AIX เวอร์ชัน 7.2

Networks และการจัดการกับการสื่อสาร

AIX เวอร์ชัน 7.2

Networks และการจัดการกับการสื่อสาร

IBM

หมายเหตุ

ก่อนที่คุณจะใช้ข้อมูลนี้และผลิตภัณฑ์ที่สนับสนุน โปรดอ่านข้อมูลใน “คำประกาศ” ในหน้า 743

เอดิชันนี้ใช้กับ AIX เวอร์ชัน 7.2 และกับรีลีสและโมดิฟิเคชันในลำดับต่อมาทั้งหมด จนกว่ามีการบ่งชี้เป็นอย่างอื่นในเอดิชันใหม่

สารบัญ

เกี่ยวกับเอกสารนี้

เอกสารนี้ให้ข้อมูลที่ครบถ้วนแก่แอ็พพลิเคชันโปรแกรมเมอร์ เกี่ยวกับการเปิดใช้งานแอ็พพลิเคชันสำหรับการทำให้เป็นโกลบอลสำหรับระบบปฏิบัติการ AIX® รวมทั้ง ให้ข้อมูลที่ครบถ้วนแก่ผู้ดูแลระบบเกี่ยวกับ การเปิดใช้งานสภาวะแวดล้อมเครือข่ายสำหรับการทำให้เป็นโกลบอลสำหรับระบบปฏิบัติการ AIX โปรแกรมเมอร์ และผู้ดูแลระบบสามารถใช้เอกสารนี้เพื่อหาความรู้ เกี่ยวกับแนวทางการทำให้เป็นโกลบอล และหลักการ หัวข้อจะรวมถึง โลแคล ชุดของโค้ด วิธีอินพุต รูทีนย่อย ตัวแปลง การแม็พอักขระ ข้อมูลที่ระบุสำหรับวัฒนธรรม และสิ่งอำนวยความสะดวกให้กับข้อความ

การไฮไลต์

ระเบียบการไฮไลต์ต่อไปนี้ถูกใช้ในเอกสารนี้:

ไอเท็ม	คำอธิบาย
ตัวหนา	ระบุคำสั่ง รูทีนย่อย คีย์เวิร์ด ไฟล์ โครงสร้าง ไดเร็กทอรี และไอเท็มอื่นที่เป็นเจ้าของชื่อ ที่กำหนดไว้ล่วงหน้าโดยระบบ และยังระบุอ็อบเจ็กต์รูปภาพ เช่น ปุ่ม เลเบล และไอคอนที่ผู้ใช้เลือก
ตัวเอน	ระบุพารามิเตอร์ที่ชื่อแท้จริง หรือค่าที่ถูกกำหนด โดยผู้ใช้
`Monospace`	ระบุตัวอย่างของค่าข้อมูลเฉพาะ ตัวอย่างของข้อความที่คล้ายกับที่คุณจะเห็นแสดงขึ้น ตัวอย่างของส่วนของโปรแกรม โค้ดที่คล้ายกับที่คุณอาจบันทึกในฐานะโปรแกรมเมอร์ ข้อความจากระบบ หรือข้อมูลที่คุณควรจะพิมพ์จริง

การตรงตามตัวพิมพ์ใน AIX

ทุกสิ่งในระบบปฏิบัติการ AIX เป็นแบบตรงตาม ตัวพิมพ์ ซึ่งหมายความว่ามีการแยกแยะความแตกต่างระหว่างตัวอักษรพิมพ์ใหญ่ และพิมพ์เล็ก ตัวอย่างเช่น คุณสามารถใช้คำสั่ง **ls** เพื่อ แสดงรายชื่อไฟล์ ถ้าคุณพิมพ์ `LS` ระบบจะตอบกลับว่า `is not found` คำสั่ง ในลักษณะคล้ายกัน **FILEA**, **FiLea**, และ **filea** คือชื่อไฟล์ที่แตกต่างกันสามไฟล์ แม้ว่า จะอยู่ในไดเร็กทอรีเดียวกันก็ตาม เพื่อหลีกเลี่ยงการทำการดำเนินการที่ไม่ต้องการ ตรวจสอบให้แน่ใจว่าคุณใช้ตัวพิมพ์ที่ถูกต้องเสมอ

ISO 9000

ระบบรับรองคุณภาพที่ลงทะเบียน ISO 9000 ใช้ในการพัฒนาและการผลิตผลิตภัณฑ์นี้

การจัดการเครือข่ายและการสื่อสาร

ทั้งผู้ดูแลระบบและผู้ใช้ทำภารกิจการสื่อสารผ่านทางเครือข่าย หลายรูปแบบ ผู้ดูแลระบบสามารถค้นหาข้อมูล ในหัวข้อนี้เกี่ยวกับวิธีการทำภารกิจ เช่น การตั้งค่าคอนฟิกค่าติดตั้ง TCP/IP การพัฒนาความปลอดภัยของเครือข่าย และการมอนิเตอร์ระบบของคุณ ผู้ใช้สามารถ ค้นหาข้อมูลที่สมบูรณ์เกี่ยวกับวิธีการทำภารกิจ เช่น การใช้แอ็พพลิเคชัน การสื่อสารและบริการสำหรับระบบปฏิบัติการ ข้อมูลอื่น อธิบายการกำหนดคอนฟิก และการแก้ไขปัญหา Mail, Message Handler (MH), Network File System (NFS), High Availability-NFS (HA-NFS), Transmission Control Protocol/Internet Protocol (TCP/IP), Basic Networking Utilities (BNU), อุปกรณ์การสื่อสารแบบซีเรียล และ TTY, Asynchronous Terminal Emulation (ATE) และ Simple Network Management Protocol (SNMP) ข้อมูล เกี่ยวกับการรับและการส่งเมลและข้อความ การถ่ายโอนไฟล์ (คำสั่ง **ftp**) การพิมพ์ไฟล์จากหรือไปยังระบบรีโมต การรันคำสั่งบนระบบอื่น การสื่อสารระหว่างระบบโลคัลและรีโมต และรวมถึงการปรับแต่งสภาวะแวดล้อมการสื่อสาร หัวข้อนี้ ยังมีอยู่บนแผ่นซีดีเอกสารคู่มือที่จัดส่งมาพร้อมกับ ระบบปฏิบัติการ

มีอะไรใหม่ในการจัดการเครือข่ายและการสื่อสาร

อ่านเกี่ยวกับข้อมูลใหม่หรือข้อมูลสำคัญที่มีการเปลี่ยนแปลงสำหรับ คอลเล็กชันหัวข้อการจัดการเครือข่ายและการสื่อสาร

วิธีการดู มีอะไรใหม่หรือมีอะไรที่เปลี่ยนแปลง

ในไฟล์ PDF นี้ คุณอาจเห็นแถบการแก้ไข (|) ในขอบด้านซ้ายเพื่อระบุข้อมูลใหม่ และที่เปลี่ยนแปลง

ธันวาคม 2015

ข้อมูลต่อไปนี้เป็นการสรุปของอัพเดตที่ทำไว้กับชุดของหัวข้อนี้:

- ข้อมูลเพิ่มเติมเกี่ยวกับส่วนสนับสนุน RDMA บน Converged Ethernet (RoCE) บน RDSv3 ในหัวข้อ “Reliable Datagram Sockets บน InfiniBand และ RoCE” ในหน้า 160
- ข้อมูลเพิ่มเติมเกี่ยวกับอินเตอร์เฟส loopback ในหัวข้อ “ข้อควรพิจารณาเกี่ยวกับผู้เริ่มต้นซอฟต์แวร์ iSCSI” ในหน้า 424 และ “ข้อควรพิจารณาเกี่ยวกับตัวเริ่มต้นซอฟต์แวร์ iSCSI” ในหน้า 426

การสื่อสารและเน็ตเวิร์ก

เข้าใจถึงหลักการทั่วไปของคอมพิวเตอร์เน็ตเวิร์กเป็นกรอบความคิดพื้นฐาน ผู้ดูแลระบบที่ไม่คุ้นเคยกับหลักการทั่วไปของเน็ตเวิร์กจำเป็นต้องอ่านหัวข้อนี้ ผู้ที่คุ้นเคยกับเน็ตเวิร์กแบบ UNIXสามารถข้ามหัวข้อนี้ได้

เน็ตเวิร์กเป็นการรวมกันของคอมพิวเตอร์สองเครื่องหรือมากกว่าและลิงก์ที่เชื่อมถึงกัน เน็ตเวิร์กแบบ *ฟิสิคัล*เป็นฮาร์ดแวร์ (อุปกรณ์ เช่น อะแด็ปเตอร์การ์ด สายเคเบิล และสายโทรศัพท์) ที่ประกอบกันเป็นเน็ตเวิร์ก ซอฟต์แวร์และโมเดลของกรอบความคิดประกอบเป็นเน็ตเวิร์กแบบ *โลจิคัล* เน็ตเวิร์กชนิดต่างๆและอีมูเลเตอร์จัดเตรียมการทำงานแบบต่างๆ

การสื่อสาร

เน็ตเวิร์กจะยอมให้ผู้ใช้จำนวนมากและฟังก์ชันการสื่อสารของแอ็พพลิเคชัน

ตัวอย่างเช่น มันให้ผู้ใช้ทำสิ่งต่อไปนี้ :

- ส่งจดหมายอิเล็คทรกนิคส์ (อีเมล)
- อีมูเลตเทอร์มินัลอื่น หรือล็อกอินคอมพิวเตอร์อื่น
- ถ่ายโอนข้อมูล
- รันโปรแกรมที่อยู่บนโหนดแบบรีโมต

One of the most popular applications for computer networks is email, which allows a user to send a message to another user. ผู้ใช้สองคนอาจอยู่บนระบบเดียวกัน (ในกรณีนี้ไม่ต้องการเน็ตเวิร์กการสื่อสาร) ระบบที่ต่างกันที่อยู่คนละตึก หรือแม้แต่คนละประเทศ เลเยอร์ที่จำเป็นของซอฟต์แวร์ และ ฮาร์ดแวร์รวมถึงเน็ตเวิร์กแบบฟิสิคัล ทำให้ผู้ใช้สามารถ สร้าง ส่ง รับ และประมวลผลข้อความ จดหมาย บันทึก คำเชิญ และไฟล์ข้อมูล การสื่อสารเหล่านี้สามารถสามารถไปยังหรือมาจากผู้ใช้อื่นที่อยู่บนเน็ตเวิร์กแบบฟิสิคัล จดหมายอิเล็คทรอนิคส์มีความสามารถสำหรับการทำคำอธิบายประกอบข้อความ เรียงลำดับข้อความ การแพ็กข้อความ การเรียงวันที่ และการจัดการโฟลเดอร์ของเมล

โดยผ่านเน็ตเวิร์กการสื่อสาร คอมพิวเตอร์หนึ่งสามารถการ *อีมูเลต* หรือเลียนแบบคอมพิวเตอร์อื่น และเข้าถึงข้อมูลเหมือนกับว่าเป็นคอมพิวเตอร์หรือเทอร์มินัลชนิดอื่น ความสามารถในการล็อกอินแบบรีโมต ทำให้ผู้ใช้สามารถใช้อินเตอร์เฟสบรรทัดรับคำสั่งแบบโต้ตอบเพื่อล็อกอินระบบรีโมต และเข้าถึงไฟล์โปรแกรมและไฟล์เดียวกัน เหมือนกับใช้เครื่องแบบโลคัล

เน็ตเวิร์กยังใช้สำหรับการถ่ายโอนข้อมูลจากระบบหนึ่งไปยังอีกระบบ ไฟล์ไดเร็กทอรี และระบบไฟล์ทั้งหมดสามารถถูกโอนย้ายจากเครื่องหนึ่งไปยังระบบอื่นข้ามเน็ตเวิร์ก ทำให้สามารถทำการแบ็กอัพข้อมูลแบบรีโมต หร้อมกับมีการทำระบบซ้ำในการที่เครื่องล้มเหลว การป้องกันด้วยรหัสผ่านยังถูกจัดเตรียมเป็นส่วนหนึ่งของโปรโตคอล โดยใช้การถ่ายโอนไฟล์ จะมีความสัมพันธ์แบบไคลเอ็นต์/เซิร์ฟเวอร์ระหว่างผู้ใช้ที่ร้องขอและระบบรีโมตที่ผู้ใช้เข้าถึง บ่อยครั้งที่โปรโตคอลการถ่ายโอนไฟล์จะรวมฟังก์ชันสำหรับแสดงและควบคุมเพื่อที่ผู้ใช้ที่มีสิทธิอ่าน/เขียนสามารถแสดง กำหนด หรือลบไฟล์และไดเร็กทอรี

หลายๆโปรโตคอลที่มีอยู่จะทำให้ผู้ใช้และแอ็พพลิเคชันบนระบบหนึ่งสามารถใช้โพรซีเดอร์และแอ็พพลิเคชันบนระบบอื่น นี่จะเป็นประโยชน์สำหรับหลายสภาวะแวดล้อม รวมถึงการลดโหลดของรูทีนที่ใช้คอมพิวเตอร์มากในแอ็พพลิเคชันทางวิศวกรรมหรือวิทยาศาสตร์

เครือข่าย

ความซับซ้อนของคอมพิวเตอร์เน็ตเวิร์กสมัยใหม่ทำให้เกิดหลายการจำลองเชิงแนวคิดสำหรับอธิบายวิธีที่เน็ตเวิร์กทำงาน

หนึ่งในโมเดลที่ใช้กันทั่วไปคือโมเดลการอ้างอิง International Standards Organization's Open Systems Interconnection (OSI) ยังถูกอ้างอิงเป็นโมเดล OSI 7 เลเยอร์

เจ็ดเลเยอร์ของ OSI โมเดลจะมีหมายเลขดังต่อไปนี้ :

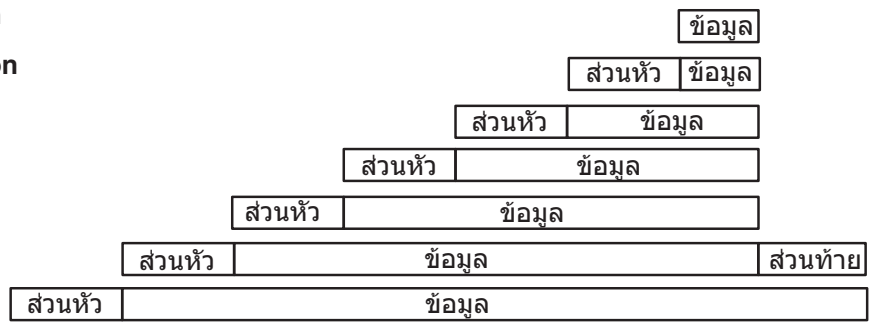
ไอเท็ม	คำอธิบาย
7	แอ็พพลิเคชัน
6	พรีเซนเตชัน
5	Session
4	Transport
3	เครือข่าย
2	Data Link
1	แบบฟิสิคัล

ระดับที่ 1 ถึง 3 ระบุถึงเน็ตเวิร์ก และแตกต่างกันโดยขึ้นอยู่กับฟิสิคัลเน็ตเวิร์กที่คุณใช้ ระดับที่ 4 ถึง 7 ประกอบด้วยส่วนที่เป็นอิศระจากเน็ตเวิร์ก ซึ่งเป็นฟังก์ชันระดับสูงกว่า แต่ละเลเยอร์จะอธิบายแต่ละฟังก์ชัน (แทนที่จะเป็นโปรโตคอลที่ระบุ) ที่ปรากฏในการสื่อสารข้อมูล ฟังก์ชัน 7 เลเยอร์จากระดับล่างสุด (ระดับเครื่อง) ถึงระดับสูงสุด (ระดับที่คนเข้ามาเกี่ยวข้อง) ดังต่อไปนี้ :

ไอเท็ม	คำอธิบาย
แอ็พพลิเคชัน	ประกอบด้วยแอ็พพลิเคชันที่ใช้เน็ตเวิร์ก
พรีเซนเตชัน	ต้องแน่ใจว่าข้อมูลที่แสดงกับแอ็พพลิเคชันมีความถูกต้อง
Session	จัดการดารเชื่อมต่อระหว่างแอ็พพลิเคชัน
Transport	ทำให้แน่ใจว้เป็นการส่งข้อมูลที่ไม่มีข้อผิดพลาด
เครือข่าย	จัดการการเชื่อมต่อไปยังเครื่องอื่นในเน็ตเวิร์ก
Data Link	จัดเตรียมความน่าเชื่อถือในการนำส่งข้อมูลข้ามฟิสิคัลเลเยอร์ (ซึ่งในความเป็นจริงแล้วเชื่อถือไม่ได้)
แบบฟิสิคัล	อธิบายสื่อแบบฟิสิคัลของเน็ตเวิร์ก ตัวอย่างเช่น สายเส้นใยนำแสงที่ต้องการสำหรับ Fiber Distributed Data Interface (FDDI) เน็ตเวิร์ก เป็นส่วนของฟิสิคัลเลเยอร์

หมายเหตุ: ในขณะที่โมเดลการอ้างอิง OSI มีประโยชน์สำหรับการอธิบายแนวความคิดของเน็ตเวิร์ก หลายๆเน็ตเวิร์กโปรโตคอลไม่ได้เป็นไปตามโมเดล OSI ตัวอย่างเช่น เมื่ออธิบายถึง Transmission Control Protocol/Internet Protocol (TCP/IP) ฟังก์ชันของ Application และ Presentation เลเยอร์จะถูกรวมกัน เหมือนกับ Session และ Transport เลเยอร์ และ Data Link และ Physical เลเยอร์

แต่ละเลเยอร์ในโมเดล OSI สื่อสารกับเลเยอร์ที่เกี่ยวข้องบนเครื่องรีโมตดังแสดงในรูปโมเดลเลเยอร์ OSI



รูปที่ 1. โมเดลการอ้างอิง OSI

นี่จะแสดงระดับการสื่อสารต่างๆของโมเดล OSI ดังอธิบายข้างต้น

เลเยอร์จะผ่านข้อมูลไปยังเลเยอร์ที่อยู่ติดกันข้างบนและล่างเท่านั้น แต่ละเลเยอร์เพิ่มข้อมูลส่วนหัวของมัน (และข้อมูลส่วนท้าย ในกรณีของ Data Link) encapsulate ข้อมูลที่ได้รับจากเลเยอร์ที่สูงกว่าอย่างมีประสิทธิภาพ

ผู้ใช้แต่ละคนพร้อมด้วยองค์กรใช้เน็ตเวิร์กด้วยหลายสาเหตุ รวมถึง :

- การบันทึกข้อมูล
- การเคียวรีข้อมูล
- รีโมตแบตช์ entry
- การแบ่งใช้รีซอร์ส
- การแบ่งใช้ข้อมูล
- จดหมายอิเล็กทรอนิก

การบันทึกข้อมูลประกอบด้วยการใส่ข้อมูลโดยตรงกับไฟล์ข้อมูลแบบโลคัลหรือรีโมต ความเที่ยงตรงและประสิทธิภาพที่เพิ่มขึ้นเป็นธรรมชาติของผลิตภัณฑ์ของการถ่ายโอนข้อมูลขั้นตอนเดียว การเคียวรีข้อมูลนำมาซึ่งการต้นหาไฟล์ข้อมูลสำหรับข้อมูลที่ระบุ การอัพเดตข้อมูลเกี่ยวข้องกับการเปลี่ยน เพิ่ม หรือลบข้อมูลที่ถูกเก็บในโลคัลหรือรีโมตไฟล์ รีโมตแบตช์ entry ประกอบด้วยการใส่แบตช์ของข้อมูลจากตำแหน่งรีโมต โดยมักถูกทำตอนกลางคืนหรือระหว่างช่วงเวลาที่มีการใช้ระบบน้อย เนื่องจากความสามารถนั้น การสื่อสารและเน็ตเวิร์กถึงต้องการและมีความจำเป็น

การแบ่งใช้รีซอร์สเป็นอีกฟังก์ชันหนึ่งของเน็ตเวิร์ก ผู้ใช้สามารถแบ่งใช้ข้อมูลรวมถึงโปรแกรม พื้นที่การเก็บไฟล์ และอุปกรณ์เสริม เช่น เครื่องพิมพ์ โมเด็ม เทอร์มินัล และฮาร์ดดิสก์ การแบ่งใช้รีซอร์สของระบบเป็นการลงทุนที่มีประสิทธิภาพ เนื่องจากช่วยกำจัดปัญหาของการมีโปรแกรมหลายชุดและเก็บข้อมูลมีความถูกต้อง (ในกรณีของการแบ่งใช้โปรแกรมและไฟล์)

ฟิสิคัลเน็ตเวิร์ก

ฟิสิคัลเน็ตเวิร์กประกอบด้วยสายเคเบิล (โคแอ็กเชียลเคเบิล คู่แบบ twisted เส้นใยนำแสง และสายโทรศัพท์) ที่เชื่อมโยงกับฮาร์ดแวร์ที่แตกต่างกัน ซึ่งวางอยู่บนเน็ตเวิร์ก อะแด็ปเตอร์การดที่ใช้บนคอมพิวเตอร์ซึ่งเชื่อมต่อกับเน็ตเวิร์ก (โฮสต์) และตัวเชื่อมต่อ ตัวทำซ้ำ เราเตอร์ หรือบริดจ์ ที่ใช้ในเน็ตเวิร์ก

ฟิสิคัลเน็ตเวิร์กเปลี่ยนแปลงทั้งขนาดและชนิดของฮาร์ดแวร์ที่ใช้ มีชนิดของเน็ตเวิร์ก อยู่สองชนิดคือ *local area networks* (LANs) และ *wide area networks* (WANs) LAN คือเน็ตเวิร์กที่การสื่อสารถูกจำกัด พื้นที่กราฟิกที่มีขนาดกลางประมาณ 1 ถึง 10 กม. (1 ถึง 6 ไมล์) เช่น ตึกสำนักงานเดี่ยว คลังสินค้า หรือแคมปัส WAN คือเน็ตเวิร์ก ที่จัดเตรียมความสามารถในการสื่อสารกับข้อมูลตลอดทั้งพื้นที่ภูมิภาคที่ใหญ่กว่า การให้บริการแบบ LAN เช่น ระหว่างประเทศหรือระหว่างทวีป คลาสระดับกลางของ เน็ตเวิร์กที่มีอยู่ยังเรียกว่า *metropolitan area networks* (MANs) คำแนะนำนี้ไม่ได้แบ่งแยก MAN ซึ่งถูกจัดกลุ่มด้วย WAN

LAN ใช้ Standard Ethernet, IEEE 802.3 Ethernet หรือฮาร์ดแวร์โทเค็นริง สำหรับฟิสิคัลเน็ตเวิร์ก ขณะที่ WAN และเน็ตเวิร์กแบบอะซิงโครนัส ใช้เน็ตเวิร์การสื่อสารที่จัดเตรียมไว้โดยบริษัทคลื่นพาหะทั่วไป การดำเนินการของ ฟิสิคัลเน็ตเวิร์กในทั้งสองกรณีถูกควบคุมโดยมาตรฐานการวางเน็ตเวิร์ก จากองค์กร เช่น Electronics Industry Association (EIA) หรือ International Telecommunication Union (ITU)

ระบบเน็ตเวิร์ก

การสื่อสารเน็ตเวิร์กทั้งหมดเกี่ยวข้องกับการใช้ฮาร์ดแวร์และซอฟต์แวร์ การสนับสนุนระบบการสื่อสารฮาร์ดแวร์และซอฟต์แวร์ถูกกำหนดโดยฮาร์ดแวร์ที่ถูกใช้และ ซอฟต์แวร์ที่จำเป็นเพื่อรันฮาร์ดแวร์นั้นและอินเตอร์เฟสกับเน็ตเวิร์ก

ฮาร์ดแวร์ ประกอบด้วยอุปกรณ์ฟิสิคัลที่เชื่อมต่อกับฟิสิคัลเน็ตเวิร์ก *ซอฟต์แวร์* ประกอบด้วยโปรแกรมและไดรเวอร์อุปกรณ์ที่เกี่ยวข้องกับการทำงานของระบบนั้นๆ ฮาร์ดแวร์ระบบประกอบด้วยอะแด็ปเตอร์การ์ดหรืออุปกรณ์อื่นที่ให้พาธหรืออินเตอร์เฟสระหว่างซอฟต์แวร์ระบบและฟิสิคัลเน็ตเวิร์ก อะแด็ปเตอร์การ์ดต้องการ อินพุต/เอาต์พุต (I/O) การ์ดสล็อตใน

ระบบ อะแด็ปเตอร์การ์ดเชื่อมต่อ *data terminal equipment* (DTE) กับ *data circuit-terminating equipment* (DCE) ซึ่งมันจะให้การกำหนดโลคัลแอดเดรสแบบฟิสิคัลให้กับพอร์ตของ DTE อุปกรณ์อื่น เช่นโมเด็ม สามารถเชื่อมกับหนึ่งในพอร์ตมาตรฐานบนคอมพิวเตอร์

อะแด็ปเตอร์การ์ดเตรียมข้อมูลขาเข้าและขาออกทั้งหมด ทำการค้นหาแอดเดรส จัดเตรียมไดรเวอร์ ตัวรับและป้องกันไฟกระชาก สนับสนุนอินเตอร์เฟสที่ต่างกัน และและโดยทั่วไปจะลดงานการสื่อสารของโพรเซสเซอร์ระบบ อะแด็ปเตอร์การ์ดสนับสนุนมาตรฐานที่ต้องการโดยฟิสิคัลเน็ตเวิร์ก (ตัวอย่างเช่น EIA 232D, Smartmodem, V.25 bis, EIA 422A, X.21 หรือ V.35) และในเวลาเดียวกันอาจสนับสนุนซอร์ฟแวร์ *โปรโตคอล* ตัวอย่างเช่น synchronous data link control (SDLC), high-level data link control (HDLC) และฟิสิคัลปรโตคอล bisynchronous ถ้าอะแด็ปเตอร์ไม่ได้ประกอบด้วยซอฟต์แวร์ที่สนับสนุน ดังนั้นการสนับสนุนนี้ต้องถูกจัดเตรียมโดยไดรเวอร์อุปกรณ์ของอะแด็ปเตอร์

โปรโตคอล

ซอฟต์แวร์การสื่อสารทั้งหมดใช้ *โปรโตคอล* ซึ่งเป็นชุดของกฎเกี่ยวกับ ความหมายและไวยากรณ์ที่กำหนดลักษณะการทำงานของยูนิตการทำงานเพื่อให้ สามารถสื่อสารได้

โปรโตคอลกำหนดวิธีการจัดส่งข้อมูล วิธีการแนบเพื่อให้ไปถึงปลายทางอย่าง ปลอดภัย และพาธที่จะใช้ โปรโตคอลยังจัดการกับ โฟลว์ของข้อความและการยอมรับด้วย

โปรโตคอลมีอยู่ที่ระดับที่แตกต่างกันภายในเคอร์นัลและไม่สามารถจัดการ โดยตรง อย่างไรก็ตาม โปรโตคอลมีการจัดการทางอ้อมโดยสิ่งที่ผู้ใช้เลือก ที่จะทำที่ระดับ application programming interface (API) ตัวเลือกที่ผู้ใช้เลือก เมื่อเรียกใช้การโอนย้ายไฟล์ รีโมตล็อกอิน หรือโปรแกรมการเลียนแบบเทอร์มินัล กำหนดโปรโตคอลที่ใช้ในการดำเนินการของโปรแกรมเหล่านั้น

แอดเดรส

แอดเดรส มีความเชื่อมโยงกับทั้งซอฟต์แวร์และฮาร์ดแวร์ แอดเดรสเป็นสื่อกลางซึ่งสเตชันการส่งหรือควบคุมใช้ในการเลือกสเตชันที่จะส่งข้อมูลไป

แอดเดรสระบุที่ตั้งค่าการรับหรือหน่วยเก็บ ฟิสิคัลแอดเดรสคือ โค้ดที่ไม่ซ้ำกันซึ่งกำหนดให้กับแต่ละอุปกรณ์หรือเวิร์กสเตชันที่เชื่อมต่อกับเครือข่าย

ตัวอย่างเช่น บนเครือข่าย token-ring คำสั่ง **netstat -iv** แสดง แอดเดรสของ token-ring การ์ด นี่เป็นแอดเดรสของเครือข่ายฟิสิคัล คำสั่ง **netstat -iv** ยังแสดงข้อมูลแอดเดรสระดับคลาส และระดับผู้ใช้ด้วย แอดเดรสมักถูกกำหนดโดยซอฟต์แวร์ แต่สามารถสร้างโดยผู้ใช้ได้เช่นกัน

โดเมน

ลักษณะหนึ่งของแอดเดรสที่พบได้ทั่วไปในเครือข่ายการสื่อสารจำนวนมากคือ แนวคิดของ *โดเมน* โดเมนวางรีซอร์สการประมวลผลข้อมูลไว้ใน เครือข่ายภายใต้การควบคุมทั่วไป

ตัวอย่างเช่น โครงสร้างของอินเทอร์เน็ตแสดงวิธีการที่โดเมนกำหนด แอดเดรส Internet Protocol (IP) อินเทอร์เน็ตนับเป็นเครือข่ายแบบครอบคลุมที่ประกอบด้วย เครือข่ายขนาดเล็กซึ่งแตกต่างกันจำนวนมาก เพื่อสนับสนุนการเราต์และการกำหนดแอดเดรส อินเทอร์เน็ตแอดเดรสจึงมีการจัดโครงสร้างตามลำดับชั้นในโดเมน โดยมีหมวดหมู่ ที่กว้างมากอยู่ที่ด้านบนสุด เช่น `com` สำหรับผู้ใช้เชิงพาณิชย์, `edu` สำหรับ ผู้ใช้ในภาคการศึกษา, และ `gov` สำหรับผู้ใช้ภาครัฐ

ภายในโดเมน `com` คือโดเมนขนาดเล็กจำนวนมากที่สอดคล้องกับ แต่ละธุรกิจ ตัวอย่างเช่น `ibm` ภายในโดเมน `ibm.com` ยังประกอบด้วย โดเมนขนาดเล็กลงไปอีกซึ่งสอดคล้องกับอินเทอร์เน็ตแอดเดรสของที่ตั้ง หลากหลาย เช่น `austin.ibm.com`

หรือ raleigh.ibm.com ที่ระดับนี้ เราเริ่มต้นเห็นชื่อของ *โฮสต์* โฮสต์ในคอนเท็กซ์นี้ คือคอมพิวเตอร์ใดๆ ที่เชื่อมต่อกับเครือข่าย *ภายใน* austin.ibm.com อาจมีโฮสต์ที่มีชื่อว่า hamlet และ lear ซึ่งมีแอดเดรสเป็น hamlet.austin.ibm.com และ lear.austin.ibm.com

เกตเวย์และบริดจ์

เครือข่ายจำนวนมากที่อยู่บนอินเทอร์เน็ตมักใช้ ฮาร์ดแวร์ที่แตกต่างกันและรันซอฟต์แวร์ที่แตกต่างกัน *เกตเวย์* และ *บริดจ์* ช่วยให้ เครือข่ายที่แตกต่างกันเหล่านี้สามารถสื่อสารซึ่งกันและกันได้

บริดจ์เป็นยูนิตการทำงานที่เชื่อมต่อสอง LANs ซึ่งอาจจะใช้โพรซีเดอร์ logical link control (LLC) เดียวกัน เช่น อีเทอร์เน็ต แต่โพรซีเดอร์ medium access control (MAC) ต่างกัน เกตเวย์มีช่วงที่กว้างกว่า บริดจ์ เกตเวย์ทำงานเหนือระดับลิงก์ และเมื่อต้องการ จะแปล อินเตอร์เฟสและโปรโตคอลที่ใช้โดยเครือข่ายหนึ่งเป็นอินเตอร์เฟสและโปรโตคอล ที่ใช้โดยเครือข่ายที่แตกต่างอื่น เกตเวย์ช่วยให้สามารถโอนย้ายข้อมูลผ่านเครือข่ายต่างๆ ซึ่งประกอบเป็นอินเทอร์เน็ตได้

การเราต์ข้อมูล

การใช้ชื่อโดเมนสำหรับการระบุแอดเดรสและเกตเวย์สำหรับการแปล ช่วยสนับสนุน *การเราต์* ของข้อมูลที่กำลังโอนย้ายได้เป็นอย่างมาก การเราต์ เป็นการกำหนดพาธซึ่งข้อความใช้เพื่อให้ไปถึงปลายทาง

ชื่อโดเมนกำหนดปลายทางของข้อความได้อย่างมีประสิทธิภาพ ในเครือข่ายขนาดใหญ่ เช่นอินเทอร์เน็ต ข้อมูลมีการเราต์จากเครือข่ายการสื่อสารหนึ่ง ไปยังเครือข่ายถัดไปจนกว่าข้อมูลนั้นไปถึงปลายทาง แต่ละเครือข่ายการสื่อสาร ตรวจสอบชื่อโดเมนและเราต์ข้อมูลไปยังจุดพักทางโลจิคัลถัดไปตามข้อมูลโดเมน ซึ่งเครือข่ายนั้นคุ้นเคย ในวิธีนี้ แต่ละเครือข่ายการสื่อสารที่ได้รับข้อมูลมีส่วนร่วมในโปรเซส การเราต์

โลคัลและรีโมตโหนด

เครือข่ายฟิสิคัลมีการใช้โดยโฮสต์ที่ตั้งอยู่บนเครือข่ายนั้น แต่ละโฮสต์เป็น *โหนด* บนเครือข่าย โหนดคือที่ตั้งซึ่งระบุแอดเดรสได้ในเครือข่ายการสื่อสารซึ่งนำเสนอเซอร์วิสการประมวลผลโฮสต์ การสื่อสารระหว่าง โหนดต่างๆ เหล่านี้มีการกำหนดเป็นแบบ *โลคัล* หรือ *รีโมต*

โลคัล เกี่ยวข้องกับอุปกรณ์ไฟล์ หรือระบบที่เข้าถึงโดยตรงจาก ระบบของคุณ โดยไม่ได้ใช้สายการสื่อสาร *รีโมต* เกี่ยวข้องกับอุปกรณ์ไฟล์ หรือระบบที่เข้าถึงโดยระบบของคุณผ่านทางสาย การสื่อสาร โลคัลไฟล์ตั้งอยู่บนระบบของคุณ ในขณะที่รีโมตไฟล์ตั้งอยู่บนไฟล์เซิร์ฟเวอร์ หรือที่โหนดอื่นซึ่งคุณสื่อสารโดยใช้เครือข่ายฟิสิคัล ตัวอย่างเช่น อีเทอร์เน็ต, token-ring, หรือสายโทรศัพท์

ไคลเอ็นต์และเซิร์ฟเวอร์

เซิร์ฟเวอร์ คือคอมพิวเตอร์ที่มีข้อมูล หรือนำเสนอสิ่งอำนวยความสะดวก ที่เข้าถึงโดยคอมพิวเตอร์เครื่องอื่นบนเครือข่าย *ไคลเอ็นต์* คือคอมพิวเตอร์ ที่ร้องขอเซอร์วิสหรือข้อมูลจากเซิร์ฟเวอร์

ชนิดเซิร์ฟเวอร์ที่พบได้ทั่วไปคือ ไฟล์เซิร์ฟเวอร์ซึ่งจัดเก็บไฟล์ เนมเซิร์ฟเวอร์ซึ่งจัดเก็บชื่อและแอดเดรส แอ็พพลิเคชันเซิร์ฟเวอร์ซึ่งจัดเก็บ โปรแกรมและแอ็พพลิเคชัน และเซิร์ฟเวอร์การพิมพ์ซึ่งจัดตารางเวลาและกำหนด ทิศทางงานพิมพ์ไปยังปลายทาง

ไคลเอ็นต์สามารถร้องขอโค้ดโปรแกรมที่อัพเดต หรือการใช้แอ็พพลิเคชัน ได้จากโค้ดเซิร์ฟเวอร์ เพื่อให้ได้ชื่อหรือแอดเดรส ไคลเอ็นต์จะติดต่อ เนมเซิร์ฟเวอร์ ไคลเอ็นต์ยังสามารถร้องขอไฟล์และข้อมูลสำหรับรายการข้อมูล การสอบถาม หรือเร็กคอร์ดที่อัพเดตได้จากไฟล์เซิร์ฟเวอร์

การสื่อสารกับระบบปฏิบัติการอื่นๆ

ชนิดที่แตกต่างกันของคอมพิวเตอรีสามารถเชื่อมต่อบนเน็ตเวิร์ก คอมพิวเตอร์ สามารถมาจากผู้ผลิตที่แตกต่างกันหรือเป็นโมเดลที่แตกต่างจาก ผู้ผลิตรายเดียวกัน โปรแกรมการสื่อสารสร้างการเชื่อมต่อที่แตกต่างกันในระบบปฏิบัติการ ของคอมพิวเตอร์ตั้งแต่สองชนิดขึ้นไป

ในบางครั้ง โปรแกรมเหล่านี้ต้องการให้โปรแกรมอื่นๆ ติดตั้ง อยู่บนเน็ตเวิร์ก โปรแกรมอื่นๆ อาจต้องการให้โปรโตคอลที่มีภาวะเชื่อมต่อการสื่อสาร เป็น TCP/IP หรือ Systems Network Architecture (SNA) ที่มีอยู่บนเน็ตเวิร์ก

แอ็พพลิเคชันโฮสต์อีมูเลชัน

อีมูเลเตอร์ เป็นซอฟต์แวร์แอ็พพลิเคชันทำให้ระบบคุณเพื่อทำหน้าที่เหมือนกับคุณใช้เทอร์มินัลหรือเครื่องพิมพ์อื่น

เทอร์มินัลอีมูเลเตอร์ จะเชื่อมต่อกับระบบโฮสต์เพื่อเข้าถึงข้อมูลหรือแอ็พพลิเคชัน เทอร์มินัลอีมูเลเตอร์บางตัวจะอำนวยความสะดวกในการถ่ายโอนข้อมูลไปยังหรือจากโฮสต์ ตัวอื่นให้ application programming interface (API) เพื่ออนุญาตการสื่อสารแบบโปรแกรมถึงโปรแกรม และงานของโฮสต์แบบอัตโนมัติ

อีมูเลเตอร์เครื่องพิมพ์ อนุญาตให้โฮสต์สามารถพิมพ์ไฟล์บนเครื่องพิมพ์แบบโลคัล หรือเก็บมันในรูปแบบที่สามารถพิมพ์ได้เพื่อจะถูกพิมพ์หรือแก้ไขภายหลัง

หลายๆแอ็พพลิเคชันจะมีให้ใช้เพื่อให้ระบบของคุณสามารถอีมูเลตเทอร์มินัลชนิดอื่น หัวข้อนี้ให้ข้อมูลเกี่ยวกับเทอร์มินัลอีมูเลเตอร์หรืออีมูเลเตอร์เครื่องพิมพ์

หมายเหตุ : The **bterm**ี คำสั่งอีมูเลตเทอร์มินัลในโหมดสองทิศ (BIDI)

คำสั่ง TCP/IP สำหรับอีมูเลชัน

ซอฟต์แวร์ Transmission Control Protocol/Internet Protocol (TCP/IP) มีคำสั่ง **telnet** และ **rlogin** ซึ่งทำให้คุณสามารถเชื่อมต่อและเข้าถึงระบบ TCP/IP แบบรีโมต

ไอเท็ม	คำอธิบาย
telnet	ให้คุณสามารถล็อกอินเข้าสู่รีโมตโฮสต์โดยใช้โปรโตคอล **TELNET** มันจะแตกต่างกับคำสั่ง **rlogin** ที่มันเป็นคำสั่งแบบ trusted คำสั่งแบบ *trusted* เป็นคำสั่งที่เข้ากับระดับของความปลอดภัยที่ถูกตั้งบนคอมพิวเตอร์ของคุณ ระบบที่ต้องการความปลอดภัยเป็นพิเศษควรยอมให้ใช้เฉพาะคำสั่งแบบ trusted มาตรฐานสำหรับคำสั่งแบบ trusted กระบวนการ และโปรแกรมถูกตั้งไว้และดูแลโดย U.S. Department of Defense
tn	ทำหน้าที่เดียวกับคำสั่ง **telnet**
rlogin	ให้ผู้ใช้สามารถล็อกอินเข้ากับรีโมตโฮสต์ มันจะแตกต่างจากคำสั่ง **telnet** ที่มันเป็นคำสั่งแบบ *nontrusted* และสามารถถูกปิดการใช้งานถ้าระบบของคุณต้องการความปลอดภัยเป็นพิเศษ

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ **TCP/IP** ดูที่ “Transmission Control Protocol/Internet Protocol” ในหน้า 111

คำสั่ง BNU สำหรับอีมูเลชัน

ซอฟต์แวร์ Basic Networking Utilities (BNU) จะรวมคำสั่ง **ct, cu** และ **tip** ซึ่งอนุญาตให้คุณเชื่อมต่อกับระบบรีโมตที่ใช้ระบบปฏิบัติการ AIX

ไอเท็ม	คำอธิบาย
ct	ทำให้ผู้ใช้บนรีโมตเทอร์มินัล เช่น 3161 สื่อสารกับเทอร์มินัลอื่นผ่านสายโทรศัพท์ จากนั้นผู้ใช้บนรีโมตเทอร์มินัลสามารถล็อกอินและทำงานบนเทอร์มินัลอื่น คำสั่ง **ct** จะเหมือนกับคำสั่ง **cu** แต่ไม่มีความยืดหยุ่นเท่า ตัวอย่างเช่น ผู้ใช้ไม่สามารถเรียกใช้คำสั่งบนระบบโลคัลขณะที่เชื่อมต่อ กับระบบรีโมตผ่านคำสั่ง **ct** อย่างไรก็ตาม คุณสามารถบอกให้คำสั่ง **ct** หมุนโทรศัพท์ต่อไปจนกว่าจะทำการเชื่อมต่อสำเร็จ หรือระบุหมายเลขโทรศัพท์มากกว่าหนึ่งหมายเลขในเวลาเดียวกัน
cu	เชื่อมต่อเทอร์มินัลของคุณกับเทอร์มินัลอื่นที่เชื่อมต่อกับระบบ UNIX หรือที่ไม่ใช่ UNIX เมื่อสร้างการเชื่อมต่อขึ้นแล้ว ผู้ใช้สามารถล็อกอิน ทั้งสองระบบพร้อมกัน ทำการเรียกใช้คำสั่งบนระบบใดระบบหนึ่งโดยไม่ทำให้ต้องดร็อปลิงก์การเชื่อมต่อ BNU ถ้ารีโมตคอมพิวเตอร์กำลังรัน ภายใต้ UNIX เช่นกัน ผู้ใช้จะสามารถถ่ายโอนไฟล์ ASCII ระหว่างระบบทั้งสองได้ คุณสามารถใช้คำสั่ง **cu** เพื่อเชื่อมต่อกับหลายระบบ และคำสั่งสามารถถูกเรียกใช้บนระบบที่เชื่อมต่อใดๆ
tip	เชื่อมต่อเทอร์มินัลของคุณกับรีโมตเทอร์มินัลและให้คุณสามารถทำงานบนรีโมตเทอร์มินัล เหมือนกับล็อกอินโดยตรง คุณสามารถใช้คำสั่ง **tip** เพื่อถ่ายโอนไฟล์ไปยังหรือจากระบบรีโมต คุณสามารถใช้สคริปต์เพื่อบันทึกการสนทนาที่คุณมีกับคำสั่ง **tip** **หมายเหตุ :** คุณต้องล็อกอินบนระบบรีโมตเพื่อใช้คำสั่ง **tip**

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ BNU ดูที่ “Basic Networking Utilities” ในหน้า 461

Asynchronous Terminal Emulation

โปรแกรม Asynchronous Terminal Emulation (ATE) ทำให้เทอร์มินัลของคุณเชื่อมต่อกับระบบส่วนใหญ่ที่สนับสนุนเทอร์มินัลแบบอะซิงโครนัส รวมถึงระบบใดๆที่สนับสนุนการเชื่อมต่อ RS-232C หรือ RS-422A

ATE ยอมให้ระบบรีโมตสื่อสารกับเทอร์มินัลของคุณเป็นหน้าจอแลลอะซิงโครนัส หรือเป็นเทอร์มินัล DEC VT100

ATE ให้ผู้ใช้สามารถรันคำสั่งบนระบบรีโมต ส่งและรับไฟล์ และตรวจสอบความถูกต้องของข้อมูลในไฟล์ที่ถูกถ่ายโอนระหว่างระบบ คุณยังสามารถดักจับไฟล์เพื่อบันทึก หรือ*ดักจับ* ข้อมูลที่เข้ามาจากระบบรีโมต ATE เป็นแบบใช้เมนูและใช้คำสั่งย่อย

เมื่อถูกติดตั้ง ATE จะสามารถเข้าถึงได้เฉพาะผู้ใช้ที่ลงทะเบียนเป็นสมาชิกของกลุ่ม UUCP โดยผู้ใช้ที่มีสิทธิ root

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ ATE ดูที่ “Asynchronous Terminal Emulation” ในหน้า 677

การสื่อสารของคำสั่งของระบบ

นี่จะอธิบายคำสั่งที่มีสำหรับแสดงข้อมูลที่ระบุผู้ใช้บนระบบของคุณ ระบบที่คุณใช้ และผู้ใช้ที่ล็อกอินกับระบบอื่น

โปรดดูหัวข้อต่อไปนี้สำหรับคำสั่งต่างๆ ที่ใช้เพื่อให้ ข้อมูลระบบและผู้ใช้

การแสดงชื่อล็อกอินของคุณ

ใช้คำสั่ง **whoami** เพื่อดูชื่อล็อกอินของคุณ

เพื่อแสดงชื่อของผู้ใช้ปัจจุบัน ใส่:

```
whoami
```

หน้าจอเหมือนดังต่อไปนี้จะถูกแสดง :

```
denise
```

ในตัวอย่างนี้ ชื่อล็อกอินคือ `denise`

การแสดงชื่อระบบของคุณ

ใช้คำสั่ง **uname** เพื่อระบุชื่อระบบของคุณ

1. เมื่อต้องการแสดงชื่อของระบบของคุณ ถ้าคุณอยู่บนเครือข่าย ป้อน:

   ```
   uname -n
   ```

   หน้าจอเหมือนดังต่อไปนี้จะถูกแสดง:

   ```
   barnard
   ```

   ในตัวอย่างนี้ ชื่อระบบคือ barnard
2. เพื่อหาชื่อโหนดของระบบอื่น ขอให้ผู้ใช้บนระบบนั้นใช้คำสั่ง **uname –n**

การดูว่าระบบของคุณเข้าถึง

ใช้ **host** เพื่อดูว่าระบบของคุณเข้าถึงข้อมูลที่ระบุระบบอื่นหรือไม่

เมื่อต้องการเข้าถึงระบบอื่นบนเครือข่าย ระบบโลคัลของคุณต้องเข้าถึงข้อมูลที่ระบุระบบอื่น เพื่อดูว่าระบบโลคัลของคุณมีข้อมูลนี้หรือไม่ ใช่คำสั่ง **host** พร้อมกับชื่อของระบบอื่น

เพื่อดูว่าระบบของคุณมีข้อมูลเส้นทางสำหรับระบบ zeus หรือไม่ ใส่:

```
host zeus
```

ถ้าระบบของคุณมีข้อมูลที่เหมาะสม หน้าจอเหมือนดังต่อไปนี้จะถูกแสดง :

```
zeus is 192.9.200.4 (300,11,310,4)
```

คุณสามารถส่งข้อความไปยังระบบ zeus แอดเดรส 192.9.200.4 ถูกใช้โดยระบบเพื่อส่งเมล ถ้าระบบของคุณไม่มีข้อมูล หน้าจอเหมือนดังต่อไปนี้จะถูกแสดง:

```
zeus: unknown host
```

ถ้าคุณได้รับข้อความ unknown host ดังนั้นชื่อระบบที่ต้องการ:

- ไม่ถูกต้อง (ตรวจสอบการสะกดคำในแอดเดรส)
- อยู่บนเน็ตเวิร์กของคุณ แต่ไม่ถูกกำหนดกับระบบของคุณ (ติดต่อบุคคลที่รับผิดชอบสำหรับการติดตั้งเน็ตเวิร์กของคุณ)
- อยู่บนเน็ตเวิร์กอื่น (ดูที่ “การกำหนดแอดเดรสเมลไปยังผู้ใช้บนเน็ตเวิร์กอื่น” ในหน้า 25) และต้องการการระบแอดเดรสที่ละเอียดมากขึ้น
- ไม่ได้เชื่อมต่อกับเน็ตเวิร์กของคุณ

คุณยังอาจได้รับข้อความ unknown host ถ้าเน็ตเวิร์กของคุณไม่ทำงาน และระบบโลคัลของคุณขึ้นอยู่กับระบบรีโมตที่จะให้เน็ตเวิร์กแอดเดรส

การแสดงข้อมูลเกี่ยวกับผู้ใช้ที่ล็อกอิน

ใช้คำสั่ง **finger** หรือ **f** เพื่อแสดงข้อมูลเกี่ยวกับผู้ใช้ปัจจุบันบนโฮสต์ที่ระบุ

ข้อมูลนี้สามารถระบุชื่อล็อกอินของผู้ใช้ และชื่อเทอร์มินัล พร้อมกับวันและเวลาที่ล็อกอิน

1. เมื่อต้องการแสดงข้อมูลเกี่ยวกับผู้ใช้ที่ล็อกอินกับโฮสต์ @alcatraz ใส่ :

```
finger @alcatraz
```

หน้าจอเหมือนดังต่อไปนี้จะถูกแสดง :

```
brown  Console Mar 15 13:19
smith  pts0    Mar 15 13:01
jones  tty0    Mar 15 13:01
```

ผู้ใช้ brown มีการล็อกอินที่คอนโซล ผู้ใช้ smith มี การล็อกอินจาก pseudo teletype line pts0 และผู้ใช้ jones มี การล็อกอินจาก tty0

2. เพื่อให้ได้ข้อมูลเกี่ยวกับผู้ใช้ brown จากตัวอย่างก่อนหน้านี้ ใส่ :

```
finger brown@alcatraz
```

หรือ

```
finger brown
```

หน้าจอเหมือนดังต่อไปนี้จะถูกแสดง :

```
Login name: brown
In real life: Marta Brown
Directory:/home/brown  Shell: /bin/ksh
On since May 8 07:13:49 on console
No Plan.
```

การจัดการเมล

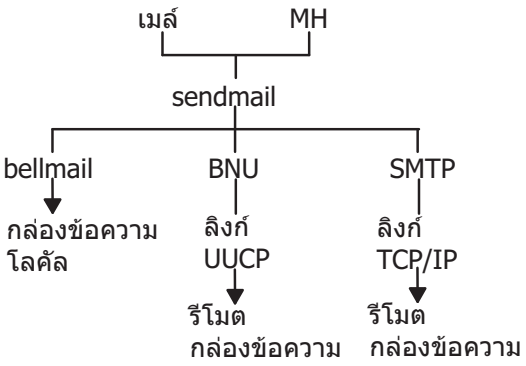
สิ่งอำนวยความสะดวกของเมลจัดเตรียมวิธีสำหรับการแลกเปลี่ยนจดหมายอิเล็กทรอนิก (อีเมล) กับผู้ใช้บนระบบเดียวกันหรือบนหลายระบบที่เชื่อมต่อกันโดยเน็ตเวิร์ก ระบบเมล ส่วนติดต่อผู้ใช้เมลมาตรฐาน **Internet Message Access Protocol (IMAP)** และ **Post Office Protocol (POP)** จะถูกอธิบายที่นี่

ระบบเมลเป็นเป็นเครื่องอำนวยความสะดวกในการส่งเมลภายในเน็ตเวิร์กที่ประกอบด้วยส่วนติดต่อผู้ใช้โปรแกรมการเราต์ข้อความ และส่วนติดต่อผู้ใช้การนำส่งข้อความ (หรือ mailer) ระบบเมลจะส่งทอดข้อความจากผู้ใช้หนึ่งไปยังผู้อื่นบนโฮสต์เดียวกัน และข้ามเน็ตเวิร์ก มันยังทำการแก้ไขส่วนหัวของของข้อความที่จำกัดจำนวนหนึ่งเพื่อใส่ข้อความในรูปแบบที่เหมาะสมสำหรับโฮสต์ที่รับ

ส่วนติดต่อผู้ใช้ ของเมลจะทำให้ผู้ใช้สามารถสร้างและส่งข้อความไปยัง และรับข้อความจากผู้ใช้อื่น ระบบเมลจัดเตรียมส่วนติดต่อผู้ใช้ 2 ส่วน **mail** และ **mhmail** คำสั่ง **mail** เป็นส่วนติดต่อผู้ใช้เมลมาตรฐานที่มีบนระบบ UNIX ทุกระบบ คำสั่ง **mhmail** เป็นส่วนติดต่อผู้ใช้ Message Handler (MH) ที่ปรับปรุงส่วนติดต่อผู้ใช้เมลที่ถูกออกแบบสำหรับผู้ใช้ที่มีประสบการณ์

โปรแกรมการเราต์ข้อความ จะเราต์ข้อความไปยังปลายทางของมัน โปรแกรมการเราต์ข้อความระบบเมลเป็นโปรแกรม **sendmail** ซึ่งเป็นส่วนของ Base Operating System (BOS) และถูกติดตั้งกับ BOS โปรแกรม **sendmail** เป็น daemon ที่ใช้ข้อมูลในไฟล์ /etc/mail/sendmail.cf ไฟล์ /etc/mail/aliasesเพื่อทำการเราต์ที่จำเป็น

ขึ้นอยู่กับชนิดของเส้นทางไปยังปลายทาง คำสั่ง **sendmail** จะใช้ *mailers* ที่ต่างกันในการส่งข้อความ



รูปที่ 2. Mailers จะถูกใช้โดยคำสั่ง sendmail

การแสดงนี้เป็นชนิดของแผนภูมิเกี่ยวกับการจัดระเบียบแบบบนลงล่างโดย Mail และ MH อยู่ด้านบน สาขาของมันคือ bellmail, BNU และ SMTP ใต้ระดับก่อนหน้านี้เป็นโลคัลเมลบ็อกซ์, UUCP link และ **TCP/IP** ลิงก์ ตามลำดับ ใต้ UUCP ลิงก์เป็นรีโมตเมลบ็อกซ์ และใต้ **TCP/IP** ลิงก์เป็นรีโมตเมลบ็อกซ์

ดังภาพจะแสดง:

- เพื่อส่งโลคัลเมล โปรแกรม **sendmail** จะเราต์ข้อความไปยังโปรแกรม **bellmail** โปรแกรม **bellmail** ส่งโลคัลเมลทั้งหมดโดยการต่อท้ายข้อความไปยังระบบเมลบ็อกซ์ของผู้ใช้ ซึ่งอยู่ในไดเร็กทอรี /var/spool/mail
- เพื่อส่งเมลบน UNIX-to-UNIX Copy Program (UUCP) ลิงก์ โปรแกรม **sendmail** จะเราต์ข้อความโดยใช้ Basic Network Utilities (BNU)
- เพื่อส่งเมลที่ถูกเราต์ผ่าน **Transmission Control Protocol/Internet Protocol (TCP/IP)** คำสั่ง **sendmail** จะสร้างการเชื่อมต่อ **TCP/IP** ไปยังระบบรีโมต จากนั้นใช้ **Simple Mail Transfer Protocol (SMTP)** เพื่อถ่ายโอนข้อความไปยังระบบรีโมต

โปรแกรมเมลเอเจนต์ของผู้ใช้

ก่อนที่คุณจะสามารถใช้ระบบเมล คุณต้องเลือกโปรแกรมเอเจนต์ของผู้ใช้ คุณสามารถเลือกโปรแกรมเมล **mail**) ตัวจัดการข้อความ (**mh**) หรือคำสั่ง **bellmail**

โปรแกรมเอเจนต์ของผู้ใช้ให้ความสะดวกในการสร้าง รับ ส่งและจัดเก็บเมลเข้าแฟ้ม นอกจากนี้คุณต้องการ transport-agent โปรแกรม **sendmail** ซึ่งจะกระจายเมลขาเข้าจากระบบอื่นหรือแพ็กเกจ และกระจายแต่ละไอเท็มของเมลขาออก และจากนั้นส่งมันไปยังโปรแกรมเดียวกันในหนึ่งหรือหลายระบบรีโมต

> **หมายเหตุ:** โปรแกรม **mail** และ **mh** ไม่สามารถเข้ากันกับวิธีที่มันเก็บเมล คุณต้องเลือกตัวจัดการเมลหนึ่งตัว หรืออย่างอื่น

อินเตอร์เฟสของโปรแกรมเมล

โปรแกรม **mail** จัดเตรียมส่วนติดต่อผู้ใช้เพื่อจัดการกับเมลที่ไปยังและจากทั้งผู้ใช้ของโลคัลเน็ตเวิร์กและผู้ใช้ของระบบรีโมต

ข้อความเมลสามารถเป็นเท็กซ์ ที่ถูกใส่โดยใช้เอดิเตอร์ หรือไฟล์ ASCII นอกเหนือจากพิมพ์ข้อความหรือไฟล์ คุณสามารถส่ง :

ไอเท็ม	คำอธิบาย
ข้อความระบบ	จะบอกผู้ใช้ว่าระบบถูกอัพเดตแล้ว ข้อความระบบจะเหมือนกับข้อความบรอดคาสต์ แต่จะถูกส่งบนเน็ตเวิร์กแบบโลคัลเท่านั้น
เมลความลับ	ใช้เพื่อส่งข้อมูลที่เป็นความลับ เมลความลับจะถูกเข้ารหัส ผู้รับต้องใส่รหัสผ่านเพื่ออ่านมัน
ข้อความวันหยุดพักผ่อน	บอกว่าผู้ใช้พักร้อนอยู่ เมื่อระบบของคุณได้รับเมลว่าคุณไม่อยู่ มันจะส่งข้อความกลับไปที่ต้นทาง ข้อความจะบอกว่าคุณพักร้อนอยู่ เมลใดๆที่คุณได้รับขณะพักร้อนจะสามารถถูกส่ง

เมื่อคุณรับเมลโดยใช้คำสั่งย่อย **mail** คุณสามารถ :

- ทิ้งเมลไว้ในระบบเมลบ็อกซ์
- อ่านและลบเมล
- ฟอร์เวิร์ดเมล
- เพิ่มหมายเหตุเข้ากับเมล
- เก็บเมลในเมลบ็อกซ์ส่วนตัวของคุณ (mbox)
- เก็บเมลในโฟลเดอร์ที่คุณสร้าง
- สร้างและรักษาไฟล์ alias หรือไฟล์ดีสทริบิวชัน ซึ่งส่งเมลและข้อความเมล

การติดตั้ง **sendmail** จะเป็นแบบอัตโนมัติ

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับโปรแกรม **mail** อ้างถึง “ฟังก์ชันเมล” ในหน้า 13

Messenger handler (mh)

โปรแกรม **mh** เป็นชุดของคำสั่งที่ให้คุณสามารถทำแต่ละฟังก์ชันการประมวลผลเมลโดยตรงจากบรรทัดรับคำสั่ง

คำสั่งเหล่านี้จัดเตรียมช่วงของฟังก์ชันที่กว้างกว่าคำสั่งย่อยของ **mail** นอกจากนี้ เนื่องจากคำสั่งสามารถถูกใช้ได้ทุกเวลาที่จุดรับคำสั่งถูกแสดง คุณจะได้รับความสามารถและความยืดหยุ่นในการสร้างเมลและประมวลผลเมลที่ได้รับ ตัวอย่างเช่น คุณสามารถอ่านข้อความเมล ค้นหาไฟล์ หรือรันโปรแกรมเพื่อหาวิธีการแก้ปัญหาและตอบข้อความ โดยทั้งหมดอยู่ในเชลล์เดียวกัน

โปรแกรม **mh** ให้คุณสามารถสร้าง กระจาย รับ ดู ประมวลผล และเก็บข้อความโดยใช้คำสั่งต่อไปนี้ :

ไอเท็ม	คำอธิบาย
ali	แสดง alias เมลและแอดเดรส
anno	เพิ่มความคิดเห็นข้อความ
ap	วิเคราะห์คำและปรับรูปแบบแอดเดรส
burst	กระจายส่วนย่อยไปยังข้อความ
comp	สตาร์ทเอดิเตอร์สำหรับการสร้างและแก้ไขข้อความ
dist	จัดสรรข้อความใหม่ไปที่แอดเดรสเพิ่มเติม
dp	วิเคราะห์คำและจัดรูปแบบวันที่ใหม่
folder	เลือกและแสดงรายการโฟลเดอร์และข้อความ
folders	ลิสต์โฟลเดอร์และข้อความทั้งหมดในเมลไดเร็กทอรี
forw	ส่งต่อข้อความ
inc	รวมเมลใหม่ในโฟลเดอร์
mark	สร้าง แก้ไข และแสดงลำดับข้อความ
mhl	สร้างลิสต์ที่ได้รับการจัดรูปแบบของข้อความ
mhmail	ส่งหรือรับเมล
mhpath	พิมพ์ชื่อพาธแบบเต็มของข้อความและโฟลเดอร์
msgchk	ตรวจหาข้อความ
msh	สร้างเชลล์ของ mail handler (mh)

ไอเท็ม	คำอธิบาย
next	แสดงข้อความถัดไป
packf	บีบอัดเนื้อหาของโฟลเดอร์ลงในไฟล์
pick	เลือกข้อมูลตามเนื้อหาและสร้างและแก้ไข ลำดับ
prev	แสดงข้อความก่อนหน้า
refile	ย้ายไฟล์ระหว่างโฟลเดอร์
repl	ตอบกลับข้อความ
rmf	ลบโฟลเดอร์และข้อความที่มี
rmm	ลบข้อความออกจากสถานะแอ็คทีฟ
scan	สร้างลิสต์แบบหนึ่งบรรทัดต่อข้อความที่สามารถสแกนได้
send	ส่งข้อความ
show	แสดงข้อความ
sortm	เรียงลำดับข้อความ
vmh	สตาร์ทอินเตอร์เฟสเสมือนสำหรับใช้กับคำสั่ง **mh**
whatnow	เริ่มทำงานอินเตอร์เฟสการพร้อมต์สำหรับการจัดการแบบร่าง
whom	จัดการกับแอดเดรส **mh**

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ **mh** อ้างถึง *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 3*

คำสั่ง bellmail

คำสั่ง **bellmail** เป็นคำสั่งเมล AT&T UNIX แบบดั้งเดิม ซึ่งจะจัดการเมลสำหรับผู้ใช้บนระบบเดียวกัน และยังสำหรับผู้ใช้บนระบบรีโมตที่สามารถถูกเข้าถึงโดย Basic Network Utilities (BNU) บางครั้งรู้จักในชื่อ UNIX-to-UNIX Copy Program (UUCP)

โปรแกรมเหล่านี้สนับสนุนเฉพาะเน็ตเวิร์กของระบบที่ถูกเชื่อมต่อโดยการหมุนโทรศัพท์ หรือสายการสื่อสารแบบเช่าจุดต่อจุด คำสั่งจะเปิดเชลล์ที่คำสั่งย่อยให้คุณสามารถที่จะ :

- รับข้อมูลจากอินพุตมาตรฐาน (พิมพ์เข้าไปหรือถูกเปลี่ยนทิศทางจากไฟล์ที่มีอยู่) เพิ่มแอดเดรสหนึ่งแอดเดรสหรือมากกว่า (ถูกกำหนดเป็นอาร์กิวเมนต์กับคำสั่งของมันเอง) และเวลาประทับ จากนั้นต่อคัดลอกเข้ากับแต่ละไฟล์ระบบเมลบ็อกซ์ของผู้รับ (`/var/spool/mail/`*`UserID`*)
- อ่านไอเท็มของเมลจากไฟล์ระบบเมลบ็อกซ์ของคุณ
- ต่อท้านไอเท็มของเมลเข้ากับไฟล์เมลบ็อกซ์ส่วนตัวของคุณ (`$HOME/mbox`) หรือเพื่อระบุไฟล์
- ส่งเมลโดยใช้ BNU ไปยังผู้ใช้บนระบบอื่น
- เปลี่ยนทิศทางเมลโดยอัตโนมัติจากระบบเมลบ็อกซ์ของคุณไปยังเมลบ็อกซ์อื่นบนระบบอื่นโดยการเพิ่มข้อความ *.forward* เข้ากับส่วนเริ่มต้นของไฟล์ระบบเมลบ็อกซ์ของคุณ

อย่างไรก็ตาม คุณต้องมีความชำนาญบางอย่างเป็นผู้ใช้ UNIX ก่อนที่คุณจะใช้ตัวจัดการเมลนี้ได้อย่างเต็มที่ สำหรับข้อมูลเพิ่มเติม อ้างถึง คำสั่ง **bellmail** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1*

ฟังก์ชันเมล

คุณลักษณะของโปรแกรม **เมล** แนะนำไว้ที่นี่

โปรแกรม**เมล** ช่วยให้คุณรับ สร้าง และส่ง เมลถึงผู้ใช้ในระบบโลคัลหรือรีโมตของคุณ

ที่เก็บข้อมูลเมล

เมลถูกจัดเก็บหลายวิธีขึ้นอยู่กับสถานการณ์เฉพาะ

เมื่อเมลถูกส่งถึงแอดเดรสของคุณ เมลจะถูกเก็บไว้ในไดเร็กทอรีระบบที่ เฉพาะสำหรับเมล ไดเร็กทอรีระบบนี้มีไฟล์สำหรับผู้ใช้ทุกคน บนระบบโลคัล ไดเร็กทอรีนี้เก็บเมลของคุณไว้จนกว่าคุณจะทำอย่างอื่นกับมัน

เมลบ็อกซ์ของระบบ:

เมลบ็อกซ์ของระบบจะเหมือนกับกล่องไปรษณีย์ : ไปรษณีย์จะนำส่งจดหมายที่มีแอดเดรสถึงบุคคลที่เป็นเจ้าของกล่องนั้น

เช่นเดียวกัน เมลบ็อกซ์ของระบบเป็นไฟล์ที่ข้อความถูกนำส่งไปยังผู้ใช้นั้นๆ ถ้าไฟล์ไม่มีอยู่เมื่อเมลมาถึง มันจะถูกสร้าง ไฟล์จะถูกลบเมื่อข้อความทั้งหมดถูกลบ

เมลบ็อกซ์ของระบบจะอยู่ในไดเร็กทอรี `/var/spool/mail` แต่ละเมลบ็อกซ์ของระบบจะถูกตั้งชื่อโดย ID ID ผู้ใช้ที่เกี่ยวข้องกับมัน ตัวอย่างเช่น ถ้า ID ID ผู้ใช้ของคุณคือ `karen` เมลบ็อกซ์ของระบบของคุณคือ :

```
/var/spool/mail/karen
```

เมลบ็อกซ์ส่วนตัวแบบดีฟอลต์:

เมลบ็อกซ์ส่วนตัวของคุณจะเหมือนกับตะกร้าขาเข้าในสำนักงาน คุณใส่เมลในตะกร้าขาเข้าหลังจากที่คุณได้รับมัน ก่อนที่คุณจะเก็บมันเข้าแฟ้ม

ผู้ใช้แต่ละคนจะมีเมลบ็อกซ์ส่วนตัว เมื่อคุณอ่านเมลจะระบบเมลบ็อกซ์ และถ้ามันไม่ถูกมาร์กสำหรับลบหรือบันทึกในไฟล์ มันจะถูกเขียนไปยังเมลบ็อกซ์ส่วนตัวของคุณ `$HOME/mbox` (`$HOME` เป็นไดเร็กทอรีล็อกอินของคุณ ไฟล์ `mbox` จะมีอยู่เฉพาะเมื่อมันมีข้อความ

ไฟล์ dead.letter สำหรับข้อความที่ไม่สมบูรณ์:

ถ้าคุณต้องการอินเตอร์รัปต์ข้อความที่คุณสร้างเพื่อทำงานอื่น ระบบจะบันทึกข้อความที่ไม่สมบูรณ์ในไฟล์ `dead.letter` ในไดเร็กทอรี `$HOME`

ถ้าไฟล์ `dead.letter` ไม่มีอยู่ ไฟล์จะถูกสร้าง หลังจากนั้นคุณสามารถแก้ไขไฟล์เพื่อทำข้อความของคุณให้สมบูรณ์

> **ข้อควรใส่ใจ:** ห้ามใช้ไฟล์ `dead.letter` เพื่อเก็บข้อความ เนื้อหาของไฟล์นี้จะถูกเขียนทับแต่ละครั้งที่อินเตอร์รัปต์ถูกใช้เพื่อบันทึกข้อความบางส่วนเข้ากับไปยัง `dead.letter`

เมลโฟลเดอร์:

โฟลเดอร์ทำให้คุณสามารถบันทึกข้อความในแบบที่มีระเบียบ โดยใช้โปรแกรมเมล คุณสามารถใส่ข้อความเข้าไปยังโฟลเดอร์จากระบบเมลบ็อกซ์ เมลบ็อกซ์ส่วนตัว หรือโฟลเดอร์อื่นๆ

แต่ละโฟลเดอร์เป็นเท็กซ์ไฟล์ แต่ละโฟลเดอร์จะอยู่ในไดเร็กทอรีที่คุณระบุในไฟล์ `.mailrc` ด้วยอ็อพชัน **set folder** คุณต้องสร้างไดเร็กทอรีนี้ก่อนที่จะใช้โฟลเดอร์เพื่อเก็บข้อความ เมื่อไดเร็กทอรีมีอยู่แล้ว โปรแกรมเมลจะสร้างโฟลเดอร์ในไดเร็กทอรีนั้นตามต้องการ ถ้าคุณไม่ได้ระบุไดเร็กทอรีในไฟล์ `.mailrc` ของคุณ โฟลเดอร์จะถูกสร้างในไดเร็กทอรีปัจจุบัน โปรดดู “การจัดระเบียบเมล” ในหน้า 20

> **หมายเหตุ :** หลายโปรแกรมจะพร้อมที่จะส่งและรับเมล รวมถึง Message Handler (MH) และโปรแกรม **bellmail** โดยโปรแกรมที่คุณใช้จะขึ้นอยู่กับอะไรที่ถูกติดตั้งและตั้งค่าบนระบบของคุณ สำหรับข้อมูลเกี่ยวรู้การตั้งค่าระบบของคุณ ติดต่อผู้ดูแลระบบของคุณ

การจัดการและการรับเมล

โปรแกรม **mail** ให้คุณสามารถตรวจสอบแต่ละข้อความในเมลบ็อกซ์ และจากนั้นลบหรือไฟล์ข้อความในไดเร็กทอรีเมลส่วนบุคคล

เชลล์คำสั่งจะแจ้งคุณว่าเมลมาถึงแล้ว การแจ้งจะถูกแสดงก่อนพร้อมต์ถัดไป ถูกจัดเตรียมว่าตัวแปรสภาวะแวดล้อม **MAIL** ถูกตั้งและถูกจัดเตรียมว่าช่วงเวลาที่ถูกระบุโดย **MAILCHECK** ได้ผ่านไปตั้งแต่เชลล์ตรวจสอบเมลครั้งสุดท้าย ข้อความแจ้งเป็นค่าของตัวแปรสภาวะแวดล้อม **MAILMSG** ขึ้นอยู่กับเชลล์ใดที่คุณใช้ (bourne, korn หรือ C shell) การแจ้งจะเหมือนดังต่อไปนี้ :

```
YOU HAVE NEW MAIL
```

เมลบ็อกซ์เริ่มทำงาน:

ใช้คำสั่ง **mail** เพื่ออ่านและลบข้อความ จากเมลบ็อกซ์ระบบของคุณ

ห้ามใช้เมลบ็กซ์ระบบเพื่อเก็บข้อความ ให้เก็บข้อความของคุณไว้ในเมลบ็อกซ์ส่วนบุคคล และในโฟลเดอร์เมล

การตรวจสอบระบบเมลบ็อกซ์ของคุณสำหรับเมล:

ใช้คำสั่ง **mail** เพื่อตรวจสอบระบบเมลบ็อกซ์ของคุณสำหรับเมล

ที่พร้อมต์ของบรรทัดรับคำสั่งของคุณ ใส่คำสั่ง **mail** :

```
mail
```

ถ้าไม่มีเมลในระบบเมลบ็อกซ์ของคุณ ระบบจะตอบสนองด้วยข้อความ :

```
No mail for YourID
```

ถ้ามีเมลในเมลบ็อกซ์ของคุณ ระบบจะแสดงลิสต์ของข้อความในระบบเมลบ็อกซ์ของคุณ :

```
Mail Type ? สำหรับความช่วยเหลือ
"/usr/mail/lance": 3 messages 3 new
  >N    1 karen Tue Apr 27 16:10 12/321 "Dept Meeting"
   N    2 lois  Tue Apr 27 16:50 10/350 "System News"
   N    3 tom   Tue Apr 27 17:00 11/356 "Tools Available"
```

ข้อความปัจจุบันจะถูกนำหน้าด้วยเครื่องหมายมากกว่า (>) เสมอ แต่ละรายการหนึ่งบรรทัดนั้นแสดงฟิลด์ต่อไปนี้:

ไอเท็ม	คำอธิบาย
สถานะ	ระบุคลาสของข้อความ
จำนวน	ระบุชิ้นของเมลกับโปรแกรมเมล
sender	ระบุแอดเดรสของบุคคลที่ส่งเมล
date	ระบุวันที่ที่ได้รับข้อความ
ขนาด	กำหนดจำนวนบรรทัดและอักขระที่มีในข้อความ (นี้รวมส่วนหัวด้วย)
subject	ระบุเรื่องของข้อความ ถ้ามี

สถานะสามารถเป็น ค่าใดต่อไปนี้:

ไอเท็ม	คำอธิบาย
N	ข้อความใหม่
P	ข้อความที่ถูกสงวนไว้ในระบบเมลบ็อกซ์ของคุณ
U	ข้อความที่ยังไม่เปิดอ่าน นี่เป็นข้อความที่ถูกลิสต์ในเมลบ็อกซ์ครั้งสุดท้ายที่คุณใช้โปรแกรมเมล แต่เนื้อหายังไม่ได้ถูกตรวจสอบ
*	ข้อความที่ถูกบันทึกหรือเขียนไปยังไฟล์หรือโฟลเดอร์

ข้อความที่ไม่มีสถานะระบุว่า ข้อความได้ถูกอ่านแล้วแต่ยังไม่ถูกลบหรือบันทึก

การตรวจสอบเมลบ็อกซ์ส่วนตัวหรือเมลโฟลเดอร์ของคุณสำหรับเมล:

คุณสามารถใช้คำสั่ง **mail** เพื่อตรวจสอบเมลบ็อกซ์ส่วนตัวหรือเมลโฟลเดอร์ของคุณสำหรับเมล

ที่พร้อมต์ของบรรทัดรับคำสั่งของระบบของคุณ คุณสามารถใช้คำสั่ง **mail** ในวิธีที่ถูกแสดงในขั้นตอนต่อไปนี้ :

1. เพื่อแสดงลิสต์ของข้อความในเมลบ็อกซ์ส่วนตัวของคุณ `$HOME/mbox` ใส่ :

   ```
   mail -f
   ```

   ถ้าไม่มีเมลในเมลบ็อกซ์ส่วนตัวของคุณ ระบบจะตอบสนองด้วยข้อความที่เหมือนดังต่อไปนี้ :

   ```
   "/u/george/mbox": 0 messages
   ```

   หรือ

   ```
   ไฟล์หรือไดเร็กทอรีในชื่อพาธไม่มีอยู่
   ```

2. เพื่อแสดงลิสต์ของข้อความในโฟลเดอร์ `dept` ใส่ :

   ```
   mail -f +dept
   ```

   ถ้าไม่มีเมลในเมลโฟลเดอร์ของคุณ ระบบจะตอบสนองด้วยข้อความที่เหมือนดังต่อไปนี้ :

   ```
   ไฟล์หรือไดเร็กทอรีในชื่อพาธไม่มีอยู่
   ```

อ็อพชันการแสดงเนื้อหาของเมลบ็อกซ์:

จากพร้อมต์ของเมลบ็อกซ์ คุณสามารถใส่คำสั่งย่อยของเมลบ็อกซ์เพื่อจัดการเนื้อหาของเมลบ็อกซ์

ข้อกำหนดเบื้องต้น

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. โปรแกรมเมลต้องถูกสตาร์ท
3. ต้องมีเมลในเมลบ็อกซ์ของคุณ

ช่วงของข้อความ:

ใช้คำสั่งย่อย **h** เพื่อดูข้อความที่อยู่ภายในลิสต์ของข้อความที่คุณกำหนด เพื่อที่คุณจะไม่ต้องเรียกดูข้อความของคุณทั้งหมด

ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย **h** ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
h	ประมาณ 20 ข้อความจะถูกแสดงพร้อมกัน จำนวนที่แท้จริงที่ถูกแสดงจะถูกกำหนดโดยชนิดของเทอร์มินัลที่ถูกใช้และอ็อพชัน **set screen** ในไฟล์ .mailrc ของคุณ ถ้าคุณใส่คำสั่งย่อย **h** อีกครั้ง ช่วงของข้อความเดิมจะถูกแสดง
h 21	ข้อความที่ 21 และข้อความที่ตามมา ถึงข้อความที่ 40 (ถ้าคุณมีข้อความจำนวนนั้นในเมลบ็อกซ์ของคุณ) จะถูกแสดง พิมพ์คำสั่งย่อย **h** ต่อด้วยหมายเลขข้อความที่ตามมาจนกว่าข้อความทั้งหมดจะถูกแสดง
h 1	เพื่อแสดงกลุ่มของของ 20 ข้อความแรก ใส่จำนวนใดๆภายในช่วง 1-20

การเลื่อนเมลบ็อกซ์:

ใช้คำสั่งย่อย **z** เพื่อเลื่อนผ่านเมลบ็อกซ์ของคุณ

ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย **z** ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
z	ประมาณ 20 ข้อความจะถูกแสดงพร้อมกัน จำนวนที่แท้จริงที่ถูกแสดงจะถูกกำหนดโดยชนิดของเทอร์มินัลที่ถูกใช้และอ็อพชัน **set screen** ในไฟล์ .mailrc ของคุณ ใส่คำสั่งย่อย **z** อีกครั้งเพื่อเลื่อนไปยัง 20 ข้อความถัดไป
z +	อาร์กิวเมนต์เครื่องหมายบวก (+) จะเลื่อนไปยัง 20 ข้อความถัดไป ข้อความที่ 21 และข้อความที่ตามมา ถึงข้อความที่ 40 (ถ้าคุณมีข้อความจำนวนนั้นในเมลบ็อกซ์ของคุณ) จะถูกแสดง พิมพ์คำสั่งย่อย **z+** ต่อจนกระทั่งข้อความทั้งหมดถูกแสดง ระบบจะตอบสนองด้วยข้อความต่อไปนี้ : `On last screenful of messages.`
z-	อาร์กิวเมนต์เครื่องหมายลบ (-) จะเลื่อนไปยัง 20 ข้อความก่อนหน้านี้ เมื่อคุณไปถึงชุดแรกของข้อความ ระบบจะตอบสนองด้วยข้อความต่อไปนี้ : `On first screenful of messages.`

การฟิลเตอร์ข้อความสำหรับข้อมูลที่ระบุ:

ที่พร้อมต์ของเมลบ็อกซ์ คุณสามารถใช้คำสั่งย่อย **f** ในวิธีที่แสดงต่อไปนี้เพื่อฟิลเตอร์ข้อความเพื่อข้อมูลที่คุณต้องการ

ไอเท็ม	คำอธิบาย
f	แสดงข้อมูลส่วนหัวสำหรับข้อความปัจจุบัน
f 1 4 7	แสดงข้อมูลส่วนหัวสำหรับข้อความที่ระบุ 1, 4 และ 7
f 1-10	แสดงข้อความส่วนหัวสำหรับช่วงของข้อความ 1 ถึง 10
f *	แสดงข้อความทั้งหมด

f ron	ข้อความ ถ้ามี จากผู้ใช้ ron จะถูกแสดง อักขระที่ถูกใส่สำหรับแอดเดรสที่จำเป็นต้องตรงกับแอดเดรสทุกประการ ดังนั้น คำร้องขอสำหรับแอดเดรส ron ในแบบตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็กจะตรงกับแอดเดรสต่อไปนี้ทั้งหมด :
	`RoN`
	`ron@topdog`
	`hron`
	`rOn`

fmeet	ข้อความ ถ้ามีโดยที่ฟิลด์ **Subject:** ประกอบด้วยตัวอักษร meet จะถูกแสดง อักขระที่ถูกใส่สำหรับรูปแบบไม่จำเป็นต้องตรงกับฟิลด์ **Subject:** ทุกประการ มันจะต้องอยู่ในฟิลด์ **Subject:** โดยเป็นตัวอักษรแบบตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็กเท่านั้น ดังนั้น คำร้องขอสำหรับหัวเรื่อง meet จะตรงกับหัวเรื่องต่อไปนี้ :
	`Meeting on Thursday`
	`Come to meeting tomorrow`
	`MEET ME IN ST. LOUIS`

หมายเลขข้อความ ปัจจุบัน:

คำสั่งย่อย = จะแสดงหมายเลขข้อความ

ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย = ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
=	หมายเลขของข้อความปัจจุบันจะถูกแสดง

จำนวนทั้งหมดของข้อความในเมลบ็อกซ์ของคุณ:

ใช้คำสั่งย่อย **folder** เพื่อตรวจสอบจำนวนของข้อความในเมลบ็อกซ์ของคุณ

ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย **folder** sในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
โฟลเดอร์	ลิสต์ข้อมูลเกี่ยวกับโฟลเดอร์หรือเมลบ็อกซ์ของคุณ ระบบจะตอบสนองเหมือนกับต่อไปนี้ : `"/u/lance/mbox": 29 messages.`

อ็อพชันการอ่านเมล:

คุณสามารถอ่านเมลได้หลายวิธี ตัวอย่างของแต่ละวิธีถูกอธิบายในที่นี้

เลือกวิธีที่เหมาะสมที่สุด และใช้มันอ่านเมลของคุณ ก่อนที่จะพยายามอ่านเมลของคุณ ต้องแน่ใจว่าเงื่อนไขต่อไปนี้เป็นจริง :

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. โปรแกรมเมลต้องถูกสตาร์ท
3. ต้องเป็นเมลในระบบเมลบ็อกซ์ของคุณ

การอ่านข้อความในเมลบ็อกซ์ของคุณ:

ใช้คำสั่งย่อย **t** หรือ **p** เพื่ออ่านข้อความในเมลบ็อกซ์ของคุณ

ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย **t** หรือ**p** ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
3	ถ้าคุณใช้หมายเลขของข้อความ โดยดีฟอลต์เท็กซ์ของข้อความจะถูกแสดง
t	ถ้าคุณใช้คำสั่งย่อย **t** โดยดีฟอลต์ เท็กซ์ของข้อความปัจจุบันจะถูกแสดง
t 3	เท็กซ์ของข้อความที่ 3 จะถูกแสดง
t 2 4 9	ข้อความสำหรับข้อความที่ 2, 4 และ 9 จะถูกแสดง
t 2-4	ข้อความสำหรับช่วงของข้อความที่ 2 ถึง 4 จะถูกแสดง
t	ถ้าคุณใช้คำสั่งย่อย **p** โดยดีฟอลต์ เท็กซ์ของข้อความปัจจุบันจะถูกแสดง
p 3	เท็กซ์ของข้อความที่ 3 จะถูกแสดง
p 2 4 9	ข้อความสำหรับข้อความที่ 2, 4 และ 9 จะถูกแสดง
p 2-4	ข้อความสำหรับช่วงของข้อความที่ 2 ถึง 4 จะถูกแสดง

การอ่านข้อความถัดไปในเมลบ็อกซ์ของคุณ:

ใช้คำสั่งย่อย **n** เพื่ออ่านข้อความถัดไปในเมลบ็อกซ์ของคุณ

ที่พร้อมต์ของเมลบ็อกซ์ คุณสามารถใช้คำสั่งย่อย (**n**)ext หรือเครื่องหมายบวก (+) ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
n หรือ +	แสดงเท็กซ์ของข้อความถัดไป และข้อความนี้จะกลายเป็นข้อความปัจจุบัน

คุณยังสามารถกดคีย์ Enter เพื่อแสดงเท็กซ์ของข้อความถัดไป

การอ่านข้อความก่อนหน้านี้ในเมลบ็อกซ์ของคุณ:

ใช้คำสั่งย่อย – เพื่ออ่านข้อความก่อนหน้านี้

ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย – ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
–	เท็กซ์ของข้อความก่อนหน้านี้จะถูกแสดง

การลบเมล:

เมื่อลบข้อความ คุณสามารถลบข้อความปัจจุบัน ลบข้อความที่ระบุ หรือลบช่วงของข้อความ

คุณยังสามารถลบข้อความปัจจุบันและแสดงข้อความถัดไปโดยการรวมคำสั่งย่อย ต้องแน่ใจว่าเงื่อนไขต่อไปนี้มีครบ :

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. ต้องเป็นเมลในระบบเมลบ็อกซ์ของคุณ
3. โปรแกรมเมลต้องถูกสตาร์ท

การลบข้อความ:

ใช้รูปแบบต่างๆของคำสั่งย่อย **d** เพื่อลบข้อความ

ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย (**d**)elete ในวิธีการที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
d	ข้อความปัจจุบันจะถูกลบ
dp หรือ **dt**	ข้อความปัจจุบันจะถูกลบและข้อความถัดไปจะถูกแสดง นี่ยังสามารถทำได้โดยการรวมอ็อพชัน **set autoprint** ในไฟล์ `.mailrc` ซึ่งจะตั้งคำสั่งย่อย **d** ให้ทำงานเหมือนกับการรวมคำสั่งย่อย **dp** หรือ **dt**
d 4	ลบข้อความ 4 ที่ระบุ
d 4-6	ลบช่วงของข้อความ 4 ถึง 6
d 2 6 8	ลบข้อความ 2, 6 และ 8

การยกเลิกการลบข้อความ:

ใช้คำสั่งย่อย **u** สำหรับการยกเลิกการลบข้อความ

ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย **u** ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
u	ข้อความปัจจุบันจะถูกยกเลิกการลบ
u 4	การยกเลิกการลบข้อความที่ 4
u 4-6	การยกเลิกการลบช่วงของข้อความที่s 4 ถึง 6
u 2 6 8	การยกเลิกข้อความที่ 2, 6 และ 8

การออกจากเมล:

ต้องแน่ใจว่าข้อกำหนดต่อไปนี้มีพร้อมก่อนที่จะออกจากโปรแกรมเมล

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. ต้องเป็นเมลในระบบเมลบ็อกซ์ของคุณ
3. โปรแกรมเมลต้องถูกสตาร์ท

การออกจากเมลและ่บันทึกการเปลี่ยนแปลง:

ใช้คำสั่งย่อย **q** เพื่อออกจากเมลและบันทึกการเปลี่ยนแปลง

ถ้าคุณออกจากระบบเมลบ็อกซ์ :

ไอเท็ม	คำอธิบาย
q	คำสั่งย่อย **q** จะออกจากระบบเมลบ็อกซ์และกลับไปยังระบบปฏิบัติการ เมื่อคุณออกจากเมลบ็อกซ์ ข้อความทั้งหมดที่ถูกมาร์กเพื่อลบจะถูกลบจากเมลบ็อกซ์และไม่สามารถเรียกคืนได้ โปรแกรมเมลจะบันทึกข้อความที่คุณอ่านในเมลบ็อกซ์ส่วนตัว (**mbox**) ถ้าคุณไม่ได้อ่านเมลของคุณ ข้อความจะยังคงอยู่ในระบบเมลบ็อกซ์จนกว่าจะกระทำการใดๆ

ถ้าคุณออกจากเมลบ็อกซ์ส่วนตัวหรือเมลโฟลเดอร์ของคุณ :

ไอเท็ม	คำอธิบาย
q	เมื่อใช้คำสั่งย่อย **q** ในเมลบ็อกซ์ส่วนตัวหรือเมลโฟลเดอร์ของคุณ ข้อความที่ถูกอ่านและไม่ได้อ่านจัยังคงอยู่ในเมลบ็อกซ์ส่วนตัวหรือเมลโฟลเดอร์จนกว่าจะถูกกระทำใดๆ

การออกจากเมลโดยไม่บันทึกการเปลี่ยนแปลง:

ใช้คำสั่งย่อย **x** หรือ **ex** เพื่อออกจากเมลโดยไม่ทำการแก้ไขเมลบ็อกซ์

ไอเท็ม	คำอธิบาย
x or **ex**	คำสั่งย่อย **x** หรือ **ex** ยอมให้คุณออกจากเมลบ็อกซ์และกลับไปยังระบบปฏิบัติการโดยไม่มีการเปลี่ยนแปลงเนื้อหาดั้งเดิมของเมลบ็อกซ์ โปรแกรมจะไม่สนใจการร้องขอใดๆของคุณก่อนหน้าคำร้องขอ **x** อย่างไรก็ตาม ถ้าคุณทำการบันทึกข้อความไปยังโฟลเดอร์อื่น การบันทึกจะเกิดขึ้น

การจัดระเบียบเมล:

ใช้โฟลเดอร์เพื่อบันทึกข้อความในแบบที่ได้รับการจัดระเบียบ

คุณสามารถสร้างโฟลเดอร์ได้มากเท่าที่คุณต้องการ ตั้งชื่อให้แต่ละโฟลเดอร์ด้วยชื่อที่สอดคล้องกับเรื่องของข้อความ เหมือนกับโฟลเดอร์ของไฟล์ในระบบการเข้าแฟ้มของสำนักงาน แต่ละโฟลเดอรเป็นเท็กซ์ไฟล์ที่ถูกวางในไดเร็กทอรีที่คุณระบุในไฟล์ `.mailrc` ด้วยอ็อพชัน **set folder** คุณต้องสร้างไดเร็กทอรีนี้ก่อนที่จะใช้โฟลเดอร์เพื่อเก็บข้อความ เมื่อไดเร็กทอรีมีอยู่แล้ว

โปรแกรมเมลจะสร้างโฟลเดอร์ในไดเร็กทอรีนั้นตามต้องการ ถ้าคุณไม่ได้ระบุไดเร็กทอรีด้วยอ็อพชัน **set folder** ในไฟล์ `.mailrc` ของคุณ โฟลเดอร์จะถูกสร้างในไดเร็กทอรีปัจจุบัน โดยใช้โปรแกรมเมล คุณสามารถใส่ข้อความเข้าไปยังโฟลเดอร์จากระบบเมลบ็อกซ์ เมลบ็อกซ์ส่วนตัว หรือโฟลเดอร์อื่นๆ

คุณสามารถเพิ่มเนื้อหาของข้อความเข้ากับไฟล์หรือโฟลเดอร์โดยใช้ คำสั่งย่อย **s** หรือ **w** คำสั่งย่อยทั้งสองเหล่านี้จะต่อท้ายข้อมูลเข้ากับไฟล์ที่มีอยู่ หรือสร้างไฟล์ใหม่ถ้ามันไม่มีอยู่ ข้อมูลที่อยู่ในไฟล์อยู่แล้วจะไม่ถูกทำลาย ถ้าคุณบันทึกข้อความจากระบบเมลบ็อกซ์ของคุณไปยังไฟล์หรือโฟลเดอร์ ข้อความจะถูกลบจากระบบเมลบ็อกซ์ของคุณและถูกถ่ายโอนไปยังไฟล์หรือโฟลเดอร์ที่ระบุ ถ้าคุณบันทึกข้อความจากเมลบ็อกซ์ส่วนตัวหรือโฟลเดอร์ของคุณไปยังไฟล์หรือโฟลเดอร์อื่น ข้อความจะไม่ถูกลบจากเมลบ็อกซ์ส่วนตัวของคุณแต่จะถูกคัดลอกไปยังไฟล์หรือโฟลเดอร์ที่ระบุ เมื่อใช้คำสั่งย่อย **s** คุณสามารถอ่านโฟลเดอร์เหมือนกับเมลบ็อกซ์ เนื่องจากข้อความและข้อมูลส่วนหัวจะถูกต่อท้ายที่ท้ายของโฟลเดอร์ เมื่อใช้คำสั่งย่อย **w** คุณสามารถอ่านโฟลเดอร์เหมือนกับไฟล์ เนื่องจากข้อความจะถูกต่อท้ายโดยไม่มีข้อมูลส่วนหัวที่ท้ายของไฟล์

ก่อนที่คุณจะจัดระเบียบเมล ต้องแน่ใจว่าข้อกำหนดต่อไปนี้พร้อมแล้ว :

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. ต้องมีเมลในเมลบ็อกซ์ เมลบ็อกซ์ส่วนตัว หรือโฟลเดอร์ของคุณ ที่คุณกำหนด
3. โปรแกรมเมลต้องถูกสตาร์ท

การสร้างไดเร็กทอรีเมลบ็อกซ์ตัวอักษรเพื่อเก็บข้อความในโฟลเดอร์:

ข้อความสามารถถูกบันทึกในโฟลเดอร์ไดเร็กทอรีเมลบ็อกซ์โดยใช้คำสั่งย่อย **set folder**

ใช้โพรซีเดอร์ต่อไปนี้เพื่อเก็บข้อความในโฟลเดอร์ :

1. เพื่อตรวจสอบว่าอ็อพชัน **set folder** ถูกเปิดใช้งานในไฟล์ `.mailrc` หรือไม่ ใส่คำสั่งย่อยต่อไปนี้ที่พร้อมต์ของเมลบ็อกซ์ :

   ```
   set
   ```

   คำสั่งย่อย **set** จะแสดงลิสต์ของอ็อพชันที่ถูกเปิดใช้งานในไฟล์ `.mailrc` ของคุณ

   ถ้าอ็อพชัน **set folder** ถูกเปิดใช้งาน ระบบจะตอบสนองด้วยข้อความที่เหมือนดังต่อไปนี้ :

   ```
   folder /home/george/letters
   ```

   ในตัวอย่างนี้ `letters` เป็นไดเร็กทอรีที่เมลโฟลเดอร์จะถูกเก็บ
2. ถ้าอ็อพชัน **set folder** ไม่ถูกเปิดใช้งาน เพิ่มบรรทัดเหมือนดังต่อไปนี้ในไฟล์ `.mailrc` :

   ```
   set folder=/home/george/letters
   ```

   ในตัวอย่างนี้ `/home/george` เป็นไดเร็กทอรีหลักของจอร์จ และ `letters` เป็นไดเร็กทอรีที่เมลโฟลเดอร์จะถูกเก็บ อ็อพชัน **set folder** จะให้คุณสามารถใช้เครื่องหมายบวก (+) สัญลักษณ์ชวเลขที่พร้อมต์ของเมลบ็อกซ์ของคุณเพื่อบันทึกข้อความในไดเร็กทอรีของคุณ `letters`
3. คุณต้องสร้างไดเร็กทอรี `letters` ในไดเร็กทอรีหลักของคุณ ในไดเร็กทอรีหลักของคุณที่พร้อมต์บรรทัดรับคำสั่งของระบบ พิมพ์ :

   ```
   mkdir letters
   ```

การบันทึกข้อความพร้อมกับส่วนหัว:

คำสั่งย่อย **s** จะบันทึกข้อความพร้อมกับส่วนหัว

ใช้คำสั่งย่อย **s** ในวิธีต่อไปนี้:

ไอเท็ม	คำอธิบาย
s 1-4 notes	บันทึกข้อความ 1, 2, 3 และ 4 พร้อมกับข้อมูลส่วนหัวของมันยังโฟลเดอร์ชื่อ notes ในไดเร็กทอรีปัจจุบัน โปรแกรมเมลจะตอบสนองด้วยข้อความต่อไปนี้ : "notes" [Appended] 62/1610
s +admin	บันทึกข้อความปัจจุบันยังโฟลเดอร์ที่มีอยู่แล้วชื่อ admin ในโฟลเดอร์ไดเร็กทอรีของคุณ ถ้าโฟลเดอร์ไดเร็กทอรีถูกกำหนดเป็น /home/george/lettersในไฟล์ .mailrc ของคุณ ระบบจะตอบสนองด้วย : "/home/george/letters/admin" [Appended] 14/321
s 6 +admin	บันทึกข้อความ 6 ยังโฟลเดอร์ที่มีอยู่แล้วชื่อ admin ในโฟลเดอร์ไดเร็กทอรีของคุณ ถ้าโฟลเดอร์ไดเร็กทอรีถูกกำหนดเป็น /home/george/lettersในไฟล์ .mailrc ของคุณ ระบบจะตอบสนองด้วย : "/home/george/letters/admin" [Appended] 14/321

การบันทึกข้อความโดยไม่มีส่วนหัว:

ใช้คำสั่งย่อย **w** เพื่อบันทึกข้อความเป็นไฟล์แทนที่จะเป็นโฟลเดอร์

เพื่ออ่านหรือแก้ไขไฟล์ที่ถูกบันทึกด้วยคำสั่งย่อย **w** คุณต้องใช้ **vi** หรือเท็กซ์เอดิเตอร์อื่น ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย **w** ในวิธีต่อไปนี้ :

ไอเท็ม	คำอธิบาย
w 6 pass	บันทึกเฉพาะเท็กซ์ของข้อความที่ 6 ไปยังไฟล์ชื่อ pass ในไดเร็กทอรีปัจจุบัน ถ้าไฟล์ pass ไม่มีอยู่ ระบบจะตอบสนองด้วยข้อความต่อไปนี้ : "pass" [New file] 12/30 ถ้าไฟล์ pass มีอยู่แล้ว ระบบจะตอบสนองด้วยข้อความต่อไปนี้ : "pass" [Appended] 12/30
w 1-3 safety	บันทึกเฉพาะเท็กซ์ของข้อความที่ 1, 2 และ 3 ไปยังไฟล์ชื่อ safety ในไดเร็กทอรีปัจจุบัน เท็กซ์ของข้อความในตัวอย่างนี้จะถูกต่อท้ายข้อความอื่นในไฟล์เดียว ถ้าไฟล์ safety ไม่มีอยู่ ระบบจะตอบสนองด้วยข้อความต่อไปนี้ : "safety" [New file] 12/30

การกำหนดเมลบ็อกซ์หรือโฟลเดอร์ปัจจุบัน:

ใช้คำสั่งย่อย **folder** เพื่อกำหนดเมลบ็อกซ์หรือโฟลเดอร์ปัจจุบัน

แม้ว่าคำสั่ง **mail** จะแสดงชื่อของเมลบ็อกซ์ปัจจุบันเมื่อมันสตาร์ท มันอาจจะไม่รู้ว่าเมลบ็อกซ์ใดที่คุณอยู่ ที่พร้อมต์ของเมลบ็อกซ์ของคุณ คุณสามารถใช้คำสั่งย่อย **folder** ที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
folder	หาชื่อของเมลบ็อกซ์หรือโฟลเดอร์ปัจจุบันของคุณ ถ้าเมลบ็อกซ์ปัจจุบันคือ `/home/lance/mbox` ต่อไปนี้จะถูกแสดง : `/home/lance/mbox: 2 messages 1 deleted` ข้อความนี้จะระบุว่า `/home/lance/mbox` เป็นเมลบ็อกซ์ปัจจุบันที่คุณอยู่ มันประกอบด้วย 2 ข้อความ และหนึ่งในข้อความเหล่านั้นจะถูกลบเมื่อคุณเสร็จกับเมลบ็อกซ์นี้

การเปลี่ยนไปยังเมลบ็อกซ์อื่น:

การเปลี่ยนไปยังเมลบ็อกซ์อื่นเหมือนกับการออกจากเมลบ็อกซ์หรือโฟลเดอร์

ข้อความใดๆที่คุณมาร์กเพื่อจะลบจะถูกลบเมื่อคุณออกจากเมลบ็อกซ์ ข้อความที่ถูกลบจะไม่สามารถเรียกคืน ที่พร้อมต์ของเมลบ็อกซ์ คุณสามารถใช้คำสั่งย่อย **file** หรือ **folder** ที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย

folder `+project`	หลังจากโปรแกรมเมลถูกสตาร์ทด้วยหนึ่งเมลบ็อกซ์ ใช้คำสั่งย่อย **file** หรือ **folder** เพื่อเปลี่ยนไปยังเมลบ็อกซ์อื่น
	
	ถ้าคุณเปลี่ยนจากไฟล์ `mbox` เป็นโฟลเดอร์ `mbox` และคุณลบข้อความทั้งหมดในไฟล์ `mbox` โปรแกรมจะถูกแสดง :
	`/home/dee/mbox removed`
	`+project: 2 messages 2 new`
	
	ตามด้วยลิสต์ของข้อความในโฟลเดอร์ `project`

การสร้างและส่งเมล

คุณสามารถใช้โปรแกรม **mail** เพื่อสร้าง ส่ง ตอบ และฟอร์เวิร์ดข้อความไปยังผู้อื่น หรือส่งไฟล์ ASCII ไปยังผู้ใช้อื่น

ตัวอย่างเช่น ในไฟล์ ASCII อาจเป็นเอกสารที่คุณเขียนโดยใช้เอดิเตอร์ที่ต้องการหรือไฟล์ต้นฉบับสำหรับโปรแกรม

คุณสามารถส่งข้อความและไฟล์ไปยังผู้ใช้บนระบบโลคัลของคุณ บนเน็ตเวิร์กของคุณ หรือผู้ใช้บนเน็ตเวิร์กอื่นที่เชื่อมต่ออยู่ The recipient does not need to be logged on to the system when you send the information. เมลจะถูกส่งไปยังแอดเดรสของผู้ใช้

การกำหนดแอดเดรสของเมล:

เมลจะถูกส่งไปยังแอดเดรสของผู้ใช้ แอดเดรส ประกอบด้วยชื่อล็อกอินและชื่อระบบ จะกำหนดการส่งข้อความเมล

โดยทั่วไป เพื่อส่งข้อความไปยังผู้ใช้อื่น คุณต้องใช้คำสั่ง **mail** และแอดเดรส ดังต่อไปนี้ :

`mail` *`User@Address`*

รูปแบบของพารามิเตอร์ *Address* ขึ้นอยู่กับสถานที่ของผู้รับ แนวคิดเหมือนกับวิธีที่คุณส่งบันทึกให้กับผู้ร่วมงานในสำนักงาน เพื่อส่งบันทึกถึงไรอัน ที่ทำงานในแผนกเล็กๆที่มี 6–8 คน คุณอาจเขียนชื่อบนซองจดหมายและใส่มันในระบบจดหมายของสำนักงาน อย่างไรก็ตาม ถ้าไรอันอยู่ในแผนกอื่น คุณอาจต้องให้ข้อมูลเพิ่มเติมบนซองจดหมาย :

```
Ryan
Payroll
```

ถ้าไรอันอยู่ในสถานที่อื่น คุณอาจต้องการเพิ่มข้อมูลเพื่อให้แน่ใจว่าข้อความจะส่งถึงไรอัน :

```
Ryan
Payroll
Gaithersburg
```

เพื่อส่งจดหมายแบบอิเล็คทรอนิค ใช้กระบวนการกำหนดแอดเดรสแบบเดียวกัน :

ไอเท็ม	คำอธิบาย
mail ryan	เพื่อส่งเมลไปยังผู้ใช้บนระบบโลคัลของคุณ ชื่อล็อกอินเป็นเพียงส่วนเดียวของแอดเดรสที่ต้องการ
mail ryan@tybalt	เพื่อส่งเมลไปยังผู้ใช้บนโลคัลเน็ตเวิร์กของคุณ ใส่แอดเดรสแบบเต็มของระบบของคุณ (โหนด)
mail ryan@mars.aus.dbm.com	เพื่อส่งเมลไปยังผู้ใช้บนเน็ตเวิร์กอื่นที่ถูกเชื่อมต่อ ใส่แอดเดรสแบบเต็มของระบบและเน็ตเวิร์กแอดเดรส
mail dept71	คุณสามารถส่งเมลไปยังกลุ่มของบุคคลที่ระบุโดยใช้ alias หรือดีสทริบิวชันลิสต์ เพื่อทำดังกล่าวคุณต้องสร้าง alias หรือดีสทริบิวชันลิสต์ในไฟล์ .mailrc ของคุณ ถ้าคุณต้องการข้อมูลเพิ่มเติมเกี่ยวกับการสร้าง aliase ดูที่ "Alias และดีสทริบิวชันลิสต์" ในหน้า 40

การจัดส่งอีเมลถึงผู้ใช้มากกว่าหนึ่งราย:

เมื่อต้องการส่งเมลถึงผู้ใช้มากกว่าหนึ่งรายพร้อมกัน ให้แยกชื่อผู้ใช้ แต่ละคนด้วยช่องว่าง

ตัวอย่างเช่น:

```
ryan@tybalt suemc@julius dmorgan@ophelia
```

การกำหนดแอดเดรสเมลไปยังผู้ใช้บนระบบโลคัลของคุณ:

เพื่อส่งข้อความไปยังผู้ใช้บนระบบโลคัลของคุณ (ไปยังบางคนที่ชื่อล็อกอินถูกลิสต์ในไฟล์ /etc/passwd ของคุณ) ใช้ชื่อล็อกอินสำหรับแอดเดรส

ที่พร้อมต์ของบรรทัดรับคำสั่งของระบบของคุณ คุณสามารถใช้คำสั่ง **mail** ดังแสดงในตัวอย่างต่อไปนี้ :

```
mail LoginName
```

ไอเท็ม	คำอธิบาย
mail ryan	ถ้าไรอันอยู่บนระบบของคุณและมีชื่อล็อกอิน ryan คำสั่งนี้จะเปิดใช้งานโปรแกรมเมล และพยายามส่งข้อความไปยังชื่อล็อกอินแบบโลคัลของ ryan หากข้อความของคุณถูกนำส่งสำเร็จ คุณจะไม่ได้รับ การแจ้งเตือน ถ้าไรอันไม่อยู่บนระบบของคุณ ระบบเมลจะส่งข้อความแสดงข้อผิดพลาดกลับมาทันที และส่งข้อความที่ไม่สามารถส่งกลับมายังระบบเมลบ็อกซ์ของคุณ

การกำหนดแอดเดรสเมลไปยังผู้ใช้บนเน็ตเวิร์ของคุณ:

ใช้คำสั่ง **mail** เพื่อส่งข้อความไปยังผู้ใช้บนเน็ตเวิร์กของคุณ รวมชื่อล็อกอินของผู้ใช้และชื่อระบบในแอดเดรส

เพื่อส่งข้อความผ่านโลคัลเน็ตเวิร์กไปยังผู้ใช้บนระบบอื่น ที่บรรทัดรับคำสั่ง พิมพ์ :

ไอเท็ม	คำอธิบาย
mail *LoginName@SystemName*	ตัวอย่างเช่น ถ้าไรอันอยู่บนระบบ `zeus` ใช้คำสั่งต่อไปนี้เพื่อสร้างและส่งข้อความไปยังไรอัน : `mail ryan@zeus` คำสั่งนี้จะเปิดใช้งานโปรแกรมเมล ให้คุณสามารถสร้างข้อความ และลองส่งข้อความไปยังชื่อล็อกอิน `ryan` บนระบบ `zeus` ถ้าข้อความถูกส่งสำเร็จ คุณจะได้รับพร้อมต์ระบบโดยไม่มีการแจ้งเตือน ถ้าแอดเดรสของเมลไม่ถูกต้อง คุณจะได้รัลข้อความแสดงข้อผิดพลาด

หมายเหตุ: เพื่อส่งข้อความผ่านโลคัลเน็ตเวิร์กไปยังผู้ใช้บนระบบอื่น คุณต้องรู้ชื่อล็อกอินและชื่อของระบบอื่น สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการแสดงข้อมูลที่ระบุผู้ใช้ ดูที่ "การสื่อสารของคำสั่งของระบบ" ในหน้า 8

การกำหนดแอดเดรสเมลไปยังผู้ใช้บนเน็ตเวิร์กอื่น:

ถ้าเน็ตเวิร์กของคุณเชื่อมต่อกับเน็ตเวิร์กอื่น คุณสามารถส่งเมลไปยังผู้ใช้บนเน็ตเวิร์กอื่น

พารามิเตอร์แอดเดรสจะแตกต่างกันโดยขึ้นอยู่กับวิธีที่เน็ตเวิร์กของคุณและเน็ตเวิร์กอื่นถูกกำหนดแอดเดรสถึงกันและกันและวิธีที่มันเชื่อมต่อ โดยขึ้นอยู่กับวิธีการตั้งค่าเน็ตเวิร์กของคุณ ใช้หนึ่งในแอ็คชันต่อไปนี้ :

- ถ้าคุณใช้ฐานข้อมูลของชื่อและแอดเดรสส่วนกลาง ใช้คำสั่ง **mail** ดังแสดงในตำแหน่งต่อไปนี้ :

  mail *LoginName@SystemName*

  ถ้าเน็ตเวิร์กใช้ฐานข้อมูลของชื่อที่เป็นส่วนกลาง คุณไม่ต้องการข้อมูลเพิ่มเติมใดๆเพื่อส่งเมลไปยังผู้ใช้บนเน็ตเวิร์กที่เชื่อมต่อ ใช้รูปแบบแอดเดรสเหมือนสำหรับผู้บนโลคัลเน็ตเวิร์กของคุณ

  การกำหนดแอดเดรสชนิดนี้ทำงานได้ดีเมื่อธรรมชาติของเน็ตเวิร์กยอมให้ฐานข้อมูลส่วนกลางของชื่อถูกบำรุงรักษา
- ถ้าคุณใช้การกำหนดแอดเดรสชื่อของโดเมน ใช้คำสั่ง **mail** ที่แสดงในตัวอย่างต่อไปนี้ :

  mail *LoginName@SystemName.DomainName*

  สำหรับเน็ตเวิร์กที่มีขนาดใหญ่ เน็ตเวิร์กที่ไม่เกี่ยวข้องในตำแหน่งขนาดกว้าง ฐานข้อมูลส่วนกลางของชื่อจะใช้ไม่ได้ พารามิเตอร์ *DomainName* จะกำหนดรีโมตเน็ตวิร์ก ความสัมพันธ์กับโลคัลเน็ตเวิร์กของคุณ ภายในโครงสร้างที่ถูกกำหนดสำหรับกลุ่มขนาดใหญ่ของการเชื่อมต่อของเน็ตเวิร์ก

  ตัวอย่างเช่น ถ้าคุณใส่คำสั่งต่อไปนี้ :

  mail `kelly@merlin.odin.valryanl`

  เมลของคุณจะถูกส่งไปยังผู้ใช้ `kelly` บนระบบ `merlin` ซึ่งอยู่บนโลคัลเน็ตเวิร์กที่ชื่อ `odin` ที่เชื่อมต่ออยู่กับเน็ตเวิร์กที่สองของโดเมนชื่อ `valryanl`

เมลแอดเดรสบน BNU หรือ UUCP ลิงก์:

คุณสมามารถส่งข้อความถึงผู้ใช้บนระบบอื่นบน Basic Networking Utilities (BNU) หรือ UNIX-to-UNIX Copy Program (UUCP) ลิงก์

เพื่อส่งข้อความไปยังผู้ใช้บนระบบอื่นที่เชื่อมต่ออยู่กับระบบของคุณโดย BNU หรือเวอร์ชันอื่นของ UUCP คุณต้องรู้ :

- ชื่อล็อกอิน
- ชื่อของระบบอื่น
- เส้นทางแบบฟิสิคัลไปยังระบบอื่นนั้น

บุคคลี้รับผิดชอบสำหรับเชื่อมต่อระบบของคุณเข้ากับระบบอื่นควรจะสามารถให้ข้อมูลเส้นทางไปยังแอดเดรสของระบบอื่น

เมื่อคอมพิวเตอร์ของคุณมี BNU หรือ UUCP ลิงก์: ที่บรรทัดรับคำสั่งของระบบของคุณ ใช้คำสั่ง **mail** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
mail *UUCPRoute*!*LoginName*	ถ้าโลคัลคอมพิวเตอร์ของคุณมีการเชื่อมต่อ BNU หรือ UUCP ที่สามารถถูกใช้เพื่อเข้าถึงระบบรีโมต ใช้รูปแบบในตัวอย่างนี้เพื่อกำหนดแอดเดรสของข้อความ ตัวแปร *LoginName* เป็นชื่อล็อกอินบนระบบรีโมตสำหรับผู้รับข้อความ ตัวแปร *UUCPRoute* จะอธิบายเส้นทางแบบฟิสิคัลที่ข้อความต้องใช้ใน UUCP เน็ตเวิร์ก ถ้าระบบของคุณเชื่อมต่อกับระบบรีโมตโดยไม่มีระบบ UUCP ระดับกลางอยู่ตรงกลาง ตัวแปรนี้จะเป็นชื่อของระบบรีโมต
mail`arthur!lancelot!merlin!ken`	ถ้าข้อความต้องเดินทางผ่านระบบ UUCP ระดับกลางก่อนที่จะถึงระบบรีโมตที่ต้องการ ตัวแปรนี้จะเป็นลิสต์ของแต่ละระบบระดับกลาง ลิสต์จะเริ่มต้นด้วยระบบที่ใกล้ที่สุดไปจนถึงระบบที่ไกลที่สุด โดยคั่นด้วยเครื่องหมายตกใจ (!). คุณสามารถทำตามตัวอย่างนี้ ถ้าข้อความต้องเดินทางผ่านระบบ `arthur` และ `lancelot` (ในลำดับนั้น) ก่อนที่จะถึง `merlin`
mail `merlin!ken`	ถ้าระบบโลคัลของคุณมีลิงก์ UUCP ไปยังระบบที่ชื่อ `merlin` และไม่มีระบบ UUCP อื่นระหว่างระบบของคุณและ `merlin` คุณสามารถส่งข้อความไปยัง `ken` บนระบบนั้น

เมื่อ BNU หรือ UUCP ลิงก์อยู่บนคอมพิวเตอร์อื่น : ในสภาวะแวดล้อม local หรือ wide area network หนึ่งในระบบบนเน็ตเวิร์กอาจมีการเชื่อมต่อ BNU หรือ UUCP ชนิดอื่นไปยังระบบรีโมต คุณสามารถใช้การเชื่อมต่อ UUCP นั้นเพื่อส่งข้อความไปยังผู้ใช้บนระบบรีโมต UUCP ที่บรรทัดรับคำสั่งของระบบของคุณ ใช้คำสั่ง **mail** ดังแสดงในตัวอย่างต่อไปนี้ :

mail `@arthur:merlin!ken`

ส่งเมลไปยัง `ken` บนระบบ UUCP `merlin` จากระบบอินเตอร์เน็ต `arthur` ตัวคั่น @ ใช้สำหรับการกำหนดอินเตอร์เน็ตแอดเดรส ตัวคั่น ! ใช้สำหรับการกำหนดแอดเดรส UUCP และตัวคั่น : เชื่อม 2 แอดเดรสเข้าด้วยกัน โปรดสังเกตว่าในรูปแบบนี้จะจะไม่ส่งเมลไปยังผู้ใช้ที่ระบบระดับกลางใดๆ ดังนั้นไม่ต้องมีชื่อล็อกอินนำหน้า @ ในโดเมนแอดเดรส

mail `@arthur:odin!acct.dept!kelly`

เพื่อส่งเมลไปยัง `kelly` บนระบบ UUCP `acct.dept` ผ่านระบบ `odin` จากระบบอินเตอร์เน็ต `arthur`

mail`@odin.uucp:@dept1.UUCP:@dept2:bill@dept3`

ส่งเมลไปยัง `bill@dept3` บน `odin` และ `dept1` UUCP ลิงก์ และจากนั้นโลคัลเน็ตเวิร์กลิงก์ระหว่างระบบ `dept2` และ `dept3` ไฟล์ `/etc/sendmail.cf` ต้องถูกตั้งค่าให้สอดคล้องเพื่อใช้ชนิดของแอดเดรส UUCP นี้ ปรึกษาผู้ดูแลระบบของคุณสำหรับข้อมูลเพิ่มเติม

ถ้าคุณส่งเมลไปยังผู้ใช้บนเน็ตเวิร์กอื่นบ่อยๆ การสร้าง alias ที่รวมแอดเดรสของผู้ใช้สามารถช่วยประหยัดเวลาของคุณ โปรดดู “Alias และดีสทริบิวชันลิสต์” ในหน้า 40

การสตาร์ทเมลเอดิเตอร์:

โปรแกรม **mail** จัดเตียวโปรแกรมแบบรายบรรทัดสำหรับการสร้างเมล

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. โปรแกรมเมลต้องถูกสตาร์ท

เอดิเตอร์นี้ทำให้คุณสามารถพิมพ์แต่ละบรรทัดของข้อความ กดคีย์ Enter เพื่อขึ้นบรรทัดใหม่ และพิมพ์เท็กซ์เพิ่มเติม คุณไม่สามารถเปลี่ยนบรรทัดหลังจากคุณกดคีย์ Enter แล้ว อย่างไรก็ตาม ก่อนที่จะกดคีย์ Enter คุณสามารถเปลี่ยนข้อมูลบนบรรทัดเดียวนั้นโดยใช้คีย์ Backspace และ Delete เพื่อลบ คุณยังสามารถใช้คำสั่งย่อยเมลเอดิเตอร์เพื่อเข้าสู่เอดิเตอร์แบบเต็มจอและเปลี่ยนข้อความ

เมื่อคุณสร้างเมลด้วยเมลเอดิเตอร์ ฟิลด์ **date:** และ **from:** จะถูกเติมโดยระบบโดยอัตโนมัติ คุณมีทางเลือกที่จะเติมฟิลด์ **subject:** และ **cc:** ฟิลด์เหล่านี้จะเหมือนกับส่วนเนื้อหาของจดหมายธุรกิจมาตรฐาน

เมลเอดิเตอร์จะรวมคำสั่งย่อยที่ใช้ควบคุมหลายคำสั่งที่ให้คุณสามารถดำเนินการอื่นบนข้อความ คำสั่งย่อยเหล่านี้แต่ละคำสั่งต้องถูกใส่บนบรรทัดใหม่และต้องเริ่มต้นด้วยอักขระ *escape* พิเศษ โดยดีฟอลต์ อักขระ escape คือ tilde (~) คุณสามารถเปลี่ยนมันเป็นอักขระอื่นโดยการรวมอ็อพชัน **set escape** ในไฟล์ `.mailrc` ของคุณ

ที่พร้อมต์บรรทัดรับคำสั่งของระบบของคุณ หรือพร้อมต์ของเมลบ็อกซ์ คุณสามารถใช้คำสั่ง **mail** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
mail *User@Address*	ใช้คำสั่งนี้จากพร้อมต์ของบรรทัดรับคำสั่ง ข้อความจะถูกกำหนดแอดเดรสถึง *User@Address* พารามิเตอร์ *Address* จะขึ้นอยู่กับตำแหน่งของผู้รับ
m*User@Address*	ใช้คำสั่งย่อยนี้จากพร้อมต์ของเมลบ็อกซ์ ข้อความจะถูกกำหนดแอดเดรสถึง *User@Address* พารามิเตอร์ *Address* จะขึ้นอยู่กับตำแหน่งของผู้รับ

เมลเอดิเตอร์ยังถูกเปิดใช้งาน ถ้าคุณใช้คำสั่งย่อย **R** หรือ **r** เพื่อตอบข้อความ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีตอบข้อความ ดูที่ “การส่งเมล” ในหน้า 32 และ “การตอบเมล” ในหน้า 33

การแก้ไขข้อความ:

ขณะที่อยู่ในเมลบ็อกซ์ของคุณ คุณสามารถเพิ่มข้อมูลเข้ากับข้อความที่มีอยู่ โดยการพิมพ์คำสั่งย่อย (**e**)dit หรือ (**v**)isual ที่พร้อมต์ของเมลบ็อกซ์

ขณะที่อยู่ในเมลเอดิเตอร์ คุณไม่สามารถเปลี่ยนข้อมูลบนบรรทัดหลังจากที่คุณได้กดคีย์ Enter และไปยังบรรทัดใหม่แล้ว คุณสามารถเปลี่ยนเนื้อหาของข้อความของคุณก่อนที่จะส่งมันโดยแก้ไขข้อความด้วยเอดิเตอร์อื่น

ก่อนที่จะแก้ไขข้อความในเอดิเตอร์อื่น ต้องแน่ใจว่าเงื่อนไขต่อไปนี้เป็นจริง :

1. เมลโปรแกรมต้องถูกติดตั้งบนระบบของคุณ
2. เอดิเตอร์อื่นต้องถูกกำหนดในไฟล์ `.mailrc` ด้วย :

   ```
   set EDITOR=PathName
   ```

   นี่จะกำหนดเอดิเตอร์ที่คุณเปิดใช้งานด้วยคำสั่งย่อย **~e** ค่าของ *PathName* ต้องเป็นชื่อพาธแบบเต็มไปยังโปรแกรมเอดิเตอร์ที่คุณต้องการใช้ ตัวอย่างเช่น คำจำกัดความ `set EDITOR=/usr/bin/vi` จะกำหนดเอดิเตอร์ **vi** สำหรับใช้กับคำสั่งย่อย **~e**
3. เพื่อเพิ่มข้อมูลเข้ากับข้อความในเมลบ็อกซ์ของคุณ คุณต้องสตาร์ทคำสั่ง **mail** เพื่ออ่านเมลในระบบเมลบ็อกซ์ เมลบ็อกซ์หรือโฟลเดอร์อื่น
4. เพื่อสตาร์ทเอดิเตอร์อื่นขณะที่สร้างข้อความ คุณต้องอยู่ที่พร้อมต์ของเมลเอดิเตอร์

การเพิ่มข้อมูลเข้ากับข้อความที่ระบุในเมลบ็อกซ์ของคุณ:

เพื่อเพิ่มข้อมูลเข้ากับข้อความในเมลบ็อกซ์ของคุณ ใส่คำสั่งย่อย **e** หรือคำสั่งย่อย **v** ตามด้วยหมายเลขของข้อความ

ที่พร้อมต์ของเมลบ็อกซ์ คุณสามารถใช้คำสั่งย่อย **e** หรือคำสั่งย่อย **v** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
e 13	เพื่อเพิ่มบันทึกเข้ากับข้อความ 13 โดยใช้ **e** เอดิเตอร์ (หรือเอดิเตอร์ใดก็ตามที่ถูกกำหนดในไฟล์ `.mailrc`)
v 15	เพื่อเพิ่มบันทึกเข้ากับข้อความ 15 โดยใช้ **vi** เอดิเตอร์ (หรือเอดิเตอร์ใดก็ตามที่ถูกกำหนดในไฟล์ `.mailrc`)

ถ้าคุณไม่ระบุหมายเลขข้อความ คำสั่ง **mail** จะเปิดใช้งานเอดิเตอร์โดยใช้ข้อความปัจจุบัน เมื่อคุณออกจากเอดิเตอร์ คุณจะกลับมาที่พร้อมต์ของเมลบ็อกซ์เพื่อดำเนินการกับข้อความในเมลบ็อกซ์ต่อ

การเปลี่ยนแปลงข้อความปัจจุบันขณะที่อยู่ในเมลเอดิเตอร์:

ที่เริ่มต้นของบรรทัดในเมลเอดิเตอร์ คุณสามารถใช้คำสั่งย่อย **~e** หรือคำสั่งย่อย **~v** ดังแสดงในตัวอย่างเหล่านี้

ไอเท็ม	คำอธิบาย
~e	เปิดใช้งาน **e** เอดิเตอร์หรือเอดิเตอร์อื่นที่คุณกำหนดในไฟล์ `.mailrc`
~v	เปิดใช้งาน **vi** เอดิเตอร์หรือเอดิเตอร์อื่นที่คุณกำหนดในไฟล์ `.mailrc`

ซึ่งจะให้คุณสามารถแก้ไขข้อความของข้อความปัจจุบัน เมื่อคุณออกจากเอดิเตอร์อื่น คุณจะกลับมาที่เมลเอดิเตอร์

แสดงผลบรรทัดของข้อความขณะที่อยู่ในเมลเอดิเตอร์:

ใช้คำสั่งย่อย **~p** เพื่อแสดงบรรทัดของข้อความขณะที่อยู่ในเมลเอดิเตอร์

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. เพื่อแสดงข้อความขณะที่อยู่ในเมลเอดิเตอร์ คุณต้องสตาร์ทเมลเอดิเตอร์แล้ว ถ้าคุณต้องการข้อมูลเพิ่มเติม ดูที่ “การสตาร์ทเมลเอดิเตอร์” ในหน้า 26

ที่เริ่มต้นของบรรทัดขณะที่อยู่ในเมลเอดิเตอร์ ใช้คำสั่งย่อย **~p** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
~p	เอดิเตอร์จะแสดงเนื้อหาของข้อความรวมถึงข้อมูลส่วนหัวสำหรับข้อความ ข้อความจะเลื่อนขึ้นจากล่างสุดของหน้าจอ ที่ท้ายของข้อความจะตามด้วยพร้อมต์ (`Continue`) ของเมลเอดิเตอร์

ถ้าข้อความมีขนาดใหญ่กว่าหนึ่งหน้าจอและคุณไม่ได้ตั้งขนาดของเพจสำหรับเทอร์มินัลของคุณโดยใช้คำสั่ง **stty** ข้อความจะเลื่อนจากด้านบนของหน้าจอไปด้านล่าง เพื่อดูเนื้อหาของข้อความขนาดใหญ่ ใช้คำสั่งย่อยของเมลเอดิเตอร์เพื่อดูข้อความด้วยเอดิเตอร์อื่น ถ้าคุณต้องการข้อมูลเพิ่มเติม ดูที่ “การแก้ไขข้อความ” ในหน้า 27

การออกจากเมลเอดิเตอร์:

เพื่อออกจากเมลเอดิเตอร์โดยไม่มีการส่งข้อความ ใช้คำสั่งย่อย **~q** หรือลำดับคีย์อินเตอร์รัปต์ (โดยทั่วไปเป็น Alt-Pause หรือลำดับของคีย์ Ctrl-C)

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. เพื่อแสดงข้อความขณะที่อยู่ในเมลเอดิเตอร์ คุณต้องสตาร์ทเมลเอดิเตอร์แล้ว ถ้าคุณต้องการข้อมูลเพิ่มเติม ดูที่ “การสตาร์ทเมลเอดิเตอร์” ในหน้า 26

ถ้าคุณใส่เท็กซ์ใดๆ คำสั่ง **mail** จะบันทึกข้อความในไฟล์ `dead.letter`

ที่เริ่มต้นของบรรทัดขณะอยู่ในเมลเอดิเตอร์ คุณสามารถใช้คำสั่งย่อย **~q** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
~q	ออกจากเมลเอดิเตอร์และข้อความจะไม่ถูกส่ง ข้อความถูกบันทึกในไฟล์ `dead.letter` ในไดเร็กทอรีหลักของคุณ ยกเว้นคุณไม่ได้ใส่เท็กซ์ใดๆ พร้อมต์ของระบบจะถูกแสดง

Ctrl-C	เพื่อออกจากเอดิเตอร์โดยใช้ลำดับของอินเตอร์รัปต์คีย์ การ break (ลำดับของคีย์ Ctrl-C) หรืออินเตอร์รัปต์ (ลำดับของคีย์ Alt-Pause) ข้อความต่อไปนี้แสดงขึ้น:
	`(Interrupt -- one more to kill letter)`
	กด break หรืออินเตอร์รัปต์อีกครั้ง
	`(Last Interrupt -- letter saved in dead.letter)`
	ข้อความจะไม่ถูกส่ง ข้อความถูกบันทึกในไฟล์ `dead.letter` ในไดเร็กทอรีหลักของคุณ ยกเว้นคุณไม่ได้ใส่เท็กซ์ใดๆ พร้อมต์ของระบบจะถูกแสดง

หมายเหตุ: เมื่อคุณออกจากเมลเอดิเตอร์โดยไม่ส่งข้อความ เนื้อหาก่อนหน้านี้ของไฟล์ `dead.letter` จะถูกแทนที่ด้วยข้อความที่ไม่สมบูรณ์ เพื่อดึงไฟล์ ดูที่ "อ็อพชันสำหรับเพิ่มไฟล์ และระบุข้อความภายในข้อความ"

อ็อพชันสำหรับเพิ่มไฟล์ และระบุข้อความภายในข้อความ:

ข้อกำหนดหลายอย่างต้องผ่านเกณฑ์ก่อนเพิ่มไฟล์ และข้อความเฉพาะภายในเนื้อตวามเมล

ข้อกำหนดเบื้องต้น

1. โปรแกรมเมลต้องถูกติดตั้งไว้ในระบบของคุณ
2. คุณต้องทราบชื่อและแอดเดรสของผู้รับเมล
3. ตัวแก้ไขเมลต้องเริ่มต้นใช้งาน

การรวมไฟล์ในข้อความ:

ใช้คำสั่งย่อย **~r** เพื่อเพิ่มไฟล์เข้ากับข้อความ

ที่เริ่มต้นของบรรทัดขณะอยู่ในเมลเอดิเตอร์ คุณสามารถใช้คำสั่งย่อย **~r** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
~r `schedule`	โดยที่ `schedule` เป็นชื่อของไฟล์เพื่อจะรวม ในตัวอย่างนี้ ข้อมูลในไฟล์ `schedule` จะถูกรวมที่ท้ายของข้อความปัจจุบันที่จะถูกเขียน

การรวมข้อความที่ระบุภายในข้อความ:

ใช้คำสั่งย่อย **~f** หรือ **~m** เพื่อรวมข้อความที่ระบุภายในข้อความของคุณ

ที่เริ่มต้นของบรรทัดใหม่ในเมลเอดิเตอร์ คุณสามารถใช้คำสั่งย่อย **~f** หรือ **~m** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
~f *MessageList*	ต่อท้ายข้อความที่ถูกระบุหรือข้อความเข้ากับท้ายของข้อความปัจจุบัน แต่จะ *ไม่* ย่อหน้าข้อความที่ถูกต่อท้าย นอกจากนี้ ใช้คำสั่งย่อยนี้เพื่อต่อท้ายข้อความสำหรับการอ้างอิงที่มีระยะขอบกว้างเกินกว่าที่ถูกฝังด้วยคำสั่งย่อย **~m** **หมายเหตุ :** พารามิเตอร์ *MessageList* เป็นลิสต์ของจำนวนเต็มที่อ้างถึงหมายเลขข้อความที่ถูกต้องในเมลบ็อกซ์หรือโฟลเดอร์ที่ถูกจัดการโดยเมล คุณยังสามารถใส่ช่วงง่ายๆของจำนวนด้วย ตัวอย่างเช่น: **~f** 1-4 ต่อท้ายข้อความ 1, 2, 3 และ 4 เข้ากับท้ายของข้อความที่ถูกเขียน ข้อความเหล่านี้จะถูกจัดตำแหน่งกับระยะขอบด้านซ้าย (ไม่ถูกย่อหน้า)
~m 2	ต่อท้ายข้อความที่ระบุเข้ากับท้ายของข้อความปัจจุบัน ข้อความ ที่รวมจะถูกย่อหน้าหนึ่งอักขระแท็บจากระยะขอบซ้ายปกติของข้อความ ในตัวอย่างนี้ ข้อความที่ 2 จะถูกต่อท้ายกับข้อความปัจจุบัน
~m 1 3	ต่อท้ายข้อความที่ 1 และจากนั้นข้อความที่ 3 เข้ากับท้ายของข้อความที่ถูกเขียน ย่อหน้าหนึ่งแท็บจากระยะขอบด้านซ้าย

การเพิ่มเนื้อหาของไฟล์ dead.letter เข้ากับข้อความปัจจุบันของคุณ:

ใช้คำสั่งย่อย **~d** เพื่อเพิ่มเนื้อหา `dead.letter` เข้ากับข้อความของคุณ

ที่เริ่มต้นของบรรทัดใหม่ขณะอยู่ในเมลเอดิเตอร์ คุณสามารถใช้คำสั่งย่อย **~d** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
~d	ดึงออกมาหรือต่อท้ายเนื้อหาของไฟล์ `dead.letter` ที่ท้ายของข้อความปัจจุบัน At the พร้อมต์ (`Continue`) จะทำต่อโดยการเพิ่มข้อความหรือโดยการส่งข้อความ

การแก้ไขข้อมูลส่วนหัว:

ส่วนหัวของข้อความประกอบด้วยข้อมูลเส้นทางและข้อความสั้นๆของเรื่อง คุณต้องระบุอย่างน้อยหนึ่งผู้รับของข้อความ

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. สตาร์ทเมลเอดิเตอร์และเริ่มแก้ไขข้อความ สำหรับข้อมูลเพิ่มเติม ดูที่ สตาร์ทเมลเอดิเตอร์

ข้อมูลส่วนหัวที่เหลือไม่ถูกต้องการ ข้อมูลในส่วนหัวสามารถประกอบด้วยต่อไปนี้ :

ไอเท็ม	คำอธิบาย
To:	ประกอบด้วยแอดเดรสหรือผู้รับสำหรับการส่งข้อความ
Subject:	ประกอบด้วยการสรุปแบบสั้นของหัวข้อของข้อความ
Cc:	ประกอบด้วยแอดเดรสหรือผู้รับสำหรับการส่งคัดลอกของข้อความ เนื้อหาของฟิลด์นี้เป็นส่วนของข้อความที่ถูกส่งไปยังผู้ที่ได้รับข้อความทั้งหมด
Bcc:	ประกอบด้วยแอดเดรสหรือผู้รับสำหรับการส่ง *blind* คัดลอกของข้อความ ฟิลด์นี้ *ไม่* ถูกรวมเป็นส่วนของข้อความที่ถูกส่งไปยังผู้รับข้อความทั้งหมด

คุณสามารถปรับโปรแกรมเมลตามความต้องการเพื่อให้ถามถึงข้อมูลในฟิลด์เหล่านี้โดยอัตโนมัติ โดยการใส่ entry ในไฟล์ `.mailrc` สำหรับข้อมูล เพิ่มเติม ให้ดูที่ “อ็อพชันกำหนดลักษณะเฉพาะโปรแกรมเมล” ในหน้า 38

การตั้งค่าหรือรีเซ็ตฟิลด์ Subject::

ใช้คำสั่งย่อย **~s** เพื่อตั้งค่าฟิลด์ **Subject:** เป็นกลุ่มคำหรือประโยคที่ระบุ

ใช้คำสั่งย่อยนี้เพื่อแทนเนื้อหาก่อนหน้านี้ (ถ้ามี) ของฟิลด์ **Subject:** ที่เริ่มต้นของบรรทัดใหม่ขณะอยู่ในเมลเอดิเตอร์ คุณสามารถใช้คำสั่งย่อย **~s** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย

~s Fishing Trip	นี่จะเปลี่ยนฟิลด์ **Subject:** ปัจจุบัน:
	`Subject: Vacation`
	เป็นต่อไปนี้ :
	`Subject: Fishing Trip`
	หมายเหตุ: คุณไม่สามารถต่อท้ายฟิลด์ **Subject:** ด้วยคำสั่งย่อยนี้ ใช้คำสั่งย่อย **~h** ดังที่อธิบายใน “การแก้ไขข้อมูลส่วนหัว” ในหน้า 30

การเพิ่มผู้ใช้เข้ากับฟิลด์ To:, Cc: และ Bcc::

ใช้คำสั่งย่อย ~t, ~c หรือ ~b เพื่อเพิ่มผู้ใช้เข้ากับฟิลด์ส่วนหัว

เริ่มต้นของบรรทัดใหม่ขณะอยู่ในเมลเอดิเตอร์ คุณสามารถใช้คำสั่งย่อย **~t, ~c** หรือ **~b** ดังแสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย

~t geo@austin mel@gtwn	นี่จะเปลี่ยนฟิลด์ **To:** ปัจจุบัน:
	`To: mark@austin`
	เป็นดังต่อไปนี้ :
	`To: mark@austin geo@austin mel@gtwn`

~c geo@austin mel@gtwn	นี่จะเปลี่ยนฟิลด์ **Cc:** ปัจจุบัน:
	`Cc: mark@austin amy`
	เป็นดังต่อไปนี้ :
	`Cc: mark@austin amy geo@austin mel@gtwn`

~b geo@austin mel@gtwn	นี่จะเปลี่ยนฟิลด์ **Bcc:** ปัจจุบัน:
	`Bcc: mark@austin`
	เป็นดังต่อไปนี้ :
	`Bcc: mark@austin geo@austin mel@gtwn`

หมายเหตุ: คุณไม่สามารถใช้คำสั่งย่อย **~t, ~c,** หรือ **~b** เพื่อเปลี่ยนหรือลบเนื้อหาของฟิลด์ **To:, Cc:** และ **Bcc:** ใช้คำสั่งย่อย **~h** ดังที่อธิบายใน “การแก้ไขข้อมูลส่วนหัว” ในหน้า 30

จัดรูปแบบข้อความใหม่ในเมลเอดิเตอร์:

หลังจากพิมพ์ข้อความและก่อนที่จะส่งมัน คุณสามารถจัดรูปแบบข้อความใหม่เพื่อปรับปรุงการแสดงของมันโดยการใช้โปรแกรมเชลล์ **fmt**

ก่อนที่จะจัดรูปแบบข้อความใหม่ ต้องแน่ใจว่าเงื่อนไขต่อไปนี้เป็นจริง :

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. คำสั่ง **fmt** ต้องถูกติดตั้งบนระบบของคุณ

ที่เริ่มต้นของบรรทัดใหม่ขณะที่อยู่ในเมลเอดิเตอร์ คุณสามารถใช้ใช้คำสั่ง **fmt** ดังแสดงในตำแหน่งต่อไปนี้ :

ไอเท็ม	คำอธิบาย
~! fmt	เปลี่ยนการแสดงของข้อความโดยการเปลี่ยนการโฟลว์ของข้อมูลสำหรับแต่ละพารากราฟภายในระยะขอบที่กำหนด (บรรทัดว่างต้องแยกแต่ละพารากราฟ) คำสั่งย่อยไพพ์ (!) จะไพพ์ข้อความไปยังอินพุตมาตรฐานของคำสั่งและแทนที่ข้อความด้วยเอาต์พุตมาตรฐานจากคำสั่งนั้น

หมายเหตุ: ห้ามใช้คำสั่ง **fmt** ถ้าข้อความประกอบด้วยข้อความที่ฝังอยู่หรือข้อมูลที่จัดรูปแบบล่วงหน้าจากไฟล์ภายนอก คำสั่ง **fmt** จะจัดรูปแบบข้อมูลส่วนหัวในข้อความที่ฝังอยู่ใหม่ และอาจเปลี่ยนรูปแบบของข้อมูลที่จัดรูปแบบไว้ล่วงหน้า ใช้คำสั่งย่อย **~e** หรือ **~v** แทนเพื่อเข้าสู่เอดิเตอร์แบบเต็มจอและจัดรูปแบบของข้อความใหม่

การตรวจสอบการสะกดคำผิดในเมลเอดิเตอร์:

คำสั่ง **spell** จะตรวจสอบการสะกดคำในข้อความของคุณ

ก่อนที่จะตรวจสอบการสะกดคำผิดในข้อความ ต้องแน่ใจว่าเงื่อนไขต่อไปนี้เป็นจริง :

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. โปรแกรมการจัดรูปแบบของเท็กซ์ต้องถูกติดตั้งบนระบบของคุณ

ใช้คำสั่ง **spell** เพื่อตรวจสอบคำที่สะกดผิดในข้อความของคุณ จากเมลเอดิเตอร์ :

1. เขียนข้อความไปยังไฟล์ชั่วคราว ตัวอย่างเช่น เพื่อเขียนข้อความไปยังไฟล์ `checkit` พิมพ์ :

   ```
   ~w checkit
   ```

2. รันคำสั่ง **spell** โดยใช้ไฟล์ชั่วคราวเป็นอินพุต พิมพ์:

   ```
   ~! spell checkit
   ```

   ในตัวอย่างนี้ เครื่องหมายตกใจ (!) เป็นคำสั่งย่อยที่สตาร์ทเชลล์ รันคำสั่ง และกลับไปยังเมลบ็อกซ์ คำสั่ง **spell** จะตอบสนองกับลิสต์ของคำที่ไม่อยู่ในลิสต์คำที่รู้จักของมัน ตามด้วยเครื่องหมายตกใจ (!) เพื่อบอกว่าคุณกลับมายังโปรแกรมเมล
3. ตรวจสอบลิสต์ของคำ ดูว่าคุณต้องใช้เอดิเตอร์เพื่อแก้ไขหรือไม่
4. พิมพ์ต่อไปนี้เพื่อลบไฟล์ชั่วคราว :

   ```
   ~! rm checkit
   ```

การส่งเมล:

ใช้โพรซีเดอร์นี้เพื่อส่งข้อความหลังจากคุณสร้างมัน

- โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
- คุณต้องทราบชื่อและแอดเดรสของผู้รับเมล

1. ใส่คำสั่ง **mail** บนบรรทัดรับคำสั่ง ตามด้วยชื่อและแอดเดรสของผู้รับของข้อความ ตัวอย่างเช่น:

   ```
   >mail jan@brown
   ```

   ระบบตอบกลับ ด้วย:

   ```
   Subject:
   ```

2. พิมพ์ชื่อเรื่องของข้อความ ตัวอย่าง เช่น:

   ```
   Subject: Dept Meeting
   ```

และกด Enter ตอนนี้คุณสามารถพิมพ์เนื้อความได้

3. พิมพ์ข้อความของคุณ ตัวอย่าง เช่น:

```
There will be a short department meeting this afternoon
in my office. Please plan on attending.
```

4. เพื่อส่งข้อความที่คุณพิมพ์ด้วยเมลเอดิเตอร์ กดอักขระ end-of-text ซึ่งโดยทั่วไปเป็นลำดับของคีย์ Ctrl-D หรือจุด .) ขณะอยู่ที่จุดเริ่มต้นของบรรทัดใหม่ในข้อความ

   ระบบจะแสดงฟิลด์ **Cc:** :

```
Cc:
```

5. พิมพ์ชื่อและแอดเดรสของผู้ใช้เหล่านั้นที่จะรับคัดลอกของข้อความ ตัวอย่างเช่น:

```
Cc: karen@hobo cliff@cross
```

   หมายเหตุ : ถ้าคุณไม่ต้องการส่งคัดลอก กด Enter โดยไม่ต้องพิมพ์

เมื่อคุณกดคีย์ Enter ข้อความจะถูกส่งไปยังแอดเดรสที่ระบุ

หมายเหตุ: ถ้าคุณใส่แอดเดรสที่ระบบไม่รู้จัก หรือไม่ถูกระบบใน alias หรือลิสต์การกระจาย ระบบจะตอบสนองด้วยชื่อล็อกอินตามด้วยข้อความแสดงข้อผิดพลาด : `[user ID]... User unknown`

การตอบเมล:

ที่พร้อมต์ของเมลบ็อกซ์ คุณสามารถใช้คำสั่งย่อย **r** และ **R** เพื่อตอบเมลตามแสดงในตัวอย่างต่อไปนี้

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. ต้องเป็นเมลในระบบเมลบ็อกซ์ของคุณ

ไอเท็ม	คำอธิบาย
r	สร้างข้อความใหม่ที่ถูกกำหนดแอดเดรสไปยังผู้ส่งของข้อความที่เลือก และคัดลอกบุคคลในฟิลด์ **Cc:** (ถ้ามี) ฟิลด์ **Subject:** ของข้อความใหม่จะอ้างถึงข้อความที่เลือก ค่าดีฟอลต์ของคำสั่งย่อย **r** จะเป็นข้อความปัจจุบัน ค่าดีฟอลต์นี้สามารถถูกทับโดยการพิมพ์หมายเลขของข้อความหลัง **r**
R	สตาร์ทการตอบเฉพาะไปยังผู้ส่งของข้อความ ค่าดีฟอลต์ของคำสั่งย่อย **R** คือข้อความปัจจุบัน

R 4	สตาร์ทการตอบเฉพาะไปยังผู้ส่งของข้อความ คุณสามารถทับข้อความปัจจุบันโดยการพิมพ์หมายเลขข้อความหลัง **R** ตัวอย่างนี้จะเริ่มการตอบกับข้อความ 4 ระบบจะตอบสนองด้วยข้อความเหมือนดังต่อไปนี้ :
	`To: karen@thor`
	`Subject: Re: Department Meeting`
	
	จากนั้นคุณสามารถพิมพ์การตอบสนองของคุณ :
	`I'll be there.`
	
	เมื่อคุณเสร็จสิ้น การพิมพ์ข้อความ ให้กดจุด (.) หรือกดปุ่ม Ctrl-D เพื่อส่งข้อความ หลังจากการตอบถูกส่ง คุณจะกลับมาที่พร้อมต์ของเมลบ็อกซ์

การสร้างข้อความใหม่ขณะอยู่ในเมลบ็อกซ์:

ที่พร้อมต์ของเมลบ็อกซ์ คุณสามารถใช้คำสั่งย่อย **m** ดังแสดงในตัวอย่างต่อไปนี้เพื่อสร้างข้อความใหม่

ไอเท็ม	คำอธิบาย
m *Address*	พารามิเตอร์ *Address* เป็นแอดเดรสของผู้ใช้ที่ถูกต้อง คำสั่งย่อยนี้จะสตาร์ทเมลเอดิเตอร์และให้คุณสามารถสร้างข้อความใหม่ขณะที่อยู่ในเมลบ็อกซ์ เมื่อคุณส่งข้อความ คุณจะกลับไปยังพร้อมต์ของเมลบ็อกซ์

การฟอร์เวิร์ดเมล:

ขณะที่คุณอ่านเมล คุณอาจต้องการฟอร์เวิร์ดบันทึกที่ระบุไปยังผู้ใช้อื่น

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. ถ้าฟอร์เวิร์ดข้อความที่เลือก สตาร์ทเครื่องมือของเมลด้วยคำสั่ง **mail** จดหมายเลขของเมลที่คุณต้องการฟอร์เวิร์ด

งานนี้สามารถทำได้โดยใช้คำสั่งย่อย **~f** และ **~m**

ถ้าคุณอยู่ห่างจากเน็ตเวิร์กแอดเดรสปกติของคุณ คุณสามารถให้เมลของคุณส่งไปยังเน็ตเวิร์กแอดเดรสอื่นโดยการสร้างไฟล์ `.forward` โปรดดู “ไฟล์ .forward” ในหน้า 35 แอดเดรสใหม่สามารถเป็นเมลแอดเดรสที่ถูกต้องใดๆบนเน็ตเวิร์กของคุณ หรือบนเน็ตเวิร์กที่เชื่อมต่อกับเน็ตเวิร์กของคุณ มันสามารถเป็นแอดเดรสของผู้ร่วมงานที่จะจัดการข้อความของคุณขณะที่คุณไม่อยู่ เมื่อคุณเลือกที่จะฟอร์เวิร์ดเน็ตเวิร์กเมลของคุณ คุณจะไม่ได้รับคัดลอกของเมลที่เข้ามาในเมลบ็อกซ์ของคุณ เมลทั้งหมดจะถูกฟอร์เวิร์ดโดยตรงไปยังแอดเดรสหรือผู้รับที่คุณระบุ

การฟอร์เวิร์ดข้อความที่เลือกจากภายในเมลบ็อกซ์:

ใช้โพรซีเดอร์นี้เพื่อฟอร์เวิร์ดข้อความเมลที่ระบุภายในเมลบ็อกซ์

เพื่อฟอร์เวิร์ดข้อความเมลที่เลือก :

1. สร้างข้อความใหม่โดยใช้คำสั่งย่อย **m** และระบุผู้รับโดยพิมพ์ต่อไปนี้ที่พร้อมต์ของเมลบ็อกซ์ :

   ```
   m User@Host
   ```

   โดยที่ *User* อ้างถึงชื่อล็อกอินของผู้ใช้อื่น และ *Host* เป็นชื่อของระบบของผู้ใช้ ถ้าผู้ใช้อยู่บยระบบของคุณ คุณสามารถตัดส่วน `@Host` ของแอดเดรส
2. พิมพ์ชื่อเรื่องที่พร้อมต์ **Subject:**
3. เพื่อระบุจำนวนของข้อความเมลที่จะถูกฟอร์เวิร์ด พิมพ์ :

   ```
   ~f MessageNumber
   ```

   OR

   ```
   ~m MessageNumber
   ```

   MessageNumber จะระบุชิ้นของเมลที่จะฟอร์เวิร์ด

   คำสั่ง **mail** จะแสดงข้อความเหมือนดังต่อไปนี้ :

   ```
   Interpolating: 1
   (continue)
   ```
4. เพื่อออกจากเมล ให้พิมพ์จุด (`.`) บนบรรทัดว่าง ที่พร้อมต์ **Cc:** พิมพ์ชื่อเพิ่มเติมกับคนที่คุณต้องการฟอร์เวิร์ดข้อความเมล

การฟอร์เวิร์ดเมลทั้งหมด:

ใช้โพรซีเดอร์นี้เพื่อฟอร์เวิร์ดเมลทั้งหมดของคุณไปยังบุคคลอื่น

เพื่อฟอร์เวิร์ดเมลทั้งหมดของคุณไปยังบุคคลอื่น :

1. ใส่คำสั่ง **cd** โดยไม่มีพารามิเตอร์เพื่อให้แน่ใจว่าคุณอยู่ในไดเร็กทอรีหลักของคุณ ตัวอย่างเช่น พิมพ์ต่อไปนี้สำหรับชื่อล็อกอิน `mary`:

   ```
   cd
   pwd
   ```

   ระบบตอบกลับ ด้วย:

   ```
   /home/mary
   ```

2. สร้างไฟล์ `.forward` ในไดเร็กทอรีหลักของคุณ โปรดดู "ไฟล์ .forward"

   หมายเหตุ : คุณจะไม่ได้รับเมลใดๆจนกว่าคุณจะลบไฟล์ `.forward`

ไฟล์ .forward:

ไฟล์ `.forward` ประกอบด้วยเน็ตเวิร์กแอดเดรสหรือแอดเดรสที่จะรับเน็ตเวิร์กเมลที่ถูกฟอร์เวิร์ดของคุณ

แอดเดรสต้องมีรูปแบบเป็น *User@Host User* อ้างถึงชื่อล็อกอินของผู้ใช้อื่น และ *Host* เป็นชื่อของระบบของผู้ใช้ ถ้าผู้ใช้อยู่บยระบบของคุณ คุณสามารถตัดส่วน `@`*`Host`* ของแอดเดรส คุณสามารถใช้คำสั่ง **cat** เพื่อสร้างไฟล์ `.forward` ดังต่อไปนี้ :

```
cat > .forward
mark
joe@saturn
[END OF FILE]
```

`[END OF FILE]` จะแทนอักขระสิ้นสุดของไฟล์ ซึ่งคือลำดับของคีย์ Ctrl–D บนเทอร์มินัลส่วนใหญ่ ซึ่งควรถูกพิมพ์บนบรรทัดว่างๆ

ไฟล์ `.forward` ประกอบด้วยแอดเดรสของผู้ใช้ที่คุณต้องการให้เมลของคุณฟอร์เวิร์ดไป เมลของคุณจะถูกฟอร์เวิร์ดไปยัง `mark` บนระบบโลคัลของคุณ และ `joe` บนระบบ `saturn`

ไฟล์นี้ต้องประกอบด้วยแอดเดรสที่ถูกต้อง ถ้ามันเป็น null ไฟล์ (ความยาวเป็นศูนย์) เมลของคุณจะไม่ถูกฟอร์เวิร์ดและจะถูกเก็บในเมลบ็อกซ์ของคุณ

หมายเหตุ : คุณจะไม่ได้รับเมลใดๆจนกว่าคุณจะลบไฟล์ `.forward`

การยกเลิกเมลที่ถูกฟอร์เวิร์ด:

เพื่อหยุดการฟอร์เวิร์ดเมล ลบไฟล์ `.forward` ดังต่อไปนี้

ใช้คำสั่ง **rm** เพื่อลบไฟล์ `.forward` จากไดเร็กทอรีหลักของคุณ

```
rm .forward
```

การส่งประกาศข้อความวันหยุด:

ใช้ขั้นตอนนี้เพื่อจัดเตรียม และส่งประกาศข้อความวันหยุด

โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ

1. เตรียมข้อมูลเบื้องต้นข้อความวันหยุดในไดเร็กทอรี $HOME (ล็อกอิน) โดยการพิมพ์:

   ```
   vacation
   -I
   ```

   นี่จะสร้างไฟล์ .vacation.dir และไฟล์ .vacation.pag โดยที่ชื่อของคนที่ส่งข้อความจะถูกเก็บ
2. แก้ไขไฟล์ .forward ตัวอย่างเช่น carl พิมพ์ข้อความต่อไปนี้ในไฟล์ .forward:

   ```
   carl, |"/usr/bin/vacation carl"
   ```

   entry แรก carl เป็นชื่อผู้ใช้ที่เมลจะถูกฟอร์เวิร์ด entry ที่สอง carl เป็นชื่อผู้ใช้ของผู้ส่งข้อความวันหยุด ผู้ส่งของข้อความเมลจะได้รับข้อความวันหยุดหนึ่งข้อความจาก carl ต่อหนึ่งอาทิตย์ โดยไม่สนใจว่าจะมีกี่ข้อความที่ส่งถึง carl จากผู้ส่ง ถ้าคุณมีเมลที่ถูกฟอร์เวิร์ดไปยังผู้อื่น ข้อความเมลจากผู้ส่งจะถูกฟอร์เวิร์ดไปยังบุคคลที่ถูกกำหนดในไฟล์ .forward

   ใช้แฟล็ก **–f** เพื่อเปลี่ยนช่วงความถี่ที่จะส่งข้อความ ตัวอย่างเช่น carl พิมพ์ข้อความต่อไปนี้ในไฟล์ .forward:

   ```
   carl, |"/usr/bin/vacation -f10d carl"
   ```

   ผู้ส่งของข้อความเมลจะได้รับข้อความวันหยุดหนึ่งข้อความจาก carl ทุก 10 วัน โดยไม่สนใจว่าจะมีกี่ข้อความที่ส่งถึง carl จากผู้ส่ง
3. เพื่อส่งข้อความไปยังแต่ละคนที่ส่งเมลถึงคุณ สร้างไฟล์ $HOME/.vacation.msg และเพิ่มข้อความของคุณในไฟล์นี้ ต่อไปนี้คือตัวอย่างของข้อความแจ้งวันหยุดพักผ่อน:

   ```
   From: carl@odin.austin (Carl Jones)
   Subject: I am on vacation.
   I am on vacation until October 1. If you have something urgent,
   please contact Jim Terry <terry@zeus.valhalla>.
     --carl
   ```

   ผู้ส่งได้รับข้อความที่ อยู่ในไฟล์ $HOME/.vacation.msg หรือหากไฟล์ไม่มีอยู่ จะได้รับข้อความดีฟอลต์ที่พบในไฟล์ /usr/share/lib/vacation.def หากไม่มีไฟล์ทั้งสองนี้อยู่ จะไม่ส่งการตอบกลับอัตโนมัติ ไปยังผู้ส่งข้อความเมลและไม่มีการสร้างข้อความแสดงความผิดพลาด

ในการยกเลิกข้อความวันหยุด ให้ลบไฟล์ .forward ไฟล์ .vacation.dir ไฟล์ .vacation.pag และไฟล์ .vacation.msg จากไดเร็กทอรี $HOME (ล็อกอิน) ของคุณ:

```
rm .forward .vacation.dir .vacation.pag .vacation.msg
```

การส่งและการรับเมลความลับ:

เพื่อสางเมลความลับ ที่พร้อมต์ของบรรทัดรับคำสั่งของระบบ ใช้คำสั่ง **xsend** ในวิธีที่แสดงในตัวอย่างต่อไปนี้

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. รหัสผ่านต้องถูกตั้งโดยใช้คำสั่ง **enroll**

ไอเท็ม	คำอธิบาย
xsend barbara	ในตัวอย่างนี้ เมลความลับจะถูกกำหนดแอดเดรสให้กับชื่อล็อกอิน barbara เมื่อคุณกด Enter เอดิเตอร์แบบบรรทัดเดียวจะถูกใช้เพื่อพิมพ์เท็กซ์ของข้อความ เมื่อคุณเสร็จสิ้นการพิมพ์ข้อความ ให้กดลำดับคีย์ Ctrl–D หรือจุด (.) เพื่อออกเอดิเตอร์เมล และส่งข้อความ คำสั่ง **xsend** เข้ารหัสข้อความก่อนที่จะส่ง

1. เพื่อรับเมลความลับ ที่พร้อมต์ของบรรทัดรับคำสั่งของระบบของคุณ พิมพ์ :

   ```
   mail
   ```

   ระบบจะแสดงลิสต์ของข้อความในระบบเมลบ็อกซ์ของคุณ โปรแกรมเมลความลับจะส่งการแจ้งให้คุณว่าคุณได้รับเมลความลับ บรรทัดของข้อความจะเหมือนดังต่อไปนี้ :

   ```
   Mail [5.2 UCB] Type ? สำหรับความช่วยเหลือ
   "/usr/spool/mail/linda": 4 messages 4 new
   >N 1 robert Wed Apr 14 15:23 4/182 "secret mail from robert@Zeus"
   ```

   ข้อความเท็กซ์จะบอกให้คุณอ่านเมลความลับของคุณบนโฮสต์ของคุณโดยใช้คำสั่ง **xget**

2. ที่พร้อมต์ของบรรทัดรับคำสั่งของระบบ พิมพ์ :

   ```
   xget
   ```

   คุณจะได้รับพร้อมต์สำหรับรหัสผ่านที่ถูกตั้งไว้ล่วงหน้าโดยใช้คำสั่ง **enroll** หลังจากคุณพิมพ์รหัสผ่านของคุณ จุดรับคำสั่ง **xget** จะถูกแสดง ตามด้วยลิสต์ของเมลความลับ โปรแกรมเมลจะถูกใช้เพื่อแสดงเมลความลับ คุณต้องใส่คำสั่งย่อย **q** ถ้าคุณต้องการปล่อยให้ข้อความที่อ่านแล้วและยังไม่ได้อ่านอยู่ในเมลบ็อกซ์ความลับ และป้องกันไม่ให้คำสั่ง **xget** ลบข้อความ

ข้อมูลความช่วยเหลือสำหรับเมล

คุณสามารถได้รับข้อมูลความช่วยเหลือเกี่ยวกับการใช้โปรแกรมเมลโดยใช้คำสั่ง **?**, **man** หรือ **info**

ไอเท็ม	คำอธิบาย
เพื่อให้ได้รับความช่วยเหลือในเมลบ็อกซ์	ใส่ ? หรือ help ที่พร้อมต์ของเมลบ็อกซ์ คำสั่งย่อย **?** และคำสั่งย่อย **help** จะแสดงสรุปของคำสั่งย่อยทั่วไปของเมลบ็อกซ์ คุณสามารถแสดงลิสต์ของคำสั่งย่อยเมลบ็อกซ์ (โดยไม่สรุป) โดยการใส่คำสั่งย่อย **(1)**ist
เพื่อให้ได้รับความช่วยเหลือในเมลเอดิเตอร์	พิมพ์ **~?** ที่พร้อมต์ของเมลเอดิเตอร์ คำสั่งย่อย **~?** จะแสดงสรุปของคำสั่งย่อยทั่วไปของเมลเอดิเตอร์
เพื่อให้ได้รับความช่วยเหลือในเมลที่เป็นความลับ	พิมพ์ **?** ที่พร้อมต์ของเมลเอดิเตอร์ คำสั่งย่อย **?** คำสั่งย่อยจะแสดงสรุปของคำสั่งย่อยทั่วไปของเมลที่เป็นความลับ
เพื่อให้ได้รับความช่วยเหลือในการใช้เพจแบบแมนวล	พิมพ์ man mail ที่พร้อมต์ของบรรทัดรับคำสั่งของระบบ ในตัวอย่างนี้ mail เป็นชื่อคำสั่งที่ถูกค้นหา ระบบจะให้เอกสาร ASCII ที่เกี่ยวกับคำสั่ง **mail** ที่เครื่องหมายต่อเนื่อง (:) กด Enter เพื่อดูเอกสารที่เหลือ คำสั่ง **man** ให้ข้อมูล ในรูปแบบ ASCII สำหรับหัวข้ออ้างอิงเกี่ยวกับคำสั่ง รูทีนย่อย และไฟล์

อ็อพชันกำหนดลักษณะเฉพาะโปรแกรมเมล

คำสั่งและอ็อพชันในไฟล์ .mailrc และ /usr/share/lib/Mail.rc สามารถกำหนดลักษณะเฉพาะให้เหมาะกับความต้องการใช้งานเมลของคุณ

โปรดดู "อ็อพชันการเปิดใช้งานและปิดใช้งานเมล"สำหรับข้อมูล เกี่ยวกับอ็อพชันเมล

ลักษณะเฉพาะของเซสชันเมลที่คุณสามารถกำหนดลักษณะเฉพาะประกอบด้วย:

- **พร้อมต์สำหรับหัวเรื่องของข้อความ** เมื่อคุณป้อนคำสั่ง **mail** โปรแกรมจะขอให้คุณกรอกฟิลด์ **หัวเรื่อง:** ให้สมบูรณ์ เมื่อพร้อมนี้แสดงผล คุณสามารถกรอกสรุปเรื่องที่เกี่ยวข้องกับข้อความ ซึ่งสรุปผลดังกล่าวจะแสดงที่บรรทัดเริ่มต้นของข้อความเมื่อผู้รับ อ่านข้อความ โปรดดู "พร้อมต์ฟิลด์ หัวเรื่อง: และสำเนา (Cc:)" ในหน้า 40
- **พร้อมต์สำหรับผู้ใช้เพื่อขอสำเนาของข้อความ** คุณสามารถกำหนดลักษณะเฉพาะ ของไฟล์ .mailrc ดังนั้นเมื่อคุณส่งข้อความ โปรแกรมเมลจะแสดงพร้อมต์สำหรับชื่อของผู้ใช้อื่นที่ควรได้รับ สำเนาของข้อความ โปรดดู "พร้อมต์ฟิลด์ หัวเรื่อง: และสำเนา (Cc:)" ในหน้า 40
- **ชื่อย่อหรือรายการแจกจ่าย** ถ้าคุณส่งเมลในเครือข่ายขนาดใหญ่ หรือส่งข้อความเดิมบ่อยๆ ให้แก่กลุ่มผู้ใช้จำนวนมาก การพิมพ์แอดเดรสยาวๆ สำหรับผู้รับแต่ละคนเป็นเรื่องยุ่งยาก เมื่อต้องการทำให้ขั้นตอนนี้ง่ายขึ้น ให้สร้างชื่อย่อหรือ รายการแจกจ่ายในไฟล์ .mailrc ของคุณ *ชื่อย่อ* คือชื่อที่คุณกำหนดที่สามารถใช้วางเป็นแอดเดรสผู้ใช้เดี่ยว *รายการแจกจ่าย* คือชื่อที่คุณกำหนดที่สามารถใช้วางเป็นกลุ่มของแอดเดรสผู้ใช้ โปรดดู "Alias และดีสทริบิวชันลิสต์" ในหน้า 40
- **จำนวนบรรทัดที่แสดงผลเมื่ออ่านข้อความ** คุณสามารถเปลี่ยน จำนวนบรรทัดของส่วนหัวข้อความ หรือของข้อความที่เลื่อนผ่านจอแสดงผล โปรดดู "เปลี่ยนแปลงจำนวนส่วนหัวข้อความ หรือบรรทัดของเนื้อควาที่แสดงผล ในโปรแกรมเมล" ในหน้า 41
- **ข้อมูลที่แสดงในข้อความ** คุณสามารถปิดส่วนหัวของข้อความ เช่น ฟิลด์ machine-set message-id โปรดดู "การแสดงข้อมูลในข้อความ" ในหน้า 43
- **โฟลเดอร์ไดเร็กทอรีที่เก็บข้อความ** คุณสามารถสร้างไดเร็กทอรีพิเศษ สำหรับเก็บข้อความได้ คุณสามารถใช้คำสั่งย่อย เครื่องหมายบวก (+) เพื่อกำหนดไดเร็กทอรีเมื่อเก็บข้อความหรือดูโฟลเดอร์ โปรดดู "การสร้างดีฟอลต์โฟลเดอร์เพื่อเก็บข้อความ" ในหน้า 45
- **ไฟล์ล็อกสำหรับบันทึกข้อความขาออก** คุณสามารถสั่งให้โปรแกรม**เมล** บันทึกข้อความขาออกทั้งหมดในไฟล์ หรือไดเร็กทอรีย่อยในไดเร็กทอรีโฮมของคุณ โปรดดู "การสร้างดีฟอลต์โฟลเดอร์เพื่อเก็บข้อความ" ในหน้า 45
- **ตัวแก้ไขสำหรับพิมพ์ข้อความ** นอกจากตัวแก้ไขเมล คุณสามารถเลือกตัวแก้ไขอีก 2 โปรแกรมที่ใช้แก้ไขข้อความ โปรดดู "เท็กซ์เอดิเตอร์สำหรับการพิมพ์ข้อความ" ในหน้า 46

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการกำหนดลักษณะโปรแกรมเมล โปรดดูหัวข้อ ต่อไปนี้

อ็อพชันการเปิดใช้งานและปิดใช้งานเมล:

อ็อพชันสามารถเป็นทั้งไบนารีหรือค่า

อ็อพชันไบนารีเป็นทั้ง **set** หรือ **unset** ขณะที่อ็อพชัน valued สามารถเป็น **set** เพื่อระบุค่า

> **หมายเหตุ :** รูปแบบอ็อพชัน **unset** จะเหมือนกับอ็อพชัน **set no**

ใช้คำสั่ง **pg** เพื่อดูไฟล์ /usr/share/lib/Mail.rc เนื้อหาของไฟล์ /usr/share/lib/Mail.rc จะกำหนดการตั้งค่าของโปรแกรมเมล เปลี่ยนการตั้งค่าระบบสำหรับโปรแกรมเมลของคุณโดยการสร้างไฟล์ $HOME/.mailrc เมื่อคุณรันคำสั่ง **mail** คำสั่งย่อยในไฟล์ **.mailrc** จะทับคำสั่งย่อยที่เหมือนกันในไฟล์ /usr/share/lib/Mail.rc อ็อพชัน .mailrc สามารถถูกปรับตามความต้องการและใช้ได้แต่ละครั้งที่คุณใช้โปรแกรมเมล

เพื่อเรียกใช้งานคำสั่งเมลที่ถูกเก็บในไฟล์ ใช้คำสั่งย่อย **source**

ข้อกำหนดเบื้องต้น

โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ

การเปิดใช้งานอ็อพชันเมล:

คำสั่งย่อยเมลบ็อกซ์เหล่านี้ถูกใช้โดยทั่วไปเพื่อเปลี่ยนคุณลักษณะของของเมลเซสชัน

ไอเท็ม	คำอธิบาย
set	เปิดใช้งานอ็อพชันเมล
source	เปิดใช้งานอ็อพชันเมลที่ถูกเก็บในไฟล์ เมื่ออ่านเมล คุณสามารถใช้คำสั่งย่อยนี้ที่พร้อมต์ของเมลบ็อกซ์ : `source` *PathName* โดยที่ *PathName* เป็นพาธและไฟล์ที่ประกอบด้วยคำสั่งเมล คำสั้งในไฟล์นี้จะทับการตั้งค่าของคำสั่งที่เหมือนกันก่อนหน้านี้สำหรับช่วงเวลาของเซสชันปัจจุบัน คุณยังสามารถเปลี่ยนคุณลักษณะของเซสชันเมลปัจจุบันโดยการพิมพ์คำสั่งที่พร้อมต์ของเมลบ็อกซ์

คุณสามารถตั้งอ็อพชันเหล่านี้ขณะอยู่ในเมลบ็อกซ์หรือโดยการสร้าง entry ในไฟล์ .mailrc

การดูอ็อพชันเมลที่ถูกเปิดใช้งาน:

เมื่ออ่านเมลของคุณ ใส่คำสั่งย่อย **set** ดดยไม่มีอาร์กิวเมนต์ใดๆเพื่อลิสต์อ็อพชัน .mailrc ทั้งหมดที่ถูกเปิดใช้งาน

ในลิสต์นี้ คุณยังสามารถเห็นว่าถ้าโฟลเดอร์ไดเร็กทอรีถูกเลือกและถ้าล็อกไฟล์ถูกตั้งค่าเพื่อบันทึกข้อความขาออก

ที่พร้อมต์ของเมลบ็อกซ์ พิมพ์:

```
set
```

ข้อความจะคล้ายกับที่แสดงต่อไปนี้:

```
ask
metoo
toplines 10
```

ในตัวอย่างนี้ 2 ไบนารีอ็อพชันจะถูกเปิดใช้งาน: **ask** และ **metoo** จะไม่มี entry **askcc** ในลิสต์ นี่จะระบุว่าอ็อพชัน **askcc** ไม่ถูกเปิดใช้งาน อ็อพชัน **toplines** ถูกกำหนดค่าเป็น 10 อ็อพชัน **ask, metoo, askcc** และ **toplines** ถูกอธิบายในส่วนของ รูปแบบของไฟล์ .mailrc ของ *การอ้างอิงไฟล์*

การปิดใช้งานอ็อพชันเมล:

คำสั่งย่อยเมลบ็อกซ์เหล่านี้ถูกใช้โดยทั่วไปเพื่อเปลี่ยนคุณลักษณะของของเมลเซสชัน

ไอเท็ม	คำอธิบาย
unset	ปิดใช้งานอ็อพชันเมล
unalias	ลบชื่อ alias ที่ระบุ
ignore	ยกเลิกฟิลด์ส่วนหัวส่วนหัวของข้อความ

คุณสามารถตั้งอ็อพชันเหล่านี้ขณะอยู่ในเมลบ็อกซ์หรือโดยการสร้าง entry ในไฟล์ .mailrc

หมายเหตุ: รูปแบบอ็อพชัน **unset** จะเหมือนกับอ็อพชัน **set no**

พร้อมต์ฟิลด์ หัวเรื่อง: และสำเนา (Cc:):

เมื่อพร้อมต์ฟิลด์ **หัวเรื่อง:** และ **Cc:** ถูกแก้ไข ข้อกำหนดเบื้องต้นต่อไปนี้ต้องผ่านเกณฑ์

ข้อกำหนดเบื้องต้น

โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ

เปิดใช้งานหรือปิดใช้งานการพร้อมต์ของฟิลด์ Subject ::

ใช้คำสั่ง **set** และ **unset** เพื่อเปิดใช้งานและปิดใช้งานฟิลด์ **Subject:**

คุณสามารถเปิดใช้งานหรือปิดใช้งานฟิลด์ **Subject:** ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
set ask	การพร้อมต์ของฟิลด์ **Subject:** จะถูกเปิดใช้งานโดยการแก้ไขไฟล์ .mailrc อ็อพชัน ask
unset ask	การพร้อมต์ของฟิลด์ **Subject:** จะถูกปิดใช้งานโดยการแก้ไขไฟล์ .mailrc อ็อพชัน ask

การเปิดใช้งานหรือปิดใช้งานการพร้อมต์ของฟิลด์ Carbon Copy (Cc:):

ใช้คำสั่ง **set** และ **unset** เพื่อเปิดใช้งานและปิดใช้งานฟิลด์ **Cc:**

คุณสามารถเปิดใช้งานหรือปิดใช้งานฟิลด์ **Cc:** ในวิธีที่แสดงในตัวอย่างต่อไปนี้ :

ไอเท็ม	คำอธิบาย
set askcc	การพร้อมต์ของฟิลด์ Carbon copy (**Cc:**) จะถูกเปิดใช้งานโดยการแก้ไขไฟล์ .mailrcอ็อพชัน **askcc**
unset askcc	การพร้อมต์ของฟิลด์ Carbon copy (**Cc:**) จะถูกปิดใช้งานโดยการแก้ไขไฟล์ .mailrcอ็อพชัน **askcc**

Alias และดีสทริบิวชันลิสต์:

โดยการสร้าง alias และดีสทริบิวชันลิสต์ คุณสามารถจัดการผู้รับจดหมายและแอดเดรสที่คุณใช้ทั่วไปได้อย่างง่ายดาย

ก่อนที่จะสร้าง alias หรือดีสทริบิวชันลิสต์ ต้องแน่ใจว่าเงื่อนไขต่อไปนี้เป็นจริง :

1. โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ
2. คุณต้องรู้ชื่อและแอดเดรสของผู้ใช้ที่คุณต้องการรวมใน alias หรือดีสทริบิวชันลิสต์ของคุณ

คุณสามารถ alias หรือดิสทริบิวชันลิสต์ในวิธีต่อไปนี้ :

ไอเท็ม	คำอธิบาย

alias	`kath kathleen@gtwn`
	
	ในตัวอย่างนี้ alias `kath` ถูกลิสต์สำหรับผู้ใช้ `kathleen` ที่แอดเดรส `gtwn` หลังจากที่คุณเพิ่มบรรทัดนี้เข้ากับไฟล์ `$HOME/.mailrc` ของคุณ เพื่อส่งข้อความไปถึงแคทเทอลีน พิมพ์ต่อไปนี้ที่พร้อมต์บรรทัดรับคำสั่ง :
	`mail kath`
	
	ตอนนี้คุณสามารถส่งเมลไปถึงแคทเทอลีนโดยใช้ alias `kath`

alias	`dept dee@merlin anne@anchor jerry@zeus bill carl`
	
	หลังจากที่คุณเพิ่มบรรทัดนี้เข้ากับไฟล์ `$HOME/.mailrc` ของคุณ เพื่อส่งข้อความไปยังแผนกของคุณ พิมพ์ต่อไปนี้ที่พร้อมต์บรรทัดรับคำสั่ง :
	`mail dept`
	
	ข้อความที่คุณสร้างและส่งจะไปยัง `dee` บนระบบ `merlin`, `anne` บนระบบ `anchor`, `jerry` บนระบบ`zeus` และไปยัง `bill` และ `carl` บนระบบโลคัล

เพื่อลิสต์ alias และดีสทริบิวชันลิสต์ พิมพ์ต่อไปนี้ที่พร้อมต์ของเมลบ็อกซ์ :

```
alias
```

หรือ

```
a
```

ลิสต์ของ alias และดีสตริบิวชันลิสต์จะถูกแสดง

เปลี่ยนแปลงจำนวนส่วนหัวข้อความ หรือบรรทัดของเนื้อควาที่แสดงผล ในโปรแกรมเมล:

โดยการเปลี่ยนแปลงไฟล์ `.mailrc` คุณสามารถกำหนดลักษณะ คุณลักษณะเพื่อเลื่อนผ่านรายการเมลบ็อกซ์ หรือผ่านข้อความจริง

เพื่อทำการเปลี่ยนแปลงเหล่านี้ โปรแกรมเมลต้องถูกติดตั้ง บนระบบของคุณ

การเปลี่ยนจำนวนบรรทัดที่ถูกแสดงของลิสต์ของข้อความ:

แต่ละข้อความในเมลบ็อกซ์ของคุณจะมีส่วนหัวบรรทัดเดียวในลิสต์ของข้อความ ถ้าคุณมีมากกว่า 24 ข้อความ ส่วนหัวแรกจากลิสต์ของข้อความจะเลื่อนผ่านด้านบนของหน้าจอคุณ อ็อพชัน **set screen** จะควบคุมจำนวนของบรรทัดของลิสต์ที่ถูกแสดงในเวลาหนึ่ง

เพื่อเปลี่ยนจำนวนของบรรทัดของลิสต์ของข้อความที่ถูกแสดงในเวลาหนึ่ง ในไฟล์ **$HOME/.mailrc** ของคุณ พิมพ์ :

```
set screen=20
```

ในตัวอย่างนี้ ระบบจะแสดงส่วนหัวของ 20 ข้อความในเวลาหนึ่ง ใช้คำสั่งย่อย **h** หรือ **z** เพื่อดูกลุ่มของส่วนหัวเพิ่มเติม คุณยังสามารถพิมพ์คำสั่งย่อยนี้ที่พร้อมต์ของเมลบ็อกซ์

การเปลี่ยนจำนวนบรรทัดที่ถูกแสดงในข้อความที่ยาว:

ถ้าคุณแสดงข้อความที่มีมากกว่า 24 บรรทัด บรรทัดแรกของข้อความจะเลื่อนผ่านด้านบนของหน้าจอ คุณสามารถใช้คำสั่ง **pg** จากภายในเมลเพื่อเรียกดูผ่านข้อความขนาดยาวถ้าคุณรวมอ็อพชัน **set crt** ในไฟล์ `.mailrc`

อ็อพชัน **set crt** จะควบคุมว่าข้อความต้องมีกี่บรรทัดก่อนที่คำสั่ง **pg** จะถูกสตาร์ท

ตัวอย่างเช่น ถ้าคุณใช้คำสั่งย่อย **t** เพื่ออ่านข้อความขนาดยาว จะมีเพียงหน้าจอเดียว (หรือเพจ) ที่ถูกแสดง เพจจะตามด้วยพร้อมต์โคลอนที่ให้คุณรู้ว่ายังมีเพจอื่นอีก กดคีย์ Enter เพื่อแสดงหน้าต่อไปของข้อความ หลังจากหน้าสุดท้ายของข้อความถูกแสดง จะมีพร้อมต์เหมือนต่อไปนี้ :

```
EOF:
```

ที่พร้อมต์ คุณสามารถใส่คำสั่งย่อย **pg** ใดๆที่ถูกต้อง คุณสามารถแสดงหน้าก่อนหน้านี้ ค้นหาข้อความสำหรับสตริงอักขระ หรือออกจากการอ่านข้อความและกลับไปยังพร้อมต์ของเมลบ็อกซ์

อ็อพชัน **set crt** จะถูกใส่ในไฟล์ `.mailrc` เป็น :

set crt=*Lines*

ตัวอย่างเช่น:

```
set crt=20
```

ระบุว่าข้อความต้องมี 20 บรรทัดก่อนที่คำสั่ง **pg** จะถูกสตาร์ท คำสั่ง **pg** จะถูกสตาร์ทเมื่อคุณอ่านข้อความที่มีมากกว่า 20 บรรทัด

การเปลี่ยนจำนวนบรรทัดที่ถูกแสดงที่ด้านบนของข้อความ:

คำสั่งย่อย **top** จะให้คุณสามารถสแกนผ่านข้อความโดยไม่อ่านข้อความทั้งหมด

คุณควบคุมจำนวนของบรรทัดของข้อความที่ถูกแสดงโดยการตั้งอ็อพชัน **toplines** ดังต่อไปนี้ :

set toplines=*Lines*

ในคำสั่งย่อยนี้ ตัวแปร *Lines* เป็นจำนวนของรรทัด เริ่มจากด้านบนและรวมถึงฟิลด์ส่วนหัวทั้งหมด ที่ถูกแสดงด้วยคำสั่งย่อย **top**

ตัวอย่างเช่น ถ้าผู้ใช้ `Amy`มีบรรทัดต่อไปนี้ในไฟล์ `.mailrc` :

```
set toplines=10
```

เมื่อเอมี่รันคำสั่ง **mail** เพื่ออ่านข้อความใหม่ เท็กซ์ต่อไปนี้จะถูกแสดง :

```
Mail Type ? สำหรับความช่วยเหลือ
"/usr/mail/amy": 2 messages 2 new>
N 1 george  Wed Jan 6 9:47  11/257 "Dept Meeting"
N 2 mark    Wed Jan 6 12:59 17/445 "Project Planner"
```

เมื่อเอมีใช้คำสั่งย่อย **top** เพื่อเรียกดูผ่านข้อความ ข้อความบางส่วนต่อไปนี้จะถูกแสดง :

```
top 1
Message 1:
From george Wed Jan 6 9:47 CST 1988
Received: by zeus
    id AA00549; Wed, 6 Jan 88 9:47:46 CST
Date: Wed, 6 Jan 88 9:47:46 CST
From: george@zeus
Message-Id: <8709111757.AA00178>
To: amy@zeus
Subject: Dept Meeting
Please plan to attend the department meeting on Friday
at 1:30 in the planning conference room. We will be
```

ข้อความจะถูกแสดงบางส่วนเนื่องจาก **toplines** ถูกตั้งเป็น 10 เฉพาะบรรทัดที่ 1 (ฟิลด์ **Received:**) ถึง 10 (บรรทัดที่สองของส่วนเนื้อหาของข้อความ) จะถูกแสดง บรรทัดแรก `From george Wed Jan 6 9:47 CST 1988` จะถูกแสดงเสมอ และไม่ถูกนับในอ็อพชัน **toplines**

การแสดงข้อมูลในข้อความ:

โดยการเปลี่ยนไฟล์ `.mailrc` คุณสามารถควบคุมว่าข้อมูลส่วนหัวอะไรจะถูกแสดงในข้อความ

ข้อมูลส่วนหัวบางอย่างอาจถูกปิดอยู่แล้ว ตรวจสอบไฟล์ `/usr/share/lib/Mail.rc` ของคุณสำหรับฟิลด์ส่วนหัวที่ถูกข้าม ข้อกำหนดเบื้องต้น

โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ

การป้องกันการแสดงส่วนหัว Date, From และ To:

ทุกข้อความจะมีหลายฟิลด์ส่วนหัวที่ด้านบน ฟิลด์ส่วนหัวเหล่านี้จะถูกแสดงเมื่อคุณอ่านข้อความ คุณสามารถใช้คำสั่งย่อย **ignore** เพื่อยกเลิกการแสดงฟิลด์ส่วนหัวเมื่อข้อความถูกอ่าน

รูปแบบสำหรับคำสั่งย่อย **ignore** คือ :

```
ignore [FieldList]
```

FieldList สามารถประกอบด้วยชื่อฟิลด์หนึ่งหรือมากกว่าที่คุณต้องการข้ามเมื่อคุณแสดงข้อความ ตัวอย่างเช่น ถ้าผู้ใช้เอมีจะรวมบรรทัดต่อไปนี้ในไฟล์ `.mailrc` :

```
ignore date from to
```

และไฟล์ `/usr/share/lib/Mail.rc` มีบรรทัด:

```
ignore received message-id
```

ผลของการใช้คำสั่งย่อย **t** คือ :

```
t 1
Message 1:
From george Wed Jan 6 9:47 CST 1988
Subject: Dept Meeting
```

```
Please plan to attend the department meeting on Friday
at 1:30 in the planning conference room. We will be
discussing the new procedures for using the project
planning program developed by our department.
```

ฟิลด์ **Received:**, **Date:**, **From:**, **Message-Id:** และ **To:** จะไม่ถูกแสดง เพื่อแสดงฟิลด์เหล่านี้ ใช้คำสั่งย่อย **T** หรือ **P** หรือคำสั่งย่อย **top**

> **หมายเหตุ :** ในตัวอย่าง บรรทัด **From** จะถูกแสดง นี่จะไม่เหมือนกับฟิลด์ **From:** ที่ถูกลิสต์ใน *FieldList* สำหรับคำสั่งย่อย **ignore**

การลิสต์ฟิลด์ ignored header:

ใช้คำสั่งย่อย **ignore** เพื่อลิสต์ฟิลด์ ignored header

เพื่อให้ได้ลิสต์ของฟิลด์ ignored header ในปัจจุบัน ที่พร้อมต์ของเมลบ็อกซ์ พิมพ์ :

```
ignore
```

ลิสต์ของ ignored headers ในปัจจุบันจะถูกแสดง ตัวอย่างเช่น:

```
mail-from
message-id
return-path
```

การรีเซ็ตฟิลด์ส่วนหัว:

เมื่อต้องการรีเซ็ตฟิลด์ส่วนหัว ใช้คำสั่งย่อย **retain**

ตัวอย่างเช่น:

```
retain date
```

การลิสต์ฟิลด์ส่วนหัวที่จองไว้:

ใช้คำสั่งย่อย **retain** เพื่อลิสต์ฟิลด์ส่วนหัวที่จองไว้

เพื่อดูว่าฟิลด์ส่วนหัวใดที่จองไว้ใส่คำสั่งย่อย **retain** โดยไม่มีพารามิเตอร์ฟิลด์ส่วนหัว

วรป้องกันการแสดงแบนเนอร์:

เมลแบนเนอร์เป็นบรรทัดที่ด้านบนของลิสต์ของข้อความที่แสดงชื่อของโปรแกรมเมลเมื่อคุณใช้คำสั่ง **mail**

มันจะเหมือนกับบรรทัดต่อไปนี้ :

```
Mail [5.2 UCB] [Workstation 3.1] Type ? สำหรับความช่วยเหลือ
```

เพื่อป้องกันการแสดงแบนเนอร์เมื่อคุณสตาร์ทโปรแกรมเมล เพิ่มบรรทัดต่อไปนี้เข้ากับไฟล์ `$HOME/.mailrc` ของคุณ :

```
set quiet
```

อ็อพชันอื่นที่ยกเลิกแบนเนอร์ **mail** ของคุณคือ:

```
set noheader
```

ด้วยอ็อพชันนี้ในไฟล์ .mailrc ลิสต์ของข้อความในเมลบ็อกซ์ของคุณจะไม่ถูกแสดง เมื่อคุณสตาร์ทโปรแกรม **mail** การตอบสนองเดียวคือพร้อมต์ของเมลบ็อกซ์ คุณสามารถได้รับลิสต์ของข้อความโดยการพิมพ์คำสั่งย่อย (**h**)eader

การรวมคำสั่งลบและพิมพ์:

ใช้อ็อพชัน autoprint เพื่อรวมคำสั่งย่อยการลบและพิมพ์

หลังจากคุณอ่านข้อความ คุณสามารถลบมันด้วยคำสั่งย่อย **d** คุณสามารถแสดงข้อความถัดไปด้วยคำสั่งย่อย **p** รวมคำสั่งย่อยเหล่านี้โดยการพิมพ์บรรทัดต่อไปนี้ในไฟล์ .mailrc ของคุณ:

```
set autoprint
```

ด้วยอ็อพชัน **set autoprint** ในไฟล์ .mailrc คำสั่งย่อย **d** จะลบข้อความปัจจุบันและแสดงข้อความถัดไป

การสร้างดีฟอลต์โฟลเดอร์เพื่อเก็บข้อความ:

ดีฟอลต์โฟลเดอร์ให้คุณสามารถเก็บข้อความ

โปรแกรมเมลต้องถูกติดตั้งบนระบบของคุณ

ใช้โพรซีเดอร์ต่อไปนี้เพื่อสร้างไดเร็กทอรีของเมลบ็อกซ์ letter เพื่อเก็บข้อความในโฟลเดอร์ :

1. เพื่อตรวจสอบว่าอ็อพชัน **set folder** ถูกเปิดใช้งานในไฟลไฟล์ .mailrc หรือไม่ ที่พร้อมต์ของเมลบ็อกซ์ พิมพ์ :

   ```
   set
   ```

   ถ้าอ็อพชัน **set folder** ถูกเปิดใช้งาน ระบบจะตอบสนองด้วยต่อไปนี้ :

   ```
   folder /home/george/letters
   ```

   ในตัวอย่างนี้ letters เป็นไดเร็กทอรีที่เมลโฟลเดอร์จะถูกเก็บ
2. ถ้าอ็อพชัน **set folder** ไม่ถูกเปิดใช้งาน สร้าง entry **set folder** ในไฟล์ .mailrc :

   ```
   set folder=/home/george/letters
   ```

   ในตัวอย่างนี้ /home/george เป็นไดเร็กทอรีหลักของจอร์จ และ letters เป็นไดเร็กทอรีที่เมลโฟลเดอร์จะถูกเก็บ อ็อพชัน **set folder** จะให้คุณสามารถใช้เครื่องหมายบวก (+) สัญลักษณ์ชวเลขเพื่อบันทึกข้อความในไดเร็กทอรี letters ของคุณ
3. ถ้าไดเร็กทอรี letters ไม่มีอยู่ คุณต้องสร้างไดเร็กทอรี letters ในไดเร็กทอรีหลักของคุณ จากไดเร็กทอรีหลักของคุณที่บรรทัดรับคำสั่งของระบบ พิมพ์ :

   ```
   mkdir letters
   ```

ใช้โพรซีเดอร์ต่อไปนี้เพื่อเก็บเร็กคอร์ดของข้อความที่คุณส่งไปยังผู้อื่น :

1. พิมพ์ข้อความต่อไปนี้ในไฟล์ .mailrc ของคุณ :

   ```
   set record=letters/mailout
   ```
2. ถ้าไดเร็กทอรี letters ไม่มีอยู่ คุณต้องสร้างไดเร็กทอรี letters ในไดเร็กทอรีหลักของคุณ จากไดเร็กทอรีหลักของคุณที่บรรทัดรับคำสั่งของระบบ พิมพ์ :

   ```
   mkdir letters
   ```

3. เพื่ออ่านคัดลอกของข้อความที่คุณส่งไปยังผู้อื่น พิมพ์ :

   ```
   mail -f +mailout
   ```

   ในตัวอย่างนี้ ไฟล์ `mailout` ประกอบด้วยคัดลอกของข้อความที่คุณส่งไปยังผู้อื่น

เท็กซ์เอดิเตอร์สำหรับการพิมพ์ข้อความ:

ใช้อ็อพชัน **set EDITOR=***PathName* เพื่อกำหนดเท็กซ์เอดิเตอร์ที่คุณจะพิมพ์ข้อความ

เมลโปรแกรมต้องถูกติดตั้งบนระบบของคุณ

ไอเท็ม	คำอธิบาย
set EDITOR=*PathName*	อ็อพชันนี้ในไฟล์ `.mailrc` ของคุณจะกำหนดเอดิเตอร์ที่คุณเปิดใช้งานด้วยลำดับของคีย์ **~e** ค่าของ *PathName* ต้องเป็นชื่อพาธแบบเต็มไปยังโปรแกรมเอดิเตอร์ที่คุณต้องการใช้ เพื่อเปลี่ยนเป็นเอดิเตอร์ e ขณะอยู่ในโปรแกรมเมล พิมพ์: `~e` ลำดับนี้จะเปิดใช้งานเอดิเตอร์ e หรือเอดิเตอร์ที่คุณกำหนดในไฟล์ `.mailrc` แก้ไขข้อความเมลของคุณโดยใช้เอดิเตอร์นี้
set VISUAL=*PathName*	อ็อพชันนี้ในไฟล์ `.mailrc` ของคุณจะกำหนดเอดิเตอร์ที่คุณเปิดใช้งานด้วยลำดับของคีย์ **~v** ค่าของ *PathName* ต้องเป็นชื่อพาธแบบเต็มไปยังโปรแกรมเอดิเตอร์ที่คุณต้องการใช้ ค่าดีฟอลต์คือ `/usr/bin/vi` เพื่อเปลี่ยนเป็นเอดิเตอร์ vi ขณะอยู่ในโปรแกรมเมล พิมพ์: `~v` ลำดับนี้จะเปิดใช้งานเอดิเตอร์ vi หรือเอดิเตอร์ที่คุณกำหนดในไฟล์ `.mailrc` แก้ไขข้อความเมลของคุณโดยใช้เอดิเตอร์นี้

คำสั่งย่อยของคำสั่งเมล

คำสั่ง **mail** ใช้คำสั่งย่อยต่างๆ ที่ดำเนินการฟังก์ชันแตกต่างกัน

หัวข้อนี้ใช้เป็นการอ้างอิงสำหรับคำสั่ง **mail** และคำสั่งย่อย

คำสั่งในการเรียกใช้งานเมล:

ใช้คำสั่งของระบบเหล่านี้เพื่อเรียกใช้งานเมล

ไอเท็ม	คำอธิบาย
mail	แสดงระบบเมลบ็อกซ์
mail -f	แสดงเมลบ็อกซ์ส่วนตัวของคุณ (**mbox**)
mail -f +*folder*	แสดงเมลโฟลเดอร์
mail*user@address*	กำหนดแอดเดรสข้อความไปยังผู้ใช้ที่ระบุ

คำสั่งย่อยเมลบ็อกซ์ในโปรแกรมเมล:

เมื่อโปรแกรมเมลกำลังประมวลผลเมลบ็อกซ์ มันจะแสดงผล พร้อมต์เมลบ็อกซ์เพื่อบ่งชี้ว่ากำลังรออินพุต

พร้อมต์เมลบ็อกซ์คือเครื่องหมาย ampersand (&) ที่แสดงอยู่ที่จุดเริ่มต้น บรรทัดใหม่ ที่พร้อมต์ คุณสามารถป้อนคำสั่งย่อยของเมลบ็อกซ์

คำสั่งย่อยสำหรับควบคุมโปรแกรมเมล:

ใช้คำสั่งย่อยเหล่านี้สำหรับควบคุมโปรแกรมเมล

ไอเท็ม	คำอธิบาย
q	ออกจากและใช้คำสั่งย่อยของเมลบ็อกซ์ที่ถูกใส่ในเซสชันนี้
x	ออกจากและเรียกคืนเมลบ็อกซ์เป็นสถานะดั้งเดิม
!	สตาร์ทเชลล์ รันคำสั่ง และกลับไปยังเมลบ็อกซ์
cd *dir*	เปลี่ยนไดเร็กทอรีเป็น `dir` หรือ `$HOME`

คำสั่งย่อยในการแสดงโปรแกรมเมล:

ใช้คำสั่งย่อยเหล่านี้เพื่อควบคุมการแสดงโปรแกรมเมล

ไอเท็ม	คำอธิบาย
t	จะแสดงข้อความใน *msg_list* หรือข้อความปัจจุบัน
n	แสดงข้อความถัดไป
f *msg_list*	แสดงส่วนหัวของข้อความใน *msg_list* หรือข้อความปัจจุบัน ถ้า *msg_list* ไม่ถูกระบุ
h *num*	แสดงส่วนหัวของกลุ่มที่ประกอบด้วยข้อความ *num*
top *num*	แสดงข้อความบางส่วน
set	แสดงลิสต์ของอ็อพชัน **.mailrc** ทั้งหมดที่ถูกเปิดใช้งาน
ignore	แสดงลิสต์ของฟิลด์ ignored header ทั้งหมด
โฟลเดอร์	แสดงจำนวนของข้อความในโฟลเดอร์ปัจจุบันตามด้วยชื่อพาธของโฟลเดอร์

การจัดการข้อความ:

ใช้คำสั่งย่อยเหล่านี้เพื่อแก้ไข ลบ เรียกกลับมาใหม่ ต่อท้าย หรือเก็บข้อความ

ไอเท็ม	คำอธิบาย
e *num*	แก้ไขข้อความ *num* (ดีฟอลต์เอดิเตอร์คือ e)
d *msg_list*	ลบข้อความใน *msg_list* หรือข้อความปัจจุบัน
u *msg_list*	เรียกข้อความที่ถูกลบกลับคืนใน *msg_list*
s *msg_list* **+***file*	ต่อท้ายข้อความ (พร้อมกับส่วนหัว) เข้ากับ *file*
w *msg_list* **+***file*	ต่อท้ายข้อความ (เท็กซ์เท่านั้น) เข้ากับ *file*
pre *msg_list*	เก็บข้อความในระบบเมลบ็อกซ์

คำสั่งย่อยเมลใหม่:

ใช้คำสั่งย่อยเหล่านี้เมื่อสร้างข้อความเมลใหม่

ไอเท็ม	คำอธิบาย
m *addrlist*	สร้างและส่งข้อความใหม่ไปยังแอดเดรสใน *addrlist*
r *msg_list*	ส่งการตอบไปยังผู้ส่งและผู้รับของข้อความ
R *msg_list*	ส่งการตอบเฉพาะผู้ส่งของข้อความ
a	แสดงลิสต์ของ alias และแอดเดรสของมัน

คำสั่งย่อยตัวแก้ไขเมล:

เมื่อตัวแก้ไขเมลถูกประมวลผล ซึ่งแสดงผลพร้อมต์ของตัวแก้ไขเมล เพื่อบ่งชี้ว่ากำลังรออินพุต

ที่พร้อมต์ คุณสามารถป้อนคำสั่งย่อยของตัวแก้ไขเมล

คำสั่งย่อยในการควบคุมเมลเอดิเตอร์:

ใช้คำสั่งย่อยต่อไปนี้เพื่อควบคุมเมลเอดิเตอร์

ไอเท็ม	คำอธิบาย
~q	ออกจากเอดิเตอร์โดยไม่บันทึกหรือส่งข้อความปัจจุบัน
~p	แสดงเนื้อหาของบัฟเฟอร์ของข้อความ
~: *mcmd*	รันคำสั่งย่อยของเมลบ็อกซ์ (*mcmd*)
EOT	ส่งข้อความ (Ctrl-D บนหลายเทอร์มินัล)
.	ส่งข้อความปัจจุบัน

คำสั่งย่อย Add to heading:

ใช้คำสั่งย่อยเหล่านี้เพื่อเพิ่มไอเท็มของส่วนหัวอื่นเข้ากับข้อความ

ไอเท็ม	คำอธิบาย
~h	เพิ่มเข้ากับฟิลด์ **To:**, **Subject:**, **Cc:**, และ **Bcc:**
~t *addrlist*	เพิ่มแอดเดรสของผู้ใช้ใน *addrlist* เข้ากับฟิลด์ **To:**
~s *subject*	ตั้งฟิลด์ **Subject** เป็นสตริงที่ถูกระบุโดย *subject*
~c *addrlist*	เพิ่มแอดเดรสของผู้ใช้ใน *addrlist* เข้ากับฟิลด์ **Cc:** (carbon copy)
~b *addrlist*	เพิ่มแอดเดรสของผู้ใช้ใน *addrlist* เข้ากับฟิลด์ `Bcc:` (blind carbon copy)

คำสั่งย่อย Add to message:

ใช้คำสั่งย่อยเหล่านี้เพื่อเพิ่มเนื้อหาเข้ากับข้อความ

ไอเท็ม	คำอธิบาย
~d	ต่อท้ายเนื้อหาของ `dead.letter` เข้ากับข้อความ
~r *filename*	ต่อท้ายเนื้อหาของ *filename* เข้ากับข้อความ
~f *numlist*	ต่อท้ายเนื้อหาของหมายเลขข้อความ *numlist*
~m*numlist*	ต่อท้ายและย่อหน้าเนื้อหาของหมายเลขข้อความ *numlist*

คำสั่งย่อยสำหรับเปลี่ยนข้อความ:

ใช้คำสั่งย่อยเหล่านี้เพื่อแก้ไขข้อความ

ไอเท็ม	คำอธิบาย
~e	แก้ไขข้อความโดยใช้ e เอดิเตอร์ (ดีฟอลต์คือ e)
~v	แก้ไขข้อความโดยใช้ vi เอดิเตอร์ (ดีฟอลต์คือ vi)
~w *filename*	เขียนข้อความไปยัง *filename*
~! *คำสั่ง*	สตาร์ทเชลล์ รันคำสั่ง *command* และกลับไปยังเอดิเตอร์
~| *command*	ไพพ์ข้อความไปยังอินพุตมาตรฐานของ *command* และแทนที่ข้อความด้วยเอาต์พุตมาตรฐานจากคำสั่งนั้น

คำสั่งย่อย Secret mail:

เมื่อโปรแกรมเมลที่เป็นความลับประมวลผลเมลบ็อกซ์ที่เป็นความลับ มันจะแสดงพร้อมต์ของเมลบ็อกซ์ที่เป็นความลับเพื่อระบุว่ามันกำลังรออินพุต

พร้อมต์ของเมลบ็อกซ์ที่เป็นความลับเป็นเครื่องหมายคำถาม (?) ถูกแสดงที่ต้นบรรทัดใหม่ ที่พร้อมต์ คุณสามารถป้อนคำสั่งย่อยของเมลบ็อกซ์ที่เป็นความลับ

คำสั่งย่อย Secret mail:

ใช้คำสั่งย่อยต่อไปนี้เพื่อส่งเมลความลับ

ไอเท็ม	คำอธิบาย
xsend barbara	กำหนดแอดเดรสข้อความไปยังผู้ใช้ที่ระบุ
xget	จะแสดงเมลบ็อกซ์ที่เป็นความลับ

งานเมลบ็อกซ์:

คำสั่งย่อยต่อไปนี้จะทำงานต่างๆของเมลบ็อกซ์

ไอเท็ม	คำอธิบาย
q	ออก ปล่อยข้อความที่ยังไม่อ่าน
n	ลบข้อความปัจจุบันและแสดงข้อความถัดไป
d	ลบข้อความปัจจุบันและแสดงข้อความถัดไป
Return key	ลบข้อความปัจจุบันและแสดงข้อความถัดไป
!	เรียกใช้งานคำสั่งเชลล์
s	บันทึกข้อความในไฟล์ named หรือ mbox
w	บันทึกข้อความในไฟล์ named หรือ mbox

งานการจัดการเมล

ตัวจัดการเมลจะรับผิดชอบสำหรับงานเหล่านี้

1. ตั้งค่าไฟล์ /etc/rc.tcpip เพื่อที่ **sendmail** daemon จะถูกสตาร์ทเมื่อเวลาที่ระบบบูต โปรดดู “การตั้งค่าไฟล์ /etc/rc.tcpip เพื่อสตาร์ท sendmail daemon”
2. ปรับไฟล์คอนฟิกูเรชัน /etc/mail/sendmail.cf ตามต้องการ ไฟล์ /etc/mail/sendmail.cf ดีฟอลต์ถูกตั้งค่า เพื่อที่ทั้งโลคัลเมลและ TCP/IP เมลสามารถถูกส่งได้ เพื่อที่จะส่งเมลผ่าน BNU คุณต้องปรับแต่งไฟล์ /etc/mail/sendmail.cf ดูที่ ไฟล์ sendmail.cf ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติม
3. กำหนดเมล aliad แบบระบบและโดเมนในไฟล์ /etc/mail/aliases โปรดดู “เมล alias”สำหรับ ข้อมูลเพิ่มเติม
4. การจัดการคิวของเมล โปรดดู “คิวของเมล” ในหน้า 52 สำหรับ ข้อมูลเพิ่มเติม
5. การจัดการล็อกของเมล โปรดดู “การล็อกเมล” ในหน้า 56 สำหรับ ข้อมูลเพิ่มเติม

การตั้งค่าไฟล์ /etc/rc.tcpip เพื่อสตาร์ท sendmail daemon

ใช้โพรซีเดอร์นี้ เพื่อตั้งค่าไฟล์ /etc/rc.tcpip เพื่อที่ **sendmail** daemon จะถูกสตาร์ทเมื่อเวลาที่ระบบบูต

1. แก้ไขไฟล์ /etc/rc.tcpip ด้วยเท็กซ์เอดิเตอร์ที่คุณชอบ
2. หาบรรทัดที่เริ่มต้นด้วย start /usr/lib/sendmail โดยดีฟิลต์ บรรทัดนี้ควรถูกยกเลิกหมายเหตุ นั่นคือ ไม่มีเครื่องหมาย # ที่เริ่มต้นของบรรทัด อย่างไรก็ตาม ถ้ามันถูกทำหมายเหตุ ลบเครื่องหมาย # ออก
3. บันทึกไฟล์

โดยการเปลี่ยนแปลงนี้ ระบบจะสตาร์ท **sendmail** daemon เมื่อเวลาที่บูต

เมล alias

alias จะแม็พชื่อกับลิสต์ของแอดเดรสโดยใช้ไฟล์ alias ของบุคคล ระบบ และโดเมน

คุณสามารถกำหนดชนิดของ alias 3 ชนิด :

ไอเท็ม	คำอธิบาย
บุคคล	ถูกกำหนดโดยแต่ละบุคคลในไฟล์ `$HOME/.mailrc` ของผู้ใช้
local system	ถูกกำหนดโดยผู้ดูแลระบบของเมลในไฟล์ `/etc/mail/aliases` alias เหล่านี้จะใช้กับเมลที่ถูกจัดการโดย **sendmail** บนระบบโลคัล alias ของระบบโลคัลไม่ค่อยจะถูกเปลี่ยนแปลง
domainwide	โดยดีฟอลต์ **sendmail** จะอ่าน `/etc/alias` เพื่อแปลง alias เพื่อทับค่าดีฟอลต์และใช้ NIS แก้ไขหรือสร้าง `/etc/netsvc.conf` และเพิ่มบรรทัด : `aliases=nis`

ไฟล์ /etc/mail/aliases

คุณสมบัติ เนื้อหา และตำแหน่งของไฟล์ `/etc/mail/aliases` ถูกอธิบายในที่นี้

ไฟล์ `/etc/mail/aliases` ประกอบด้วยชุดของ entry ในรูปแบบต่อไปนี้ :

Alias: *Name1*, *Name2*, ... *NameX*

โดยที่ *Alias* สามารถเป็นสตริงตัวอักษรผสมตัวเลขใดๆที่คุณสามารถเลือก (ไม่รวมอักขระพิเศษ เช่น @ or !). *Name1* ถึง *NameX* เป็นชุดของชื่อผู้รับหนึ่งชื่อหรือมากกว่า ลิสต์ของชื่อสามารถแตกออกเป็นหนึ่งหรือหลายบรรทัด แต่ละบรรทัดที่ต้อกันจะเริ่มต้นด้วยช่องว่างหรือแท็บ บรรทัดว่างและบรรทัดที่เริ่มต้นด้วย # (เครื่องหมายปาวน์) เป็นหมายเหตุ

ไฟล์ `/etc/mail/aliases` ต้องประกอบด้วย 3 alias ต่อไปนี้ :

ไอเท็ม	คำอธิบาย
MAILER-DAEMON	ID ของผู้ใช้ที่รับข้อความที่ถูกกำหนดแอดเดรสไปยัง mailer daemon ชื่อนี้จะถูกกำหนดเริ่มต้นกับผู้ใช้รูท : `MAILER-DAEMON: root`
postmaster	ID ของผู้ใช้ที่รับผิดชอบสำหรับการดำเนินการของระบบเมลแบบโลคัล alias ของ **postmaster** จะกำหนดแอดเดรสของเมลบ็อกซ์เดียวที่ใช้ได้ที่แต่ละระบบในเน็ตเวิร์ก แอดเดรสนี้จะให้ผู้ใช้สามารถส่งการสอบถามกับ alias ของ **postmaster** ที่ระบบใดๆ โดยไม่ต้องรู้แอดเดรสที่ถูกต้องของผู้ใช้ใดๆ ที่ระบบนั้น ชื่อนี้จะถูกกำหนดเริ่มต้นกับผู้ใช้รูท : `postmaster: root`
nobody	ID ที่เพื่อรับข้อความโดยตรงกับโปรแกรม เช่น **news** และ **msgs** ชื่อนี้จะถูกกำหนดเริ่มต้นกับ `/dev/null`: `nobody: /dev/null` เพื่อรับข้อความเหล่านี้ กำหนด alias นี้เป็นผู้ใช้ที่ถูกต้อง

เมื่อใดก็ตามที่คุณเปลี่ยนไปล์นี้ คุณต้องคอมไพล์มันใหม่เป็นรูปแบบฐานข้อมูลที่คำสั่ง **sendmail** สามารถใช้ได้ โปรดดู “การสร้างฐานข้อมูล alias” ในหน้า 51

การสร้างสมนามโลคัลสำหรับเมล

การสร้างสมนามโลคัลสำหรับเมลช่วยให้คุณสามารถสร้างรายชื่อกลุ่มหรือ การแจกจ่ายซึ่งสามารถส่งเมลได้

ในสถานการณ์จำลองนี้ จะมีการเพิ่ม `geo@medussa`, `mark@zeus`, `ctw@athena`, และ `dsf@plato` ลงในสมนามเมล `testers` หลังจากสมนาม `testers` ถูกสร้างขึ้นแล้ว glenda@hera จะได้รับ มอบความเป็นเจ้าของสมนาม

หลังจากเพิ่มสมนาม `testers` ลงในไฟล์ `/etc/mail/aliases` แล้ว ฐานข้อมูลสมนาม จะถูกคอมไพล์อีกครั้งโดยใช้คำสั่ง **sendmail** หลังจากคอมไพล์ฐานข้อมูลอีกครั้ง อีเมลสามารถส่งไปยัง `testers` alias

สิ่งที่ต้อง พิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ได้ผ่านการทดสอบโดยใช้เวอร์ชันเฉพาะของ AIX ผลลัพธ์ที่คุณได้อาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชันและระดับ AIX ของคุณ

ใช้ขั้นตอนต่อไปนี้เพื่อสร้างสมนามโลคัลสำหรับเมล:

1. เปิดไฟล์ /etc/mail/aliases โดยใช้โปรแกรมแก้ไขข้อความที่โปรดปรานของคุณ
2. บนบรรทัดเปล่า ให้เพิ่มชื่อสมนาม ตามด้วยเครื่องหมายจุดคู่และรายการ ของผู้รับที่คั่นด้วยเครื่องหมายคอมมา ตัวอย่าง เช่น รายการต่อไปนี้กำหนดสมนาม testers:

   ```
   testers: geo@medussa, mark@zeus, ctw@athena, dsf@plato
   ```

3. สร้างเจ้าของสมนาม หากคำสั่ง **sendmail** ส่งเมลไปยังสมนามไม่สำเร็จ คำสั่งจะส่งข้อความแสดงข้อผิดพลาดไปยัง เจ้าของ เพิ่มบรรทัดใน /etc/mail/aliases เพื่อระบุ เจ้าของ รูปแบบสำหรับบรรทัดนี้คือ owner-*groupname*: *owner* โดยที่ *groupname* คือชื่อของสมนาม และ *owner* คืออีเมลแอดเดรสของเจ้าของ ในตัวอย่างนี้ glenda@hera มีการ กำหนดเป็นเจ้าของสมนาม testers:

   ```
   testers: geo@medussa, mark@zeus, ctw@athena, dsf@plato owner-testers: glenda@hera
   ```

4. หลังจากสร้างสมนามแล้ว ให้รันคำสั่ง **sendmail -bi** เพื่อ คอมไพล์ฐานข้อมูลสมนามอีกครั้ง คุณจะต้องรันคำสั่งนี้ในทุกครั้ง ที่คุณอัพเดตไฟล์ /etc/mail/aliases

คุณสามารถส่งอีเมลไปยัง testers alias

การสร้างฐานข้อมูล alias

คำสั่ง **sendmail** ไม่ได้ใช้คำจำกัดความของ alias ในไฟล์ /etc/mail/aliases ของระบบโลคัลโดยตรง คำสั่ง **sendmail** จะอ่านเวอร์ชันของตัวจัดการฐานข้อมูลของไฟล์ /etc/mail/aliases ที่ถูกประมวลผลแทน

คุณสามารถคอมไพล์ฐานข้อมูลของ alias โดยใช้หนึ่งในวิธีต่อไปนี้ :

- รันคำสั่ง /usr/sbin/sendmail โดยใช้แฟล็ก **-bi**
- รันคำสั่ง **newaliases** คำสั่งนี้จะทำให้คำสั่ง **sendmail** อ่านไฟล์ /etc/mail/aliases ของระบบโลคัล และสร้างไฟล์ใหม่ที่ประกอบด้วยข้อมูลฐานข้อมูลของ alias ไฟล์นี้จะอยู่ในรูปแบบของ Berkeley ที่มีประสิทธิภาพมากกว่า :

  ```
  /etc/mail/aliases.db
  ```

- รันคำสั่ง **sendmail** โดยใช้แฟล็ก **Rebuild Aliases** ซึ่งจะสร้างฐานข้อมูลของ alias ใหม่โดยอัตโนมัติเมื่อมันไม่ทันสมัย การสร้างใหม่โดยอัตโนมัติอาจมีอันตรายกับเครื่องที่มีโหลดมากด้วยไฟล์ alias ขนาดใหญ่ ถ้ามันใช้เวลามากกว่า rebuild timeout (โดยทั่วไปคือ 5 นาที) เพื่อสร้างฐานข้อมูลใหม่ จะมีโอกาสที่หลายกระบวนการจะเริ่มกระบวนการ rebuild พร้อมกัน

หมายเหตุ:

1. ถ้าไฟล์เหล่านี้ไม่มีอยู่ คำสั่ง **sendmail** จะไม่สามารถประมวลผลเมลและจะสร้างข้อความแสดงข้อผิดพลาด
2. ถ้าคุณระบุหลายฐานข้อมูลของ alias แฟล็ก **-bi** จะสร้างชนิดของฐานข้อมูลที่มันเข้าใจใหม่ (ตัวอย่างเช่น มันสามารถสร้างฐานข้อมูล Network Database Management (NDBM) ใหม่แต่ไม่ใช่ฐานข้อมูล NIS)

ไฟล์ /etc/netsvc.conf ประกอบด้วยลำดับของของเซอร์วิสของระบบ เพื่อระบุลำดับของเซอร์วิสของ alias เพิ่มบรรทัดต่อไปนี้ :

```
aliases=service, service
```

โดยที่ service สามารถเป็น ไฟล์ หรือ nis ตัวอย่าง เช่น:

```
aliases=files, nis
```

บอกคำสั่ง **sendmail** ให้ลองใช้ไฟล์ alias แบบโลคัลก่อน และถ้าล้มเหลว ลองใช้ nis ถ้า nis ถูกกำหนดเป็นเซอร์วิส มันควรรันได้

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับไฟล์ /etc/netsvc.conf ดูที่ *การอ้างอิงไฟล์*

คิวของเมล

คิวของเมลเป็นไดเร็กทอรีที่เก็บข้อมูลและไฟล์ควบคุมสำหรับข้อความเมลที่คำสั่ง **sendmail** นำส่ง โดยดีฟอลต์ คิวของเมลคือ /var/spool/mqueue

ข้อความเมลอาจจะถูกคิวโดยหลายสาเหตุ

ตัวอย่างเช่น:

1. คำสั่ง **sendmail** สามารถถูกตั้งค่าเพื่อประมวลผลคิวที่ประกอบด้วยช่วงเวลา แทนที่จะเป็นทันที ถ้าเป็นดังนี้ ข้อความเมลต้องถูกเก็บชั่วคราว
2. ถ้ารีโมตโฮสต์ไม่ตอบคำร้องขอสำหรับการเชื่อมต่อเมล ระบบเมลจะเข้าคิวข้อความและลองอีกครั้งภายหลัง

คิวการพิมพ์ของเมล

เนื้อหาของคิวสามารถถูกพิมพ์โดยใช้คำสั่ง **mailq** (หรือโดยระบุแฟล็ก **-bp** กับคำสั่ง **sendmail**)

คำสั่งเหล่านี้จะสร้างลิสต์ของ ID ของคิว ขนาดของข้อความ วันที่ข้อความถูกใส่ในคิว และผู้ส่ง และผู้รับ

ไฟล์คิวของเมล

แต่ละข้อความในคิวมีหมายเลขของไฟล์ที่เชื่อมโยงกับมัน

ไฟล์ถูกตั้งชื่อตามหลักการตั้งชื่อต่อไปนี้ :

*Type*f*ID*

โดยที่ *ID* เป็น ID ของคิวของข้อความที่เป็นหนึ่งเดียว และ *Type* เป็นหนึ่งในตัวอักษรต่อไปนี้ที่ระบุชนิดของไฟล์ :

ไอเท็ม	คำอธิบาย
d	ไฟล์ข้อมูลจะประกอบด้วยส่วนเนื้อหาของข้อความโดยไม่มีข้อมูลส่วนหัว
q	ไฟล์สำหรับควบคุมคิว ไฟล์นี้ประกอบด้วยข้อมูลที่จำเป็นเพื่อดำเนินการกับงาน
t	ไฟล์ชั่วคราว ไฟล์นี้เป็นอิมเมจของไฟล์ q เมื่อมันถูกสร้างใหม่ มันจะถูกเปลี่ยนชื่อเป็นไฟล์ q อย่างรวดเร็ว
x	ไฟล์ transcript ที่มีอยู่ระหว่างช่วงเวลาของเซสชันและแสดงที่อย่างที่เกิดขึ้นระหว่างเซสชันนั้น

ตัวอย่างเช่น ถ้าข้อความมี ID ของคิวเป็น AA00269 ไฟล์ต่อไปนี้จะถูกสร้างและลบในไดเร็กทอรีคิวของเมลขณะที่คำสั่ง **sendmail** พยายามนำส่งข้อความ:

ไอเท็ม	คำอธิบาย
dfAA00269	ไฟล์ข้อมูล
qfAA00269	ไฟล์ควบคุม
tfAA00269	ไฟล์ชั่วคราว
xfAA00269	ไฟล์ Transcript

q control file:

ไฟล์ควบคุม q ประกอบด้วยชุดของบรรทัด แต่ละบรรทัดเริ่มต้นด้วยโค้ดตัวอักษร

ไอเท็ม	คำอธิบาย
B	ระบุ body type บรรทัดที่เหลือเป็นเท็กซ์สตริงที่กำหนด body type ถ้าฟิลด์ทั้งหมดนี้หายไป body type จะเป็น 7 บิตโดยดีฟอลต์ และไม่มีการดำเนินการพิเศษที่จะถูกพยายาม ค่าที่ถูกต้องคือ **7BIT** และ **8BITMIME**
C	ประกอบด้วยผู้ใช้ที่ควบคุม สำหรับแอดเดรสของผู้รับที่เป็นไฟล์หรือโปรแกรม **sendmail** จะทำการนำส่งโดยเป็นเจ้าของไฟล์หรือโปรแกรม ผู้ใช้ที่ควบคุมถูกตั้งเป็นเจ้าของของไฟล์หรือโปรแกรม แอดเดรสของผู้รับที่ถูกอ่านจากไฟล์ **.forward** หรือ **:include:** จะมีผู้ใช้ที่ควบคุมถูกตั้งเป็นเจ้าของของไฟล์ เมื่อ **sendmail** นำส่งเมลไปยังผู้รับเหล่านี้ มันจะนำส่งโดยเป็นผู้ใช้ที่ควบคุม จากนั้นแปลงกลับเป็น root
F	ประกอบด้วยแฟล็กซอง แฟล็กเป็นการรวมกันของ **w** ซึ่งจะตั้งแฟล็ก **EF_WARNING** ซึ่งจะตั้ง **r** ซึ่งจะตั้งแฟล็ก **EF_RESPONSE** **8** ซึ่งจะตั้งแฟล็ก **EF_HAS8BIT** และ **b** ซึ่งจะตั้งแฟล็ก **EF_DELETE_BCC** ตัวอักษรอื่นจะถูกข้ามไปอย่างเงียบๆ
H	ประกอบด้วยคำจำกัดความของส่วนหัว มันอาจเป็นหมายเลขใดๆของบรรทัดเหล่านี้ เพื่อที่บรรทัด **H** จะกำหนดลำดับของมันในข้อความสุดท้าย บรรทัดเหล่านี้ใช้ไวยากรณ์เดียวกับคำจำกัดความของส่วนหัวในไฟล์คอนฟิกูเรชัน /etc/mail/sendmail.cf
I	ระบุ inode และข้อมูลของอุปกรณ์สำหรับไฟล์ df นี่สามารถถูกใช้เพื่อเรียกคืนคิวของเมลของคุณหลังจากที่ดิสก์เสียหาย
K	ระบุเวลา (เป็นวินาที) ของความพยายามนำส่งล่าสุด
M	เมื่อข้อความถูกใส่เข้าไปในคิวเนื่องจากมีข้อผิดพลาดเกิดขึ้นระหว่างความพยายามนำส่ง ธรรมชาติของข้อผิดพลาดจะถูกเก็บในบรรทัด **M**
N	ระบุจำนวนทั้งหมดของความพยายามนำส่ง
O	ระบุค่าของ message transfer system (MTS) ดั้งเดิมจาก ESMTP มันถูกใช้สำหรับ Delivery Status Notifications เท่านั้น
P	ประกอบด้วยระดับความสำคัญของข้อความปัจจุบัน ระดับความสำคัญถูกใช้เพื่อจัดลำดับคิว หมายเลขที่สูงกว่า หมายถึงระดับความสำคัญที่ต่ำกว่า ระดับความสำคัญจะเพิ่มเมื่อข้อความอยู่ในคิว ตัวขึ้นต้นเริ่มต้นขึ้นอยู่กับคลาสของข้อความและขนาดของข้อความ
Q	ประกอบด้วยผู้รับดั้งเดิมที่ถูกระบุโดยฟิลด์ ORCPT= ในรายการ ESMTP มันถูกใช้เป็นพิเศษสำหรับ Delivery Status Notifications มันใช้เฉพาะกับที่ต่อจากบรรทัด **R** ต่อไปนี้
R	ประกอบด้วยแอดเดรสของผู้รับ จะมีหนึ่งบรรทัดสำหรับแต่ละผู้รับ
S	ประกอบด้วยแอดเดรสของผู้ส่ง จะมีเพียงหนึ่งของบรรทัดเหล่านี้
T	ประกอบด้วยเวลาการสร้างข้อความที่ถูกใช้เพื่อคำนวณว่าเมื่อไรที่ข้อความจะหมดเวลา
V	ระบุตัวเลขเวอร์ชันของรูปแบบของไฟล์คิวที่ถูกใช้เพื่อให้ไบนารี **sendmail** อ่านไฟล์คิวที่ถูกสร้างโดยเวอร์ชันที่เก่ากว่า ดีฟอลต์เป็นเวอร์ชัน **zero** ต้องเป็นบรรทัดแรกของไฟล์ ถ้ามี
Z	ระบุ ID ของซองดั้งเดิม (จากรายการ ESMTP) ถูกใช้สำหรับ Delivery Status Notifications เท่านั้น
$	ประกอบด้วยคำจำกัดความขนาดไมโคร ค่าของไมโครนั้นๆ (**$r** และ **$s**) จะถูกผ่านไปยังเฟสของการรันคิว

ไฟล์ q สำหรับข้อความส่งไปยัง amy@zeus จะเหมือนกับ :

```
P217031
T566755281
MDeferred: Connection timed out during user open with zeus
Sgeo
Ramy@zeus
H?P?return-path: <geo>
Hreceived: by george (0.13 (NL support)/0.01)
        id AA00269; Thu, 17 Dec 87 10:01:21 CST
H?D?date: Thu, 17 Dec 87 10:01:21 CST
H?F?From: geo
Hmessage-id: <8712171601.AA00269@george>
HTo: amy@zeus
Hsubject: test
```

โดยที่:

ไอเท็ม	คำอธิบาย
`P217031`	ระดับความสำคัญของข้อความ
`T566755281`	เวลาการส่งเป็นวินาที
`MDeferred: การหมดเวลาการเชื่อมต่อระหว่างผู้ใช้เปิดกับ zeus`	ข้อความสถานะ
`Sgeo`	ID ของผู้ส่ง
`Ramy@zeus`	ID ของผู้รับ
`H lines`	ข้อมูลส่วนหัวสำหรับข้อความ

ค่าเวลาใน sendmail

หากต้องการตั้งค่าข้อความการหมดเวลาใช้งานและช่วงเวลาในการประมวลผลคิว คุณต้องใช้รูปแบบเฉพาะสำหรับค่าเวลา

รูปแบบค่าเวลาคือ:

`-q`*`NumberUnit`*

โดยที่ *Number* คือค่าเลขจำนวนเต็มและ *Unit* คือตัวอักษรหน่วย *Unit* สามารถมีหนึ่งในค่า ต่อไปนี้:

ไอเท็ม	คำอธิบาย
s	วินาที
m	นาที
h	ชั่วโมง
d	วัน
w	สัปดาห์

หากไม่ได้ระบุ *Unit* ไว้ **sendmail** daemon จะใช้นาที (**m**) เป็นค่าดีฟอลต์ นี่คือสามตัวอย่างที่แสดงให้เห็นภาพของข้อกำหนดคุณสมบัติ เกี่ยวกับค่าเวลา:

```
/usr/sbin/sendmail -q15d
```

คำสั่งนี้แจ้งให้ **sendmail** daemon ทราบถึงการประมวลผลคิว ทุกๆ 15 วัน

```
/usr/sbin/sendmail -q15h
```

คำสั่งนี้แจ้งให้ **sendmail** daemon ทราบถึงการประมวลผลคิว ทุกๆ 15 ชั่วโมง

```
/usr/sbin/sendmail -q15
```

คำสั่งนี้แจ้งให้ **sendmail** daemon ทราบถึงการประมวลผลคิว ทุกๆ 15 นาที

คิวของเมลที่ไม่ทำงาน

ในบางกรณี คุณอาจพบว่าคิวไม่ทำงานโดยสาเหตุบางประการ คุณสามารถบังคับให้คิวทำงานโดยใช้แฟล็ก **–q** (โดยไม่ใส่ค่า)

คุณยังสามารถใช้แฟล็ก **–v** (verbose) เพื่อดูว่าเกิดอะไรขึ้น :

```
/usr/sbin/sendmail -q -v
```

คุณยังสามารถจำกัดงานกับเหล่านั้นกับตัวระบุคิวนั้นๆ ผู้ส่ง หรือผู้รับโดยใช้หนึ่งในตัวระบุคิว ตัวอย่างเช่น **–qRsally** จะจำกัดคิวที่ทำงานกับงานที่มีสตริง `sally` ในหนึ่งในแอดเดรสของผู้รับ เช่นเดียวกัน **–qS***string* จะจำกัดการทำงานกับผู้ส่งนั้นๆ และ **–qI***string* จะจำกัดมันกับตัวระบุคิวนั้นๆ

การตั้งช่วงเวลาการประมวลผลคิว

ค่าของแฟล็ก **–q** เมื่อ daemon เริ่มกำหนดช่วงเวลาที่ **sendmail** daemon ประมวลผลคิวของเมล

sendmail daemon โดยทั่วไปจะถูกสตาร์ทโดยไฟล์ `/etc/rc.tcpip` ตอนที่ระบบเริ่มต้น ไฟล์ `/etc/rc.tcpip` ประกอบด้วยตัวแปรชื่อ queue processing interval (QPI) ซึ่งมันใช้เพื่อระบุค่าของแฟล็ก **–q** เมื่อมันสตาร์ท **sendmail** daemon โดยดีฟอลต์ ค่าของ **qpi** คือ 30 นาที เพื่อระบุช่วงเวลาอื่นของการประมวลผลคิว :

1. แก้ไขไฟล์ `/etc/rc.tcpip` ด้วยเอดิเตอร์ที่คุณชอบ
2. หาบรรทัดที่กำหนดค่าให้กับตัวแปร *qpi* เช่น :

   ```
   qpi=30m
   ```

3. เปลี่ยนค่าที่กำหนดให้กับตัวแปร *qpi* เป็นค่าเวลาที่คุณต้องการ

การเปลี่ยนแปลงเหล่านี้จะมีผลเมื่อรีสตาร์ทระบบครั้งถัดไป ถ้าคุณต้องการให้การเปลี่ยนแปลงมีผลทันที หยุดและรีสตาร์ท **sendmail** daemon ระบุค่าแฟล็ก **–q** ใหม่ โปรดดู “การหยุด sendmail daemon” ในหน้า 56 and “การเริ่มต้น sendmail daemon”สำหรับข้อมูลเพิ่มเติม

การย้ายคิวของเมล

เมื่อโฮสต์หยุดทำงานเป็นระยะเวลานาน ข้อความหลายข้อความที่ถูกเราต์ (หรือผ่าน) ไปยังโฮสต์อาจถูกเก็บในคิวของเมลของคุณ เป็นผลให้ คำสั่ง sendmail จะใช้เวลานานในการเรียงลำดับของคิว ทำให้ประสิทธิภาพของระบบของคุณลดลง ถ้าคุณย้ายคิวไปยังที่อื่นชั่วคราวและสร้างคิวใหม่ คิวเก่าจะสามารถรันหลังจากนั้นเมื่อโฮสต์กลับมาให้บริการอีกครั้ง

เพื่อย้ายคิวไปยังสถานที่อื่นชั่วคราวและสร้างคิวใหม่ :

1. หยุด **sendmail** daemon โดยวิธีการต่อไปนี้ใน “การหยุด sendmail daemon” ในหน้า 56
2. ย้ายไดเร็กทอรีคิวทั้งหมดโดยการใส่ :

   ```
   cd /var/spool
   mv mqueue omqueue
   ```

3. รีสตาร์ท **sendmail** daemon โดยวิธีการต่อไปนี้ใน “การเริ่มต้น sendmail daemon”
4. โปรเซสคิวของเมลเก่าโดยการใส่ :

   ```
   /usr/sbin/sendmail -oQ/var/spool/omqueue -q
   ```

   แฟล็ก **–oQ** จะระบุไดเร็กทอรีของคิวอื่น แฟล็ก **–q** จะระบุเพื่อรันทุกงานในคิว เพื่อให้ได้รายงานเกี่ยวกับความคืบหน้าของการดำเนินการ ใช้แฟล็ก **–v**

   หมายเหตุ: การดำเนินการนี้อาจใช้เวลาสักครู่

5. ลบล็อกไฟล์และไดเร็กทอรีชั่วคราวเมื่อคิวว่าง โดยการใส่ :

   ```
   rm /var/spool/omqueue/*
   rmdir /var/spool/omqueue
   ```

การเริ่มต้น sendmail daemon

มีคำสั่งสองคำสั่งที่สตาร์ท **sendmail** daemon

หากต้องการสตาร์ท **sendmail** daemon ให้ป้อน คำสั่งต่อไปนี้:

```
startsrc -s sendmail -a "-bd -q15"
```

```
/usr/lib/sendmail -bd -q15
```

หาก **sendmail** daemon แอ็คทีฟอยู่แล้ว เมื่อคุณป้อนหนึ่งในคำสั่งเหล่านี้ คุณมองเห็นข้อความต่อไปนี้บนหน้าจอ:

```
ระบบย่อย sendmail แอ็คทีฟอยู่แล้ว อินสแตนซ์จำนวนมากไม่ได้รับการสนับสนุน
```

หาก **sendmail** daemon ไม่แอ็คทีฟแล้ว จากนั้น คุณมองเห็นข้อความที่บ่งชี้ว่า **sendmail** daemon ได้ถูกสตาร์ทแล้ว

การหยุด sendmail daemon

หากต้องการหยุด **sendmail** daemon ให้รันคำสั่ง **stopsrc -s sendmail**

หากไม่ได้สตาร์ท **sendmail** daemon พร้อมกับคำสั่ง **startsrc**:

- ให้ค้นหา ID กระบวนการสำหรับ **sendmail**
- ป้อนคำสั่ง **kill** *sendmail_pid* (โดยที่ *sendmail_pid* คือ ID กระบวนการของกระบวนการ **sendmail**)

การล็อกเมล

คำสั่ง **sendmail** จะบันทึกกิจกรรมของระบบเมลผ่าน **syslogd** daemon

syslogd daemon ต้องถูกตั้งค่าและรันอยู่เพื่อให้การล็อกเกิดขึ้น โดยเฉพาะอย่างยิ่ง ไฟล์ `/etc/syslog.conf` ควรประกอบด้วยบรรทัดที่ยกเลิกหมายเหตุ :

```
mail.debug          /var/spool/mqueue/log
```

ไม่เช่นนั้น ใช้เอดิเตอร์ที่คุณชอบเพื่อทำการแก้ไขนี้ ต้องแน่ใจว่าชื่อพาธถูกต้อง ถ้าคุณแก้ไขไฟล์ `/etc/syslog.conf` ขณะที่ **syslogd** daemon รันอยู่ รีเฟรช **syslogd** daemon โดยการพิมพ์คำสั่งต่อไปนี้ที่บรรทัดรับคำสั่ง :

```
refresh -s syslogd
```

ถ้าไฟล์ `/var/spool/mqueue/log` ไม่มีอยู่ คุณต้องสร้างมันโดยการพิมพ์คำสั่งต่อไปนี้ :

```
touch /var/spool/mqueue/log
```

ข้อความในล็อกไฟล์จะปรากฏในรูปแบบต่อไปนี้ :

แต่ละบรรทัดในล็อกไฟล์ประกอบด้วย เวลาประทับ ชื่อเครื่องที่สร้างมัน (สำหรับการล็อกจากหลายเครื่องบน local area network) คำว่า `sendmail:` และข้อความ ข้อความส่วนใหญ่เป็นลำดับของคู่ของ *name=value*

บรรทัดทั่วไปที่สุด 2 บรรทัดจะถูกล็อกเมื่อข้อความที่ถูกประมวลผลเป็นบรรทัด **receipt** และบรรทัด **delivery attempt** บรรทัด **receipt** จะล็อกผู้รับของข้อความ โดยจะมีหนึ่งล็อกต่อข้อความ บางฟิลด์อาจถูกตัดออก ฟิลด์ข้อความเหล่านี้คือ :

ไอเท็ม	คำอธิบาย
`from`	ระบุแอดเดรสของผู้ส่ง
`ขนาด`	ระบุขนาดของข้อความเป็นไบต์
`class`	ระบุคลาส (ตัวเลขที่มาก่อน) ของข้อความ
`pri`	ระบุตัวขึ้นต้นระดับความสำคัญของข้อความ (ใช้สำหรับการเรียงลำดับคิว)
`nrcpts`	ระบุจำนวนของผู้รับสำหรับข้อความนี้ (หลังการจากทำ alias และฟอร์เวิร์ด)
`proto`	ระบุโปรโตคอลที่ใช้เพื่อรับข้อความ ตัวอย่างเช่น ESMTP หรือ UNIX-to-UNIX Copy Program (UUCP)
`relay`	ระบุเครื่องจากที่มันได้รับ

บรรทัด **delivery attempt** จะถูกล็อกแต่ละครั้งที่มีการพยายามส่ง (ดังนั้นอาจมีหลายบรรทัดต่อข้อความถ้าการส่งต่างกัน หรือมีหลายผู้รับ) ฟิลด์เหล่านี้คือ :

ไอเท็ม	คำอธิบาย
ถึง	ประกอบด้วยลิสต์แบบคั่นด้วยคอมมาของผู้รับของ mailer นี้
ctladdr	ระบุ *controlling user* ซึ่งคือชื่อของผู้ใช้ที่มีสิทธิที่จะถูกใช้สำหรับส่ง
delay	ระบุเวลาหน่วงทั้งหมดระหว่างเวลาที่ข้อความนี้ได้รับและเวลาที่มันถูกส่ง
xdelay	ระบุจำนวนเวลาที่ต้องการในความพยายามส่งนี้
mailer	ระบุชื่อของ mailer ที่ใช้เพื่อส่งไปยังผู้รับนี้
relay	ระบุชื่อของโฮสต์ที่ยอมรับ (หรือปฏิเสธ) ผู้รับนี้
stat	ระบุสถานะการส่ง

เนื่องจากข้อมูลจำนวนมากสามารถถูกล็อก ล็อกไฟล์สามารถถูกจัดเป็นระดับของความสำเร็จ เริ่มต้นที่ระดับที่ 1 ระดับต่ำที่สุดจะมีเฉพาะสถานะการที่ไม่ปกติอย่างมากที่จะถูกล็อก ที่ระดับที่สูงที่สุด แม่ว่าเหตุการที่ไม่สำคัญจะถูกล็อก โดยทั่วไป ล็อกระดับ 10 และต่ำกว่านั้นจะเป็นข้อมูลที่เป็นประโยชน์ที่สุด ล็อกระดับที่สูงกว่า 64 ถูกจองไว้สำหรับการดีบัก ระดับจาก 11–64 ถูกจองไว้สำหรับข้อมูล verbose

ชนิดของกิจกรรมที่คำสั่ง **sendmail** ใส่ไว้ในล็อกไฟล์จะถูกระบุโดยอ็อพชัน **L** ในไฟล์ `/etc/mail/sendmail.cf`

การจัดการล็อก

เนื่องจากข้อมูลจะต่อท้ายล็อกไปเรื่อยๆ ไฟล์จะมีขนาดใหญ่มาก นอกจากนี้ เงื่อนไขข้อผิดพลาดสามารถทำให้เกิด entry ที่ไม่คาดคิดกับคิวของเมล เพื่อป้องกันไม่ให้คิวของเมลและล็อกไฟล์มีขนาดใหญ่เกินไป ใช้เชลล์สคริปต์ `/usr/lib/smdemon.cleanu`

สคริปต์นี้จะบังคับให้คำสั่ง **sendmail** ประมวลผลคิวและเก็บคัดลอกเก่าของล็อกไฟล์ 4 คัดลอก โดยใช้ชื่อ `log.0`, `log.1`, `log.2` และ `log.3` ทุกครั้งที่สคริปต์รันมันจะย้าย :

- `log.2` เป็น `log.3`
- `log.1` เป็น `log.2`
- `log.0` เป็น `log.1`
- `log` เป็น `log.0`

การรันสคริปต์นี้ทำให้ล็อกเริ่มต้นด้วยไฟล์ใหม่ รันสคริปต์นี้แบบแมนวลหรือช่วงเวลาที่ถูกระบุโดย **cron** daemon

ไฟล์บันทึกการรับส่งข้อมูล

ใช้แฟล็ก **–X** ของคำสั่ง **sendmail** เพื่อเซ็ตการบันทึกการรับส่งข้อมูล

การนำ **Simple Mail Transfer Protocols (SMTPs)** มาใช้จำนวนมาก ไม่ ได้นำคุณสมบัติของโปรโตคอลมาใช้ทั้งหมด ตัวอย่าง **SMTPs** ของคอมพิวเตอร์ส่วนบุคคล บางส่วนไม่เข้าใจบรรทัดการดำเนินการต่อในโค้ดตอบกลับ ซึ่งยากมากในการ ติดตาม ถ้าคุณสงสัยว่าจะมีปัญหานี้ คุณสามารถเซ็ตการบันทึกการรับส่งข้อมูลโดยใช้ แฟล็ก **–X** ตัวอย่าง เช่น:

```
/usr/sbin/sendmail -X /tmp/traffic -bd
```

คำสั่งนี้บันทึกการรับส่งข้อมูลทั้งหมดในไฟล์ `/tmp/traffic`

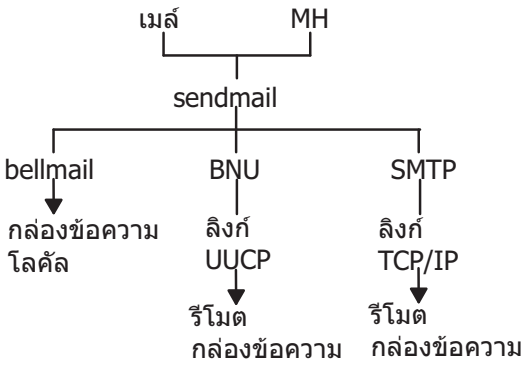
เนื่องจากคำสั่งนี้บันทึกข้อมูลจำนวนมากอย่างรวดเร็ว จึงไม่ควรใช้ ระหว่างการทำงานปกติ หลังจากรันคำสั่ง, จะบังคับให้ **errant** ส่งข้อมูลไปที่โฮสต์ของคุณ การรับส่งข้อมูลข้อความทั้งหมดเข้าและออก ของ **sendmail,** รวมถึงการรับส่งข้อมูล **SMTP** ขาเข้าจะถูกบันทึก ในไฟล์นี้

การใช้ **sendmail,** คุณสามารถบันทึกดัมพ์ของไฟล์ที่เปิด และแคชการเชื่อมต่อ โดยการส่งสัญญาณ **SIGUSR1** ผลลัพธ์ ถูกบันทึกที่ระดับความสำคัญ **LOG_DEBUG**

ล็อกของสถิติของ mailer

คำสั่ง **sendmail** จะติดตามปริมาณของเมลที่ถูกจัดการโดยโปรแกรม mailer แต่ละตัวที่อินเตอร์เฟสกับอัน

mailer เหล่านั้นถูกกำหนดในไฟล์ `/etc/mail/sendmail.cf`



รูปที่ 3. Mailers จะถูกใช้โดยคำสั่ง sendmail

การแสดงนี้เป็นชนิดของแผนภูมิเกี่ยวกับการจัดระเบียบแบบบนลงล่างโดย Mail และ MH อยู่ด้านบน สาขาของมันคือ bellmail, BNU และ SMTP ใต้ระดับก่อนหน้านี้เป็นโลคัลเมลบ็อกซ์, UUCP ลิงก์ และ TCP/IP ลิงก์ตามลำดับ ใต้ UUCP ลิงก์เป็นรีโมตเมลบ็อกซ์ และใต้ TCP/IP ลิงก์เป็นรีโมตเมลบ็อกซ์

เพื่อสตาร์ทการรวมสถิติของ mailer สร้างไฟล์ `/etc/mail/statistics` โดยการพิมพ์ต่อไปนี้ :

```
touch /etc/mail/statistics
```

ถ้าคำสั่ง **sendmail** พบข้อผิดพลาดเมื่อพยายามบันทึกข้อมูลสถิติ คำสั่งจะเขียนข้อความผ่านรูทีนย่อย **syslog** ข้อผิดพลาดเหล่านี้จะไม่มีผลกับการดำเนินการอื่นของคำสั่ง **sendmail**

คำสั่ง **sendmail** จะอัพเดตข้อมูลในไฟล์แต่ละครั้งที่มันประมวลผลเมล ขนาดของไฟล์จะไม่โตขึ้น แต่สมาชิกในไฟล์จะโตขึ้น มันจะแทนปริมาณของเมลตั้งแต่เวลาที่คุณสร้างหรือรีเซ็ตไฟล์ `/etc/mail/statistics`

การแสดงข้อมูล mailer

สถิติที่เก็บในไฟล์ `/etc/mail/statistics` จะอยู่ในรูปแบบที่ไม่สามารถอ่านได้เป็นเท็กซ์ไฟล์

เพื่อแสดงสถิติของ mailer พิมพ์ข้อความต่อไปนี้ที่จุดรับคำสั่ง :

```
/usr/sbin/mailstats
```

นี่จะอ่านข้อมูลในไฟล์ /etc/mail/statistics จัดรูปแบบมัน และเขียนมันไปยังเอาต์พุตมาตรฐาน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับเอาต์พุตของคำสั่ง **/usr/sbin/mailstats** อ่านคำอธิบายของมันใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 3*

sendmail Mail Filter API

sendmail Mail Filter API (อ้างอิงเป็น *Milter*) ช่วยให้โปรแกรมของบริษัทอื่นสามารถเข้าถึงข้อความเมลตามที่ถูกประมวลผลไว้เพื่อกรองข้อมูล–เมต้าและเนื้อหา

ข้อกำหนดฟิลเตอร์ sendmail

เนื่องจากฟิลเตอร์ใช้ threads ฟิลเตอร์ต้องไม่มีผลกับการทำงานของ thread คุณสามารถตั้งค่าฟิลเตอร์เพื่อให้แน่ใจถึงความเข้ากันได้กับ threads

หลายๆระบบปฏิบัติการให้การสนับสนุนสำหรับ POSIX threads ในไลบรารี C มาตรฐาน คอมไพลเลอร์จะแฟล็กกับลิงก์ด้วยการสนับสนุน thread ที่แตกต่างกันโดยขึ้นอยู่กับคอมไพลเลอร์และตัวเชื่อมโยงที่ใช้ ถ้าคุณไม่แน่ใจว่าโลคัลแฟล็กใดที่ใช้ ตรวจสอบ `Makefile` ในไดเร็กทอรีย่อย `obj.*/libmilter` build ที่เหมาะสม

หมายเหตุ: เนื่องจากฟิลเตอร์ใช้ threads มันอาจจำเป็นที่จะเปลี่ยนการจำกัดของกระบวนการบนฟิลเตอร์ของคุณ ตัวอย่างเช่น คุณอาจต้องการใช้ setrlimit เพื่อเพิ่มจำนวนของ file descriptors ที่เปิดถ้าฟิลเตอร์ของคุณยุ่งอยู่ ไม่เช่นนั้นเมลจะถูกปฏิเสธ

การตั้งค่าฟิลเตอร์ sendmail

ใช้แนวทางนี้เพื่อระบุฟิลเตอร์ที่คุณต้องการขณะตั้งค่า **sendmail**

ระบุฟิลเตอร์โดยใช้คีย์ตัวอักษร X (สำหรับ eXternal) ในตัวอย่างต่อไปนี้ ฟิตเตอร์ 3 ตัวจะถูกระบุ :

```
Xfilter1, S=local:/var/run/f1.sock, F=R
Xfilter2, S=inet6:999@localhost, F=T, T=C:10m;S:1s;R:1s;E:5m
Xfilter3, S=inet:3333@localhost
```

คุณสามารถระบุฟิลเตอร์ในไฟล์ .mc ของคุณโดยในไวยากรณ์ต่อไปนี้ :

```
INPUT_MAIL_FILTER(`filter1', `S=local:/var/run/f1.sock, F=R')
INPUT_MAIL_FILTER(`filter2', `S=inet6:999@localhost, F=T, T=C:10m;S:1s;R:1s;E:5m')
INPUT_MAIL_FILTER(`filter3', `S=inet:3333@localhost')
```

โดยที่ filter(*number*) เป็นชื่อของฟิลเตอร์ของคุณ บรรทัดแรกของไวยากรณ์จะระบุว่าฟิลเตอร์เชื่อมอยู่กับซ็อกเก็ตใน UNIX โดเมนในไดเร็กทอรี `/var/run` บรรทัดที่สองระบุว่าฟิลเตอร์ใช้ IPv6 ซ็อกเก็ตบนพอร์ต 999 ของโลคัลโฮสต์ บรรทัดที่สามระบุว่าฟิลเตอร์ใช้ IPv4 ซ็อกเก็ตบนพอร์ต 3333 ของโลคัลโฮสต์

`F=` แสดงว่าแฟล็กใดต่อไปนี้ถูกใช้ :

ไอเท็ม	คำอธิบาย
R	ปฏิเสธการเชื่อมต่อถ้าฟิลเตอร์ไม่พร้อมใช้งาน
T	การเชื่อมต่อล้มเหลวชั่วคราวถ้าฟิลเตอร์ไม่พร้อมใช้งาน

ถ้าไม่มีแฟล็กถูกระบุ ข้อความจะถูกผ่านไปยัง **sendmail** เหมือนกับว่าฟิลเตอร์ไม่มีอยู่

โดยการระบุค่าสำหรับ T= คุณสามารถใช้ฟิลเตอร์เพื่อทับค่าดีฟอลต์ของการหมดเวลาที่ถูกใช้โดย **sendmail** T= จะเท่ากับการใช้ฟิลด์ต่อไปนี้ :

ไอเท็ม	คำอธิบาย
C	การหมดเวลาสำหรับการเชื่อมต่อกับฟิลเตอร์ (ถ้าเป็น 0 ใช้ค่าการหมดเวลาของระบบ)
S	การหมดเวลาใช้งานสำหรับการส่งข้อมูลจาก MTA ไปยังตัวกรอง
R	ค่าการหมดเวลาสำหรับการอ่านการตอบจากฟิลเตอร์
E	การหมดเวลาโดยรวมระหว่างการส่งการเตือนสิ้นสุดข้อความไปยังฟิลเตอร์และรอสำหรับการตอบรับสุดท้าย

ดังที่ระบุในตัวอย่างก่อนหน้านี้ ตัวคั่นระหว่างแต่ละค่าการหมดเวลาคือเซมิโคลอน (;) และตัวคั่นระหว่างแต่ละการทำให้เท่ากันคือคอมมา (,)

ค่าดีฟอลต์สำหรับการหมดเวลาเป็นดังต่อไปนี้ :

```
T=C:0m;S:10s;R:10s;E:5m
```

โดยที่ s เป็นวินาที และ m เป็นนาที

อ็อพชัน **InputMailFilters** จะกำหนดว่าฟิลเตอร์ใดจะถูกใช้และในลำดับใด

หมายเหตุ: ถ้าไม่ระบุ **InputMailFilters** จะไม่มีฟิลเตอร์ใดถูกใช้
อ็อพชัน **InputMailFilters** จะถูกตั้งโดยอัตโนมัติโดยขึ้นอยู่กับลำดับของคำสั่ง INPUT_MAIL_FILTER ในไฟล์ .mc ของคุณ คุณสามารถรีเซ็ตค่านี้โดยการตั้งค่าสำหรับ confINPUT_MAIL_FILTERS ในไฟล์ .mc ของคุณ ตัวอย่างเช่น ถ้าอ็อพชัน **InputMailFilters** ถูกตั้งเป็น :

```
InputMailFilters=filter1, filter2, filter3
```

ฟิลเตอร์ 3 ฟิลเตอร์จะถูกเรียกใช้ในลำดับเดียวกันที่มันถูกระบุ

โดยการใช้ MAIL_FILTER() แทนที่ INPUT_MAIL_FILTER() ในไฟล์ .mc ของคุณ คุณสามารถกำหนดฟิลเตอร์โดยไม่ต้องเพิ่มมันเข้ากับลิสต์ของอินพุตฟิลเตอร์

ฟังก์ชันการควบคุมไลบรารี

ตัวกรอง sendmail เรียกฟังก์ชันการควบคุมไลบรารี เพื่อตั้งค่าพารามิเตอร์ libmilter ก่อนที่จะส่ง การควบคุมไปยัง libmilter พารามิเตอร์ libmilter ถูกตั้งค่าโดยเรียกฟังก์ชัน **smfi_main** ตัวกรองยังเรียกฟังก์ชัน **smfi_register** เพื่อลงทะเบียน callbacks แต่ละฟังก์ชันส่งคืนค่า MI_SUCCESS หรือ MI_FAILURE เพื่อบ่งชี้สถานะของการดำเนินการ ฟังก์ชันเหล่านี้ ไม่ได้สื่อสารกับ mail transfer agent (MTA) แต่เปลี่ยนสถานะไลบรารี ซึ่งสื่อสารกับ MTA ภายในฟังก์ชัน **smfi_main**

ตารางที่ 1. ฟังก์ชันการควบคุมไลบรารี

ไอเท็ม	description
smfi_opensocket	**ฟังก์ชัน smfi_opensocket สร้าง ซ็อกเก็ตอินเตอร์เฟส**
smfi_register	ฟังก์ชัน **smfi_register** ลงทะเบียนตัวกรอง
smfi_setconn	ฟังก์ชัน **smfi_setconn** ระบุซ็อกเก็ตที่ต้องการใช้
smfi_settimeout	ฟังก์ชัน **smfi_settimeout** ตั้งค่าการหมดเวลา
smfi_setbacklog	ฟังก์ชัน **smfi_setbacklog** นิยามขนาดคิว **listen** (2) ขาเข้า
smfi_setdbg	ฟังก์ชัน **smfi_setdbg** ตั้งค่าระดับการดีบัก (การติดตาม) ไลบรารี `milter`
smfi_stop	ฟังก์ชัน **smfi_stop** เป็นสาเหตุทำให้ระบบปิดตามลำดับ
smfi_main	ฟังก์ชัน **smfi_main** ส่งการควบคุมไปยัง `libmilter`

ฟังก์ชัน smfi_opensocket: วัตถุประสงค์

ฟังก์ชัน **smfi_opensocket** พยายามสร้างซ็อกเก็ตอินเตอร์เฟส mail transfer agents (MTA) ที่ถูกใช้เพื่อเชื่อมต่อกับตัวกรอง

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_opensocket(
bool rmsocket
);
```

คำอธิบาย

ฟังก์ชัน **smfi_opensocket** ถูกเรียกจากบรรทัดหลักของโปรแกรมเท่านั้น หลังจากที่เรียกฟังก์ชัน **smfi_setconn** และฟังก์ชัน **smfi_register** แต่ก่อนที่จะเรียกฟังก์ชัน **smfi_main** ฟังก์ชัน **smfi_opensocket** สร้างซ็อกเก็ตที่ระบุไว้ก่อนหน้านี้โดยเรียกฟังก์ชัน **smfi_setconn** ที่เป็นอินเตอร์เฟสระหว่าง MTAs และตัวกรอง ฟังก์ชัน **smfi_opensocket** อนุญาตให้การเรียกแอ็พพลิเคชันสร้างซ็อกเก็ต หากฟังก์ชัน **smfi_opensocket** ไม่ได้ถูกเรียก ฟังก์ชัน **smfi_main** จะเรียกฟังก์ชันแบบ implicitly

อาร์กิวเมนต์

ตารางที่ 2. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
rmsocket	แฟล็กบ่งชี้ว่าไลบรารีต้องพยายามลบซ็อกเก็ตโดเมน UNIX ออกก่อนที่จะพยายาม สร้างขึ้นใหม่

ค่าส่งคืน

ฟังก์ชัน **smfi_opensocket** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- ซ็อกเก็ตอินเตอร์เฟสไม่สามารถสร้างขึ้นได้
- ค่า *rmsocket* เป็นจริงและซ็อกเก็ต ไม่สามารถตรวจสอบได้ หรือซ็อกเก็ตที่มีอยู่ไม่สามารถลบออกได้

- ฟังก์ชัน **smfi_setconn** หรือฟังก์ชัน **smfi_register** ไม่ได้ถูกเรียก

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน smfi_register"

"ฟังก์ชัน smfi_setconn" ในหน้า 65

ฟังก์ชัน smfi_register:
วัตถุประสงค์

ฟังก์ชัน **smfi_register** ลงทะเบียนชุดของฟังก์ชัน sendmail filter callback

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_register(
smfiDesc descr)
);
```

คำอธิบาย

ฟังก์ชัน **smfi_register** สร้างตัวกรอง sendmail โดยใช้ข้อมูลที่จัดเตรียมไว้ในอาร์กิวเมนต์ `smfiDesc` ฟังก์ชัน **smfi_register** ต้องถูกเรียกก่อนฟังก์ชัน **smfi_main**

หมายเหตุ: การเรียกที่เป็นผลสำเร็จ จำนวนมาไปยังฟังก์ชัน **smfi_register** ภายในการประมวลผลเดี่ยว ไม่ได้รับอนุญาต เฉพาะตัวกรอง sendmail เดี่ยวเท่านั้นที่สามารถลงทะเบียน ได้เป็นผลสำเร็จ อย่างไรก็ตาม ไลบรารีไม่สามารถตรวจสอบว่าข้อจำกัดถูกปฏิบัติตาม

ฟิลด์ xxfi_flags ต้องมี bitwise หรือค่าศูนย์หรือค่าใดๆ ต่อไปนี้ ซึ่งกล่าวถึงการดำเนินของตัวกรอง sendmail สามารถนำมาใช้ได้

ตารางที่ 3. ค่า

ไอเท็ม	คำอธิบาย
SMFIF_ADDHDRS	ฟังก์ชัน **smfi_addheader** เพิ่มส่วนหัว
SMFIF_CHGHDRS	ฟังก์ชัน **smfi_chgheader** แก้ไขส่วนหัว หรือลบส่วนหัว
SMFIF_CHGBODY	ฟังก์ชัน **smfi_replacebody** แทนที่เนื้อความ ในระหว่างการกรอง ตัวกรองมีผลการทำงานที่ได้รับผลกระทบ หากตัวกรองอื่นทำการกรองเนื้อหาหลังตัวกรองนี้
SMFIF_ADDRCPT	ฟังก์ชัน **smfi_addrcpt** เพิ่มผู้รับให้กับข้อความ
SMFIF_ADDRCPT_PAR	ฟังก์ชัน **smfi_addrcpt_par** เพิ่มผู้รับ ซึ่งประกอบด้วยอาร์กิวเมนต์ simple mail transfer protocol (ESMTP) ที่ขยายเพิ่ม
SMFIF_DELRCPT	ฟังก์ชัน **smfi_delrcpt** ลบผู้รับจาก ข้อความ
SMFIF_QUARANTINE	ฟังก์ชัน **smfi_quarantine** กักกันข้อความ
SMFIF_CHGFROM	ฟังก์ชัน **smfi_chgfrom** แก้ไขผู้ส่งจดหมาย (เมลจาก)

ตารางที่ 3. ค่า (ต่อ)

ไอเท็ม	คำอธิบาย
SMFIF_SETSYMLIST	ฟังก์ชัน **smfi_setsymlist** ส่งชุดของสัญลักษณ์ (แมโคร) ที่จำเป็นต้องมี

อาร์กิวเมนต์

ตารางที่ 4. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย

descr	คำอธิบายตัวกรองของชนิด `smfiDesc` ที่กล่าวถึงฟังก์ชันตัวกรอง โครงสร้างมีสมาชิกต่อไปนี้:
	`struct smfiDesc`
	`{`
	` char  *xxfi_name; /* filter name */`
	` int  xxfi_version; /* version code`
	`-- do not change */`
	` unsigned long xxfi_flags; /* flags */`
	
	` /* connection info filter */`
	` sfsistat (*xxfi_connect)(SMFICTX *,`
	`char`
	`*, _SOCK_ADDR *);`
	` /* SMTP HELO command filter */`
	` sfsistat (*xxfi_helo)(SMFICTX *,`
	`char *);`
	` /* envelope sender filter */`
	` sfsistat (*xxfi_envfrom)(SMFICTX *,`
	`char **);`
	` /* envelope recipient filter */`
	` sfsistat (*xxfi_envrcpt)(SMFICTX *,`
	`char **);`
	` /* header filter */`
	` sfsistat (*xxfi_header)(SMFICTX *,`
	`char *,`
	` char *);`
	` /* end of header */`
	` sfsistat (*xxfi_eoh)(SMFICTX *);`
	` /* body block */`
	` sfsistat (*xxfi_body)(SMFICTX *,`
	`unsigned char`
	`*, size_t);`
	` /* end of message */`
	` sfsistat (*xxfi_eom)(SMFICTX *);`
	` /* message aborted */`
	` sfsistat (*xxfi_abort)(SMFICTX *);`
	` /* connection cleanup */`
	` sfsistat (*xxfi_close)(SMFICTX *);`
	
	` /* any unrecognized or unimplemented`
	`command filter */`
	` sfsistat (*xxfi_unknown)(SMFICTX`
	`*, const char *);`
	
	` /* SMTP DATA command filter */`
	` sfsistat (*xxfi_data)(SMFICTX *);`
	
	` /* negotiation callback */`
	` sfsistat (*xxfi_negotiate)(SMFICTX *,`
	`  unsigned long, unsigned long,`
	`unsigned long,`
	` unsigned long,`
	`  unsigned long *, unsigned long *,`
	`unsigned long *,`
	`unsigned long *`
	`);`
	`};`
	
	ค่า NULL สำหรับฟังก์ชัน callback ใดๆ บ่งชี้ว่า ตัวกรองไม่ได้ประมวลผลชนิดของข้อมูลที่กำหนดเอง และส่งคืน **SMFIS_CONTINUE**

ตารางที่ 4. อาร์กิวเมนต์ (ต่อ)

ไอเท็ม	คำอธิบาย
headerf	ชื่อส่วนหัวเป็นสตริง null-terminated ที่ไม่ใช่ค่า NULL
headerv	ค่าส่วนหัวที่ต้องถูกเพิ่มสามารถเป็นสตริง non-NULL null-terminated หรือสตริงว่าง

ค่าส่งคืน

ฟังก์ชัน **smfi_register** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- การจัดสรรหน่วยความจำล้มเหลว
- เวอร์ชันที่ทำงานร่วมกันไม่ได้หรือค่าแฟล็กที่ผิดกฎเกณฑ์

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน smfi_addheader" ในหน้า 77

"ฟังก์ชัน smfi_chgheader" ในหน้า 79

"ฟังก์ชัน smfi_replacebody" ในหน้า 86

"ฟังก์ชัน smfi_addrcpt" ในหน้า 83

"ฟังก์ชัน smfi_addrcpt_par" ในหน้า 84

"ฟังก์ชัน smfi_delrcpt" ในหน้า 85

"ฟังก์ชัน smfi_quarantine" ในหน้า 88

"ฟังก์ชัน smfi_chgfrom" ในหน้า 82

"ฟังก์ชัน smfi_setsymlist" ในหน้า 105

ฟังก์ชัน smfi_setconn:

วัตถุประสงค์

ฟังก์ชัน **smfi_setconn** ที่ตั้งค่าผ่านตัวกรองนี้สามารถสื่อสารกับคำสั่ง **sendmail**

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_setconn(
char *oconn;
);
```

คำอธิบาย

ฟังก์ชัน **smfi_setconn** ต้องถูกเรียกก่อนที่จะเรียกฟังก์ชัน **smfi_main**

ตัวกรอง ต้องไม่รันเป็น root เมื่อสื่อสารผ่าน UNIX หรือโลคัลโดเมนซ็อกเก็ต

สิทธิ สำหรับ UNIX หรือโลคัลซ็อกเก็ตต้อง มีการตั้งค่าเป็น 0600 (สิทธิในการอ่านหรือเขียนสำหรับ เจ้าของหรือกลุ่มของซ็อกเก็ตเท่านั้น) หรือ 0660 (สิทธิในการอ่าน/เขียน สำหรับเจ้าของซ็อกเก็ตและกลุ่ม) สิทธิเหล่านี้มีประโยชน์ หากใช้อ็อพชัน **sendmail RunAsUser**

สิทธิสำหรับ UNIX หรือ โลคัลโดเมนซ็อกเก็ตมีการกำหนดโดย **umask** ซึ่งต้องมีการตั้งค่าเป็น 007 หรือ 077 สำหรับระบบปฏิบัติการ เช่น ระบบปฏิบัติการ Solaris ที่ไม่ใช่สิทธิของซ็อกเก็ต ให้วางซ็อกเก็ตลงใน ไดเร็กทอรีที่ได้รับการปกป้อง

อาร์กิวเมนต์

ตารางที่ 5. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย

oconn	แอดเดรสของซ็อกเก็ตการสื่อสารที่ต้องการ แอดเดรสต้องเป็นสตริง NULL-terminated ในรูปแบบใน `proto:address`:
	`* {unix|local}:/path /to/file -- A named pipe.`
	`* inet:port @{hostname|ip-address} -- An IPV4 socket.`
	`* inet6:port @{hostname|ip-address} -- An IPV6 socket.`

ค่าส่งคืน

ฟังก์ชัน **smfi_setconn** ไม่ได้ล้มเหลว หากมีแอดเดรสที่ไม่ถูกต้อง อย่างไรก็ตาม ฟังก์ชัน **smfi_setconn** ล้มเหลวในการตั้งค่าซ็อกเก็ตหากไม่มีหน่วยความจำ ความล้มเหลวถูกตรวจพบเฉพาะในฟังก์ชัน **smfi_main** เท่านั้น

ข้อมูลที่เกี่ยวข้อง

“ฟังก์ชัน smfi_main” ในหน้า 69

ฟังก์ชัน smfi_settimeout: วัตถุประสงค์

ฟังก์ชัน **smfi_settimeout** ตั้งค่าตัวกรองการหมดเวลาของ I/O

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_settimeout((
int otimeout
);
```

คำอธิบาย

ฟังก์ชัน **smfi_settimeout** ถูกเรียกจากฟังก์ชัน **smfi_main** เท่านั้น ฟังก์ชัน **smfi_settimeout** ตั้งค่าช่วงระยะเวลา (ในหน่วยวินาที) สำหรับพารามิเตอร์ `libmilter` เพื่อรอการสื่อสาร mail transfer agent (MTA) (อ่านหรือเขียน) ก่อนที่จะหมดเวลา

หมายเหตุ: หากฟังก์ชัน **smfi_settimeout** ไม่ได้ถูกเรียก ระยะเวลาการหมดเวลาที่เป็นค่าดีฟอลต์คือ 7210 วินาที

อาร์กิวเมนต์

ตารางที่ 6. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
otimeout	ระยะเวลาในหน่วยวินาทีสำหรับพารามิเตอร์ `libmilter` จะรอ MTA ก่อนที่จะหมดเวลา ค่า *otimeout* ต้องมากกว่าศูนย์ หากค่า *otimeout* เป็นศูนย์ พารามิเตอร์ `libmilter` จะไม่รอ MTA

ค่าส่งคืน

ฟังก์ชัน **smfi_settimeout** ยังคงส่งคืนค่า MI_SUCCESS

ข้อมูลที่เกี่ยวข้อง

smfi_main

ฟังก์ชัน smfi_setbacklog:
วัตถุประสงค์

ฟังก์ชัน **smfi_setbacklog** ตั้งค่า **listen(2)** backlog ของตัวกรอง

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_setbacklog(
int obacklog
);
```

คำอธิบาย

ฟังก์ชัน **smfi_setbacklog** ถูกเรียกก่อนที่จะเรียกฟังก์ชัน **smfi_main** ฟังก์ชัน **smfi_setbacklog** ตั้งค่า backlog ซ็อกเก็ตขาเข้าซึ่งถูกใช้โดยค่า **listen(2)** backlog หากฟังก์ชัน **smfi_setbacklog** ไม่ได้ถูกเรียก ระบบปฏิบัติการดีฟอลต์จะถูกนำมาใช้

อาร์กิวเมนต์

ตารางที่ 7. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
obacklog	จำนวนของการเชื่อมต่อขาเข้ายอมให้ใช้ในคิว listen

ค่าส่งคืน

ฟังก์ชัน **smfi_setbacklog** ส่งคืน MI_FAILURE หากอาร์กิวเมนต์ *obacklog* ถูกตั้งค่าให้น้อยกว่าหรือเท่ากับ null

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน smfi_main" ในหน้า 69

ฟังก์ชัน smfi_setdbg:
วัตถุประสงค์

ฟังก์ชัน **smfi_setdbg** ตั้งค่าระดับของการดีบัก (การติดตาม) สำหรับไลบรารี `milter`

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_setdbg(
 int level;
);
```

คำอธิบาย

ฟังก์ชัน **smfi_setdbg** ตั้งค่าระดับของการดีบักภายในของไลบรารี `milter` ไปเป็นระดับใหม่ เพื่อให้รายละเอียดของโค้ดสามารถติดตามได้ ระดับศูนย์ ปิดการดีบัก ระดับที่สูงกว่า (จำนวนจริงบวกเยอะกว่า) จะมีรายละเอียด ในการดีบักมากกว่า หกคือค่าปัจจุบัน สูงที่สุด และค่าที่มีประโยชน์

อาร์กิวเมนต์

ตารางที่ 8. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
level	ระดับของการดีบักใหม่

ค่าส่งคืน

ฟังก์ชัน **smfi_setdbg** ส่งคืนค่า MI_SUCCESS ตามค่าดีฟอลต์

ฟังก์ชัน smfi_stop:
วัตถุประสงค์

ฟังก์ชัน **smfi_stop** ปิด `milter` การเชื่อมต่อไม่ได้ถูกยอมรับ หลังจากการเรียกในครั้งนี้

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_stop(void);
);
```

คำอธิบาย

ฟังก์ชัน **smfi_stop** ถูกเรียกจากฟังก์ชัน Callback หรือรูทีน error-handling ในเวลาใดๆ รูทีน **smfi_stop** ไม่ได้อนุญาตให้ใช้การเชื่อมต่อใหม่ อย่างไรก็ตาม ฟังก์ชันไม่ได้รอการเชื่อมต่อที่มีอยู่ (เธรด) เพื่อยกเลิก ฟังก์ชันนี้เป็นสาเหตุทำให้ฟังก์ชัน **smfi_main** ส่งคืนการเรียกโปรแกรมซึ่งสามารถออกหรือ warm-restart ได้

อาร์กิวเมนต์

ตารางที่ 9. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
void	อาร์กิวเมนต์นี้ไม่ได้ใช้ค่าใดๆ

ค่าส่งคืน

ฟังก์ชัน **smfi_stop** ส่งคืนค่า SMFI_CONTINUE ในกรณีต่อไปนี้:

- รูทีนภายในเป็นสาเหตุทำให้ไลบรารี `milter` หยุดทำงาน
- รูทีนเป็นสาเหตุทำให้ไลบรารี `milter` หยุดทำงาน
- กระบวนการที่สตาร์ทแล้วไม่สามารถหยุดทำงานได้

ตัวอย่าง

```
int     ret;
SMFICTX *ctx;
...
ret = smfi_addheader(ctx, "Content-Type",
"multipart/mixed;\n\tboundary=\"foobar\"");
```

ข้อมูลที่เกี่ยวข้อง

ฟังก์ชัน Callback

ฟังก์ชัน smfi_main:

วัตถุประสงค์

ฟังก์ชัน **smfi_main** ส่งการควบคุมไปยังการวนซ้ำเหตุการณ์ `libmilter`

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_main(
);
```

คำอธิบาย

ฟังก์ชัน **smfi_main** ถูกเรียกหลังจากกำหนดค่าเริ่มต้นของตัวกรองที่เสร็จสมบูรณ์

ค่าส่งคืน

ฟังก์ชัน **smfi_main** ส่งคืนค่า MI_FAILURE หากการเชื่อมต่อไม่สามารถสร้างขึ้นได้ และฟังก์ชันส่งคืน MI_SUCCESS

ความล้มเหลว เกิดขึ้นด้วยเหตุผลที่แตกต่างกันและเหตุผลสำหรับความล้มเหลวเป็นสิ่งที่ดี ตัวอย่างเช่น การส่งแอดเดรสที่ไม่ถูกต้องภายในฟังก์ชัน **smfi_setconn** ทำให้ฟังก์ชันล้มเหลว

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน smfi_setconn" ในหน้า 65

ฟังก์ชันการเข้าถึงข้อมูล

ฟังก์ชันการเข้าถึงข้อมูลถูกเรียกจากภายในฟังก์ชัน callback ซึ่งนิยามไว้ภายในตัวกรองเพื่อเข้าถึงข้อมูลเกี่ยวกับ การเชื่อมต่อหรือข้อความปัจจุบัน

ตารางที่ 10. ฟังก์ชันการเข้าถึงข้อมูล

ไอเท็ม	คำอธิบาย
smfi_getsymal	ฟังก์ชัน **smfi_getsymal** ส่งคืนค่าของ สัญลักษณ์
smfi_getpriv	ฟังก์ชัน **smfi_getpriv** ดึงตัวชี้ ข้อมูลส่วนบุคคล
smfi_setpriv	ฟังก์ชัน **smfi_setpriv** ตั้งค่าตัวชี้ข้อมูลส่วนบุคคล
smfi_setreply	ฟังก์ชัน **smfi_setreply** ตั้งค่าโค้ดการตอบกลับที่ระบุเฉพาะ ซึ่งต้องใช้
smfi_setmlreply	ฟังก์ชัน **smfi_setmlreply** ตั้งค่าการตอบกลับแบบหลายบรรทัด ที่ระบุเฉพาะ ซึ่งต้องใช้

ฟังก์ชัน smfi_getsymval:
วัตถุประสงค์

ฟังก์ชัน **smfi_getsymval** ดึงค่าของแมโคร **sendmail**

ไวยากรณ์

```
#include <libmilter/mfapi.h>
char* smfi_getsymval(
SMFICTX *ctx,
char *headerf,
char *symname
);
```

คำอธิบาย

ฟังก์ชัน **smfi_getsymval** ถูกเรียกจากฟังก์ชัน **xxfi_*** **callback** เพื่อเพิ่ม ส่วนหัวไปยังข้อความ นิยามแมโครขึ้นอยู่กับฟังก์ชันที่ถูกเรียก

โดยค่าดีฟอลต์ แมโครต่อไปนี้คือค่าที่ถูกต้อง:

ตารางที่ 11. คำอธิบาย

ไอเท็ม	คำอธิบาย
xxfi_connect	**daemon_name, if_name, if_addr, j, _**
xxfi_hello	**tls_version, cipher, cipher_bits, cert_subject, cert_issuer**
xxfi_envfrom	**i, auth_type, auth_authen, auth_ssf, auth_author, mail_mailer, mail_host, mail_addr**
xxfi_envrcpt	**rcpt_mailer, rcpt_host, rcpt_addr**
xxfi_data	ไม่มี
xxfi_eoh	ไม่มี
xxfi_eom	**msg_id**

แมโครทั้งหมดยังคงได้รับผลกระทบจากจุด ที่ได้รับจนกว่าจะสิ้นสุดการเชื่อมต่อสำหรับฟังก์ชัน **xxfi_connect, xxfi_hello**

แมโครทั้งหมด ได้รับผลกระทบจนกว่าจะสิ้นสุดข้อความสำหรับฟังก์ชัน **xxfi_envfrom** และฟังก์ชัน **xxfi_eom**

แมโครทั้งหมดยังคงได้รับผลกระทบ สำหรับแต่ละผู้รับฟังก์ชัน **xxfi_envrcpt**

รายการแมโคร สามารถเปลี่ยนแปลงได้โดยใช้อ็อพชัน **confMILTER_MACROS_*** ใน **sendmail.mc** ขอบเขตของแมโครบางตัว ถูกกำหนดไว้เมื่อตั้งค่าโดยคำสั่ง **sendmail** สำหรับคำอธิบายของค่าแมโคร โปรดดู *คู่มือการติดตั้ง Sendmail และการใช้งาน*

อาร์กิวเมนต์

ตารางที่ 12. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
symname	ชื่อของแมโคร **sendmail** แมโครตัวอักษรเดียว สามารถล้อมรอบด้วยวงเล็บปีกกา (″{″ และ ″}″) ชื่อแมโครที่ยาวกว่า ต้องล้อมรอบอยู่ในวงเล็บปีกกาตามที่อยู่ในไฟล์ **sendmail.cf**

ค่าส่งคืน

ฟังก์ชัน **smfi_getsymval** ส่งคืนค่าของแมโครที่กำหนดไว้เป็น null-terminated และ ฟังก์ชัน **smfi_getsymval** ส่งคืนค่า NULL หากไม่ได้กำหนดแมโครไว้

ข้อมูลที่เกี่ยวข้อง

“ฟังก์ชัน xxfi_connect callback” ในหน้า 91

“ฟังก์ชัน xxfi_helo callback” ในหน้า 92

“ฟังก์ชัน xxfi_envfrom callback” ในหน้า 92

“ฟังก์ชัน xxfi_envrcpt callback” ในหน้า 93

“ฟังก์ชัน xxfi_data callback” ในหน้า 94

“ฟังก์ชัน xxfi_eoh callback” ในหน้า 97

“ฟังก์ชัน xxfi_eom callback” ในหน้า 99

ฟังก์ชัน smfi_getpriv:
วัตถุประสงค์

ฟังก์ชัน **smfi_getpriv** ดึงตัวชี้ข้อมูลการเชื่อมต่อเฉพาะสำหรับการเชื่อมต่อนี้

ไวยากรณ์

```
#include <libmilter/mfapi.h>
void* smfi_getpriv(
SMFICTX *ctx
);
```

คำอธิบาย

ฟังก์ชัน **smfi_getpriv** สามารถเรียกอยู่ในฟังก์ชัน **xxfi_*** **callback** ใดๆ

อาร์กิวเมนต์

ตารางที่ 13. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter

ค่าส่งคืน

ฟังก์ชัน **smfi_getpriv** ที่ถูกเก็บไว้โดยการส่งคืนตัวชี้ข้อมูลส่วนบุคคลที่เก็บไว้โดยการเรียก ก่อนฟังก์ชัน **smfi_setpriv** มิฉะนั้น ฟังก์ชัน **smfi_setpriv** จะส่งคืน NULL หากไม่ได้ตั้งค่าไว้

ข้อมูลที่เกี่ยวข้อง

“ฟังก์ชัน smfi_setpriv”

ฟังก์ชัน smfi_setpriv:
วัตถุประสงค์

ฟังก์ชัน **smfi_setpriv** ตั้งค่าตัวชี้ข้อมูลส่วนบุคคลสำหรับการเชื่อมต่อนี้

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_setpriv
SMFICTX *ctx,
void *privatedata
();
```

คำอธิบาย

ฟังก์ชัน **smfi_setpriv** ถูกเรียกจากฟังก์ชัน **xxfi_*** **callback** ใดๆ เพื่อตั้งค่า ตัวชี้ข้อมูลส่วนบุคคลสำหรับ *ctx*

หมายเหตุ: มีหนึ่งตัวชี้ข้อมูลส่วนบุคคล ต่อการเชื่อมต่อ การเรียกจำนวนมากไปยังฟังก์ชัน **smfi_setpriv** ที่มีค่าที่แตกต่างกันเป็นสาเหตุทำให้ค่าก่อนหน้านี้หายไป ก่อนที่ตัวกรองจะถูกยกเลิก ตัวกรองต้องรีลีสข้อมูลส่วนบุคคลและตั้งค่า ตัวชี้ให้เป็นค่า NULL

อาร์กิวเมนต์

ตารางที่ 14. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
privatedata	อาร์กิวเมนต์ชี้ไปยังข้อมูลไพรเวต ค่านี้ส่งคืนโดย การเรียกไปยังฟังก์ชัน **smfi_getpriv** ในลำดับถัดมาโดยใช้ *ctx*

ค่าส่งคืน

ฟังก์ชัน **smfi_setpriv** ส่งคืนค่า MI_FAILURE หาก *ctx* คือคอนเท็กซ์ ที่ไม่ถูกต้อง และฟังก์ชันส่งคืน MI_SUCCESS

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน smfi_setpriv" ในหน้า 72

ฟังก์ชัน smfi_setreply:

วัตถุประสงค์

ฟังก์ชัน **smfi_setreply** ตั้งค่าโค้ดระบุความผิดพลาดในการตอบกลับของ simple mail transfer protocol (SMTP) และยอมรับโค้ดการตอบกลับเฉพาะ `4XX` และ `5XX` เท่านั้น

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_setreply
SFICTX *ctx,
char *rcode,
char *xcode,
char *message
);
```

คำอธิบาย

ฟังก์ชัน **smfi_setreply** ถูกเรียกจากฟังก์ชัน **xxfi_ callback** ใดๆ ที่นอกเหนือจากฟังก์ชัน **xxfi_connect** ฟังก์ชัน **smfi_setreply** ตั้งค่าโค้ดระบุความผิดพลาดในการตอบกลับ SMTP สำหรับการเชื่อมต่อ โค้ดนี้ถูกใช้ สำหรับข้อผิดพลาดในการตอบกลับถัดมาซึ่งเป็นผลมาจากการดำเนินการที่ใช้โดย ตัวกรองนี้

ค่าต่างๆ ที่ส่งไปยังฟังก์ชัน **smfi_setreply** ไม่ได้ตรวจสอบความยินยอมมาตรฐาน

อาร์กิวเมนต์ *message* ต้องมีอักขระที่สามารถพิมพ์ได้เท่านั้น อักขระอื่นๆ สามารถนำหน้าด้วย ลักษณะการทำงานที่ไม่ได้กำหนดไว้ ตัวอย่างเช่น อักขระที่คล้ายกับ CR หรือ LF ทำให้การเรียกล้มเหลว อักขระ '%' เดี่ยวทำให้ข้อความถูกละทิ้ง

หมายเหตุ: หากสตริง '%' จำเป็นต้องมีในพารามิเตอร์ให้ใช้ '%%' ที่คล้ายกับ printf(3)

สำหรับรายละเอียด เกี่ยวกับโค้ดการตอบกลับและความหมายของโค้ดเหล่านั้น โปรดดู RFC 821 หรือ 2821 และ RFC 1893 หรือ 2034

หากอาร์กิวเมนต์ *rcode* ถูกตั้งค่าเป็น 4XX แต่ค่า SMFI_REJECT ถูกใช้สำหรับข้อความ การตอบกลับแบบกำหนดเองจะไม่ถูกนำมาใช้

หากอาร์กิวเมนต์ *rcode* ถูกตั้งค่าเป็น 5XX แต่ค่า SMFI_TEMPFAIL ถูกใช้สำหรับข้อความ การตอบกลับแบบกำหนดเองจะไม่ถูกนำมาใช้

หมายเหตุ: ในทั้งสองกรณี ข้างต้น ข้อผิดพลาดถูกส่งคืนกลับไปยังพารามิเตอร์ `milter` พารามิเตอร์ `Libmilter` จะละเว้นโค้ดการตอบกลับ ข้างต้น

หากพารามิเตอร์ `milter` ส่งคืนค่า SMFI_TEMPFAIL และตั้งค่าโค้ดการตอบกลับไปยัง 421 เซิร์ฟเวอร์ SMTP ยกเลิกเซสชัน SMTP ที่มีโค้ดระบุความผิดพลาด 421

อาร์กิวเมนต์

ตารางที่ 15. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
rcode	โค้ดการตอบกลับ (RFC 821 or 2821) SMTP สามหลักที่เป็นสตริง null-terminated *rcode* ไม่สามารถเป็นค่า NULL และต้องเป็นโค้ดการตอบกลับ 4XX หรือ 5XX ที่ถูกต้อง
xcode	โค้ดการตอบกลับ (RFC 1893 หรือ 2034) ที่ขยายเพิ่ม หาก *xcode* คือค่า NULL จะไม่มีโค้ดที่ขยายเพิ่มถูกใช้ หรือ *xcode* must conform to RFC 1893 หรือ 2034
message	ส่วนของข้อความของการตอบกลับ SMTP หากข้อความมีค่า NULL ข้อความที่ว่างเปล่าจะถูกใช้

ค่าส่งคืน

ฟังก์ชัน **smfi_setreply** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *rcode* หรือ *xcode* ไม่ถูกต้อง
- ความล้มเหลวในการจัดสรรหน่วยความจำจะเกิดขึ้น

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน xxfi_connect callback" ในหน้า 91

ฟังก์ชัน smfi_setmlreply:
วัตถุประสงค์

ฟังก์ชัน **smfi_setmlreply** ตั้งค่าโค้ดระบุความผิดพลาดในการตอบกลับของ simple mail transfer protocol (SMTP) ที่เป็นค่าดีฟอลต์ไปเป็นการตอบกลับแบบหลายบรรทัด ฟังก์ชัน **smfi_setmlreply** ยอมรับเฉพาะการตอบกลับ `4XX` และ `5XX`

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_setmlreply(
SMFICTX *ctx,
char *rcode,
char *xcode,
...
);
```

คำอธิบาย

ฟังก์ชัน **smfi_setmlreply** ถูกเรียกจากฟังก์ชัน **xxfi_callback** ใดๆ ยกเว้นฟังก์ชัน **xxfi_connect** ฟังก์ชัน **smfi_setmlreply** จัดเตรียมโค้ดระบุความผิดพลาดในการตอบกลับของ SMTP สำหรับการเชื่อมต่อที่กล่าวถึงอยู่ใต้ *xcode* รายชื่ออาร์กิวเมนต์ ต้องเป็นค่า null-terminated โค้ดนี้ถูกใช้สำหรับข้อผิดพลาดในการตอบกลับถัดมา ซึ่งเป็นผลมาจากการดำเนินการที่ใช้โดยตัวกรองนี้

ค่าต่างๆ ที่ส่งไปยังฟังก์ชัน **smfi_setmlreply** ไม่ได้ตรวจสอบความยินยอมมาตรฐาน

พารามิเตอร์ `message` ต้องมีอักขระที่สามารถพิมพ์ได้ อักขระอื่นๆ ที่นำไปสู่ ลักษณะการทำงานที่นิยามไม่ได้ ตัวอย่างเช่น อักขระที่คล้ายกับ CR หรือ LF ทำให้การเรียกล้มเหลว อักขระ '`%`' เดี่ยวทำให้ข้อความถูกละทิ้ง

หมายเหตุ: หากสตริง '`%`' จำเป็นต้องมีในพารามิเตอร์ `message` ให้ใช้สตริง '`%%`' ที่คล้ายกับสตริง `printf(3)` ที่ถูกใช้

สำหรับโค้ดการตอบกลับและความหมายของโค้ด โปรดดู RFC 821 หรือ 2821 และ RFC 1893 หรือ 2034

หาก *rcode* ถูกตั้งค่าเป็น `4XX` แต่ค่า SMFI_REJECT ถูกใช้สำหรับข้อความ การตอบกลับแบบกำหนดเอง ไม่ได้ถูกใช้

ถ้า *rcode* ถูกตั้งค่าเป็น `5XX` แต่ค่า SMFI_TEMPFAIL ถูกใช้สำหรับข้อความ การตอบกลับแบบกำหนดเอง ไม่ได้ถูกใช้

หมายเหตุ: ในทั้งสองกรณีแรก ข้อผิดพลาดจะถูกส่งคืนกลับไปยังพารามิเตอร์ `milter` และพารามิเตอร์ `Libmilter` ละเว้นข้อผิดพลาดนี้

หากพารามิเตอร์ milter ส่งคืนค่า SMFI_TEMPFAIL และตั้งค่าโค้ดการตอบกลับไปเป็น 421 เซิร์ฟเวอร์ SMTP ยกเลิก SMTP เซสชันด้วยโค้ดระบุความผิดพลาด 421

อาร์กิวเมนต์

ตารางที่ 16. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter
rcode	โค้ดการตอบกลับ (RFC 821 หรือ 2821) SMTP สามหลักที่เป็นสตริง null-terminated อาร์กิวเมนต์ *rcode* ไม่สามารถเป็นค่า NULL ได้และต้องเป็นโค้ดการตอบกลับ 4XX หรือ 5XX ที่ถูกต้อง
xcode	โค้ดการตอบกลับ (RFC 1893 หรือ 2034) ที่ขยายเพิ่ม หาก *xcode* คือค่า NULL จะไม่มีโค้ดที่ขยายเพิ่มถูกใช้ หรือ *xcode* ต้องอยู่ในรูป RFC 1893 หรือ 2034
...	ส่วนที่เหลือของอาร์กิวเมนต์คือบรรทัดข้อความเดี่ยว ที่มีได้สูงสุด 32 อาร์กิวเมนต์ ซึ่งถูกใช้เป็นส่วนของข้อความของการตอบกลับ SMTP รายการ ต้องเป็นค่า null-terminated

ค่าส่งคืน

ฟังก์ชัน **smfi_setmlreply** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *rcode* หรือ *xcode* ไม่ถูกต้อง
- ความล้มเหลวในการจัดสรรหน่วยความจำจะเกิดขึ้น
- บรรทัดข้อความมีการขึ้นบรรทัดใหม่หรือป้อนบรรทัด
- ความยาวของบรรทัดข้อความจะมากกว่า MAXREPLYLEN(980)
- ข้อความที่ตอบกลับจะมากกว่า 32 บรรทัด

ตัวอย่าง

```
ret = smfi_setmlreply(ctx, "550", "5.7.0",
"Spammer access rejected",
"Please see our policy at:",
 "http://www.example.com/spampolicy.html",
NULL);
```

ตัวอย่างก่อนหน้านี้จะส่งผลดังนี้ :

```
550-5.7.0 Spammer access rejected
550-5.7.0 Please see our policy at:
550 5.7.0 http://www.example.com/spampolicy.html
```

ข้อมูลที่เกี่ยวข้อง

“ฟังก์ชัน xxfi_connect callback” ในหน้า 91

ฟังก์ชันการแก้ไขข้อความ

ฟังก์ชันการแก้ไขข้อความเปลี่ยนเนื้อหาและแอ็ตทริบิวต์ ข้อความ ฟังก์ชันถูกเรียกเฉพาะโดยฟังก์ชัน **xxfi_eom** ฟังก์ชันการแก้ไขข้อความสามารถเรียกทำงานการสื่อสารเพิ่มเติม กับ mail transfer agent (MTA) ฟังก์ชันเหล่านี้ส่งคืนค่า MI_SUCCESS หรือ MI_FAILURE เพื่อบ่งชี้สถานะของการดำเนินการ

หมายเหตุ: ข้อความ (ผู้ส่ง ผู้รับ ส่วนหัว และชิ้นเนื้อความ) ที่ถูกส่งผ่านไปยังการแก้ไขข้อความในพารามิเตอร์ซึ่งถูกคัดลอกและไม่ต้องการสงวนไว้ (หน่วยความจำที่จัดสรรแล้วสามารถป้อนได้)

หากต้องการเรียกฟังก์ชันการแก้ไขข้อความ ตัวกรองต้องตั้งค่า แฟล็กให้เหมาะสมในคำอธิบายที่ถูกส่งผ่านฟังก์ชัน **smfi_register** หากแฟล็กไม่ได้ถูกตั้งค่าไว้ MTA จัดการกับการเรียกฟังก์ชันเนื่องจาก ความล้มเหลวของตัวกรอง และยกเลิกการเชื่อมต่อ

หมายเหตุ: สถานะ ถูกส่งคืนโดยฟังก์ชันที่บ่งชี้ว่าตัวกรองข้อความ จะถูกส่งไปยัง MTA ได้เป็นผลสำเร็จ สถานะไม่ได้บ่งชี้ว่า MTA ดำเนินการกับการดำเนินการที่ร้องขอ ตัวอย่างเช่น ฟังก์ชัน **smfi_header** เมื่อเรียกด้วยชื่อส่วนหัวที่ผิดกฎเกณฑ์ ฟังก์ชันจะส่งคืนแฟล็ก MI_SUCCESS แม้ว่า MTA สามารถปฏิเสธการเพิ่มส่วนหัวที่ผิดกฎเกณฑ์ในภายหลัง

ตารางที่ 17. ฟังก์ชัน Mod

ไอเท็ม	คำอธิบาย	ฟังก์ชัน
smfi_addheader	ฟังก์ชัน **smfi_addheader** เพิ่มส่วนหัวให้กับ ข้อความ	**SMFIF_ADDHDRS**
smfi_chgheader	ฟังก์ชัน **smfi_chgheader** แก้ไขหรือลบ ส่วนหัว	**SMFIF_CHGHDRS**
smfi_insheader	ฟังก์ชัน **smfi_insheader** แทรกส่วนหัวลงใน ข้อความ	**SMFIF_ADDHDRS**
smfi_chgfrom	ฟังก์ชัน **smfi_chgfrom** แก้ไขแอดเดรส ผู้ส่งจดหมาย	**SMFIF_CHGFROM**
smfi_addrcpt	ฟังก์ชัน **smfi_addrcpt** เพิ่มผู้รับให้กับ ซองจดหมาย	**SMFIF_ADDRCPT**
smfi_addrcpt_par	ฟังก์ชัน **smfi_addrcpt_par** เพิ่มผู้รับที่สอดแทรกพารามิเตอร์ simple mail transfer protocol (ESMTP) ไปยังซองจดหมาย	**SMFIF_ADDRCPT_PAR**
smfi_delrcpt	ฟังก์ชัน **smfi_delrcpt** ลบผู้รับออกจาก ซองจดหมาย	**SMFIF_DELRCPT**
smfi_replacebody	ฟังก์ชัน **smfi_replacebody** แทนที่เนื้อความ ของข้อความ	**SMFIF_CHGBODY**

ฟังก์ชัน smfi_addheader:
วัตถุประสงค์

ฟังก์ชัน **smfi_addheader** เพิ่มส่วนหัวให้กับข้อความปัจจุบัน

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_addheader(
SMFICTX *ctx,
char *headerf,
char *headerv
);
```

คำอธิบาย

ฟังก์ชัน **smfi_addheader** ถูกเรียกจากฟังก์ชัน **xxfi_eom** เพื่อเพิ่มส่วนหัวไปยัง ข้อความ

ฟังก์ชัน **smfi_addheader** ไม่ได้แก้ไข ส่วนหัวที่มีอยู่ของข้อความ

หากแก้ไขค่าส่วนหัวปัจจุบัน ให้ใช้ฟังก์ชัน **smfi_chgheader**

ตัวกรอง ที่เรียกฟังก์ชัน **smfi_addheader** ต้องการตั้งค่าแฟล็ก **SMFIF_ADDHDRS** ในอาร์กิวเมนต์ `smfiDesc_str` ตัวกรองส่งผ่านค่า ไปยังฟังก์ชัน **smfi_register**

ฟังก์ชัน **smfi_addheader** จำเป็นต้องมีลำดับการกรองที่ต้องถูกระบุเฉพาะ คุณสามารถดูการแก้ไขในส่วนหัว โดยใช้ตัวกรองที่ถูกสร้างก่อนหน้านี้

ชื่อ หรือค่าของส่วนหัวไม่ได้ถูกตรวจสอบสำหรับการยินยอมมาตรฐาน อย่างไรก็ตาม แต่ละบรรทัดต้องเป็นอักขระต่ำกว่า `998` ตัวอักษร หาก คุณต้องการชื่อส่วนหัวที่ยาวกว่า ให้ใช้ส่วนหัวแบบหลายบรรทัด หากคุณต้องการสร้าง ส่วนหัวแบบหลายบรรทัด ให้แทรกการป้อนบรรทัด (`ASCII 0x0a` หรือ `\n` ในภาษาโปรแกรม C) ตามด้วยอักขระช่องว่าง เช่น พื้นที่ว่าง (`ASCII 0x20`) หรือแท็บ (`ASCII 0x09` หรือ `\t` ในภาษาโปรแกรม C) การป้อนบรรทัดไม่สามารถนำหน้าด้วยการขึ้นบรรทัดใหม่ (`ASCII 0x0d`) mail transfer agent (MTA) จะเพิ่มให้โดยอัตโนมัติ ความรับผิดชอบของตัวเขียนตัวกรองต้องมั่นใจว่าไม่มีมาตรฐาน ที่ถูกละเมิด

MTA เพิ่มการนำหน้าด้วยช่องว่างไปยังค่าส่วนหัวที่เพิ่มขึ้น ยกเว้นว่าตั้งค่าแฟล็ก **SMFIP_HDR_LEADSPC** ในกรณีที่พารามิเตอร์ `milter` ต้องสอดแทรกการนำหน้าด้วยช่องว่างใดๆ

อาร์กิวเมนต์

ตารางที่ 18. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
headerf	ชื่อส่วนหัวเป็นสตริง null-terminated ที่ไม่ใช่ค่า NULL
headerv	ค่าส่วนหัวที่ต้องถูกเพิ่มสามารถเป็นสตริง non-NULL null-terminated หรือสตริงว่าง

ค่าส่งคืน

ฟังก์ชัน **smfi_addheader** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *headerf* หรือ *headerv* เป็นค่า NULL

- การเพิ่มส่วนหัวในสถานะการเชื่อมต่อปัจจุบันไม่ถูกต้อง
- การจัดสรรหน่วยความจำล้มเหลว
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก **SMFIF_ADDHDRS** ไม่ตั้งค่าเมื่อฟังก์ชัน **smfi_register** ถูกเรียก

ตัวอย่าง

```
int     ret;
SMFICTX *ctx;
...
ret = smfi_addheader(ctx, "Content-Type",
"multipart/mixed;\n\tboundary=\"foobar\"");
```

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน xxfi_eom callback" ในหน้า 99

"ฟังก์ชัน smfi_chgheader"

"ฟังก์ชัน smfi_register" ในหน้า 62

ฟังก์ชัน smfi_chgheader:

วัตถุประสงค์

ฟังก์ชัน **smfi_chgheader** แก้ไขหรือลบส่วนหัวข้อความ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_chgheader(
SMFICTX *ctx,
char *headerf,
mi_int32 hdridx,
char *headerv
);
```

คำอธิบาย

ฟังก์ชัน **smfi_chgheader** ถูกเรียกจากฟังก์ชัน **xxfi_eom** เพื่อแก้ไขค่าส่วนหัว สำหรับข้อความปัจจุบัน

ฟังก์ชัน **smfi_chgheader** สามารถใช้เพื่อเพิ่มส่วนหัวใหม่ อย่างไรก็ตาม ส่วนหัวมีประสิทธิภาพ และปลอดภัยต่อการใช้ฟังก์ชัน **smfi_addheader**

ตัวกรองที่เรียกฟังก์ชัน **smfi_chgheader** ต้องตั้งค่าแฟล็ก **SMFIF_CHGHDRS** ในอาร์กิวเมนต์ `smfiDesc_str` ตัวกรองส่งผ่านค่า ไปยังฟังก์ชัน **smfi_register**

ฟังก์ชัน **smfi_chgheader** จำเป็นต้องมีลำดับการกรองที่ต้องระบุไว้ คุณสามารถดูการแก้ไขในส่วนหัว โดยใช้ตัวกรองที่ถูกสร้างก่อนหน้านี้

ชื่อ หรือค่าของส่วนหัวไม่ได้ถูกตรวจสอบสำหรับการยินยอมมาตรฐาน อย่างไรก็ตาม แต่ละบรรทัดต้องเป็นอักขระต่ำกว่า 998 ตัวอักษร หาก คุณต้องกรชื่อส่วนหัวที่ยากกว่า ให้ใช้ส่วนหัวแบบหลายบรรทัด หากคุณต้องการสร้าง ส่วนหัวแบบหลายบรรทัด ให้แทรกการป้อนบรรทัด (ASCII 0x0a หรือ \n ในภาษาโปรแกรม C) ตามด้วยอักขระช่องว่าง เช่น พื้นที่ว่าง (ASCII 0x20) หรือแท็บ (ASCII 0x09 หรือ \t ในภาษาโปรแกรม C) การป้อนบรรทัดใหม่ไม่สามารถนำหน้าด้วยการขึ้นบรรทัดใหม่ (ASCII 0x0d), mail transfer agent (MTA) เพิ่มไว้โดยอัตโนมัติ ความรับผิดชอบ ของตัวเขียนตัวกรองต้องมั่นใจว่าไม่มีมาตรฐานที่ถูกละเมิด

MTA เพิ่มการนำหน้าด้วยช่องว่างไปยังค่าส่วนหัวที่เพิ่มขึ้น ยกเว้นว่าตั้งค่าแฟล็ก **SMFIP_HDR_LEADSPC** ในกรณีที่พารามิเตอร์ `milter` ต้องสอดแทรกการนำหน้าด้วยช่องว่างใดๆ

อาร์กิวเมนต์

ตารางที่ 19. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
headerf	ชื่อส่วนหัวเป็นสตริง null-terminated ที่ไม่ใช่ค่า NULL
hdridx	ค่าดัชนีส่วนหัว (หนึ่งฐาน) ค่า *hdridx* ของ 1 แก้ไขเหตุการณ์ที่เกิดขึ้นในครั้งแรกของส่วนหัวที่ชื่อ *headerf* หาก *hdridx* ที่มีค่ามากกว่าจำนวนครั้ง *headerf* ปรากฏขึ้น สำเนาใหม่ของ *headerf* จะถูกเพิ่มขึ้น
headerv	ค่าส่วนหัวที่ต้องถูกเพิ่มสามารถเป็นสตริง non-NULL null-terminated หรือสตริงว่าง

ค่าส่งคืน

ฟังก์ชัน **smfi_chgheader** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *headerf* เป็นค่า NULL
- การแก้ไขส่วนหัวในสถานะการเชื่อมต่อปัจจุบันไม่ถูกต้อง
- การจัดสรรหน่วยความจำล้มเหลว
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก **SMFIF_CHGHDRS** ไม่ได้ตั้งค่าเมื่อฟังก์ชัน **smfi_register** ถูกเรียก

ตัวอย่าง

```
int     ret;
SMFICTX *ctx;
...

ret = smfi_chgheader(ctx, "Content-Type", 1,
"multipart/mixed;\n\tboundary=\"foobar\"");
```

ข้อมูลที่เกี่ยวข้อง

“ฟังก์ชัน xxfi_eom callback” ในหน้า 99

“ฟังก์ชัน smfi_addheader” ในหน้า 77

ฟังก์ชัน smfi_insheader:
วัตถุประสงค์

ฟังก์ชัน smfi_insheader เพิ่มส่วนหัวไปยังข้อความปัจจุบัน

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_insheader(
SMFICTX ,
int hdridx,
char *headerf,
char *headerv
);
```

คำอธิบาย

ฟังก์ชัน smfi_insheader ถูกเรียกจากฟังก์ชัน **xxfi_eom** เพื่อเพิ่มส่วนหัวไปยัง ข้อความปัจจุบัน

ฟังก์ชัน smfi_insheader ไม่ได้แก้ไขส่วนหัวที่มีอยู่ของข้อความ

หากเปลี่ยนค่าปัจจุบัน ของส่วนหัว ให้ใช้ฟังก์ชัน **smfi_chgheader**

ตัวกรอง ที่เรียกฟังก์ชัน **smfi_insheader** ต้องตั้งค่าแฟล็ก **SMFIF_ADDHDRS** ในอาร์กิวเมนต์ `smfiDesc_str` ที่ส่งผ่านใน ฟังก์ชัน **smfi_register**

ฟังก์ชัน smfi_insheader จำเป็นต้องมีลำดับการกรองที่ต้องระบุไว้ คุณสามารถดูการแก้ไขในส่วนหัว โดยใช้ตัวกรองที่ถูกสร้างก่อนหน้านี้

ตัวกรอง ได้รับส่วนหัวที่ถูกส่งโดยไคลเอ็นต์ simple mail transfer protocol (SMTP) และส่วนหัวที่แก้ไขโดย ตัวกรองก่อนหน้า ส่วนหัวที่แทรกโดยคำสั่ง **sendmail** และส่วนหัว ที่แทรกด้วยส่วนหัวเองจะไม่ถูกรับ ตำแหน่งในการแทรกส่วนหัว ขึ้นอยู่กับส่วนหัวที่มีอยู่ในข้อความขาเข้า และยังอยู่บนส่วนหัวที่ถูกกำหนดคอนฟิกที่ต้องเพิ่มโดยคำสั่ง **sendmail**

ตัวอย่างเช่น คำสั่ง **sendmail** จะเพิ่ม **ส่วนหัว รับแล้ว:** ที่จุดเริ่มต้นของส่วนหัว สำหรับการตั้งค่า *hdridx* ไปเป็น `0` ส่วนหัวจะถูกแทรกไว้ก่อนพารามิเตอร์ **ส่วนหัว รับแล้ว:** อย่างไรก็ตาม ตัวกรองถัดมาได้รับสิ่งที่ผิดตามที่ได้รับส่วนหัวที่เพิ่มเติม แต่ไม่ใช่ **ส่วนหัว รับแล้ว:** ดังนั้น ซึ่งทำให้ยากต่อการแทรกส่วนหัว ที่ตำแหน่งที่แก้ไข

หากค่า *hdridx* มากกว่าจำนวนของส่วนหัวในข้อความ ส่วนหัวจะถูกต่อท้าย

ชื่อ หรือค่าของส่วนหัวไม่ได้ถูกตรวจสอบสำหรับการยินยอมมาตรฐาน อย่างไรก็ตาม แต่ละบรรทัดต้องเป็นอักขระต่ำกว่า 998 ตัวอักษร หาก คุณต้องการชื่อส่วนหัวที่ยาวกว่า ให้ใช้ส่วนหัวแบบหลายบรรทัด หากคุณต้องการสร้าง ส่วนหัวแบบหลายบรรทัด ให้แทรกการป้อนบรรทัด (`ASCII 0x0a` หรือ `\n` ในภาษาโปรแกรม C) ตามด้วยอักขระช่องว่าง เช่น พื้นที่ว่าง (`ASCII 0x20`) หรือแท็บ (`ASCII 0x09` หรือ `\t` ในภาษาโปรแกรม C) การป้อนบรรทัดไม่สามารถนำหน้าด้วยการขึ้นบรรทัดใหม่ (`ASCII 0x0d`) mail transfer agent (MTA) เพิ่มการขึ้นบรรทัดใหม่โดยอัตโนมัติ ความรับผิดชอบของตัวเขียนตัวกรองต้องมั่นใจว่าไม่มีมาตรฐาน ที่ถูกละเมิด

MTA เพิ่มการนำหน้าด้วยช่องว่างไปยังค่าส่วนหัวที่แทรกไว้ ยกเว้นว่าตั้งค่าแฟล็ก **SMFIP_HDR_LEADSPC** ในกรณีที่พารามิเตอร์ `milter` ต้องสอดแทรกการนำหน้าด้วยช่องว่างใดๆ

อาร์กิวเมนต์

ตารางที่ 20. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
headerf	ชื่อส่วนหัวเป็นสตริง null-terminated ที่ไม่ใช่ค่า NULL
headerv	ค่าส่วนหัวที่ต้องถูกเพิ่มสามารถเป็นสตริง non-NULL null-terminated หรือสตริงว่าง

ค่าส่งคืน

ฟังก์ชัน **smfi_insheader** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ นอกเหนือจากฟังก์ชันที่ส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *headerf* หรือ *headerv* เป็นค่า NULL
- การเพิ่มส่วนหัวในสถานะการเชื่อมต่อปัจจุบันไม่ถูกต้อง
- การจัดสรรหน่วยความจำล้มเหลว
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก **SMFIF_ADDHDRS** ไม่ถูกตั้งค่าเมื่อฟังก์ชัน **smfi_register** ถูกเรียก

ตัวอย่าง

```
int     ret;
SMFICTX *ctx;
...
ret = smfi_insheader( ctx, 0, "First", "See me?");;
```

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน xxfi_eom callback" ในหน้า 99

"ฟังก์ชัน smfi_register" ในหน้า 62

"ฟังก์ชัน smfi_chgheader" ในหน้า 79

ฟังก์ชัน smfi_chgfrom: วัตถุประสงค์

ฟังก์ชัน **smfi_chgfrom** แก้ไขผู้ส่งจดหมาย (MAIL จาก) สำหรับข้อความปัจจุบัน

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_chgfrom(
SMFICTX *ctx,
const char *mail,
char *args
);
```

คำอธิบาย

ฟังก์ชัน **smfi_chgfrom** ถูกเรียกจากฟังก์ชัน **xxfi_eom** เพื่อแก้ไขผู้ส่งจดหมาย และ MAIL จากข้อความปัจจุบัน

ตัวกรองที่เรียกฟังก์ชัน **smfi_chgfrom** ต้องตั้งค่าแฟล็ก **SMFIF_CHGFROM** ในอาร์กิวเมนต์ `smfiDesc_str` ตัวกรองส่งผ่านค่า ไปยังฟังก์ชัน **smfi_register**

อาร์กิวเมนต์ simple mail transfer protocol (ESMTP) ที่ขยายเพิ่มทั้งหมดสามารถตั้งค่าผ่าน การเรียกได้ แต่การตั้งค่าสำหรับอาร์กิวเมนต์บางตัว เช่น `SIZE` และ `BODY` อาจทำให้เกิดปัญหา ดังนั้น จึงต้องใช้ด้วยความระมัดระวัง เมื่อตั้งค่าอาร์กิวเมนต์ไม่มีผลป้อนกลับจาก mail transfer agent (MTA) ไปยังพารามิเตอร์ `milter` หากการเรียกเป็นผลสำเร็จ

อาร์กิวเมนต์

ตารางที่ 21. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
mail	แอดเดรสของผู้ส่งใหม่
args	อาร์กิวเมนต์ simple mail transfer protocol (ESMTP) ที่ขยายเพิ่ม

ค่าส่งคืน

ฟังก์ชัน **smfi_chgfrom** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *mail* เป็นค่า NULL
- การเปลี่ยนผู้ส่งในสถานะการเชื่อมต่อปัจจุบันไม่ถูกต้อง
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก **SMFIF_CHGFROM** ไม่ได้ถูกตั้งค่าเมื่อฟังก์ชัน **smfi_register** ถูกเรียก

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน xxfi_eom callback" ในหน้า 99

"ฟังก์ชัน smfi_register" ในหน้า 62

ฟังก์ชัน smfi_addrcpt:

วัตถุประสงค์

ฟังก์ชัน **smfi_addrcpt** เพิ่มผู้รับสำหรับข้อความปัจจุบัน

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_addrcpt(
SMFICTX *ctx
char *rcpt
);
```

คำอธิบาย

ฟังก์ชัน **smfi_addrcpt** ถูกเรียกจากฟังก์ชัน **xxfi_eom** เท่านั้น เพื่อเพิ่มผู้รับ ให้กับจดหมายข้อความ

หมายเหตุ: ตัวกรองที่เรียกฟังก์ชัน **smfi_addrcpt** ต้องตั้งค่าแฟล็ก **SMFIF_ADDRCPT** ในโครงสร้าง **smfiDesc_str** ที่ส่งผ่านไปยังฟังก์ชัน **smfi_register**

อาร์กิวเมนต์

ตารางที่ 22. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter
rcpt	แอดเดรสของผู้รับใหม่

ค่าส่งคืน

ฟังก์ชัน **smfi_addrcpt** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *rcpt* เป็นค่า NULL
- การเพิ่มผู้รับในสถานะการเชื่อมต่อปัจจุบันไม่ถูกต้อง
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก **SMFIF_ADDRCPT** จะไม่ถูกตั้งค่าเมื่อฟังก์ชันยกเลิกฟังก์ชัน **smfi_register**

ข้อมูลที่เกี่ยวข้อง

“ฟังก์ชัน xxfi_eom callback” ในหน้า 99

“ฟังก์ชัน smfi_register” ในหน้า 62

ฟังก์ชัน smfi_addrcpt_par:

วัตถุประสงค์

ฟังก์ชัน **smfi_addrcpt_par** เพิ่มผู้รับสำหรับข้อความปัจจุบันซึ่งประกอบด้วยอาร์กิวเมนต์ simple mail transfer protocol (ESMTP) ที่ขยายเพิ่ม

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_addrcpt_par(
SMFICTX *ctx,
char *rcpt,
char *args
);
```

คำอธิบาย

ฟังก์ชัน **smfi_addrcpt_par** ถูกเรียกจากฟังก์ชัน **xxfi_eom** เพื่อเพิ่มผู้รับไปยัง จดหมายข้อความ

อาร์กิวเมนต์

ตารางที่ 23. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
rcpt	แอดเดรสของผู้รับใหม่
args	พารามิเตอร์ผู้รับ ESMTP

ค่าส่งคืน

ฟังก์ชัน **smfi_addrcpt** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *rcpt* เป็นค่า NULL
- การเพิ่มผู้รับในสถานะการเชื่อมต่อปัจจุบันไม่ถูกต้อง
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก **SMFIF_ADDRCPT_PAR** ไม่ได้ถูกตั้งค่าเมื่อฟังก์ชัน **smfi_register** ถูกเรียก

ข้อมูลที่เกี่ยวข้อง

"ฟังก์ชัน smfi_addrcpt" ในหน้า 83

"ฟังก์ชัน smfi_register" ในหน้า 62

ฟังก์ชัน smfi_delrcpt:
วัตถุประสงค์

ฟังก์ชัน **smfi_delrcpt** ลบผู้รับออกจากจดหมายของข้อความปัจจุบัน

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_delrcpt(
 SMFICTX *ctx;
 char *rcpt;
);
```

คำอธิบาย

ฟังก์ชัน **smfi_delrcpt** ถูกเรียกจากฟังก์ชัน **xxfi_eom** callback เพื่อลบผู้รับที่ตั้งชื่อไว้ ออกจากจดหมายข้อความปัจจุบัน

หมายเหตุ: แอดเดรสต่างๆ ที่ต้องถูกลบทิ้งต้องตรงกันอย่างชัดเจน ตัวอย่างเช่น แอดเดรสและรูปแบบที่ขยายเพิ่ม ไม่ตรงกัน

อาร์กิวเมนต์

ตารางที่ 24. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
rcpt	แอดเดรสผู้รับที่ต้องถูกลบออก นั่นคือสตริง non-NULL, null-terminated

ค่าส่งคืน

ฟังก์ชัน **smfi_delrcpt** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- อาร์กิวเมนต์ *rcpt* เป็นค่า NULL
- การลบผู้รับในสถานะการเชื่อมต่อปัจจุบันไม่ถูกต้อง
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก SMFIF_DELRCPT ไม่ได้ตั้งค่าเมื่อฟังก์ชัน **smfi_register** ถูกเรียก

ข้อมูลที่เกี่ยวข้อง

smfi_register

xxfi_eom

ฟังก์ชัน smfi_replacebody:
วัตถุประสงค์

ฟังก์ชัน **smfi_replacebody** แทนที่เนื้อความของข้อความ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_replacebody(
 SMFICTX *ctx,
 unsigned char *bodyp,
 int bodylen
);
```

คำอธิบาย

ฟังก์ชัน **smfi_replacebody** แทนที่เนื้อความของข้อความปัจจุบัน หากฟังก์ชันถูกเรียกมากกว่าหนึ่งครั้ง การเรียกลำดับถัดมาจะส่งผลให้ข้อมูลต่อท้าย เนื้อความใหม่ ฟังก์ชันอาจถูกเรียกได้มากกว่าหนึ่งครั้ง

เนื่องจาก เนื้อข้อความอาจใหญ่ขึ้น การตั้งค่าแฟล็ก SMFIF_CHGBODY อาจมีผลต่อผลการทำงานของตัวกรอง

หากตัวกรอง ตั้งค่าแฟล็ก SMFIF_CHGBODY แต่ไม่เรียกฟังก์ชัน **smfi_replacebody** เนื้อความเดิมยังคงไม่เปลี่ยนแปลง

ลำดับตัวกรองเป็นสิ่งสำคัญ สำหรับฟังก์ชัน **smfi_replacebody** เนื้อความใหม่ถูกสร้างขึ้นโดย ตัวกรองเก่าในไฟล์ตัวกรองใหม่

อาร์กิวเมนต์

ตารางที่ 25. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
bodyp	ตัวชี้ไปยังจุดเริ่มต้นข้อมูลเนื้อความใหม่ ซึ่งไม่ได้เป็น null-terminated หาก *bodyp* เป็นค่า NULL ตัวชี้นั้นจะมีความยาว == 0 ข้อมูลเนื้อความควรอยู่ในรูปแบบ CR หรือ LF
bodylen	จำนวนของไบต์ข้อมูลถูกชี้ไปโดย *bodyp*

ค่าส่งคืน

ฟังก์ชัน **smfi_replacebody** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- *bodyp* == NULL และ *bodylen* > 0
- การเปลี่ยนเนื้อความในสถานะการเชื่อมต่อปัจจุบันไม่ถูกต้อง
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก `SMFIF_CHGBODY` ไม่ถูกตั้งค่าเมื่อฟังก์ชัน **smfi_register** ถูกยกเลิก

ข้อมูลที่เกี่ยวข้อง

smfi_register

ฟังก์ชันการจัดการกับข้อความ

ฟังก์ชันการจัดการกับข้อความจัดเตรียมคำสั่งในการจัดการกับกรณีพิเศษ สำหรับพารามิเตอร์ `milter` หรือ mail transfer agent (MTA) โดยไม่เปลี่ยนเนื้อหาหรือสถานะ ของข้อความ ฟังก์ชันการจัดการกับข้อความสามารถเรียกได้เฉพาะใน **xxfi_eom** ฟังก์ชัน **xxfi_eom** สามารถ เรียกใช้งานการสื่อสารเพิ่มเติมด้วย MTA และส่งคืนค่า MI_SUCCESS หรือ MI_FAILURE เพื่อบ่งชี้สถานะของการดำเนินการ

หมายเหตุ: สถานะ ที่ส่งคืนโดยฟังก์ชันบ่งชี้ว่าข้อความตัวกรอง ได้ส่งไปยัง MTA เป็นผลสำเร็จแล้ว สถานะไม่ได้บ่งชี้ว่า MTA ดำเนินการกับการดำเนินการที่ร้องขอ

ตารางที่ 26. ข้อความ การจัดการข้อความ

ไอเท็ม	คำอธิบาย
smfi_progress	ฟังก์ชัน **smfi_progress** รายงานการดำเนินการว่ากำลังดำเนินการอยู่
smfi_quarantine	ฟังก์ชัน **smfi_quarantine** กักกันข้อความ

ฟังก์ชัน smfi_progress:
วัตถุประสงค์

ฟังก์ชัน **smfi_progress** รายงานความคืบหน้าของการดำเนินการ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_progress(
 SMFICTX *ctx;
);
```

คำอธิบาย

ฟังก์ชัน smfi_progress ถูกเรียกจากฟังก์ชัน **xxfi_eom** callback เพื่อแจ้งให้ mail transfer agent (MTA) ทราบว่าตัวกรองยังคงทำงานกับข้อความ ฟังก์ชันนี้เป็นสาเหตุทำให้ MTA รีสตาร์ทการหมดเวลา

อาร์กิวเมนต์

ตารางที่ 27. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter

ค่าส่งคืน

ฟังก์ชัน smfi_progress ส่งคืนค่า MI_FAILURE หากมีข้อผิดพลาดด้านเครือข่าย และฟังก์ชันส่งคืน MI_SUCCESS

ข้อมูลที่เกี่ยวข้อง

xxfi_eom

ฟังก์ชัน smfi_quarantine:
วัตถุประสงค์

ฟังก์ชัน smfi_quarantine กักกันข้อความ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_quarantine(
 SMFICTX *ctx;
 char *reason;
);
```

คำอธิบาย

ฟังก์ชัน smfi_quarantine ถูกเรียกจากฟังก์ชัน **xxfi_eom** callback เพื่อกักกันข้อความโดยใช้เหตุผลที่กำหนดไว้

อาร์กิวเมนต์

ตารางที่ 28. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
reason	เหตุผลของการกักกันซึ่งเป็นสตริง non-NULL และ non-empty null-terminated

ค่าส่งคืน

ฟังก์ชัน **smfi_quarantine** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- *reason* คือค่า NULL หรือค่าว่าง
- ข้อผิดพลาดเน็ตเวิร์กจะเกิดขึ้น
- แฟล็ก `SMFIF_QUARANTINE` จะไม่ถูกตั้งค่าเมื่อฟังก์ชัน **smfi_register** ถูกเรียก

ข้อมูลที่เกี่ยวข้อง

smfi_register

xxfi_eom

ฟังก์ชัน Callback

ตัวกรอง sendmail ต้องนำฟังก์ชัน callback หนึ่งฟังก์ชันหรือมากกว่าไปใช้งาน ซึ่งได้ลงทะเบียนผ่านฟังก์ชัน **smfi_register**

ตารางที่ 29. ฟังก์ชัน Call back

ไอเท็ม	คำอธิบาย
xxfi_connect	ฟังก์ชัน **xxfi_connect** ถูกเรียกเพียงครั้งเดียวที่จุดเริ่มต้นของ การเชื่อมต่อ SMTP แต่ละครั้ง ฟังก์ชันส่งคืนค่า SMFIS_CONTINUE
xxfi_hello	ฟังก์ชัน **xxfi_hello** ถูกเรียกเมื่อใดก็ตามที่ไคลเอ็นต์ ส่งคำสั่ง HELO/EHLO
xxfi_envfrom	ฟังก์ชัน **xxfi_envfrom** ถูกเรียกที่จุดเริ่มต้นของ ข้อความ
xxfi_envrcpt	ฟังก์ชัน **xxfi_envrcpt** ถูกเรียกสำหรับผู้รับแต่ละราย
xxfi_data	ฟังก์ชัน **xxfi_data** จัดการกับคำสั่ง DATA
xxfi_unknown	ฟังก์ชัน **xxfi_unknown** จัดการกับคำสั่ง simple mail transfer protocol (SMTP) ที่ไม่รู้จัก
xxfi_header	ฟังก์ชัน **xxfi_header** จัดการกับส่วนหัวของข้อความ
xxfi_eoh	ฟังก์ชัน **xxfi_eoh** จัดการกับส่วนหัวของข้อความ
xxfi_body	ฟังก์ชัน **xxfi_body** จัดการกับชิ้นข้อมูลของ เนื้อความ
xxfi_eom	ฟังก์ชัน **xxfi_eom** จัดการกับส่วนท้ายของข้อความ
xxfi_abort	ฟังก์ชัน **xxfi_abort** จัดการกับข้อความที่ ถูกยกเลิก
xxfi_close	ฟังก์ชัน **xxfi_close** ถูกเรียกเพื่อปิด การเชื่อมต่อปัจจุบัน
xxfi_negotiate	ฟังก์ชัน **xxfi_negotiate** ถูกเรียกที่จุดเริ่มต้นของ การเชื่อมต่อ SMTP

ฟังก์ชัน callback ต้องส่งคืนค่าที่ถูกต้อง หากฟังก์ชัน callback ส่งคืนค่าอื่นใดที่ไม่ใช่ค่าที่นิยามไว้ ฟังก์ชันจะแสดงข้อผิดพลาด และคำสั่ง **sendmail** จะยกเลิกการเชื่อมต่อไปยังตัวกรอง

พารามิเตอร์ `Milter` แยกระหว่างรูทีน **recipient-, message-** และ **connection-oriented** :

- ฟังก์ชัน **recipient-oriented** มีผลต่อการประมวลผล ผู้รับข้อความเดี่ยว
- ฟังก์ชัน **message-oriented** มีผลต่อ ข้อความเดี่ยว
- ฟังก์ชัน **connection-oriented** callback มีผลต่อการเชื่อมต่อทั้งหมด (ในระหว่างข้อความจำนวนมากที่สามารถส่งผ่านไปยัง ชุดของผู้รับจำนวนมาก)
- ฟังก์ชัน **xxfi_envrcpt** เป็นแบบ recipient-oriented ฟังก์ชัน **xxfi_conect, xxfi_hello** และ **xxfi_close** เป็นแบบ connection-oriented ฟังก์ชัน callback อื่นๆ ทั้งหมดเป็นแบบ message-oriented

ตารางที่ 30. ฟังก์ชัน Callback

ไอเท็ม	คำอธิบาย
SMFIS_CONTINUE	ดำเนินการประมวลผลการเชื่อมต่อ ข้อความ หรือผู้รับ ปัจจุบัน
SMFIS_REJECT	• สำหรับรูทีน **connection-oriented** ให้ปฏิเสธการเชื่อมต่อนี้ แล้วเรียก **xxfi_close** • สำหรับรูทีน **message-oriented** (ยกเว้นฟังก์ชัน **xxfi_eom** หรือฟังก์ชัน **xxfi_abort**) ให้ปฏิเสธข้อความนี้ • สำหรับรูทีน **recipient-oriented** ให้ปฏิเสธผู้รับปัจจุบัน (แต่ดำเนินการประมวลผลข้อความปัจจุบันต่อ)
SMFIS_DISCARD	• สำหรับรูทีน **message-** หรือ **recipient-oriented** ให้ยอมรับข้อความนี้ แต่ละเลยข้อความนี้ • SMFIS_DISCARD ต้องไม่ส่งคืนโดยรูทีน **connection-oriented**
SMFIS_ACCEPT	• สำหรับรูทีน **connection-oriented** ให้ยอมรับการเชื่อมต่อนี้ โดยไม่ต้องประมลผลตัวกรองเพิ่มเติม แล้วเรียกฟังก์ชัน **xxfi_close** • สำหรับรูทีน **message- หรือ recipient- oriented** ให้ยอมรับข้อความนี้โดยไม่มีการกรองเพิ่มเติม
SMFIS_TEMPFAIL	ส่งคืนความล้มเหลวชั่วคราว นั่นคือคำสั่ง simple mail transfer protocol (SMTP) ที่สอดคล้องกันจะส่งคืนโค้ดสถานะ 4xx • สำหรับรูทีน **message-oriented** (ยกเว้นฟังก์ชัน **xxfi_envfrom**) เกิดความล้มเหลวสำหรับข้อความนี้ • สำหรับรูทีน **connection-oriented** เกิดความล้มเหลวสำหรับการเชื่อมต่อนี้ เรียกฟังก์ชัน **xxfi_close** • สำหรับรูทีน **recipient-oriented** เกิดความล้มเหลวสำหรับผู้รับปัจจุบัน ดำเนินการประมวลผลข้อความต่อ

ตารางที่ 30. ฟังก์ชัน Callback (ต่อ)

ไอเท็ม	คำอธิบาย
SMFIS_SKIP	ข้าม callbacks ที่มีชนิดเดียวกันเพิ่มเติมในธุรกรรมนี้ ณ ปัจจุบัน ค่าส่งคืนนี้ได้รับอนุญาตให้ใช้ในฟังก์ชัน **xxfi_body** เท่านั้น ค่าส่งคืนสามารถใช้ได้หากพารามิเตอร์ `milter` ได้รับชิ้นข้อมูลเนื้อความที่เพียงพอต่อการตัดสินใจ แต่หากค่าส่งคืน ยังคงต้องการเรียกใช้ฟังก์ชันการแก้ไขข้อความ ที่ได้รับอนุญาตให้ใช้โดยเรียกจากฟังก์ชัน **xxfi_eom** เท่านั้น **หมายเหตุ:** พารามิเตอร์ `milter` ต้องเจรจาถึงลักษณะการทำงานนี้กับ mail transfer agent (MTA) พารามิเตอร์ `milter` ตรวจสอบการดำเนินการกับโปรโตคอล SMFIP_SKIP ที่พร้อมใช้งาน หากการดำเนินการของโปรโตคอล SMFIP_SKIP พร้อมใช้งาน พารามิเตอร์ `milter` ต้องร้องขอการดำเนินการนี้
SMFIS_NOREPLY	• ห้ามส่งการตอบกลับไปยัง MTA พารามิเตอร์ `milter` ต้องเจรจาถึงลักษณะการทำงานนี้กับ MTA พารามิเตอร์ `milter` ต้องตรวจสอบถึงการดำเนินการกับโปรโตคอล SMFIP_NR_* ที่เหมาะสมซึ่งพร้อมใช้งาน หากการดำเนินการกับโปรโตคอล SMFIP_NR_* พร้อมใช้งาน พารามิเตอร์ `milter` ต้องร้องขอการดำเนินการนี้ • หากคุณตั้งค่าการดำเนินการกับโปรโตคอล SMFIP_NR_* สำหรับ callback callback นั้นต้องตอบกลับด้วย SMFIS_NOREPLY เสมอ การใช้โค้ดการตอบกลับอื่นใด จะเป็นข้อห้ามของ application programming interface (API) ในบางกรณี หาก callback ของคุณสามารถส่งคืนค่าอื่น (เนื่องจาก ขาดแคลนรีซอร์ส) คุณต้องไม่ตั้งค่า SMFIP_NR_* และคุณต้องใช้ SMFIS_CONTINUE เป็นโค้ดส่งคืนดีฟอลต์ หรือ คุณสามารถหน่วงเวลาการรายงานปัญหาไปยัง callback ที่ใหม่กว่าซึ่ง SMFIP_NR_* ไม่ได้ตั้งค่าไว้

ฟังก์ชัน xxfi_connect callback:

วัตถุประสงค์

ฟังก์ชัน **xxfi_connect** callback จัดเตรียมข้อมูลการเชื่อมต่อ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_connect)(
        SMFICTX    *ctx,
        char       *hostname,
        _SOCK_ADDR *hostaddr);
```

คำอธิบาย

ฟังก์ชัน **xxfi_connect** callback ถูกเรียกหนึ่งครั้ง ณ เวลาเริ่มต้นการเชื่อมต่อ simple mail transfer protocol (SMTP) ในแต่ละครั้งและส่งคืนแฟล็ก `SMFIS_CONTINUE`

หมายเหตุ: หาก ตัวกรองก่อนหน้านี้ปฏิเสธการเชื่อมต่อในรูทีนฟังก์ชัน **xxfi_connect** callback ฟังก์ชัน **xxfi_connect** callback ของตัวกรอง จะไม่ถูกเรียก

อาร์กิวเมนต์

ตารางที่ 31. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
hostname	ชื่อโฮสต์ของผู้ส่งข้อความตามที่ได้กำหนดไว้โดย การค้นหาแบบย้อนกลับบนแอดเดรสของโฮสต์ หากการค้นหาแบบย้อนกลับล้มเหลวหรือ หากไม่มี IP แอดเดรสของชื่อโฮสต์ที่ได้รับการแก้ไขตรงกับ IP แอดเดรสเดิม ชื่อโฮสต์จะมี IP แอดเดรสของผู้ส่งข้อความ ซึ่งล้อมรอบด้วยวงเล็บเหลี่ยม (ตัวอย่างเช่น `[a.b.c.d]') หากการเชื่อมต่อ simple mail transfer protocol (SMTP) ถูกทำผ่าน stdin ค่าจะเป็น localhost
hostaddr	โฮสต์แอดเดรสตามที่กำหนดไว้โดยการเรียก **getpeername(2)** บนซ็อกเก็ต SMTP ค่าจะเป็น NULL หากซนิดไม่ได้รับการสนับสนุน ในเวอร์ชันปัจจุบัน หรือหากการเชื่อมต่อ SMTP ถูกทำขึ้นผ่าน **stdin**

ฟังก์ชัน xxfi_helo callback:

วัตถุประสงค์

ฟังก์ชัน **xxfi_helo** callback จัดการกับคำสั่ง **HELO หรือ EHLO**

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_helo)(
 SMFICTX *ctx,
 char *helohost
);
```

คำอธิบาย

ฟังก์ชัน **xxfi_hello** callback ถูกเรียกเมื่อใดก็ตามที่ไคลเอ็นต์ส่งคำสั่ง **HELO or EHLO** และส่งคืนแฟล็ก SMFIS_CONTINUE ดังนั้น callback อาจถูกเรียกหลายครั้งหรืออาจไม่ถูกเรียกเลย ข้อจำกัดบางอย่าง สามารถกำหนดได้โดยคอนฟิกูเรชัน mail transfer agent (MTA)

อาร์กิวเมนต์

ตารางที่ 32. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
helohost	ค่าที่ส่งไปยังคำสั่ง **HELO หรือ EHLO** ควรเป็นชื่อโดเมนของโฮสต์ที่ส่ง

ฟังก์ชัน xxfi_envfrom callback:

วัตถุประสงค์

ฟังก์ชัน **xxfi_envfrom** callback จัดการกับคำสั่ง `MAIL` (ผู้รับจดหมาย)

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_envfrom)(
 SMFICTX *ctx,
 char **argv
);
```

คำอธิบาย

ฟังก์ชัน xxfi_envfrom callback ถูกเรียกเมื่อไคลเอ็นต์ใช้คำสั่ง DATA และส่งคืนแฟล็ก SMFIS_CONTINUE

หมายเหตุ: สำหรับรายละเอียดเพิ่มเติม เกี่ยวกับการตอบกลับ ESMTP โปรดดู RFC 1869

อาร์กิวเมนต์

ตารางที่ 33. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter
argv	อาร์กิวเมนต์คำสั่ง null-terminated SMTP นั่นคือ argv[0] ถูกกักกันไว้เป็นแอดเดรสของผู้ส่ง อาร์กิวเมนต์ถัดมาคืออาร์กิวเมนต์ simple mail transfer protocol (ESMTP) ที่ขยายเพิ่ม

ค่าส่งคืน

ตารางที่ 34. ค่าส่งคืน

ไอเท็ม	คำอธิบาย
SMFIS_TEMPFAIL	ผู้ส่งและข้อความได้รับการปฏิเสธด้วยข้อผิดพลาดชั่วคราว ผู้ส่งใหม่ (และข้อความใหม่) อาจถูกระบุไว้ในภายหลัง และฟังก์ชัน **xxfi_abort** callback ไม่ได้ถูกเรียก
SMFIS_REJECT	ผู้ส่งและข้อความได้รับการปฏิเสธ ผู้ส่งใหม่ และข้อความอาจถูกระบุไว้และฟังก์ชัน **xxfi_abort** callback ไม่ได้ถูกเรียก
SMFIS_DISCARD	ข้อความได้รับการยอมรับและละทิ้ง และฟังก์ชัน **xxfi_abort** callback ไม่ได้ถูกเรียก
SMFIS_ACCEPT	ข้อความได้รับการยอมรับและฟังก์ชัน **xxfi_abort** callback ไม่ได้ถูกเรียก

ข้อมูลที่เกี่ยวข้อง

xxfi_abort

ฟังก์ชัน xxfi_envrcpt callback:

วัตถุประสงค์

ฟังก์ชัน xxfi_envrcpt callback จัดการกับคำสั่ง RCPT ของจดหมาย

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_envrcpt)(
 SMFICTX *ctx,
 char **argv
);
```

คำอธิบาย

ฟังก์ชัน xxfi_envrcpt callback ถูกเรียกหนึ่งครั้งต่อผู้รับ และมากกว่าหนึ่งครั้งต่อข้อความในทันที หลังฟังก์ชัน **xxfi_envfrom** callback และส่งคืนแฟล็ก SMFIS_CONTINUE

หมายเหตุ: สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับการตอบกลับ simple mail transfer protocol (ESMTP) โปรดดู RFC 1869

อาร์กิวเมนต์

ตารางที่ 35. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างดอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter
argv	อาร์กิวเมนต์คำสั่ง null-terminated SMTP นั่นคือ argv[0] ถูกกักกันไว้เป็นแอดเดรสของผู้รับ อาร์กิวเมนต์ถัดมาคืออาร์กิวเมนต์ simple mail transfer protocol (ESMTP) ที่ขยายเพิ่ม

ค่าส่งคืน

ตารางที่ 36. ค่าส่งคืน

ไอเท็ม	คำอธิบาย
SMFIS_TEMPFAIL	ผู้รับได้รับความล้มเหลวชั่วคราว ผู้รับเพิ่มเติมอาจยังคงถูกส่ง และฟังก์ชัน **xxfi_abort** callback ไม่ถูกเรียก
SMFIS_REJECT	ผู้รับได้รับการปฏิเสธ ผู้รับเพิ่มเติมอาจยังคงถูกส่ง และฟังก์ชัน **xxfi_abort** callback ไม่ถูกเรียก
SMFIS_DISCARD	ข้อความได้รับการยอมรับหรือละทิ้ง และฟังก์ชัน **xxfi_abort** callback ถูกเรียก
SMFIS_ACCEPT	ผู้รับได้รับการยอมรับและฟังก์ชัน **xxfi_abort** callback ไม่ได้ถูกเรียก

ข้อมูลที่เกี่ยวข้อง

xxfi_envfrom

xxfi_abort

ฟังก์ชัน xxfi_data callback: วัตถุประสงค์

ฟังก์ชัน xxfi_data callback จัดการกับคำสั่ง DATA

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_data)(
 SMFICTX *ctx
);
```

คำอธิบาย

ฟังก์ชัน **xxfi_data** callback ถูกเรียกเมื่อไคลเอ็นต์ใช้คำสั่ง DATA และส่งคืนแฟล็ก SMFIS_CONTINUE

หมายเหตุ: สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับการตอบกลับ ESMTP โปรดดู RFC 1869

อาร์กิวเมนต์

ตารางที่ 37. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter

ค่าส่งคืน

ตารางที่ 38. ค่าส่งคืน

ไอเท็ม	คำอธิบาย
SMFIS_TEMPFAIL	ข้อความได้รับการปฏิเสธด้วยข้อผิดพลาดชั่วคราว
SMFIS_REJECT	ข้อความได้รับการปฏิเสธ
SMFIS_DISCARD	ข้อความได้รับการยอมรับและละทิ้ง
SMFIS_ACCEPT	ข้อความได้รับการยอมรับ

ฟังก์ชัน xxfi_unknown callback:

วัตถุประสงค์

ฟังก์ชัน **xxfi_unknown** callback จัดการกับคำสั่ง simple mail transfer protocol (SMTP) ที่ไม่รู้จักและไม่ได้นำไปใช้งาน

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_unknown)(
 SMFICTX *ctx,
 const char *arg
);
```

คำอธิบาย

ฟังก์ชัน **xxfi_unknown** callback ถูกเรียกใช้เมื่อไคลเอ็นต์ใช้คำสั่ง SMTP ที่ไม่รู้จัก หรือไม่ได้นำไปใช้งานโดย mail transfer agent (MTA) และส่งคืนแฟล็ก SMFIS_CONTINUE

หมายเหตุ: เซิร์ฟเวอร์จะปฏิเสธคำสั่ง SMTP เสมอ ซึ่งเป็นไปได้ที่จะส่งคืนโค้ดระบุความผิดพลาด อื่นๆ

อาร์กิวเมนต์

ตารางที่ 39. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
arg	คำสั่ง SMTP ประกอบด้วยอาร์กิวเมนต์ทั้งหมด

ค่าส่งคืน

ตารางที่ 40. ค่าส่งคืน

ไอเท็ม	คำอธิบาย
`SMFIS_TEMPFAIL`	คำสั่งได้รับการปฏิเสธด้วยข้อผิดพลาดชั่วคราว
`SMFIS_REJECT`	คำสั่งได้รับการปฏิเสธ

ฟังก์ชัน xxfi_header callback:

วัตถุประสงค์

ฟังก์ชัน **xxfi_header** callback จัดการกับส่วนหัวข้อความ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_header)(
 SMFICTX *ctx,
 char *headerf,
 char *headerv
);
```

คำอธิบาย

ฟังก์ชัน **xxfi_header** callback ถูกเรียกหนึ่งครั้งสำหรับส่วนหัวข้อความแต่ละส่วนและส่งคืนแฟล็ก `SMFIS_CONTINUE`

หมายเหตุ:

- การเริ่มต้นด้วย sendmail 8.14 ให้ใส่ช่องว่างหลังโคลอนในฟิลด์ส่วนหัว ที่สงวนไว้หากร้องขอโดยใช้แฟล็ก `SMFIP_HDR_LEADSPC` ตัวอย่างมีดังต่อไปนี้:

  ```
  From: sender <f@example.com>
  To:  user <t@example.com>
  Subject:no
  ```

  จะถูกส่งไปยังพารามิเตอร์ `milter` ดังต่อไปนี้:

  ```
  "From", " sender <f@example.com>"
  "To", "  user <t@example.com>"
  "Subject", "no"
  ```

  ขณะที่ก่อนหน้านี้ (หรือไม่มีแฟล็ก `SMFIP_HDR_LEADSPC`) ซึ่งเป็นดังต่อไปนี้:

```
"From", "sender <f@example.com>"
"To", "user <t@example.com>"
"Subject", "no"
```

- ตัวกรองที่เก่ากว่าทำให้ส่วนหัวเปลี่ยนหรือเพิ่มไปเป็นตัวกรองใหม่
- สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับรูปแบบส่วนหัว โปรดดู RFC 822 และ RFC 2822.

อาร์กิวเมนต์

ตารางที่ 41. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
headerf	ชื่อฟิลด์ส่วนหัว
headerv	ค่าฟิลด์ส่วนหัว เนื้อหาของส่วนหัวอาจสอดแทรก ช่องว่างไว้ นั่นคือ บรรทัดจำนวนมากที่มีช่องว่างต่อไปนี้ ซึ่งบรรทัดจะถูกแบ่งโดย LF (ไม่ใช่ CR หรือ LF) ตัวยกเลิกบรรทัดส่วนท้าย (CR หรือ LF) จะถูกลบออก

ข้อมูลที่เกี่ยวข้อง:

RFC 2822

RFC 822

ฟังก์ชัน xxfi_eoh callback:
วัตถุประสงค์

ฟังก์ชัน **xxfi_eoh** callback จัดการกับส่วนท้ายของส่วนหัวของข้อความ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_eoh)(
 SMFICTX *ctx
);
```

คำอธิบาย

ฟังก์ชัน **xxfi_eoh** callback ถูกเรียกหนึ่งครั้งหลังจากส่วนหัวทั้งหมดถูกส่งและประมวลผลแล้ว และส่งคืนแฟล็ก `SMFIS_CONTINUE`

อาร์กิวเมนต์

ตารางที่ 42. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`

ฟังก์ชัน xxfi_body callback:
วัตถุประสงค์

ฟังก์ชัน xxfi_body callback จัดการกับชนิดของเนื้อความ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_body)(
 SMFICTX *ctx,
 unsigned char *bodyp,
 size_t len
);
```

คำอธิบาย

ฟังก์ชัน xxfi_body callback ไม่ได้ถูกเรียกหรือถูกเรียกหลายครั้งระหว่างฟังก์ชัน **xxfi_eoh** และ **xxfi_eom** callback และส่งคืนแฟล็ก SMFIS_CONTINUE

หมายเหตุ:

- **bodyp** ชี้ไปยังลำดับของไบต์ ซึ่งไม่ใช่สตริง C (ลำดับของอักขระที่ยกเลิกโดย '\0') ดังนั้น ห้ามใช้ฟังก์ชันสตริง C ตามปกติ เช่น **strlen(3)** บนบล็อกของไบต์ ลำดับของไบต์อาจมีอักขระ '\0' อยู่ภายใน บล็อก ดังนั้น แม้ว่า จะเพิ่มส่วนท้ายด้วย '\0' ฟังก์ชันสตริง C อาจยังคงล้มเหลวตามที่คาดการณ์ไว้
- เนื่องจากเนื้อข้อความสามารถขยายให้ใหญ่ได้ การนิยามฟังก์ชัน **xxfi_body** callback สามารถมีผลต่อผลการทำงานของตัวกรองได้
- End-of-lines จะถูกแทนค่าตามที่ได้รับจาก SMTP (ปกติคือ CR/LF)
- ตัวกรองที่เก่ากว่าทำให้เนื้อความเปลี่ยนไปเป็นตัวกรองใหม่
- เนื้อข้อความอาจถูกส่งในชิ้นข้อมูลหลายชุดพร้อมกับหนึ่งการเรียกไปยังฟังก์ชัน **xxfi_body** callback ต่อชิ้นข้อความ
- ฟังก์ชันนี้ส่งคืนแฟล็ก SMFIS_SKIP หากพารามิเตอร์ milter ได้รับชิ้นเนื้อความจำนวนมากเพื่อทำการตัดสินใจ แต่ยังคงต้องการเรียกทำงานฟังก์ชันการแก้ไขข้อความ ซึ่งจะได้รับอนุญาตให้เรียกจากฟังก์ชัน **xxfi_eom** callback
- พารามิเตอร์ milter ต้องเจรจาถึงลักษณะการทำงานนี้ด้วย mail transfer agent (MTA) นั่นคือ ต้องตรวจสอบว่าแฟล็กการดำเนินการกับโปรโตคอล SMFIP_SKIP พร้อมใช้งานและหากพร้อมใช้งาน พารามิเตอร์ `milter` ต้องรอขอแฟล็กนั้น

อาร์กิวเมนต์

ตารางที่ 43. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
bodyp	ตัวชี้ไปยังจุดเริ่มต้นของบล็อกของข้อมูลเนื้อความนี้ *bodyp* ไม่ถูกต้องภายนอกการเรียกไปยังฟังก์ชัน **xxfi_body** callback
len	จำนวนข้อมูลที่ชี้ไปโดย *bodyp*

ข้อมูลที่เกี่ยวข้อง

xxfi_eoh

xxfi_eom

ฟังก์ชัน xxfi_eom callback:
วัตถุประสงค์

ฟังก์ชัน **xxfi_eom** callback จัดการกับส่วนท้ายของข้อความ

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_eom)(
 SMFICTX *ctx
);
```

คำอธิบาย

ฟังก์ชัน **xxfi_eom** callback ถูกเรียกหนึ่งครั้งหลังจากการเรียกทั้งหมดไปยังฟังก์ชัน **xxfi_body** callback สำหรับข้อความที่กำหนดไว้และส่งคืนแฟล็ก `SMFIS_CONTINUE`

หมายเหตุ: ตัวกรอง ต้องการทำให้การเปลี่ยนแปลงทั้งหมดไปยังส่วนหัวข้อความ เนื้อความ และซองจดหมายในฟังก์ชัน **xxfi_eom** callback การแก้ไข จะทำผ่านรูทีน `smfi_*`

อาร์กิวเมนต์

ตารางที่ 44. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`

ข้อมูลที่เกี่ยวข้อง

xxfi_body

ฟังก์ชัน xxfi_abort callback:

วัตถุประสงค์

ฟังก์ชัน **xxfi_abort** callback จัดการกับข้อความปัจจุบันที่กำลังถูกยกเลิก

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_abort)(
 SMFICTX *ctx
);
```

คำอธิบาย

ฟังก์ชัน **xxfi_abort** callback ถูกเรียก ณ เวลาใดๆ ในระหว่างที่ประมวลผลข้อความ (นั่นคือ ระหว่างรูทีน message-oriented บางตัวและฟังก์ชัน **xxfi_eom** callback) และส่งคืนแฟล็ก SMFIS_CONTINUE

หมายเหตุ:

- ฟังก์ชัน **xxfi_abort** callback ต้องเรียกคืนรีซอร์สใดๆ ที่จัดสรรไว้บนพื้นฐานต่อข้อความ และต้องทนต่อการถูกเรียกระหว่าง message-oriented callbacks สองตัวใดๆ
- การเรียกฟังก์ชัน **xxfi_abort** and **xxfi_eom** callback เป็นการเรียกแบบ exclusive
- ฟังก์ชัน **xxfi_abort** callback ไม่ได้รับผิดชอบต่อการเรียกคืน ข้อมูลที่ระบุเฉพาะการเชื่อมต่อ เนื่องจากฟังก์ชัน **xxfi_close** callback ถูกเรียกเมื่อปิดการเชื่อมต่อ
- เนื่องจากข้อความปัจจุบันได้ถูกยกเลิก ค่าส่งคืนจะถูกละเว้นในปัจจุบัน
- ฟังก์ชัน **xxfi_abort** callback จะถูกเรียก หากข้อความถูกยกเลิกภายนอกการควบคุมของตัวกรองเท่านั้น และตัวกรองไม่ได้ประมวลผล message-oriented จนเสร็จสิ้น ตัวอย่างเช่น หากตัวกรอง ได้ส่งคืนแฟล็ก SMFIS_ACCEPT, SMFIS_REJECT หรือ SMFIS_DISCARD จากรูทีน message-oriented ฟังก์ชัน **xxfi_abort** callback จะไม่ถูกเรียกแม้ว่าข้อความจะถูกยกเลิกในภายหลัง ซึ่งอยู่ภายนอกการควบคุม

อาร์กิวเมนต์

ตารางที่ 45. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter

ข้อมูลที่เกี่ยวข้อง

xxfi_close

xxfi_eom

ฟังก์ชัน xxfi_close callback:
วัตถุประสงค์

ฟังก์ชัน **xxfi_close** callback ปิดการเชื่อมต่อปัจจุบัน

ไวยากรณ์

```
#include <libmilter/mfapi.h>
sfsistat (*xxfi_close)(
 SMFICTX *ctx
);
```

คำอธิบาย

ฟังก์ชัน **xxfi_close** callback ถูกเรียกเพียงหนึ่งครั้งที่ส่วนท้ายของการเชื่อมต่อเสมอ และส่งคืนแฟล็ก SMFIS_CONTINUE

ฟังก์ชัน **xxfi_close** callback อาจถูกเรียกโดยใช้งานไม่ได้ นั่นคือ ก่อนที่จะเรียกฟังก์ชัน **xxfi_connect** callback หลังจากที่สร้างการเชื่อมต่อโดย mail transfer agent (MTA) ไปยังตัวกรองแล้ว หาก MTA ตัดสินว่า ทราฟิกของการเชื่อมต่อนี้จะถูกละทิ้ง (ตัวอย่างเช่น ผ่านผลลัพธ์ access_db) จะไม่มีข้อมูลใดๆ ที่ส่งผ่านไปยังตัวกรองจาก MTA จนกว่าไคลเอ็นต์ จะปิด ณ เวลานั้น ฟังก์ชัน **xxfi_close** callback จะถูกเรียก ดังนั้น จึงสามารถ callback เฉพาะที่ใช้กับการเชื่อมต่อที่กำหนดไว้ และคุณควรสื่อสารถึงความเป็นไปได้เมื่อ ทำโค้ดฟังก์ชัน **xxfi_close** callback เอง โดยเฉพาะอย่างยิ่ง เมื่อมีการตั้งสมมติฐานที่ไม่ถูกต้องว่า ตัวชี้คอนเท็กซ์ส่วนบุคคล จะมีค่าเป็นค่าอื่นที่ไม่ใช่ NULL ใน callback นี้

ฟังก์ชัน **xxfi_close** callback ถูกเรียกเมื่อปิดแม้ว่าธุรกรรมเมลก่อนหน้านี้ ถูกยกเลิก

ฟังก์ชัน **xxfi_close** callback รับผิดชอบต่อการล้างรีซอร์สใดๆ ที่จัดสรรไว้บนพื้นฐานต่อการเชื่อมต่อ

เนื่องจาก การเชื่อมต่อกำลังปิด ค่าส่งคืนจะถูกละเว้น

อาร์กิวเมนต์

ตารางที่ 46. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ libmilter

ข้อมูลที่เกี่ยวข้อง

xxfi_connect

ฟังก์ชัน xxfi_negotiate callback: วัตถุประสงค์

ฟังก์ชัน **xxfi_negotiate** callback จัดการกับการเจรจา

ไวยากรณ์

```
#include <libmilter/mfapi.h>
#include <libmilter/mfdef.h>
sfsistat (*xxfi_negotiate)(
        SMFICTX    "ฟังก์ชัน xxfi_negotiate callback",
 unsigned long f0,
 unsigned long f1,
 unsigned long f2,
 unsigned long f3,
```

```
unsigned long *pf0,
unsigned long *pf1,
unsigned long *pf2,
unsigned long *pf3);
```

คำอธิบาย

ฟังก์ชัน **xxfi_negotiate** callback ถูกเรียกที่จุดเริ่มต้นของการเชื่อมต่อ simple mail transfer protocol (SMTP) แต่ละครั้งและส่งคืนแฟล็ก SMFIS_ALL_OPTS

ด้วยฟังก์ชันนี้ พารามิเตอร์ milter สามารถกำหนดแบบไดนามิกได้ และร้องขอการดำเนินการและการดำเนินการในระหว่างการเริ่มทำงาน ในเวอร์ชันก่อนหน้านี้ การดำเนินการ (f0) ถูกแก้ไขในฟิลด์แฟล็กของโครงสร้าง **smfiDesc** และขั้นตอนโปรโตคอล (f1) ถูกได้รับโดยตรวจสอบ callback ที่นิยามไว้ เนื่องจาก ส่วนขยายในเวอร์ชัน milter ใหม่ เช่น การเลือกแบบสแตติกไม่ได้ทำงานหากพารามิเตอร์ milter จำเป็นต้องมีการดำเนินการใหม่ที่ไม่พร้อมใช้งานเมื่อกล่าวถึง mail transfer agent (MTA) แบบเก่า ดังนั้น การเจรจากับ callback สามารถกำหนดการดำเนินการที่พร้อมใช้งานและเลือกฟังก์ชัน callback แบบไดนามิกที่ต้องการและถูกนำเสนอ หากการดำเนินการบางอย่างไม่พร้อมใช้งาน พารามิเตอร์ milter อาจกลับสู่โหมดที่เก่ากว่าหรือหยุดเซสชันและบอกให้ผู้ใช้ อัพเกรด

ขั้นตอนของโปรโตคอล

(f1, *pf1)

:

- SMFIP_RCPT_REJ: ด้วยการตั้งค่าบิตนี้ พารามิเตอร์ milter สามารถร้องขอให้ MTA ส่งคำสั่ง RCPT ที่ได้รับการปฏิเสธเนื่องจากผู้ใช้ไม่รู้จัก (หรือเหตุผลที่คล้ายกัน) แต่ไม่ใช่ฟังก์ชันเหล่านั้นทั้งหมดที่ถูกปฏิเสธเนื่องจากข้อผิดพลาดทางทางไวยากรณ์ หาก milter ร้องขอขั้นตอนของโปรโตคอลนี้ พารามิเตอร์ควรตรวจสอบแมโคร **{rcpt_mailer}**: หากพารามิเตอร์นั้นถูกตั้งค่า ให้มีข้อผิดพลาด ผู้รับจะได้รับการปฏิเสธโดย MTA โดยปกติ แมโคร **{rcpt_host}** และ **{rcpt_addr}** จะมีโค้ดสถานะที่ปรับปรุงแล้ว และข้อความแสดงข้อผิดพลาดในกรณีนั้น
- SMFIP_SKIP บ่งชี้ว่า MTA เข้าใจถึงโค้ดส่งคืน SMFIS_SKIP
- SMFIP_NR_* บ่งชี้ว่า MTA เข้าใจถึงโค้ดส่งคืน SMFIS_NOREPLY มีแฟล็กสำหรับขั้นตอนของโปรโตคอลที่หลากหลาย:
  - SMFIP_NR_CONN: “ฟังก์ชัน xxfi_connect callback” ในหน้า 91
  - SMFIP_NR_HELO: “ฟังก์ชัน xxfi_helo callback” ในหน้า 92
  - SMFIP_NR_MAIL: “ฟังก์ชัน xxfi_envfrom callback” ในหน้า 92
  - SMFIP_NR_RCPT: “ฟังก์ชัน xxfi_envrcpt callback” ในหน้า 93
  - SMFIP_NR_DATA: “ฟังก์ชัน xxfi_data callback” ในหน้า 94
  - SMFIP_NR_UNKN: “ฟังก์ชัน xxfi_unknown callback” ในหน้า 95
  - SMFIP_NR_EOH: “ฟังก์ชัน xxfi_eoh callback” ในหน้า 97
  - SMFIP_NR_BODY: “ฟังก์ชัน xxfi_body callback” ในหน้า 98
  - SMFIP_NR_HDR: “ฟังก์ชัน xxfi_header callback” ในหน้า 96
- แฟล็ก SMFIP_HDR_LEADSPC บ่งชี้ว่า MTA สามารถส่งค่าส่วนหัวด้วยการนำหน้าด้วยช่องว่างที่ไม่เปลี่ยนแปลง หากขั้นตอนของโปรโตคอลนี้ ถูกร้องขอ MTA จะไม่เพิ่มการนำหน้าด้วยช่องว่างให้กับส่วนหัว เมื่อเพิ่ม แทรก หรือเปลี่ยนแปลง
- MTA สามารถสั่งไม่ให้ส่งข้อมูลเกี่ยวกับขั้นตอน SMTP ที่หลากหลาย แฟล็กเหล่านี้จะขึ้นต้นด้วย: SMFIP_NO*

- SMFIP_NOCONNECT: “ฟังก์ชัน xxfi_connect callback” ในหน้า 91
- SMFIP_NOHELO: “ฟังก์ชัน xxfi_header callback” ในหน้า 96
- SMFIP_NOMAIL: “ฟังก์ชัน xxfi_envfrom callback” ในหน้า 92
- SMFIP_NORCPT: “ฟังก์ชัน xxfi_envrcpt callback” ในหน้า 93
- SMFIP_NOBODY: “ฟังก์ชัน xxfi_body callback” ในหน้า 98
- SMFIP_NOHDRS: “ฟังก์ชัน xxfi_header callback” ในหน้า 96
- SMFIP_NOEOH: “ฟังก์ชัน xxfi_eoh callback” ในหน้า 97
- SMFIP_NOUNKNOWN: “ฟังก์ชัน xxfi_unknown callback” ในหน้า 95
- SMFIP_NODATA: “ฟังก์ชัน xxfi_data callback” ในหน้า 94

สำหรับแต่ละ **xxfi_* callbacks** เหล่านั้นซึ่งพารามิเตอร์ `milter` ไม่ได้ใช้ แฟล็กที่สอดคล้องกันควรถูกตั้งค่าไว้ใน `*pf1`.

การดำเนินการที่พร้อมใช้งาน

`(f0, *pf0)`

ถูกกล่าวถึงใน (xxfi_flags)

หาก `milter` ส่งคืนแฟล็ก SMFIS_CONTINUE `milter` ตั้งค่าการดำเนินการตามความต้องการและขั้นตอนของโปรโตคอลผ่านพารามิเตอร์ (เอาต์พุต) `pf0` และ `pf1` (ซึ่งสอดคล้องกับ `f0` และ `f1` ตามลำดับ) พารามิเตอร์ (เอาต์พุต) `pf2` และ `pf3` ควรถูกตั้งค่าเป็น 0 สำหรับความเข้ากันได้กับเวอร์ชันในอนาคต

อาร์กิวเมนต์

ตารางที่ 47. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
`ctx`	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
`f0`	การดำเนินการถูกนำเสนอโดย MTA
`f1`	ขั้นตอนของโปรโตคอลถูกนำเสนอโดย MTA
`f2`	สำหรับส่วนขยายในอนาคต
`f3`	สำหรับส่วนขยายในอนาคต
`pf0`	แอ็คชันที่ร้องขอโดย `milter`
`pf1`	ขั้นตอนของโปรโตคอลถูกร้องขอโดย `milter`
`pf2`	สำหรับส่วนขยายในอนาคต
`pf3`	สำหรับส่วนขยายในอนาคต

ค่าส่งคืน

ตารางที่ 48. ค่าส่งคืน

ไอเท็ม	คำอธิบาย
SMFIS_ALL_OPTS	หาก milter ต้องการตรวจสอบขั้นตอนของโปรโตคอล ที่พร้อมใช้งานและการดำเนินการ ซึ่งสามารถส่งคืนแฟล็ก SMFIS_ALL_OPTS และ MTA ควรสร้างขั้นตอนโปรโตคอลทั้งหมดและการดำเนินการที่พร้อมใช้งานกับ milter ในกรณีนี้ ไม่มีค่าที่ควรกำหนดให้กับพารามิเตอร์เอาต์พุต pf0 –pf3 ตามที่ควรจะถูกละทิ้ง
SMFIS_REJECT	การเริ่มทำงาน milter จะล้มเหลวและจะไม่ถูกติดต่อ อีกครั้ง (สำหรับการเชื่อมต่อปัจจุบัน)
SMFIS_CONTINUE	ดำเนินการประมวลผลต่อ ในกรณีนี้พารามิเตอร์ milter ต้องตั้งค่าพารามิเตอร์เอาต์พุตทั้งหมด pf0 – pf3 โปรดดูสิ่งต่อไปนี้สำหรับคำอธิบายถึงวิธีการตั้งค่าพารามิเตอร์เอาต์พุตเหล่านี้

ฟังก์ชันอื่นๆ และค่าคงที่

ฟังก์ชันอื่นๆ และค่าคงที่ดึงพารามิเตอร์ข้อมูลเวอร์ชันของพารามิเตอร์ libmilter

ตารางที่ 49. ฟังก์ชันค่าคงที่

ไอเท็ม	คำอธิบาย
smfi_version	ฟังก์ชัน **smfi_version** ดึงข้อมูลเวอร์ชัน (รันไทม์) ของพารามิเตอร์ libmilter
smfi_setsymlist	**smfi_setsymlist** ตั้งค่ารายชื่อแมโครที่พารามิเตอร์ libmilter ต้องการรับจาก mail transfer agent (MTA) สำหรับขั้นตอน ของโปรโตคอล

ตารางที่ 50. ฟังก์ชันค่าคงที่

ไอเท็ม	คำอธิบาย
SMFI_VERSION	**SMFI_VERSION** ขอรับเวอร์ชันรันไทม์ของพารามิเตอร์ libmilter

ฟังก์ชัน smfi_version:
วัตถุประสงค์

ฟังก์ชัน **smfi_version** จัดเตรียมข้อมูลเวอร์ชัน libmilter (รันไทม์)

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_version(
 unsigned int *pmajor,
 unsigned int *pminor,
 unsigned int *ppl
);
```

คำอธิบาย

ฟังก์ชัน **smfi_version** callback อาจถูกเรียกได้ ณ เวลาใดๆ

เวอร์ชัน compile-time ของไลบรารี libmilter พร้อมใช้งานในแมโคร SMFI_VERSION หากต้องการ แตกเวอร์ชันหลักและเวอร์ชันรองพร้อมกับระดับของแพ็ตช์ปัจจุบัน จากแมโครนี้ แมโคร SM_LM_VRS_MAJOR(v), SM_LM_VRS_MINOR (v) และแมโคร SM_LM_VRS_PLVL(v) อาจถูกนำมาใช้ พารามิเตอร์ milter สามารถตรวจสอบแมโคร SMFI_VERSION เพื่อกำหนดฟังก์ชัน ที่ต้องการใช้ (ณ เวลาคอมไพล์ผ่านข้อความสั่งตัวประมวลผลก่อน C) การใช้แมโครนี้ และฟังก์ชัน **smfi_version** นี้ พารามิเตอร์ milter สามารถกำหนดได้ ณ เวลารันไทม์ซึ่งได้ถูกลิงก์ (แบบไดนามิก) กับเวอร์ชัน libmilter ที่คาดการณ์ไว้ ฟังก์ชันบางตัว ต้องเปรียบเทียบเวอร์ชันหลักและเวอร์ชันรองเท่านั้น ไม่ใช่ระดับของแพ็ตช์ นั่นคือ ไลบรารี libmilter จะทำงานร่วมกันได้กับระดับของแพ็ตช์ที่แตกต่างกัน

อาร์กิวเมนต์

ตารางที่ 51. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
pmajor	ตัวชี้ไปยังตัวแปร int ที่ไม่ได้ลงนามเพื่อเก็บหมายเลข เวอร์ชันหลัก
pminor	ตัวชี้ไปยังตัวแปร int ที่ไม่ได้ลงนามเพื่อเก็บหมายเลข เวอร์ชันรอง
ppl	ตัวชี้ไปยังตัวแปร int ที่ไม่ได้ลงนามเพื่อเก็บหมายเลข ระดับของแพ็ตช์

ค่าส่งคืน

ฟังก์ชัน **smfi_version** ส่งคืนค่า MI_SUCCESS

ฟังก์ชัน smfi_setsymlist:
วัตถุประสงค์

ฟังก์ชัน **smfi_setsymlist** ตั้งค่ารายการของแมโครที่พารามิเตอร์ milter ต้องการรับจาก mail transfer agent (MTA) สำหรับขั้นตอนของโปรโตคอล

ไวยากรณ์

```
#include <libmilter/mfapi.h>
int smfi_setsymlist(
        SMFICTX    *ctx,
 int        stage,
 char     *macros
);
```

คำอธิบาย

ฟังก์ชัน **smfi_setsymlist** callback ต้องถูกเรียกในระหว่างฟังก์ชัน **xxfi_negotiate** และฟังก์ชันนี้สามารถใช้เพื่อเขียนทับรายการแมโครที่พารามิเตอร์ milter ต้องการรับจาก mail transfer agent (MTA)

หมายเหตุ: มีข้อจำกัดภายใน เกี่ยวกับจำนวนของแมโครที่สามารถตั้งค่าได้ (ปัจจุบันคือ 5) อย่างไรก็ตาม ข้อจำกัดนี้ไม่ได้บังคับไว้โดยพารามิเตอร์ milter แต่บังคับไว้โดย MTA เท่านั้น และการละเมิดที่อาจเกิดขึ้นได้ของข้อจำกัดนี้ ไม่ได้สื่อสารกลับไปยังพารามิเตอร์ milter

อาร์กิวเมนต์

ตารางที่ 52. อาร์กิวเมนต์

ไอเท็ม	คำอธิบาย
ctx	โครงสร้างคอนเท็กซ์แบบทึบถูกเก็บไว้ในพารามิเตอร์ `libmilter`
stage	ขั้นตอนของโปรโตคอลระหว่างรายการแมโครควร ถูกนำมาใช้โปรดดูไฟล์ `include/libmilter/mfapi.h` สำหรับค่าที่ถูกต้อง และมองหาแมโคร C ที่มี SMFIM_ prefix ขั้นตอนของโปรโตคอลที่พร้อมใช้งานอย่างน้อยที่สุดคือ การเชื่อมต่อเริ่มแรก HELO หรือ EHLO, MAIL, RCPT, DATA ส่วนท้าย ของส่วนหว และส่วนท้ายของข้อความ
แมโคร	รายการแมโคร (คั่นด้วยช่องว่าง) ตัวอย่างเช่น: "**{rcpt_mailer} {rcpt_host}**"

ค่าส่งคืน

ฟังก์ชัน **smfi_setsymlist** ส่งคืนค่า MI_FAILURE ในกรณีต่อไปนี้ และ ฟังก์ชันส่งคืน MI_SUCCESS

- มีหน่วยความจำอิสระไม่เพียงพอต่อการทำสำเนาของรายการแมโคร
- *แมโคร* คือค่า NULL หรือค่าว่าง
- *ขั้นตอน* ไม่ใช่ขั้นตอนโปรโตคอลที่ถูกต้อง
- รายการแมโครสำหรับขั้นตอนได้ถูกตั้งค่าไว้ก่อน

ข้อมูลที่เกี่ยวข้อง

xxfi_negotiate

ดีบักแฟล็กสำหรับ sendmail

มีจำนวนของดีบักแฟล็กมากมายที่ถูกสร้างไว้ในคำสั่ง **sendmail**

ดีบักแฟล็กแต่ละตัวมีหมายเลข และระดับ โดยที่ระดับสูงกว่าจะพิมพ์ข้อมูลมากกว่า ระเบียบเป็นระดับที่สูงกว่า 9 การพิมพ์ดังนั้นข้อมูลที่มากกว่าที่มันถูกใช้ เฉพาะสำหรับการดีบักส่วนของโค้ดนั้นๆ ดีบักแฟล็กถูกตั้งโดยใช้แฟล็ก **-d** ดังแสดงในตัวอย่างต่อไปนี้ :

```
debug-flag:     -d debug-list
debug-list:     debug-flag[.debug-flag]*
debug-flag:     debug-range[.debug-level]
debug-range:    integer|integer-integer
debug-level:    integer
```

```
-d12            Set flag 12 to level 1
-d12.3          Set flag 12 to level 3
-d3-17          Set flags 3 through 17 to level 1
-d3-17.4        Set flags 3 through 17 to level 4
```

ดีบักแฟล็กที่มีให้ใช้ คือ :

ไอเท็ม	คำอธิบาย
-d0	การดีบักทั่วไป
-d1	แสดงข้อมูลที่ส่ง
-d2	ลงท้ายด้วย *finis*( )
-d3	พิมพ์โหลดโดยเฉลี่ย
-d4	พื้นที่ดิสก์เพียงพอ
-d5	แสดงเหตุการณ์
-d6	แสดงเมลที่ล้มเหลว
-d7	ชื่อไฟล์ของคิว
-d8	DNS name resolution
-d9	ติดตามเคียวรี RFC1413
-d9.1	ทำให้ชื่อโฮสต์เป็นที่ยอมรับ
-d10	แสดงการนำส่งผู้รับ
-d11	ติดตามการนำส่ง
-d12	แสดงการแม็พของโฮสต์ที่เกี่ยวข้อง
-d13	แสดงการนำส่ง
-d14	แสดงฟิลด์ส่วนหัวคอมมา
-d15	แสดงเน็ตเวิร์กได้รับกิจกรรมการร้องขอ
-d16	การเชื่อมต่อขาออก
-d17	รายชื่อโฮสต์ MX

หมายเหตุ: มีดีบักแฟล็กเกือบ 200 ตัวที่ถูกกำหนดใน **sendmail**

Internet Message Access Protocol และ Post Office Protocol

AIX จัดเตรียมการใช้เซิร์ฟเวอร์การส่งอินเตอร์เน็ตเมล 2 ชนิดสำหรับการเข้าถึงเมลแบบรีโมต

- **Post Office Protocol** (**POP** หรือ **POP3DS**)
- **Internet Message Access Protocol** (**IMAP** หรือ **IMAPDS**)

เซิร์ฟเวอร์แต่ละชนิดจะเก็บและให้การเข้าถึงข้อความและอิเล็กทรอนิก การใช้โปรโตคอลการเข้าถึงเมลเหล่านี้บนเซิร์ฟเวอร์กำจัดข้อกำหนดที่ว่า เพื่อที่จะรับเมล คอมพิวเตอร์ต้องทำงานอยู่ตลอดเวลา

เซิร์ฟเวอร์ **POP** หรือ **POP3DS** จัดเตรียมระบบเมลแลลออฟไลน์โดยที่ไคลเอ็นต์ใช้ **POP** หรือ **POP3DS** ไคลเอ็นต์ซอฟต์แวร์สามารถเข้าถึงเมลเซิร์ฟเวอร์แบบรีโมตเพื่อดึงข้อความเมล ไคลเอ็นต์สามารถดาวน์โหลดข้อความเมลและลบข้อความจากเซิร์ฟเวอร์ได้ทันที หรือดาวน์โหลดข้อความและปล่อยให้ข้อความยังอยู่บนเซิร์ฟเวอร์ **POP** หรือ **POP3DS** หลังจากที่เมลถูกดาวน์โหลดไปยังเครื่องไคลเอ็นต์ ประมวลผลเมลทั้งหมดจะเป็นแบบโลคัลกับเครื่องไคลเอ็นต์ เซิร์ฟเวอร์ **POP** จะยอมให้เข้าถึงเมลบ็อกซ์ของผู้ใช้ทีละหนึ่งไคลเอ็นต์ **POP3DS** เวอร์ชันจะใช้ไลบรารี OpenSSL ซึ่งต้องการ certificate ความปลอดภัย

เซิร์ฟเวอร์ **IMAP** หรือ **IMAPDS** จัดเตรียมซุปเปอร์เซ็ตของการทำงานของ **POP** แต่มีอินเตอร์เฟสที่แตกต่าง เซิร์ฟเวอร์ **IMAP** หรือ **IMAPDS** จัดเตรียมเซอร์วิสแบบออฟไลน์ พร้อมด้วยเซอร์วิสแบบออนไลน์ และเซอร์วิสที่ถูกตัดการเชื่อมต่อ โปรโตคอลถูกออกแบบเพื่อยอมให้การจัดการกับเมลบ็อกซ์แบบรีโมตเหมือนกับว่ามันเป็นแบบโลคัล ตัวอย่างเช่น ไคลเอ็นต์สามารถทำการค้นหาและมาร์กข้อความด้วยแฟล็กสถานะ เช่น **deleted** หรือ **answered** นอกจากนี้ ข้อความสามารถคงอยู่บนฐานข้อมูลของเซิร์ฟเวอร์จนกว่ามันจะถูกลบอย่างแน่นอน เซิร์ฟเวอร์ **IMAP** ยังยอมให้มีการเข้าถึงเมลบ็อกซ์ของผู้ใช้แบบโต้ตอบพร้อมๆกันโดยหลายไคลเอ็นต์ **IMAPDS** เวอร์ชันจะใช้ไลบรารี OpenSSL ซึ่งต้องการ certificates ความปลอดภัย

เซิร์ฟเวอร์แต่ละชนิดถูกใช้สำหรับการเข้าถึงเมลเท่านั้น เซิร์ฟเวอร์เหล่านี้ขึ้นอยู่กับ **Simple Mail Transfer Protocol** (**SMTP**) สำหรับการส่งเมล

แต่ละโปรโตคอลเป็นโปรโตคอลแบบเปิด โดยขึ้นอยู่กับมาตรฐานที่ถูกอธิบายใน RFCs เซิร์ฟเวอร์ **IMAP** จะขึ้นอยู่กับ RFC 2060 และ 2061 และเซิร์ฟเวอร์ **POP** ขึ้นอยู่กับ RFC 1939 ทั้งสองเซิร์ฟเวอร์เป็นแบบ connection-oriented โดยใช้ซ็อกเก็ต TCP เซิร์ฟเวอร์ **IMAP** จะรับฟังบนพอร์ต 143 และเซิร์ฟเวอร์ **IMAPDS** จะรับฟังบนพอร์ต 993 เซิร์ฟเวอร์ **POP** จะรับฟังบนพอร์ต 110 และเซิร์ฟเวอร์ **POP3DS** จะรับฟังบนพอร์ต 995 เซิร์ฟเวอร์ทั้งหมดจะถูกจัดการโดย **inetd** daemon

ข้อกำหนด: เพื่อใช้ OpenSSL เวอร์ชัน คุณต้องติดตั้ง OpenSSL OpenSSL จะมีบน*AIX Toolbox สำหรับ Linux Applications* CD

การตั้งค่าเซิร์ฟเวอร์ IMAP และ POP

ใช้โพรซีเดอร์นี้เพื่อตั้งค่าเซิร์ฟเวอร์ **IMAP** และ **POP**

ในการทำงานนี้ คุณต้องมีสิทธิของ root

1. ยกเลิกหมายเหตุ entry ของการตั้งค่า **imapd** หรือ **imapds** และ **pop3d** หรือ **pop3ds** ในไฟล์ `/etc/inetd.conf` ต่อไปนี้เป็นตัวอย่างของ entry ของการตั้งค่า:

   ```
   #imap2  stream  tcp     nowait  root    /usr/sbin/imapd imapd
   #pop3   stream  tcp     nowait  root    /usr/sbin/pop3d pop3d
   #imaps  stream  tcp     nowait  root    /usr/sbin/imapds imapds
   pop3s   stream  tcp     nowait  root    /usr/sbin/pop3ds pop3ds
   ```

2. รีเฟรช **inetd** daemon โดยการรันคำสั่งต่อไปนี้ :

   ```
   refresh -s inetd
   ```

การรันการทดสอบการตั้งค่า:

รันการทดสอบเล็กน้อยเพื่อตรวจสอบว่าเซิร์ฟเวอร์พร้อมที่จะทำงาน

1. ลำดับแรก ตรวจสอบว่าเซิร์ฟเวอร์กำลังฟังบนพอร์ตของมัน ในการทำดังกล่าว พิมพ์คำสั่งต่อไปนี้ที่จุดรับคำสั่ง กด Enter หลังจากพิมพ์แต่ละคำสั่ง:

   ```
   netstat -a | grep imap
   netstat -a | grep pop
   ```

   ต่อไปนี้เป็นเอาต์พุตจากคำสั่ง **netstat** :

   ```
   tcp     0       0     *.imap2         *.*         LISTEN
   tcp     0       0     *.imaps         *.*         LISTEN
   tcp     0       0     *.pop3          *.*         LISTEN
   tcp     0       0     *.pop3s         *.*         LISTEN
   ```

2. ถ้าคุณไม่ได้รับเอาต์พุตเหมือยแบบนี้ ให้ตรวจสอบ entry ในไฟล์ `/etc/inetd.conf` อีกครั้ง และจากนั้น รันคำสั่ง **refresh -s inetd** อีกครั้ง
3. เพื่อทดสอบการตั้งค่าของเซิร์ฟเวอร์ imapd ใช้ Telnet เพื่อเข้าถึงเซิร์ฟเวอร์ imap2 พอร์ต 143 (สำหรับ IMAPDS Telnet พอร์ต 993) เมื่อคุณเชื่อมต่อโดยใช้ Telnet คุณจะได้รับพร้อมต์ของ imapd จากนั้นคุณสามารถใส่คำสั่ง IMAP เวอร์ชัน 4 ดังที่ถูกกำหนดใน RFC 1730 เพื่อรันคำสั่ง พิมพ์จุดหนึ่งจุด (.)ตามด้วยช่องว่าง จากนั้นโทเค็น ชื่อคำสั่ง และพารามิเตอร์ใดๆ โทเค็นถูกใช้เพื่อเรียงลำดับชื่อของคำสั่ง ตัวอย่าง เช่น:

   ```
   . token CommandName parameters
   ```

   รหัสผ่านจะถูก echo เมื่อคุณ Telnet ไปยังเซิร์ฟเวอร์ imapd

   ในตัวอย่างของ Telnet ต่อไปนี้ คุณต้องใส่รหัสผ่านของคุณ โดยที่ *id_password* ถูกระบุในคำสั่ง **login**

คำแนะนำ: สำหรับ IMAPDS คำสั่งและเอาต์พุตจะต่างไปเล็กน้อย

```
telnet e-xbelize 143
Trying...
Connected to e-xbelize.austin.ibm.com.
Escape character is '^]'.
* OK e-xbelize.austin.ibm.com IMAP4 server ready
. 1 login id id_password
. OK
. 2 examine /usr/spool/mail/root
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
* 0 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 823888143]
. OK [READ-ONLY] Examine completed
. 3 logout
* BYE Server terminating connection
. OK Logout completed
Connection closed.
```

4. เพื่อทดสอบการตั้งค่าของเซิร์ฟเวอร์ pop3d ใช้ Telnet เพื่อเข้าถึงพอร์ต POP3 110 (สำหรับ POP3DS Telnet พอร์ต 995) เมื่อคุณเชื่อมต่อโดยใช้ Telnet คุณจะได้รับพร้อมต์ของ pop3d คุณสามารถใส่คำสั่ง POP ที่ถูกกำหนดใน RFC 1725 เพื่อรันหนึ่งในคำสั่ง พิมพ์จุดหนึ่งจุด (.)ตามด้วยช่องว่าง และจากนั้นชื่อของคำสั่ง ตัวอย่าง เช่น:

   ```
   . CommandName
   ```

   รหัสผ่านจะถูก echo เมื่อคุณ Telnet ไปยังเซิร์ฟเวอร์ pop3d

   ในตัวอย่างของ Telnet ต่อไปนี้ คุณต้องใส่รหัสผ่านของคุณ โดยที่ *id_password* ถูกระบุในคำสั่ง **pass**

   คำแนะนำ: สำหรับ POP3DS คำสั่งและเอาต์พุตจะต่างไปเล็กน้อย

   ```
   telnet e-xbelize 110
   Trying...
   Connected to e-xbelize.austin.ibm.com.
   Escape character is '^]'.
   +OK e-xbelize.austin.ibm.com POP3 server ready
   user id
   +OK Name is a valid mailbox
   pass id_password
   +OK Maildrop locked and ready
   แสดงรายการ
   +OK scan listing follows
   .
   stat
   +OK 0 0
   quit
   +OK
   Connection closed.
   ```

การล็อกอินด้วยสิ่งอำนวยความสะดวก SYSLOG

เซิร์ฟเวอร์ซอฟต์แวร์ **IMAP** (และ **IMAPDS**) และ **POP** (และ **POP3DS**) ส่งข้อความบันทึกไปยังสิ่งอำนวยความสะดวก **SYSLOG**

1. เพื่อกำหนดคอนฟิกระบบสำหรับการล็อกอิน **IMAP** และ **POP** โดยใช้ สิ่งอำนวยความสะดวก **SYSLOG** คุณต้องเป็นผู้ใช้ root แก้ไขไฟล์ `/etc/syslog.conf` และเพิ่มรายการสำหรับ ***.debug** ดังนี้:

   ```
   *.debug /usr/adm/imapd.log
   ```

2. ไฟล์ `usr/adm/imapd.log` ต้องมีอยู่ก่อน **syslogd** daemon อ่านไฟล์คอนฟิกูเรชัน `/etc/syslog.conf` เมื่อต้องการสร้างไฟล์นี้ ให้พิมพ์ดังต่อไปนี้ที่พร้อมต์บรรทัดรับคำสั่ง และ กด Enter:

   ```
   touch /usr/adm/imapd.log
   ```

3. รีเฟรช **syslogd** daemon เพื่ออ่านไฟล์คอนฟิกูเรชัน อีกครั้ง พิมพ์ดังต่อไปนี้ที่พร้อมต์บรรทัดรับคำสั่ง และกด Enter:

   ```
   refresh -s syslogd
   ```

คำสั่งการจัดการเมล

คำสั่งการจัดการเมลจะถูกสรุปที่นี่

ไอเท็ม	คำอธิบาย
bugfiler	เก็บรายงานบั๊กในเมลไดเร็กทอรีที่ระบุ
comsat	แจ้งผู้ใช้ของเมลขาเข้า (daemon)
mailq	พิมพ์เนื้อหาของเมลคิว
mailstats	แสดงสถิติเกี่ยวกับปริมาณรับส่งข้อมูลเมล
newaliases	สร้างสำเนาชุดใหม่ของฐานข้อมูล alias จากไฟล์ `/etc/mail/aliases`
rmail	จัดการกับรีโมตเมลที่ได้รับผ่านคำสั่ง **uucp** ของ Basic Networking Utilities (BNU)
sendbug	ส่งรายงานบั๊กของระบบไปยังแอดเดรสที่ระบุ
sendmail	จัดเส้นทางเมลสำหรับการนำส่งในโลคัลหรือเน็ตเวิร์ก
smdemon.cleanu	คลีนอัพคิว **sendmail** สำหรับการดูแลรักษาเป็นระยะ

เมลไฟล์และไดเร็กทอรี

เมลไฟล์และไดเร็กทอรีสามารถถูกจัดเรียงโดยฟังก์ชัน

ไอเท็ม	คำอธิบาย
`/usr/share/lib/Mail.rc`	ตั้งระบบโลคัลดีฟอลต์สำหรับผู้ใช้ทั้งหมดของโปรแกรมเมล เท็กซ์ไฟล์สามารถถูกแก้ไขเพื่อตั้งคุณลักษณะดีฟอลต์ของคำสั่ง **mail**
`$HOME/.mailrc`	ให้ผู้ใช้สามารถเปลี่ยนระบบโลคัลดีฟอลต์สำหรับอำนวยความสะดวกกับเมล
`$HOME/mbox`	เก็บเมลที่ถูกประมวลผลสำหรับแต่ละผู้ใช้
`/usr/bin/Mail`, `/usr/bin/mail` หรือ `/usr/bin/mailx`	ระบุ 3 ชื่อที่ถูกลิงก์กับโปรแกรมเดียวกัน โปรแกรมเมลเป็น*หนึ่ง*ของส่วนติดต่อผู้ใช้กับระบบเมล
`/var/spool/mail`	ระบุไดเร็กทอรีดร็อบเมลดีฟอลต์ โดยดีฟอลต์ เมลทั้งหมดจะถูกนำส่งไปยังไฟล์ `/var/spool/mail/UserName`
`/usr/bin/bellmail`	ทำการนำส่งเมลแบบโลคัล
`/usr/bin/rmail`	ทำการอินเตอร์เฟสรีโมตเมลสำหรับ BNU
`/var/spool/mqueue`	มีล็อกไฟล์และไฟล์ชั่วคราวที่สัมพันธ์กับข้อความ ในคิวเมล

ไอเท็ม	คำอธิบาย
/usr/sbin/sendmail	คำสั่ง **sendmail**
/usr/ucb/mailq	ลิงก์ไปยัง /usr/sbin/sendmail การใช้ mailq จะเหมือนกับการใช้คำสั่ง /usr/sbin/sendmail -bp
/usr/ucb/newaliases	ลิงก์ไปยังไฟล์ /usr/sbin/sendmail การใช้ newaliases จะเหมือนกับการใช้คำสั่ง /usr/sbin/sendmail -bi
/etc/netsvc.conf	ระบุลำดับของเซอร์วิสการแปลงชื่อนั้นๆ
/usr/sbin/mailstats	จัดรูปแบบและพิมพ์สถิติ **sendmail** ที่พบในไฟล์ /etc/sendmail.st ถ้ามันมีอยู่ ไฟล์ /etc/sendmail.st เป็นค่าดีฟอลต์ แต่คุณสามารถระบุไฟล์อื่น
/etc/mail/aliases	อธิบายเท็กซ์เวอร์ชันของไฟล์ alias สำหรับคำสั่ง **sendmail** คุณสามารถแก้ไขไฟล์นี้เพื่อสร้าง แก้ไข หรือลบ alias สำหรับระบบของคุณ
/etc/aliasesDB	อธิบายไดเร็กทอรีที่ประกอบด้วยไฟล์ฐานข้อมูล alias DB.dir และ DB.pag ที่ถูกสร้างจากไฟล์ /etc/mail/aliases เมื่อคุณรันคำสั่ง**sendmail –bi**
/etc/mail/sendmail.cf	ประกอบด้วยข้อมูลการตั้งค่า **sendmail** ในรูปแบบของเท็กซ์ แก้ไขไฟล์เพื่อเปลี่ยนข้อมูลนี้
/usr/lib/smdemon.cleanu	ระบุเซลล์ไฟล์ที่รันคิวของเมลและบำรุงรักษาล็อกไฟล์ **sendmail** ในไดเร็กทอรี /var/spool/mqueue
/etc/mail/statistics	รวบรวมสถิติเกี่ยวกับทราฟิกของเมล ไฟล์จะไม่โตขึ้น ใช้คำสั่ง /usr/sbin/mailstats เพื่อแสดงเนื้อหาของไฟล์นี้ ลบไฟล์นี้ถ้าคุณไม่ต้องการเก็บรวบรวมข้อมูลนี้
/var/spool/mqueue	อธิบายไดเร็กทอรีที่ประกอบด้วยไฟล์ชั่วคราวที่เกี่ยวข้องกับแต่ละข้อความในคิว ไดเร็กทอรีสามารถประกอบด้วยล็อกไฟล์
/var/spool/cron/crontabs	อธิบายไดเร็กทอรีที่ประกอบด้วยไฟล์ที่ **cron** daemon อ่านเพื่อระบุว่างานใดที่จะสตาร์ท ไฟล์ root จะประกอบด้วยบรรทัดเพื่อสตาร์ทเชลล์สคริปต์ smdemon.cleanu

คำสั่ง IMAP และ POP

คำสั่งเมล **imapd** และ **pop3d** ถูกใช้สำหรับ IMAP และ POP

ไอเท็ม	คำอธิบาย
/usr/sbin/**imapd**	กระบวนการของ Internet Message Access Protocol (IMAP) เซิร์ฟเวอร์
/usr/sbin/**pop3d**	กระบวนการของ Post Office Protocol Version 3 (POP3) เซิร์ฟเวอร์

Transmission Control Protocol/Internet Protocol

เมื่อคอมพิวเตอร์สื่อสารกับเครื่องอื่น กฎบางอย่าง หรือ *โปรโตคอล* อนุญาตให้คอมพิวเตอร์ส่งหรือรับข้อมูลตามแบบแผนทั่วโลก หนึ่งในชุดโปรโตคอลที่ใช้กันมากที่สุดคือ **Transmission Control Protocol/Internet Protocol (TCP/IP)** (อย่างไรก็ตาม ส่วนมากในยุโรป ใช้โปรโตคอล X.25) ฟังก์ชันทั่วไปบางอย่างสำหรับการใช้ **TCP/IP** คือ อิเล็กทรอนิกส์เมล, การโอนย้ายไฟล์ระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ และรีโมตล็อกอิน

คำสั่งผู้ใช้ **mail** คำสั่งผู้ใช้ Message Handling (MH) และคำสั่งเซิร์ฟเวอร์ **sendmail** สามารถใช้ **TCP/IP** สำหรับ ส่งและรับเมลระหว่างระบบ และ Basic Networking Utilities (BNU) สามารถใช้ **TCP/IP** สำหรับส่งและรับไฟล์และคำสั่งระหว่างระบบ

TCP/IP คือชุดโปรโตคอลที่ระบุการสื่อสารมาตรฐาน ระหว่างคอมพิวเตอร์ และรายละเอียดข้อตกลงสำหรับกำหนดเส้นทางและการเชื่อมต่อภายในเครือข่าย ซึ่งใช้อย่างแพร่หลายในอินเตอร์เน็ต และต่อมาอนุญาตให้หน่วยงานวิจัย วิทยาลัย และมหาวิทยาลัย รัฐบาล และอุตสาหกรรมติดต่อสื่อสารระหว่างกัน

TCP/IP ช่วยให้การสื่อสารระหว่างกลุ่มคอมพิวเตอร์ (เรียกว่า โฮสต์) เชื่อมต่อเครือข่าย แต่ละเครือข่ายสามารถเชื่อมต่อกับเครือข่ายอื่น เพื่อติดต่อกับโฮสต์บนเครือข่าย แม้ว่ามีเทคโนโลยีเครือข่ายมากมาย หลายเทคโนโลยีทำงานด้วย packet-switching และ stream transport, **TCP/IP** เสนอข้อดีที่สำคัญ: ไม่ขึ้นกับฮาร์ดแวร์

เนื่องจากอินเตอร์เน็ตโปรโตคอลกำหนดยูนิตของการโอนย้าย และระบุวิธีการส่ง **TCP/IP** สามารถซ่อนรายละเอียดของฮาร์ดแวร์เครือข่าย ช่วยให้ เทคโนโลยีเครือข่ายหลายชนิดเชื่อมต่อและแลกเปลี่ยนข้อมูล อินเตอร์เน็ตแอดเดรส ช่วยให้เครื่องบนเครือข่ายสื่อสารกับเครื่องอื่นบนเครือข่าย **TCP/IP** จัดเตรียมมาตรฐานสำหรับหลายๆ เซอร์วิสการสื่อสาร ที่ผู้ใช้ต้องการ

TCP/IP จัดเตรียมโครงสร้างพื้นฐานที่ทำให้ระบบคอมพิวเตอร์เป็น อินเตอร์เน็ตโฮสต์ ซึ่งสามารถติดกับเครือข่ายและสื่อสารกับอินเตอร์เน็ตโฮสต์อื่น **TCP/IP** รวม คำสั่งและโครงสร้างพื้นฐานที่อนุญาตให้คุณ:

- โอนย้ายไฟล์ระหว่างระบบ
- ล็อกอินเข้าสู่ระบบรีโมต
- รันคำสั่งบนระบบรีโมต
- พิมพ์ไฟล์บนระบบรีโมต
- ส่งเมลอิเล็กทรอนิกส์ให้กับผู้ใช้รีโมต
- สนทนาโต้ตอบกับผู้ใช้รีโมต
- จัดการเครือข่าย

> **หมายเหตุ: TCP/IP** จัดเตรียมคุณลักษณะการจัดการเครือข่ายพื้นฐานไว้ให้ **Simple Network Management Protocol (SNMP)** จัดเตรียม คำสั่งและฟังก์ชันการจัดการเครือข่ายเพิ่มเติม

TCP/IP terminology

คุณอาจจะพบว่าเป็นการดีที่จะคุ้นเคยกับประโยคอินเตอร์เน็ต ต่อไปนี้ เนื่องจากประโยคเหล่านี้ถูกใช้สัมพันธ์กับ TCP/IP

ไอเท็ม	คำอธิบาย
ไคลเอ็นต์	คอมพิวเตอร์หรือกระบวนการที่เข้าถึง ข้อมูล เซอร์วิส หรือรีซอร์ส ของคอมพิวเตอร์หรือกระบวนการอื่นบนเน็ตเวิร์ก
โฮสต์	คอมพิวเตอร์ที่ถูกเชื่อมต่อกับอินเตอร์เน็ตเน็ตเวิร์กและสามารถสื่อสาร กับอินเตอร์เน็ตโฮสต์อื่น *โลคัลโฮสต์* สำหรับผู้ใช้คือคอมพิวเตอร์ซึ่งผู้ใช้ทำงานอยู่ *โฮสต์ foreign* คือชื่อโฮสต์ อื่นบนเน็ตเวิร์ก จากมุมมองของเน็ตเวิร์กการสื่อสาร โฮสต์เป็นทั้งต้นทางและปลายทางของแพ็กเก็ต ทุกโฮสต์เป็นได้ไคลเอ็นต์ เซิร์ฟเวอร์ หรือทั้งสองแบบ บนอินเตอร์เน็ตเน็ตเวิร์ก โฮสต์ถูกระบุโดยชื่อและแอดเดรส อินเตอร์เน็ต
เน็ตเวิร์ก	การรวมกันของสองโฮสต์หรือมากกว่านั้น และเชื่อมต่อลิงก์ระหว่าง กัน *ฟิสิคัลเน็ตเวิร์ก* คือฮาร์ดแวร์ที่สร้างเน็ตเวิร์ก *โลจิคัลเน็ตเวิร์ก* คือการการจัดกลุ่มแบบนามธรรมทั้งหมดหรือบางส่วนบนหนึ่งฟิสิคัลเน็ตเวิร์ก หรือมากกว่านั้น อินเตอร์เน็ตเน็ตเวิร์กเป็นตัวอย่างของโลจิคัลเน็ตเวิร์ก อินเตอร์เฟสโปรแกรมจัดการการแปลของการดำเนินการ โลจิคัลเน็ตเวิร์ก ไปเป็นการดำเนินการฟิสิคัลเน็ตเวิร์ก
แพ็กเก็ต	บล็อกของสารสนเทศและข้อมูลควบคุมสำหรับหนึ่ง transaction ระหว่าง โฮสต์และเน็ตเวิร์ก แพ็กเก็ตคือสื่อแลกเปลี่ยนที่ใช้โดยกระบวนการ เพื่อส่งและรับข้อมูลผ่านอินเตอร์เน็ตเน็ตเวิร์ก แพ็กเก็ตถูกส่งจาก *ต้นทาง* ไปที่ *ปลายทาง*
พอร์ต	จุดการเชื่อมต่อโลจิคัลสำหรับกระบวนการ ข้อมูลถูกส่งระหว่าง กระบวนการผ่านพอร์ต (หรือ *ซ็อกเก็ต*) แต่ละพอร์ตจัดเตรียมคิวสำหรับ การส่งและการรับข้อมูล ในอินเตอร์เฟสโปรแกรมเน็ตเวิร์ก แต่ละพอร์ตมี *หมายเลขพอร์ต* อินเตอร์เน็ต ขึ้นกับวิธีที่พอร์ตถูกใช้ พอร์ต ถูกระบุด้วยอินเตอร์เน็ต *ซ็อกเก็ตแอดเดรส*, ซึ่งเป็นการรวมกัน ของอินเตอร์เน็ตโฮสต์แอดเดรสและหมายเลขพอร์ต
กระบวนการ	โปรแกรมที่รันอยู่ กระบวนการคือส่วนประกอบที่แอ็คทีฟในคอมพิวเตอร์ เทอร์มินัล ไฟล์ และอุปกรณ์ I/O อื่นสื่อสารกันผ่าน กระบวนการ ดังนั้น การสื่อสารของเน็ตเวิร์กคือ *การสื่อสาร interprocess* (คือ การสื่อสารระหว่างกระบวนการ)
โปรโตคอล	ชุดกฎสำหรับการจัดการการสื่อสารที่ระดับฟิสิคัลหรือ โลจิคัล โปรโตคอลบ่อยครั้งใช้โปรโตคอลอื่นเพื่อจัดเตรียมเซอร์วิส ตัวอย่าง *connection-level protocol* ใช้ *transport-level protocol* เพื่อ ส่งแพ็กเก็ตที่ดูแลการเชื่อมต่อระหว่างสองโฮสต์
เซิร์ฟเวอร์	คอมพิวเตอร์หรือกระบวนการที่จัดเตรียม ข้อมูล เซอร์วิส หรือรีซอร์สที่ สามารถเข้าถึงได้โดยคอมพิวเตอร์หรือกระบวนการอื่นบนเน็ตเวิร์ก

การวางแผนเน็ตเวิร์ก TCP/IP ของคุณ

เนื่องจาก **TCP/IP** เป็นเครื่องมือการใช้งานเน็ตเวิร์กที่มีความยืดหยุ่น คุณสามารถ กำหนดเครื่องได้เองเพื่อให้เหมาะสมกับองค์กรของคุณ พิจารณา ประเด็นสำคัญในหัวข้อนี้เมื่อวางแผนเน็ตเวิร์กของคุณ รายละเอียดของประเด็น เหล่านี้จะกล่าวถึงในหัวข้ออื่น รายการนี้มุ่งหวังเพียงแนะนำ ให้คุณทราบถึงประเด็นปัญหาต่างๆ

1. ตัดสินใจว่าฮาร์ดแวร์เน็ตเวิร์กชนิดใดที่คุณต้องการใช้: โทเค็นริง, Ethernet เวอร์ชัน 2, IEEE 802.3, Fiber Distributed Data Interface (FDDI), Serial Optical Channel (SOC) หรือ Serial Line Interface Protocol (SLIP)
2. วางแผนโครงร่างฟิสิคัลของเน็ตเวิร์ก พิจารณาว่าหน้าที่ใด ที่โฮสต์แต่ละเครื่องจะให้บริการ ตัวอย่างเช่น คุณต้องตัดสินใจว่าเครื่อง ใดบ้างที่จะทำหน้าที่เป็นเกตเวย์ก่อนที่คุณจะวางสายสายเคเบิลเชื่อมต่อเน็ตเวิร์ก
3. ตัดสินใจว่าโครงสร้างเน็ตเวิร์กแบบ *ราบ* หรือแบบ *ลำดับชั้น* ที่ตรงตามความต้องการของคุณมากที่สุด

   ถ้าเน็ตเวิร์กของคุณมีขนาดค่อนข้าง เล็ก ตั้งอยู่ที่ไซต์เดียว และให้ความสำคัญเรื่องเน็ตเวิร์กฟิสิคัล ดังนั้นเน็ตเวิร์ก แบบราบน่าจะเป็นทางเลือกที่เหมาะกับความต้องการของคุณ ถ้าเน็ตเวิร์กของคุณมีขนาดใหญ่มาก หรือมีความ ซับซ้อนโดยมีหลายไซต์ หรือมีหลายเน็ตเวิร์กฟิสิคัล การใช้เน็ตเวิร์กแบบลำดับชั้น จะมีประสิทธิภาพต่อโครงสร้างองค์กรเน็ตเวิร์กสำหรับคุณมากกว่า
4. ถ้าเน็ตเวิร์กของคุณจะต้องเชื่อมต่อกับเน็ตเวิร์กอื่น คุณต้อง วางแผนว่าจะตั้งค่าและกำหนดค่าเกตเวย์ของคุณอย่างไร สิ่งที่ต้องพิจารณาได้แก่:
   a. ตัดสินใจว่าเครื่องใดบ้างที่จะทำหน้าที่เป็นเกตเวย์
   b. ตัดสินใจว่าคุณต้องใช้การจัดเส้นทางแบบคงที่หรือแบบไดนามิก หรือใช้รวมกัน ทั้งสองแบบ ถ้าคุณเลือกการจัดเส้นทางแบบไดนามิก ให้ตัดสินใจว่า daemons การจัดเส้นทางใด ที่แต่ละเกตเวย์จะใช้เนื่องจากชนิดของโปรโตคอลสื่อสารที่คุณจำเป็นต้องให้ การสนับสนุน
5. ตัดสินใจเกี่ยวกับรูปแบบการกำหนดแอดเดรส

   ถ้าเน็ตเวิร์กของคุณ จะไม่เป็นส่วนหนึ่งของอินเตอร์เน็ตเวิร์กขนาดใญ่ ให้เลือกรูปแบบการกำหนดแอดเดรสที่ เหมาะกับความต้องการของคุณมากที่สุด ถ้าคุณต้องการให้เน็ตเวิร์กของคุณเชื่อมต่อกับอินเตอร์เน็ตเวิร์กขนาดใหญ่ เช่น อินเตอร์เน็ต คุณจะต้องมีชุดแอดเดรสที่เป็นค่าทางการ จาก internet service provider (ISP) ของคุณ
6. ตัดสินใจว่าระบบของคุณจำเป็นต้องแบ่งออกเป็นเน็ตย่อยหรือไม่ ถ้า จำเป็น ให้ตัดสินใจว่าคุณจะกำหนด subnet masks อย่างไร
7. ตัดสินใจเกี่ยวกับรูปแบบการตั้งชื่อ แต่ละเครื่องบนเน็ตเวิร์กจำเป็นต้อง มีชื่อโฮสต์เฉพาะของตนเอง
8. ตัดสินใจว่าเน็ตเวิร์กของคุณจำเป็นต้องใช้เนมเซิร์ฟเวอร์สำหรับการระบุชื่อหรือไม่ หรือถ้าใช้ไฟล์ `/etc/hosts` จะ เพียงพอต่อความต้องการหรือไม่

   ถ้าคุณเลือกใช้เนมเซิร์ฟเวอร์ ให้พิจารณา ชนิดเนมเซิร์ฟเวอร์ที่คุณจำเป็นต้องใช้ และจำนวนที่คุณจะให้บริการเน็ตเวิร์กของคุณ ได้อย่างมีประสิทธิภาพ
9. ตัดสินใจชนิดบริการที่คุณต้องการให้เน็ตเวิร์กของคุณมีสำหรับ ผู้ใช้รีโมต ตัวอย่างเช่น บริการเมล บริการการพิมพ์ การใช้ไฟล์ร่วมกัน การล็อกอิน แบบรีโมต การเรียกใช้คำสั่งแบบรีโมต และอื่นๆ

การติดตั้ง TCP/IP

ส่วนนี้จะอธิบายถึงการติดตั้ง **Transmission Control Protocol/Internet Protocol (TCP/IP)**

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการติดตั้ง **Transmission Control Protocol/Internet Protocol (TCP/IP)** โปรดดู การติดตั้งและการโอนย้าย

คอนฟิกูเรชันของ TCP/IP

หลังจากติดตั้งซอฟต์แวร์ **TCP/IP** บนระบบแล้ว คุณพร้อม ที่จะเริ่มต้นการกำหนดคอนฟิกระบบ

ภารกิจคอนฟิกูเรชัน **TCP/IP** จำนวนมากสามารถทำได้มากกว่า หนึ่งวิธี โดย:

- การใช้ System Management Interface Tool (SMIT)
- การแก้ไขรูปแบบไฟล์
- การออกใช้คำสั่งที่เชลล์พร้อมต์

ตัวอย่างเช่น เชลล์สคริปต์ `rc.net` ทำคอนฟิกูเรชันโฮสต์ต่ำสุด ที่จำเป็นสำหรับ **TCP/IP** ในระหว่างโปรเซสสตาร์ทอัพระบบ (สคริปต์ `rc.net` มีการรันโดยโปรแกรมตัวจัดการคอนฟิกูเรชัน ในระหว่างระยะที่สองของการบูต) โดยใช้ SMIT เพื่อดำเนินการกำหนดคอนฟิกโฮสต์ ไฟล์ `rc.net` จะถูกกำหนดคอนฟิกโดยอัตโนมัติ

หรือคุณสามารถกำหนดคอนฟิกไฟล์ `/etc/rc.bsdnet` โดยใช้โปรแกรมแก้ไขข้อความมาตรฐาน ด้วยเมธอดนี้ คุณสามารถระบุคำสั่งคอนฟิกูเรชัน UNIX **TCP/IP** แบบที่ใช้กันมา เช่น **ifconfig, hostname,** และ **route** หากใช้เมธอดการแก้ไขไฟล์ คุณต้องป้อนพาธด่วน `smit configtcp` แล้วเลือก **BSD Style rc Configuration** โปรดดูที่ รายการของการอ้างอิงการเขียนโปรแกรม TCP/IP ใน *หลักการเขียนโปรแกรมการสื่อสาร* สำหรับ ข้อมูลเกี่ยวกับไฟล์ **TCP/IP** และรูปแบบไฟล์

งานบางอย่าง เช่นการกำหนดคอนฟิกเนมเซิร์ฟเวอร์ ไม่สามารถทำได้โดยใช้ SMIT

คอนฟิกูเรชันโฮสต์

ต้องกำหนดคอนฟิกโฮสต์แต่ละเครื่องบนเครือข่ายเพื่อให้ทำงาน ตามความต้องการของผู้ใช้ขั้นปลายและเครือข่ายโดยรวม

สำหรับแต่ละโฮสต์บนเครือข่าย คุณต้องกำหนดคอนฟิกอินเตอร์เฟสเครือข่าย ตั้งค่าอินเทอร์เน็ตแอดเดรส และตั้งค่าชื่อโฮสต์ คุณยังต้องตั้งค่าสแตติก เราต์ไปยังเกตเวย์หรือโฮสต์อื่น ระบุ daemons ที่จะเริ่มต้นโดยค่าดีฟอลต์ และตั้งค่าไฟล์ `/etc/hosts` สำหรับการแก้ไข ชื่อด้วย (หรือตั้งค่าโฮสต์ที่จะใช้เนมเซิร์ฟเวอร์สำหรับการแก้ไขชื่อ)

คอนฟิกูเรชันโฮสต์เป็นเซิร์ฟเวอร์

หากเครื่องโฮสต์จะมีฟังก์ชันเฉพาะ เช่น ทำหน้าที่เป็น เกตเวย์ เซิร์ฟเวอร์ไฟล์ หรือเนมเซิร์ฟเวอร์ คุณต้องทำภารกิจคอนฟิกูเรชันที่จำเป็น หลังจากคอนฟิกูเรชันพื้นฐานเสร็จสมบูรณ์แล้ว

ตัวอย่างเช่น ถ้าเครือข่ายมีการจัดระเบียบตามลำดับชั้นและคุณต้องการ ใช้โปรโตคอล **Domain Name** เพื่อแก้ไขชื่อเป็น อินเทอร์เน็ตแอดเดรส คุณจะต้องกำหนดคอนฟิกอย่างน้อยหนึ่งเนมเซิร์ฟเวอร์เพื่อ นำเสนอฟังก์ชันนี้สำหรับเครือข่าย

โปรดจำไว้ว่า เซิร์ฟเวอร์โฮสต์ไม่จำเป็นต้องเป็นเครื่องเฉพาะงาน และสามารถ ใช้สำหรับสิ่งอื่นได้ด้วย หากฟังก์ชันเนมเซิร์ฟเวอร์สำหรับเครือข่าย ของคุณค่อนข้างเล็ก ยังอาจใช้เครื่องเป็นเวิร์กสเตชันหรือเป็นเซิร์ฟเวอร์ไฟล์ สำหรับเครือข่ายได้ด้วย

หมายเหตุ: ถ้าระบบของคุณติดตั้ง NIS ไว้แล้ว เซอร์วิสเหล่านี้ยังสามารถจัดเตรียมการแก้ไขชื่อ

คอนฟิกูเรชันเกตเวย์

หากเครือข่ายของคุณจะสื่อสารกับเครือข่ายอื่น คุณ จะต้องกำหนดคอนฟิกเกตเวย์โฮสต์อย่างน้อยหนึ่งรายการ

คุณต้องพิจารณาโปรโตคอลการสื่อสารซึ่งคุณต้องการสนับสนุน จากนั้น ใช้ daemon (**routed** หรือ **gated** daemon) การเราต์ที่สนับสนุนโปรโตคอลดังกล่าว

คำสั่งคอนฟิกูเรชันและการจัดการ TCP/IP

คุณสามารถใช้คำสั่งได้หลายคำสั่งเพื่อกำหนดคอนฟิกและจัดการกับเครือข่าย **TCP/IP** ซึ่งอธิบายไว้ในตารางนี้

ไอเท็ม	คำอธิบาย
arp	แสดงหรือเปลี่ยนอินเทอร์เน็ตแอดเดรสในตารางการแปลฮาร์ดแวร์ แอดเดรสที่ใช้โดยโปรโตคอล **Address Resolution**
finger	ส่งคืนข้อมูลเกี่ยวกับผู้ใช้บนโฮสต์ที่ระบุ
host	แสดงอินเทอร์เน็ตแอดเดรสของโฮสต์ที่ระบุ หรือชื่อโฮสต์ของ อินเทอร์เน็ตแอดเดรสที่ระบุ
hostname	แสดงหรือตั้งค่าชื่ออินเทอร์เน็ตและแอดเดรสของโลคัลโฮสต์
ifconfig	กำหนดคอนฟิกอินเตอร์เฟสเครือข่ายและลักษณะ
netstat	แสดงโลคัลแอดเดรสและแอดเดรสต่างประเทศ ตารางการเราต์ สถิติฮาร์ดแวร์ และข้อมูลสรุปของแพ็กเก็ตที่โอนย้าย
no	ตั้งค่าหรือแสดงอ็อพชันเคอร์นัลเครือข่ายปัจจุบัน
ping	กำหนดว่าโฮสต์สามารถเข้าถึงได้หรือไม่
route	อนุญาตให้คุณจัดการตารางการเราต์ด้วยตนเอง
ruptime	แสดงข้อมูลสถานะบนโฮสต์ที่เชื่อมต่อกับเครือข่ายโลคัลฟิสิคัล และกำลังรันเซิร์ฟเวอร์ **rwhod**
rwho	แสดงข้อมูลสถานะของผู้ใช้บนโฮสต์ที่เชื่อมต่อกับเครือข่ายโลคัลฟิสิคัล และกำลังรันเซิร์ฟเวอร์ **rwhod**
setclock	อ่านเซอร์วิสเวลาเครือข่าย และตั้งค่าวันที่และเวลาของ โลคัลโฮสต์ตามลำดับ
timedc	ส่งคืนข้อมูลเกี่ยวกับ **timed** daemon
trpt	รายงานการติดตามโปรโตคอลบนซ็อกเก็ต TCP
whois	แสดงเซอร์วิสไดเร็กทอรีชื่ออินเทอร์เน็ต

การกำหนดค่าเน็ตเวิร์ก TCP/IP

ใช้โพรซีเดอร์นี้เป็นแนวทางสำหรับการกำหนดค่าเน็ตเวิร์กของคุณ ตรวจสอบ ให้แน่ใจว่าคุณได้อ่าน และทำความเข้าใจเอกสารที่เหมาะสม

ก่อนเริ่มทำงานกระบวนการนี้ ตรวจสอบให้แน่ใจว่าสิ่งที่จำเป็นต้องมีต่อไปนี้ เป็นจริง:

1. ฮาร์ดแวร์เน็ตเวิร์กถูกติดตั้งและต่อสายแล้ว สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการติดตั้ง และการวางสายเคเบิลฮาร์ดแวร์ของคุณ ดูที่ “การ์ดอะแด็ปเตอร์เครือข่ายพื้นที่โลคัล TCP/IP” ในหน้า 173
2. ติดตั้งซอฟต์แวร์ **TCP/IP** แล้ว สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการติดตั้ง ดูที่ *การติดตั้งและการย้าย*

หลังจากคุณทำให้เน็ตเวิร์กของคุณทำงาน และกำลังรันอย่างเหมาะสม คุณ อาจพบว่าเป็นประโยชน์ในการอ้างอิงรายการตรวจสอบนี้เพื่อวัตถุประสงค์การดีบัก

เมื่อ ต้องการกำหนดค่าเน็ตเวิร์ก **TCP/IP** ของคุณให้ใช้ขั้นตอนต่อไปนี้:

1. อ่าน “โปรโตคอล TCP/IP” ในหน้า 131 เกี่ยวกับ โครงสร้างการทำงานพื้นฐานของ **TCP/IP** คุณควรทำความเข้าใจ:
   - ลักษณะธรรมชาติแบบเลเยอร์ของ **TCP/IP** (นั่นคือ โปรโตคอลต่างกันอยู่ บนเลเยอร์ที่ต่างกัน)
   - วิธีการที่ข้อมูลไหลจากผ่านเลเยอร์ต่างๆ
2. อย่างน้อยที่สุดต้องกำหนดค่าโฮสต์แต่ละเครื่องบนเน็ตเวิร์ก นี้ หมายความว่าต้องเพิ่มเน็ตเวิร์กอะแด็ปเตอร์ การกำหนด IP แอดเดรสและการกำหนด ชื่อโฮสต์ให้แก่แต่ละโฮสต์ รวมถึงการกำหนดเส้นทางดีฟอลต์ไปยังเน็ตเวิร์กของคุณ สำหรับข้อมูลเบื้องหลังเกี่ยวกับงานเหล่านี้ อ้างอิงที่ “อินเตอร์เฟสเครือข่าย TCP/IP” ในหน้า 184, “การกำหนดแอดเดรส TCP/IP” ในหน้า 193 และ “การตั้งชื่อโฮสต์บนเน็ตเวิร์กของคุณ” ในหน้า 202

   หมายเหตุ: แต่ละ เครื่องบนเน็ตเวิร์กจำเป็นต้องใช้การกำหนดค่าพื้นฐานนี้ ไม่ว่าจะเป็น โฮสต์ของผู้ใช้ขั้นปลาย ไฟล์เซิร์ฟเวอร์ เกตเวย์ หรือเนมเซิร์ฟเวอร์
3. กำหนดค่าและเริ่มทำงาน **inetd** daemon บนโฮสต์ แต่ละเครื่องบนเน็ตเวิร์ก อ่าน “TCP/IP daemons” ในหน้า 376 และจากนั้นปฏิบัติตามคำแนะนำใน “การกำหนดคอนฟิก inetd daemon” ในหน้า 377

4. กำหนดค่าโฮสต์แต่ละเครื่องเพื่อให้ดำเนินการระบุชื่อโลคัล หรือเพื่อใช้เซิร์ฟเวอร์ใช้ ถ้าคุณกำลังตั้งค่าเน็ตเวิร์ก Domain Name แบบลำดับชั้น ให้กำหนดค่าอย่างน้อยหนึ่งโฮสต์เพื่อทำให้หน้าที่เป็นเนมเซิร์ฟเวอร์ อ่านและปฏิบัติตามคำสั่งใน "การระบุชื่อ" ในหน้า 204
5. ถ้าเน็ตเวิร์กของคุณจะสื่อสารกับเน็ตเวิร์กรีโมตใดๆ ให้กำหนดค่า อย่างน้อยหนึ่งโฮสต์เพื่อให้ทำหน้าที่เป็นเกตเวย์ เกตเวย์สามารถใช้ เส้นทางคงที่ หรือ daemon การจัดเส้นทางเพื่อดำเนินการจัดเส้นทางระหว่างเน็ตเวิร์ก อ่านและ ปฏิบัติตามคำสั่งใน "การจัดเส้นทาง TCP/IP" ในหน้า 378
6. เลือกว่าบริการใดที่โฮสต์แต่ละเครื่องบนเน็ตเวิร์กจะใช้โดยค่าดีฟอลต์ บริการทั้งหมดจะพร้อมใช้งานได้ ปฏิบัติตามคำแนะนำใน "เซอร์วิสเครือข่ายไคลเอ็นต์" ในหน้า 377 ถ้าคุณ ต้องการทำให้บริการเฉพาะบางอย่างไม่สามารถทำงานได้
7. ตัดสินใจว่าโฮสต์ใดบนเน็ตเวิร์กที่จะใช้เป็นเซิร์ฟเวอร์ และบริการใด จะเซิร์ฟเวอร์เฉพาะนั้นจะให้บริการ ปฏิบัติตามคำแนะนำใน "เซอร์วิสเครือข่ายเซิร์ฟเวอร์" ในหน้า 378 เพื่อเริ่มทำงาน daemons เซิร์ฟเวอร์ที่คุณต้องการรัน
8. กำหนดค่าเซิร์ฟเวอร์การพิมพ์รีโมตที่คุณจะต้องใช้ ดูที่ Printing administration ใน *เครื่องพิมพ์และการพิมพ์* เพื่อ ดูข้อมูลเพิ่มเติม
9. **ทางเลือก:** ถ้าต้องการ ให้กำหนดค่าโฮสต์เพื่อใช้ หรือเพื่อให้บริการ เป็นเซิร์ฟเวอร์เวลามาสเตอร์สำหรับเน็ตเวิร์ก ดูที่ **timed** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 5* เพื่อดูข้อมูลเพิ่มเติม

การพิสูจน์ตัวตนและ rcmds ที่ปลอดภัย

มีการปรับปรุงคำสั่งเหล่านี้เพื่อนำเสนอเมธอดการพิสูจน์ตัวตนเพิ่มเติม จากที่ใช้ในทุกวันนี้

Rcmds ที่ปลอดภัยคือ **rlogin, rcp, rsh, telnet,** และ **ftp** โดยดีฟอลต์ คำสั่งเหล่านี้ใช้เมธอด *มาตรฐาน AIX* ของการพิสูจน์ตัวตน เมธอดเพิ่มเติมสองเมธอดคือ Kerberos V.5 และ Kerberos V.4

เมื่อใช้เมธอดการพิสูจน์ตัวตน Kerberos V.5 ไคลเอ็นต์จะได้รับตั๋ว Kerberos V.5 จากเซิร์ฟเวอร์ความปลอดภัย DCE หรือเซิร์ฟเวอร์ Native Kerberos ตั๋ว เป็นส่วนหนึ่งของหลักฐาน DCE หรือ Native ปัจจุบันที่เข้ารหัสของผู้ใช้สำหรับ เซิร์ฟเวอร์ **TCP/IP** ซึ่งต้องการเชื่อมต่อ Daemon บนเซิร์ฟเวอร์ **TCP/IP** ถอดรหัสตั๋ว ซึ่งช่วยให้เซิร์ฟเวอร์ **TCP/IP** สามารถระบุผู้ใช้ได้โดยสัมบูรณ์ หากหลักการ DCE หรือ Native ที่อธิบายไว้ในตั๋วได้รับอนุญาตให้เข้าถึง แอคเคาต์ของผู้ใช้ระบบปฏิบัติการ การเชื่อมต่อจะดำเนินต่อไป

> **หมายเหตุ:** เริ่มต้นด้วย DCE เวอร์ชัน 2.2 เซิร์ฟเวอร์ความปลอดภัย DCE สามารถส่งคืนตั๋ว Kerberos V.5 ได้ Rcmds ที่ปลอดภัย ในระบบปฏิบัติการ AIX จะใช้ไลบรารี Kerberos V.5 และไลบรารี GSSAPI ที่จัดให้โดย NAS (Network Authentication Service) เวอร์ชัน 1.3

นอกจากการพิสูจน์ตัวตนไคลเอ็นต์แล้ว Kerberos V.5 จะส่งต่อหลักฐานของ ผู้ใช้ปัจจุบันไปยังเซิร์ฟเวอร์ **TCP/IP** หากหลักฐานมีการทำเครื่องหมายว่า ส่งต่อได้ ไคลเอ็นต์จะส่งหลักฐานไปยังเซิร์ฟเวอร์เป็น Kerberos TGT (Ticket Granting Ticket) บนด้านเซิร์ฟเวอร์ **TCP/IP** ถ้ากำลังสื่อสารกับ เซิร์ฟเวอร์ความปลอดภัย DCE daemon จะอัพเกรด TGT เป็นหลักฐาน DCE เต็มรูปแบบ โดยใช้คำสั่ง **k5dcecreds**

คำสั่ง **ftp** ใช้เมธอดการพิสูจน์ตัวตนที่แตกต่างจาก คำสั่งอื่น คำสั่งนี้ใช้กลไกการรักษาความปลอดภัย GSSAPI เพื่อส่งผ่านการพิสูจน์ตัวตนระหว่างคำสั่ง **ftp** และ **ftpd** daemon โดยใช้คำสั่งย่อย **clear/safe/private** ไคลเอ็นต์ **ftp** สนับสนุนการเข้ารหัสข้อมูล

ระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ระบบปฏิบัติการ **ftp** ได้รับ การพัฒนาขึ้นเพื่อให้สามารถโอนย้ายข้อมูลหลายไบต์สำหรับการเชื่อมต่อข้อมูลที่เข้ารหัสได้ มาตรฐานกำหนดการโอนย้ายไบต์เดียวเท่านั้นสำหรับการเชื่อมต่อข้อมูลที่เข้ารหัส เมื่อเชื่อมต่อกับเครื่องของบุคคลที่สามและใช้การเข้ารหัสข้อมูล **ftp** จะใช้ขีดจำกัดการโอนย้ายข้อมูลไบต์เดียว

หมายเหตุ: The **rlogin**, **rsh**, and **telnet** secure rcmds commands, along with the **klogin** and **kshell** Kerberos V.5 authentication methods, allow three attempts before the connection to the remote host is closed.

การกำหนดค่าระบบสำหรับ secure rcmds

สำหรับ secure rcmds ทั้งหมด จะมีวิธีการกำหนดค่า ระดับระบบ เพื่อใช้พิจารณาว่าวิธีการพิสูจน์ตัวตนแบบใดที่ได้รับอนุญาตให้กระทำได้สำหรับระบบนั้น การตั้งค่าจะทำหน้าที่ควบคุมการเชื่อมต่อทั้งขาออกและขาเข้า

คอนฟิกูเรชันการพิสูจน์ตัวตนประกอบด้วยไลบรารี `libauthm.a` และสองคำสั่ง **lsauthent** และ **chauthent** ที่ให้การเข้าถึงจากบรรทัดรับคำสั่งไปยังสองรูทีนของไลบรารีคือ: **get_auth_methods** และ **set_auth_methods**

ระบบสนับสนุนวิธีการพิสูจน์ตัวตนที่ต่างกันสามวิธี: Kerberos V.5, Kerberos V.4 และ *Standard AIX* วิธีการพิสูจน์ตัวตนเป็นตัวกำหนดวิธีที่ใช้ในการพิสูจน์ตัวตน การเข้าถึงเน็ตเวิร์กของผู้ใช้

- Kerberos V.5 เป็นวิธีการที่ใช้ทั่วไปมากสุด เนื่องจากเป็นวิธีพื้นฐานสำหรับ Distributed Computing Environment (DCE) ระบบปฏิบัติการจะอัพเกรด Kerberos V.5 tickets ขาเข้าให้เป็น DCE credentials แบบเต็ม หรือใช้ Native Kerberos V.5 tickets ขาเข้า
- Kerberos V.4 ถูกใช้โดยคำสั่งสองคำสั่งของ secure rcmds เท่านั้น: **rsh** และ **rcp** มีการจัดให้เพื่อสนับสนุนความเข้ากันได้กับเวอร์ชันก่อนหน้าบนระบบ SP และ สามารถทำงานบนหนึ่งระบบเท่านั้น Kerberos V.4 ticket ไม่ถูกอัพเกรดเป็น DCE credentials
- คำว่าเมธอด *การพิสูจน์ตัวตน AIX* มาตรฐาน หมายถึงเมธอดการพิสูจน์ตัวตนที่ใช้โดยระบบปฏิบัติการ AIX

โดยจะมีการนำไปปฏิบัติสำรองเมื่อมีวิธีการพิสูจน์ตัวตนมากกว่าหนึ่งวิธี ถูกกำหนดค่า ถ้าเมธอดแรกไม่สามารถเชื่อมต่อได้ ไคลเอ็นต์ จะพยายามพิสูจน์ตัวตนโดยใช้เมธอดการพิสูจน์ตัวตนถัดไปที่ กำหนดคอนฟิกไว้

วิธีการพิสูจน์ตัวตนสามารถตั้งค่าลำดับใดๆ ก็ได้ มีข้อยกเว้น อย่างเดียวคือ *AIX* มาตรฐานต้อง เป็นเมธอดการพิสูจน์ตัวตนสุดท้ายที่กำหนดคอนฟิกไว้ เนื่องจากไม่มีอ็อพชัน สำรองสำหรับใช้ ถ้าเมธอดการพิสูจน์ตัวตน *AIX* มาตรฐานไม่ได้ กำหนดคอนฟิกไว้ ระบบจะไม่พยายามพิสูจน์ตัวตนรหัสผ่าน และ ความพยายามเชื่อมต่อใดๆ ที่ใช้เมธอดนี้จะถูกปฏิเสธ

โดยสามารถตั้งค่าระบบได้โดยไม่ต้องใช้วิธีการพิสูจน์ตัวตนใดๆ ในกรณีนี้ ระบบปฏิเสธการเชื่อมต่อทั้งหมดที่มาจาก หรือไปยัง เทอร์มินัลใดๆ ที่ใช้ secure rcmds นอกจากนี้ เนื่องจาก Kerberos V.4 ได้รับการสนับสนุนด้วยคำสั่ง **rsh** และ **rcp** เท่านั้น ระบบที่กำหนดคอนฟิกเพื่อใช้เฉพาะ Kerberos V.4 จึงไม่อนุญาต การเชื่อมต่อที่ใช้ **telnet**, **ftp** หรือ **rlogin**

ข้อมูลที่เกี่ยวข้อง:

รูทีนย่อย get_auth_method

รูทีนย่อย set_auth_method

คำสั่ง lsauthent

คำสั่ง chauthent

การตรวจสอบความถูกต้องผู้ใช้ Kerberos V.5 สำหรับ secure rcmds

เมื่อใช้วิธีการพิสูจน์ตัวตน Kerberos V.5 ไคลเอ็นต์ **TCP/IP** จะได้รับ ticket การบริการที่เข้ารหัสสำหรับเซิร์ฟเวอร์ **TCP/IP** เมื่อเซิร์ฟเวอร์ ถอดรหัส ticket จะมีวิธีการระบุผู้ใช้อย่างปลอดภัย (โดย DCE หรือ Native principal)

อย่างไรก็ตาม ยังจำเป็นต้องพิจารณาว่า DCE หรือ Native principal นี้ ที่ได้รับอนุญาตให้เข้าถึงบัญชีผู้ใช้โลคัล การแม็พ DCE หรือ Native principal กับ บัญชีผู้ใช้ระบบปฏิบัติการโลคัลจะได้รับการจัดการโดยไลบรารีที่แบ่งใช้ `libvaliduser.a` ซึ่งมีรูทีนย่อยเดียวคือ **kvalid_user** ถ้าต้องการใช้วิธีการแม็พแบบอื่น ผู้ดูแลระบบต้องให้ทางลือก สำหรับไลบรารี `libvaliduser.a`

การกำหนดค่า DCE สำหรับ secure rcmds

ในการใช้ rcmds ที่ปลอดภัย ต้องมีสอง DCE principals อยู่ สำหรับทุกอินเตอร์เฟสเน็ตเวิร์กที่สามารถใช้เชื่อมต่อได้

สองพูลคือ:

```
host/FullInterfaceName
ftp/FullInterfaceName
```

โดยที่ *FullInterfaceName* คือชื่ออินเตอร์เฟส และโดเมนเนมสำหรับ *HostName.DomainName* หลัก

การกำหนดค่าดั้งเดิมสำหรับ secure rcmds

เมื่อต้องการใช้ secure rcmds ต้องมีหลักปฏิบัติสองข้อสำหรับทุกเน็ตเวิร์ก อินเตอร์เฟสที่สามารถเชื่อมต่อได้

สองพูลคือ:

```
host/FullInterfaceName@Realmname
ftp/FullInterfaceName@Realmname
```

โดยที่ *FullInterfaceName* คือ ชื่ออินเตอร์เฟส และโดเมนเนมสำหรับ `HostName.DomainName` หลัก *Realmname* คือชื่อของ Native Kerberos V realm

การกำหนดลักษณะเฉพาะของ TCP/IP

เมื่อต้องการลักษณะเฉพาะของ **TCP/IP** ให้สร้างไฟล์ `.netrc`

ไฟล์ `.netrc` ระบุข้อมูลการล็อกอินอัตโนมัติ สำหรับคำสั่ง **ftp** และ **rexec** คุณสามารถเขียนแมโคร **ftp** ใหม่ ซึ่งกำหนด ในไฟล์ `$HOME/.netrc` เมื่อต้องการกำหนดลักษณะเฉพาะฟังก์ชันคีย หรือลำดับ ให้สร้างและแก้ไขไฟล์ `$HOME/.3270keys` นอกจากนี้ ไฟล์ `.k5login` ระบุ DCE แรก ที่เซลล์อนุญาตให้เข้าถึงแอคเคาต์ของผู้ใช้

การสร้างไฟล์ .netrc

ขั้นตอนเหล่านี้อธิบายวิธีสร้างและแก้ไขไฟล์ `$HOME/.netrc`:

1. คุณต้องมีสำเนาของไฟล์ `/usr/samples/tcpip/netrc`
2. คำสั่ง **securetcpip** ต้องไม่รันอยู่บนระบบ ของคุณ

เมื่อต้องการสร้างไฟล์ `.netrc`:

1. คัดลอกไฟล์ `/usr/samples/tcpip/netrc` ไปยังไดเร็กทอรี `$HOME` ของคุณโดยพิมพ์ คำสั่งต่อไปนี้:

   ```
   cp /usr/samples/tcpip/netrc $HOME
   ```

2. แก้ไขไฟล์ `$HOME/netrc` เพื่อระบุตัวแปร *HostName*, *LoginName* และ *Password* ที่เหมาะสม ตัวอย่าง เช่น:

```
machine host1.austin.century.com login fred password bluebonnet
```

3. เมื่อต้องการตั้งค่าสิทธิบนไฟล์ $HOME/netrc เป็น 600 โดยใช้คำสั่ง **chmod** ที่พร้อมต์บรรทัดคำสั่ง ($) ให้พิมพ์:

```
chmod 600 $HOME/.netrc
```

4. เปลี่ยนชื่อไฟล์ $HOME/netrc เป็นไฟล์ $HOME/.netrc จุดเริ่มต้น (.) เป็นสาเหตุทำให้ไฟล์ถูกซ่อน

```
mv $HOME/netrc $HOME/.netrc
```

ไฟล์ $HOME/.netrc สามารถมีหลาย นิยามการล็อกอิน และได้สูงสุด 16 แมโครต่อหนึ่งนิยามการล็อกอิน

การเขียนแมโคร ftp

ขั้นตอนเหล่านี้อธิบายวิธีการสร้างแมโคร **ftp**

คุณต้องสร้างไฟล์ $HOME/.netrc

เมื่อต้องการเขียนแมโคร **ftp**:

1. แก้ไขไฟล์ $HOME/.netrc เพื่อรวมคำสั่งต่อไปนี้:

```
macdef init
put schedule
```

ให้แน่ใจว่าแทรกบรรทัดว่างไว้ที่ ส่วนล่างของแมโคร **ftp** บรรทัดว่างจะสิ้นสุดแมโคร **ftp** ในตัวอย่างข้างต้น คำสั่งย่อย **macdef** กำหนดแมโครคำสั่งย่อย init บรรทัดต่อไปนี้คือ คำสั่งระบุแมโคร ในกรณีนี้ put *schedule*, โดย *schedule* คือ ชื่อไฟล์

2. หลังจากที่คุณสร้างแมโคร **ftp** ที่บรรทัดรับคำสั่ง ให้พิมพ์:

```
ftp hostname
```

โดย *hostname* คือ ชื่อของโฮสต์ทีคุณกำลังเชื่อมต่อ **ftp** จะสแกน ไฟล์ $HOME/.netrc สำหรับนิยามล็อกอินที่ตรงกับ ชื่อโฮสต์ของคุณ และใช้นิยามล็อกอินเพื่อเข้าสู่ระบบ

3. หลังจากเข้าสู่ระบบ ที่พร้อมของบรรทัดรับคำสั่ง ให้พิมพ์:

```
ftp init
```

ในตัวอย่างนี้ **ftp** สแกนหาแมโครชื่อ init และเรียกใช้ คำสั่ง หรือหลายคำสั่งที่ระบุในแมโคร

แมโคร **ftp** เชื่อมโยงกับรายการล็อกอินก่อนหน้าทันที แมโคร **ftp** ไม่ใช่แบบโกลบอลกับไฟล์ $HOME/.netrc แมโคร init ถูก เรียกใช้อัตโนมัติเมื่อล็อกอิน แมโครอื่นสามารถเรียกใช้จากพร้อมต์ **ftp** (ftp>) โดยพิมพ์ต่อไปนี้:

```
$getit
```

ในตัวอย่างนี้ $ เรียกใช้ **ftp** macro **getit**

การเปลี่ยนแปลงการกำหนดค่าของชุดคีย์

เมื่อกำหนดค่า **TCP/IP** คุณสามารถใช้โพรซีเดอร์นี้เพื่อเปลี่ยนแปลง ฟังก์ชันและลำดับคีย์

1. คุณต้องมีความรู้เกี่ยวกับการใช้งานเอดิเตอร์ **vi**
2. เอดิเตอร์ vi ต้องอยู่บนระบบของคุณ

ขั้นตอนต่อไปนี้อธิบายวิธีสร้างและแก้ไขไฟล์ $HOME/.3270keys เพื่อกำหนดฟังก์ชันหรือลำดับคีย์เอง:

1. คัดลอกไฟล์ /etc/3270.keys ไปที่ไดเร็กทอรี $HOME และเปลี่ยนชื่อไฟล์ .3270keys โดยใช้คำสั่ง ต่อไปนี้:

```
cp /etc/3270.keys $HOME/.3270keys
```

2. เปลี่ยนคำสั่ง bind ในไฟล์ $HOME/.3270keys เพื่อเปลี่ยนการกำหนดค่าของชุดคีย์โดยใช้ขั้นตอนต่อไปนี้:
   a. เริ่มทำงานเอดิเตอร์ vi บนไฟล์ใหม่และเข้าสู่โหมดแก้ไข
   b. กดลำดับคีย์ Ctrl-V จากนั้นคีย์ที่คุณต้องการ แม็พ คำสั่งนี้แสดงค่าสำหรับคีย์ที่กด
   c. วางค่าที่แสดงบนบรรทัดที่เหมาะสมในคอลัมน์ Sequence ของไฟล์ $HOME/.3270keys

   ตัวอย่างเช่น หลังจากคุณเรียกใช้เอดิเตอร์ vi และเข้าสู่โหมดแทรก ให้กด Ctrl-V ตามด้วย Alt-Insert ซึ่งจะแสดง [[141q [ตัวแรกถูกแทนที่ด้วย \e ในคอลัมน์ Sequence ดังนั้นบรรทัดที่ถูกกำหนดจะมีลักษณะต่อไปนี้:

```
3270 Function Sequence  Key
bind  pa1      "\e[141q" #a_insert
```

ไฟล์ .k5login:

ไฟล์ .k5login ถูกใช้เมื่อใช้การพิสูจน์ตัวตน Kerberos V.5 สำหรับ secure rcmds ไฟล์นี้ระบุว่า DCE principals ในเซลล์ที่ที่ได้รับอนุญาตให้เข้าถึงบัญชีผู้ใช้ของผู้ใช้

ไฟล์อยู่ที่ $HOME/.k5login ไฟล์ควรเป็น ของผู้ใช้โลคัล และเจ้าของควรมีสิทธิการอ่านใน ไฟล์นี้ ค่าติดตั้งสิทธิขั้นต่ำสำหรับไฟล์นี้คือ 400

ไฟล์ .k5login มีรายการคู่ค่า DCE principal/ เซลล์ที่อนุญาตให้เข้าถึงบัญชีผู้ใช้ คู่ค่า principal/เซลล์ถูกเก็บใน รูปแบบ Kerberos (ซึ่งตรงข้ามกับรูปแบบ DCE) ตัวอย่างเช่น ถ้ามีไฟล์

```
UserA@Cell1
```

ดังนั้น DCE principal UserA บนเซลล์ DCE Cell1 สามารถ เข้าถึงบัญชีผู้ใช้

ถ้า DCE principal เหมือนกับชื่อบัญชีผู้ใช้ของผู้ใช้ และถ้า ไมมีไฟล์ $HOME/.k5login สำหรับบัญชีผู้ใช้นั้น DCE principal จะมีสิทธิเข้าถึงบัญชีผู้ใช้นั้น (การพิสูจน์ตัวตน Kerberos V.5 ที่มี ถูกกำหนดค่า)

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการพิสูจน์ตัวตน Kerberos V.5 ดูที่ “การพิสูจน์ตัวตนและ rcmds ที่ปลอดภัย” ในหน้า 116

วิธีการสำหรับการสื่อสารกับระบบและผู้ใช้อื่น

มีวิธีการสื่อสารกับระบบและผู้ใช้อื่นหลายวิธี โดยสองวิธีแรกจะถูกกล่าวถึงตอนนี้ วิธีแรกคือเชื่อมต่อโลคัลโฮสต์ กับรีโมตโฮสต์ วิธีที่สองคือสื่อสารกับผู้ใช้รีโมต

การเชื่อมต่อโลคัลโฮสต์กับรีโมตโฮสต์

คำสั่งการเชื่อมต่อโฮสต์ **TCP/IP** เหล่านี้ใช้สำหรับรีโมตล็อกอิน และการดำเนินการคำสั่ง

มีหลายเหตุผลซึ่งคุณอาจต้องการเข้าถึงคอมพิวเตอร์ ที่ไม่ใช่ของคุณ ตัวอย่างเช่น ผู้ดูแลระบบอาจต้องการ กำหนดสิทธิอนุญาตอีกครั้งในไฟล์ที่ไวต่อการเปลี่ยนแปลงซึ่งคุณกำลังทำงาน หรือ คุณอาจต้องการเข้าถึงไฟล์ส่วนบุคคลจากเวิร์กสเตชันของบุคลลอื่น คุณ สามารถแม้แต่เชื่อมต่อกับคอมพิวเตอร์ของคุณเองจากคอมพิวเตอร์สเตชันของบุคคลอื่น ฟังก์ชันรีโมตล็อกอิน เช่น คำสั่ง **rlogin, rexec,** และ **telnet** ช่วยให้โลคัลโฮสต์สามารถทำงานเป็น อินพุต/เอาต์พุตเทอร์มินัลโฮสต์ Key strokes ถูกส่งไปยังรีโมตโฮสต์ และ ผลลัพธ์แสดงขึ้นบนโลคัลมอนิเตอร์ เมื่อคุณสิ้นสุดรีโมตล็อกอิน เซสชัน ฟังก์ชันทั้งหมดกลับไปยังโลคัลโฮสต์ของคุณ

TCP/IP มีคำสั่งต่อไปนี้สำหรับรีโมตล็อกอิน และการดำเนินการคำสั่ง:

ไอเท็ม	คำอธิบาย
rexec	คำสั่ง **rexec** ทำให้สามารถ ดำเนินการคำสั่งแบบโต้ตอบบนโฮสต์ต่างประเทศที่แตกต่างอื่น เมื่อคุณล็อกอิน เข้าสู่รีโมตโฮสต์โดยใช้คำสั่ง **rlogin** คำสั่งนี้ถูกปิดใช้งาน โดยผู้จัดการระบบถ้าเครือข่ายของคุณต้องการความปลอดภัยมากเป็นพิเศษ เมื่อคุณออกใช้คำสั่ง **rexec** โลคัลโฮสต์จะค้นหา ไฟล์ `$HOME/.netrc` ของรีโมตโฮสต์เพื่อหาชื่อผู้ใช้ ของคุณและรหัสผ่านจากโลคัลโฮสต์ หากพบข้อมูลเหล่านี้ คำสั่ง ซึ่งคุณร้องขอให้รันบนโลคัลโฮสต์จะรัน มิฉะนั้น คุณ จะต้องระบุชื่อล็อกอินและรหัสผ่านก่อนสามารถดำเนินการคำร้องขอ ได้
rlogin	คำสั่ง **rlogin** ทำให้สามารถ ล็อกอินเข้าสู่โฮสต์ต่างประเทศที่คล้ายกัน ไม่เหมือนกับ **telnet** ซึ่ง สามารถใช้กับรีโมตโฮสต์ที่แตกต่างได้ คำสั่ง **rlogin** สามารถ ใช้บน UNIX โฮสต์เท่านั้น คำสั่งนี้ถูกปิดใช้งาน โดยผู้จัดการระบบถ้าเครือข่ายของคุณต้องการความปลอดภัยมาก เป็นพิเศษ คำสั่ง **rlogin** คล้ายกับ คำสั่ง **telnet** ในแง่ที่ว่า ทั้งสองคำสั่งช่วยให้โลคัลโฮสต์ เชื่อมต่อกับรีโมตโฮสต์ได้ ความแตกต่างเพียงอย่างเดียว คือคำสั่ง **rlogin** ไม่ใช่ คำสั่งที่เชื่อถือได้ และอาจถูกปิดใช้งานถ้าระบบของคุณต้องการความปลอดภัยมากเป็นพิเศษ คำสั่ง **rlogin** ไม่ใช่ คำสั่งที่เชื่อถือได้เนื่องจากทั้งไฟล์ `$HOME/.rhosts` ซึ่ง เป็นของผู้ใช้โลคัล และไฟล์ `/etc/hosts.equiv` ซึ่ง เป็นของผู้จัดการระบบ เก็บรักษารายการของรีโมตโฮสต์ที่ มีสิทธิเข้าถึงโลคัลโฮสต์ ด้วยเหตุนี้ ถ้าคุณปล่อยเทอร์มินัลทิ้งไว้โดยไม่มีการ เฝ้าระวัง ผู้ใช้ที่ไม่ได้รับอนุญาตสามารถตรวจสอบชื่อและรหัสผ่านที่มีอยู่ใน ไฟล์ดังกล่าว หรือกรณีเลวร้ายที่สุด อาจทำให้รีโมตโฮสต์เสียหายใน บางวิธี สิ่งที่ดีที่สุดคือ ผู้ใช้รีโมตควรต้องพิมพ์รหัสผ่านหลังจากออกใช้คำสั่ง **rlogin** แต่อาจสามารถข้ามคุณลักษณะที่แนะนำนี้ได้ หากทั้งไฟล์ `$HOME/.rhosts` และไฟล์ `/etc/hosts.equiv` ไม่มีชื่อของรีโมตโฮสต์ที่กำลังพยายามล็อกอิน โลคัลโฮสต์จะพร้อมต์ ขอรหัสผ่าน ไฟล์รหัสผ่านรีโมตจะถูกตรวจสอบก่อนเป็นอันดับแรกเพื่อตรวจสอบ รหัสผ่านที่ป้อน จากนั้น ล็อกอินพร้อมต์แสดงขึ้นอีกครั้งถ้ารหัสผ่าน ไม่ถูกต้อง การกด tilde และจุด (~.) ที่พร้อมต์ล็อกอินจะสิ้นสุดความพยายาม login แบบรีโมต คำสั่ง **rlogin** ยังสามารถมีการกำหนดคอนฟิก เพื่อใช้ Kerberos V.5 ในการพิสูจน์ตัวตนผู้ใช้ได้ด้วย อ็อพชันนี้ช่วยให้สามารถระบุ ผู้ใช้โดยไม่ต้องใช้ไฟล์ `$HOME/.rhosts` หรือ ส่งผ่านรหัสผ่านบนเครือข่าย สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการใช้นี้ของ คำสั่ง **rlogin** ให้ดูที่ “การพิสูจน์ตัวตนและ rcmds ที่ปลอดภัย” ในหน้า 116
rsh และ **remsh**	คำสั่ง **rsh** และ **remsh** ทำให้ สามารถดำเนินการคำสั่งบนโฮสต์ต่างประเทศที่คล้ายกันได้ อินพุตที่จำเป็นทั้งหมด ต้องทำโดยรีโมตโฮสต์ คำสั่ง **rsh** และ **remsh** ถูกปิดใช้งาน โดยผู้จัดการระบบถ้าเครือข่ายของคุณต้องการความปลอดภัยมากเป็นพิเศษ คำสั่ง **rsh** สามารถใช้ได้ในสองวิธีดังนี้: • การดำเนินการคำสั่งเดียวบนรีโมตโฮสต์เมื่อมีการระบุชื่อคำสั่ง • การดำเนินการคำสั่ง **rlogin** เมื่อไม่ได้ระบุ ชื่อคำสั่ง เมื่อออกใช้คำสั่ง **rsh** โลคัลโฮสต์จะค้นหา ไฟล์ `/etc/hosts.equiv` บนรีโมตโฮสต์ เพื่อหาสิทธิอนุญาตล็อกอิน หากไม่สำเร็จ จะค้นหาไฟล์ `$HOME/.rhosts` ทั้งสองไฟล์เหล่านี้เป็นรายการของรีโมตโฮสต์ที่มีสิทธิอนุญาตล็อกอิน ผู้ใช้รีโมตควรต้องพิมพ์รหัสผ่านหลังจากออกใช้คำสั่ง **rsh** นอกจากนี้ ยังสามารถตัดความจำเป็นในการออกใช้คำสั่ง **rlogin** ได้ด้วย คำสั่ง **rsh** อนุญาตการดำเนินการคำสั่งบน รีโมตโฮสต์ แต่ไม่ได้เป็นสื่อที่ใช้ข้ามความต้องการรหัสผ่าน หากต้องการรหัสผ่านเพื่อเข้าถึงรีโมตโฮสต์ ต้องระบุรหัสผ่านเพื่อ ใช้คำสั่ง **rsh** ด้วยเนื่องจากคำสั่งทั้งสองเข้าถึง ไฟล์ `$HOME/.rhosts` และไฟล์ `/etc/hosts.equiv` คำสั่ง **rsh** ยังสามารถมี การกำหนดคอนฟิกเพื่อใช้ Kerberos V.5 ในการพิสูจน์ตัวตนผู้ใช้ได้ด้วย อ็อพชันนี้ช่วยให้ สามารถระบุผู้ใช้โดยไม่ต้องใช้ไฟล์ `$HOME/.rhosts` หรือ ส่งผ่านรหัสผ่านบนเครือข่าย สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการใช้นี้ของ คำสั่ง **rsh** ให้ดูที่ “การพิสูจน์ตัวตนและ rcmds ที่ปลอดภัย” ในหน้า 116

ไอเท็ม	คำอธิบาย
telnet, tn, และ **tn3270**	คำสั่ง **telnet** เป็นโปรแกรมการเลียนแบบ เทอร์มินัลที่นำโปรโตคอล TELNET ไปใช้และช่วยให้คุณสามารถล็อกอินบน โฮสต์ต่างประเทศที่คล้ายหรือต่างกันได้ คำสั่งนี้ใช้ TCP/IP เพื่อสื่อสารกับ โฮสต์อื่นในเครือข่าย **หมายเหตุ:** เพื่อความสะดวก ต่อไปในเอกสารนี้ **telnet** หมายถึงคำสั่ง **telnet, tn,** และ **tn3270**
	คำสั่ง **telnet** เป็นวิธีหนึ่งที่ผู้ใช้สามารถล็อกอินเข้าสู่รีโมตโฮสต์ คุณลักษณะที่สำคัญที่สุด ของคำสั่ง **telnet** คือเป็นคำสั่ง *ที่เชื่อถือได้* ในทางตรงกันข้าม คำสั่ง **rlogin** ซึ่งสามารถใช้สำหรับรีโมตล็อกอิน ได้เช่นกัน ไม่ถือว่าเป็นคำสั่งที่เชื่อถือได้
	ระบบอาจต้องการความปลอดภัย มากเป็นพิเศษเพื่อป้องกันไม่ให้ผู้ใช้ที่ไม่ได้รับอนุญาตเข้าถึงไฟล์ของระบบ ขโมยข้อมูลที่สำคัญ ลบไฟล์ หรือใส่ไวรัสหรือหนอนบน ระบบ คุณลักษณะความปลอดภัยของ TCP/IP ได้รับการออกแบบมาเพื่อช่วยป้องกัน ไม่ให้เกิดเหตุการณ์ดังกล่าว
	ผู้ใช้ที่ต้องการล็อกอินเข้าสู่รีโมตโฮสต์โดยใช้คำสั่ง **telnet** ต้องระบุชื่อผู้ใช้และรหัสผ่านของผู้ใช้ที่ได้รับอนุมัติสำหรับคอมพิวเตอร์นั้น ซึ่งคล้ายกับโพรซีเดอร์ที่ใช้สำหรับการล็อกอินเข้าสู่โลคัลโฮสต์ เมื่อล็อกอิน เข้าสู่รีโมตโฮสต์เรียบร้อยแล้ว เทอร์มินัลของผู้ใช้จะรันราวกับว่าเชื่อมต่อเข้ากับ โฮสต์โดยตรง
	คำสั่ง **telnet** สนับสนุน อ็อพชันที่เรียกว่า *การเจรจาระหว่างเทอร์มินัล* หากรีโมตโฮสต์สนับสนุน การเจรจาระหว่างเทอร์มินัล คำสั่ง **telnet** จะส่งชนิดโลคัล เทอร์มินัลไปยังรีโมตโฮสต์ หากรีโมตโฮสต์ไม่ยอมรับชนิดโลคัล เทอร์มินัล คำสั่ง **telnet** จะพยายามเลียนแบบเทอร์มินัล 3270 และเทอร์มินัล DEC VT100 ถ้าคุณระบุเทอร์มินัลที่จะเลียนแบบ คำสั่ง **telnet** จะไม่เจรจาเกี่ยวกับชนิดเทอร์มินัล หากโลคัลและรีโมตโฮสต์ไม่สามารถตกลงเกี่ยวกับชนิดเทอร์มินัล โลคัลโฮสต์ จะใช้ค่าดีฟอลต์เป็น **ไม่มี**
	คำสั่ง **telnet** สนับสนุน ชนิดเทอร์มินัล 3270 เหล่านี้: 3277-1, 3278-1, 3278-2, 3278-3, 3278-4, และ 3278-5 หากคุณกำลังใช้คำสั่ง **telnet** ในโหมด 3270 บนจอแสดงผลสี สีและฟิลด์จะแสดงขึ้นเหมือนกับ บนจอแสดงผล 3279 โดยค่าดีฟอลต์ คุณสามารถเลือกสีอื่นโดยการแก้ไขไฟล์การแม็พคีย์บอร์ด ไฟล์ใดไฟล์หนึ่งในรายการก่อนหน้านี้ของชนิดเทอร์มินัล เมื่อ เซสชัน **telnet** สิ้นสุดลง จอแสดงผลจะถูกรีเซ็ตเป็นสี ที่ใช้ก่อนเซสชันเริ่มขึ้น
	คำสั่ง **telnet** ยังสามารถมีการกำหนดคอนฟิก เพื่อใช้ Kerberos V.5 ในการพิสูจน์ตัวตนผู้ใช้ได้ด้วย อ็อพชันนี้ช่วยให้ สามารถระบุผู้ใช้โดยไม่ต้องใช้ไฟล์ `$HOME/.rhosts` หรือ ส่งผ่านรหัสผ่านบนเครือข่าย สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการใช้นี้ของ คำสั่ง **telnet** ให้ดูที่ “การพิสูจน์ตัวตนและ rcmds ที่ปลอดภัย” ในหน้า 116

หมายเหตุ: คำสั่ง **rsh** และ **rexec** สามารถ ใช้เพื่อดำเนินการคำสั่งบนรีโมตโฮสต์ แต่ทั้งสองคำสั่งไม่ใช่คำสั่ง ที่เชื่อถือได้ ดังนั้นจึงอาจไม่ตรงกับระดับความปลอดภัยทั้งหมดซึ่งกำหนดคอนฟิกในคอมพิวเตอร์ของคุณ ผลคือคำสั่งเหล่านี้อาจถูกปิดใช้งาน ถ้าระบบของคุณต้องการ ความปลอดภัยมากเป็นพิเศษ

การล็อกอินเข้าสู่รีโมตโฮสต์

คุณสามารถล็อกอินเข้าสู่รีโมตโฮสต์ได้โดยใช้คำสั่ง **telnet**

เพื่อทำเช่นนี้ คุณต้องมี ID ผู้ใช้และรหัสผ่านที่ถูกต้องสำหรับ รีโมตโฮสต์

เมื่อต้องการล็อกอินเข้าสู่รีโมตโฮสต์ (`host1` ในตัวอย่าง นี้) ให้พิมพ์:

```
telnet host1
```

ข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้แสดงขึ้นบนหน้าจอของคุณ:

```
Trying . . .
Connected to host1
Escape character is '^T'.

AIX telnet (host1)
```

```
AIX Operating System
Version 7.1
(/dev/pts0)
login:_
```

หลังจากคุณล็อกอินแล้ว คุณสามารถออกใช้คำสั่งได้ เมื่อต้องการล็อกเอาต์ออกจากระบบ และปิดการเชื่อมต่อ ให้กดปุ่ม Ctrl-D ตามลำดับ

หากคุณไม่สามารถล็อกอินได้ ให้ยกเลิกการเชื่อมต่อโดยกดปุ่ม Ctrl-T ตาม ลำดับ

การสนทนากับผู้ใช้รีโมต

ใช้คำสั่ง **talk** เพื่อทำการสนทนาแบบเรียลไทม์ กับผู้ใช้อื่นบนรีโมตโฮสต์

1. **talkd** daemon ต้องใช้งานอยู่บนทั้งโลคัลและ รีโมตโฮสต์
2. ผู้ใช้บนรีโมตโฮสต์ต้องมีการล็อกอิน

คำสั่ง **talk** ต้องการแอดเดรสที่ถูกต้อง ซึ่งจะผูกไว้ชื่อโฮสต์ของรีโมตเทอร์มินัลต้องถูกผูกเข้ากับ อินเตอร์เฟสเครือข่ายที่กำลังทำงานซึ่งใช้ได้โดยคำสั่งเครือข่ายอื่น เช่น คำสั่ง **ping** หากเครื่องไม่มีอินเตอร์เฟสเครือข่าย ที่เป็นเทอร์มินัลแบบสแตนด์อะโลน เครื่องต้องผูกชื่อโฮสต์ของตนเข้ากับ loopback address (127.0.0.1) เพื่อให้คำสั่ง **talk** ทำงานได้

โดยใช้ อิเล็กทรอนิกส์เมล คุณสามารถส่งข่าวสารข้อความไปยังผู้ใช้อื่นบนเครือข่ายโลคัล และรับเมลจากผู้ใช้อื่นได้เช่นกัน ถ้าระบบคอมพิวเตอร์มีการตั้งค่าคอนฟิกอย่างเหมาะสม และคุณทราบอิเล็กทรอนิกส์แอดเดรสที่เหมาะสม คุณสามารถส่งข้อความ อิเล็กทรอนิกส์เมลไปยังบางคนบนระบบรีโมตได้ทั่วโลก

TCP/IP มีคำสั่ง ต่อไปนี้สำหรับการสื่อสารรีโมต:

ไอเท็ม	คำอธิบาย
mail	ส่งและรับบันทึกและจดหมายอิเล็กทรอนิกส์
talk	ช่วยให้คุณสามารถสนทนาเชิงโต้ตอบกับผู้ใช้บนรีโมตโฮสต์

1. เมื่อต้องการพูดคุยกับผู้ใช้รีโมต `dale@host2` ที่ล็อกอิน บนรีโมตโฮสต์ `jane@host1` ให้พิมพ์:

   ```
   talk dale@host2
   ```

   ข้อความคล้ายกับตัวอย่างต่อไปนี้แสดงขึ้นบนหน้าจอของ `dale@host2`:

   ```
   Message from TalkDaemon@host1 at 15:16...
   talk: connection requested by jane@host1.
   talk: respond with: talk jane@host1
   ```

   ข้อความนี้แจ้ง `dale@host2` ว่า `jane@host1` กำลัง พยายามสนทนากับเธอ
2. เมื่อต้องการยอมรับคำเชิญ `dale@host2` จะพิมพ์:

   ```
   talk jane@host1
   ```

   ขณะนี้ ผู้ใช้ `dale@host2` และ `jane@host1` สามารถทำการสนทนาเชิงโต้ตอบได้
3. เมื่อต้องการสิ้นสุดการสนทนาในเวลาใดๆ ผู้ใช้รายใดรายหนึ่งสามารถกดปุ่ม Ctrl-C ตามลำดับ ซึ่งจะส่งคืนผู้ใช้ไปยังพร้อมต์บรรทัดคำสั่ง

การถ่ายโอนไฟล์

แม้ว่าจะสามารถส่งไฟล์ขนาดเล็กโดยใช้อิเล็กทรอนิกส์เมล แต่ยังมีวิธีที่มีประสิทธิภาพกว่าในการส่งไฟล์ขนาดใหญ่

โปรแกรมอิเล็กทรอนิกส์เมลถูกออกแบบมาเพื่อส่งข้อความขนาดเล็ก ดังนั้นสื่ออื่นจึงเหมาะสำหรับการโอนย้ายไฟล์ขนาดใหญ่อย่างมีประสิทธิภาพกว่า คำสั่ง **ftp, rcp** และ **tftp** ขึ้นอยู่กับ **TCP/IP** เพื่อสร้างการเชื่อมต่อจากโลคัลโฮสต์ของคุณ ไปยังรีโมตโฮสต์โดยตรง Basic Network Utilities (BNU) สามารถใช้ **TCP/IP** เพื่อจัดเตรียมการเชื่อมต่อกับโฮสต์อื่นโดยตรง

การโอนย้ายไฟล์โดยใช้คำสั่ง ftp และ rcp

ใช้คำสั่ง **ftp** เพื่อคัดลอกไฟล์จากรีโมตโฮสต์ คำสั่ง **ftp** จะไม่รักษาแอ็ตทริบิวต์ของไฟล์ หรือคัดลอกไดเร็กทอรีย่อย ถ้าสภาวะเหล่านี้เป็นสิ่งจำเป็น ให้ใช้คำสั่ง **rcp**

ไอเท็ม	คำอธิบาย
ftp	ใช้ **File Transfer Protocol (FTP)** เพื่อโอนย้ายไฟล์ระหว่างโฮสต์ที่ใช้ระบบไฟล์ หรืออักขระแสดงผลแตกต่างกัน เช่น EBCDIC และ ASCII ซึ่งจัดเตรียมสำหรับความปลอดภัยโดยส่งรหัสผ่านไปให้รีโมตโฮสต์ และอนุญาตให้ล็อกอินอัตโนมัติ, โอนย้ายไฟล์ และล็อกออฟ
rcp	คัดลอกไฟล์หนึ่งไฟล์หรือมากกว่าระหว่างโลคัลโฮสต์กับรีโมตโฮสต์ ระหว่างรีโมตโฮสต์ ที่แยกกัน หรือระหว่างไฟล์ในรีโมตโฮสต์เดียวกัน คำสั่งนี้เหมือนคำสั่ง **cp** ยกเว้นคำสั่งจะทำงาน เฉพาะการดำเนินการไฟล์รีโมต ถ้าต้องการความปลอดภัยมากยิ่งขึ้นสำหรับเครือข่ายของคุณ คำสั่งนี้จะถูกปิดใช้งานโดยผู้จัดการระบบ

ก่อนที่จะพยายามโอนย้ายไฟล์โดยใช้คำสั่ง **ftp** และ **rcp** ตรวจดูให้แน่ใจว่าสภาวะต่อไปนี้เป็นจริง:

1. คุณต้องมีสิทธิในการล็อกอินรีโมตระบุไว้ในไฟล์ `$HOME/.netrc` ของรีโมตโฮสต์ ถ้าใช้คุณลักษณะการล็อกอินอัตโนมัติ มิฉะนั้น คุณต้องทราบชื่อล็อกอินและ รหัสผ่านสำหรับรีโมตโฮสต์ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับไฟล์ `.netrc` โปรดดู “การสร้างไฟล์ .netrc” ในหน้า 118

   อีกวิธีหนึ่ง ระบบสามารถกำหนดคอนฟิกเพื่อใช้การพิสูจน์ตัวตนแบบ Kerberos V.5 นี่ถูกใช้ แทนไฟล์ `.netrc` หรือ `$HOME/.rhosts` โปรดดู “การพิสูจน์ตัวตนและ rcmds ที่ปลอดภัย” ในหน้า 116
2. ถ้าคุณต้องการคัดลอกไฟล์จากรีโมตโฮสต์ คุณต้องมีสิทธิในการอ่านไฟล์นั้น

   หมายเหตุ: สิทธิในการอ่านและเขียนสำหรับไฟล์และไดเร็กทอรีบนรีโมตโฮสต์ กำหนดโดยชื่อล็อกอินที่ใช้งาน
3. ถ้าคุณต้องการคัดลอกไฟล์จากโลคัลโฮสต์มายังรีโมตโฮสต์ คุณต้องมี สิทธิในการเขียนสำหรับไดเร็กทอรีที่เก็บไฟล์ที่คัดลอก นอกจากนี้ ถ้าไดเร็กทอรีบนรีโมตโฮสต์มีไฟล์ที่มีชื่อเดียวกับไฟล์ที่คุณต้องการ วาง คุณต้องมีสิทธิในการเขียนเพื่อเพิ่ม ไฟล์บนรีโมตโฮสต์

การล็อกอินเข้าสู่รีโมตโฮสต์โดยตรง:

เมื่อใช้ **TCP/IP** เพื่อโอนย้ายไฟล์ คุณสามารถ ใช้โพรซีเดอร์นี้เพื่อล็อกอินเข้าสู่รีโมตโฮสต์โดยตรง

1. ใช้คำสั่ง **cd** เพื่อเปลี่ยนไปยังไดเร็กทอรี ที่มีไฟล์ที่คุณต้องการส่ง (การส่งไฟล์) หรือไปยังไดเร็กทอรี ที่คุณต้องการเก็บไฟล์ที่โอนย้าย (การรับไฟล์)
2. ล็อกอินเข้าสู่รีโมตโฮสต์โดยตรง โดยพิมพ์:

   ```
   ftp HostName
   ```

   ถ้าคุณมีสิทธิใช้งานล็อกอินอัตโนมัติ ข้อมูลคล้ายด้านล่างจะปรากฏบน โลคัลโฮสต์ของคุณ:

```
Connected to canopus.austin.century.com.
220 canopus.austin.century.com FTP server
(Version 4.1 Sat Nov 23 12:52:09 CST 1995) ready.
331 Password required for dee.
230 User dee logged in.
ftp>
```

หรือ ข้อมูลคล้ายด้านล่าง จะปรากฏบนโลคัลโฮสต์:

```
Connected to canopus.austin.century.com.
220 canopus.austin.century.com FTP server
(Version 4.1 Sat Nov 23 12:52:09 CST 1995) ready.
Name (canopus:eric): dee
331 Password required for dee.
Password:
230 User dee logged in.
ftp>
```

3. **ป้อนชื่อล็อกอินและรหัสผ่านของคุณเมื่อพร้อมต์โดยระบบ**

คุณพร้อมที่จะคัดลอกไฟล์ระหว่างสองโฮสต์แล้ว

การล็อกอินเข้าสู่รีโมตโฮสต์ทางอ้อม:

เมื่อใช้ TCP/IP เพื่อโอนย้ายไฟล์ คุณสามารถ ใช้โพรซีเดอร์นี้เพื่อล็อกอินเข้าสู่รีโมตโฮสต์ทางอ้อม

1. **ใช้คำสั่ง cd เพื่อเปลี่ยนไปยังไดเร็กทอรี ที่มีไฟล์ที่คุณต้องการส่ง (การส่งไฟล์) หรือไปยังไดเร็กทอรี ที่คุณต้องการเก็บไฟล์ที่โอนย้าย (การรับไฟล์)**
2. **ล็อกอินเข้าสู่รีโมตโฮสต์ทางอ้อมโดยพิมพ์:**

   ```
   ftp
   ```

3. **เมื่อพร้อมต์ ftp> ปรากฏขึ้น ให้พิมพ์:**

   open *HostName*

   ถ้าคุณมีสิทธิใช้งานล็อกอินอัตโนมัติ ข้อมูลคล้ายด้านล่างจะปรากฏบน โลคัลโฮสต์ของคุณ:

   ```
   Connected to canopus.austin.century.com.
   220 canopus.austin.century.com FTP server
   (Version 4.1 Sat Nov 23 12:52:09 CST 1995) ready.
   331 Password required for dee.
   230 User dee logged in.
   ftp>
   ```

   หรือ ข้อมูลคล้ายด้านล่าง จะปรากฏบนโลคัลโฮสต์:

   ```
   Connected to canopus.austin.century.com.
   220 canopus.austin.century.com FTP server
   (Version 4.1 Sat Nov 23 12:52:09 CST 1995) ready.
   Name (canopus:eric): dee
   331 Password required for dee.
   Password:
   230 User dee logged in.
   ftp>
   ```

4. **พิมพ์ชื่อผู้ใช้และรหัสผ่านเมื่อพร้อมต์โดยระบบ**

การคัดลอกไฟล์จากรีโมตโฮสต์ไปยังโลคัลโฮสต์:

ใช้คำสั่ง **ftp** เพื่อคัดลอกไฟล์จากรีโมต โฮสต์ไปยังโลคัลโฮสต์

เมื่อต้องการคัดลอกไฟล์จากรีโมตโฮสต์ไปยังโลคัลโฮสต์โดยใช้คำสั่ง **ftp** อันดับแรก คุณต้องล็อกอินเข้าสู่ระบบรีโมตโดยทางตรงหรือทางอ้อม อย่างใดอย่างหนึ่ง ให้ดูที่ การล็อกอินเข้าสู่รีโมตโฮสต์โดยทางตรง หรือ การล็อกอินเข้าสู่รีโมตโฮสต์โดยทางอ้อม สำหรับคำแนะนำ

หมายเหตุ: คำสั่ง **ftp** ใช้ ชนิดการโอนย้ายดีฟอลต์ ASCII เพื่อคัดลอกไฟล์

เมื่อต้องการคัดลอกไฟล์ จากรีโมตโฮสต์ไปยังโลคัลโฮสต์:

1. กำหนดว่าไฟล์ซึ่งคุณต้องการคัดลอกอยู่ในไดเร็กทอรีปัจจุบันหรือไม่ โดยการรันคำสั่งย่อย **dir** (คำสั่งย่อย **dir** ของ คำสั่ง **ftp** ทำงานในลักษณะเดียวกันกับคำสั่ง **ls –l**) หากไฟล์ไม่ได้อยู่ในไดเร็กทอรีปัจจุบัน ให้ใช้คำสั่งย่อย **cd** เพื่อย้าย ไปยังไดเร็กทอรีที่เหมาะสม
2. เมื่อต้องการคัดลอกโลคัลไฟล์ของคุณโดยใช้รูปภาพไบนารี ให้พิมพ์:

   ```
   binary
   ```
3. เมื่อต้องการคัดลอกไฟล์ไปยังโฮสต์ของคุณ ให้พิมพ์:

   ```
   get FileName
   ```

   ไฟล์ถูกวางไว้ในไดเร็กทอรีซึ่งคุณออกใช้คำสั่ง **ftp**
4. เมื่อต้องการสิ้นสุดเซสชัน ให้กดปุ่ม Ctrl-D ตามลำดับ หรือพิมพ์ `quit`

การคัดลอกไฟล์จากโลคัลโฮสต์ไปยังรีโมตโฮสต์:

ใช้คำสั่ง **ftp** เพื่อคัดลอกไฟล์จากโลคัล โฮสต์ไปยังรีโมตโฮสต์

เมื่อต้องการคัดลอกไฟล์จากโลคัลโฮสต์ไปยังรีโมตโฮสต์โดยใช้คำสั่ง **ftp** อันดับแรก คุณต้องล็อกอินเข้าสู่ระบบรีโมตโดยทางตรงหรือทางอ้อม อย่างใดอย่างหนึ่ง ให้ดูที่ การล็อกอินเข้าสู่รีโมตโฮสต์โดยทางตรง หรือ การล็อกอินเข้าสู่รีโมตโฮสต์โดยทางอ้อม สำหรับคำแนะนำ

หมายเหตุ: คำสั่ง **ftp** ใช้ ชนิดการโอนย้ายดีฟอลต์ ASCII เพื่อคัดลอกไฟล์

เมื่อต้องการคัดลอกไฟล์ จากโลคัลโฮสต์ไปยังรีโมตโฮสต์:

1. ถ้าคุณต้องการวางไฟล์ไว้ในไดเร็กทอรีอื่นที่ไม่ใช่ไดเร็กทอรี `$HOME` ให้ใช้คำสั่ง **cd** เพื่อย้ายไปยังไดเร็กทอรีที่ต้องการ
2. เมื่อต้องการคัดลอกโลคัลไฟล์ของคุณโดยใช้รูปภาพไบนารี ให้พิมพ์:

   ```
   binary
   ```
3. เมื่อต้องการคัดลอกไฟล์ไปยังรีโมตโฮสต์ ให้พิมพ์:

   ```
   put FileName
   ```
4. เมื่อต้องการสิ้นสุดเซสชัน ให้กดปุ่ม Ctrl-D ตามลำดับ หรือพิมพ์ `quit`

การถ่ายโอนไฟล์โดยใช้คำสั่ง tftp และ utftp

ใช้คำสั่ง **tftp** และ **utftp** สำหรับ **Trivial File Transfer Protocol** (**TFTP**) เพื่อถ่ายโอนไฟล์ ไปและจาก โฮสต์

เนื่องจาก **TFTP** เป็นโปรโตคอลการถ่ายโอนไฟล์เดี่ยว คำสั่ง **tftp** and **utftp** จะไม่มีคุณลักษณะทั้งหมดของคำสั่ง **ftp** ถ้าจำเป็นต้องมีการรักษาความปลอดภัยเป็นพิเศษสำหรับเน็ตเวิร์กของคุณ ผู้จัดการระบบสามารถปิด คำสั่งนี้

หมายเหตุ: คำสั่ง **tftp** ใช้ไม่ได้เมื่อ โฮสต์ของคุณทำงานที่ระดับการรักษาความปลอดภัยสูง

ก่อนทำการถ่ายโอนไฟล์โดยใช้คำสั่ง **tftp** และ **utftp** ตรวจสอบว่าเงื่อนไขต่อไปนี้เป็นจริง:

1. ถ้าคุณต้องการคัดลอกไฟล์ จา*ก* รีโมตโฮสต์ คุณต้องมีสิทธิ *read* สำหรับไดเร็กทอรีที่มีไฟล์ที่ต้องการ
2. ถ้าคุณต้องการคัดลอกไฟล์ *ไปที่* รีโมตโฮสต์ คุณต้องมีสิทธิ *write* สำหรับไดเร็กทอรีซึ่งไฟล์จะถูกกำหนดไว้

การคัดลอกไฟล์จากรีโมตโฮสต์:

เมื่อใช้ **TCP/IP** เพื่อคัดลอกไฟล์ คุณสามารถใช้โพรซีเดอร์นี้ เพื่อคัดลอกไฟล์จากรีโมตโฮสต์

1. เมื่อต้องการสร้างการเชื่อมต่อกับรีโมตโฮสต์ ให้พิมพ์:

   ```
   tftp host1
   ```

   ในตัวอย่างนี้ `host1` เป็นชื่อของโฮสต์ ที่คุณต้องการเชื่อมต่อ

   พร้อมต์ `tftp>` จะปรากฏขึ้น
2. เมื่อต้องการกำหนดให้สร้างการเชื่อมต่อ ให้พิมพ์:

   ```
   status
   ```

   ข้อความจะคล้ายกับที่แสดงต่อไปนี้:

   ```
   Connected to host1
   Mode: netascii Verbose: off Tracing: off
   Remxt-interval: 5 seconds, Max-timeout: 25 seconds
   tftp>
   ```
3. ป้อนคำสั่งย่อย **get** ชื่อของไฟล์ ที่จะโอนย้าย และชื่อที่ตั้งให้ไฟล์บนระบบรีโมต :

   ```
   get /home/alice/update update
   ```

   ไดเร็กทอรี `/home/alice` บนรีโมตโฮสต์ต้องมี สิทธิในการอ่านตั้งค่าไว้สำหรับผู้ใช้อื่น ในตัวอย่างนี้ ไฟล์ `/home/alice/update` จะโอนย้ายจาก `host1` เป็นไฟล์ `update` ในไดเร็กทอรีปัจจุบันบนระบบโลคัล
4. เมื่อต้องการสิ้นสุดเซสชัน ให้พิมพ์:

   ```
   quit
   ```

   หรือ กดลำดับคีย์ Ctrl-D

การคัดลอกไฟล์ไปยังรีโมตโฮสต์:

เมื่อใช้ **TCP/IP** เพื่อคัดลอกไฟล์ คุณสามารถใช้โพรซีเดอร์นี้ เพื่อคัดลอกไฟล์ไปยังรีโมตโฮสต์

1. เมื่อต้องการสร้างการเชื่อมต่อกับรีโมตโฮสต์ ให้พิมพ์:

   ```
   tftp host1
   ```

   ในตัวอย่างนี้ `host1` เป็นชื่อของโฮสต์ ที่คุณต้องการเชื่อมต่อ

   พร้อมต์ `tftp>` จะปรากฏขึ้น
2. เมื่อต้องการกำหนดให้สร้างการเชื่อมต่อ ให้พิมพ์:

   ```
   status
   ```

ข้อความจะคล้ายกับที่แสดงต่อไปนี้:

```
Connected to host1
Mode: netascii Verbose: off Tracing: off
Remxt-interval: 5 seconds, Max-timeout: 25 seconds
tftp>
```

3. ป้อนคำสั่งย่อย **put** ชื่อของไฟล์ ที่จะโอนย้ายจากโลคัลโฮสต์ และพาธและชื่อไฟล์สำหรับไฟล์ บนรีโมตโฮสต์:

   ```
   put myfile /home/alice/yourfile
   ```

   ไดเร็กทอรี `/home/alice` บนรีโมตโฮสต์ต้องมีสิทธิในการเขียน ตั้งค่าไว้สำหรับผู้ใช้อื่น ไฟล์ `myfile` อยู่ในไดเร็กทอรีทำงานปัจจุบันของผู้ใช้ จะถูกโอนย้ายไปที่ `host1` ชื่อพาธต้องถูกระบุ นอกจากตั้งค่าดีฟอลต์ไว้ ไฟล์ `myfile` จะปรากฏบนรีโมตโฮสต์เป็น `yourfile`
4. เมื่อต้องการสิ้นสุดเซสชัน ให้พิมพ์:

   ```
   quit
   ```

   หรือ ใช้ลำดับคีย์ Ctrl-D

การพิมพ์ไฟล์ในระบบรีโมต

ถ้าคุณมีเครื่องพิมพ์โลคัลติดตั้งไว้กับโฮสต์ของคุณ ให้ปฏิบัติตาม โพรซีเดอร์ต่อไปนี้เพื่ออ้างถึงการพิมพ์บนเครื่องพิมพ์รีโมต ถ้าคุณไม่มีเครื่องพิมพ์โลคัล ให้ปฏิบัติตามโพรซีเดอร์ต่อไปนี้อ้างถึงเครื่องพิมพ์รีโมตที่ไม่ใช่ดีฟอลต์

1. ชื่อโฮสต์ของคุณต้องปรากฏอยู่ในไฟล์ `/etc/hosts.lpd` ของรีโมตโฮสต์

   หมายเหตุ: ระบบการคิว ไม่สนับสนุนชื่อโฮสต์มัลติไบต์

   เมื่อต้องการนำการเปลี่ยนแปลง ของไฟล์ `/etc/hosts.lpd` ไปใช้โดยไม่ต้องรีสตาร์ทระบบ ให้ใช้คำสั่ง System Resource Controller (SRC) **refresh**
2. คุณต้องสามารถกำหนดชื่อคิวและชื่อเครื่องพิมพ์รีโมต ในไฟล์ `/usr/lib/lpd/qconfig` โลคัลของคุณ

คุณสามารถใช้คำสั่ง **enq** หรือ System Management Interface Tool (SMIT)เพื่อดำเนินการภารกิจนี้ให้เสร็จ

> **หมายเหตุ:** ส่วนนี้ อธิบายถึงวิธีพิมพ์ในรีโมตโฮสต์ที่ระดับง่ายสุด สำหรับข้อมูลและแนวคิดเพิ่มเติมเกี่ยวกับการพิมพ์รีโมต โปรดดูคำสั่ง **enq**

การวางงานพิมพ์ในคิวพิมพ์รีโมต

เมื่อใช้ **TCP/IP** ในการพิมพ์ไฟล์ คุณสามารถใช้โพรซีเดอร์นี้ เพื่อวางงานพิมพ์ในคิวพิมพ์รีโมต

เพื่อให้คุณวางงานในคิวการพิมพ์รีโมต ชื่อโฮสต์ของคุณ ต้องปรากฏอยู่ในไฟล์ `/etc/hosts.lpd` ของรีโมตโฮสต์ (ระบบคิวไม่สนับสนุนชื่อโฮสต์แบบหลายไบต์) เมื่อต้องการนำการเปลี่ยนแปลง ของไฟล์ `/etc/hosts.lpd` ไปใช้โดยไม่ต้องรีสตาร์ทระบบ ให้ใช้คำสั่ง System Resource Controller (SRC) **refresh** นอกจากนี้ คุณต้องสามารถกำหนดชื่อคิวและชื่อเครื่องพิมพ์รีโมตในไฟล์ `/usr/lib/lpd/qconfig` โลคัลของคุณ

1. ค้นหาชื่อคิวและชื่ออุปกรณ์รีโมตที่เหมาะสม ชื่อคิวมักขึ้นต้นด้วยอักษร `rp` และตามด้วยตัวเลขหรือ ชุดตัวเลข ชื่อเครื่องพิมพ์รีโมตมักขึ้นต้นด้วยอักษร `drp` แล้วตามด้วย ตัวเลขหรือชุดตัวเลข
2. ป้อนคำสั่งต่อไปนี้:

   ```
   enq -P QueueName:PrinterName FileName
   ```

โดย *QueueName* คือ ชื่อของคิว (เช่น `rp1`) และ *PrinterName* คือชื่อเครื่องพิมพ์ (เช่น `drp1`) ที่พบในไฟล์ `/usr/lib/lpd/qconfig` ห้ามลืมเครื่องหมายโคลอน (`:`) ระหว่าง *QueueName* กับ *PrinterName FileName* คือ ชื่อของไฟล์ที่คุณต้องการพิมพ์

ต่อไปนี้คือตัวอย่างวิธีใช้งานคำสั่ง **enq** :

- เมื่อต้องการพิมพ์ไฟล์ `memo` บนเครื่องพิมพ์ดีฟอลต์ ให้พิมพ์:

  ```
  enq memo
  ```

- เมื่อต้องการพิมพ์ไฟล์ `prog.c` พร้อมหมายเลขหน้า ให้พิมพ์:

  ```
  pr prog.c | enq
  ```

  คำสั่ง **pr** วางส่วนหัวไว้ที่ด้านบน สุดของแต่ละหน้า ซึ่งรวมถึงวันที่ไฟล์ถูกแก้ไขล่าสุด ชื่อของไฟล์และหมายเลขหน้า จากนั้น คำสั่ง **enq** พิมพ์ไฟล์

- เมื่อต้องการพิมพ์ไฟล์ `report` บนเครื่องพิมพ์ที่พร้อมใช้งานถัดไป ที่กำหนดคอนฟิกสำหรับคิว `fred` ให้พิมพ์:

  ```
  enq -P fred report
  ```

- เมื่อต้องการพิมพ์หลายไฟล์ที่ขึ้นตอนด้วย `sam` บนเครื่องพิมพ์ที่พร้อมใช้งานถัดไปสำหรับคิว `fred` ให้พิมพ์:

  ```
  enq -P fred sam*
  ```

  ไฟล์ทั้งหมดที่ขึ้นต้นด้วย `sam` จะถูกรวมไว้ในหนึ่งงานพิมพ์ คำสั่งสถานะปกติแสดงเฉพาะ ชื่อของงานพิมพ์ ซึ่งในกรณีนี้คือ ชื่อของไฟล์แรกในคิว นอกจากค่าอื่นถูกระบุด้วยแฟล็ก **–T** เมื่อต้องการแสดงรายชื่อของไฟล์ทั้งหมดใน งานพิมพ์ ให้ใช้คำสั่งสถานะนาน **enq –A –L**

การจัดคิวงานโดยใช้ SMIT

เมื่อใช้ **TCP/IP** เพื่อจัดคิวไฟล์ คุณสามารถ ใช้คำสั่ง **smit**

1. เมื่อต้องการจัดคิวโดยใช้ SMIT, พิมพ์คำสั่งต่อไปนี้:

   ```
   smit
   ```

2. เลือก **Spooler** และเริ่มต้นเมนูงานพิมพ์
3. เลือกอ็อพชัน **File to Print** และพิมพ์ชื่อไฟล์ที่คุณต้องการพิมพ์
4. เลือกอ็อพชัน **Print Queue** และเลือก ชื่อของเครื่องพิมพ์รีโมตที่คุณต้องการพิมพ์

คุณพร้อมที่จะพิมพ์ไปยังเครื่องพิมพ์รีโมต

การพิมพ์ไฟล์จากระบบรีโมต

ในบางครั้ง คุณอาจต้องการพิมพ์ไฟล์ที่อยู่ในรีโมตโฮสต์ ตำแหน่งของงานพิมพ์ขึ้นอยู่กับเครื่องพิมพ์รีโมตใดที่ พร้อมใช้งานกับรีโมตโฮสต์

1. คุณต้องสามารถล็อกอินเข้าสู่ระบบรีโมตได้ โดยใช้คำสั่ง **rlogin** หรือ **telnet**
2. คุณต้องมีสิทธิในการอ่านสำหรับรีโมตไฟล์ที่คุณต้องการพิมพ์ บนเครื่องพิมพ์โลคัลของคุณ

   หมายเหตุ: โพรซีเดอร์นี้อธิบายวิธีการพิมพ์ไปที่ รีโมตโฮสต์ในระดับที่ง่ายที่สุด สำหรับข้อมูลและแนวคิดเพิ่มเติมเกี่ยวกับการพิมพ์แบบรีโมต โปรดอ่านเกี่ยวกับคำสั่ง **enq**

เมื่อต้องการพิมพ์ จากระบบรีโมต:

1. ล็อกอินเข้าสู่ระบบรีโมต โดยใช้คำสั่ง **rlogin** or **telnet**

2. ค้นหาชื่อคิวและชื่ออุปกรณ์รีโมตที่เหมาะสม ชื่อคิวมักขึ้นต้นด้วยอักษร rp และตามด้วยตัวเลขหรือ ชุดตัวเลข ชื่อเครื่องพิมพ์รีโมตมักขึ้นต้นด้วยอักษร drp แล้วตามด้วย ตัวเลขหรือชุดตัวเลข
3. พิมพ์คำสั่งต่อไปนี้:

   ```
   enq -P QueueName:PrinterName FileName
   ```

   โดย *QueueName* คือ ชื่อของคิว (เช่น rp1) และ *PrinterName* คือชื่อเครื่องพิมพ์ (เช่น drp1) ที่พบในไฟล์ /usr/lib/lpd/qconfig ห้ามลืมเครื่องหมาย : (โคลอน) ระหว่าง *QueueName* กับ *PrinterName* *FileName* คือชื่อของไฟล์ที่คุณต้องการพิมพ์
4. สิ้นสุดการเชื่อมต่อกับรีโมตโฮสต์โดยกดลำดับปุ่ม Ctrl-D หรือโดยพิมพ์ quit

การแสดงผลข้อมูลสถานะ

คุณสามารถใช้คำสั่ง **TCP/IP** เพื่อกำหนดสถานะ ของเครือข่าย แสดงผลข้อมูลเกี่ยวกับผู้ใช้ และแก้ไขข้อมูลโฮสต์ ที่จำเป็นต่อการสื่อสารกับโฮสต์หรือผู้ใช้อื่น

คำสั่งสถานะ TCP/IP

TCP/IP มีคำสั่งสถานะเพื่อกำหนดสถานะของ โลคัลและรีโมตโฮสต์และเน็ตเวิร์ก

ไอเท็ม	คำอธิบาย
finger หรือ **f**	แสดงข้อมูลเกี่ยวกับผู้ใช้ปัจจุบันบนโฮสต์ที่ระบุ ข้อมูลนี้สามารถระบุชื่อล็อกอินของผู้ใช้ และชื่อเทอร์มินัล พร้อมกับวันและเวลาที่ล็อกอิน
โฮสต์	เปลี่ยนชื่อโฮสต์ให้เป็นอินเตอร์เน็ตแอดเดรส หรืออินเตอร์เน็ตแอดเดรสเป็น ชื่อโฮสต์
ping	ช่วยกำหนดสถานะของเน็ตเวิร์กหรือโฮสต์ เป็นเรื่องปกติที่สุด ที่ใช้เพื่อตรวจสอบว่าเน็ตเวิร์กหรือโฮสต์ที่รันอยู่ขณะนี้
rwho	แสดงผู้ใช้ที่ถูกบันทึกในโฮสต์บนโลคัลเน็ตเวิร์ก คำสั่งนี้ แสดงชื่อผู้ใช้ ชื่อโฮสต์ และวันที่และเวลาของล็อกอินสำหรับทุกคนบนล็อกอินเน็ตเวิร์ก
whois	ระบุเจ้าของ ID ผู้ใช้หรือ nickname คำสั่งนี้สามารถใช้เฉพาะ ถ้าโลคัลเน็ตเวิร์กของคุณถูกเชื่อมต่อกับอินเตอร์เน็ต

การแสดงข้อมูลเกี่ยวกับผู้ใช้ทั้งหมดที่ล็อกอินเข้าสู่โฮสต์

ใช้โพรซีเดอร์นี้เพื่อดูข้อมูลเกี่ยวกับผู้ใช้ *ทั้งหมด* ที่ ล็อกอินเข้าสู่รีโมตโฮสต์

เมื่อต้องการแสดงข้อมูลเกี่ยวกับผู้ใช้ทั้งหมดที่ล็อกอินเข้าสู่ รีโมตโฮสต์:

1. ล็อกอินเข้าสู่รีโมตโฮสต์ซึ่งคุณต้องการสื่อสาร
2. เมื่อต้องการแสดงข้อมูลเกี่ยวกับผู้ใช้ทั้งหมดที่ล็อกอินเข้าสู่โฮสต์ alcatraz ให้พิมพ์:

   ```
   finger @alcatraz
   ```

   ข้อมูลคล้ายกับ ตัวอย่างต่อไปนี้แสดงขึ้น:

   ```
   brown    console  Mar 15 13:19
   smith    pts0     Mar 15 13:01
   jones    tty0    Mar 15 13:01
   ```

ผู้ใช้ brown มีการล็อกอินที่คอนโซล ผู้ใช้ smith มี การล็อกอินจาก pseudo teletype line pts0 และผู้ใช้ jones มี การล็อกอินจาก tty0 ผู้ดูแลระบบของคุณสามารถตั้งค่าระบบของคุณเพื่อให้ คำสั่ง **finger** ทำงานในลักษณะที่แตกต่างอื่นได้ หากคุณพบ ปัญหาใดๆ เกี่ยวกับการใช้คำสั่ง **finger** โปรดติดต่อผู้ดูแลระบบ ของคุณ

การแสดงข้อมูลเกี่ยวกับผู้ใช้รายหนึ่งที่ล็อกอินเข้าสู่โฮสต์

ใช้โพรซีเดอร์นี้เพื่อดูข้อมูลเกี่ยวกับผู้ใช้ *เฉพาะ* ที่ ล็อกอินเข้าสู่รีโมตโฮสต์

เมื่อต้องการแสดงข้อมูลเกี่ยวกับผู้ใช้รายหนึ่งที่ล็อกอินเข้าสู่ รีโมตโฮสต์:

1. ล็อกอินเข้าสู่รีโมตโฮสต์ซึ่งคุณต้องการสื่อสาร
2. เมื่อต้องการแสดงข้อมูลเกี่ยวกับผู้ใช้ `brown` บนโฮสต์ `alcatraz` ให้พิมพ์:

   ```
   finger brown@alcatraz
   ```

   ข้อมูลคล้ายกับ ตัวอย่างต่อไปนี้แสดงขึ้น:

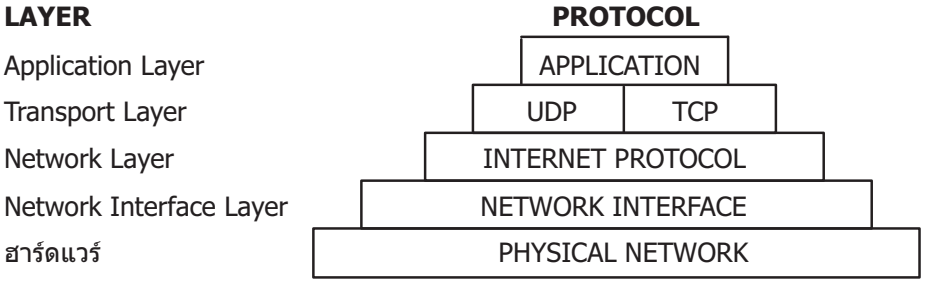
   ```
   Login name:  brown
   Directory:  /home/brown     Shell:  /home/bin/xinit -L -n Startup
   On since May 8 07:13:49 on console
   No Plan.
   ```

ผู้ดูแลระบบของคุณสามารถตั้งค่าระบบของคุณเพื่อให้ คำสั่ง **finger** ทำงานในลักษณะที่แตกต่างอื่นได้ หากคุณพบ ปัญหาใดๆ เกี่ยวกับการใช้คำสั่ง **finger** โปรดติดต่อผู้ดูแลระบบ ของคุณ

โปรโตคอล TCP/IP

โปรโตคอลคือชุดของกฎสำหรับรูปแบบข้อความและโพรซีเดอร์ ที่อนุญาตให้เครื่องและแอ็พพลิเคชันโปรแกรมแลกเปลี่ยนข้อมูล กฎเหล่านี้ต้องเป็นไปตามเครื่องแต่ละเครื่องที่เกี่ยวข้องในการสื่อสาร เพื่อให้โฮสต์การรับสามารถเข้าใจข้อความนั้นๆ *ชุด* TCP/IP ของโปรโตคอลสามารถเข้าใจได้ในรูปของ เลเยอร์ (หรือระดับ)

รูปภาพนี้แสดงเลเยอร์ของโปรโตคอล **TCP/IP** จากด้านบนสุด ประกอบด้วย Application Layer, Transport Layer, Network Layer, Network Interface Layer และ Hardware

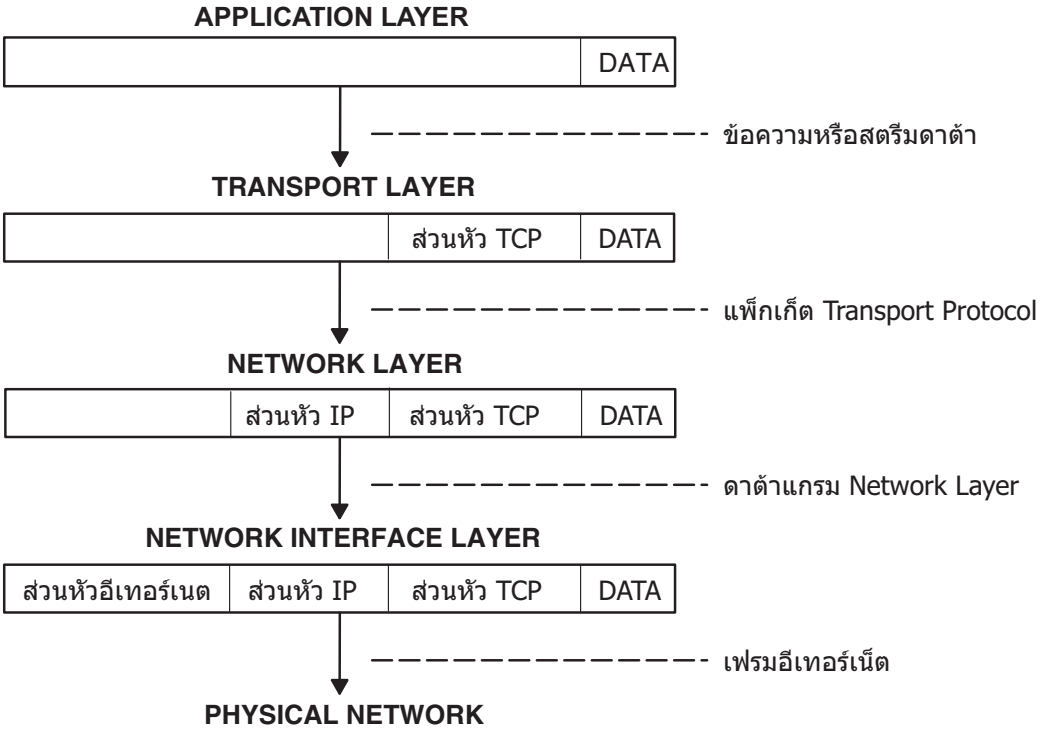


รูปที่ 4. ชุด TCP/IP ของโปรโตคอล

TCP/IP กำหนดวิธีการย้ายข้อมูลจากผู้ส่งไปยังผู้รับอย่างระมัดระวัง อันดับแรก แอ็พพลิเคชันโปรแกรมส่งข้อความ หรือสตรีม ข้อมูลของ Internet Transport Layer Protocols หนึ่งซึ่งอาจเป็น **User Datagram Protocol** (**UDP**) หรือ **Transmission Control Protocol** (**TCP**) โปรโตคอลเหล่านี้จะได้รับข้อมูลจาก แอ็พพลิเคชัน จากนั้นแบ่งข้อมูลออกเป็นส่วนย่อยๆ เรียกว่า *แพ็กเก็ต* เพิ่มแอดเดรสปลายทาง และจากนั้นส่งแพ็กเก็ตไปยัง เลเยอร์โปรโตคอลถัดไป คือเลเยอร์ Internet Network

เลเยอร์ Internet Network layer รวมแพ็กเก็ตให้อยู่ในดาตาแกรม **Internet Protocol** (**IP**) วางในส่วนหัวและส่วนท้ายของ ดาตาแกรม ตัดสินใจว่าจะส่งดาตาแกรมไปที่ใด (ไปยังปลายทางโดยตรง หรือไปยังเกตเวย์) และส่งดาตาแกรมต่อไปยัง เลเยอร์ Network Interface

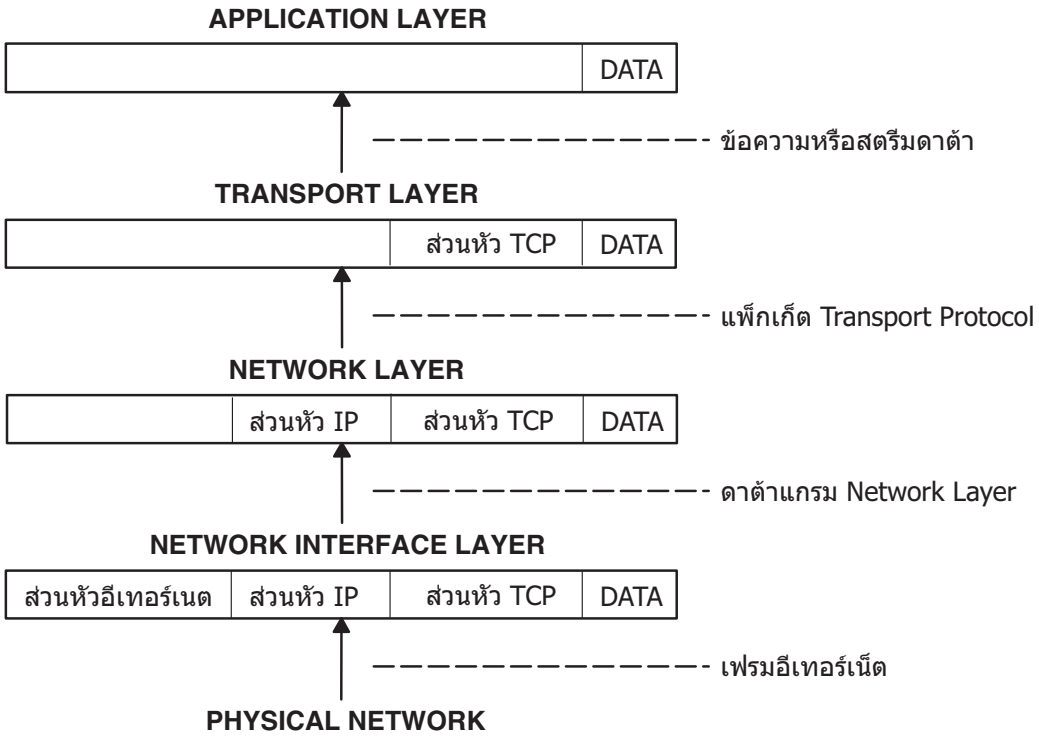
เลเยอร์ Network Interface ยอมรับดาตาแกรม **IP** และส่ง เป็นแบบ *เฟรม* บนฮาร์ดแวร์เน็ตเวิร์กที่เจาะจง เช่นเน็ตเวิร์ก Ethernet หรือ Token-Ring



รูปที่ 5. การย้ายข้อมูลจากแอ็พพลิเคชันผู้ส่งไปยัง โฮสต์ผู้รับ

รูปภาพนี้แสดงโฟลว์ของข้อมูลลงไปถึงเลเยอร์โปรโตคอล TCP/IP จากผู้ส่งไปยังโฮสต์

เฟรมที่โฮสต์ได้รับจะส่งไปยังเลเยอร์โปรโตคอลในทางตรงกันข้าม แต่ละเลเยอร์จะถอดข้อมูลส่วนข้องที่เกี่ยวข้องออก จนกระทั่ง เหลือเฉพาะข้อมูลที่ส่งกลับไปยังเลเยอร์แอ็พพลิเคชัน

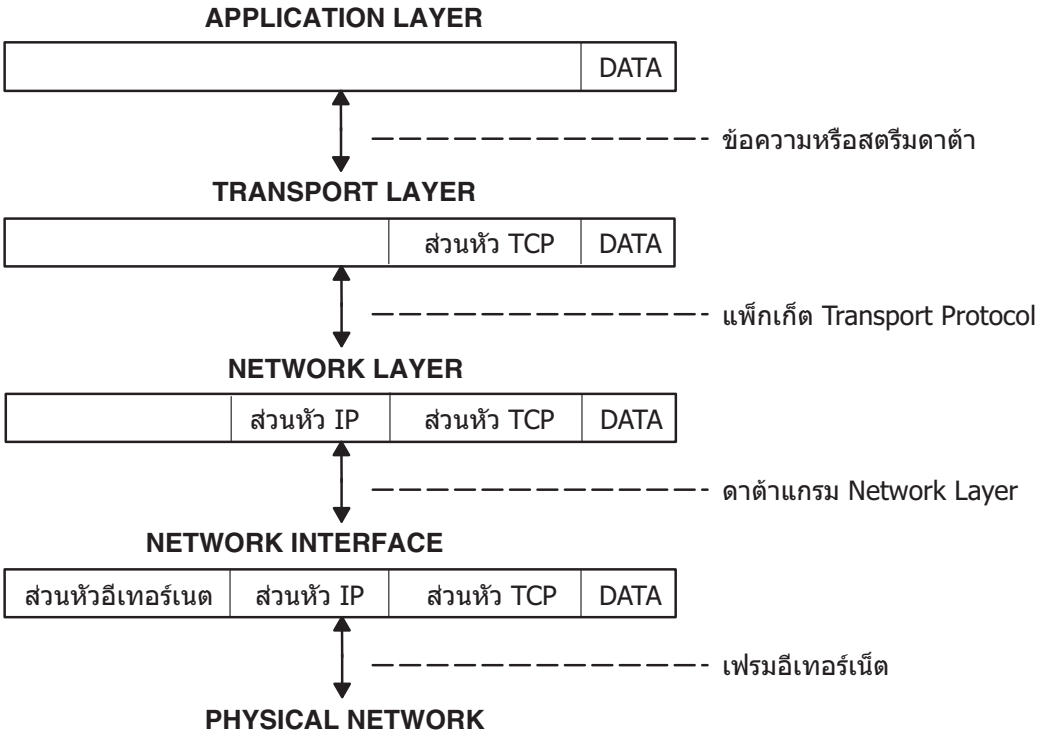


รูปที่ 6. การย้ายข้อมูลจากโฮสต์ไปยังแอ็พพลิเคชัน

รูปภาพนี้แสดงโฟลว์ของข้อมูลขึ้นไปถึงเลเยอร์โปรโตคอล **TCP/IP** จากโฮสต์ไปยังผู้ส่ง

เฟรมที่เลเยอร์ Network Interface ได้รับ (ในกรณีนี้ คืออะแด็ปเตอร์ Ethernet) เลเยอร์ Network Interface ถอดส่วนหัว Ethernet ออก และส่งดาตาแกรมขึ้นไปยังเลเยอร์ Network ในเลเยอร์ Network Internet Protocol ถอดส่วนหัว IP ออก และส่งแพ็กเก็ตขึ้นไปยังเลเยอร์ Transport ในเลเยอร์ Transport นั้น **TCP** (ใน กรณีนี้) จะถอดส่วนหัว **TCP** ออก และส่งข้อมูลขึ้น ไปยังเลเยอร์ Application

โฮสต์บนเน็ตเวิร์กจะส่งและรับข้อมูลพร้อมๆ กัน รูปที่ 7 ในหน้า 134 แสดงโฮสต์ ได้อย่างถูกต้องยิ่งขึ้นขณะที่สื่อสาร



หมายเหตุ: ส่วนหัวได้ถูกเพิ่มในแต่ละเลเยอร์โปโตคอลที่ข้อมูลถูกส่งและรับโดยโฮสต์

รูปที่ 7. การส่งและการรับ ข้อมูลโฮสต์

รูปภาพนี้แสดงการโฟลว์ข้อมูลทั้งสองทางผ่านเลเยอร์ **TCP/IP**

Internet Protocol (IP) version 6

Internet Protocol (**IP**) version 6 (**IPv6** หรือ IP*ng*) คือ **IP** รุ่นใหม่และถูกออกแบบ เพื่อให้เป็นขั้นตอนการพัฒนาจาก **IP** version 4 (**IPv4**)

ในขณะที่ **IPv4** ช่วยให้เกิดการพัฒนาขึ้นในโลกของอินเตอร์เน็ต แต่มันไม่สามารถรองรับความต้องการที่เพิ่มขึ้นในอนาคต เนื่องจากปัจจัยหลักสองประการ: พื้นที่แอดเดรสที่จำกัด และเส้นทางที่ซับซ้อน **IPv4** แอดเดรสแบบ 32 บิต ไม่มีความยืดหยุ่นพอสำหรับเส้นทางอินเตอร์เน็ตโกลบอล การพัฒนา ของ Classless InterDomain Routing (CIDR) ได้ช่วงต่ออายุ เส้นทางของ **IPv4** มานานหลายปี แต่ความพยายามที่จะจัดการเส้นทางให้ดีขึ้น ยังคงดำเนินต่อไป แม้ว่าเส้นทางของ **IPv4** สามารถขยายเพิ่มขึ้นได้ แต่อินเตอร์เน็ตก็ยังมีจำนวนเครือข่ายไม่เพียงพอ

Internet Engineering Task Force (IETF) ตระหนักดีว่า **IPv4** ไม่สามารถ สนับสนุนปรากฏการณ์การขยายตัวของอินเตอร์เน็ตได้ ดังนั้นกลุ่มทำงาน IETF IP*ng* จึงถูกก่อตั้งขึ้น ข้อเสนอได้ถูกจัดทำขึ้น **Simple Internet Protocol Plus** (**SIPP**) ถูกเลือกเป็นวิวัฒนาการก้าวกระโดด ในพัฒนาการของ IP ซึ่งเปลี่ยนชื่อเป็น IP*ng* และ RFC1883 ได้ สรุปข้อมูลในเดือนธันวาคมปี 1995

IPv6 ขยายจำนวนสูงสุดของอินเตอร์เน็ตแอดเดรสเพื่อรับมือกับ การเพิ่มขึ้นของผู้ใช้อินเตอร์เน็ต วิวัฒนาการที่เปลี่ยนแปลงจาก **IPv4**, **IPv6** มีข้อดี ที่ช่วยให้ IP แบบเก่าและใหม่สามารถใช้ร่วมกันได้ในเครือข่ายเดียวกัน การใช้งานร่วมกันนี้ช่วยให้การโอนย้ายจาก **IPv4** (แอดเดรสแบบ 32 บิต) เป็น **IPv6** (แอดเดรสแบบ 128 บิต) ได้อย่างราบรื่นในเครือข่ายที่ดำเนินการ

ภาพรวมนี้มุ่งหวังที่จะให้ผู้อ่านเข้าใจถึงแนวคิดทั่วไปของ โปรโตคอล IP*ng* สำหรับรายละเอียดเพิ่มเติม โปรดดู RFCs 2460, 2373, 2465, 1886, 2461, 2462 และ 2553

การรักษาความปลอดภัย จัดเตรียมข้อมูลด้านความปลอดภัย เกี่ยวกับโปรโตคอล **TCP/IP** รวมถึง **IPv6** สำหรับรายละเอียดเกี่ยวกับความปลอดภัยของ IP เวอร์ชัน 4 และ 6 โปรดดู ความปลอดภัยของอินเตอร์เน็ตโปรโตคอล

การจัดเส้นทางและการกำหนดแอดเดรสขยายของ IPv6:

IPv6 เพิ่มขนาด **IP** address จาก 32 บิตเป็น 128 บิต ดังนั้นให้การสนับสนุนระดับของลำดับชั้นการกำหนดแอดเดรสได้มากยิ่งขึ้น จำนวนโหนดที่กำหนด แอดเดรสได้เพิ่มมากยิ่งขึ้น และทำการกำหนดแอดเดรสอัตโนมัติได้ง่ายขึ้น

IPv6 มีแอดเดรสสามชนิดดังนี้:

ไอเท็ม	คำอธิบาย
unicast	แพ็กเก็ตส่งไปยังแอดเดรส unicast จะถูกนำส่งไปยังอินเตอร์เฟสที่ระบุ โดยแอดเดรสนั้น แอดเดรส unicast มีขอบเขตเฉพาะ: link-local, site-local, global. โดยจะมีแอดเดรส unicast พิเศษสองแอดเดรส: • ::/128 (แอดเดรสที่ไม่ระบุ) • ::1/128 (แอดเดรส loopback address)
multicast	แพ็กเก็ตส่งไปยังแอดเดรส multicast จะถูกนำส่งไปยังอินเตอร์เฟสทั้งหมด ที่ระบุโดยแอดเดรส แอดเดรส multicast ถูกระบุโดยใช้คำนำหน้า ff::/8 เหมือนกับแอดเดรส unicast แอดเดรส multicast มีขอบเขตที่คล้ายกัน คือ: node-local, link-local, site-local และ organization-local
anycast	แอดเดรส anycast คือแอดเดรสที่มีผู้ส่งรายเดียว หลาย listeners และมีผู้ตอบกลับรายเดียวเท่านั้น (โดยปกติคือผู้ที่อยู่ "ใกล้ที่สุด" ตาม การวัดระยะของโปรโตคอลการจัดเส้นทาง) ตัวอย่างเช่น หลายๆ เว็บเซิร์ฟเวอร์ จะ listen แอดเดรส anycast เมื่อมีการร้องขอถูกส่งไปยังแอดเดรส anycast จะมีเพียงหนึ่งรายที่ตอบกลับเท่านั้น แอดเดรส anycast ไม่สามารถแยกความแตกต่างจากแอดเดรส unicast แอดเดรส unicast จะกลายเป็นแอดเดรส anycast เมื่อมีการกำหนดค่าอินเตอร์เฟส มากกว่าหนึ่งอินเตอร์เฟสกับแอดเดรสนั้น

หมายเหตุ: ไม่มีแอดเดรสการกระจายใน **IPv6** ฟังก์ชัน เหล่านี้ถูกแทนที่ด้วยแอดเดรส multicast

คอนฟิกูเรชันอัตโนมัติของ IPv6:

กลไกหลักที่มีอยู่ซึ่งช่วยให้โหนดเริ่มทำงาน และสื่อสารกับโหนดอื่นผ่านเครือข่าย **IPv4** เป็นแบบฮาร์ดโค้ด **BOOTP** และ **DHCP**

IPv6 แนะนำแนวคิดของ *ขอบเขต* ให้กับ **IP** แอดเดรส หนึ่งในลิงก์โลคัล ซึ่งช่วยให้โฮสต์สร้างแอดเดรสที่ถูกต้อง จากคำนำหน้าลิงก์โลคัลที่กำหนดล่วงหน้าและตัวบ่งชี้โลคัล ตัวบ่งชี้โลคัลนี้ มักสืบทอดจากแอดเดรส medium access control (MAC) หรืออินเตอร์เฟสที่ถูกกำหนดคอนฟิก การใช้แอดเดรสนี้ โหนดสามารถสื่อสารกับ โฮสต์อื่นบนซับเน็ตเดียวกัน และสำหรับซับเน็ตที่แยกสมบูรณ์ อาจไม่จำเป็นต้อง กำหนดคอนฟิกแอดเดรสอื่น

แอดเดรสที่มีความหมาย IPv6:

สำหรับ **IPv4**, ส่วนที่มีความหมายและจดจำได้ในแอดเดรสคือ บอร์ดคาสต์ (มักเป็น 1 ทั้งหมดหรือ 0 ทั้งหมด) และคลาส (เช่น คลาส D คือมัลติคาสต์) แต่สำหรับ **IPv6** คำนำหน้าสามารถใช้กำหนด *ขอบเขต* ได้อย่างรวดเร็ว (เช่น link-local), มัลติคาสต์ กับยูนิคาสต์ และกลไกการกำหนด (ตามผู้ให้บริการหรือตามภูมิศาสตร์)

ข้อมูลเส้นทางอาจถูกโหลดอย่างเปิดเผยลงในบิตบนของแอดเดรสด้วย แต่ประเด็นนี้ยังไม่มีการสรุปผลโดย IETF (สำหรับแอดเดรสตามผู้ให้บริการ ข้อมูลเส้นทางจะปรากฏอยู่ในแอดเดรส)

การตรวจจับแอดเดรสซ้ำซ้อนของ IPv6:

เมื่ออินเตอร์เฟสถูกเตรียมข้อมูลหรือเตรียมข้อมูลใหม่ ซึ่งใช้การกำหนดคอนฟิกอัตโนมัติ ให้กับลิงก์โลคัลที่เกี่ยวข้องชั่วคราว ด้วยอินเตอร์เฟส (แอดเดรส ยังไม่ถูกกำหนดให้กับอินเตอร์เฟสตามปกติ) ณ จุดนี้ อินเตอร์เฟสจะรวมโหนดทั้งหมดและ กลุ่มมัลติคาสต์ของโหนดที่เรียกร้อง และส่งข้อความสำรวจใกล้เคียงไปยังกลุ่มเหล่านี้ โดยใช้มัลติคาสต์แอดเดรส โหนด สามารถกำหนดแอดเดรสลิงก์โลคัลจำเพาะที่ถูกกำหนดก่อนหน้านี้ และเลือกแอดเดรสสำรอง

สิ่งนี้จะลบแอดเดรสเดียวกันในสองแอดเดรสที่ต่างกัน บนลิงก์เดียวกันโดยไม่ได้ตั้งใจ (ซึ่งเป็นไปได้ที่จะสร้างแอดเดรส ขอบเขตโกลบอล สำหรับโหนดที่ไม่อยู่บนลิงก์เดียวกัน)

การค้นหาเครื่องใกล้เคียง/การกำหนดค่าแอดเดรสอัตโนมัติแบบไม่เก็บค่าสถานะ:

Neighbor Discovery Protocol (**NDP**) สำหรับ **IPv6** ถูกใช้ โดยโหนด (โฮสต์และเราเตอร์) เพื่อใช้พิจารณาแอดเดรสในเลเยอร์ การลิงก์สำหรับ เครื่องใกล้เคียงที่รู้จัก ที่อยู่บนลิงก์ที่เชื่อมต่ออยู่ และดูแลรักษาตารางการจัดเส้นทาง ต่อหนึ่งปลายทางสำหรับ การเชื่อมต่อที่แอ็คทีฟ **IPv6** กำหนดทั้งกลไกการกำหนดค่า แอดเดรสอัตโนมัติแบบเก็บค่าสถานะ และแบบไม่เก็บค่าสถานะ *การกำหนดค่าอัตโนมัติแบบไม่เก็บค่าสถานะ* ไม่จำเป็น ต้องทำการกำหนดค่าโฮสต์เอง ถ้ามีที่ต้องกำหนดอย่างน้อยที่สุดคือ การกำหนดค่าเราเตอร์ และไม่สามารถมีเซิร์ฟเวอร์เพิ่ม

โฮสต์ยังใช้ **NDP** เพื่อค้นหาเราเตอร์ที่อยู่ใกล้เคียงที่ต้องการ ส่งต่อแพ็กเก็ตในนามของเราเตอร์ และตรวจหาแอดเดรสในเล เยอร์การลิงก์ที่เปลี่ยนแปลง **NDP** ใช้ **Internet Control Message Protocol** (**ICMP**) เวอร์ชัน 6 ที่มี ชนิดข้อความเฉพาะของตน เอง โดยทั่วไปโปรโตคอล **IPv6** Neighbor Discovery สอดคล้องตาม **IPv4 Address Resolution Protocol** (**ARP**), ICMP Router Discovery (RDISC) และ **ICMP** Redirect (ICMPv4) รวมกัน แต่มีการปรับปรุงให้ดียิ่งขึ้นหลายอย่างเหนือกว่าของ โปรโตคอล **IPv4**

กลไกแบบไม่เก็บค่าสถานะยอมให้โฮสต์สร้างแอดเดรสของตนเองโดยใช้ ข้อมูลที่มีแบบโลคัล และข้อมูลที่ได้จากเราเตอร์ รวมกัน เราเตอร์เสนอคำนำหน้าที่ใช้ระบุซับเน็ตที่เชื่อมโยง กับลิงก์ ในขณะที่โฮสต์จะสร้างโทเค็นการอินเตอร์เฟสที่ระบุถึงอิน เตอร์เฟส บนซับเน็ตโดยเฉพาะ แอดเดรสถูกจัดรูปแบบโดยการรวมของทั้งสอง หาก ไม่มีเราเตอร์ โฮสต์จะสามารถสร้างได้ เฉพาะแอดเดรสการลิงก์โลคัลเท่านั้น อย่างไรก็ตาม แอดเดรสการลิงก์โลคัลก็พอเพียงสำหรับการอนุญาตให้ทำการสื่อสาร ระหว่างโหนด ที่เชื่อมต่อบนลิงก์เดียวกัน

การทำให้จัดเส้นทางง่ายขึ้น:

เมื่อต้องการให้จัดเส้นทางได้ง่ายขึ้น **IPv6** addresses จะถูกพิจารณา ในสองส่วนคือ : คำนำหน้า และ ID ซึ่งคล้ายกับว่าจะ เหมือนกับการแตกย่อย **IPv4** net-host address แต่มีข้อดีสองข้อ

ไอเท็ม	**คำอธิบาย**
ไม่มีคลาส	ไม่มีจำนวนบิตคงที่สำหรับคำนำหน้าหรือ ID ซึ่งช่วยลด การสูญเสียเนื่องจากการจัดสรรมากเกินไป
การซ้อนภายใน	จำนวนการแบ่งอิสระสามารถถูกนำไปใช้โดยการพิจารณาจำนวนบิต ที่ต่างกันเป็นคำนำหน้า

กรณีที่ 1

128 บิต
แอดเดรสโหนด

กรณีที่ 2

ไอเท็ม	คำอธิบาย
n บิต	128-*n* บิต
คำนำหน้าซับเน็ต	ID อินเตอร์เฟส

กรณีที่ 3:

ไอเท็ม	คำอธิบาย	
n บิต	80-*n* บิต	48 บิต
คำนำหน้า Subscriber	ID ซับเน็ต	ID อินเตอร์เฟส

กรณีที่ 4:

ไอเท็ม	คำอธิบาย		
s บิต	*n* บิต	*m* บิต	128-*s*-*n*-*m* บิต
คำนำหน้า Subscribe	ID พื้นที่	ID ซับเน็ต	ID อินเตอร์เฟส

โดยทั่วไป IPv4 จะไม่สามารถเกินกรณีที่ 3 แม้กับ Variable Length Subnet Mask (VLSM เป็นค่าเฉลี่ยของการจัดสรรรีซอร์ส **IP** addressing ให้แก่ซับเน็ต ตามความต้องการใช้แต่ละรายการมากกว่ากฎที่ใช้ทั้งเน็ตเวิร์กโดยทั่วไปบางกฎ) ซึ่งเทียบได้กับการส่งความยาวแอดเดรสที่สั้นลงเหมือนกับนิยาม ของคำนำหน้าความยาวผันแปรได้ แต่ถึงอย่างไรก็ไม่มีประโยชน์ใด

การทำรูปแบบส่วนหัวให้เข้าใจง่าย:

IPv6 ทำให้เข้าใจส่วนหัว **IP** ง่ายขึ้นโดยการเอาทั้งหมดออก หรือโดยการย้ายฟิลด์บางส่วนไปยังส่วนหัวที่ขยายเพิ่มที่พบในส่วนหัว **IPv4** โดยกำหนดรูปแบบที่มีความยืดหยุ่นมากขึ้นสำหรับใช้กับข้อมูลเพิ่มเติม (ส่วนหัว ที่ขยายเพิ่ม)

โดยเฉพาะ หมายเหตุสิ่งที่ไม่มี:

- ความยาวส่วนหัว (ความยาวคงที่)
- การระบุ
- แฟล็ก
- ออฟเซ็ตแฟรกเมนต์ (ย้ายไปไว้ในส่วนหัวที่ขยายเพิ่มของการแตกแฟรกเมนต์)
- ค่าเช็กซัมส่วนหัว (โปรโตคอลระดับบน หรือส่วนหัวที่ขยายเพิ่มด้านความปลอดภัยจัดการ ในส่วน data integrity)

ตารางที่ 53. ส่วนหัว IPv4

ไอเท็ม	คำอธิบาย	คำอธิบาย	คำอธิบาย	คำอธิบาย
Version	IHL	ชนิดเซอร์วิส	ความยาวทั้งหมด	
Identification	Identification	Identification	แฟล็ก	ออฟเซ็ตแฟรกเมนต์
Time to Live	Time to Live	Protocol	เช็กซัมส่วนหัว	เช็กซัมส่วนหัว
แอดเดรสต้นทาง	แอดเดรสต้นทาง	แอดเดรสต้นทาง	แอดเดรสต้นทาง	แอดเดรสต้นทาง
แอดเดรสปลายทาง	แอดเดรสปลายทาง	แอดเดรสปลายทาง	แอดเดรสปลายทาง	แอดเดรสปลายทาง
อ็อพชัน	อ็อพชัน	อ็อพชัน	อ็อพชัน	เครื่องรอง

ตารางที่ 54. ส่วนหัว IPv6

ไอเท็ม	คำอธิบาย	คำอธิบาย	คำอธิบาย	คำอธิบาย
Version	Prio		เลเบลไฟลว์	
ความยาว Payload	ความยาว Payload	ความยาว Payload	ส่วนหัวถัดไป	ขีดจำกัดฮ็อบ
แอดเดรสต้นทาง	แอดเดรสต้นทาง	แอดเดรสต้นทาง	แอดเดรสต้นทาง	แอดเดรสต้นทาง
แอดเดรสปลายทาง	แอดเดรสปลายทาง	แอดเดรสปลายทาง	แอดเดรสปลายทาง	แอดเดรสปลายทาง

IP*ng* มีกลไกอ็อพชันที่ปรับปรุงเพิ่มเหนือกว่า **IPv4** อ็อพชัน **IPv6** อยู่ในส่วนหัวที่ขยายเพิ่มแยกต่างกัน ที่อยู่ในตำแหน่งระหว่างส่วนหัว **IPv6** และส่วนหัวเลเยอร์การนำส่งข้อมูลในแพ็กเก็ต ส่วนขยายที่ขยายเพิ่มส่วนใหญ่จะไม่ถูก ตรวจสอบหรือประมวลผลโดยเราเตอร์ที่อยู่ตามเส้นทางการนำส่งแพ็กเก็ตจนกว่าจะ ไปถึงที่หมายปลายทาง กลไกการทำงานนี้ช่วยให้ปรับปรุงประสิทธิ การทำงานของเราเตอร์เป็นอย่างมากสำหรับแพ็กเก็ตที่มีอ็อพชัน ใน **IPv4** การมี อ็อพชันใดๆ อยู่จำเป็นที่เราเตอร์จะต้องตรวจสอบอ็อพชันทั้งหมด

การปรับปรุงอีกข้อหนึ่ง ที่แตกต่างจากอ็อพชัน **IPv4** ก็คือส่วนหัวที่ขยายเพิ่ม **IPv6** ที่สามารถมีความยาวเท่าใดก็ได้ และจำนวนอ็อพชันทั้งหมดที่เก็บ ในหนึ่งแพ็กเก็ตไม่ถูกจำกัดแค่ 40 ไบต์ คุณลักษณะนี้ บวกกับรูปแบบ ที่ถูกประมวลผล ยอมให้อ็อพชัน **IPv6** ถูกใช้สำหรับฟังก์ชันที่ ใช้ไม่ได้ใน **IPv4** เช่นอ็อพชันการพิสูจน์ตัวตน **IPv6** และ Security Encapsulation

เมื่อต้องการปรับปรุงสิทธิภาพการทำงานในการจัดการส่วนตัวของอ็อพชันที่ตามมาและ โปรโตคอลการนำส่งข้อมูลที่ตามหลัง อ็อพชัน IPv6 มักเป็นเลขจำนวนเต็มของเลขฐานแปด แปดหลักเพื่อคงการจัดตำแหน่งนี้สำหรับส่วนหัวที่ตามมา

ด้วยการใช้ส่วนหัวที่ขยายเพิ่มแทนตัวระบุโปรโตคอลและฟิลด์ อ็อพชัน จะช่วยให้ส่วนขยายที่กำหนดใหม่สามารถนำมารวมเข้าได้ง่ายยิ่งขึ้น

ข้อมูลจำเพาะ ปัจจุบัน กำหนดส่วนหัวของส่วนขยายในวิธีต่อไปนี้:

- อ็อพชัน Hop-by-hop ที่ใช้กับแต่ละ hop (เราเตอร์) ในพาธ
- ส่วนหัวการจัดเส้นทางสำหรับการจัดเส้นทางต้นทางแบบ loose/strict (ใช้ไม่บ่อย)
- แฟรกเมนต์กำหนดแพ็กเก็ตเป็นหนึ่งแฟรกเมนต์ และมีข้อมูลเกี่ยวกับ แฟรกเมนต์ (เราเตอร์ **IPv6** จะไม่แตกแฟรกเมนต์)
- การพิสูจน์ตัวตน (ดูที่การรักษาความปลอดภัย **TCP/IP** ใน *การรักษาความปลอดภัย*
- การเข้ารหัส (ดูที่การรักษาความปลอดภัย **TCP/IP** ใน *การรักษาความปลอดภัย*

• อ็อพชันปลายทางสำหรับโหนดปลายทาง (ถูกข้ามโดยเราเตอร์)

การควบคุมคุณภาพบริการ/ปริมาณรับส่งข้อมูลที่ปรับปรุง:

ขณะที่คุณภาพบริการสามารถควบคุมโดยใช้โปรโตคอล การควบคุมเช่น **RSVP** นั้น **IPv6** ยังมีนิยามการกำหนดระดับความสำคัญชัดเจน สำหรับแพ็กเก็ตโดยการใช้ฟิลด์ระดับความสำคัญในส่วนหัว **IP**

โหนดสามารถตั้งค่านี้เพื่อระบุระดับความสำคัญที่สัมพันธ์กันของแพ็กเก็ต เฉพาะ หรือชุดของแพ็กเก็ต ซึ่งสามารถใช้โดยโหนด เราเตอร์อย่างน้อยหนึ่ง เราเตอร์ หรือปลายทางเพื่อทำการตัดสินใจที่เกี่ยวกับแพ็กเก็ต (นั่นคือ ตัดทิ้งหรือไม่)

IPv6 ระบุระดับความสำคัญสองชนิด คือสำหรับการรับส่งข้อมูลที่ควบคุม ความหนาแน่น และสำหรับการรับส่งข้อมูลที่ไม่ควบคุมความหนาแน่น ไม่มีการใช้การจัดลำดับที่ สัมพันธ์กันระหว่างสองชนิดนี้

การรับส่งข้อมูลที่ควบคุมความหนาแน่น ถูกกำหนดเป็นการรับส่งข้อมูลที่ตอบสนอง ต่อความหนาแน่นผ่านทางอัลกอริทึมแบบ "back-off" หรือการจำกัดอื่นๆ ระดับความสำคัญสำหรับการรับส่งข้อมูลที่ควบคุมความหนาแน่น ได้แก่:

ไอเท็ม	คำอธิบาย
0	การรับส่งข้อมูลที่ไม่มีการกำหนดลักษณะพิเศษ
1	การรับส่งข้อมูลที่มี "การกรอง" (ตัวอย่างเช่น netnews)
2	การถ่ายโอนข้อมูลที่ไม่ต้องมีการโต้ตอบ (ตัวอย่างเช่น เมล)
3	(สำรอง)
4	การถ่ายโอนจำนวนมากที่ต้องมีการโต้ตอบ (ตัวอย่างเช่น **FTP**)
5	(สำรอง)
6	การรับส่งข้อมูลแบบไม่มีการโต้ตอบ (ตัวอย่างเช่น Telnet)
7	การรับส่งข้อมูลการควบคุม (ตัวอย่างเช่น โปรโตคอลการจัดเส้นทาง)

การรับส่งข้อมูลที่ไม่ควบคุมความหนาแน่น ถูกกำหนดเป็นการรับส่งข้อมูลที่ตอบสนอง ต่อความหนาแน่นโดยการตัดทิ้ง (หรือเพียงการไม่ส่งใหม่) แพ็กเก็ต เช่นวิดีโอ เสียง หรือการรับส่งข้อมูล ณ เวลาจริงอื่นๆ ระดับที่แน่นอนไม่ถูกกำหนดแก่ตัวอย่าง แต่การจัดลำดับจะคล้ายกับของการรับส่งข้อมูลที่ควบคุมความหนาแน่น:

- ค่าต่ำสุดที่ต้นทางต้องการละทิ้งมากที่สุดควรถูกใช้ สำหรับการรับส่งข้อมูล
- ค่าสูงสุดที่ต้นทางต้องการละทิ้งน้อยที่จะควร ถูกใช้สำหรับการรับส่งข้อมูล

การควบคุมระดับความสำคัญนี้ใช้ได้กับการรับส่งข้อมูลจากแอดเดรสต้นทางที่เจาะจง เท่านั้น การรับส่งข้อมูลการควบคุมจากแอดเดรสหนึ่งไม่ได้มีระดับความสำคัญที่สูงกว่าการรับส่ง ข้อมูลจากแอดเดรสอื่นจำนวนมากที่มีการโต้ตอบอย่างเด่นชัด

เลเบลการไหล:

ภายนอกลำดับความสำคัญพื้นฐานของทราฟฟิก **IPv6** กำหนด กลไกสำหรับระบุการไหลของแพ็กเก็ตจำเพาะเอาไว้ในความหมายของ **IPv6** *การไหล* ถูกกำหนดตามลำดับแพ็กเก็ตที่ส่งจากแหล่งข้อมูลจำเพาะ ไปยังปลายทางจำเพาะ (unicast หรือ multicast) สำหรับแหล่งข้อมูลที่ต้องการ การจัดการพิเศษโดยเราเตอร์แบบ intervening

ตัวบ่งชี้การไหลนี้สามารถใช้สำหรับควบคุมความสำคัญ แต่อาจใช้สำหรับ ตัวควบคุมอื่นได้

เลเบลการไหลถูกเลือกแบบสุ่ม และไม่ได้บ่งชี้ลักษณะเฉพาะใดๆ ของทราฟฟิก นอกจากการไหลที่ดำเนินการ ซึ่งหมายความว่า เราเตอร์ไม่สามารถกำหนดว่าแพ็กเก็ตเป็นแบบจำเพาะได้จากการตรวจสอบ เลเบลการไหล อย่างไรก็ตาม ยังสามารถกำหนดว่าเป็นส่วนของแพ็กเก็ตลำดับเดียวกัน กับแพ็กเก็ตสุดท้ายที่มีเลเบล

หมายเหตุ: จนกว่า **IPv6** อยู่ในการใช้ทั่วไป เลเบลการไหล เป็นการทดลอง ผู้ใช้และตัวควบคุมเลเบลโฟลว์ที่เกี่ยวข้อง ที่ยังไม่ได้กำหนด หรือไม่กำหนดมาตรฐาน

การสร้างช่องสัญญาณ IPv6:

การสร้างช่องสัญญาณมีวิธีในการใช้โครงสร้างพื้นฐานการจัดเส้นทาง **IPv4** ที่มีอยู่แล้วเพื่อจัดการการรับส่งข้อมูล **IPv6**

กุญแจสู่การเปลี่ยนเป็น **IPv6** ได้สำเร็จคือความเข้ากันได้กับ พื้นฐานของโฮสต์และเราเตอร์ **IPv4** ที่ติดตั้งซึ่งมีอยู่แล้ว ความเข้ากันได้ในการดูแลรักษา กับ **IPv4** ขณะที่มีการปรับใช้ **IPv6** ให้มีประสิทธิภาพมากขึ้นสำหรับงานการเปลี่ยน อินเตอร์เน็ตเป็น **IPv6** ขณะที่โครงสร้างพื้นฐาน **IPv6** กำลังถูก นำไปใช้โครงสร้างพื้นฐานการจัดเส้นทาง **IPv4** ที่มีอยู่แล้วยังสามารถใช้งานได้ และสามารถใช้เพื่อดำเนินการรับส่งข้อมูล **IPv6**

โฮสต์และเราเตอร์ **IPv6** หรือ **IPv4** สามารถทำช่องสัญญาณดาตาแกรม **IPv6** เหนือขอบเขตของทอโพโลยีการจัดเส้นทาง **IPv4** โดยการเอ็นแคปซูเลตให้อยู่ภายในแพ็กเก็ต **IPv4** การสร้างช่องสัญญาณสามารถใช้ได้หลายวิธี:

ไอเท็ม	คำอธิบาย
Router-to-Router	เราเตอร์ **IPv6** หรือ **IPv4** ถูกเชื่อมต่อถึงกันโดยโครงสร้างพื้นฐาน **IPv4** สามารถสร้างช่องสัญญาณแพ็กเก็ต **IPv6** ระหว่างกัน ในกรณีนี้ ช่องสัญญาณ ขยายออกหนึ่งเซกเมนต์ของพาธ end-to-end ที่แพ็กเก็ต **IPv6** ใช้
Host-to-Router	**โฮสต์ IPv6** หรือ **IPv4** สามารถสร้างช่องสัญญาณแพ็กเก็ต **IPv6** ไปยัง เราเตอร์ **IPv6** หรือ **IPv4** ระหว่างกลางที่สามารถไปถึงได้โดยผ่าน โครงสร้างพื้นฐาน **IPv4** ช่องสัญญาณชนิดนี้ขยายเซกเมนต์แรก ของพาธ end-to-end ของแพ็กเก็ต
Host-to-Host	**โฮสต์ IPv6** หรือ **IPv4** ที่สามารถเชื่อมต่อถึงกันโดยโครงสร้างพื้นฐาน **IPv4** สามารถสร้างช่องสัญญาณแพ็กเก็ต **IPv6** ระหว่างกัน ในกรณีนี้ ช่องสัญญาณ ขยายออกทั้งพาธ end-to-end ที่แพ็กเก็ตใช้
Router-to-Host	เราเตอร์ **IPv6/IPv4** สามารถสร้างช่องสัญญาณแพ็กเก็ต **IPv6** ไปยัง โฮสต์ **IPv6** หรือ **IPv4** ปลายทางสุดท้าย ช่องสัญญาณนี้ขยายเฉพาะ เซกเมนต์สุดท้ายของพาธ end-to-end เท่านั้น

เทคนิคการสร้างช่องสัญญาณโดยปกติถูกจัดหมวดหมู่ตามกลไก ที่โหนดการเอ็นแคปซูเลตใช้พิจารณาแอดเดรสของโหนดที่ท้ายช่องสัญญาณ สำหรับวิธี router-to-router หรือ host-to-router แพ็กเก็ต **IPv6** กำลังถูกสร้างช่องช่องสัญญาณไปยังเราเตอร์ในวิธี host-to-host หรือ router-to-host นั้น แพ็กเก็ต **IPv6** จะถูกสร้างช่องสัญญาณตลอดทางไปยังปลายทางสุดท้าย

โหนดรายการของช่องสัญญาณ (โหนดที่กำลังเอ็นแคปซูเลต) จะสร้างส่วนหัว **IPv4** การเอ็นแคปซูเลต และส่งแพ็กเก็ตที่เอ็นแคปซูเลต โหนดออกของช่องสัญญาณ (โหนด ที่ยกเลิกการเอ็นแคปซูเลต) ได้รับแพ็กเก็ตที่เอ็นแคปซูเลต ให้เอาส่วนหัว **IPv4** ออก อัพเดต ส่วนหัว **IPv6** และประมวลผลแพ็กเก็ต **IPv6** ที่ได้รับ อย่างไรก็ตาม โหนดการเอ็นแคปซูเลตจำเป็นต้องดูแลข้อมูลสถานะแบบ soft สำหรับแต่ละช่องสัญญาณ เช่น maximum transmission unit (MTU) ของช่องสัญญาณ เพื่อประมวลผลแพ็กเก็ต **IPv6** ที่ส่งต่อไปในช่องสัญญาณ

ช่องสัญญาณมีสองชนิดใน **IPv6**:

ช่องสัญญาณอัตโนมัติ

ช่องสัญญาณอัตโนมัติถูกกำหนดค่าโดยใช้ข้อมูล **IPv4** address ที่ฝังอยู่ใน **IPv6** address คือ **IPv6** address ของโฮสต์ปลายทางที่มีข้อมูลเกี่ยวกับ **IPv4** address ที่แพ็กเก็ตควร ถูกสร้างช่องสัญญาณ

ช่องสัญญาณที่กำหนดค่า

ช่องสัญญาณที่กำหนดค่าต้องถูกกำหนดค่าด้วยตนเอง ช่องสัญญาณเหล่านี้ใช้ เมื่อใช้ **IPv6** addresses ที่ไม่มีข้อมูล **IPv4** ฝังตัว **IPv6** และ **IPv4** ของจุดปลายของช่องสัญญาณต้อง ถูกระบุ

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการกำหนดค่าอัตโนมัติและช่องสัญญาณที่กำหนดค่า ดูที่ “การตั้งค่า tunneling ใน IPv6” ในหน้า 149

การสนับสนุน IPv6 multihomed link-local และ site-local:

โฮสต์สามารถถูกกำนดให้มีมากกว่าหนึ่งอินเตอร์เฟสโฮสต์ที่มี อินเตอร์เฟสที่แอ็คทีฟมากกว่าสองจะถูกเรียกว่า multihomed แต่ละอินเตอร์เฟสมีแอดเดรส link-local ที่เชื่อมโยงกับอินเตอร์เฟส

แอดเดรส Link-local เพียงพอสำหรับการอนุญาตให้ทำการสื่อสารระหว่างโหนด ที่เชื่อมต่อบนลิงก์เดียวกัน

โฮสต์ multihomed มีแอดเดรส link-local ที่เชื่อมโยงอยู่อย่างน้อยสองแอดเดรส การนำ AIX **IPv6** ไปใช้มี 4 อ็อพชันให้ใช้จัดการการกำหนดแอดเดรส link-layer บนโฮสต์ multihomed อ็อพชัน 1 เป็นค่าดีฟอลต์

ไอเท็ม	คำอธิบาย
อ็อพชัน 0	ไม่มีการดำเนินการ multihomed การส่งข้อมูลจะส่งออกจากอินเตอร์เฟส link-local แรก เมื่อ **Neighbor Discovery Protocol** (**NDP**) ต้องดำเนินการกำหนดแอดเดรส โปรโตคอล multicasts ข้อความ Neighbor Solicitation ออกไปยังแต่ละอินเตอร์เฟสที่มีแอดเดรสโลคัลลิงก์ถูกกำหนด NDP เข้าคิวแพ็กเก็ตข้อมูล จนกระทั่งได้รับข้อความ Neighbor Advertisement แรก จากนั้น แพ็กเก็ตข้อมูลจะถูกส่งออกทางลิงก์นี้
อ็อพชัน 1	เมื่อ **NDP** ต้องดำเนินการกำหนดแอดเดรส นั่นคือ เมื่อ ส่งแพ็กเก็ตข้อมูลไปยังปลายทาง และข้อมูล link-layer สำหรับ hop ถัดไปไม่อยู่ใน Neighbor Cache โปรโตคอลต้อง multicasts ข้อความ Neighbor Solicitation ออกไปยังแต่ละอินเตอร์เฟสที่มีแอดเดรส link-local ถูกกำหนด จากนั้น **NDP** เข้าคิวแพ็กเก็ตข้อมูลจนกระทั่งได้รับข้อมูล link-layer จากนั้น **NDP** จะรอจนกระทั่งได้รับการตอบกลับสำหรับแต่ละอินเตอร์เฟส เพื่อรับประกันว่า แพ็กเก็ตข้อมูลถูกส่งบนอินเตอร์เฟสขาออกที่เหมาะสม ถ้า **NDP** ไม่มีการรอ แต่ตอบกลับไปยัง Neighbor Advertisement แรกที่ได้รับ เป็นไปได้ที่ โปรโตคอลอาจจะส่งแพ็กเก็ตข้อมูลออกไปบนลิงก์ที่ไม่เชื่อมโยงกับ แอดเดรสของต้นทางแพ็กเก็ต เนื่องจาก **NDP** ต้องรอ จะเกิดการหน่วงเวลาขึ้น ในแพ็กเก็ตแรกที่กำลังจะถูกส่ง อย่างไรก็ตาม การหน่วงเวลาจะเกิดขึ้นเพื่อรอ การตอบกลับแรก
อ็อพชัน 2	อนุญาตให้มีการดำเนินการ Multihomed แต่การส่งแพ็กเก็ตข้อมูลถูก จำกัดตามอินเตอร์เฟสที่ระบุโดย main_if6 เมื่อ **NDP** ต้องดำเนินการ กำหนดแอดเดรส โปรโตคอลจะ multicasts ข้อความ Neighbor Solicitation ออกไปยังแต่ละ อินเตอร์เฟสที่มีแอดเดรส link-local ถูกกำหนด จากนั้นจะรอข้อความ Neighbor Advertisement จากอินเตอร์เฟสที่ระบุโดย main_if6 (ดูที่คำสั่ง **no**) เมื่อได้รับการตอบกลับจากอินเตอร์เฟสนี้ แพ็กเก็ตข้อมูลจะถูกส่งออกไป บนลิงก์นี้
อ็อพชัน 3	อนุญาตให้มีการดำเนินการ Multihomed แต่การส่งแพ็กเก็ตข้อมูล จะถูกจำกัดตามอินเตอร์เฟสที่ระบุโดย main_if6 และแอดเดรส site-local จะถูกจัดเส้นทางสำหรับอินเตอร์เฟสที่ระบุโดย main_site6 (ดูที่คำสั่ง **no**) เท่านั้น NDP ดำเนินงานเหมือนกับที่ทำสำหรับอ็อพชัน 2 สำหรับแอ็พพลิเคชันที่จัดเส้นทาง แพ็กเก็ตข้อมูลโดยใช้แอดเดรส site-local บนโฮสต์ multihomed เฉพาะแอดเดรส site-local ที่ระบุโดย main_site6 เท่านั้นที่ถูกใช้

การอัพเกรดเป็น IPv6 ด้วย IPv4 ที่ตั้งค่าคอนฟิก:

สถานการณ์จำลองนี้นำคุณผ่านขั้นตอนการอัพเกรดด้วยตนเองจาก **IPv4** เป็น **IPv6**

เครือข่ายที่ใช้ในตัวอย่างนี้ประกอบด้วยเราเตอร์หนึ่งเครื่องและ subnets สองเครื่อง มีอยู่สองโฮสต์บนแต่ละ subnet คือ: เราเตอร์ และโฮสต์อื่น คุณจะอัพเกรดแต่ละเครื่องบนเครือข่ายนี้เป็น **IPv6** ที่ตอนท้ายของ สถานการณ์จำลอง เราเตอร์จะใช้คำเติมหน้า `3ffe:0:0:aaaa::/64` บน อินเตอร์เฟสเครือข่าย `en0` และคำเติมหน้า `3ffe:0:0:bbbb::/64` บนอินเตอร์เฟสเครือข่าย `en1` อันดับแรก คุณจะตั้งค่าคอนฟิกเครื่องเพื่อ สนับสนุน **IPv6** เป็นการชั่วคราวเพื่อให้คุณสามารถทดสอบได้ จากนั้น คุณจะ ตั้งค่าคอนฟิกเครื่องเพื่อให้พร้อมจะเป็น **IPv6** ณ เวลาบูต

ถ้าคุณกำลังรันระบบปฏิบัติการ AIX และไม่ได้กำหนดคอนฟิกค่าติดตั้ง **IPv4** ไว้โปรดดู “การอัพเกรดเป็น IPv6 ด้วย IPv4 ที่ไม่ได้กำหนดคอนฟิก” ในหน้า 143

สิ่งที่ต้อง พิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ได้ผ่านการทดสอบโดยใช้เวอร์ชันเฉพาะของ AIX ผลลัพธ์ที่คุณได้อาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชันและระดับ AIX ของคุณ

ขั้นตอนที่ 1: ตั้งค่าโฮสต์สำหรับ IPv6

บนโฮสต์บนทั้งสอง subnets ให้ทำดังต่อไปนี้:

1. ตรวจสอบให้แน่ใจว่ามีการตั้งค่าคอนฟิก **IPv4** โดยการพิมพ์คำสั่งต่อไปนี้:

   ```
   netstat -ni
   ```

   ผลลัพธ์ของคุณ ควรมีลักษณะคล้ายกับตัวอย่างต่อไปนี้:

   ```
   Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
   en0   1500  link#2      0.6.29.4.55.ec      279393     0     2510     0     0
   en0   1500  9.3.230.64  9.3.230.117         279393     0     2510     0     0
   lo0   16896 link#1                            913     0      919     0     0
   lo0   16896 127         127.0.0.1              913     0      919     0     0
   lo0   16896 ::1                               913     0      919     0     0
   ```

2. ด้วยสิทธิ root กำหนดคอนฟิกค่าติดตั้ง **IPv6** โดยการพิมพ์ คำสั่งต่อไปนี้:

   ```
   autoconf6
   ```

3. รันคำสั่งต่อไปนี้อีกครั้ง:

   ```
   netstat -ni
   ```

   ผลลัพธ์ของคุณ ควรมีลักษณะคล้ายกับตัวอย่างต่อไปนี้:

   ```
   Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
   en0   1500  link#2      0.6.29.4.55.ec      279679     0     2658     0     0
   en0   1500  9.3.230.64  9.3.230.117         279679     0     2658     0     0
   en0   1500  fe80::206:29ff:fe04:55ec        279679     0     2658     0     0
   sit0  1480  link#3      9.3.230.117              0     0        0     0     0
   sit0  1480  ::9.3.230.117                        0     0        0     0     0
   lo0   16896 link#1                           2343     0     2350     0     0
   lo0   16896 127         127.0.0.1            2343     0     2350     0     0
   lo0   16896 ::1                             2343     0     2350     0     0
   ```

4. เริ่มต้น **ndpd-host** daemon โดยการพิมพ์คำสั่ง ต่อไปนี้:

   ```
   startsrc -s ndpd-host
   ```

ขั้นตอนที่ 2: ตั้งค่าเราเตอร์สำหรับ IPv6

1. ตรวจสอบให้แน่ใจว่ามีการตั้งค่าคอนฟิกค่าติดตั้ง **IPv4** โดยการพิมพ์คำสั่ง ต่อไปนี้:

   ```
   netstat -ni
   ```

2. ด้วยสิทธิ root ให้พิมพ์คำสั่งต่อไปนี้:

   ```
   autoconf6
   ```

3. กำหนดคอนฟิกแอดเดรสสากลด้วยตนเองบนอินเตอร์เฟสของเราเตอร์ที่เป็นสมาชิก ของ subnets แต่ละเครื่องจากทั้งหมดสองเครื่อง โดยการพิมพ์คำสั่งต่อไปนี้:

   ```
   # ifconfig en0 inet6 3ffe:0:0:aaaa::/64 eui64 alias
   # ifconfig en1 inet6 3ffe:0:0:bbbb::/64 eui64 alias
   ```

   คุณจะต้อง ทำเช่นนี้สำหรับทุก subnet ที่เราเตอร์ของคุณกำลังส่งแพ็กเก็ตไป

4. เมื่อต้องการเรียกใช้การส่งต่อ IPv6 ให้พิมพ์ดังต่อไปนี้:

   ```
   no -o ip6forwarding=1
   ```

5. เมื่อต้องการเริ่มต้น **ndpd-router** daemon ให้พิมพ์ดังต่อไปนี้:

```
startsrc -s ndpd-router
```

ndpd-router daemon จะใช้คำเติมหน้าที่สอดคล้องกับแอดเดรสสากลซึ่ง คุณตั้งค่าคอนฟิกไว้บนเราเตอร์ ในกรณีนี้ เราเตอร์ ndpd จะใช้ คำเติมหน้า 3ffe:0:0:aaaa::/64 บน en0 และคำเติมหน้า 3ffe:0:0:bbbb::/64 บน en1

ขั้นตอนที่ 3. ตั้งค่า IPv6 ซึ่งจะตั้งค่าคอนฟิกบนโฮสต์ ณ เวลา บูต

IPv6 ที่ตั้งค่าคอนฟิกใหม่ของคุณจะถูกลบออกเมื่อคุณ รีบูตเครื่อง เมื่อต้องการเปิดใช้งานการทำงานของโฮสต์ **IPv6** ในทุกครั้งที่คุณรีบูต ให้ทำดังต่อไปนี้:

1. เปิดไฟล์ /etc/rc.tcpip โดยใช้โปรแกรมแก้ไขข้อความที่โปรดปรานของคุณ
2. ยกเลิกการแสดงข้อคิดเห็นเกี่ยวกับบรรทัดต่อไปนี้ในไฟล์นั้น:

```
# Start up autoconf6 process
start /usr/sbin/autoconf6 ""

# Start up ndpd-host daemon
start /usr/sbin/ndpd-host "$src_running"
```

3. เพิ่มแฟล็ก **-A** ลงใน start /usr/sbin/autoconf6 "":

```
start /usr/sbin/autoconf6 "" -A
```

เมื่อคุณรีบูต จะมีการตั้งค่าคอนฟิก **IPv6** ทำซ้ำกระบวนการนี้สำหรับแต่ละโฮสต์

ขั้นตอนที่ 4: ตั้งค่า IPv6 ซึ่งจะตั้งค่าคอนฟิกบนเราเตอร์ ณ เวลา บูต

IPv6 ที่ตั้งค่าคอนฟิกใหม่ของคุณจะถูกลบออกเมื่อคุณ รีบูต เมื่อต้องการเปิดใช้งานการทำงานของเราเตอร์ **IPv6** ในทุกครั้งที่คุณรีบูต ให้ทำดังต่อไปนี้:

1. เปิดไฟล์ /etc/rc.tcpip ในโปรแกรมแก้ไขข้อความที่โปรดปรานของคุณ
2. ยกเลิกการแสดงข้อคิดเห็นเกี่ยวกับบรรทัดต่อไปนี้ในไฟล์นั้น:

```
# Start up autoconf6 process
start /usr/sbin/autoconf6 ""
```

3. เพิ่มบรรทัดต่อไปนี้ตามหลังต่อจากบรรทัดซึ่งคุณเพิ่งยกเลิกการแสดงข้อคิดเห็น ในขั้นตอนก่อนหน้านี้:

```
# Configure global addresses for router
ifconfig en0 inet6 3ffe:0:0:aaaa::/64 eui64 alias
ifconfig en1 inet6 3ffe:0:0:bbbb::/64 eui64 alias
```

ในสถานการณ์จำลองนี้ เครือข่ายของเรามีเพียงสอง subnets เท่านั้นคือ en0 และ en1 คุณจะต้องเพิ่มหนึ่งบรรทัดลงในไฟล์นี้สำหรับทุก subnet ที่เราเตอร์ ของคุณกำลังส่งแพ็กเก็ตไป

4. ยกเลิกการแสดงข้อคิดเห็นเกี่ยวกับบรรทัดต่อไปนี้ในไฟล์:

```
# Start up ndpd-router daemon
start /usr/sbin/ndpd- router "$src_running"
```

เมื่อคุณรีบูต **IPv6** จะเริ่มต้นขึ้นโดยอัตโนมัติ

การอัพเกรดเป็น IPv6 ด้วย IPv4 ที่ไม่ได้กำหนดคอนฟิก:

สถานการณ์จำลองนี้แสดงวิธีการตั้งค่าโฮสต์และเราเตอร์สำหรับ **IPv6** โดยไม่มีค่าติดตั้ง **IPv4** ที่ตั้งค่าคอนฟิก

เครือข่ายที่ใช้ในตัวอย่างนี้ประกอบด้วยเราเตอร์หนึ่งเครื่องและ subnets สองเครื่อง มีอยู่สองโฮสต์บนแต่ละ subnet คือ: เราเตอร์ และโฮสต์อื่น ที่ตอนท้ายของสถานการณ์จำลอง เราเตอร์จะใช้คำเติมหน้า 3ffe:0:0:aaaa::/64 บน อินเตอร์เฟสเครือข่าย en0 และคำเติมหน้า 3ffe:0:0:bbbb::/64 บน อินเตอร์เฟสเครือข่าย en1 อันดับแรก คุณจะตั้งค่าคอนฟิกเครื่องเพื่อสนับสนุน **IPv6** เป็นการชั่วคราวเพื่อให้คุณสามารถทดสอบได้ จากนั้น คุณจะ ตั้งค่าคอนฟิกเครื่องเพื่อให้พร้อมจะเป็น IPv6 ณ เวลาบูต

สถานการณ์จำลอง นี้สมมติว่ามีการติดตั้งชุดไฟล์ bos.net.tcp.client ไว้แล้ว

เมื่อต้องการอัพเกรดเป็น **IPv6** ด้วย **IPv4** ที่ตั้งค่าคอนฟิกแล้ว ให้ดูที่ “การอัพเกรดเป็น IPv6 ด้วย IPv4 ที่ตั้งค่าคอนฟิก” ในหน้า 141

สิ่งที่ต้อง พิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ได้ผ่านการทดสอบโดยใช้เวอร์ชันเฉพาะของ AIX ผลลัพธ์ที่คุณได้อาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชันและระดับ AIX ของคุณ

ขั้นตอนที่ 1: ตั้งค่าโฮสต์สำหรับ IPv6

1. ด้วยสิทธิ root ให้พิมพ์คำสั่งต่อไปนี้บนแต่ละโฮสต์บน subnet:

   ```
   autoconf6 -A
   ```

   คำสั่งนี้ จะแสดงอินเตอร์เฟสทั้งหมดที่สามารถจัดการกับ **IPv6** บนระบบ

   หมายเหตุ: เมื่อต้องการแสดง ชุดย่อยของอินเตอร์เฟส ให้ใช้แฟล็ก **–i** ตัวอย่างเช่น `autoconf6 -i en0 en1` จะแสดงอินเตอร์เฟส en0 และ en1
2. พิมพ์คำสั่งต่อไปนี้เพื่อดูอินเตอร์เฟสของคุณ:

   ```
   netstat -ni
   ```

   ผลลัพธ์ของคุณ ควรมีลักษณะคล้ายกับตัวอย่างต่อไปนี้:

   ```
   Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
   en0   1500  link#3      0.4.ac.17.b4.11        7     0       17     0     0
   en0   1500  fe80::204:acff:fe17:b411            7     0       17     0     0
   lo0   16896 link#1                           436     0      481     0     0
   lo0   16896 127         127.0.0.1            436     0      481     0     0
   lo0   16896 ::1                              436     0      481     0     0
   ```
3. เริ่มต้น **ndpd-host** daemon โดยการพิมพ์คำสั่ง ต่อไปนี้:

   ```
   startsrc -s ndpd-host
   ```

ขั้นตอนที่ 2: ตั้งค่าเราเตอร์สำหรับ IPv6

1. ด้วยสิทธิ root ให้พิมพ์คำสั่งต่อไปนี้บนเราเตอร์โฮสต์:

   ```
   autoconf6 -A
   ```

   คำสั่งนี้ จะแสดงอินเตอร์เฟสทั้งหมดที่สามารถจัดการกับ **IPv6** บนระบบ

   หมายเหตุ: เมื่อต้องการแสดง ชุดย่อยของอินเตอร์เฟส ให้ใช้แฟล็ก `-i` ตัวอย่างเช่น `autoconf6 -i en0 en1` จะแสดงอินเตอร์เฟส en0 และ en1
   ผลลัพธ์ของคุณ ควรมีลักษณะคล้ายกับตัวอย่างต่อไปนี้:

```
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en1   1500  link#2      0.6.29.dc.15.45        0     0        7     0     0
en1   1500  fe80::206:29ff:fedc:1545             0     0        7     0     0
en0   1500  link#3      0.4.ac.17.b4.11        7     0       17     0     0
en0   1500  fe80::204:acff:fe17:b411             7     0       17     0     0
lo0   16896 link#1                            436     0      481     0     0
lo0   16896 127         127.0.0.1              436     0      481     0     0
lo0   16896 ::1                               436     0      481     0     0
```

2. ตั้งค่าคอนฟิกแอดเดรสสากลด้วยตนเองบนอินเตอร์เฟสของเราเตอร์ที่เป็นสมาชิก ของ subnets แต่ละเครื่องจากทั้งหมดสองเครื่อง โดยการพิมพ์คำสั่งต่อไปนี้:

   ```
   # ifconfig en0 inet6 3ffe:0:0:aaaa::/64 eui64 alias
   # ifconfig en1 inet6 3ffe:0:0:bbbb::/64 eui64 alias
   ```

   หมายเหตุ: คุณจะต้อง ทำเช่นนี้สำหรับทุก subnet ที่เราเตอร์ของคุณกำลังส่งแพ็กเก็ตไป

3. เมื่อต้องการเรียกใช้การส่งต่อ **IPv6** ให้พิมพ์ดังต่อไปนี้:

   ```
   no -o ip6forwarding=1
   ```

4. เมื่อต้องการเริ่มต้น **ndpd-router** daemon ให้พิมพ์ดังต่อไปนี้:

   ```
   startsrc -s ndpd-router
   ```

   ndpd-router daemon จะใช้คำเติมหน้าที่สอดคล้องกับแอดเดรสสากลซึ่ง คุณตั้งค่าคอนฟิกไว้บนเราเตอร์ ในกรณีนี้ เราเตอร์ ndpd จะใช้ คำเติมหน้า `3ffe:0:0:aaaa::/64` บน `en0` และคำเติมหน้า `3ffe:0:0:bbbb::/64` บน `en1`

5. กด Enter เพื่อทำต่อไป
6. กด Enter ครั้งที่สองเพื่อยืนยันการตัดสินใจของคุณ และเริ่มต้นการติดตั้ง กลุ่มซอฟต์แวร์

ขั้นตอนที่ 3. ตั้งค่า IPv6 ซึ่งจะตั้งค่าคอนฟิกบนโฮสต์ ณ เวลา บูต

หลังจากทำขั้นตอนที่ 1 เสร็จสมบูรณ์แล้วสำหรับแต่ละโฮสต์ **IPv6** จะ ถูกลบออกเมื่อคุณรีบูตเครื่อง เมื่อต้องการเปิดใช้งานการทำงานของโฮสต์ **IPv6** ในทุกครั้งที่คุณรีบูต ให้ทำดังต่อไปนี้:

1. เปิดไฟล์ `/etc/rc.tcpip` โดยใช้โปรแกรมแก้ไขข้อความที่โปรดปรานของคุณ
2. ยกเลิกการแสดงข้อคิดเห็นเกี่ยวกับบรรทัดต่อไปนี้ในไฟล์นั้น:

   ```
   # Start up autoconf6 process
   start /usr/sbin/autoconf6 ""

   # Start up ndpd-host daemon
   start /usr/sbin/ndpd-host "$src_running"
   ```

3. เพิ่มแฟล็ก **-A** ลงใน `start /usr/sbin/autoconf6 ""`:

   ```
   start /usr/sbin/autoconf6 "" -A
   ```

4. ทำซ้ำกระบวนการนี้สำหรับแต่ละโฮสต์

เมื่อคุณรีบูต **IPv6** จะเริ่มต้นขึ้นโดยอัตโนมัติ

ขั้นตอนที่ 4: ตั้งค่า IPv6 ซึ่งจะตั้งค่าคอนฟิกบนเราเตอร์ ณ เวลา บูต

หลังจากทำขั้นตอนที่ 2 เสร็จสมบูรณ์แล้วสำหรับเราเตอร์ของคุณ **IPv6** จะถูกลบออกเมื่อคุณรีบูต เมื่อต้องการเปิดใช้งานการทำงานของเราเตอร์ **IPv6** ในทุกครั้งที่คุณรีบูต ให้ทำดังต่อไปนี้:

1. เปิดไฟล์ `/etc/rc.tcpip` ในโปรแกรมแก้ไขข้อความที่โปรดปรานของคุณ

2. ยกเลิกการแสดงข้อคิดเห็นเกี่ยวกับบรรทัดต่อไปนี้ในไฟล์นั้น:

```
# Start up autoconf6 process
start /usr/sbin/autoconf6 ""
```

3. เพิ่มแฟล็ก **-A** ลงในบรรทัดนั้น:

```
start /usr/sbin/autoconf6 "" -A
```

4. เพิ่มบรรทัดต่อไปนี้ตามหลังต่อจากบรรทัดซึ่งคุณเพิ่งยกเลิกการแสดงข้อคิดเห็น ในขั้นตอนก่อนหน้านี้:

```
# Configure global addresses for router
ifconfig en0 inet6 3ffe:0:0:aaaa::/64 eui64 alias
ifconfig en1 inet6 3ffe:0:0:bbbb::/64 eui64 alias
```

ในสถานการณ์จำลองนี้ เครือข่ายของเรามีเพียงสอง subnets เท่านั้นคือ `en0` และ `en1` คุณจะต้องเพิ่มหนึ่งบรรทัดลงในไฟล์นี้สำหรับทุก subnet ที่เราเตอร์ ของคุณกำลังส่งแพ็กเก็ตไป

5. ยกเลิกการแสดงข้อคิดเห็นเกี่ยวกับบรรทัดต่อไปนี้ในไฟล์:

```
# Start up ndpd-router daemon
start /usr/sbin/ndpd-router "$src_running"
```

6. รันคำสั่งต่อไปนี้เพื่อเปิดใช้งานการส่งต่อ IP ณ เวลาบูต:

```
no -r -o ip6forwarding=1
```

เมื่อคุณรีบูต **IPv6** จะเริ่มต้นขึ้นโดยอัตโนมัติ

คอนฟิกูเรชันรันไทม์แบบสแตติก:

สถานการณ์จำลองนี้นำคุณผ่านคอนฟิกูเรชันรันไทม์ ของโหนดโดยใช้สแตติก IPs และเราต์

เครือข่ายที่ใช้ในตัวอย่างนี้ประกอบด้วยโฮสต์หนึ่งเครื่องและ เราเตอร์หนึ่งเครื่อง เมื่อสิ้นสุดสถานการณ์จำลอง อินเตอร์เฟส IPv6 มีการตั้งค่า บนโฮสต์ อันดับแรก คุณกำหนดคอนฟิกเครื่องเพื่อสนับสนุน IPv6 เป็นการชั่วคราวเพื่อให้คุณสามารถทดสอบได้ จากนั้น คุณกำหนดคอนฟิก เครื่องเพื่อให้พร้อมจะเป็น IPv6 ณ เวลาบูต

สิ่งที่ต้องพิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ผ่านการทดสอบแล้วโดยใช้ เวอร์ชันเฉพาะของ AIX ผลลัพธ์ ที่คุณได้รับอาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชัน และระดับ AIX ของคุณ
- ตัวอย่างสมมติว่า **2001:1:2::/48** เป็น Aggregate Global Unicast Address สำหรับอินเตอร์เฟส IPv6 ที่กำหนดโดย Internet Assigned Numbers Authority (IANA) ให้กับผู้ให้บริการ และ **2001:1:2:3:4::/64** เป็น ชุดย่อยที่ใช้บิต 49 – 64 ซึ่งกำหนดโดยผู้ดูแลระบบเครือข่าย
- คุณต้องอ้างอิง RFC 3587 เพื่อให้เข้าใจรูปแบบ IPv6 Global Unicast Address

ข้อมูลที่เกี่ยวข้อง:

คำสั่งคอนฟิกูเรชันรันไทม์

คำสั่ง autoconf6

ขั้นตอนที่ 1 การตั้งค่าโฮสต์สำหรับ IPv6:

ปฏิบัติตามโพรซีเดอร์นี้เพื่อตั้งค่าโฮสต์สำหรับ IPv6

1. ด้วยสิทธิ root ให้กำหนดคอนฟิกค่าติดตั้ง IPv6 โดยป้อน คำสั่งต่อไปนี้:

```
# autoconf6
```

2. รันคำสั่งต่อไปนี้อีกครั้ง:

```
# netstat -ni
```

ผลลัพธ์ของคุณ ควรมีลักษณะคล้ายกับเอาต์พุตต่อไปนี้:

```
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      0.6.29.4.55.ec      279679     0     2658     0     0
en0   1500  9.3.230.64  9.3.230.117         279679     0     2658     0     0
en0   1500  fe80::206:29ff:fe04:55ec        279679     0     2658     0     0
sit0  1480  link#3      9.3.230.117              0     0        0     0     0
sit0  1480  ::9.3.230.117                        0     0        0     0     0
lo0   16896 link#1                           2343     0     2350     0     0
lo0   16896 127         127.0.0.1           2343     0     2350     0     0
lo0   16896 ::1                             2343     0     2350     0     0
```

3. ใช้คำสั่ง **chdev** เพื่อเพิ่มแอดเดรส IPv6 ลงในโฮสต์ อินเตอร์เฟส สำหรับตัวอย่างนี้ low-order 64 บิตถูกนำมาจาก low-order 64 บิตของ Link-Local IP ที่สร้างขึ้นโดย **autoconf6** บน อินเตอร์เฟส **en0**

```
# chdev -l en0 -a netaddr6='2001:2:3:4:206:29ff:fe04:55ec' -a prefixlen=64
```

4. ลบเราต์ของลิงก์คำเติมหน้าที่มีอยู่สำหรับคำเติมหน้าต่อไปนี้:

```
# route delete -inet6 2001:2:3:4::/64
```

5. กำหนดคอนฟิกเราต์แบบสแตติกของคำเติมหน้าบนโฮสต์ เพื่อเพิ่มความสามารถในการเข้าถึง เราเตอร์โดยที่ **fe80::206:29ff:fe04:66e** คือ เราเตอร์หรือเกตเวย์ซึ่งมีภาวะเชื่อมต่อกับเราเตอร์

```
# route add -inet6 -net 2001:2:3:4::/64 fe80::206:29ff:fe04:66e -static
```

หมายเหตุ: ถ้า การเปลี่ยนแปลงเป็นสิ่งจำเป็นสำหรับเราต์ดีฟอลต์ ตรวจสอบให้แน่ใจว่า **autoconf6** รันด้วยอ็อพชัน **-R** ที่ป้องกันไม่ให้เพิ่มหรือเขียนทับ เราต์ดีฟอลต์ใดๆ บนโหนด จากนั้น ทำซ้ำขั้นตอน 3-5

ขั้นตอนที่ 2 การตั้งค่าเราเตอร์สำหรับ IPv6:

ปฏิบัติตามโพรซีเดอร์นี้เพื่อตั้งค่าเราเตอร์สำหรับ IPv6

1. ตรวจสอบให้แน่ใจว่ามีการกำหนดคอนฟิกค่าติดตั้ง IPv4 โดยป้อน คำสั่งต่อไปนี้:

```
# netstat -ni
```

2. ด้วยสิทธิ root ให้ป้อนคำสั่งต่อไปนี้:

```
# autoconf6
```

3. เมื่อต้องการเรียกใช้การส่งต่อ IPv6 ให้ป้อนคำสั่งต่อไปนี้:

```
# no -o ip6forwarding=1
```

4. กำหนดคอนฟิกโกลบอล IP บนเราเตอร์อินเตอร์เฟสโดยป้อนคำสั่ง ต่อไปนี้:

```
# chdev -l en0 -a netaddr6='2001:4:5:6:207:30ff:fe05:66ec' -a prefixlen=64
```

5. กำหนดคอนฟิกเราต์บนเราเตอร์ด้วยตนเองเพื่อเปิดใช้งานการจัดส่ง แพ็กเก็ตที่ถูกต้อง ตัวอย่างเช่น ถ้า **fe80::3ca6:70ff:fe00:3004/64** เป็น เกตเวย์สำหรับคำเติมหน้า **2001:2:3:4::/64** ให้เพิ่มเราต์ คำเติมหน้าดังนี้:

```
# route add -inet6 -net 2001:2:3:4::/64 fe80::3ca6:70ff:fe00:3004 -static
```

ขั้นตอนที่ 3 การตั้งค่า IPv6 ที่จะกำหนดคอนฟิกบนโฮสต์ในทุกครั้งที่ รีสตาร์ท:

ค่าติดตั้งโฮสต์ IPv6 ที่กำหนดคอนฟิกไว้ใน **ขั้นตอน 1. การตั้งค่า โฮสต์สำหรับ IPv6** จะถูกลบออกเมื่อคุณรีสตาร์ทเครื่อง เมื่อต้องการ เปิดใช้งานการทำงานโฮสต์ IPv6 ในทุกครั้งที่คุณรีสตาร์ทเครื่อง ให้ปฏิบัติตามโพรซีเดอร์นี้

1. เปิดไฟล์ **/etc/rc.tcpip** ในเท็กซ์เอดิเตอร์
2. ยกเลิกการแสดงข้อคิดเห็นบรรทัดต่อไปนี้ในไฟล์ **/etc/rc.tcpip**:

   ```
   # Start up autoconf6 process
   start /usr/sbin/autoconf6 ""
   ```

   หมายเหตุ: ถ้าบรรทัดก่อนหน้า ไม่มีอยู่ในไฟล์ **/etc/rc.tcpip** ให้เพิ่มบรรทัดนั้น ในไฟล์
3. เพิ่มแฟล็ก **-A** ลงใน **start /usr/sbin/autoconf6** *""*

   ```
   start /usr/sbin/autoconf6 "" -A
   ```
4. เพิ่มบรรทัดต่อไปนี้ในไฟล์ **/etc/rc.tcpip** ต่อจากบรรทัดที่คุณยกเลิกการแสดงข้อคิดเห็น (หรือเพิ่ม):

   ```
   chdev -l en0 -a netaddr6='2001:2:3:4:206:29ff:fe04:55ec' -a prefixlen=64
   ```
5. ลบเราต์คำเติมหน้าที่มีอยู่ก่อนหน้านี้ โดยป้อน คำสั่งต่อไปนี้:

   ```
   chdev -l inet0 -a delrout6='-net, 2001:2:3:4::/64'
   ```
6. ตั้งค่าเราต์โดยป้อนคำสั่งต่อไปนี้:

   ```
   chdev -l inet0 -a rout6='-net, 2001:2:3:4::/64 ,fe80::206:29ff:fe04:66e,-static'
   ```

เมื่อคุณรีสตาร์ทเครื่อง จะมีการตั้งค่าคอนฟิกูเรชัน IPv6 ของคุณ

หมายเหตุ: คุณ ต้องทำซ้ำโพรซีเดอร์นี้สำหรับแต่ละโฮสต์

ขั้นตอนที่ 4 การตั้งค่า IPv6 ที่จะกำหนดคอนฟิกบนเราเตอร์ในทุกครั้งที่ รีสตาร์ท:

ค่าติดตั้งเราเตอร์ IPv6 ที่กำหนดคอนฟิกไว้ใน **ขั้นตอน 2. การตั้งค่า เราเตอร์สำหรับ IPv6** จะถูกลบออกเมื่อคุณรีสตาร์ทเครื่อง เมื่อต้องการเปิดใช้งานการทำงานเราเตอร์ IPv6 ในทุกครั้งที่คุณรีสตาร์ทเครื่อง ให้ปฏิบัติตามโพรซีเดอร์นี้

1. เปิดไฟล์ **/etc/rc.tcpip** ในเท็กซ์เอดิเตอร์
2. ยกเลิกการแสดงข้อคิดเห็นบรรทัดต่อไปนี้ในไฟล์ **/etc/rc.tcpip**:

   ```
   # Start up autoconf6 process
   start /usr/sbin/autoconf6 ""
   ```

   หมายเหตุ: ถ้าบรรทัดก่อนหน้า ไม่มีอยู่ในไฟล์ **/etc/rc.tcpip** ให้เพิ่มบรรทัดนั้น ในไฟล์
3. เพิ่มแฟล็ก **-A** ลงใน **start /usr/sbin/autoconf6** *""*

   ```
   start /usr/sbin/autoconf6 "" -A
   ```
4. เพิ่มบรรทัดต่อไปนี้ต่อจากบรรทัดที่คุณยกเลิกการแสดงข้อคิดเห็น (หรือ เพิ่ม) ในขั้นตอน 2 เพื่อกำหนดคอนฟิกโกลบอล IP บนเราเตอร์ และเพื่อกำหนดคอนฟิก เราต์คำเติมหน้า

   ```
   chdev -l en0 -a netaddr6='2001:4:5:6:207:30ff:fe05:66ec' -a prefixlen=64
   chdev -l inet0 -a rout6='-net,2001:2:3:4::/64,fe80::3ca6:70ff:fe00:3004,-static'
   ```

   ใน สถานการณ์จำลองนี้ เครือข่ายมีเพียงหนึ่ง subnet คือ **en0** คุณต้องเพิ่มบรรทัดในไฟล์นี้สำหรับทุก subnet ซึ่งเราเตอร์จะส่งแพ็กเก็ต

เมื่อคุณรีสตาร์ทเครื่อง IPv6 จะมีการเริ่มต้นโดยอัตโนมัติบน เครื่อง

หมายเหตุ: เมื่อคุณใช้คอนฟิกูเรชันแบบสแตติกพร้อมกันกับ **ndpd-host** ตรวจสอบ ให้แน่ใจว่าแฟล็กต่างๆ ใน **ndpd-host** มีการสำรวจเพื่อ รักษาสแตติก IPs และเราต์ ถ้าจำเป็น

การตั้งค่า tunneling ใน IPv6:

คุณสามารถใช้วิธีการอย่างใดอย่างหนึ่งจากสองวิธีเพื่อตั้งค่า tunneling ใน IPv6 วิธีการแรกตั้งค่า tunnel อัตโนมัติ วิธีการที่สอง ตั้งค่า tunnel ที่ตั้งค่าคอนฟิก

สิ่งที่ต้องพิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ได้ผ่านการทดสอบโดยใช้เวอร์ชันเฉพาะของ AIX ผลลัพธ์ที่คุณได้อาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชันและระดับ AIX ของคุณ

การตั้งค่า tunnel อัตโนมัติใน IPv6

ในสถานการณ์จำลองนี้ จะใช้คำสั่ง **autoconf6** เพื่อตั้งค่าคอนฟิก IPv6 และ ตั้งค่า tunnel อัตโนมัติผ่านทางอินเตอร์เฟสหลัก, en2 จากนั้น จะใช้คำสั่ง **autoconf6** เพื่อตั้งค่าคอนฟิก tunnel ผ่านทางอินเตอร์เฟสรอง, en0

ข้อมูลต่อไปนี้ เป็นผลลัพธ์ของคำสั่ง **netstat -ni** ซึ่งแสดงการตั้งค่าคอนฟิก เครือข่ายปัจจุบันของระบบ:

```
en0   1500  link#2      MAC address here         0     0       33     0     0
en0   1500  1.1         1.1.1.3                  0     0       33     0     0
en2   1500  link#3      MAC address here     79428     0      409     0     0
en2   1500  10.1        10.1.1.1             79428     0      409     0     0
```

- เมื่อต้องการเปิดใช้งาน IPv6 และ tunnel อัตโนมัติหนึ่งรายการ ให้พิมพ์คำสั่งต่อไปนี้:

  ```
  autoconf6
  ```

  การรันคำสั่ง **netstat -ni** ในขณะนี้ทำให้เกิด ผลลัพธ์ต่อไปนี้:

  ```
  # netstat -in
  en0   1500  link#2      MAC address here         0     0       33     0     0
  en0   1500  1.1         1.1.1.3                  0     0       33     0     0
  en0   1500  fe80::204:acff:fe49:4910              0     0       33     0     0
  en2   1500  link#3      MAC address here     79428     0      409     0     0
  en2   1500  10.1        10.1.1.1             79428     0      409     0     0
  en2   1500  fe80::220:35ff:fe12:3ae8
  sit0  1480  link#7      10.1.1.1                 0     0        0     0     0
  sit0  1480  ::10.1.1.1
  ```

  ถ้า en2 (IP แอดเดรส 10.1.1.1) เป็นอินเตอร์เฟสหลัก แอดเดรส ::10.1.1.1 จะมีอยู่ในตอนนี้สำหรับ tunneling อัตโนมัติบนอินเตอร์เฟส en2

- เมื่อต้องการเปิดใช้งาน tunnel อัตโนมัติผ่านทางอินเตอร์เฟส en0 ให้พิมพ์คำสั่งต่อไปนี้:

  ```
  autoconf6 -s -i en0
  ```

  การรันคำสั่ง **netstat -ni** ในขณะนี้ทำให้เกิด ผลลัพธ์ต่อไปนี้:

  ```
  # netstat -in
  en0   1500  link#2      MAC address here         0     0       33     0     0
  en0   1500  1.1         1.1.1.3                  0     0       33     0     0
  ```

```
en0   1500  fe80::204:acff:fe49:4910            0     0       33     0     0
en2   1500  link#3      MAC address here     79428     0      409     0     0
en2   1500  10.1        10.1.1.1             79428     0      409     0     0
en2   1500  fe80::220:35ff:fe12:3ae8
sit0  1480  link#7      1.1.1.3                  0     0        3     0     0
sit0  1480  ::10.1.1.1                           0     0        3     0     0
sit0  1480  ::1.1.1.3                            0     0        3     0     0
```

การดำเนินการนี้ทำให้มีการเพิ่มแอดเดรส IPv6 ที่เข้ากันได้กับ IPv4 ลงใน อินเตอร์เฟส SIT ที่มีอยู่, `sit0` ในขณะนี้ tunneling ยังมีการ เปิดใช้งานสำหรับอินเตอร์เฟส `en0` โดยใช้แอดเดรส ::1.1.1.3 ด้วย จะใช้อินเตอร์เฟส เดียวกัน, `sit0`, สำหรับทั้งสอง tunnels

หมายเหตุ: Tunnels อัตโนมัติถูกลบออกเมื่อรีสตาร์ทระบบ เพื่อให้ tunnel อัตโนมัติ มีอยู่ในเวลาบูต ให้เพิ่มอาร์กิวเมนต์ที่จำเป็นลงในคำสั่ง **autoconf6** ในไฟล์ `/etc/rc.tcpip`

การตั้งค่า tunnels ที่ตั้งค่าคอนฟิก

ในสถานการณ์จำลองนี้ จะใช้ SMIT เพื่อตั้งค่า tunnel ที่ตั้งค่าคอนฟิก Tunnel นี้จะมีอยู่ เมื่อระบบรีสตาร์ทเนื่องจากจะถูกจัดเก็บไว้ใน ODM Tunnel จะมีการ ตั้งค่าคอนฟิกระหว่างระบบ `alpha` และ `beta` IPv4 address ของ `alpha` คือ 10.1.1.1 และ IPv4 address ของ `beta` คือ 10.1.1.2

เมื่อต้องการตั้งค่า tunnels ที่ตั้งค่าคอนฟิก ให้ปฏิบัติตามขั้นตอนเหล่านี้:

1. เมื่อต้องการตั้งค่าคอนฟิก tunnel ระหว่าง `alpha` และ `beta` ให้พิมพ์ดังต่อไปนี้บนทั้งสองระบบ:

   ```
   smit ctinet6
   ```

2. เลือก **เพิ่ม IPV6 ใน IPV4 Tunnel Interface** บน ทั้งสองระบบ

   ```
   autoconf6
   ```

3. ในสถานการณ์จำลองนี้ เรากรอกค่าดังนี้บน `alpha` ตามข้อมูล IPv4 addresses:

   ```
   * IPV4 SOURCE ADDRESS (dotted decimal)              [10.1.1.1]
   * IPV4 DESTINATION ADDRESS (dotted decimal)         [10.1.1.2]
     IPV6 SOURCE ADDRESS (colon separated)            []
     IPV6 DESTINATION ADDRESS (colon separated)       []
   ```

   บน `beta` มีการป้อนค่าต่อไปนี้:

   ```
   * IPV4 SOURCE ADDRESS (dotted decimal)              [10.1.1.2]
   * IPV4 DESTINATION ADDRESS (dotted decimal)         [10.1.1.1]
     IPV6 SOURCE ADDRESS (colon separated)            []
     IPV6 DESTINATION ADDRESS (colon separated)       []
   ```

4. เมื่อต้องการดูอินเตอร์เฟสที่ตั้งค่าคอนฟิก ให้พิมพ์คำสั่งต่อไปนี้:

   ```
   ifconfig cti X
   ```

   โดยที่ *X* คือหมายเลขของอินเตอร์เฟส ในสถานการณ์จำลองนี้ มีการส่งคืนผลลัพธ์ ต่อไปนี้: บน `alpha`:

   ```
   cti0: flags=8080051<UP,POINTOPOINT,RUNNING,MULTICAST>
           inet6 fe80::a01:101/128 --> fe80::a01:102
   ```

   บน `beta`:

   ```
   cti0: flags=8080051 <UP,POINTOPOINT,RUNNING,MULTICAST>
           inet6 fe80::a01:102/128 --> fe80::a01:101
   ```

SMIT สร้าง IPv6 addresses สำหรับปลายทั้งสองด้านของ tunnel ให้โดยอัตโนมัติโดยใช้วิธีการต่อไปนี้:

- 32 บิตด้านล่างมี IPv4 address
- 96 บิตด้านบนมีคำเติมหน้า `fe80::/96`

คุณสามารถกรอก IPv6 addresses เฉพาะถ้าต้องการ

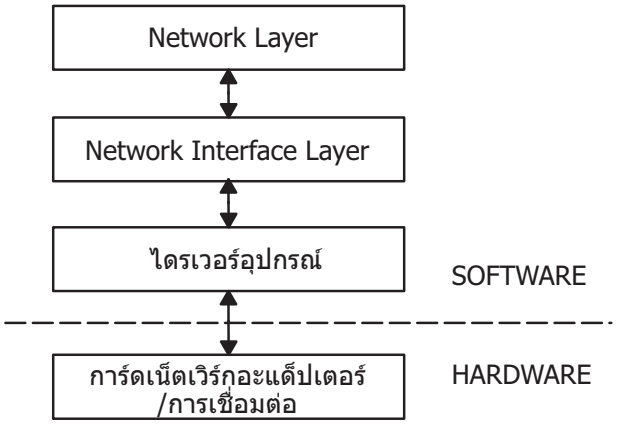
การติดตามแพ็กเก็ต

การติดตามแพ็กเก็ตคือกระบวนการที่คุณสามารถตรวจสอบพาธของแพ็กเก็ต ผ่านเลเยอร์ไปยังปลายทาง

คำสั่ง **iptrace** ดำเนินการกับการติดตามแพ็กเก็ตของ ระดับของเน็ตเวิร์กอินเตอร์เฟส คำสั่ง **ipreport** ออกเอาต์พุตบนการติดตามแพ็กเก็ต ทั้งในรูปแบบเลขฐานสิบหกและรูปแบบ ASCII คำสั่ง **trpt** ดำเนินการกับการติดตามแพ็กเก็ตระดับโปรโตคอลการส่งผ่านสำหรับ **TCP** เอาต์พุตคำสั่ง **trpt** มีรายละเอียดเพิ่มเติม ซึ่งประกอบด้วยข้อมูลเกี่ยวกับเวลา สถานะของ **TCP** และการจัดลำดับแพ็กเก็ต

ส่วนหัวของเน็ตเวิร์กอินเตอร์เฟสแพ็กเก็ต

ที่เลเยอร์เน็ตเวิร์กอินเตอร์เฟส ส่วนหัวของแพ็กเก็ตจะถูกแนบกับข้อมูลขาออก



รูปที่ 8. แพ็กเก็ตจะไหลผ่าน Network Interface Structure

ภาพประกอบนี้จะแสดงการไหลของข้อมูลแบบสองทิศทางผ่านเลเยอร์ของ Network Interface Structure จากด้านบน (ซอฟต์แวร์) จะมี เน็ตเวิร์กเลเยอร์ เน็ตเวิร์กอินเตอร์เฟสเลเยอร์ ไดรเวอร์อุปกรณ์ และ (ฮาร์ดแวร์) เน็ตเวิร์กอะแด็ปเตอร์การ์ด หรือการเชื่อมต่อ

จากนั้นแพ็กเก็ตจะถูกส่งผ่านเน็ตเวิร์กอะแด็ปเตอร์ไปยังเน็ตเวิร์กที่เหมาะสม แพ็กเก็ตสามารถผ่านหลายเกตเวย์ก่อนที่จะถึงเป้าหมายของมัน ที่เน็ตเวิร์กเป้าหมาย ส่วนหัวจะถูกแยกออกจากแพ็กเก็ตและข้อมูลถูกส่งไปยังโฮสต์ที่เหมาะสม

ส่วนต่อไปนี้ประกอบด้วยข้อมูลส่วนหัวของแพ็กเก็ตสำหรับหลายเน็ตเวิร์กอินเตอร์เฟสทั่วไป

ส่วนหัวของเฟรมของ Ethernet อะแด็ปเตอร์:

ส่วนหัวของ **Internet Protocol (IP)** หรือ **Address Resolution Protocol (ARP)** สำหรับ Ethernet อะแด็ปเตอร์จะประกอบด้วย 3 ฟิลด์เหล่านี้

ตารางที่ 55. ส่วนหัวของเฟรมของ Ethernet อะแด็ปเตอร์

ฟิลด์	ความยาว	คำนิยาม
DA	6 ไบต์	แอดเดรสปลายทาง
SA	6 ไบต์	แอดเดรสต้นทาง ถ้าบิต 0 ของฟิลด์นี้ถูกตั้งเป็น 1 มันจะระบุว่า routing information (RI) มีอยู่
Type	2 ไบต์	ระบุว่าแพ็กเก็ตเป็น **IP** หรือ **ARP** ค่าของตัวเลขของชนิดจะถูกลิสต์ด้านล่าง

ตัวเลขชนิดของฟิลด์:

ไอเท็ม	คำอธิบาย
IP	0800
ARP	0806

ส่วนหัวเฟรม Token-Ring:

มีห้าฟิลด์ที่เป็นส่วนหัวของ medium access control (MAC) สำหรับโทเค็นริงอะแด็ปเตอร์

ตารางที่ 56. ส่วนหัว Token-ring MAC

ฟิลด์	ความยาว	นิยาม
AC	1 ไบต์	ค่าควบคุมการเข้าใช้ ค่าในฟิลด์นี้ x`00' กำหนดระดับความสำคัญส่วนหัวเป็น 0
FC	1 ไบต์	การควบคุมฟิลด์ ค่าในฟิลด์นี้ x`40' ระบุเฟรม Logical Link Control
DA	6 ไบต์	แอดเดรสปลายทาง
SA	6 ไบต์	แอดเดรสต้นทาง ถ้าบิต 0 ของฟิลด์นี้ถูกตั้งเป็น 1 มันจะระบุว่า routing information (RI) มีอยู่
RI	18 ไบต์	ข้อมูลการจัดเส้นทาง ฟิลด์ที่ใช้ได้อธิบายดังต่อไปนี้

ส่วนหัว MAC ประกอบด้วยฟิลด์ข้อมูลการจัดเส้นทางโดยแต่ละส่วน มีสองไบต์: routing control (RC) และหมายเลขเซกเมนต์ หมายเลขเซกเมนต์สูงสุด ที่ใช้ได้คือแปดตัว ในการระบุผู้รับของการกระจายข้อมูลที่จำกัด ข้อมูล RC อยู่ในไบต์ 0 และ 1 ของฟิลด์ RI ค่าติดตั้งของสองบิตแรก ของฟิลด์ RC มีความหมายต่อไปนี้:

ไอเท็ม	คำอธิบาย
bit (0) = 0	ใช้เส้นทางแบบไม่กระจายข้อมูลที่ระบุในฟิลด์ RI
bit (0) = 1	สร้างฟิลด์ RI และกระจายข้อมูลไปที่วงแหวนทั้งหมด
bit (1) = 0	กระจายข้อมูลผ่านบริดจ์ทั้งหมด
bit (1) = 1	กระจายข้อมูลผ่านบริดจ์ที่จำกัดไว้

ส่วนหัว logical link control (LLC) ประกอบด้วยห้าฟิลด์ ตามที่แสดง ในตารางส่วนหัว LLC ต่อไปนี้

ตารางที่ 57. ส่วนหัว 802.3 LLC

ฟิลด์	ความยาว	นิยาม
DSAP	1 ไบต์	Destination service access point ค่าในฟิลด์นี้คือ x`aa'
SSAP	1 ไบต์	Source service access point ค่าในฟิลด์นี้คือ x`aa'
CONTROL	1 ไบต์	กำหนดคำสั่งและการตอบกลับของ LLC ค่าที่ใช้ได้สามค่า สำหรับฟิลด์นี้ถูกอธิบายดังต่อไปนี้
PROT_ID	3 ไบต์	Protocol ID ฟิลด์นี้ถูกสงวนไว้ มีค่าเป็น x`0'
TYPE	2 ไบต์	ระบุว่าแพ็กเก็ตเป็น **IP** หรือ **ARP**

ค่าฟิลด์ควบคุม:

ฟิลด์ควบคุมโทเค็นริงมีเฟรมข้อมูลที่ไม่ได้กำหนดตัวเลข, เฟรมการระบุการแลกเปลี่ยนข้อมูล และเฟรมทดสอบ ค่าต่างๆ มีคำอธิบาย ดังนี้

ไอเท็ม	คำอธิบาย
x`03'	เฟรม Unnumbered Information (UI) นี่เป็นวิธี ปกติหรือไม่มีการจัดลำดับ ซึ่งข้อมูลโทเค็นริงอะแด็ปเตอร์ถูกส่งผ่านเน็ตเวิร์ก **TCP/IP** จัดลำดับ ข้อมูล
x`AF'	เฟรม Exchange identification (XID) เฟรมนี้มีข้อมูลคุณสมบัติ ของโฮสต์ที่ส่งข้อมูล
x`E3'	เฟรมทดสอบ เฟรมนี้สนับสนุนการทดสอบพาธการส่งข้อมูล ตอบกลับถึงข้อมูลที่ได้รับ

ส่วนหัวของเฟรม 802.3:

ส่วนหัวของ MAC สำหรับ 802.3 อะแด็ปเตอร์ประกอบด้วย 2 ฟิลด์ ดังแสดงในตารางส่วนหัวของ MAC นี้

ตารางที่ 58. 802.3 MAC header

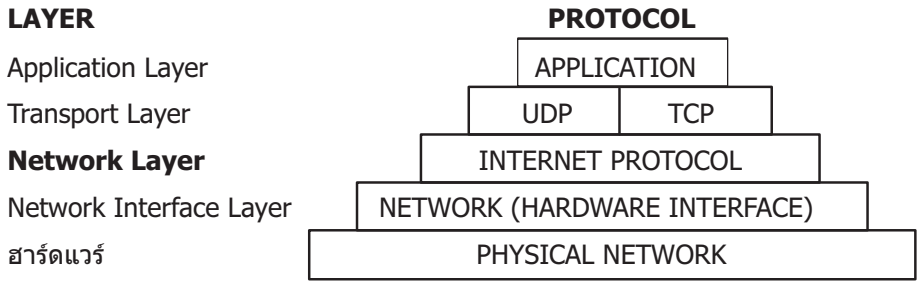
ฟิลด์	ความยาว	คำนิยาม
DA	6 ไบต์	แอดเดรสปลายทาง
SA	6 ไบต์	แอดเดรสต้นทาง ถ้าบิต 0 ของฟิลด์นี้ถูกตั้งเป็น 1 มันจะระบุว่า routing information (RI) มีอยู่

ส่วนหัว LLC สำหรับ 802.3 จะเหมือนกับส่วนหัวของ Token-Ring MAC

อินเตอร์เน็ตโปรโตคอลระดับเน็ตเวิร์ก

อินเตอร์เน็ตโปรโตคอลระดับเน็ตเวิร์กจัดการกับการสื่อสารแบบเครื่อง ต่อเครื่อง

อีกนัยหนึ่ง เลเยอร์นี้นำการเราต์ **TCP/IP** ไปใช้โปรโตคอลเหล่านี้ ยอมรับคำร้องขอเพื่อส่งแพ็กเก็ต (พร้อมกับเน็ตเวิร์กแอดเดรส ของเครื่องปลายทาง) จากเลเยอร์ Transport ซึ่งแปลงแพ็กเก็ต ไปเป็นรูปแบบดาตาแกรม และส่งลงไปยังเลเยอร์ Network Interface สำหรับประมวลผลเพิ่มเติม



รูปที่ 9. เลเยอร์ Network ของ TCP/IP Suite of Protocols

รูปภาพประกอบนี้แสดงเลเยอร์ต่างๆ ของ **TCP/IP** Suite of Protocols จากด้านบนสุด เลเยอร์แอ็พพลิคเชันสอดคล้องกับ แอ็พพลิเคชัน เลเยอร์ transport มี **UDP** และ **TCP** เลเยอร์ network มีเน็ตเวิร์กอินเตอร์เฟส (ฮาร์ดแวร์) และท้ายสุด เลเยอร์ฮาร์ดแวร์มีฟิสิคัลเน็ตเวิร์ก

TCP/IP จัดเตรียมโปรโตคอลที่จำเป็นต้องการคอมไพล์ด้วย RFC 1100 *Official Internet Protocols* พร้อมกับโปรโตคอลอื่นๆ ที่ใช้โดยโฮสต์ในอินเตอร์เน็ต community

หมายเหตุ: การใช้เน็ตเวิร์กอินเตอร์เน็ต เวอร์ชัน ซ็อกเก็ต เซอร์วิส และหมายเลขโปรโตคอลใน **TCP/IP** ยังคอมไพล์ด้วย RFC 1010 *Assigned Numbers*.

Address Resolution Protocol:

โปรโตคอลระดับเน็ตเวิร์กคือ **Address Resolution Protocol** (**ARP**) **ARP** แปลอินเตอร์เน็ตแอดเดรสลงในแอดเดรสของฮาร์ดแวร์บน local area networks

หากต้องการแสดงวิธีการทำงานกับ **ARP** ให้พิจารณาโหนดสองโหนด X และ Y หากโหนด X ต้องการสื่อสารด้วย Y และ X และ Y อยู่บน local area networks (LANs) ที่แตกต่างกัน X และ Y สื่อสารผ่าน *bridges*, *routers* หรือ *gateways* โดยใช้ IP แอดเดรสภายใน LAN โหนดการสื่อสารโดยใช้แอดเดรสฮาร์ดแวร์ ระดับต่ำ

โหนดบนเซ็กเมนต์เดียวกันของ LAN เดียวกันใช้ **ARP** เพื่อกำหนดแอดเดรสของฮาร์ดแวร์ของโหนดอื่นๆ อันดับแรก โหนด X กระจายสัญญาณ **ARP** สำหรับโหนดแอดเดรสฮาร์ดแวร์ของ Y คำร้องขอ **ARP** มี **IP** X และแอดเดรสของฮาร์ดแวร์และ **IP** แอดเดรสของ Y เมื่อ Y ได้รับคำร้องขอ **ARP** ซึ่งวางรายการ X ในแคช **ARP** (ถูกใช้เพื่อแม็พจาก IP แอดเดรสกับแอดเดรสของฮาร์ดแวร์อย่างรวดเร็ว) จากนั้นตอบกลับไปยัง X ด้วยการตอบกลับ **ARP** ที่มี **IP** ของ Y และ แอดเดรสของฮาร์ดแวร์ เมื่อโหนด X รับการตอบกลับ **ARP** ของ Y โหนดจะวางรายการสำหรับ Y ในแคช **ARP**

หากรายการแคช **ARP** มีอยู่ที่ X สำหรับ Y โหนด X สามารถส่งแพ็กเก็ตโดยตรงไปยัง Y โดยไม่มีการเรียงลำดับอีกครั้งกับ **ARP** (ยกเว้นรายการแคช **ARP** สำหรับ Y ถูกลบทิ้ง ซึ่งเป็นกรณีของ **ARP** ถูกใช้เพื่อติดต่อกับ Y)

ไม่เหมือนกับโปรโตคอลส่วนมาก แพ็กเก็ต **ARP** ไม่ได้มีส่วนหัวในรูปแบบคงที่ แต่ ข้อความจะถูกออกแบบให้ใช้ประโยชน์ด้วยความหลากหลายของเทคโนโลยีเน็ตเวิร์ก เช่น:

- Ethernet LAN adapter (สนับสนุน Ethernet และโปรโตคอล 802.3)
- เน็ตเวิร์กอะแด็ปเตอร์ของโทเค็นริง
- เน็ตเวิร์กอะแด็ปเตอร์ Fiber Distributed Data Interface (FDDI)

อย่างไรก็ตาม ARP ไม่ได้แปลแอดเดรสสำหรับ **Serial Line Interface Protocol (SLIP)** หรือ **Serial Optical Channel Converter (SOC)** เนื่องจากมีการเชื่อมต่อแบบ point-to-point

เคอร์เนลรักษาตารางการแปล และ **ARP** ไม่พร้อมใช้งานกับผู้ใช้หรือแอ็พพลิเคชัน เมื่อแอ็พพลิเคชันส่ง Internet packet ไปยังหนึ่งในอินเตอร์เฟสไดรเวอร์ ไดรเวอร์ร้องขอ การแม็พแอดเดรสที่เหมาะสม หากการแม็พไม่ได้อยู่ในตาราง แพ็กเก็ตการกระจายสัญญาณ **ARP** ถูกส่งผ่านไดรเวอร์อินเตอร์เฟสที่ร้องขอไปยังโฮสต์บน local area network

รายการในตารางการแม็พ **ARP** ถูกลบทิ้งหลังเวลา 20 นาที รายการที่ไม่เสร็จสิ้นจะถูกลบทิ้งภายใน 3 นาที หากต้องการสร้างรายการให้เป็นแบบถาวรในตารางการแม็พ **ARP** ให้ใช้คำสั่ง **arp** ด้วยพารามิเตอร์ *pub*:

```
arp -s 802.3 host2 0:dd:0:a:8s:0 pub
```

เมื่อโฮสต์ใดๆ ที่สนับสนุน **ARP** รับแพ็กเก็ตคำร้องขอ **ARP** โฮสต์จดบันทึก **IP** และแอดเดรสของฮาร์ดแวร์ของระบบที่ร้องขอและอัพเดตตารางการแม็พ หากจำเป็น หากได้รับโฮสต์ **IP** address ไม่ตรงกับแอดเดรสคำร้องขอ โฮสต์ละทิ้งแพ็กเก็ตที่ร้องขอ หากแอดเดรส **IP** ตรงกัน โฮสต์การรับส่งแพ็กเก็ตการตอบกลับ ไปยังระบบที่ร้องขอ ระบบการร้องขอเก็บการแม็พใหม่ และใช้เพื่อส่งผ่านแพ็กเก็ตอินเตอร์เน็ตที่คงค้างอยู่

Internet Control Message Protocol:

โปรโตคอลระดับเน็ตเวิร์กสำรองคือ **Internet Control Message Protocol (ICMP)** **ICMP** คือส่วนของการนำ **IP** ไปใช้งาน **ICMP** จัดการกับข้อผิดพลาดและข้อความควบคุมสำหรับ **IP**

โปรโตคอลนี้อนุญาตให้เกตเวย์และโฮสต์ส่งรายงานปัญหาไปยังเครื่อง ที่ส่งแพ็กเก็ต **ICMP** ทำขั้นตอนต่อไปนี้:

- ทดสอบว่าปลายทางใช้งานได้และเข้าถึงได้
- รายงานปัญหาเกี่ยวกับพารามิเตอร์พร้อมกับส่วนหัวของดาตาแกรม
- ดำเนินการกับการซิงโครไนซ์เวลาและส่งการประมาณการเวลา
- ขอรับอินเตอร์เน็ตแอดเดรสและ subnet masks

หมายเหตุ: **ICMP** ใช้ส่วนสนับสนุนพื้นฐานของ **IP** หากเป็นโปรโตคอล ระดับสูงกว่า อย่างไรก็ตาม **ICMP** คือส่วนที่สำคัญของ **IP** และต้องนำไปใช้กับทุกๆ โมดูล **IP**

ICMP จัดเตรียมการตอบกลับเกี่ยวกับปัญหาในสภาพแวดล้อมการสื่อสาร แต่ะไม่ทำให้ **IP** เชื่อถือได้ นั่นคือ **ICMP** ไม่ได้รับประกันว่า **IP** packet ถูกจัดส่งด้วยความเชื่อถือได้หรือข้อความ **ICMP** ถูกส่งคืนไปยังโฮสต์ต้นทางเมื่อ **IP** packet ไม่ได้ถูกส่งผ่านหรือ ส่งผ่านไม่ถูกต้อง

ข้อความ **ICMP** อาจถูกส่งในสถานการณ์ต่อไปนี้:

- เมื่อแพ็กเก็ตไม่สามารถเข้าถึงปลายทางได้
- เมื่อโฮสต์เกตเวย์ไม่ได้มีความสามารถในการบัฟเฟอร์เพื่อส่งต่อ แพ็กเก็ต
- เมื่อเกตเวย์สามารถส่งไปยังโฮสต์เพื่อส่งทราฟิกบนเราต์ที่สั้นกว่า

TCP/IP ส่งและรับชนิดข้อความ ICMP จำนวนมาก (โปรดดู “ชนิดข้อความ Internet Control Message Protocol” ในหน้า 156) **ICMP** ถูกฝังอยู่ในเคอร์เนล และไม่มี application programming interface (API) ที่จัดเตรียมโปรโตคอลนี้ไว้

ชนิดข้อความ Internet Control Message Protocol:

ICMP ส่งและรับชนิดข้อความเหล่านี้

ไอเท็ม	คำอธิบาย
echo request	ส่งโดยโฮสต์และเกตเวย์เพื่อทดสอบปลายทางว่ายังคงอยู่ และสามารถเข้าถึงได้
information request	ส่งโดยโฮสต์และเกตเวย์เพื่อขอรับอินเตอร์เน็ตแอดเดรสสำหรับเน็ตเวิร์ก ที่ต้องถูกพ่วงต่อ ชนิดข้อความนี้ถูกส่งพร้อมกับส่วนของเน็ตเวิร์กของ **IP** address ปลายทางที่ตั้งค่าเป็น 0
timestamp request	ส่งไปยังคำร้องขอเครื่องปลายทางที่ส่งคืนค่าปัจจุบัน สำหรับเวลาของวัน
address mask request	ส่งโดยโฮสต์เพื่อเรียนรู้ subnet mask โฮสต์สามารถส่งไปยังเกตเวย์ หากรู้แอดเดรสของเกตเวย์ หรือส่งข้อความกระจายสัญญาณ
destination unreachable	ส่งเมื่อเกตเวย์ไม่สามารถส่งดาตาแกรม **IP**
source quench	ส่งโดยละทิ้งเครื่องเมื่อดาตาแกรมมาถึงเร็วเกินไปสำหรับ เกตเวย์เพื่อประมวลผล หากคำร้องขอแหล่งที่มาต้นทางช้ากว่า อัตราส่วนของดาตาแกรมที่ส่ง
redirect message	ส่งเมื่อเกตเวย์ตรวจพบโฮสต์กำลังใช้เราต์ อย่างไม่มีประสิทธิภาพ
echo reply	ส่งโดยเครื่องที่ได้รับคำร้องขอ echo ในการตอบกลับไปยังเครื่อง ซึ่งส่งคำร้องขอ
information reply	ส่งเกตเวย์ตอบกลับไปยังคำร้องขอสำหรับแอดเดรสเน็ตเวิร์ก พร้อมกับฟิลด์ต้นทางและปลายทางของดาตาแกรม **IP** ที่ระบุ
timestamp reply	ส่งพร้อมกับค่าของเวลาของวันปัจจุบัน
address mask reply	ส่งไปยังเครื่องที่ร้องขอ subnet masks
parameter problem	ส่งเมื่อโฮสต์หรือเกตเวย์ค้นหาปัญหาด้วยส่วนหัวของดาตาแกรม
time exceeded	ส่งเมื่อต่อไปนี้เป็นจริง: • แต่ละดาตาแกรม **IP** มีตัวนับ time-to-live (จำนวน hop) ที่ลดเกตเวย์แต่ละตัว • เกตเวย์ละทิ้งดาตาแกรมเนื่องจากจำนวน hop เข้าถึงค่า 0
Internet Timestamp	ใช้เพื่อเร็กคอร์ดการประทับเวลาผ่านเราต์

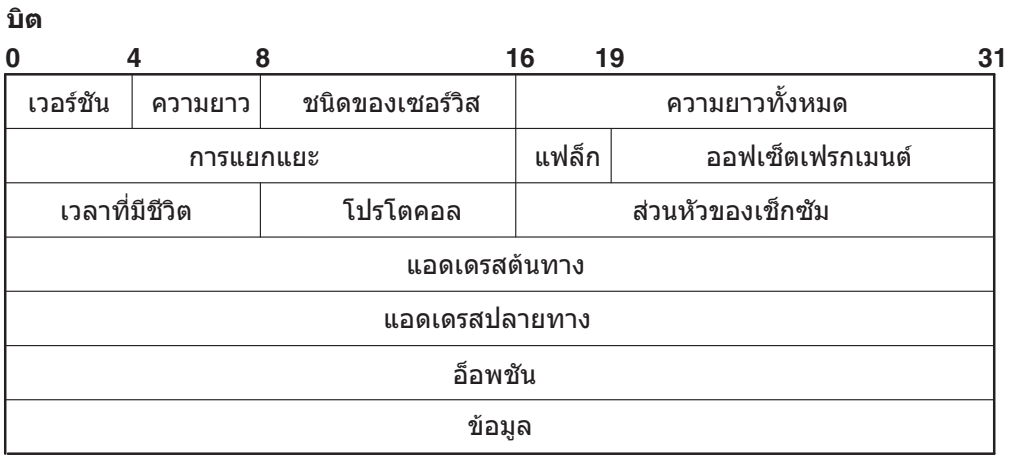
Internet Protocol:

โปรโตคอลระดับของเน็ตเวิร์กที่สามคือ **อินเตอร์เน็ตโปรโตคอล (IP)** ซึ่งจัดเตรียมการส่งมอบแพ็กเก็ตที่เชื่อถือไม่ได้ ซึ่งไม่ได้เชื่อมต่อสำหรับ อินเตอร์เน็ต

IP คือการเชื่อมต่อแบบ connectionless เนื่องจากใช้ข้อมูลแต่ละแพ็กเก็ต อย่างเป็นอิสระ และไม่สามารถเชื่อถือได้เนื่องจากไม่มีการรับประกันในการส่งมอบ ซึ่งหมายความว่า ไม่ต้องการการตอบรับจากโฮสต์การส่ง โฮสต์การรับ หรือโฮสต์ระดับกลาง

IP จัดเตรียมอินเตอร์เฟสไปยังโปรโตคอล ระดับของเน็ตเวิร์กอินเตอร์เฟส การเชื่อมต่อแบบฟิสิคัลของข้อมูลการถ่ายโอนเน็ตเวิร์ก ในกรอบที่มีส่วนหัวและข้อมูล ส่วนหัวมีแอดเดรสต้นทาง และแอดเดรสปลายทาง **IP** ใช้อินเตอร์เน็ตดาตาแกรม ที่มีข้อมูลที่คล้ายกับกรอบแบบฟิสิคัล The datagram also has a header containing Internet Protocol addresses of both source and destination of the data.

IP กำหนดรูปแบบของข้อมูลทั้งหมดที่ส่งผ่านอินเตอร์เน็ต



รูปที่ 10. ส่วนหัวแพ็กเก็ตของอินเตอร์เน็ตโปรโตคอล

รูปประกอบนี้แสดงส่วนหัวแพ็กเก็ต IP ทั่วไป 32 บิตแรก ตารางด้านล่างแสดงเอ็นทิตีต่างๆ

นิยามฟิลด์ส่วนหัว IP

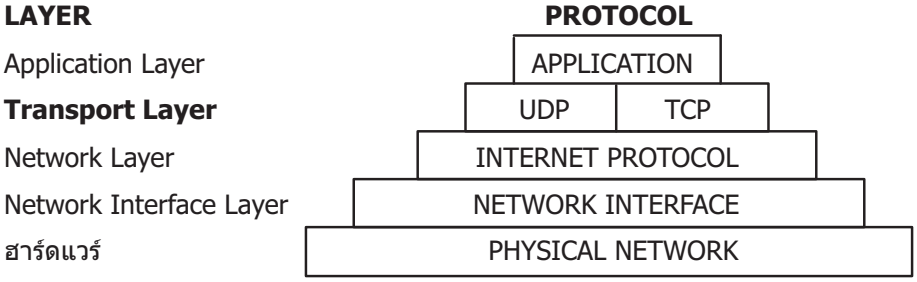
ไอเท็ม	คำอธิบาย
Version	ระบุเวอร์ชันของ **IP** ที่ใช้ เวอร์ชันปัจจุบัน ของ **IP** โปรโตคอลคือ 4
ความยาว	ระบุความยาวส่วนหัวดาตาแกรมที่วัดในหน่วนคำที่มีขนาด 32 บิต
ชนิดของเซอร์วิส	มีห้าฟิลด์ย่อยที่ระบุชนิดของการนำหน้า หน่วงเวลา ปริมาณงาน และความเชื่อถือได้ที่ต้องการ สำหรับแพ็กเก็ต (อินเตอร์เน็ต ไม่ได้รับประกันคำร้องขอนี้) ค่ากำหนดดีฟอลตีสำหรับห้าฟิลด์ย่อยเหล่านี้ คือการนำหน้ารูทีน หน่วงเวลาปกติ ปริมาณงานปกติ และความเชื่อถือได้ตามปกติ ฟิลด์นี้ไม่ได้ถูกใช้โดย อินเตอร์เน็ต ในเวลานี้ การนำ **IP** ไปใช้งานคอมไพล์ด้วยข้อกำหนดของข้อกำหนดคุณสมบัติ **IP** นั่นคือ RFC 791 *อินเตอร์เน็ตโปรโตคอล*
ความยาวทั้งหมด	ระบุความยาวของดาตาแกรมที่สอดแทรกทั้งส่วนหัว และข้อมูลที่วัดได้ใน octet การแตกแฟรกเมนต์ของ เกตเวย์ ที่มีการประกอบขึ้นใหม่ที่ปลายทางถูกจัดเตรียมไว้ ความยาวทั้งหมดของ แพ็กเก็ต **IP** สามารถถูก กำหนดคอนฟิกในลักษณะ interface-by-interface ด้วยคำสั่ง **ifconfig** หรือ System Management Interface Tool (SMIT) fast path, `smit chinet` ใช้ SMIT เพื่อตั้งค่าถาวรในฐานข้อมูลการกำหนดคอนฟิก ใช้คำสั่ง **ifconfig** เพื่อตั้งค่าหรือเปลี่ยนแปลงค่า ในระบบที่กำลังรัน
Identification	มีเลขจำนวนเต็มเฉพาะที่ระบุดาตาแกรม
แฟล็ก	ควบคุมการแตกแฟรกเมนต์พร้อมกับฟิลด์ Identification แฟล็กการแตกแฟรกเมนต์ระบุว่า ดาตาแกรม สามารถแตกแฟรกเมนต์ และแฟรกเมนต์ปัจจุบันคือแฟรกเมนต์ล่าสุด
ออฟเซ็ตการแตกแฟรกเมนต์	ระบุออฟเซ็ตของแฟร็กเมนต์นี้ในดาตาแกรมต้นฉบับ ที่วัดได้ในหน่วย 8 octet
Time to Live	ระบุระยะเวลาที่ดาตาแกรมสามารถคงอยู่บนอินเตอร์เน็ตได้ซึ่งจะเก็บดาตาแกรมที่มีเราต์ที่ไม่ถูกต้องจาก ที่ยังคงอยู่บนอินเตอร์เน็ต ค่าดีฟอลต์ของ time to live คือ 255 วินาที
Protocol	ระบุชนิดของโปรโตคอลระดับสูง
ส่วนหัวของเช็กซัม	บ่งชี้จำนวนที่คำนวณได้เพื่อทำให้มั่นใจถึง integrity ของค่าส่วนหัว
แอดเดรสต้นทาง	ระบุอินเตอร์เน็ตแอดเดรสของโฮสต์การส่ง
แอดเดรสปลายทาง	ระบุอินเตอร์เน็ตแอดเดรสของโฮสต์การรับ

ไอเท็ม อ็อพชัน	คำอธิบาย จัดเตรียมการทดสอบเน็ตเวิร์กและการดีบัก ฟิลด์นี้ไม่จำเป็นต้องมี สำหรับดาตาแกรมทุกๆ ตัว
	ส่วนท้ายของรายการอ็อพชัน ระบุส่วนท้ายของรายการอ็อพชัน ซึ่งถูกใช้ที่ส่วนท้ายของอ็อพชันสุดท้าย ซึ่งไม่ใช่ที่ส่วนท้ายของแต่ละอ็อพชัน อ็อพชันนี้ ควรถูกใช้เฉพาะหากส่วนท้ายของอ็อพชันจะไม่ได้เกิดขึ้นพร้อมกัน กับส่วนท้ายของส่วนหัว **IP** ส่วนท้ายของรายการอ็อพชัน ถูกใช้หากอ็อพชันมาความยาวของดาตาแกรมมากเกินไป
	ไม่มีการดำเนินการ จัดเตรียมการจัดตำแหน่งระหว่างอ็อพชันอื่นๆ ตัวอย่างเช่น การจัดตำแหน่งจุดเริ่มต้นของอ็อพชันตามลำดับบนขอบเขต 32 บิต
	การปล่อยเราต์ต้นทางและการบันทึกเราต์ จัดเตรียมความหมายสำหรับแหล่งที่มาของอินเตอร์เน็ตดาตาแกรม เพื่อจัดหาข้อมูลการเราต์ที่ใช้โดยเกตเวย์ในการส่งผ่านดาตาแกรม ไปยังปลายทางและในการบันทึกข้อมูลการเราต์ นี่คือ *การปล่อย* เราต์ต้นทาง: เกตเวย์หรือโฮสต์ **IP** ถูกอนุญาตให้ใช้เราต์ใดๆ ของจำนวนเกตเวย์ระดับกลางใดๆ เพื่อเข้าถึง แอดเดรสถัดไปในเราต์
	การจำกัดเราต์ต้นทางและการบันทึกเราต์ จัดเตรียมความหมายสำหรับแหล่งที่มาของอินเตอร์เน็ตดาตาแกรม เพื่อจัดหาข้อมูลการเราต์ที่ใช้โดยเกตเวย์ในการส่งผ่านดาตาแกรม ไปยังปลายทางและในการบันทึกข้อมูลการเราต์ นี่คือ *จำกัด* เราต์ต้นทาง : หากต้องการเข้าถึงเกตเวย์ถัดไปหรือโฮสต์ที่ระบุในเราต์ เกตเวย์หรือโฮสต์ **IP** ต้องส่งดาตาแกรมโดยตรงไปยัง แอดเดรสถัดไปในเราต์ต้นทางและเฉพาะกับเน็ตเวิร์กการเชื่อมต่อโดยตรง ที่ถูกบ่งชี้ในแอดเดรสถัดไป
	การบันทึกเราต์ จัดเตรียมความหมายให้กับเร็กคอร์ดการเราต์ของอินเตอร์เน็ตดาตาแกรม
	ตัวระบุสตรีม จัดเตรียมวิธีการสำหรับตัวระบุสตรีมที่ต้องถูกใช้ผ่านเน็ตเวิร์ก ที่ไม่สนับสนุนแนวคิดแบบสตรีม
	การประทับเวลาอินเตอร์เน็ต จัดเตรียมเร็กคอร์ดของการประทับเวลาผ่านเราต์

แพ็กเก็ตขาออกมีส่วนหัว **IP** แบบอัตโนมัติที่นำหน้า แพ็กเก็ตเหล่านั้น แพ็กเก็ตขาเข้ามีส่วนหัว **IP** ถูกลบทิ้งก่อน ที่จะส่งไปยังโปรโตคอลระดับที่สูงกว่า โปรโตคอล **IP** จัดเตรียมสำหรับการกำหนดแอดเดรสสากลของโฮสต์ในอินเตอร์เน็ตเน็ตเวิร์ก

Internet Transport-Level Protocols

โปรโตคอล **TCP/IP** transport–level อนุญาตให้แอ็พพลิเคชันโปรแกรม สื่อสารกับแอ็พพลิเคชันโปรแกรมอื่น



รูปที่ 11. เลเยอร์ Transport ของ TCP/IP Suite of Protocols.

รูปภาพประกอบนี้แสดงเลเยอร์ต่างๆ ของ **TCP/IP** Suite of Protocols จากด้านบนสุด เลเยอร์แอ็พพลิคเชันสอดคล้องกับ แอ็พพลิเคชัน เลเยอร์ transport มี **UDP** และ **TCP** เลเยอร์ network มีเน็ตเวิร์กอินเตอร์เฟส (ฮาร์ดแวร์) และท้ายสุด เลเยอร์ฮาร์ดแวร์มีฟิสิคัลเน็ตเวิร์ก

User Datagram Protocol (**UDP**) และ **TCP** มีโปรโตคอล transport-level พื้นฐานสำหรับการสร้างการเชื่อมต่อระหว่าง อินเตอร์เน็ตโฮสต์ ทั้ง **TCP** และ **UDP** อนุญาตให้โปรแกรมส่งข้อความ และรับข้อความจากแอ็พพลิคเชันบนโฮสต์อื่น เมื่อแอ็พพลิเคชันส่งคำร้องขอไปยังเลเยอร์ Transport เพื่อส่งข้อความ **UDP** และ **TCP** แตกข้อมูลลงในแพ็กเก็ต เพิ่มส่วนหัวแพ็กเก็ตซึ่งรวมถึงแอดเดรสปลายทาง และส่งข้อมูล ไปยังเลเยอร์ Network เพื่อประมวลผลเพิ่มเติม ทั้ง **TCP** และ **UDP** ใช้พอร์ตโปรโตคอลบนโฮสต์เพื่อระบุปลายทางเฉพาะของ ข้อความ

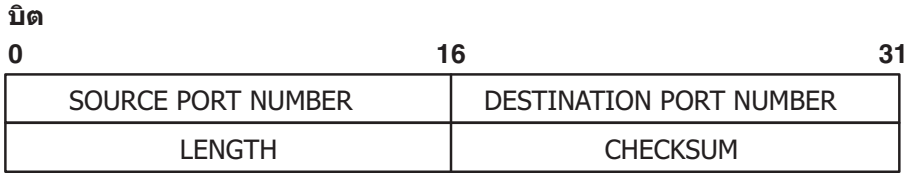
โปรโตคอลระดับที่สูงกว่าและแอ็พพลิคเชันใช้ **UDP** เพื่อทำให้การเชื่อมต่อ datagram และ **TCP** ทำการเชื่อมต่อกับสตรีม อินเตอร์เฟส ซ็อกเก็ตระบบปฏิบัติการใช้โปรโตคอลเหล่านี้

User Datagram Protocol:

บางครั้งแอ็พพลิเคชันบนเน็ตเวิร์กต้องการส่งข้อความ ไปยังแอ็พพลิเคชันเฉพาะหรือกระบวนการบนเน็ตเวิร์กอื่นๆ **UDP** จัดเตรียมดาตาแกรมของการสื่อสารระหว่างแอ็พพลิเคชันบนอินเตอร์เน็ต โฮสต์

เนื่องจากผู้ส่งไม่รู้จักกระบวนการที่แอ็คทีฟที่กำหนดไว้ในขณะนั้น **UDP** ใช้พอร์ตโปรโตคอล (หรือจุดปลายทางแบบย่อ ภายในเครื่อง) ซึ่งระบุโดยตัวเลขบวก เพื่อส่งข้อความไปยังหนึ่งในปลายทางจำนวนมากบนโฮสต์ พอร์ตโปรโตคอล รับและจัดการกับข้อความในคิวจนกว่าแอ็พพลิเคชันบนการรับเน็ตเวิร์ก สามารถเรียกคืนได้

เนื่องจาก **UDP** อยู่ภายใต้ **IP** เพื่อส่งดาตาแกรม **UDP** จัดเตรียมข้อความ connectionless การส่งมอบเป็น **IP** ซึ่งนำเสอนความไม่แน่นอนของดาตาแกรมที่ส่งมอบหรือทำซ้ำ การปกป้อง อย่างไรก็ตาม **UDP** อนุญาตให้ผู้ใช้ระบุหมายเลขพอร์ตต้นทาง และปลายทางสำหรับข้อความและคำนวณเช็กซัม ของข้อมูลและส่วนหัว คุณลักษณะทั้งสองนี้อนุญาตให้แอ็พพลิเคชันการส่ง และรับมั่นใจได้ว่า การส่งมอบข้อความนั้นถูกต้อง



รูปที่ 12. User Datagram Protocol (UDP) packet header

รูปภาพประกอบแสดง 32 บิตแรกของส่วนหัวแพ็กเก็ต **UDP** 16 บิตแรกมีหมายเลขพอร์ตต้นทางและความยาว 16 บิตต่อมามีหมายเลขพอร์ตปลายทางและเช็กซัม

แอ็พพลิเคชันที่ต้องการการส่งมอบที่เชื่อถือได้ของดาตาแกรมต้องใช้ ความเชื่อถือได้ในการตรวจสอบเมื่อใช้ **UDP** แอ็พพลิเคชันที่ต้องการ การส่งมอบที่เชื่อถือได้ของข้อมูลสตรีมควรใช้ **TCP**

นิยามฟิลด์ส่วนหัวของ UDP

ไอเท็ม	คำอธิบาย
Source Port Number	แอดเดรสของพอร์ตโปรโตคอลที่ส่งข้อมูล
Destination Port Number	แอดเดรสของพอร์ตโปรโตคอลที่รับข้อมูล
ความยาว	ความยาวในหน่วย octets ของดาตาแกรม **UDP**
Checksum	จัดเตรียมการตรวจสอบเกี่ยวกับดาตาแกรม **UDP** โดยใช้อัลกอริทึมเดียวกันกับ **IP**

applications programming interface (API) กับ **UDP** คือชุดของรูทีนย่อยไลบรารีที่จัดเตรียมโดยอินเตอร์เฟสซ็อกเก็ต

Reliable Datagram Sockets บน InfiniBand และ RoCE:

Reliable Datagram Sockets (RDS) คือโปรโตคอลที่ไม่มีการเชื่อมต่อหรือมุ่งเน้นที่เร็กคอร์ด ซึ่งจัดเตรียมเซอร์วิสตามลำดับและไม่ซ้ำกันบน InfiniBand และ RDMA บน Converged Ethernet (RoCE) RDS แสดงชุดย่อย User Datagram Protocol (UDP) ของซ็อกเก็ต API

RDS เป็นส่วนหนึ่งของโดเมน **AF_BYPASS** ที่ใช้สำหรับโปรโตคอลซึ่งจะเลี่ยงเคอร์เนล TCP/IP สแต็ก

ระบบปฏิบัติการ AIX มี RDS สองเวอร์ชันคือ: RDSv2 และ RDSv3 RDSv3 คือเวอร์ชัน ล่าสุดและมีการสนับสนุน Remote
| Direct Memory Access (RDMA) RDSv3 บน AIX 7.2 และเวอร์ชันถัดมา สนับสนุน Open Fabrics Enterprise Distribution
| (OFED) อ้างอิงตาม RDMA บน Converged Ethernet (RoCE)

การสร้างซ็อกเก็ต RDS: เมื่อต้องการสร้างซ็อกเก็ต RDS ให้เรียกใช้การเรียกระบบ **socket( )** โดยเพิ่มบรรทัดต่อไปนี้ลงในแอ็พพลิเคชันโปรแกรม:

```
#include <sys/bypass.h>
include <net/rds_rdma.h>          /* for RDSv3 only */
sock = socket (AF_BYPASS, SOCK_SEQPACKET,BYPASSPROTO_RDS);
```

ถ้าโปรโตคอล **BYPASSPROTO_RDS** เป็นโปรโตคอลเดทาแกรมเพียงอย่างเดียวที่เชื่อถือได้ ซึ่งได้รับการสนับสนุนในตระกูล **AF_BYPASS** คุณยังสามารถเรียก การเรียกระบบ **socket( )** ดังนี้:

```
sock = socket (AF_BYPASS, SOCK_SEQPACKET,0);
```

การเรียกระบบ

RDS ยังสนับสนุนการเรียกระบบ ต่อไปนี้ด้วย:

- blind()
- close()
- getsockopt()
- recvform()
- recvmsg()
- sendmsg()
- sendto()
- setsockopt()

นอกจากนั้น RDSv3 ยังสนับสนุนการเรียกระบบต่อไปนี้ด้วย:

- connect()
- read()
- recv()
- send()
- write()

หมายเหตุ: แม้ว่าซ็อกเก็ต RDS ไม่มีการเชื่อมต่อ แต่การเรียกระบบ **connect( )** ได้รับการสนับสนุนจาก RDSv3 อย่างไรก็ตาม ในกรณีนี้ **connect( )** ไม่ได้ สร้างเอนทิตีการเชื่อมต่อระดับซ็อกเก็ตระหว่างจุดปลาย RDS สองจุด เพียงแต่เชื่อมโยงจุดปลายของปลายทางดีฟอลต์กับซ็อกเก็ตเท่านั้น ด้วยเหตุผลนี้ การเรียกระบบ **listen( ), accept( )** และ **shutdown( )** จึงไม่ได้รับการสนับสนุนสำหรับซ็อกเก็ต RDS

ยูทิลีตี rdsctrl สำหรับ RDSv2: ใช้ยูทิลีตี **rdsctrl** (`/usr/sbin/rdsctrl`) เพื่อเปลี่ยนการปรับและการวินิจฉัยสำหรับสถิติ RDS สำหรับ RDSv2 สามารถใช้ยูทิลีตีหลังจากโหลด RDS แล้ว (**bypassctrl load rds**) สำหรับข้อมูลเพิ่มเติมเกี่ยวกับยูทิลีตีนี้ ให้รันคำสั่ง rdsctrl โดยไม่มีอาร์กิวเมนต์

สถิติ

เมื่อต้องการแสดงสถิติ RDS ต่างๆ ให้รันคำสั่ง # `rdsctrl stats`

เมื่อต้องการรีเซ็ตสถิติ ให้รันคำสั่ง # `rdsctrl stats reset`

พารามิเตอร์ที่ปรับได้

พารามิเตอร์ RDS ต่อไปนี้สามารถปรับได้หลังจากโหลด RDS แล้ว แต่ก่อนจะรันแอ็พพลิเคชัน RDS:

rds_sendspace

ระบุเครื่องหมาย high-water ของบัฟเฟอร์การส่งสำหรับแต่ละโฟลว์ แต่ละ ซ็อกเก็ตอาจมีหลายโฟลว์ ค่าดีฟอลต์คือ 524288 ไบต์ (512 KB) ค่ามีการตั้งโดยใช้คำสั่งต่อไปนี้: # `rdsctrl set rds_sendspace=`<*ค่าในหน่วยไบต์*>

rds_recvspace

ระบุเครื่องหมาย high-water สำหรับแต่ละโฟลว์ของบัฟเฟอร์การรับสำหรับแต่ละ ซ็อกเก็ต สำหรับทุกโฟลว์เพิ่มเติม

ในซ็อกเก็ตนี้ เครื่องหมาย **high-water ของการได้รับ** เพิ่มขึ้นตามค่านี้ ค่าดีฟอลต์คือ 524288 ไบต์ (512 KB) ค่ามีการตั้งโดยใช้คำสั่งต่อไปนี้: # `rdsctrl set rds_recvspace=<value in bytes>`

หมายเหตุ: เพื่อให้ได้ ประสิทธิภาพการสตรีม RDS ที่ดีขึ้น อย่างน้อยที่สุด ค่าของพารามิเตอร์ **rds_sendspace** และพารามิเตอร์ **rds_recvspace** ต้องเป็นค่าของขนาด RDS **sendmsg( )** ที่ใหญ่ที่สุดคูณด้วยสี่ RDS จะส่ง ACK สำหรับแต่ละชุดของสี่ข้อความที่ได้รับ ถ้า **rds_recvspace** ไม่ใหญ่กว่า ขนาดข้อความอย่างน้อยสี่เท่า ผลผลิต จะต่ำมาก

rds_mclustsize

ระบุขนาดของแต่ละคลัสเตอร์หน่วยความจำ ซึ่งเป็น ขนาดของแฟรกเมนต์ข้อความด้วย ขนาดดีฟอลต์คือ 16384 ไบต์ (16 KB) ค่าซึ่งเป็นผลคูณของ 4096 เสมอ มีการตั้งค่าโดยใช้คำสั่ง ต่อไปนี้: # `rdsctrl set rds_mclustsize=<ผลคูณ ของ 4096 ในหน่วย ไบต์>`

ข้อควรสนใจ: ค่า **rds_mclustsize** ต้องเหมือนกันบนระบบทั้งหมด (โหนด) ในคลัสเตอร์ การเปลี่ยนค่านี้ ยังเกี่ยวข้องกับประสิทธิภาพด้วย

ค่าปัจจุบันของพารามิเตอร์ก่อนหน้านี้สามารถดึงข้อมูลได้ โดยใช้คำสั่ง # `rdsctrl get <parameter>`

เมื่อต้องการรับรายการของค่าที่ปรับได้ทั้งหมดและค่า ให้รันคำสั่ง # `rdsctrl get`

ยูทิลีตี rdsctrl สำหรับ RDSv3: สำหรับ RDSv3 คำสั่ง **rdsctrl** สนับสนุนอ็อพชัน อ็อพชันเหล่านี้แสดงรายการที่นี่:

ไอเท็ม	คำอธิบาย
วิธีใช้ [<***ชื่อค่าที่ปรับได้***>]	อ็อพชัน **วิธีใช้** แสดงข้อความคำอธิบายของค่าที่ปรับได้ของ RDSv3 ซึ่งระบุไว้ ถ้า ไม่ได้ระบุค่าที่ปรับได้ อ็อพชันนี้แสดงรายการของค่าที่ปรับได้ทั้งหมด ซึ่งได้รับการสนับสนุนสำหรับ RDSv3 พร้อมกับคำอธิบายของแต่ละค่าที่ปรับได้
ตั้งค่า [**-p**]{<***ชื่อค่าที่ปรับได้***> = <***ค่า***>}	อ็อพชัน **ตั้งค่า** ตั้งค่าของค่าที่ปรับได้ของ RDSv3 ซึ่งระบุไว้ อ็อพชันนี้จะตรวจสอบว่า ผู้ใช้มีสิทธิที่จำเป็นในการป้องกันไม่ให้ผู้ใช้ที่ไม่ได้รับอนุญาต เปลี่ยนค่าที่ปรับได้ของ RDS และยังตรวจสอบความถูกต้องของช่วงสำหรับค่าของ ค่าที่ปรับได้ใหม่ด้วย แฟล็ก **-p** ทำการกำหนด แบบถาวรบนการดำเนินการรีบูต
รับ [<***ชื่อค่าที่ปรับได้***>]	อ็อพชัน **รับ** รับค่าปัจจุบันของค่าที่ปรับได้ซึ่งเดียวรี เมื่อไม่ได้ระบุฟิลด์ชื่อ ในคำสั่งนี้ คำสั่งจะส่งคืนค่าปัจจุบันของค่าที่ปรับได้ของ RDS ทั้งหมดซึ่งพร้อมใช้งาน
ดีฟอลต์ [**-p**] [<***ชื่อค่าที่ปรับได้***>]	อ็อพชัน **ดีฟอลต์** มีการ ใช้เพื่อรีเซ็ตค่าที่ปรับได้เป็นค่าดีฟอลต์ เมื่อระบุฟิลด์ ชื่อ ระบบจะรีเซ็ตเฉพาะค่าที่ปรับได้นั้น ถ้าไม่ได้ระบุฟิลด์ชื่อ คำสั่งนี้จะรีเซ็ตค่าที่ปรับได้ทั้งหมดเป็นค่าดีฟอลต์ อ็อพชันนี้ ยังเป็นวิธีในการทำการเปลี่ยนแปลงอย่างถาวรเมื่อรีบูต โดยใช้แฟล็ก **-p**
load [`ofed|aixib`]	อ็อพชัน **โหลด** จะโหลดส่วนขยายเคอร์เนล RDSv3 (ถ้ายังไม่ได้โหลด) อาร์กิวเมนต์ `ofed` โหลดส่วนขยายเคอร์เนลใน RDSv3 บน OFED verbs ในโหมด RoCE อาร์กิวเมนต์ `aixib` โหลดส่วนขยายเคอร์เนลใน RDSv3 ในโหมด InfiniBand การระบุอาร์กิวเมนต์สำหรับอ็อพชัน `load` เป็นตัวเลือก อ็อพชัน `load` เป็นดีฟอลต์ของอาร์กิวเมนต์ `aixib` เมื่อไม่ได้ระบุอาร์กิวเมนต์ไว้ ตามค่าดีฟอลต์แล้ว ยูทิลีตี **rdsctrl** โหลดอุปกรณ์ InfiniBand ยกเว้นว่าระบุแอ็ตทริบิวต์ใหม่ (`ofed`) ที่บรรทัดรับคำสั่ง
ยกเลิกการโหลด	อ็อพชัน **ยกเลิกการโหลด** มีการ ใช้เพื่อยกเลิกการโหลดส่วนขยายเคอร์เนล RDSv3

ไอเท็ม	คำอธิบาย
ras [**-p**] <***minimal*** / ***normal*** / ***detail*** / ***maximal***>	อ็อพชัน **ras** ตั้งค่าระบบปฏิบัติการ AIX สำหรับ ค่าติดตั้งการติดตาม RAS และการตรวจสอบข้อผิดพลาดของ RDSv3 เป็นระดับที่ ระบุไว้ คำสั่งนี้เรียกคำสั่งของระบบปฏิบัติการ **errctrl** และ **ctctrl** AIX แบบภายใน แฟล็ก **-p** ทำให้ค่าติดตั้งมีอยู่ในการดำเนินการรีบูต
ras extract	อ็อพชัน **ras extract** ดัมพ์เนื้อหาของข้อผิดพลาด RAS และบัฟเฟอร์การติดตามที่ไม่มีข้อผิดพลาดสำหรับ RDS ไปยังเอาต์พุตมาตรฐาน
info [<***flags***>]	อ็อพชัน **info** เป็นสมนามสำหรับคำสั่ง **rds-info**
ping [<***IP v4 address***>]	อ็อพชัน **ping** เป็นสมนามสำหรับคำสั่ง **rds-ping**
conn <***restart*** / ***kill***> <***source IP address***> <***destination IP address***>	อ็อพชัน **conn** รีสตาร์ทการเชื่อมต่อ RDS ที่ระบุ (อ็อพชันย่อย **รีสตาร์ท**) หรือสิ้นสุดการเชื่อมต่อ RDS ที่ระบุอย่างถาวร (อ็อพชันย่อย **kill**) การเชื่อมต่อ RDS ที่จะรีสตาร์ทหรือสิ้นสุดมีการระบุโดยให้ IP แอดเดรสของโหนดแบบโลคัลและแบบรีโมตสำหรับการเชื่อมต่อ การรีสตาร์ทการเชื่อมต่อจะดร็อปการเชื่อมต่อ InfiniBand ที่เกี่ยวข้อง และพยายามสร้างการเชื่อมต่อขึ้นอีกครั้ง ในทางตรงกันข้าม การสิ้นสุด การเชื่อมต่อ (อ็อพชันย่อย kill) จะดร็อปการเชื่อมต่อ InfiniBand ที่เกี่ยวข้องและยกเลิกการจัดสรรรีซอร์สทั้งหมดซึ่งเชื่อมโยงกับ การเชื่อมต่อ RDS ที่สอดคล้องกัน
เริ่มต้น การติดตาม <***พาธไปยังไฟล์การติดตาม***> <***ข้อมูล สูงสุดที่ดักจับต่อแฟรกเมนต์ RDS***>	อ็อพชัน **เริ่มต้นการติดตาม** เริ่มต้นเซสชันการติดตามเพื่อดักจับทราฟฟิก over-the-wire สำหรับโปรโตคอล RDSv3 ข้อความ RDSv3 มีการส่งผ่านในแฟรกเมนต์ แต่ละแฟรกเมนต์ RDS ที่ส่งผ่านหรือที่ได้รับถูกดักจับเป็นแพ็กเก็ตการติดตามใน ไฟล์การติดตามที่ระบุ สำหรับแต่ละแฟรกเมนต์ RDS ระบบดักจับเพย์โหลด ได้มากถึง <*ข้อมูลสูงสุดที่ดักจับต่อแฟรกเมนต์ RDS*> ไบต์ เฉพาะผู้ใช้ที่มีสิทธิพิเศษสามารถติดตามทราฟฟิก RDS และสามารถมีเซสชันการติดตาม ที่แอ็คทีฟได้ครั้งละหนึ่งเซสชันเท่านั้น
หยุดการติดตาม	อ็อพชัน **หยุดการติดตาม** สิ้นสุดเซสชันการติดตามที่เริ่มต้นก่อนหน้านี้โดยคำสั่ง **เริ่มต้นการติดตาม** อ็อพชันปิดไฟล์การติดตามที่เชื่อมโยงกับเซสชันการติดตาม หลังจากคำสั่งนี้ สามารถใช้คำสั่ง **รายงานการติดตาม** เพื่อสร้างรายงานข้อความของไฟล์การติดตาม
รายงาน การติดตาม <***พาธไปยังไฟล์การติดตาม***>	อ็อพชัน **รายงานการติดตาม** พิมพ์รายงานข้อความไปยังเอาต์พุตมาตรฐาน จากไฟล์การติดตามโปรโตคอล RDS ที่ดักจับไว้ก่อนหน้านี้
เวอร์ชัน	อ็อพชัน **เวอร์ชัน** พิมพ์เวอร์ชันโปรโตคอล RDS ที่โหลดในปัจจุบันในระบบ

ค่าที่ปรับได้ของ RDSv3: เมื่อต้องการดูรายการของค่าที่ปรับได้ซึ่งได้รับการสนับสนุนสำหรับ RDSv3 ให้รัน คำสั่ง **rdsctrl help** โดยไม่มีอาร์กิวเมนต์

RDMA API (RDSv3 เท่านั้น): โมเดลการเขียนโปรแกรมสำหรับการทำงานบน RDMA ด้วยซ็อกเก็ต RDS สร้างขึ้นจากข้อมูล โมเดลไคลเอ็นต์/เซิร์ฟเวอร์ ไคลเอ็นต์ RDMA คือแอ็พพลิเคชันที่ เริ่มต้นการดำเนินการอ่านหรือเขียน RDMA จากเซิร์ฟเวอร์ RDMA ที่ระบุ เซิร์ฟเวอร์ RDMA คือแอ็พพลิเคชันที่ประมวลผลการโอนย้ายข้อมูล RDMA การดำเนินการ อ่าน RDMA คือการโอนย้ายข้อมูลจากพื้นที่แอดเดรสของไคลเอ็นต์ ไปยังพื้นที่แอดเดรสของเซิร์ฟเวอร์ ในขณะที่การดำเนินการเขียน RDMA คือ การโอนย้ายข้อมูลจากพื้นที่แอดเดรสของเซิร์ฟเวอร์ไปยังพื้นที่แอดเดรสของไคลเอ็นต์ ในกรณีอย่างใดอย่างหนึ่ง ข้อมูลมีการโอนย้ายโดยตรงระหว่างหน่วยความจำพื้นที่ผู้ใช้ บนทั้งสองด้าน โดยไม่มีการคัดลอกไปยังหน่วยความจำพื้นที่เคอร์นัลบนด้านใด ด้านหนึ่ง

แอ็พพลิเคชันไคลเอ็นต์ RDMA สามารถเริ่มต้นการดำเนินการอ่านหรือเขียน RDMA โดยส่งคำร้องขอระดับแอ็พพลิเคชันพร้อมกับคุกกี้ RDMA ไปยังแอ็พพลิเคชันเซิร์ฟเวอร์ RDMA คำร้องขอระดับแอ็พพลิเคชันต้อง ระบุว่าการดำเนินการเป็นการดำเนินการอ่านหรือเขียน RDMA ตลอดจน แอดเดรสและความยาวของพื้นที่ของหน่วยความจำของไคลเอ็นต์ที่จะ อ่านหรือเขียนแบบรีโมตโดยเซิร์ฟเวอร์ RDMA

มีสองเมธอดสำหรับการส่งคำร้องขอ RDMA จากไคลเอ็นต์ RDMA ไปยังเซิร์ฟเวอร์ RDMA

เมธอดแรกคือการส่งข้อความควบคุม **RDS_CMSG_RDMA_MAP** (ส่งโครงสร้าง **rds_get_mr_args**) พร้อมกับ คำร้องขอ RDMA ระดับแอ็พพลิเคชันโดยใช้การเรียกระบบ **sendmsg( )** บนซ็อกเก็ต RDS เคอร์เนลระบบปฏิบัติการ AIX ที่ด้านไคลเอ็นต์จะประมวลผลข้อความควบคุม **RDS_CMSG_RDMA_MAP** โดยแม็พพื้นที่ที่ระบุของหน่วยความจำโลคัล (จากพื้นที่แอดเดรสของ ไคลเอ็นต์แอ็พพลิเคชัน) สำหรับการเข้าถึง DMA และสร้างคุกกี้ RDMA จากนั้น ส่งคำร้องขอระดับแอ็พพลิเคชันไปยังเซิร์ฟเวอร์พร้อมกับ คุกกี้ RDMA

เมธอดที่สองมีสองขั้นตอน ขั้นตอนแรกคือการเรียก การเรียกระบบ **setsockopt( )** ด้วยอ็อพชันซ็อกเก็ต **RDS_GET_MR** เพื่อส่งผ่านโครงสร้าง **rds_get_mr_args** การเรียกนี้ จะแม็พพื้นที่ที่ระบุของหน่วยความจำโลคัลสำหรับการเข้าถึง DMA และส่งคืนคุกกี้ RDMA ขั้นตอนที่สองคือการส่งข้อความควบคุม **RDS_CMSG_RDMA_DEST** (ส่งคุกกี้ RDMA ที่ได้รับมาจากขั้นตอนแรก) พร้อมกับคำร้องขอ RDMA ระดับแอ็พพลิเคชันโดยใช้การเรียกระบบ **sendmsg( )**

เมธอดแรกซึ่งต้องใช้หนึ่งการเรียกระบบ เหมาะสมกว่า เมธอดที่สองซึ่งต้องใช้สองการเรียกระบบ

เมื่อแอ็พพลิเคชันเซิร์ฟเวอร์ RDMA ได้รับคำร้องขอ **การอ่าน RDMA** ระดับแอ็พพลิเคชัน จากไคลเอ็นต์ แอ็พพลิเคชันเซิร์ฟเวอร์ยังได้รับข้อความควบคุม **RDS_CMSG_RDMA_DEST** (ส่งคุกกี้ RDMA จากไคลเอ็นต์) ด้วย จากนั้น เซิร์ฟเวอร์ จะเริ่มต้นการดำเนินการ **อ่าน RDMA** โดยส่งการตอบระดับแอ็พพลิเคชัน ไปยังไคลเอ็นต์ พร้อมกับข้อความควบคุม **RDS_CMSG_RDMA_ARGS** (ส่งโครงสร้าง **rds_rdma_args**) เคอร์เนลระบบปฏิบัติการ AIX ที่ ด้านเซิร์ฟเวอร์จะประมวลผลข้อความควบคุม **RDS_CMSG_RDMA_ARGS** โดยแม็พพื้นที่ที่ระบุของหน่วยความจำโลคัล (จากพื้นที่แอดเดรสของ เซิร์ฟเวอร์แอ็พพลิเคชัน) สำหรับการเข้าถึง DMA และเริ่มต้นการดำเนินการอ่าน RDMA ทางฟิสิคัล การดำเนินการอ่าน RDMA ทำโดย อ หลังจากการดำเนินการอ่าน RDMA เสร็จสมบูรณ์แล้ว อะแด็ปเตอร์ด้านเซิร์ฟเวอร์จะส่งการตอบระดับแอ็พพลิเคชันไปยังไคลเอ็นต์ การทำเช่นนี้ ทำให้ไคลเอ็นต์แอ็พพลิเคชันทราบว่าการดำเนินการอ่าน RDMA เสร็จสมบูรณ์แล้ว

หมายเหตุ: การดำเนินการ RDMA มีการร้องขอโดยไคลเอ็นต์โดยใช้ตัวควบคุม **RDS_CMSG_RDMA_MAP** ซึ่งมีการตั้งค่าแฟล็ก **RDS_RDMA_USE_ONCE** สำหรับคำร้องขอนี้ พื้นที่ของหน่วยความจำซึ่งแม็พสำหรับ DMA ในพื้นที่แอดเดรสของไคลเอ็นต์ของหน่วยความจำมีการยกเลิกการแม็พโดยอัตโนมัติสำหรับ DMA เมื่อ ไคลเอ็นต์ได้รับการตอบระดับแอ็พพลิเคชันจากเซิร์ฟเวอร์

แม้ว่ากลไกการแม็พหรือการยกเลิกการแม็พ DMA โดยนัยนี้ทำให้ เขียนแอ็พพลิเคชัน RDMA ได้ง่ายขึ้น แต่ผู้พัฒนาต้องระวังว่า การลงทะเบียนหน่วยความจำสำหรับ DMA บนระบบปฏิบัติการ AIX เป็นการดำเนินการที่เสียค่าใช้จ่ายสูง ดังนั้น ถ้าพื้นที่ของหน่วยความจำเดียวกัน กำลังจะมีการเข้าถึงโดยใช้ RDMA หลายครั้ง การลงทะเบียน DMA เฉพาะครั้งแรกจะมีประสิทธิภาพมากกว่า ในการทำกิจกรรมนี้ ไคลเอ็นต์แอ็พพลิเคชันต้องใช้ข้อความควบคุม **RDS_CMSG_RDMA_MAP** โดยไม่มีการตั้งค่าแฟล็ก **RDS_RDMA_USE_ONCE** เมื่อส่ง คำร้องขอ RDMA ไปยังเซิร์ฟเวอร์ จากนั้น การโอนย้าย RDMA ในลำดับต่อมาไปยัง พื้นที่เดียวกันของหน่วยความจำของไคลเอ็นต์สามารถเริ่มต้นได้โดยแอ็พพลิเคชันเซิร์ฟเวอร์ RDMA โดยไคลเอ็นต์ไม่ต้องส่งคำร้องขออื่น ไปยังเซิร์ฟเวอร์ ในตอนท้าย ไคลเอ็นต์แอ็พพลิเคชันจะต้อง ยกเลิกการแม็พหน่วยความจำ DMA ที่แม็พไว้อย่างชัดแจ้งโดยใช้การเรียกระบบ **setsockopt( )** พร้อมด้วยอ็อพชันซ็อกเก็ต **RDS_FREE_MR**

RDS-specific socket options are specified by using **SOL_RDS** as the level parameter for the **setsockopt( )** or **getsockopt( )** system call.

Transmission Control Protocol:

TCP จัดเตรียมการส่งมอบสตรีมของข้อมูลระหว่าง อินเตอร์เน็ตโฮสต์

เหมือนกับ **UDP** นั่นคือ **TCP** ใช้อินเทอร์เน็ตโปรโตคอล โปรโตคอลที่อยู่ต่ำกว่า เพื่อส่งผ่านเดทาแกรม และสนับสนุนการส่งผ่านบล็อค ของสตรีมที่ต่อเนื่องของเดทาแกรมระหว่างพอร์ตของกระบวนการ ไม่เหมือนกับ **UDP** นั่นคือ **TCP** จัดเตรียมการส่งมอบข้อความที่เชื่อถือได้ **TCP** ตรวจสอบให้แน่ใจว่าข้อมูลไม่ได้เสียหาย สูญหาย ทำซ้ำ หรือส่งมอบภายนอกลำดับของกระบวนการรับ ความแน่นอนของความเชื่อถือได้นี้เก็บโปรแกรมเมอร์แอ็พพลิเคชัน จากการสร้างการสื่อสารที่ป้องกันความปลอดภัยให้กับซอฟต์แวร์

ต่อไปนี้คือคุณสมบัติของการดำเนินการของ **TCP**:

ไอเท็ม	คำอธิบาย
Basic Data Transfer	**TCP** สามารถถ่ายโอนสตรีมต่อเนื่องของ octets ขนาด 8 บิตในแต่ละทิศทางระหว่างผู้ใช้โดยจัดทำแพ็กเกจจำนวนไบต์ ให้กับเซกเมนต์สำหรับการส่งข้อมูลผ่านระบบอินเตอร์เน็ต การนำ **TCP** ไปใช้อนุญาตให้ขนาดของเซกเมนต์มีค่าอย่างน้อย 1024 ไบต์โดยทั่วไป **TCP** ตัดสินใจว่า เมื่อบล็อกและส่งต่อแพ็กเก็ตในเวลาที่สะดวก
Reliability	**TCP** ต้องกู้คืนข้อมูลที่เสียหาย สูญหาย ทำซ้ำ หรือส่งมอบภายนอกลำดับโดยอินเตอร์เน็ต **TCP** บรรลุ ความเชื่อถือได้นี้โดยกำหนดหมายเลขลำดับให้กับแต่ละ octet ที่ส่งข้อมูลและต้องการการตอบรับทางบวก (ACK) จากการรับ **TCP** หาก ACK ไม่ได้รับภายในช่วงเวลาของการหมดเวลาข้อมูล จะถูกส่งผ่าน การส่งผ่าน **TCP** สำหรับค่าหมดเวลาอีกครั้ง ถูกกำหนดเป็นแบบไดนามิกสำหรับแต่ละการเชื่อมต่อ อ้างอิงตามเวลาแบบไปกลับ ที่ receiver หมายเลขลำดับถูกใช้เพื่อเรียงลำดับเซกเมนต์ ที่อาจรับภายนอกลำดับและกำจัดความซ้ำซ้อน ความเสียหายถูกจัดการโดยการเพิ่มเช็กซัมให้กับแต่ละเซกเมนต์ที่ส่งข้อมูล การตรวจสอบที่ receiver และการละทิ้งเซกเมนต์ที่เสียหาย
Flow Control	**TCP** กำหนดจำนวนข้อมูลที่ส่งโดยส่งคืนหน้าต่างด้วย ACK ทุกตัวเพื่อบ่งชี้ช่วงของหมายเลขลำดับที่ยอมรับได้ ซึ่งเข้าใกล้เซกเมนต์สุดท้ายที่รับได้เป็นผลสำเร็จ หน้าต่างบ่งชี้จำนวนของ octets ที่ได้รับอนุญาตซึ่งผู้ส่งอาจส่งก่อนที่จะรับสิทธิ เพิ่มเติม
Multiplexing	**TCP** อนุญาตให้กระบวนการจำนวนมากภายในโฮสต์เดียวกัน ใช้การสื่อสารแบบ **TCP** พร้อมเพียงกัน **TCP** ได้รับชุดของแอดเดรสของพอร์ตภายในแต่ละโฮสต์ **TCP** รวมหมายเลขพอร์ตด้วยเน็ตเวิร์กแอดเดรสและโฮสต์แอดเดรส เพื่อระบุแต่ละซ็อกเก็ตโดยเฉพาะ คู่ของซ็อกเก็ตระบุแต่ละการเชื่อมต่อ
การเชื่อมต่อ	**TCP** ต้องเริ่มต้นและรักษาข้อมูลสถานะ สำหรับแต่ละ data stream ชุดของข้อมูลนี้ ซึ่งประกอบด้วยซ็อกเก็ต หมายเลขลำดับ และขนาดหน้าต่างถูกเรียกว่าการเชื่อมต่อ แต่ละการเชื่อมต่อถูกระบุโดยคู่ของซ็อกเก็ตที่ระบุไว้ สองฝั่ง
Precedence and Security	ผู้ใช้ **TCP** อาจบ่งชี้ความปลอดภัยและการมาก่อนของ การสื่อสาร ค่าดีฟอลต์ถูกใช้เมื่อคุณลักษณะเหล่านี้ ไม่จำเป็น

ตัวเลข **TCP Packet Header** แสดงรูปภาพประกอบสำหรับคุณสมบัติเหล่านี้

รูปที่ 13. แพ็กเก็ตส่วนหัว Transmission Control Protocol (TCP)

รูปประกอบนี้แสดงสิ่งที่มีอยู่ในแพ็กเก็ตส่วนหัว **TCP** รายการเดี่ยวถูกแสดงอยู่ในข้อความด้านล่าง

นิยามฟิลด์ส่วนหัว TCP:

คำอธิบายอย่างสั้นของฟิลด์ **Transmission Control Protocol (TCP)** แต่ละฟิลด์ต่อไปนี้

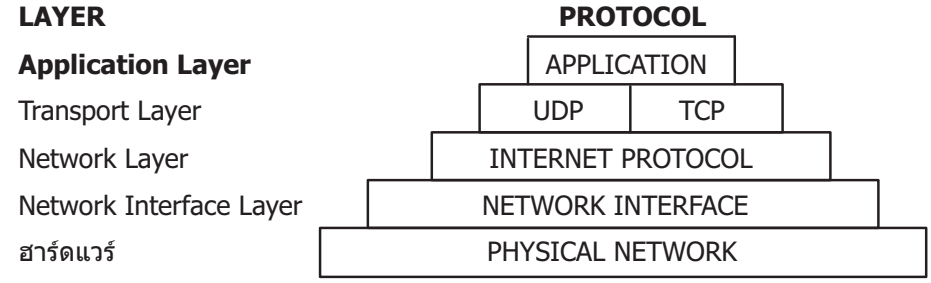
ไอเท็ม	คำอธิบาย
Source Port	ระบุจำนวนพอร์ตของแอ็พพลิเคชันโปรแกรมต้นทาง
Destination Port	ระบุจำนวนพอร์ตของแอ็พพลิเคชันโปรแกรมปลายทาง
Sequence Number	ระบุหมายเลขลำดับของไบต์ของข้อมูลแรกในเซกเมนต์นี้
Acknowledgment Number	ระบุตำแหน่งของไบต์ที่สูงที่สุดที่ได้รับ
Data Offset	ระบุออฟเซ็ตของส่วนของข้อมูลของเซกเมนต์
สงวนไว้	สงวนไว้สำหรับใช้ในอนาคต
โค้ด	ควบคุมบิตที่บ่งชี้วัตถุประสงค์ของเซกเมนต์: **URG** ฟิลด์จุดเร่งด่วนถูกต้อง **ACK** ฟิลด์การตอบรับถูกต้อง **PSH** คำร้องขอเซกเมนต์ PUSH. **RTS** รีเซ็ตการเชื่อมต่อ **SYN** ซิงโครไนซ์หมายเลขลำดับ **FIN** ผู้ส่งเข้าถึงส่วนท้ายของสตีมไบต์
Window	ระบุจำนวนข้อมูลปลายทางที่กำลังยอมรับ
Checksum	ตรวจสอบ integrity ของส่วนหัวของเซกเมนต์และข้อมูล
Urgent Pointer	บ่งชี้ข้อมูลที่ถูกส่งเร็วที่สุดเท่าที่จะเร็วได้ ตัวชี้นี้ ระบุตำแหน่งที่ข้อมูลเร่งด่วนสิ้นสุด

ไอเท็มอ็อพชัน	คำอธิบาย
ส่วนท้ายของรายการอ็อพชัน	ระบุส่วนท้ายของรายการอ็อพชัน ซึ่งถูกใช้ที่ส่วนท้ายของอ็อพชันสุดท้าย ซึ่งไม่ใช่ที่ส่วนท้ายของแต่ละอ็อพชัน อ็อพชันนี้ ควรถูกใช้เฉพาะหากส่วนท้ายของอ็อพชันจะไม่ได้เกิดขึ้นพร้อมกันกับส่วนท้ายของส่วนหัว **TCP**
ไม่มีการดำเนินการ	ระบุขอบเขตระหว่างอ็อพชัน สามารถใช้ระหว่างอ็อพชันอื่น ตัวอย่างเช่น การจัดตำแหน่งจุดเริ่มต้นของอ็อพชันตามลำดับบนขอบเขตของคำ ไม่มีการรับประกันที่ผู้ส่งจะใช้อ็อพชันนี้ ดังนั้น receiver ต้องถูกจัดเตรียมเพื่อประมวลผลอ็อพชันแม้ว่าไม่ได้เริ่มต้นบนขอบเขตของคำ
ขนาดเซกเมนต์สูงสุด	บ่งชี้ขนาดของเซกเมนต์สูงสุด **TCP** สามารถรับ ซึ่งถูกส่งใน คำร้องขอการเชื่อมต่อเริ่มต้น

applications programming interface to **TCP** ประกอบด้วยชุดของรูทีนย่อยไลบรารีที่จัดเตรียมโดยอินเตอร์เฟสของซ็อกเก็ต

Internet Application-Level Protocols

TCP/IP ใช้อินเตอร์เน็ตโปรโตคอลระดับสูง ที่ระดับของแอ็พพลิเคชันโปรแกรม



รูปที่ 14. เลเยอร์แอ็พพลิเคชันของ TCP/IP Suite of Protocols

รูปภาพประกอบนี้แสดงเลเยอร์ต่างๆ ของ **TCP/IP** Suite of Protocols จากด้านบนสุด เลเยอร์แอ็พพลิคเซันสอดคล้องกับ แอ็พพลิเคชัน เลเยอร์ transport มี **UDP** และ **TCP** เลเยอร์ network มีเน็ตเวิร์กอินเตอร์เฟส (ฮาร์ดแวร์) และท้ายสุด เลเยอร์ฮาร์ดแวร์มีฟิสิคัลเน็ตเวิร์ก

เมื่อแอ็พพลิเคชันต้องการส่งข้อมูลไปยังแอ็พพลิเคชันบนโฮสต์อื่น แอ็พพลิเคชันส่งข้อมูลลงไปยังระดับโปรโตคอล transport เพื่อจัดเตรียมข้อมูลสำหรับการส่งข้อมูล

โปรโตคอลระดับแอ็พพลิเคชันของอินเตอร์เน็ตประกอบด้วย:

- **Domain Name Protocol**
- **Exterior Gateway Protocol**
- **File Transfer Protocol**
- **Name/Finger Protocol**
- **Telnet Protocol**
- **Trivial File Transfer Protocol**

TCP/IP ใช้โปรโตคอลระดับสูงที่ไม่ใช่โปรโตคอลอินเตอร์เน็ตไม่ใช่เชิงพาณิชย์ ต่แใช้ใน Internet community ที่ระดับของโปรแกรมแอ็พพลิเคชัน โปรโตคอลเหล่านี้สอดแทรก:

- Distributed Computer Network (DCN) **Local-Network Protocol**
- **Remote Command Execution Protocol**
- **Remote Login Protocol**
- **Remote Shell Protocol**
- **Wake On LAN Protocol**
- **Routing Information Protocol**
- **Time Server Protocol**

TCP/IP ไม่ได้จัดเตรียม APIs ให้กับโปรโตคอลระดับ แอ็พพลิเคชันเหล่านี้

Domain Name Protocol:

Domain Name Protocol (**DOMAIN**) อนุญาตให้โฮสต์ในโดเมน ทำงานเป็น *เซิร์ฟเวอร์รายชื่อ* สำหรับโฮสต์อื่นๆ ภายในโดเมน

DOMAIN ใช้ **UDP** หรือ **TCP** เป็นโปรโตคอลที่จำเป็นต้องมี และอนุญาตให้โลคัลเน็ตเวิร์กเพื่อกำหนดชื่อโฮสต์ภายในโดเมนแบบอิสระ จากโดเมนอื่น ตามปกติแล้ว โปรโตคอล **DOMAIN** ใช้ **UDP** อย่างไรก็ตาม หากการตอบกลับของ **UDP** ถูกตัดปลาย **TCP** สามารถใช้งานได้โปรโตคอล **DOMAIN** ใน **TCP/IP** สนับสนุนทั้งสองแบบ

ในระบบการตั้งชื่อตามลำดับชั้นของ **DOMAIN** รูทีนตัวแก้ไขปัญหาโลคัล สามารถแก้ไขชื่ออินเตอร์เน็ตและแอดเดรสที่ใช้ฐานข้อมูลการแก้ปัญหาชื่อโลคัล ที่ดูแลรักษาโดย **named** daemon หากชื่อที่ร้องขอโดยโฮสต์ ไม่ได้อยู่ในฐานข้อมูลโลคัล รูทีนตัวแก้ไขปัญหาต้องการเคียวรีเซิร์ฟเวอร์รายชื่อ **DOMAIN** แบบรีโมต ในกรณีอื่น หากข้อมูลการแก้ปัญหาเรื่องชื่อไม่พร้อมใช้งาน รูทีนตัวแก้ไขปัญหาพยายามใช้ไฟล์ `/etc/hosts` สำหรับการแก้ไขชื่อ

หมายเหตุ: **TCP/IP** ตั้งค่ารูทีนตัวแก้ไขท้องถิ่นสำหรับโปรโตคอล **DOMAIN** หากโลคัลไฟล์ `/etc/resolv.conf` มีอยู่ หากไฟล์นี้ไม่มีอยู่ **TCP/IP** ตั้งค่ารูทีนตัวแก้ไขบนโลคัล เพื่อใช้ฐานข้อมูล `/etc/hosts`

TCP/IP นำโปรโตคอล **DOMAIN** ไปใช้ใน **named** daemon และในรูทีนตัวแก้ไขและไม่ได้จัดเตรียม API กับโปรโตคอลนี้

Exterior Gateway Protocol:

Exterior Gateway Protocol (**EGP**) คือกลไกที่ช่วยให้ เกตเวย์ภายนอกของ *ระบบอัตโนมัติ* แบ่งใช้ข้อมูลเส้นทาง กับเกตเวย์ภายนอกบนระบบอัตโนมัติอื่น

ระบบแบบอัตโนมัติ:

ระบบแบบอัตโนมัติคือกลุ่มของเน็ตเวิร์กและเกตเวย์ที่มีหนึ่งสิทธิ ในการดูแลซึ่งเป็นความรับผิดชอบ

เกตเวย์คือ *interior neighbors* หากตั้งอยู่บนระบบแบบอัตโนมัติ และ *exterior neighbors* หากตั้งอยู่บน ระบบอัตโนมัติที่แตกต่างกัน เกตเวย์ที่แลกเปลี่ยนข้อมูลการเราต์โดยใช้ **EGP** พูดถึง **EGP** *peers* หรือ *neighbors* เกตเวย์ระบบแบบอัตโนมัติ ใช้ **EGP** เพื่อจัดเตรียมข้อมูลการเข้าถึงใน **EGP** neighbors

EGP อนุญาตให้เกตเวย์เพื่อถามเกตเวย์ภายนอกอื่น เพื่อยอมรับการแลกเปลี่ยนข้อมูลการเข้าถึง ตรวจสอบว่า **EGP** neighbors กำลังตอบกลับ และช่วย **EGP** neighbors เพื่อแลกเปลี่ยนข้อมูลการเข้าถึง โดยส่งผ่านข้อความอัพเดตการเราต์

EGP จำกัดเกตเวย์ภายนอกโดยอนุญาตให้ประกาศ ปลายทางของเน็ตเวิร์กที่เข้าถึงภายในระบบแบบอัตโนมัติ ของเกตเวย์ ดังนั้น เกตเวย์ภายอกโดยใช้ **EGP** ส่งข้อมูลไปยัง **EGP** neighbor แต่ไม่ได้ประกาศข้อมูลการเข้าถึงเกี่ยวกับ **EGP** neighbors ภายนอกระบบแบบอัตโนมัติ

EGP ไม่ได้ตีความเมทริกระยะทางใดๆ ที่ปรากฏขึ้นใน ข้อความอัพเดตการเราต์จากโปรโตคอลอื่นๆ **EGP** ใช้ฟิลด์ระยะทาง เพื่อระบุพาธที่มีอยู่ (ค่า 255 หมายถึงเน็ตเวิร์ก ไม่สามารถเข้าถึงได้) ค่าไม่สามารถใช้เพื่อคำนวณเราต์ที่สั้นกว่าสองทาง ยกเว้นเราต์เหล่านั้นมีอยู่ในระบบอัตโนมัติเดี่ยว ดังนั้น **EGP** จึงไม่สามารถใช้เป็นอัลกอริทึมการเราต์ได้ ตามผลลัพธ์แล้ว มีเพียงหนึ่งพาธจากเกตเวย์ภายนอกไปยังเน็ตเวิร์กใดๆ

ในทางตรงกันข้าม **Routing Information Protocol** (**RIP**) ซึ่งสามารถใช้ภายในระบบแบบอัตโนมัติของอินเตอร์เน็ตเน็ตเวิร์กที่ตั้งค่าเราต์แบบไดนามิก เราต์ **EGP** ถูกกำหนดไว้ล่วงหน้าในไฟล์ **/etc/gated.conf EGP** สมมติว่า **IP** อยู่ภายใต้โปรโตคอล

ชนิดข้อความ EGP:

ชนิดข้อความ exterior gateway protocol (EGP) ที่หลากหลายถูกนิยาม ไว้ที่นี่

ไอเท็ม	คำอธิบาย
Neighbor Acquisition Request	ถูกใช้โดยเกตเวย์ภายนอกเพื่อรอขอ neighbor ของแต่ละคำร้องขอ
Neighbor Acquisition Reply	ถูกใช้โดยเกตเวย์ภายนอกเพื่อยอมรับคำร้องขอเป็น neighbor
Neighbor Acquisition Refusal	ถูกใช้โดยเกตเวย์ภายนอกเพื่อปฏิเสธคำร้องขอเป็น neighbors ข้อความปฏิเสธสอดแทรกเหตุผลสำหรับการปฏิเสธ เช่น `out of table space`
Neighbor Cease	ถูกใช้โดยเกตเวย์ภายนอกเพื่อสิ้นสุดความสัมพันธ์ neighbor ข้อความ ที่สิ้นสุดสอดคล้องเหตุผลสำหรับสิ้นสุด เช่น `going down`
Neighbor Cease Acknowledgment	ถูกใช้เกตเวย์ภายนอกเพื่อตอบรับคำร้องขอเพื่อสิ้นสุดความสัมพันธ์ neighbor
Neighbor Hello	ถูกใช้เกตเวย์ภายนอกเพื่อพิจารณาภาวะเชื่อมต่อ. เกตเวย์ออกใช้ข้อความ `Hello` และเกตเวย์อื่นๆ ออกใช้ข้อความ `I Heard You`
I Heard You	ถูกใช้เกตเวย์ภายนอกเพื่อตอบกลับข้อความ `Hello` ข้อความ `I Heard You` สอดแทรกการเข้าถึงของการตอบเกตเวย์ และหากเกตเวย์ไม่สามารถเข้าถึง เหตุผลสำหรับการขาดแคลนของการเข้าถึง เช่น `You are unreachable because of problems with my network interface`
NR Poll	ถูกใช้โดยเกตเวย์ภายนอกเพื่อเคียวรีเกตเวย์ neighbor เกี่ยวกับ ความสามารถในการเข้าถึงเกตเวย์อื่น
Network Reachability	ถูกใช้เกตเวย์ภายนอกเพื่อตอบข้อความ `NR Poll` สำหรับเกตเวย์แต่ละตัวในข้อความ ข้อความ `Network Reachability` มีข้อมูลบนแอดเดรสที่เกตเวย์สามารถเข้าถึงผ่าน neighbors
EGP Error	ถูกใช้เกตเวย์ภายนอกเพื่อตอบกลับข้อความ EGP ที่มีเช็กซัมที่ไม่ดี หรือมีฟิลด์มีค่าไม่ถูกต้อง

TCP/IP นำโปรโตคอล **EGP** ในเซิร์ฟเวอร์ **gated** คำสั่งและไม่ได้จัดเตรียม API ให้กับโปรโตคอลนี้

File Transfer Protocol:

File Transfer Protocol (**FTP**) อนุญาตให้โฮสต์ถ่ายโอนข้อมูลระหว่าง โฮสต์ที่ไม่เหมือนกัน พร้อมกับไฟล์ระหว่างสองโฮสต์ภายนอกทางอ้อม

FTP จัดเตรียมภารกิจที่แสดงไดเร็กทอรีแบบรีโมต การเปลี่ยนไดเร็กทอรีแบบรีโมตในปัจจุบัน การสร้างและการลบไดเร็กทอรีแบบรีโมต และการถ่ายโอนไฟล์จำนวนมากในคำร้องขอเดี่ยว **FTP** เก็บการส่งผ่านข้อมูล ที่ปลอดภัยโดยส่งผ่านผู้ใช้และรหัสผ่านไปยังโฮสต์ภายนอก แม้ว่า **FTP** ถูกออกแบบมาให้ใช้โดยแอ็พพลิเคชันเป็นหลัก และยังอนุญาตให้ใช้เซสชันแบบ user-oriented แบบโต้ตอบ

FTP ใช้การส่งมอบสตรีมที่ไว้ใจได้ (**TCP/IP**) เพื่อส่งไฟล์ และใช้การเชื่อมต่อแบบ Telnet เพื่อถ่ายโอนคำสั่งและตอบกลับ **FTP** ยังเข้าใจพื้นฐานรูปแบบไฟล์จำนวนมากซึ่งรวมถึง NETASCII, IMAGE และLocal 8

TCP/IP ใช้ **FTP** ในคำสั่งผู้ใช้ **ftp** และคำสั่งเซิร์ฟเวอร์ **ftpd** และไม่ได้จัดเตรียม applications programming interface (API) กับโปรโตคอลนี้

เมื่อสร้างผู้ใช้ ftp แบบไม่ระบุชื่อและไดเร็กทอรี โปรดมั่นใจว่า ไดเร็กทอรีหลักสำหรับผู้ใช้ ftp และแบบไม่ระบุชื่อ (ตัวอย่างเช่น `/u/ftp`) เป็นเจ้าของโดย root และไม่อนุญาตให้ใช้สิทธิในการเขียน (ตัวอย่างเช่น `dr-xr-xr-x`) สคริปต์ `/usr/samples/tcpip/anon.ftp` สามารถใช้เพื่อสร้างแอคเคาต์ผู้ใช้ไฟล์ และไดเร็กทอรีเหล่านี้

Telnet Protocol:

Telnet Protocol (**TELNET**) จัดเตรียมเมธอดมาตรฐาน สำหรับอุปกรณ์เทอร์มินัลและกระบวนการ terminal-oriented กับอินเตอร์เฟส

TELNET ถูกใช้โดยโปรแกรมเทอร์มินัลอีมูเลชันที่อนุญาตให้คุณ ล็อกอินเข้าสู่รีโมตโฮสต์ อย่างไรก็ตาม **TELNET** ยังสามารถใช้สำหรับการสื่อสารแบบ terminal-to-terminal และการสื่อสารแบบ interprocess **TELNET** ยังถูกใช้โดยโปรโตคอลอื่น (ตัวอย่างเช่น **FTP**) สำหรับการสร้าง ช่องสัญญาณการควบคุมโปรโตคอล

TCP/IP ใช้ **TELNET** ในคำสั่งผู้ใช้ **tn, telnet** หรือ **tn3270 telnetd** daemon ไม่ได้จัดเตรียม API ให้กับ **TELNET**

TCP/IP สนับสนุนอ็อพชัน **TELNET** ต่อไปนี้ซึ่งต่อรองระหว่าง ไคลเอ็นต์และเซิร์ฟเวอร์:

ไอเท็ม	คำอธิบาย
BINARY TRANSMISSION (Used in **tn3270** sessions)	ส่งอักขระเป็นข้อมูลแบบไบนารี
SUPPRESS GO_AHEAD (ระบบปฏิบัติการหยุดอ็อพชัน GO-AHEAD)	บ่งชี้ว่า เมื่อผลกระทบกับการเชื่อมต่อระหว่างผู้ส่งข้อมูล และผู้รับข้อมูล ผู้ส่งจำเป็นต้องส่งอ็อพชัน GO_AHEAD หากอ็อพชัน GO_AHEAD ไม่ได้ต้องการ กลุ่มในการเชื่อมต่อ จะหยุดทำงานในทั้งสองทิศทาง การดำเนินการนี้ต้องแทนที่ทั้งใน ทิศทางทั้งสองอย่างเป็นอิสระ
TIMING MARK (จดจำ แต่ไม่มีการตอบกลับทางลบ)	ตรวจสอบให้แน่ใจว่า ข้อมูลที่ส่งได้ประมวลผลแล้ว
EXTENDED OPTIONS LIST	ส่วนขยายรายการอ็อพชัน **TELNET** สำหรับอ็อพชันอื่น 256 อ็อพชัน หากไม่มีอ็อพชันนี้ อ็อพชัน **TELNET** อนุญาตให้ใช้เฉพาะ 256 อ็อพชันเท่านั้น
ECHO (คำสั่ง User-changeable)	ส่งอักขระข้อมูล echo ที่ได้รับแล้วกลับไปยังผู้ส่ง ดั้งเดิม
TERM TYPE	เปิดใช้งานเซิร์ฟเวอร์เพื่อกำหนดชนิดของเทอร์มินัลที่เชื่อมต่อกับโปรแกรม **TELNET** ผู้ใช้
SAK (Secure Attention Key)	สร้างสภาพแวดล้อมที่จำเป็นสำหรับการสื่อสารที่ปลอดภัยระหว่าง คุณและระบบ
NAWS (การต่อรองเกี่ยวกับขนาดของหน้าต่าง)	เปิดใช้งานไคลเอ็นต์และเซิร์ฟเวอร์เพื่อต่อรองแบบไดนามิกสำหรับขนาด หน้าต่าง ซึ่งถูกใช้โดยแอ็พพลิเคชันที่สนับสนุนการเปลี่ยนแปลงขนาดของหน้าต่าง

หมายเหตุ: **TELNET** ต้องได้รับอนุญาตให้ส่งข้อมูลที่มีอักขระแปดบิตเมื่อไม่ได้อยู่ในโหมด ไบนารีหากใช้โค้ดเพจ ISO 8859 Latin

Trivial File Transfer Protocol:

Trivial File Transfer Protocol (**TFTP**) สามารถอ่านและเขียนไฟล์จากและไปยังโฮสต์ภายนอก

เนื่องจาก **TFTP** ใช้ **User Datagram Protocol** ที่ไม่สามารถเชื่อถือได้กับไฟล์การส่งออก ซึ่งจะเร็วกว่า **FTP** ซึ่งเหมือนกับ **FTP** นั่นคือ **TFTP** สามารถถ่ายโอนไฟล์เป็นอักขระ NETASCII หรือข้อมูลไบนารีแบบ 8 บิต อย่างใดอย่างหนึ่ง และไม่เหมือนกับ **FTP** นั่นคือ **TFTP** ไม่สามารถใช้เพื่อแสดงหรือเปลี่ยนไดเร็กทอรีที่โฮสต์ภายนอกและไม่มีการข้อกำหนด สำหรับความปลอดภัยเช่น การป้องกันรหัสผ่าน และ ข้อมูลสามารถเขียนหรือเรียกคืนใน ไดเร็กทอรีพับลิกเท่านั้น

TCP/IP ใช้ **TFTP** ในคำสั่งผู้ใช้ **tftp** และ **utftp** และในคำสั่งเซิร์ฟเวอร์ **tftpd** คำสั่ง **utftp** เป็นรูปแบบของคำสั่ง **tftp** สำหรับใช้ในไพพ์ **TCP/IP** ไม่ได้จัดเตรียม API ไปยังโปรโตคอลนี้

สำหรับข้อมูลเพิ่มเติม โปรดดูคำอธิบายคำสั่ง **tftp** หรือ **utftp** และคำอธิบาย **tftpd** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 5*

Name/Finger Protocol:

Name/Finger Protocol (**FINGER**) คืออินเตอร์เน็ตโปรโตคอล ระดับของแอ็พพลิเคชันที่จัดเตรียมอินเตอร์เฟสระหว่างคำสั่ง **finger** และ **fingerd** daemon

fingerd daemon ส่งคืนข้อมูลเกี่ยวกับผู้ใช้ ที่ล็อกอินในปัจจุบันเข้าสู่รีโมตโฮสต์ที่ระบุไว้ หากคุณเรียกใช้คำสั่ง finger ซึ่งระบุผู้ใช้ที่โฮสต์เฉพาะ คุณจะขอรับข้อมูลเฉพาะ เกี่ยวกับผู้ใช้นั้น **FINGER** Protocol ต้องแสดงอยู่ที่รีโมตโฮสต์ และที่การร้องขอโฮสต์ **FINGER** ใช้ **Transmission Control Protocol** ("Transmission Control Protocol" ในหน้า 164) เป็นโปรโตคอลที่จำเป็น

หมายเหตุ: TCP/IP ไม่ได้จัดเตรียม API ไปยังโปรโตคอลนี้

สำหรับข้อมูลเพิ่มเติม โปรดดูคำอธิบายคำสั่ง **finger** และคำอธิบาย **fingerd** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2*

โปรโตคอล Distributed Computer Network Local-Network:

gated daemon จัดเตรียมเน็ตเวิร์กโปรโตคอล **Distributed Computer Network** (**DCN**) บนโลคัล

Local-Network Protocol (**HELLO**) คือเกตเวย์โปรโตคอลที่ออกแบบภายใน สำหรับใช้ภายในระบบแบบอัตโนมัติ (สำหรับข้อมูลเพิ่มเติม โปรดดู "ระบบแบบอัตโนมัติ" ในหน้า 168) **HELLO** รักษาสภาวะเชื่อมต่อ การเราต์ และข้อมูลการเก็บเวลา ซึ่งอนุญาตให้แต่ละเครื่อง ที่อยู่ในเน็ตเวิร์กพิจารณาพาธที่สั้นที่สุดไปยังปลายทางโดยอ้างอิงเวลาหน่วง จากนั้น อัพเดตข้อมูลการเราต์ไปยังปลายทางนั้น

สำหรับข้อมูลเพิ่มเติม โปรดดูคำอธิบาย **gated** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2*

Remote Command Execution Protocol:

คำสั่งผู้ใช้ **rexec** และ **rexecd** daemon จัดเตรียม remote command execution protocol ที่อนุญาตให้ผู้ใช้รันคำสั่ง บนรีโมตโฮสต์ที่ทำงานร่วมกันได้

สำหรับข้อมูลเพิ่มเติม โปรดดูคำอธิบายคำสั่ง **rexec** และคำอธิบาย **rexecd** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*

Remote Login Protocol:

คำสั่งผู้ใช้ **rlogin** และ **rlogind** daemon จัดเตรียม **remote login protocol** ซึ่งอนุญาตให้ผู้ใช้ล็อกอินเข้าสู่รีโมตโฮสต์ และใช้เทอร์มินัลหากผู้ใช้เหล่านั้นถูกเชื่อมต่อโดยตรงกับ รีโมตโฮสต์

สำหรับข้อมูลเพิ่มเติม โปรดดูคำอธิบายคำสั่ง **rlogin** และคำอธิบาย **rlogind** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*

Remote Shell Protocol:

คำสั่งผู้ใช้ **rsh** และ **rshd** daemon จัดเตรียม **remote command shell protocol** ซึ่งอนุญาตให้ผู้ใช้เปิดเชลล์ บนโฮสต์ภายนอกที่สามารถทำงานร่วมกันได้สำหรับการรันคำสั่ง

สำหรับข้อมูลเพิ่มเติม โปรดดูคำอธิบายคำสั่ง **rsh** และคำอธิบาย **rshd** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*

Wake On LAN Protocol:

Wake On LAN (WOL) อนุญาตให้คุณปลุกโฮสต์ตั้งแต่หนึ่งโฮสต์ขึ้นไป ที่เชื่อมต่อกับเน็ตเวิร์กในโหมดหยุดทำงานชั่วคราว โดยส่ง Magic Packet ไปยังแอดเดรสที่ระบุไว้บน subnet ที่ระบุไว้

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการใช้ **WOL** โปรดดูคำอธิบายคำสั่ง **wol** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 6*

โปรโตคอลข้อมูลการเราต์:

โปรโตคอลข้อมูลการเราต์ (RIP) และ **routed** และ **gated** daemons ที่นำไปใช้เพื่อเก็บการติดตามข้อมูลการเราต์แบบอ้างอิง hop ของเกตเวย์ และคงไว้ซึ่งรายการตารางการเราต์เคอร์เนล

สำหรับข้อมูลเพิ่มเติม โปรดดู **routed** และ **gated** daemons

Time Server Protocol:

timed daemon ถูกใช้เพื่อซิงโครไนซ์ หนึ่งโฮสต์ด้วยเวลาของโฮสต์อื่น

daemon นี้อ้างอิงตามแนวคิดแบบไคลเอ็นต์/เซิร์ฟเวอร์ สำหรับข้อมูลเพิ่มเติม โปรดดูคำสั่ง **timedc** และคำอธิบาย และคำอธิบาย **timed** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 5*

หมายเลขที่กำหนด

สำหรับความเข้ากันได้ที่มีสภาพแวดล้อมของเน็ตเวิร์กทั่วไป หมายเลขที่รู้จักกันดี ถูกมอบหมายไว้สำหรับเวอร์ชันอินเตอร์เน็ต เน็ตเวิร์ก พอร์ต โปรโตคอล และอ็อพชันโปรโตคอล นอกจากนี้ ชื่อที่รู้จักกันดียังถูกกำหนดให้กับเครื่อง เน็ตเวิร์ก ระบบปฏิบัติการ โปรโตคอล เซอร์วิส และเทอร์มินัล

TCP/IP คอมไพล์ด้วยหมายเลขที่กำหนดไว้และชื่อที่นิยามอยู่ใน RFC 1010 *หมายเลขที่กำหนดไว้*

อินเตอร์เน็ตโปรโตคอล (IP) นิยามฟิลด์ขนาด 4 บิตในส่วนหัว **IP** ที่ระบุเวอร์ชันของโปรโตคอล Internetwork ทั่วไปที่ใช้งานสำหรับ **IP** หมายเลขเวอร์ชันนี้ในฐานสิบคือ 4 สำหรับรายละเอียดเกี่ยวกับหมายเลขที่กำหนดไว้และชื่อที่ถูกใช้โดย **TCP/IP** โปรดดูไฟล์ `/etc/protocols` และ `/etc/services` ซึ่งสอดแทรกด้วย **TCP/IP** สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับหมายเลขที่กำหนดไว้ และชื่อ โปรดอ้างถึง RFC 1010 และไฟล์ `/etc/services`

การ์ดอะแด็ปเตอร์เครือข่ายพื้นที่โลคัล TCP/IP

การ์ดอะแด็ปเตอร์เครือข่ายเป็นฮาร์ดแวร์ที่ต่อพ่วงเข้ากับ สายเคเบิลเครือข่ายทางกายภาพ การ์ดนี้รับผิดชอบในการรับและการส่งข้อมูล ที่ระดับฟิสิคัล

การ์ดอะแด็ปเตอร์เครือข่ายมีการควบคุมโดยไดรเวอร์อุปกรณ์อะแด็ปเตอร์เครือข่าย

เครื่องต้องมีหนึ่งการ์ดอะแด็ปเตอร์เครือข่าย (หรือการเชื่อมต่อ) สำหรับแต่ละ เครือข่าย (ไม่ใช่ชนิดเครือข่าย) ซึ่งเครื่องเชื่อมต่อ ตัวอย่างเช่น ถ้าโฮสต์ต่อพ่วงเข้ากับเครือข่าย token-ring สองเครือข่าย โฮสต์ต้องมีสองการ์ดอะแด็ปเตอร์เครือข่าย

TCP/IP ใช้การ์ดอะแด็ปเตอร์เครือข่ายและการเชื่อมต่อดังต่อไปนี้:

- อีเทอร์เน็ตมาตรฐานเวอร์ชัน 2
- IEEE 802.3
- Token-ring
- อะแด็ปเตอร์อะซิงโครนัสและพอร์ตอนุกรมแบบ native
- Fiber Distributed Data Interface (FDDI)
- Serial Optical Channel Converter (ที่อธิบายใน *หลักการเขียนโปรแกรมการสนับสนุนส่วนขยายเคอร์นัลและอุปกรณ์*)
- Asynchronous Transfer Mode (ATM)
- Fibre Channel

เทคโนโลยีเครือข่ายอีเทอร์เน็ตและ 802.3 ใช้อะแด็ปเตอร์ชนิดเดียวกัน

แต่ละเครื่องมีสล็อตส่วนขยายจำนวนจำกัด ซึ่งคุณอาจจะใช้บางตัวหรือทั้งหมด สำหรับอะแด็ปเตอร์การสื่อสาร นอกจากนี้ แต่ละเครื่องยังสนับสนุนอะแด็ปเตอร์การสื่อสารของชนิดที่กำหนดในจำนวน จำกัด ภายในขีดจำกัดเหล่านี้ (ขีดจำกัดซอฟต์แวร์) คุณสามารถติดตั้งชุดของ อะแด็ปเตอร์ใดๆ ได้สูงสุดถึงจำนวนทั้งหมดของสล็อตส่วนขยายที่มีอยู่ในเครื่องของคุณ (ขีดจำกัดฮาร์ดแวร์)

อย่างไรก็ตาม สามารถกำหนดคอนฟิกได้เพียงหนึ่งอินเตอร์เฟส **Transmission Control Protocol/Internet Protocol (TCP/IP)** เท่านั้น โดยไม่คำนึงถึงจำนวนของ Serial Optical Channel Converters ที่สนับสนุนโดยระบบ ไดรเวอร์อุปกรณ์ Serial Optical ใช้ตัวเปลี่ยนแชนเนลทั้งสองตัว แม้ว่ามีการกำหนดคอนฟิกโลจิคัลอินเตอร์เฟส **TCP/IP** เพียงตัวเดียว เท่านั้น

การติดตั้งเน็ตเวิร์กอะแด็ปเตอร์

ใช้โพรซีเดอร์นี้เพื่อติดตั้งเน็ตเวิร์กอะแด็ปเตอร์

เมื่อต้องการติดตั้งเน็ตเวิร์กอะแด็ปเตอร์:

1. ปิดระบบคอมพิวเตอร์ ดูที่คำสั่ง **shutdown** สำหรับวิธีเกี่ยวกับการปิดระบบ
2. ปิดกำลังไฟของคอมพิวเตอร์
3. ถอดฝาครอบคอมพิวเตอร์ออก
4. หาสล็อตว่างและใส่เน็ตเวิร์กอะแด็ปเตอร์ ต้องแน่ใจว่าใส่อะแด็ปเตอร์ในสล็อตดีแล้ว
5. ปิดฝาครอบคอมพิวเตอร์
6. รีสตาร์ทคอมพิวเตอร์

การจัดการและตั้งค่าอะแด็ปเตอร์

เพื่อตั้งค่าและจัดการ token-ring หรือ Ethernet อะแด็ปเตอร์ใช้งานในตารางต่อไปนี้

ตารางที่ 59. งานการการตั้งค่าและจัดการอะแด็ปเตอร์

งาน	วิธีลัด SMIT	คำสั่งหรือไฟล์
ตั้งค่าอะแด็ปเตอร์	`smit chgtok` (token-ring) `smit chgenet` (Ethernet)	1. กำหนดชื่ออะแด็ปเตอร์:[1] `lsdev -C -c adapter -t tokenring -H` หรือ `lsdev -C -c adapter -t ethernet -H` 2. รีเซ็ตความเร็วของ ring (token-ring) หรือชนิดของตัวเชื่อมต่อ (Ethernet) ถ้าจำเป็น ตัวอย่างเช่น : `chdev -l tok0 -a ring_speed=16 -P` หรือ `chdev -l ent0 -a bnc_select=dix -P`
การกำหนดเน็ตเวิร์กอะแด็ปเตอร์ฮาร์ดแวร์แอดเดรส	`smit chgtok` (token-ring) `smit chgenet` (Ethernet)	`lscfg -l tok0 -v` (token-ring)[2] `lscfg -l ent0 -v` (Ethernet)[2]

การตั้งฮาร์ดแวร์แอดเดรสอื่น	`smit chgtok` (token-ring) `smit chgenet` (Ethernet)	1. กำหนดฮาร์ดแวร์แอดเดรสอื่น ตัวอย่างเช่น สำหรับ token-ring:[2,3]
		`chdev -l tok0 -a alt_addr=0X10005A4F1B7F`
		สำหรับ Ethernet:[2,3]
		`chdev -l ent0 -a alt_addr=0X10005A4F1B7F -p`
		2. เริ่มใช้แอดเดรสอื่น สำหรับ token-ring:[4]
		`chdev -l tok0 -a use_alt_addr=yes`
		สำหรับ Ethernet:[4]
		`chdev -l ent0 -a use_alt_addr=yes`

หมายเหตุ:

1. ชื่อของเน็ตเวิร์กอะแด็ปเตอร์สามารถเปลี่ยนได้ถ้าคุณย้ายมันจากสล็อตหนึ่งไปยังสล็อตอื่น หรือลบมันจากระบบ ถ้าคุณเคยย้ายอะแด็ปเตอร์ใช้คำสั่ง **diag -a** เพื่ออัพเดตฐานข้อมูลคอนฟิกูเรชัน
2. แทนที่ชื่อของอะแด็ปเตอร์ของคุณด้วย `tok0` และ `ent0`
3. แทนที่ฮาร์ดแวร์แอดเดรสของคุณด้วย `0X10005A4F1B7F`
4. หลังจากทำโพรซีเดอร์นี้ คุณอาจพบการการการสื่อสารกับโฮสต์อื่นขาดไปจนกว่ามันจะ flush แคชของ Address Resolution Protocol (ARP) ของมัน และรับฮาร์ดแวร์แอดเดรสใหม่ของโฮสต์นี้

Virtual Local Area Networks

VLANs (Virtual Local Area Networks) สามารถคิดได้ว่าเป็นโดเมนการบรอดคาสต์แบบโลจิคัล VLAN จะแยกกลุ่มของผู้ใช้เน็ตเวิร์กบนเน็ตเวิร์กแบบฟิสิคัลจริงเป็นเซกเมนต์ของเน็ตเวิร์กแบบโลจิคัล

การใช้งานนี้สนับสนุนมาตรฐานการแท็ก IEEE 802.1Q VLAN ด้วยความสามารถที่จะสนับสนุนหลาย VLAN ID ที่รันบน Ethernet อะแด็ปเตอร์ แต่ละ VLAN ID จะถูกเชื่อมโยงกับ Ethernet อินเตอร์เฟสที่แยกกันกับเลเยอร์ข้างบน (IP เป็นต้น) และสร้างอินสแตนซ์ของ Ethernet อะแด็ปเตอร์แบบโลจิคัลที่เป็นหนึ่งเดียวต่อ VLAN ตัวอย่างเช่น `ent1`, `ent2` เป็นต้น

การสนับสนุน IEEE 802.1Q VLAN สามารถถูกตั้งค่าบน Ethernet อะแด็ปเตอร์ที่ได้รับการสนับสนุนใดๆ อะแด็ปเตอร์ต้องเชื่อมต่อกับสวิตช์ที่สนับสนุน IEEE 802.1Q VLAN

คุณสามารถตั้งค่าหลายอุปกรณ์ VLAN แบบโลจิคัลบนระบบเดียว แต่ละอุปกรณ์ VLAN แบบโลจิคัลจะกันเป็นอินสแตนซ์ของ Ethernet อะแด็ปเตอร์เพิ่มเติม อุปกรณ์แบบโลจิคัลเหล่านี้สามารถถูกใช้เพื่อตั้งค่า Ethernet IP อินเตอร์เฟสเดียวกัน กับที่ใช้กับ Ethernet อะแด็ปเตอร์แบบฟิสิคัล ดังนั้น อ็อพชัน **no** *ifsize* (ดีฟอลต์ 8) ต้องถูกเพิ่มค่าเพื่อที่จะได้ไม่รวมเฉพาะ Ethernet กำหนดคอนฟิกสำหรับแต่ละอะแด็ปเตอร์ แต่ยังรวมอุปกรณ์ VLAN แบบโลจิคัลใดๆที่ถูกตั้งค่า ดูที่เอกสารสำหรับคำสั่ง **no**

แต่ละ VLAN สามารถมีค่า maximum transmission unit (MTU) ที่ต่างกัน แม้ว่าจะแบ่งใช้อะแด็ปเตอร์ Ethernet แบบฟิสิคัลเดียวกัน

การสนับสนุน VLAN ถูกจัดการผ่าน SMIT พิมพ์ `smit vlan` fast path จากบรรทัดรับคำสั่งและเลือกทางเลือกของคุณ จากเมนูหลักของ VLAN ความช่วยเหลือแบบออนไลน์พร้อมใช้งาน

หลังจากคุณกำหนดค่า VLAN, กำหนดค่าอินเตอร์เฟส IP ตัวอย่างเช่น `en1` สำหรับ Ethernet มาตรฐานหรือ `et1` สำหรับ IEEE 802.3 โดยใช้ SMIT หรือคำสั่ง

AIX 5.3 และหลังจากนั้น สนับสนุน Ethernet เสมือนโดยใช้สวิตช์ I/O เสมือนเป็นวิธีที่จะทำการสื่อสาร แบบในหน่วยความจำระหว่างพาร์ติชันในระบบPOWER5 สวิตช์ ยังสนับสนุนการแท็ก IEEE 802.1Q ซึ่งยอมให้อะแด็ปเตอร์ Ethernet เสมือนสามารถเป็นของ VLANs ที่ต่างกันบนสวิตช์ อะแด็ปเตอร์ Ethernet เสมือนจะถูกสร้าง และตั้งค่าบนพาร์ติชันโดยใช้ Hardware Management Console (HMC) หลังจากมันถูกสร้าง พาร์ติชันจะเห็นอะแด็ปเตอร์ Ethernet เสมือนในทรีของเฟิร์มแวร์แบบเปิดเมื่อมันสแกนหาอุปกรณ์ หลังจากมันถูกตรวจพบอะแด็ปเตอร์ Ethernet เสมือนจะถูกตั้งค่า และถูกใช้เหมือนกับ Ethernet อะแด็ปเตอร์แบบฟิสิคัล สำหรับข้อมูลเพิ่มเติม โปรดดูที่เอกสารฮาร์ดแวร์สำหรับระบบ POWER5 ของคุณ

หมายเหตุ:

1. ถ้าคุณพยายามที่จะกำหนดคอนฟิกค่า VLAN ID ที่ถูกใช้สำหรับอะแด็ปเตอร์ที่ระบุ คอนฟิกูเรชันจะล้มเหลวโดยมีข้อผิดพลาดต่อไปนี้:

   ```
   Method error (/usr/lib/methods/chgvlan):
     0514-018 The values specified for the following attributes
              are not valid:
              vlan_tag_id    VLAN Tag ID
   ```

2. ถ้าผู้ใช้ (ตัวอย่างเช่น IP อินเตอร์เฟส) กำลังใช้อุปกรณ์ VLAN แบบโลจิคัล ความพยายามลบอุปกรณ์ VLAN แบบโลจิคัลใดๆ จะล้มเหลว ข้อความคล้ายดังต่อไปนี้จะถูกแสดง :

   ```
   Method error (/usr/lib/methods/ucfgcommo):
           0514-062 Cannot perform the requested function because the
                    specified device is busy.
   ```

   เพื่อลบอุปกรณ์ VLAN แบบโลจิคัล ขั้นแรกให้ถอดผู้ใช้ก่อน ตัวอย่างเช่น ถ้าผู้ใช้เป็น IP อินเตอร์เฟส `en1` ดังนั้นคุณสามารถใช้คำสั่งต่อไปนี้ :

   ```
   ifconfig en1 detach
   ```

   จากนั้นลบเน็ตเวิร์กอินเตอร์เฟสโดยใช้เมนู SMIT TCP/IP

3. ถ้าผู้ใช้ (ตัวอย่างเช่น IP อินเตอร์เฟส) กำลังใช้อุปกรณ์ VLAN แบบโลจิคัล ความพยายามแก้ไขคุณลักษณะของ VLAN (ID ของแท็กของ VLAN หรืออะแด็ปเตอร์ฐาน) จะล้มเหลว ข้อความคล้ายดังต่อไปนี้จะถูกแสดง :

```
Method error (/usr/lib/methods/chgvlan):
        0514-062 Cannot perform the requested function because the
                 specified device is busy.
```

เพื่อแก้ไขอุปกรณ์ VLAN แบบโลจิคัล ขั้นแรกให้ถอดผู้ใช้ก่อน ตัวอย่างเช่น ถ้าผู้ใช้เป็น IP อินเตอร์เฟส en1 ดังนั้นคุณสามารถใช้คำสั่งต่อไปนี้ :

```
ifconfig en1 detach
```

จากนั้นแก้ไข VLAN และเพิ่มเน็ตเวิร์กอินเตอร์เฟสอีกครั้งโดยใช้เมนู SMIT TCP/IP

การแก้ปัญหา VLAN:

tcpdump และ **trace** สามารถถูกใช้เพื่อแก้ปัญหา VLAN

การติดตามจะเชื่อม ID สำหรับแต่ละชนิดของแพ็กเก็ตที่ส่งต่อไปนี้ :

ไอเท็ม	คำอธิบาย
แพ็กเก็ตที่ส่ง	3FD
แพ็กเก็ตที่ได้รับ	3FE
เหตุการณ์อื่น	3FF

คำสั่ง**entstat** ให้การรวมสถิติของอะแด็ปเตอร์แบบฟิสิคัลที่ VLAN ถูกตั้งค่า มันจะ *ไม่* ให้สถิติเฉพาะสำหรับอุปกรณ์ VLAN แบบโลจิคัลนั้นๆ

ข้อขำกัดของ VLAN:

การดัมพ์แบบรีโมตไม่ได้รับการสนับสนุนบน VLAN นอกจากนี้ อุปกรณ์ VLAN แบบโลจิคัลไม่สามารถใช้เพื่อสร้าง Etherchannel ของระบบ Cisco

อะแด็ปเตอร์ ATM

Asynchronous Transfer Mode (ATM) เป็นมาตรฐานสากลที่กำหนดวิธีของเน็ตเวิร์กความเร็วสูงเพื่อส่ง เสียง วิดีโอ และข้อมูลคอมพิวเตอร์แบบดั้งเดิมไปยังเน็ตเวิร์กแบบโลคัล แบบเมือง และเน็ตเวิร์กแบบกว้าง (LANs, MANs, และ WANs)

ATM อะแด็ปเตอร์จัดเตรียมการเชื่อมต่อแบบ full-duplex สำหรับRS/6000® เซิร์ฟเวอร์ หรือไคลเอ็นโดยใช้วงจรเสมือนแบบถาวร (PVCs) และวงจรเสมือนแบบเซอร์กิต (SVCs) การใช้งาน PVC และ SVC ถูกออกแบบเพื่อให้สอดคล้องกับข้อกำหนดของ ATM Forum จำนวนสูงสุดของวงจรเสมือนได้รับการสนับสนุนโดยขึ้นอยู่กับอะแด็ปเตอร์ อะแด็ปเตอร์ส่วนใหญ่จะสนับสนุนอย่างน้อย 1024 วงจรเสมือน

เทคโนโลยี ATM:

Asynchronous Transfer Mode (ATM) เป็น cell-switching ที่ใช้เทคโนโลยีแบบ connection-oriented

ใน ATM เน็ตเวิร์ก สเตชันปลายทางเชื่อมกับเน็ตเวิร์กโดยใช้การเชื่อมต่อแบบ full duplex เฉพาะ ATM เน็ตเวิร์กถูกสร้างจากการใช้สวิตช์ และสวิตช์เชื่อมต่อกันโดยใช้การเชื่อมต่อแบบฟิสิคัลเฉพาะ ก่อนที่การถ่ายโอนข้อมูลจะเริ่มขึ้น จะต้องมีการสร้างการเชื่อมต่อจากปลายทางข้างหนึ่งถึงอีกข้างหนึ่ง หลายการเชื่อมต่อสามารถทำและมีอยู่บนอินเตอร์เฟสแบบฟิสิคัลเดียว การส่งข้อมูลที่ส่งของสเตชันโดยการแบ่งเซกเมนต์ของ Protocol Data Units (PDUs) เป็นเซลล์ขนาด 53-ไบต์ การบรรทุกจะยังอยู่ในรูปของเซลล์ระหว่างการส่งในเน็ตเวิร์ก สเตชันที่รับจะประกอบเซลล์เป็น PDUs ใหม่ การเชื่อมต่อถูกระบุโดยใช้ตัวระบุพาธเสมือน (VPI) ตัวระบุแชนแนลเสมือน (VCI) ฟิลด์ VPI จะใช้หนึ่งไบต์ในเซลล์ส่วนหัวจำนวน 5 ไบต์ของ ATM โดยที่ VCI

ใช้สองไบต์ในเซลล์ส่วนหัวจำนวน 5 ไบต์ของ ATM โดยพื้นฐาน VPI:คู่ของ VCI จะระบุซอร์สของ ATM เซลล์ ฟังก์ชันของ ATM สวิตช์จะรู้จักต้อนกำเนิดของเซลล์ กำหนดฮ็อบต่อไป และส่งเซลล์ไปยังพอร์ต VPI:VCI จะเปลี่ยนบนฐานฐานของฮ็อบต่อฮ็อบ ดังนั้น VPI:ค่าของ VCI จะไม่เป็นสากล แต่ละวงจรเสมือนถูกอธิบายเป็นการรวมของ VPI: ค่าของ VCI ทั้งเน็ตเวิร์ก

การเชื่อมต่อ ATM:

สถาปัตยกรรม ATM มีวงจรเสมือนอยู่ 2 ชนิด : แบบถาวร (PVCs) และแบบสวิตช์ (SVCs)

ไอเท็ม	คำอธิบาย
วงจรเสมือนแบบถาวร	PVCs ถูกกำหนดคอนฟิกแบบ static และแบบแมนวล สวิตช์จะ จัดรูปแบบของ ATM เน็ตเวิร์กต้องถูกตั้งค่าให้รู้จัก VPI:VCI เป็นการรวมกันของแต่ละ endpoint และเราต์ endpoint ATM เซลล์ไปยัง endpoint ปลายทางผ่าน ATM เน็ตเวิร์ก เมื่อเชื่อมต่อผ่าน เน็ตเวิร์กถูกสร้างจาก endpoint หนึ่งไปนัง endpoint อื่น ATM เซลล์สามารถถูกส่งผ่าน ATM และ ATM สวิตช์ เน็ตเวิร์กสวิตช์จะแปล VPI: ค่า VCI ในวิธีที่เหมาะสม ที่จะเราต์เซลล์ไปยังปลายทางของมัน
วงจรเสมือนแบบสวิตช์	SVCs ถูกตั้งค่าแบบไดนามิกบนพื้นฐานของความต้องการ ATM ปลายทางจะถูกกำหนดแอดเดรสแบบ 20 ไบต์ SVCs ใช้ระนาบการควบคุม และระนาบของข้อมูล ระนาบการควบคุมใช้แชนแนลการให้สัญญาณ VPI:VCI 0:5 SVCs จะเกี่ยวข้องกับการจัดตั้งการเรียกใช้แบบ on demand โดยที่ ATM สเตชัน ส่งอิลิเมนต์ของข้อมูลที่ระบุ ATM ปลายทาง (และ ATM แอดเดรสต้นทาง ที่เป็นอ็อพชัน) โดยทั่วไป สเตชันที่เป็นผู้เรียก เน็ตเวิร์ก และสเตชันที่ถูกเรียกจะมีส่วนร่วมในการสื่อสาร สุดท้าย การเรียกจะได้การยอมรับหรือถูกปฏิเสธ ถ้าการเรียกถูกยอมรับ เน็ตเวิร์กจะกำหนด VPI:ค่า VCI สำหรับระนาบของข้อมูลให้กับสเตชันที่เรียกและสเตชันที่ถูกเรียก ในระนาบการควบคุม ATM เน็ตเวิร์กจะเราต์ (หรือสวิตช์) แพ็กเกจการส่งสัญญาณ บนพื้นฐานของ ATM แอดเดรส เมื่อแพ็กเก็ตเหล่านี้ถูกเราต์ สวิตช์จะตั้งตาราง สำหรับการเราต์เซลล์ของระนาบของข้อมูล ในระนาบของข้อมูล ATM เน็ตเวิร์กจะสวิตช์เซลล์บนพื้นฐานของ VPI:VCI เหมือนกับกรณีของ PVCs เมื่อการถ่ายโอนข้อมูลเสร็จแล้ว การเชื่อมต่อจะถูกยกเลิก

ATM แอดเดรสจะถูกสร้างโดยการลงทะเบียนกับ ATM เน็ตเวิร์ก และโดยได้มาจาก 13 ไบต์ซ้ายสุด 6 ไบต์ถัดมา ประกอบด้วย MAC แอดเดรสที่เป็นหนึ่งเดียวที่ถูกกำหนดให้กับผู้ผลิตอะแด็ปเตอร์ ไบต์ขวาสุด เป็นตัวเลือก การใช้ไบต์นี้ถูกใช้โดยดุลยพินิจของสเตชันปลายทาง ATM เน็ตเวิร์กจะไม่แปล ไบต์นี้

TCP/IP บน ATM:

มาตรฐาน *Internet Engineering Task Force RFC1577: Classical IP and ARP over ATM* จะระบุกลไกสำหรับการใช้ Internet Protocol (IP) บน ATM เนื่องจาก ATM เป็นเทคโนโลยีแบบ connection-oriented และ IP เป็นเทคโนโยยีแบบ datagram-oriented การแม็พ IP บน ATM จึงไม่ง่าย

โดยทั่วไป ATM เน็ตเวิร์กจะถูกแบ่งออกเป็น IP ซับเน็ตเวิร์กแบบโลจิคัล (LISs) แต่ละ LIS ประกอบด้วย ATM สเตชันจำนวนหนึ่ง LISs จะคล้ายกับการแบ่งเซกเมนต์ของ LAN ทั่วไป LISs เชื่อมต่อกันโดยเราเตอร์ อะแด็ปเตอร์นั้นๆ (บน ATM สเตชัน) สามารถเป็นส่วนของหลาย LISs คุณลักษณะนี้จะมีประโยชน์สำหรับการใช้เราเตอร์

RFC1577 จะระบุ RFC1483 ซึ่งจะระบุการ encapsulate แบบ Logical link control/Sub-Network Access Protocol (LLC/SNAP) โดยดีฟอลต์ PVC เน็ตเวิร์กสำหรับแต่ละ IP สเตชัน PVCs ทั้งหมดต้องถูกกำหนดแบบแมนวลโดยการตั้งค่า VPI: ค่า VPI ถ้าการตั้งค่า encapsulate แบบ LLC/SNAP ไม่ถูกใช้ IP แอดเดรสปลายทางจะเชื่อมโยงกับแต่ละ VPI:VCIต้องถูกกำหนด การการ encapsulate แบบ LLC/SNAP ถูกใช้ IP สเตชันสามารถเรียนรู้รีโมต IP แอดเดรสโดยกลไกของ InARP

สำหรับ SVC เน็ตเวิร์ก RFC1577 จะระบุหนึ่ง ARP ต่อ LIS วัตถุประสงค์ของ ARP เซิร์ฟเวอร์คือเพื่อ resolve IP แอดเดรสเป็น ATM แอดเดรสโดยไม่ใช้การบรอร์ดคาส แต่ละ IP สเตชันถูกตั้งค่าด้วย ATM แอดเดรสของ ARP เซิร์ฟเวอร์ IP สเตชันตั้ง

SVCs กับ ARP เซิร์ฟเวอร์ ซึ่งคือ จะส่งคำร้องขอ InARP ไปยัง IP สเตชัน ขึ้นอยู่กับการตอบของ InARP ARP เซิร์ฟเวอร์จะตั้งการแม็ต IP กับ ATM แอดเดรส IP สเตชันจะส่ง ARP แพ็กเก็ตไปยัง ARP เซิร์ฟเวอร์เพื่อ resolve แอดเดรส ซึ่งคือ ATM แอดเดรส จากนั้น IP สเตชันจะตั้ง SVC ไปยังสเตชันปลายทางและเริ่มการถ่ายโอนข้อมูล อายุของ entry ของ ARP ใน IP สเตชันและ ARP เซิร์ฟเวอร์จะขึ้นอยู่กับกลไกที่ถูกกำหนดไว้อย่างดี สำหรับทั้งสภาวะแวดล้อม PVC และ SVC แต่ละ IP สเตชันจะมีอย่างน้อยหนึ่งวงจรเสมือนต่อแอดเดรสปลายทาง

Internet Engineering Task Force RFC2225 เพิ่มการสนับสนุนของลิสต์ของ ATM ARP Request Address กับ RFC1577 ลิสต์ของ ATM ARP Request Address เป็นลิสต์ที่ประกอบด้วยหนึ่งหรือมากกว่า ATM แอดเดรสของแต่ละ ATM ARP เซิร์ฟเวอร์ที่อยู่ใน LIS RFC2225 ไคลเอ็นต์จำกัดจุดของความล้มเหลวที่เชื่อมโยงกับ ATM ARP เซอร์วิสของ 1577 ไคลเอ็นต์ 2225 ไคลเอ็นต์มีความสามารถที่จะสลับไปยัง ARP เซิร์ฟเวอร์สำรอง เมื่อ ATM ARP เซิร์ฟเวอร์ปัจจุบันล้มเหลว

RS/6000 จะตั้ง entry ลำดับแรกในลิสต์ของ ATM ARP Request Address เป็น ATM ARP เซิร์ฟเวอร์ลำดับแรกและ entry ที่เหลือเป็น ATM ARP เซิร์ฟเวอร์ลำดับที่สอง

ไคลเอ็นต์จะพยายามใช้ ATM ARP เซิร์ฟเวอร์ลำดับแรกเสมอ ถ้าความพยายามในการเชื่อมต่อกับ ATM ARP เซิร์ฟเวอร์ลำดับแรกล้มเหลว ไคลเอ็นต์จะพยายามเชื่อมต่อกับเซิร์ฟเวอร์ลำดับที่สองตัวแรก (ตำแหน่งในลิสต์ของ ATM ARP Request Address จะกำหนดลำดับของ ATM ARP เซิร์ฟเวอร์ลำดับที่สอง) ถ้าความพยายามในการเชื่อมต่อกับ ATM ARP เซิร์ฟเวอร์ลำดับที่สองตัวแรกล้มเหลว ไคลเอ็นต์จะพยายามเชื่อมต่อกับเซิร์ฟเวอร์ลำดับที่สองตัวถัดมา กระบวนการนี้จะถูกทำจนการเชื่อมต่อจะสำเร็จ

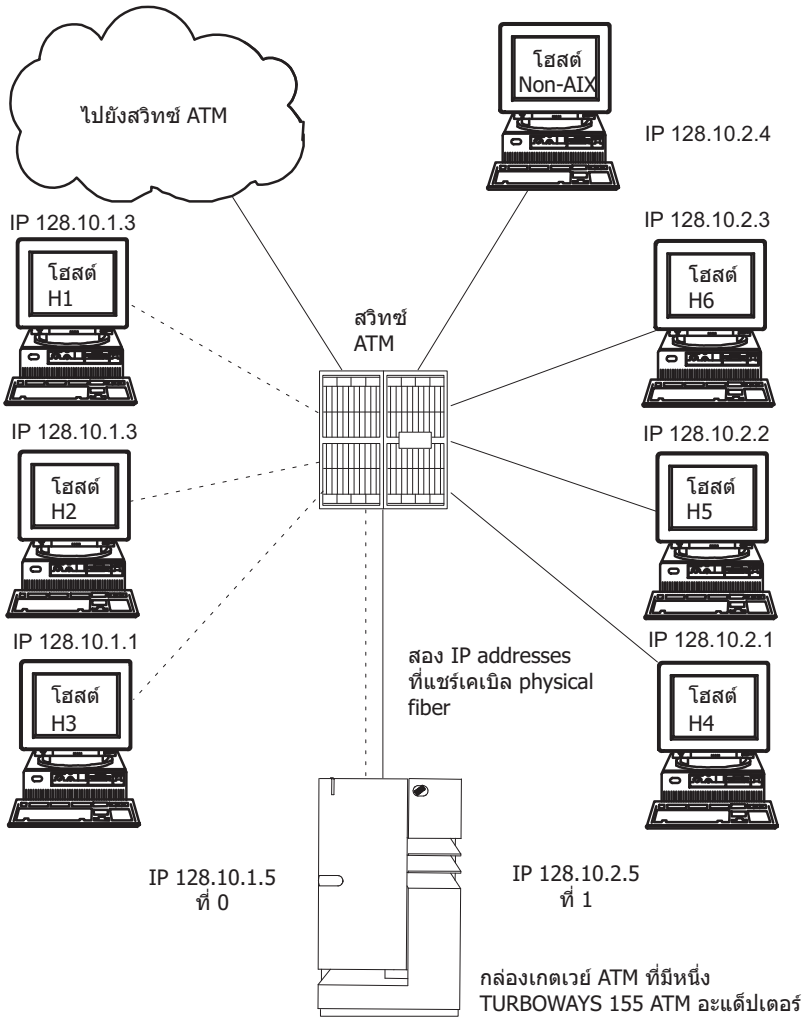
ถ้าการเชื่อมต่อกับ ATM ARP เซิร์ฟเวอร์ลำดับแรกล้มเหลว โดยไม่สนใจว่า ATM ARP เซิร์ฟเวอร์ลำดับที่สองที่มันเชื่อมต่ออยู่ หรือพยายามเชื่อมต่อด้วย ไคลเอ็นต์จะพยายามเชื่อมต่อกับ ATM ARP เซิร์ฟเวอร์ลำดับแรกทุกๆ 15 นาที ถ้าในที่สุดมันสามารถเชื่อมต่อกับ ATM ARP เซิร์ฟเวอร์ลำดับแรก ดังนั้นการเชื่อมต่อกับ ATM ARP เซิร์ฟเวอร์ลำดับที่สองในปัจจุบันจะถูกดร็อบ

ลิสต์ของ ATM ARP Request Address จะถูกใส่เข้าไปโดยแมนวลโดยผ่าน SMIT หรือโดยใช้คำสั่ง **ifconfig** ลิสต์ของ ATM ARP Request Address ไม่สามารถถูกตั้งค่ากับ Management Information Base (MIB)

PVC เน็ตเวิร์ก:

ใช้ตัวแทนของ ATM เน็ตเวิร์กต่อไปนี้เป็นตัวอย่างสำหรับตั้งค่าเน็ตเวิร์กของคุณ

ในรูป รูปที่ 15 ในหน้า 179 IP ซับเน็ต แบบโลจิคัลหนึ่งถูกแทนโดยเส้นประจากแต่ละโฮสต์ไปยังสวิตช์ IP ซับเน็ต อื่นถูกแทนด้วยเส้นทึบจากแต่ละโฮสต์ไปยังสวิตช์



รูปที่ 15. ตัวแทนของ ATM เน็ตเวิร์ก

ภาพนี้แสดงโครงร่างของ ATM เน็ตเวิร์กในโทโปโลยีแบบดาวทั่วไป ที่ศูนย์กลางของดาวเป็น ATM สวิตช์ จำนวนของโฮสต์ของ IP จะเป็นสาขาออกมาจากสวิตช์ จะลิงก์กับ ATM สวิตช์อื่น และหนึ่งของ ATM เกตเวย์และอะแด็ปเตอร์

ตาราง Representative Host Configuration ต่อไปนี้ระบุวิธีที่โฮสต์ H3 และ H4 ถูกตั้งค่าเพื่อสื่อสารกับเกตเวย์และกับแต่ละโฮสต์บน IP ซับเน็ต แบบโลจิคัลของมัน

ตารางที่ 60. การแทนการตั้งค่าโฮสต์. การแทนการตั้งค่าโฮสต์

เน็ตเวิร์กอินเตอร์เฟสไดรเวอร์	VPI:VCI	หมายเหตุ
at0	0:40	เชื่อมต่อกับ 128.10.1.5 (เกตเวย์)
at0	0:42	เชื่อมต่อกับ 128.10.1.2
at0	0:43	เชื่อมต่อกับ 128.10.1.3
at0	0:50	เชื่อมต่อกับ 128.10.2.5 (เกตเวย์)
at0	0:52	เชื่อมต่อกับ 128.10.2.2
at0	0:53	เชื่อมต่อกับ 128.10.2.3

ตารางที่ 60. การแทนการตั้งค่าโฮสต์ (ต่อ). การแทนการตั้งค่าโฮสต์

เน็ตเวิร์กอินเตอร์เฟสไดรเวอร์	VPI:VCI	หมายเหตุ
at0	0:54	เชื่อมต่อกับ 128.10.2.4

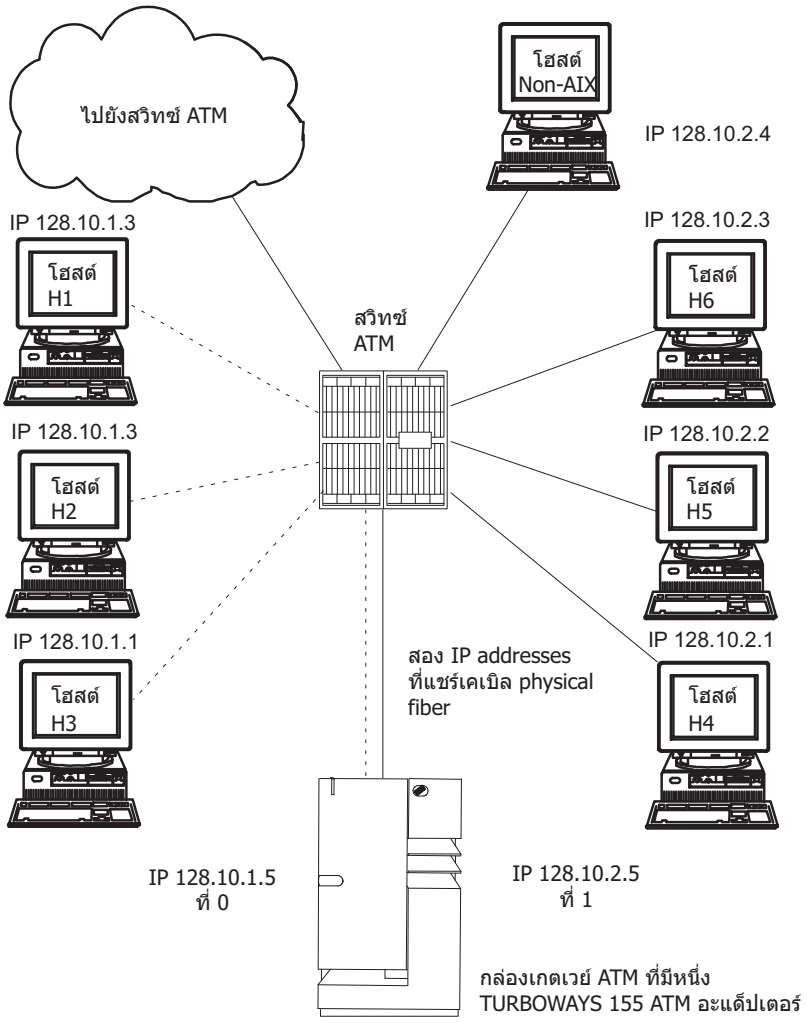
เพื่อเชื่อมต่อกับโฮสต์บน IP ซับเน็ต แบบโลจิคัลอื่น เฉพาะ VPI: การเชื่อมต่อ VCI ไปยังเกตเวย์ต้องถูกสร้าง (VPI:VCIs สำหรับใช้เพื่อการแสดงเท่านั้น)

ATM เกตเวย์จะมีหนึ่ง ATM ที่มี 2 IP แอดเดรสที่แบ่งใช้สายเคเบิลแบบฟิสิคัลเดียวกัน

SVC เน็ตเวิร์ก:

ใช้ตัวแทนของ SVC เน็ตเวิร์กต่อไปนี้เป็นตัวอย่างสำหรับตั้งค่าเน็ตเวิร์กของคุณ

การใช้ รูปที่ 16 ในหน้า 181 เป็นตัวอย่าง ลองนึกภาพที่โฮสต์ H3 ต้องการเรียก H4 H1 เป็น ARP เซิร์ฟเวอร์สำหรับซับเน็ตที่ 1 และ H6 เป็น ARP เซิร์ฟเวอร์สำหรับซับเน็ตที่ 2 สมมุติว่า subnet mask เป็น 255.255.255.0 สเตชันที่มีแอดเดรส 128.10.1.X จะเป็นสมาชิกของซับเน็ตหนึ่ง โดยที่สเตชันที่มีแอดเดรส 128.10.2.X จะเป็นสมาชิกของซับเน็ตที่สอง ดูลิสต์ของตัวแทนการตั้งค่าโฮสต์โดยใช้ SVCs ต่อไปนี้



รูปที่ 16. ตัวแทนของ ATM เน็ตเวิร์ก

ภาพนี้แสดงโครงร่างของ ATM เน็ตเวิร์กในโทโปโลยีแบบดาวทั่วไป ที่ศูนย์กลางของดาวเป็น ATM สวิตช์ จำนวนของโฮสต์ของ IP จะเป็นสาขาออกมาจากสวิตช์ จะลิงก์กับ ATM สวิตช์อื่น และหนึ่งของ ATM เกตเวย์และอะแด็ปเตอร์

ตารางที่ 61. ลิสต์ของตัวแทนการตั้งค่าโฮสต์

เน็ตเวิร์กอินเตอร์เฟสไดรเวอร์	IP แอดเดรส	ARP เซิร์ฟเวอร์	ARP เซิร์ฟเวอร์แอดเดรส	เกตเวย์แอดเดรส
Host H1 at0	128.10.1.3	ใช่		128.10.1.5
Host H3 at0	128.10.1.1	ไม่	ATM แอดเดรสของ H1	128.10.1.5
Gateway at0	128.10.1.5	ไม่	ATM แอดเดรสของ H1	
at1	128.10.2.5	ไม่	ATM แอดเดรสของ H6	
Host H4 at0	128.10.2.1	ไม่	ATM แอดเดรสของ H6	128.10.2.5
Host H6 at0	128.10.2.3	ใช่		128.10.2.5

หมายเหตุ: แต่ละซับเน็ตต้องการเพียง ARP เดียวเท่านั้น

เนื่องจาก H3 รู้ว่าแอดเดรส 128.10.2.1 ไม่อยู่บนซับเน็ตของมัน H3 จะถาม H1 เพื่อให้ resolve IP แอดเดรสของดีฟอลต์เกตเวย์เป็น ATM แอดเดรส จากนั้น H3 จะเรียกไปยังเกตเวย์ เกตเวย์จะรู้ว่าข้อมูลจะออกไปยังซับเน็ตที่สองและถาม H6 เพื่อให้สามารถ resolve IP แอดเดรสของ H4 เป็น ATM แอดเดรส จากนั้นการเชื่อมต่อจะถูกสร้างระหว่าง H3 และเกตเวย์ และเกตเวย์และ H4

การตั้งค่า ATM อะแด็ปเตอร์:

เมื่อต้องการกำหนดค่าอะแด็ปเตอร์ ATM ของคุณ ใช้ SMIT fast path `smit chg_atm`

เลือกชื่ออะแด็ปเตอร์ จากนั้นใช้ความช่วยเหลืออแบบ และลิสต์แบบหลายตัวเลือกเพื่อตัดสินใจว่าจะแก้ไขใดที่จะทำกับคอนฟิกูเรชันของคุณ

สถิติของ ATM อะแด็ปเตอร์:

คำสั่ง **atmstat** สามารถใช้เพื่อให้ได้รับสถิติของ ATM อะแด็ปเตอร์

การใช้คำสั่ง **atmstat** กับแฟล็ก **-r** จะรีเซ็ตสถิติ รูปแบบของคำสั่งคือ `atmstat` *`DeviceName`* คำสั่งนี้ส่งคืนชุดของสถิติต่อไปนี้

สถิติการส่ง:

คำสั่ง **atmstat** สามารถใช้เพื่อให้ได้รับสถิติการส่งของ ATM

`Packets:`
ฟิลด์นี้ประกอบด้วยจำนวนของแพ็กเก็ต (หรือ PDUs) ที่ถูกส่ง

`Bytes:` ฟิลด์นี้ประกอบด้วยจำนวนของไบต์ที่ถูกส่ง เหล่านี้เป็นไบต์ของผู้ใช้โอเวอร์เฮดของ ATM ตัวอย่างเช่นส่วนหัวของ ATM เซลล์ และส่วนหางของ AAL 5 PDU จะไม่ถูกรวม

`Interrupts:`
ฟิลด์นี้ไม่ถูกใช้

`Transmit Errors:`
ฟิลด์นี้ประกอบด้วยจำนวนของข้อผิดพลาดในการส่งสำหรับอุปกรณ์นี้

`Packets Dropped:`
ฟิลด์นี้ประกอบด้วยจำนวนของ Transmit Packets ที่ถูกดร็อบ ตัวอย่างเช่น เนื่องจากเงื่อนไขที่บัฟเฟอร์เต็ม

`Max Packets on S/W Transmit Queue:`
ฟิลด์นี้ใช้ไม่ได้กับ ATM

`S/W Transmit Queue Overflow:`
ฟิลด์นี้ใช้ไม่ได้กับ ATM

`Current S/W + H/W Transmit Queue Length:`
ฟิลด์นี้ประกอบด้วยความยาวของคิวการส่งในปัจจุบัน

`Cells Transmitted:`
ฟิลด์นี้ประกอบด้วยจำนวนของเซลล์ที่ถูกส่งโดยอุปกรณ์นี้

`Out of Xmit Buffers:`
ฟิลด์นี้ประกอบด้วยจำนวนของแพ็กเก็ตที่ถูกดร็อบเนื่องจากเงื่อนไขไม่มีบัฟเฟอร์ xmit

Current HW Transmit Queue Length:
ฟิลด์นี้ประกอบด้วยจำนวนปัจจุบันแพ็กเก็ตการส่งบนคิวของฮาร์ดแวร์

Current SW Transmit Queue Length:
ฟิลด์นี้ใช้ไม่ได้กับ ATM

สถิติที่ได้รับ:

สถิติต่อไปนี้จะติดตามรายการต่างๆที่ได้รับโดย ATM

Packets:
ฟิลด์นี้ประกอบด้วยจำนวนของแพ็กเก็ต (หรือ PDUs) ที่ได้รับ

Bytes: ฟิลด์นี้ประกอบด้วยจำนวนของไบต์ที่ได้รับ เหล่านี้เป็นไบต์ของผู้ใช้โอเวอร์เฮดของ ATM ตัวอย่างเช่นส่วนหัวของ ATM เซลล์ และส่วนหางของ AAL 5 PDU จะไม่ถูกรวม

Interrupts:
ฟิลด์นี้ประกอบด้วยจำนวนของอินเตอร์รัปต์ที่ได้รับโดยระบบสำหรับตัวระบุ adapter-to-system บางเหตุการณ์ที่เป็นสาเหตุของอินเตอร์รัปต์เหล่านี้เป็นแพ็กเก็ตที่ได้รับ การส่งการระบุว่าทำแล้ว และอื่นๆ

Receive Errors:
ฟิลด์นี้ประกอบด้วยจำนวนของข้อผิดพลาดที่ได้รับสำหรับอุปกรณ์นี้

Packets Dropped:
ฟิลด์นี้ประกอบด้วยจำนวนของแพ็กเก็ตที่ได้รับที่ถูกดร็อบ ตัวอย่างเช่น เนื่องจากเงื่อนไขที่บัฟเฟอร์เต็ม

Bad Packets:
ฟิลด์นี้ใช้ไม่ได้กับ ATM

Cells Received:
ฟิลด์นี้ประกอบด้วยจำนวนของเซลล์ที่ได้รับโดยอุปกรณ์นี้

Out of Rcv Buffers:
ฟิลด์นี้ประกอบด้วยจำนวนของแพ็กเก็ตที่ถูกดร็อบเนื่องจากเงื่อนไขจากบัฟเฟอร์ที่รับเต็ม

CRC Errors:
ฟิลด์นี้ประกอบด้วยจำนวนของแพ็กเก็ตที่ได้รับที่มีข้อผิดพลาด CRC

Packets Too Long:
ฟิลด์นี้ประกอบด้วยจำนวนของแพ็กเก็ตที่ได้รับที่มีขนาดเกินขนาดสูงสุดของ PDU

Incomplete Packets:
ฟิลด์นี้ประกอบด้วยจำนวนของแพ็กเก็ตที่ได้รับที่ไม่สมบูรณ์

Cells Dropped:
ฟิลด์นี้ประกอบด้วยจำนวนของเซลล์ที่ถูกดร็อบ เซลล์อาจถูกดร็อบเนื่องจากหลายสาเหตุ เช่น bad header error control (HEC) เงื่อนไขที่บัฟเฟอร์เต็ม และอื่นๆ

สถิติทั่วไป:

ฟิลด์เหล่านี้ให้สถิติทั่วไปสำหรับ ATM อะแด็ปเตอร์

`No mbuf Errors:`
ฟิลด์นี้ประกอบด้วยจำนวนของคำร้องขอ mbuf ที่ถูกปฏิเสธ

`Adapter Loss of Signals:`
ฟิลด์นี้ประกอบด้วยจำนวนครั้งที่อะแด็ปเตอร์พบว่าสัญญาณหายไป

`Adapter Reset Count:`
ฟิลด์นี้ประกอบด้วยจำนวนครั้งที่อะแด็ปเตอร์ถูกรีเซ็ต

`Driver Flags: Up Running Simplex`
ฟิลด์นี้ประกอบด้วยแฟล็ก neighborhood discovery daemon (NDD)

`Virtual Connections in use:`
ฟิลด์นี้ประกอบด้วยจำนวนของการเชื่อมต่อเสมือนที่ถูกจัดสรรหรือถูกใช้งาน

`Max Virtual Connections in use:`
ฟิลด์นี้ประกอบด้วยจำนวนสูงสุดของการเชื่อมต่อเสมือนที่ถูกจัดสรรตั้งแต่การรีเซ็ตครั้งล่าสุดของสถิติ

`Virtual Connections Overflow:`
ฟิลด์นี้ประกอบด้วยจำนวนของคำร้องขอจัดสรรการเชื่อมต่อเสมือนที่ถูกปฏิเสธ

`SVC UNI Version:`
ฟิลด์นี้ประกอบด้วย UNI เวอร์ชันปัจจุบันของโปรโตคอลการส่งสัญญาณที่ถูกใช้

สถิติที่ระบุว่าหรับ PCI ATM อะแด็ปเตอร์:

สถิติต่อไปนี้ถูกระบุกับ PCI ATM อะแด็ปเตอร์

`Total 4K byte Receive Buffers: 768 Using: 512`
ข้อความนี้ประกอบด้วยจำนวนของบัฟเฟอร์ตัวรับที่ถูกจัดสรรพร้อมกับจำนวนที่ถูกใช้ในปัจจุบัน

`Max 4K byte Receive Buffers limit: 1228 max_used: 514`
ข้อความนี้ประกอบด้วยจำนวนสูงสุดของบัฟเฟอร์ตัวรับพร้อมกับจำนวนที่ถูกใช้ตั้งแต่อะแด็ปเตอร์ถูกตั้งค่าครั้งล่าสุดหรือถูกเปิด

อินเตอร์เฟสเครือข่าย TCP/IP

อินเตอร์เฟสเครือข่าย **TCP/IP** จัดวางรูปแบบ IP ดาตาแกรม ที่เลเยอร์เครือข่ายเป็นแพ็กเก็ตที่ระบุเทคโนโลยีเครือข่ายสามารถทำความเข้าใจ และถ่ายโอน

อินเตอร์เฟสเครือข่ายเป็นซอฟต์แวร์เฉพาะเครือข่ายที่ติดต่อ กับไดรเวอร์อุปกรณ์เฉพาะเครือข่าย และเลเยอร์ IP เพื่อจัดเตรียม เลเยอร์ IP ด้วยอินเตอร์เฟสที่สอดคล้องกับอะแด็ปเตอร์เครือข่ายทั้งหมดที่ อาจปรากฏ

เลเยอร์ IP เลือกอินเตอร์เฟสเครือข่ายที่เหมาะสมตามแอดเดรสปลายทาง ของแพ็กเก็ตที่ถ่ายโอน แต่ละอินเตอร์เฟสเครือข่ายมีแอดเดรสเครือข่าย เลเยอร์อินเตอร์เฟสเครือข่ายรับผิดชอบการเพิ่มหรือลบส่วนหัวโปรโตคอลของ ของลิงก์เลเยอร์จำเป็นต้องส่งข้อความไปยังปลายทาง ไดรเวอร์อุปกรณ์ของ **อะแด็ปเตอร์เครือข่าย** ควบคุมการ์ดอะแด็ปเตอร์เครือข่าย

แม้ว่าจะไม่จำเป็น อินเตอร์เฟสเครือข่ายเชื่อมโยงกับอะแด็ปเตอร์เครือข่าย สำหรับอินสแตนซ์ อินเตอร์เฟสแบบ loopback ไม่มีอะแด็ปเตอร์เครือข่าย เชื่อมโยงกับมัน เครื่องต้องมีการ์ดอะแด็ปเตอร์หนึ่งการ์ดสำหรับแต่ละเครือข่าย (ไม่ใช่ชนิดเครือข่าย) เพื่อเชื่อมต่อ อย่างไรก็ตาม เครื่องต้องการหนึ่งสำเนา ของซอฟต์แวร์อินเตอร์เฟสเครือข่ายสำหรับให้แต่ละอะแด็ปเตอร์

เครือข่ายใช้งาน สำหรับอินสแตนซ์ ถ้าโฮสต์เชื่อมต่อกับเครือข่ายโทเค็นริงสองวง ก็ต้องมี การ์ดอะแด็ปเตอร์เครือข่ายสองการ์ด อย่างไรก็ตาม ต้องการเพียงสำเนาเดียวของซอฟต์แวร์อินเตอร์เฟสเครือข่าย **โทเค็นริง** และหนึ่งสำเนาของไดรเวอร์อุปกรณ์โทเค็นริง

TCP/IP สนับสนุนชนิดของอินเตอร์เฟสเครือข่าย:

- Standard Ethernet Version 2 (en)
- IEEE 802.3 (et)
- Token-ring (tr)
- **Serial Line Internet Protocol (SLIP)**
- Loopback (lo)
- FDDI
- Serial Optical (so)
- ATM (at)
- **Point-to-Point Protocol (PPP)**
- Virtual IP Address (vi)

อินเตอร์เฟสอีเทอร์เน็ต, 802.3 และโทเค็นริงสำหรับใช้กับเครือข่ายท้องถิ่น (LAN) อินเตอร์เฟส **SLIP** สำหรับใช้กับการเชื่อมต่ออนุกรม อินเตอร์เฟส loopback ใช้สำหรับโฮสต์เพื่อส่งข้อความกลับให้โฮสต์เอง อินเตอร์เฟส Serial Optical ใช้สำหรับเครือข่ายจุดต่อจุดแบบอ็อพติคัล โดยใช้ตัวจัดการอุปกรณ์ Serial Optical Link อินเตอร์เฟส ATM สำหรับใช้กับการเชื่อมต่อแบบ ATM 100 Mb/วินาที และ 155 Mb/วินาที **โปรโตคอล Point to Point** มักใช้บ่อยเมื่อเชื่อมต่อกับคอมพิวเตอร์หรือเครือข่ายอื่น ผ่านโมเด็ม อินเตอร์เฟส Virtual IP Address (หรือเรียกว่า *อินเตอร์เฟสเสมือน*) ไม่เชื่อมโยงกับอะแด็ปเตอร์เครือข่ายจำเพาะ หลายอินสแตนซ์ ของอินเตอร์เฟสเสมือนสามารถกำหนดคอนฟิกบนโฮสต์ เมื่ออินเตอร์เฟสเสมือน ถูกกำหนดคอนฟิก แอดเดรสของอินเตอร์เฟสเสมือนแรกกลายเป็นแอดเดรสต้นทาง จนกว่าแอ็พพลิเคชันจะเลือกอินเตอร์เฟสอื่น โปรเซสที่ใช้ IP แอดเดรสเสมือนเป็นแอดเดรสต้นทางสามารถส่งแพ็กเก็ตผ่านอินเตอร์เฟสเครือข่ายใดๆ ที่จัดเตรียมเส้นทางที่ดีที่สุดสำหรับปลายทาง แพ็กเก็ตขาเข้า กำหนดเป้าหมายสำหรับ IP แอดเดรสเสมือนถูกส่งไปยังโปรเซสโดยไม่คำนึง ถึงอินเตอร์เฟส

การตั้งค่าเน็ตเวิร์กอินเตอร์เฟสโดยอัตโนมัติ

เมื่อเน็ตเวิร์กอะแด็ปเตอร์ใหม่ถูกติดตั้งบนระบบ ระบบปฏิบัติการจะเพิ่มเน็ตเวิร์กอะแด็ปเตอร์ที่เหมาะสมสำหรับอะแด็ปเตอร์นั้นโดยอัตโนมัติ

ตัวอย่างเช่น ถ้าคุณติดตั้งโทเค็นริงอะแด็ปเตอร์ในระบบของคุณ ระบบปฏิบัติการจะกำหนดชื่อ `tok0` และเพิ่มเน็ตเวิร์กอินเตอร์เฟสชื่อ `tr0` ถ้าคุณติดตั้ง Ethernet อะแด็ปเตอร์ในระบบของคุณ ระบบปฏิบัติการจะกำหนดชื่อ `ent0` และเพิ่มทั้ง Ethernet เวอร์ชัน 2 และ IEEE 802.3 อินเตอร์เฟส ชื่อ `en0` และ `et0` ตามลำดับ

ในกรณีส่วนใหญ่ จะมีความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างชื่ออะแด็ปเตอร์และชื่อเน็ตเวิร์กอินเตอร์เฟส ตัวอย่างเช่น โทเค็นริงอะแด็ปเตอร์ `tok0` จะสอดคล้องกับอินเตอร์เฟส `tr0` อะแด็ปเตอร์ `tok1` จะสอดคล้องกับอินเตอร์เฟส `tr1` เป็นต้น เช่นเดียวกัน Ethernet อะแด็ปเตอร์ `ent0` จะสอดคล้องกับอินเตอร์เฟส `en0` (สำหรับ Ethernet เวอร์ชัน 2) และ `et0` (สำหรับ IEEE 802.3) และอะแด็ปเตอร์ `ent1` จะสอดคล้องกับอินเตอร์เฟส `en1` (สำหรับ Ethernet เวอร์ชัน 2) และ `et1` (สำหรับ IEEE 802.3)

ในกรณีของ ATM ตาม RFC1577 เป็นไปได้ที่สเตชัน ATM จะเป็นส่วนของหลาย IP Subnetworks แบบโลจิคัล ในกรณีนี้หลายอินเตอร์เฟสจะเชื่อมโยงกับอุปกรณ์เดียว นี่ต้องการให้อินเตอร์เฟสถูกเพิ่มและชื่ออุปกรณ์ถูกกำหนดให้กับมัน

หมายเหตุ: ภาพใต้เงื่อนไขปกติ คุณไม่ต้องลบหรือเพิ่มเน็ตเวิร์กอินเตอร์เฟสแบบแมนวล อย่างไรก็ตาม บางโพรซีเดอร์การกำหนดปัญหาบางโพรซีเดอร์อาจต้องการให้คุณทำเช่นนั้น ในกรณีนี้ ใช้ SMIT fast path, **smit inet** เพื่อลบหรือสร้างอินเตอร์เฟส ที่เหมาะสมใหม่

TCP/IP ค่าการตั้งค่าดีฟอลต์

ที่การเริ่มทำงานของระบบแต่ละครั้ง ระบบปฏิบัติการจะตั้งค่าเน็ตเวิร์กอินเตอร์เฟสซอฟต์แวร์โดยอัตโนมัติโดยขึ้นอยู่กับข้อมูลในฐานข้อมูล ODM เมื่อเริ่มต้น เน็ตเวิร์กอินเตอร์เฟสจะถูกตั้งค่าด้วยค่าดีฟอลต์

เพื่อที่จะสื่อสารผ่านเน็ตเวิร์กอินเตอร์เฟสที่กำหนด อินเตอร์เน็ตแอดเดรสต้องถูกตั้ง นี่เป็นแอ็ตทริบิวต์เดียวที่คุณต้องตั้ง แอ็ตทริบิวต์ที่จำเป็นอื่นทั้งหมดสามารถใช้ค่าดีฟอลต์ ค่าดีฟอลต์สำหรับแต่ละเน็ตเวิร์กอินเตอร์เฟสเป็นดังต่อไปนี้

TCP/IP ค่า Ethernet ดีฟอลต์:

แอ็ตทริบิวต์อะแด็ปเตอร์เครือข่าย Ethernet ที่ถูกต้องสามารถมี ค่าที่เปลี่ยนแปลงโดยใช้เมนู Network Interface Selection ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
netaddr		
state	ไม่ทำงาน	up, down, detach
arp	ใช่	ใช่, ไม่ใช่
netmask		
broadcast		

แอ็ตทริบิวต์ไดรเวอร์อุปกรณ์เครือข่าย Ethernet ที่ถูกต้อง ถูกแสดงคู่กับค่าดีฟอลต์ ซึ่งสามารถเปลี่ยนแปลงโดยใช้ เมนู Network Interface Drivers ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
mtu	1500	60 ถึง 1500

TCP/IP ค่า 802.3 ดีฟอลต์:

แอ็ตทริบิวต์อะแด็ปเตอร์เครือข่าย 802.3 ที่ถูกต้องสามารถมีค่า ที่เปลี่ยนแปลงโดยใช้เมนู Network Interface Selection ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
netaddr		
state	ไม่ทำงาน	up, down, detach
arp	ใช่	ใช่, ไม่ใช่
netmask		
broadcast		

แอ็ตทริบิวต์ไดรเวอร์อุปกรณ์เครือข่าย 802.3 ที่ถูกต้อง ถูกแสดงคู่กับค่าดีฟอลต์ ซึ่งสามารถเปลี่ยนแปลงโดยใช้ เมนู Network Interface Drivers ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
mtu	1492	60 ถึง 1492

TCP/IP ค่าโทเค็นริงดีฟอลต์:

แอ็ตทริบิวต์อะแด็ปเตอร์เครือข่ายโทเค็นริงที่ถูกต้องสามารถมีค่า ที่เปลี่ยนแปลงโดยใช้เมนู Network Interface Selection ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
netaddr		
netmask		
state	ไม่ทำงาน	up, down, detach
arp	ใช่	ใช่, ไม่ใช่
hwloop	ไม่	ใช่, ไม่ใช่
netmask		
broadcast		
allcast	ไม่	ใช่, ไม่ใช่

แอ็ตทริบิวต์ไดรเวอร์อุปกรณ์เครือข่ายโทเค็นริงที่ถูกต้อง ถูกแสดงคู่กับค่าดีฟอลต์ ซึ่งสามารถเปลี่ยนแปลงโดยใช้ เมนู Network Interface Drivers ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
mtu (4Mbps)	1500	60 ถึง 4056
mtu (16Mbps)	1500	60 ถึง 17960

หมายเหตุ: เมื่อทำงานผ่านบริดจ์ ค่าดีฟอลต์ 1500 สำหรับ maximum transmission unit (MTU) ควรถูกเปลี่ยนเป็นค่าที่น้อยกว่าฟิลด์ maximum information (maximum I-frame) ที่ถูกประกาศโดยบริดจ์ในฟิลด์ routing control อยู่ 8 ตัวอย่างเช่น

ถ้าค่าของ maximum I-frame เป็น 1500 ในฟิลด์ routing control ขนาดของ MTU ควรถูกตั้งเป็น 1492 นี่สำหรับโทเค็นริงเน็ตเวิร์กอินเตอร์เฟสเท่านั้น สำหรับข้อมูล เพิ่มเติม ให้ดูที่ “ปัญหา TCP/IP กับบริดจ์ Token-Ring/Token-Ring” ในหน้า 455

เมื่อใช้ อะแด็ปเตอร์ IBM® 16/4 PowerPC token-ring (ISA) MTU ถูกจำกัดที่ 2000

TCP/IP ค่า SLIP ดีฟอลต์:

แอ็ตทริบิวต์อะแด็ปเตอร์เครือข่าย SLIP ที่ถูกต้องสามารถมีค่า ที่เปลี่ยนแปลงโดยใช้เมนู Network Interface Selection ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
netaddr		
dest		
state	ทำงาน	up, down, detach
netmask		

แอ็ตทริบิวต์ไดรเวอร์อุปกรณ์เครือข่าย SLIP ต่อไปนี้แสดง คู่กับค่าดีฟอลต์ดังแสดงภายใต้เมนู Network Interface Drivers ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
mtu	1006	60 ถึง 4096

TCP/IP ค่า Serial Optical ดีฟอลต์:

ตัวแปลงแชนเนลเครือข่าย Valid Serial Optical สามารถ เปลี่ยนแปลงค่าโดยใช้เมนู Network Interface Selection ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
netaddr		
state	ไม่ทำงาน	up, down, detach
netmask		

แอ็ตทริบิวต์ตัวจัดการอุปกรณ์เครือข่ายออพติคัลอนุกรมที่ถูกต้องต่อไปนี้ แสดงคู่กับค่าดีฟอลต์ดังแสดงภายใต้เมนู Network Interface Drivers ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
mtu	61428	1 ถึง 61428

TCP/IP ค่า ATM ดีฟอลต์:

แอ็ตทริบิวต์อะแด็ปเตอร์เครือข่าย ATM ที่ถูกต้องสามารถมีค่า ที่เปลี่ยนแปลงโดยใช้เมนู Network Interface Selection ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
netaddr		
netmask		
state	ทำงาน	up, down, detach
ชนิดของการเชื่อมต่อ	svc_s	svc_c, svc_s, pvc
ATM Server Address		
อุปกรณ์อื่น		
เวลาที่ไม่ได้ทำงาน	60	1 ถึง 60
Best Effort Bit Rate (UBR) ในหน่วย kbits/วินาที	0	1 ถึง 155,000

แอ็ตทริบิวต์ไดรเวอร์อุปกรณ์เครือข่าย ATM ต่อไปนี้แสดง คู่กับค่าดีฟอลต์ดังแสดงภายใต้เมนู Network Interface Drivers ใน SMIT

แอ็ตทริบิวต์	ค่าดีฟอลต์	ค่าที่เป็นไปได้
mtu	9180	1 ถึง 64K

หมายเหตุ: ผู้บริหารเน็ตเวิร์กต้องทำด้วยความระมัดระวังขณะที่เปลี่ยนขนาดของ MTU จากค่าดีฟอลต์เป็นค่าอื่น ขนาดของ MTU ต้องเท่ากับสเตชันอื่นบนเน็ตเวิร์ก

ถ้า PVCs ถูกใช้บนอินเตอร์เฟส VPI:VCIs ต้องถูกกำหนด นี่จะถูกกระทำผ่านเมนู Network Interface Selection อ็อพชัน `PVCs for IP over ATM Network` บนเมนูนี้ถูกใช้เพื่อลิสต์ เพิ่ม เปลี่ยน หรือลบ PVCs

ความสัมพันธ์ของหลายเน็ตเวิร์กอินเตอร์เฟสบนเน็ตเวิร์กเดียวกัน

ถ้าหลายเน็ตเวิร์กอินเตอร์เฟสถูกเชื่อมอยู่บนเน็ตเวิร์กเดียว แต่ละอินเตอร์เฟสต้องมี IP แอดเดรสที่เป็นหนึ่งเดียว

คุณลักษณะการเราต์หลายพาธอนุญาตให้เพิ่มเราต์ลงใน ตารางการเราต์ IP สำหรับอินเตอร์เฟสหลายพาธบน subnet เดียวกัน นี่จะยอมให้ทราฟิกขาออกที่จะสลับระหว่างอินเตอร์เฟสแทนที่จะเป็นถูกส่งผ่านอินเตอร์เฟสเดียว

การจัดการเน็ตเวิร์กอินเตอร์เฟส

เมื่อต้องการจัดการอินเตอร์เฟสเครือข่าย ใช้ WSM Network, FastPath (แอ็พพลิเคชัน) หรืองานในตารางนี้

ตารางที่ 62. งานการจัดการเน็ตเวิร์กอินเตอร์เฟส

ภารกิจ	วิธีลัด SMIT	คำสั่งหรือไฟล์
ลิสต์อุปกรณ์เน็ตเวิร์กทั้งหมด	`smit lsinet`	**lsdev -C -c if**
ตั้งค่าอุปกรณ์เน็ตเวิร์ก	`smit chinet`	ดูคำสั่ง **ifconfig** และไฟล์ `rc.net`
การเปลี่ยนข้อมูลเน็ตเวิร์กอินเตอร์เฟสด้วย `/usr` ที่ถูกเมาต์แบบรีโมต	`smit chdev`[1,2]	**chgif**[1,2]
ดูสถิติสำหรับเน็ตเวิร์กอินเตอร์เฟส		**netstat -v**

หมายเหตุ:

1. เปลี่ยนจากการเมาต์ `/usr` แบบรีโมตจะมีผลเฉพาะ Information Database (ODM) จนกว่าเน็ตเวิร์กจะถูกรีสตาร์ท หรือจนกว่าคำสั่ง **ifconfig** จะถูกใช้เพื่อให้การเปลี่ยนแปลงมีผลใช้งาน
2. เมื่อใช้การเมาต์ `/usr` แบบรีโมต ต้องระวังไม่แก้ไขอินเตอร์เฟสที่กำลังถูกใช้ เนื่องจากเป็นตำแหน่งของไลบรารี คำสั่ง และเคอร์เนล

อ็อพชันที่ระบุของเน็ตเวิร์กอินเตอร์เฟส

TCP/IP อินเตอร์เฟสต้องถูกปรับแต่งเป็นพิเศษเพื่อให้ได้ประสิทธิภาพของเน็ตเวิร์กความเร็วสูงที่ดี (100 Mb หรือมากกว่า) การกระทำดังกล่าวมีความซับซ้อนโดยความจริงที่ว่าหลายเน็ตเวิร์กอินเตอร์เฟสและการรวมกันของ **TCP/IP** อินเตอร์เฟสความเร็วสูงแบบดั้งเดิมสามารถถูกใช้บนระบบเดียว

ในระบบปฏิบัติการ AIX Interface Specific Network Options (ISNO) ช่วยให้ผู้ดูแลระบบ สามารถปรับแต่ละอินเตอร์เฟส **TCP/IP** ทีละตัวเพื่อให้ได้ประสิทธิภาพที่ดีที่สุด

มีพารามิเตอร์ ISNO 5 พารามิเตอร์สำหรับแต่ละอินเตอร์เฟสที่ได้รับการสนับสนุน: **rfc1323, tcp_nodelay, tcp_sendspace, tcp_recvspace** และ **tcp_mssdflt** เมื่อตั้งค่า ค่าของพารามิเตอร์เหล่านี้จะทับพารามิเตอร์ที่ชื่อเดียวกันที่เป็นของทั้งระบบ ที่ถูกตั้งด้วยคำสั่ง **no** เมื่ออ็อพชัน ISNO ไม่ถูกตั้งสำหรับอินเตอร์เฟสนั้นๆ อ็อพชันของระบบจะถูกใช้ เมื่ออ็อพชันถูกตั้งโดยแอ็พพลิเคชันสำหรับซ็อกเก็ตนั้นๆโดยใช้รูทีนย่อย **setsockopt** อ็อพชันนั้นจะทับ ISNOs

เน็ตเวิร์กอ็อพชัน **use_isno** ตั้งด้วยคำสั่ง **no** ต้องมีค่าเป็น 1 เพื่อให้ ISNOs มีผลใช้งาน ค่าดีฟอลต์สำหรับ **use_isno** คือ 1

อะแด็ปเตอร์ความเร็วสูงบางตัวมีพารามิเตอร์ ISNO ที่ถูกตั้งโดยดีฟอลต์ในฐานข้อมูล ODM

Gigabit Ethernet อินเตอร์เฟส เมื่อถูกตั้งค่าเพื่อใช้ MTU เป็น 9000 ใช้ค่า ISNO ต่อไปนี้โดยดีฟอลต์ :

ชื่อ	AIX 4.3.3 Value	AIX 4.3.3 (4330-08) Value	AIX 5.1 (และหลังจากนั้น) ค่า
tcp_sendspace	131072	262144	262144
tcp_recvspace	92160	131072	131072
rfc1323	1	1	1

Gigabit Ethernet อินเตอร์เฟส เมื่อถูกตั้งค่าเพื่อใช้ MTU เป็น 1500 ใช้ค่า ISNO ต่อไปนี้โดยดีฟอลต์ :

ชื่อ	AIX 4.3.3 Value	AIX 4.3.3 (4330-08) Value	AIX 5.1 (และหลังจากนั้น) ค่า
tcp_sendspace	65536	131072	131072
tcp_recvspace	16384	65536	65536
rfc1323	0	ไม่ได้ตั้ง	ไม่ได้ตั้ง

ATM อินเตอร์เฟส เมื่อถูกตั้งค่าเพื่อใช้ MTU เป็น 1500 ใช้ค่า ISNO ต่อไปนี้โดยดีฟอลต์ :

ชื่อ	AIX 4.3.3 Value	AIX 4.3.3 (4330-08) Value	AIX 5.1 (และหลังจากนั้น) ค่า
tcp_sendspace	16384	ไม่ได้ตั้ง	ไม่ได้ตั้ง
tcp_recvspace	16384	ไม่ได้ตั้ง	ไม่ได้ตั้ง
rfc1323	0	ไม่ได้ตั้ง	ไม่ได้ตั้ง
tcp_nodelay	0	ไม่ได้ตั้ง	ไม่ได้ตั้ง
tcp_mssdflt	512	ไม่ได้ตั้ง	ไม่ได้ตั้ง

ATM อินเตอร์เฟส เมื่อถูกตั้งค่าเพื่อใช้ MTU เป็น 65527 ใช้ค่า ISNO ต่อไปนี้โดยดีฟอลต์ :

ชื่อ	AIX 4.3.3 Value	AIX 4.3.3 (4330-08) Value	AIX 5.1 (และหลังจากนั้น) ค่า
tcp_sendspace	655360	655360	655360
tcp_recvspace	655360	655360	655360
rfc1323	0	1	1
tcp_nodelay	0	ไม่ได้ตั้ง	ไม่ได้ตั้ง
tcp_mssdflt	512	ไม่ได้ตั้ง	ไม่ได้ตั้ง

ATM อินเตอร์เฟส เมื่อถูกตั้งค่าเพื่อใช้ MTU เป็น 9180 ใช้ค่า ISNO ต่อไปนี้โดยดีฟอลต์ :

ชื่อ	AIX 4.3.3 Value	AIX 4.3.3 (4330-08) Value	AIX 5.1 (และหลังจากนั้น) ค่า
tcp_sendspace	65536	65536	65536
tcp_recvspace	65536	65536	65536
rfc1323	0	ไม่ได้ตั้ง	ไม่ได้ตั้ง
tcp_nodelay	0	ไม่ได้ตั้ง	ไม่ได้ตั้ง
tcp_mssdflt	512	ไม่ได้ตั้ง	ไม่ได้ตั้ง

FDDI อินเตอร์เฟส เมื่อถูกตั้งค่าเพื่อใช้ MTU เป็น 4352 ใช้ค่า ISNO ต่อไปนี้โดยดีฟอลต์ :

ชื่อ	Value
tcp_sendspace	45046
tcp_recvspace	45046

พารามิเตอร์ ISNO ไม่สามารถถูกแสดงหรือเปลี่ยนโดยใช้ SMIT มันสามารถถูกตั้งโดยใช้คำสั่ง **chdev** หรือคำสั่ง **ifconfig** คำสั่ง **ifconfig** จะเปลี่ยนค่าเฉพาะหลังจากการรีบูตครั้งต่อไป คำสั่ง **chdev** จะเปลี่ยนค่าในฐานข้อมูล ODM เพื่อที่มันจะถูกใช้ในการรีบูตครั้งต่อๆไป คำสั่ง **lsattr** หรือ **ifconfig** สามารถถูกใช้เพื่อแสดงค่าปัจจุบัน

ตัวอย่างต่อไปนี้แสดงคำสั่งที่สามารถถูกใช้ก่อนเพื่อตรวจสอบระบบและการสนับสนุนอินเตอร์เฟส และจากนั้นตั้งและตรวจสอบค่าใหม่

1. ตรวจสอบค่าทั่วไปและการสนับสนุนอินเตอร์เฟสโดยใช้คำสั่ง **no** และ **lsattr**
   - ต้องแน่ใจว่าอ็อพชัน **use_isno** ถูกเปิดใช้งานโดยใช้คำสั่งเหมือนดังต่อไปนี้ :

     ```
     $ no -a | grep isno
              use_isno=1
     ```

   - ต้องแน่ใจว่าอินเตอร์เฟสสนับสนุน 5 ISNOs ใหม่โดยใช้คำสั่ง **lsattr -El** ดังแสดงในต่อไปนี้ :

     ```
     $ lsattr -E -l en0 -H
              attribute   value  description
              rfc1323                     N/A
              tcp_nodelay                 N/A
              tcp_sendspace               N/A
              tcp_recvspace               N/A
              tcp_mssdflt                 N/A
     ```

2. ตั้งค่าที่ระบุของเซ็ตโดยใช้คำสั่ง **ifconfig** หรือ **chdev** คำสั่ง **ifconfig** จะตั้งค่าที่ถูกแนะนำเป็นการชั่วคราว สำหรับการทดสอบ คำสั่ง **chdev** จะเปลี่ยน ODM ดังนั้นค่าที่ถูกปรับแต่งจะยังคงใช้ได้หลังจากรีบูต
   - ตั้ง **tcp_recvspace** และ **tcp_sendspace** เป็น 64K และเปิดใช้งาน **tcp_nodelay** โดยใช้หนึ่งในคำสั่งต่อไปนี้:

     ```
     $ ifconfig en0 tcp_recvspace 65536 tcp_sendspace 65536 tcp_nodelay 1
     $ chdev -l en0 -a tcp_recvspace=65536 -a tcp_sendspace=65536 -a tcp_nodelay=1
     ```

   - นอกจากนี้ การใช้คำสั่ง **no** จะรายงานค่าโกลบอล rfc1323=1 ผู้ใช้ root สามารถปิด **rfc1323** สำหรับการเชื่อมต่อทั้งหมดบน en0 ด้วยคำสั่งต่อไปนี้ :

     ```
     $ ifconfig en0 rfc1323 0
     $ chdev -l en0 -a rfc1323=0
     ```

3. ตรวจสอบการตั้งค่าโดยใช้คำสั่ง **ifconfig** หรือ **lsattr** ดังแสดงในตัวอย่างต่อไปนี้ :

   ```
   $ ifconfig en0 <UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
            en0: flags=e080863
                inet 9.19.161.100 netmask 0xffffff00 broadcast 9.19.161.255
                tcp_sendspace 65536 tcp_recvspace 65536 tcp_nodelay 1 rfc1323 0
   $ lsattr -El en0
            rfc1323         0           N/A              True
            tcp_nodelay     1           N/A              True
            tcp_sendspace   65536       N/A              True
            tcp_recvspace   65536       N/A              True
            tcp_mssdflt                 N/A              True
   ```

การกำหนดแอดเดรส TCP/IP

TCP/IP มีแบบแผนการกำหนดแอดเดรสอีเทอร์เน็ตที่อนุญาตให้ ผู้ใช้และแอ็พพลิเคชันระบุเครือข่ายหรือโฮสต์เฉพาะซึ่งจะสื่อสาร ด้วย

อีเทอร์เน็ตแอดเดรสทำงานคล้ายกับรหัสไปรษณีย์ โดยอนุญาตให้เราต์ข้อมูล ไปยังปลายทางที่เลือก **TCP/IP** นำเสนอมาตรฐานสำหรับ การกำหนดแอดเดรสให้กับเครือข่าย เครือข่ายย่อย โฮสต์ และซ็อกเก็ต และสำหรับ การใช้แอดเดรสพิเศษสำหรับการแพร่สัญญาณและโลคัล loopback

อินเทอร์เน็ตแอดเดรสประกอบด้วยแอดเดรสเครือข่ายและโฮสต์ (หรือโลคัล) แอดเดรส แอดเดรสสองส่วนนี้อนุญาตให้ผู้ส่งระบุเครือข่าย และโฮสต์เฉพาะบนเครือข่าย มีการกำหนดแอดเดรสเครือข่ายอย่างเป็นทางการที่ไม่ซ้ำกันให้กับ แต่ละเครือข่ายเมื่อเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตอื่น อย่างไรก็ตาม หาก เครือข่ายโลคัลจะไม่เชื่อมต่อกับเครือข่ายอินเทอร์เน็ตอื่น สามารถกำหนดแอดเดรส เครือข่ายใดๆ ที่สะดวกสำหรับการใช้งานแบบโลคัลได้

แบบแผนการกำหนดแอดเดรสอินเทอร์เน็ตประกอบด้วย Internet Protocol (IP) แอดเดรส และสองกรณีพิเศษของ IP แอดเดรสคือ: แอดเดรสการแพร่สัญญาณและ loopback แอดเดรส

อินเตอร์เน็ตแอดเดรส

อินเตอร์เน็ตโปรโตคอล (IP) ใช้ 32-บิต แอดเดรสฟิลด์แบบ 2 ส่วน

32 บิตถูกแบ่งเป็นสี่ *octets* ดังต่อไปนี้;

```
01111101   00001101   01001001   00001111
```

เลขไบนารีเหล่านี้จะแปลเป็น :

```
125              13              73              15
```

2 ส่วนของอินเตอร์เน็ตแอดเดรสเป็นส่วนของเน็ตเวิร์กแอดเดรสและส่วนของโฮสต์แอดเดรส นี่จะยอมให้รีโมตโฮสต์เพื่อระบุทั้งรีโมตเน็ตเวิร์กและโฮสต์บนรีโมตเน็ตเวิร์กเมื่อส่งข้อมูล ตามธรรมเนียม หมายเลขโฮสต์ที่เป็น 0 ถูกใช้เพื่ออ้างถึงตัวเน็ตเวิร์กเอง

TCP/IP สนับสนุน 3 คลาสของอินเตอร์เน็ตแอดเดรส : คลาส A, คลาส B, และคลาส C คลาสที่แตกต่างกันของอินเตอร์เน็ตแอดเดรสถูกกำหนดโดยวิธีที่ 32 บิตของแอดเดรสถูกจัดสรร คลาสของแอดเดรสนั้นๆ เน็ตเวิร์กถูกกำหนดโดยขึ้นอยู่กับขนาดของเน็ตเวิร์ก

คลาส A คลาส:

คลาส A คลาสประกอบด้วย 8-บิตเน็ตเวิร์กแอดเดรสและ 24-บิตโลคัลหรือโฮสต์แอดเดรส

บิตแรกในเน็ตเวิร์กแอดเดรสถูกใช้เพื่อระบุคลาสของเน็ตเวิร์ก ปล่อยให้ 7 บิตสำหรับเน็ตเวิร์กแอดเดรสจริง เนื่องจากหมายเลขสูงสุดที่ 7 บิตสามารถแทนได้ในแบบไบนารี คือ 128 จะมีเน็ตเวิร์กแอดเดรสของคลาส A ที่เป็นไปได้ 128 เน็ตเวิร์ก จาก 128 เน็ตเวิร์กแอดเดรสที่เป็นไปได้ ถูกจองไว้ 2 แอดเดรสไว้ใช้ในกรณีพิเศษ : เน็ตเวิร์กแอดเดรส 127 ถูกจองไว้สำหรับ loopback แอดเดรส และเน็ตเวิร์กแอดเดรสที่เป็นหนึ่งทั้งหมดจะระบุถึงบรอดคาสต์แอดเดรส

จะมีเน็ตเวิร์กแอดเดรสของคลาส A ที่เป็นไปได้ 126 แอดเดรส และโลคัสโฮสต์แอดเดรสที่เป็นไปได้ 16,777,216 โฮสต์ในคลาส A แอดเดรส บิตลำดับสูงสุดจะถูกตั้งเป็น 0

ที่อยู่เน็ตเวิร์ค (8 บิต)	**ที่อยู่โลคัลโฮส** (24 บิต)		
01111101	00001101	01001001	00001111

หมายเหตุ: บิตที่อยู่ลำดับแรกๆ (หรือบิตแรก) จะเป็น 0 เสมอในคลาส A

รูปที่ 17. คลาส A แอดเดรส

ภาพนี้จะแสดงโครงสร้างของคลาส A แอดเดรสทั่วไป 8 บิตแรกประกอบด้วยเน็ตเวิร์กแอดเดรส (เริ่มต้นด้วย 0 เสมอ) 24 บิตที่เหลือจะประกอบด้วยโลคัลโฮสต์แอดเดรส

octet แรกของคลาส A แอดเดรสจะอยู่ในช่วง 1 ถึง 126

คลาส B แอดเดรส:

คลาส B แอดเดรสประกอบด้วย 16-บิตเน็ตเวิร์กแอดเดรส และ 16-บิตโลคัลหรือโฮสต์แอดเดรส

สองบิตแรกในเน็ตเวิร์กแอดเดรสถูกใช้เพื่อระบุคลาสของเน็ตเวิร์ก ปล่อยให้ 14 บิตสำหรับเน็ตเวิร์กแอดเดรสจริง จะมีเน็ตเวิร์กแอดเดรสที่เป็นไปได้ 16,384 เน็ตเวิร์ก และ 65,536 โลคัลโฮสต์แอดเดรส ในคลาส B แอดเดรส บิตลำดับสูงสุดจะถูกตั้งเป็น 1 และ 0

ที่อยู่เน็ตเวิร์ค (16 บิต)		**ที่อยู่โลคัลโฮส** (16 บิต)	
10011101	00001101	01001001	00001111

หมายเหตุ: บิตที่อยู่ลำดับแรกๆ (หรือบิตแรก) จะเป็น 0 เสมอในคลาส B

รูปที่ 18. คลาส B แอดเดรส

ภาพนี้จะแสดงโครงสร้างของคลาส B แอดเดรสทั่วไป 16 บิตแรกจะประกอบด้วยเน็ตเวิร์กแอดเดรส บิตลำดับสูงสุด 2 บิตจะเป็น 1 และ 0 เสมอ 16 บิตที่เหลือจะประกอบด้วยโลคัลโฮสต์แอดเดรส

octet แรกของคลาส B แอดเดรสจะอยู่ในช่วง 128 ถึง 191

คลาส C แอดเดรส:

คลาส C แอดเดรสประกอบด้วย 24-บิตเน็ตเวิร์กแอดเดรส และ 8-บิต โลคัลโฮสต์แอดเดรส

สามบิตแรกในแอดเดรสเครือข่ายบ่งชี้ คลาสเครือข่าย ปล่อยไว้ 21 บิตสำหรับแอดเดรสเครือข่ายจริง ดังนั้น จะมีเน็ตเวิร์กที่เป็นไปได้ 2,097,152 เน็ตเวิร์ก และโลคัลโฮสต์แอดเดรสที่เป็นไปได้ 256 แอดเดรส ในคลาส C แอดเดรส บิตลำดับสูงสุดมีการตั้งค่าเป็น 1-1-0

ที่อยู่เน็ตเวิร์ค (24 บิต)			ที่อยู่โลคัลโฮส (8 บิต)
11011101	00001101	01001001	00001111

หมายเหตุ: บิตลำดับสูงสุดสามตัว (หรือบิตสามตัวแรก) จะเป็น 1-1-0 ในแอดเดรส Class C เสมอ

รูปที่ 19. คลาส C แอดเดรส

ภาพนี้แสดงโครงสร้างของคลาส C แอดเดรสทั่วไป 24 บิต แรกมีแอดเดรสเครือข่าย (บิตลำดับสูงสุดสามบิตแรก จะเป็น 1-1-0 เสมอ) 8 บิตที่เหลือจะประกอบด้วยโลคัลโฮสต์แอดเดรส

อีกนัยหนึ่ง octet ของคลาส C แอดเดรสจะอยู่ในช่วง 192 ถึง 223

เมื่อตัดสินใจว่าจะใช้เน็ตเวิร์กแอดเดรสคลาสใด คุณต้องพิจารณาจำนวนของโลคัลโฮสต์ที่จะมีบนเน็ตเวิร์ก และจำนวนของเน็ตเวิร์กย่อยที่จะมีในองค์กร ถ้าองค์กรมีขนาดเล็กและมีน้อยกว่า 256 โฮสต์ คลาส C แอดเดรสน่าจะเพียงพอ ถ้าองค์กรมีขนาดใหญ่ ดังนั้นคลาส B หรือคลาส A อาจจะเหมาะสม

หมายเหตุ: คลาส D (1-1-1-0 ในบิตลำดับสูงสุด) แอดเดรสมีไว้สำหรับมัลติคาสแอดเดรส และได้รับการสนับสนุนโดย UDP/IP ภายใต้ระบบปฏิบัติการนี้

เครื่องจะอ่านแอดเดสเป็นไบนารีโค้ด เครื่องหมายแบบดั้งเดิมสำหรับอินเตอร์เน็ตโฮสต์แอดเดรส คือ *จุดทศนิยม* ซึ่งจะแบ่ง 32-บิตแอดเดรสเป็นฟิลด์ขนาด 8-บิต 4 ฟิลด์ ค่าไบนารีต่อไปนี้:

```
0001010   00000010   00000000   00110100
```

สามารถถูกแสดงเป็น :

`010.002.000.052` หรือ `10.2.0.52`

โดยที่ค่าของแต่ละฟิลด์ถูกระบุเป็นเลขฐานสิบ และฟิลด์ถูกแยกด้วยจุด

หมายเหตุ: คำสั่ง **hostent** ไม่รู้จัก แอดเดรสต่อไปนี้: .08, .008, .09 และ .009 แอดเดรสที่มีศูนย์นำหน้า จะถูกแปลเป็นเลขฐานแปด และตัวเลขฐานแปดไม่สามารถมีเลข 8 หรือ 9

TCP/IP ต้องการอินเตอร์เน็ตแอดเดรสที่เป็นหนึ่งเดียวสำหรับแต่ละเน็ตเวิร์กอินเตอร์เฟส (อะแด็ปเตอร์) บนเน็ตเวิร์ก แอดเดรสเหล่านี้ถูกกำหนดโดย entry ในฐานข้อมูลคอนฟิกูเรชัน ซึ่งต้องตรงกับ entry ในไฟล์ `/etc/hosts` หรือฐานข้อมูล **named** ถ้าเน็ตเวิร์กใช้เนมเซิร์ฟเวอร์

อินเตอร์เน็ตแอดเดรสโดยใช้ศูนย์:

เมื่ออินเตอร์เน็ตแอดเดรสคลาส C ประกอบด้วย 0 เป็นส่วนของโฮสต์แอดเดรส (ตัวอย่างเช่น 192.9.200.0) TCP/IP จะส่ง wildcard แอดเดรสบนเน็ตเวิร์ก

เครื่องทั้งหมดที่เป็นคลาส C แอดเดรสเป็น 192.9.200.*X* (โดยที่ *X* แทนค่าระหว่าง 0 และ 254) ควรตอบสนองต่อคำร้องขอ ซึ่งจะเป็นผลให้เกิด network flood ด้วยคำร้องขอไปยังเครื่องที่ไม่มีอยู่

เช่นเดียวกัน ปัญหานี้เกิดขึ้นสำหรับคลาส B แอดเดรส เช่น 129.5.0.0 เครื่องทั้งหมดที่มีคลาส B แอดเดรสเป็น 129.5.*X*.*X* (โดยที่*X* จะแทนค่าระหว่าง 0 และ 254) จะถูกบังคับให้ตอบสนองต่อคำร้องขอ ในกรณีนี้ เนื่องจากคลาส B แอดเดรสมีเน็ตเวิร์กที่ใหญ่กว่าคลาส C แอดเดรส เน็ตเวิร์กจะถูก flood ด้วยคำร้องขอไปยังเครื่องที่มีอยู่ที่มากกว่าสำหรับเน็ตเวิร์กคลาส C

ซับเน็ตแอดเดรส

การกำหนดซับเน็ตแอดเดรสจะยอมให้ระบบแบบอิสระสร้างหลายเน็ตเวิร์กเพื่อแบ่งใช้อินเตอร์เน็ตแอดเดรสเดียวกัน

ความสามารถของซับเน็ตเวิร์กของ TCP/IP ยังทำให้เป็นไปได้ที่จะแบ่งเน็ตเวิร์กเดียว เป็นหลายโลจิคัลเน็ตเวิร์ด (ซับเน็ต) ตัวอย่างเช่น องค์กรหนึ่งสามารถมีอินเตอร์เน็ตเน็ตเวิร์กแอดเดรสเดียว ที่รู้จักโดยผู้ใช้ภายนอกองค์กร มันสามารถถูกตั้งค่าเน็ตเวิร์กภายในเป็นซับเน็ตของแผนก ในทั้งสองกรณี จะใช้ อินเตอร์เน็ตเน็ตเวิร์กแอดเดรสน้อยลงขณะที่ความสามารถในการเราต์แบบโลคัล ถูกปรับปรุง

ฟิลด์ของอินเตอร์เน็ตโปรโตคอลแอดเดรสมาตรฐานมีสองส่วน: เน็ตเวิร์ก แอดเดรส และโลคัลแอดเดรส เพื่อให้ซับเน็ตเป็นไปได้ ส่วนของโลคัลแอดเดรส ของอินเตอร์เน็ตแอดเดรสจะถูกแบ่งเป็นเลขของซับเน็ตและ หมายเลขของโฮสต์ ซับเน็ตจะถูกระบุเพื่อที่ระบบแบบอิสระแบบโลคัลสามารถเราต์ข้อความได้อย่างน่าเชื่อถือ

ในอินเตอร์เน็ตแอดเดรสคลาส A พื้นฐาน ซึ่งประกอบด้วย 8 บิตเน็ตเวิร์กแอดเดรส และ 24 บิตโลคัลแอดเดรส โลคัลแอดเดรสจะระบุเครื่องโฮสต์ที่ระบุบนเน็ตเวิร์ก

ที่อยู่เน็ตเวิร์ค (8 บิต)	**ที่อยู่โลคัลโฮส** (24 บิต)		
01111101	00001101	01001001	00001111

รูปที่ 20. คลาส A แอดเดรส

ภาพนี้จะแสดงโครงสร้างของคลาส A แอดเดรสทั่วไป 8 บิตแรกประกอบด้วยเน็ตเวิร์กแอดเดรส (เริ่มต้นด้วย 0 เสมอ) 24 บิตที่เหลือจะประกอบด้วยโลคัลโฮสต์แอดเดรส

เพื่อสร้างซับเน็ตแอดเดรสสำหรับอินเตอร์เน็ตแอดเดรสคลาส A นี้ โลคัลแอดเดรสสามารถถูกแบ่งเป็นตัวเลขที่ระบุฟิสิคัลเน็ตเวิร์ก (หรือซับเน็ต) และตัวเลขที่ระบุโฮสต์บนซับเน็ต ผู้ส่งจะเราต์ข้อความไปยังเน็ตเวิร์กแอดเดรสที่ถูกเผยแพร่ และระบบแบบโลคัลจะรับผิดชอบสำหรับการเราต์ข้อความไปยังซับเน็ตและโฮสต์ของมัน เมื่อตัดสินใจวิธีแบ่งโลคัลแอดเดรสเป็นซับเน็ตแอดเดรสและโฮสต์แอดเดรส คุณควรพิจารณาจำนวนของซับเน็ตและจำนวนของโฮสต์บนซับเน็ตนั้น

ในรูปต่อไปนี้ โลคัลแอดเดรสจะถูกแบ่งเป็น 12-บิตซับเน็ตแอดเดรส และ 12-บิตโฮสต์แอดเดรส

ที่อยู่เน็ตเวิร์ค (8 บิต)	ที่อยู่โลคัลโฮส (24 บิต)			
ที่อยู่เน็ตเวิร์ค	**ที่อยู่ซับเน็ต**		**ที่อยู่โฮส**	
01111101	00001101	0100	1001	00001111

หมายเหตุ: บิตที่อยู่ลำดับแรกๆ (หรือบิตแรก) จะเป็น 0 เสมอในคลาส A

รูปที่ 21. คลาส A แอดเดรสที่สอดคล้องกับซับเน็ตแอดเดรส

ภาพนี้จะแสดงโครงสร้างของคลาส A แอดเดรสทั่วไป 8 บิตแรกประกอบด้วยเน็ตเวิร์กแอดเดรส (เริ่มต้นด้วย 0 เสมอ) 24 บิตที่เหลือจะประกอบด้วยโลคัลโฮสต์แอดเดรสที่ซับเน็ตแอดเดรสจะใช้ 8 บิตแรกและโฮสต์แอดเดรสจะใช้ 8 บิตสุดท้าย

คุณมีความยืดหยุ่นเมื่อกำหนดซับเน็ตแอดเดรสและโฮสต์แอดเดรส บิตของโลคัลแอดเดรสสามารถถูกแบ่งตามความต้องการและแนวโน้มการขยายตัวขององค์กรและโครงสร้างเน็ตเวิร์กของมัน ข้อจำกัดคือ:

- `network_address` เป็นอินเตอร์เน็ตแอดเดรสสำหรับเน็ตเวิร์ก
- `subnet_address` เป็นฟิลด์ที่มีความกว้างคงที่สำหรับเน็ตเวิร์กนั้นๆ
- `host_address` เป็นฟิลด์ที่มีความกว้างอย่างน้อย 1 บิต

ถ้าความกว้างของฟิลด์ `subnet_address` เป็น 0 เน็ตเวิร์กจะไม่ถูกจัดเป็นซับเน็ต และการกำหนดแอดเดรสให้กับเน็ตเวิร์กจะถูกกระทำโดยใช้โดยใช้อินเตอร์เน็ตเน็ตเวิร์กแอดเดรส

บิตที่ระบุซับเน็ตจะถูกระบุโดยบิตมาสก์ และดังนั้น ไม่ถูกต้องการที่จะอยู่ติดกันในแอดเดรส ดังนั้น โดยทั่วไปมันจะถูกต้องการที่บิตของซับเน็ตต้องอยู่ติดกันและเป็นบิตที่อยู่ซ้ายสุดของโลคัลแอดเดรส

Subnet masks:

เมื่อโฮสต์ส่งข้อความไปยังปลายทาง ระบบต้องตรวจสอบว่าปลายทางอยู่บนเน็ตเวิร์กเดียวกันกับต้นทาง หรือถ้าปลายทางสามารถไปถึงได้โดยตรงผ่านหนึ่งในโลคัลอินเตอร์เฟส ระบบจะเปรียบเทียบแอดเดรสปลายทางกับแอดเดรสของโฮสต์โดยใช้ *subnet mask*

ถ้าปลายทางไม่ใช่โลคัล ระบบจะส่งข้อความไปยังเกตเวย์ เกตเวย์จะทำการเปรียบเทียบแบบเดียวกันเพื่อดูว่าแอดเดรสปลายทางอยู่บนเน็ตเวิร์กที่สามารถไปถึงได้แบบโลคัลหรือไม่

subnet mask จะบอกระบบว่า scheme ของการแบ่งซับเน็ตคืออะไร บิตมาสก์นี้ประกอบด้วยส่วนของเน็ตเวิร์กแอดเดรสและส่วนของซับเน็ตแอดเดรสของอินเตอร์เน็ตแอดเดรส

ที่อยู่เน็ตเวิร์ค (8 บิต)	ที่อยู่โลคัลโฮส (24 บิต)			
ที่อยู่เน็ตเวิร์ค	**ที่อยู่ซับเน็ต**		**ที่อยู่โฮส**	
01111101	00001101	0100	1001	00001111

คลาส A ที่สอดคล้องกับซับเน็ตแอดเดรส

ที่อยู่เน็ตเวิร์ค (8 บิต)	ที่อยู่โลคัลโฮส (24 บิต)			
ที่อยู่เน็ตเวิร์ค	ที่อยู่ซับเน็ต		ที่อยู่โฮส	
ซับเน็ตแอดเดรส			**ที่อยู่โฮส**	
01111101	00001101	0100	1001	00001111

คลาส A ที่สอดคล้องกับซับเน็ตแอดเดรส

รูปที่ 22. คลาส A คลาสที่สอดคล้องกับซับเน็ตแอดเดรส

ภาพนี้จะแสดงโครงสร้างของคลาส A แอดเดรสทั่วไป 8 บิตแรกประกอบด้วยเน็ตเวิร์กแอดเดรส (เริ่มต้นด้วย 0 เสมอ) 24 บิตที่เหลือจะประกอบด้วยโลคัลโฮสต์แอดเดรสที่ซับเน็ตแอดเดรสจะใช้ 8 บิตแรกและโฮสต์แอดเดรสจะใช้ 8 บิตสุดท้าย

ตัวอย่างเช่น subnet mask ของคลาส A แอดเดรสโดยมี scheme การแบ่งที่ถูกกำหนดด้านบนถูกแสดงในรูปนี้

subnet mask เป็นชุดของ 4 ไบต์ เหมือนกับอินเตอร์เน็ตแอดเดรส subnet mask ประกอบด้วยบิตสูง (1) ที่สอดคล้องกับตำแหน่ง บิตของแอดเดรสของเน็ตเวิร์กและเน็ตเวิร์กย่อย และบิตต่ำ (0) ที่สอดคล้อง กับตำแหน่งบิตของโฮสต์แอดเดรส subnet mask สำหรับแอดเดรสก่อนหน้านี้จะเหมือนดังรูปต่อไปนี้

ที่อยู่เน็ตเวิร์ค (8 บิต)	ที่อยู่โลคัลโฮส (24 บิต)			
ที่อยู่เน็ตเวิร์ค	**ซับเน็ตแอดเดรส**		**ที่อยู่โฮส**	
11111111	11111111	1111	0000	00000000

รูปที่ 23. ตัวอย่างของ subnet mask

ภาพนี้แสดงตัวอย่างของโครงสร้างของ subnet mask 8 บิตแรกจะประกอบด้วยเน็ตเวิร์กแอดเดรส 24 บิตที่เหลือจะประกอบด้วยโลคัลโฮสต์แอดเดรสที่ซับเน็ตแอดเดรสจะใช้ 8 บิตแรกและโฮสต์แอดเดรสจะใช้ 8 บิตสุดท้าย

การเปรียบเทียบแอดเดรส:

แอดเดรสปลายทางและโลคัลเน็ตเวิร์กแอดเดรสจะถูกเปรียบเทียบโดยทำตรรกะ AND และ exclusive OR บน subnet mask ของโฮสต์ต้นทาง

กระบวนการการเปรียบเทียบมีแนวทางดังข้างล่าง :

1. ทำตรรกะ AND ของแอดเดรสปลายทางและมาสก์ของโลคัลซับเน็ตแอดเดรส

2. ทำตรรกะ exclusive OR บนผลลัพธ์ของการดำเนินการก่อนหน้านี้ และโลคัลเน็ตแอดเดรสของโลคัลอินเตอร์เฟส ถ้าผลลัพธ์ที่เป็น 0 ทั้งหมดปลายทางถูกคาดว่าจะสามารถเข้าถึงได้โดยตรงผ่านหนึ่งในโลคัลอินเตอร์เฟส
3. ถ้าระบบแบบอิสระมีมากกว่าหนึ่งอินเตอร์เฟส (ดังนั้นจะมีมากกว่าหนึ่งอินเตอร์เน็ตแอดเดรส) กระบวนการการเปรียบเทียบจะถูกทำซ้ำสำหรับแต่ละโลคัลอินเตอร์เฟส

ตัวอย่างเช่น สมมุติว่ามี 2 โลคัลอินเตอร์เฟสถูกกำหนดสำหรับโฮสต์เน็ตเวิร์ก T125 อินเตอร์เน็ตแอดเดรสของมันและการแทนค่าไบนารีของแอดเดรสเหล่านั้นจะถูกแสดงในตังอย่างต่อไปนี้ :

```
CLASS A   73.1.5.2    =  01001001 00000001 00000101 00000010


CLASS B   145.21.6.3  =  10010001 00010101 00000110 00000011
```

subnet mask ที่สอดคล้องสำหรับโลคัลเน็ตเวิร์กอินเตอร์เฟสถูกแสดงในตัวอย่างต่อไปนี้ :

```
CLASS A   73.1.5.2    =  11111111 11111111 11100000 00000000


CLASS B   145.21.6.3  =  11111111 11111111 11111111 11000000
```

ถ้าเน็ตเวิร์กต้นทาง T125 ถูกต้องการให้ส่งข้อความไปยังเน็ตเวิร์กปลายทางที่มีโฮสต์แอดเดรส 114.16.23.8 (ถูกแทนในไบนารีเป็น : 01110010 00010000 00010111 00001000) ระบบจะตรวจสอบว่าปลายทางสามารถไปถึงได้ผ่านโลคัลอินเตอร์เฟสหรือไม่

หมายเหตุ: คีย์เวิร์ด **subnetmask** ต้องถูกตั้งในฐานข้อมูลคอนฟิกูเรชันของแต่ละโฮสต์เพื่อสนับสนุนซับเน็ต ดังนั้นความสามารถของซับเน็ตเวิร์กสามารถถูกใช้โฮสต์ทั้งหมดบนเน็ตเวิร์กต้องสนับสนุนมัน ตั้งค่า subnet mask ถาวร ในฐานข้อมูลการกำหนดคอนฟิกโดยใช้เมนู Network Interface Selection ใน SMIT subnet mask ยังสามารถถูกตั้งในระบบที่รันอยู่โดยใช้คำสั่ง **ifconfig** การใช้คำสั่ง **ifconfig** เพื่อตั้ง subnet mask จะไม่เป็นการเปลี่ยนแบบถาวร

บรอดคาสต์แอดเดรส

TCP/IP สามารถส่งข้อมูลที่ยังทุกโฮสต์บนโลคัลเน็ตเวิร์ก หรือทุกโฮสต์บนเน็ตเวิร์กที่ถูกเชื่อมต่อโดยตรงทั้งหมด การส่งนั้นเรียกว่า *ข้อความบรอดคาสต์*

ตัวอย่างเช่น **routed** routing daemon ใช้ข้อความบรอดคาสต์เพื่อเคียวรีและตอบสนองต่อการเราต์เคียวรี

สำหรับข้อมูลที่จะบรอดคาสต์ไปยังโฮสต์ทั้งหมดบนเน็ตเวิร์กที่เชื่อมต่อโดยตรง User Datagram Protocol (UDP) และ Internet Protocol (IP) ถูกใช้เพื่อส่งข้อมูล และแอดเดรสปลายทางของโฮสต์ในส่วนหัวของ IP มีทุกบิตถูกตั้งเป็น 1 สำหรับข้อมูลที่จะบรอดคาสต์ไปยังทุกโฮสต์บนเน็ตเวิร์กที่ระบุ ทุกบิตในส่วนของโลคัลแอดเดรสของ IP แอดเดรสถูกตั้งเป็น 0 ไม่มีคำสั่งของผู้ใช้ที่ใช้ความสามารถในการบรอดคาสต์ แม้ว่าคำสั่งนั้นหรือโปรแกรมจะถูกพัฒนา

บรอดคาสต์แอดเดรสสามารถถูกเปลี่ยนชั่วคราวโดยการเปลี่ยนพารามิเตอร์ *broadcast* ในคำสั่ง **ifconfig** เปลี่ยนแอดเดรสการกระจาย อย่างถาวรโดยใช้ SMIT fast path `smit chinet` การเปลี่ยนบรอดคาสต์แอดเดรสอาจมีประโยชน์ถ้าคุณต้องการที่จะให้เข้ากันได้กับซอฟต์แวร์เวอร์ชันเก่าที่ใช้บรอดคาสต์แอดเดรสที่แตกต่างออกไป ตัวอย่างเช่น โฮสต์ ID ที่ถูกตั้งเป็น 0 ทั้งหมด

loopback แอดเดรสแบบโลคัล

Internet Protocol จะกำหนดเน็ตเวิร์กแอดเดรสแอดเดรสพิเศษ 127.0.0.1 เป็น local แอดเดรสแบบโลคัล

โฮสต์ใช้ local แอดเดรสแบบโลคัลเพื่อส่งข้อความถึงตัวเอง loopback แอดเดรสแบบโลคัลถูกตั้งค่าโดยตัวจัดการคอนฟิกูเรชันระหว่างกระบวนการเริ่มต้นระบบ Loopback แบบโลคัลถูกนำไปใช้ในเคอร์นัลและสามารถถูกตั้งโดยคำสั่ง **ifconfig** Loopback ถูกใช้เมื่อระบบถูกสตาร์ท

การระบุชื่อ TCP/IP

แม้อินเตอร์เน็ตแอดเดรสแบบ 32 บิตจะทำให้เครื่องมือที่มี ประสิทธิภาพในการระบุต้นทางและปลายทางของดาตาแกรมที่ส่งอินเตอร์เน็ตเวิร์ก ด้วยชื่อที่ผู้ใช้ต้องการ และมีความหมาย จดจำได้ง่าย **Transmission Control Protocol/Internet Protocol (TCP/IP)** มีระบบการตั้งชื่อที่ สนับสนุนองค์กรทั้งแบบเครือข่ายแบบราบ และแบบลำดับชั้น

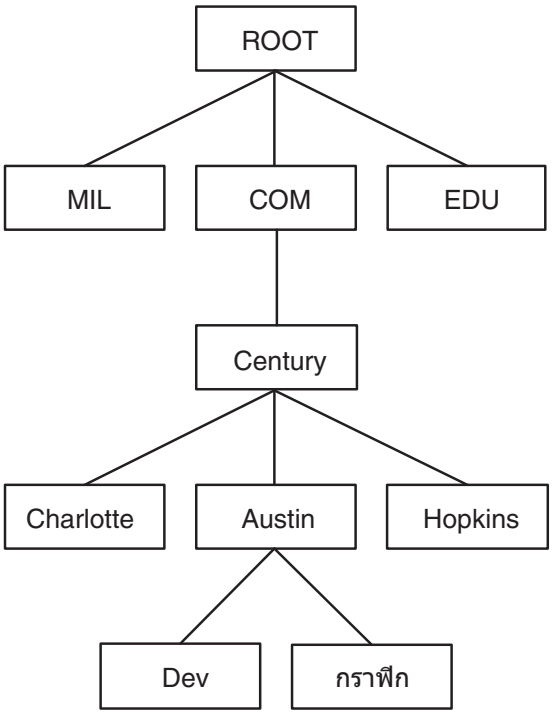
หลักการตั้งชื่อในเน็ตเวิร์กแบบราบทำได้ง่ายมาก ชื่อโฮสต์ประกอบด้วยชุดอักขระ ชุดเดียว และโดยทั่วไปจะถูกดูแลจัดการบนโลคัล ในเน็ตเวิร์ก **TCP/IP** แบบราบ แต่ละเครื่องบนเน็ตเวิร์กจะมีไฟล์ (`/etc/hosts`) ที่มี ข้อมูลการแม็พ ชื่อ-กับ-อินเตอร์เน็ตแอดเดรสสำหรับทุกโฮสต์บนเน็ตเวิร์ก ภาระด้านการดูแลจัดการเก็บรักษาไฟล์การตั้งชื่อของแต่ละเครื่องที่เติบโตขึ้นในขณะนี้ เป็นการเติบโตของเน็ตเวิร์ก **TCP/IP** เมื่อเน็ตเวิร์ก **TCP/IP** เริ่มมีขนาดใหญ่ มากๆ อย่างบนอินเตอร์เน็ต การตั้งชื่อจะถูกแบ่งออกเป็นลำดับชั้น โดยทั่วไป การแบ่ง จะแบ่งตามองค์กรแบบเครือข่าย ใน **TCP/IP** การตั้งชื่อ แบบลำดับชั้นจะเรียกว่า *domain name system* (DNS) และใช้โปรโตคอล DOMAIN โปรโตคอล DOMAIN ถูกนำไปปฏิบัติใช้โดย **named** daemon ใน **TCP/IP**

เนื่องจากในการตั้งชื่อสำหรับเน็ตเวิร์กแบบราบ ลำดับชั้นโดเมนเนมมีขึ้นเพื่อให้ การกำหนดค่าชื่อสัญลักษณ์ไปยังเน็ตเวิร์กและโฮสต์มีความหมาย และผู้ใช้จดจำได้ง่าย อย่างไรก็ตาม แทนที่จะให้แต่ละเครื่องบนเน็ตเวิร์ก เก็บรักษาไฟล์ที่มีการแม็พชื่อ-กับ-แอดเดรสสำหรับโฮสต์อื่นทั้งหมด บนเน็ตเวิร์กเอง ก็จะมีอย่างน้อยหนึ่งโฮสต์ถูกเลือกเพื่อทำหน้าที่เป็น *เนมเซิร์ฟเวอร์* แทน เซิร์ฟเวอร์ ชื่อจะแปล (แก้ไข) ชื่อสัญลักษณ์ที่กำหนดไปยังเน็ตเวิร์ก และโฮสต์ให้เป็นอินเตอร์เน็ตแอดเดรสที่มีประสิทธิภาพที่ใช้โดยเครื่องใดๆ เนมเซิร์ฟเวอร์ มีข้อมูลครบถ้วนเกี่ยวกับส่วนของโดเมน ที่ถูกอ้างอิงเป็น *โซน* และมี *สิทธิ* ในโซนของตน

สิทธิการตั้งชื่อ

ในเน็ตเวิร์กแบบราบ โฮสต์ทั้งหมดในเน็ตเวิร์กจะถูกจัดการ โดยตัวควบคุมสิทธิกลาง เน็ตเวิร์กรูปแบบนี้จำเป็นที่โฮสต์ทั้งหมดในเน็ตเวิร์กต้องมีชื่อโฮสต์เฉพาะ ในเน็ตเวิร์กขนาดใหญ่ ข้อกำหนดนี้ จะสร้างภาระในการดูแลอย่างมากต่อตัวควบคุมสิทธิกลาง

ในเน็ตเวิร์กโดเมน กลุ่มของโฮสต์จะถูกดูแลจัดการแยกต่างหาก ภายในลำดับชั้นแบบโครงสร้างของต้นไม้ของโดเมนและโดเมนย่อย ในกรณี นี้ ชื่อโฮสต์ต้องเป็นชื่อเฉพาะเท่านั้นภายในโดเมนโลคัล และ มีเพียง *โดเมน root* เท่านั้นที่ถูกดูแลจัดการโดยตัวควบคุมสิทธิกลาง โครงสร้างนี้อนุญาตให้โดเมนย่อยถูกดูแลจัดการแบบโลคัล และช่วยลด ภาระของตัวจัดการสิทธิกลาง ตัวอย่างเช่น โดเมน root ของอินเตอร์เน็ตจะประกอบด้วยโดเมนต่างๆ เช่น `com` (องค์กร เชิงพานิชย์), `edu` (องค์กรการศึกษา), `gov` (หน่วยงาน รัฐ) และ `mil` (กลุ่มทางทหาร) โดเมนระดับบนสุด ใหม่นี้จะสามารถถูกเพิ่มได้โดยตัวควบคุมสิทธิกลางเท่านั้น การตั้งชื่อที่ ระดับที่สองจะถูกมอบสิทธิให้แก่เอเจนต์ที่กำหนดภายในโดเมน ที่เกี่ยวข้อง ตัวอย่างเช่น ในภาพต่อไปนี้ `com` มีสิทธิการตั้งชื่อสำหรับโดเมนย่อยที่เป็นองค์กรเชิงพานิชย์ทั้งหมดที่อยู่ ภายใต้ เช่นเดียวกับ การตั้งชื่อในระดับที่สาม (และอื่นๆ) จะถูกมอบให้แก่ เอเจนต์ภายในระดับนั้น ตัวอย่างเช่น ในภาพ โครงสร้างโดเมนของ อินเตอร์เน็ต นั้น Century มีสิทธิการตั้งชื่อสำหรับโดเมนย่อยของตนคือ Austin, Hopkins และ Charlotte



รูปที่ 24. โครงสร้างโดเมนของอินเตอร์เน็ต

ภาพนี้แสดงโครงสร้างแบบลำดับขั้นของอินเตอร์เน็ต โดยเริ่มต้นที่ด้านบนสุดที่มี root และสาขาแยกย่อยไปยังระดับถัดไปที่มีโดเมน mil, com และ edu ด้านล่างของโดเมน com เป็น อีกระดับที่มี Charlotte, Austin และ Hopkins ด้านล่างของ Austin คือ Dev และ Graphics

โดเมนย่อย Austin ของ Century ก็อาจแบ่งออกเป็นหลายโซน เช่น Dev และ Graphics เช่นกัน ในกรณีนี้โซน `austin.century.com` จะมี ข้อมูลทั้งหมดที่มีในโดเมน `austin.century.com` ยกเว้นที่ถูกมอบสิทธิให้แก่ Dev และ Graphics โซน `dev.century.com` จะ มีเฉพาะข้อมูลที่มอบสิทธิให้แก่ Dev เท่านั้น โดยจะไม่ทราบอะไรเลยเกี่ยวกับ Graphics เป็นต้น โซน `austin.century.com` (ตรง ข้ามกับโดเมนของชื่อเดียวกัน) จะมีเฉพาะข้อมูลที่ ไม่ถูกมอบสิทธิให้แก่โซนอื่นๆ

ระเบียบการตั้งชื่อ

ในระบบโดเมนเนมแบบลำดับชั้น ชื่อจะประกอบด้วยลำดับ ของชื่อย่อยที่ไม่คำนึงถึงขนาดตัวพิมพ์คั่นด้วยจุดโดยไม่มีช่องว่างอยู่ภายในชื่อ

โปรโตคอล DOMAIN ระบุว่าชื่อโดเมนบนโลคัลจะต้องมีขนาดน้อยกว่า 64 อักขระ และชื่อโฮสต์ต้องมีความยาวน้อยกว่า 32 อักขระ ชื่อโฮสต์นี้ถูกกำหนดไว้เป็นอันดับแรก ซึ่งตามด้วยจุด (.) ชุดของโดเมนเนม แบบโลคัลจะคั่นด้วยจุด และสุดท้ายจึงเป็นโดเมน root ชื่อโดเมน ที่ระบุแบบสมบูรณ์สำหรับโฮสต์ โดยรวมจุด ต้องมีความยาวน้อยกว่า 255 อักขระ และมีรูปแบบต่อไปนี้:

```
host.subdomain1.[subdomain2 . . . subdomain].rootdomain
```

เนื่องจากชื่อโฮสต์ต้องเป็นชื่อเฉพาะภายในโดเมน คุณสามารถ ใช้ชื่อย่อเมื่อส่งข้อความไปยังโฮสต์ภายในโดเมนเดียวกันได้ ตัวอย่างเช่น แทนการส่งข้อความไปยัง `smith.eng.lsu.edu` โฮสต์ในโดเมน `eng` สามารถส่งข้อความไปยัง `smith` แทนได้ นอกจากนั้น แต่ละโฮสต์ยังสามารถมีได้หลาย alias ที่โฮสต์อื่นๆ สามารถใช้ เมื่อส่งข้อความ

การตั้งชื่อโฮสต์บนเน็ตเวิร์กของคุณ

วัตถุประสงค์ของการใช้ชื่อสำหรับโฮสต์ก็คือเพื่อให้มีวิธีที่รวดเร็ว ง่ายดาย และไม่ทำให้สับสน ในการอ้างอิงคอมพิวเตอร์ในเน็ตเวิร์กของคุณ ผู้ดูแลระบบ อินเตอร์เน็ตพบว่ามีทั้งตัวเลือกที่ดี และไม่ดี สำหรับชื่อโฮสต์ ข้อเสนอแนะเหล่านี้มุ่งหวังที่จะช่วยคุณหลีกเลี่ยงหลุมพรางทั่วไป ในการเลือกชื่อโฮสต์

ต่อไปนี้เป็นข้อเสนอแนะบางประการสำหรับการเลือกชื่อโฮสต์ที่ชัดเจน จดจำได้ง่าย:

- คำที่ไม่ค่อยใช้ เช่น `sphinx` หรือ `eclipse`
- ชื่อธีม เช่น สี องค์ประกอบ (ตัวอย่างเช่น `helium`, `argon` หรือ `zinc`) ดอกไม้ ปลา และอื่นๆ
- คำจริง (ตรงข้ามกับสตริงอักขระแบบสุ่ม)

ต่อไปนี้คือตัวอย่างการเลือกที่แย่ โดยทั่วไป เหล่านี้ คือชื่อที่แย่ เนื่องจากจดจำได้ยาก หรือทำให้เกิดความสับสน (ไม่ว่า จะต่อมนุษย์ หรือคอมพิวเตอร์):

- คำที่ใช้อยู่แล้วโดยทั่วไป ตัวอย่างเช่น `up`, `down` หรือ `crash`
- ชื่อที่มีแต่ตัวเลขเท่านั้น
- ชื่อที่มีเครื่องหมายวรรคตอน
- ชื่อที่ต้องยึดตามขนาดตัวพิมพ์ เช่น `Orange` และ `orange`
- ชื่อ หรือตัวขึ้นต้นของผู้ใช้หลักของระบบ
- ชื่อที่ยาวมากกว่า 8 อักขระ
- การสะกดแบบผิดปกติ หรือตั้งใจให้ไม่ถูกต้อง เช่น `czek` ซึ่งสามารถทำให้เกิดความสับสนกับคำว่า "check" หรือ "czech"
- ชื่อที่เป็น หรือคล้ายกับชื่อโดเมน เช่น `yale.edu`

เนมเซิร์ฟเวอร์

ใน flat name space ชื่อทั้งหมดต้องถูกเก็บในไฟล์ `/etc/hosts` บนแต่ละโฮสต์บนเน็ตเวิร์ก ถ้าเน็ตเวิร์กมีขนาดใหญ่มาก ปัญหานี้อาจ กลายเป็นภาระในรีซอร์สของแต่ละเครื่อง ในเน็ตเวิร์ก แบบลำดับชั้น โฮสต์ที่แน่นอนจะถูกกำหนดเป็น *เนมเซิร์ฟเวอร์* เพื่อระบุชื่อ ให้เป็นอินเตอร์เน็ตแอดเดรสสำหรับโฮสต์อื่นๆ

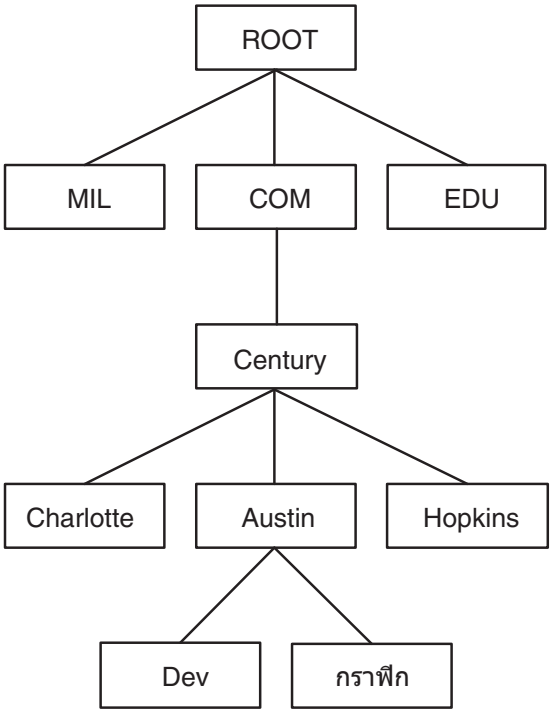
ซึ่งมีข้อดีกว่าเนมสเปซแบบ flat สองข้อ โดยเก็บ รีซอร์สของแต่ละโฮสต์บนเน็ตเวิร์กมิให้ต้องยุ่งยากในการระบุ ชื่อ และช่วยให้ผู้มีหน้าที่ดูแลระบบไม่จำเป็นต้อง ดูแลไฟล์การระบุชื่อบนแต่ละเครื่องบนเน็ตเวิร์ก ชุดของชื่อที่ดูแลโดยเนมเซิร์ฟเวอร์เดียวจะเรียกว่า *โซน สิทธิ*

หมายเหตุ: แม้ว่าเครื่องโฮสต์ที่ทำหน้าที่ในการระบุชื่อ สำหรับโซนสิทธิโดยทั่วไปจะถูกอ้างอิงเป็นโฮสต์ *เซิร์ฟเวอร์ชื่อ* การควบคุมกระบวนการทำงาน **named** daemon เป็นกระบวนการเนมเซิร์ฟเวอร์ที่แท้จริง

เมื่อต้องการลดกิจกรรมเน็ตเวิร์กที่ไม่จำเป็นออก เนมเซิร์ฟเวอร์ทั้งหมด *แคช* (เก็บ ในช่วงระยะเวลาหนึ่ง) การแม็พชื่อ-กับ-แอดเดรส เมื่อไคลเอ็นต์ขอให้ เซิร์ฟเวอร์ระบุชื่อ เซิร์ฟเวอร์จะตรวจสอบก่อนว่า ชื่อได้รับถูกระบุเมื่อเร็วๆ นี้หรือไม่ เนื่องจากชื่อโดเมนและชื่อโฮสต์ มีการเปลี่ยนแปลง แต่ละรายการยังคงอยู่ในแคชที่มีระยะเวลาที่ระบุ โดย TTL ของเร็กคอร์ดเหลือจำกัด ด้วยวิธีนี้ สิทธิ สามารถระบุระยะเวลาที่คาดว่าการระบุจะยังคงถูกต้อง

ภายในระบบอัตโนมัติใดๆ สามารถมีได้หลายเนมเซิร์ฟเวอร์ โดยปกติ เนมเซิร์ฟเวอร์ถูกจัดลำดับขั้น และสอดคล้องกับ ระบบองค์กรของเน็ตเวิร์ก การอ้างอิงภาพ "โครงสร้างโดเมนของ อินเตอร์เน็ต" แต่ละโดเมนอาจมีเนมเซิร์ฟเวอร์ที่รับผิดชอบ โด

เมนย่อยทั้งหมดภายในโดเมน แต่ละเนมเซิร์ฟเวอร์โดเมนย่อยจะสื่อสาร กับเนมเซิร์ฟเวอร์ของโดเมนที่อยู่เหนือขึ้นไป (เรียกว่าเนมเซิร์ฟเวอร์ *พาเรนต์*) รวมถึงกับเนมเซิร์ฟเวอร์ของโดเมนย่อยอื่นๆ



รูปที่ 25. โครงสร้างโดเมนของอินเตอร์เน็ต

ภาพนี้แสดงโครงสร้างแบบลำดับขั้นของอินเตอร์เน็ต โดยเริ่มต้นที่ด้านบนสุดที่มี root และสาขาแยกย่อยไปยังระดับถัดไปที่มีโดเมน mil, com และ edu ด้านล่างของโดเมน com เป็น อีกระดับที่มี Charlotte, Austin และ Hopkins ด้านล่างของ Austin คือ Dev และ Graphics

ตัวอย่างเช่น ในภาพ "โครงสร้างโดเมนของอินเตอร์เน็ต" Austin, Hopkins และ Charlotte คือโดเมนย่อยทั้งหมดของโดเมน Century ถ้าลำดับชั้นแผนผังเป็นไปตามการออกแบบเน็ตเวิร์ก เนมเซิร์ฟเวอร์ Austin จะสื่อสารกับเนมเซิร์ฟเวอร์ของ Charlotte และ Hopkins รวมทั้งกับเนมเซิร์ฟเวอร์พาเรนต์ Century เนมเซิร์ฟเวอร์ Austin ยังสื่อสารกับเนมเซิร์ฟเวอร์ที่รับผิดชอบในส่วนโดเมนย่อย

เนมเซิร์ฟเวอร์มีหลายชนิด:

ไอเท็ม	คำอธิบาย
เนมเซิร์ฟเวอร์มาสเตอร์ **เนมเซิร์ฟเวอร์สเลฟ**	โหลดข้อมูลจากไฟล์หรือดิสก์ และสามารถมอบสิทธิให้แก่เซิร์ฟเวอร์อื่นในโดเมน รับข้อมูลในตอนเริ่มทำงานระบบสำหรับโซน ของสิทธิที่กำหนดจากเนมเซิร์ฟเวอร์มาสเตอร์ จากนั้นขอ เซิร์ฟเวอร์มาสเตอร์อัพเดตข้อมูลเป็นระยะ เมื่อค่ารีเฟรช หมดอายุใน start of authority (SOA) Resource Record บนเนมเซิร์ฟเวอร์สเลฟ หรือเมื่อรับข้อความ Notify จาก เนมเซิร์ฟเวอร์มาสเตอร์ ตัวสเลฟจะโหลดฐานข้อมูลจากมาสเตอร์ใหม่ ถ้าหมายเลขลำดับของฐานข้อมูลบนมาสเตอร์มีค่ามากกว่า หมายเลขลำดับในฐานข้อมูลปัจจุบันบนสเลฟ ถ้ามีความจำเป็น ต้องบังคับใช้การถ่ายโอนโซนใหม่จากมาสเตอร์ ทำได้ง่ายๆ เพียงเอา ฐานข้อมูลสเลฟที่มีอยู่ออก และรีเฟรช **named** daemon บนเนมเซิร์ฟเวอร์สเลฟ

ไอเท็ม	คำอธิบาย
เนมเซิร์ฟเวอร์ Stub	แม้ว่าวิธีการทำสำเนาฐานข้อมูลจะคล้ายกับวิธี ของเนมเซิร์ฟเวอร์สเลฟ เนมเซิร์ฟเวอร์ stub จะทำเพียงทำสำเนาเร็กคอร์ด เนมเซิร์ฟเวอร์ของฐานข้อมูลมาสเตอร์แทนทั้งฐานข้อมูล
เซิร์ฟเวอร์แนะนำ	ระบุเนมเซิร์ฟเวอร์ที่ตอบกลับเมื่อให้คำแนะนำที่ เซิร์ฟเวอร์ได้สร้างจากการสอบถามก่อนหน้าไปยังเนมเซิร์ฟเวอร์ เนมเซิร์ฟเวอร์ แนะนำตอบกลับการสอบถามนั้นโดยถามไปยังเซิร์ฟเวอร์อื่นที่มี สิทธิในการให้ข้อมูลที่จำเป็น ถ้าเนมเซิร์ฟเวอร์แนะนำ ไม่มีการแม็พชื่อ-กับ-แอดเดรสในแคช
ตัวส่งต่อ หรือ เซิร์ฟเวอร์ไคลเอ็นต์	ส่งต่อการสอบถามที่สามารถดำเนินการบนโลคัลได้ เพื่ออแก้ไขปัญหารายการ ของเซิร์ฟเวอร์การส่งต่อ เซิร์ฟเวอร์ที่ทำการส่งต่อเท่านั้น (ตัวส่งต่อที่มี ข้อมูลและส่งไปยังไคลเอ็นต์อื่นๆ แต่ไม่ใช่เซิร์ฟเวอร์ แท้จริง) ไม่มีการโต้ตอบกับเนมเซิร์ฟเวอร์มาสเตอร์สำหรับโดเมน root และโดเมนอื่นๆ เคียวรีเกี่ยวกับเซิร์ฟเวอร์การส่งต่อ จะเป็นแบบเรียกซ้ำ โดยสามารถมีอย่างน้อยหนึ่งเซิร์ฟเวอร์การส่งต่อ ซึ่ง จะพยายามเปิดให้ใช้ได้จนกระทั่ง รายการครบหมด การกำหนดค่าไคลเอ็นต์ และตัวส่งต่อโดยปกติจะถูกใช้เมื่อคุณไม่ต้องการให้เซิร์ฟเวอร์ทั้งหมด ที่ไซต์ที่กำหนดโต้ตอบกับไซต์อินเตอร์เน็ตที่เหลือ หรือเมื่อคุณต้องการสร้างแคชขนาดใหญ่บนเนมเซิร์ฟเวอร์ตามจำนวน ที่เลือก
รีโมตเซิร์ฟเวอร์	รันโปรแกรมเน็ตเวิร์กทั้งหมดที่ใช้เนมเซิร์ฟเวอร์โดยไม่มี กระบวนการเนมเซิร์ฟเวอร์กำลังรันอยู่บนโฮสต์โลคัล เคียวรีทั้งหมด ให้บริการโดยเนมเซิร์ฟเวอร์ที่กำลังทำงานอยู่บนเครื่องอื่นบน เน็ตเวิร์ก

หนึ่งโฮสต์เนมเซิร์ฟเวอร์สามารถดำเนินการโดยมีความสามารถแตกต่างกันสำหรับโซน ของสิทธิที่แตกต่างกัน ตัวอย่างเช่น โฮสต์เนมเซิร์ฟเวอร์เดียวสามารถ เป็นเนมเซิร์ฟเวอร์มาสเตอร์สำหรับโซนหนึ่ง และเป็นเนมเซิร์ฟเวอร์สเลฟสำหรับอีก โซนหนึ่ง

การระบุชื่อ

กระบวนการเพื่อให้ได้อินเตอร์เน็ตแอดเดรสจากชื่อโฮสต์จะ รู้จักเป็นการระบุชื่อ และดำเนินการโดยรูทีนย่อย **gethostbyname**

กระบวนการของการแปลอินเตอร์เน็ตแอดเดรสให้เป็น ชื่อโฮสต์จะ รู้จักเป็นการระบุชื่อสำรอง และดำเนินการโดยรูทีนย่อย **gethostbyaddr** รูทีนเหล่านี้เป็นตัวเข้าถึงไลบรารีของรูทีนการแปล ชื่อที่สำคัญที่รู้จักในชื่อ *ตัวแก้ไข*

รูทีนตัวแก้ไขบนโฮสต์ที่กำลังรัน **TCP/IP** โดยปกติพยายามแก้ไข ชื่อโดยใช้แหล่งที่มาต่อไปนี้:

1. BIND/DNS (named)
2. Network Information Service (NIS)
3. ไฟล์ `/etc/hosts` โลคัล

เมื่อต้องการแก้ไขชื่อในโดเมนเน็ตเวิร์ก อันดับแรกรูทีนตัวแก้ไขจะเคียวรี ฐานข้อมูลโดเมนเนมเซิร์ฟเวอร์ ซึ่งอาจเป็นค่าโลคัล ถ้าโฮสต์เป็นโดเมนเนม เซิร์ฟเวอร์ หรือบนโฮสต์อื่น เนมเซิร์ฟเวอร์จะแปลโดเมนเนมให้เป็น อินเตอร์เน็ตแอดเดรส กลุ่มของชื่อซึ่งเนมเซิร์ฟเวอร์รับผิดชอบ คือส่วนของโซนการกำหนดสิทธิ ถ้ารูทีนตัวแก้ไขกำลังใช้เนมเซิร์ฟเวอร์รีโมต รูทีนจะใช้โปรโตคอลโดเมนเนม (DOMAIN) เพื่อเคียวรีการแม็พ เมื่อต้องการแก้ไขชื่อในเน็ตเวิร์กแนวราบ รูทีนตัวแก้ไขจะตรวจหารายการในไฟล์ `/etc/hosts` โลคัล เมื่อใช้ NIS แล้ว ไฟล์ `/etc/hosts` บนเซิร์ฟเวอร์มาสเตอร์จะถูกตรวจสอบ

โดยค่าดีฟอลต์ รูทีนตัวแก้ไขจะพยายามแก้ไขชื่อโดยใช้รีซอร์ส ข้างต้น BIND/DNS จะถูกใช้เป็นอันดับแรก ถ้าไม่มีไฟล์ `/etc/resolv.conf` หรือถ้า BIND/DNS ไม่พบชื่อ NIS จะถูกเคียวรีถ้า กำลังรันอยู่ NIS ได้รับสิทธิเหนือ `/etc/hosts` โลคัล ดังนั้นการค้นหาจะสิ้นสุดที่นี่ถ้า NIS กำลังรัน ถ้า NIS ไม่ได้รันอยู่ ไฟล์ `/etc/hosts` จะถูกค้นหา ถ้าไม่มีบริการใดเหล่านี้ ที่สามารถหาชื่อพบ รูทีนตัวแก้ไขจะส่งกลับค่า HOST_NOT_FOUND ถ้าบริการทั้งหมดไม่สามารถใช้ได้ รูทีนตัวแก้ไขจะส่งกลับ ค่า SERVICE_UNAVAILABLE

ลำดับดีฟอลต์ที่อธิบายข้างต้นสามารถแทนที่ได้ด้วยการสร้างไฟล์คอนฟิกูเรชัน /etc/irs.conf และระบุลำดับที่ต้องการ รวมทั้ง การจัดลำดับดีฟอลต์ และ /etc/irs.conf ทั้งสอง สามารถแทนที่ด้วยตัวแปรสภาวะแวดล้อม **NSORDER** ถ้าไฟล์ /etc/irs.conf หรือตัวแปรสภาวะแวดล้อม **NSORDER** อย่างใดอย่างหนึ่งถูกกำหนดค่า ดังนั้นจะมีอย่างน้อยหนึ่งค่าที่ต้องระบุพร้อม กับอ็อพชัน

เมื่อต้องการระบุลำดับโฮสต์ด้วยไฟล์ /etc/irs.conf :

```
hosts value [ continue ]
```

ลำดับถูกระบุด้วยแต่ละเมธอดที่ระบุในบรรทัดเอง *value* คือหนึ่งในเมธอดที่แสดงรายการ และคีย์เวิร์ด **continue** ระบุว่ายังมีเมธอดตัวแก้ไขอื่นตามมาในบรรทัดถัดไป

เมื่อต้องการระบุลำดับโฮสต์ด้วยตัวแปรสภาวะแวดล้อม **NSORDER** :

```
NSORDER=value,value,value
```

ให้ระบุลำดับ บนบรรทัดเดียวกันกับค่าโดยคั่นด้วยเครื่องหมายจุลภาค อนุญาตให้มีพื้นที่ว่างสีขาวระหว่าง เครื่องหมายจุลภาค และเครื่องหมายเท่ากับได้

ตัวอย่างเช่น ถ้าเครือข่ายโลคัลมีการจัดระเบียบ เป็นเครือข่ายระนาบเดียว (flat) ผลคือต้องการไฟล์ /etc/hosts เพียง อย่างเดียวเท่านั้น สำหรับตัวอย่างที่กำหนดนี้ ไฟล์ /etc/irs.conf จะมีบรรทัดต่อไปนี้:

```
hosts local
```

หรือ ตัวแปรสภาวะแวดล้อม **NSORDER** สามารถตั้งค่าเป็น:

```
NSORDER=local
```

ถ้าเครือข่ายโลคัลเป็นเครือข่ายโดเมนที่ใช้เนมเซิร์ฟเวอร์สำหรับการแก้ไข ชื่อและไฟล์ /etc/hosts สำหรับแบ็คอัพ ควรระบุการบริการ ทั้งสองอย่าง สำหรับตัวอย่างที่กำหนดนี้ ไฟล์ /etc/irs.conf จะมีบรรทัดต่อไปนี้:

```
hosts dns continue
hosts local
```

ตัวแปรสภาพแวดล้อม **NSORDER** ถูกตั้งค่าเป็น:

```
NSORDER=bind,local
```

หมายเหตุ: ค่าที่แสดงรายการต้องเป็นตัวพิมพ์เล็ก

เมื่อทำตามลำดับตัวแก้ไขที่กำหนด หรือค่าดีฟอลต์ใดๆ อัลกอริทึมการค้นหา จะดำเนินการต่อจากตัวแก้ไขหนึ่งไปยังตัวถัดไป ต่อเมื่อ:

- เซอร์วิสปัจจุบันไม่ได้รันอยู่ ดังนั้น จึงไม่สามารถใช้งาน
- เซอร์วิสปัจจุบันไม่พบชื่อและไม่ได้รับอนุญาต

ถ้าไม่มีไฟล์ /etc/resolv.conf ดังนั้น BIND/DNS จะถูกพิจารณาว่าไม่ถูกตั้งค่า หรือไม่ได้รันอยู่ จึงไม่สามารถใช้งานได้ ถ้ารูทีนย่อย **getdomainname** และ **yp_bind** ล้มเหลว ดังนั้น บริการ NIS ถูกพิจารณาว่าไม่ถูกตั้งค่า หรือไม่ได้รันอยู่ จึง ไม่สามารถใช้งานได้ ถ้าไม่สามารถเปิดไฟล์ /etc/hosts ดังนั้นจะไม่สามารถค้นหาแบบโลคัล ดังนั้นไฟล์และบริการ จะไม่สามารถใช้งานได้

เมื่อเซอร์วิสถูกแสดงรายการเป็น *authoritative* หมายความว่าเซอร์วิสนี้ เป็นผู้เชี่ยวชาญสำหรับตัวที่สืบต่อ และมีชื่อและแอดเดรสที่เกี่ยวข้อง รูทีนตัวแก้ไขจะไม่ลองใช้เซอร์วิสตัวที่สืบต่อ เนื่องจากตัวที่สืบต่ออาจ มีเฉพาะเซ็ตย่อยของข้อมูลในบริการที่มีสิทธิ การระบุ ชื่อสิ้นสุดที่เซอร์วิสที่แสดงรายการว่ามีสิทธิ แม้ว่าจะไม่พบ ชื่อ (ไม่ว่ากรณีใด รูทีนตัวแก้ไขจะส่งกลับ HOST_NOT_FOUND) ถ้าเซอร์วิสที่มีสิทธิไม่สามารถใช้งานได้ เซอร์วิสถัดไปที่ระบุ จะถูกเคียวรี

ต้นทางที่มีสิทธิจะถูกระบุด้วยสตริง =auth หลัง ค่า โดยสามารถพิมพ์คำทั้งคำ authoritative หรือใช้เฉพาะสตริง auth ตัวอย่างเช่น ถ้าตัวแปรสภาวะแวดล้อม **NSORDER** มีค่าต่อไปนี้:

```
hosts = nis=auth,dns,local
```

การค้นหาสิ้นสุดหลังจากเคียวรี NIS (ถ้า NIS กำลังรัน) ไม่ว่า จะพบชื่อหรือไม่ ถ้า NIS ไม่ได้รันอยู่ ต้นทางถัดไปจะถูก เคียวรี ซึ่งคือ DNS

เนมเซิร์ฟเวอร์ **TCP/IP** ใช้การแคชเพื่อลดค่าใช้จ่ายในการค้นหา ชื่อโฮสต์บนเน็ตเวิร์กรีโมต แทนที่จะค้นหาชื่อโฮสต์ ในแต่ละครั้งที่มีการร้องขอ เนมเซิร์ฟเวอร์จะค้นหาที่แคชก่อนเพื่อดูว่า ชื่อโฮสต์เพิ่มถูกระบุชื่อเมื่อเร็วๆ นี้หรือไม่ เนื่องจากโดเมนและชื่อโฮสต์มีการเปลี่ยนแปลง แต่ละรายการจะยังคงอยู่ในแคชในช่วงเวลา จำกัด ตามที่ระบุโดยค่า time-to-live (TTL) ของเร็กคอร์ด วิธีนี้ เนมเซิร์ฟเวอร์สามารถระบุระยะเวลาที่คาดว่าการตอบกลับจะถูกพิจารณาว่า มีสิทธิ

ชื่อโฮสต์ที่น่าจะขัดแย้งกันระหว่างชื่อเซิร์ฟเวอร์กับ sendmail:

ในสภาพแวดล้อม DNS ชื่อโฮสต์ถูกตั้งค่าโดยใช้คำสั่ง **hostname** จากบรรทัดรับคำสั่ง หรือในรูปแบบไฟล์ rc.net ต้องเป็นชื่ออย่างเป็นทางการของโฮสต์ที่คืนค่าโดยเนมเซิร์ฟเวอร์

ตามปกติแล้ว ชื่อนี้เป็นชื่อโดเมนแบบสมบูรณ์ของโฮสต์ในรูปแบบ:

```
host.subdomain.subdomain.rootdomain
```

หมายเหตุ: รูทีนการแก้ไขต้องการดีฟอลต์โดเมนเพื่อตั้งค่า ถ้าดีฟอลต์โดเมน ไม่ถูกตั้งค่าไว้ในคำสั่ง **hostname** แล้ว ต้องตั้งค่าไว้ในไฟล์ /etc/resolv.conf

ถ้าชื่อโฮสต์ไม่ถูกตั้งค่าเป็นชื่อโดเมนแบบสมบูรณ์ และถ้าระบบถูกตั้งค่าให้ใช้เนมเซิร์ฟเวอร์โดเมนร่วมกับโปรแกรม **sendmail** ไฟล์คอนฟิกูเรชัน **sendmail** (/etc/sendmail.cf) ต้องถูกแก้ไขเพื่อแสดงชื่อโฮสต์ที่เป็นทางการ นอกจากนี้ แมโครของชื่อโดเมน ในไฟล์คอนฟิกูเรชันต้องตั้งค่าสำหรับโปรแกรม **sendmail** เพื่อให้ทำงานได้อย่างถูกต้อง

หมายเหตุ: โดเมนที่ระบุในไฟล์ /etc/sendmail.cf มาก่อนชุดโดเมนโดยคำสั่ง **hostname** สำหรับฟังก์ชัน **sendmail** ทั้งหมด

ชื่อโดเมนที่น่าจะขัดแย้งกันระหว่างชื่อเซิร์ฟเวอร์กับ sendmail:

ชื่อโดเมนโลคัลและเนมเซิร์ฟเวอร์โดเมนถูกระบุในไฟล์ที่แตกต่างกัน ขึ้นอยู่กับโฮสต์เป็นเนมเซิร์ฟเวอร์ DOMAIN

สำหรับโฮสต์ที่อยู่ในเครือข่าย DOMAIN แต่ไม่ใช่เนมเซิร์ฟเวอร์ ชื่อโดเมนโลคัล และเนมเซิร์ฟเวอร์โดเมนถูกระบุในไฟล์ /etc/resolv.conf ในโฮสต์ของเนมเซิร์ฟเวอร์ DOMAIN, โดเมนโลคัลและเนมเซิร์ฟเวอร์อื่นถูกกำหนดใน ไฟล์ที่อ่านโดย **named** daemon เมื่อเริ่มทำงาน

Reverse Address Resolution Protocol

Reverse Address Resolution Protocol (RARP) แปล แอดเดรสฮาร์ดแวร์เฉพาะให้เป็นอินเตอร์เน็ตแอดเดรสบนอะแด็ปเตอร์ Ethernet local area network (LAN) (โปรโตคอล Ethernet เท่านั้น)

โปรโตคอล Standard Ethernet ได้รับการสนับสนุนโดยมีข้อจำกัดต่อไปนี้:

- เซิร์ฟเวอร์ตอบกลับการร้องขอ **RARP** เท่านั้น
- เซิร์ฟเวอร์ใช้รายการตาราง **ARP** ถาวรเท่านั้น
- เซิร์ฟเวอร์ไม่ใช้รายการตาราง **ARP** ไดนามิก
- เซิร์ฟเวอร์ไม่ตอบกลับตนเองโดยอัตโนมัติ

ผู้ดูแลระบบต้องสร้างและดูแลรักษรตารางรายการ **ARP** ถาวร ด้วยตนเองโดยใช้คำสั่ง **arp** รายการตาราง **ARP** เจาะจงต้องถูกเพิ่มบนเซิร์ฟเวอร์สำหรับแต่ละโฮสต์ที่ต้องมีการตอบกลับ **RARP** จากต้นทางที่มีสิทธิ

งานการระบุชื่อโลคัล (/etc/hosts)

กำหนดค่าไฟล์ /etc/hosts ถ้าเน็ตเวิร์กของคุณ มีขนาดเล็ก และคุณกำลังใช้รูปแบบการตั้งชื่อแบบราบ

แม้ว่าคุณกำลังใช้รูปแบบการตั้งชื่อแบบลำดับชั้น (หรือโดเมน) กับเซิร์ฟเวอร์ชื่อ คุณอาจต้องกำหนดค่าไฟล์ /etc/hosts เพื่อระบุโฮสต์ที่เนมเซิร์ฟเวอร์ไม่รู้จัก

กำหนดค่าระบบของคุณสำหรับการกำหนดโฮสต์โลคัลโดยใช้ System Management Interface Tool (SMIT) หรือคำสั่ง ถ้าคุณเลือกวิธีใช้คำสั่ง ขอให้แน่ใจว่าคงรูปแบบของไฟล์ /etc/hosts ดังอธิบายใน รูปแบบไฟล์โฮสต์สำหรับ TCP/IP ใน *การอ้างอิงไฟล์*

ตารางที่ 63. งานการระบุชื่อโลคัล

ภารกิจ	วิธีลัด SMIT	คำสั่งหรือไฟล์
List All the Hosts	**smit lshostent**	Use the **hostent** command or **view** /etc/hosts
เพิ่มโฮสต์	**smit mkhostent**	Use the **hostent** command or **edit** /etc/hosts
Change/Show Characteristics of a Host	**smit chhostent**	Use the **hostent** command or **edit** /etc/hosts
Remove a Host	**smit rmhostent**	Use the **hostent** command or **edit** /etc/hosts

การวางแผนสำหรับการระบุชื่อ DOMAIN

ข้อเสนอแนะเหล่านี้สามารถช่วยคุณวางแผนระบบการระบุชื่อ DOMAIN ของคุณ

ถ้าคุณเป็นส่วนหนึ่งของอินเทอร์เน็ตเวิร์กขนาดใหญ่ ประสานการตั้งค่า โดเมนของคุณ กับเนมเซิร์ฟเวอร์ที่มีสิทธิกลาง

- เนื่องจากมีความเป็นไปได้อย่างสูงในสถาปัตยกรรมและการกำหนดค่า จะมี ความคุ้นเคยกับ **TCP/IP**, DNS และ BIND ก่อนที่คุณจะสรุปแผนที่ใช้ ถ้าคุณวางแผนที่จะใช้บริการข้อมูลเน็ตเวิร์ก ขอให้ทำความคุ้นเคยกับ NFS และ NIS ด้วย หนังสือที่เกี่ยวกับหัวข้อเหล่านี้มีให้อ่านได้มากมาย
- วางแผนเลย

  การเปลี่ยนชื่อมีความยุ่งยาก *มาก* กว่าการตั้งค่า เริ่มต้น ขอคำแนะนำจากองค์กรของคุณเกี่ยวกับเน็ตเวิร์ก เกตเวย์ เนมเซิร์ฟเวอร์ และชื่อโฮสต์ก่อนที่คุณจะตั้งค่าไฟล์

- ตั้งค่าเนมเซิร์ฟเวอร์ที่ซ้ำกัน

  ถ้าคุณไม่สามารถตั้งค่าเนมเซิร์ฟเวอร์ ที่ซ้ำกัน ขอให้แน่ใจว่าได้ตั้งค่าเนมเซิร์ฟเวอร์ที่เป็นสเลฟและแนะนำเพื่อให้คุณมีข้อมูลสำรอง
- ในการเลือกเนมเซิร์ฟเวอร์ โปรดจดจำสิ่งต่อไปนี้:
  - เลือกเครื่องที่ตั้งอยู่ใกล้กับระบบภายนอกมากที่สุด
  - เนมเซิร์ฟเวอร์ควรเป็นอิสระที่สุดเท่าที่เป็นได้ ลองใช้ตัวจ่ายไฟ อื่น และการวางสายเคเบิลแบบอิสระ
  - ค้นหาเน็ตเวิร์กอื่นเพื่อสำรองบริการการระบุชื่อของคุณ และดำเนินการ เช่นเดียวกันกับเน็ตเวิร์กอื่น
- ทดสอบเซิร์ฟเวอร์
  - ทดสอบทั้งการระบุชื่อปกติ และสำรอง
  - ทดสอบการถ่ายโอนโซนจากเนมเซิร์ฟเวอร์ต้นแบบไปยังสเลฟ
  - ทดสอบแต่ละเนมเซิร์ฟเวอร์หลังจากระบบขัดข้อง หรือบูตใหม่
- ส่งการร้องขอการระบุชื่อไปยังเซิร์ฟเวอร์ตัวส่งต่อก่อนที่จะไปยังเนมเซิร์ฟเวอร์ ภายนอก ซึ่งอนุญาตให้เนมเซิร์ฟเวอร์ของคุณแบ่งใช้แคช และปรับปรุงประสิทธิภาพการทำงาน โดยการลดปริมาณงานบนเนมเซิร์ฟเวอร์ต้นแบบของคุณ

```
objectclass container
        requires
                objectclass,
                cn
objectclass hosts
        requires
                objectclass,
                hname
        allows
                addr
                halias,
                comment
```

การระบุเนมเซิร์ฟเวอร์

ในเน็ตเวิร์กแบบลำดับชั้น โฮสต์จะถูกกำหนดเป็น *เซิร์ฟเวอร์ชื่อ* โฮสต์เหล่านี้จะแปลงชื่อเป็น IP addresses สำหรับโฮสต์อื่นๆ

named daemon ควบคุมฟังก์ชันเนมเซิร์ฟเวอร์ ดังนั้น ต้องรันอยู่บนโฮสต์เนมเซิร์ฟเวอร์

ก่อนคุณกำหนดคอนฟิกเซิร์ฟเวอร์ชื่อ ให้ตัดสินใจเลือกชนิดที่เหมาะสมกับ เครือข่ายที่จะให้บริการมากที่สุด เนมเซิร์ฟเวอร์มีหลายชนิด

เซิร์ฟเวอร์ชื่อหลัก จัดเก็บฐานข้อมูลที่มีข้อมูลการแม็พ ชื่อกับแอดเดรสอย่างแท้จริง โดยโหลดข้อมูลจากไฟล์หรือดิสก์ และสามารถมอบสิทธิ ให้แก่เซิร์ฟเวอร์อื่นในโดเมน *เซิร์ฟเวอร์ชื่อ slave* หรือ *เซิร์ฟเวอร์ชื่อ stub* ได้รับ ข้อมูลในเวลาสตาร์ทอัพระบบสำหรับโซนเฉพาะของสิทธิ จากเซิร์ฟเวอร์ชื่อหลัก จากนั้น ขอให้เซิร์ฟเวอร์หลัก อัพเดตข้อมูลเป็นระยะ *เนมเซิร์ฟเวอร์ hint* ตอบกลับ การร้องขอเพื่อระบุชื่อโดยการสอบถามเนมเซิร์ฟเวอร์ที่มีสิทธิ ให้ข้อมูลที่จำเป็น

หมายเหตุ: เนมเซิร์ฟเวอร์ **named** รุ่นก่อนหน้าที่ระบุ เนมเซิร์ฟเวอร์หลักเป็นเนมเซิร์ฟเวอร์เริ่มต้น เนมเซิร์ฟเวอร์ย่อยเป็นเนมเซิร์ฟเวอร์รอง และเนมเซิร์ฟเวอร์แนะนำเป็นเนมเซิร์ฟเวอร์การแคช เท่านั้น

โปรดจำไว้ว่าเนมเซิร์ฟเวอร์สามารถทำงานโดยมีความสามารถแตกต่างกันสำหรับ โซนของสิทธิที่ต่างกัน ตัวอย่างเช่น โฮสต์เนมเซิร์ฟเวอร์หนึ่งสามารถเป็นเนมเซิร์ฟเวอร์ หลักสำหรับโซนหนึ่ง และเป็นเนมเซิร์ฟเวอร์ย่อยสำหรับอีกโซนหนึ่ง ถ้าระบบของคุณติดตั้ง NIS ไว้แล้ว เซอร์วิสเหล่านี้ยังสามารถจัดเตรียม การแก้ไขชื่อ

มีหลายไฟล์ที่เกี่ยวข้องในการกำหนดคอนฟิกเซิร์ฟเวอร์ชื่อ

ไอเท็ม	คำอธิบาย
`conf`	ไฟล์นี้ถูกอ่านเมื่อ **named** daemon เริ่มทำงาน เร็กคอร์ดในไฟล์ `conf` แจ้งให้ **named** daemon ทราบว่าเซิร์ฟเวอร์ชนิดใด, โดเมนใดที่มีสิทธิเหนือ (โซน ของสิทธิ) และรับข้อมูลจากที่ใดเพื่อเริ่มการตั้งค่าฐานข้อมูล ชื่อดีฟอลต์ของไฟล์นี้คือ `/etc/named.conf` อย่างไรก็ตาม คุณสามารถเปลี่ยนชื่อของไฟล์นี้ได้โดยการระบุชื่อและพาธของ ไฟล์บนบรรทัดคำสั่งเมื่อ **named** daemon เริ่มทำงาน ถ้าคุณปรารถนาที่จะใช้ `/etc/named.conf` เป็นไฟล์ `conf` และไม่มีอยู่ จะมีข้อความสร้างขึ้นในไฟล์ `syslog` และ **named** จบการทำงาน อย่างไรก็ตาม ถ้ามีการระบุไฟล์ `conf` ทางเลือก และไฟล์ทางเลือกไม่มีอยู่ จะไม่มีการสร้างข้อความแสดงข้อผิดพลาด และ **named** ทำงานต่อไป
แคช	มีข้อมูลเกี่ยวกับแคชโลคัล ไฟล์แคชโลคัลมี ชื่อและแอดเดรสของเนมเซิร์ฟเวอร์สิทธิสูงสุดในเน็ตเวิร์ก ไฟล์แคชสามารถใช้ Standard Resource Record Format ชื่อของไฟล์แคช ถูกตั้งค่าในไฟล์ `conf`

ข้อมูลโดเมน	มีไฟล์ข้อมูลโดเมนปกติสามไฟล์ โดยเรียกเป็นไฟล์ข้อมูล **named** ไฟล์ **named** *local* มี ข้อมูลการระบุแอดเดรสสำหรับลูปแบคบนโลคัล ไฟล์ **named** *data* มีข้อมูลการระบุแอดเดรสสำหรับทุกเครื่องในโซนสิทธิ ของเนมเซิร์ฟเวอร์ ไฟล์ **named** *reverse data* มีข้อมูลการระบุแอดเดรสสำรองสำหรับทุกเครื่องใน โซนสิทธิของเนมเซิร์ฟเวอร์ ไฟล์ข้อมูลโดเมนใช้ Standard Resource Record Format โดยชื่อไฟล์ผู้ใช้สามารถกำหนดได้ และถูกตั้งค่าในไฟล์ `conf` โดยหลักการแล้ว โดยทั่วไปชื่อของไฟล์เหล่านี้จะรวมชื่อของ daemon (`named`) และชนิดไฟล์ และชื่อของโดเมน ถูกกำหนดในส่วนขยาย ตัวอย่างเช่น เนมเซิร์ฟเวอร์สำหรับโดเมน `abc` จะ มีไฟล์ต่อไปนี้:
	`named.abc.data`
	`named.abc.rev`
	`named.abc.local`
	
	เมื่อแก้ไขไฟล์ข้อมูล **named** นั้น หมายเลขลำดับใน SOA Resource Record ต้องถูกเพิ่มสำหรับ เนมเซิร์ฟเวอร์ย่อยเพื่อให้ทราบการเปลี่ยนแปลงโซนใหม่ที่เหมาะสม

`resolv.conf`	การมีไฟล์นี้อยู่แสดงว่าโฮสต์จะไปที่เนมเซิร์ฟเวอร์ เพื่อระบุชื่อเป็นอันดับแรก ถ้าไม่มีไฟล์ `resolv.conf` โฮสต์จะค้นหาในไฟล์ `/etc/hosts` เพื่อ ทำการระบุชื่อ บนเนมเซิร์ฟเวอร์ ไฟล์ `resolv.conf` ต้องมีอยู่ และสามารถมีโลคัลโฮสต์แอดเดรส แอดเดรสลูปแบค (127.0.0.1) หรือเป็นค่าว่าง **หมายเหตุ:** รูทีนตัวแก้ไขจำเป็นต้องให้โดเมนดีฟอลต์ถูกตั้งค่า ถ้าโดเมนดีฟอลต์ไม่ถูกตั้งค่าในไฟล์ `/etc/resolv.conf` ดังนั้นต้องถูกตั้งค่าใน `hostname`

Time-to-live (TTL) ถูกระบุในเร็กคอร์ดรีซอร์ส ถ้า TTL ไม่ถูกระบุ ในเร็กคอร์ด ความยาวของช่วงเวลานี้จะใช้ค่าดีฟอลต์เป็นฟิลด์ต่ำสุด ที่กำหนดในเร็กคอร์ด start of authority (SOA) สำหรับโซนนั้น TTL ถูกใช้ เมื่อข้อมูลถูกเก็บภายนอกโซน (ในแคช) เพื่อให้แน่ใจว่าข้อมูล ไม่ถูกเก็บไว้โดยไม่มีกำหนด

การตั้งค่าคอนฟิกเนมเซิร์ฟเวอร์โดเมน:

ในสถานการณ์จำลองนี้ จะมีการตั้งค่าคอนฟิกเนมเซิร์ฟเวอร์ master, เนมเซิร์ฟเวอร์ slave, และเนมเซิร์ฟเวอร์ hint เพื่อทำการแก้ไขชื่อ เนมเซิร์ฟเวอร์ แต่ละรายการจะมีเครื่องแยกต่างหากกัน และแต่ละเซิร์ฟเวอร์จะมีไฟล์ที่ตั้งค่าคอนฟิก `/etc/named.conf` แม้ว่าข้อมูลในแต่ละเซิร์ฟเวอร์จะแตกต่างกัน ไฟล์ `/etc/named.conf` ถูก อ่านในทุกครั้งที่ **named** daemon เริ่มต้นขึ้น และระบุชนิดของ เซิร์ฟเวอร์ (master, slave, หรือ hint) และตำแหน่งที่จะรับข้อมูลการแก้ไข ชื่อ เนมเซิร์ฟเวอร์แต่ละเครื่องเหล่านี้จะกำลังรัน BIND 8

เนมเซิร์ฟเวอร์หลักจะมีการตั้งค่าคอนฟิกเพื่อนำเสนอการแก้ไขชื่อ สำหรับโซน `abc.aus.century.com` ในสถานการณ์จำลองนี้ IP แอดเดรสของเนมเซิร์ฟเวอร์หลักคือ `192.9.201.1` และชื่อโฮสต์ ของเซิร์ฟเวอร์คือ `venus.abc.aus.century.com` เซิร์ฟเวอร์จะนำเสนอการแก้ไขชื่อ สำหรับชื่อโฮสต์ venus, earth, mars, และ jupiter ไฟล์ `/etc/named.conf` จะมีการ ตั้งค่าคอนฟิกเพื่อระบุว่า **named** daemon ควรค้นหาไฟล์ข้อมูล ในไดเร็กทอรี `/usr/local/domain` ไฟล์ข้อมูลที่จะถูกตั้งค่าคอนฟิกสำหรับเนมเซิร์ฟเวอร์หลัก คือ `named.ca`, `named.abc.local`, `named.abc.data`, และ `named.abc.rev`

จากนั้น จะตั้งค่าคอนฟิกเนมเซิร์ฟเวอร์ย่อย ชื่อโฮสต์ของเนมเซิร์ฟเวอร์ย่อย จะเป็น earth.abc.aus.century.com และ IP แอดเดรส จะเป็น 192.9.201.5 ในไฟล์ /etc/named.conf ของเนมเซิร์ฟเวอร์ย่อย เราจะระบุแอดเดรสของเนมเซิร์ฟเวอร์หลัก เพื่อให้เนมเซิร์ฟเวอร์ย่อย สามารถทำซ้ำไฟล์ named.abc.data และ named.abc.rev ของเนมเซิร์ฟเวอร์หลักได้ นอกจากนี้ จะมีการตั้งค่าคอนฟิกไฟล์ข้อมูล named.ca และ named.abc.local สำหรับเซิร์ฟเวอร์นี้

จากนั้น จะตั้งค่าคอนฟิกเนมเซิร์ฟเวอร์ hint เนมเซิร์ฟเวอร์ hint จะจัดเก็บโลคัลแคชของการแม็พชื่อโฮสต์ และแอดเดรส หากแอดเดรสหรือเนมเซิร์ฟเวอร์ที่ร้องขอไม่ได้อยู่ในแคช เซิร์ฟเวอร์ hint จะติดต่อกับเนมเซิร์ฟเวอร์หลัก เพื่อรับข้อมูลการแก้ไขและเพิ่มข้อมูลนั้นลงในแคช นอกจากนี้ จะมีการตั้งค่าคอนฟิกไฟล์ข้อมูล named.ca และ named.abc.local สำหรับเซิร์ฟเวอร์นี้

ข้อมูลทั้งหมดในไฟล์ข้อมูล named (ไม่ใช่ไฟล์ /etc/named.conf) บนเนมเซิร์ฟเวอร์ ต้องอยู่ในรูปแบบเร็กคอร์ดรีซอร์สมาตรฐาน หากต้องการคำอธิบายข้อมูล เกี่ยวกับไฟล์ข้อมูล named ให้ดูที่ รูปแบบ เร็กคอร์ดรีซอร์สมาตรฐานสำหรับ TCP/IP ใน *การอ้างอิงไฟล์*

ผู้ดูแลระบบ สำหรับเนมเซิร์ฟเวอร์แต่ละเครื่องจะเป็น gail.zeus.abc.aus.century.com ซึ่งมีการระบุอยู่ในไฟล์ข้อมูลโลคัลบนเนมเซิร์ฟเวอร์แต่ละเครื่อง นอกจากนั้น ในสถานการณ์จำลองนี้ root ของเนมเซิร์ฟเวอร์คือ relay.century.com ที่มี IP แอดเดรส 129.114.1.2

ที่ตอนท้ายของสถานการณ์จำลองนี้ จะมีการนำเสนอการแก้ไขชื่อสำหรับโฮสต์ venus, earth, mars, และ jupiter นอกจากนี้ ยังจะมีการนำเสนอการแก้ไขชื่อย้อนกลับด้วย (IP แอดเดรส-ไปยัง- ชื่อโฮสต์) เมื่อได้รับคำร้องขอที่ไม่สามารถแก้ไขได้ เนมเซิร์ฟเวอร์หลัก จะติดต่อกับ relay.century.com เพื่อค้นหาข้อมูลที่ ต้องการ

สิ่งที่ต้องพิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ได้ผ่านการทดสอบโดยใช้เวอร์ชันเฉพาะของ AIX ผลลัพธ์ที่คุณได้อาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชันและระดับ AIX ของคุณ

ขั้นตอนที่ 1. ตั้งค่าคอนฟิกเนมเซิร์ฟเวอร์หลัก

1. บนเนมเซิร์ฟเวอร์หลัก ให้เปิดไฟล์ /etc/named.conf หากไม่มีไฟล์ /etc/named.conf อยู่ในไดเร็กทอรี /etc ให้สร้างไฟล์ขึ้นโดยการรันคำสั่งต่อไปนี้:

   ```
   touch /etc/named.conf
   ```

   ทำ ดังต่อไปนี้เพื่อตั้งค่าคอนฟิกไฟล์ /etc/named.conf:

   a. ระบุส่วนคำสั่งไดเร็กทอรีในอ็อพชัน stanza ส่วนคำสั่งนี้ช่วยให้ไฟล์ข้อมูล named สามารถใช้พาธที่สัมพันธ์กับไดเร็กทอรี /usr/local/domain ในสถานการณ์จำลองนี้ มีการเพิ่มข้อมูลต่อไปนี้:

      ```
      options {
          directory "/usr/local/domain";
      };
      ```

      ถ้าคุณเลือกไม่ระบุไดเร็กทอรีที่นี่ จะมีการค้นหาไดเร็กทอรี /etc เพื่อหาไฟล์ข้อมูลซึ่งจำเป็น

   b. เพื่อให้สามารถแคชข้อมูลเร็กคอร์ดภายนอกโซนที่กำหนดได้ ให้ระบุ ชื่อของไฟล์โซน hint ในสถานการณ์จำลองนี้ มีการเพิ่มข้อมูล ต่อไปนี้:

      ```
      zone "." IN {
          type hint;
          file "named.ca";
      };
      ```

c. เพิ่ม stanzas ต่อไปนี้เพื่อระบุแต่ละโซน ชนิดของเนมเซิร์ฟเวอร์ ซึ่งคุณกำลังตั้งค่าคอนฟิก และไฟล์ข้อมูลโดเมนของเนมเซิร์ฟเวอร์ของคุณ ในสถานการณ์จำลองนี้ เซิร์ฟเวอร์หลักสำหรับทั้งโซนไปข้างหน้าและย้อนกลับเป็นดังต่อไปนี้:

```
zone "abc.aus.century.com" in {
    type master;
    file "named.abc.data";
};
zone "201.9.192.in-addr.arpa" in {
    type master;
    file "named.abc.rev";
};
```

d. กำหนดชื่อของไฟล์โลคัล named ตัวอย่างเช่น:

```
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "named.abc.local";
};
```

หลังจากแก้ไขไฟล์แล้ว ให้บันทึกและปิดไฟล์

2. เปิดไฟล์ /usr/local/domain/named.ca เพิ่ม แอดเดรสของ root ของเนมเซิร์ฟเวอร์สำหรับโดเมน มีการเพิ่มข้อมูลต่อไปนี้ ในสถานการณ์จำลองนี้:

```
; root name servers.
.          IN    NS    relay.century.com.
relay.century.com.   3600000    IN    A     129.114.1.2
```

หลังจากแก้ไข ไฟล์แล้ว ให้บันทึกและปิดไฟล์

3. เปิดไฟล์ /usr/local/domain/named.abc.local เพิ่มข้อมูลต่อไปนี้:
   - ข้อมูลการเริ่มต้นสิทธิ (SOA) ของโซนและ time-to-live ดีฟอลต์ มีการเพิ่มข้อมูลต่อไปนี้ในสถานการณ์จำลองนี้:

```
$TTL 3h    ;3 hour

@ IN SOA venus.abc.aus.century.com. gail.zeus.abc.aus.century.com.  (

                            1       ;serial
                            3600    ;refresh
                            600     ;retry
                            3600000 ;expire
                            3600    ;negative caching TTL
)
```

   - เร็กคอร์ดเนมเซิร์ฟเวอร์ (NS) แทรกพื้นที่ว่างแท็บที่ตอนต้นของ บรรทัด; **named** daemon จะแทนที่พื้นที่ว่างแท็บด้วยชื่อโซน:

```
<tab> IN    NS     venus.abc.aus.century.com.
```

   - เร็กคอร์ดตัวชี้ (PTR)

```
1      IN    PTR    localhost.
```

หลังจากแก้ไข ไฟล์แล้ว ให้บันทึกและปิดไฟล์

4. เปิดไฟล์ /usr/local/domain/named.abc.data เพิ่มข้อมูลต่อไปนี้:
   - ข้อมูลการเริ่มต้นสิทธิของโซนและ time-to-live ดีฟอลต์ของโซน เร็กคอร์ดนี้ออกแบบการเริ่มต้นของโซน ใช้ได้เพียงหนึ่งการเริ่มต้น ของเร็กคอร์ดสิทธิต่อโซนเท่านั้น ในสถานการณ์จำลองนี้ มีการเพิ่มข้อมูลต่อไปนี้:

```
$TTL 3h    ;3 hour

@ IN    SOA     venus.abc.aus.century.com. gail.zeus.abc.aus.century.com. (
                1       ;serial
                3600    ;refresh
                600     ;retry
                3600000 ;expire
                3600    ;negative caching TTL
)
```

- เร็กคอร์ดเนมเซิร์ฟเวอร์สำหรับเนมเซิร์ฟเวอร์หลักในโซน แทรกพื้นที่ว่างแท็บที่ตอนต้นของ บรรทัด; **named** daemon จะแทนที่พื้นที่ว่างแท็บด้วย ชื่อโซน:

```
<tab> IN    NS       venus.abc.aus.century.com.
```

- ข้อมูลการแก้ไขชื่อเป็นแอดเดรสบนโฮสต์ทั้งหมดในโซนสิทธิของเซิร์ฟเวอร์ชื่อ:

```
venus        IN    A        192.9.201.1
earth        IN    A        192.9.201.5
mars         IN    A        192.9.201.3
jupiter      IN    A        192.9.201.7
```

รวมรายการชนิดอื่น เช่น เร็กคอร์ดชื่อ canonical และเร็กคอร์ด mail exchanger ตามต้องการ

หลังจากแก้ไขไฟล์แล้ว ให้บันทึกและปิดไฟล์

5. เปิดไฟล์ /usr/local/domain/named.abc.rev เพิ่มข้อมูลต่อไปนี้:

- ข้อมูลการเริ่มต้นสิทธิของโซนและ time-to-live ดีฟอลต์ เร็กคอร์ดนี้ออกแบบการเริ่มต้นของโซน ใช้ได้เพียงหนึ่งการเริ่มต้น ของเร็กคอร์ดสิทธิต่อโซนเท่านั้น

```
$TTL 3h    ;3 hour

@  IN  SOA  venus.abc.aus.century.com. gail.zeus.abc.aus.century.com.  (
                          1       ;serial
                          3600    ;refresh
                          600     ;retry
                          3600000 ;expire
                          3600    ;negative caching TTL
)
```

- รายการชนิดอื่น เช่น เร็กคอร์ดเนมเซิร์ฟเวอร์ หากคุณกำลังรวม เร็กคอร์ดเหล่านี้ให้แทรกพื้นที่ว่างแท็บที่ตอนต้นของบรรทัด; **named** daemon จะแทนที่พื้นที่ว่างแท็บด้วยชื่อโซน ในสถานการณ์จำลองนี้ มีการเพิ่มข้อมูลต่อไปนี้:

```
<tab> IN      NS      venus.abc.aus.century.com.
```

- ข้อมูลการแก้ไขแอดเดรสเป็นชื่อบนโฮสต์ทั้งหมดในโซนสิทธิของ เนมเซิร์ฟเวอร์

```
1                    IN    PTR    venus.abc.aus.century.com.
5                    IN    PTR    earth.abc.aus.century.com.
3                    IN    PTR    mars.abc.aus.century.com.
7                    IN    PTR    jupiter.abc.aus.century.com.
```

หลังจากแก้ไข ไฟล์แล้ว ให้บันทึกและปิดไฟล์

6. สร้างไฟล์ /etc/resolv.conf โดยการรันคำสั่ง ต่อไปนี้:

```
touch /etc/resolv.conf
```

ถ้าไฟล์นี้ มีอยู่ แสดงว่าโฮสต์ควรจะใช้เนมเซิร์ฟเวอร์สำหรับการแก้ไขชื่อ

7. เพิ่มรายการต่อไปนี้ลงในไฟล์ /etc/resolv.conf:

   ```
   nameserver 127.0.0.1
   ```

   แอดเดรส 127.0.0.1 คือ loopback แอดเดรส ซึ่งทำให้โฮสต์เข้าถึงตัวเอง เป็นเนมเซิร์ฟเวอร์ไฟล์ /etc/resolv.conf ยังสามารถ ประกอบด้วยรายการที่คล้ายกับตัวอย่างต่อไปนี้:

   ```
   domain abc.aus.century.com
   ```

   ใน กรณีนี้ abc.aus.century.com คือชื่อโดเมน

   หลังจากแก้ไข ไฟล์แล้ว ให้บันทึกและปิดไฟล์

8. ใช้พาธด่วน smit stnamed SMIT เพื่อเปิดใช้งาน **named** daemon พาธด่วนนี้จะเริ่มต้น daemon ด้วยสตาร์ทอัพระบบแต่ละรายการ บ่งชี้ว่าคุณ ต้องการเริ่มต้น **named** daemon เดี๋ยวนี้ หรือในการรีสตาร์ทระบบ ครั้งถัดไป หรือทั้งสองอย่าง

ขั้นตอนที่ 2. ตั้งค่าคอนฟิกเนมเซิร์ฟเวอร์ย่อย

เมื่อต้องการตั้งค่าคอนฟิก เนมเซิร์ฟเวอร์ย่อย ให้ใช้โพรซีเดอร์ต่อไปนี้ คุณจะแก้ไขชุดของไฟล์ จากนั้นใช้ SMIT เพื่อเริ่มต้น **named** daemon

1. บนเนมเซิร์ฟเวอร์หลัก ให้เปิดไฟล์ /etc/named.conf หากไม่มีไฟล์ /etc/named.conf อยู่ในไดเร็กทอรี /etc ให้สร้างไฟล์ขึ้นโดยการรันคำสั่งต่อไปนี้:

   ```
   touch /etc/named.conf
   ```

   ทำ ดังต่อไปนี้เพื่อตั้งค่าคอนฟิกไฟล์ /etc/named.conf:

   a. ระบุส่วนคำสั่งไดเร็กทอรีในอ็อพชัน stanza ส่วนคำสั่งนี้ช่วยให้ไฟล์ข้อมูล named สามารถใช้พาธที่สัมพันธ์กับไดเร็กทอรี /usr/local/domain ในสถานการณ์จำลองนี้ มีการเพิ่มข้อมูลต่อไปนี้:

      ```
      options {
          directory "/usr/local/domain";
      };
      ```

      ถ้าคุณเลือกไม่ระบุไดเร็กทอรีที่นี่ **named** daemon จะค้นหาไดเร็กทอรี /etc เพื่อหาไฟล์ข้อมูลซึ่ง จำเป็น

   b. เพื่อให้สามารถแคชข้อมูลเร็กคอร์ดภายนอกโซนที่กำหนดได้ ให้ระบุ ชื่อของไฟล์โซน hint สำหรับเนมเซิร์ฟเวอร์:

      ```
      zone "." IN {
          type hint;
          file "named.ca";
      };
      ```

   c. ระบุส่วนคำสั่งโซน slave แต่ละ stanza ประกอบด้วยชนิดโซน ชื่อไฟล์ซึ่งเนมเซิร์ฟเวอร์สามารถทำสำเนาสำรองข้อมูล และ IP แอดเดรส ของเนมเซิร์ฟเวอร์หลัก ซึ่งเนมเซิร์ฟเวอร์ย่อยจะทำซ้ำไฟล์ข้อมูลจาก แอดเดรสนั้น ในสถานการณ์จำลองนี้ เราเพิ่มส่วนคำสั่งโซนย่อยต่อไปนี้:

      ```
      zone "abc.aus.century.com" IN {
          type slave;
          file "named.abc.data.bak";
          masters { 192.9.201.1; };
      };

      zone "201.9.192.in-addr.arpa" IN {
          type slave;
          file "named.abc.rev.bak";
          masters { 192.9.201.1; };
      };
      ```

d. เพื่อสนับสนุนการแก้ไข loopback เน็ตเวิร์กแอดเดรส ให้ระบุโซนชนิด *master* ที่มี ซอร์สเป็น `named.abc.local` และโดเมนซึ่งเซิร์ฟเวอร์ชื่อรับผิดชอบ

```
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "named.abc.local";
};
```

หลังจากแก้ไขไฟล์แล้ว ให้บันทึกและปิดไฟล์

2. แก้ไขไฟล์ `/usr/local/domain/named.ca`

ไฟล์นี้ มีเซิร์ฟเวอร์แอดเดรสซึ่งไม่ใช่เซิร์ฟเวอร์โดเมน root ของเน็ตเวิร์ก ในสถานการณ์จำลองนี้ มีการเพิ่มข้อมูลต่อไปนี้:

```
; root name servers.
.          IN    NS    relay.century.com.
relay.century.com.   3600000    IN    A     129.114.1.2
```

หลังจากแก้ไข ไฟล์แล้ว ให้บันทึกและปิดไฟล์

3. เปิดไฟล์ `/usr/local/domain/named.abc.local` ในสถานการณ์จำลองนี้ มีการเพิ่มข้อมูลต่อไปนี้:
   - ข้อมูลการเริ่มต้นสิทธิ (SOA) ของโซนและ time-to-live ดีฟอลต์:

     ```
     $TTL 3h    ;3 hour

     @ IN SOA earth.abc.aus.century.com. gail.zeus.abc.aus.century.com.  (

                                1       ;serial
                                3600    ;refresh
                                600     ;retry
                                3600000 ;expire
                                3600    ;negative caching TTL
     )
     ```

   - เร็กคอร์ดเนมเซิร์ฟเวอร์ (NS) แทรกพื้นที่ว่างแท็บที่ตอนต้นของ บรรทัด; **named** daemon จะแทนที่พื้นที่ว่างแท็บด้วยชื่อโซน ตัวอย่างเช่น:

     ```
     <tab> IN    NS     earth.abc.aus.century.com.
     ```

   - เร็กคอร์ดตัวชี้ (PTR)

     ```
     1      IN    PTR    localhost.
     ```

หลังจากแก้ไขไฟล์แล้ว ให้บันทึกและปิดไฟล์

4. สร้างไฟล์ `/etc/resolv.conf` โดยการรันคำสั่ง ต่อไปนี้:

```
touch /etc/resolv.conf
```

5. เพิ่มรายการต่อไปนี้ในไฟล์นั้น:

```
nameserver 127.0.0.1
domain abc.aus.century.com
```

หลังจากแก้ไขไฟล์แล้ว ให้บันทึกและปิดไฟล์

6. ใช้พาธด่วน `smit stnamed` SMIT เพื่อเปิดใช้งาน **named** daemon พาธด่วนนี้จะเริ่มต้น daemon ด้วยสตาร์ทอัพระบบแต่ละรายการ บ่งชี้ว่าคุณ ต้องการเริ่มต้น **named** daemon เดี๋ยวนี้ หรือในการรีสตาร์ทระบบ ครั้งถัดไป หรือทั้งสองอย่าง

ขั้นตอนที่ 3. ตั้งค่าคอนฟิกเนมเซิร์ฟเวอร์ Hint

เมื่อต้องการตั้งค่า คอนฟิก hint หรือเนมเซิร์ฟเวอร์ *แคชอย่างเดียว* ให้ใช้โพรซีเดอร์ต่อไปนี้ ซึ่ง จะแก้ไขชุดของไฟล์ จากนั้นใช้ SMIT หรือบรรทัดคำสั่งเพื่อเริ่มต้น **named** daemon

1. บนเนมเซิร์ฟเวอร์ hint ให้แก้ไขไฟล์ /etc/named.conf หากไม่มีไฟล์ /etc/named.conf อยู่ในไดเร็กทอรี /etc ให้สร้างไฟล์ขึ้นโดยการรันคำสั่งต่อไปนี้:

   ```
   touch /etc/named.conf
   ```

   ทำ ดังต่อไปนี้เพื่อตั้งค่าคอนฟิกไฟล์ /etc/named.conf:

   a. ระบุส่วนคำสั่งไดเร็กทอรีในอ็อพชัน stanza ส่วนคำสั่งนี้ช่วยให้ไฟล์ข้อมูล named สามารถใช้พาธที่สัมพันธ์กับไดเร็กทอรี /usr/local/domain ในสถานการณ์จำลองนี้ มีการเพิ่มข้อมูลต่อไปนี้:

      ```
      options {
          directory "/usr/local/domain";
      };
      ```

   b. เพื่อสนับสนุนการแก้ไข loopback เน็ตเวิร์กแอดเดรส ให้ระบุโซนชนิด *master* ที่มี ซอร์สเป็น named.abc.local และโดเมนซึ่งเซิร์ฟเวอร์ชื่อรับผิดชอบ ในตัวอย่างนี้ อ็อพชันไดเร็กทอรีคีย์เวิร์ด มีการระบุในไฟล์ /etc/named.conf

      ```
      zone "0.0.127.in-addr.arpa" IN {
          type master;
         file "named.abc.local";
      };
      ```

   c. ระบุชื่อของไฟล์โซนแคช ตัวอย่างเช่น:

      ```
      zone "." IN {
          type hint;
          file "named.ca";
      };
      ```

   หลังจากแก้ไขไฟล์แล้ว ให้บันทึกและปิดไฟล์

2. แก้ไขไฟล์ /usr/local/domain/named.ca

   ไฟล์นี้ มีแอดเดรสของเซิร์ฟเวอร์ที่เป็นเนมเซิร์ฟเวอร์ซึ่งได้รับอนุญาต สำหรับโดเมน root ของเครือข่าย ตัวอย่างเช่น:

   ```
   ; root name servers.
   .          IN    NS    relay.century.com.
   relay.century.com.   3600000    IN    A     129.114.1.2
   ```

   หลังจากแก้ไข ไฟล์แล้ว ให้บันทึกและปิดไฟล์

3. แก้ไขไฟล์ /usr/local/domain/named.local ในสถานการณ์ จำลองนี้ มีการเพิ่มข้อมูลต่อไปนี้ในไฟล์นี้:

   - ข้อมูลการเริ่มต้นสิทธิ (SOA) ของโซนและ time-to-live ดีฟอลต์:

     ```
     $TTL 3h    ;3 hour

     @ IN SOA venus.abc.aus.century.com. gail.zeus.abc.aus.century.com.  (

                                  1       ;serial
                                  3600    ;refresh
                                  600     ;retry
                                  3600000 ;expire
                                  3600    ;negative caching TTL
     )
     ```

- เร็กคอร์ดเนมเซิร์ฟเวอร์ (NS) แทรกพื้นที่ว่างแท็บที่ตอนต้นของ บรรทัด; **named** daemon จะแทนที่พื้นที่ว่างแท็บด้วยชื่อโซน:

  ```
  <tab> IN    NS     venus.abc.aus.century.com.
  ```

- เร็กคอร์ดตัวชี้ (PTR)

  ```
  1      IN    PTR    localhost.
  ```

  หลังจากแก้ไขไฟล์แล้ว ให้บันทึกและปิดไฟล์

4. สร้างไฟล์ `/etc/resolv.conf` โดยการรันคำสั่ง ต่อไปนี้:

   ```
   touch /etc/resolv.conf
   ```

5. เพิ่มรายการต่อไปนี้ในไฟล์นั้น:

   ```
   nameserver 127.0.0.1
   domain abc.aus.century.com
   ```

   หลังจากแก้ไขไฟล์แล้ว ให้บันทึกและปิดไฟล์

6. ใช้พาธด่วน `smit stnamed` SMIT เพื่อเปิดใช้งาน **named** daemon พาธด่วนนี้จะเริ่มต้น daemon ด้วยสตาร์ทอัพระบบแต่ละรายการ บ่งชี้ว่าคุณ ต้องการเริ่มต้น **named** daemon เดี๋ยวนี้ หรือในการรีสตาร์ทระบบ ครั้งถัดไป หรือทั้งสองอย่าง

เมื่อคุณรีบูต จะมีการตั้งค่าคอนฟิก IPv6 ทำซ้ำกระบวนการนี้ สำหรับแต่ละโฮสต์

การตั้งค่าโดเมนของเมลเซิร์ฟเวอร์:

การตั้งค่าโดเมนของเมลเซิร์ฟเวอร์จัดเตรียมวิธีแบบง่ายให้กับผู้ใช้ที่อยู่ภายนอกองค์กรของคุณสำหรับส่งเมลถึงผู้ใช้ของคุณ นั่นคือ ถ้าไม่มีโดเมนเมลเซิร์ฟเวอร์ แอดเดรสของเมลต้องระบุโฮสต์ในองค์กรของคุณ

ตัวอย่างเช่น `sam@orange.widget.com` โดยที่ `widget.com` เป็นชื่อโดเมนขององค์กรของคุณ และ `orange` เป็นโฮสต์ที่ `sam` ใช้ แต่โดยการใช้โดเมนเซิร์ฟเวอร์ ผู้ใช้ภายนอกองค์กรของคุณสามารถระบุชื่อผู้ใช้และชื่อโดเมน โดยไม่ต้องรู้จักว่าผู้ใช้ใช้โฮสต์ใด ตัวอย่างเช่น `sam@widget.com`

เมื่อต้องการกำหนดค่าโดเมน เมลเซิร์ฟเวอร์ใช้โพรซีเดอร์ต่อไปนี้

1. สร้างเร็กคอร์ดของ mail exchanger (MX) และเร็กคอร์ดของ address (A) สำหรับเมลเซิร์ฟเวอร์ `black.widget.com`:

   ```
   widget.com           IN    MX     10 black.widget.com
   widget.com           IN     A     192.10.143.9
   black.widget.com     IN     A     192.10.143.9
   ```

2. แก้ไข `sendmail.cf` บนเมลเซิร์ฟเวอร์ (`black.widget.com`) เพื่อเพิ่ม alias ของโดเมน (คลาส **w**):

   ```
   Cw $w $?D$w.$D$. widget.com
   ```

3. เมลไคลเอ็นต์ต้องรู้ว่าจะส่งเมลแบบไม่ใช่โลคัลของตนไปที่ไหน ดังนั้นแก้ไข `sendmail.cf` บนแต่ละไคลเอ็นต์เพื่อชี้ไปยังเมลเซิร์ฟเวอร์ (แมโคร **S**):

   ```
   DRblack.widget.com
   ```

4. ใช้อ็อพชัน **NameServOpt** เพื่อกำหนดคอนฟิก **sendmail** daemon เพื่อที่ทุกคนสามารถใช้เร็กคอร์ดของ MX ที่ถูกกำหนดในเนมเซิร์ฟเวอร์ `brown.widget.com`
5. เพิ่ม alias สำหรับผู้ใช้ในโดเมนที่ไม่มีแอคเคาต์ผู้ใช้บนเมลเซิร์ฟเวอร์โดยใช้ไฟล์ aliases ตัวอย่างเช่น :

```
sam:sam@orange.widget.com
david:david@green.widget.com
judy:judy@red.widget.com
```

หมายเหตุ: เร็กคอร์ดของ Mailbox (MB) สามารถทำหน้าที่เดียวกัน

6. หมายเลขลำดับใน SOA Resource Record ต้องถูกเพิ่มขึ้น เนื่องจากฐานข้อมูลถูกแก้ไข
7. รีเฟรชฐานข้อมูลของเนมเซิร์ฟเวอร์โดยใช้คำสั่ง `refresh -s named`
8. บนไคลเอ็นต์ รันคำสั่ง `refresh -s sendmail` เพื่อทำให้การเปลี่ยนแปลงมีผลใช้งาน

มีวิธีอื่นเพื่อตั้งค่าโดเมนเมลเซิร์ฟเวอร์โพรซีเดอร์นี้เกี่ยวข้องกับการใช้เร็กคอร์ดของ mailbox (MB), mail rename (MR) และ mail group (MG)

การตั้งค่าโดเมนของเมลเซิร์ฟเวอร์โดยใช้เร็กคอร์ดของ mailbox:

ใช้โพรซีเดอร์ต่อไปนี้เพื่อตั้งค่าโดเมนเมลเซิร์ฟเวอร์โดยใช้เร็กคอร์ดของ mailbox

1. กำหนดเร็กคอร์ดของ mailbox (MB) สำหรับแต่ละผู้ใช้ในโดเมน เพิ่ม entry เช่น :

   ```
   sam IN MB orange.widget.com.
   ```

   เข้ากับไฟล์ `/usr/local/domain/named.abc.data` บนโฮสต์ `brown.widget.com` entry เหล่านี้จะระบุกับเมลเซิร์ฟเวอร์ `black.widget.com` ที่จะส่งเมลสำหรับแต่ละผู้ใช้ในโดเมน
2. ตั้งค่า **sendmail** daemon บนเมลเซิร์ฟเวอร์ `black.widget.com` เพื่อใช้เร็กคอร์ด MB ที่กำหนดในเนมเซิร์ฟเวอร์ `brown.widget.com` ใช้อ็อพชัน **NameServOpt**
3. เพิ่มหมายเลขลำดับใน SOA Resource Record เนื่องจากฐานข้อมูลถูกแก้ไข
4. รีเฟรชฐานข้อมูลของเนมเซิร์ฟเวอร์โดยการรัน `refresh -s named`
5. พิมพ์คำสั่ง `refresh -s sendmail` เพื่อทำให้การเปลี่ยนแปลงมีผลใช้งาน

การกำหนดเร็กคอร์ดของ mail rename สำหรับผู้ใช้:

ใช้โพรซีเดอร์นี้เพื่อกำหนดเร็กคอร์ดของ mail rename

1. แก้ไขไฟล์ `/usr/local/domain/named.abc.data` บนโดเมนเมลเซิร์ฟเวอร์ของคุณ
2. เพิ่มเร็กคอร์ดของ Mail Rename (MR) สำหรับแต่ละ alias ตัวอย่างเช่น ถ้าผู้ใช้ `sam` มี alias `sammy` เร็กคอร์ดของ Mail Rename คือ :

   ```
   sammy IN MR sam
   ```

   เร็กคอร์ดนี้จะทำให้เมลทั้งหมดที่มีแอดเดรสไปยัง `sammy` จะถูกส่งไปยัง `sam` แต่ละเร็กคอร์ดของ MR ควรถูกใส่บนบรรทัดของมันเอง
3. หมายเลขลำดับใน SOA Resource Record ต้องถูกเพิ่มขึ้น เนื่องจากฐานข้อมูลถูกแก้ไข
4. รีเฟรชฐานข้อมูลของเนมเซิร์ฟเวอร์โดยการพิมพ์คำสั่ง `refresh -s named`
5. พิมพ์คำสั่ง `refresh -s sendmail` เพื่อทำให้การเปลี่ยนแปลงมีผลใช้งาน

การกำหนดเร็กคอร์ดของสมาชิกของ mail group:

ใช้โพรซีเดอร์ต่อไปนี้เพื่อกำหนดเร็กคอร์ดของสมาชิกของ mail group

1. แก้ไขไฟล์ `/usr/local/domain/named.abc.data` บนโดเมนเมลเซิร์ฟเวอร์ของคุณ

2. เพิ่มเร็กคอร์ด MG สำหรับแต่ละ mail group (MG) เร็กคอร์ดของ MG ทำงานเหมือนกับไฟล์ /etc/aliases โดยที่ aliases ถูกเก็บอยู่บนเนมเซิร์ฟเวอร์ ตัวอย่าง เช่น:

   ```
   users IN HINFO users-request widget.com
   users IN MG sam
   users IN MG david
   users IN MG judy
   ```

   ตัวอย่างนี้ทำให้เมลทั้งหมดที่มีแอดเดรสไปยัง users@widget.com ถูกส่งไปยัง sam, david และ judy ใส่แต่ละเร็กคอร์ด MG บนบรรทัดโดยมันเอง

   หมายเหตุ: ผู้ใช้ sam, david และ judy ต้องถูกกำหนดเร็กคอร์ดของ MB
3. หมายเลขลำดับใน SOA Resource Record ต้องถูกเพิ่มขึ้น เนื่องจากฐานข้อมูลถูกแก้ไข
4. รีเฟรชฐานข้อมูลของเนมเซิร์ฟเวอร์โดยการพิมพ์คำสั่ง refresh -s named
5. พิมพ์คำสั่ง refresh -s sendmail เพื่อทำให้การเปลี่ยนแปลงมีผลใช้งาน

การกำหนดเร็กคอร์ดของ mail exchanger:

ใช้โพรซีเดอร์ต่อไปนี้เพื่อกำหนดเร็กคอร์ดของ mail exchanger

1. แก้ไขไฟล์ /usr/local/domain/named.abc.data บนโดเมนเมลเซิร์ฟเวอร์ของคุณ
2. เพิ่มเร็กคอร์ดของ mail exchanger (MX) สำหรับแต่ละเครื่องที่ไม่ได้เชื่อมต่อโดยตรงกับเน็ตเวิร์กของคุณที่คุณต้องการฟอร์เวิร์ดเมล ตัวอย่างเช่น ถ้าเมลแอดเดรสไปยังผู้ใช้บน purple.widget.com ควรถูกฟอร์เวิร์ดไปยัง post.office.widget เร็กคอร์ด MX จะมีลักษณะเหมือนต่อไปนี้ :

   ```
   purple.widget.com IN MX 0 post.office.widget.
   ```

   คุณต้องระบุทั้งชื่อของโฮสต์และเครื่องเมื่อใช้เร็กคอร์ด MX ใส่แต่ละเร็กคอร์ด MG บนบรรทัดโดยมันเอง คุณสามารถใช้ wildcards ตัวอย่างเช่น:

   ```
   *.widget.com IN MX 0 post.office.widget.
   ```

   ตัวอย่างนี้ทำให้เมลที่ไปยังโฮสต์ที่ไม่รู้จัก (โฮสต์ที่ไม่ระบุเร็กคอร์ด MX) ในโดเมน widget.com จะถูกฟอร์เวิร์ดไปยัง post.office.widget.

   หมายเหตุ: เร็กคอร์ด MX แบบ wildcard จะไม่เหมาะที่จะใช้บนอินเตอร์เน็ต
3. หมายเลขลำดับใน SOA Resource Record ต้องถูกเพิ่มขึ้น เนื่องจากฐานข้อมูลถูกแก้ไข
4. รีเฟรชฐานข้อมูลของเนมเซิร์ฟเวอร์โดยการพิมพ์คำสั่ง refresh -s named
5. พิมพ์คำสั่ง refresh -s sendmail เพื่อทำให้การเปลี่ยนแปลงมีผลใช้งาน

การตั้งค่า forwarder

เมื่อต้องการกำหนดคอนฟิกเซิร์ฟเวอร์ตัวฟอร์เวิร์ด ใช้โพรซีเดอร์นี้ ซึ่งแก้ไขแก้ไขของไฟล์ จากนั้นใช้ SMIT หรือบรรทัดรับคำสั่ง เพื่อเริ่มทำงาน **named** daemon

1. แก้ไขไฟล์ /etc/named.conf ถ้าไม่มีไฟล์ named.conf fในไดเร็กทอรี /etc คัดลอกไฟล์ /usr/samples/tcpip/named.conf ตัวอย่างไปยังไดเร็กทอรี /etc และแก้ไขมัน ดูที่ "named.conf File Format for TCP/IP" ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติมและตัวอย่างแบบละเอียดของไฟล์ conf
   - ระบุบรรทัด forwarders ในอ็อพชัน stanza ของไฟล์ /etc/named.conf ที่ลิสต์ IP แอดเดรสของเนมเซิร์ฟเวอร์ที่ควรได้รับคำร้องขอ forwarded ตัวอย่างเช่น:

```
options {
    ...
    directory "/usr/local/domain";
    forwarders { 192.100.61.1; 129.35.128.222; };
    ...
};
```

- ระบุโซนของ loopback ตัวอย่างเช่น:

```
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "named.abc.local";
};
```

- ระบุ hint โซน ตัวอย่างเช่น:

```
zone "." IN {
    type hint;
    file "named.ca";
};
```

2. แก้ไขไฟล์ /usr/local/domain/named.ca ดูที่ "DOMAIN Cache File Format for TCP/IP" ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติมและตัวอย่างแบบละเอียดของไฟล์ cache

   ไฟล์นี้ มีแอดเดรสของเซิร์ฟเวอร์ที่เป็นเนมเซิร์ฟเวอร์ซึ่งได้รับอนุญาต สำหรับโดเมน root ของเครือข่าย ตัวอย่าง เช่น:

```
; root name servers.
.          IN    NS    relay.century.com.
relay.century.com.   3600000    IN    A     129.114.1.2
```

   หมายเหตุ: ทุกบรรทัดในไฟล์นี้ต้องอยู่ใน Standard Resource Record Format

3. แก้ไขไฟล์ /usr/local/domain/named.abc.local ดูที่ DOMAIN Local Data File Format for TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติมและตัวอย่าง แบบละเอียด ของไฟล์ข้อมูลโลคัล

   a. ระบุ start of authority (SOA) ของโซนและข้อมูล time-to-live ดีฟอลต์ ตัวอย่าง เช่น:

```
$TTL 3h    ;3 hour

@ IN SOA venus.abc.aus.century.com. gail.zeus.abc.aus.century.com.  (

                            1       ;serial
                            3600    ;refresh
                            600     ;retry
                            3600000 ;expire
                            86400   ;negative caching TTL
)
```

   b. ระบุเร็กคอร์ดของเนมเซิร์ฟเวอร์ (NS) ตัวอย่าง เช่น:

```
<tab> IN    NS     venus.abc.aus.century.com.
```

   c. ระบุเร็กคอร์ด pointer (PTR)

```
1      IN    PTR    localhost.
```

   หมายเหตุ: ทุกบรรทัดในไฟล์นี้ต้องอยู่ใน Standard Resource Record Format

4. สร้างไฟล์ /etc/resolv.conf โดยการพิมพ์คำสั่ง ต่อไปนี้:

```
touch /etc/resolv.conf
```

ถ้าไฟล์นี้มีอยู่ แสดงว่าโฮสต์ควรใช้เนมเซิร์ฟเวอร์ ไม่ใช่ไฟล์ /etc/hosts สำหรับการแปลงชื่อ

นอกจากนี้ ไฟล์ /etc/resolv.conf อาจประกอบด้วย entry ต่อไปนี้ :

```
nameserver 127.0.0.1
```

แอดเดรส 127.0.0.1 คือ loopback แอดเดรส ซึ่งทำให้โฮสต์เข้าถึงตัวเอง เป็นเนมเซิร์ฟเวอร์ ไฟล์ /etc/resolv.conf ยังอาจประกอบด้วย entry เหมือนดังต่อไปนี้ :

```
domain domainname
```

ในตัวอย่างก่อนหน้านี้ ค่า *domainname* คือ austin.century.com

5. ทำหนึ่งในขั้นตอนต่อไปนี้ :
   - เปิดใช้งาน **named** daemon โดยใช้ smit stnamed SMIT fast path พาธด่วนนี้จะเริ่มต้น daemon ด้วยสตาร์ทอัพระบบแต่ละรายการ บ่งชี้ว่าคุณ ต้องการเริ่มต้น **named** daemon เดี๋ยวนี้ หรือในการรีสตาร์ทระบบ ครั้งถัดไป หรือทั้งสองอย่าง
   - แก้ไขไฟล์ /etc/rc.tcpip ยกเลิกหมายเหตุบรรทัดสำหรับ **named** daemon โดยการลบเครื่องหมายหมายเหตุ (#) จากบรรทัดต่อไปนี้ :

     ```
     #start /etc/named "$src_running"
     ```

     พาธด่วนนี้จะเริ่มต้น daemon ด้วยสตาร์ทอัพระบบแต่ละรายการ
6. ถ้าคุณเลือกที่จะเริ่มต้น named daemon ผ่าน SMIT สตาร์ท daemon สำหรับเซสชันนี้โดยการพิมพ์คำสั่งต่อไปนี้ :

   ```
   startsrc -s named
   ```

การกำหนดคอนฟิกการฟอร์เวิร์ดเฉพาะเนมเซิร์ฟเวอร์

เมื่อต้องการกำหนดคอนฟิกฟอร์เวิร์ดเฉพาะเนมเซิร์ฟเวอณี ใช้โพรซีเดอร์นี้ ซึ่งแก้ไขแก้ไขของไฟล์ จากนั้นใช้ SMIT หรือบรรทัดรับคำสั่ง เพื่อเริ่มทำงาน **named** daemon

หมายเหตุ: คุณสามารถทำการตั้งค่าที่เหมือนกันโดยไม่ต้องรันการฟอร์เวิร์ดเนมเซิร์ฟเวอร์เฉพาะ โดยสร้างไฟล์ /etc/resolv.conf ที่ประกอบด้วยบรรทัดของเนมเซิร์ฟเวอร์ที่ชี้ไปยัง forwarders ที่คุณต้องการใช้

1. แก้ไขไฟล์ /etc/named.conf ถ้าไม่มีไฟล์ named.conf fในไดเร็กทอรี /etc คัดลอกไฟล์ /usr/samples/tcpip/named.conf ตัวอย่างไปยังไดเร็กทอรี /etc และแก้ไขมัน ดูที่ named.conf File Format for TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติมและตัวอย่างแบบละเอียดของไฟล์ conf
   - ระบุบรรทัด forwarders และ forward เท่านั้นในอ็อพชัน stanza ของไฟล์ /etc/named.conf ที่ลิสต์ IP แอดเดรสของเนมเซิร์ฟเวอร์ที่รับคำร้องขอ forwarded ตัวอย่างเช่น:

     ```
     options {
         ...
         directory "/usr/local/domain";
         forwarders { 192.100.61.1;  129.35.128.222; };
         forward only;
         ...
     };
     ```

   - ระบุโซนของ loopback ตัวอย่างเช่น:

     ```
     zone "0.0.127.in-addr.arpa" in {
         type master;
         file "named.abc.local";
     };
     ```

   - ระบุ hint โซน ตัวอย่างเช่น:

```
zone "." IN {
    type hint;
    file "named.ca";
};
```

2. แก้ไขไฟล์ /usr/local/domain/named.ca ตัวอย่าง เช่น: ดูที่ DOMAIN Cache File Format for TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติมและตัวอย่าง แบบละเอียด ของไฟล์แคช ไฟล์นี้ มีแอดเดรสของเซิร์ฟเวอร์ที่เป็นเนมเซิร์ฟเวอร์ซึ่งได้รับอนุญาต สำหรับโดเมน root ของเครือข่าย

```
; root name servers.
.          IN    NS    relay.century.com.
relay.century.com.   3600000    IN    A     129.114.1.2
```

หมายเหตุ: ทุกบรรทัดในไฟล์นี้ต้องอยู่ใน Standard Resource Record Format

3. แก้ไขไฟล์ /usr/local/domain/named.abc.local ดูที่ DOMAIN Local Data File Format for TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติมและตัวอย่าง แบบละเอียด ของไฟล์ข้อมูลโลคัล

   a. ระบุ start of authority (SOA) ของโซนและข้อมูล time-to-live ดีฟอลต์ ตัวอย่างเช่น:

   ```
   $TTL 3h    ;3 hour

   @ IN SOA venus.abc.aus.century.com. gail.zeus.abc.aus.century.com.  (

                              1       ;serial
                              3600    ;refresh
                              600     ;retry
                              3600000 ;expire
                              86400   ;negative caching TTL
   )
   ```

   b. ระบุเร็กคอร์ดของเนมเซิร์ฟเวอร์ (NS) ตัวอย่างเช่น:

   ```
   <tab> IN    NS     venus.abc.aus.century.com.
   ```

   c. ระบุเร็กคอร์ด pointer (PTR)

   ```
   1      IN    PTR    localhost.
   ```

   หมายเหตุ: ทุกบรรทัดในไฟล์นี้ต้องอยู่ใน Standard Resource Record Format

4. สร้างไฟล์ /etc/resolv.conf โดยการพิมพ์คำสั่ง ต่อไปนี้:

   ```
   touch /etc/resolv.conf
   ```

   ถ้าไฟล์นี้มีอยู่ แสดงว่าโฮสต์ควรใช้โฮสต์ ไม่ใช่ไฟล์ /etc/hosts สำหรับการแปลงชื่อ

   นอกจากนี้ ไฟล์ /etc/resolv.conf อาจประกอบด้วย entry ต่อไปนี้ :

   ```
   nameserver 127.0.0.1
   ```

   แอดเดรส 127.0.0.1 คือ loopback แอดเดรส ซึ่งทำให้โฮสต์เข้าถึงตัวเอง เป็นเนมเซิร์ฟเวอร์ ไฟล์ /etc/resolv.conf ยังสามารถประกอบด้วย entry เช่น:

   ```
   domain domainname
   ```

   ในตัวอย่างก่อนหน้านี้ ค่า *domainname* คือ austin.century.com

5. ทำหนึ่งในขั้นตอนต่อไปนี้ :

- เปิดใช้งาน **named** daemon โดยใช้ `smit stnamed` SMIT fast path พาธด่วนนี้จะเริ่มต้น daemon ด้วยสตาร์ทอัพระบบแต่ละรายการ บ่งชี้ว่าคุณ ต้องการเริ่มต้น **named** daemon เดี๋ยวนี้ หรือในการรีสตาร์ทระบบ ครั้งถัดไป หรือทั้งสองอย่าง
- แก้ไขไฟล์ `/etc/rc.tcpip` ยกเลิกหมายเหตุบรรทัดสำหรับ **named** daemon โดยการลบเครื่องหมายหมายเหตุ (`#`) จากบรรทัดต่อไปนี้ :

  ```
  #start /etc/named "$src_running"
  ```

  พาธด่วนนี้จะเริ่มต้น daemon ด้วยสตาร์ทอัพระบบแต่ละรายการ

6. ถ้าคุณเลือกที่จะไม่เริ่มต้น **named** daemon ผ่าน SMIT สตาร์ท daemon สำหรับเซสชันนี้โดยการพิมพ์คำสั่งต่อไปนี้ :

   ```
   startsrc -s named
   ```

การกำหนดคอนฟิกโฮสต์เพื่อใช้เนมเซิร์ฟเวอร์

เมื่อต้องการกำหนดค่าโฮสต์เพื่อใช้เนมเซิร์ฟเวอร์ ใช้โพรซีเดอร์นี้

1. สร้างไฟล์ `/etc/resolv.conf` โดยการรันคำสั่ง ต่อไปนี้:

   ```
   touch /etc/resolv.conf
   ```

2. บนบรรทัดแรกของไฟล์ `/etc/resolv.conf` พิมพ์คำว่า `domain` ตามด้วยชื่อแบบเต็มของโดเมนที่โฮสต์นี้อยู่ ตัวอย่าง เช่น:

   ```
   domain abc.aus.century.com
   ```

3. บนบรรทัดว่างใดๆ ด้านล่างของบรรทัด `domain` พิมพ์คำว่า `nameserver` ตามด้วยช่องว่างอย่างน้อยหนึ่งช่องว่าง ตามด้วยอินเตอร์เน็ตแอดเดรสแบบ dotted decimal ของเนมเซิร์ฟเวอร์ที่โฮสต์นี้จะใช้ (เนมเซิร์ฟเวอร์ต้องให้บริการกับโดเมนที่ถูกระบุโดยข้อความ `domain`) คุณสามารถมีเนมเซิร์ฟเวอร์ได้ถึง 3 ตัว ตัวอย่างเช่น ไฟล์ `/etc/resolv.conf` ของคุณอาจประกอบด้วย entry:

   ```
   nameserver 192.9.201.1
   nameserver 192.9.201.2
   ```

   ระบบจะเคียวรีเนมเซิร์ฟเวอร์ที่ถูกลิสต์แบบลำดับ

   ```
   search domainname_list
   ```

   นอกจากนี้ การค้นหาคีย์เวิร์ดสามารถถูกใช้เพื่อที่ resolver จะเคียวรีลิสต์ของโดเมน ในกรณีนี้ค่า `domainname_list` คือ `abc.aus.century.com` และ `aus.century.com` `domainname_list` สามารถเป็นสตริงอักขระที่มีความยาวสูงสุด 1024 อักขระ โดยแต่ละตัวถูกคั่นด้วยช่องว่าง

4. สมมุติว่าเนมเซิร์ฟเวอร์เป็นทำงานได้ คุณสามารถทดสอบการสื่อสารระหว่างโฮสต์และเนมเซิร์ฟเวอร์โดยการพิมพ์คำสั่งต่อไปนี้ :

   ```
   host hostname
   ```

   ใช้ชื่อของโฮสต์ที่ควรถูกหามาโดยเนมเซิร์ฟเวอร์เพื่อดูว่ากระบวนการทำงานหรือไม่ เอาต์พุตที่ได้ควรมีลักษณะเหมือนต่อไปนี้ :

   ```
   brown.abc.aus.century.com is 129.35.145.95
   ```

งานการตั้งค่าอื่นถูกแสดงในตารางต่อไปนี้

ตารางที่ 64. งานการตั้งค่าโฮสต์เพื่อใช้เนมเซิร์ฟเวอร์

งาน	วิธีลัด SMIT	คำสั่งหรือไฟล์
สร้างไฟล์ /etc/resolv.conf	smit stnamerslv2	สร้างและแก้ไข /etc/resolv.conf[1]
แสดงรายชื่อเนมเซิร์ฟเวอร์ทั้งหมดที่ใช้งานโดยโฮสต์	smit lsnamerslv	ดู /etc/resolv.conf
เพิ่มเนมเซิร์ฟเวอร์	smit mknamerslv	แก้ไข /etc/resolv.conf[2]
ลบเนมเซิร์ฟเวอร์	smit rmnamerslv	แก้ไข /etc/resolv.conf
เริ่ม/รีสตาร์ทการใช้การแก้ไขชื่อโดเมน	smit stnamerslv	
หยุดใช้งานการแก้ไขชื่อโดเมน	smit spnamerslv	
เปลี่ยน/แสดง โดเมน	smit mkdomain	แก้ไข /etc/resolv.conf
ลบโดเมน	smit rmdomain	แก้ไข /etc/resolv.conf

ข้อมูลที่เกี่ยวข้อง

ไฟล์ netsvc.conf

ไดนามิกโซนบนเนมเซิร์ฟเวอร์ DNS

คำสั่ง **named** ใช้สำหรับการอัพเดต แบบไดนามิก ฐานข้อมูล named และไฟล์คอนฟิกูเรชันต้องถูกตั้งค่า เพื่อยอมให้เครื่องไคลเอ็นต์ส่งการอัพเดต โซนสามารถถูกตั้งเป็น แบบไดนามิกหรือสแตติก โซนแบบดีฟอลต์คือสแตติก

เมื่อต้องการทำให้โซนเป็นแบบไดนามิก คุณต้องเพิ่มคีย์เวิร์ด **allow-update** เข้ากับ stanza ของโซนในไฟล์ /etc/named.conf คีย์เวิร์ด **allow-update** จะระบุลิสต์ที่ตรงของอินเตอร์เน็ตแอดเดรส ที่ระบุกฎที่ถูกอนุญาตให้ส่งอัพเดต โปรดดูที่ named.conf File Format for TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติม และตัวอย่างแบบละเอียดของไฟล์ conf ในตัวอย่างต่อไปนี้ โฮสต์ทั้งหมดถูกอนุญาตให้อัพเดตไดนามิกโซน:

```
zone "abc.aus.century.com" IN {
    type master;
    file "named.abc.data";
    allow-update { any; };
};
```

หลังจากที่โซนถูกทำเครื่องหมายว่าเป็นไดนามิก ความปลอดภัย 3 โหมดสามารถ เริ่มต้น:

ไอเท็ม	คำอธิบาย
UNsecured	อนุญาตให้ทุกคนอัพเดตข้อมูลใดๆในโซนที่ทุกเวลา **ข้อควรสนใจ:** ไม่แนะนำให้ใช้โหมดนี้ ซึ่งสามารถ เป็นสาเหตุให้ข้อมูลหาย ข้อมูลถูกดัก และการทำร้ายผู้ใช้ สุดท้าย โซนที่ไม่ปลอดภัยควรถูกจำกัดเพื่ออัพเดตเฉพาะอินเตอร์เน็ตแอดเดรส ที่ระบุ
Conntrolled	อนุญาตสำหรับการสร้างข้อมูลใหม่และการแทนที่ ข้อมูลที่มีอยู่แล้ว นี่อาจเป็นโหมดที่ง่ายที่สุดที่ใช้ สำหรับสภาวะแวดล้อมที่มีการเปลี่ยนแปลงความปลอดภัย โหมดนี้ยังต้องการให้อัพเดต ที่เข้ามาทั้งหมดถูกประทับเวลาและมีคีย์การลงชื่อ
Presecured	ต้องการให้อัพเดตทั้งหมดกับข้อมูลที่มีอยู่ถูกแทนที่ ด้วยข้อมูลที่เหมือนกัน ไม่อนุญาตสำหรับการสร้างข้อมูลใหม่ โหมดนี้ยังต้องการให้อัพเดตที่เข้ามาทั้งหมดถูกประทับเวลา และมีคีย์การลงชื่อ

ไดนามิกโซนแบบดีฟอลต์กับโหมดแบบไม่ปลอดภัย เพื่อใช้หนึ่งในโหมดอื่นๆ พิมพ์ controlled หรือ presecured หลังคีย์เวิร์ด **update-security** ใน stanza ของโซนของไฟล์ /etc/named.conf นี่จะบอก **named** เซิร์ฟเวอร์ถึงระดับของความปลอดภัยที่ใช้ กับโซนนั้น ตัวอย่างเช่น:

```
zone "abc.aus.century.com" IN {
    type master;
    file "named.abc.data";
    allow-update { any; };
    update-security controlled;
};
```

หลังจากโหมดถูกเลือก ไฟล์ข้อมูลจริงๆต้องถูกแก้ไข สำหรับระดับของความปลอดภัยของคุณ ในโหมด unsecured ไฟล์ข้อมูลถูก ใช้"ตามที่เป็นอยู่" สำหรับโหมด controlled หรือ presecured คุณต้องสร้าง ชุดของคู่ของคีย์เซิร์ฟเวอร์หลัก/ชื่อโฮสต์ สำหรับแต่ละชื่อในโซน ซึ่งทำได้โดยใช้คำสั่ง **nsupdate** โดยใช้อ็อพชัน **-g** คำสั่งนี้จะสร้างคู่ของคีย์ (ไพรเวตคีย์และพับลิกคีย์) คีย์เหล่านี้ถูกต้องการเพื่อพิสูจน์การลงชื่อสำหรับอัพเดต หลังจากสร้าง คีย์ทั้งหมดสำหรับลิสต์ของชื่อโซนของคุณ คุณต้องเพิ่มมันเข้ากับ ไฟล์ข้อมูล รูปแบบของ KEY เป็นดังต่อไปนี้ :

Index ttl Class Type KeyFlags Protocol Algorithm KeyData

โดยที่:

ไอเท็ม	คำอธิบาย
Index	ระบุชื่อที่ถูกใช้เพื่ออ้างอิงข้อมูลในโซน
ttl	ระบุ time-to-live (TTL) สำหรับข้อมูลนี้ นี่เป็นฟิลด์ ที่เป็นอ็อพชัน
Class	ระบุคลาสของข้อมูล นี่จะขึ้นอยู่กับโซน แต่โดยทั่วไปจะเป็น IN
Type	ระบุชนิดของเร็กคอร์ด ในกรณีนี้ จะเป็น KEY
KeyFlags	ให้ข้อมูล named เกี่ยวกับคีย์ 0x0000 จะระบุ เร็กคอร์ดของคีย์ทั่วไปสำหรับโฮสต์ 0x0100 จะระบุเร็กคอร์ดของคีย์ที่เกี่ยวข้อง กับชื่อโซน
Protocol	ระบุโปรโตคอลที่ใช้ ปัจจุบันมีเฉพาะ 0
Algorithm	ระบุอัลกอริทึมของคีย์ ปัจจุบันมีเฉพาะ 1 นี่คือวิธีการพิสูจน์ตัวตน MD5 Private/Public
KeyData	ระบุคีย์ในการแทนแบบ base64 คำสั่ง **nsupdate** จะสร้างทั้งพับลิกคีย์และไพรเวตคีย์ในการแทนแบบ base64 พับลิกคีย์จะถูกลิสต์สุดท้ายในเอาต์พุตไฟล์

ตัวอย่างเช่น เพื่อให้แน่ใจถึงความปลอดภัยบนชื่อโฮสต์ในไดนามิกโซน บรรทัดที่เหมือนกับต่อไปนี้ต้องถูกเพิ่มเข้ากับไฟล์โซน สำหรับโซนที่ประกอบด้วยชื่อโฮสต์

```
bears    4660    IN    KEY    0x0000    0    1    AQOtg......
```

ตัวอย่างก่อนหน้านี้ระบุว่า bears มีการกำหนดเร็กคอร์ด KEY บางคนต้องการอัพเดต bears ต้องลงชื่ออัพเดตด้วยไพรเวตคีย์ที่ตรงกับพับลิกคีย์ ในฐานข้อมูล เพื่อจะให้คำสั่ง **nsupdate** ทำสำเร็จ ไพรเวตคีย์ต้องถูกเก็บไว้บนไคลเอ็นต์ในคีย์ไฟล์ (ดีฟอลต์ คือ /etc/keyfile) ซึ่งควรจะมีรูปแบบ :

hostname mastername base64 key

entry ของ KEY ที่เหมือนกันถูกต้องการในส่วนของคำจำกัดความของโซน *คีย์ของโซน จำเป็นสำหรับทั้งโหมด presecured และ controlled ไม่เช่นนั้นจะถูกพิจารณาว่า เป็นโหมด unsecured* นี่สามารถทำได้ดังแสดง ในตัวอย่างของ bears ก่อนหน้านี้ แต่ไพรเวตคีย์ จะมีให้ผู้ดูแลระบบใช้กับโหมด administrative ของคำสั่ง **nsupdate**

1. เพื่อสร้างคู่ของคีย์โดยใช้คำสั่ง **nsupdate** พิมพ์ต่อไปนี้:

   ```
   nsupdate -g -h ZoneName -p ServerName -k AdminKeyFile
   ```

   นี่จะ สร้างคีย์สำหรับโซน ในตัวอย่างนี้ **nsupdate** ถูกเชื่อมโยงกับ **nsupdate4** ซึ่งสามารถทำได้โดยการพิมพ์ ต่อไปนี้:

```
ln -fs /usr/sbin/nsupdate4 /usr/sbin/nsupdate
```

2. วางคู่ของคีย์ล่าสุดในส่วนเริ่มต้น สำหรับโซนดังต่อไปนี้:

```
IN     KEY     0x0100  0  1  Key
```

entry สำหรับไฟล์ named.abc.data จะเป็นดังต่อไปนี้:

```
$TTL 3h    ;3 hour

@ IN    SOA     venus.abc.aus.century.com. gail.zeus.abc.aus.century.com. (
                1       ;serial
                3600    ;refresh
                600     ;retry
                3600000 ;expire
                86400   ;negative caching TTL
)
        IN      NS      venus.abc.aus.century.com.
        IN      KEY     0x0100  0 1 AQPlwHmIQeZzRk6Q/nQYhs3xwnhfTgF/
                        8YlBVzKSoKxVKPNLINnYWOmB7attTcfhHaZZcZr4u/
                        vDNikKnhnZwgn/
venus         IN    A         192.9.201.1
earth         IN    A         192.9.201.5
mars          IN    A         192.9.201.3
```

3. ตอนนี้โซนจะพร้อมที่จะถูกโหลดโดยการรีเฟรชเนมเซิร์ฟเวอร์ วาง AdminKeyFile บนไคลเอ็นต์หรือ DHCP เซิร์ฟเวอร์ที่อัพเดตโซน คีย์ของโซนใน AdminKeyFile สามารถถูกใช้เพื่อนำอัพเดตไปใช้และการทำการบำรุงรักษาเนมเซิร์ฟเวอร์

ความปลอดภัยของ BIND 9

BIND 9 ให้ Transaction Signatures (TSIG) และ Signatures (SIG) เป็นการวัดความปลอดภัยสำหรับ **named**

โดยดีฟอลต์เนมเซิร์ฟเวอร์ที่มี BIND 9 ไม่ยอมให้มีการอัพเดตแบบไดนามิกกับ authoritative โซน เหมือนกับ BIND 8

BIND 9 ส่วนใหญ่สนับสนุน Transaction Signatures (TSIG) สำหรับการสื่อสารแบบเซิร์ฟเวอร์ถึงเซิร์ฟเวอร์ นี่จะรวมถึงการถ่ายโอนโซน การแจ้งเตือน และการส่งข้อความเคียวรีซ้ำ TSIG ยังมีประโยชน์สำหรับการอัพเดตแบบไดนามิก เซิร์ฟเวอร์ลำดับแรกสำหรับโซนแบบไดนามิกควรใช้แอ็คเซสคอนโทรลเพื่อควบคุมการอัพเดต แต่แอ็คเซสคอนโทรลแบบ IP-based ไม่เพียงพอ

โดยการใช้การเข้ารหัสแบบใช้คีย์แทนที่จะเป็นลิสต์ของแอ็คเซสคอนโทรลปัจจุบัน TSIG สามารถถูกใช้เพื่อจำกัดว่าใครสามารถอัพเดตกับโซนแบบไดนามิก ไม่เหมือนกับวิธี Access Control List (ACL) ของการอัพเดตแบบไดนามิก คีย์ของ TSIG สามารถถูกกระจายไปยังผู้อัพเดตอื่นโดยไม่ต้องแก้ไขไฟล์คอนฟิกูเรชันบนเนมเซิร์ฟเวอร์ ซึ่งหมายความว่าเนมเซิร์ฟเวอร์ที่จะต้องอ่านไฟล์คอนฟิกูเรชันใหม่

มีความจำเป็นที่ต้องสังเกตว่า BIND 9 ไม่ได้มีคีย์เวิร์ดที่ถูกนำมาใช้ใน BIND 8 ในตัวอย่างนี้ เราใช้การตั้งค่าตัวอย่างแบบง่ายจาก BIND 8

หมายเหตุ: เพื่อใช้ named 9 คุณต้องเชื่อมลิงก์สัญลักษณ์กับ **named** daemon กับ **named9** และ **nsupdate** กับ**nsupdate9** ใหม่โดยใช้คำสั่งต่อไปนี้ :

1. `ln -fs /usr/sbin/named9 /usr/sbin/named`
2. `ln -fs /usr/sbin/nsupdate9 /usr/sbin/nsupdate`

1. สร้างคีย์โดยใช้คำสั่ง **dnssec-keygen** :

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST keyname
```

- HMAC-MD5 เป็นอัลกอริทึมที่ใช้สำหรับการเข้ารหัส
- 128 เป็นความยาวของคีย์ที่ใช้ (หรือจำนวนของบิต)
- HOST: HOST เป็นคีย์เวิร์ด TSIG ที่ใช้เพื่อสร้างโฮสต์คีย์สำหรับคีย์การเข้ารหัสที่ถูกแบ่งใช้

คำสั่ง

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST venus-batman.abc.aus.century.com
```

จะสร้างคีย์ไฟล์ 2 ไฟล์ ดังต่อไปนี้:

```
Kvenus-batman.abc.aus.century.com.+157+35215.key
Kvenus-batman.abc.aus.century.com.+157+35215.private
```

- 157 เป็นอัลกอริทึมที่ใช้ (HMAC–MD5)
- 35215 เป็น finger print ซึ่งมีประโยชน์ใน DNNSEC เนื่องจากยอมให้ใช้ได้หลายคีย์ต่อโซน

2. เพิ่ม entry เข้ากับ named.conf บน เนมเซิร์ฟเวอร์หลัก:

```
// TSIG Key
key venus-batman.abc.aus.century.com. {
        algorithm hmac-md5;
        secret "+UWSvbpxHWFdNwEAdy1Ktw==";
};
```

สมมุติว่า HMAC–MD5 ถูกใช้ keyfile ทั้งสองประกอบด้วยคีย์ที่ถูกแบ่งใช้ ซึ่งถูกเก็บเป็น entry สุดท้ายในไฟล์ หาวิธีที่ปลอดภัยที่สุดเพื่อคัดลอกคีย์ลับที่ถูกแบ่งใช้ไปยังไคลเอ็นต์ คุณไม่ต้องคัดลอก keyfile แค่แบ่งใช้คีย์ลับ

ต่อไปนี้เป็น entry สำหรับไฟล์ Kvenus-batman.abc.aus.century.com.+157+35215.private:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: +UWSvbpxHWFdNwEAdy1Ktw==
```

ด้านล่างเป็นตัวอย่างของไฟล์ named.conf สำหรับ เนมเซิร์ฟเวอร์หลัก โซน abc.aus.century.com ยอมให้มีการถ่ายโอนโซน และอัพเดตแบบไดนามิกเฉพาะกับเซิร์ฟเวอร์ที่มีคีย์ venus-batman.abc.aus.century.com ทำอย่างเดียวกันกับโซนแบบ reverse ซึ่งผู้อัพเดตต้องมีคีย์ที่ถูกแบ่งใช้

```
// TSIG Key
key venus-batman.abc.aus.century.com. {
        algorithm hmac-md5;
        secret "+UWSvbpxHWFdNwEAdy1Ktw==";
};

options {
        directory "/usr/local/domain";
};

zone "abc.aus.century.com" in {
        type master;
        file "named.abc.data";
        allow-transfer { key venus-batman.abc.aus.century.com.;};
        allow-update{ key venus-batman.abc.aus.century.com.; };
};
```

เนื่องจากตอนนี้การถ่ายโอนโซนถูกจำกัดสำหรับผู้ที่มีคีย์ไฟล์ named.conf ของเนมเซิร์ฟเวอร์ย่อยต้องถูกแก้ไขด้วย คำร้องขอทั้งหมดที่ไปยัง 192.9.201.1 (venus.abc.aus.century.com) จะถูกลงชื่อโดยคีย์ บันทึกชื่อของคีย์ (venus-batman.abc.aus.century.com.) ต้องตรงคีย์เหล่านั้นบนเซิร์ฟเวอร์ที่ใช้มัน

ด้านล่างเป็นตัวอย่างของไฟล์ named.conf สำหรับเนมเซิร์ฟเวอร์ย่อย

```
// TSIG Key
key venus-batman.abc.aus.century.com. {
        algorithm hmac-md5;
        secret "+UWSvbpxHWFdNwEAdy1Ktw==";
};

server 192.9.201.1{
        keys { venus-batman.abc.aus.century.com.;};
};

options {
        directory "/usr/local/domain";
};

zone "abc.aus.century.com" IN {
    type slave;
    file "named.abc.data.bak";
    masters { 192.9.201.1; };
};
```

BIND 9 Transaction Signatures:

BIND 9 ส่วนใหญ่สนับสนุน Transaction Signatures (TSIG) สำหรับการสื่อสารแบบเซิร์ฟเวอร์ถึงเซิร์ฟเวอร์

นี่จะรวมถึงการถ่ายโอนโซน การแจ้งเตือน และการส่งข้อความเคียวรีซ้ำ TSIG ยังมีประโยชน์สำหรับการอัพเดตแบบไดนามิก เซิร์ฟเวอร์ลำดับแรกสำหรับโซนแบบไดนามิกควรใช้แอ็คเซสคอนโทรลเพื่อควบคุมการอัพเดต แต่แอ็คเซสคอนโทรลแบบ IP-based ไม่เพียงพอ

โดยการใช้การเข้ารหัสแบบใช้คีย์แทนที่จะเป็นลิสต์ของแอ็คเซสคอนโทรลปัจจุบัน TSIG สามารถถูกใช้เพื่อจำกัดว่าใครสามารถอัพเดตกับโซนแบบไดนามิก ไม่เหมือนกับวิธี Access Control List (ACL) ของการอัพเดตแบบไดนามิก คีย์ของ TSIG สามารถถูกกระจายไปยังผู้อัพเดตอื่นโดยไม่ต้องแก้ไขไฟล์คอนฟิกูเรชันบนเนมเซิร์ฟเวอร์ ซึ่งหมายความว่าเนมเซิร์ฟเวอร์ที่จะต้องอ่านไฟล์คอนฟิกูเรชันใหม่

มีความจำเป็นที่ต้องสังเกตว่า BIND 9 ไม่ได้มีคีย์เวิร์ดที่ถูกนำมาใช้ใน BIND 8 ในตัวอย่างนี้ เราใช้การตั้งค่าตัวอย่างแบบง่ายจาก BIND 8

หมายเหตุ: เพื่อใช้ named 9 คุณต้องเชื่อมลิงก์สัญลักษณ์กับ **named** daemon กับ **named9** และ **nsupdate** กับ**nsupdate9** ใหม่ โดยใช้คำสั่งต่อไปนี้:

1. `ln -fs /usr/sbin/named9 /usr/sbin/named`
2. `ln -fs /usr/sbin/nsupdate9 /usr/sbin/nsupdate`

1. สร้างคีย์โดยใช้คำสั่ง **dnssec-keygen**:

   ```
   dnssec-keygen -a HMAC-MD5 -b 128 -n HOST keyname
   ```

   - HMAC-MD5 เป็นอัลกอริทึมที่ใช้สำหรับการเข้ารหัส

- 128 เป็รความยาวของคีย์ที่ใช้ (หรือจำนวนของบิต)
- HOST: HOST เป็นคีย์เวิร์ด TSIG ที่ใช้เพื่อสร้างโฮสต์คีย์สำหรับคีย์การเข้ารหัสที่ถูกแบ่งใช้

คำสั่ง

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST venus-batman.abc.aus.century.com
```

จะสร้างคีย์ไฟล์ 2 ไฟล์ ดังต่อไปนี้:

```
Kvenus-batman.abc.aus.century.com.+157+35215.key
Kvenus-batman.abc.aus.century.com.+157+35215.private
```

- 157 เป็นอัลกอริทึมที่ใช้ (HMAC-MD5)
- 35215 เป็น finger print ซึ่งมีประโยชน์ใน DNNSEC เนื่องจากยอมให้ใช้ได้หลายคีย์ต่อโซน

2. เพิ่ม entry เข้ากับ named.conf บนเนมเซิร์ฟเวอร์หลัก:

```
// TSIG Key
key venus-batman.abc.aus.century.com. {
        algorithm hmac-md5;
        secret "+UWSvbpxHWFdNwEAdy1Ktw==";
};
```

สมมุติว่า HMAC-MD5 ถูกใช้ keyfile ทั้งสองประกอบด้วยคีย์ที่ถูกแบ่งใช้ ซึ่งถูกเก็บเป็น entry สุดท้ายในไฟล์ หาวิธีที่ปลอดภัยที่สุดเพื่อคัดลอกคีย์ลับที่ถูกแบ่งใช้ไปยังไคลเอ็นต์ คุณไม่ต้องคัดลอก keyfile แค่แบ่งใช้คีย์ลับ

ต่อไปนี้เป็น entry สำหรับไฟล์ Kvenus-batman.abc.aus.century.com.+157+35215.private:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: +UWSvbpxHWFdNwEAdy1Ktw==
```

ต่อไปนี้เป็นตัวอย่างของไฟล์ named.conf สำหรับ เนมเซิร์ฟเวอร์หลัก โซน abc.aus.century.com ยอมให้มีการถ่ายโอนโซน และอัพเดตแบบไดนามิกเฉพาะกับเซิร์ฟเวอร์ที่มีคีย์ venus-batman.abc.aus.century.com ทำอย่างเดียวกันกับโซนแบบ reverse ซึ่งผู้อัพเดตต้องมีคีย์ที่ถูกแบ่งใช้

```
// TSIG Key
key venus-batman.abc.aus.century.com. {
        algorithm hmac-md5;
        secret "+UWSvbpxHWFdNwEAdy1Ktw==";
};

options {
        directory "/usr/local/domain";
};
zone "abc.aus.century.com" in {
        type master;
        file "named.abc.data";
        allow-transfer { key venus-batman.abc.aus.century.com.;};
        allow-update{ key venus-batman.abc.aus.century.com.; };
};
```

เนื่องจากตอนนี้การถ่ายโอนโซนถูกจำกัดสำหรับผู้ที่มีคีย์ ไฟล์ named.conf ของเนมเซิร์ฟเวอร์ย่อยต้องถูกแก้ไขด้วย คำร้องขอทั้งหมดที่ไปยัง 192.9.201.1 (venus.abc.aus.century.com) จะถูกลงชื่อโดยคีย์ บันทึกชื่อของคีย์ (venus-batman.abc.aus.century.com.) ต้องตรงคีย์เหล่านั้นบนเซิร์ฟเวอร์ที่ใช้มัน

ด้านล่างเป็นตัวอย่างของไฟล์ named.conf สำหรับเนมเซิร์ฟเวอร์ย่อย

```
// TSIG Key
key venus-batman.abc.aus.century.com. {
        algorithm hmac-md5;
        secret "+UWSvbpxHWFdNwEAdy1Ktw==";
};

server 192.9.201.1{
        keys { venus-batman.abc.aus.century.com.;};
};

options {
        directory "/usr/local/domain";
};

zone "abc.aus.century.com" IN {
     type slave;
     file "named.abc.data.bak";
     masters { 192.9.201.1; };
};
```

BIND 9 Signature:

BIND 9 สนับสนุนการลงชื่อรายการ DNSSEC SIG บางส่วน ดังระบุใน RFC 2535

SIG ใช้พับลิกคีย์และไพรเวตคีย์เพื่อพิสูจน์ตัวตนข้อความ

เร็กคอร์ดของ SIG จะให้ผู้ดูแลระบบลงชื่อข้อมูลโซนของตน เพื่อเป็นการบอกว่ามันเชื่อถือได้

การรักษาความปลอดภัยให้กับโซน root:

เมื่อใช้ขั้นตอนนี้เพื่อรักษาความปลอดภัยให้กับโซน root ให้สันนิษฐานว่า เนมเซิร์ฟเวอร์อื่นบนอินเตอร์เน็ตไม่ได้ใช้ BIND 9 และคุณต้องรักษาความปลอดภัยข้อมูลโซนของคุณ และยอมให้เซิร์ฟเวอร์อื่นตรวจสอบข้อมูลโซนของคุณ

คุณต้องการบอกว่าโซนของคุณ (ในกรณีของเราคือ `aus.century.com`) เป็น root ที่ปลอดภัย และจะตรวจสอบข้อมูลของโซนที่ปลอดภัยที่อยู่ใต้มัน

1. สร้างคีย์โดยใช้คำสั่ง **dnssec-keygen** :

   ```
   dnssec-keygen -a RSA -b 512 -r /usr/sbin/named -n ZONE aus.century.com.
   ```

   หมายเหตุ: การเข้ารหัส RSA สามารถถูกใช้เป็นอัลกอริทึมในการสร้างคีย์ถ้า OpenSSL ถูกติดตั้ง แม้ว่าคุณต้องเชื่อมไลบรารีของ DNS กับไลบรารี DNS ที่ปลอดภัยใหม่ โดยใช้คำสั่งต่อไปนี้ :

   ```
   ln -fs /usr/lib/libdns_secure.a /usr/lib/libdns.a
   ```

   - ZONE: ZONE เป็นคีย์เวิร์ด DNSSEC ที่ถูกใช้เพื่อสร้างคีย์ของโซนสำหรับคีย์การเข้ารหัสแบบ ไพรเวต/พับลิก
   - แฟล็ก r ระบุอุปกรณ์แบบสุ่ม
2. เพิ่ม entry ของพับลิกคีย์เหมือนกับไฟล์ `named.conf` entry ถูกใช้ในกรณีของเราเป็นดังต่อไปนี้ ด้านล่างเป็นเนื้อหาของ keyfile `Kaus.century.com.+001+03254.key`

   ```
   abc.aus.century.com. IN KEY 256 3 1
   AQOnfGEAgOxpzSdNRe7KePq3Dl4NqQiq7HkwKl6TygUfaw6vz6ldmauB4UQFcGKOyL68/
   Zv5ZnEvyB1fMTAaDLYz
   ```

พับลิกคีย์จะอยู่ในไฟล์ Kzonename.+algor.+fingerprint.key หรือในกรณีของเรา Kaus.century.com.+001+03254.key คุณต้องลบคลาส IN และพิมพ์ KEY พร้อมกับอ้างอิงคีย์ เมื่อคุณเพิ่ม entry นี้เข้ากับไฟล์ /etc/named.conf และรีเฟรชเนมเซิร์ฟเวอร์โซน aus.century.com จะเป็น root ที่ปลอดภัย

```
trusted-keys {
        aus.century.com. 256 3 1 "AQOnfGEAgOxpzSdNRe7KePq3Dl4NqQiq7HkwKl6Tyg
Ufaw6vz6ldmauB 4UQFcGKOyL68/Zv5ZnEvyB1fMTAaDLYz";
};
options {
        directory "/usr/local/domain";
};

zone "abc.aus.century.com" in {
        type master;
        file "named.abc.data.signed";
        allow-update{192.9.201.1;};
};
```

การใช้ chain of trust:

ตอนนี่คุณมี root ที่มีความปลอดภัย คุณสามารถทำความปลอดภัยให้กับโซนซายน์ ที่เหลือของคุณ ในกรณีนี้ เราจะทำการให้ความปลอดภัยกับโซน abc.aus.century.com

ทำขั้นตอนต่อไปนี้เพื่อให้ความปลอดภัยโซนซายน์ที่เหลือของคุณ :

1. สร้างคู่ของคีย์โดยใช้คำสั่ง **dnssec-keygen**:

   ```
   dnssec-keygen -a RSA -b 512 -r /usr/sbin/named -n ZONE abc.aus.century.com.
   ```

   แฟล็ก r จะระบุอินพุตไฟล์แบบสุ่ม
2. สร้าง keyset โดยใช้คำสั่ง **dnssec-makekeyset** :

   ```
   dnssec-makekeyset -t 172800 Kabc.aus.century.com.+001+11515.key
   ```

   โดยที่ *Kabc.aus.century.com.+001+11515.key* เป็นพับลิกคีย์ของคุณ
   นี่จะสร้างไฟล์ keyset ที่ชื่อ keyset-abc.aus.century.com
3. ส่งไฟล์ keyset นี้ไปยังโซนพาเรนต์เพื่อให้ลงชื่อ ในกรณีนี้ โซนพาเรนต์ของเราคือโซน root ที่ปลอดภัย aus.century.com
4. พาเรนต์ต้องลงชื่อคีย์โดยใช้ไพรเวตคีย์ของมัน

   ```
   dnssec-signkey keyset-abc.aus.century.com. Kaus.century.com.+001+03254.private
   ```

   นี่จะสร้างไฟล์ชือ signedkey-abc.aus.century.com และพาเรนต์ต้องส่งไฟล์นี้กลับไปยังโซนซายน์
5. บนซายน์เนมเซิร์ฟเวอร์ของโซน abc.aus.century.com เพิ่ม $INCLUDE Kabc.aus.century.com.+001+11515.key เข้ากับไฟล์โซนแบบธรรมดา named.abc.data อย่าลืมใส่ไฟล์ signedkey-abc.aus.century.com ในตำแหน่งเดียวกับไฟล์โซน named.abc.data เมื่อโซนถูกลงชื่อในขั้นตอนต่อไปนี้ โปรแกรมจะรู้ว่าต้องรวม signedkey-abc.aus.century.com ซึ่งได้รับจากพาเรนต์

   ```
   $TTL 3h    ;3 hour

   @ IN SOA venus.abc.aus.century.com. gail.zeus.abc.aus.century.com.  (
                  1       ;serial
                  3600    ;refresh
   ```

```
                600     ;retry
                3600000 ;expire
                86400   ;negative caching TTL
)
$INCLUDE Kabc.aus.century.com.+001+03254.key
```

6. ลงชื่อโซนโดยใช้คำสั่ง **dnssec-signzone** :

```
dnssec-signzone -o abc.aus.century.com. named.abc.data
```

7. แก้ไขไฟล์ **named.conf** บนโซน child `abc.aus.century.com`เพื่อใช้ไฟล์โซนที่ถูกลงชื่อใหม่ (`named.abc.data.signed`) ตัวอย่าง เช่น:

```
options {
        directory "/usr/local/domain";
};

zone "abc.aus.century.com" in {
        type master;
        file "named.abc.data.signed";
        allow-update{192.9.201.1;};
};
```

8. รีเฟรชเนมเซิร์ฟเวอร์

สำหรับข้อมูลเกี่ยวกับการแก้ไขปัญหา ดูที่ “ปัญหาการระบุชื่อ” ในหน้า 447

การวางแผนและการกำหนดคอนฟิกความละเอียดของชื่อ LDAP (สกีมา IBM SecureWay Directory)

Lightweight Directory Access Protocol (LDAP) เป็นมาตรฐานอุตสาหกรรมแบบเปิดที่กำหนดวิธีสำหรับการเข้าถึงและอัพเดตข้อมูลในไดเร็กทอรี

LDAP schema จะกำหนดกฎสำหรับการเรียงลำดับข้อมูล คลาสของ **ibm-HostTable** ส่วนของ IBM SecureWay ไดเร็กทอรี schema สามารถถูกใช้เพื่อเก็บข้อมูลการแม็พชื่อกับอินเตอร์เน็ตแอดเดรสสำหรับทุกๆโฮสต์บนเน็ตเวิร์ก

คลาสของ **ibm-HostTable** object ถูกกำหนดดังต่อไปนี้ :

```
Object Class name:     ibm-HostTable
Description:           Host Table entry which has a collection of hostname to
                       IP address mappings.
OID:                   TBD
RDN:                   ipAddress
Superior object class: top
Required Attributes:   host, ipAddress
Optional Attributes:   ibm-hostAlias, ipAddressType, description
```

คำจำกัดความของแอ็ตทริบิวต์เป็นดังต่อไปนี้ :

```
Attribute Name: ipAddress
Description:    IP Address of the hostname in the Host Table
OID:            TBD
Syntax:         caseIgnoreString
Length:         256
Single Valued:  Yes
```

```
Attribute Name: ibm-hostAlias
Description:    Alias of the hostname in the Host Table
OID:            TBD
Syntax:         caseIgnoreString
Length:         256
Single Valued:  Multi-valued
Attribute Name: ipAddressType
Description:    Address Family of the IP Address (1=IPv4, 2=IPv6)
OID:            TBD
Syntax:         Integer
Length:         11
Single Valued:  Yes
Attribute Name: host
Description:    The hostname of a computer system.
OID:            1.13.18.0.2.4.486
Syntax:         caseIgnoreString
Length:         256
Single Valued:  Multi-valued
Attribute Name: description
Description:    Comments that provide a description of a directory object entry.
OID:            2.5.4.13
Syntax:         caseIgnoreString
Length:         1024
Single Valued:  Multi-valued
```

ใช้โพรซีเดอร์ต่อไปนี้เพื่อตั้งค่า **LDAP** เซิร์ฟเวอร์ให้สอดคล้องกับ IBM SecureWay ไดเร็กทอรี schema สำหรับเก็บข้อมูลโฮสต์การแม็ตชื่อกับอินเตอร์เน็ตแอดเดรส

1. เพิ่มส่วนต่อท้ายบน **LDAP** เซิร์ฟเวอร์ ส่วนต่อท้ายเป็นจุดเริ่มต้นของฐานข้อมูลของโฮสต์ ตัวอย่างเช่น "cn=hosts" ซึ่งสามารถทำได้โดยใช้เครื่องมือ web-based IBM SecureWay Directory Server Administration
2. สร้างไฟล์ LDAP Data Interchange Format (LDIF) ซึ่งสามารถทำแบบแมนวล หรือด้วยคำสั่ง **hosts2ldif** ซึ่งจะสร้างไฟล์ LDIF จากไฟล์ /etc/hosts ดูที่ คำสั่ง hosts2ldif สำหรับข้อมูลเพิ่มเติม ต่อไปนี้เป็นตัวอย่างของไฟล์ LDIF :

   ```
   dn: cn=hosts
   objectclass: top
   objectclass: container
   cn: hosts
   dn: ipAddress=1.1.1.1, cn=hosts
   host: test
   ipAddress: 1.1.1.1
   objectclass: ibm-HostTable
   ipAddressType: 1
   ibm-hostAlias: e-test
   ibm-hostAlias: test.austin.ibm.com
   description: first ethernet interface
   dn: ipAddress=fe80::dead, cn=hosts
   host: test
   ipAddress: fe80::dead
   objectclass: ibm-HostTable
   ipAddressType: 2
   ibm-hostAlias: test-ll
   ibm-hostAlias: test-ll.austin.ibm.com
   description: v6 link level interface
   ```

3. อิมปอร์ตข้อมูลโฮสต์ไดเร็กทอรีจากไฟล์ LDIF บนเซิร์ฟเวอร์ **LDAP** ซึ่งสามารถทำได้โดยคำสั่ง **ldif2db** หรือผ่านเครื่องมือ web-based IBM SecureWay Directory Server Administration

เพื่อตั้งค่าไคลเอ็นต์เพื่อเข้าถึงฐานข้อมูลของโฮสต์บนเซิร์ฟเวอร์ LDAP โดยใช้กลไก **LDAP** ทำตามขั้นตอนเหล่านี้ :

1. สร้างไฟล์ /etc/resolv.ldap ดูที่ รูปแบบของไฟล์ resolv.ldap สำหรับ TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติมและตัวอย่างแบบละเอียดของไฟล์ resolv.ldap
2. เปลี่ยนการแปลงชื่อดีฟอลต์ผ่านตัวแปรสภาวะแวดล้อม **NSORDER** ไฟล์ /etc/netsvc.conf หรือไฟล์ /etc/irs.conf ดที่ รูปแบบไฟล์ netsvc.conf สำหรับ TCP/IP หรือ รูปแบบไฟล์ irs.conf สำหรับ TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติม

แม้ว่ายังได้รับการสนับสนุน ไม่เห็นด้วยที่จะใช้กลไก ldap กลไก ldap ที่มีอยู่นี้ทำงาน กับสกีมา IBM SecureWay Directory ขณะที่ nis_ldap (NIS_LDAP) ทำงานกับสกีมา RFC 2307 แนะนำให้ใช้กลไก nis_ldap แทนกลไก ldap สำหรับข้อมูลเกี่ยวกับการแปลงชื่อ nis_ldap ดูที่ "การวางแผนและตั้งค่าการแก้ไขปัญหาเรื่องชื่อ NIS_LDAP (RFC 2307 schema)"

การวางแผนและตั้งค่าการแก้ไขปัญหาเรื่องชื่อ NIS_LDAP (RFC 2307 schema)

AIX 5.2 นำเสนอกลไการตั้งชื่อใหม่ที่เรียกว่า NIS_LDAP

ความแตกต่างระหว่างกลไก LDAP ที่มีอยู่เดิมและกลไก NIS_LDAP อยู่ใน LDAP schema (ชุดของแอ็ตทริบิวต์และคลาสอ็อบเจ็กต์ ที่กำหนดแอ็ตทริบิวต์ที่จัดกลุ่มพร้อมกับการอธิบายถึง เอ็นทิตี) กลไก LDAP ที่มีอยู่จะทำงานกับ IBM SecureWay Directory Schema ที่ยอมรับ เซิร์ฟเวอร์ LDAP และให้การสนับสนุนเฉพาะเซอร์วิสการตั้งชื่อโฮสต์เท่านั้น กลไก NIS_LDAP ทำงานกับ RFC 2307 schema ที่ยอมรับเซิร์ฟเวอร์ LDAP และสนับสนุนเซอร์สวิ NIS ทั้งหมด คือ: ผู้ใช้และกลุ่ม โฮสต์ เซอร์วิส โปรโตคอล เน็ตเวิร์ก และ netgroup RFC 2307 นิยามชุดของแอ็ตทริบิวต์และคลาสอ็อบเจ็กต์ ที่สามารถใช้เพื่ออธิบายถึงการให้บริการข้อมูลเน็ตเวิร์ก ซึ่งรวมถึงผู้ใช้ และกลุ่ม

- To configure the LDAP server, you will need to set up the LDAP server and migrate the required data to the server.
  1. ใช้คำสั่ง **mksecldap** เพื่อตั้งค่าเซิร์ฟเวอร์ กลไก nis_ldap ทำงานกับ RFC 2307 schema ขณะที่ติดตั้งเซิร์ฟเวอร์ LDAP คำสั่ง **mksecldap** ควรถูกเรียกใช้พร้อมกับอ็อพชัน -S rfc2307 หรือ -S rfc2307aix อย่างใดอย่างหนึ่ง (ซึ่งไม่ใช่ -S aix โดยที่ระบุ IBM SecureWay Directory schema) ตามค่าดีฟอลต์แล้ว คำสั่ง **mksecldap** จะถ่ายโอนผู้ใช้และกลุ่มที่นิยามอยู่บนระบบโลคัลไปยังเซิร์ฟเวอร์ LDAP หากคุณต้องการปิดใช้งาน การโอนย้ายระบบนี้ ให้ใช้อ็อพชัน -u NONE

     ```
     mksecldap -s -a cn=admin -p adminpwd -S rfc2307aix
     ```

     อ็อพชันนี้ตั้งค่าเซิร์ฟเวอร์ LDAP พร้อมกับผู้ดูแลระบบ DN เป็น cn=admin และรหัสผ่านเป็น adminpwd คำต่อท้ายดีฟอลต์ cn=aixdata ยังถูกเพิ่มให้กับไฟล์ /etc/slapd32.conf ซึ่งเป็นไฟล์คอนฟิกูเรชันสำหรับ LDAP

     ตามค่าดีฟอลต์แล้ว คำสั่ง **mksecldap** จะถ่ายโอนผู้ใช้และกลุ่มที่นิยามอยู่บนระบบโลคัลไปยังเซิร์ฟเวอร์ LDAP หากคุณต้องการปิดใช้งานการโอนย้ายระบบนี้ ให้ใช้อ็อพชัน -u NONE ซึ่งปกป้องการโอนย้ายระบบของผู้ใช้และกลุ่มบนโลคัลไปยังเซิร์ฟเวอร์ LDAP ดังนั้น คุณจึงสามารถเพิ่มผู้ใช้และกลุ่ม NIS ได้ในภายหลัง

     ```
     mksecldap -s -a cn=admin -p adminpwd -u NONE
     ```

     สำหรับข้อมูล เพิ่มเติมเกี่ยวกับคำสั่ง **mksecldap** โปรดดูคำอธิบายคำสั่งใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 3*
  2. ถ่ายโอนข้อมูล NIS ใช้คำสั่ง **nistoldif** จากเซิร์ฟเวอร์ NIS เพื่อถ่ายโอนข้อมูล NIS ที่แม็พกับเซิร์ฟเวอร์ LDAP คำสั่ง **nistoldif** ยังสามารถถูกใช้เพื่อถ่ายโอนข้อมูลจากไฟล์มิติเดียว รันคำสั่ง **nistoldif** บนระบบที่มีข้อมูล NIS ที่จำเป็นต้องถ่ายโอนไปยังเซิร์ฟเวอร์ LDAP

```
nistoldif -h server1.ibm.com -a cn=admin -p adminpwd -d cn=aixdata
```

คำสั่งนี้ ถ่ายโอนข้อมูล NIS ที่แม็พจากระบบโลคัลไปยังเซิร์ฟเวอร์ LDAP นั่นคือ server1.ibm.com ข้อมูล NIS ถูกวางอยู่ใต้ cn=aixdata DN คุณยังสามารถรันคำสั่ง **nistoldif** เพื่อถ่ายโอนข้อมูลจากไฟล์มิติเดียว บนระบบใดๆ ไปยังเซิร์ฟเวอร์ LDAP ไฟล์มิติเดียวจะถูกใช้สำหรับการแม็พใดๆ ที่หายไปจากเซิร์ฟเวอร์ NIS

สำหรับข้อมูลเกี่ยวกับคำสั่ง **nistoldif** โปรดดูคำอธิบายคำสั่งใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*

หมายเหตุ: ชื่อ จะถูกแทนค่าด้วยแอ็ตทริบิวต์ cn ของเซิร์ฟเวอร์ LDAP แอ็ตทริบิวต์ cn ที่นิยามโดย RFC 2307 ไม่สนใจขนาดตัวพิมพ์ ชื่อที่แตกต่างกันตามขนาดตัวพิมพ์ จะถูกผสานอยู่บนเซิร์ฟเวอร์ และการจับคู่จะไม่สนใจขนาดตัวพิมพ์ด้วยเช่นกัน การค้นหา TCP, tcp หรือ Tcp จะส่งคืนรายการโปรโตคอลทั้งหมดสำหรับ TCP

- หากต้องการตั้งค่าไคลเอ็นต์ LDAP เพื่อเข้าถึงชื่อจากเซิร์ฟเวอร์ LDAP ให้รันคำสั่ง **mksecldap** ด้วยอ็อพชันการตั้งค่าไคลเอ็นต์
  1. คำสั่ง **mksecldap** จะบันทึกชื่อเซิร์ฟเวอร์ LDAP พอร์ต admindn รหัสผ่าน และ basedn ลงในไฟล์ /etc/security/ldap/ldap.cfg ที่ถูกอ่านโดย **secldapclntd** ณ เวลา ที่เริ่มต้นการทำงาน คำสั่ง **mksecldap** สตาร์ท **secldapclntd** daemon โดยอัตโนมัติ หากการตั้งค่าเป็นผลสำเร็จแล้ว

     See the /etc/security/ldap/ldap.cfg file ใน *ข้อมูลอ้างอิงไฟล์* and the **secldapclntd** daemon in the *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 5* for more information.
  2. คำสั่ง **mksecldap** เพิ่มกลไก nis_ldap ให้กับไฟล์ /etc/netsvc.conf และไฟล์ /etc/irs.conf เพื่อให้การแก้ปัญหาเรื่องชื่อสามารถส่งตรงไปยัง LDAP ได้ คุณยังสามารถตั้งค่าตัวแปรสภาพแวดล้อม **NSORDER** แบบแมนวลไปเป็น nis_ldap เพื่อใช้การแก้ไขปัญหาเรื่องชื่อ NIS_LDAP

     ```
     mksecldap -c -a cn=admin -p adminpwd -h server1.ibm.com
     ```

     คำสั่งนี้ ตั้งค่าระบบโลคัลเพื่อใช้ server1.ibm.com สำหรับเซิร์ฟเวอร์ LDAP DN ของผู้ดูแลระบบเซิร์ฟเวอร์ LDAP และรหัสผ่านต้องถูกจัดหาไว้ สำหรับไคลเอ็นต์นี้เพื่อพิสูจน์ตัวตนกับเซิร์ฟเวอร์ ไฟล์ /etc/netsvc.conf และ /etc/irs.conf ถูกอัพเดต ดังนั้น การแก้ไขปัญหาเรื่องชื่อ ได้รับการแก้ไขผ่าน NIS_LDAP

     โปรดดูรูปแบบไฟล์ /etc/netsvc.conf สำหรับ TCP/IP หรือรูปแบบไฟล์ /etc/irs.conf สำหรับ TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติม
  3. การแก้ไขปัญหาเรื่องชื่อสำหรับผู้ใช้และกลุ่มไม่ได้ถูกควบคุมโดยไฟล์ /etc/netsvc.conf หรือ /etc/irs.conf แต่จะถูกควบคุมผ่านไฟล์ /etc/security/user หากต้องการเปิดใช้งาน สำหรับผู้ใช้ LDAP เพื่อล็อกอินเข้าสู่ระบบ AIX ให้ตั้งค่าตัวแปร SYSTEM and registry ของผู้ใช้ไปเป็นไฟล์ LDAP in the /etc/security/user ของระบบไคลเอ็นต์ คุณสามารถรันคำสั่ง **chuser** เพื่อทำสิ่งนี้ได้

     ```
     chuser -R LDAP SYSTEM=LDAP registry=LDAP foo
     ```

     คุณสามารถ ตั้งค่าระบบของคุณเพื่ออนุญาตให้ผู้ใช้ LDAP ทั้งหมดล็อกอินเข้าสู่ระบบ หากต้องการทำ สิ่งนี้ให้แก้ไขไฟล์ /etc/security/user เพิ่ม registry = files ให้กับ root stanza จากนั้น เพิ่มการลงทะเบียน SYSTEM = LDAP และ registry = LDAP ให้กับ stanza ดีฟอลต์

     สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการพิสูจน์ตัวตนผู้ใช้ โปรดอ้างถึง Light Directory Access Protocol ใน *การรักษาความปลอดภัย*

ข้อมูลที่เกี่ยวข้อง:

การโอนย้ายจากเซอร์วิส NIS ไปเป็น RFC 2307–compliant LDAP

TCP/IP แอดเดรสและการกำหนด พารามิเตอร์ - Dynamic Host Configuration Protocol

Transmission Control Protocol/Internet Protocol (TCP/IP) ช่วยให้สามารถสื่อสารระหว่างเครื่องที่มีแอดเดรสซึ่งกำหนดคอนฟิก ภาระส่วนหนึ่งที่ ผู้ดูแลเครือข่ายต้องพบคือการกำหนดแอดเดรสและการแจกจ่ายพารามิเตอร์ สำหรับเครื่องทั้งหมดบนเครือข่าย โดยทั่วไป นี่เป็นโปรเซสซึ่ง ผู้ดูแลระบบกำหนดคอนฟิกูเรชันให้กับผู้ใช้แต่ละราย เพื่อให้ผู้ใช้สามารถ กำหนดคอนฟิกเครื่องของตนเองได้ อย่างไรก็ตาม คอนฟิกูเรชันที่ไม่ถูกต้องและความเข้าใจผิด อาจทำให้มีการเรียกเซอร์วิสซึ่งผู้ดูแลระบบต้องจัดการทีละกรณีไป **Dynamic Host Configuration Protocol (DHCP)** เป็นเมธอดที่ผู้ดูแล เครือข่ายใช้ในการเอาผู้ใช้ขั้นปลายออกจากปัญหาคอนฟิกูเรชันนี้ และรักษา คอนฟิกูเรชันเครือข่ายไว้ในที่ตั้งส่วนกลาง

DHCP เป็นโปรโตคอลชั้นแอ็พพลิเคชันที่ช่วยให้เครื่องไคลเอ็นต์ บนเครือข่ายสามารถเรียกใช้ IP แอดเดรสและพารามิเตอร์คอนฟิกูเรชันอื่นจาก เซิร์ฟเวอร์ ไคลเอ็นต์เรียกใช้ข้อมูลโดยการแลกเปลี่ยนแพ็กเก็ตระหว่าง daemon บน ไคลเอ็นต์และ daemon อื่นบนเซิร์ฟเวอร์ ขณะนี้ ระบบปฏิบัติการส่วนใหญ่นำเสนอไคลเอ็นต์ **DHCP** ในแพ็กเกจพื้นฐาน

เพื่อให้ได้รับแอดเดรส **DHCP** client daemon (**dhcpcd**) จะแพร่สัญญาณข้อความค้นหา **DHCP** ซึ่งได้รับและประมวลผลโดยเซิร์ฟเวอร์ (สามารถกำหนดคอนฟิกหลายเซิร์ฟเวอร์บนเครือข่ายสำหรับการทำซ้ำได้) หากมีแอดเดรสว่างสำหรับไคลเอ็นต์นั้น จะมีการสร้างข้อความการนำเสนอ **DHCP** ข้อความนี้มี IP แอดเดรสและอ็อพชันอื่นที่เหมาะสมสำหรับ ไคลเอ็นต์นั้น ไคลเอ็นต์ได้รับการนำเสนอ **DHCP** ของเซิร์ฟเวอร์ และจัดเก็บไว้ในขณะที่รอการนำเสนออื่น เมื่อไคลเอ็นต์เลือกการนำเสนอที่ดีที่สุด ไคลเอ็นต์จะแพร่สัญญาณคำร้องขอ **DHCP** ที่ระบุเซิร์ฟเวอร์ซึ่ง นำเสนอสิ่งที่ไคลเอ็นต์ต้องการ

เซิร์ฟเวอร์ **DHCP** ที่กำหนดคอนฟิกทั้งหมดได้รับคำร้องขอ แต่ละเซิร์ฟเวอร์จะตรวจสอบเพื่อ ดูว่าเป็นเซิร์ฟเวอร์ที่ร้องขอหรือไม่ ถ้าไม่ เซิร์ฟเวอร์จะทำให้แอดเดรสที่กำหนดให้กับ ไคลเอ็นต์นั้นว่าง The requested server marks the address as assigned and returns a **DHCP** acknowledgment, at which time, the transaction is complete. ไคลเอ็นต์มีแอดเดรสตามระยะเวลา (เช่า) ที่กำหนดโดย เซิร์ฟเวอร์

เมื่อใช้ไปครึ่งหนึ่งของเวลาเช่า ไคลเอ็นต์จะส่งแพ็กเก็ต *ต่ออายุ* ไปยัง เซิร์ฟเวอร์เพื่อขยายเวลาเช่า If the server is willing to renew, it sends a **DHCP** acknowledgment. หากไคลเอ็นต์ไม่ได้รับการตอบกลับจากเซิร์ฟเวอร์ที่เป็นเจ้าของแอดเดรสปัจจุบัน ไคลเอ็นต์จะแพร่สัญญาณแพ็กเก็ต **DHCP** rebind เพื่อให้ถึงเซิร์ฟเวอร์ถ้า ตัวอย่างเช่น เซิร์ฟเวอร์ถูกย้ายจากเครือข่ายหนึ่งไปยังเครือข่ายอื่น หากไคลเอ็นต์ ยังไม่ได้ต่ออายุแอดเดรสหลังจากหมดเวลาเช่า อินเตอร์เฟสจะ ดาวน์และโปรเซสเริ่มต้นขึ้นใหม่ วงจรนี้ช่วยป้องกันไม่ให้หลายไคลเอ็นต์ บนเครือข่ายได้รับการกำหนดแอดเดรสเดียวกัน

เซิร์ฟเวอร์ **DHCP** กำหนดแอดเดรสตามข้อมูลคีย์ คีย์ทั่วไปสี่รายการคือ เครือข่าย คลาส ผู้ขาย และ ID ไคลเอ็นต์ เซิร์ฟเวอร์ใช้คีย์เหล่านี้เพื่อเรียกใช้ แอดเดรสและชุดของคอนฟิกูเรชันอ็อพชันเพื่อส่งกลับไปยังไคลเอ็นต์

เครือข่าย
ระบุเซกเมนต์เครือข่ายต้นทางของแพ็กเก็ต คีย์เครือข่าย ช่วยให้เซิร์ฟเวอร์สามารถตรวจสอบฐานข้อมูลแอดเดรสและกำหนดแอดเดรส โดยใช้เซกเมนต์เครือข่าย

คลาส เป็นไคลเอ็นต์ที่กำหนดคอนฟิกได้โดยสมบูรณ์ สามารถระบุแอดเดรสและอ็อพชัน สามารถใช้คีย์นี้เพื่อแสดงฟังก์ชันเครื่องในเครือข่าย หรือเพื่ออธิบาย วิธีการจัดกลุ่มเครื่องสำหรับวัตถุประสงค์การจัดการ ตัวอย่างเช่น ผู้ดูแลเครือข่ายอาจต้องการสร้างคลาส `netbios` ที่มีอ็อพชัน สำหรับไคลเอ็นต์ NetBIOS หรือคลาส `accounting` ที่แสดงถึง เครื่องแผนกบัญชีซึ่งต้องการสิทธิเข้าถึงเครื่องพิมพ์เฉพาะ

ผู้ขาย ช่วยระบุไคลเอ็นต์โดยใช้ฮาร์ดแวร์/ซอฟต์แวร์แพล็ตฟอร์ม (ตัวอย่างเช่น ไคลเอ็นต์ Microsoft Windows 95 หรือไคลเอ็นต์ OS/2 Warp)

ID ไคลเอ็นต์

ระบุไคลเอ็นต์โดยใช้ชื่อโฮสต์ของเครื่อง หรือแอดเดรสชั้น medium access control (MAC) ID ไคลเอ็นต์มีการระบุในไฟล์คอนฟิกูเรชัน ของ **dhcpcd** daemon นอกจากนี้ เซิร์ฟเวอร์สามารถใช้ ID ไคลเอ็นต์ เพื่อส่งผ่านอ็อพชันไปยังไคลเอ็นต์เฉพาะ หรือห้ามไคลเอ็นต์เฉพาะ ไม่ให้ได้รับพารามิเตอร์ใดๆ

คอนฟิกูเรชันสามารถใช้คีย์เหล่านี้เพียงอย่างเดียวหรือใช้ร่วมกับคีย์อื่นก็ได้ หากไคลเอ็นต์ระบุหลายคีย์และสามารถกำหนดได้หลายแอดเดรส จะมีการเลือกเพียงคีย์เดียว และได้รับชุดอ็อพชันมาจากคีย์แรก ที่เลือก สำหรับข้อมูลรายละเอียดเพิ่มเติมเกี่ยวกับการเลือกคีย์และแอดเดรส ให้ดูที่ “คอนฟิกูเรชัน DHCP” ในหน้า 239

ต้องการรีเลย์เอเจนต์เพื่อให้การแพร่สัญญาณแรกเริ่มจากไคลเอ็นต์สามารถ ออกจากเครือข่ายโลคัลได้ เอเจนต์นี้เรียกว่ารีเลย์เอเจนต์ BOOTP รีเลย์เอเจนต์ ทำหน้าที่เป็นเอเจนต์ส่งต่อสำหรับแพ็กเก็ต **DHCP** และ **BOOTP**

เซิร์ฟเวอร์ DHCP

ในระบบปฏิบัติการ AIX เซิร์ฟเวอร์ **DHCP** ถูกแบ่งเซ็กเมนต์เป็นสามส่วน หลักๆ

คอมโพเนนต์หลักของเซิร์ฟเวอร์ **DHCP** คือเซิร์ฟเวอร์โปรโตคอลเอ็นจิ้น และชุดเธรดเซอร์วิส โดยแต่ละส่วนมีข้อมูลคอนฟิกูเรชัน ของตัวเอง

ฐานข้อมูล DHCP:

ฐานข้อมูล `db_file.dhcpo` ใช้เพื่อติดตาม ไคลเอ็นต์และแอดเดรสและสำหรับการควบคุมการเข้าถึง (ตัวอย่างเช่น การอนุญาต บางไคลเอ็นต์บนบางเครือข่ายแต่ไม่อนุญาตไคลเอ็นต์ที่เหลือ หรือการปิดใช้งานไคลเอ็นต์ **BOOTP** บนเครือข่ายเฉพาะ)

อ็อพชันยังมีการจัดเก็บไว้ในฐานข้อมูลสำหรับการดึงข้อมูลและการจัดส่งไปยังไคลเอ็นต์ มีการนำฐานข้อมูลไปใช้เป็นอ็อบเจ็กต์ที่โหลดได้แบบไดนามิก ซึ่งช่วยให้ สามารถอัพเกรดและบำรุงรักษาเซิร์ฟเวอร์ได้ง่าย

โดยใช้ข้อมูลในไฟล์คอนฟิกูเรชัน ฐานข้อมูลถูกเตรียมพร้อม และตรวจสอบความสอดคล้อง ชุดของไฟล์ checkpoint จัดการกับอัพเดตใน ฐานข้อมูลและลดค่าใช้จ่ายในการเขียนลงในไฟล์หน่วยเก็บหลัก ฐานข้อมูลยังมีพูลแอดเดรสและอ็อพชัน แต่ข้อมูลเหล่านี้เป็นแบบสแตติก และมีการอธิบายไว้ใน “คอนฟิกูเรชัน DHCP” ในหน้า 239

ไฟล์หน่วยเก็บหลักและสำเนาสำรองเป็นไฟล์ flat ASCII ที่สามารถ แก้ไขได้ รูปแบบสำหรับไฟล์หน่วยเก็บหลักของฐานข้อมูลคือ:

```
DF01
"CLIENT ID" "0.0.0.0" State LeaseTimeStart LeaseTimeDuration LeaseTimeEnd
  "Server IP Address" "Class ID" "Vendor ID" "Hostname" "Domain Name"
"CLIENT ID" "0.0.0.0" State LeaseTimeStart LeaseTimeDuration LeaseTimeEnd
  "Server IP Address" "Class ID" "Vendor ID" "Host Name" "Domain Name"
...
```

บรรทัดแรกคือตัวระบุเวอร์ชันสำหรับไฟล์: `DF01c` บรรทัดต่อมาเป็นบรรทัดนิยามไคลเอ็นต์เร็กคอร์ด เซิร์ฟเวอร์อ่าน ตั้งแต่บรรทัดที่สองไปถึงตอนท้ายของไฟล์ (พารามิเตอร์ในอัญประกาศต้อง ถูกใส่ไว้ในอัญประกาศ)

"*CLIENT ID*"

ID ซึ่งไคลเอ็นต์ใช้เพื่อแสดงแทนตัวเองที่เซิร์ฟเวอร์

"0.0.0.0"

เป็น IP แอดเดรสที่กำหนดให้กับเซิร์ฟเวอร์ **DHCP** ในปัจจุบัน หากไม่มี การกำหนดแอดเดรส ค่านี้จะเป็น "0.0.0.0"

State สถานะปัจจุบันของไคลเอ็นต์ **DHCP** โปรโตคอลเอ็นจินมี ชุดที่ใช้ได้ และมีการเก็บรักษาสถานะไว้ในฐานข้อมูล **DHCP** ตัวเลขถัดจาก *State* แสดงถึงค่า สถานะ สามารถเป็น:

(1) FREE

แสดงถึงแอดเดรสที่พร้อมใช้งาน โดยทั่วไป ไคลเอ็นต์ไม่มี สถานะนี้ยกเว้นว่าไคลเอ็นต์ไม่ได้รับการกำหนดแอดเดรส **dadmin** และ เอาต์พุตจาก **lssrc** รายงานสถานะนี้เป็น Free

(2) BOUND

บ่งชี้ว่าไคลเอ็นต์และแอดเดรสถูกโยงเข้าด้วยกัน และไคลเอ็นต์ได้รับการกำหนด แอดเดรสนี้ในช่วงเวลาหนึ่ง **dadmin** และ เอาต์พุตจาก **lssrc** รายงานสถานะนี้เป็น Leased

(3) EXPIRED

บ่งชี้ว่าไคลเอ็นต์และแอดเดรสถูกโยงเข้าด้วยกัน แต่ใช้เพื่อเป็นข้อมูลเท่านั้น ในลักษณะคล้ายกับแอดเดรสที่เช่า อย่างไรก็ตาม สถานะหมดอายุ แสดงถึงไคลเอ็นต์ที่ปล่อยให้การเช่าหมดอายุ แอดเดรสที่หมดอายุสามารถนำมาใช้ได้ และกำหนดใหม่หลังจากแอดเดรสที่ว่างทั้งหมดไม่มีอยู่ และก่อนจะกำหนด แอดเดรสที่เช่าอีกครั้ง **dadmin** และ เอาต์พุตจาก **lssrc** รายงานสถานะนี้เป็น Expired

(4) RELEASED

บ่งชี้ว่าไคลเอ็นต์และแอดเดรสถูกโยงเข้าด้วยกันเพื่อใช้เป็นข้อมูลเท่านั้น โปรโตคอล **DHCP** แนะนำให้เซิร์ฟเวอร์ **DHCP** เก็บรักษาข้อมูล เกี่ยวกับไคลเอ็นต์ที่ให้บริการไปแล้วเพื่อใช้อ้างอิงในอนาคต (โดยส่วนใหญ่ พยายาม กำหนดแอดเดรสเดียวกันให้กับไคลเอ็นต์ซึ่งเคยได้รับการกำหนดแอดเดรสนั้นใน อดีต) สถานะนี้บ่งชี้ว่าไคลเอ็นต์รีลีสแอดเดรสแล้ว แอดเดรสนั้นพร้อม ให้ไคลเอ็นต์อื่นใช้งานได้ ถ้าไม่มีแอดเดรสอื่น **dadmin** และ เอาต์พุตจาก **lssrc** รายงานสถานะนี้เป็น Released

(5) RESERVED

บ่งชี้ว่าไคลเอ็นต์และแอดเดรสถูกโยงเข้าด้วยกัน แต่ในแบบหลวมๆ ไคลเอ็นต์ออกใช้ข้อความค้นหา **DHCP** และเซิร์ฟเวอร์ **DHCP** ตอบกลับแล้ว แต่ ไคลเอ็นต์ยังไม่ได้ตอบกลับด้วยคำร้องขอ **DHCP** สำหรับแอดเดรสนั้น **dadmin** และ เอาต์พุตจาก **lssrc** รายงานสถานะนี้เป็น Reserved

(6) BAD

แสดงถึงแอดเดรสที่ถูกใช้อยู่ในเครือข่ายแต่ยังไม่ได้ส่ง โดยเซิร์ฟเวอร์ **DHCP** สถานะนี้ยังแสดงถึงแอดเดรสที่ไคลเอ็นต์ ปฏิเสธด้วย สถานะนี้ไม่ได้ใช้กับไคลเอ็นต์ **dadmin** และ เอาต์พุตจาก **lssrc** รายงานสถานะนี้เป็น Used และ Bad ตามลำดับ

LeaseTimeStart

เป็นการเริ่มต้นของเวลาเช่าปัจจุบัน (ในจำนวนวินาทีตั้งแต่ วันที่ 1 มกราคม 1970)

LeaseTimeDuration

แสดงถึงช่วงเวลาของการเช่า (ในหน่วยวินาที)

LeaseTimeEnd

ใช้รูปแบบเดียวกับ *LeaseTimeStart* แต่ แสดงถึงการสิ้นสุดของการเช่า คอนฟิกูเรชันอ็อพชันบางรายการใช้ค่าที่แตกต่าง สำหรับการเริ่มต้นและการสิ้นสุดของการเช่า และค่าเหล่านี้สามารถถูกยกเลิก โดยอ็อพชันไฟล์คอนฟิกูเรชัน โปรดดูที่ “ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับฐานข้อมูล db_file” ในหน้า 257

"Server IP Address"
เป็น IP แอดเดรสของเซิร์ฟเวอร์ DHCP ที่เป็นเจ้าของเร็กคอร์ดนี้

"Class ID" "Vendor ID" "Host Name" "Domain Name"
ค่าที่เซิร์ฟเวอร์ใช้เพื่อกำหนดอ็อพชันซึ่งจะถูกส่งไปยัง เซิร์ฟเวอร์ (ที่จัดเก็บเป็นสตริงในอัญประกาศ) พารามิเตอร์เหล่านี้ช่วยให้ประสิทธิภาพดีขึ้น เนื่องจากสามารถ สร้างรายการอ็อพชันล่วงหน้าสำหรับไคลเอ็นต์เหล่านี้เมื่อเซิร์ฟเวอร์ **DHCP** สตาร์ทอัพ

ไฟล์ DHCP checkpoint:

ไม่มีการระบุไวยากรณ์สำหรับไฟล์ checkpoint

หากเซิร์ฟเวอร์เสียหายหรือคุณต้องปิดและไม่สามารถทำการปิดฐานข้อมูล ตามปกติได้ เซิร์ฟเวอร์สามารถประมวลผลไฟล์ checkpoint และสำเนาสำรอง เพื่อสร้างฐานข้อมูลที่ถูกต้องขึ้นใหม่ ไคลเอ็นต์ที่กำลังถูกเขียนลงในไฟล์ checkpoint เมื่อเซิร์ฟเวอร์เสียหายจะสูญหายไป ไฟล์ดีฟอลต์คือ:

`/etc/db_file.cr`
การดำเนินงานฐานข้อมูลปกติ

`/etc/db_file.crbk`
สำเนาสำรองของฐานข้อมูล

`/etc/db_file.chkpt` และ `/etc/db_file.chkpt2`
การหมุนเวียนไฟล์ checkpoint

เซิร์ฟเวอร์ **DHCP** มีการเธรด เพื่อรักษาปริมาณงานระดับสูง การดำเนินงานฐานข้อมูล (รวมถึง การดำเนินงานบันทึก) จึงเป็นเธรดที่มีประสิทธิภาพ เมื่อร้องขอการบันทึก ไฟล์ checkpoint ที่มีอยู่จะหมุนเวียนไปยังไฟล์ checkpoint ถัดไป ไฟล์ฐานข้อมูลที่มีอยู่ถูกคัดลอกไปยังไฟล์สำเนาสำรอง และมีการสร้างไฟล์บันทึกใหม่ขึ้น จากนั้น มีการบันทึก แต่ละไคลเอ็นต์เร็กคอร์ดและมีการสลับบิตเพื่อบ่งชี้ว่า ไคลเอ็นต์ควรจะใช้ไฟล์ checkpoint ใหม่สำหรับการบันทึก เมื่อบันทึกไคลเอ็นต์เร็กคอร์ดทั้งหมดแล้ว การบันทึกจะปิด และไฟล์สำเนาสำรองและ checkpoint เก่า ถูกลบออก ไคลเอ็นต์ยังคงสามารถถูกประมวลผลได้และ ขึ้นอยู่กับว่าบันทึกไคลเอ็นต์ เร็กคอร์ดแล้วหรือไม่ การเปลี่ยนฐานข้อมูลเข้าสู่ไฟล์บันทึกใหม่หรือเข้าสู่ไฟล์ checkpoint ใหม่

DHCP โปรโตคอลเอ็นจิน:

เอ็นจินโปรโตคอล **DHCP** สนับสนุน RFC 2131 และยังคง เข้ากันได้กับ RFC 1541 (เซิร์ฟเวอร์ยังสามารถประมวลผลอ็อพชันดังที่กำหนด ใน RFC 2132) โปรโตคอลเอ็นจินใช้ฐานข้อมูลเพื่อกำหนดข้อมูลที่จะ ส่งคืนไปยังไคลเอ็นต์

คอนฟิกูเรชันของพูลแอดเดรสมีคอนฟิกูเรชันอ็อพชันบางอย่าง ที่กระทบต่อสถานะของแต่ละเครื่อง ตัวอย่างเช่น เซิร์ฟเวอร์ **DHCP** pings แอดเดรสก่อนจะส่งออกไป ในขณะนี้ สามารถกำหนดคอนฟิกเวลาที่เซิร์ฟเวอร์รอ การตอบกลับสำหรับแต่ละพูลแอดเดรส

การดำเนินงาน DHCP ที่เธรด:

ชิ้นส่วนสุดท้ายของเซิร์ฟเวอร์ **DHCP** แท้จริงแล้วคือ ชุดของการดำเนินงานที่ใช้เพื่อรักษาให้สิ่งต่างๆ รันต่อไป เนื่องจากเซิร์ฟเวอร์ **DHCP** มีการ เธรด การดำเนินงานเหล่านี้จึงมีการตั้งค่าจริงเป็นเธรดที่ จะทำสิ่งต่างๆ ในบางโอกาสเพื่อทำให้แน่ใจว่าทุกสิ่งเข้ากันได้ดี

เธรดแรกซึ่งเป็นเธรด **หลัก** จัดการกับคำร้องขอ SRC (เช่น **startsrc, stopsrc, lssrc, traceson,** และ **refresh**) เธรดนี้ยังประสานการดำเนินงานทั้งหมดที่มีผลต่อเธรดทั้งหมดและจัดการกับ สัญญาณด้วย ตัวอย่างเช่น

- SIGHUP (-1) ส่งผลให้รีเฟรชฐานข้อมูลทั้งหมดในไฟล์คอนฟิกูเรชัน
- SIGTERM (-15) ส่งผลให้เซิร์ฟเวอร์หยุดอย่างเรียบร้อย

เธรดถัดไปคือ **dadmin** ทำหน้าที่อินเตอร์เฟส กับโปรแกรมไคลเอ็นต์ **dadmin** และเซิร์ฟเวอร์ **DHCP** สามารถใช้เครื่องมือ **dadmin** เพื่อเรียกใช้สถานะ และแก้ไขฐานข้อมูลเพื่อหลีกเลี่ยงการแก้ไขไฟล์ฐานข้อมูล ด้วยตนเอง เวอร์ชันก่อนหน้านี้ของเซิร์ฟเวอร์ **DHCP** ป้องกันไม่ให้ ไคลเอ็นต์ใดๆ เรียกใช้แอดเดรสถ้าคำร้องขอสถานะกำลังรัน ด้วยการเพิ่ม เธรด **dadmin** และ **src** เซิร์ฟเวอร์สามารถจัดการกับคำร้องขอเซอร์วิสและยังคงจัดการกับคำร้องขอไคลเอ็นต์ได้

เธรดถัดไปคือเธรด **garbage** ซึ่งรันตัวจับเวลาที่จะทำความสะอาดฐานข้อมูลเป็นระยะๆ บันทึกฐานข้อมูล ล้างข้อมูลไคลเอ็นต์ที่ไม่มีแอดเดรส และลบแอดเดรสที่สงวนไว้ซึ่งอยู่ในสถานะสงวนไว้เป็นเวลานานเกินไป ตัวจับเวลาทั้งหมดเหล่านี้ สามารถกำหนดคอนฟิกได้ (โปรดดูที่ “คอนฟิกูเรชัน DHCP”) เธรดอื่นคือตัวประมวลผลแพ็กเก็ต จำนวนของ เธรดเหล่านี้สามารถกำหนดคอนฟิกได้ ค่าดีฟอลต์คือ 10 แต่ละเธรดสามารถจัดการกับ คำร้องขอจากไคลเอ็นต์ **DHCP** จำนวนของตัวประมวลผลแพ็กเก็ต ที่ต้องการขึ้นอยู่กับโหลดและเครื่อง หากใช้เครื่อง สำหรับเซอร์วิสอื่นที่ไม่ใช่ **DHCP** ไม่ควรสตาร์ทอัพ 500 เธรด

การวางแผน DHCP

เพื่อใช้โปรโตคอลนี้ ผู้ดูแลเครือข่ายต้องตั้งค่า เซิร์ฟเวอร์ **DHCP** และกำหนดคอนฟิก BOOTP รีเลย์เอเจนต์บนลิงก์ที่ไม่ มีเซิร์ฟเวอร์ **DHCP** การวางแผนขั้นสูงสามารถลดโหลด **DHCP** บน เครือข่ายได้

ตัวอย่างเช่น สามารถกำหนดคอนฟิกเซิร์ฟเวอร์หนึ่งเพื่อจัดการกับไคลเอ็นต์ ทั้งหมด แต่ต้องส่งผ่านแพ็กเก็ตทั้งหมดผ่านทางนั้น หากคุณมีเราเตอร์ตัวเดียวระหว่าง เครือข่ายขนาดใหญ่สองเครือข่าย ควรวางเซิร์ฟเวอร์สองตัวในเครือข่าย หนึ่งตัว บนแต่ละลิงก์

ลักษณะอีกอย่างหนึ่งที่ควรพิจารณาคือ **DHCP** แสดงรูปแบบของการจราจร ตัวอย่างเช่น ถ้าคุณตั้งค่าเวลาเช่าดีฟอลต์เป็นน้อยกว่าสองวันและ เครื่องของคุณปิดในวันหยุด เช้าวันจันทร์จะกลายเป็นช่วงเวลาที่มี การจราจร **DHCP** สูง แม้ว่าการจราจร **DHCP** ไม่ได้ทำให้เกิดค่าใช้จ่ายสูงนัก สำหรับเครือข่าย แต่ต้องพิจารณาเมื่อเลือกตำแหน่งในการวาง เซิร์ฟเวอร์ **DHCP** บนเครือข่ายและจำนวนที่จะใช้

หลังจากเปิดใช้งาน **DHCP** เพื่อเรียกใช้ไคลเอ็นต์บนเครือข่าย ไคลเอ็นต์ไม่ต้องป้อนข้อมูลใดๆ ไคลเอ็นต์ **DHCP**, `dhcpcd`, อ่านไฟล์ `dhcpcd.ini` ซึ่งมีข้อมูล เกี่ยวกับการบันทึกและพารามิเตอร์อื่นที่จำเป็นในการเริ่มต้นการรัน หลังจากการติดตั้ง ให้เลือกเมธอดที่จะใช้สำหรับคอนฟิกูเรชัน **TCP/IP**: คอนฟิกูเรชันต่ำสุด หรือ **DHCP** ถ้าเลือก **DHCP** ให้เลือกอินเตอร์เฟสและระบุ พารามิเตอร์ทางเลือกบางตัว เมื่อต้องการเลือกอินเตอร์เฟส ให้เลือกคีย์เวิร์ด **any** ซึ่งบอกให้ `dhcpcd` ค้นหาอินเตอร์เฟสแรกที่ทำงาน และใช้อินเตอร์เฟสนั้น เมธอดนี้ลดจำนวนอินพุตบนด้านไคลเอ็นต์

คอนฟิกูเรชัน DHCP

โดยค่าดีฟอลต์ เซิร์ฟเวอร์ **DHCP** มีการกำหนดคอนฟิกโดยการอ่านไฟล์ `/etc/dhcpsd.cnf` ซึ่งระบุฐานข้อมูลแรกเริ่มของอ็อพชันและแอดเดรส

เซิร์ฟเวอร์เริ่มต้นขึ้นในไฟล์ `/etc/rc.tcpip` ทั้งสามารถเริ่มทำงานจาก SMIT หรือผ่านคำสั่ง SRC ไคลเอ็นต์ **DHCP** สามารถถูกกำหนดคอนฟิกโดยการรัน System Management Interface Tool (SMIT) หรือการแก้ไขไฟล์ flat ASCII

โดยปกติ การกำหนดคอนฟิกเซิร์ฟเวอร์ **DHCP** เป็นส่วนที่ยากที่สุดของการใช้ **DHCP** ใน เครือข่าย อันดับแรก ตัดสินใจเลือกเครือข่ายซึ่งคุณต้องการให้มีไคลเอ็นต์ **DHCP** แต่ละ subnet ในเครือข่ายแสดงถึงพูลของแอดเดรสที่เซิร์ฟเวอร์ **DHCP** ต้องเพิ่มลงในฐานข้อมูล ตัวอย่างเช่น:

```
database db_file
{

    subnet 9.3.149.0 255.255.255.0
      {  option 3 9.3.149.1 # The default gateway clients on this network should use
         option 6 9.3.149.2 # The nameserver clients on this network should use
      }
    ... options or other containers added later
}
```

ตัวอย่างข้างบนแสดง subnet, `9.3.149.0`, ที่มี subnet mask `255.255.255.0` แอดเดรสทั้งหมดใน subnet นี้ตั้งแต่ 9.3.149.1 ถึง 9.3.149.254 อยู่ในพูล หรือสามารถเลือกที่จะระบุช่วง ที่ตอนท้ายของบรรทัด หรือสามารถรวมช่วงหรือคำสั่ง exclude ไว้ใน คอนเทนเนอร์ subnet โปรดดูที่ “อ็อพชันที่รู้จักของไฟล์เซิร์ฟเวอร์ DHCP” ในหน้า 249 สำหรับ เมธอดคอนฟิกูเรชันทั่วไปและนิยาม

ส่วนคำสั่งฐานข้อมูลที่มี `db_file` บ่งชี้เมธอดฐานข้อมูล ที่จะใช้สำหรับการประมวลผลไฟล์คอนฟิกูเรชันส่วนนี้ ข้อคิดเห็น ขึ้นต้นด้วย # (เครื่องหมายสี่เหลี่ยม) ข้อความตั้งแต่ # จนถึงตอนท้ายของบรรทัด ถูกละเว้นโดยเซิร์ฟเวอร์ **DHCP** เซิร์ฟเวอร์ใช้แต่ละบรรทัด `option` เพื่อบอกสิ่งที่ ไคลเอ็นต์ต้องทำ “อ็อพชันที่รู้จักของไฟล์เซิร์ฟเวอร์ DHCP” ในหน้า 249 อธิบาย อ็อพชันที่ได้รับการสนับสนุนและรู้จักในปัจจุบัน โปรดดูที่ “ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับการดำเนินงานเซิร์ฟเวอร์ทั่วไป” ในหน้า 252 สำหรับ วิธีการระบุอ็อพชันที่เซิร์ฟเวอร์ไม่ทราบ

หากเซิร์ฟเวอร์ไม่เข้าใจวิธีการแจงส่วนอ็อพชัน เซิร์ฟเวอร์จะใช้ดีฟอลต์เมธอด เพื่อส่งอ็อพชันไปยังไคลเอ็นต์ ซึ่งเซิร์ฟเวอร์ **DHCP** ยังสามารถส่ง อ็อพชันเฉพาะไซต์ที่ไม่ได้กำหนด RFC แต่สามารถใช้ ไคลเอ็นต์หรือคอนฟิกูเรชันไคลเอ็นต์บางรายการได้ด้วย

ไฟล์คอนฟิกูเรชัน DHCP:

ไฟล์คอนฟิกูเรชันมีส่วนแอดเดรสและส่วนนิยาม อ็อพชัน ส่วนเหล่านี้ใช้คอนเทนเนอร์เพื่อจัดเก็บอ็อพชัน ตัวแก้ไข และอาจมีคอนเทนเนอร์อื่น

คอนเทนเนอร์ (โดยพื้นฐาน เมธอดการจัดกลุ่มอ็อพชัน) ใช้ตัวระบุเพื่อจัดประเภทไคลเอ็นต์เป็นกลุ่มต่างๆ ชนิดคอนเทนเนอร์คือ subnet, คลาส, ผู้ขาย, และไคลเอ็นต์ ในปัจจุบัน ไม่มีคอนเทนเนอร์ทั่วไปซึ่งผู้ใช้สามารถ กำหนดได้ ตัวระบุกำหนดไคลเอ็นต์โดยไม่ซ้ำกันเพื่อให้สามารถติดตามไคลเอ็นต์ ได้ถ้า ตัวอย่างเช่น ย้ายระหว่าง subnets สามารถใช้คอนเทนเนอร์ได้ มากกว่าหนึ่งชนิดเพื่อกำหนดการเข้าถึงไคลเอ็นต์

อ็อพชัน คือตัวระบุที่ถูกส่งคืนไปยังไคลเอ็นต์ เช่น ดีฟอลต์เกตเวย์และ DNS แอดเดรส

ตัวแก้ไข เป็นคำสั่งเดียวที่แก้ไขลักษณะบางอย่างของคอนเทนเนอร์ เช่น เวลาเช่าดีฟอลต์

คอนเทนเนอร์ DHCP:

เมื่อเซิร์ฟเวอร์ **DHCP** ได้รับคำร้องขอ จะมีการแจงส่วนแพ็กเก็ต และคีย์การระบุกำหนดคอนเทนเนอร์ อ็อพชัน และแอดเดรสที่จะ แยก

ตัวอย่างใน “คอนฟิกูเรชัน DHCP” ในหน้า 239 แสดง คอนเทนเนอร์ subnet คีย์การระบุคือตำแหน่งของไคลเอ็นต์ในเครือข่าย หากไคลเอ็นต์มาจากเครือข่ายนั้น ไคลเอ็นต์จะอยู่ในคอนเทนเนอร์นั้น

คอนเทนเนอร์แต่ละชนิดใช้อ็อพชันที่แตกต่างกันในการระบุไคลเอ็นต์:

- คอนเทนเนอร์ subnet ใช้ฟิลด์ **giaddr** หรือ อินเตอร์เฟสแอดเดรสของอินเตอร์เฟสที่ได้รับเพื่อกำหนด subnet ต้นทางของไคลเอ็นต์
- คลาสคอนเทนเนอร์ใช้ค่าในอ็อพชัน 77 (ตัวระบุคลาสของไซต์ผู้ใช้)
- ผู้ขายใช้ค่าในอ็อพชัน 60 (ตัวระบุคลาสของผู้ขาย)
- ไคลเอ็นต์คอนเทนเนอร์ใช้อ็อพชัน 61 (ตัวระบุไคลเอ็นต์) สำหรับไคลเอ็นต์ **DHCP** และฟิลด์ **chaddr** ในแพ็กเก็ต **BOOTP** สำหรับไคลเอ็นต์ **BOOTP**

ยกเว้นสำหรับ subnets แต่ละคอนเทนเนอร์อนุญาตการระบุค่า ที่ตรงกัน รวมถึงการจับคู่นิพจน์ปกติ

และยังมีคอนเทนเนอร์โดยปริยายคือ คอนเทนเนอร์ *สากล* อ็อพชัน และตัวแก้ไขถูกวางไว้ในคอนเทนเนอร์สากล ยกเว้นว่าถูกยกเลิกหรือปฏิเสธ คอนเทนเนอร์ส่วนใหญ่สามารถวางไว้ภายในคอนเทนเนอร์อื่น ที่ใช้ขอบเขตของการมองเห็นได้ คอนเทนเนอร์อาจหรืออาจไม่มีช่วงแอดเดรส ที่เชื่อมโยง Subnets โดยธรรมชาติแล้ว มีช่วงที่เชื่อมโยงด้วย

กฎพื้นฐานสำหรับคอนเทนเนอร์และคอนเทนเนอร์ย่อยมีดังนี้:

- คอนเทนเนอร์ทั้งหมดถูกต้องที่ระดับสากล
- Subnets ไม่สามารถวางไว้ภายในคอนเทนเนอร์อื่น
- คอนเทนเนอร์ที่จำกัดไม่สามารถมีคอนเทนเนอร์ปกติชนิดเดียวกัน อยู่ภายใน (ตัวอย่างเช่น คอนเทนเนอร์ที่มีอ็อพชันที่อนุญาตเฉพาะคลาส `Accounting` ไม่สามารถมีคอนเทนเนอร์ที่มีอ็อพชันซึ่ง อนุญาตทุกคลาสที่ขึ้นต้นด้วยตัวอักษร ″a″ นี่ไม่ถูกต้อง)
- ไคลเอ็นต์คอนเทนเนอร์ที่จำกัดไม่สามารถมีคอนเทนเนอร์ย่อย

ภายใต้กฎข้างบน คุณสามารถสร้างลำดับชั้นของคอนเทนเนอร์ที่ แบ่งเซกเมนต์อ็อพชันของคุณออกเป็นกลุ่มต่างๆ สำหรับไคลเอ็นต์หรือชุดของไคลเอ็นต์เฉพาะ

หากไคลเอ็นต์ตรงกับหลายคอนเทนเนอร์ จะจัดการกับอ็อพชันและแอดเดรส อย่างไร? เซิร์ฟเวอร์ **DHCP** ได้รับข้อความ ส่งผ่านคำร้องขอ ไปยังฐานข้อมูล (`db_file` ในกรณีนี้) และมีการสร้างรายการ คอนเทนเนอร์ขึ้น รายการแสดงขึ้นในลำดับของความลึกและระดับความสำคัญ ระดับความสำคัญ มีการกำหนดเป็นลำดับชั้นปริยายในคอนเทนเนอร์ คอนเทนเนอร์จำกัดมีระดับ ความสำคัญสูงกว่าคอนเทนเนอร์ปกติ ไคลเอ็นต์ คลาส ผู้ขาย และสุดท้าย subnets มีการเรียงลำดับในลำดับนั้น และภายในคอนเทนเนอร์ชนิดเดียวกันมีการเรียงตามความลึก ซึ่ง สร้างรายการที่เรียงลำดับตามข้อมูลเฉพาะมากที่สุดไปยังน้อยที่สุด ตัวอย่างเช่น:

```
Subnet 1
--Class 1
--Client 1
Subnet 2
--Class 1
----Vendor 1
----Client 1
--Client 1
```

ตัวอย่างแสดงสอง subnets, Subnet 1 และ Subnet 2 มีหนึ่งชื่อคลาส, Class 1, หนึ่งชื่อผู้ขาย, Vendor 1, และหนึ่งชื่อไคลเอ็นต์, Client 1 Class 1 และ Client 1 มีการกำหนดในหลายที่ เนื่องจาก อยู่ในคอนเทนเนอร์ที่แตกต่างกัน ชื่ออาจเหมือนกันได้แต่ค่าภายใน อาจแตกต่างกัน หาก Client 1 ส่งข้อความไปยังเซิร์ฟเวอร์ **DHCP** จาก Subnet 1 โดยมีการระบุ Class 1 ใน รายการอ็อพชัน เซิร์ฟเวอร์ **DHCP** อาจสร้างพาธคอนเทนเนอร์ ต่อไปนี้:

```
Subnet 1, Class 1, Client 1
```

คอนเทนเนอร์เฉพาะที่สุดแสดงอยู่ในลำดับสุดท้าย เพื่อให้ได้แอดเดรส รายการ จะถูกตรวจสอบในลำดับชั้นย้อนกลับเพื่อค้นหาแอดเดรสแรกที่มีอยู่ จากนั้น จะตรวจสอบรายการในลำดับชั้นไปข้างหน้าเพื่อให้ได้อ็อพชัน อ็อพชันยกเลิก ค่าก่อนหน้านี้ยกเว้นว่ามีอ็อพชัน **deny** อยู่ในคอนเทนเนอร์ นอกจากนี้ เนื่องจาก Class 1 และ Client 1 อยู่ใน Subnet 1 จะมีการเรียงลำดับตามระดับความสำคัญของคอนเทนเนอร์ หากไคลเอ็นต์ เดียวกันอยู่ใน Subnet 2 และส่งข้อความเดียวกัน รายการคอนเทนเนอร์ ที่สร้างขึ้นคือ:

```
Subnet 2, Class 1, Client 1 (ที่ระดับ Subnet 2), Client 1 (ที่ระดับ Class 1)
```

Subnet 2 แสดงขึ้นก่อน ตามด้วย Class 1 ตามด้วย Client 1 ที่ระดับ Subnet 2 (เนื่องจากคำสั่งไคลเอ็นต์นี้ต่ำลงเพียงหนึ่งระดับในลำดับชั้น) ลำดับชั้น ตีความว่าไคลเอ็นต์ที่ตรงกับคำสั่งไคลเอ็นต์แรกมีความเฉพาะ น้อยกว่าไคลเอ็นต์ที่ตรงกับ Client 1 ของ Class 1 ภายใน Subnet 2

ระดับความสำคัญที่เลือกโดยความลึกภายในลำดับชั้นไม่ได้ถูกแทนที่โดย ระดับความสำคัญของตัวคอนเทนเนอร์เอง ตัวอย่างเช่น ถ้าไคลเอ็นต์เดียวกันออกใช้ข้อความเดียวกันและระบุตัวระบุผู้ขาย รายการคอนเทนเนอร์จะเป็น:

```
Subnet 2, Class 1, Vendor 1, Client 1 (ที่ระดับ Subnet 2), Client 1 (ที่ระดับ Class 1)
```

ระดับความสำคัญคอนเทนเนอร์ช่วยพัฒนาประสิทธิภาพการค้นหา เนื่องจาก เป็นไปตามแนวคิดทั่วไปที่ไคลเอ็นต์คอนเทนเนอร์เป็นวิธีเฉพาะที่สุดในการกำหนด หนึ่งไคลเอ็นต์ขึ้นไป คลาสคอนเทนเนอร์มีแอดเดรสเฉพาะน้อยกว่าไคลเอ็นต์ คอนเทนเนอร์ ผู้ขายมีความเฉพาะน้อยไปอีก และ subnet มีความเฉพาะน้อยที่สุด

DHCP แอดเดรสและช่วงแอดเดรส:

ชนิดคอนเทนเนอร์ต่างๆ สามารถมีช่วงแอดเดรสที่เชื่อมโยง และ subnets ต้องมีช่วงแอดเดรสที่เชื่อมโยง แต่ละช่วงภายในคอนเทนเนอร์ต้องเป็นชุดย่อยของช่วง และต้องไม่ซ้อนเหลื่อมกับช่วงของคอนเทนเนอร์อื่น

ตัวอย่างเช่น ถ้าคลาสถูกกำหนดไว้ภายใน subnet และคลาสมีช่วง ช่วงต้องเป็นชุดย่อยของช่วงของ subnet นอกจากนี้ ช่วงภายในคลาสคอนเทนเนอร์นั้น ต้องไม่ซ้อนเหลื่อมกับช่วงอื่นใดๆ ที่ระดับ

ช่วงสามารถระบุได้บนบรรทัดคอนเทนเนอร์และแก้ไขโดยใช้ช่วง และคำสั่ง exclude เพื่อให้สามารถแยกชุดแอดเดรสที่เชื่อมโยงกับคอนเทนเนอร์ได้ ดังนั้น ถ้าคุณมีแอดเดรสสิบรายการแรกและสิบรายการที่สองของ subnet อยู่ subnet สามารถระบุแอดเดรสเหล่านี้โดยใช้ช่วงในส่วนคำสั่ง subnet เพื่อลดทั้งการใช้หน่วยความจำและโอกาสการปะทะของแอดเดรสกับ ไคลเอ็นต์อื่นที่ไม่ได้อยู่ในช่วงที่ระบุ

หลังจากเลือกแอดเดรสแล้ว คอนเทนเนอร์ในลำดับต่อมาใดๆ ในรายการ ที่มีช่วงแอดเดรสจะถูกลบออกจากรายการพร้อมกับชายด์ เหตุผลที่เป็นเช่นนี้คือ อ็อพชันเฉพาะเครือข่ายในคอนเทนเนอร์ที่ลบออก ไม่ถูกต้องถ้าไม่ได้ใช้แอดเดรสจากภายในคอนเทนเนอร์นั้น

อ็อพชันไฟล์คอนฟิกูเรชัน DHCP:

หลังจากรายการถูกเลือกเพื่อกำหนดแอดเดรส ชุดของอ็อพชัน จะถูกสร้างขึ้นสำหรับไคลเอ็นต์

ในขั้นตอนการเลือกนี้ อ็อพชันจะเขียนทับอ็อพชันที่เลือกก่อนหน้า จนกว่าจะตรวจพบ *ปฏิเสธ* ซึ่งในกรณีดังกล่าว อ็อพชันที่ปฏิเสธจะถูกลบ ออกจากรายการที่ส่งให้กับไคลเอ็นต์ วิธีนี้อนุญาตให้สืบทอดจาก คอนเทนเนอร์หลักเพื่อลดปริมาณข้อมูลที่ต้องระบุ

ตัวแก้ไข DHCP:

ตัวแก้ไขคือไอเท็มที่เปลี่ยนลักษณะบางอย่างของคอนเทนเนอร์เฉพาะ เช่น การเข้าถึง หรือเวลาเช่า

กำหนดพูลแอดเดรสและอ็อพชันก่อนการแก้ไขคอนเทนเนอร์ ตัวแก้ไข ที่ใช้กันมากที่สุดคือ **leasetimedefault, supportBootp,** และ **supportUnlistedclients**

leasetimedefault
: กำหนดเวลาที่เช่าแอดเดรสให้กับไคลเอ็นต์

supportBootp
: กำหนดว่าเซิร์ฟเวอร์จะตอบกลับไคลเอ็นต์ **BOOTP** หรือไม่

supportUnlistedclients
: บ่งชี้ว่าไคลเอ็นต์จะถูกกำหนดอย่างชัดแจ้งโดยคำสั่งไคลเอ็นต์ เพื่อรับแอดเดรสหรือไม่ ค่าสำหรับ **supportUnlistedClients** สามารถ เป็น **none, dhcp, bootp,** หรือ **both** ซึ่งช่วยให้คุณสามารถจำกัดการเข้าถึงใน ไคลเอ็นต์ bootp และอนุญาตให้ไคลเอ็นต์ DHCP ทั้งหมดเรียกใช้แอดเดรสได้

ตัวแก้ไขอื่นมีการแสดงรายการอยู่ใน “ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับฐานข้อมูล db_file” ในหน้า 257

การบันทึก DHCP:

หลังจากเลือกตัวแก้ไขแล้ว ไอเท็มถัดไปที่จะตั้งค่าคือการบันทึก

พารามิเตอร์การบันทึกมีการระบุในคอนเทนเนอร์เช่นเดียวกับฐานข้อมูล แต่คอนเทนเนอร์คีย์เวิร์ดคือ **logging_info** เมื่อศึกษาเพื่อกำหนดคอนฟิก **DHCP** ขอแนะนำให้เปิดการบันทึกเป็น ระดับสูงสุด นอกจากนี้ สิ่งที่ดีที่สุดคือการระบุคอนฟิกูเรชันการบันทึกก่อนหน้า ข้อมูลไฟล์คอนฟิกูเรชันอื่นใด เพื่อให้มั่นใจว่าจะมีการบันทึกข้อผิดพลาด คอนฟิกูเรชันหลังจากเริ่มต้นระบบย่อยการบันทึกแล้ว ใช้คีย์เวิร์ด **logitem** เพื่อ เปิดระดับการบันทึกหรือลบคีย์เวิร์ด **logitem** เพื่อ ปิดใช้งานระดับการบันทึก คีย์เวิร์ดอื่นสำหรับการบันทึกช่วยให้สามารถระบุ ชื่อไฟล์บันทึก ขนาดไฟล์ และจำนวนของไฟล์บันทึกที่หมุนเวียน

อ็อพชันเฉพาะเซิร์ฟเวอร์ DHCP:

ชุดสุดท้ายของพารามิเตอร์ที่จะระบุคืออ็อพชันเฉพาะเซิร์ฟเวอร์ ซึ่งช่วยให้ผู้ใช้สามารถควบคุมจำนวนของตัวประมวลผลแพ็กเก็ต ความบ่อย ในการรันของเธรดการรวบรวมขยะ และอื่นๆ

ตัวอย่างเช่น อ็อพชันเฉพาะเซิร์ฟเวอร์สองอ็อพชันคือ:

reservedTime
: บ่งชี้ระยะเวลาที่แอดเดรสอยู่ในสถานะที่สงวนไว้หลังจากการส่ง OFFER ไปยังไคลเอ็นต์ **DHCP**

reservedTimeInterval

บ่งชี้ความบ่อยที่เซิร์ฟเวอร์ **DHCP** สแกนผ่านแอดเดรส เพื่อดูว่ามีรายการใดๆ ซึ่งอยู่ในสถานะที่สงวนไว้นานกว่า **reservedTime** หรือไม่

อ็อพชันเหล่านี้มีประโยชน์ถ้าคุณมีหลายไคลเอ็นต์ที่แพร่สัญญาณข้อความ DISCOVER และไม่ได้แพร่สัญญาณข้อความ REQUEST หรือข้อความ REQUEST หายไปในเครือข่าย อย่างใดอย่างหนึ่ง การใช้พารามิเตอร์เหล่านี้ช่วยไม่ให้แอดเดรสถูกสำรองไว้โดยไม่สิ้นสุดสำหรับไคลเอ็นต์ที่ไม่ได้ใช้งาน

อีกอ็อพชันหนึ่งที่มีประโยชน์มากคือ **SaveInterval** ซึ่งบ่งชี้ความบ่อยในการบันทึก อ็อพชันเฉพาะเซิร์ฟเวอร์ทั้งหมดแสดงอยู่ใน “ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับการดำเนินงานเซิร์ฟเวอร์ทั่วไป” ในหน้า 252 พร้อมกับ คีย์เวิร์ดการบันทึก

ข้อควรพิจารณาเกี่ยวกับประสิทธิภาพ DHCP:

สิ่งสำคัญคือการทำความเข้าใจว่าคอนฟิกูเรชันคีย์เวิร์ดต่างๆ และโครงสร้างของไฟล์คอนฟิกูเรชันมีผลกระทบต่อการใช้หน่วยความจำ และประสิทธิภาพของเซิร์ฟเวอร์ **DHCP** อย่างไร

อันดับแรก การใช้หน่วยความจำที่มากเกินความจำเป็นสามารถหลีกเลี่ยงได้โดยการ ทำความเข้าใจกับโมเดลการสืบทอดของอ็อพชันจากคอนเทนเนอร์พาเรนต์ไปยังชายด์ ในสภาวะแวดล้อมซึ่งสนับสนุน ไคลเอ็นต์ที่ไม่มีการแสดงรายการ ผู้ดูแลระบบต้องแสดงรายการแต่ละไคลเอ็นต์ใน ไฟล์อย่างชัดแจ้ง เมื่อแสดงรายการอ็อพชันสำหรับไคลเอ็นต์เฉพาะ เซิร์ฟเวอร์จะใช้ หน่วยความจำเพื่อจัดเก็บแผนผังคอนฟิกูเรชันนั้นมากกว่าเมื่อได้รับสืบทอดอ็อพชัน จากคอนเทนเนอร์พาเรนต์ (ตัวอย่างเช่น คอนเทนเนอร์ subnet, เครือข่าย, หรือสากล) ดังนั้น ผู้ดูแลระบบจึงควรตรวจสอบว่าอ็อพชันใดมีการทำซ้ำ ที่ระดับไคลเอ็นต์ภายในไฟล์คอนฟิกูเรชันหรือไม่ และถ้ามี ควรพิจารณาว่า สามารถระบุอ็อพชันดังกล่าวในคอนเทนเนอร์พาเรนต์และแบ่งใช้โดยชุด ของไคลเอ็นต์โดยรวมได้หรือไม่

นอกจากนี้ เมื่อใช้รายการ **logItem** INFO และ TRACE จะมีการ บันทึกข้อความเป็นจำนวนมากในระหว่างการประมวลผลทุกข้อความของไคลเอ็นต์ **DHCP** การผนวกบรรทัดเข้ากับไฟล์บันทึกอาจเป็นการดำเนินงานที่มีค่าใช้จ่ายสูง ด้วยเหตุนี้ การจำกัดจำนวนของการบันทึกจึงช่วยปรับปรุงประสิทธิภาพของเซิร์ฟเวอร์ **DHCP** ได้ หากสงสัยว่ามีข้อผิดพลาดเกี่ยวกับเซิร์ฟเวอร์ **DHCP** สามารถเปิดใช้งานการ บันทึกอีกครั้งแบบไดนามิกโดยใช้คำสั่ง SRC **traceson** หรือ **dadmin**

สุดท้าย การเลือกค่า **numprocessors** ขึ้นอยู่กับ ขนาดของเครือข่ายที่ **DHCP** สนับสนุน พารามิเตอร์คอนฟิกูเรชัน **pingTime** **db_file** และเวลาหน่วงการแพร่กระจายปกติ บนเครือข่าย เนื่องจากแต่ละเธรดตัวประมวลผลแพ็กเก็ตออกใช้ ICMP Echo Request เพื่อตรวจสอบสถานะของแอดเดรสของเซิร์ฟเวอร์ก่อนนำเสนอให้กับ ไคลเอ็นต์ ระยะเวลาที่รอ Echo Response จึงส่งผลกระทบโดยตรง กับระยะเวลาการประมวลผลสำหรับข้อความ DISCOVER สิ่งสำคัญคือ เธรดตัวประมวลผลแพ็กเก็ตไม่สามารถทำอะไรได้มากไปกว่าการรอการตอบกลับ หรือรอไทม์เอาต์ **pingTime** การลดค่า **numprocessors** ช่วยปรับปรุงเวลาตอบกลับของเซิร์ฟเวอร์โดยการลดจำนวนของการส่งผ่าน ไคลเอ็นต์อีกครั้ง แต่ยังคงรักษาประโยชน์ ping ของการออกแบบเซิร์ฟเวอร์ไว้

สำหรับประสิทธิภาพที่ดีที่สุด ให้เลือก **pingTime** ตามข้อมูล เวลาหน่วงการแพร่กระจายของเครือข่ายรีโมตใดๆ ที่สนับสนุนโดยเซิร์ฟเวอร์ **DHCP** นอกจากนี้ ให้เลือกค่า **numprocessors** ตามข้อมูลค่า **pingTime** นี้ และขนาดของเครือข่าย การเลือกค่าที่น้อยเกินไปสามารถส่งผลให้ เธรดการประมวลผลแพ็กเก็ตทั้งหมดหยุด จากนั้น ทำให้เซิร์ฟเวอร์ต้อง รอ Echo Responses ในขณะที่กำลังจัดคิวข้อความไคลเอ็นต์ **DHCP** ขาเข้า บนเซิร์ฟเวอร์พอร์ต ซึ่งส่งผลให้เซิร์ฟเวอร์จัดการกับข้อความไคลเอ็นต์ในแบตช์แทนสตรีมคงที่

การเลือกค่าที่น้อยเกินไปสามารถส่งผลให้เธรดการประมวลผลแพ็กเก็ตทั้งหมด หยุดรอ Echo Responses

เพื่อป้องกันสถานการณ์นี้ ให้ตั้งค่า **numprocessors** เป็น ตัวเลขที่สูงกว่าจำนวนที่ประเมินของข้อความ DISCOVER ซึ่งสามารถรับได้ภายในหนึ่งช่วงเวลา **pingTime** ในระหว่างระยะเวลา ที่มีกิจกรรมไคลเอ็นต์ **DHCP** สูง อย่างไรก็ตาม อย่าตั้งค่า **numprocessors** สูงจน สร้างภาระให้กับเคอร์นัลที่มีการจัดการเธรด

ตัวอย่างเช่น ค่า **numprocessors 5** และ **pingTime 300** ส่งผลให้ประสิทธิภาพไม่ดีในสภาวะแวดล้อมที่อาจมี 10 ข้อความ DISCOVER ต่อวินาทีเนื่องจากในเวลาที่มีความต้องการสูงสุด จะมีการจัดการเพียง 5 ข้อความ ทุก 3 วินาที กำหนดคอนฟิกสภาวะแวดล้อมนี้ด้วยค่าคล้ายกับ **numprocessors 20** และ **pingTime 80**

การกำหนดไฟล์คอนฟิกูเรชัน DHCP เอง:

มีปัจจัยหลายอย่างที่เกี่ยวข้องในกำหนดไฟล์คอนฟิกูเรชัน **DHCP** ของคุณเอง

เครือข่ายจำนวนมากมีไคลเอ็นต์อยู่หลายชนิด ตัวอย่างเช่น เครือข่ายหนึ่ง อาจมีคอมพิวเตอร์หลายเครื่องซึ่งกำลังรันระบบปฏิบัติการหลากหลาย เช่น Windows, OS/2, Java™ OS, และ UNIX แต่ละระบบปฏิบัติการเหล่านี้ต้องการตัวระบุผู้ขายที่ไม่ซ้ำกัน (ฟิลด์ที่ใช้ในการระบุ ชนิดของเครื่องที่เซิร์ฟเวอร์ DHCP) Java OS ไคลเอ็นต์และเครื่อง IBM Thin Client สามารถต้องการพารามิเตอร์ที่ไม่ซ้ำกัน เช่น bootfiles และคอนฟิกูเรชันอ็อพชันที่ต้องปรับแต่ง สำหรับเครื่องโดยเฉพาะ คอมพิวเตอร์ Windows 95 ไม่ได้จัดการอ็อพชันเฉพาะ Java

อ็อพชันเฉพาะเครื่องอาจมีอยู่ภายในคอนเทนเนอร์ผู้ขายถ้า การใช้งานหลักของเครื่องขึ้นอยู่กับชนิดของผู้ใช้สำหรับเครื่องเหล่านั้น ตัวอย่างเช่น ทีมงานฝ่ายพัฒนาอาจใช้ไคลเอ็นต์ของระบบปฏิบัติการนี้ สำหรับการเขียนโปรแกรม ทีมงานฝ่ายการตลาดอาจใช้ไคลเอ็นต์ OS/2 ฝ่ายขายอาจใช้ Java OS ไคลเอ็นต์และเครื่อง IBM Thin Client และฝ่ายบัญชี อาจใช้เครื่อง Windows 95 เพื่อนร่วมงานของผู้ใช้ในแต่ละฝ่ายเหล่านี้ อาจต้องการคอนฟิกูเรชันอ็อพชันที่แตกต่างกัน (เครื่องพิมพ์ เนมเซิร์ฟเวอร์ หรือดีฟอลต์เว็บเซิร์ฟเวอร์ที่แตกต่างกัน และอื่นๆ) ในกรณีนี้ สามารถรวมอ็อพชันดังกล่าว ไว้ในคอนเทนเนอร์ผู้ขายได้ เนื่องจากแต่ละกลุ่มใช้ชนิดเครื่อง ที่แตกต่างกัน

หากหลายกลุ่มใช้เครื่องชนิดเดียวกัน การวางอ็อพชันไว้ภายใน ตัวระบุคลาสรองแทน จะช่วยให้ผู้จัดการฝ่ายการตลาด ตัวอย่างเช่น สามารถใช้ชุดของเครื่องพิมพ์ซึ่งพนักงานรายอื่น ไม่สามารถเข้าถึงได้

หมายเหตุ: ตัวอย่างที่สมมติขึ้นต่อไปนี้แสดงถึงส่วนของไฟล์คอนฟิกูเรชัน ข้อคิดเห็นมีเครื่องหมายสี่เหลี่ยม (#) นำหน้าและอธิบายวิธีการกำหนดการติดตั้ง ของแต่ละบรรทัด

```
vendor "AIX_CLIENT"
{
# No specific options, handles things based on class
}

vendor "OS/2 Client"
{
# No specific options, handles things based on class
}

vendor "Windows 95"
{ option 44 9.3.150.3          # Default NetBIOS Nameserver
}

vendor "Java OS"
{ bootstrapserver 9.3.150.4    # Default TFTP server for the Java OS boxes
  option 67 "javaos.bin"       # The bootfile of the Java OS box
}
```

```
vendor "IBM Thin Client"
{ bootstrapserver 9.3.150.5    # Default TFTP server for Thin Client boxes
  option 67 "thinos.bin"       # Default bootfile for the Thin Client boxes
}

subnet 9.3.149.0 255.255.255.0
{ option 3 9.3.149.1           # The default gateway for the subnet
  option 6 9.3.150.2           # This is the nameserver for the subnet
  class accounting 9.3.149.5-9.3.149.20
  {       # The accounting class is limited to address range 9.3.149.5-9.3.149.20
          # The printer for this group is also in this range, so it is excluded.
     exclude 9.3.149.15
     option 9 9.3.149.15       # The LPR server (print server)
     vendor "Windows 95"
     {
     option 9 deny              # This installation of Windows 95 does not support
                                # this printer, so the option is denied.
     }
  }
 . . .
}
```

DHCP และ Dynamic Domain Name System

เซิร์ฟเวอร์ **DHCP** นำเสนออ็อพชันที่ช่วยให้สามารถดำเนินงานใน สภาวะแวดล้อม Dynamic Domain Name System (DDNS) ได้

เพื่อใช้ **DHCP** ในสภาวะแวดล้อม DDNS คุณต้องตั้งค่าและใช้ Dynamic Zone บนเซิร์ฟเวอร์ DNS

หลังจากกำหนดคอนฟิกเซิร์ฟเวอร์ DDNS แล้ว ให้เลือกว่าเซิร์ฟเวอร์ **DHCP** จะ ทำการอัพเดตเร็กคอร์ด A, อัพเดตเร็กคอร์ด PTR, อัพเดตเร็กคอร์ด ทั้งสองชนิด, หรือไม่อัพเดตเลย การตัดสินใจนี้ขึ้นอยู่กับว่าเครื่องไคลเอ็นต์สามารถ ทำบางส่วนหรือทั้งหมดของงานนี้

- หากไคลเอ็นต์สามารถแบ่งความรับผิดชอบในการอัพเดต ให้กำหนดคอนฟิก เซิร์ฟเวอร์เพื่อทำอัพเดตเร็กคอร์ด PTR และกำหนดคอนฟิกไคลเอ็นต์เพื่อทำอัพเดตเร็กคอร์ด A
- ถ้าไคลเอ็นต์สามารถทำทั้งสองอัพเดต ให้กำหนดคอนฟิกเซิร์ฟเวอร์เพื่อไม่ทำสิ่งใดเลย
- ถ้าไคลเอ็นต์ไม่สามารถทำอัพเดต ให้กำหนดคอนฟิกเซิร์ฟเวอร์เพื่อทำทั้งสองอย่าง

เซิร์ฟเวอร์ **DHCP** มีชุดของคีย์เวิร์ดคอนฟิกูเรชันซึ่งช่วยให้คุณ สามารถระบุคำสั่งที่จะรันเมื่อต้องการอัพเดต ซึ่งคือ:

updatedns
(ถูกคัดค้าน) แสดงถึงคำสั่งที่จะออกใช้เพื่อทำอัพเดตชนิดต่างๆ มีการเรียกสำหรับอัพเดตทั้งเร็กคอร์ด PTR และเร็กคอร์ด A

updatednsA
ระบุคำสั่งที่จะอัพเดตเร็กคอร์ด A

updatednsP
ระบุคำสั่งที่จะอัพเดตเร็กคอร์ด PTR

คีย์เวิร์ดเหล่านี้ระบุสตริงปฏิบัติการที่เซิร์ฟเวอร์ **DHCP** จะรัน เมื่อต้องการอัพเดต คีย์เวิร์ดสตริงต้องมีสี่ %s (สัญลักษณ์เปอร์เซ็นต์, ตัวอักษร s) %s ลำดับแรกคือชื่อโฮสต์ ลำดับที่สอง คือชื่อโดเมน ลำดับที่สามคือ IP แอดเดรส และลำดับที่สี่คือเวลาเช่า ค่าเหล่านี้ใช้เป็นพารามิเตอร์สี่ตัวแรกสำหรับคำสั่ง **dhcpaction** พารามิเตอร์อีกสองตัวที่เหลือสำหรับคำสั่ง **dhcpaction** บ่งชี้เร็กคอร์ดที่จะอัพเดต (A, PTR, NONE, หรือ BOTH) และควรจะอัพเดต NIM หรือไม่ (NIM หรือ NONIM) โปรดดูที่ "คำแนะนำเกี่ยวกับ DHCP และ Network Installation Management" ในหน้า 301 สำหรับ ข้อมูลเพิ่มเติมเกี่ยวกับการโต้ตอบ NIM และ **DHCP** ตัวอย่างเช่น:

```
updatednsA "/usr/sbin/dhcpaction '%s' '%s' '%s' '%s' '%s' A NONIM"
                  # This does the dhcpaction command only on the A record
updatednsP "/usr/sbin/dhcpaction '%s' '%s' '%s' '%s' '%s' PTR NONIM"
                  # This does the command only on the PTR record
updatedns "/usr/sbin/dhcpaction '%s' '%s' '%s' '%s' '%s' BOTH NIM"
                  # This does the command on both records and updates NIM
```

เซิร์ฟเวอร์ **DHCP** ยังมีชุดของคีย์เวิร์ดที่จะลบรายการ DNS เมื่อการเช่าถูกรีลีสหรือหมดอายุ คีย์เวิร์ดคือ:

releasednsA
ลบเร็กคอร์ด A

releasednsP
ลบเร็กคอร์ด PTR

removedns
ลบเร็กคอร์ดทั้งสองชนิด

คีย์เวิร์ดเหล่านี้ระบุสตริงปฏิบัติการที่เซิร์ฟเวอร์ **DHCP** จะรัน เมื่อแอดเดรสถูกรีลีสหรือหมดอายุ คำสั่ง **dhcpremove** ทำงานคล้ายกับ **dhcpaction** แต่ใช้เพียงสามพารามิเตอร์เท่านั้น:

1. IP แอดเดรสที่ระบุเป็น %s ในสตริงคำสั่ง
2. เร็กคอร์ดที่จะลบ (A, PTR, NONE, หรือ BOTH)
3. ควรจะอัพเดต NIM หรือไม่ (NIM หรือ NONIM)

ตัวอย่างเช่น:

```
releasednsA "/usr/sbin/dhcpremove '%s' A NONIM"
                   # This does the dhcpremove command only the A record
releasednsP "/usr/sbin/dhcpremove '%s' PTR NONIM"
                   # This does the command only on the PTR record
removedns "/usr/sbin/dhcpremove '%s' BOTH NIM"
                   # This does the command on both records and updates NIM
```

สคริปต์ **dhcpaction** และ **dhcpremove** ทำ การตรวจสอบพารามิเตอร์บางตัว จากนั้นตั้งค่าการเรียกไปยัง **nsupdate** ซึ่งมีการอัพเดตแล้วเพื่อทำงานกับเซิร์ฟเวอร์ของระบบปฏิบัติการนี้และกับเซิร์ฟเวอร์ OS/2 DDNS โปรดดูที่คำอธิบายคำสั่ง **nsupdate** สำหรับข้อมูลเพิ่มเติม

หากอัพเดตชื่อ **NOT** ต้องการการโต้ตอบ NIM เซิร์ฟเวอร์ DHCP สามารถมีการกำหนดคอนฟิกเพื่อใช้การโอนย้ายซ็อกเก็ตระหว่าง **DHCP** daemon และคำสั่ง **nsupdate** เพื่อปรับปรุงประสิทธิภาพและเปิดใช้งานอัพเดต DNS ที่จะลองซ้ำเมื่อล้มเหลว เพื่อกำหนดคอนฟิกอ็อพชันนี้ คีย์เวิร์ด **updateDNSA**, **updateDNSP**, **releaseDNSA**, หรือ **releaseDNSP** ต้องระบุ "nsupdate_daemon" เป็นคำในเครื่องหมายอัญประกาศคำแรก พารามิเตอร์และแฟล็กสำหรับอัพเดตนี้เหมือนกับ ที่ยอมรับโดยคำสั่ง **nsupdate** นอกจากนี้ สามารถใช้ชื่อตัวแปรต่อไปนี้สำหรับการทดแทน:

ไอเท็ม	คำอธิบาย
$hostname	แทนที่โดยชื่อโฮสต์ของไคลเอ็นต์ในอัพเดต DNS หรือชื่อโฮสต์ ที่เชื่อมโยงก่อนหน้านี้กับไคลเอ็นต์สำหรับการลบ DNS
$domain	แทนที่โดยโดเมน DNS สำหรับอัพเดต หรือโดเมนที่ใช้ก่อนหน้านี้ ของชื่อโฮสต์ไคลเอ็นต์สำหรับการลบ DNS
$ipadress	แทนที่โดย IP แอดเดรสที่จะเชื่อมโยงหรือตัดการเชื่อมโยงจากชื่อไคลเอ็นต์ **DHCP**
$leasetime	แทนที่โดยเวลาเช่า (ในหน่วยวินาที)
$clientid	แทนที่โดยสตริงแสดงแทนตัวระบุไคลเอ็นต์ **DHCP** หรือ ชุดของชนิดฮาร์ดแวร์และฮาร์ดแวร์แอดเดรสสำหรับไคลเอ็นต์ **BOOTP**

ตัวอย่างเช่น:

```
updateDNSA "nsupdate_daemon -p 9.3.149.2 -h $hostname -d $domain
 -s"d;a;*;a;a;$ipaddress;s;$leasetime;3110400""

updateDNSP "nsupdate_daemon -p 9.3.149.2 -r $ipaddress
 -s"d;ptr;*;a;ptr;$hostname.$domain.;s;$leasetime;3110400""

releaseDNSA "nsupdate_daemon -p 9.3.149.2 -h $hostname -d $domain -s"d;a;*;s;1;3110400""

releaseDNSP "nsupdate_daemon -p 9.3.149.2 -r $ipaddress -s"d;ptr;*;s;1;3110400""
```

โปรดดูที่คำอธิบายคำสั่ง **nsupdate** สำหรับข้อมูลเพิ่มเติม

นอกจากนี้ ยังมีการเพิ่มนโยบายที่กำหนดโดยผู้ดูแลระบบสำหรับการแลกเปลี่ยน ชื่อโฮสต์ระหว่างเซิร์ฟเวอร์และไคลเอ็นต์ โดยดีฟอลต์ ชื่อโฮสต์ที่ส่งกลับ ไปยังไคลเอ็นต์และใช้สำหรับอัพเดต DDNS คืออ็อพชัน 12 (ที่กำหนดไว้ในไฟล์ คอนฟิกูเรชันเซิร์ฟเวอร์) หรือชื่อโฮสต์ดีฟอลต์สามารถเป็นชื่อโฮสต์ที่ไคลเอ็นต์แนะนำ ผ่านทางอ็อพชัน 81 (อ็อพชัน DHCPDDNS) หรือผ่านทางอ็อพชัน 12 (อ็อพชัน HOSTNAME) อย่างไรก็ตาม ผู้ดูแลระบบสามารถยกเลิกชื่อโฮสต์ดีฟอลต์ โดยใช้คีย์เวิร์ดคอนฟิกูเรชัน **hostnamepolicy**, **proxyarec**, และ **appenddomain** อ็อพชันเหล่านี้ และพารามิเตอร์ของอ็อพชันมีการกำหนดไว้ใน “ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับฐานข้อมูล db_file” ในหน้า 257

ความเข้ากันได้กับเวอร์ชันเก่าของ DHCP

เซิร์ฟเวอร์ **DHCP** รับรู้คอนฟิกูเรชันและไฟล์ฐานข้อมูล เวอร์ชันก่อนหน้านี้ นั่นคือ `dhcps.ar` และ `dhcps.cr`

DHCP แจงส่วนไฟล์คอนฟิกูเรชันเก่าและสร้างไฟล์ฐานข้อมูลใหม่ ในที่ตั้งเก่า ฐานข้อมูลเก่าจะถูกแปลงเป็นไฟล์ใหม่โดยอัตโนมัติ ตัวไฟล์คอนฟิกูเรชันเองไม่ถูกแปลง

โมดูลฐานข้อมูลเซิร์ฟเวอร์ **DHCP**, `db_file`, สามารถ อ่านรูปแบบเก่าได้ เซิร์ฟเวอร์ **DHCP** สามารถรับรู้ได้เมื่อคอนเทนเนอร์ฐานข้อมูลไม่อยู่ในไฟล์คอนฟิกูเรชันและจัดการกับทั้งไฟล์เป็นพารามิเตอร์ การกำหนดคอนฟิกเซิร์ฟเวอร์ พารามิเตอร์การบันทึก และพารามิเตอร์ฐานข้อมูล `db_file`

หมายเหตุ:

1. ไวยากรณ์ไฟล์คอนฟิกูเรชันเก่าบางส่วนถูกคัดค้าน แต่ยังคงได้รับการสนับสนุน การคัดค้านอื่นๆ มีดังนี้:
2. คอนเทนเนอร์เครือข่ายถูกคัดค้านโดยสมบูรณ์ เพื่อระบุอย่างถูกต้อง ให้แปลงส่วนคำสั่งเครือข่ายที่มีช่วงเป็นคอนเทนเนอร์ subnet ที่ถูกต้อง ซึ่งมี subnet แอดเดรส, subnet netmask, และช่วง หากคอนเทนเนอร์เครือข่าย มีคอนเทนเนอร์ subnet ให้ลบคีย์เวิร์ดคอนเทนเนอร์เครือข่ายและวงเล็บปีกกา จากนั้นวาง subnet mask ในตำแหน่งที่เหมาะสมบนบรรทัดเมื่อต้องการเริ่มต้น การใช้คอนเทนเนอร์ฐานข้อมูล ให้จัดกลุ่มทุกสิ่งที่เกี่ยวข้องกับเครือข่ายและ สิทธิเข้าถึงไคลเอ็นต์ไว้ในคอนเทนเนอร์ฐานข้อมูลเดียวชนิด `db_file`

3. คีย์เวิร์ด **updatedns** และ **removedns** ถูกคัดค้านและแทนที่เพื่อสนับสนุนการระบุการดำเนินการสำหรับเร็กคอร์ด A และ PTR แยกต่างหากกัน
4. คีย์เวิร์ด **clientrecorddb** และ **addressrecorddb** ถูกเปลี่ยนเป็น **clientrecorddb** และ **backupfile** ตามลำดับ
5. คีย์เวิร์ด **option sa** และ **option ga** ถูกแทนที่โดยคีย์เวิร์ด **bootstrapserver** และ **giaddrfield** ตามลำดับ โปรดดู "ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับการดำเนินงานเซิร์ฟเวอร์ทั่วไป" ในหน้า 252 และ "ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับฐานข้อมูล db_file" ในหน้า 257 สำหรับข้อมูลเพิ่มเติม

อ็อพชันที่รู้จักของไฟล์เซิร์ฟเวอร์ DHCP

อ็อพชันที่รู้จักของไฟล์เซิร์ฟเวอร์ **DHCP** มีการระบุ ที่นี่

หมายเหตุ: อ็อพชันที่แสดงในตารางนี้ว่าไม่อนุญาตให้ ระบุ (ไม่ ในคอลัมน์ สามารถระบุ?) สามารถระบุได้ในไฟล์คอนฟิกูเรชัน แต่จะถูกเขียนทับโดยค่าที่ถูกต้อง สำหรับนิยามที่ดีขึ้นของ แต่ละอ็อพชัน ให้ดูที่ RFC 2132

หมายเลขอ็อพชัน	ชนิดข้อมูลดีฟอลต์	สามารถระบุ?	คำอธิบาย/การใช้งาน
0	ไม่มี	ไม่	เซิร์ฟเวอร์ pads ฟิลด์อ็อพชัน ถ้าจำเป็น
1	จุด quad	ไม่	Net mask ของ subnet ซึ่งดึงแอดเดรส
2	เลขจำนวนเต็ม 32 บิต	ใช่	ระบุออฟเซ็ตของไคลเอ็นต์ subnet ในหน่วยวินาทีจาก Coordinated Universal Time (UTC)
3	หนึ่งหรือหลายจุด quads	ใช่	รายการ IP แอดเดรสของดีฟอลต์เกตเวย์
4	หนึ่งหรือหลายจุด quads	ใช่	รายการ IP แอดเดรสของเซิร์ฟเวอร์เวลา
5	หนึ่งหรือหลายจุด quads	ใช่	รายการ IP แอดเดรสของเนมเซิร์ฟเวอร์
6	หนึ่งหรือหลายจุด quads	ใช่	รายการของ DNS IP แอดเดรส
7	หนึ่งหรือหลายจุด quads	ใช่	รายการ IP แอดเดรสของเซิร์ฟเวอร์บันทึก
8	หนึ่งหรือหลายจุด quads	ใช่	รายการ IP แอดเดรสของเซิร์ฟเวอร์คุกกี้
9	หนึ่งหรือหลายจุด quads	ใช่	รายการ IP แอดเดรสของเซิร์ฟเวอร์ LPR
10	หนึ่งหรือหลายจุด quads	ใช่	รายการ IP แอดเดรสของเซิร์ฟเวอร์ Impress
11	หนึ่งหรือหลายจุด quads	ใช่	รายการ IP แอดเดรสของเซิร์ฟเวอร์ที่ตั้งรีซอร์ส
12	สตริง ASCII	ใช่	ชื่อโฮสต์สำหรับไคลเอ็นต์ที่จะใช้
13	เลขจำนวนเต็ม 16 บิตที่ไม่มีเครื่องหมาย	ใช่	ขนาดของ bootfile
14	สตริง ASCII	ใช่	พาธสำหรับไฟล์ Merit Dump
15	สตริง ASCII	ใช่	ชื่อโดเมน DNS ดีฟอลต์
16	IP แอดเดรส	ใช่	แอดเดรสของเซิร์ฟเวอร์ Swap
17	สตริง ASCII	ใช่	พาธ root ดีฟอลต์
18	สตริง ASCII	ใช่	พาธไปยังส่วนขยายของไคลเอ็นต์
19	ใช่, ไม่, จริง, เท็จ, 1, 0	ใช่	ระบุว่าควรจะเปิด IP Forwarding หรือไม่
20	ใช่, ไม่, จริง, เท็จ, 1, 0	ใช่	ระบุว่าควรจะใช้การเราต์ต้นทางแบบไม่ใช่โลคัลหรือไม่
21	หนึ่งหรือหลายคู่ของจุด quads ในรูปแบบ *DottedQuad*:*DottedQuad*	ใช่	นโยบายตัวกรองสำหรับ IP แอดเดรส
22	เลขจำนวนเต็ม 16 บิตที่ไม่มีเครื่องหมาย	ใช่	ขนาดสูงสุดที่ใช้ได้สำหรับ datagram แฟรกเมนต์
23	เลขจำนวนเต็ม 8 บิตที่ไม่มีเครื่องหมาย	ใช่	IP time-to-live (TTL)
24	เลขจำนวนเต็ม 32 บิตที่ไม่มีเครื่องหมาย	ใช่	จำนวนวินาทีที่จะใช้ในช่วงเวลาไทม์เอาต์ Path MTU
25	รายการของเลขจำนวนเต็ม 16 บิตที่ไม่มีเครื่องหมายตั้งแต่หนึ่งรายการขึ้นไป	ใช่	ตาราง path MTU Plateau ระบุชุดค่าที่แสดงถึง ขนาด MTU ที่จะใช้เมื่อใช้การค้นพบ Path MTU
26	เลขจำนวนเต็ม 16 บิตที่ไม่มีเครื่องหมาย	ใช่	ระบุขนาด MTU สำหรับอินเตอร์เฟสการรับ
27	ใช่, ไม่, จริง, เท็จ, 1, 0	ใช่	ระบุว่า subnets ทั้งหมดเป็นแบบโลคัลหรือไม่
28	IP แอดเดรส (จุด quad)	ใช่	ระบุแอดเดรสแพร่สัญญาณสำหรับอินเตอร์เฟส
29	ใช่, ไม่, จริง, เท็จ, 1, 0	ใช่	ระบุว่าควรจะใช้การค้นพบ ICMP netmask หรือไม่
30	ใช่, ไม่, จริง, เท็จ, 1, 0	ใช่	ระบุว่าไคลเอ็นต์ควรเป็นผู้ระบุ ICMP netmask หรือไม่
31	ใช่, ไม่, จริง, เท็จ, 1, 0	ใช่	ระบุว่าควรจะใช้ข้อความ ICMP Router Discovery หรือไม่

หมายเลขอ็อพ

ชัน	ชนิดข้อมูลดีฟอลต์	สามารถระบุ?	คำอธิบาย/การใช้งาน
32	IP แอดเดรส (จุด quad)	ใช่	ระบุแอดเดรสที่จะใช้สำหรับการร้องขอเราเตอร์
33	หนึ่งหรือหลายคู่ IP แอดเดรส ในรูปแบบ *DottedQuad*:*DottedQuad*	ใช่	คู่แอดเดรสแต่ละคู่แสดงถึงเราต์แบบสแตติก
34	ใช่/ไม่, จริง/เท็จ, 1/0	ใช่	ระบุว่าควรจะใช้การห่อหุ้มส่วนปลายหรือไม่
35	เลขจำนวนเต็ม 32 บิตที่ไม่มีเครื่องหมาย	ใช่	ค่าไทม์เอาต์ ARP แคช
36	ใช่/ไม่, จริง/เท็จ, 1/0	ใช่	ระบุว่าควรจะใช้การห่อหุ้มอีเทอร์เน็ตหรือไม่
37	เลขจำนวนเต็ม 8 บิตที่ไม่มีเครื่องหมาย	ใช่	TCP time-to-live (TTL)
38	เลขจำนวนเต็ม 32 บิตที่ไม่มีเครื่องหมาย	ใช่	ช่วงเวลา **TCP** keep alive
39	ใช่/ไม่, จริง/เท็จ, 1/0	ใช่	ระบุว่าควรจะใช้ TCP keep alive หรือไม่
40	สตริง ASCII	ใช่	NIS ดีฟอลต์โดเมน
41	หนึ่งหรือหลายจุด quads	ใช่	ระบุ IP แอดเดรสของเซิร์ฟเวอร์ NIS
42	หนึ่งหรือหลายจุด quads	ใช่	ระบุ IP แอดเดรสของเซิร์ฟเวอร์ NTP
43	สตริงฐานหกของตัวเลข ในรูปแบบของ hex *"digits"*, hex *"digits"*, หรือ 0x*digits*	ใช่ แต่มีการระบุจริงในคอนเทนเนอร์ผู้ขายเท่านั้น	อ็อพชันคอนเทนเนอร์สำหรับคอนเทนเนอร์ผู้ขายเท่านั้น
44	หนึ่งหรือหลายจุด quads	ใช่	ระบุ IP แอดเดรสของเนมเซิร์ฟเวอร์ NetBIOS
45	หนึ่งหรือหลายจุด quads	ใช่	ระบุ IP แอดเดรสของเซิร์ฟเวอร์การแจกจ่าย NetBIOS datagram
46	เลขจำนวนเต็ม 8 บิตที่ไม่มีเครื่องหมาย	ใช่	ระบุชนิดโหนด NetBIOS
47	สตริงฐานหกของตัวเลข ในรูปแบบของ hex *"digits"*, hex *"digits"*, หรือ 0x*digits*	Yes	NetBIOS Scope
48	หนึ่งหรือหลายจุด quads	ใช่	ระบุ IP แอดเดรสของเซิร์ฟเวอร์ X Windows font
49	หนึ่งหรือหลายจุด quads	ใช่	ระบุ X Windows Display Manager
50	ไม่มี	ไม่	ร้องขอ IP แอดเดรสที่ใช้โดยไคลเอ็นต์เพื่อบ่งชี้แอดเดรสที่ต้องการ
51	เลขจำนวนเต็ม 32 บิตที่ไม่มีเครื่องหมาย	ใช่	เวลาเช่าสำหรับแอดเดรสที่ส่งคืน โดยค่าดีฟอลต์ เซิร์ฟเวอร์ **DHCP** ใช้คีย์เวิร์ด **leasetimedefault** แต่การระบุ อ็อพชัน 51 โดยตรงยกเลิกคีย์เวิร์ดดังกล่าว
52	ไม่มี	ไม่	อ็อพชันโอเวอร์โหลด ไคลเอ็นต์ใช้ค่านี้เพื่อบ่งชี้ว่าฟิลด์ **sname** และ **file** ของแพ็กเก็ต **BOOTP** อาจมีอ็อพชัน
53	ไม่มี	ไม่	เซิร์ฟเวอร์ **DHCP** หรือไคลเอ็นต์ใช้อ็อพชันนี้เพื่อบ่งชี้ชนิด ของข้อความ **DHCP**
54	ไม่มี	ไม่	เซิร์ฟเวอร์ **DHCP** หรือไคลเอ็นต์ใช้อ็อพชันนี้เพื่อบ่งชี้แอดเดรสของ เซิร์ฟเวอร์หรือเซิร์ฟเวอร์ซึ่งจะกำหนดทิศทางข้อความไป
55	ไม่มี	ไม่	ไคลเอ็นต์ **DHCP** ใช้ค่านี้เพื่อบ่งชี้อ็อพชันที่ต้องการ
56	สตริง ASCII	ใช่	สตริงที่เซิร์ฟเวอร์ **DHCP** ส่งไปยังไคลเอ็นต์โดยทั่วไป เซิร์ฟเวอร์ **DHCP** และไคลเอ็นต์สามารถใช้ค่านี้เพื่อบ่งชี้ปัญหา
57	ไม่	ไม่	ไคลเอ็นต์ **DHCP** ใช้อ็อพชันนี้เพื่อแจ้งให้เซิร์ฟเวอร์ **DHCP** ทราบขนาดแพ็กเก็ต **DHCP** สูงสุดที่ไคลเอ็นต์สามารถรับได้
58	เลขจำนวนเต็ม 32 บิตที่ไม่มีเครื่องหมาย	ใช่	ระบุจำนวนวินาทีจนกว่าไคลเอ็นต์ควรจะส่งแพ็กเก็ต ต่ออายุ
59	เลขจำนวนเต็ม 32 บิตที่ไม่มีเครื่องหมาย	ใช่	ระบุจำนวนวินาทีจนกว่าไคลเอ็นต์ควรจะส่งแพ็กเก็ต ผูกใหม่
60	ไม่มี	ไม่	ไคลเอ็นต์ **DHCP** ใช้อ็อพชันนี้เพื่อบ่งชี้ชนิดของผู้ขาย เซิร์ฟเวอร์ **DHCP** ใช้ ฟิลด์นี้เพื่อจับคู่คอนเทนเนอร์ผู้ขาย
61	ไม่มี	ไม่	ไคลเอ็นต์ **DHCP** ใช้ค่านี้เพื่อระบุถึงตัวเองโดยเฉพาะ เซิร์ฟเวอร์ **DHCP** ใช้ ฟิลด์นี้เพื่อจับคู่ไคลเอ็นต์คอนเทนเนอร์
66	สตริง ASCII	ใช่	ระบุชื่อเซิร์ฟเวอร์ **TFTP** นี่เป็นชื่อโฮสต์และใช้ แทนฟิลด์ **siaddr** ถ้าไคลเอ็นต์เข้าใจ อ็อพชันนี้
67	สตริง ASCII	ใช่	ระบุชื่อ bootfile สามารถใช้ค่านี้แทนคีย์เวิร์ด **bootfile** ซึ่งวางไฟล์ไว้ในฟิลด์ **filename** ของ แพ็กเก็ต
68	หนึ่งหรือหลายจุด quads หรือ NONE	ใช่	ระบุแอดเดรสของโฮมเอเจนต์
69	หนึ่งหรือหลายจุด quads	ใช่	ระบุเซิร์ฟเวอร์ SMTP ดีฟอลต์ที่จะใช้
70	หนึ่งหรือหลายจุด quads	ใช่	ระบุเซิร์ฟเวอร์ POP3 ดีฟอลต์ที่จะใช้

หมายเลขอ็อพ

ชัน	ชนิดข้อมูลดีฟอลต์	สามารถระบุ?	คำอธิบาย/การใช้งาน
71	หนึ่งหรือหลายจุด quads	ใช่	ระบุเซิร์ฟเวอร์ NNTP ดีฟอลต์ที่จะใช้
72	หนึ่งหรือหลายจุด quads	ใช่	ระบุเซิร์ฟเวอร์ WWW ดีฟอลต์ที่จะใช้
73	หนึ่งหรือหลายจุด quads	ใช่	ระบุเซิร์ฟเวอร์ Finger ดีฟอลต์ที่จะใช้
74	หนึ่งหรือหลายจุด quads	ใช่	ระบุเซิร์ฟเวอร์ IRC ดีฟอลต์ที่จะใช้
75	หนึ่งหรือหลายจุด quads	ใช่	ระบุเซิร์ฟเวอร์ Street Talk ดีฟอลต์ที่จะใช้
76	หนึ่งหรือหลายจุด quads	ใช่	ระบุเซิร์ฟเวอร์ความช่วยเหลือไดเร็กทอรี Street Talk ดีฟอลต์ที่จะใช้
77	สตริง ASCII	ใช่	ตัวระบุคลาสไซต์ของผู้ใช้ เซิร์ฟเวอร์ **DHCP** ใช้ ฟิลด์นี้เพื่อจับคู่คลาสคอนเทนเนอร์
78	ไบต์บังคับ, หนึ่งหรือหลายจุด quads	ใช่	SLP directory Agent Option ระบุรายการ IP แอดเดรสสำหรับ Directory Agents
79	ไบต์บังคับ, และสตริง ASCII	ใช่	สตริง ASCII เป็นรายการขอบเขต ซึ่งเป็นรายการที่คั่นด้วยเครื่องหมายจุลภาคและบ่งชี้ขอบเขตที่กำหนดคอนฟิกให้ SLP Agent ใช้งาน
81	สตริง ASCII บวกไอเท็มอื่น	ไม่	ไคลเอ็นต์ DHCP ใช้อ็อพชันนี้เพื่อกำหนดนโยบายที่เซิร์ฟเวอร์ **DHCP** ควรจะใช้เกี่ยวกับ DDNS
85	หนึ่งหรือหลายจุด quads	ใช่	NDS เซิร์ฟเวอร์อ็อพชันระบุเซิร์ฟเวอร์ NDS ตั้งแต่หนึ่งเครื่อง ขึ้นไปสำหรับไคลเอ็นต์ที่จะติดต่อเพื่อเข้าถึงฐานข้อมูล DNS เซิร์ฟเวอร์ควรมีการแสดง ตามลำดับการกำหนดค่าตามความชอบ
86	สตริง ASCII	ใช่	อ็อพชันชื่อแผนผัง NDS ระบุชื่อของแผนผัง NDS ที่ไคลเอ็นต์จะติดต่อ
87	สตริง ASCII	ใช่	NDS คอนเท็กซ์อ็อพชันระบุ NDS คอนเท็กซ์แรกเริ่ม ที่ไคลเอ็นต์ควรจะใช้
93	ไม่มี	ไม่	ไคลเอ็นต์ DHCP ใช้อ็อพชันนี้เพื่อกำหนดสถาปัตยกรรมระบบไคลเอ็นต์
94	ไม่มี	ไม่	ไคลเอ็นต์ DHCP ใช้อ็อพชันนี้เพื่อกำหนดตัวระบุอินเตอร์เฟสเครือข่ายไคลเอ็นต์

117	เลขจำนวนเต็ม 16 บิตที่ไม่มีเครื่องหมายตั้งแต่หนึ่งรายการขึ้นไป	ใช่	Name Service Search Option ให้ลำดับที่ต้องการ ของโค้ดอ็อพชันเลขจำนวนเต็มสำหรับเซอร์วิสชื่อ ตัวอย่างเช่น:
			`Name Services value`
			`Domain Name Server Option 6`
			`NIS Option 41`

118	หนึ่งจุด quads	ไม่	Subnet Selection Option เป็นอ็อพชันที่ส่งโดยไคลเอ็นต์ เพื่อขอให้เซิร์ฟเวอร์ dhcp จัดสรร IP แอดเดรสจาก subnet ที่ระบุ
255	ไม่มี	ไม่	เซิร์ฟเวอร์ DHCP และไคลเอ็นต์ใช้อ็อพชันนี้เพื่อบ่งชี้การสิ้นสุดของรายการ อ็อพชัน

อ็อพชันย่อยของคอนเทนเนอร์ผู้ขาย preboot execution environment

เมื่อสนับสนุนไคลเอ็นต์ preboot execution environment (PXE) เซิร์ฟเวอร์ **DHCP** จะส่งผ่านอ็อพชันต่อไปนี้ไปยังเซิร์ฟเวอร์ BINLD ซึ่งใช้โดย BINLD เพื่อ กำหนดคอนฟิกตัวเอง

Opt Num	ชนิดข้อมูลดีฟอลต์	สามารถระบุ?	คำอธิบาย
7	หนึ่งจุด quads	ใช่	Multicast IP แอดเดรส เซิร์ฟเวอร์บูต ค้นพบ multicast IP แอดเดรส

ตัวอย่างต่อไปนี้แสดงวิธีการใช้อ็อพชันนี้:

```
pxeservertype    proxy_on_dhcp_server

Vendor pxeserver
{
     option   7    9.3.4.68
}
```

ในตัวอย่างข้างบน เซิร์ฟเวอร์ **DHCP** แจ้งไคลเอ็นต์ว่า เซิร์ฟเวอร์พร็อกซีกำลังรันบนเครื่องเดียวกันแต่รับฟังคำร้องขอไคลเอ็นต์อยู่ บนพอร์ต 4011 ต้องใช้คอนเทนเนอร์ผู้ขายที่นี่เนื่องจากเซิร์ฟเวอร์ BINLD แพร่สัญญาณข้อความ INFORM/REQUEST บนพอร์ต 67 ที่มีการตั้งค่าอ็อพชัน 60 เป็น "PXEServer" ในการตอบกลับ เซิร์ฟเวอร์ **DHCP** จะส่ง Multicast IP แอดเดรสซึ่ง BINLD ต้องรับฟังคำร้องขอของ PXEClient

ในตัวอย่างต่อไปนี้ เซิร์ฟเวอร์ **dhcpsd** ให้ชื่อ bootfile แก่ PXEClient หรือกำหนดทิศทาง PXEClient ไปยังเซิร์ฟเวอร์ BINLD โดยการส่งอ็อพชันย่อย คีย์เวิร์ด **pxeboofile** ใช้เพื่อสร้างรายการของไฟล์บูตสำหรับสถาปัตยกรรมไคลเอ็นต์ที่กำหนดและเวอร์ชันหลักและรองของระบบไคลเอ็นต์

```
pxeservertype     dhcp_pxe_binld

subnet default
{
     vendor pxe
     {
         option 6 2    # Disable Multicast
         option 8 5 4 10.10.10.1 12.1.1.15 12.5.5.5 12.6.6.6\
                  2 2 10.1.1.10 9.3.4.5 1 1 10.5.5.9\
                  1 1 9.3.149.15\
                  4 0
         option 9 5 "WorkSpace On Demand" 2 "Intel"\
                  1 "Microsoft Windows NT" 4 "NEC ESMPRO"
         option 10 2 "Press F8 to View Menu"
     }
     vendor pxeserver
     {
         option 7 239.0.0.239
     }

}

subnet  9.3.149.0  255.255.255.0
{
     option 3   9.3.149.1
     option 6   9.3.149.15

     vendor pxe
     {
       option    6    4    # bootfile is present in the offer packet
       pxebootfile   1   2   1   os2.one
       pxebootfile   2   2   1   aix.one
     }
}
```

เซิร์ฟเวอร์ใช้แต่ละบรรทัดอ็อพชันในคอนเทนเนอร์ PXE เพื่อ บอกสิ่งที่ไคลเอ็นต์ต้องทำ “อ็อพชันย่อยของคอนเทนเนอร์ผู้ผลิต PXE” ในหน้า 332 อธิบาย อ็อพชันย่อย PXE ที่ได้รับการสนับสนุนและรู้จักในปัจจุบัน

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับการดำเนินงานเซิร์ฟเวอร์ทั่วไป

ไวยากรณ์ไฟล์ **DHCP** สำหรับการดำเนินงานเซิร์ฟเวอร์ทั่วไปและ ค่าที่ถูกต้องสำหรับแต่ละฟิลด์มีการกำหนดอยู่ที่นี่

หมายเหตุ: หน่วยเวลา (*time_units*) ที่แสดงในตารางต่อไปนี้ เป็นทางเลือก และแสดงถึงตัวแก้ไขเวลาจริง หน่วยเวลาดีฟอลต์คือนาที ค่าที่ถูกต้องคือวินาที (1), นาที (60), ชั่วโมง (3600), วัน (86400), สัปดาห์ (604800), เดือน

(2392000), และปี (31536000) ตัวเลข ที่แสดงในวงเล็บเป็นตัวคูณที่ใช้กับค่า *n* ที่ระบุ เพื่อระบุค่าในหน่วยวินาที

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ฐานข้อมูล	ฐานข้อมูล *db_type*	ใช่	ไม่มี	คอนเทนเนอร์หลักที่จัดเก็บนิยามสำหรับแอดเดรสพูล อ็อพชัน และคำสั่งการเข้าถึงไคลเอ็นต์ *db_type* คือ ชื่อของโมดูลที่ถูกโหลดเพื่อประมวลผลส่วนนี้ของไฟล์ ค่าเดียว ที่มีอยู่ในปัจจุบันคือ **db_file**
logging_info	logging_info	ใช่	ไม่มี	คอนเทนเนอร์การบันทึกหลักที่กำหนดพารามิเตอร์การบันทึก
logitem	logitem NONE	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem SYSERR	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem OBJERR	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem PROTOCOL	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem PROTERR	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem WARN	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem WARNING	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem CONFIG	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem EVENT	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem PARSEERR	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem ACTION	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
logitem	logitem ACNTING	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem STAT	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem TRACE	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem RTRACE	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem START	ไม่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
numLogFiles	numLogFiles *n*	ไม่ใช่	0	ระบุจำนวนของไฟล์บันทึกที่จะสร้าง บันทึกจะหมุนเวียนเมื่อกรอกข้อมูล บันทึกแรก *n* คือจำนวนของไฟล์ที่จะสร้าง
logFileSize	logFileSize *n*	ไม่ใช่	0	ระบุขนาดของแต่ละไฟล์บันทึกในหน่วย 1024-ไบต์
logFileName	logFileName *พาธ*	ไม่	ไม่มี	ระบุพาธไปยังไฟล์บันทึกแรก ไฟล์บันทึกดั้งเดิมมีชื่อว่า *filename* หรือ *filename.extension* เมื่อหมุนเวียนไฟล์ ไฟล์จะถูกเปลี่ยนชื่อโดยเริ่มต้นด้วยฐาน *filename* โดยการผนวกตัวเลขหรือการแทนที่นามสกุลด้วยตัวเลข ตัวอย่าง เช่น ถ้าชื่อไฟล์ดั้งเดิมเป็น `file` ชื่อของไฟล์ที่หมุนเวียน จะกลายเป็น `file01` ถ้าชื่อไฟล์ดั้งเดิมเป็น `file.log` จะกลายเป็น `file.01`
CharFlag	charflag yes	ไม่	จริง	ใช้ไม่ได้กับเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการนี้ แต่เซิร์ฟเวอร์ OS/2 **DHCP** ใช้เพื่อจัดทำหน้าต่างดีบัก
CharFlag	charflag true	ไม่	จริง	ใช้ไม่ได้กับเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการนี้ แต่เซิร์ฟเวอร์ OS/2 **DHCP** ใช้เพื่อจัดทำหน้าต่างดีบัก
CharFlag	charflag false	ไม่	จริง	ใช้ไม่ได้กับเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการนี้ แต่เซิร์ฟเวอร์ OS/2 **DHCP** ใช้เพื่อจัดทำหน้าต่างดีบัก

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
CharFlag	charflag no	ไม่	จริง	ใช้ไม่ได้กับเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการนี้ แต่เซิร์ฟเวอร์ OS/2 **DHCP** ใช้เพื่อจัดทำหน้าต่างดีบัก
StatisticSnapShot	StatisticSnapShot *n*	ไม่	-1, ไม่เคย	ระบุความบ่อยในการเขียนสถิติลงในไฟล์บันทึกในหน่วยวินาที
UsedIpAddressExpireInterval	UsedIpAddressExpireInterval *n time_units*	ไม่	-1, ไม่เคย	ระบุความบ่อยในการ recouped และทดสอบความถูกต้องอีกครั้งของแอดเดรส ที่วางไว้ในสถานะ BAD
leaseExpireInterval	leaseExpireInterval *n time_units*	ไม่	900 วินาที	ระบุความบ่อยในการตรวจสอบแอดเดรสในสถานะ BOUND เพื่อดูว่า หมดอายุหรือไม่ หากแอดเดรสหมดอายุ สถานะจะถูกย้ายไปยัง EXPIRED
reservedTime	reservedTime *n time_units*	ไม่	-1, ไม่เคย	ระบุระยะเวลาที่แอดเดรสควรอยู่ในสถานะ RESERVED ก่อนถูก recouped เข้าในสถานะ FREE
reservedTimeInterval	reservedTimeInterval *n time_units*	ไม่	900 วินาที	ระบุความบ่อยในการตรวจสอบแอดเดรสในสถานะ RESERVE เพื่อดูว่า ควรถูก recouped เข้าในสถานะ FREE หรือไม่
saveInterval	saveInterval *n time_units*	ไม่	3600 วินาที	ระบุความบ่อยที่เซิร์ฟเวอร์ **DHCP** ควรบังคับใช้การบันทึกของ ฐานข้อมูลเปิด สำหรับเซิร์ฟเวอร์ที่โหลดหนัก ค่านี้ควรเป็น 60 หรือ 120 วินาที
clientpruneintv	clientpruneintv *n time_units*	ไม่	3600 วินาที	ระบุความบ่อยที่เซิร์ฟเวอร์ **DHCP** มีไคลเอ็นต์ลบฐานข้อมูลและไม่ได้เชื่อมโยงกับแอดเดรสใดๆ (ในสถานะ UNKNOWN) ซึ่งช่วยลด การใช้หน่วยความจำของเซิร์ฟเวอร์ **DHCP**
numprocessors	numprocessors *n*	ไม่	10	ระบุจำนวนของตัวประมวลผลแพ็กเก็ตที่จะสร้าง ค่าต่ำสุดคือหนึ่ง

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
userObject	userObject *obj_name*	ใช่	ไม่มี	บ่งชี้ว่าเซิร์ฟเวอร์ควรโหลดอ็อบเจ็กต์แบบแบ่งใช้ที่กำหนดโดยผู้ใช้ และเรียกรูทีนภายในอ็อบเจ็กต์นี้ผ่านทางการโต้ตอบกับไคลเอ็นต์ **DHCP** ในแต่ละครั้งอ็อบเจ็กต์ที่จะโหลดตั้งอยู่ในไดเร็กทอรี `/usr/sbin` โดยใช้ชื่อ `obj_name.dhcpo` โปรดดูที่ DHCP Server User-Defined Extension API สำหรับข้อมูลเพิ่มเติม
pxeservertype	pxeservertype *server_type*	ไม่	dhcp_only	บ่งชี้ชนิดของเซิร์ฟเวอร์ **dhcpd** *server_type* สามารถ เป็นอย่างใดอย่างหนึ่งต่อไปนี้: **dhcp_pxe_binld** **DHCP** ทำฟังก์ชัน **dhcpsd, pxed,** และ **bindl** **proxy_on_dhcp_server** **DHCP** อ้างอิงไคลเอ็นต์ PXE ไปยังพร็อกซีเซิร์ฟเวอร์พอร์ตบนเครื่อง เดียวกัน ค่าดีฟอลต์คือ `dhcp_only` ซึ่งหมายความว่า **dhcpsd** ไม่สนับสนุนไคลเอ็นต์ PXE ในโหมดดีฟอลต์
supportsubnetselection	supportsubnetselection **global** supportsubnetselection **subnetlevel** supportsubnetselection **no**	ไม่	ไม่มี	บ่งชี้ว่าเซิร์ฟเวอร์ dhcp จะสนับสนุน อ็อพชัน 118 (อ็อพชันการเลือก subnet) ในแพ็กเก็ต DISCOVER หรือ REQUEST ของไคลเอ็นต์หรือไม่ **global**: subnets ทั้งหมดในไฟล์คอนฟิกูเรชันจะสนับสนุนอ็อพชัน 118 **subnetlevel**: subnets มีการกำหนดคอนฟิกเพื่อสนับสนุนอ็อพชันนี้โดยคีย์เวิร์ด supportoption118 จะสนับสนุนอ็อพชันนี้ **no**: ไม่สนับสนุน อ็อพชัน 118

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ DHCP สำหรับฐานข้อมูล db_file

ไวยากรณ์ไฟล์สำหรับฐานข้อมูล `db_file` มี คุณสมบัติต่อไปนี้

หมายเหตุ:

1. หน่วยเวลา (*time_units*) ที่แสดงในตารางต่อไปนี้ เป็นทางเลือก และแสดงถึงตัวแก้ไขเวลาจริง หน่วยเวลา ดีฟอลต์คือ นาที ค่าที่ถูกต้องคือวินาที (1), นาที (60), ชั่วโมง (3600), วัน (86400), สัปดาห์ (604800), เดือน (2392000), และปี (31536000) ตัวเลข ที่แสดงในวงเล็บเป็นตัวคูณที่ใช้กับค่า *n* ที่ระบุ เพื่อระบุค่าในหน่วยวินาที
2. ไอเท็มที่ระบุในคอนเทนเนอร์หนึ่งสามารถถูกยกเลิกภายในคอนเทนเนอร์ย่อย ตัวอย่างเช่น คุณสามารถกำหนดไคลเอ็นต์ **BOOTP** แบบสากล แต่ภายใน บาง subnet อนุญาตไคลเอ็นต์ **BOOTP** โดยระบุคีย์เวิร์ด supportBootp ในทั้งสองคอนเทนเนอร์
3. ไคลเอ็นต์ คลาส และคอนเทนเนอร์ผู้ขายอนุญาตการสนับสนุนนิพจน์ ปกติ สำหรับคลาสและผู้ขาย สตริงที่อยู่ใน อัญประกาศที่มีอักขระตัวแรกหลังจาก อัญประกาศเป็นเครื่องหมายอัศเจรีย์ (!) บ่งชี้ว่าส่วนที่เหลือของสตริง ควรถูกจัดการเป็นนิพจน์ปกติ ไคลเอ็นต์คอนเทนเนอร์อนุญาตการใช้ นิพจน์ปกติบนทั้งฟิลด์ hwtype และ hwaddr สตริงเดี่ยว ใช้เพื่อแสดงแทนทั้งสองฟิลด์ด้วยรูปแบบต่อไปนี้:

   ```
   decimal_number-data
   ```

   หาก decimal_number เป็นศูนย์ ข้อมูลจะเป็นสตริง ASCII หากเป็นตัวเลขอื่น ข้อมูลจะเป็นตัวเลขฐานหก

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
subnet	subnet ดีฟอลต์	ใช่	ไม่มี	ระบุ subnet ที่ไม่มีช่วงเชื่อมโยง เซิร์ฟเวอร์ใช้ subnet นี้ เฉพาะถ้าตอบกลับไปยังแพ็กเก็ต INFORM/REQUEST จาก ไคลเอ็นต์และแอดเดรสของไคลเอ็นต์ไม่มี subnet คอนเทนเนอร์ที่ตรงกันอื่น
subnet	subnet *subnet id netmask*	ใช่	ไม่มี	ระบุ subnet และพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วงหรือคำสั่ง exclude ช่วงทางเลือก เป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบลและระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
subnet	subnet *subnet id netmask ช่วง*	ใช่	ไม่มี	ระบุ subnet และพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมด อยู่ในพูล ยกเว้นว่ามีการระบุช่วง บนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์ โดยช่วงหรือคำสั่ง exclude ช่วง ทางเลือก เป็นคู่ของ IP แอดเดรส ในรูปแบบจุด quad ที่แบ่งโดย เครื่องหมายขีด สามารถ ระบุเลเบลทางเลือกและระดับ ความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่ง โดยเครื่องหมายจุดคู่ คอนเทน เนอร์เหล่านี้ใช้ได้ที่ระดับ คอน เทนเนอร์สากลหรือฐานข้อมูลเท่า นั้น
subnet	subnet *subnet id netmask เลเบล :ระดับความสำคัญ*	ใช่	ไม่มี	ระบุ subnet และพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมด อยู่ในพูล ยกเว้นว่ามีการระบุช่วง บนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์ โดยช่วงหรือคำสั่ง exclude ช่วง ทางเลือก เป็นคู่ของ IP แอดเดรส ในรูปแบบจุด quad ที่แบ่งโดย เครื่องหมายขีด สามารถ ระบุเลเบลทางเลือกและระดับ ความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่ง โดยเครื่องหมายจุดคู่ คอนเทน เนอร์เหล่านี้ใช้ได้ที่ระดับ คอน เทนเนอร์สากลหรือฐานข้อมูลเท่า นั้น

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
subnet	subnet *subnet id netmask ช่วง เลเบล :ระดับความสำคัญ*	ใช่	ไม่มี	ระบุ subnet และพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วงหรือคำสั่ง exclude ช่วงทางเลือก เป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบลและระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
subnet	subnet *subnet id ช่วง*	ใช่	ไม่มี	ระบุ subnet ที่อยู่ภายในคอนเทนเนอร์เครือข่าย กำหนด ช่วงของแอดเดรสที่เป็น subnet ทั้งหมดยกเว้นว่ามีการระบุส่วนช่วง ทางเลือก Netmask ที่เชื่อมโยงกับ subnet นำมาจากคอนเทนเนอร์เครือข่ายล้อมรอบ **หมายเหตุ:** เมธอดนี้ใช้เพื่อสนับสนุน รูปแบบ subnet อื่น
อ็อพชัน	อ็อพชัน *ตัวเลข ข้อมูล* ...	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณี ของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ ส่วนคำสั่งอ็อพชัน * deny หมายความว่าอ็อพชันทั้งหมดที่ไม่ได้ระบุในคอนเทนเนอร์ปัจจุบัน จะไม่ถูกส่งคืนไปยังไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8–บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* or hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วยอ็อพชันอื่นในอ็อพชัน 43

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
อ็อพชัน	อ็อพชัน *ตัวเลข*deny	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณี ของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ ส่วนคำสั่งอ็อพชัน * deny หมายความว่าอ็อพชันทั้งหมดที่ไม่ได้ระบุในคอนเทนเนอร์ปัจจุบัน จะไม่ถูกส่งคืนไปยังไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8-บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* or hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วยอ็อพชันอื่นในอ็อพชัน 43
อ็อพชัน	อ็อพชัน * deny	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณี ของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ ส่วนคำสั่งอ็อพชัน * deny หมายความว่าอ็อพชันทั้งหมดที่ไม่ได้ระบุในคอนเทนเนอร์ปัจจุบัน จะไม่ถูกส่งคืนไปยังไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8-บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* or hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วยอ็อพชันอื่นในอ็อพชัน 43
exclude	exclude *IP แอดเดรส*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง exclude คำสั่ง exclude ไม่ถูกต้องในระดับคอนเทนเนอร์สากล หรือ ฐานข้อมูล คำสั่ง exclude ลบแอดเดรสหรือช่วงที่ระบุ ออกจากหน้าปัจจุบันบนคอนเทนเนอร์ คำสั่ง exclude ช่วยให้คุณสามารถสร้าง ช่วงที่ไม่ต่อเนื่องสำหรับ subnets หรือคอนเทนเนอร์อื่น

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
exclude	exclude *dotted_quad-dotted_quad*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง exclude คำสั่ง exclude ไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล คำสั่ง exclude ลบแอดเดรสหรือช่วงที่ระบุ ออกจากหน้าปัจจุบันบนคอนเทนเนอร์ คำสั่ง exclude ช่วยให้คุณสามารถสร้าง ช่วงที่ไม่ต่อเนื่องสำหรับ subnets หรือคอนเทนเนอร์อื่น
ช่วง	ช่วง *IP_address*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง ช่วง คำสั่งช่วงไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล ถ้าช่วงเป็นช่วงแรกในคอนเทนเนอร์ที่ไม่ได้ ระบุช่วงบนบรรทัดนิยามคอนเทนเนอร์ ช่วงสำหรับคอนเทนเนอร์ จะกลายเป็นช่วงที่ระบุโดยคำสั่งช่วง คำสั่งช่วงหลังจาก ช่วงแรก หรือคำสั่งช่วงทั้งหมดสำหรับคอนเทนเนอร์ที่ระบุช่วง ในนิยามจะถูกเพิ่มลงในช่วงปัจจุบัน ด้วยคำสั่งช่วง สามารถเพิ่มแอดเดรสหนึ่งหรือชุดของแอดเดรสลงในช่วงได้ ช่วงต้อง พอดีภายในนิยามคอนเทนเนอร์ subnet
ช่วง	ช่วง *dotted_quad-dotted_quad*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง ช่วง คำสั่งช่วงไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล ถ้าช่วงเป็นช่วงแรกในคอนเทนเนอร์ที่ไม่ได้ ระบุช่วงบนบรรทัดนิยามคอนเทนเนอร์ ช่วงสำหรับคอนเทนเนอร์ จะกลายเป็นช่วงที่ระบุโดยคำสั่งช่วง คำสั่งช่วงหลังจาก ช่วงแรก หรือคำสั่งช่วงทั้งหมดสำหรับคอนเทนเนอร์ที่ระบุช่วง ในนิยามจะถูกเพิ่มลงในช่วงปัจจุบัน ด้วยคำสั่งช่วง สามารถเพิ่มแอดเดรสหนึ่งหรือชุดของแอดเดรสลงในช่วงได้ ช่วงต้อง พอดีภายในนิยามคอนเทนเนอร์ subnet

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ไคลเอ็นต์	ไคลเอ็นต์ *hwtype hwaddr* NONE	ใช่	ไม่มี	ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุโดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรสของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศ ล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติเพื่อจับคู่หลายไคลเอ็นต์
ไคลเอ็นต์	ไคลเอ็นต์ *hwtype hwaddr* ANY	ใช่	ไม่มี	ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุโดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรสของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศ ล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติเพื่อจับคู่หลายไคลเอ็นต์
ไคลเอ็นต์	ไคลเอ็นต์ *hwtype hwaddr dotted_quad*	ใช่	ไม่มี	ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุโดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรสของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศ ล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติเพื่อจับคู่หลายไคลเอ็นต์

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ไคลเอ็นต์	ไคลเอ็นต์ *hwtype hwaddr ช่วง*	ใช่	ไม่มี	ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุ โดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรสของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศ ล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้ รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์
คลาส	คลาส *สตริง*	ใช่	ไม่มี	ระบุคลาสคอนเทนเนอร์ที่มีชื่อเป็น *สตริง* สตริงอาจอยู่ใน อัญประกาศหรือไม่ก็ได้ หากอยู่ใน อัญประกาศ อัญประกาศจะถูกลบออกก่อนการเปรียบเทียบ อัญประกาศเป็นสิ่งจำเป็นสำหรับสตริงที่มีช่องว่างหรือแท็บ คอนเทนเนอร์นี้ถูกต้อง ที่ทุกระดับ สามารถระบุช่วงเพื่อบ่งชี้ชุดของแอดเดรสที่จะส่งไปยัง ไคลเอ็นต์ที่มีคลาสนี้ ช่วงเป็น IP แอดเดรส quad จุดเดียวหรือ IP แอดเดรส quad สองจุดที่แบ่งด้วยเครื่องหมายขีด
คลาส	คลาส *สตริง ช่วง*	ใช่	ไม่มี	ระบุคลาสคอนเทนเนอร์ที่มีชื่อเป็น *สตริง* สตริงอาจอยู่ใน อัญประกาศหรือไม่ก็ได้ หากอยู่ใน อัญประกาศ อัญประกาศจะถูกลบออกก่อนการเปรียบเทียบ อัญประกาศเป็นสิ่งจำเป็นสำหรับสตริงที่มีช่องว่างหรือแท็บ คอนเทนเนอร์นี้ถูกต้อง ที่ทุกระดับ สามารถระบุช่วงเพื่อบ่งชี้ชุดของแอดเดรสที่จะส่งไปยัง ไคลเอ็นต์ที่มีคลาสนี้ ช่วงเป็น IP แอดเดรส quad จุดเดียวหรือ IP แอดเดรส quad สองจุดที่แบ่งด้วยเครื่องหมายขีด

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
เครือข่าย	เครือข่าย *network id netmask*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
เครือข่าย	เครือข่าย *id เครือข่าย*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
เครือข่าย	เครือข่าย *id เครือข่าย ช่วง*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id* hex""	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id* hex""	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id* 0xdata	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id""*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id ช่วง*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id ช่วง* hex""	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id ช่วง* hex ""	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id ช่วง* 0xdata	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id ช่วง ""*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
inoption	inoption *ตัวเลข option_data*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
inoption	inoption *ตัวเลข option_data ช่วง*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเองที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุหมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชันสำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ `0x` นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะเดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย `"!` (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันไม่เป็นที่รู้จัก ของเซิร์ฟเวอร์จะเป็นสตริงฐานสิบหกของไบต์ที่ไม่นำหน้าด้วย อักขระ `0x`
เสมือน	virtual fill *id id* ...	ไม่	ไม่มี	ระบุ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
เสมือน	virtual sfill *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน
เสมือน	virtual rotate *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
เสมือน	virtual srotate *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน
inorder:	inorder: *id id* ...	ไม่	ไม่มี	ระบุ subnet เสมือนที่มีนโยบาย fill ซึ่งหมายความว่าใช้ แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet ID เหมือนกัน
balance:	balance: *id id* ...	ไม่	ไม่มี	ระบุ subnet เสมือนที่มีนโยบาย rotate ซึ่งหมายความว่าใช้ แอดเดรสถัดไปในคอนเทนเนอร์ถัดไป สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet ID เหมือนกัน
supportBootp	supportBootp true	ไม่	ใช่	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ **BOOTP** หรือไม่
supportBootp	supportBootp 1	ไม่	ใช่	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ **BOOTP** หรือไม่

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
supportBootp	supportBootp yes	ไม่	ใช่	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ **BOOTP** หรือไม่
supportBootp	supportBootp false	ไม่	ใช่	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ **BOOTP** หรือไม่
supportBootp	supportBootp 0	ไม่	ใช่	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ **BOOTP** หรือไม่
supportBootp	supportBootp no	ไม่	ใช่	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ **BOOTP** หรือไม่
supportBootp				ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ **BOOTP** หรือไม่
supportUnlistedclients	supportUnlistedclients BOTH	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
supportUnlistedclients	supportUnlistedclients DHCP	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE
supportUnlistedclients	supportUnlistedclients BOOTP	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE
supportUnlistedclients	supportUnlistedclients NONE	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
supportUnlistedclients	supportUnlistedclients true	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE
supportUnlistedclients	supportUnlistedclients yes	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE
supportUnlistedclients	supportUnlistedclients 1	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
supportUnlistedclients	supportUnlistedclients false	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE
supportUnlistedclients	supportUnlistedclients no	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE
supportUnlistedclients	supportUnlistedclients 0	ไม่	ทั้งสอง	ระบุว่าคอนเทนเนอร์ปัจจุบันและที่ต่ำกว่าทั้งหมด (จนกว่าถูกยกเลิก) ควรสนับสนุนไคลเอ็นต์ที่ไม่ได้แสดงรายการหรือไม่ ค่าบ่งชี้ว่าไคลเอ็นต์ทั้งหมดควรได้รับอนุญาตให้เข้าถึงโดยไม่มีคำสั่งไคลเอ็นต์เฉพาะ, ไคลเอ็นต์ **DHCP** อย่างเดียว, ไคลเอ็นต์ **BOOTP** อย่างเดียว, หรือไม่อนุญาตไคลเอ็นต์ใดเลย **หมายเหตุ:** ค่า จริง และ เท็จ ได้รับการสนับสนุนเพื่อให้เข้ากันได้กับเวอร์ชันก่อนหน้านี้และถูกคัดค้าน ค่า จริง สอดคล้องกับ BOTH และค่า เท็จ สอดคล้องกับ NONE

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
addressrecorddb	addressrecrddb *พาธ*	ไม่	ไม่มี	หากระบุ จะทำงานคล้ายกับคีย์เวิร์ด **backupfile** ถูกต้องในระดับคอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น **หมายเหตุ:** เมธอด นี้ถูกคัดค้าน
backupfile	backupfile *พาธ*	ไม่	/etc/db_file.crbk	ระบุไฟล์ที่จะใช้สำหรับสำเนาสำรองฐานข้อมูล ถูกต้องในระดับคอนเทนเนอร์ สากลหรือฐานข้อมูลเท่านั้น
checkpointfile	checkpointfile *พาธ*	ไม่	/etc/db_file.chkpt	ระบุไฟล์ checkpoint ฐานข้อมูล ไฟล์ checkpoint แรก คือ *พาธ* ไฟล์ checkpoint ที่สองคือ *พาธ* ที่มีอักขระตัวสุดท้ายถูกแทนที่ด้วย 2 ดังนั้น ไฟล์ checkpoint จึงไม่ควรลงท้าย ด้วย 2 ถูกต้องในระดับคอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
clientrecorddb	clientrecorddb *พาธ*	ไม่	/etc/db_file.cr	ระบุไฟล์บันทึกฐานข้อมูล ไฟล์มีไคลเอ็นต์เร็กคอร์ดทั้งหมด ที่เซิร์ฟเวอร์ **DHCP** ให้บริการแล้ว ถูกต้องในระดับคอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
bootstrapserver	bootstrapserver *IP แอดเดรส*	ไม่	ไม่มี	ระบุเซิร์ฟเวอร์ที่ไคลเอ็นต์ควรจะใช้ไฟล์ **TFTP** หลังจากได้รับแพ็กเก็ต **BOOTP** หรือ **DHCP** ค่านี้กรอกข้อมูลในฟิลด์ **siaddr** ในแพ็กเก็ต ค่านี้ถูกต้อง ที่คอนเทนเนอร์ทุกระดับ
giaddrfield	giaddrfield *IP แอดเดรส*	ไม่	ไม่มี	ระบุ giaddrfield สำหรับแพ็กเก็ตการตอบกลับ **หมายเหตุ:** การระบุนี้ไม่ถูกต้องในโปรโตคอล **BOOTP** และ **DHCP** แต่ บางไคลเอ็นต์ต้องการฟิลด์ **giaddr** เป็นดีฟอลต์เกตเวย์สำหรับเครือข่าย เนื่องจากความขัดแย้งที่อาจเกิดขึ้นได้นี้ จึงควรใช้ giaddrfield ภายในไคลเอ็นต์คอนเทนเนอร์เท่านั้น แม้ว่าสามารถทำงาน ได้ในทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
pingTime	pingTime *n time_unit*	ไม่	3 วินาที	ระบุจำนวนเวลาที่จะรอสำหรับการตอบกลับ ping ก่อนการส่งแอดเตรส หน่วยเวลาดีฟอลต์คือเศษหนึ่งส่วนร้อยของวินาที ค่าหน่วยเวลา มีการกำหนดไว้ในหมายเหตุก่อนหน้าตารางนี้ ค่านี้ถูกต้องที่คอนเทนเนอร์ ทุกระดับ พารามิเตอร์ *time_unit* เป็นทางเลือก
bootptime	bootptime *n time_unit*	ไม่	-1, ไม่สิ้นสุด	ระบุระยะเวลาในการเช่าแอดเตรสให้กับไคลเอ็นต์ **BOOTP** ค่าดีฟอลต์คือ -1 ซึ่งหมายถึงไม่สิ้นสุด มีค่าหน่วยเวลาปกติ พารามิเตอร์ *time unit* เป็นทางเลือก ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
AllRoutesBroadcast	allroutesbroadcast no	ไม่	0	ระบุว่าควรจะแพร่สัญญาณการตอบกลับไปยังเราต์ทั้งหมดหรือไม่ ถ้าต้องการการตอบกลับการแพร่สัญญาณ ค่านี้ถูกต้องที่คอนเทนเนอร์ ทุกระดับ ค่านี้จะถูกละเว้นโดยเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการ เนื่องจาก MAC แอดเตรสจริงของไคลเอ็นต์ รวมถึง RIFs มีการจัดเก็บไว้สำหรับแพ็กเก็ต ส่งคืน ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
AllRoutesBroadcast	allroutesbroadcast false	ไม่	0	ระบุว่าควรจะแพร่สัญญาณการตอบกลับไปยังเราต์ทั้งหมดหรือไม่ ถ้าต้องการการตอบกลับการแพร่สัญญาณ ค่านี้ถูกต้องที่คอนเทนเนอร์ ทุกระดับ ค่านี้จะถูกละเว้นโดยเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการ เนื่องจาก MAC แอดเตรสจริงของไคลเอ็นต์ รวมถึง RIFs มีการจัดเก็บไว้สำหรับแพ็กเก็ต ส่งคืน ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
AllRoutesBroadcast	allroutesbroadcast 0	ไม่	0	ระบุว่าควรจะแพร่สัญญาณการตอบกลับไปยังเราต์ทั้งหมดหรือไม่ ถ้าต้องการการตอบกลับการแพร่สัญญาณ ค่านี้ถูกต้องที่คอนเทนเนอร์ ทุกระดับ ค่านี้จะถูกละเว้นโดยเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการ เนื่องจาก MAC แอดเตรสจริงของไคลเอ็นต์ รวมถึง RIFs มีการจัดเก็บไว้สำหรับแพ็กเก็ต ส่งคืน ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
AllRoutesBroadcast	allroutesbroadcast yes	ไม่	0	ระบุว่าควรจะแพร่สัญญาณการตอบกลับไปยังเราต์ทั้งหมดหรือไม่ ถ้าต้องการการตอบกลับการแพร่สัญญาณ ค่านี้ถูกต้องที่คอนเทนเนอร์ ทุกระดับ ค่านี้จะถูกละเว้นโดยเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการ เนื่องจาก MAC แอดเดรสจริงของไคลเอ็นต์ รวมถึง RIFs มีการจัดเก็บไว้สำหรับแพ็กเก็ต ส่งคืน ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
AllRoutesBroadcast	allroutesbroadcast true	ไม่	0	ระบุว่าควรจะแพร่สัญญาณการตอบกลับไปยังเราต์ทั้งหมดหรือไม่ ถ้าต้องการการตอบกลับการแพร่สัญญาณ ค่านี้ถูกต้องที่คอนเทนเนอร์ ทุกระดับ ค่านี้จะถูกละเว้นโดยเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการ เนื่องจาก MAC แอดเดรสจริงของไคลเอ็นต์ รวมถึง RIFs มีการจัดเก็บไว้สำหรับแพ็กเก็ต ส่งคืน ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
AllRoutesBroadcast	allroutesbroadcast 1	ไม่	0	ระบุว่าควรจะแพร่สัญญาณการตอบกลับไปยังเราต์ทั้งหมดหรือไม่ ถ้าต้องการการตอบกลับการแพร่สัญญาณ ค่านี้ถูกต้องที่คอนเทนเนอร์ ทุกระดับ ค่านี้จะถูกละเว้นโดยเซิร์ฟเวอร์ **DHCP** ของระบบปฏิบัติการ เนื่องจาก MAC แอดเดรสจริงของไคลเอ็นต์ รวมถึง RIFs มีการจัดเก็บไว้สำหรับแพ็กเก็ต ส่งคืน ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
addressassigned	addressassigned *"string"*	ไม่	ไม่มี	ระบุสตริงในอัญประกาศที่จะดำเนินการเมื่อกำหนดแอดเดรสให้กับ ไคลเอ็นต์ สตริงควรมีสอง %s โดย %s แรกเป็น id ไคลเอ็นต์ ใน รูปแบบ *ชนิด-สตริง* %s ที่สองเป็น IP แอดเดรสในรูปแบบจุด quad ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
addressreleased	addressreleased *"string"*	ไม่	ไม่มี	ระบุสตริงในอัญประกาศที่จะดำเนินการเมื่อรีลีสแอดเดรสโดยไคลเอ็นต์ สตริงควรมีหนึ่ง %s โดย %s เป็น IP แอดเดรสที่กำลังรีลีส ในรูปแบบจุด quad ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
appenddomain	appenddomain 0	ไม่	ไม่	ระบุว่าจะผนวกชื่อโดเมนอ็อพชัน 15 ที่กำหนดไว้เข้ากับชื่อโฮสต์ที่ไคลเอ็นต์แนะนำหรือไม่ในกรณีที่ไคลเอ็นต์ไม่ได้ แนะนำชื่อโดเมนด้วย ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
appenddomain	appenddomain no	ไม่	ไม่	ระบุว่าจะผนวกชื่อโดเมนอ็อพชัน 15 ที่กำหนดไว้เข้ากับชื่อโฮสต์ที่ไคลเอ็นต์แนะนำหรือไม่ในกรณีที่ไคลเอ็นต์ไม่ได้ แนะนำชื่อโดเมนด้วย ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
appenddomain	appenddomain false	ไม่	ไม่	ระบุว่าจะผนวกชื่อโดเมนอ็อพชัน 15 ที่กำหนดไว้เข้ากับชื่อโฮสต์ที่ไคลเอ็นต์แนะนำหรือไม่ในกรณีที่ไคลเอ็นต์ไม่ได้ แนะนำชื่อโดเมนด้วย ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
appenddomain	appenddomain 1	ไม่	ไม่	ระบุว่าจะผนวกชื่อโดเมนอ็อพชัน 15 ที่กำหนดไว้เข้ากับชื่อโฮสต์ที่ไคลเอ็นต์แนะนำหรือไม่ในกรณีที่ไคลเอ็นต์ไม่ได้ แนะนำชื่อโดเมนด้วย ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
appenddomain	appenddomain yes	ไม่	ไม่	ระบุว่าจะผนวกชื่อโดเมนอ็อพชัน 15 ที่กำหนดไว้เข้ากับชื่อโฮสต์ที่ไคลเอ็นต์แนะนำหรือไม่ในกรณีที่ไคลเอ็นต์ไม่ได้ แนะนำชื่อโดเมนด้วย ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
appenddomain	appenddomain true	ไม่	ไม่	ระบุว่าจะผนวกชื่อโดเมนอ็อพชัน 15 ที่กำหนดไว้เข้ากับชื่อโฮสต์ที่ไคลเอ็นต์แนะนำหรือไม่ในกรณีที่ไคลเอ็นต์ไม่ได้ แนะนำชื่อโดเมนด้วย ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
canonical	canonical 0	ไม่	0	ระบุว่า id ไคลเอ็นต์อยู่ในรูปแบบที่บัญญัติ ค่านี้ ถูกต้องในไคลเอ็นต์คอนเทนเนอร์เท่านั้น
canonical	canonical no	ไม่	0	ระบุว่า id ไคลเอ็นต์อยู่ในรูปแบบที่บัญญัติ ค่านี้ ถูกต้องในไคลเอ็นต์คอนเทนเนอร์เท่านั้น
canonical	canonical false	ไม่	0	ระบุว่า id ไคลเอ็นต์อยู่ในรูปแบบที่บัญญัติ ค่านี้ ถูกต้องในไคลเอ็นต์คอนเทนเนอร์เท่านั้น

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
canonical	canonical 1	ไม่	0	ระบุว่า id ไคลเอ็นต์อยู่ในรูปแบบที่บัญญัติ ค่านี้ ถูกต้องในไคลเอ็นต์คอนเทนเนอร์เท่านั้น
canonical	canonical yes	ไม่	0	ระบุว่า id ไคลเอ็นต์อยู่ในรูปแบบที่บัญญัติ ค่านี้ ถูกต้องในไคลเอ็นต์คอนเทนเนอร์เท่านั้น
canonical	canonical true	ไม่	0	ระบุว่า id ไคลเอ็นต์อยู่ในรูปแบบที่บัญญัติ ค่านี้ ถูกต้องในไคลเอ็นต์คอนเทนเนอร์เท่านั้น
leaseTimeDefault	leaseTimeDefault *n time_unit*	ไม่	86400 วินาที	ระบุเวลาเช่าดีฟอลต์สำหรับไคลเอ็นต์ ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ พารามิเตอร์ *time_unit* เป็นทางเลือก
proxyarec	proxyarec never	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS `never` หมายถึงไม่เคยอัพเดต เร็กคอร์ด A `usedhcpddns` หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น `usedhcpddnsplus` หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ `always` หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด `XXXXprotected` แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต `standard` เป็นคำพ้อง ของ `always` `protected` เป็นคำพ้องของ `alwaysprotected` ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
proxyarec	proxyarec usedhcpddns	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS `never` หมายถึงไม่เคยอัพเดต เร็กคอร์ด A `usedhcpddns` หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น `usedhcpddnsplus` หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ `always` หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด `XXXXprotected` แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต `standard` เป็นคำพ้อง ของ `always` `protected` เป็นคำพ้องของ `alwaysprotected` ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
proxyarec	proxyarec usedhcpddnsplus	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS never หมายถึงไม่เคยอัพเดต เร็กคอร์ด A usedhcpddns หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น usedhcpddnsplus หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ always หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด XXXXprotected แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต standard เป็นคำพ้อง ของ always protected เป็นคำพ้องของ alwaysprotected ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
proxyarec	proxyarec always	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS never หมายถึงไม่เคยอัพเดต เร็กคอร์ด A usedhcpddns หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น usedhcpddnsplus หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ always หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด XXXXprotected แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต standard เป็นคำพ้อง ของ always protected เป็นคำพ้องของ alwaysprotected ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
proxyarec	proxyarec usedhcpddnsprotected	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS never หมายถึงไม่เคยอัพเดต เร็กคอร์ด A usedhcpddns หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น usedhcpddnsplus หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ always หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด XXXXprotected แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต standard เป็นคำพ้อง ของ always protected เป็นคำพ้องของ alwaysprotected ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
proxyarec	proxyarec usedhcpddnsplusprotected	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS never หมายถึงไม่เคยอัพเดต เร็กคอร์ด A usedhcpddns หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น usedhcpddnsplus หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ always หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด XXXXprotected แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต standard เป็นคำพ้อง ของ always protected เป็นคำพ้องของ alwaysprotected ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
proxyarec	proxyarec alwaysprotected	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS never หมายถึงไม่เคยอัพเดต เร็กคอร์ด A usedhcpddns หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น usedhcpddnsplus หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ always หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด XXXXprotected แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต standard เป็นคำพ้อง ของ always protected เป็นคำพ้องของ alwaysprotected ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
proxyarec	proxyarec standard	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS never หมายถึงไม่เคยอัพเดต เร็กคอร์ด A usedhcpddns หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น usedhcpddnsplus หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ always หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด XXXXprotected แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต standard เป็นคำพ้อง ของ always protected เป็นคำพ้องของ alwaysprotected ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
	proxyarec protected	ไม่	usedhcpddnsplus	ระบุอ็อพชันและเมธอดที่ควรจะใช้สำหรับอัพเดตเร็กคอร์ด A ใน DNS `never` หมายถึงไม่เคยอัพเดต เร็กคอร์ด A `usedhcpddns` หมายถึงใช้อ็อพชัน 81 ถ้าไคลเอ็นต์ ระบุอ็อพชันนั้น `usedhcpddnsplus` หมายถึงใช้อ็อพชัน 81 หรืออ็อพชัน 12 และ 15 ถ้ามีการระบุ `always` หมายถึงทำอัพเดตเร็กคอร์ด A สำหรับ ไคลเอ็นต์ทั้งหมด `XXXXprotected` แก้ไขคำสั่ง **nsupdate** เพื่อให้แน่ใจว่าไคลเอ็นต์ได้รับอนุญาต `standard` เป็นคำพ้อง ของ `always` `protected` เป็นคำพ้องของ `alwaysprotected` ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
releasednsA	releasednsA *"string"*	ไม่	ไม่มี	ระบุสตริงดำเนินการที่จะใช้เมื่อรีลีสแอดเดรส สตริงใช้เพื่อลบเร็กคอร์ด A ที่เชื่อมโยงกับแอดเดรสที่รีลีส ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
releasednsP	releasednsP *"string"*	ไม่	ไม่มี	ระบุสตริงดำเนินการที่จะใช้เมื่อรีลีสแอดเดรส สตริงใช้เพื่อลบเร็กคอร์ด PTR ที่เชื่อมโยงกับแอดเดรสที่รีลีส ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
removedns	removedns *"string"*	ไม่	ไม่มี	ระบุสตริงดำเนินการที่จะใช้เมื่อรีลีสแอดเดรส สตริงใช้เพื่อลบเร็กคอร์ด PTR และ A ที่เชื่อมโยงกับแอดเดรสที่ รีลีส ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ **หมายเหตุ:** เมธอดนี้ใช้เพื่อสนับสนุนคีย์เวิร์ด **releasednsA** และ **releasednsP**
updatedns	updatedns *"string"*	ไม่	ไม่มี	ระบุสตริงดำเนินการที่จะใช้เมื่อผูกแอดเดรส สตริง ใช้เพื่ออัพเดตทั้งเร็กคอร์ด A และ PTR ที่เชื่อมโยงกับ แอดเดรส ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ **หมายเหตุ:** เมธอดนี้ใช้เพื่อสนับสนุนคีย์เวิร์ด **updatednsA** และ **updatednsP**

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
updatednsA	updatednsA *"string"*	ไม่	ไม่มี	ระบุสตริงดำเนินการที่จะใช้เมื่อผูกแอดเดรส สตริง ใช้เพื่ออัพเดตเร็กคอร์ด A ที่เชื่อมโยงกับแอดเดรส ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
updatednsP	updatednsP *"string"*	ไม่	ไม่มี	ระบุสตริงดำเนินการที่จะใช้เมื่อผูกแอดเดรส สตริง ใช้เพื่ออัพเดตเร็กคอร์ด PTR ที่เชื่อมโยงกับแอดเดรส ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
hostnamepolicy	hostnamepolicy suggested	ไม่	ดีฟอลต์	ระบุชื่อโฮสต์ที่จะส่งกลับไปยังไคลเอ็นต์ นโยบาย ดีฟอลต์คือแนะนำให้ใช้ชื่อโฮสต์และชื่อโดเมนที่กำหนดมากกว่าชื่อที่แนะนำ นโยบายอื่นมีการบังคับใช้อย่างเข้มงวด (ตัวอย่างเช่น: `defined` จะ ส่งคืนชื่อที่กำหนด หรือ ไม่มีถ้าไม่มีการกำหนดชื่อไว้ในคอนฟิกูเรชัน) นอกจากนี้ นโยบายที่ใช้ตัวแก้ไข `always` จะสั่งให้ เซิร์ฟเวอร์ส่งคืนอ็อพชันชื่อโฮสต์โดยไม่คำนึงว่าไคลเอ็นต์ร้องขอ ผ่านทางอ็อพชันรายการพารามิเตอร์ หรือไม่ โปรดทราบว่าการแนะนำชื่อโฮสต์ยัง หมายถึงการร้องขอด้วย และสามารถแนะนำชื่อโฮสต์ผ่านทางอ็อพชัน 81 หรือ ผ่านทางอ็อพชัน 12 และ 15 คีย์เวิร์ดนี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
hostnamepolicy	hostnamepolicy resolved	ไม่	ดีฟอลต์	ระบุชื่อโฮสต์ที่จะส่งกลับไปยังไคลเอ็นต์ นโยบาย ดีฟอลต์คือแนะนำให้ใช้ชื่อโฮสต์และชื่อโดเมนที่กำหนดมากกว่าชื่อที่แนะนำ นโยบายอื่นมีการบังคับใช้อย่างเข้มงวด (ตัวอย่างเช่น: `defined` จะ ส่งคืนชื่อที่กำหนด หรือ ไม่มีถ้าไม่มีการกำหนดชื่อไว้ในคอนฟิกูเรชัน) นอกจากนี้ นโยบายที่ใช้ตัวแก้ไข `always` จะสั่งให้ เซิร์ฟเวอร์ส่งคืนอ็อพชันชื่อโฮสต์โดยไม่คำนึงว่าไคลเอ็นต์ร้องขอ ผ่านทางอ็อพชันรายการพารามิเตอร์ หรือไม่ โปรดทราบว่าการแนะนำชื่อโฮสต์ยัง หมายถึงการร้องขอด้วย และสามารถแนะนำชื่อโฮสต์ผ่านทางอ็อพชัน 81 หรือ ผ่านทางอ็อพชัน 12 และ 15 คีย์เวิร์ดนี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
hostnamepolicy	hostnamepolicy always_resolved	ไม่	ดีฟอลต์	ระบุชื่อโฮสต์ที่จะส่งกลับไปยังไคลเอ็นต์ นโยบาย ดีฟอลต์คือแนะนำให้ใช้ชื่อโฮสต์และชื่อโดเมนที่กำหนดมากกว่าชื่อที่แนะนำ นโยบายอื่นมีการบังคับใช้อย่างเข้มงวด (ตัวอย่างเช่น: `defined` จะ ส่งคืนชื่อที่กำหนด หรือ ไม่มี ถ้าไม่มีการกำหนดชื่อไว้ในคอนฟิกูเรชัน) นอกจากนี้ นโยบายที่ใช้ตัวแก้ไข `always` จะสั่งให้ เซิร์ฟเวอร์ส่งคืนอ็อพชันชื่อโฮสต์ โดยไม่คำนึงว่าไคลเอ็นต์ร้องขอ ผ่านทางอ็อพชันรายการพารามิเตอร์หรือไม่ โปรดทราบว่าการแนะนำชื่อโฮสต์ยัง หมายถึงการร้องขอด้วย และสามารถแนะนำชื่อโฮสต์ผ่านทางอ็อพชัน 81 หรือ ผ่านทางอ็อพชัน 12 และ 15 คีย์เวิร์ดนี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
hostnamepolicy	hostnamepolicy defined	ไม่	ดีฟอลต์	ระบุชื่อโฮสต์ที่จะส่งกลับไปยังไคลเอ็นต์ นโยบาย ดีฟอลต์คือแนะนำให้ใช้ชื่อโฮสต์และชื่อโดเมนที่กำหนดมากกว่าชื่อที่แนะนำ นโยบายอื่นมีการบังคับใช้อย่างเข้มงวด (ตัวอย่างเช่น: `defined` จะ ส่งคืนชื่อที่กำหนด หรือ ไม่มี ถ้าไม่มีการกำหนดชื่อไว้ในคอนฟิกูเรชัน) นอกจากนี้ นโยบายที่ใช้ตัวแก้ไข `always` จะสั่งให้ เซิร์ฟเวอร์ส่งคืนอ็อพชันชื่อโฮสต์ โดยไม่คำนึงว่าไคลเอ็นต์ร้องขอ ผ่านทางอ็อพชันรายการพารามิเตอร์หรือไม่ โปรดทราบว่าการแนะนำชื่อโฮสต์ยัง หมายถึงการร้องขอด้วย และสามารถแนะนำชื่อโฮสต์ผ่านทางอ็อพชัน 81 หรือ ผ่านทางอ็อพชัน 12 และ 15 คีย์เวิร์ดนี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
hostnamepolicy	hostnamepolicy always_defined	ไม่	ดีฟอลต์	ระบุชื่อโฮสต์ที่จะส่งกลับไปยังไคลเอ็นต์ นโยบาย ดีฟอลต์คือแนะนำให้ใช้ชื่อโฮสต์และชื่อโดเมนที่กำหนดมากกว่าชื่อที่แนะนำ นโยบายอื่นมีการบังคับใช้อย่างเข้มงวด (ตัวอย่างเช่น: `defined` จะ ส่งคืนชื่อที่กำหนด หรือ ไม่มี ถ้าไม่มีการกำหนดชื่อไว้ในคอนฟิกูเรชัน) นอกจากนี้ นโยบายที่ใช้ตัวแก้ไข `always` จะสั่งให้ เซิร์ฟเวอร์ส่งคืนอ็อพชันชื่อโฮสต์ โดยไม่คำนึงว่าไคลเอ็นต์ร้องขอ ผ่านทางอ็อพชันรายการพารามิเตอร์หรือไม่ โปรดทราบว่าการแนะนำชื่อโฮสต์ยัง หมายถึงการร้องขอด้วย และสามารถแนะนำชื่อโฮสต์ผ่านทางอ็อพชัน 81 หรือ ผ่านทางอ็อพชัน 12 และ 15 คีย์เวิร์ดนี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
hostnamepolicy	hostnamepolicy default	ไม่	ดีฟอลต์	ระบุชื่อโฮสต์ที่จะส่งกลับไปยังไคลเอ็นต์ นโยบาย ดีฟอลต์คือแนะนำให้ใช้ชื่อโฮสต์และชื่อโดเมนที่กำหนดมากกว่าชื่อที่แนะนำ นโยบายอื่นมีการบังคับใช้อย่างเข้มงวด (ตัวอย่างเช่น: `defined` จะ ส่งคืนชื่อที่กำหนด หรือ ไม่มี ถ้าไม่มีการกำหนดชื่อไว้ในคอนฟิกูเรชัน) นอกจากนี้ นโยบายที่ใช้ตัวแก้ไข `always` จะสั่งให้ เซิร์ฟเวอร์ส่งคืนอ็อพชันชื่อโฮสต์ โดยไม่คำนึงว่าไคลเอ็นต์ร้องขอ ผ่านทางอ็อพชันรายการพารามิเตอร์หรือไม่ โปรดทราบว่าการแนะนำชื่อโฮสต์ยัง หมายถึงการร้องขอด้วย และสามารถแนะนำชื่อโฮสต์ผ่านทางอ็อพชัน 81 หรือ ผ่านทางอ็อพชัน 12 และ 15 คีย์เวิร์ดนี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
bootfilepolicy	bootfilepolicy suggested	ไม่	ที่แนะนำ	ระบุการกำหนดค่าตามความชอบสำหรับการส่งคืนชื่อ bootfile ไปยังไคลเอ็นต์ `suggested` แนะนำให้ใช้ชื่อ bootfile ที่ไคลเอ็นต์แนะนำมากกว่าชื่อที่เซิร์ฟเวอร์กำหนด คอนฟิก `merge` จะผนวกชื่อที่ไคลเอ็นต์แนะนำ เข้ากับไดเร็กทอรีโฮมที่เซิร์ฟเวอร์กำหนดคอนฟิก `defined` แนะนำให้ใช้ชื่อที่กำหนดไว้มากกว่าชื่อ bootfile ที่แนะนำ `always` ส่งคืน ชื่อที่กำหนดไว้โดยไม่คำนึงว่าไคลเอ็นต์ร้องขออ็อพชัน bootfile ผ่านทางอ็อพชันรายการพารามิเตอร์หรือไม่
bootfilepolicy	bootfilepolicy merge	ไม่	ที่แนะนำ	ระบุการกำหนดค่าตามความชอบสำหรับการส่งคืนชื่อ bootfile ไปยังไคลเอ็นต์ `suggested` แนะนำให้ใช้ชื่อ bootfile ที่ไคลเอ็นต์แนะนำมากกว่าชื่อที่เซิร์ฟเวอร์กำหนด คอนฟิก `merge` จะผนวกชื่อที่ไคลเอ็นต์แนะนำ เข้ากับไดเร็กทอรีโฮมที่เซิร์ฟเวอร์กำหนดคอนฟิก `defined` แนะนำให้ใช้ชื่อที่กำหนดไว้มากกว่าชื่อ bootfile ที่แนะนำ `always` ส่งคืน ชื่อที่กำหนดไว้โดยไม่คำนึงว่าไคลเอ็นต์ร้องขออ็อพชัน bootfile ผ่านทางอ็อพชันรายการพารามิเตอร์หรือไม่
bootfilepolicy	bootfilepolicy defined	ไม่	ที่แนะนำ	ระบุการกำหนดค่าตามความชอบสำหรับการส่งคืนชื่อ bootfile ไปยังไคลเอ็นต์ `suggested` แนะนำให้ใช้ชื่อ bootfile ที่ไคลเอ็นต์แนะนำมากกว่าชื่อที่เซิร์ฟเวอร์กำหนด คอนฟิก `merge` จะผนวกชื่อที่ไคลเอ็นต์แนะนำ เข้ากับไดเร็กทอรีโฮมที่เซิร์ฟเวอร์กำหนดคอนฟิก `defined` แนะนำให้ใช้ชื่อที่กำหนดไว้มากกว่าชื่อ bootfile ที่แนะนำ `always` ส่งคืน ชื่อที่กำหนดไว้โดยไม่คำนึงว่าไคลเอ็นต์ร้องขออ็อพชัน bootfile ผ่านทางอ็อพชันรายการพารามิเตอร์หรือไม่

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
bootfilepolicy	bootfilepolicy always	ไม่	ที่แนะนำ	ระบุการกำหนดค่าตามความชอบสำหรับการส่งคืนชื่อ bootfile ไปยังไคลเอ็นต์ `suggested` แนะนำให้ใช้ชื่อ bootfile ที่ไคลเอ็นต์แนะนำมากกว่าชื่อที่เซิร์ฟเวอร์กำหนด คอนฟิก `merge` จะผนวกชื่อที่ไคลเอ็นต์แนะนำ เข้ากับไดเร็กทอรีโฮมที่เซิร์ฟเวอร์กำหนดคอนฟิก `defined` แนะนำให้ใช้ชื่อที่กำหนดไว้มากกว่าชื่อ bootfile ที่แนะนำ `always` ส่งคืน ชื่อที่กำหนดไว้โดยไม่คำนึงว่าไคลเอ็นต์ร้องขออ็อพชัน bootfile ผ่านทางอ็อพชันรายการพารามิเตอร์หรือไม่
stealfromchildren	stealfromchildren true	ไม่	ไม่	ระบุว่าคอนเทนเนอร์พาเรนต์ควร "ขโมย" จากคอนเทนเนอร์ซายด์หรือไม่ เมื่อคอนเทนเนอร์พาเรนต์ขาดแอดเดรส นี่หมายความว่าถ้าคุณมี subnet ซึ่งมีคลาสที่กำหนดด้วยช่วงของแอดเดรส แอดเดรสเหล่านั้นจะถูกสงวนไว้สำหรับไคลเอ็นต์ดังกล่าวที่ระบุคลาสนั้น ถ้า `stealfromchildren` เป็น จริง แอดเดรสจะถูกดึงมาจากซายด์เพื่อพยายามและตอบสนองต่อคำร้องขอ ค่าดีฟอลต์คือไม่ขโมยแอดเดรส
stealfromchildren	stealfromchildren 1	ไม่	ไม่	ระบุว่าคอนเทนเนอร์พาเรนต์ควร "ขโมย" จากคอนเทนเนอร์ซายด์หรือไม่ เมื่อคอนเทนเนอร์พาเรนต์ขาดแอดเดรส นี่หมายความว่าถ้าคุณมี subnet ซึ่งมีคลาสที่กำหนดด้วยช่วงของแอดเดรส แอดเดรสเหล่านั้นจะถูกสงวนไว้สำหรับไคลเอ็นต์ดังกล่าวที่ระบุคลาสนั้น ถ้า `stealfromchildren` เป็น จริง แอดเดรสจะถูกดึงมาจากซายด์เพื่อพยายามและตอบสนองต่อคำร้องขอ ค่าดีฟอลต์คือไม่ขโมยแอดเดรส

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
stealfromchildren	stealfromchildren yes	ไม่	ไม่	ระบุว่าคอนเทนเนอร์พาเรนต์ควร "ขโมย" จากคอนเทนเนอร์ชายด์หรือไม่ เมื่อคอนเทนเนอร์พาเรนต์ขาดแอดเดรส นี่หมายความว่าถ้าคุณมี subnet ซึ่งมีคลาสที่กำหนดด้วยช่วงของแอดเดรส แอดเดรสเหล่านั้นจะถูกสงวนไว้สำหรับไคลเอ็นต์ดังกล่าวที่ระบุคลาสนั้น ถ้า `stealfromchildren` เป็น จริง แอดเดรสจะถูกดึงมาจากชายด์เพื่อพยายามและตอบสนองต่อคำร้องขอ ค่าดีฟอลต์คือไม่ขโมยแอดเดรส
stealfromchildren	stealfromchildren false	ไม่	ไม่	ระบุว่าคอนเทนเนอร์พาเรนต์ควร "ขโมย" จากคอนเทนเนอร์ชายด์หรือไม่ เมื่อคอนเทนเนอร์พาเรนต์ขาดแอดเดรส นี่หมายความว่าถ้าคุณมี subnet ซึ่งมีคลาสที่กำหนดด้วยช่วงของแอดเดรส แอดเดรสเหล่านั้นจะถูกสงวนไว้สำหรับไคลเอ็นต์ดังกล่าวที่ระบุคลาสนั้น ถ้า `stealfromchildren` เป็น จริง แอดเดรสจะถูกดึงมาจากชายด์เพื่อพยายามและตอบสนองต่อคำร้องขอ ค่าดีฟอลต์คือไม่ขโมยแอดเดรส
stealfromchildren	stealfromchildren 0	ไม่	ไม่	ระบุว่าคอนเทนเนอร์พาเรนต์ควร "ขโมย" จากคอนเทนเนอร์ชายด์หรือไม่ เมื่อคอนเทนเนอร์พาเรนต์ขาดแอดเดรส นี่หมายความว่าถ้าคุณมี subnet ซึ่งมีคลาสที่กำหนดด้วยช่วงของแอดเดรส แอดเดรสเหล่านั้นจะถูกสงวนไว้สำหรับไคลเอ็นต์ดังกล่าวที่ระบุคลาสนั้น ถ้า `stealfromchildren` เป็น จริง แอดเดรสจะถูกดึงมาจากชายด์เพื่อพยายามและตอบสนองต่อคำร้องขอ ค่าดีฟอลต์คือไม่ขโมยแอดเดรส

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย	ค่าดีฟอลต์	ความหมาย
stealfromchildren	stealfromchildren no	ไม่	ไม่	ระบุว่าคอนเทนเนอร์พาเรนต์ควร "ขโมย" จากคอนเทนเนอร์ซายด์หรือไม่ เมื่อคอนเทนเนอร์พาเรนต์ขาดแอดเดรส นี่หมายความว่าถ้าคุณมี subnet ซึ่งมีคลาสที่กำหนดด้วยช่วงของแอดเดรส แอดเดรสเหล่านั้นจะถูกสงวนไว้สำหรับไคลเอ็นต์ดังกล่าวที่ระบุคลาสนั้น ถ้า `stealfromchildren` เป็น จริง แอดเดรสจะถูกดึงมาจากซายด์เพื่อพยายามและตอบสนองต่อคำร้องขอ ค่าดีฟอลต์คือไม่ขโมยแอดเดรส
homedirectory	homedirectory *พาธ*	ไม่	ไม่มี	ระบุไดเร็กทอรีโฮมที่จะใช้ในส่วนไฟล์ของแพ็กเก็ต การตอบกลับ ค่านี้สามารถระบุที่คอนเทนเนอร์ทุกระดับ นโยบาย bootfile กำหนด จำนวนไอเท็มที่ระบุในส่วนไฟล์ของแพ็กเก็ตขาเข้าที่โต้ตอบกับ bootfile และคำสั่งไดเร็กทอรีโฮม
bootfile	bootfile *พาธ*	ไม่	ไม่มี	ระบุ bootfile ที่จะใช้ในส่วนไฟล์ของแพ็กเก็ตการตอบกลับ ค่านี้สามารถระบุที่คอนเทนเนอร์ทุกระดับ นโยบาย bootfile กำหนด จำนวนไอเท็มที่ระบุในส่วนไฟล์ของแพ็กเก็ตขาเข้าที่โต้ตอบกับ bootfile และคำสั่งไดเร็กทอรีโฮม
pxebootfile	pxebootfile *system_architecture major_version minor_version boofilename*	ไม่	ไม่มี	ระบุ bootfile ที่จะกำหนดให้กับไคลเอ็นต์ ค่านี้ใช้เฉพาะ ถ้า **dhcpsd** สนับสนุนไคลเอ็นต์ PXE (**pxeservertype** เป็น **dhcp_pxe_binld**) ตัวแจงส่วนไฟล์คอนฟิกูเรชันสร้างข้อผิดพลาดถ้าจำนวนของพารามิเตอร์หลังจาก `pxebootfile` น้อยกว่าสี่ และละเว้นพารามิเตอร์เพิ่มเติมใดๆ `pxebootfile` สามารถใช้ภายในคอนเทนเนอร์เท่านั้น
supportoption118	supportoption118 *no/yes*	ไม่ สามารถกำหนดใน subnet คอนเทนเนอร์เท่านั้น	ไม่มี	คีย์เวิร์ดนี้ระบุว่าคอนเทนเนอร์นี้สนับสนุน อ็อพชัน 118 หรือไม่ใช่ หมายความว่าสนับสนุน และ ไม่หมายความว่าไม่สนับสนุน เพื่อให้อ็อพชันนี้มีผลบังคับใช้ คุณยังต้องใช้คีย์เวิร์ด **supportsubnetselection** ด้วย

คำแนะนำเกี่ยวกับ DHCP และ Network Installation Management

แนวคิดการกำหนดแอดเดรส Internet Protocol (IP) แบบไดนามิก เป็นเรื่องค่อนข้างใหม่ มีการจัดเตรียมคำแนะนำต่อไปนี้เพื่อช่วยในการโต้ตอบกับ **DHCP** และ Network Installation Management (NIM)

1. เมื่อกำหนดคอนฟิกอ็อบเจ็กต์ในสภาวะแวดล้อม NIM ให้ใช้ชื่อโฮสต์ในทุกเมื่อที่เป็นไปได้ การทำเช่นนี้ช่วยให้คุณสามารถใช้เนมเซิร์ฟเวอร์ไดนามิกที่จะอัพเดต IP แอดเดรสเมื่อชื่อโฮสต์ถูกแปลงเป็น IP แอดเดรสในสภาวะแวดล้อม NIM
2. วางต้นแบบ NIM และเซิร์ฟเวอร์ **DHCP** ไว้บนระบบเดียวกัน เซิร์ฟเวอร์ **DHCP** มีอ็อพชันในอัพเดต DNS สตริงซึ่ง เมื่อตั้งค่า เป็น `NIM` จะพยายามรักษาอ็อบเจ็กต์ NIM ให้อยู่นอกสถานะเหล่านั้น ซึ่งต้องการ IP แอดเดรสแบบสแตติกเมื่อแอดเดรสดังกล่าวเปลี่ยนไป
3. สำหรับไคลเอ็นต์ NIM ให้ตั้งค่าเวลาเช่าดีฟอลต์เป็นสองเท่าของเวลาที่ใช้ในการติดตั้งไคลเอ็นต์ การทำเช่นนี้ช่วยให้ IP แอดเดรสที่เช่าถูกต้องในระหว่าง การติดตั้ง หลังจากการติดตั้ง ให้รีสตาร์ทไคลเอ็นต์ **DHCP** จะ สตาร์ทหรือจะต้องถูกกำหนดคอนฟิก ขึ้นอยู่กับชนิดของการติดตั้ง
4. เซิร์ฟเวอร์ `dhcpsd` ควรรับผิดชอบสำหรับทั้งเร็กคอร์ด PTR และ A DNS เมื่อ NIM ติดตั้งเครื่องอีกครั้ง ไฟล์ที่มี RSA จะถูกลบออก และไคลเอ็นต์ไม่สามารถอัพเดตเร็กคอร์ดของตนได้ เซิร์ฟเวอร์อัพเดต เร็กคอร์ดของระบบ เมื่อต้องการทำเช่นนี้ ให้เปลี่ยนบรรทัด `updatedns` ใน `/etc/dhcpcd.ini` เป็น:

   ```
   updatedns "/usr/sbin/dhcpaction '%s' '%s' '%s' '%s' '%s' NONE NONIM"
   ```

   ในไฟล์ `/etc/dhcpsd.cnf` ให้เปลี่ยนบรรทัด `updatedns` เป็น:

   ```
   updatedns "/usr/sbin/dhcpaction '%s' '%s' '%s' '%s' '%s' BOTH NIM"
   ```

   หมายเหตุ: เมื่ออ็อบเจ็กต์ NIM ถูกวางไว้ในสถานะคงค้างการติดตั้ง BOS เซิร์ฟเวอร์ `dhcpsd` อาจส่งผ่านอาร์กิวเมนต์ซึ่งแตกต่างจากอาร์กิวเมนต์ที่ตั้งใจในครั้งแรก ดังนั้นควรลดเวลาที่ ไคลเอ็นต์อยู่ในสถานะคงค้างนี้ให้เหลือน้อยที่สุด เพื่อหลีกเลี่ยงสถานการณ์นี้

คำแนะนำเหล่านี้ช่วยให้สภาวะแวดล้อม NIM สามารถทำงานกับไดนามิก ไคลเอ็นต์ได้

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ Network Installation Management ให้ดูที่ *AIX 5L™ เวอร์ชัน 5.3 คู่มือและข้อมูลอ้างอิงการจัดการติดตั้งเครือข่าย*

Dynamic Host Configuration Protocol เวอร์ชัน 6

Dynamic Host Configuration Protocol (DHCP) จัดเตรียม วิธีควบคุมคอนฟิกูเรชันเครือข่ายในตำแหน่งรวมศูนย์ หัวข้อนี้เป็นหัวข้อเฉพาะ **DHCPv6** การอ้างอิงทั้งหมดถึง "IP แอดเดรส" หมายถึง IPv6 แอดเดรส และการอ้างอิงทั้งหมดถึง "**DHCP**" หมายถึง **DHCPv6** (ยกเว้นว่ามีการระบุเป็นอย่างอื่น)

เซิร์ฟเวอร์ **DHCPv4** สามารถอยู่ร่วมกันบนลิงก์เดียวกับเซิร์ฟเวอร์ **DHCPv6** สำหรับคำอธิบายเชิงลึกของโปรโตคอล โปรดดู RFC 3315

DHCP เป็นโปรโตคอลชั้นแอ็พพลิเคชันที่ช่วยให้เครื่องไคลเอ็นต์ บนเครือข่ายสามารถเรียกใช้ IP แอดเดรสและพารามิเตอร์คอนฟิกูเรชันอื่นจาก เซิร์ฟเวอร์ พารามิเตอร์เหล่านี้ถูกกำหนดใน *options* ซึ่ง Options ได้รับ โดยแลกเปลี่ยนแพ็กเก็ตระหว่าง daemon บนไคลเอ็นต์กับ daemon อื่นบนเซิร์ฟเวอร์ ข้อความเหล่านี้แลกเปลี่ยนกันในรูปแบบแพ็กเก็ต **UDP** ไคลเอ็นต์ใช้ แอด

เดรสลิงก์โลคัล ผ่านคำสั่ง **autoconf6** หรือวิธีการอื่น เพื่อจำแนกแอดเดรสต้นทางถึงเซิร์ฟเวอร์ เซิร์ฟเวอร์รอสัญญาณ บนแอดเดรสแบบมัลติคาสต์ขอบเขตลิงก์ที่สงวนไว้ รีเลย์เอเจนต์จะอนุญาตให้ไคลเอ็นต์ และเซิร์ฟเวอร์ติดต่อกันถ้าทั้งสองไม่ได้อยู่บนลิงก์เดียวกัน

หัวข้อนี้อธิบาย handshake การแลกเปลี่ยนสี่ข้อความสำหรับอินเตอร์เฟส เดียวที่มีหนึ่ง IA_NA และหนึ่งแอดเดรสสำหรับ IA_NA นี้ เพื่อให้ได้รับ IP แอดเดรส **DHCP** client daemon (**dhcpcd6**) ส่งข้อความ SOLICIT ให้กับแอดเดรส **All_DHCP_Relay_Agents_and_Servers** ซึ่งได้รับโดยเซิร์ฟเวอร์และประมวลผล (สามารถกำหนดคอนฟิกหลายเซิร์ฟเวอร์บนเครือข่ายสำหรับการทำซ้ำได้) ถ้ามีแอดเดรสว่างสำหรับไคลเอ็นต์ ข้อความ ADVERTISE ถูกสร้างและส่งกลับไคลเอ็นต์ ข้อความนี้มี IP แอดเดรสและอ็อพชันอื่นที่เหมาะสมสำหรับไคลเอ็นต์ ไคลเอ็นต์ได้รับข้อความ DHCP ADVERTISE ของเซิร์ฟเวอร์และจัดเก็บไว้ขณะรอการแจ้งอื่นๆ เมื่อไคลเอ็นต์ เลือกการแสดงที่ดีที่สุด ไคลเอ็นต์จะส่ง DHCP REQUEST ไปที่ แอดเดรส **All_DHCP_Relay_Agents_and_Servers** ระบุ การแจ้งเซิร์ฟเวอร์ที่ต้องการ

เซิร์ฟเวอร์ **DHCP** ที่กำหนดคอนฟิกทั้งหมดได้รับข้อความ REQUEST แต่ละเซิร์ฟเวอร์จะตรวจสอบเพื่อ ดูว่าเป็นเซิร์ฟเวอร์ที่ร้องขอหรือไม่ เซิร์ฟเวอร์ไม่ประมวลผลแพ็กเก็ตใดๆ ที่มีเซิร์ฟเวอร์ DUID ที่ไม่ตรงกับของตัวเอง เซิร์ฟเวอร์ที่ร้องขอจะทำเครื่องหมาย แอดเดรสเป็นกำหนดแล้ว และคืนค่า DHCP REPLY ตามเวลา ธุรกรรมเสร็จสมบูรณ์ ไคลเอ็นต์มีแอดเดรสสำหรับช่วงเวลา (valid-lifetime) กำหนดโดยเซิร์ฟเวอร์

เมื่อหมดเวลา preferred-lifetime สำหรับแอดเดรส ไคลเอ็นต์ส่งเซิร์ฟเวอร์ แพ็กเก็ต RENEW เพื่อขยายเวลาเช่า หากเซิร์ฟเวอร์เต็มใจจะต่ออายุ เซิร์ฟเวอร์จะส่ง DHCP REPLY หาก ไคลเอ็นต์ไม่ได้รับการตอบสนองจากเซิร์ฟเวอร์ที่เป็นเจ้าของแอดเดรสปัจจุบัน ตัวอย่างเช่น ซึ่งมัลติคาสต์แพ็กเก็ต DHCP REBIND เซิร์ฟเวอร์ ถูกย้ายจากเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่ง หากไคลเอ็นต์ยังไม่ต่ออายุ แอดเดรสหลัง valid-lifetime แอดเดรสจะถูกลบออกจากอินเตอร์เฟส และเริ่มโปรเซสใหม่ วงจรนี้ช่วยป้องกันไม่ให้หลายไคลเอ็นต์ บนเครือข่ายได้รับการกำหนดแอดเดรสเดียวกัน

ไคลเอ็นต์สามารถมีหลายอ็อพชัน IA_NA และแต่ละ IA_NA สามารถมีหลายแอดเดรส ไคลเอ็นต์สามารถหลายอ็อพชัน IA_TA และแต่ละอ็อพชัน สามารถมีหลายแอดเดรส:

- **การเชื่อมโยงเอกลักษณ์สำหรับแอดเดรสที่ไม่ใช่ชั่วคราว (IA_NA)**: IA ที่กำหนดแอดเดรสที่ไม่ใช่แอดเดรสชั่วคราว
- **การเชื่อมโยงเอกลักษณ์สำหรับแอดเดรสชั่วคราว (IA_TA)**: IA ที่มีแอดเดรสชั่วคราว (โปรดดู RFC 3041)
- **DUID**: **DHCP**-ตัวบ่งชี้จำเพาะสำหรับผู้เข้าร่วม **DHCP** แต่ละไคลเอ็นต์และเซิร์ฟเวอร์ของ **DHCP** มี DUID จำเพาะที่ยังเหมือนเดิม หลังรีบูต

เซิร์ฟเวอร์ **DHCP** กำหนดแอดเดรสตามข้อมูลคีย์ คีย์ทั่วไปสีรายการคือ **คลาส, ผู้ผลิต, ID ไคลเอ็นต์** และ **อินอ็อพชัน** เซิร์ฟเวอร์ใช้ คีย์เหล่านี้จัดสรรแอดเดรส และชุดของอ็อพชันคอนฟิกูเรชันเพื่อคืนค่า ไปยังไคลเอ็นต์

คลาส คีย์ **class** กำหนดคอนฟิกไคลเอ็นต์ได้ สามารถระบุแอดเดรสและอ็อพชัน สามารถใช้คีย์นี้เพื่อแสดงฟังก์ชันเครื่องในเครือข่าย หรือเพื่ออธิบาย วิธีการจัดกลุ่มเครื่องสำหรับวัตถุประสงค์การจัดการ ตัวอย่างเช่น ผู้ควบคุมระบบเครือข่ายอาจต้องการสร้างคลาส NetBIOS ที่มีอ็อพชันสำหรับไคลเอ็นต์ NetBIOS หรือคลาสบัญชี แสดงเครื่องของแผนกบัญชีที่จำเป็นต้องเข้าถึงเครื่องพิมพ์

vendor คีย์ **vendor** ช่วยจำแนกไคลเอ็นต์ตาม แพล็ตฟอร์มของฮาร์ดแวร์และซอฟต์แวร์

ID ไคลเอ็นต์

คีย์ **ID ไคลเอ็นต์** จำแนกไคลเอ็นต์ผ่าน DUID ID ไคลเอ็นต์ถูกระบุในไฟล์ duid ของ **dhcpcd** daemon นอกจากนี้ เซิร์ฟเวอร์สามารถใช้ ID ไคลเอ็นต์ เพื่อส่งผ่านอ็อพชันไปยังไคลเอ็นต์เฉพาะ หรือห้ามไคลเอ็นต์เฉพาะ ไม่ให้ได้รับพารามิเตอร์ใดๆ

Inoption

คีย์ **inoption** จำแนกไคลเอ็นต์โดย อ็อพชันร้องขอโดยอ็อพชัน

คีย์เหล่านี้สามารถใช้อย่างเดียวหรือใช้ร่วมกับคีย์อื่นก็ได้ หากไคลเอ็นต์ระบุหลายคีย์และสามารถกำหนดได้หลายแอดเดรส จะมีการเลือกเพียงคีย์เดียว และได้รับชุดอ็อพชันมาจากคีย์แรก ที่เลือก

รีเลย์เอเจนต์จำเป็นสำหรับเริ่มมัลติคาสต์จากไคลเอ็นต์สามารถ ออกจากเครือข่ายโลคัล รีเลย์เอเจนต์ทำหน้าที่ส่งต่อเอเจนต์สำหรับแพ็กเก็ต **DHCP**

เซิร์ฟเวอร์ DHCPv6

มีสามคอมโพเนนต์หลักของเซิร์ฟเวอร์ **DHCPv6**

เซิร์ฟเวอร์ **DHCP** ถูกแบ่งเซกเมนต์เป็นสามคอมโพเนนต์หลัก: ฐานข้อมูล, โปรโตคอลเอ็นจิ้น และชุดเธรดเซอร์วิส โดยแต่ละคอมโพเนนต์ จะมีข้อมูลคอนฟิกูเรชันของตัวเอง

ฐานข้อมูล DHCPv6:

ฐานข้อมูล `db_filev6.dhcpo` ถูกใช้ติดตาม ไคลเอ็นต์และแอดเดรส และการควบคุมการเข้าถึง

อ็อพชันยังมีการจัดเก็บไว้ในฐานข้อมูลสำหรับการดึงข้อมูลและการจัดส่งไปยังไคลเอ็นต์ ฐานข้อมูลถูกอิมพลีเมนต์เป็นอ็อบเจ็กต์ที่สามารถโหลดได้แบบไดนามิก

การใช้ข้อมูลในไฟล์คอนฟิกูเรชัน ฐานข้อมูลถูกเตรียมพร้อม และตรวจสอบความสอดคล้อง ฐานข้อมูลมีพูลอ็อพชัน และแอดเดรส

ไฟล์หน่วยเก็บข้อมูลหลักและสำรองเป็นไฟล์ ASCII รูปแบบ สำหรับไฟล์หน่วยเก็บข้อมูลหลักของฐานข้อมูลคือ:

หมายเหตุ: ห้ามแก้ไขไฟล์เหล่านี้ ด้วยตนเอง

```
DB6-1.0
Client-Info {
duid 1-0006085b68e20004ace491d3
state 7
authinfo {
    protocol 2
    algorithm 1
    rdm 0
    replay 1206567640
}
  Interface 0 {
    Inoptions {
    interface-id "en1"
    policies 2
    maxopcode 16
    numiana 1
      Ianalist {
      option 3 40 000000010000003200000050000050018deaddeadaaaaaaaa0000000000000006000000064000000c8
      }
    numiata 0
      Optiontable {
```

```
      option 6 10 00030004001700180237
      option 8 2 e659
      option 15 14 000369626d000373756e00026870
      option 16 18 000004d2000730783131313131000369626d
      }
    }
    Ianarec {
    IAID 1
    t1 50
    t2 80
      Addrec {
      Address dead:dead:aaaa:aaaa::6
      state 3
      starttime  1087592918
      preferred-lifetime 100
      valid-lifetime 200
      }
    }
  }
}
```

บรรทัดแรกคือตัวบ่งชี้เวอร์ชันสำหรับไฟล์: DB6-1.0 บรรทัดต่อมาเป็นบรรทัดนิยามไคลเอ็นต์เร็กคอร์ด เซิร์ฟเวอร์อ่าน ตั้งแต่บรรทัดที่สองไปถึงตอนท้ายของไฟล์ (พารามิเตอร์ในอัญประกาศต้อง ถูกใส่ไว้ในอัญประกาศ)

duid ID ซึ่งไคลเอ็นต์ใช้เพื่อแสดงแทนตัวเองที่เซิร์ฟเวอร์

อินเตอร์เฟส
ไคลเอ็นต์สามารถมีหลายอินเตอร์เฟส ถ้าไคลเอ็นต์มีอินเตอร์เฟสเดียว และสร้างข้อความ **SOLICIT** เฉพาะสำหรับแต่ละ IA_NA หรือ IA_TA, ไฟล์จะมีหลายอินเตอร์เฟสสำหรับไคลเอ็นต์นี้

Inoptions
อ็อพชันขาเข้าจากไคลเอ็นต์

policies กำหนดแฟล็กเพื่อจำแนก unicast, reconfig-option และ rapid-commit

maxopcode
โค้ดอ็อพชันที่ใหญ่ที่สุด

numiana
จำนวนของ IA_NAs สำหรับอินเตอร์เฟสนี้

Ianalist รายการของอ็อพชันขาเข้า IA_NA จากไคลเอ็นต์

numiata
จำนวนของ IA_TAs สำหรับอินเตอร์เฟสนี้

Optiontable
รายการของอ็อพชันที่ร้องขอโดยไคลเอ็นต์ไม่รวมอ็อพชัน IA_NA และ IA_TA

Ianarec
เร็กคอร์ดคอนเทนเนอร์ IA_NA ที่บันทึกจากเซิร์ฟเวอร์ฐานข้อมูล

IAID ID ของ IA_NA

t1 เปอร์เซ็นต์ preferred-lifetime สำหรับ IA_NA นี้

t2 เปอร์เซ็นต์ valid-lifetime สำหรับ IA_NA นี้

Addrec คอนเทนเนอร์เร็กคอร์ดของแอดเดรสจากเซิร์ฟเวอร์ฐานข้อมูล

Address

แอดเดรสที่กำหนดให้ไคลเอ็นต์สำหรับเร็กคอร์ดของแอดเดรส

state สถานะปัจจุบันของไคลเอ็นต์ **DHCP** โปรโตคอลเอ็นจิ้นมี ชุดที่ใช้ได้ และมีการเก็บรักษาสถานะไว้ในฐานข้อมูล **DHCP** ตัวเลขถัดจาก **state** แสดงถึงค่า สถานะสามารถเป็น:

(1) FREE

แสดงถึงแอดเดรสที่พร้อมใช้งาน โดยทั่วไป ไคลเอ็นต์ไม่มี สถานะนี้ยกเว้นว่าไคลเอ็นต์ไม่ได้รับการกำหนดแอดเดรส คำสั่ง **dadmin** และ **lssrc** รายงานสถานะนี้เป็น `Free`

(2) BOUND

บ่งชี้ว่าไคลเอ็นต์และแอดเดรสถูกโยงเข้าด้วยกัน และไคลเอ็นต์ได้รับการกำหนด แอดเดรสนี้ในช่วงเวลาหนึ่ง คำสั่ง **dadmin** และ **lssrc** รายงานสถานะนี้เป็น `Leased`

(3) EXPIRED

บ่งชี้ว่าไคลเอ็นต์และแอดเดรสถูกโยงเข้าด้วยกัน แต่ใช้เพื่อเป็นข้อมูลเท่านั้น ในลักษณะคล้ายกับแอดเดรสที่เช่า อย่างไรก็ตาม สถานะหมดอายุ แสดงถึงไคลเอ็นต์ที่ปล่อยให้การเช่าหมดอายุ แอดเดรสที่หมดอายุสามารถนำมาใช้ได้ และกำหนดใหม่หลังจากแอดเดรสที่ว่างทั้งหมดไม่มีอยู่ และก่อนจะกำหนด แอดเดรสที่เช่าอีกครั้ง คำสั่ง **dadmin** และ **lssrc** รายงานสถานะนี้เป็น `Expired`

(4) RELEASED

บ่งชี้ว่าไคลเอ็นต์และแอดเดรสถูกโยงเข้าด้วยกันเพื่อใช้เป็นข้อมูลเท่านั้น โปรโตคอล **DHCP** แนะนำให้เซิร์ฟเวอร์ **DHCP** เก็บรักษาข้อมูล เกี่ยวกับไคลเอ็นต์ที่ให้บริการไปแล้วเพื่อใช้อ้างอิงในอนาคต (โดยส่วนใหญ่ พยายาม กำหนดแอดเดรสเดียวกันให้กับไคลเอ็นต์ซึ่งเคยได้รับการกำหนดแอดเดรสนั้นใน อดีต) สถานะนี้บ่งชี้ว่าไคลเอ็นต์รีลีสแอดเดรสแล้ว แอดเดรสนั้นพร้อม ให้ไคลเอ็นต์อื่นใช้งานได้ ถ้าไม่มีแอดเดรสอื่น คำสั่ง **dadmin** และ **lssrc** รายงานสถานะนี้เป็น `Released`

(5) RESERVED

บ่งชี้ว่าไคลเอ็นต์และแอดเดรสถูกโยงเข้าด้วยกัน แต่ในแบบหลวมๆ ไคลเอ็นต์ออกใช้ข้อความค้นหา **DHCP** และเซิร์ฟเวอร์ **DHCP** ตอบกลับแล้ว แต่ ไคลเอ็นต์ยังไม่ได้ตอบกลับด้วยคำร้องขอ **DHCP** สำหรับแอดเดรสนั้น คำสั่ง **dadmin** และ **lssrc** รายงานสถานะนี้เป็น `Reserved`

(6) BAD

แสดงถึงแอดเดรสที่ถูกใช้อยู่ในเครือข่ายแต่ยังไม่ได้ส่ง โดยเซิร์ฟเวอร์ **DHCP** สถานะนี้ยังแสดงถึงแอดเดรสที่ไคลเอ็นต์ ปฏิเสธด้วย สถานะนี้ไม่ได้ใช้กับไคลเอ็นต์ คำสั่ง **dadmin** รายงานสถานะนี้ เป็น `Used` และคำสั่ง **lssrc** รายงานสถานะนี้เป็น `Bad`

Starttime

เวลาที่แอดเดรสนี้ถูกส่งออก แสดงเป็นวินาที ตั้งแต่ January 1, 2000

preferred-lifetime

จำนวนเป็นวินาทีก่อนที่แอดเดรสนี้จำเป็นต้องเรียกใช้ใหม่

valid-lifetime
จำนวนเป็นวินาทีก่อนที่แอดเดรสนี้จะไม่ถูกต้อง และไม่สามารถใช้ได้

protocol
โปรโตคอลการพิสูจน์ตัวตนที่ไคลเอ็นต์ใช้:

(1) DELAYED
ไคลเอ็นต์ใช้การพิสูจน์ตัวตนแบบดีเลย์

(2) RECONFIGURE KEY
ไคลเอ็นต์ใช้การพิสูจน์ตัวตนแบบ reconfigure key

algorithm
อัลกอริทึมของการพิสูจน์ตัวตนที่ไคลเอ็นต์ใช้:

(1) HMAC-MD5
ไคลเอ็นต์ใช้อัลกอริทึม keyed MD5 เพื่อสร้างข้อความย่อ

rdm วิธีตรวจจับรีเพลย์ที่ไคลเอ็นต์ใช้:

(0) Monotonically increasing counter
ไคลเอ็นต์ใช้ตัวนับเพิ่มแบบ monotonically เพื่อแก้ไข ค่ารีเพลย์

replay ค่าปัจจุบันของฟิลด์รีเพลย์

ไม่มีการระบุไวยากรณ์สำหรับไฟล์ checkpoint หากเซิร์ฟเวอร์เสียหายหรือคุณต้องปิดและไม่สามารถทำการปิดฐานข้อมูลตามปกติได้ เซิร์ฟเวอร์สามารถประมวลผลไฟล์ checkpoint และสำเนาสำรอง เพื่อสร้างฐานข้อมูลที่ถูกต้องขึ้นใหม่ ไคลเอ็นต์ไม่ถูกเขียนลงไฟล์ checkpoint เมื่อเซิร์ฟเวอร์หยุดทำงานหายไป ในตอนนี้ ไม่มีการบันทึกเป็นระยะๆ เมื่อไคลเอ็นต์ประมวลผล ไฟล์ดีฟอลต์คือ:

`/etc/dhcpv6/db_file6.cr`
การดำเนินการฐานข้อมูลปกติ

`/etc/dhcpv6/db_file6.crbk`
สำรองฐานข้อมูล

การดำเนินการเธรด DHCP:

ชิ้นสุดท้ายของเซิร์ฟเวอร์ **DHCP** ตั้งค่าการดำเนินการ ที่ใช้เพื่อทำให้ระบบทำงาน

เนื่องจากเซิร์ฟเวอร์ **DHCP** เป็นเธรด การดำเนินการเหล่านี้ ตั้งค่าเป็นเธรดเพื่อให้ทุกอย่างทำงานร่วมกัน

เธรดหลัก
เธรดนี้จัดการสัญญาณ ตัวอย่างเช่น,

- SIGHUP (-1) ทำให้รีเฟรชฐานข้อมูลทั้งหมดในไฟล์คอนฟิกูเรชัน
- A SIGTERM (-15) will ทำให้เซิร์ฟเวอร์หยุดแบบปกติ
- A SIGUSR1 (-30) จะทำให้เซิร์ฟเวอร์ทำการดัมพ์ฐานข้อมูลคอนฟิกูเรชัน

เธรด src
เธรดนี้จัดการการร้องขอ SRC (เช่น **startsrc, stopsrc, lssrc, traceson,** และ **refresh**)

เธรด dadmin

เธรดนี้ติดต่อกับโปรแกรมไคลเอ็นต์ **dadmin** และเซิร์ฟเวอร์ **DHCP** เครื่องมือ **dadmin** สามารถใช้เพื่อรับสถานะเช่นเดียวกับการแก้ไขฐานข้อมูลเพื่อหลีกเลี่ยง การแก้ไขไฟล์ฐานข้อมูลด้วยตนเอง ด้วยการเพิ่ม เธรด **dadmin** และ **src** เซิร์ฟเวอร์สามารถจัดการกับคำร้องขอเซอร์วิสและยังคงจัดการกับคำร้องขอไคลเอ็นต์ได้

เธรด garbage

เธรดนี้รันตัวจับเวลาที่จะทำความสะอาดฐานข้อมูลเป็นระยะ, บันทึกฐานข้อมูล, ล้างข้อมูลไคลเอ็นต์ที่ไม่มีแอดเดรสและลบแอดเดรส ที่สงวนไว้ซึ่งอยู่ในสถานะสงวนไว้เป็นเวลานานเกินไป ตัวจับเวลาทั้งหมดนี้ สามารถกำหนดคอนฟิกได้

packet processors

แต่ละแพ็กเก็ตเหล่านี้สามารถจัดการการร้องขอจากไคลเอ็นต์ **DHCPv6** จำนวนแพ็กเก็ตตัวประมวลผลต้องการโหลดและขึ้นกับเครื่อง จำนวนของเธรดเหล่านี้สามารถกำหนดคอนฟิกได้ ค่าดีฟอลต์คือ 1 จำนวนสูงสุด ของแพ็กเก็ตเธรดคือ 50

เธรด logging

ในระบบที่ปริมาณข้อมูลที่ถูกบันทึกลงไฟล์ล็อกมีความสำคัญ จำนวนเธรดการล็อกสามารถเพิ่มขึ้นมากกว่า ค่าดีฟอลต์ (1) ถึงสูงสุด (50)

เธรด table manager

เธรดนี้ให้แน่ใจว่า **dhcpsdv6** daemon ไม่ประมวลผลแพ็กเก็ตที่ซ้ำ

เธรด process

เธรดเหล่านี้ประมวลผลไคลเอ็นต์แพ็กเก็ต **DHCPv6**

เธรด reconfigure

เธรดนี้จัดการกำหนดคอนฟิกใหม่เมื่อเซิร์ฟเวอร์รีเฟรช (เช่น ด้วยคำสั่ง `dadmin -x 6 -i`)

คอนฟิกูเรชัน DHCPv6

ตามค่าดีฟอลต์ เซิร์ฟเวอร์ **DHCP** ถูกกำหนดคอนฟิกโดยอ่านไฟล์ `/etc/dhcpv6/dhcpsdv6.cnf` ซึ่งระบุฐานข้อมูลเริ่มต้นของอ็อพชันและแอดเดรส

เซิร์ฟเวอร์เริ่มต้นจากคำสั่ง SRC ถ้า **dhcpsdv6** ตั้งใจ ว่าจะเริ่มต้นข้ามการรีบูต ให้เพิ่มรายการลงในไฟล์ `/etc/rc.tcpip`

กำหนดคอนฟิกเซิร์ฟเวอร์ **DHCP** คือส่วนที่ยากที่สุดในการใช้งาน **DHCP** ในเครือข่ายของคุณ ขั้นแรก ให้ตัดสินใจว่าไคลเอ็นต์ใดที่คุณต้องการให้มีไคลเอ็นต์ **DHCP** แต่ละซับเน็ตในเครือข่ายของคุณแสดงพูลของแอดเดรสที่เซิร์ฟเวอร์ **DHCP** ต้องเพิ่มลงในฐานข้อมูล ตัวอย่างเช่น:

```
subnet dead:dead:aaaa:: 48 {
        option 23 dead::beef beef:aaaa::bbbb:c aaaa:bbbb::cccc  #nameserver list
        option 24 austin.ibm.com ibm.com                        # domain list
 }
```

ตัวอย่างข้างต้นแสดงซับเน็ต `dead:dead:aaaa::`, ด้วย คำนำหน้า 48 บิต แอดเดรสทั้งหมดในซับเน็ตนี้ `dead:dead:aaaa::1` ถึง `dead:dead:aaaa:ffff:ffff:ffff:ffff:ff7f` อยู่ในพูล อีกวิธีที่คือ ระบุช่วงไว้ที่สิ้นสุดบรรทัดก่อน '`{`' หรือช่วง หรือคำสั่ง exclude สามารถรวมไว้ใน ซับเน็ตคอนเทนเนอร์

บรรทัดความคิดเห็นจะขึ้นต้นด้วย # (เครื่องหมายปอนด์) ข้อความหลัง # จนสุดบรรทัด จะถูกละเว้นโดยเซิร์ฟเวอร์ **DHCP** แต่ละบรรทัดอ็อพชันจะถูกใช้โดยเซิร์ฟเวอร์เพื่อบอกให้ไคลเอ็นต์ทำงาน

ถ้าเซิร์ฟเวอร์ไม่เข้าใจวิธีการวิเคราะห์อ็อพชัน เซิร์ฟเวอร์จะใช้วิธีการดีฟอลต์ เพื่อส่งอ็อพชันให้แก่ไคลเอ็นต์ ซึ่งอนุญาตให้เซิร์ฟเวอร์ **DHCP** ส่งอ็อพชัน site-specific ที่ไม่กำหนด RFC แต่อาจถูกใช้โดยบางไคลเอ็นต์ หรือไคลเอ็นต์คอนฟิกูเรชัน

ไฟล์คอนฟิกูเรชัน DHCPv6:

ไฟล์คอนฟิกูเรชันมีส่วนแอดเดรส และส่วนนิยามอ็อพชัน ส่วนเหล่านี้ใช้คอนเทนเนอร์เพื่อเก็บอ็อพชัน โมดิฟายเออร์ และคอนเทนเนอร์อื่น

คอนเทนเนอร์ (วิธีจัดกลุ่มอ็อพชัน) ใช้ตัวบ่งชี้เพื่อจำแนก ไคลเอ็นต์เป็นกลุ่ม ชนิดคอนเทนเนอร์ได้แก่ **ซับเน็ต, คลาส, ผู้ผลิต, อินอ็อพชัน** และ **ไคลเอ็นต์** ในปัจจุบัน ยังไม่มีคอนเทนเนอร์ที่ผู้ใช้นิยามได้เอง ตัวบ่งชี้จำเพาะกำหนดไคลเอ็นต์ เพื่อให้สามารถติดตามไคลเอ็นต์ได้ เช่น ไคลเอ็นต์ย้ายระหว่างซับเน็ต สามารถใช้ชนิดคอนเทนเนอร์มากกว่าหนึ่งชนิดเพื่อกำหนดการเข้าถึงไคลเอ็นต์

อ็อพชันคือตัวบ่งชี้ที่คืนค่าให้ไคลเอ็นต์ เช่น แอดเดรส DNS หรือชื่อโดเมน

หลังจากเลือกโมดิฟายเออร์ ไอเท็มถัดไปเพื่อตั้งค่าการล็อก พารามิเตอร์การล็อก ถูกระบุในคอนเทนเนอร์อย่างฐานข้อมูล แต่คีย์เวิร์ดคอนเทนเนอร์ คือ **logging_info** เมื่อศึกษาการกำหนดคอนฟิก **DHCP** ขอแนะนำให้ปรับการล็อกเป็นระดับสูงสุด นอกจากนี้ ควรอย่างยิ่ง ที่จะระบุคอนฟิกูเรชันการล็อกก่อนข้อมูลของไฟล์คอนฟิกูเรชันอื่น เพื่อให้แน่ใจว่าข้อผิดพลาดของคอนฟิกูเรชันถูกล็อกหลังจากที่ระบบย่อยการล็อก เริ่มทำงาน ใช้คีย์เวิร์ด **logitem** เพื่อเปิดระดับการล็อก หรือลบคีย์เวิร์ด **logitem** เพื่อปิดระดับการล็อก คีย์เวิร์ดอื่นสำหรับการล็อกอนุญาตให้ระบุชื่อไฟล์ของล็อก, ขนาดไฟล์ และจำนวนไฟล์ล็อกเวียน

คอนเทนเนอร์ DHCPv6:

เมื่อเซิร์ฟเวอร์ **DHCP** ได้รับการร้องขอ แพ็กเก็ตถูกวิเคราะห์ และจำแนกคีย์กำหนดคอนเทนเนอร์ อ็อพชัน และแอดเดรสถูกดึงออก

แต่ละชนิดของคอนเทนเนอร์ใช้อ็อพชันแตกต่างกันเพื่อแยกแยะไคลเอ็นต์:

- คอนเทนเนอร์ **subnet** ใช้ฟิลด์ **hintlist** หรืออินเตอร์เฟสแอดเดรสของการรับอินเตอร์เฟสเพื่อกำหนดจาก ซับเน็ตที่ไคลเอ็นต์เป็นเจ้าของ
- คอนเทนเนอร์ **class** ใช้ค่าในอ็อพชัน 15 (OPTION_USER_CLASS Identifier)
- **vendor** ใช้ค่าอ็อพชัน 16 (OPTION_VENDOR_CLASS)
- คอนเทนเนอร์ **client** ใช้อ็อพชัน 1 (OPTION_CLIENTID) จาก DUID ของไคลเอ็นต์ DHCP
- คอนเทนเนอร์ **inoption** ตรงกับอ็อพชันตอบสนองของไคลเอ็นต์

ยกเว้นสำหรับซับเน็ต แต่ละคอนเทนเนอร์อนุญาตให้ระบุค่าที่ตรงกับมัน รวมถึงการจับคู่นิพจน์ปกติ

อีกทั้งยังมีคอนเทนเนอร์โดยนัย, โกลบอลคอนเทนเนอร์ อ็อพชันและโมดิฟายเออร์ อยู่ในโกลบอลคอนเทนเนอร์จนกว่าจะถูกลบล้างหรือปฏิเสธ คอนเทนเนอร์ส่วนใหญ่สามารถอยู่ในคอนเทนเนอร์อื่นเพื่อแสดงขอบเขตการมองเห็น คอนเทนเนอร์อาจมีหรือไม่มีช่วงแอดเดรสที่เกี่ยวข้องกับคอนเทนเนอร์ ซับเน็ตโดยธรรมชาติจะมีช่วงที่สัมพันธ์กับซับเน็ต

กฎพื้นฐานสำหรับคอนเทนเนอร์และคอนเทนเนอร์ย่อยคือ:

- เฉพาะซับเน็ตคอนเทนเนอร์ใช้งานได้ที่ระดับโกลบอล
- ซับเน็ตไม่สามารถอยู่ในคอนเทนเนอร์อื่น รวมถึงซับเน็ตเอง

- คอนเทนเนอร์ที่จำกัดไม่สามารถมีคอนเทนเนอร์ปกติที่มีชนิดเดียวกันอยู่ภายในได้ (ตัวอย่าง เช่น คอนเทนเนอร์ที่มีอ็อพชันที่อนุญาตให้เฉพาะคลาสของบัญชี จะไม่สามารถรวมอ็อพชันที่อนุญาตให้คลาสทั้งหมด ที่ขึ้นต้นด้วยอักษร a)
- ไคลเอ็นต์คอนเทนเนอร์ที่จำกัดไม่สามารถมีคอนเทนเนอร์ย่อยได้
- อินอ็อพชันคอนเทนเนอร์ไม่สามารถมีคอนเทนเนอร์ย่อยได้

กฎที่กำหนดข้างต้น คุณสามารถสร้างลำดับขั้นของคอนเทนเนอร์ ที่แบ่งเซกเมนต์อ็อพชันของคุณเป็นกลุ่มสำหรับไคลเอ็นต์เฉพาะ หรือชุดของไคลเอ็นต์

ถ้าไคลเอ็นต์ตรงกับหลายคอนเทนเนอร์ เซิร์ฟเวอร์ **DHCP** จะส่งผ่าน การร้องขอไปที่ฐานข้อมูล และรายการคอนเทนเนอร์จะถูกสร้างขึ้น ซึ่งรายการจะแสดง ตามลำดับความลึกและลำดับความสำคัญ ลำดับความสำคัญถูกกำหนดเป็นลำดับขั้นโดยนัยในคอนเทนเนอร์ คอนเทนเนอร์ที่เข้มงวดมีลำดับความสำคัญสูงกว่าคอนเทนเนอร์ปกติ ไคลเอ็นต์ คลาส ผู้ผลิต และซับเน็ตถูกจัดเรียง ตามลำดับดังกล่าว และภายในชนิดคอนเทนเนอร์ตามความลึก สิ่งนี้สร้างลำดับรายการตามที่ระบุไว้สูงสุด ไปยังต่ำสุด ตัวอย่างเช่น:

```
Subnet 1
    --Class 1
    --Client 1
Subnet 2
    --Class 1
    ----Vendor 1
    ----Client 1
    --Client 1
```

ตัวอย่างแสดงซับเน็ต 2 รายการคือ Subnet 1 และ Subnet 2 มีชื่อคลาสหนึ่งรายการคือ Class 1, ชื่อผู้ผลิตหนึ่งรายการ, Vendor 1 และชื่อไคลเอ็นต์หนึ่งรายการคือ Client 1 Class 1 และ Client 1 ถูกกำหนดไว้ในหลายที่ เนื่องจาก รายการแตกต่างกันในคอนเทนเนอร์ ชื่อรายการสามารถเหมือนกัน แต่ค่าภายในสามารถ แตกต่างกันได้ ถ้า Client 1 ส่งข้อความถึงเซิร์ฟเวอร์ **DHCP** จาก Subnet 1 ด้วย Class 1 ที่ระบุในรายการ อ็อพชัน เซิร์ฟเวอร์ **DHCP** ควรสร้างพาธคอนเทนเนอร์ต่อไปนี้:

```
Subnet 1, Class 1, Client 1
```

คอนเทนเนอร์ที่ระบุสำคัญสุดจะแสดงเป็นรายการสุดท้าย เมื่อต้องการขอรับแอดเดรส รายการจะถูกตรวจสอบ ตามลำดับขั้นย้อนกลับเพื่อค้นหาแอดเดรสแรกที่พร้อมใช้งาน จาก รายการจะตรวจสอบตามลำดับขั้นเพื่อรับอ็อพชัน อ็อพชันจะลบล้างค่าก่อนหน้า นอกจากการปฏิเสธอ็อพชันจะปรากฏในคอนเทนเนอร์ นอกจากนี้ เนื่องจาก Class 1 และ Client 1 อยู่ใน Subnet 1, รายการเรียงลำดับตามความสำคัญของคอนเทนเนอร์ ถ้าไคลเอ็นต์เดียวกันอยู่ใน Subnet 2 และส่งข้อความเดียวกัน รายการคอนเทนเนอร์ที่สร้างคือ:

```
Subnet 2, Class 1, Client 1 (ที่ระดับ Subnet 2), Client 1 (ที่ระดับ Class 1)
```

Subnet 2 แสดงเป็นอันดับแรก แล้ว Class 1, แล้ว Client 1 ที่ระดับ Subnet 2 (เนื่องจากคำสั่งของไคลเอ็นต์นี้เป็นระดับเดียวลงในลำดับขั้น) ลำดับขั้นโดยนัย ที่ไคลเอ็นต์ตรงกับคำสั่งแรกของไคลเอ็นต์ต่ำกว่าที่ระบุ การจับคู่ไคลเอ็นต์ Client 1 ของ Class 1 ภายใน Subnet 2

ลำดับความสำคัญที่เลือกตามความลึกภายในลำดับขั้นไม่ถูกแทนที่โดย ลำดับขั้นของคอนเทนเนอร์เอง ตัวอย่าง เช่น ถ้าไคลเอ็นต์เดียวกันใช้ข้อความ เดียวกัน และระบุตัวบ่งชี้ผู้ผลิต รายการคอนเทนเนอร์คือ:

```
Subnet 2, Class 1, Vendor 1, Client 1 (ที่ระดับ Subnet 2), Client 1 (ที่ระดับ Class 1)
```

ลำดับความสำคัญของคอนเทนเนอร์ช่วยปรับปรุงประสิทธิภาพการค้นหา เนื่องจากลำดับความสำคัญเป็นไปตาม แนวคิดทั่วไปที่ไคลเอ็นต์คอนเทนเนอร์สำคัญที่สุดระบุวิธีกำหนดหนึ่งไคลเอ็นต์หรือมากกว่า คลาสคอนเทนเนอร์มีแอดเดรสที่สำคัญน้อยกว่าไคลเอ็นต์คอนเทนเนอร์ ผู้ผลิตมีความสำคัญต่ำ และซับเน็ตมีความสำคัญต่ำ

แอดเดรสและช่วงแอดเดรสของ DHCPv6:

ชนิดคอนเทนเนอร์ใดๆ สามารถมีช่วยแอดเดรสที่เกี่ยวข้อง ซับเน็ต ต้องมีช่วงแอดเดรสที่เกี่ยวข้อง

แต่ละช่วงภายในคอนเทนเนอร์ต้องเป็นซับเน็ตของช่วง และต้องไม่ซ้อนทับ กับช่วงของคอนเทนเนอร์อื่น ตัวอย่างเช่น ถ้าคลาสถูกกำหนดภายในซับเน็ต และคลาสมีช่วง ช่วงนั้นต้องเป็นเซ็ตย่อยของช่วงซับเน็ต นอกจากนี้ ช่วงภายในคลาสคอนเทนเนอร์ไม่สามารถซ้อนทับกับ ช่วงอื่นที่ระดับของมัน

ช่วงสามารถแสดงบนบรรทัดคอนเทนเนอร์และแก้ไขตามช่วง และไม่รวมคำสั่งที่อนุญาตสำหรับแยกชุดแอดเดรสที่เกี่ยวข้องกับคอนเทนเนอร์ ถ้าคุณมีแอดเดรส 10 อันดับแรก และ 10 อันดับรองของซับเน็ตพร้อมใช้งานอยู่ ซับเน็ตสามารถระบุแอดเดรสเหล่านี้ตามช่วงในวลีซับเน็ตเพื่อลด ทั้งหน่วยความจำที่ใช้ และโอกาสที่แอดเดรสชนกันกับไคลเอ็นต์อื่น ที่ไม่อยู่ในช่วงที่ระบุ

หลังจากแอดเดรสถูกเลือก, คอนเทนเนอร์ที่ตามมา ในรายการที่มีช่วงแอดเดรสถูกลบออกจากรายการพร้อม รายการย่อยอ็อพชัน Network-specific ในคอนเทนเนอร์ที่ลบไม่ถูกต้อง ถ้าแอดเดรสไม่ได้ใช้งานจากภายในคอนเทนเนอร์

อ็อพชันไฟล์คอนฟิกูเรชัน DHCPv6:

หลังจากรายการถูกเลือกเพื่อกำหนดแอดเดรส ชุดของอ็อพชัน จะถูกสร้างขึ้นสำหรับไคลเอ็นต์

ในขั้นตอนการเลือกนี้ อ็อพชันจะเขียนทับอ็อพชันที่เลือกก่อนหน้า จนกว่าจะตรวจพบการปฏิเสธ ซึ่งในกรณีดังกล่าว อ็อพชันที่ปฏิเสธจะถูกลบ ออกจากรายการที่ส่งให้กับไคลเอ็นต์ วิธีนี้อนุญาตให้สืบทอดจาก คอนเทนเนอร์หลักเพื่อลดปริมาณข้อมูลที่ต้องระบุ

อ็อพชันเฉพาะเซิร์ฟเวอร์ของ DHCPv6:

ชุดสุดท้ายของพารามิเตอร์ที่ระบุอ็อพชันเฉพาะเซิร์ฟเวอร์ ที่อนุญาตให้ผู้ใช้ควบคุมจำนวนแพ็กเก็ตโปรเซสเซอร์ ความถี่ของเธรดรวบรวมขยะที่รัน และอื่นๆ

ตัวอย่าง อ็อพชันเฉพาะเซิร์ฟเวอร์สองรายการคือ:

reservedTime
: บ่งชี้ระยะเวลาที่แอดเดรสอยู่ในสถานะสงวนหลังจากการส่ง ADVERTISE ไปที่ไคลเอ็นต์ **DHCP**

reservedTimeInterval
: บ่งชี้ความถี่ที่เซิร์ฟเวอร์ **DHCP** สแกนผ่านแอดเดรส เพื่อดูว่ามีแอดเดรสใดที่อยู่ในสภาวะสำรองนานกว่า *reservedTime*

อ็อพชันเหล่านี้เป็นประโยชน์เมื่อคุณมีหลายไคลเอ็นต์ที่มัลติคาสต์ข้อความ SOLICIT และไม่ทำการมัลติคาสต์ข้อความ REQUEST หรือข้อความ REQUEST สูญหายในเครือข่าย การใช้พารามิเตอร์เหล่านี้เก็บแอดเดรส จากการถูกสงวนแบบไม่มีกำหนดสำหรับไคลเอ็นต์ที่ไม่ยินยอม

อ็อพชันที่มีประโยชน์อีกรายการคือ *SaveInterval*, ซึ่งบ่งชี้ความถี่ในการบันทึก

ไฟล์ /etc/dhcpv6/dhcpsdv6.cnf:

เซิร์ฟเวอร์ **DHCPv6** ถูกกำหนดคอนฟิกโดยแก้ไขไฟล์ /etc/dhcpv6/dhcpsdv6.cnf

คีย์เวิร์ดเป็นแบบตรงตัวพิมพ์ เมื่อ '{' ถูกแสดงรายการ ซึ่งต้องแสดง บนบรรทัดเดียวกับคีย์เวิร์ด ตัวอย่างไฟล์คอนฟิกูเรชัน สามารถพบได้ใน /usr/samples/tcpip/dhcpv6

ต่อไปนี้คือคำอธิบายของไฟล์ /etc/dhcpv6/dhcpsdv6.cnf กลุ่มบรรทัดต่อไปนี้อนุญาตในไฟล์นี้:

- การบันทึกการทำงาน
- โกลบอลคีย์เวิร์ด
- คำสั่งคอนเทนเนอร์ที่ไม่ซ้อน
- คำสั่งคอนเทนเนอร์ที่ซ้อน
- อ็อพชัน
- อ็อพชันทั่วไป

การล็อก DHCPv6:

คีย์เวิร์ดเซิร์ฟเวอร์ **DHCPv6** ที่นี่สำหรับ รายการในกลุ่มบรรทัดการล็อก

กลุ่มบรรทัดนี้ไม่จำเป็นต้องใช้ แต่ถ้ามีอยู่ ต้องอยู่ที่ด้านบน ของไฟล์คอนฟิกูเรชัน โดยมีรูปแบบดังต่อไปนี้:

```
logging_info { log_options }
```

ค่า *log_options* สามารถเป็นได้ดังนี้:

ตารางที่ 65. คีย์เวิร์ด ค่า และคำอธิบายสำหรับรายการ ในกลุ่มบรรทัดการล็อก

คีย์เวิร์ด	ค่า	คำอธิบาย
logFileSize	*num*	ระบุขนาดของไฟล์ล็อก ค่า *num* คือขนาดสูงสุดของไฟล์ล็อกเป็นกิโลไบต์ ไฟล์ล็อกจะถูกเปลี่ยน หลังจากเกินขนาดไฟล์นี้ ให้ใช้ขนาดไม่จำกัดถ้า logFileSize ไม่ระบุ ค่าไว้
logFileName	"*filename*"	ระบุชื่อของไฟล์ล็อก ค่า *filename* จะเป็นชื่อของไฟล์ล็อก ชื่อไฟล์และตำแหน่งดีฟอลต์คือ /var/tmp/dhcpsdv6.log
numLogFiles	*num*	ระบุจำนวนไฟล์ล็อกสำหรับเวียน ดีฟอลต์คือ 0

ตารางที่ 65. คีย์เวิร์ด ค่า และคำอธิบายสำหรับรายการ ในกลุ่มบรรทัดการล็อก (ต่อ)

คีย์เวิร์ด	ค่า	คำอธิบาย
logItem	*type*	ระบุชนิดของล็อกที่ต้องการ ค่าเหล่านี้เป็นค่าที่ถูกต้อง : **SYSERR** ข้อผิดพลาดระบบ ที่อินเตอร์เฟสไปยังแพล็ตฟอร์ม **OBJERR** ข้อผิดพลาดอ็อบเจ็กต์ ระหว่างอ็อบเจ็กต์ในโปรเซส **PROTERR** ข้อผิดพลาดโปรโตคอล ระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ **WARNING** คำเตือน เรียกความสนใจจากผู้ใช้ **EVENT** เหตุการณ์ที่เกิดขึ้นกับโปรเซส **ACTION** เกิดการดำเนินการโดยโปรเซส **INFO** ข้อมูลที่อาจเป็นประโยชน์ **ACNTING** บุคคลที่ทำหน้าที่เมื่อ **TRACE** การไหลของโค้ด สำหรับดีบัก

คีย์เวิร์ดโกลบอล DHCPv6:

ค่าคีย์เวิร์ดถูกอธิบายที่นี่สำหรับรายการใน กลุ่มบรรทัดคีย์เวิร์ดโกลบอล

คีย์เวิร์ดโกลบอลสามารถใช้ได้เฉพาะนอกคอนเทนเนอร์ ค่าต่อไปนี้สามารถใช้ได้:

ตารางที่ 66. คีย์เวิร์ด ค่า และคำอธิบายสำหรับรายการในกลุ่มบรรทัดคีย์เวิร์ดโกลบอล

คีย์เวิร์ด	ค่า	คำอธิบาย
UsedIpAddressExpiredInterval	num [*units*]	ระบุความถี่แอดเดรสที่วางในสถานะ BAD ถูกชดเชยและทดสอบใหม่เพื่อความถูกต้อง ถ้าไม่ได้ตั้งค่าหน่วย ระบบดีฟอลต์ถูกตั้งค่าเป็นวินาที ค่าดีฟอลต์คือ -1
leaseExpiredInterval	num [*units*]	ระบุความถี่ของแอดเดรสในสถานะ BOUND ถูกตรวจสอบเพื่อดูว่าหมดอายุหรือไม่ ถ้าแอดเดรสหมดอายุ สถานะจะเปลี่ยนเป็น EXPIRED ถ้าไม่ได้ตั้งค่าหน่วย ระบบดีฟอลต์ถูกตั้งค่าเป็นวินาที ค่าดีฟอลต์คือ 900 วินาที
reservedTime	num [*units*]	ระบุระยะเวลาที่แอดเดรสควรอยู่ในสถานะ RESERVED ก่อนชดเชยในสถานะ FREE ถ้าไม่ได้ตั้งค่าหน่วย ระบบดีฟอลต์ถูกตั้งค่าเป็นวินาที ค่าดีฟอลต์คือ -1
reservedTimeInterval	num [*units*]	ระบุความถี่ของแอดเดรสในสถานะ RESERVE ถูกตรวจสอบเพื่อดูว่าควรชดเชยในสถานะ FREE ถ้าไม่ได้ตั้งค่าหน่วย ระบบดีฟอลต์ถูกตั้งค่าเป็นวินาที ค่าดีฟอลต์คือ 900 วินาที

ตารางที่ 66. คีย์เวิร์ด ค่า และคำอธิบายสำหรับรายการในกลุ่มบรรทัดคีย์เวิร์ดโกลบอล (ต่อ)

คีย์เวิร์ด	ค่า	คำอธิบาย
saveInterval	num [*units*]	ระบุความบ่อยที่เซิร์ฟเวอร์ **DHCP** ควรบังคับใช้การบันทึกของ ฐานข้อมูลเปิด สำหรับเซิร์ฟเวอร์ที่โหลดหนัก ค่านี้ควรเป็น 60 หรือ 120 วินาที ถ้าไม่ได้ตั้งค่าหน่วย ระบบดีฟอลต์ถูกตั้งค่าเป็นวินาที ค่าดีฟอลต์คือ 3600 วินาที
clientpruneintv	num [*units*]	ระบุความถี่ของเซิร์ฟเวอร์ **DHCP** มี ฐานข้อมูลลบไคลเอ็นต์ไม่ได้เชื่อมโยงกับแอดเดรสใดๆ (ใน สถานะ UNKNOWN) ซึ่งช่วยลด การใช้หน่วยความจำของเซิร์ฟเวอร์ **DHCP** ถ้าไม่ได้ตั้งค่าหน่วย ระบบดีฟอลต์ถูกตั้งค่าเป็น วินาที ค่าดีฟอลต์คือ 3600 วินาที
numprocessthreads	num	ระบุจำนวนของเธรดตัวประมวลผลแพ็กเก็ต ที่จะสร้าง ค่าต่ำสุดคือหนึ่ง แต่ละเธรดโปรเซสจัดการหนึ่งไคลเอ็นต์ ตามค่าดีฟอลต์คือ 30
numpacketthreads	num	ระบุจำนวนเธรดแพ็กเก็ตที่จะสร้าง ค่าต่ำสุดนี้คือ 1 แต่ตามค่าดีฟอลต์ถูกตั้งค่าเป็น 5
numloggingthreads	num	ระบุจำนวนเธรดการล็อก ค่าดีฟอลต์คือ 1
numduidbuckets	num	นี่ถูกใช้โดยตัวจัดการตาราง และมี การเทียบโดยตรงกับ **numprocessthreads** ตามค่าดีฟอลต์ ถูกตั้งค่าเป็น 53
numclientbuckets	num	จำนวนกลุ่มที่จะใช้เก็บเร็กคอร์ดไคลเอ็นต์ ตามค่าดีฟอลต์คือ 1021
ignoreinterfacelist	*interface* [*interface*]	รายการอินเตอร์เฟสที่ละเว้น ซึ่งสามารถเป็นอินเตอร์เฟสเดี่ยว หรือหลายอินเตอร์เฟส
backupfile	"*filename*"	ไฟล์ที่จะใช้สำหรับฐานข้อมูลสำรอง ไฟล์ดีฟอลต์คือ /etc/dhcpv6/db_file6.crbk
checkpointfile	"*filename*"	ระบุไฟล์ checkpoint ฐานข้อมูล ไฟล์ checkpoint แรกคือพาธ ไฟล์ checkpoint ที่สองคือพาธ ที่อักขระสุดท้ายแทนที่ด้วย 2 ดังนั้นไฟล์ checkpoint ไม่ควรลงท้ายด้วย 2
clientrecorddb	"*filename*"	ระบุไฟล์บันทึกฐานข้อมูล ไฟล์มีไคลเอ็นต์เร็กคอร์ดทั้งหมด ที่เซิร์ฟเวอร์ **DHCP** ให้บริการแล้ว ไฟล์ดีฟอลต์คือ **/etc/dhcpv6/db_file6.cr**
duid	idtype *value* [*value*]	ใช้เพื่อจำแนกเซิร์ฟเวอร์ ค่าต่อไปนี้สามารถใช้ได้: • duid 1 *interface* • duid 2 *interface* • duid 3 enterprise number *identifier* • duid number 0x*hexdigit*
preference-number	num	อนุญาตให้ไคลเอ็นต์จำแนกเซิร์ฟเวอร์ ที่ต้องการข้อมูล ค่ายิ่งสูงยิ่งมีโอกาสที่ไคลเอ็นต์จะใช้เซอร์วิสนี้สำหรับเซอร์วิส ค่าดีฟอลต์และค่าสูงสุด คือ 255
unicast-enable	*policy*	นโยบาย Unicast สำหรับเซิร์ฟเวอร์ นี่ช่วยให้ เซิร์ฟเวอร์สื่อสารกันโดยใช้ unicast ตามค่าดีฟอลต์ ค่านี้ถูกเปิดไว้
tablemgr-policy	*policy*	อนุญาตให้เซิร์ฟเวอร์มีตัวจัดการตาราง เพื่อจัดการไคลเอ็นต์ขาเข้าได้ดีขึ้น ตามค่าดีฟอลต์ ค่านี้ถูกเปิดไว้
auth	*policy*	อนุญาตให้เซิร์ฟเวอร์สนับสนุนการพิสูจน์ตัวตนแบบดีเลย์ ตามค่าดีฟอลต์ ค่านี้ถูกปิดไว้
auth-keyfile	"*filename*"	ไฟล์ที่มีคีย์การพิสูจน์ตัวตนแบบดีเลย์ สำหรับไคลเอ็นต์ ไฟล์ดีฟอลต์คือ **/etc/dhcpv6/dhcpsdv6.keys**

คำสั่งคอนเทนเนอร์ที่ไม่ซ้อนของ DHCPv6:

คีย์เวิร์ดของเซิร์ฟเวอร์ **DHCPv6 subnet** สำหรับ รายการในคำสั่งคอนเทนเนอร์ที่ไม่ซ้อน

คำสั่งคอนเทนเนอร์ที่ไม่ซ้อนสามารถมีได้เป็นส่วนหนึ่งของคีย์เวิร์ด โกลบอล

ตารางที่ 67. คีย์เวิร์ด ค่า และคำอธิบายสำหรับรายการ ในคำสั่งคอนเทนเนอร์ที่ไม่ซ้อน

ไอเท็ม	คำอธิบาย	
subnet	*subnetid prefix-length* [*range*] {OPTIONS}	ระบุซับเน็ตที่จะใช้งาน *subnetid* ต้องเป็น แอดเดรส IPv6 ส่วน *prefix-length* ต้องเป็นจำนวนเต็มบวก ที่น้อยกว่า 128

คำสั่งคอนเทนเนอร์ที่ซ้อนของ DHCPv6:

คำสั่งคอนเทนเนอร์ที่ซ้อนสามารถใช้เป็นอ็อพชัน ภายในซับเน็ต

คอนเทนเนอร์ทั้งหมดสามารถมีคอนเทนเนอร์อื่นซ้อนอยู่ภายในคอนเทนเนอร์ นอกจากกำหนดสภาวะไว้ ความลึกสูงสุดของการซ้อนคือ 7 รวมถึงซับเน็ต และโกลบอลคอนเทนเนอร์ (เฉพาะคอนเทนเนอร์ที่ซ้อนกันห้าชั้นสามารถอยู่ภายใต้ซับเน็ต คอนเทนเนอร์)

คอนเทนเนอร์ผู้ผลิตและอินอ็อพชันไม่สามารถมีคอนเทนเนอร์อื่นซ้อน อยู่ภายในได้

ตารางที่ 68. คีย์เวิร์ด ค่า และคำอธิบายสำหรับรายการ ในคำสั่งคอนเทนเนอร์ที่ซ้อน

คีย์เวิร์ด	ค่า	คำอธิบาย
class	*name* [*range*] {OPTIONS|COMMON OPTIONS }	คลาสคอนเทนเนอร์ ค่า *name* คือสตริงเดี่ยว, ช่องว่างแยกสตริง, นิพจน์ปกติ, เลขฐานสิบหก 0x*ตัวเลขฐานสิบหก*, 0x*ตัวเลขฐานสิบหก*
vendor	*name* [*range*] {OPTIONS|COMMON OPTIONS }	คอนเทนเนอร์ผู้ผลิต ค่า *name* คือสตริงเดี่ยว, ช่องว่างแยกสตริง, นิพจน์ปกติ, เลขฐานสิบหก 0x*ตัวเลขฐานสิบหก*, 0x*ตัวเลขฐานสิบหก*
client	<*id* | 0 0xhexdigit | regular expression> <ip | *range* | none | any> {OPTIONA|COMMON OPTIONS }	ไคลเอ็นต์คอนเทนเนอร์ *id*- 1-hexdigit, 2-hexdigit, 3-hexdigit <ip|range|none|any> - IP แอดเดรสที่ให้ไคลเอ็นต์ที่ตรงกับ ID
inoption	*icode keytomatch* [*range*] { OPTIONS|COMMON OPTIONS }	อินอ็อพชันคอนเทนเนอร์ *icode* - โค้ดหรือตัวเลขของอ็อพชันขาเข้าที่ถูกระบุไว้โดยไคลเอ็นต์ *keytomatch* - ข้อมูลอ็อพชันที่ตรงกัน

อ็อพชันไฟล์ cnf ของ DHCPv6:

อ็อพชันไฟล์ cnf ที่กล่าวถึงที่นี่ สำหรับ **DHCPv6** อาจมีอยู่ภายในคอนเทนเนอร์เท่านั้น

ตารางที่ 69. คีย์เวิร์ด ค่า และคำอธิบาย สำหรับรายการในกลุ่มบรรทัดอ็อพชัน

คีย์เวิร์ด	ค่า	คำอธิบาย
exclude	*range*	ช่วง IP ที่ไม่รวมจากช่วงปัจจุบัน มักใช้ เมื่อไม่ได้ระบุช่วงเป็นส่วนหนึ่งของคำสั่งคอนเทนเนอร์
exclude	`ip`	IP แอดเดรสที่ไม่รวมจากช่วงปัจจุบัน
range	*range*	ช่วง IP ที่ใช้เพื่อขยายช่วงปัจจุบัน มักใช้ เมื่อไม่ได้ระบุช่วงเป็นส่วนหนึ่งของคำสั่งคอนเทนเนอร์
range	`ip`	IP แอดเดรสที่เพิ่ม ใช้เพื่อขยายช่วง
stealfromchildren	*policy*	ยึดแอดเดรสจากคอนเทนเนอร์ย่อย ถ้าไม่มีแอดเดรสเหลือแล้ว ตามค่าดีฟอลต์ ค่านี้ถูกปิดไว้
stealfrompeer	*policy*	ยึดแอดเดรสจากเพียร์คอนเทนเนอร์ ถ้าไม่มีแอดเดรสเหลือแล้ว ตามค่าดีฟอลต์ ค่านี้ถูกปิดไว้
stealfromparent	*policy*	ยึดแอดเดรสจากพาเรนต์คอนเทนเนอร์ ถ้าไม่มีแอดเดรสเหลือแล้ว ตามค่าดีฟอลต์ ค่านี้ถูกปิดไว้
balance-option	`{ balance-policy | <option | option option ...> }`	สมดุลอ็อพชันคอนเทนเนอร์ อ็อพชันระบุภายใน คอนเทนเนอร์นี้จะกำหนดให้ไคลเอ็นต์ฐานตามนโยบาย คีย์เวิร์ดนี้ สามารถมีได้เฉพาะภายใต้ซับเน็ตคอนเทนเนอร์
balance-policy	`b_policy`	ค่า b_policy สามารถเป็น `fill` หรือ `rotate` ค่าดีฟอลต์คือ `rotate`
fill-count	`num`	จำนวนครั้งที่อ็อพชันจะถูกส่งก่อน มอบอ็อพชันเดียวกันให้กับอินสแตนซ์ถัดไป
interface-id	"*interface*"	นี่สามารถแสดงรายการภายใต้ซับเน็ตเท่านั้น การร้องขอไคลเอ็นต์ ที่ได้รับบนอินเตอร์เฟสนี้จะได้รับอนุญาตให้รับแอดเดรส

อ็อพชันทั่วไปของ DHCPv6:

คีย์เวิร์ดเหล่านี้เป็นอ็อพชันทั่วไปของ **DHCPv6**

ซึ่งสามารถอยู่ภายในคอนเทนเนอร์หรือในส่วนโกลบอล:

ตารางที่ 70. คีย์เวิร์ด ค่า และคำอธิบายสำหรับอ็อพชันทั่วไป

คีย์เวิร์ด	ค่า	คำอธิบาย
reconfig-policy	*policy*	อนุญาตให้เซิร์ฟเวอร์ส่งข้อความคอนฟิกูเรชันใหม่ ไปให้ไคลเอ็นต์ ตามค่าดีฟอลต์ นี่ไม่ถูกตั้งค่าและถือเป็นปิด
rapid-commit	*policy*	อนุญาตสำหรับเซิร์ฟเวอร์เพื่อส่งมอบอย่างรวดเร็วสำหรับคอนเทนเนอร์หรือชุดโกลบอล ตามค่าดีฟอลต์ นี่ไม่ถูกตั้งค่าและถือเป็นปิด
prefered-lifetime	`num [`*units*`]`	ช่วงชีวิตที่ต้องการ IANA หรือ IATA ค่าดีฟอลต์ คือ 43200 วินาที
valid-lifetime	`num [`*units*`]`	ช่วงชีวิตที่ถูกต้องของ IANA หรือ IATA ค่าดีฟอลต์คือ 86400 วินาที
rebind	`num`	เปอร์เซ็นต์ของเวลาผูกรวมใหม่ 0-100 สำหรับแอดเดรส ค่าดีฟอลต์คือ 80 เปอร์เซ็นต์
renew	`num`	เปอร์เซ็นต์ของเวลาเริ่มใหม่ 0-100 สำหรับแอดเดรส ค่าดีฟอลต์คือ 50 ตัวอักษร

ตารางที่ 70. คีย์เวิร์ด ค่า และคำอธิบายสำหรับอ็อพชันทั่วไป (ต่อ)

คีย์เวิร์ด	ค่า	คำอธิบาย
unicast-option	policy	อนุญาตให้คอนเทนเนอร์แสดงการแลกเปลี่ยนข้อความโดยใช้ unicasting ซึ่งสามารถใช้เพื่อเปิดและปิดคอนเทนเนอร์เฉพาะและซับเน็ต แม้ว่านโยบายเซิร์ฟเวอร์แตกต่างกัน ตามค่าดีฟอลต์ นี่ไม่ถูกตั้งค่าและถือเป็นปิด
option	num <string|stings|hex>	สำหรับรายการอ็อพชัน โปรดดู “อ็อพชันไฟล์ที่รู้จักของเซิร์ฟเวอร์ DHCPv6”
change-optiontable	optiontable	อนุญาตเฉพาะภายในคอนเทนเนอร์ของผู้ผลิต

อ็อพชันไฟล์ที่รู้จักของเซิร์ฟเวอร์ DHCPv6:

อ็อพชันไฟล์ที่รู้จักของเซิร์ฟเวอร์ **DHCPv6** ถูกอธิบายไว้ที่นี่

อ็อพชันต่อไปนี้คืออ็อพชันไฟล์ที่รู้จักของเซิร์ฟเวอร์ **DHCPv6** อ็อพชันที่มี ″No″ ในคอลัมน์ **สามารถระบุได้** ไม่สามารถระบุได้ในไฟล์คอนฟิกูเรชัน ถ้าถูกระบุไว้ ก็จะถูกละเว้น

หมายเลขอ็อพชัน	ชนิดข้อมูลดีฟอลต์	สามารถระบุได้?	รายละเอียด
1	None	No	Solicit
2	None	No	Advertise
3	None	No	Request
4	None	No	Confirm
5	None	No	Address
6	None	No	Option Request
7	number	No	The preference number of the server
8	None	No	Elapse Time
9	None	No	Relay Message
11	None	No	Auth
12	ASCII string yes, no, true, false	Yes	Unicast
13	None	No	Status
14	ASCII string yes, no, true, false	Yes	Rapid Commit
15	None	No	User Class
16	None	No	Vendor Class
17	None	No	Vendor Option
18	None	No	Interface ID
19	None	No	Reconfiguration Message
20	ASCII string yes, no, true, false	Yes	Reconfiguration Accept

หมายเลขอ็อพชัน	ชนิดข้อมูลดีฟอลต์	สามารถระบุได้?	รายละเอียด
23	Space separated IPv6 addresses	Yes	DNS servers
24	ASCII string	Yes	Domain list

ค่าพารามิเตอร์ DHCPv6:

ค่าเหล่านี้สามารถใช้สำหรับพารามิเตอร์ **DHCPv6**

units: second, seconds, minute, minutes, hour, hours, day, days, week, weeks, month, months, year, years

interface: en0, en1, tr0

identifier: numbers หรือ characters

policy: yes, no, true, false

range: ipv6addresss-ipv6addresss

regular expression: "!epression to match$", "!epression to match^"

ตัวอย่าง ไฟล์ /etc/dhcpv6/dhcpsdv6.cnf:

ตัวอย่าง ไฟล์ `/etc/dhcpv6/dhcpsdv6.cnf` ที่แสดงไว้ที่นี่ เป็นส่วนหนึ่งของเนื้อหาไฟล์

```
logging_info{
        logFileSize 4000
        logItem     SYSERR
        logItem     PROTERR
        logItem     WARNING
        logItem     EVENT
        logItem     ACTION
        logItem     INFO
        logItem     ACNTING
        logItem     TRACE
        numLogFiles     3
        logFileName "/var/tmp/dhcpsdv6.log"
}
duid 1 en0
numprocessthreads 10
numpacketthreads  5
preference-number 255
reconfig-policy no
rapid-commit  no
unicast-option    yes
leaseExpiredInterval  3000 seconds
unicast-enable          yes
saveInterval            60 seconds
reservedTimeInterval    8000 seconds
reservedTime            10000 seconds
clientpruneintv         20  seconds
```

```
subnet bbbb:aaaa:: 40 bbbb:aaaa::0004-bbbb:aaaa::000f {
        balance-option {
               option 23 dead::beef
               option 23 beef::aaaa
               option 24 yahoo.com
        }

subnet dead:dead:aaaa:: 48 dead:dead:aaaa:aaaa::0006-dead:dead:aaaa:aaaa::000a {
        interface-id "en1"
        preferred-lifetime      100 seconds
        valid-lifetime          200 seconds
        rapid-commit  yes
        option 23 dead::beef beef:aaaa::bbbb:c aaaa:bbbb::cccc
        option 24 ibm.com austin.ibm.com
}
```

DHCPv6 ไคลเอ็นต์คอนฟิกูเรชัน

ไฟล์ /etc/dhcpv6/dhcpc6.cnf ถูกใช้เพื่อกำหนดคอนฟิก ไคลเอ็นต์ **DHCPv6**

คำสั่งที่สามารถระบุได้ในไฟล์นี้จะถูกรวมไว้ที่นี่ ถ้า **dhcpcd6** ตั้งใจว่าจะเริ่มต้นข้ามการรีบูต, เพิ่มรายการลงในไฟล์ /etc/rc.tcpip

คีย์เวิร์ดการล็อก:

คีย์เวิร์ดการล็อกที่ถูกต้องของเซิร์ฟเวอร์ **DHCPv6** ถูกอธิบายไว้ที่นี่

คีย์เวิร์ดต้อไปนี้จะใช้ได้:

ตารางที่ 71. คีย์เวิร์ดและคำอธิบายสำหรับคีย์เวิร์ดการล็อก

คีย์เวิร์ด	คำอธิบาย
log-file-name	พาธและชื่อไฟล์ของไฟล์ล็อกล่าสุดที่สำคัญมาก ชื่อไฟล์ล่าสุดที่สำคัญน้อยจะมีหมายเลข 1 ถึง (n-1) ต่อท้ายชื่อไฟล์ หมายเลขยิ่งมากไฟล์ยิ่งมีความสำคัญน้อย
log-file-size	ระบุขนาดสูงสุดของไฟล์ล็อกเป็น KB เมื่อ ขนาดของไฟล์ล็อกล่าสุดที่มีความสำคัญถึงค่านี้ ไฟล์จะถูกเปลี่ยนชื่อ และไฟล์ใหม่จะถูกสร้างขึ้น
log-file-num	ระบุจำนวนสูงสุดของไฟล์ล็อกที่เก็บไว้เมื่อขนาด ไฟล์ล็อกล่าสุดที่มีความสำคัญถึงค่า **log-file-size** และไฟล์จะถูกเปลี่ยนชื่อเพื่อสร้างไฟล์ใหม่

ตารางที่ 71. คีย์เวิร์ดและคำอธิบายสำหรับคีย์เวิร์ดการล็อก (ต่อ)

คีย์เวิร์ด	คำอธิบาย
log-item	ระบุไอเท็มล็อกที่จำเป็นต้องล็อก **SYSERR** ข้อผิดพลาดระบบ **OBJERR** ข้อผิดพลาดอ็อบเจ็กต์ **PROTERR** ข้อผิดพลาดเกี่ยวกับโปรโตคอล **WARNING** คำเตือน **EVENT** เกิดเหตุการณ์ขึ้น **ACTION** เกิดการดำเนินการโดยโปรเซส **INFO** ข้อมูลเพิ่มเติม **ACNTING** บุคคลที่ทำหน้าที่เมื่อ **TRACE** การไหลของโค้ด การดีบัก

คีย์เวิร์ด DUID:

ค่าคีย์เวิร์ดต่อไปนี้สำหรับรายการ DUID

รูปแบบของรายการ DUID มีดังต่อไปนี้:

```
duid  <duid_type> <value>  <value> ...
```

ชนิด DUID สามารถเป็นคีย์เวิร์ดหรือตัวเลข เตรียมที่ว่างสำหรับชนิด DUID ที่อาจถูกกำหนดในอนาคต ซึ่งในตอนนี้มี DUID 3 ชนิดตาม RFC 3315 :

ตารางที่ 72. คีย์เวิร์ดและค่าสำหรับรายการ DUID

คีย์เวิร์ด	คำอธิบาย
LLT	ชนิด DUID-LLT (ค่า 1)
LL	DUID-LL (ค่า 2)
TH	ชนิด DUID-EN (ค่า 3)

รูปแบบเฉพาะของรายการ DUID ขึ้นกับคีย์เวิร์ดที่ใช้งาน

```
duid  LLT    <interface name>
duid  LL     <interface name>
duid  EN     <enterprise number>  <enterprise identifier>
duid  <number>  <hex data (prefixed with '0x')
```

คีย์เวิร์ดเฉพาะข้อมูล:

คีย์เวิร์ดเฉพาะข้อมูลอยู่ในรูปแบบ `info-only` *`interface name`*

ปฏิบัติตามคีย์เวิร์ดเฉพาะข้อมูล:

ตารางที่ 73. คีย์เวิร์ด และคำอธิบายของคีย์เวิร์ดเฉพาะข้อมูล

คีย์เวิร์ด	คำอธิบาย
`info-only` *`interface name`*	คีย์เวิร์ดนี้ระบุชื่ออินเตอร์เฟสสำหรับ ไคลเอ็นต์ที่ได้รับเฉพาะข้อมูลคอนฟิกูเรชันและไม่ใช่แอดเดรส จากเซิร์ฟเวอร์

คีย์เวิร์ดต่อสัญญาเช่าและผูกรวมใหม่:

คีย์เวิร์ดต่อสัญญาเช่าและผูกรวมใหม่อธิบายไว้ที่นี่สำหรับเซิร์ฟเวอร์ **DHCPv6**

ตารางที่ 74. คีย์เวิร์ดและคำอธิบายสำหรับคีย์เวิร์ดต่อสัญญาเช่า และการโยงอีกครั้ง

คีย์เวิร์ด	คำอธิบาย
`rebind-time` *`value`*	ในเหตุการณ์ที่ไคลเอ็นต์ล้มเหลวการต่ออายุการเช่าซื้อ (เนื่องจากเซิร์ฟเวอร์ไม่ตอบสนอง), rebind-time จะระบุเวลา ที่ไคลเอ็นต์ติดต่อเซิร์ฟเวอร์อื่นเพื่อผูกรวมการเช่าซื้อ
`renew-time` *`value`*	renew-time ระบุเวลาที่ไคลเอ็นต์ติดต่อ เซิร์ฟเวอร์จากไคลเอ็นต์ที่ได้รับข้อมูลการเช่าซื้อ เพื่อต่ออายุการเช่าซื้อ

คีย์เวิร์ดการส่งผ่านที่เรียกร้อง:

คีย์เวิร์ดการส่งผ่านอีกครั้งที่เรียกร้องประกอบด้วย **solicit-maxcount** และ **solicit-timeout**

ตารางที่ 75. คีย์เวิร์ดและคำอธิบายสำหรับคีย์เวิร์ด การส่งผ่านอีกครั้งที่เรียกร้อง

คีย์เวิร์ด	คำอธิบาย
solicit-maxcount	คีย์เวิร์ด solicit-maxcount ระบุจำนวนข้อความเรียกร้อง ที่ไคลเอ็นต์ส่งไปให้เซิร์ฟเวอร์ ก่อนที่ไคลเอ็นต์จะได้รับการตอบกลับ จากเซิร์ฟเวอร์
solicit-timeout	คีย์เวิร์ด solicit-timeout ระบุเวลาจนถึง ที่ไคลเอ็นต์พยายามส่งข้อความเรียกร้องไปยังเซิร์ฟเวอร์ ก่อน ที่ไคลเอ็นต์จะได้รับการตอบกลับจากเซิร์ฟเวอร์

คีย์เวิร์ด Option:

ถ้าคีย์เวิร์ด option ปรากฏอยู่นอกกลุ่มบรรทัด 'interface' แล้วจะถือว่าเป็นโกลบอล ซึ่งอ็อพชันจะถูกนำไปใช้กับอินเตอร์เฟสทั้งหมด ถ้า คีย์เวิร์ด option ปรากฏอยู่นอกกลุ่มบรรทัด 'interface' อ็อพชันเหล่านี้จะนำไป ใช้เฉพาะกับอินเตอร์เฟสนั้น

กลุ่มบรรทัด option มีรูปแบบดังต่อไปนี้:

```
option <keyword | option code>
option <keyword | option code> exec "exec string"
option <keyword | option code> { option specific parameters }
option <keyword | option code> { option specific parameters } exec "exec string"
```

โค้ด option สามารถระบุได้โดยใช้โค้ดอ็อพชันที่ลงทะเบียน IANA อย่างไรก็ตาม บางอ็อพชันสามารถระบุได้โดยใช้คีย์เวิร์ดที่ปรากฏ ด้านล่าง:

คีย์เวิร์ด	โค้ดอ็อพชัน
ia-na	3
ia-ta	4
request-option	6
rapid-commit	14
user-class	15
vendor-class	16
vendor-opts	17
reconf-accept	20
dns-servers	23
domain-list	24

คำอธิบายเพิ่มเติมของแต่ละคีย์เวิร์ดต่อไปนี้:

คีย์เวิร์ด	วัตถุประสงค์ รูปแบบ และพารามิเตอร์
ia-na	**วัตถุประสงค์** ระบุ option 3 ถ้าระบุไว้ไคลเอ็นต์จะร้องขอแอดเดรสที่ไม่ใช่แบบชั่วคราว จากเซิร์ฟเวอร์ **รูปแบบ** `option ia-na [ { parameters } ] [ exec "exec string" ]` **พารามิเตอร์** Option **ia-na** ใช้พารามิเตอร์ต่อไปนี้: ia-id *value* renew-time *value* rebind-time *value* พารามิเตอร์เหล่านี้ระบุ ค่า user-preferred และเป็นทางเลือก *value* ระบุ เป็นเลขฐานสิบหรือเลขฐานสิบหกโดยนำหน้าด้วย '0x'
ia-ta	**วัตถุประสงค์** ระบุ option 4 ถ้าระบุไว้ไคลเอ็นต์จะร้องขอแอดเดรสชั่วคราว จากเซิร์ฟเวอร์ **รูปแบบ** `option ia-ta [ { parameters } ] [ exec "exec string" ]` **พารามิเตอร์** option **ia-ta** ใช้พารามิเตอร์ต่อไปนี้: ia-id *value* พารามิเตอร์นี้ ระยุค่าที่ผู้ใช้ต้องการ และเป็นทางเลือก *value* ระบุ เป็นเลขฐานสิบหรือเลขฐานสิบหกโดยนำหน้าด้วย '0x'

คีย์เวิร์ด	วัตถุประสงค์ รูปแบบ และพารามิเตอร์
request-option	**วัตถุประสงค์** ระบุ option 6 ถ้าระบุไว้ ไคลเอ็นต์จะร้องขอรายการอ็อพชัน จากเซิร์ฟเวอร์ **รูปแบบ** `option request-option { parameters } [ exec "exec string" ]` **พารามิเตอร์** option **request-option** ใช้ช่องว่างแบ่ง รายการโค้ดอ็อพชัน (ฐานสิบ) เป็นอาร์กิวเมนต์
rapid-commit	**วัตถุประสงค์** ระบุ option 14 ถ้าระบุไว้ ไคลเอ็นต์จะบ่งชี้ว่าไคลเอ็นต์ได้ เตรียมการดำเนินการแลกเปลี่ยนข้อความตอบกลับแบบ Solicit **รูปแบบ** `option rapid-commit [exec "exec string"]` **พารามิเตอร์** ไม่รับพารามิเตอร์ใดๆ นอกจากคำสั่ง optional exec
user-class	**วัตถุประสงค์** ระบุ option 15 ถ้าระบุไว้ ไคลเอ็นต์จะบ่งชี้ชนิดหรือหมวดหมู่ ของผู้ใช้หรือแอ็พพลิเคชันที่แสดง **รูปแบบ** `option user-class { parameters } [ exec "exec string" ]` **พารามิเตอร์** Option **user-class** รับอินสแตนซ์หนึ่งรายการหรือมากกว่า ของข้อมูลคลาสของผู้ใช้ แต่ละอินสแตนซ์ของข้อมูลคลาสของผู้ใช้คือสตริงที่มีหรือไม่เครื่องหมายคำพูดที่มีความยาวกำหนดเอง ถ้าสตริงมีช่องว่างอยู่ สตริงนั้นต้องล้อมรอบด้วย เครื่องหมายคำพูด พารามิเตอร์เป็นสิ่งจำเป็น รูปแบบสำหรับพารามิเตอร์ คือ: class *value* class *value* โดย *value* คือ สตริงที่มีหรือไม่มีเครื่องหมายคำพูด
vendor-class	**วัตถุประสงค์** ระบุ option 16 ถ้าระบุไว้ ไคลเอ็นต์จะบ่งชี้ผู้ผลิตที่ ผลิตฮาร์ดแวร์ที่ไคลเอ็นต์กำลังรันอยู่ **รูปแบบ** option vendor-class { *parameters* } [exec *"*exec string*"*] **พารามิเตอร์** Option **vendor-class** รับหมายเลของค์กรที่ลงทะเบียนไว้ ของผู้ผลิต และอินสแตนซ์หนึ่งอินสแตนซ์หรือมากกว่าของข้อมูลคลาสของผู้ผลิต แต่ละอินสแตนซ์ ของข้อมูลคลาสของผู้ผลิตคือสตริงที่มีหรือไม่มีเครื่องหมายคำพูดที่มีความยาวกำหนดเอง ซึ่งแต่ละ สตริงอธิบายลักษณะเฉพาะบางอย่างของฮาร์ดแวร์คอนฟิกูเรชันของไคลเอ็นต์ พารามิเตอร์*ไม่ใช่*ทางเลือก รูปแบบ คือ: vendor-id *value* class *value* class *value* โดย *value* คือ สตริงที่มีหรือไม่มีเครื่องหมายคำพูด

คีย์เวิร์ด	วัตถุประสงค์ รูปแบบ และพารามิเตอร์
vendor-opts	**วัตถุประสงค์** ระบุ option 17 ถ้าระบุไว้ไคลเอ็นต์จะบ่งชี้ข้อมูลที่เกี่ยวข้องกับผู้ผลิต ให้แก่เซิร์ฟเวอร์ **รูปแบบ** `option vendor-opts <enterprise-number> { parameters } [ exec "exec string" ]]` **พารามิเตอร์** Option **vendor-opts** รับหมายเลของค์กรที่ลงทะเบียนไว้ของผู้ผลิต และอินสแตนซ์หนึ่งอินสแตนซ์หรือมากกว่าของข้อมูลของผู้ผลิต แต่ละอินสแตนซ์ ของข้อมูลอ็อพชันของผู้ผลิตคือโค้ดอ็อพชันของผู้ผลิตตามด้วยข้อมูลอ็อพชันในรูปแบบสตริงหรือเลขฐานสิบหก พารามิเตอร์*ไม่ใช่*ทางเลือก รูปแบบ คือ: vendor-id *value* option *opcode option-data* option *opcode option-data* โดย *option-data* คือสตริงที่ มีหรือไม่มีเครื่องหมายคำพูดหรือสตริงเลขฐานสิบหก (นำหน้าด้วย '0x')
reconf-accept	**วัตถุประสงค์** ระบุ option 20 ถ้าระบุไว้ไคลเอ็นต์จะบ่งชี้ให้กับเซิร์ฟเวอร์ ว่าไคลเอ็นต์ต้องการยอมรับข้อความกำหนดคอนฟิกใหม่จากเซิร์ฟเวอร์ **รูปแบบ** `option reconf-accept [ { exec "exec string" } ]` **พารามิเตอร์** option **reconf-accept** ไม่รับอ็อพชันระบุ พารามิเตอร์ นอกจากคำสั่ง exec
dns-servers	**วัตถุประสงค์** ระบุ option 23 ถ้าระบุไว้ไคลเอ็นต์จะบ่งชี้ให้กับเซิร์ฟเวอร์ ถึงชุดที่ต้องการของเซิร์ฟเวอร์ DNS **รูปแบบ** `option dns-servers [ { parameters } ] [ exec "exec string" ]` **พารามิเตอร์** option **dns-servers** ใช้ช่องว่าง/เว้นบรรทัด แบ่งรายการแอดเดรส IPv6 เป็นอาร์กิวเมนต์
domain-list	**วัตถุประสงค์** ระบุ option 24 ถ้าระบุไว้ไคลเอ็นต์จะบ่งชี้รายการโดเมน ที่ต้องการ **รูปแบบ** `option domain-list [ { parameters } ] [ exec "exec string" ]` **พารามิเตอร์** option **domain-list** ใช้ช่องว่าง/เว้นบรรทัด แบ่งรายการสตริงชื่อโดเมน

คีย์เวิร์ดอินเตอร์เฟส:

คีย์เวิร์ดอินเตอร์เฟสอยู่ในรูปแบบ `interface <interface name> [ { option declaration/s } ]`

ตารางที่ 76. คีย์เวิร์ดและคำอธิบายสำหรับคีย์เวิร์ดอินเตอร์เฟส

คีย์เวิร์ด	คำอธิบาย
interface <*interface name*> [{ option declaration/s }]	คำสั่งอินเตอร์เฟสรับการประกาศอ็อพชันหนึ่งรายการหรือมากกว่า เป็นอาร์กิวเมนต์ อ็อพชันเหล่านี้ระบุไว้ภายในกลุ่มข้อความอินเตอร์เฟสระบุไว้กับอินเตอร์เฟสนี้ ไม่เหมือนอ็อพชันที่ประกาศไว้นอกกลุ่มข้อความอินเตอร์เฟส ที่ใช้กับอินเตอร์เฟสทั้งหมด

```
interface  en1  {
             option ia-na {
            ia-id  01
            renew-time 0x40
            rebind-time 0x60
 }

 option  request-option  { 3 23 24 }

       option  user-class {
          class ibm
               class "userclassA and B"
               class "userclassB"
       }

      option  vendor-class {
            vendor-id 1234
               class "vendorclassA"
               class "vendorclassB"
      }

      option  vendor-opts {
          vendor-id 2343
               option 89       vendoroption89
               option 90       vendoroption90
      }

 option reconf-accept
```

เอเจนต์รีเลย์ DHCP

ไฟล์ /etc/dhcprd.cnf คือไฟล์คอนฟิกูเรชัน สำหรับเอเจนต์รีเลย์ **DHCP** และ **BOOTP** รูปแบบของไฟล์ และคำสั่งและคีย์เวิร์ดที่อนุญาตถูกอธิบายไว้ที่นี่

คำสั่งถูกระบุในรูปแบบต่อไปนี้:

<*keyword*> <*value1*> ... <*valueN*>

การปรากฏของพารามิเตอร์และค่าเหล่านี้ถูกใช้โดยเอเจนต์รีเลย์ ที่เริ่มทำงานหรือรีสตาร์ท

ชุดพารามิเตอร์นี้ระบุไฟล์ล็อกที่จะควบคุมดูแล โดยเซิร์ฟเวอร์นี้ แต่ละพารามิเตอร์ถูกจำแนกโดยคีย์เวิร์ดและตามด้วยค่า

คีย์เวิร์ด	ค่า	นิยาม
numLogFiles	0 ถึง *n*	จำนวนไฟล์ล็อก ถ้าระบุ 0 ไว้ไม่มีล็อกไฟล์ จะถูกควบคุมดูแล และไม่มีข้อความล็อกแสดงผลใดๆ *n* คือ จำนวนสูงสุดของไฟล์ล็อกควบคุมตามขนาดของไฟล์ล็อกล่าสุด ที่เกินค่าสูงสุดและไฟล์ล็อกใหม่ถูกสร้างขึ้น
logFileSize	เป็น KB	ขนาดสูงสุดของไฟล์ล็อก เมื่อขนาดของไฟล์ล็อกล่าสุด เกิดค่านี้ ไฟล์จะถูกเปลี่ยนชื่อและไฟล์ล็อกใหม่จะถูกสร้างขึ้น
logFileName	พาธไฟล์	ชื่อของไฟล์ล็อกล่าสุด ไฟล์ล็อกล่าสุดที่สำคัญน้อยจะมีหมายเลข 1 ถึง (*n* – 1) ต่อท้ายชื่อไฟล์นั้น หมายเลขยิ่งมากไฟล์ยิ่งมีความสำคัญน้อย
logItem	หนึ่งไอเท็มที่จะล็อก	**SYSERR** ข้อผิดพลาดระบบ ที่อินเตอร์เฟสไปยังแพล็ตฟอร์ม **OBJERR** ข้อผิดพลาดอ็อบเจ็กต์ ระหว่างอ็อบเจ็กต์ในโปรเซส **PROTERR** ข้อผิดพลาดโปรโตคอล ระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ **WARNING** คำเตือน เรียกความสนใจจากผู้ใช้ **EVENT** เหตุการณ์ที่เกิดขึ้นกับโปรเซส **ACTION** เกิดการดำเนินการโดยโปรเซส **INFO** ข้อมูลที่อาจเป็นประโยชน์ **ACNTING** บุคคลที่ทำหน้าที่เมื่อ **TRACE** การไหลของโค้ด สำหรับดีบัก

ตัวอย่างเช่น ไฟล์ /etc/dhcprd.cnf อาจมีรายการ ต่อไปนี้:

```
numLogFiles  4
logFileSize  1000
logFileName    /usr/tmp/dhcprd.log
logItem     SYSERR
logItem     OBJERR
logItem     PROTERR
logItem     WARNING
logItem     EVENT
logItem     ACTION
logItem     INFO
logItem     ACNTING
logItem     TRACE
```

คีย์เวิร์ด	ค่า	นิยาม
relay	**IPv4**, **IPv6**, หรือ **ALL**	ระบุโหมดของแพ็กเก็ตรีเลย์ ถ้า **IPv4** ถูกระบุไว้ รีเลย์เอเจนต์จะทำหน้าที่เป็นรีเลย์เอเจนต์แบบ **DHCPv4** นี่คือโหมดดีฟอลต์ ของรีเลย์เอเจนต์ ถ้า **IPv6** ถูกระบุไว้รีเลย์เอเจนต์จะทำหน้าที่เป็น รีเลย์เอเจนต์ **DHCPv6** ถ้า **ALL** ถูกระบุไว้รีเลย์เอเจนต์ทำหน้าที่เป็นทั้ง รีเลย์เอเจนต์ **DHCPv4** และ **DHCPv6**
server	IP แอดเดรส	ระบุ IP แอดเดรสของเซิร์ฟเวอร์ **BOOTP** หรือ **DHCP** แพ็กเก็ตจะถูกส่งต่อไปที่เซิร์ฟเวอร์ที่แสดง ในไฟล์นี้
server6	แอดเดรส IPv6	ระบุแอดเดรส IPv6 ของเซิร์ฟเวอร์ **DHCPv6** แพ็กเก็ตจะถูกส่งต่อไปยังเซิร์ฟเวอร์ที่แสดงที่นี่
option6	*<option code> <option data>*	ระบุอ็อพชันรีเลย์เอเจนต์ **DHCPv6** คีย์เวิร์ดใช้ได้ ก็ต่อเมื่อโหมดรีเลย์ถูกตั้งค่าเป็น IPv6 ค่า *option code* ถูกระบุเป็นเลขฐานสิบ ค่า *option data* ถูก ระบุเป็นสตริงที่มีหรือไม่มีเครื่องหมายคำพูด ใน รูปแบบเลขฐานสิบหก (นำหน้าด้วย 0x)
single-site		ระบุอุปกรณ์ที่รีเลย์เอเจนต์กำลังรัน เป็นของไซต์ เพียงไซต์เดียว

Preboot Execution Environment Proxy DHCP daemon

เซิร์ฟเวอร์ PXE Proxy **DHCP** ทำหน้าที่เหมือนเซิร์ฟเวอร์ **DHCP** โดยรอสัญญาณสำหรับทราฟฟิกของไคลเอ็นต์ **DHCP** ปกติ และตอบสนองการร้องขอบางอย่าง ของไคลเอ็นต์ อย่างไรก็ตาม แตกต่างจากเซิร์ฟเวอร์ **DHCP** เซิร์ฟเวอร์ PXE Proxy **DHCP** ไม่ควบคุมดูแลแอดเดรสเครือข่าย และตอบสนองเฉพาะไคลเอ็นต์ที่ แสดงตัวว่าเป็นไคลเอ็นต์ PXE

การตอบสนองที่ให้โดยเซิร์ฟเวอร์ PXE Proxy **DHCP** มีกลไกทีไคลเอ็นต์ ค้นหาบูตเซิร์ฟเวอร์ หรือแอดเดรสเครือข่าย และคำ อธิบายของการสนับสนุน ความเข้ากันได้ของบูตเซิร์ฟเวอร์

การใช้เซิร์ฟเวอร์ PXE Proxy **DHCP** นอกเหนือจากเซิร์ฟเวอร์ **DHCP** ควรพิจารณาสามประเด็นสำคัญ ประการแรก คุณ สามารถแยกการควบคุมดูแลของ แอดเดรสเครือข่ายจากการควบคุมดูแลของบูตอิมเมจ การใช้สองโปรเซสที่แตกต่างกันบน ระบบ เดียวกัน คุณสามารถกำหนดคอนฟิกข้อมูลการบูตที่จัดการโดยเซิร์ฟเวอร์ PXE Proxy **DHCP** โดยไม่รบกวน หรือร้องขอ การเข้าถึง คอนฟิกูเรชันของเซิร์ฟเวอร์ **DHCP** ประการที่สอง คุณสามารถกำหนดเซิร์ฟเวอร์แบบบูตหลายระบบ และให้ไคล เอ็นต์ PXE เลือกเซิร์ฟเวอร์เฉพาะระหว่างเวลาบูต ตัวอย่างเช่น แต่ละบูตเซิร์ฟเวอร์สามารถเสนอระบบปฏิบัติการ หรือคอนฟิ กูเรชันระบบที่แตกต่างกัน ประการสุดท้าย การใช้พร็อกซีเซิร์ฟเวอร์ช่วยให้ กำหนดคอนฟิกไคลเอ็นต์ PXE ที่จะใช้ IP แอดเด รสแบบมัลติคาสต์เพื่อสำรวจ ตำแหน่งของบูตเซิร์ฟเวอร์ที่เข้ากันได้

เซิร์ฟเวอร์ PXE Proxy **DHCP** สามารถกำหนดคอนฟิกให้รันบน ระบบเดียวกันที่กำลังรันเซิร์ฟเวอร์ **DHCP** หรือบนระบบที่ แตกต่างกัน นอกจากนี้ ยังสามารถกำหนดคอนฟิกให้รันบนระบบเดียวกันที่กำลังรัน daemon ของบูตเซิร์ฟเวอร์หรือระบบที่ แตกต่างกัน

PXE Proxy DHCP เซิร์ฟเวอร์คอมโพเนนต์

มีสามคอมโพเนนต์ของเซิร์ฟเวอร์ PXED

เซิร์ฟเวอร์ PXED ถูกแบ่งเซกเมนต์เป็นสามส่วนหลักๆ คือฐานข้อมูล โปรโตคอลเอ็นจิ้น และชุดเธรดของเซอร์วิส โดยแต่ละส่วนมีข้อมูลคคอนฟิกูเรชันของตัวเอง

ฐานข้อมูล PXED:

ฐานข้อมูล `db_file.dhcpo` ถูกใช้เพื่อสร้าง อ็อพชันที่ส่งให้กับไคลเอ็นต์เมื่อไคลเอ็นต์ส่งแพ็กเก็ต REQUEST

อ็อพชันคืนกลับตามฐานข้อมูลขึ้นอยู่กับชนิดของเซิร์ฟเวอร์ที่เลือก นี่ถูกตั้งค่าโดยใช้คีย์เวิร์ด **pxeservertype** ในไฟล์ `pxed.cnf`

การใช้ข้อมูลในไฟล์คอนฟิกูเรชัน ฐานข้อมูลถูกเตรียมพร้อม และตรวจสอบความสอดคล้อง

โปรโตคอลเอ็นจิน PXED:

โปรโตคอลเอ็นจินใช้ฐานข้อมูลเพื่อกำหนดสิ่งที่ข้อมูลควร คืนกลับให้ไคลเอ็นต์

เอ็นจินโปรโตคอล PXED อ้างอิงข้อมูลจำเพาะ Intel Preboot Execution Environment (PXE) เวอร์ชัน 2.1 และยังคง เข้ากันได้กับข้อมูลจำเพาะ Intel PXE เวอร์ชัน 1.1

การดำเนินการเธรด PXED:

ชิ้นสุดท้ายของเซิร์ฟเวอร์ PXED ตั้งค่าการดำเนินการ ที่ใช้เพื่อทำให้ระบบทำงาน เนื่องจากเซิร์ฟเวอร์ PXED เป็นเธรด การดำเนินการเหล่านี้ตั้งค่าเป็นเธรดเพื่อให้ทุกอย่างทำงานร่วมกัน

เธรดแรกคือ เธรด *main* ที่จัดการการร้องขอ SRC (เช่น **startsrc, stopsrc, lssrc, traceson,** และ **refresh**) เธรดนี้ทำงานพร้อมกับการทำงานทั้งหมด ที่มีผลกระทบกับเธรดทั้งหมดและจัดการสัญญาณ ตัวอย่างเช่น

- SIGHUP (-1) ทำให้รีเฟรชฐานข้อมูลทั้งหมดในไฟล์คอนฟิกูเรชัน
- SIGTERM (-15) ทำให้เซิร์ฟเวอร์หยุดแบบปกติ

เธรดอื่นประมวลผลแพ็กเก็ต ขึ้นอยู่กับชนิดเซิร์ฟเวอร์ ซึ่งสามารถมีหนึ่งหรือสองเธรด เธรดหนึ่งรอสัญญาณที่พอร์ต 67 และเธรดที่สองรอสัญญาณ ที่พอร์ต 4011 แต่ละเธรดสามารถจัดการการร้องขอจากไคลเอ็นต์

การกำหนดคอนฟิกเซิร์ฟเวอร์ PXED

ตามค่าดีฟอลต์ เซิร์ฟเวอร์ PXED ถูกกำหนดคอนฟิกโดยการอ่านไฟล์ `/etc/pxed.cnf` ซึ่งระบุอ็อพชันและแอดเดรสของฐานข้อมูลเริ่มต้นของเซิร์ฟเวอร์

เซิร์ฟเวอร์เริ่มทำงานจาก SMIT หรือผ่านคำสั่ง SRC

กำหนดคอนฟิกเซิร์ฟเวอร์ PXED เป็นส่วนที่ยากที่สุดในการใช้ PXED ใน เครือข่ายของคุณ ขั้นแรก ค้นหาสิ่งที่เครือข่ายที่คุณต้องการมีในไคลเอ็นต์ PXE ตัวอย่างต่อไปนี้กำหนดคอนฟิก **pxed** daemon เพื่อรัน บนเครื่องเดียวกันเป็นเซิร์ฟเวอร์ DHCP:

```
pxeservertype       proxy_on_dhcp_server

subnet default
{
```

```
    vendor pxe
    {
        option    6      2       # Disable Multicast boot server discovery
        option    8      1   2     9.3.4.5    9.3.4.6   2    1    9.3.149.29
                                   # The above option gives the list of bootservers
        option    9      0       "PXE bootstrap server" \
                                 1       "Microsoft Windows NT Boot Server"  \
                                 2       "DOS/UNDI Boot Server"
        option    10    20    "seconds left before the first item in the boot menu is auto-selected"
     }
}
```

อ็อพชันย่อยในคอนเทนเนอร์ผู้ผลิตถูกส่งไปไคลเอ็นต์ PXE ก็ต่อเมื่อ IP แอดเดรสของไคลเอ็นต์อยู่ในช่วง IPแอดเดรสของซับเน็ต (ตัวอย่าง 9.3.149.0 ถึง 9.3.149.255)

ตัวอย่างต่อไปนี้กำหนดคอนฟิก **pxed** daemon เพื่อรัน บนเครื่องที่แตกต่างกันกับเซิร์ฟเวอร์ **DHCP**:

```
subnet default
{
    vendor pxe
    {
        option    6      10      # The bootfile name is present in the client's initial pxed
                                   # offer packet.
        option    8      1   2     9.3.4.5    9.3.4.6   2    1    9.3.149.29
                                   # The above option gives the list of bootservers
        option    9      0       "PXE bootstrap server" \
                                 1       "Microsoft Windows NT Boot Server"  \
                                 2       "DOS/UNDI Boot Server"
        option    10    20    "seconds left before the first item in the boot menu is auto-selected"
        bootstrapserver          9.3.148.65
        pxebootfile    1    2    1   window.one
        pxebootfile    2    2    1   linux.one
        pxebootfile    1    2    1   hello.one
        client 6 10005a8ad14d any
        {
           pxebootfile    1    2    1   aix.one
           pxebootfile    2    2    1   window.one
        }
    }

Vendor pxeserver
{
    option     7     224.234.202.202
}
```

คีย์เวิร์ด **pxeservertype** ไม่ได้ตั้งค่าในไฟล์คอนฟิกูเรชัน เพื่อใช้ค่าดีฟอลต์ ซึ่ง **pdhcp_only**, หมายความว่าเซิร์ฟเวอร์ PXED กำลังรันบนเครื่องอื่นที่ไม่ใช่เซิร์ฟเวอร์ **DHCP** กำหนดคอนฟิกูเรชันนี้ เซิร์ฟเวอร์ PXED รอสัญญาณที่พอร์ตสองพอร์ต (67 และ 4011) สำหรับไคลเอ็นต์แพ็กเก็ต BINLD REQUEST/INFORM option 7 ถูกส่งไปให้เซิร์ฟเวอร์ BINLD เมื่อเซิร์ฟเวอร์ PXED ได้รับแพ็กเก็ต REQUEST/INFORM บนพอร์ต 67 จาก BINLD และ option 60 ถูกตั้งค่าไปยังเซิร์ฟเวอร์ PXED

วลีฐานข้อมูล `db_file` บ่งชี้ว่าวิธีที่ฐานข้อมูล ใช้ประมวลผลของไฟล์คอนฟิกูเรชัน บรรทัดความคิดเห็นจะขึ้นต้นด้วย เครื่องหมายปอนด์ (#) ตั้งแต่ # จนถึงสุดบรรทัดจะถูกละเว้น โดยเซิร์ฟเวอร์ PXED แต่ละบรรทัด `option` ถูกใช้โดยเซิร์ฟเวอร์ เพื่อ

บอกให้ไคลเอ็นต์ดำเนินการ “อ็อพชันย่อยของคอนเทนเนอร์ผู้ผลิต PXE” ในหน้า 332 อธิบาย สนับสนุนและอ็อพชันที่รู้จัก โปรดดู “ไวยากรณ์ไฟล์เซิร์ฟเวอร์ PXED สำหรับการดำเนินการเซิร์ฟเวอร์ทั่วไป” ในหน้า 333 สำหรับ วิธีที่ระบุอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จัก

ไฟล์คอนฟิกูเรชัน PXED:

ไฟล์คอนฟิกูเรชันมีส่วนแอดเดรส และส่วนนิยามอ็อพชัน ตามแนวคิดของคอนเทนเนอร์ที่เก็บอ็อพชัน โมดิฟายเออร์ และคอนเทนเนอร์อื่น

คอนเทนเนอร์ (โดยพื้นฐาน วิธีจัดกลุ่มอ็อพชัน) ใช้ตัวบ่งชี้เพื่อจำแนกไคลเอ็นต์เป็นกลุ่ม ชนิดคอนเทนเนอร์คือ ซับเน็ต คลาส ผู้ผลิต และไคลเอ็นต์ ในปัจจุบัน ยังไม่มีคอนเทนเนอร์ที่ผู้ใช้นิยามได้เอง ตัวบ่งชี้จำเพาะกำหนดไคลเอ็นต์ เพื่อให้สามารถติดตามไคลเอ็นต์ได้ เช่น ไคลเอ็นต์ย้ายระหว่างซับเน็ต สามารถใช้ชนิดคอนเทนเนอร์มากกว่าหนึ่งชนิดเพื่อกำหนดการเข้าถึงไคลเอ็นต์

อ็อพชัน คือตัวบ่งชี้ที่คืนค่าให้ไคลเอ็นต์ เช่น ดีฟอลต์เกตเวย์และแอดเดรส DNS

คอนเทนเนอร์ PXED:

เมื่อเซิร์ฟเวอร์ **DHCP** ได้รับการร้องขอ แพ็กเก็ตถูกวิเคราะห์ และจำแนกคีย์กำหนดคอนเทนเนอร์ อ็อพชัน และแอดเดรสถูกดึงออก

ตัวอย่างในคอนฟิกูเรชันของเซิร์ฟเวอร์ PXED แสดงซับเน็ตคอนเทนเนอร์ คีย์บ่งชี้คือตำแหน่งไคลเอ็นต์ในเครือข่าย ถ้าไคลเอ็นต์มาจาก เครือข่ายนั้น มันจะอยู่ในคอนเทนเนอร์นั้น

แต่ละชนิดของคอนเทนเนอร์ใช้อ็อพชันแตกต่างกันเพื่อแยกแยะไคลเอ็นต์:

- ซับเน็ตคอนเทนเนอร์ใช้ฟิลด์ `giaddr` หรืออินเตอร์เฟส แอดเดรสของอินเตอร์เฟสการรับเพื่อกำหนดซับเน็ตที่ไคลเอ็นต์มาจาก
- คลาสคอนเทนเนอร์ใช้ค่าใน option 77 (User Site Class Identifier).
- ผู้ผลิตใช้ค่าใน option 60 (Vendor Class Identifier)
- ไคลเอ็นต์คอนเทนเนอร์ใช้ option 61 (Client Identifier) สำหรับไคลเอ็นต์ PXE และฟิลด์ `chaddr` ในแพ็กเก็ต **BOOTP** สำหรับไคลเอ็นต์ **BOOTP**

ยกเว้นซับเน็ต แต่ละคอนเทนเนอร์อนุญาตให้ระบุค่า ที่ตรงกับมันรวมถึงการจับคู่นิพจน์ปกติ

อีกทั้งยังมีคอนเทนเนอร์โดยนัย *โกลบอล*คอนเทนเนอร์ อ็อพชัน และโมดิฟายเออร์อยู่ในโกลบอลคอนเทนเนอร์จนกว่าจะถูกลบล้าง หรือปฏิเสธ คอนเทนเนอร์ส่วนใหญ่สามารถอยู่ในคอนเทนเนอร์อื่น เพื่อแสดงขอบเขตการมองเห็น คอนเทนเนอร์อาจมีหรือไม่มีช่วงแอดเดรสที่เกี่ยวข้อง กับคอนเทนเนอร์ ซับเน็ตโดยธรรมชาติจะมีช่วงที่สัมพันธ์กับซับเน็ต

กฎพื้นฐานสำหรับคอนเทนเนอร์และคอนเทนเนอร์ย่อยคือ:

- คอนเทนเนอร์ทั้งหมดใช้ได้ที่ระดับโกลบอล
- ซับเน็ตไม่สามารถวางในคอนเทนเนอร์อื่น
- คอนเทนเนอร์ที่จำกัดไม่สามารถมีคอนเทนเนอร์ปกติที่มีชนิดเดียวกัน อยู่ภายในได้ (ตัวอย่าง เช่น คอนเทนเนอร์ที่มีอ็อพชันที่อนุญาตเฉพาะคลาส บัญชี จะไม่สามารถรวมคอนเทนเนอร์ด้วยอ็อพชัน ที่อนุญาตคลาสทั้งหมดที่ขึ้นต้นด้วย ″a″ นี่ไม่ถูกต้อง)

- ไคลเอ็นต์คอนเทนเนอร์ที่จำกัดไม่สามารถมีคอนเทนเนอร์ย่อยได้

กฎที่กำหนดข้างต้น คุณสามารถสร้างลำดับขั้นของคอนเทนเนอร์ ที่แบ่งเซกเมนต์อ็อพชันของคุณเป็นกลุ่มสำหรับไคลเอ็นต์เฉพาะ หรือชุดของไคลเอ็นต์

ถ้าไคลเอ็นต์ตรงกับหลายคอนเทนเนอร์ อ็อพชันและแอดเดรสจะถูกส่ง อย่างไร? เซิร์ฟเวอร์ **DHCP** รับข้อความ ซึ่งส่งผ่านการร้องขอ ไปที่ฐานข้อมูล (`db_file` ในกรณีนี้) และรายการคอนเทนเนอร์ ถูกสร้างขึ้น ซึ่งรายการจะแสดง ตามลำดับความลึกและลำดับความสำคัญ ลำดับความสำคัญถูกกำหนดเป็นลำดับขั้นโดยนัย ในคอนเทนเนอร์ คอนเทนเนอร์ที่เข้มงวดมีลำดับความสำคัญสูงกว่าคอนเทนเนอร์ปกติ ไคลเอ็นต์ คลาส ผู้ผลิต และสุดท้ายซับเน็ต ถูกจัดเรียง ตามลำดับดังกล่าว และภายในคอนเทนเนอร์ตามความลึก สิ่งนี้สร้างลำดับรายการตามที่ระบุไว้สูงสุด ไปยังต่ำสุด ตัวอย่าง เช่น:

```
Subnet 1
--Class 1
--Client 1
Subnet 2
--Class 1
----Vendor 1
----Client 1
--Client 1
```

ตัวอย่างข้างต้นแสดงซับเน็ต 2 รายการ `Subnet 1` และ `Subnet 2` มีชื่อคลาสหนึ่งรายการคือ `Class 1` ชื่อผู้ผลิตหนึ่งรายการ `Vendor 1` และชื่อไคลเอ็นต์หนึ่งรายการ `Client 1` `Class 1` และ `Client 1` ถูกกำหนดไว้หลายที่ เนื่องจาก รายการแตกต่างกันในคอนเทนเนอร์ ชื่อรายการสามารถเหมือนกัน แต่ค่าภายในสามารถ แตกต่างกันได้ ถ้า `Client 1` ส่งข้อความถึงเซิร์ฟเวอร์ **DHCP** จาก `Subnet 1` ด้วย `Class 1` ที่ระบุในรายการ อ็อพชัน เซิร์ฟเวอร์ **DHCP** ควรสร้างพาธคอนเทนเนอร์ต่อไปนี้:

`Subnet 1, Class 1, Client 1`

คอนเทนเนอร์ที่ระบุสำคัญสุดจะแสดงเป็นรายการสุดท้าย เมื่อต้องการขอรับแอดเดรส รายการจะถูกตรวจสอบ ตามลำดับขึ้นย้อนกลับเพื่อค้นหาแอดเดรสแรกที่พร้อมใช้งาน จาก รายการจะตรวจสอบตามลำดับขึ้นเพื่อรับอ็อพชัน อ็อพชันจะลบล้างค่าก่อนหน้า นอกจากอ็อพชัน **ปฏิเสธ** จะปรากฏ ในคอนเทนเนอร์ นอกจากนี้ ตั้งแต่ `Class 1` และ `Client 1` อยู่ใน `Subnet 1` รายการเรียงลำดับตาม ความสำคัญของคอนเทนเนอร์ ถ้าไคลเอ็นต์เดียวกันอยู่ใน `Subnet 2` และ ส่งข้อความเดียวกัน รายการคอนเทนเนอร์ที่สร้างคือ:

`Subnet 2, Class 1, Client 1` (ที่ระดับ `Subnet 2`), `Client 1` (ที่ระดับ `Class 1`)

`Subnet 2` แสดงเป็นอันดับแรก แล้ว `Class 1`, แล้ว `Client 1` ที่ระดับ `Subnet 2` (เนื่องจากคำสั่งของไคลเอ็นต์นี้เป็นระดับเดียวลงในลำดับขั้น) ลำดับขั้นโดยนัย ที่ไคลเอ็นต์ตรงกับคำสั่งแรกของไคลเอ็นต์ต่ำกว่าที่ระบุ การจับคู่ไคลเอ็นต์ `Client 1` ของ `Class 1` ภายใน `Subnet 2`

ลำดับความสำคัญที่เลือกตามความลึกภายในลำดับขั้นไม่ถูกแทนที่โดย ลำดับขั้นของคอนเทนเนอร์เอง ตัวอย่าง เช่น ถ้าไคลเอ็นต์เดียวกันใช้ข้อความ เดียวกัน และระบุตัวบ่งชี้ผู้ผลิต รายการคอนเทนเนอร์คือ:

`Subnet 2, Class 1, Vendor 1, Client 1` (ที่ระดับ `Subnet 2`), `Client 1` (ที่ระดับ `Class 1`)

ลำดับความสำคัญของคอนเทนเนอร์ช่วยปรับปรุงประสิทธิภาพการค้นหา เนื่องจากลำดับความสำคัญเป็นไปตาม แนวคิดทั่วไปที่ไคลเอ็นต์คอนเทนเนอร์สำคัญที่สุดระบุวิธีกำหนดหนึ่งไคลเอ็นต์หรือมากกว่า คลาสคอนเทนเนอร์มีแอดเดรสที่สำคัญน้อยกว่าไคลเอ็นต์คอนเทนเนอร์ ผู้ผลิตมีความสำคัญต่ำ และซับเน็ตมีความสำคัญต่ำ

แอดเดรสและช่วงแอดเดรสของ PXED:

ชนิดคอนเทนเนอร์ใดๆ สามารถมีช่วงแอดเดรสที่เกี่ยวข้อง ซับเน็ต ต้องมีช่วงแอดเดรสที่เกี่ยวข้อง แต่ละช่วงภายในคอนเทนเนอร์ต้องเป็นซับเน็ตของช่วง พาเรนต์คอนเทนเนอร์ และต้องไม่ซ้อนทับกับช่วงของคอนเทนเนอร์อื่น

ตัวอย่างเช่น ถ้าคลาสถูกกำหนดภายในซับเน็ต และคลาสมีช่วง ช่วงนั้นต้องเป็นเซ็ตย่อยของช่วงซับเน็ต นอกจากนี้ ช่วงภายในคลาสคอนเทนเนอร์ไม่สามารถซ้อนทับกับ ช่วงอื่นที่ระดับของมัน

ช่วงสามารถแสดงบนบรรทัดคอนเทนเนอร์และแก้ไขตามช่วง และไม่รวมคำสั่งที่อนุญาตสำหรับแยกชุดแอดเดรสที่เกี่ยวข้องกับคอนเทนเนอร์ ดังนั้น ถ้าคุณมีแอดเดรส 10 อันดับแรก และ 10 อันดับรองของซับเน็ตพร้อมใช้งานอยู่ ซับเน็ตสามารถระบุแอดเดรสเหล่านี้ตามช่วงในวลีซับเน็ตเพื่อลด ทั้งหน่วยความจำที่ใช้ และโอกาสที่แอดเดรสชนกันกับไคลเอ็นต์อื่น ที่ไม่อยู่ในช่วงที่ระบุ

หลังจากแอดเดรสถูกเลือก, คอนเทนเนอร์ที่ตามมา ในรายการที่มีช่วงแอดเดรสถูกลบออกจากรายการพร้อม รายการย่อย เหตุผลสำหรับสิ่งนี้คืออ็อพชันเฉพาะเครือข่ายในคอนเทนเนอร์ที่ลบ ไม่ถูกต้อง ถ้าแอดเดรสไม่ได้ใช้งานจากภายในคอนเทนเนอร์

อ็อพชันไฟล์คอนฟิกูเรชัน PXED:

หลังจากรายการถูกเลือกเพื่อกำหนดแอดเดรส ชุดของอ็อพชัน จะถูกสร้างขึ้นสำหรับไคลเอ็นต์

ในขั้นตอนการเลือกนี้ อ็อพชันจะเขียนทับอ็อพชันที่เลือกก่อนหน้า จนกว่าจะตรวจพบ *ปฏิเสธ* ซึ่งในกรณีดังกล่าว อ็อพชันที่ปฏิเสธจะถูกลบ ออกจากรายการที่ส่งให้กับไคลเอ็นต์ วิธีนี้อนุญาตให้สืบทอดจาก คอนเทนเนอร์หลักเพื่อลดปริมาณข้อมูลที่ต้องระบุ

การล็อก PXED:

พารามิเตอร์การล็อกถูกระบุไว้ในคอนเทนเนอร์ เช่น ฐานข้อมูล แต่คีย์เวิร์ดคอนเทนเนอร์คือ **logging_info**

เมื่อศึกษาการกำหนดคอนฟิก PXED ขอแนะนำให้ปรับการล็อกเป็นระดับสูงสุด นอกจากนี้ ควรอย่างยิ่งที่จะระบุคอนฟิกูเรชันการล็อกก่อน ข้อมูลของไฟล์คอนฟิกูเรชันอื่นๆ เพื่อให้แน่ใจว่าข้อผิดพลาดของคอนฟิกูเรชัน ถูกล็อกหลังจากที่ระบบย่อยการล็อกเริ่มทำงาน ใช้คีย์เวิร์ด **logitem** เพื่อเปิดระดับการล็อก หรือลบคีย์เวิร์ด **logitem** เพื่อปิดระดับการล็อก คีย์เวิร์ดอื่นสำหรับการล็อกอนุญาตให้ระบุชื่อไฟล์ของล็อก, ขนาดไฟล์ และจำนวนไฟล์ล็อกเวียน

ข้อควรพิจารณาเกี่ยวกับประสิทธิภาพของ PXED:

สำคัญอย่างยิ่งที่ควรทำความเข้าใจการกำหนดคอนฟิกคีย์เวิร์ด และโครงสร้างของไฟล์คอนฟิกูเรชันที่มีผลกระทบกับหน่วยความจำที่ใช้งาน และประสิทธิภาพของเซิร์ฟเวอร์ PXED

ขั้นแรก การใช้หน่วยความจำมากเกินไปสามารถหลีกเลี่ยงได้โดยทำความเข้าใจกับ โมเดลลำดับขั้นของอ็อพชันจากคอนเทนเนอร์พาเรนต์ถึงไชน์ในสภาพแวดล้อมที่สนับสนุน ที่ไม่มีไคลเอ็นที่ไม่แสดงรายการ ผู้ดูแลระบบต้องแสดงรายการแต่ละไคลเอ็นต์ไว้ในไฟล์ เมื่ออ็อพชันถูกแสดงรายการสำหรับไคลเอ็นต์ที่จำเพาะใดๆ เซิร์ฟเวอร์ใช้ หน่วยความจำมากขึ้นเพื่อเก็บผังคอนฟิกูเรชันมากกว่าเมื่ออ็อพชันสืบทอดจาก พาเรนต์คอนเทนเนอร์ (เช่น ซับเน็ต เครือข่าย หรือโกลบอลคอนเทนเนอร์) เพราะฉะนั้น ผู้ดูแลระบบควรตรวจสอบว่ามีอ็อพชันซ้ำที่ระดับไคลเอ็นต์ ภายในไฟล์คอนฟิกูเรชัน และถ้ามี กำหนดให้อ็อพชันเหล่านี้สามารถระบุไว้ใน พาเรนต์คอนเทนเนอร์และแบ่งใช้โดยชุดของคอนฟิกูเรชัน ทั้งชุด

นอกจากนี้ เมื่อใช้รายการ **logItem** INFO และ TRACE, ข้อความจำนวนมากจะถูกบันทึกระหว่างประมวลผลทุกๆ ข้อความของไคลเอ็นต์ PXE การเพิ่มบรรทัดในไฟล์ล็อกอาจเป็นการดำเนินการที่ใช้รีซอร์ส เพราะฉะนั้น จำกัดปริมาณของล็อกจะช่วยปรับปรุงประสิทธิภาพของเซิร์ฟเวอร์ PXED ได้ เมื่อข้อผิดพลาดกับเซิร์ฟเวอร์ PXED ถูกพัก การล็อกสามารถเปิดใช้งานใหม่แบบไดนามิก ได้โดยใช้คำสั่ง SRC **traceson**

อ็อพชันย่อยของคอนเทนเนอร์ผู้ผลิต PXE

เมื่อสนับสนุนไคลเอ็นต์ PXE เซิร์ฟเวอร์ **DHCP** ส่งผ่านอ็อพชัน ไปให้เซิร์ฟเวอร์ BINLD ซึ่ง BINLD ใช้ในการกำหนดคอนฟิก:

หมายเลข Opt	ชนิดข้อมูลดีฟอลต์	สามารถระบุได้?	คำอธิบาย
6	เลขฐานสิบ	ใช่	PXE_DISCOVERY_CONTROL จำกัด 0–16 นี่คือฟิลด์บิต บิต 0 เป็น บิตที่สำคัญน้อยที่สุด **บิต 0** ถ้าตั้งค่า ปิดใช้งานการสำรวจการกระจาย **บิต 1** ถ้าตั้งค่า ปิดใช้งานการสำรวจมัลติคาสต์ **บิต 2** ถ้าตั้งค่า เฉพาะเซิร์ฟเวอร์ที่ใช้/ยอมรับใน PXE_BOOT_ SERVERS **บิต 3** ถ้าตั้งค่า และชื่อไฟล์บูตอยู่ในแพ็กเก็ตเสนอ PXED เริ่มต้น ดาวน์โหลดไฟล์บูต (ไม่มีพร้อมต์/เมนู/สำรวจบูตเซิร์ฟเวอร์) **บิต 4–7** ต้องเป็น 0 ถ้าอ็อพชันนี้ไม่ถูกส่ง ไคลเอ็นต์จะถือว่าบิตทั้งหมด เป็น 0
7	One dotted quad	ใช่	IP แอดเดรสแบบมัลติคาสต์ บูตเซิร์ฟเวอร์สำหรับ IP แอดเดรสแบบมัลติคาสต์ คุณลักษณะบูตเซิร์ฟเวอร์ ของสำรวจมัลติคาสต์ต้องรับสัญญาณบนแอดเดรสมัลติคาสต์นี้ อ็อพชันนี้จำเป็นเมื่อการสำรวจมัลติคาสต์ปิดใช้งานบิต (บิต 1) ใน อ็อพชัน PXE_DISCOVERY_ CONTROL ถูกตั้งค่า

หมายเลข Opt	ชนิดข้อมูลดีฟอลต์	สามารถระบุได้?	คำอธิบาย
8	*ชนิดบูตเซิร์ฟเวอร์ (0-65535)*	ใช่	PXE_BOOT_SERVERS *ตัวนับ IP แอดเดรส (0-256)* **Type 0** Miscrosoft Windows *IP address...IP address* NT Boot Server *Boot server type IP address* **Type 1** Intel LCM Boot Server *count IP address ...* **Type 3** DOS/UNDI Boot Server *IP address* **Type 4** NEC ESMPRO Boot Server **Type 5** IBM WSoD Boot Server **Type 6** IBM LCCM Boot Server **Type 7** CA Unicenter TNG Boot Server. **Type 8** HP OpenView Boot Server. **Type 9 ถึง 32767** สงวนไว้ **Type 32768 ถึง 65534** ผู้ผลิตใช้ **Type 65535** เซิร์ฟเวอร์ทดสอบ PXE API ถ้า *ตัวนับ IP แอดเดรส* เป็นศูนย์ สำหรับชนิดเซิร์ฟเวอร์ไคลเอ็นต์อาจยอมรับข้อเสนอจากบูตเซิร์ฟเวอร์อื่นของ ชนิดนั้น บูตเซิร์ฟเวอร์ไม่ตอบสนองการร้องขอสำรวจชนิดที่ไม่สนับสนุน
9	*Boot server type (0-65535)*	ใช่	PXE_BOOT_MENU *"description"* บูตเซิร์ฟเวอร์ บูต "order" ปรากฏอยู่ในชนิด *"description"...ลำดับเมนู*
10	*หมดเวลาเป็นวินาที (0-255)*	ใช่	PXE_MENU_PROMPT *"prompt"* การหมดเวลาเป็นจำนวนวินาที ที่รอก่อนการเลือกไอเท็มเมนูแรกโดยอัตโนมัติ บนระบบไคลเอ็นต์ พร้อมต์จะปรากฏตามด้วยจำนวนวินาทีที่เหลือ ก่อนที่ไอเท็มแรกในเมนูบูตจะถูกเลือกโดยอัตโนมัติ ถ้าปุ่ม F8 ถูกกดไว้บนระบบไคลเอ็นต์ เมนูจะปรากฏขึ้น ถ้าอ็อพชันนี้ จัดเตรียมให้ไคลเอ็นต์ เมนูจะปรากฏขึ้นโดยไม่มีพร้อมต์ และการหมดเวลา ถ้าการหมดเวลาเป็น 0 ไอเท็มแรกในเมนูจะถูกเลือกโดยอัตโนมัติ ถ้าการหมดเวลาเป็น 255 เมนูและพร้อมต์จะปรากฏขึ้นโดยไม่มีการเลือกอัตโนมัติ หรือการหมดเวลา

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ PXED สำหรับการดำเนินการเซิร์ฟเวอร์ทั่วไป

คีย์เวิร์ดไฟล์เซิร์ฟเวอร์ PXED ของเซิร์ฟเวอร์ **DHCPv6** อธิบายไว้ที่นี่ เป็นการดำเนินการเซิร์ฟเวอร์ทั่วไป ฟอร์ม คอนเทนเนอร์ย่อย ค่าดีฟอลต์ และความหมายจะถูกจำแนก

หมายเหตุ: หน่วยเวลา (*time_units*) แสดงใน ตารางต่อไปนี้เป็นทางเลือก และแสดงโมดิฟายเออร์เป็นเวลาจริง หน่วยเวลาดีฟอลต์เป็นนาที ค่าที่ใช้ได้คือวินาที (1), นาที (60), ชั่วโมง (3600), วัน (86400), สัปดาห์ (604800), เดือน (2392000) และปี (31536000) จำนวนที่แสดงไว้ในวงเล็บคือตัวคูณที่ใช้กับค่าเฉพาะ *n* เพื่อ แปลงค่าเป็นวินาที

คีย์เวิร์ด	ฟอร์ม	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ฐานข้อมูล	ฐานข้อมูล *db type*	ใช่	ไม่มี	คอนเทนเนอร์หลักที่เก็บนิยามสำหรับพูลแอดเดรส อ็อพชัน และคำสั่งการเข้าถึงไคลเอ็นต์ *db type* คือชื่อโมดูลที่ถูกโหลดเพื่อประมวลผลส่วนนี้ของไฟล์ เฉพาะ ค่าที่พร้อมใช้งานในปัจจุบันคือ **db_file**
logging_info	logging_info	ใช่	ไม่มี	คอนเทนเนอร์การล็อกหลักที่กำหนดพารามิเตอร์การล็อก
logitem	logitem NONE	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem SYSERR	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem OBJERR	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem PROTOCOL	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem PROTERR	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem WARN	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem WARNING	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem CONFIG	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem EVENT	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem PARSEERR	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem ACTION	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem ACNTING	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem STAT	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem TRACE	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem RTRACE	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด
logitem	logitem START	ไม่	ดีฟอลต์ทั้งหมดไม่เปิดใช้งาน	เปิดใช้งานระดับการล็อก อนุญาตให้มี หลายบรรทัด

คีย์เวิร์ด	ฟอร์ม	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
numLogFiles	numLogFiles *n*	ไม่ใช่	0	ระบุจำนวนไฟล์ล็อกที่จะสร้าง ไฟล์ล็อกเปลี่ยนเมื่อไฟล์แรกเต็ม *n* คือจำนวนไฟล์ที่จะสร้าง
logFileSize	logFileSize *n*	ไม่ใช่	0	ระบุขนาดของแต่ละไฟล์ล็อกในหน่วย 1024 ไบต์
logFileName	logFileName *path*	ไม่	ไม่มี	ระบุพาธไปยังไฟล์ล็อกแรก ไฟล์ล็อกเดิมถูกตั้งชื่อ *filename* หรือ *filename.extension filename* ต้องเป็นอักขระไม่เกิน 8 อักขระ เมื่อไฟล์ถูกเปลี่ยน ไฟล์จะถูกเปลี่ยนชื่อโดยขึ้นต้นด้วยฐาน *filename*, แล้วต่อท้ายด้วยตัวเลข หรือแทนที่นามสกุลด้วยตัวเลข ตัวอย่างเช่น ถ้าชื่อไฟล์เดิมคือ `file` ไฟล์ที่เปลี่ยน จะกลายเป็น `file01` ถ้าชื่อไฟล์เดิมคือ `file.log`, ชื่อจะกลายเป็น `file.01`
pxeservertype	pxeservertype *servertype*	ไม่	dhcp_only	บ่งชี้ชนิดของเซิร์ฟเวอร์ **dhcpsd** *servertype* สามารถเป็น **proxy_on_dhcp_server** ซึ่งหมายถึง PXED กำลังรันอยู่ บนเครื่องเดียวกันกับเซิร์ฟเวอร์ **DHCP** และกำลังรอสัญญาณสำหรับไคลเอ็นต์ PXE ร้องขอบนพอร์ต 4011 เท่านั้น หรือค่าดีฟอลต์ **pdhcp_only** ซึ่งหมายความว่า PXED กำลังรันอยู่บนเครื่องที่แยกต่างหาก และกำลังรอสัญญาณสำหรับไคลเอ็นต์แพ็กเก็ต ที่พอร์ต 67 และ 4011

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ PXED สำหรับฐานข้อมูล db_file

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ PXED สำหรับฐานข้อมูล `db_file` ถูกอธิบายไว้ที่นี่ ฟอร์ม คอนเทนเนอร์ย่อย ค่าดีฟอลต์ และ ความหมายจะถูกจำแนก

หมายเหตุ:

1. หน่วยเวลา (*time_units*) ที่แสดงในตารางต่อไปนี้ เป็นทางเลือก และแสดงถึงตัวแก้ไขเวลาจริง หน่วยเวลา ดีฟอลต์คือ นาที ค่าที่ถูกต้องคือวินาที (1), นาที (60), ชั่วโมง (3600), วัน (86400), สัปดาห์ (604800), เดือน (2392000), และปี (31536000) ตัวเลข ที่แสดงในวงเล็บเป็นตัวคูณที่ใช้กับค่า *n* ที่ระบุ เพื่อระบุค่าในหน่วยวินาที
2. ไอเท็มที่ระบุในคอนเทนเนอร์หนึ่งสามารถถูกยกเลิกภายในคอนเทนเนอร์ย่อย ตัวอย่างเช่น คุณสามารถกำหนดไคลเอ็นต์ **BOOTP** โกลบอล แต่ภายใน บางซับเน็ตอนุญาตไคลเอ็นต์ **BOOTP** โดยระบุคีย์เวิร์ด supportBootp ในคอนเทนเนอร์ทั้งสอง
3. ไคลเอ็นต์ คลาส และคอนเทนเนอร์ผู้ขายอนุญาตการสนับสนุนนิพจน์ ปกติ สำหรับคลาสและผู้ขาย สตริงที่อยู่ในอัญประกาศที่มีอักขระตัวแรกหลังจาก อัญประกาศเป็นเครื่องหมายอัศเจรีย์ (!) บ่งชี้ว่าส่วนที่เหลือของสตริง ควรถูกจัดการเป็นนิพจน์ปกติ ไคลเอ็นต์คอนเทนเนอร์อนุญาตสำหรับ นิพจน์ทั่วไปบนฟิลด์ **hwtype** และ **hwaddr** สตริงเดี่ยว ใช้เพื่อแสดงแทนทั้งสองฟิลด์ด้วยรูปแบบต่อไปนี้:

decimal_number-data

หาก decimal_number เป็นศูนย์ ข้อมูลจะเป็นสตริง ASCII หากเป็นตัวเลขอื่น ข้อมูลจะเป็นตัวเลขฐานหก

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
subnet	ซับเน็ตดีฟอลต์	ใช่	ไม่มี	ระบุซับเน็ตที่ไม่มีช่วงใดๆ ซับเน็ตถูกใช้โดยเซิร์ฟเวอร์เฉพาะเมื่อตอบสนองแพ็กเก็ต INFORM จากไคลเอ็นต์
subnet	subnet *subnet id netmask*			ระบุซับเน็ตและพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วงหรือคำสั่ง exclude ช่วงทางเลือกเป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
subnet	subnet *subnet id netmask ช่วง*			ระบุซับเน็ตและพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วงหรือคำสั่ง exclude ช่วงทางเลือกเป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
subnet	subnet *subnet id netmask เลเบล:ระดับความสำคัญ*			ระบุซับเน็ตและพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วง หรือคำสั่ง exclude ช่วงทางเลือก เป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
subnet	subnet *subnet id netmask ช่วง เลเบล:ระดับความสำคัญ*			ระบุซับเน็ตและพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วง หรือคำสั่ง exclude ช่วงทางเลือก เป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
subnet	subnet *subnet id ช่วง*	ใช่	ไม่มี	ระบุซับเน็ตที่อยู่ภายในคอนเทนเนอร์เครือข่าย กำหนด ช่วงของแอดเดรสที่เป็นซับเน็ตทั้งหมดยกเว้นว่ามีการระบุส่วนช่วง ทางเลือก Netmask ที่เชื่อมโยงกับ subnet นำมาจากคอนเทนเนอร์เครือข่าย ล้อมรอบ **หมายเหตุ:** เมธอดนี้ล้าสมัยแล้วในการสนับสนุนซับเน็ตฟอร์ม

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
option	อ็อพชัน *ตัวเลข ข้อมูล ...*	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณีของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ วลี * deny ทางเลือกหมายถึงอ็อพชันทั้งหมดไม่ระบุในคอนเทนเนอร์ปัจจุบัน ไม่คืนค่าให้กับไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8-บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* or hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วย อ็อพชันอื่นในอ็อพชัน 43
option	อ็อพชัน *ตัวเลข*deny	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณีของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ วลี * deny ทางเลือกหมายถึงอ็อพชันทั้งหมดไม่ระบุในคอนเทนเนอร์ปัจจุบัน ไม่คืนค่าให้กับไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8-บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* or hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วย อ็อพชันอื่นในอ็อพชัน 43
option	อ็อพชัน * deny	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณีของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ วลี * deny ทางเลือกหมายถึงอ็อพชันทั้งหมดไม่ระบุในคอนเทนเนอร์ปัจจุบัน ไม่คืนค่าให้กับไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8-บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* or hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วย อ็อพชันอื่นในอ็อพชัน 43

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
exclude	exclude *IP แอดเดรส*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง exclude คำสั่ง exclude ไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือฐานข้อมูล คำสั่ง exclude ลบแอดเดรสหรือช่วงที่ระบุ ออกจากหน้าปัจจุบันบนคอนเทนเนอร์ คำสั่ง exclude ช่วยให้คุณสามารถสร้างช่วงที่ไม่ต่อเนื่องสำหรับ subnets หรือคอนเทนเนอร์อื่น
exclude	exclude *dotted_quad–dotted_quad*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง exclude คำสั่ง exclude ไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือฐานข้อมูล คำสั่ง exclude ลบแอดเดรสหรือช่วงที่ระบุ ออกจากหน้าปัจจุบันบนคอนเทนเนอร์ คำสั่ง exclude ช่วยให้คุณสามารถสร้างช่วงที่ไม่ต่อเนื่องสำหรับ subnets หรือคอนเทนเนอร์อื่น
range	ช่วง *IP_address*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง ช่วง คำสั่งช่วงไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล ถ้าช่วงเป็นช่วงแรกในคอนเทนเนอร์ที่ไม่ได้ ระบุช่วงบนบรรทัดนิยามคอนเทนเนอร์ ช่วงสำหรับคอนเทนเนอร์ จะกลายเป็นช่วงที่ระบุโดยคำสั่งช่วง คำสั่งช่วงหลังจาก ช่วงแรกหรือคำสั่งช่วงทั้งหมดสำหรับคอนเทนเนอร์ที่ระบุช่วง ในนิยามจะถูกเพิ่มลงในช่วงปัจจุบัน ด้วยคำสั่งช่วง สามารถเพิ่มแอดเดรสหนึ่งหรือชุดของแอดเดรสลงในช่วงได้ ช่วงต้องพอดีภายในนิยามคอนเทนเนอร์ subnet
range	ช่วง *dotted_quad–dotted_quad*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง ช่วง คำสั่งช่วงไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล ถ้าช่วงเป็นช่วงแรกในคอนเทนเนอร์ที่ไม่ได้ ระบุช่วงบนบรรทัดนิยามคอนเทนเนอร์ ช่วงสำหรับคอนเทนเนอร์ จะกลายเป็นช่วงที่ระบุโดยคำสั่งช่วง คำสั่งช่วงหลังจาก ช่วงแรกหรือคำสั่งช่วงทั้งหมดสำหรับคอนเทนเนอร์ที่ระบุช่วง ในนิยามจะถูกเพิ่มลงในช่วงปัจจุบัน ด้วยคำสั่งช่วง สามารถเพิ่มแอดเดรสหนึ่งหรือชุดของแอดเดรสลงในช่วงได้ ช่วงต้องพอดีภายในนิยามคอนเทนเนอร์ subnet

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
client	ไคลเอ็นต์ *hwtype hwaddr* NONE	ใช่	ไม่มี	ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุ โดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์ และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรสของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศ ล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์
client	ไคลเอ็นต์ *hwtype hwaddr* ANY	ใช่	ไม่มี	ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุ โดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์ และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรสของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศ ล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์
client	ไคลเอ็นต์ *hwtype hwaddr* *dotted_quad*	ใช่	ไม่มี	ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุ โดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์ และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรสของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศ ล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
client	ไคลเอ็นต์ *hwtype hwaddr ช่วง*	ใช่	ไม่มี	ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุโดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์ และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรสของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศ ล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์
คลาส	คลาส *สตริง*	ใช่	ไม่มี	ระบุคลาสคอนเทนเนอร์ที่มีชื่อเป็น *สตริง* สตริงอาจอยู่ในอัญประกาศหรือไม่ก็ได้ หากอยู่ในอัญประกาศ อัญประกาศจะถูกลบออกก่อนการเปรียบเทียบ อัญประกาศเป็นสิ่งจำเป็นสำหรับสตริงที่มีช่องว่าง หรือแท็บ คอนเทนเนอร์นี้ถูกต้องที่ทุกระดับ สามารถระบุช่วงเพื่อบ่งชี้ชุดของแอดเดรสที่จะส่งไปยัง ไคลเอ็นต์ที่มีคลาสนี้ ช่วงเป็น IP แอดเดรส quad จุดเดียวหรือ IP แอดเดรส quad สองจุดที่แบ่งด้วยเครื่องหมายขีด
คลาส	คลาส *สตริง ช่วง*	ใช่	ไม่มี	ระบุคลาสคอนเทนเนอร์ที่มีชื่อเป็น *สตริง* สตริงอาจอยู่ในอัญประกาศหรือไม่ก็ได้ หากอยู่ในอัญประกาศ อัญประกาศจะถูกลบออกก่อนการเปรียบเทียบ อัญประกาศเป็นสิ่งจำเป็นสำหรับสตริงที่มีช่องว่าง หรือแท็บ คอนเทนเนอร์นี้ถูกต้องที่ทุกระดับ สามารถระบุช่วงเพื่อบ่งชี้ชุดของแอดเดรสที่จะส่งไปยัง ไคลเอ็นต์ที่มีคลาสนี้ ช่วงเป็น IP แอดเดรส quad จุดเดียวหรือ IP แอดเดรส quad สองจุดที่แบ่งด้วยเครื่องหมายขีด

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
network	เครือข่าย *network id netmask*	ใช่	ไม่มี	ระบุ ID เครือข่ายที่ใช้ข้อมูลคลาส (ตัวอย่างเช่น 9.3.149.0 ที่มี netmask 255.255.255.0 จะเป็นเครือข่าย 9.0.0.0 255.255.255.0) เวอร์ชันนี้ของคอนเทนเนอร์เครือข่ายใช้เพื่อจัดเก็บ subnets ที่มี ID เครือข่าย และ netmask เหมือนกัน เมื่อระบุช่วง แอดเดรสทั้งหมดในช่วงอยู่ในพูล ช่วงต้องอยู่ในเครือข่ายของ ID เครือข่าย รูปแบบนี้ใช้การกำหนดแอดเดรสแบบเต็มของคลาส รูปแบบนี้ถูกต้องในระดับคอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น **หมายเหตุ:** คีย์เวิร์ดเครือข่ายใช้เพื่อสนับสนุน subnet คอนเทนเนอร์
network	เครือข่าย *id เครือข่าย*	ใช่	ไม่มี	ระบุ ID เครือข่ายที่ใช้ข้อมูลคลาส (ตัวอย่างเช่น 9.3.149.0 ที่มี netmask 255.255.255.0 จะเป็นเครือข่าย 9.0.0.0 255.255.255.0) เวอร์ชันนี้ของคอนเทนเนอร์เครือข่ายใช้เพื่อจัดเก็บ subnets ที่มี ID เครือข่าย และ netmask เหมือนกัน เมื่อระบุช่วง แอดเดรสทั้งหมดในช่วงอยู่ในพูล ช่วงต้องอยู่ในเครือข่ายของ ID เครือข่าย รูปแบบนี้ใช้การกำหนดแอดเดรสแบบเต็มของคลาส รูปแบบนี้ถูกต้องในระดับคอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น **หมายเหตุ:** คีย์เวิร์ดเครือข่ายใช้เพื่อสนับสนุน subnet คอนเทนเนอร์
network	เครือข่าย *id เครือข่าย ช่วง*	ใช่	ไม่มี	ระบุ ID เครือข่ายที่ใช้ข้อมูลคลาส (ตัวอย่างเช่น 9.3.149.0 ที่มี netmask 255.255.255.0 จะเป็นเครือข่าย 9.0.0.0 255.255.255.0) เวอร์ชันนี้ของคอนเทนเนอร์เครือข่ายใช้เพื่อจัดเก็บ subnets ที่มี ID เครือข่าย และ netmask เหมือนกัน เมื่อระบุช่วง แอดเดรสทั้งหมดในช่วงอยู่ในพูล ช่วงต้องอยู่ในเครือข่ายของ ID เครือข่าย รูปแบบนี้ใช้การกำหนดแอดเดรสแบบเต็มของคลาส รูปแบบนี้ถูกต้องในระดับคอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น **หมายเหตุ:** คีย์เวิร์ดเครือข่ายใช้เพื่อสนับสนุน subnet คอนเทนเนอร์

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
vendor	ผู้ขาย *vendor_id*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย
vendor	ผู้ขาย *vendor_id* hex*""*			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
vendor	ผู้ขาย *vendor_id* hex*""*			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย
vendor	ผู้ขาย *vendor_id* 0xdata			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
vendor	ผู้ขาย *vendor_id ""*			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย
vendor	ผู้ขาย *vendor_id ช่วง*			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
vendor	ผู้ขาย *vendor_id ช่วง* hex""			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย
vendor	ผู้ขาย *vendor_id ช่วง* hex""			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
vendor	ผู้ขาย *vendor_id ช่วง* 0xdata			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย
vendor	ผู้ขาย *vendor_id ช่วง ""*			ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือก เป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอก อ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
inoption	inoption *ตัวเลข option_data*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเอง ที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุ หมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชัน สำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะ เดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดีจะเป็นสตริงฐานสิบหก ของไบต์ที่ไม่นำหน้าด้วยอักขระ 0x
inoption	inoption *ตัวเลข option_data ช่วง*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเอง ที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุ หมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชัน สำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ — สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม — สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะ เดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดีจะเป็นสตริงฐานสิบหก ของไบต์ที่ไม่นำหน้าด้วยอักขระ 0x

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
virtual	virtual fill *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จากนิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน
virtual	virtual sfill *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จากนิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
virtual	virtual rotate *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จากนิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน
virtual	virtual srotate *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จากนิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน
inorder:	inorder: *id id* ...	ไม่	ไม่มี	ระบุ subnet เสมือนที่มีนโยบาย fill ซึ่งหมายความว่าใช้ แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet ID เหมือนกัน

คีย์เวิร์ด	Form	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
balance:	balance: *id id* ...	ไม่	ไม่มี	ระบุ subnet เสมือนที่มีนโยบาย rotate ซึ่งหมายความว่าใช้ แอดเดรสถัดไปในคอนเทนเนอร์ถัดไป สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบล เป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet ID เหมือนกัน
bootstrapserver	bootstrapserver *IP address*	ไม่	ไม่มี	ระบุเซิร์ฟเวอร์ที่ไคลเอ็นต์ควรจะใช้ไฟล์ **TFTP** หลังจากได้รับแพ็กเก็ต **BOOTP** หรือ **DHCP** ค่านี้กรอกข้อมูลในฟิลด์ **siaddr** ในแพ็กเก็ต ค่านี้ถูกต้อง ที่คอนเทนเนอร์ทุกระดับ
giaddrfield	giaddrfield *IP address*	ไม่	ไม่มี	ระบุ giaddrfield สำหรับแพ็กเก็ตการตอบกลับ **หมายเหตุ:** การระบุนี้ไม่ถูกต้องในโปรโตคอล **BOOTP** และ **DHCP** แต่บางไคลเอ็นต์ต้องการฟิลด์ **giaddr** เป็นดีฟอลต์เกตเวย์สำหรับเครือข่าย เนื่องจากความขัดแย้งที่อาจเกิดขึ้นได้นี้ จึงควรใช้ **giaddrfield** ภายในไคลเอ็นต์คอนเทนเนอร์ แม้ว่าสามารถทำงานได้ในทุกระดับ
bootfile	bootfile *path*	ไม่	ไม่มี	ระบุ bootfile ที่จะใช้ในส่วนไฟล์ของแพ็กเก็ตการตอบกลับ ค่านี้สามารถระบุที่คอนเทนเนอร์ทุกระดับ นโยบาย bootfile กำหนดจำนวนไอเท็มที่ระบุในส่วนไฟล์ของแพ็กเก็ตขาเข้าที่โต้ตอบกับ bootfile และคำสั่งไดเร็กทอรีโฮม
pxebootfile	pxebootfile *System Arch MajorVer MinorVer Bootfilename*	ไม่	ไม่มี	ระบุ bootfile ที่กำหนดให้กับไคลเอ็นต์ ตัววิเคราะห์ไฟล์คอนฟิก สร้างข้อผิดพลาดถ้าจำนวนพารามิเตอร์หลังจากคีย์เวิร์ดน้อยกว่า 4 และละเว้นถ้ามากกว่า 4 คีย์เวิร์ดนี้สามารถใช้ได้ในคอนเทนเนอร์เท่านั้น

สำหรับรายละเอียดเกี่ยวกับอ็อพชันอื่น โปรดดู “อ็อพชันที่รู้จักของไฟล์เซิร์ฟเวอร์ DHCP” ในหน้า 249 และ “อ็อพชันย่อยของคอนเทนเนอร์ผู้ขาย preboot execution environment” ในหน้า 251

Boot Image Negotiation Layer daemon

เซิร์ฟเวอร์ Boot Image Negotiation Layer daemon (BINLD) เป็น ระยะที่สามของการติดต่อสำหรับไคลเอ็นต์ preboot execution environment (PXE)

หลังการสื่อสารกับเซิร์ฟเวอร์ DHCP เพื่อให้ได้ IP แอดเดรส และหลังจาก สื่อสารกับเซิร์ฟเวอร์ PXE Proxy DHCP เพื่อให้ได้ที่ตั้งของ เซิร์ฟเวอร์บูตแล้ว จะมีการติดต่อกับเซิร์ฟเวอร์บูตเพื่อให้ได้รับชื่อไฟล์และที่ตั้ง ซึ่งจะดาวน์โหลดรูปภาพบูต ไคลเอ็นต์ PXE สามารถกลับไปเพื่อสื่อสาร กับเซิร์ฟเวอร์บูตได้หลายครั้งในระหว่างการบูตถ้าไคลเอ็นต์ ต้องการหลายไฟล์ในบูตโปรเซส

ระยะสุดท้ายในบูตเครือข่าย PXE คือการดาวน์โหลดรูปภาพบูตที่กำหนด โดยเซิร์ฟเวอร์บูต ที่ตั้งของเซิร์ฟเวอร์ TFTP และชื่อไฟล์ที่จะ ถูกดาวน์โหลดมีการกำหนดโดยเซิร์ฟเวอร์บูตให้กับไคลเอ็นต์ PXE

BINLD เซิร์ฟเวอร์คอมโพเนนต์

คอมโพเนนต์หลักสามคอมโพเนนต์ของเซิร์ฟเวอร์ BINLD ถูกแนะนำที่นี่

เซิร์ฟเวอร์ BINLD ถูกแบ่งเซ็กเมนต์เป็นสามส่วนหลักคือ ฐานข้อมูล เอ็นจินโปรโตคอล และชุดของเซอร์วิสเธรด โดยแต่ละส่วนมีข้อมูลคอนฟิกูเรชัน ของตนเอง

ฐานข้อมูล BINLD:

ฐานข้อมูล `db_file.dhcpo` ถูกใช้เพื่อสร้าง อ็อพชันที่ตอบสนองแพ็กเก็ต REQUEST ของไคลเอ็นต์

อ็อพชันคืนกลับตามฐานข้อมูลขึ้นอยู่กับชนิดของเซิร์ฟเวอร์ที่เลือก อ็อพชันถูกตั้งค่าโดยใช้คีย์เวิร์ด **pxeservertype** ใน ไฟล์ `binld.cnf`

การใช้ข้อมูลในไฟล์คอนฟิกูเรชัน ฐานข้อมูลถูกเตรียมพร้อม และตรวจสอบความสอดคล้อง

โปรโตคอลเอ็นจิน BINLD:

โปรโตคอลเอ็นจินใช้ฐานข้อมูลเพื่อกำหนดสิ่งที่ข้อมูลควร คืนกลับให้ไคลเอ็นต์

เอ็นจินโปรโตคอล PXED อ้างอิงตาม Intel Preboot Execution Environment (PXE) Specification เวอร์ชัน 2.1 แต่ยังคงทำงานร่วมกันได้กับ Intel PXE Specification เวอร์ชัน 1.1

การดำเนินงาน BINLD ที่เธรด:

ชิ้นส่วนสุดท้ายของเซิร์ฟเวอร์ BINLD แท้จริงแล้วคือชุดของการดำเนินงาน ที่ใช้เพื่อรักษาให้สิ่งต่างๆ รันต่อไป

เนื่องจากเซิร์ฟเวอร์ BINLD มีการเธรด การดำเนินงานเหล่านี้จึงมีการตั้งค่าจริง เป็นเธรดที่จะทำสิ่งต่างๆ ในบางโอกาสเพื่อทำให้แน่ใจว่าทุกสิ่ง เข้ากันได้ดี

เธรดแรกซึ่งเป็นเธรด *หลัก* จัดการกับคำร้องขอ SRC (เช่น **startsrc, stopsrc, lssrc, traceson,** และ **refresh**) เธรดนี้ยังประสานการดำเนินงานทั้งหมดที่มีผลต่อเธรดทั้งหมดและจัดการกับ สัญญาณด้วย ตัวอย่างเช่น

- SIGHUP (-1) ส่งผลให้รีเฟรชฐานข้อมูลทั้งหมดในไฟล์คอนฟิกูเรชัน
- SIGTERM (-15) ส่งผลให้เซิร์ฟเวอร์หยุดอย่างเรียบร้อย

เธรดอื่นประมวลผลแพ็กเก็ต ขึ้นอยู่กับชนิดเซิร์ฟเวอร์ อาจมี หนึ่งหรือสองเธรด เธรดหนึ่งรับฟังบนพอร์ต 67 และเธรดที่สองรับฟังบนพอร์ต 4011 แต่ละเธรดสามารถจัดการกับคำร้องขอจากไคลเอ็นต์

คอนฟิกูเรชัน BINLD

โดยค่าดีฟอลต์ เซิร์ฟเวอร์ BINLD มีการกำหนดคอนฟิกโดยการอ่านไฟล์ /etc/binld.cnf ซึ่งระบุฐานข้อมูลแรกเริ่มของเซิร์ฟเวอร์ของอ็อพชันและแอดเดรส

เซิร์ฟเวอร์เริ่มทำงานจาก SMIT หรือผ่านคำสั่ง SRC

โดยปกติ การกำหนดคอนฟิกเซิร์ฟเวอร์ BINLD เป็นส่วนที่ยากที่สุดของการใช้ BINLD ในเครือข่าย อันดับแรก พิจารณาว่าคุณต้องการให้ไคลเอ็นต์ PXE อยู่บนเครือข่ายใด ตัวอย่างต่อไปนี้กำหนดคอนฟิกเซิร์ฟเวอร์ BINLD เพื่อรันบนเครื่องเดียวกันกับ เซิร์ฟเวอร์ DHCP:

```
pxeservertype       binld_on_dhcp_server

subnet default
{
    vendor pxe
    {
                bootstrapserver  9.3.149.6      #TFTP server IP address
                pxebootfile   1   2   1   window.one   1   0
                pxebootfile   2   2   1   linux.one    2   3
                pxebootfile   1   2   1   hello.one    3   4
                client 6 10005a8ad14d any
                {
                   pxebootfile   1   2   1   aix.one     5   6
                   pxebootfile   2   2   1   window.one  6   7
       }
     }
}
```

จากคอนฟิกูเรชันข้างบน เซิร์ฟเวอร์ BINLD รับฟัง unicast แพ็กเก็ตของไคลเอ็นต์บนพอร์ต 4011 และ Multicast แพ็กเก็ตบนพอร์ต 4011 ถ้า BINLD ได้รับ Multicast แอดเดรสจาก dhcpsd/pxed เซิร์ฟเวอร์ BINLD ตอบกลับไปยังไคลเอ็นต์ REQUEST/INFORM แพ็กเก็ตที่มีชื่อ bootfile และ IP แอดเดรสของเซิร์ฟเวอร์ TFTP หาก BINLD ไม่พบ bootfile ที่มีชั้นตรงกันกับที่ระบุโดย ไคลเอ็นต์ เซิร์ฟเวอร์จะพยายามค้นหา bootfile สำหรับชั้นถัดไป BINLD ไม่ ตอบกลับเมื่อไม่มี boot file ที่ตรงกับความต้องการของไคลเอ็นต์ (*Type*, *SystemArch*, *MajorVers*, *MinorVers*, และ *Layer*)

ตัวอย่างต่อไปนี้กำหนดคอนฟิก BINLD เพื่อรันบนเครื่องแยกต่างหาก (นั่น คือ DHCP / PXED ไม่ได้กำลังรันบนเครื่องเดียวกัน)

```
subnet 9.3.149.0 255.255.255.0
{

     vendor pxe
     {

     bootstrapserver        9.3.149.6      # TFTP server ip address.
     pxebootfile    1    2    1   window.one    1   0
     pxebootfile    2    2    1   linux.one     2   3
     pxebootfile    1    2    1   hello.one     3   4
     client 6 10005a8ad14d any
       {
       pxebootfile    1    2    1   aix.one      5   6
```

```
        pxebootfile    2    2    1   window.one   6   7
        }
      }
}
```

ในตัวอย่างข้างบน ไม่มีการตั้งค่า *pxeservertype* ดังนั้นชนิดเซิร์ฟเวอร์ดีฟอลต์คือ **binld_only** เซิร์ฟเวอร์ BINLD รับฟัง unicast แพ็กเก็ตของไคลเอ็นต์บนพอร์ต 4011, broadcast & unicast แพ็กเก็ตบนพอร์ต 67, และ Multicast แพ็กเก็ตบนพอร์ต 4011 ถ้า BINLD ได้รับ Multicast แอดเดรสจาก dhcpsd/pxed ชื่อ bootfile และ IP แอดเดรสของเซิร์ฟเวอร์ TFTP ถูกส่งไปยังไคลเอ็นต์ PXE เฉพาะถ้า IP แอดเดรสของไคลเอ็นต์อยู่ในช่วง IP แอดเดรสของ subnet เท่านั้น (9.3.149.0 ถึง 9.3.149.255)

ตัวอย่างต่อไปนี้กำหนดคอนฟิก BINLD เพื่อรันบนเครื่องเดียวกันกับ เซิร์ฟเวอร์ PXED:

```
pxeservertype       binld_on_proxy_server
subnet default
{
     vendor
     {
     bootstrapserver        9.3.149.6          # TFTP server ip address.
     pxebootfile    1    2    1   window.one    1    0
     pxebootfile    2    2    1   linux.one     2    3
     pxebootfile    1    2    1   hello.one     3    4
     client 6 10005a8ad14d any
       {
       pxebootfile    1    2    1   aix.one     5   6
       pxebootfile    2    2    1   window.one  6   7
       }
     }
}
```

ในคอนฟิกูเรชันนี้ เซิร์ฟเวอร์ BINLD รับฟังเฉพาะบนพอร์ต 4011 สำหรับ Multicast แพ็กเก็ตเฉพาะถ้า BINLD ได้รับ Multicast แอดเดรสจาก dhcpsd/pxed เท่านั้น หากไม่ได้รับ multicast แอดเดรสใดๆ BINLD จะออกและมีการบันทึกข้อความแสดง ข้อผิดพลาดลงในไฟล์บันทึก

ส่วนคำสั่ง `db_file` ของฐานข้อมูลบ่งชี้เมธอดฐานข้อมูล ที่จะใช้สำหรับการประมวลผลไฟล์คอนฟิกูเรชันส่วนนี้ ข้อคิดเห็นขึ้นต้น ด้วยเครื่องหมายสี่เหลี่ยม (#) ตั้งแต่ # ถึงตอนท้ายของบรรทัดถูกละเว้นโดยเซิร์ฟเวอร์ PXED เซิร์ฟเวอร์ใช้แต่ละบรรทัด `option` เพื่อบอกสิ่งที่ ไคลเอ็นต์ต้องทำ “อ็อพชันย่อยของคอนเทนเนอร์ผู้ผลิต PXE” ในหน้า 332 อธิบาย อ็อพชันย่อยที่ได้รับการสนับสนุนและรู้จักในปัจจุบัน โปรดดูที่ “ไวยากรณ์ไฟล์เซิร์ฟเวอร์ BINLD สำหรับการดำเนินงานเซิร์ฟเวอร์ทั่วไป” ในหน้า 358 สำหรับ วิธีการระบุอ็อพชันที่เซิร์ฟเวอร์ไม่ทราบ

ไฟล์คอนฟิกูเรชัน BINLD:

ไฟล์คอนฟิกูเรชันมีส่วนแอดเดรสและส่วนนิยามอ็อพชัน ซึ่งสร้างขึ้นจากแนวคิดของคอนเทนเนอร์ที่จัดเก็บอ็อพชัน ตัวแก้ไข และอาจมีคอนเทนเนอร์อื่น

คอนเทนเนอร์ (โดยพื้นฐาน เมธอดการจัดกลุ่มอ็อพชัน) ใช้ตัวระบุเพื่อจัดประเภทไคลเอ็นต์เป็นกลุ่มต่างๆ ชนิดคอนเทนเนอร์คือ subnet, คลาส, ผู้ขาย, และไคลเอ็นต์ ในปัจจุบัน ไม่มีคอนเทนเนอร์ทั่วไปซึ่งผู้ใช้สามารถ กำหนดได้ ตัวระบุกำหนดไคลเอ็นต์โดยไม่ซ้ำกันเพื่อให้สามารถติดตามไคลเอ็นต์ ได้ถ้า ตัวอย่างเช่น ย้ายระหว่าง subnets สามารถใช้คอนเทนเนอร์ได้ มากกว่าหนึ่งชนิดเพื่อกำหนดการเข้าถึงไคลเอ็นต์

อ็อพชัน คือตัวระบุที่ถูกส่งคืนไปยังไคลเอ็นต์ เช่น ดีฟอลต์เกตเวย์และ DNS แอดเดรส

คอนเทนเนอร์ BINLD:

เมื่อเซิร์ฟเวอร์ DHCP ได้รับคำร้องขอ จะมีการแจงส่วนแพ็กเก็ตและ คีย์การระบุกำหนดคอนเทนเนอร์ อ็อพชัน และแอดเดรสที่จะแยก

ตัวอย่างล่าสุดใน คอนฟิกูเรชัน BINLD แสดงคอนเทนเนอร์ subnet คีย์การระบุคือตำแหน่งไคลเอ็นต์ในเครือข่าย หากไคลเอ็นต์มาจากเครือข่ายนั้น ไคลเอ็นต์จะอยู่ในคอนเทนเนอร์นั้น

คอนเทนเนอร์แต่ละชนิดใช้อ็อพชันที่แตกต่างกันในการระบุไคลเอ็นต์:

- Subnet คอนเทนเนอร์ใช้ฟิลด์ giaddr หรืออินเตอร์เฟสแอดเดรสของ อินเตอร์เฟสที่ได้รับเพื่อกำหนด subnet ต้นทางของไคลเอ็นต์
- คลาสคอนเทนเนอร์ใช้ค่าในอ็อพชัน 77 (ตัวระบุคลาสของไซต์ผู้ใช้)
- ผู้ขายใช้ค่าในอ็อพชัน 60 (ตัวระบุคลาสของผู้ขาย)
- ไคลเอ็นต์คอนเทนเนอร์ใช้อ็อพชัน 61 (ตัวระบุไคลเอ็นต์) สำหรับไคลเอ็นต์ PXED และฟิลด์ chaddr ในแพ็กเก็ต BOOTP สำหรับไคลเอ็นต์ BOOTP

ยกเว้นสำหรับ subnets แต่ละคอนเทนเนอร์อนุญาตการระบุค่า ที่ตรงกัน รวมถึงการจับคู่นิพจน์ปกติ

และยังมีคอนเทนเนอร์โดยปริยายคือ คอนเทนเนอร์ *สากล* อ็อพชัน และตัวแก้ไขที่วางไว้ในคอนเทนเนอร์สากลใช้กับคอนเทนเนอร์ทั้งหมด ยกเว้น ว่าถูกยกเลิกหรือปฏิเสธ คอนเทนเนอร์ส่วนใหญ่สามารถวางไว้ภายในคอนเทนเนอร์อื่น ที่ใช้ขอบเขตของการมองเห็นได้ คอนเทนเนอร์อาจหรืออาจไม่มีช่วงแอดเดรส ที่เชื่อมโยง Subnets โดยธรรมชาติแล้ว มีช่วงที่เชื่อมโยง ด้วย

กฎพื้นฐานสำหรับคอนเทนเนอร์และคอนเทนเนอร์ย่อยมีดังนี้:

- คอนเทนเนอร์ทั้งหมดถูกต้องที่ระดับสากล
- Subnets ไม่สามารถวางไว้ภายในคอนเทนเนอร์อื่น
- คอนเทนเนอร์ที่จำกัดไม่สามารถมีคอนเทนเนอร์ปกติชนิดเดียวกัน อยู่ภายใน (ตัวอย่างเช่น คอนเทนเนอร์ที่มีอ็อพชันที่อนุญาตเฉพาะคลาส `Accounting` ไม่สามารถมีคอนเทนเนอร์ที่มีอ็อพชันซึ่ง อนุญาตทุกคลาสที่ขึ้นต้นด้วยตัวอักษร "a" นี่ไม่ถูกต้อง)
- ไคลเอ็นต์คอนเทนเนอร์ที่จำกัดไม่สามารถมีคอนเทนเนอร์ย่อย

ภายใต้กฎข้างบน คุณสามารถสร้างลำดับชั้นของคอนเทนเนอร์ที่ แบ่งเซกเมนต์อ็อพชันของคุณออกเป็นกลุ่มต่างๆ สำหรับไคลเอ็นต์หรือชุดของไคลเอ็นต์เฉพาะ

หากไคลเอ็นต์ตรงกับหลายคอนเทนเนอร์ จะจัดการกับอ็อพชันและแอดเดรส อย่างไร? เซิร์ฟเวอร์ DHCP ได้รับข้อความ ส่งผ่านคำร้องขอไปยัง ฐานข้อมูล (`db_file` ในกรณีนี้) และมีการสร้างรายการ คอนเทนเนอร์ขึ้น รายการแสดงขึ้นในลำดับของความลึกและระดับความสำคัญ ระดับความสำคัญ มีการกำหนดเป็นลำดับชั้นปริยายในคอนเทนเนอร์ คอนเทนเนอร์จำกัดมีระดับ ความสำคัญสูงกว่าคอนเทนเนอร์ปกติ ไคลเอ็นต์ คลาส ผู้ขาย และสุดท้าย subnets มีการเรียงลำดับในลำดับนั้น และภายในคอนเทนเนอร์ชนิดเดียวกันมีการเรียงตามความลึก ซึ่ง สร้างรายการที่เรียงลำดับตามข้อมูลเฉพาะมากที่สุดไปยังน้อยที่สุด ตัวอย่างเช่น:

```
Subnet 1
--Class 1
--Client 1
Subnet 2
--Class 1
----Vendor 1
----Client 1
--Client 1
```

ตัวอย่างแสดงสอง subnets, Subnet 1 และ Subnet 2 มีหนึ่งชื่อคลาส, Class 1, หนึ่งชื่อผู้ขาย, Vendor 1, และหนึ่งชื่อไคลเอ็นต์, Client 1 Class 1 และ Client 1 มีการกำหนดในหลายที่ เนื่องจาก อยู่ในคอนเทนเนอร์ที่แตกต่างกัน ชื่ออาจเหมือนกันได้แต่ค่าภายใน อาจแตกต่างกัน หาก Client 1 ส่งข้อความไปยังเซิร์ฟเวอร์ DHCP จาก Subnet 1 โดยมีการระบุ Class 1 ใน รายการอ็อพชัน เซิร์ฟเวอร์ DHCP อาจสร้างพาธคอนเทนเนอร์ ต่อไปนี้:

```
Subnet 1, Class 1, Client 1
```

คอนเทนเนอร์เฉพาะที่สุดแสดงอยู่ในลำดับสุดท้าย เพื่อให้ได้แอดเดรส รายการ จะถูกตรวจสอบในลำดับชั้นย้อนกลับเพื่อค้นหาแอดเดรสแรกที่มีอยู่ จากนั้น จะตรวจสอบรายการในลำดับชั้นไปข้างหน้าเพื่อให้ได้อ็อพชัน อ็อพชันยกเลิก ค่าก่อนหน้านี้ยกเว้นว่ามีอ็อพชัน *deny* อยู่ในคอนเทนเนอร์ นอกจากนี้ เนื่องจาก Class 1 และ Client 1 อยู่ใน Subnet 1 จะมีการเรียงลำดับตามระดับความสำคัญของคอนเทนเนอร์ หากไคลเอ็นต์ เดียวกันอยู่ใน Subnet 2 และส่งข้อความเดียวกัน รายการคอนเทนเนอร์ ที่สร้างขึ้นคือ:

```
Subnet 2, Class 1, Client 1 (ที่ระดับ Subnet 2), Client 1 (ที่ระดับ Class 1)
```

Subnet 2 แสดงขึ้นก่อน ตามด้วย Class 1 ตามด้วย Client 1 ที่ระดับ Subnet 2 (เนื่องจากคำสั่งไคลเอ็นต์นี้ต่ำลงเพียงหนึ่งระดับในลำดับชั้น) ลำดับชั้น ตีความว่าไคลเอ็นต์ที่ตรงกับคำสั่งไคลเอ็นต์แรกมีความเฉพาะ น้อยกว่าไคลเอ็นต์ที่ตรงกับ Client 1 ของ Class 1 ภายใน Subnet 2

ระดับความสำคัญที่เลือกโดยความลึกภายในลำดับชั้นไม่ได้ถูกแทนที่โดย ระดับความสำคัญของตัวคอนเทนเนอร์เอง ตัวอย่าง เช่น ถ้าไคลเอ็นต์เดียวกันออกใช้ข้อความเดียวกันและระบุตัวระบุผู้ขาย รายการคอนเทนเนอร์จะเป็น:

```
Subnet 2, Class 1, Vendor 1, Client 1 (ที่ระดับ Subnet 2), Client 1 (ที่ระดับ Class 1)
```

ระดับความสำคัญคอนเทนเนอร์ช่วยพัฒนาประสิทธิภาพการค้นหา เนื่องจาก เป็นไปตามแนวคิดทั่วไปที่ไคลเอ็นต์คอนเทนเนอร์เป็นวิธีเฉพาะที่สุดในการกำหนด หนึ่งไคลเอ็นต์ขึ้นไป คลาสคอนเทนเนอร์มีแอดเดรสเฉพาะน้อยกว่าไคลเอ็นต์ คอนเทนเนอร์ ผู้ขายมีความเฉพาะน้อยไปอีก และ subnet มีความเฉพาะน้อยที่สุด

BINLD แอดเดรสและช่วงแอดเดรส:

ชนิดคอนเทนเนอร์ต่างๆ อาจมีช่วงแอดเดรสที่เชื่อมโยง และ subnets ต้องมีช่วงแอดเดรสที่เชื่อมโยง

แต่ละช่วงภายในคอนเทนเนอร์ต้องเป็นชุดย่อยของช่วงของพาเรนต์ คอนเทนเนอร์ และต้องไม่ซ้อนเหลื่อมกับช่วงของคอนเทนเนอร์อื่น ตัวอย่างเช่น ถ้า คลาสถูกกำหนดไว้ภายใน subnet และคลาสมีช่วง ช่วงต้องเป็น ชุดย่อยของช่วงของ subnet นอกจากนี้ ช่วงภายในคลาสคอนเทนเนอร์นั้น ต้องไม่ซ้อนเหลื่อมกับช่วงอื่นใดๆ ที่ระดับ

ช่วงสามารถระบุได้บนบรรทัดคอนเทนเนอร์และแก้ไขโดยใช้ช่วง และคำสั่ง exclude เพื่อให้สามารถแยกชุดแอดเดรสที่เชื่อมโยงกับคอนเทนเนอร์ได้ ดังนั้น ถ้าคุณมีแอดเดรสสิบรายการแรกและสิบรายการที่สองของ subnet อยู่ subnet สามารถระบุแอดเดรสเหล่านี้โดยใช้ช่วงในส่วนคำสั่ง subnet เพื่อลดทั้งการใช้หน่วยความจำและโอกาสการปะทะของแอดเดรสกับ ไคลเอ็นต์อื่นที่ไม่ได้อยู่ในช่วงที่ระบุ

เมื่อเลือกแอดเดรสแล้ว คอนเทนเนอร์ในลำดับต่อมาใดๆ ในรายการ ที่มีช่วงแอดเดรสจะถูกลบออกจากรายการพร้อมกับชายด์ เหตุผลที่เป็นเช่นนี้คือ อ็อพชันเฉพาะเครือข่ายในคอนเทนเนอร์ที่ลบออก ไม่ถูกต้องถ้าไม่ได้ใช้แอดเดรสจากภายในคอนเทนเนอร์นั้น

อ็อพชันไฟล์คอนฟิกูเรชัน BINLD:

หลังจากรายการถูกเลือกเพื่อกำหนดแอดเดรส ชุดของอ็อพชัน จะถูกสร้างขึ้นสำหรับไคลเอ็นต์

ในขั้นตอนการเลือกนี้ อ็อพชันจะเขียนทับอ็อพชันที่เลือกก่อนหน้า จนกว่าจะตรวจพบ *ปฏิเสธ* ซึ่งในกรณีดังกล่าว อ็อพชันที่ปฏิเสธจะถูกลบ ออกจากรายการที่ส่งให้กับไคลเอ็นต์ วิธีนี้อนุญาตให้สืบทอดจาก คอนเทนเนอร์หลักเพื่อลดปริมาณข้อมูลที่ต้องระบุ

การบันทึก BINLD:

พารามิเตอร์การบันทึกมีการระบุในคอนเทนเนอร์เช่นเดียวกับฐานข้อมูล แต่คอนเทนเนอร์คีย์เวิร์ดคือ **logging_info**

เมื่อศึกษาเพื่อกำหนดคอนฟิก PXED ขอแนะนำให้เปิดการบันทึกเป็น ระดับสูงสุด นอกจากนี้ สิ่งที่ดีที่สุดคือการระบุคอนฟิกูเรชันการบันทึกก่อนหน้า ข้อมูลไฟล์คอนฟิกูเรชันอื่นใด เพื่อให้มั่นใจว่าจะมีการบันทึกข้อผิดพลาด คอนฟิกูเรชันหลังจากเริ่มต้นระบบย่อยการบันทึกแล้ว ใช้คีย์เวิร์ด **logitem** เพื่อ เปิดระดับการบันทึกหรือลบคีย์เวิร์ด **logitem** เพื่อ ปิดใช้งานระดับการบันทึก คีย์เวิร์ดอื่นสำหรับการบันทึกช่วยให้สามารถระบุ ชื่อไฟล์บันทึก ขนาดไฟล์ และจำนวนของไฟล์บันทึกที่หมุนเวียน

ข้อควรพิจารณาเกี่ยวกับประสิทธิภาพ BINLD:

สิ่งสำคัญคือการทำความเข้าใจว่าคอนฟิกูเรชันคีย์เวิร์ดต่างๆ และโครงสร้างของไฟล์คอนฟิกูเรชันมีผลกระทบต่อการใช้หน่วยความจำ และประสิทธิภาพของเซิร์ฟเวอร์ PXED อย่างไร

อันดับแรก การใช้หน่วยความจำที่มากเกินความจำเป็นสามารถหลีกเลี่ยงได้โดยการ ทำความเข้าใจกับโมเดลการสืบทอดของอ็อพชันจากคอนเทนเนอร์พาเรนต์ไปยังชายด์ในสภาวะแวดล้อมซึ่งสนับสนุน ไคลเอ็นต์ที่ไม่มีการแสดงรายการ ผู้ดูแลระบบต้องแสดงรายการแต่ละไคลเอ็นต์ใน ไฟล์อย่างชัดแจ้ง เมื่อแสดงรายการอ็อพชันสำหรับไคลเอ็นต์เฉพาะ เซิร์ฟเวอร์จะใช้ หน่วยความจำเพื่อจัดเก็บแผนผังคอนฟิกูเรชันนั้นมากกว่าเมื่อได้รับสืบทอดอ็อพชัน จากคอนเทนเนอร์พาเรนต์ (ตัวอย่างเช่น คอนเทนเนอร์ subnet, เครือข่าย, หรือสากล) ดังนั้น ผู้ดูแลระบบจึงควรตรวจสอบว่าอ็อพชันใดมีการทำซ้ำ ที่ระดับไคลเอ็นต์ภายในไฟล์คอนฟิกูเรชันหรือไม่ และถ้ามี ควรพิจารณาว่า สามารถระบุอ็อพชันดังกล่าวในคอนเทนเนอร์พาเรนต์และแบ่งใช้โดยชุด ของไคลเอ็นต์โดยรวมได้หรือไม่

นอกจากนี้ เมื่อใช้รายการ **logItem** INFO และ TRACE จะมีการบันทึกข้อความเป็นจำนวนมากในระหว่างการประมวลผลทุกข้อความของไคลเอ็นต์ PXE การผนวกบรรทัดเข้ากับไฟล์บันทึกอาจเป็นการดำเนินงานที่มีค่าใช้จ่ายสูง ด้วยเหตุนี้ การจำกัดจำนวนของการบันทึกจึงช่วยปรับปรุงประสิทธิภาพของเซิร์ฟเวอร์ PXED ได้ เมื่อข้อผิดพลาดกับเซิร์ฟเวอร์ PXED ถูกพัก การล็อกสามารถเปิดใช้งานใหม่แบบไดนามิก ได้โดยใช้คำสั่ง SRC traceson

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ BINLD สำหรับการดำเนินงานเซิร์ฟเวอร์ทั่วไป

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ BINLD สำหรับการดำเนินงานเซิร์ฟเวอร์ทั่วไป มีการอธิบายที่นี่ มีการระบุแบบฟอร์ม คอนเทนเนอร์ย่อย ค่าดีฟอลต์ และความหมาย

หมายเหตุ: หน่วยเวลา (*time_units*) ที่แสดงในตารางต่อไปนี้ เป็นทางเลือก และแสดงถึงตัวแก้ไขเวลาจริง หน่วยเวลาดีฟอลต์คือนาที ค่าที่ถูกต้องคือวินาที (1), นาที (60), ชั่วโมง (3600), วัน (86400), สัปดาห์ (604800), เดือน (2392000), และปี (31536000) ตัวเลข ที่แสดงในวงเล็บเป็นตัวคูณที่ใช้กับค่า *n* ที่ระบุ เพื่อระบุค่าในหน่วยวินาที

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ฐานข้อมูล	ฐานข้อมูล *db type*	ใช่	ไม่มี	คอนเทนเนอร์หลักที่จัดเก็บนิยามสำหรับแอดเดรสพูล อ็อพชัน และคำสั่งการเข้าถึงไคลเอ็นต์ *db type* คือ ชื่อของโมดูลที่ถูกโหลดเพื่อประมวลผลส่วนนี้ของไฟล์ ค่าเดียว ที่มีอยู่ในปัจจุบันคือ **db_file**
logging_info	logging_info	ใช่	ไม่มี	คอนเทนเนอร์การบันทึกหลักที่กำหนดพารามิเตอร์การบันทึก
logitem	logitem NONE	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem SYSERR	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem OBJERR	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem PROTOCOL	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem PROTERR	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem WARN	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem WARNING	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem CONFIG	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem EVENT	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem PARSEERR	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem ACTION	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem ACNTING	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem STAT	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
logitem	logitem TRACE	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem RTRACE	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
logitem	logitem START	ไม่ใช่	ค่าดีฟอลต์ทั้งหมดเป็นไม่เปิดใช้งาน	เปิดใช้งานระดับการบันทึก ใช้ได้หลายบรรทัด
numLogFiles	numLogFiles *n*	ไม่ใช่	0	ระบุจำนวนของไฟล์บันทึกที่จะสร้าง บันทึกจะหมุนเวียนเมื่อกรอกข้อมูลบันทึกแรก *n* คือจำนวนของไฟล์ที่จะสร้าง
logFileSize	logFileSize *n*	ไม่ใช่	0	ระบุขนาดของแต่ละไฟล์บันทึกในหน่วย 1024-ไบต์
logFileName	logFileName *พาธ*	ไม่ใช่	ไม่มี	ระบุพาธไปยังไฟล์บันทึกแรก ไฟล์บันทึกดั้งเดิมมี ชื่อว่า *filename* หรือ *filename.extension* เมื่อหมุนเวียนไฟล์ ไฟล์จะถูกเปลี่ยนชื่อโดยเริ่มต้นด้วยฐาน *filename* โดยการผนวกตัวเลขหรือการแทนที่นามสกุลด้วยตัวเลข ตัวอย่าง เช่น ถ้าชื่อไฟล์ดั้งเดิมเป็น `file` ชื่อของไฟล์ที่หมุนเวียน จะกลายเป็น `file01` ถ้าชื่อไฟล์ดั้งเดิมเป็น `file.log` จะกลายเป็น `file.01`
pxeservertype	pxeservertype *servertype*	ไม่ใช่	dhcp_only	บ่งชี้ชนิดของเซิร์ฟเวอร์ dhcpsd *servertype* สามารถ เป็นอย่างใดอย่างหนึ่งต่อไปนี้ **binld_on_dhcp_server** นี่หมายความว่า **BINLD** กำลังรันบนเครื่องเดียวกันกับ เซิร์ฟเวอร์ DHCP และกำลังรับฟังคำร้องขอของ PXE Client บนพอร์ต 4011 และ Multicast แอดเดรสถ้าได้รับจาก DHCP / PXED **binld_on_proxy_server** นี่ หมายความว่า **BINLD** กำลังรันบนเครื่องเดียวกันกับเซิร์ฟเวอร์ PXED และกำลังรับฟังคำร้องขอของ PXE Client บน Multicast แอดเดรส ถ้าได้รับจาก DHCP / PXED ค่าดีฟอลต์คือ **binld_only** ซึ่งหมายความว่า BINLD กำลังรันบนเครื่องแยกต่างหาก และต้องรับฟัง แพ็กเก็ตของไคลเอ็นต์บนพอร์ต 67, 4011 และ Multicast แอดเดรสถ้าได้รับ จาก DHCP / PXED
dhcp_or_proxy _address	dhcp_or_proxy_address *IP แอดเดรส*	ไม่ใช่	ไม่มี	ให้ IP แอดเดรสของเซิร์ฟเวอร์ dhcp หรือ pxed ซึ่งเซิร์ฟเวอร์ BINLD สามารถส่ง Unicast แพ็กเก็ตชนิด REQUEST / INFORM เพื่อรับ Multicast แอดเดรส คีย์เวิร์ดนี้มีการกำหนดเฉพาะถ้า dhcp หรือ pxed อยู่บน subnet ที่แตกต่างจาก BINLD

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ BINLD สำหรับฐานข้อมูล db_file

ไวยากรณ์ไฟล์เซิร์ฟเวอร์ BINLD สำหรับฐานข้อมูล `db_file` มีการอธิบายที่นี่ มีการระบุแบบฟอร์ม คอนเทนเนอร์ย่อย ค่าดีฟอลต์ และความหมาย

หมายเหตุ:

1. หน่วยเวลา (*time_units*) ที่แสดงในตารางต่อไปนี้ เป็นทางเลือก และแสดงถึงตัวแก้ไขเวลาจริง หน่วยเวลา ดีฟอลต์คือ นาที ค่าที่ถูกต้องคือวินาที (1), นาที (60), ชั่วโมง (3600), วัน (86400), สัปดาห์ (604800), เดือน (2392000), และปี (31536000) ตัวเลข ที่แสดงในวงเล็บเป็นตัวคูณที่ใช้กับค่า *n* ที่ระบุ เพื่อระบุค่าในหน่วยวินาที
2. ไอเท็มที่ระบุในคอนเทนเนอร์หนึ่งสามารถถูกยกเลิกภายในคอนเทนเนอร์ย่อย ตัวอย่างเช่น คุณสามารถกำหนดไคลเอ็นต์ BOOTP แบบสากล แต่ภายในบาง subnet อนุญาตไคลเอ็นต์ BOOTP โดยระบุคีย์เวิร์ด supportBootp ในทั้งสอง คอนเทนเนอร์
3. ไคลเอ็นต์ คลาส และคอนเทนเนอร์ผู้ขายอนุญาตการสนับสนุนนิพจน์ ปกติ สำหรับคลาสและผู้ขาย สตริงที่อยู่ในอัญประกาศที่มีอักขระตัวแรกหลังจาก อัญประกาศเป็นเครื่องหมายอัศเจรีย์ (!) บ่งชี้ว่าส่วนที่เหลือของสตริง ควรถูกจัดการเป็นนิพจน์ปกติ ไคลเอ็นต์คอนเทนเนอร์อนุญาตการใช้ นิพจน์ปกติบนทั้งฟิลด์ hwtype และ hwaddr สตริงเดี่ยว ใช้เพื่อแสดงแทนทั้งสองฟิลด์ด้วยรูปแบบต่อไปนี้:

   ```
   decimal_number-data
   ```

   หาก decimal_number เป็นศูนย์ ข้อมูลจะเป็นสตริง ASCII หากเป็นตัวเลขอื่น ข้อมูลจะเป็นตัวเลขฐานหก

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
subnet	subnet ดีฟอลต์	ใช่	ไม่มี	ระบุ subnet ที่ไม่มีช่วงใดๆ เซิร์ฟเวอร์ใช้ subnet เฉพาะถ้าเซิร์ฟเวอร์ตอบกลับแพ็กเก็ต INFORM จากไคลเอ็นต์ และ แอดเดรสของไคลเอ็นต์ไม่มี subnet คอนเทนเนอร์ที่ตรงกันอื่น
subnet	subnet *subnet id netmask*	ใช่	ไม่มี	ระบุ subnet และพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วงหรือคำสั่ง exclude ช่วงทางเลือก เป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือน เพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
subnet	subnet *subnet id netmask ช่วง*	ใช่	ไม่มี	ระบุ subnet และพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วงหรือคำสั่ง exclude ช่วงทางเลือก เป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
subnet	subnet *subnet id netmask เลเบล:ระดับความสำคัญ*	ใช่	ไม่มี	ระบุ subnet และพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วงหรือคำสั่ง exclude ช่วงทางเลือก เป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
subnet	subnet *subnet id netmask ช่วง เลเบล:ระดับความสำคัญ*			ระบุ subnet และพูลของแอดเดรส มีการสมมติว่า แอดเดรสทั้งหมดอยู่ในพูล ยกเว้นว่ามีการระบุช่วงบนบรรทัด หรือแอดเดรสถูก แก้ไขในภายหลังในคอนเทนเนอร์โดยช่วงหรือคำสั่ง exclude ช่วงทางเลือก เป็นคู่ของ IP แอดเดรสในรูปแบบจุด quad ที่แบ่งโดยเครื่องหมายขีด สามารถระบุเลเบลทางเลือกและระดับความสำคัญได้ ซึ่งใช้โดย subnets เสมือนเพื่อระบุและจัดลำดับ subnets ใน subnet เสมือน เลเบล และระดับความสำคัญมีการแบ่งโดยเครื่องหมายจุดคู่ คอนเทนเนอร์เหล่านี้ใช้ได้ที่ระดับ คอนเทนเนอร์สากลหรือฐานข้อมูลเท่านั้น
subnet	subnet *subnet id ช่วง*	ใช่	ไม่มี	ระบุ subnet ที่อยู่ภายในคอนเทนเนอร์เครือข่าย กำหนด ช่วงของแอดเดรสที่เป็น subnet ทั้งหมดยกเว้นว่ามีการระบุส่วนช่วง ทางเลือก Netmask ที่เชื่อมโยงกับ subnet นำมาจากคอนเทนเนอร์เครือข่าย ล้อมรอบ **หมายเหตุ:** เมธอดนี้ใช้เพื่อสนับสนุนรูปแบบ subnet อื่น

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
อ็อพชัน	อ็อพชัน *ตัวเลข ข้อมูล* ...	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณี ของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ ส่วนคำสั่งอ็อพชัน * **deny** หมายความว่าอ็อพชันทั้งหมดที่ไม่ได้ระบุในคอนเทนเนอร์ปัจจุบัน จะไม่ถูกส่งคืนไปยังไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8-บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* หรือ hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วย อ็อพชันอื่นในอ็อพชัน 43
อ็อพชัน	อ็อพชัน *ตัวเลข*deny	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณี ของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ ส่วนคำสั่งอ็อพชัน * **deny** หมายความว่าอ็อพชันทั้งหมดที่ไม่ได้ระบุในคอนเทนเนอร์ปัจจุบัน จะไม่ถูกส่งคืนไปยังไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8-บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* หรือ hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วย อ็อพชันอื่นในอ็อพชัน 43
อ็อพชัน	อ็อพชัน * deny	ไม่	ไม่มี	ระบุอ็อพชันที่จะส่งไปยังไคลเอ็นต์ หรือในกรณี ของ deny ระบุอ็อพชันที่จะป้องกันไม่ให้ส่งไปยังไคลเอ็นต์ ส่วนคำสั่งอ็อพชัน * **deny** หมายความว่าอ็อพชันทั้งหมดที่ไม่ได้ระบุในคอนเทนเนอร์ปัจจุบัน จะไม่ถูกส่งคืนไปยังไคลเอ็นต์ อ็อพชัน *ตัวเลข* ปฏิเสธเฉพาะอ็อพชันที่ระบุเท่านั้น *ตัวเลข* เป็นเลขจำนวนเต็ม 8-บิตที่ไม่มีเครื่องหมาย *ข้อมูล* เป็นข้อมูลเฉพาะอ็อพชัน (โปรดดูที่ ข้างบน) หรือสามารถระบุเป็นสตริงที่อยู่ในอัญประกาศ (บ่งชี้ข้อความ ASCII) หรือ 0x*hexdigits* หรือ hex*"hexdigits"* หรือ hex *"hexdigits"* หากอ็อพชันอยู่ในคอนเทนเนอร์ผู้ขาย อ็อพชันจะถูกล้อมรอบด้วย อ็อพชันอื่นในอ็อพชัน 43

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
exclude	exclude *IP แอดเดรส*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง exclude คำสั่ง exclude ไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล คำสั่ง exclude ลบแอดเดรสหรือช่วงที่ระบุ ออกจากหน้าปัจจุบันบนคอนเทนเนอร์ คำสั่ง exclude ช่วยให้คุณสามารถสร้าง ช่วงที่ไม่ต่อเนื่องสำหรับ subnets หรือคอนเทนเนอร์อื่น
exclude	exclude *dotted_quad–dotted_quad*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง exclude คำสั่ง exclude ไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล คำสั่ง exclude ลบแอดเดรสหรือช่วงที่ระบุ ออกจากหน้าปัจจุบันบนคอนเทนเนอร์ คำสั่ง exclude ช่วยให้คุณสามารถสร้าง ช่วงที่ไม่ต่อเนื่องสำหรับ subnets หรือคอนเทนเนอร์อื่น
ช่วง	ช่วง *IP_address*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง ช่วง คำสั่งช่วงไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล ถ้าช่วงเป็นช่วงแรกในคอนเทนเนอร์ที่ไม่ได้ ระบุช่วงบนบรรทัดนิยามคอนเทนเนอร์ ช่วงสำหรับคอนเทนเนอร์ จะกลายเป็นช่วงที่ระบุโดยคำสั่งช่วง คำสั่งช่วงหลังจาก ช่วงแรกหรือคำสั่งช่วงทั้งหมดสำหรับคอนเทนเนอร์ที่ระบุช่วง ในนิยามจะถูกเพิ่มลงในช่วงปัจจุบัน ด้วยคำสั่งช่วง สามารถเพิ่มแอดเดรสหนึ่งหรือชุดของแอดเดรสลงในช่วงได้ ช่วงต้อง พอดีภายในนิยามคอนเทนเนอร์ subnet
ช่วง	ช่วง *dotted_quad–dotted_quad*	ไม่	ไม่มี	แก้ไขช่วงบนคอนเทนเนอร์ซึ่งมีคำสั่ง ช่วง คำสั่งช่วงไม่ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูล ถ้าช่วงเป็นช่วงแรกในคอนเทนเนอร์ที่ไม่ได้ ระบุช่วงบนบรรทัดนิยามคอนเทนเนอร์ ช่วงสำหรับคอนเทนเนอร์ จะกลายเป็นช่วงที่ระบุโดยคำสั่งช่วง คำสั่งช่วงหลังจาก ช่วงแรกหรือคำสั่งช่วงทั้งหมดสำหรับคอนเทนเนอร์ที่ระบุช่วง ในนิยามจะถูกเพิ่มลงในช่วงปัจจุบัน ด้วยคำสั่งช่วง สามารถเพิ่มแอดเดรสหนึ่งหรือชุดของแอดเดรสลงในช่วงได้ ช่วงต้อง พอดีภายในนิยามคอนเทนเนอร์ subnet

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ไคลเอ็นต์	ไคลเอ็นต์ *hwtype hwaddr* NONE			ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุ โดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรส ของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้ รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์
ไคลเอ็นต์	ไคลเอ็นต์ *hwtype hwaddr* ANY			ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุ โดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรส ของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้ รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์
ไคลเอ็นต์	ไคลเอ็นต์ *hwtype hwaddr dotted_quad*			ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุ โดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรส ของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้ รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ไคลเอ็นต์	ไคลเอ็นต์ *hwtype hwaddr ช่วง*			ระบุไคลเอ็นต์คอนเทนเนอร์ที่ปฏิเสธไคลเอ็นต์ซึ่งระบุ โดย *hwaddr* และ *hwtype* จาก การเรียกใช้แอดเดรส หาก *hwtype* เป็น 0 *hwaddr* จะเป็น สตริง ASCII มิฉะนั้น *hwtype* เป็นชนิดฮาร์ดแวร์ สำหรับไคลเอ็นต์และ *hwaddr* เป็นฮาร์ดแวร์แอดเดรส ของไคลเอ็นต์ ถ้า *hwaddr* เป็นสตริง สามารถมีอัญประกาศล้อมรอบสตริงได้ ถ้า *hwaddr* เป็น hexstring อาจระบุแอดเดรสเป็น 0x*hexdigits* หรือ hex *digits ช่วง* ช่วยให้ไคลเอ็นต์ที่ระบุโดย *hwaddr* และ *hwtype* ได้ รับแอดเดรสใน *ช่วง* ต้องเป็นนิพจน์ปกติ เพื่อจับคู่หลายไคลเอ็นต์
คลาส	คลาส *สตริง*	ใช่	ไม่มี	ระบุคลาสคอนเทนเนอร์ที่มีชื่อเป็น *สตริง* สตริงอาจอยู่ในอัญประกาศหรือไม่ก็ได้ หากอยู่ในอัญประกาศ อัญประกาศจะถูกลบออกก่อนการเปรียบเทียบ อัญประกาศเป็นสิ่งจำเป็นสำหรับสตริงที่มีช่องว่างหรือแท็บ คอนเทนเนอร์นี้ถูกต้อง ที่ทุกระดับ สามารถระบุช่วงเพื่อบ่งชี้ชุดของแอดเดรสที่จะส่งไปยัง ไคลเอ็นต์ที่มีคลาสนี้ ช่วงเป็น IP แอดเดรส quad จุดเดียวหรือ IP แอดเดรส quad สองจุดที่แบ่งด้วยเครื่องหมายขีด
คลาส	คลาส *สตริง ช่วง*	ใช่	ไม่มี	ระบุคลาสคอนเทนเนอร์ที่มีชื่อเป็น *สตริง* สตริงอาจอยู่ในอัญประกาศหรือไม่ก็ได้ หากอยู่ในอัญประกาศ อัญประกาศจะถูกลบออกก่อนการเปรียบเทียบ อัญประกาศเป็นสิ่งจำเป็นสำหรับสตริงที่มีช่องว่างหรือแท็บ คอนเทนเนอร์นี้ถูกต้อง ที่ทุกระดับ สามารถระบุช่วงเพื่อบ่งชี้ชุดของแอดเดรสที่จะส่งไปยัง ไคลเอ็นต์ที่มีคลาสนี้ ช่วงเป็น IP แอดเดรส quad จุดเดียวหรือ IP แอดเดรส quad สองจุดที่แบ่งด้วยเครื่องหมายขีด
เครือข่าย	เครือข่าย *network id netmask*	ใช่	ไม่มี	ระบุ ID เครือข่ายที่ใช้ข้อมูลคลาส (ตัวอย่างเช่น 9.3.149.0 ที่มี netmask 255.255.255.0 จะเป็นเครือข่าย 9.0.0.0 255.255.255.0) เวอร์ชันนี้ของคอนเทนเนอร์เครือข่ายใช้เพื่อจัดเก็บ subnets ที่มี ID เครือข่าย และ netmask เหมือนกัน เมื่อระบุช่วง แอดเดรสทั้งหมดใน ช่วงอยู่ในพูล ช่วงต้องอยู่ในเครือข่ายของ ID เครือข่าย รูปแบบนี้ใช้การกำหนดแอดเดรสแบบเต็มของคลาส รูปแบบนี้ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูลเท่านั้น **หมายเหตุ:** คีย์เวิร์ดเครือข่ายใช้เพื่อสนับสนุน subnet คอนเทนเนอร์

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
เครือข่าย	เครือข่าย *id เครือข่าย*	ใช่	ไม่มี	ระบุ ID เครือข่ายที่ใช้ข้อมูลคลาส (ตัวอย่างเช่น 9.3.149.0 ที่มี netmask 255.255.255.0 จะเป็นเครือข่าย 9.0.0.0 255.255.255.0) เวอร์ชันนี้ของคอนเทนเนอร์เครือข่ายใช้เพื่อจัดเก็บ subnets ที่มี ID เครือข่าย และ netmask เหมือนกัน เมื่อระบุช่วง แอดเดรสทั้งหมดใน ช่วงอยู่ในพูล ช่วงต้องอยู่ในเครือข่ายของ ID เครือข่าย รูปแบบนี้ ใช้การกำหนดแอดเดรสแบบเต็มของคลาส รูปแบบนี้ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูลเท่านั้น **หมายเหตุ:** คีย์เวิร์ดเครือข่ายใช้เพื่อสนับสนุน subnet คอนเทนเนอร์
เครือข่าย	เครือข่าย *id เครือข่าย ช่วง*			ระบุ ID เครือข่ายที่ใช้ข้อมูลคลาส (ตัวอย่างเช่น 9.3.149.0 ที่มี netmask 255.255.255.0 จะเป็นเครือข่าย 9.0.0.0 255.255.255.0) เวอร์ชันนี้ของคอนเทนเนอร์เครือข่ายใช้เพื่อจัดเก็บ subnets ที่มี ID เครือข่าย และ netmask เหมือนกัน เมื่อระบุช่วง แอดเดรสทั้งหมดใน ช่วงอยู่ในพูล ช่วงต้องอยู่ในเครือข่ายของ ID เครือข่าย รูปแบบนี้ ใช้การกำหนดแอดเดรสแบบเต็มของคลาส รูปแบบนี้ถูกต้องในระดับคอนเทนเนอร์สากลหรือ ฐานข้อมูลเท่านั้น **หมายเหตุ:** คีย์เวิร์ดเครือข่ายใช้เพื่อสนับสนุน subnet คอนเทนเนอร์
ผู้ขาย	ผู้ขาย *vendor_id*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id* hex""	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex"*digits*" อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer
ผู้ขาย	ผู้ขาย *vendor_id* hex""	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex"*digits*" อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id* 0xdata	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer
ผู้ขาย	ผู้ขาย *vendor_id""*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id ช่วง*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer
ผู้ขาย	ผู้ขาย *vendor_id ช่วง* hex""	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id ช่วง* hex ""	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex"*digits*" อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer
ผู้ขาย	ผู้ขาย *vendor_id ช่วง* 0xdata	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex"*digits*" อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	ผู้ขาย *vendor_id ช่วง ""*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer
ผู้ขาย	vendor pxe	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
ผู้ขาย	vendor pxeserver	ใช่	ไม่มี	ระบุคอนเทนเนอร์ผู้ขาย คอนเทนเนอร์ผู้ขายใช้เพื่อ ส่งคืนอ็อพชัน 43 ไปยังไคลเอ็นต์ Id ผู้ขายอาจระบุเป็นสตริงในอัญประกาศ หรือไบนารีสตริงในรูปแบบ 0x*hexdigits* หรือ hex*"digits"* อาจใส่ช่วงทางเลือกหลังจาก id ผู้ขาย ช่วงมีการระบุเป็น quads สองจุดที่แบ่งด้วยเครื่องหมายขีด หลังจากช่วงทางเลือก สามารถระบุ hexstring หรือสตริง ASCII ทางเลือกเป็นส่วนแรกของอ็อพชัน 43 หากอ็อพชันอยู่ในคอนเทนเนอร์ อ็อพชัน จะถูกแนบเข้ากับข้อมูลอ็อพชัน 43 หลังจากประมวลผลอ็อพชันทั้งหมดแล้ว End Of Option List Option จะถูกแนบเข้ากับข้อมูล เมื่อต้องการส่งคืนอ็อพชันภายนอกอ็อพชัน 43 ให้ใช้ไคลเอ็นต์นิพจน์ปกติที่ตรงกับไคลเอ็นต์ทั้งหมดเพื่อ ระบุอ็อพชันปกติที่จะส่งคืนตามข้อมูล ID ผู้ขาย **pxe** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEClient **pxeserver** หลังจาก คีย์เวิร์ด **ผู้ขาย** จะสร้างคอนเทนเนอร์ผู้ขาย สำหรับ PXEServer
inoption	inoption *ตัวเลข option_data*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเอง ที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุ หมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชัน สำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ – สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม – สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ 0x นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะ เดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย "! (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดีจะเป็นสตริงฐานสิบหก ของไบต์ที่ไม่นำหน้าด้วยอักขระ 0x

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
inoption	inoption *ตัวเลข option_data ช่วง*	ใช่	ไม่มี	ระบุคอนเทนเนอร์ที่จะจับคู่กับอ็อพชันขาเข้าแบบกำหนดเอง ที่ระบุโดยไคลเอ็นต์ *ตัวเลข* ระบุ หมายเลขอ็อพชัน *option_data* ระบุคีย์ ที่จะจับคู่กับคอนเทนเนอร์นี้ซึ่งจะเลือกในระหว่างการเลือกแอดเดรสและอ็อพชัน สำหรับไคลเอ็นต์ *option_data* มีการระบุในรูปแบบที่คาดไว้ – สตริงในอัญประกาศ, IP แอดเดรส, ค่าเลขจำนวนเต็ม – สำหรับอ็อพชันที่รู้จักดี หรือสามารถเลือกระบุเป็นสตริงฐานสิบหกของไบต์ถ้ามีอักขระ `0x` นำหน้า สำหรับอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดี สามารถระบุสตริงฐานสิบหกของไบต์ในลักษณะ เดียวกัน นอกจากนี้ *option_data* สามารถบ่งชี้ นิพจน์ปกติที่จะเปรียบเทียบกับสตริงแสดงแทนข้อมูล อ็อพชันของไคลเอ็นต์ นิพจน์ปกติมีการระบุในสตริงในเครื่องหมายอัญประกาศ ที่ขึ้นต้นด้วย `"!(` (อัญประกาศคู่ตามด้วยเครื่องหมายอัศเจรีย์) รูปแบบสตริงของอ็อพชันที่เซิร์ฟเวอร์ไม่รู้จักดีจะเป็นสตริงฐานสิบหก ของไบต์ที่ไม่นำหน้าด้วยอักขระ `0x`
เสมือน	virtual fill *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขายหรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรส จะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
เสมือน	virtual sfill *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรสจะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน
เสมือน	virtual rotate *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรสจะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน
เสมือน	virtual srotate *id id* ...	ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรสจะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
เสมือน		ไม่	ไม่มี	ระบ subnet เสมือนที่มีนโยบาย `fill` หมายความว่า ใช้แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป `rotate` หมายถึง เลือกแอดเดรสจากพูลถัดไปในรายการบนแต่ละคำร้องขอ `sfill` และ `srotate` เหมือนกับ `fill` และ `rotate` แต่จะทำการค้นหา เพื่อดูว่าไคลเอ็นต์ตรงกับคอนเทนเนอร์ ผู้ขาย หรือคลาสใน subnet หรือไม่ หากพบรายการที่ตรงกันซึ่งสามารถระบุแอดเดรสจะใช้แอดเดรส จากคอนเทนเนอร์นั้นแทนการปฏิบัติตามนโยบาย สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet id เหมือนกัน
inorder:	inorder: *id id ...*	ไม่	ไม่มี	ระบุ subnet เสมือนที่มีนโยบาย fill ซึ่งหมายความว่าใช้ แอดเดรสทั้งหมดในคอนเทนเนอร์ก่อนไปยังคอนเทนเนอร์ถัดไป สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จาก นิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet ID เหมือนกัน
balance:	balance: *id id ...*	ไม่	ไม่มี	ระบุ subnet เสมือนที่มีนโยบาย rotate ซึ่งหมายความว่าใช้ แอดเดรสถัดไปในคอนเทนเนอร์ถัดไป สามารถมี IDs ได้มากตามต้องการ *id* เป็น subnet ID จากนิยาม subnet หรือเลเบลจากนิยาม subnet เลเบลเป็นสิ่งจำเป็น ถ้ามีหลาย subnets ที่มี subnet ID เหมือนกัน
bootstrapserver	bootstrapserver *IP แอดเดรส*	ไม่	ไม่มี	ระบุเซิร์ฟเวอร์ที่ไคลเอ็นต์ควรจะใช้ไฟล์ TFTP หลังจาก ได้รับแพ็กเก็ต BOOTP หรือ DHCP ค่านี้กรอกข้อมูลในฟิลด์ **siaddr** ในแพ็กเก็ต ค่านี้ถูกต้องที่คอนเทนเนอร์ทุกระดับ
giaddrfield	giaddrfield *IP แอดเดรส*	ไม่	ไม่มี	ระบุ giaddrfield สำหรับแพ็กเก็ตการตอบกลับ **หมายเหตุ:** การระบุนี้ไม่ถูกต้องในโปรโตคอล BOOTP และ DHCP แต่บางไคลเอ็นต์ต้องการฟิลด์ **giaddr** เป็นดีฟอลต์เกตเวย์สำหรับเครือข่าย เนื่องจากความขัดแย้งที่อาจเกิดขึ้นได้นี้ จึงควรใช้ giaddrfield ภายในไคลเอ็นต์คอนเทนเนอร์เท่านั้น แม้ว่าสามารถทำงาน ได้ในทุกระดับ

คีย์เวิร์ด	รูปแบบ	คอนเทนเนอร์ย่อย?	ค่าดีฟอลต์	ความหมาย
bootfile	bootfile *พาธ*	ไม่	ไม่มี	ระบุ bootfile ที่จะใช้ในส่วนไฟล์ของแพ็กเก็ตการตอบกลับ ค่านี้สามารถระบุที่คอนเทนเนอร์ทุกระดับ นโยบาย bootfile กำหนด จำนวนไอเท็มที่ระบุในส่วนไฟล์ของแพ็กเก็ตขาเข้าที่โต้ตอบกับ bootfile และคำสั่งไดเร็กทอรีโฮม
pxebootfile	pxebootfile *SystemArch MajorVer MinorVer Bootfilename Type Layer*	ไม่	ไม่มี	ระบุ bootfile ที่จะกำหนดให้กับ PXEClient ตัวแจงส่วนไฟล์ คอนฟิกสร้างข้อผิดพลาดถ้าจำนวนของพารามิเตอร์หลังจากคีย์เวิร์ด น้อยกว่า 4, ละเว้นถ้ามากกว่า 7 และถ้าเป็น 4 จะสมมติ ค่าชนิด = 0 และชั้น = 0 คีย์เวิร์ดนี้สามารถใช้ได้ในคอนเทนเนอร์เท่านั้น

สำหรับรายละเอียดเกี่ยวกับอ็อพชันอื่น ให้ดูที่ “อ็อพชันที่รู้จักของไฟล์เซิร์ฟเวอร์ DHCP” ในหน้า 249 และ “อ็อพชันย่อยของคอนเทนเนอร์ผู้ผลิต PXE” ในหน้า 332

TCP/IP daemons

Daemons (เรียกอีกอย่างว่า *เซิร์ฟเวอร์*) คือโปรเซสที่รันอย่างต่อเนื่อง ในพื้นหลังและทำฟังก์ชันที่ต้องการโดยโปรเซสอื่น **Transmission Control Protocol/Internet Protocol (TCP/IP)** จัดเตรียม daemons สำหรับการนำฟังก์ชันบางอย่างไปใช้ในระบบปฏิบัติการ

Daemons เหล่านี้เป็นโปรเซสพื้นหลังที่รันโดยไม่ขัดจังหวะโปรเซส อื่น (ยกเว้นว่าเป็นส่วนหนึ่งของฟังก์ชัน daemon)

Daemons ถูกเรียกใช้โดยคำสั่งที่ระดับการจัดการระบบ หรือโดย daemons อื่น หรือโดยเชลล์สคริปต์ คุณยังสามารถควบคุม daemons โดยใช้ **inetd** daemon, เชลล์สคริปต์ **rc.tcpip**, และ System Resource Controller (SRC)

ระบบย่อยและเซิร์ฟเวอร์ย่อย

ระบบย่อย คือ daemon หรือเซิร์ฟเวอร์ที่ควบคุมโดย SRC *เซิร์ฟเวอร์ย่อย* คือ daemon ที่ควบคุมโดยระบบย่อย (โดยปกติ คำสั่ง daemon และชื่อ daemon มีการบ่งชี้โดย **d** ที่ ตอนท้ายของชื่อ)

หมวดหมู่ของระบบย่อยและเซิร์ฟเวอร์ย่อยไม่มีความเกี่ยวข้องซึ่งกันและกัน นั่นคือ daemons ไม่มีการแสดงรายการเป็นทั้งระบบย่อยและเป็นเซิร์ฟเวอร์ย่อย ระบบย่อย **TCP/IP** เดียวที่ควบคุม daemons อื่นคือ **inetd** daemon เซิร์ฟเวอร์ย่อย **TCP/IP** ทั้งหมด ยังเป็นเซิร์ฟเวอร์ย่อย **inetd** ด้วย

หากต้องการรายการของ **TCP/IP** daemons ให้ดูที่ “TCP/IP daemons” ในหน้า 460

System Resource Control

เทียบกับฟังก์ชันอื่น, SRC อนุญาตให้คุณสตาร์ท daemons, หยุดการทำงาน และติดตามกิจกรรม นอกจากนี้ SRC จัดเตรียมความสามารถในการจัดกลุ่ม daemons ลงในระบบย่อยและเซิร์ฟเวอร์ย่อย

System Resource Control เป็นเครื่องมือที่ถูกออกแบบให้ช่วยเหลือคุณในการควบคุม daemons SRC อนุญาตให้ควบคุมนอกเหนือจากแฟล็กและพารามิเตอร์ที่ใช้ได้กับแต่ละคำสั่ง daemon

ดูที่ System Resource Controller ใน *การจัดการระบบปฏิบัติการและอุปกรณ์* สำหรับข้อมูลเพิ่มเติม เกี่ยวกับ System Resource Controller

สำหรับรายการของคำสั่ง SRC ดูที่ "คำสั่ง SRC" ในหน้า 458

การกำหนดคอนฟิก inetd daemon

ใช้ขั้นตอนเหล่านี้เพื่อกำหนดคอนฟิก **TCP/IP inetd** daemon

เมื่อต้องการกำหนดคอนฟิก **inetd** daemon:

1. ระบุเซิร์ฟเวอร์ย่อยที่จะเรียกใช้โดยการเพิ่ม **inetd** daemon
2. ระบุลักษณะรีสตาร์ทโดยการเปลี่ยนลักษณะรีสตาร์ท ของ **inetd** daemon

ตารางที่ 77. ภารกิจการกำหนดคอนฟิก inetd daemon

ภารกิจ	พาธด่วน SMIT	คำสั่งหรือไฟล์
การเริ่มต้น **inetd** Daemon	`smit mkinetd`	**startsrc -s inetd**
การเปลี่ยนลักษณะรีสตาร์ทของ **inetd** Daemon	`smit chinetd` หรือ `smit lsinetd`	
การหยุด **inetd** Daemon	`smit rminetd`	**stopsrc -s inetd**
การแสดงรายการเซิร์ฟเวอร์ย่อย **inetd** ทั้งหมด	`smit inetdconf`	
การเพิ่มเซิร์ฟเวอร์ย่อย **inetd** [1]	`smit mkinetdconf`	แก้ไข `/etc/inetd.conf` แล้วรัน **refresh -s inetd** หรือ **kill -1** *inetdPID*[2]
เปลี่ยน/แสดงลักษณะของเซิร์ฟเวอร์ย่อย **inetd**	`smit inetdconf`	แก้ไข `/etc/inetd.conf` แล้วรัน **refresh -s inetd** หรือ **kill -1** *inetdPID*[2]
การลบเซิร์ฟเวอร์ย่อย **inetd**	`smit rminetd`	แก้ไข `/etc/inetd.conf` แล้วรัน **refresh -s inetd** หรือ **kill -1** *inetdPID*[2]

หมายเหตุ:

1. การเพิ่มเซิร์ฟเวอร์ย่อย **inetd** จะกำหนดคอนฟิก **inetd** daemon เพื่อให้เรียกใช้เซิร์ฟเวอร์ย่อยเมื่อต้องการ
2. ทั้งคำสั่ง **refresh** และ **kill** แจ้ง **inetd** daemon เกี่ยวกับการเปลี่ยนแปลงในไฟล์ คอนฟิกูเรชัน

เซอร์วิสเครือข่ายไคลเอ็นต์

Client Network Services (เข้าถึงได้โดยใช้ SMIT fast path, `smit clientnet`) อ้างอิงโปรโตคอล **TCP/IP** ที่มีสำหรับใช้โดยระบบปฏิบัติการนี้

แต่ละโปรโตคอล (หรือเซอร์วิส) เป็นที่รู้จักโดยหมายเลขพอร์ตที่โปรโตคอลใช้บน เครือข่าย ดังนั้นจึงเรียกว่า *พอร์ตที่รู้จักกันดี* เพื่อความสะดวกสำหรับโปรแกรมเมอร์ หมายเลขพอร์ตสามารถอ้างอิงได้โดยใช้ชื่อและหมายเลข ตัวอย่างเช่น **TCP/IP** เมล โปรโตคอลใช้พอร์ต 25 และเป็นที่รู้จักในชื่อ **smtp** หากโปรโตคอลแสดงรายการอยู่ (ไม่มีข้อคิดเห็น) ในไฟล์ `/etc/services` แสดงว่าโฮสต์สามารถใช้โปรโตคอลนั้นได้

โดยค่าดีฟอลต์โปรโตคอล **TCP/IP** ทั้งหมดมีการกำหนดไว้ในไฟล์ `/etc/services` คุณไม่ต้องกำหนดคอนฟิกไฟล์นี้ หากคุณเขียนโปรแกรมไคลเอ็นต์/เซิร์ฟเวอร์ ของคุณเอง คุณอาจต้องการเพิ่มเซอร์วิสลงในไฟล์ `/etc/services` และสงวนหมายเลขพอร์ตและชื่อเฉพาะสำหรับเซอร์วิสของคุณ หากคุณตัดสินใจ เพิ่มเซอร์วิสลงใน `/etc/services` โปรดทราบว่าหมายเลข

พอร์ต 0 ถึง 1024 ถูกสงวนไว้สำหรับการใช้ของระบบ

ตารางที่ 78. ภารกิจเซอร์วิสเครือข่ายไคลเอ็นต์

ภารกิจ	พาธด่วน SMIT	คำสั่งหรือไฟล์
การแสดงรายการเซอร์วิสทั้งหมด	`smit lsservices`	ดู `/etc/services`
การเพิ่มเซอร์วิส	`smit mkservices`	แก้ไข `/etc/services`
เปลี่ยน/แสดงลักษณะของเซอร์วิส	`smit chservices`	แก้ไข `/etc/services`
การลบเซอร์วิส	`smit rmservices`	แก้ไข `/etc/services`

เซอร์วิสเครือข่ายเซิร์ฟเวอร์

เซอร์วิสเครือข่ายเซิร์ฟเวอร์รวมถึงการควบคุมการเข้าถึงแบบรีโมต การ เริ่มต้นหรือการหยุด **TCP/IP** และการจัดการกับไดรเวอร์อุปกรณ์ **pty** ดังแสดงในตารางนี้

ไดรเวอร์อุปกรณ์ **pty** มีการติดตั้งโดยอัตโนมัติ พร้อมกับระบบ โดยค่าดีฟอลต์ มีการกำหนดคอนฟิกเพื่อสนับสนุนลิงก์สัญลักษณ์ลักษณะ 16 BSD และพร้อมให้ระบบใช้งานในเวลาบูต

ตารางที่ 79. ภารกิจเซอร์วิสเครือข่ายเซิร์ฟเวอร์

ภารกิจ	พาธด่วน SMIT	คำสั่งหรือไฟล์
การควบคุมการเข้าถึงแบบรีโมต		โปรดดูที่ ″Remote Command Execution Access″ และ ″Restricted File Transfer Program Users″ ใน *การรักษาความปลอดภัย*
เริ่มต้น รีสตาร์ท หรือหยุดระบบย่อย TCP/IP	`smit otherserv`	โปรดดู “System Resource Control” ในหน้า 376
แสดง/เปลี่ยนลักษณะของไดรเวอร์อุปกรณ์ pty	`smit chgpty`	**chdev -l pty0 -P -a num=***X* โดยที่ *X* มีช่วง ตั้งแต่ 0 ถึง 64
ทำให้ไดรเวอร์อุปกรณ์ pty ไม่พร้อมใช้งาน	`smit pty` จากนั้นเลือก **ลบ PTY; เก็บนิยาม**	ไม่มีคำสั่งหรือไฟล์ที่เกี่ยวข้อง
ทำให้ไดรเวอร์อุปกรณ์ pty พร้อมใช้งาน	`smit pty` จากนั้นเลือก **กำหนดคอนฟิก PTY ที่กำหนดไว้**	ไม่มีคำสั่งหรือไฟล์ที่เกี่ยวข้อง
สร้างรายงานข้อผิดพลาด	`smit errpt`	ไม่มีคำสั่งหรือไฟล์ที่เกี่ยวข้อง
ติดตาม pty	`smit trace`	ไม่มีคำสั่งหรือไฟล์ที่เกี่ยวข้อง

การจัดเส้นทาง TCP/IP

route กำหนดพาธสำหรับการส่งแพ็กเก็ตผ่านอินเตอร์เน็ตเน็ตเวิร์ก ไปที่แอดเดรสบนเน็ตเวิร์กอื่น

เส้นทางไม่กำหนดพาธสมบูรณ์ เฉพาะพาธเซกเมนต์จากหนึ่งโฮสต์ ไปที่เกตเวย์ที่สามารถส่งต่อแพ็กเก็ตไปที่ปลายทาง (หรือจากหนึ่งเกตเวย์ ไปที่เกตเวย์อื่น) มีเส้นทางห้าชนิด:

ไอเท็ม	คำอธิบาย
เส้นทางโฮสต์	กำหนดเกตเวย์ที่สามารถส่งต่อแพ็กเก็ตไปที่โฮสต์เจาะจงบน เน็ตเวิร์กอื่น
เส้นทางเน็ตเวิร์ก	กำหนดเกตเวย์ที่สามารถส่งต่อแพ็กเก็ตไปที่โฮสต์บน เน็ตเวิร์กที่เจาะจง
เส้นทางดีฟอลต์	กำหนดเกตเวย์เพื่อใช้เมื่อเส้นทางโฮสต์หรือเน็ตเวิร์กไปที่ปลายทาง ไม่ได้กำหนดไว้
เส้นทาง loopback	เส้นทางดีฟอลต์สำหรับแพ็กเก็ตทั้งหมดที่ส่งไปที่โลคัล เน็ตเวิร์กแอดเดรส loopback เส้นทาง IP เป็น 127.0.0.1 เสมอ
เส้นทางการกระจายข้อมูล	เส้นทางดีฟอลต์สำหรับแพ็กเก็ตการกระจายข้อมูลทั้งหมด สองเส้นทางกระจาย ข้อมูลจะถูกกำหนดให้กับแต่ละ subnet โดยอัตโนมัติบนเน็ตเวิร์ก ที่มี IP (เส้นทางหนึ่งให้กับแอดเดรส subnet และอีกเส้นทางให้กับแอดเดรสการกระจายข้อมูลของ subnet)

เส้นทางถูกกำหนดใน *ตารางการจัดเส้นทาง* เคอร์เนล นิยามเส้นทาง มีข้อมูลบนเน็ตเวิร์กที่เชื่อมต่อได้จากโลคัลโฮสต์และบนเกตเวย์ ที่สามารถถูกใช้เพื่อเข้าถึงรีโมตเน็ตเวิร์ก เมื่อเกตเวย์ได้รับดาตาแกรม จะตรวจสอบตารางการจัดเส้นทางเพื่อค้นหาจุดต่อไปที่จะส่งดาตาแกรมตาม พาธไปที่ปลายทาง

คุณสามารถเพิ่มหลายเราต์สำหรับปลายทางเดียวกันในตาราง การเราต์เคอร์เนล การค้นหาการจัดเส้นทางหาค่าเส้นทางทั้งหมดที่ตรงกับการร้องขอ จากนั้น เลือกเส้นทางที่มีเมทริกระยะทางสั้นที่สุด ถ้ามีเส้นทางที่ตรงกันหลายเส้นทาง ซึ่งระยะเท่ากัน การค้นหาจะเลือกเส้นทางที่เจาะจงที่สุด ถ้าทั้งสองเงื่อนไข เท่ากันสำหรับหลายเส้นทาง การจัดเส้นทางค้นหาตัวเลือกรองของเส้นทาง ที่พบ

การจัดเส้นทางสแตติกและไดนามิก

ใน **TCP/IP** การจัดเส้นทางเป็นหนึ่งในสองชนิด: *สแตติก* หรือ *ไดนามิก*

ด้วยการจัดเส้นทางสแตติก คุณดูแลตารางการจัดเส้นทางด้วยตัวเองโดยใช้คำสั่ง **route** การจัดเส้นทางสแตติกในทางปฏิบัติสำหรับ การสื่อสารเน็ตเวิร์กเดี่ยวโดยมีหนึ่งหรือสอง เน็ตเวิร์กอื่น อย่างไรก็ตาม ขณะที่เน็ตเวิร์กของคุณเริ่มการสื่อสารกับเน็ตเวิร์กเพิ่มเติม จำนวนของเกตเวย์เพิ่มขึ้น และเช่นเดียวกับเวลาและ การทำงานที่จำเป็นในการดูแลตารางการจัดเส้นทางด้วยตัวเอง

ด้วยการจัดเส้นทางไดนามิก, daemons อัพเดตตารางการจัดเส้นทางโดยอัตโนมัติ daemons การจัดเส้นทางได้รับข้อมูลที่ส่งโดย daemons การจัดเส้นทางอื่นอย่างต่อเนื่อง และดังนั้นอัพเดตตารางการจัดเส้นทางอย่างต่อเนื่อง

TCP/IP จัดเตรียมสอง daemons สำหรับใช้ในการจัดเส้นทางไดนามิก **routed** และ **gated** daemons **gated** daemon สนับสนุนโปรโตคอลการจัดเส้นทาง **Routing Information Protocol (RIP), Routing Information Protocol Next Generation (RIPng), Exterior Gateway Protocol (EGP), Border Gateway Protocol (BGP)** และ **BGP4+, Defense Communications Network Local-Network Protocol (HELLO), Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS)** และ **Internet Control Message Protocol (ICMP and ICMPv6)/Router Discovery** พร้อมกัน นอกจากนี้ **gated** daemon สนับสนุน **Simple Network Management Protocol (SNMP) routed** daemon สนับสนุนเพียง **Routing Information Protocol**

daemons การจัดเส้นทางสามารถทำงานในหนึ่งในสองโหมด *passive* หรือ *active*, ขึ้นกับอ็อพชันที่คุณใช้เมื่อเริ่ม daemons ในโหมด active, daemons การจัดเส้นทางทั้งส่งข้อมูลการจัดเส้นทาง เป็นระยะๆ เกี่ยวกับโลคัลเน็ตเวิร์ก ไปที่เกตเวย์และโฮสต์และรีบข้อมูลการจัดเส้นทางจาก โฮสต์และเกตเวย์ ในโหมด passive, daemons การจัดเส้นทางรับข้อมูลการจัดเส้นทาง จากโฮสต์และเกตเวย์ แต่ไม่พยายามเก็บรีโมตเกตเวย์ที่อัพเดต (ไม่มีการประกาศข้อมูลการจัดเส้นทางของตัวเอง)

การจัดเส้นทางสองชนิดนี้ไม่เพียงสามารถใช้ได้สำหรับเกตเวย์ แต่สำหรับโฮสต์อื่น ก็ใช้ได้เช่นกัน การจัดเส้นทางสแตติกทำงานเหมือนกันสำหรับเกตเวย์และ โฮสต์อื่น daemons การจัดเส้นทางไดนามิก, อย่างไรก็ตาม, ต้องถูกรันในโหมด passive (quiet) mode เมื่อรันบนโฮสต์ที่ไม่ใช่เกตเวย์

เกตเวย์การจัดเส้นทาง TCP/IP

เกตเวย์เป็นเราเตอร์ชนิดหนึ่ง *เราเตอร์* เชื่อมโยงเน็ตเวิร์กอย่างน้อย สองเน็ตเวิร์กเข้าด้วยกัน และจัดให้มีฟังก์ชันการจัดเส้นทาง ตัวอย่างเช่น เราเตอร์บางตัวจัดเส้นทาง ที่ระดับเน็ตเวิร์กอินเตอร์เฟส หรือที่ระดับฟิสิคัล อย่างไรก็ตาม *เกตเวย์* จะจัดเส้นทางที่ระดับเน็ตเวิร์ก

เกตเวย์ได้รับดาตาแกรม IP จากเกตเวย์ หรือโฮสต์อื่น สำหรับการนำส่งไปยังโฮสต์บนเน็ตเวิร์กโลคัล และจัดเส้นทางดาตาแกรม IP จากเน็ตเวิร์กหนึ่ง ไปอีกเน็ตเวิร์กหนึ่ง ตัวอย่างเช่น เกตเวย์ที่เชื่อมระหว่างเน็ตเวิร์ก Token-Ring สองเน็ตเวิร์ก มีการ์ดอะแด็ปเตอร์ Token-Ring สองการ์ด แต่ละการ์ดมีเน็ตเวิร์กอินเตอร์เฟส Token-Ring ของตนเอง เมื่อต้องการส่งข้อมูล เกตเวย์จะได้รับดาตาแกรมผ่านเน็ตเวิร์กอินเตอร์เฟส หนึ่ง และส่งออกไปผ่านทางอีกเน็ตเวิร์กอินเตอร์เฟสหนึ่ง เกตเวย์ จะทำการตรวจสอบการเชื่อมต่อเน็ตเวิร์กของตนเป็นระยะผ่านทางข้อความสถานะอินเตอร์เฟส

เกตเวย์จัดเส้นทางแพ็กเก็ตตามเน็ตเวิร์กปลายทางไม่ใช่ตามโฮสต์ ปลายทาง นั่นคือ เครื่องเกตเวย์ไม่จำเป็นต้องเก็บค่า การติดตามปลายทางโฮสต์ที่เป็นไปได้ทั้งหมดสำหรับแพ็กเก็ต เกตเวย์จะจัดเส้นทาง แพ็กเก็ตตามค่าเน็ตเวิร์กของโฮสต์ปลายทางแทน จากนั้นเน็ตเวิร์ก ปลายทางจะดูแลเกี่ยวกับการส่งแพ็กเก็ตไปยังโฮสต์ปลายทาง ดังนั้น เครื่องเกตเวย์โดยทั่วไปต้องการใช้ความจุของหน่วยเก็บดิสก์ที่จำกัดเท่านั้น (ถ้า มี) และความจุหน่วยความจำหลักที่จำกัด

ระยะห่างที่ข้อความต้องเดินทางจากโฮสต์เริ่มต้นไปยังโฮสต์ ปลายทางจะขึ้นอยู่กับจำนวน *hop เกตเวย์* ที่ต้องใช้ เกตเวย์ จะมีค่าศูนย์ hops จากเน็ตเวิร์กไปยังเครื่องที่เชื่อมต่อโดยตรง หนึ่ง hop จาก เน็ตเวิร์กที่สามารถเข้าถึงได้ผ่านทางเกตเวย์ เป็นต้น ระยะห่างข้อความ ถูกปกติถูกแสดงเป็นจำนวน hop เกตเวย์ที่จำเป็น หรือ *จำนวน hop* (รวมทั้ง เรียกว่า *เมทริก*)

การจัดเส้นทางเกตเวย์ภายในและภายนอก:

เกตเวย์ภายในคือเกตเวย์ที่เป็นของระบบ autonomous เดียวกัน ซึ่งสื่อสารระหว่างกันโดยใช้โปรโตคอล **Routing Information Protocol (RIP), Routing Information Protocol Next Generation (RIPng), Intermediate System to Intermediate System,** โปรโตคอล **Open Shortest Path First (OSPF)** หรือ **HELLO Protocol (HELLO)** เกตเวย์ภายนอกเป็นของระบบ autonomous อื่น ซึ่งใช้ **Exterior Gateway Protocol (EGP), Border Gateway Protocol (BGP)** หรือ **BGP4+**

ตัวอย่าง พิจารณาสองระบบ autonomous ระบบแรกคือ เน็ตเวิร์กทั้งหมดถูกดูแลโดย Widget Company ระบบที่สองคือ เน็ตเวิร์กทั้งหมด ถูกดูแลโดย Gadget Company Widget Company มีเครื่องหนึ่ง ชื่อ apple ซึ่งเป็นเกตเวย์ของ Widget ไปยังอินเตอร์เน็ต Gadget Company มีเครื่องหนึ่ง ชื่อ orange ซึ่งเป็นเกตเวย์ของ Gadget ไปยังอินเตอร์เน็ต ทั้งสองบริษัทมีระบบภายในเน็ตเวิร์กต่างกันระหว่างบริษัท เกตเวย์เชื่อมต่อเน็ตเวิร์กภายในคือเกตเวย์ภายใน แต่ apple และ orange เป็นเกตเวย์ภายนอก

แต่ละเกตเวย์ภายนอกไม่สื่อสารกับเกตเวย์ภายนอกอื่น แต่ เกตเวย์ภายนอกรับชุดของ neighbors (เกตเวย์ภายนอก อื่น) ซึ่งมีการสื่อสารกัน neighbors เหล่านี้ไม่ถูกกำหนดโดยระยะทางภูมิศาสตร์, แต่โดยการสื่อสารที่สร้างขึ้นระหว่างกัน เกตเวย์ neighboring, มี neighbors เกตเวย์ภายนอกอื่น ในวิธีนี้ตารางการจัดเส้นทางเกตเวย์ภายนอกถูกอัพเดตและข้อมูลการจัดเส้นทาง ถูกถ่ายทอดระหว่างเกตเวย์ภายนอก

ข้อมูลการจัดเส้นทางถูกส่งเป็นคู่ (N,D), โดยที่ N คือเน็ตเวิร์ก และ D คือระยะที่สะท้อนถึงภาระในการติดต่อเน็ตเวิร์กที่ระบุ แต่ละเกตเวย์ประกาศข้อมูลเน็ตเวิร์กที่สามารถติดต่อและภาระในการ ติดต่อเน็ตเวิร์ก เกตเวย์ที่ได้รับคำนวณพาธที่สั้นที่สุดไปที่เน็ตเวิร์กอื่น และส่งข้อมูลนี้ไปที่ neighbors ของตัวเอง ดังนั้นแต่ละเกตเวย์ภายนอก ได้รับข้อมูลการจัดเส้นทางอย่างต่อเนื่อง ทำการอัพเดตตารางการจัดเส้นทางแล้ว ส่งข้อมูลนั้นไปที่ neighbors ภายนอกของตัวเอง

โปรโตคอลเกตเวย์:

เกตเวย์ทั้งหมด ไม่ว่าภายในหรือภายนอก จะใช้โปรโตคอลเพื่อสื่อสาร ระหว่างกัน ต่อไปนี้เป็นคำอธิบายอย่างย่อสำหรับโปรโตคอลเกตเวย์ **TCP/IP** ที่ ใช้ทั่วไป:

HELLO Protocol (HELLO)

HELLO เป็นโปรโตคอลหนึ่งที่เกตเวย์ภายในใช้เพื่อสื่อสาร ระหว่างกัน **HELLO** คำนวณหาเส้นทางไปยังเน็ตเวิร์กอื่นที่สั้นที่สุด โดยการพิจารณาเส้นทางที่มีเวลาหน่วงน้อยที่สุด

Routing Information Protocol (RIP)

Routing Information Protocol คือโปรโตคอลที่เกตเวย์ภายใน ใช้เพื่อสื่อสารระหว่างกัน เหมือนกับ **HELLO Protocol** คือ **RIP** จะคำนวณ เส้นทางไปยังเน็ตเวิร์กอื่นที่สั้นที่สุด แต่ต่างจาก **HELLO** ตรงที่ **RIP** ประมาณ ระยะห่าง ไม่ใช่จากเวลาหน่วง แต่เป็นการนับจำนวน hop เนื่องจาก **gated** daemon เก็บค่าเมทริกทั้งหมดภายในเป็นค่าหน่วงเวลา โดยแปลงจำนวน hop ของ **RIP** ให้เป็นค่าการหน่วงเวลา

Routing Information Protocol Next Generation

RIPng คือโปรโตคอล **RIP** ที่ได้รับการปรับปรุงเพื่อให้การสนับสนุน **IPv6**

Open Shortest Path First (OSPF)

OPSF คือโปรโตคอลที่เกตเวย์ภายในใช้เพื่อสื่อสาร ระหว่างกัน โดยเป็นโปรโตคอลค่าสถานะลิงก์ที่มีความเหมาะสมมากกว่า **RIP** สำหรับ เน็ตเวิร์กที่ซับซับที่มีเราเตอร์จำนวนมาก โดยจัดให้มีการจัดเส้นทางหลายทางที่มีค่าใช้จ่ายเท่ากัน

Exterior Gateway Protocol (EGP)

เกตเวย์ภายนอกสามารถใช้ **Exterior Gateway Protocol** เพื่อ สื่อสารระหว่างกัน **EGP** ไม่คำนวณเส้นทาง ไปยังเน็ตเวิร์กอื่นที่สั้นที่สุด แต่จะระบุว่าเน็ตเวิร์กที่เจาะจง สามารถเข้าถึงได้หรือไม่เท่านั้น

Border Gateway Protocol (BGP)

เกตเวย์ภายนอกสามารถใช้โปรโตคอลนี้เพื่อสื่อสารระหว่างกัน โดยแลกเปลี่ยนข้อมูลการเข้าถึงได้ระหว่างระบบ autonomous แต่มี ความสามารถมากกว่า **EGP BGP** ใช้แอ็ตทริบิวต์พาธเพื่อให้ ข้อมูลเพิ่มเติมเกี่ยวกับแต่ละเส้นทาง เพื่อช่วยในการเลือกเส้นทางที่ดีที่สุด

Border Gateway Protocol 4+

BGP4+ คือโปรโตคอล **BGP** เวอร์ชัน 4 ซึ่งสนับสนุน **IPv6** และ มีการเพิ่มประสิทธิภาพอื่นๆ เหนือกว่าโปรโตคอลเวอร์ชันที่ผ่านมา

Intermediate System to Intermediate System (IS-IS)

เกตเวย์ภายในใช้โปรโตคอล **IS-IS** เพื่อสื่อสารระหว่างกัน โดยเป็นโปรโตคอลสถานะลิงก์ที่สามารถจัดเส้นทางแพ็กเก็ต IP และ ISO/CLNP และเหมือนกับ **OSPF** ตรงที่ใช้อัลกอริทึม "เลือกพาธสั้นกว่าก่อน" เพื่อพิจารณาเส้นทาง

ข้อควรพิจารณาเกี่ยวกับเกตเวย์

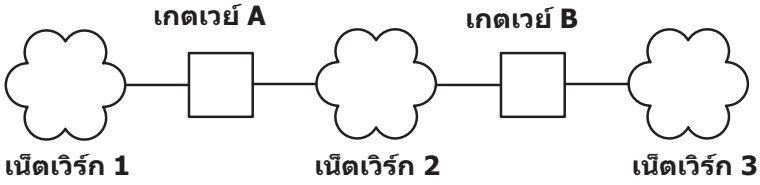
ดำเนินการเหล่านี้ก่อนกำหนดค่าเกตเวย์ของคุณ

ก่อนที่คุณจะกำหนดค่าเกตเวย์สำรับเน็ตเวิร์กของคุณ คุณ ต้องทำสิ่งต่อไปนี้ก่อน:

1. พิจารณาจำนวนเกตเวย์ที่จะใช้ จำนวน ของเกตเวย์ที่คุณจำเป็นต้องกำหนดค่าจะขึ้นอยู่กับ:
   - จำนวนเน็ตเวิร์กที่คุณต้องการเชื่อมต่อ
   - วิธีที่คุณต้องการเชื่อมต่อเน็ตเวิร์ก

- ระดับกิจกรรมบนเน็ตเวิร์กที่เชื่อมต่อ

ตัวอย่างเช่น สมมติผู้ใช้บนเน็ตเวิร์ก 1, เน็ตเวิร์ก 2 และเน็ตเวิร์ก 3 ทั้งหมดจำเป็นต้องสื่อสารระหว่างกัน



รูปที่ 26. การกำหนดค่าเกตเวย์แบบง่าย

ภาพประกอบนี้มีภาพเมฆแสดงเน็ตเวิร์กสามภาพ หมายเลขหนึ่ง สอง และสาม เน็ตเวิร์กหนึ่ง และสองเชื่อมต่อกับเกตเวย์ A เน็ตเวิร์กสองและสามเชื่อมต่อกับเกตเวย์ B

เมื่อต้องการเชื่อมต่อ เน็ตเวิร์ก 1 กับเน็ตเวิร์ก 2 โดยตรง คุณควรใช้เกตเวย์เดียว (เกตเวย์ A) เมื่อต้องการเชื่อมต่อเน็ตเวิร์ก 2 กับเน็ตเวิร์ก 3 โดยตรง คุณควรใช้อีก เกตเวย์หนึ่ง (เกตเวย์ B) ขณะนี้ สมมติว่าเส้นที่เหมาะสมได้ถูกกำหนดแล้ว ผู้ใช้ทั้งหมดบนเน็ตเวิร์กทั้งสามสามารถสื่อสารถึงกันได้

อย่างไรก็ตาม ถ้าเน็ตเวิร์ก 2 ยุ่งมาก การสื่อสารระหว่างเน็ตเวิร์ก 1 และเน็ตเวิร์ก 3 จะประสบปัญหาการหน่วงที่ไม่สามารถยอมรับได้ นอกจากนั้น ถ้าการสื่อสารระหว่างเน็ตเวิร์ก ส่วนมากเกิดขึ้นระหว่างเน็ตเวิร์ก 1 และเน็ตเวิร์ก 3 คุณอาจต้องการ เชื่อมต่อเน็ตเวิร์ก 1 กับเน็ตเวิร์ก 3 โดยตรง เมื่อต้องการสิ่งนี้ คุณควร ใช้คู่เกตเวย์เพิ่ม คือเกตเวย์ C (บนเน็ตเวิร์ก 1) และเกตเวย์ D (บนเน็ตเวิร์ก 3) ที่มีการเชื่อมต่อตรงระหว่างสองเกตเวย์เพิ่ม เหล่านี้ อย่างไรก็ตาม วิธีนี้อาจเป็นการแก้ไขที่ไม่เพียงพอ เนื่องจาก เกตเวย์หนึ่งสามารถเชื่อมต่อได้มากกว่าสองเน็ตเวิร์ก

แนวทางที่มี ประสิทธิภาพมากยิ่งขึ้นคือการเชื่อมต่อเกตเวย์ A กับเกตเวย์ B โดยตรง เช่นเดียวกับ เน็ตเวิร์ก 2 วิธีนี้จะต้องใช้เน็ตเวิร์กอะแด็ปเตอร์ที่สองในทั้ง เกตเวย์ A และเกตเวย์ B โดยทั่วไป จำนวนเน็ตเวิร์กที่คุณเชื่อมต่อ ผ่านเกตเวย์หนึ่งจะถูกจำกัดโดยจำนวนเน็ตเวิร์กอะแด็ปเตอร์เชื่อมต่อ ที่เครื่องเกตเวย์สามารถรองรับได้

2. ตัดสินใจเกี่ยวกับชนิดการจัดเส้นทางที่จะใช้

   ถ้า เน็ตเวิร์กของคุณมีขนาดเล็ก และการเปลี่ยนแปลงการกำหนดค่าไม่บ่อย คุณอาจ ต้องการใช้การจัดเส้นทางแบบสแตติก แต่ถ้าคุณมีเน็ตเวิร์กขนาดใหญ่ที่ มีการเปลี่ยนแปลงการกำหนดค่าบ่อยๆ คุณอาจต้องการใช้การจัดเส้นทาง แบบไดนามิก คุณอาจตัดสินใจใช้รวมกันทั้งการจัดเส้นทางแบบสแตติก และแบบไดนามิก นั่นคือ คุณอาจต้องการกำหนดนิยามสแตติกให้แก่เส้นทางที่เจาะจง จำนวนไม่มาก ขณะที่อนุญาตให้เส้นทางอื่นถูกอัพเดตโดย daemons เส้นทางแบบสแตติกที่คุณสร้างจะไม่ถูกประกาศไปยังเกตเวย์ อื่นๆ และไม่ถูกอัพเดตโดย daemons การจัดเส้นทาง

3. ถ้าคุณกำลังใช้การจัดเส้นทางแบบไดนามิก ให้เลือก daemon การจัดเส้นทาง ตามชนิดเกตเวย์ที่คุณต้องใช้ และโปรโตคอลที่เกตเวย์ของคุณ ต้องสนับสนุน ถ้าเกตเวย์เป็นเกตเวย์ภายใน และ ให้จำเป็นต้องสนับสนุนเฉพาะ **RIP** ให้เลือก **routed** daemon ถ้าเกตเวย์ต้องสนับสนุนโปรโตคอลอื่นๆ หรือเป็นเกตเวย์ ภายนอก ให้เลือก **gated** daemon

   หมายเหตุ: ผลลัพธ์ ที่ไม่คาดคิดสามารถเกิดขึ้นได้ถ้า **gated** และ **routed** daemons ทำงานอยู่บนโฮสต์เดียวกันในเวลาเดียวกัน

การกำหนดค่าเกตเวย์

เมื่อต้องการกำหนดค่าเครื่องให้ทำหน้าที่เป็นเกตเวย์ ให้ใช้คำสั่งเหล่านี้

เพื่อความชัดเจน โพรซีเดอร์นี้จะถือว่าเครื่องเกตเวย์ เชื่อมต่อกับสองเน็ตเวิร์ก และเครื่องเกตเวย์ได้ถูกกำหนดค่าอย่างน้อยที่สุดไว้แล้ว บนเครื่องใดเครื่องหนึ่งของเน็ตเวิร์ก

1. ติดตั้งและกำหนดค่าเน็ตเวิร์กอะแด็ปเตอร์ทั้งสอง ถ้าคุณยังไม่ได้ ติดตั้ง (ดูที่ "การติดตั้งเน็ตเวิร์กอะแด็ปเตอร์" ในหน้า 173 และ "การจัดการและตั้งค่าอะแด็ปเตอร์" ในหน้า 174)
2. เลือก IP address สำหรับเน็ตเวิร์กอินเตอร์เฟสที่สอง จากนั้น กำหนดค่าเน็ตเวิร์กอินเตอร์เฟสโดยปฏิบัติตามคำแนะนำใน "การจัดการเน็ตเวิร์กอินเตอร์เฟส" ในหน้า 189
3. เพิ่มเส้นทางที่จะไปยังเน็ตเวิร์กที่สอง
4. เมื่อต้องการใช้เครื่องเป็นเราเตอร์ระหว่างเน็ตเวิร์กบน **TCP/IP** ให้พิมพ์:

   ```
   no -o ipforwarding=1
   ```

ถึงตอนนี้เครื่องเกตเวย์สามารถเข้าถึงทั้งสองเน็ตเวิร์กซึ่ง เชื่อมต่อกันโดยตรงได้

1. ถ้าคุณต้องการใช้การจัดเส้นทางแบบสแตติกเพื่อสื่อสารกับโฮสต์ หรือเน็ตเวิร์ก นอกเหนือจากสองเน็ตเวิร์กนี้ ให้เพิ่มเส้นทางอื่นที่คุณต้องการ
2. ถ้าคุณต้องการใช้การจัดเส้นทางแบบไดนามิก ให้ปฏิบัติตามคำแนะนำใน "การตั้งค่า routed daemon" ในหน้า 385 หรือ "การกำหนดค่า gated daemon" ในหน้า 386 ถ้าอินเตอร์เน็ตเวิร์กของคุณจะรวมกับอินเตอร์เน็ต คุณควรปฏิบัติตามคำแนะนำ ใน "ตัวเลขระบบ Autonomous" ในหน้า 389 ด้วย

ตารางที่ 80. การกำหนดค่างานเกตเวย์

ภารกิจ	วิธีลัด SMIT	ไฟล์คำสั่ง
การแสดงตารางการจัดเส้นทาง	`smit lsroute`	**netstat -rn**[1]
การเพิ่มเส้นทางแบบสแตติก	`smit mkroute`	**route add** *destination gateway*[2]
การลบเส้นทางแบบสแตติกออก	`smit rmroute`	**route delete** *destination gateway*[2]
การลบตารางการจัดเส้นทาง	`smit fshrttbl`	**route flush**

หมายเหตุ:

1. ตารางถูกแบ่งออกเป็นคอลัมน์สำหรับแอดเดรสปลายทาง แอดเดรสเกตเวย์ แฟล็ก จำนวนการอ้างอิง (จำนวน hop) และเน็ตเวิร์กอินเตอร์เฟส (สำหรับคำอธิบาย โดยละเอียดสำหรับแต่ละคอลัมน์ โปรดดูที่คำสั่ง **netstat** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*) ถ้าเฟรมไปไม่ถึงปลายทาง และตารางการจัดเส้นทางระบุ เส้นทางที่ถูกต้อง แสดงว่ามีเงื่อนไขอย่างน้อยหนึ่งเงื่อนไขต่อไปนี้อยู่:
   - เน็ตเวิร์กล้มเหลว
   - โฮสต์หรือเกตเวย์รีโมตล้มเหลว
   - โฮสต์หรือเกตเวย์รีโมตหยุดทำงาน หรือไม่พร้อมรับเฟรม
   - โฮสต์รีโมตไม่มีเส้นทางกลับไปยังเน็ตเวิร์กต้นทาง
2. ค่า *destination* คือแอดเดรสแบบจุด หรือชื่อสัญลักษณ์ของโฮสต์หรือเน็ตเวิร์กปลายทาง และค่า *gateway* เป็นแอดเดรสจุด หรือชื่อสัญลักษณ์ของเกตเวย์ (เส้นทาง ดีฟอลต์ระบุ 0 เป็นปลายทาง)

ข้อจำกัดการใช้เส้นทาง

เส้นทางสามารถถูกจำกัดเพื่อที่สามารถถูกใช้เฉพาะโดยบางผู้ใช้ ข้อจำกัดมาจาก ID กลุ่มหลักของผู้ใช้

การใช้ **route** คุณสามารถระบุรายการได้ถึง 32 ID กลุ่มที่ได้หรือไม่ได้รับอนุญาต ให้ใช้เส้นทาง ถ้ารายการ เป็นกลุ่มที่ได้รับอนุญาน ผู้ใช้ที่เป็นสมาชิกกลุ่มในรายการสามารถใช้ เส้นทางได้ ถ้ารายการเป็นของกลุ่มที่ไม่ได้รับอนุญาต เฉพาะผู้ใช้ที่ไม่ได้เป็น ของกลุ่มในรายการที่สามารถใช้เส้นทางได้ ผู้ใช้ root สามารถใช้ เส้นทางได้ทั้งหมด

กลุ่มยังสามารถถูกเชื่อมโยงกับอินเตอร์เฟสโดยใช้คำสั่ง **ifconfig** ในกรณีนี้ แพ็กเก็ตที่ส่งต่อได้สามารถใช้เส้นทางทั้งหมดที่อนุญาตสำหรับกลุ่ม ที่เชื่อมโยงกับอินเตอร์เฟสขาเข้าของตัวเอง

ถ้ามีเส้นทางสองเส้นทางหรือมากกว่านั้นไปที่ปลายทางเดียวกัน การเปลี่ยนทิศทาง ICMP ที่ได้รับสำหรับปลายทางนั้นจะถูกละเว้นและพาธ MTU discovery จะไม่ถูกดำเนินการบนเส้นทางเหล่านั้น

การตรวจหาเกตเวย์ที่ไม่ทำงาน

โฮสต์สามารถมีการกำหนดคอนฟิกเพื่อตรวจหาว่าเกตเวย์ที่โฮสต์ ใช้อยู่ไม่ทำงานหรือไม่ และสามารถปรับตารางการเราต์ตามนั้น

ถ้าเน็ตเวิร์กอ็อพชัน **-passive_dgd** คือ 1, การตรวจหา เกตเวย์ที่ไม่ทำงานแบบพาสซีฟถูกเปิดใช้งานสำหรับทั้งระบบ ถ้า ไม่ได้รับการตอบกลับสำหรับคำร้องขอ **dgd_packets_lost ARP** ที่ต่อเนื่อง ไปยังเกตเวย์ จะสมมติว่าเกตเวย์ไม่ทำงานและเมทริกระยะทาง (เรียกอีกอย่างว่า *hopcount* หรือ *cost*) สำหรับเราต์ทั้งหมด ที่ใช้เกตเวย์นั้นจะถูกเพิ่มเป็นค่าสูงสุดที่เป็นไปได้ หลังจากเวลา **dgd_retry_time** นาที ผ่านไป ภาระของเส้นทางถูกเรียกคืนไปเป็นค่า user-configured และโฮสต์ยังมีการดำเนินการขึ้นกับการเชื่อมต่อ **TCP** ที่ล้มเหลว ถ้าแพ็กเก็ต **dgd_packets_lost TCP** ต่อมาสูญหา รายการ **ARP** สำหรับเกตเวย์ที่ใช้อยู่ถูกลบและการเชื่อมต่อ **TCP** พยายามใช้เส้นทางที่ดีที่สุดถัดไป ในครั้งถัดไปที่เกตเวย์ถูกใช้ การดำเนินการด้านบน จะถูกนำมาใช้ ถ้าเกตเวย์ไม่ทำงาน พารามิเตอร์ **passive_dgd, dgd_packets_lost** และ **dgd_retry_time** สามารถกำหนดคอนฟิกได้ทั้งหมดโดยใช้ คำสั่ง **no**

โฮสต์ยังสามารถถูกตั้งค่าให้ใช้การตรวจหาเกตเวย์ที่ไม่ทำงานแอ็คทีฟในแบบ per-route กับแฟล็ก **-active_dgd** ของคำสั่ง **route** การตรวจหาเกตเวย์ที่ไม่ทำงานซึ่งแอ็คทีฟอยู่ pings เกตเวย์ทั้งหมดที่ใช้โดยเราต์ ซึ่ง เปิดใช้งานในทุก **dgd_ping_time** วินาที ถ้าไม่ได้รับการตอบสนอง จากเกตเวย์ จะทำการ ping เร็วขึ้นถึง **dgd_packets_lost** เท่า ถ้ายังคงไม่มีการตอบกลับ ค่าใช้จ่ายของเราต์ทั้งหมดที่ใช้เกตเวย์ นั้นจะเพิ่มขึ้น เกตเวย์จะถูก ping ต่อไป และถ้าได้รับการตอบสนองในที่สุด ภาระบนเส้นทางถูกเรียกคืนไปเป็นค่า user-configured พารามิเตอร์ **dgd_ping_time** สามารถกำหนดคอนฟิกได้โดยใช้คำสั่ง **no**

การตรวจหาเกตเวย์ที่ไม่ทำงานมีประโยชน์มากที่สุดสำหรับโฮสต์ที่ใช้การ จัดเส้นทางแบบสแตติก มากกว่าแบบไดนามิก การตรวจหาเกตเวย์ที่ไม่ทำงานแบบพาสซีฟมี ปัญหาด้านประสิทธิภาพน้อยกว่า และแนะนำให้ใช้บนเครือข่ายใดๆ ที่มีเกตเวย์มาก อย่างไรก็ตาม การตรวจหาเกตเวย์ที่ไม่ทำงานพาสซีฟถูกดำเนินการในแบบ best-effort เท่านั้น บางโปรโตคอล เช่น **UDP**, ไม่จัดเตรียมผลตอบกลับไปที่โฮสต์ ถ้าการส่งข้อมูลล้มเหลว และในกรณีนี้ไม่มีการดำเนินการใดที่กระทำโดยการตรวจหาเกตเวย์ที่ไม่ทำงาน พาสซีฟ

การตรวจหาเกตเวย์ที่ไม่ทำงานแอ็คทีฟ มีประโยชน์ที่สุดเมื่อโฮสต์ต้องค้นพบ ในทันที เมื่อเกตเวย์ไม่ทำงาน เนื่องจากมีการเคียวรีแต่ละเกตเวย์ ที่ถูกเปิดใช้งาน ในทุกสองถึงสามวินาที ดังนั้น จึงมีการใช้งานเครือข่ายเกินกว่า ระดับที่เชื่อมโยง การตรวจหาเกตเวย์ที่ไม่ทำงานแอ็คทีฟ แนะนำให้ใช้เฉพาะกับโฮสต์ ที่จัดเตรียมเซอร์วิสที่สำคัญ และบนเน็ตเวิร์กที่มีจำนวนโฮสต์จำกัด

หมายเหตุ: การตรวจหาเกตเวย์ที่ไม่ทำงานและโปรโตคอลการเราต์ ที่ใช้โดย **gated** และ **routed** daemons ทำ ฟังก์ชันที่คล้ายกัน โดยการค้นหาการเปลี่ยนแปลงในคอนฟิกูเรชันเครือข่าย และปรับตารางการเราต์ตามนั้น อย่างไรก็ตาม จะใช้กลไกที่ต่างกันในการ ดำเนินการ และถ้าถูกรันพร้อมกัน อาจมีความขัดแย้ง ระหว่างกัน ด้วยเหตุผลนี้ ต้องไม่ใช้การตรวจหาเกตเวย์ที่ไม่ทำงาน บนระบบที่รัน **gated** หรือ **routed** daemons

เมื่อการตรวจหาเกตเวย์ที่ไม่ทำงานตรวจพบว่าเราต์หลัก ย้อนกลับไปออนไลน์ และพารามิเตอร์ **dgd_flush_cached_route** มีการ เปิดใช้งาน เราต์ที่แคชไว้ปัจจุบันของการเชื่อมต่อที่แอ็คทีฟทั้งหมดจะถูกฟลัช เราต์ของการเชื่อมต่อที่แอ็คทีฟปัจจุบันทั้งหมดจะถูกตรวจสอบความถูกต้องอีกครั้ง เพื่อค้นหาเราต์ที่ดีที่สุดสำหรับการส่งข้อมูล พารามิเตอร์ **dgd_flush_cached_route** สามารถกำหนดคอนฟิกได้โดยใช้คำสั่ง **no** โดย ดีฟอลต์ พารามิเตอร์ **dgd_flush_cached_route** ปิดใช้งาน

หมายเหตุ: พารามิเตอร์ **dgd_flush_cached_route** ต้องเปิดใช้งานเฉพาะในสภาวะแวดล้อมเครือข่ายที่มั่นคง มิฉะนั้น อาจมีปัญหาด้านประสิทธิภาพมากขึ้นเนื่องจากฮาร์ดแวร์เราเตอร์ที่ไม่ดีหรือ ไม่มั่นคง ส่งผลให้การตรวจหาเกตเวย์ที่ไม่ทำงานอัพเดตตารางการเราต์ บ่อยครั้ง การฟลัชเราต์ที่แคชไว้บ่อยครั้งยังเสียค่าใช้จ่ายสูงด้วย

การโคลนเส้นทาง

การโคลนเส้นทางยอมให้สร้างเส้นทางโฮสต์สำหรับ ทุกโฮสต์ที่ระบบสื่อสารด้วย

เมื่อการรับส่งข้อมูลจะเป็นการส่ง การค้นหาถูกดำเนินในตารางเส้นทาง เพื่อค้นหาเส้นทางไปยังโฮสต์นั้น ถ้าพบเส้นทางโฮสต์ที่เจาะจง เส้นทาง นั้นจะถูกใช้ ถ้าไม่พบเส้นทางโฮสต์ที่เจาะจง อาจพบเส้นทางเน็ตเวิร์ก หรือสเนทางดีฟอลต์ ถ้าเส้นทางที่แฟล็กแสดงการโคลน 'c' ถูกตั้งค่า เส้นทางโฮสต์สำหรับปลายาทงจะสร้างขึ้นโดยใช้เกตเวย์ จากเส้นทางที่กำลังถูกโคลน ตารางการจัดเส้นทางต่อมาที่ค้นหา ปลายทางนั้นจะถบเส้นทางโฮสต์ที่ถูกโคลน เส้นทางที่ถูกโคลนมีแฟล็ก 'W' ถูกตั้งค่า เส้นทางเหล่านี้จะมีการหมดเวลาใช้งาน และถูกลบออกจากตารางการจัดเส้นทาง ถ้าไม่ถูกใช้งานเป็นเวลา *route_expire* นาที คุณสามารถแก้ไข *route_expire* ได้โดยใช้คำสั่ง **no**

คุณลักษณะการลอกแบบเราต์ถูกใช้อย่างมากโดยโปรโตคอลการค้นหา path MTU ภายในระบบปฏิบัติการ AIX เพื่อให้สามารถติดตามข้อมูล path MTU สำหรับทุก ปลายทางที่สื่อสารด้วย ถ้าอ็อพชันเน็ตเวิร์ก **tcp_pmtu_discover** หรือ **udp_pmtu_discover** (ตั้งค่าได้ ด้วยคำสั่ง **no**) คือ 1 แฟล็กการโคลนจะถูกเปิดใช้ สำหรับเส้นทางเน็ตเวิร์กทั้งหมดบนระบบ โปรโตคอลการค้นหา path MTU เปิดโดยดีฟอลต์

หมายเหตุ: เมื่อต้องการเพิ่มรายการจัดเส้นทางการโคลนด้วยตนเอง คุณสามารถ แก้ไขตารางการจัดเส้นทางผ่านคำสั่ง **route**

ข้อมูลที่เกี่ยวข้อง:
คำสั่ง route

การลบเส้นทางแบบไดนามิก

ถ้าคุณใช้ **routed** daemon เส้นทางที่ถูกลบแบบแมนวลจะ*ไม่*ถูกแทนที่โดยข้อมูล RIP ขาเข้า (เนื่องจาก ioctl ถูกใช้)

ถ้าคุณใช้ **gated** daemon และแฟล็ก **–n** ไม่ถูกใช้ เส้นทางที่ถูกลบแบบแมนวล จะ ถูกแทนที่โดยข้อมูลโดยเส้นทางที่ถูกค้นพบในข้อมูล RIP ขาเข้า

การตั้งค่า routed daemon

ทำตามขั้นตอนเหล่านี้เพื่อตั้งค่า **routed** daemon

เมื่อต้องการตั้งค่า **routed** daemon:

1. ลบสัญลักษณ์ความคิดเห็น (#) และแก้ไขวลี routed ในเชลล์สคริปต์ `/etc/rc.tcpip` นี่จะเป็นการ สตาร์ท **routed** daemon โดยอัตโนมัติพร้อมกับแต่ละ ระบบเริ่มทำงาน
   - ระบุว่าคุณต้องการให้เกตเวย์รันในโหมด active (แฟล็ก **–s**) หรือ passive (แฟล็ก **–q**)

- ระบุว่าคุณต้องการให้การติดตามแพ็กเก็ต เปิดหรือปิด (แฟล็ก **-t**) การติดตาม แพ็กเก็ตยังสามารถถูกเปิดหลังจาก **routed** daemon ถูกสตาร์ทแล้วโดยใช้คำสั่ง **kill** เพื่อส่งสัญญาณ **SIGUSR1** ไปที่ daemon สัญญาณนี้ยังสามารถถูกใช้เพื่อเพิ่มระดับของการติดตาม ผ่านสี่ระดับ นอกจากนี้ การติดตามแพ็กเก็ตสามารถถูกปิดขณะที่ **routed** daemon รันอยู่โดยใช้คำสั่ง **kill** เพื่อส่งสัญญาณ **SIGUSR2** ไปที่ daemon สำหรับข้อมูลเพิ่มเติม ดูที่ **routed** daemon และคำสั่ง **kill**
- ระบุว่าคุณต้องการให้เปิดหรือปิดการดีบัก (แฟล็ก **-d**) ถ้าคุณใช้แฟล็กนี้ ระบุไฟล์บันทึกการทำงานที่คุณต้องการเก็บข้อมูล การดีบัก หรือเลือกไว้เพื่อให้ถูกกำหนดไปที่จอแสดงผลคอนโซล
- ระบุว่าคุณกำลังรัน **routed** daemon บน เกตเวย์ (แฟล็ก **-g**) หรือไม่

  หมายเหตุ: โฮสต์ที่ไม่ใช่เกตเวย์สามารถรัน **routed** daemon, แต่ต้องถูกรันในโหมด passive

2. ระบุเน็ตเวิร์กที่รู้จักโดยการแสดงในไฟล์ `/etc/networks` ดูที่ Networks File Format for TCP/IP ใน *การอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติม ไฟล์ `networks` ตัวอย่าง อยู่ในไดเร็กทอรี `/usr/samples/tcpip`
3. เซ็ตอัพเส้นทางในไฟล์ `/etc/gateways` ไปที่ เกตเวย์ที่รู้จักที่ไม่ได้ถูกเชื่อมต่อโดยตรงไปที่เน็ตเวิร์กของคุณ อ้างถึง Gateways File Format for TCP/IP ใน *การอ้างอิงไฟล์* สำหรับ ตัวอย่างละเอียดของรายการในไฟล์ `/etc/gateways` ไฟล์ `gateways` ตัวอย่างอยู่ในไดเร็กทอรี `/usr/samples/tcpip`

ข้อควรสนใจ: อย่ารัน **routed** daemon และ **gated** daemon บนเครื่องเดียวกัน อาจเกิดผลลัพธ์ ที่ไม่คาดคิดขึ้น

การกำหนดค่า gated daemon

เมื่อกำหนดค่า **gated** daemon คุณต้อง ตัดสินใจว่าโปรโตคอลเกตเวย์ใดที่เหมาะกับระบบของคุณมากที่สุด

เมื่อต้องการกำหนดค่า **gated** daemon:

1. ตัดสินใจว่าโปรโตคอลเกตเวย์ใดที่เหมาะกับระบบของคุณมากที่สุด ตัวเลือกสำหรับโปรโตคอลการจัดเส้นทางได้แก่ **EGP, BGP, RIP, RIPng, HELLO, OSPF, ICMP/Router Discovery** และ **IS-IS** คุณยังสามารถใช้ **SNMP** โปรโตคอลอนุญาตให้คุณเปลี่ยนแปลงหรือแสดงข้อมูลการจัดการสำหรับองค์ประกอบเน็ตเวิร์ก จากโฮสต์รีโมต

   หมายเหตุ: ใช้ **EGP, BGP** หรือ **BGP4+** เพื่อประกาศ แอดเดรสของเน็ตเวิร์กในระบบ autonomous ไปยังเกตเวย์ในระบบ autonomous อื่น ถ้าคุณอยู่บนอินเตอร์เน็ต **EGP, BGP** หรือ **BGP4+** ต้องถูกใช้เพื่อประกาศความสามารถในการเข้าถึงเน็ตเวิร์กได้ไปยังระบบเกตเวย์หลัก ใช้โปรโตคอลการจัดเส้นทางภายในเพื่อประกาศข้อมูลความสามารถในการเข้าถึงได้ภายใน ระบบ autonomous
2. ระบุเน็ตเวิร์กที่รู้จักโดยการแสดงรายการเน็ตเวิร์กในไฟล์ `/etc/networks` ดูที่ รูปแบบไฟล์เน็ตเวิร์กสำหรับ TCP/IP ใน *การอ้างอิงไฟล์* สำหรับ ข้อมูลเพิ่มเติม ตัวอย่างไฟล์ `networks` อยู่ใน ไดเร็กทอรี `/usr/samples/tcpip`
3. แก้ไขไฟล์ `/etc/gated.conf` เพื่อให้มี การกำหนดค่า **gated** daemon ที่ต้องการ

   หมายเหตุ: เวอร์ชัน **gated** บน AIX 4.3.2 และสูงกว่าคือ 3.5.9 ไวยากรณ์ของไฟล์ `/etc/gated.conf` มีการเปลี่ยนแปลง ตัวอย่าง ที่กำหนดด้านล่างใช้สำหรับ **gated** เวอร์ชัน 3.5.9 เมื่อต้องการกำหนดค่าไฟล์ `/etc/gated.conf` สำหรับเวอร์ชันก่อนหน้า AIX 4.3.2 ให้ใช้ไวยากรณ์ที่กำหนดให้ ในไฟล์ `/etc/gated.conf` เอง

   a. ระบุระดับเอาต์พุตการติดตามที่คุณต้องการ ถ้าจำเป็นต้องมีการติดตาม ก่อนที่ไฟล์ `gated.conf` จะถูกวิเคราะห์คำให้ใช้ แฟล็ก **-t** เพื่อเปิดการติดตามเมื่อ daemon เริ่มทำงาน ดูที่ gated Daemon in *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2* สำหรับข้อมูลเพิ่มเติม

   b. ระบุโปรโตคอลการจัดเส้นทางที่คุณต้องการใช้ แต่ละ โปรโตคอลมีข้อความโปรโตคอลของตนเอง ลบสัญลักษณ์ความคิดเห็น (#) และแก้ไขคำสั่งใดๆ ที่เกี่ยวข้องกับโปรโตคอลที่คุณต้องการใช้

      - ถ้าใช้ **EGP**:

- ตั้งค่าประโยคย่อย **EGP** `autonomoussystem` จัดหา ตัวเลขระบบ autonomous จากหน่วยงานอินเตอร์เน็ต ถ้าคุณอยู่บน อินเตอร์เน็ต หรือถ้าไม่ ให้กำหนดตัวเลขระบบ autonomous โดยพิจารณาตัวเลขระบบ autonomous ของระบบอื่นๆ บนเน็ตเวิร์กของคุณ
- ตั้งค่าคำสั่ง **EGP** เป็น `yes`
- ตั้งค่าประโยคย่อย `group` สำหรับแต่ละระบบ autonomous
- ตั้งค่าประโยคย่อย `neighbor` สำหรับแต่ละเครื่องที่อยู่ใกล้เคียงระบบ autonomous นั้น ตัวอย่างเช่น:

```
autonomoussystem 283 ;

egp yes {
        group maxup 1 {
             neighbor nogendefault 192.9.201.1 ;
             neighbor nogendefault 192.9.201.2 ;
       } ;
       group {
             neighbor 192.10.201.1 ;
             neighbor 192.10.201.2 ;
       } ;
} ;
```

- ถ้าใช้ **RIP** หรือ **HELLO**:
  - ตั้งค่าคำสั่ง **RIP** หรือ **HELLO** เป็น `yes`
  - ระบุ `nobroadcast` ในคำสั่ง **RIP** or **HELLO** ถ้าคุณต้องการให้เกตเวย์ยอมข้อมูลการจัดเส้นทางเท่านั้น ไม่มีการกระจาย ข้อมูล หรือระบุ `broadcast` ในคำสั่ง **RIP** หรือ **HELLO** ถ้าคุณต้องการให้เกตเวย์กระจายข้อมูลการจัดเส้นทางรวมทั้งรับ ข้อมูลการจัดเส้นทาง
  - ถ้าคุณต้องการให้เกตเวย์ส่งไปยังเกตเวย์ต้นทางโดยตรง ให้ใช้คำสั่ง `sourcegateways` ระบุชื่อเกตเวย์ หรืออินเตอร์เน็ตแอดเดรสในรูปแบบจุดทศนิยมในประโยคย่อย `sourcegateways` ตัวอย่างเช่น:

```
# Send directly to specific gateways

rip/hello yes {
          sourcegateways
                101.25.32.1
                101.25.32.2 ;
} ;
```

  ตัวอย่างต่อไปนี้แสดง **RIP/HELLO** stanza ในไฟล์ `gated.conf` ของเครื่องที่ไม่ส่งแพ็กเก็ต **RIP** และไม่รับแพ็กเก็ต **RIP** บนอินเตอร์เฟส tr0

```
rip/hello nobroadcast {
        interface tr0 noripin ;
} ;
```

- ถ้าใช้ **BGP**:
  - ตั้งค่าประโยคย่อย **BGP** `autonomoussystem` จัดหา ตัวเลขระบบ autonomous จากหน่วยงานอินเตอร์เน็ต ถ้าคุณอยู่บน อินเตอร์เน็ต หรือถ้าไม่ ให้กำหนดตัวเลขระบบ autonomous โดยพิจารณาตัวเลขระบบ autonomous ของระบบอื่นๆ บนเน็ตเวิร์กของคุณ
  - ตั้งค่าคำสั่ง **BGP** เป็น `yes`
  - ตั้งค่าประโยคย่อย `peer` สำหรับแต่ละเครื่องที่อยู่ใกล้เคียงระบบ autonomous นั้น ตัวอย่างเช่น:

```
# Perform all BGP operations

bgp yes {
        peer 192.9.201.1 ;
} ;
```

- ถ้าใช้ **SNMP**:
  - ตั้งค่าคำสั่ง **SNMP** เป็น yes

    ```
    snmp yes ;
    ```

การกำหนดค่า gated daemon เพื่อรัน IPv6:

ใช้โพรซีเดอร์นี้เพื่อกำหนดค่า **gated** daemon เพื่อรัน **Internet Protocol version 6 (IPv6)**

เมื่อต้องการกำหนดค่า **gated** daemon เพื่อรันภายใต้ **Internet Protocol version 6 (IPv6)** อันดับแรกตรวจสอบให้แน่ใจว่าระบบของคุณได้รับการกำหนดค่า สำหรับการจัดเส้นทาง **IPv6** และ **IPv6**:

1. รัน **autoconf6** เพื่อกำหนดค่าอินเตอร์เฟสของคุณ โดยอัตโนมัติสำหรับ **IPv6**
2. กำหนดค่าแอดเดรสโลคัลของไซต์สำหรับแต่ละอินเตอร์เฟส **IPv6** ที่คุณต้องการใช้การจัดเส้นทาง **IPv6** โดยใช้คำสั่งต่อไปนี้:

   ```
   ifconfig interface inet6 fec0:n::address/64 alias
   ```

   โดยที่

   interface
   คือชื่อของอินเตอร์เฟส เช่น tr0 หรือ en0

   n คือเลขทศนิยมใดๆ ตัวอย่างเช่น 11

   address คือส่วนของแอดเดรสอินเตอร์เฟส **IPv6** ที่ตามด้วยโคลอน คู่ ตัวอย่างเช่น **IPv6** address ที่กำหนด fe80::204:acff:fe86:298d รายการ *address* จะเป็น 204:acff:fe86:298d

   หมายเหตุ: คุณสามารถใช้คำสั่ง **netstat -i** เพื่อดูว่า **IPv6** address ของคุณ เป็นค่าใดสำหรับแต่ละอินเตอร์เฟสที่กำหนดค่า

   ถ้าโทเค็นริง tr0 มี **IPv6** address เป็น fe80::204:acff:fe86:298d ให้คุณเรียกใช้คำสั่งต่อไปนี้:

   ```
   ifconfig tr0 inet6 fec0:13::204:acff:fe86:298d/64 alias
   ```

3. เปิดใช้การส่งต่อ **IPv6** ด้วยคำสั่งต่อไปนี้:

   ```
   no -o ip6forwarding=1
   ```

4. เริ่มทำงาน **ndpd-router** ด้วยคำสั่งต่อไปนี้:

   ```
   ndpd-router -g
   ```

   ดูที่ **ndpd-router** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4* เพื่อ พิจารณาว่าจะใช้แฟล็กใดสำหรับการกำหนดค่าเน็ตเวิร์กของคุณ การเริ่มทำงาน **ndpd-router** จะยอมให้ ระบบของคุณทำหน้าที่เป็นเราเตอร์สำหรับ **Neighbor Discovery Protocol** เราเตอร์ **Neighbor Discovery Protocol** แจ้งโฮสต์ Neighbor Discovery โดยใช้ข้อมูล การจัดเส้นทางเพื่อให้โฮสต์สามารถจัดเส้นทางแพ็กเก็ต **IPv6**

โฮสต์ใดๆ บน เน็ตเวิร์กที่คุณต้องการให้เป็นส่วนหนึ่งของเน็ตเวิร์ก **IPv6** ต้องรัน **ndpd-host** โฮสต์บนเน็ตเวิร์กที่รัน **ndpd-host** จะยอมรับว่า ตนเองเป็นส่วนหนึ่งของเน็ตเวิร์ก **IPv6** และใช้ **Neighbor Discovery Protocol** ซึ่งยอมให้สามารถพิจารณา และมอนิเตอร์แอดเดรสในเลเยอร์ของลิงก์ ทั้งเพื่ออนุญาตให้มีการจัดเส้นทางไปที่ใกล้เคียง และเพื่อค้นหาเส้นทางใกล้เคียงสำหรับการส่งต่อ แพ็กเก็ต

ดูที่ **ndpd-router** และ **ndpd-host** in *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4* หรืออ่าน RFC 1970, *Neighbor Discovery* เพื่อดูข้อมูลเพิ่มเติม

5. จากนั้น กำหนดค่า **gated** daemon:
   a. ตัดสินใจว่าโปรโตคอลเกตเวย์ **IPv6** ใดเหมาะกับระบบของคุณ มากที่สุด ตัวเลือกสำหรับโปรโตคอลการจัดเส้นทาง **IPv6** ได้แก่ **Border Gateway Protocol** ที่ปรับปรุงสำหรับ **IPv6** (**BGP4+**) และ **Routing Information Protocol Next Generation** (**RIPng**)
   b. แก้ไขไฟล์ `/etc/gated.conf` เพื่อให้มี การกำหนดค่า **gated** daemon ที่ต้องการ

      หมายเหตุ: AIX 4.3.2 และภายหลังรัน **gated** เวอร์ชัน 3.5.9 ไวยากรณ์ของไฟล์ `gated.conf` มีการเปลี่ยนแปลงจากเวอร์ชันก่อนหน้าเล็กน้อย อ่านเอกสารคู่มือ `gated.conf` ใน ใน *ข้อมูลอ้างอิงไฟล์* หรือใช้ไฟล์ตัวอย่างที่มีมาในไดเร็กทอรี `/usr/sample/tcpip` เพื่อแก้ไขไวยากรณ์

      เมื่อกำหนดค่า **BGP4+** หรือ **RIPng** ให้ใช้ **IPv6** addresses ที่ไวยากรณ์ระบุ IP แอดเดรส

      หมายเหตุ: โด ค่าดีฟอลต์ **RIPng** multicasts แพ็กเก็ตของตน

หลังจากไฟล์ `/etc/gated.conf` ถูกแก้ไขแล้ว **gated** daemon จะสามารถเริ่มทำงาน

ตัวเลขระบบ Autonomous

ถ้าคุณใช้ **EGP** หรือ **BGP** คุณควรมี *ตัวเลข ระบบ autonomous* สำหรับเกตเวย์ของคุณ

เมื่อต้องการหาตัวเลขระบบ autonomous ทางการ ให้ติดต่อ NIC ที่ อินเตอร์เน็ตแอดเดรสต่อไปนี้:

`INFO@INTERNIC.NET`

Mobile IPv6

Mobile **IPv6** ให้การสนับสนุนการเคลื่อนย้ายสำหรับ **IPv6** ซึ่ง อนุญาตให้คุณคงค่าอินเตอร์เน็ตแอดเดรสเดิมใช้ทั่วโลก และยังอนุญาต ให้แอ็พพลิเคชันที่ใช้แอดเดรสนั้นรักษาการส่งข้อมูล และการเชื่อมต่อในเลเยอร์ระดับสูง เมื่อมีการเปลี่ยนที่ตั้ง รวมทั้งอนุญาตการเคลื่อนย้ายระหว่างสื่อบันทึกที่มีคุณสมบัติเหมือนกันและ ต่างกัน

ตัวอย่างเช่น Mobile **IPv6** ช่วยอำนวยความสะดวกในการย้ายโหนดจากเซ็กเมนต์ อีเทอร์เน็ตไปยังเซลล์ LAN แบบไร้สายขณะที่ IP แอดเดรสของโหนดโมบายล์ยังคง ไม่เปลี่ยนแปลง

ใน Mobile **IPv6** แต่ละโหนดโมบายล์จะถูกระบุโดยใช้สอง IP addresses: โฮมแอดเดรส และแอดเดรส care-of โฮมแอดเดรสคือ IP address ถาวรที่ระบุโหนดโมบายล์โดยไม่ต้องคำนึงถึงตำแหน่งที่ตั้ง แอดเดรส care-of จะเปลี่ยนแปลงแต่ละครั้งที่มีการเชื่อมต่อจุดใหม่ และให้ข้อมูล เกี่ยวกับสถานการณ์ปัจจุบันของโหนดโมบายล์ เมื่อโหนดโมบายล์ไปยังเน็ตเวิร์ก ที่เคยเยี่ยมชมโหนดต้องใช้แอดเดรส care-of ซึ่งจะถูกใช้ระหว่าง เวลาที่โหนดโมบายล์อยู่ในตำแหน่งนี้ในเน็ตเวิร์กที่เคยเยี่ยมชม โดยอาจใช้เมธอด **IPv6** Neighborhood Discovery เพื่อหาค่าแอดเดรส care-of (ดูที่ “การค้นหาเครื่องใกล้เคียง/การกำหนดค่าแอดเด

รสอัตโนมัติแบบไม่เก็บค่าสถานะ" ในหน้า 136) สามารถใช้ได้ทั้งการกำหนดค่าอัตโนมัติแบบไม่มีการบันทึกสถานะ และบันทึกสถานะ แอดเดรส care-of ยังถูกกำหนดค่าเองได้ วิธีการเพื่อให้ได้แอดเดรส care-of ไม่ใช่สิ่งสำคัญ สำหรับ Mobile **IPv6**

โดยมีอย่างน้อยหนึ่งโฮมเอเจนต์ที่กำหนดค่าบนโฮมเน็ตเวิร์ก และ โหนดโมบายล์ต้องถูกกำหนดค่าเพื่อให้รู้จัก IP address ของโฮม์เอเจนต์ โหนดโมบายล์ส่งแพ็กเก็ตที่มีการอัพเดตข้อมูลการโยงไปยังโฮมต์เอเจนต์ โฮมเอเจนต์ได้รับแพ็กเก็ต และทำการเชื่อมโยงระหว่างโฮม แอดเดรสไปยังโหนดโมบายล์ และแอดเดรส care-of ที่ได้รับ โฮมเอเจนต์ ตอบกลับด้วยแพ็กเก็ตที่มีการรับทราบข้อมูลการโยง

โฮมเอเจนต์เก็บค่าแคชการโยงที่มีการเชื่อมโยงระหว่าง โฮมแอดเดรส กับแอดเดรส care-of สำหรับโหนดโมบายล์ที่ให้บริการโฮม เอเจนต์จะสกัดแพ็กเก็ตที่มีปลายทางไปยังโฮมแอดเดรส และส่งต่อ แพ็กเก็ตเหล่านี้ไปยังโหนดโมบายล์ จากนั้นโหนดโมบายล์จะส่งการอัพเดตข้อมูลการโยงไปยัง โหนดที่เกี่ยวข้องเพื่อแจ้งให้ทราบเกี่ยวกับแอดเดรส care-of และโหนดที่ เกี่ยวข้องจะสร้างรายการแคชข้อมูลการโยงเพื่อให้สามารถใช้ส่งข้อมูลโดยตรงไปยัง โหนดโมบายล์ที่แอดเดรส care-of ในอนาคต

การเคลื่อนย้ายที่สนับสนุนใน AIX จัดให้มี พื้นฐานต่อไปนี้:

ในฐานะโหนด **Home Agent**:

- ดูแลรายการในแคชข้อมูลการโยงสำหรับแต่ละโหนดโมบายล์ที่กำลัง ให้การบริการ
- สกัดแพ็กเก็ตที่มีแอดเดรสไปยังโหนดโมบายล์ที่ซึ่งขณะนี้กำลัง ให้บริการเป็นโฮมเอเจนต์ บนโฮมลิงก์ของโหนดโมบายล์นั้น ขณะนี้โหนด โมบายล์อยู่ห่างจากโฮม
- ห่อหุ้มแพ็กเก็ตที่สกัดไว้เพื่อสร้างช่องสัญญาณไปยังแอดเดรส care-of หลักสำหรับโหนดโมบายล์ที่ระบุในการโยงในแคชข้อมูลการโยงของ โฮมเอเจน์
- ส่งกลับอ็อพชันการรับทราบการโยงในการตอบกลับไปที่อ็อพชัน การอัพเดตการโยงที่ได้รับด้วยชุดบิตการตอบรับ
- ประมวลผล Duplicate Address Detection บนแอดเดรส care-of ของโหนดโมบายล์ เพื่อให้แน่ใจว่า **IPv6** addresses เป็นค่าเฉพาะ
- สนับสนุน Dynamic Home Agent Address Discovery เพื่อช่วยโหนดโมบายล์ ในการค้นหาแอดเดรสของโฮมเอเจนต์
- สนับสนุนการรับ Mobile Prefix Solicitation และการส่ง Mobile Prefix Advertisement

ในฐานะโหนด **Stationary Correspondent**:

- ประมวลผลอ็อพชันโฮมแอดเดรสที่ได้รับในแพ็กเก็ต **IPv6**
- ประมวลผลอ็อพชันการอัพเดตการโยงที่ได้รับในแพ็กเก็ตและส่งกลับอ็อพชันการตอบกลับ การโยงถ้าบิตการตอบรับ (A) ถูกตั้งค่าในการอัพเดตการโยง ที่ได้รับ
- ดูแลแคชข้อมูลการโยงของการโยงที่ได้รับในการยอมรับการอัพเดต การโยง
- ส่งแพ็กเก็ตโดยใช้ส่วนหัวการจัดเส้นทางเมื่อมีรายการแคชข้อมูลการโยง สำหรับโหนดโมบายล์ที่มีแอดเดรส care-of ปัจจุบันของโหนดโมบายล์

ในฐานะโหนด **Router** ในเน็ตเวิร์กที่เยี่ยมชมโดยโหนด โมบายล์:

- ส่งอ็อพชันช่วงเวลาการประกาศในการประกาศเราเตอร์เพื่อ ช่วยการตรวจหาการเคลื่อนย้ายโดยโหนดโมบายล์ สามารถกำหนดได้โดยใช้พารามิเตอร์ **-m** ใน **ndpd-router** daemon
- สนับสนุนการส่งการประกาศเราเตอร์ multicast ที่ไม่ต้องการในอัตราที่เร็วขึ้น ดังอธิบายใน RFC 2461 สามารถกำหนดได้โดยใช้พารามิเตอร์ **-m** และ **-D** ใน **ndpd-router** daemon

- ส่งอ็อพชัน Home Agent Information (การกำหนดค่าตามความชอบและอายุการใช้งานโฮมเอเจนต์) ในการประกาศเราเตอร์เพื่อช่วยโหนดโมบายล์เลือกโฮมเอเจนต์ของตน สามารถกำหนดได้โดยใช้พารามิเตอร์ **-H** ใน **ndpd-router** daemon

การรักษาความปลอดภัย Mobile IPv6

ข้อความการอัพเดตข้อมูลการโยง และการตอบรับการโยงที่แลกเปลี่ยน ระหว่างโหนดโมบายล์ และโฮมเอเจนต์ ต้องได้รับการป้องกันโดย IP Security โดยใช้การป้องกัน Encapsulating Security Payload (ESP) ที่มีอัลกอริทึมการพิสูจน์ตัวตน non-NULL payload

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ IP Security ดูที่ *การรักษาความปลอดภัย*

การสร้างการโยงระหว่างโหนดโมบายล์ และโหนด ที่เกี่ยวกับจะมีความปลอดภัยโดยใช้โพรซีเดอร์ Return Routability ในโพรซีเดอร์นี้ ข้อความที่ถูกแลกเปลี่ยนระหว่างโหนดโฮมเอเจนต์และโหนด โมบายล์ควรได้รับการปกป้องโดย IP Security โดยใช้ ESP เช่นกัน เนื่องจากข้อความการอัพเดต การโยง และการตอบรับการโยงถูกแลกเปลี่ยนระหว่างโหนดที่ เกี่ยวข้องและโหนดโมบายล์จะได้รับการปกป้องโดยโพรซีเดอร์ Return Routability ไม่มีข้อกำหนด IP Security สำหรับโหนดที่เกี่ยวข้อง แต่ถ้าการเกี่ยวข้อง ใช้ IP Security เพื่อจำกัดการเข้าถึง ข้อมูลความที่มีโปรโตคอล MH (135) ต้องได้รับอนุญาตให้ใช้

ช่องสัญญาณสามารถถูกกำหนดด้วยตนเอง หรือใช้ IKE acting เป็นตัวตอบกลับ (สนับสนุน เฉพาะโหมด aggressive เท่านั้น) สำหรับค่าต่ำสุด ช่องสัญญาณ IP Security ต่อไปนี้ จะถูกกำหนดบนโฮมเอเจนต์โดยใช้ส่วนหัว ESP:

- ช่องสัญญาณในโหมดการส่งข้อมูลที่มีโปรโตคอล MH (135) ระหว่าง IP address โฮมเอเจนต์และโฮมแอดเดรสของแต่ละโหนดโมบายล์ที่ถูกลงทะเบียน บนโฮมเอเจนต์นี้ได้ง่าย
- ช่องสัญญาณในโหมดการช่องสัญญาณที่มีโปรโตคอล MH (135) ระหว่าง IP address และโฮมแอดเดรสของแต่ละโหนดโมบายล์ที่ถูกลงทะเบียน บนโฮมเอเจนต์นี้ได้ง่าย

ช่องสัญญาณที่เกี่ยวข้องต้องกำหนดค่าบนโหนดโมบายล์

หมายเหตุ: ข้อความ การอัพเดตการโอย และการตอบรับการโยงถูกส่งโดยใช้ Mobility Header และต้องได้รับการป้องกันโดย IP Security โดยใช้ ESP

ในการใช้ Mobile IPv6 ก่อนหน้าใน AIX การสนับสนุนที่มีให้สำหรับโหนดโมบายล์โดยใช้แพ็กเก็ต Destination Option เพื่อส่งข้อความการอัพเดตการโยง ข้อความเหล่านี้สามารถป้องกันได้ด้วย IP Security โดยใช้ Authentication Header

สำหรับโฮมเอเจนต์ หรือโหนดที่เกี่ยวข้องเพื่อตอบรับข้อความอัพเดต การโยงโดยใช้ Destination Option ให้แก้ไขไฟล์ `/etc/rc.mobip6` และเปิดใช้งานตัวแปร **Enable_Draft13_Mobile** ก่อนที่จะ เริ่มทำงาน Mobile **IPv6** ในกรณีนี้ ถ้าคุณใช้ IP Security เพื่อป้องกัน ข้อความอัพเดตการโยง คุณต้องการกำหนดช่องสัญญาณเอง หรือ IKE ในโหมด การส่งข้อมูลบนโปรโตคอล 60 ซึ่งจะป้องกันข้อความ Binding Update และ Acknowledgement

สำหรับโฮมเอเจนต์หรือโหนดที่เกี่ยวข้องเพื่อตอบรับข้อความอัพเดตการโยง ที่ไม่ได้รับการป้องกันโดย IP Security ให้แก้ไขไฟล์ `/etc/rc.mobip6` และปิดใช้งานตัวแปร **Check_IPsec** ไม่แนะนำให้ใช้วิธีนี้เนื่องจากมีช่องโหว่ด้านความปลอดภัยที่สำคัญตลอดจน โอกาสที่จะมีผลต่อการจัดเส้นทางของแพ็กเก็ตที่กำหนดแอดเดรสไปยังโหนดโมบายล์

การกำหนดค่า Mobile IPv6

หัวข้อนี้จะให้ข้อมูลเกี่ยวกับการกำหนดค่า Mobile **IPv6** เพื่อใช้งาน Mobile **IPv6** อันดับแรกคุณต้องติดตั้งชุดไฟล์ `bos.net.mobip6.rte`

สำหรับข้อมูลเกี่ยวกับการติดตั้งชุดไฟล์ ดูที่ การติดตั้งผลิตภัณฑ์ซอฟต์แวร์ทางเลือกและการอัพเดตเซอร์วิส ใน *การติดตั้งและการย้าย*

การเริ่มต้น Mobile IPv6 เป็นโฮมเอเจนต์:

ใช้โพรซีเดอร์นี้เพื่อเริ่มต้น Mobile **IPv6** เป็นโฮมเอเจนต์

1. กำหนดช่องสัญญาณ IKE (เฟส 1 และ 2) เป็นตัวตอบกลับโดยใช้โปรโตคอล **ESP** หรือ ESP IP Security Association กำหนดเองระหว่าง IP แอดเดรสของโฮมเอเจนต์และ แต่ละโมบายล์โฮมแอดเดรสที่ตัวโต้ตอบอาจสื่อสารด้วย
2. เปิดใช้งานระบบเป็นโฮมเอเจนต์ Mobile **IPv6** และโหนดโต้ตอบ ที่บรรทัดคำสั่ง พิมพ์ `smit enable_mobip6_home_agent`
3. เลือกเวลาที่คุณต้องการเปิดใช้งาน

การเริ่มต้น Mobile IPv6 เป็นตัวโต้ตอบ:

ใช้โพรซีเดอร์นี้เพื่อเริ่มทำงาน Mobile IPv6 เป็นตัวโต้ตอบ

1. กำหนดช่องสัญญาณ IKE (เฟส 1 และ 2) เป็นตัวตอบกลับโดยใช้โปรโตคอล **ESP** หรือ ESP IP Security Association กำหนดเองระหว่าง IP แอดเดรสของโฮมเอเจนต์และ แต่ละโมบายล์โฮมแอดเดรสที่ตัวโต้ตอบอาจสื่อสารด้วย
2. เปิดใช้งานระบบเป็นโหนดโต้ตอบ Mobile **IPv6** ที่ บรรทัดคำสั่ง ให้พิมพ์ `smit enable_mobip6_correspondent`
3. เลือกเวลาที่คุณต้องการเปิดใช้งาน

การเริ่มต้น Mobile IPv6 เป็นเราเตอร์:

ใช้โพรซีเดอร์นี้เพื่อเริ่มต้น Mobile **IPv6** เป็น เราเตอร์

รันคำสั่งต่อไปนี้เพื่อตรวจจับการเคลื่อนไหวของระบบพื้นฐาน:

```
ndpd-router -m
```

การหยุด Mobile IPv6:

ใช้โพรซีเดอร์นี้เพื่อหยุด Mobile **IPv6**

1. พิมพ์ `smit disable_mobip6` ที่บรรทัดรับคำสั่ง
2. เลือกเวลาที่คุณต้องการหยุด **IPv6**
3. เลือกว่าคุณจะหยุด **ndpd-router** หรือไม่
4. เลือกเมื่อคุณต้องการปิดการส่งต่อ **IPv6**

การแก้ไขปัญหา Mobile IPv6

ใช้คำสั่ง **mobip6ctrl -b** เพื่อแก้ไขปัญหา Mobile **IPv6**

1. รับสภาวะผูกรวมโดยรันคำสั่งต่อไปนี้:

   ```
   mobip6ctrl -b
   ```

2. โปรดดู “การแก้ปัญหา TCP/IP” ในหน้า 446 สำหรับข้อมูลเกี่ยวกับการใช้ยูทิลีตีการแก้ไขปัญหา **TCP/IP**

IP address เสมือน

IP address เสมือนขจัดความเชื่อมโยงของโฮสต์บนแต่ละ เน็ตเวิร์กอินเตอร์เฟส

แพ็กเก็ตขาเข้าถูกส่งไปที่แอดเดรส VIPA ของระบบแต่การเดินทางของแพ็กเก็ตทั้งหมด ผ่านเน็ตเวิร์กอินเตอร์เฟสจริง

ก่อนหน้านี้ ถ้าอินเตอร์เฟสล้มเหลว การเชื่อมต่อไปที่อินเตอร์เฟสนั้นจะสูญ หายไป ด้วย VIPA บนระบบของคุณและโปรโตคอลการจัดเส้นทางภายในเน็ตเวิร์กจัดเตรียม การจัดเส้นทางใหม่อัตโนมัติ กู้คืนจากความล้มเหลวที่เกิดขึ้นโดยไม่ขัดขวางการเชื่อมต่อ ของผู้ใช้ที่มีอยู่ที่กำลังใช้อินเตอร์เฟสเสมือน เนื่องจากแพ็กเก็ตแบบยาว สามารถมาถึงผ่านฟิสิคัลอินเตอร์เฟสอื่น ระบบรัน VIPA มีความพร้อมใช้ สูงกว่า เนื่องจากอะแด็ปเตอร์ที่ขัดข้องจะไม่มีผลกับการเชื่อมต่อที่แอ็คทีฟอีกต่อไป เนื่องจากหลายฟิสิคัลอะแด็ปเตอร์ดำเนินการรับส่ง IP ระบบ, โหลดโดยรวมจะไม่ ถูกจำกัดอยู่ที่อะแด็ปเตอร์เดียวและ subnet ที่เกี่ยวข้อง

ฟังก์ชัน AIX VIPA มองเห็นได้กับอุปกรณ์เน็ตเวิร์ก ไม่จำเป็นต้องมีอุปกรณ์เน็ตเวิร์กพิเศษ หรือฮาร์ดแวร์อื่น เมื่อต้องการใช้ VIPA คุณจำเป็นต้องมีรายการ ต่อไปนี้:

- อินเตอร์เฟส IP ที่มีอยู่สองอินเตอร์เฟสหรือมากกว่า ที่เป็นชนิดฟิสิคัลบน subnets ต่างกันที่เชื่อมต่อกับเน็ตเวิร์กองค์กร
- โปรโตคอลการจัดเส้นทาง IP รันอยู่ภายในเน็ตเวิร์กองค์กร

การตั้งค่า VIPA

VIPA ถูกตั้งค่า เหมือนกับ IP เน็ตเวิร์กอินเตอร์เฟสอื่นใน SMIT นอกจากนี้คุณสามารถระบุกลุ่มของอินเตอร์เฟสขณะทำการตั้งค่า VIPA

เมื่อตั้งค่าวิธีนี้ สำหรับการเชื่อมต่อขาออกทั้งหมด ที่เริ่มต้นโดยโฮสต์ VIPA ผ่านอินเตอร์เฟสเหล่านี้ ซึ่งถูกกำหนดให้ใช้ VIPA, แอดเดรสเสมือนจะกลายเป็นซอร์สแอดเดรสที่กำหนดไว้ใน ส่วนหัวแพ็กเก็ต **TCP/IP** ของแพ็กเก็ตขาออก

1. สำหรับ IPv4 VIPA, พิมพ์ `smit mkinetvi` บน บรรทัดคำสั่ง สำหรับ IPv6 VIPA, พิมพ์ `smit mkinetvi6` บน บรรทัดคำสั่ง
2. ป้อนข้อมูลในฟิลด์ที่จำเป็น สำหรับข้อมูลเพิ่มเติม ดูที่ “สภาวะแวดล้อม VIPA ตัวอย่าง” ในหน้า 394 กด Enter

การเพิ่มอะแด็ปเตอร์ให้กับ VIPA

ใช้ขั้นตอนนี้เพื่อเพิ่มอะแด็ปเตอร์ให้กับ IP address เสมือน

เมื่อต้องการเพิ่มอะแด็ปเตอร์ให้กับอินเตอร์เฟส VIPA ของคุณ ให้ทำตามขั้นตอนต่อไปนี้:

1. พิมพ์ `smit chvi` บนบรรทัดคำสั่ง
2. เลือก VIPA ซึ่งคุณต้องการเพิ่มอะแด็ปเตอร์และกด Enter
3. ป้อนอะแด็ปเตอร์ที่คุณต้องการเพิ่มในฟิลด์ **Interface Name(s)**
4. พิมพ์ `ADD` ในฟิลด์ **ADD/REMOVE interface(s)** และกด Enter

การลบอะแด็ปเตอร์จาก VIPA

ใช้ขั้นตอนนี้เพื่อเอาอะแด็ปเตอร์ออกจาก IP แอดเดรสเสมือน

เมื่อต้องการเอาอะแด็ปเตอร์ออกจาก VIPA ให้ทำตามขั้นต่อเหล่านี้:

1. พิมพ์ `smit chvi` บนบรรทัดคำสั่ง

2. เลือก VIPA ซึ่งคุณต้องการเอาอะแด็ปเตอร์ออก และกด Enter
3. ป้อนอะแด็ปเตอร์ที่คุณต้องการเอาออกในฟิลด์ **Interface Name(s)**
4. พิมพ์ REMOVE ในฟิลด์ **ADD/REMOVE interface(s)** และกด Enter

สภาวะแวดล้อม VIPA ตัวอย่าง

สภาวะแวดล้อม VIPA ตัวอย่างต่อไปนี้มีการเชื่อมต่อ Ethernet สัมพันธ์กับระบบที่มี IP address เสมือนกับสองการเชื่อมต่อฟิสิคัล

ระบบมี IP แอดเดรสเสมือน, vi0, เป็น 10.68.6.1 และสองการเชื่อมต่อฟิสิคัล, en1 ที่มี IP address 10.68.1.1 และ en5, ที่มี IP address 10.68.5.1 ในตัวอย่างนี้ ทั้งสองการเชื่อมต่อ ฟิสิคัลเป็น Ethernet, แต่การใช้งานรวมกันของอินเตอร์เฟส IP เช่น token-ring หรือ FDDI, จะถูกสนับสนุนตราบที่ subnets เชื่อมต่อกับเน็ตเวิร์กองค์กรที่ใหญ่กว่าในท้ายที่สุด และเป็นที่รู้จักของเราเตอร์องค์กร

การรันคำสั่ง **lsattr -El vi0** สร้างผลลัพธ์ต่อไปนี้:

```
netaddr          10.68.6.1      N/A                                    True
state            up             Standard Ethernet Network Interface    True
netmask          255.255.255.0  Maximum IP Packet Size for This Device  True
netaddr6                        Maximum IP Packet Size for REMOTE Networks  True
alias6                          Internet Address                       True
prefixlen                       Current Interface Status               True
alias4                          TRAILER Link-Level Encapsulation       True
interface_names  en1,en5        Interfaces using the Virtual Address   True
```

การรันคำสั่ง **ifconfig vi0** สร้างผลลัพธ์ ต่อไปนี้:

```
vi0: flags=84000041<UP,RUNNING,64BIT>
       inet 10.68.6.1 netmask 0xffffff00
       iflist : en1 en5
```

การรันคำสั่ง **netstat -rn** สร้างผลลัพธ์ต่อไปนี้:

```
Routing tables
Destination      Gateway           Flags   Refs     Use  If   PMTU Exp Groups

Route Tree for Protocol Family 2 (Internet):
default          10.68.1.2      UG        3      1055   en1     -   -
10.68.1/24       10.68.1.1      U         0      665    en1     -   -
10.68.5/24       10.68.5.1      U         0      1216   en5     -   -
127/8            127.0.0.1      U         4      236    lo0     -   -
10.68.6.1        127.0.0.1      UH        0      0      lo0     -   -
```

แพ็กเก็ต ขาออกที่ไม่มีซุดซอร์สแอดเดรสและที่ถูกส่ง ผ่านอินเตอร์เฟส en1 และ en5 จะ มีซุดซอร์สแอดเดรสเซ็ตเป็นแอดเดรสเสมือน (10.68.6.1) แพ็กเก็ตขาเข้าถูกจัดเส้นทางไปที่แอดเดรส VIPA (10.68.6.1) ที่แจ้งไว้บนเน็ตเวิร์ก เนื่องจาก vi0 คือเสมือน (คือไม่สัมพันธ์กับอุปกรณ์ใด) จึงไม่ควรมีรายการสำหรับ ค่านี้ในตารางการจัดเส้นทางทั้งระบบ ที่แสดงโดยใช้คำสั่ง **netstat -rn** ซึ่งหมายความว่าไม่มีเส้นทางอินเตอร์เฟสถูกเพิ่มเมื่อ อินเตอร์เฟสถูกตั้งค่าใน SMIT

ถ้าหนึ่งในฟิสิคัลอินเตอร์เฟส, การเชื่อมต่อเน็ตเวิร์ก หรือเน็ตเวิร์กพาธล้มเหลว เน็ตเวิร์กโปรโตคอลจัดเส้นทางไปที่ฟิสิคัลอินเตอร์เฟสอื่นบนระบบเดียวกัน ถ้าระบบรีโมต telnets ไปที่แอดเดรส `vi0` , แพ็กเก็ต ไปที่ `vi0` สามารถมาถึงโดยใช้ `en1` หรือ `en5` ถ้า `en1` ไม่ทำงาน ตัวอย่าง, แพ็กเก็ตยังคงสามารถมาถึง ที่ `en5` หมายเหตุไว้ว่าโปรโตคอลการจัดเส้นทางอาจใช้เวลา ในการกระจายเส้นทาง

เมื่อใช้ VIPA, ระบบสุดท้าย และการแทรกแทรงเราเตอร์ ต้องสามารถจัดเส้นทางปลายทางของแพ็กเก็ต สำหรับ VIPA (`vi0`) ไปที่หนึ่งในฟิสิคัลอินเตอร์เฟส (`en1` หรือ `en5`)

เปรียบเทียบ VIPA กับ IP แบบย่อ

แนวคิดของ VIPA คล้ายกับ IP แบบย่อ ยกเว้นแอดเดรส ไม่เชื่อมโยงกับอินเตอร์เฟสของฮาร์ดแวร์

VIPA มีข้อดีหลายประการที่ IP แบบย่อไม่มี:

- VIPA มีอุปกรณ์เสมือนที่สามารถเลื่อนขึ้นลงได้อย่างอิสระ โดยไม่มีผลกระทบใดๆ ต่อฟิสิคัลอินเตอร์เฟส
- แอดเดรสของ VIPA สามารถเปลี่ยนแปลงได้ได้ในขณะที่ IP แบบย่อทำได้เพียงเพิ่มหรือลบเท่านั้น

การเข้าถึงโดยใช้ IP address ของอะแด็ปเตอร์จริง

แต่ละอินเตอร์เฟสยังสามารถเข้าถึงระบบอื่นได้หลัง การนำ VIPA มาใช้งาน อย่างไรก็ตาม การใช้ IP addresses จริงสำหรับ ping และ telnet เซสชันเลี่ยงข้อดีของ VIPA ในการสื่อสารอิสระของ ฟิสิคัลอะแด็ปเตอร์ VIPA ซ่อนความล้มเหลวของฟิสิคัลอะแด็ปเตอร์จาก ไคลเอ็นต์ การใช้แอดเดรสจริงมีความเชื่อมโยงกับฟิสิคัล อะแด็ปเตอร์

ถ้าระบบรีโมตติดต่อกับระบบ VIPA โดยใช้ VIPA แอดเดรสหรือถ้าแอ็พพลิเคชันบนระบบ VIPA เริ่มการสื่อสาร กับระบบอื่น VIPA แอดเดรสจะถูกใช้เป็นซอร์ส IP address ในแพ็กเก็ต อย่างไรก็ตาม ถ้าระบบรีโมตเริ่มต้นเซสชันโดยใช้ IP address ของอินเตอร์เฟสจริงที่ IP address จริงจะเป็นซอร์ส IP address ในแพ็กเก็ตการตอบกลับ มีข้อยกเว้นหนึ่งข้อ สำหรับแอ็พพลิเคชัน ที่เชื่อมโยงกับ IP interface เฉพาะ แพ็กเก็ตขาออกจะถือ ซอร์สแอดเดรสของอินเตอร์เฟสซึ่งแพ็กเก็ตถูกเชื่อมโยง

VIPA และโปรโตคอลการจัดเส้นทาง

gated daemon ถูกแก้ไขสำหรับ VIPA ดังนั้นมันไม่ควรเพิ่ม เส้นทางอินเตอร์เฟส หรือส่งการแจ้งผ่านอินเตอร์เฟสเสมือน

โปรโตคอล **OSPF** สนับสนุนโดย gated จะแจ้ง อินเตอร์เฟสเสมือนให้กับเราเตอร์ใกล้เคียง โฮสต์อื่นบนเครือข่าย จะสามารถคุยกับโฮสต์ VIPA ผ่านทางเราเตอร์ hop แรก

แอดเดรส VIPA หลายแอดเดรส

อินเตอร์เฟสเสมือนหลายอินเตอร์เฟสสามารถถูกตั้งค่าได้ อินเตอร์เฟส VIPA หลาย อินเตอร์เฟสมีประโยชน์ เช่น ถ้าเน็ตเวิร์กเราเตอร์สามารถให้การทำงานพิเศษ กับแพ็กเก็ตที่ส่งไปที่หรือมาจาก แอดเดรส VIPA

หรือคุณอาจใช้หลายอินเตอร์เฟส VIPA ถ้าอินเตอร์เฟสถูกเชื่อมโยง แอ็พพลิเคชันกับอินเตอร์เฟส VIPA ตัวอย่าง เมื่อต้องการรันหลายเว็บเซิร์ฟเวอร์ สำหรับหลายบริษัทบนเครื่องเดียว คุณสามารถตั้งค่า ต่อไปนี้:

- vi0 200.1.1.1 www.companyA.com
- vi1 200.1.1.2 www.companyB.com
- vi2 200.1.1.3 www.companyC.com

EtherChannel และ IEEE 802.3ad Link Aggregation

EtherChannel และ IEEE 802.3ad Link Aggregation เป็นเทคโนโลยีการรวบรวมพอร์ตเน็ตเวิร์กที่ยอมให้หลาย Ethernet อะแด็ปเตอร์ถูกรวมเป็นอุปกรณ์ Ethernet แฝงเดียว

ตัวอย่างเช่น `ent0` และ `ent1` สามารถ มีการรวมเข้าในอะแด็ปเตอร์ EtherChannel ที่เรียกว่า `en3` จากนั้น อินเตอร์เฟส `en3` จะมีการกำหนดคอนฟิกด้วย IP แอดเดรส ระบบจะพิจารณาว่าอะแด็ปเตอร์ที่ถูกรวมเหล่านี้เป็นอะแด็ปเตอร์เดียว ดังนั้น มีการกำหนดคอนฟิก IP บนนั้น เช่นเดียวกับบนอะแด็ปเตอร์อีเทอร์เน็ตใดๆ นอกจากนี้ อะแด็ปเตอร์ทั้งหมด ใน EtherChannel หรือ Link Aggregation ได้กำหนดแอดเดรสของฮาร์ดแวร์ (Mac) ที่เหมือนกันไว้ ดังนั้น จึงใช้เป็นระบบรีโมตหากมีหนึ่งอะแด็ปเตอร์ ทั้ง EtherChannel และ การรวมลิงก์ IEEE 802.3ad ต้องการการสนับสนุนในสวิตช์ เพื่อให้ สองเทคโนโลยีเหล่านี้รับรู้ว่าพอร์ตของสวิตช์ใดต้องถูกจัดการ เป็นพอร์ตเดียว

หมายเหตุ: ไดร์เวอร์ EtherChannel กำหนด media access control (MAC) address ที่ไม่ถูกต้อง `02:00:00:00:00:00` ให้กับพอร์ต Host Ethernet Adapter (HEA) ของแชนเนลที่ไม่แอ็คทีฟของ คอนฟิกูเรชัน EtherChannel MAC address ที่ไม่ถูกต้องนี้ถูกกำหนดไว้เมื่อ EtherChannel ถูกสร้างขึ้นหรือเมื่อพอร์ต HEA ถูกเพิ่มไปยังแชนเนลที่ไม่แอ็คทีฟ ณ รันไทม์ ในระหว่างที่เกิดความล้มเหลวของ EtherChannel หรือการกู้คืน MAC address ที่ไม่ถูกต้องจะถูกสลับกับ MAC address ที่ถูกต้อง และ MAC address ที่ถูกต้องถูกสลับกับ MAC address ที่ไม่ถูกต้อง ณ รันไทม์

ประโยชน์หลักของ EtherChannel และ IEEE 802.3ad Link Aggregation คือมันมีเน็ตเวิร์กแบนด์วิดธ์ของอะแด็ปเตอร์ทั้งหมดของมันในการมีอยู่ของเน็ตเวิร์กเดียว ถ้าอะแด็ปเตอร์หนึ่งล้มเหลว เน็ตเวิร์กทราฟิกจะถูกส่งบนอะแด็ปเตอร์ต่อไปที่ว่างโดยอัตโนมัติโดยไม่มีการขัดจังหวะการเชื่อมต่อของผู้ใช้ที่มีอยู่ อะแด็ปเตอร์จะกลับไปใช้เซอร์วิสบน EtherChannel หรือ Link Aggregation โดยอัตโนมัติเมื่อมันถูกแก้ไขแล้ว

มีความแตกต่างบางอย่างระหว่าง EtherChannel และ IEEE 802.3ad Link Aggregation พิจารณาความแตกต่างที่แสดงรายการใน ตารางที่ 81เพื่อกำหนดว่าเทคโนโลยีใด เหมาะสมกับความต้องการของคุณมากที่สุด

ตารางที่ 81. ความแตกต่างระหว่าง EtherChannel และ IEEE 802.3ad Link Aggregation

EtherChannel	IEEE 802.3ad Link Aggregation
ต้องการการตั้งค่าสวิตช์	ต้องการคอนฟิกูเรชันสวิตช์ สำหรับการแลกเปลี่ยน Link Aggregation Control Protocol Data Unit (LACPDU)
Heartbeats ไม่มีการแลกเปลี่ยนระหว่าง พอร์ตของสวิตช์และพอร์ตของระบบที่ติดกัน	Heartbeats (LACPDU) มีการแลกเปลี่ยน ที่ช่วงเวลาซึ่งกำหนดโดยมาตรฐาน IEEE 802.3ad Heartbeats ให้การป้องกันพิเศษในกรณีที่เกิดความล้มเหลว

ฟังก์ชัน Dynamic Adapter Membership มีอยู่ในระบบปฏิบัติการ AIX คุณสามารถใช้ ฟังก์ชันนี้เพื่อเพิ่มหรือลบอะแด็ปเตอร์จาก EtherChannel โดยไม่ต้องขัดจังหวะการเชื่อมต่อผู้ใช้

หลักการที่เกี่ยวข้อง:

"Dynamic Adapter Membership" ในหน้า 407
ก่อนหน้า AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03 เพื่อที่จะเพิ่มหรือลบอะแด็ปเตอร์จาก EtherChannel อินเตอร์เฟสของมันต้องถูกถอดออกก่อน ซึ่งจะขัดจังหวะทราฟิกทั้งหมดของผู้ใช้ เพื่อข้ามข้อจำกัดนี้ Dynamic Adapter Membership (DAM) ได้ถูกเพิ่มใน AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03

"EtherChannel" ในหน้า 397
อะแด็ปเตอร์ที่เป็นของ EtherChannel ต้องถูกเชื่อมกับสวิตช์ EtherChannel-enabled เดียวกัน ถ้า อะแด็ปเตอร์มีการเชื่อมต่อกับสวิตช์อื่น ต้องสแต็กสวิตช์เหล่านั้น และทำหน้าที่เป็นสวิตช์เดียว

“การตั้งค่า IEEE 802.3ad Link Aggregation” ในหน้า 410
IEEE 802.3ad เป็นวิธีมาตรฐานของการรวมลิงก์ ตามแนวคิด มันทำงานในแบบเดียวกับ EtherChannel ที่หลาย Ethernet อะแด็ปเตอร์ถูกรวมเข้ากับอะแด็ปเตอร์เสมือนเดียว เพื่อให้ได้แบนด์วิดธ์ที่กว้างและป้องกันความล้มเหลว

“สถานการณ์จำลองความสามารถในการทำงานข้ามระบบ” ในหน้า 415
พิจารณาสถานการณ์จำลองความสามารถในการทำงานข้ามระบบต่อไปนี้เมื่อตั้งค่า EtherChannel หรือ E 802.3ad Link Aggregation ของคุณ

EtherChannel

อะแด็ปเตอร์ที่เป็นของ EtherChannel ต้องถูกเชื่อมกับสวิตช์ EtherChannel-enabled เดียวกัน ถ้า อะแด็ปเตอร์มีการเชื่อมต่อกับสวิตช์อื่น ต้องสแต็กสวิตช์เหล่านั้น และทำหน้าที่เป็นสวิตช์เดียว

คุณต้องตั้งค่าสวิตช์นี้แบบแมนวลที่จัดการพอร์ตที่เป็นของ EtherChannel เป็นลิงก์ที่ถูกรวม เอกสารของสวิตช์ของคุณอาจอ้างถึงความสามารถนี้เป็น *link aggregation* หรือ *trunking*

เพื่อให้ EtherChannel ทำงานอย่างถูกต้อง กลไกการโพลลิงก์ ที่จะตรวจสอบสถานะของลิงก์เป็นระยะๆ ต้องมีการเปิดใช้งานบนแต่ละอะแด็ปเตอร์ก่อนจะมีการสร้าง EtherChannel ทราฟิกจะถูกกระจายข้ามอะแด็ปเตอร์ในวิธีแบบมาตรฐาน (ที่อะแด็ปเตอร์ที่แพ็กเก็ตจะถูกส่งถูกเลือกโดยขึ้นอยู่กับอัลกอริทึม) หรือบนพื้นฐานแบบround-robin basis (ที่แพ็กเก็ตจะถูกส่งอย่างเท่ากันบนอะแด็ปเตอร์ทั้งหมด) ทราฟิกที่เข้ามาจะถูกกระจายตามการตั้งค่าของสวิตช์และไม่ถูกควบคุมโดยโหมดการทำงานของ EtherChannel

คุณสามารถกำหนดคอนฟิกหลาย EtherChannels สำหรับแต่ละระบบ ถ้าลิงก์ทั้งหมดในหนึ่ง EtherChanel มีการต่อพ่วงกับสวิตช์เดียวและ ถ้าสวิตช์ไม่ได้เสียบปลั๊กหรือล้มเหลว ทั้ง EtherChannel จะสูญหายไป เพื่อแก้ไขปัญหานี้ มีอ็อพชันแบ็คอัพที่จะรักษาเซอร์วิสให้ แอ็คทีฟเมื่อ EtherChannel หลักล้มเหลว แบ็คอัพและอะแด็ปเตอร์ EtherChannel ต้องต่อพ่วงกับสวิตช์เครือข่ายที่ต่างกัน ซึ่งต้องมีการ เชื่อมต่อสำหรับเซ็ตอัพนี้เพื่อให้ทำงานได้อย่างถูกต้อง ถ้าอะแด็ปเตอร์ทั้งหมด ใน EtherChannel ล้มเหลว ระบบจะใช้แบ็คอัพอะแด็ปเตอร์เพื่อส่งและรับ ทราฟฟิกทั้งหมด เมื่อลิงก์ใดๆใน EtherChannel ถูกกู้คืน เวอร์วิสจะถูกย้ายกลับไปยัง EtherChannel

ตัวอย่างเช่น `ent0` และ `ent1` สามารถมีการ กำหนดคอนฟิกเป็นอะแด็ปเตอร์ EtherChannel หลัก และ `ent2` เป็น แบ็คอัพอะแด็ปเตอร์ เพื่อสร้าง EtherChannel ที่เรียกว่า `en3` ตามแนวคิด `ent0` และ `ent1` มีการเชื่อมต่อ กับสวิตช์ที่เปิดใช้งาน EtherChannel ตัวเดียวกัน และ `ent2` มีการ เชื่อมต่อกับสวิตช์อื่น ในตัวอย่างนี้ ทราฟฟิกทั้งหมดที่ส่ง ผ่าน `en3` (อินเตอร์เฟสของ EtherChannel) ถูกส่ง ผ่าน `ent0` หรือ `ent1` โดยดีฟอลต์ (ขึ้นอยู่กับ แบบแผนการกระจายแพ็กเก็ตของ EtherChannel) โดยที่ `ent2` ไม่ทำงาน ถ้าเมื่อเวลาใดที่ทั้ง `ent0` และ `ent1` ล้มเหลว ทราฟฟิกทั้งหมดจะถูกส่งผ่านแบ็คอัพอะแด็ปเตอร์ `ent2` เมื่อ `ent0` หรือ `ent1` ถูกกู้คืน มันจะถูกใช้สำหรับทราฟิกทั้งหมดอีกครั้ง

แบ็คอัพอินเตอร์เฟสเครือข่าย ซึ่งเป็นโหมดของการดำเนินการที่มีสำหรับ EtherChannel จะป้องกันความล้มเหลวของเครือข่ายอีเทอร์เน็ตจุดเดียว ไม่ต้องการ ฮาร์ดแวร์พิเศษเพื่อใช้แบ็คอัพอินเตอร์เฟสเครือข่าย แต่แบ็คอัพ อะแด็ปเตอร์ต้องมีการเชื่อมต่อกับสวิตช์แยกต่างหากเพื่อให้ได้ความน่าเชื่อถือสูงสุด ในโหมด Network Interface Backup จะมีเพียงอะแด็ปเตอร์เดียวที่ถูกใช้สำหรับเน็ตเวิร์กทราฟิกในเวลาเดียวกัน EtherChannel จะทดสอบอะแด็ปเตอร์ที่แอ็คทีฟอยู่ ในปัจจุบัน และพาธเครือข่ายไปยังโหนดที่ผู้ใช้ระบุซึ่งเป็นทางเลือก เมื่อตรวจพบความล้มเหลว อะแด็ปเตอร์ถัดไปจะถูกใช้สำหรับทราฟิกทั้งหมด Network Interface Backup จัดเตรียมการตรวจจับและ failover โดยไม่มีการขัดจังหวะการเชื่อมต่อของผู้ใช้โดยดั้งเดิม แบ็คอัพอินเตอร์เฟสเครือข่าย มีการใช้เป็นโหมดในเมนู system management interface tool (SMIT) ของ EtherChannel แบ็คอัพอะแด็ปเตอร์ให้ฟังก์ชันที่ เท่าเทียมกัน ดังนั้น โหมดจึงถูกตัดออกจากเมนู SMIT เมื่อต้องการกำหนดคอนฟิก แบ็คอัพอินเตอร์เฟสเครือข่าย โปรดดู “การตั้งค่า Network Interface Backup” ในหน้า 402

ข้อพิจารณาเกี่ยวกับการตั้งค่า EtherChannel

ศึกษาลิสต์ของคำแนะนำนี้ก่อนที่จะตั้งค่า EtherChannel

- คุณสามารถมีได้มากถึงแปด Ethernet อะแด็ปเตอร์หลัก และมีแบ็กอัพ Ethernet อะแด็ปเตอร์เพียงอะแด็ปเตอร์เดียวต่อ EtherChannel
- คุณสามารถตั้งค่าหลาย EtherChannels บนระบบเดียว แต่ แต่ละ EtherChannel จะประกอบด้วย Ethernet อินเตอร์เฟสเพิ่มเติม อ็อพชัน **ifsize** ของคำสั่ง **no** อาจต้องถูกเพิ่มค่าเพื่อรวมไม่เฉพาะ Ethernet อินเตอร์เฟสสำหรับแต่ละอะแด็ปเตอร์ แต่ยังรวม EtherChannels ที่ถูกตั้งค่า ใน AIX 5.2 และก่อนหน้านี้ ค่าดีฟอลต์ของ **ifsize** คือแปด ขนาดดีฟอลต์คือ 256
- คุณสามารถใช้ Ethernet อะแด็ปเตอร์ที่ได้รับการสนับสนุนใดๆใน EtherChannel (ดูที่ "อะแด็ปเตอร์ที่ได้รับการสนับสนุน" ในหน้า 415) อย่างไรก็ตาม Ethernet อะแด็ปเตอร์ต้องถูกเชื่อมต่อกับสวิตช์ที่สนับสนุน EtherChannel ดูที่เอกสารที่มากับสวิตช์ของคุณเพื่อดูว่ามันสนับสนุน EtherChannel หรือไม่ (เอกสารของสวิตช์ของคุณอาจอ้างถึงความสามารถนี้เป็น link aggregation หรือ trunking)
- อะแด็ปเตอร์ทั้งหมดใน EtherChannel ควรถูกตั้งค่าที่ความเร็วเดียวกัน (ตัวอย่างเช่น 100 Mbps) และควรเป็น full duplex
- อะแด็ปเตอร์ที่ใช้ใน EtherChannel จะไม่สามารถถูกเข้าถึงโดยระบบหลังจากที่ EtherChannel ถูกตั้งค่า เพื่อแก้ไขแอ็ตทริบิวต์ใดๆของมัน เช่น ความเร็วของสื่อ ขนาดของคิวการส่งและรับ และอื่นๆ คุณต้องทำดังกล่าวก่อนที่จะรวมมันใน EtherChannel
- อะแด็ปเตอร์ที่คุณวางแผนที่จะใช้สำหรับ EtherChannel ของคุณต้องไม่ถูกตั้งค่า IP แอดเดรสก่อนที่คุณจะเริ่มโพรซีเดอร์นี้ เมื่อตั้งค่า EtherChannel กับอะแด็ปเตอร์ที่ถูกตั้งค่า IP แอดเดรสก่อนหน้านี้ ต้องแน่ใจว่าอินเตอร์เฟสของมันอยู่ในสถานะ `detach` อะแด็ปเตอร์ที่จะถูกเพิ่มเข้ากับ EtherChannel ไม่สามารถมีอินเตอร์เฟสที่ถูกตั้งค่าที่มีสถานะเป็น `up` ใน Object Data Manager (ODM) ซึ่งจะเกิดขึ้นถ้า IP แอดเดรสของมันถูกตั้งค่าโดยใช้ SMIT นี่อาจจะทำให้เกิดปัญหาในการทำให้ EtherChannel ทำงานเมื่อเครื่องถูกรีบูต เนื่องจากอินเตอร์เฟสนั้นถูกตั้งค่าก่อน EtherChannel พร้อมกับข้อมูลถูกพบใน ODM ดังนั้น เมื่อ EtherChannel ถูกตั้งค่า มันจะพบว่าหนึ่งในอะแด็ปเตอร์ของมันถูกใช้อยู่ เพื่อเปลี่ยนสิ่งนี้ ก่อนที่จะสร้าง EtherChannel พิมพ์ `smitty chinet` เลือกแต่ละอินเตอร์เฟสของอะแด็ปเตอร์ที่จะถูกรวมใน EtherChannel และเปลี่ยนค่า **สถานะ** ของ มันเป็น `detach` นี่จะทำให้แน่ใจว่าเมื่อเครื่องถูกรีบูต EtherChannel สามารถถูกตั้งค่าโดยไม่มีข้อผิดพลาด สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ ODM ดูที่ Object Data Manager (ODM) ใน *หลักการเขียนโปรแกรมทั่วไป: การบันทึกและการดีบักโปรแกรม*
- ถ้าคุณจะใช้ 10/100 Ethernet อะแด็ปเตอร์ใน EtherChannel สำหรับ AIX เวอร์ชันก่อน AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03 คุณอาจต้องเปิดใช้งาน link polling บนอะแด็ปเตอร์เหล่านั้นก่อนที่จะเพิ่มมันเข้ากับ EtherChannel พิมพ์ `smitty chgenet` ที่บรรทัดรับคำสั่ง เปลี่ยนค่า **Enable Link Polling** เป็น `yes` และกด Enter

  หมายเหตุ: ใน AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03 และหลังจากนั้น ไม่มีความจำเป็นต้องเปิดใช้งานกลไก link polling link poller จะถูกสตาร์ทโดยอัตโนมัติ
- ถ้าคุณวางแผนจะใช้เฟรม jumbo คุณอาจต้องเปิดใช้งานคุณลักษณะนี้ในทุกอะแด็ปเตอร์ก่อนการสร้าง EtherChannel และในตัว EtherChannel เอง พิมพ์ `smitty chgenet` ที่บรรทัดรับคำสั่ง เปลี่ยนค่า **Enable Jumbo Frames** เป็น `yes` และกด Enter ทำสิ่งนี้สำหรับทุกอะแด็ปเตอร์ที่คุณต้องการเปิดใช้งาน Jumbo Frames คุณจะเปิดใช้งาน jumbo frames ใน EtherChannel เองภายหลัง

  หมายเหตุ: การเปิดใช้งานเฟรม frames ในทุกอะแด็ปเตอร์ที่สำคัญไม่มีความจำเป็น เนื่องจากมีการเปิดใช้งานในตัว EtherChannel เอง คุณลักษณะจะถูกเปิดใช้งานโดยอัตโนมัติ ถ้าคุณตั้งแอ็ตทริบิวต์ **Enable Jumbo Frames** เป็น `yes`
- ระดับ AIX 5.3และ AIX 6.1 จะสนับสนุนการตั้งค่าต่อไปนี้โดยขึ้นอยู่กับ Host Ethernet Adapters (HEA)

- การรวมลิงก์ระหว่างพอร์ตเฉพาะของ HEA และ PCI/PCI-E อะแด็ปเตอร์ได้รับการสนับสนุน นี่ได้รับการสนับสนุนสำหรับทั้งการรวมแบบแมนวลและการรวมแบบ LACP
- สำหรับการตั้งค่า EtherChannel ที่เกี่ยวข้องกับพอร์ต non-dedicated HEA EtherChannel พร้อมด้วยแบ็กอัพอะแด็ปเตอร์ เช่น PCI/PCI-E หรือ ethernet เสมือนจะได้รับการสนับสนุน

  หมายเหตุ: สำหรับพอร์ต non-dedicated HEA ในการตั้งค่า EtherChannel ข้อจำกัดที่มีอยู่ที่เกี่ยวกับการรวมลิงก์ยังถูกใช้

- AIX เวอร์ชัน 6.1 ที่มีระดับเทคโนโลยี 6100-06 และใหม่กว่าสนับสนุน EtherChannel บนสวิตช์ที่ทำสแต็กไว้

การตั้งค่า EtherChannel

ใช้โพรซีเดอร์นี้เพื่อลบ EtherChannel

1. พิมพ์ `smitty etherchannel` บนบรรทัดรับคำสั่ง
2. เลือก **Add an EtherChannel / Link Aggregation** จากลิสต์และกด Enter
3. เลือก Ethernet อะแด็ปเตอร์ลำดับแรกที่คุณต้องการบน EtherChannel ของคุณและกด Enter ถ้าคุณวางแผนที่จะใช้ EtherChannel แบ็กอัพ ห้ามเลือกอะแด็ปเตอร์ที่คุณวางแผนที่จะใช้สำหรับแบ็กอัพในเวลานี้

   หมายเหตุ: **Available Network Adapters** จะแสดง Ethernet อะแด็ปเตอร์ทั้งหมด ถ้าคุณเลือก Ethernet อะแด็ปเตอร์ที่ถูกใช้อยู่ (มีการกำหนดอินเตอร์เฟส) คุณจะได้รับข้อความแสดงข้อผิดพลาด คุณต้องถอดมันออกก่อน ถ้าคุณต้องการใช้มัน
4. ใส่ข้อมูลในฟิลด์ตามแนวทางต่อไปนี้ :
   - **Parent Adapter:** ให้ข้อมูลเกี่ยวกับอุปกรณ์พาเรนต์ของ EtherChannel (ตัวอย่างเช่น เมื่อ EtherChannel เป็นของ Shared Ethernet Adapter) ฟิลด์นี้จะแสดงค่าของ `NONE` ถ้า EtherChannel ไม่ได้อยู่ในอะแด็ปเตอร์อื่น (โดยดีฟอลต์) ถ้า EtherChannel อยู่ในอะแด็ปเตอร์อื่น ฟิลด์นี้จะแสดงชื่อของพาเรนต์อะแด็ปเตอร์ (ตัวอย่างเช่น `ent6`) ฟิลด์นี้จะให้ข้อมูลเท่านั้นและไม่สามารถแก้ไขได้ อ็อพชัน parent adapter จะมีใน AIX 5.3 และหลังจากนั้น
   - **EtherChannel / Link Aggregation Adapters:** คุณควรเห็นอะแด็ปเตอร์หลักทั้งหมดที่คุณใช้ใน EtherChannel ของคุณ คุณเลือกอะแด็ปเตอร์เหล่านี้ในขั้นตอนก่อนหน้านี้
   - **Enable Alternate Address:** ฟิลด์นี้เป็นอ็อพชัน ตั้งเป็น `yes` จะให้คุณสามารถระบุ MAC แอดเดรสที่คุณต้องการใช้ EtherChannel ถ้าคุณตั้งอ็อพชันนี้เป็น `no` EtherChannel จะใช้ MAC แอดเดรสของอะแด็ปเตอร์แรก
   - **Alternate Address:** ถ้าคุณตั้ง **Enable Alternate Address** เป็น `yes` ระบุ MAC แอดเดรสที่คุณต้องการใช้ที่นี่ แอดเดรสที่คุณระบุต้องเริ่มต้นด้วย `0x` และเป็นแอดเดรสเลขฐานสิบหก 12-หลัก (ตัวอย่างเช่น `0x001122334455`)
   - **Enable Gigabit Ethernet Jumbo Frames:** ฟิลด์นี้เป็นอ็อพชัน เพื่อที่จะใช้อ็อพชันนี้ การสวิตช์ของคุณต้องสนับสนุนเฟรมขนาดใหญ่ นี่จะใช้ได้กับอินเตอร์เฟส Standard Ethernet (en) เท่านั้น ไม่ใช่กับอินเตอร์เฟส IEEE 802.3 (et) ตั้งค่านี้เป็น `yes` ถ้าคุณต้องการใช้มัน
   - **Mode:** คุณสามารถเลือกจากโหมดต่อไปนี้ :
     - **standard:** ในโหมดนี้ EtherChannel จะใช้อัลกอริทึมเพื่อเลือกว่าอะแด็ปเตอร์ใดที่มันจะใช้เพื่อส่งแพ็กเก็ต อัลกอริทึมประกอบด้วยการนำค่าของข้อมูลมา หารด้วยจำนวนของอะแด็ปเตอร์ใน EtherChannel และใช้เศษที่เหลือ (การใช้การหารแบบหาเศษ) เพื่อระบุลิงก์ขาออก ค่าของ Hash Mode จะระบุว่าค่าของข้อมูลใดที่จะป้อนให้กับอัลกอริทึมนี้ (ดูที่แอ็ตทริบิวต์ Hash Mode สำหรับคำอธิบายของโหมด hash ที่แตกต่างกัน) ตัวอย่างเช่น ถ้า Hash Mode เป็น `standard` มันจะใช้ IP แอดเดรสเป้าหมายของแพ็กเก็ต ถ้านี่คือ 10.10.10.11 และมี 2 อะแด็ปเตอร์

ใน EtherChannel (1 / 2) = 0 โดยมีเศษ 1 ดังนั้นอะแด็ปเตอร์ที่สองจะถูกใช้ (อะแด็ปเตอร์จะมีหมายเลขเริ่มต้นจาก 0) อะแด็ปเตอร์จะถูกกำหนดหมายเลขตามลำดับที่มันถูกลิสต์ในเมนู SMIT นี่เป็นโหมดการทำงานดีฟอลต์

- **round_robin**: ในโหมดนี้ EtherChannel จะหมุนผ่านอะแด็ปเตอร์ และให้แต่ละอะแด็ปเตอร์หนึ่งแพ็กเก็ตก่อนที่จะทำซ้ำ แพ็กเก็ตอาจถูกส่งออกในลำดับที่แตกต่างกันเล็กน้อยกว่าที่มันถูกให้กับ EtherChannel แต่มันให้การใช้แบนด์วิดธ์ที่ดีที่สุด มันเป็นการรวมที่ไม่ถูกต้องถ้าเลือกโหมดนี้กับ Hash Mode แทนที่จะเป็น `default` ถ้าคุณเลือกโหมด round-robin ปล่อยให้ค่าของโหมด Hash Mode เป็น `default`
- **netif_backup**: เพื่อเปิดใช้งานโหมด Network Interface Backup คุณสามารถกำหนดคอนฟิกหลายอะแด็ปเตอร์ใน EtherChannel หลัก และแบ็คอัพอะแด็ปเตอร์ สำหรับข้อมูลเพิ่มเติม ดูที่ "การกำหนดคอนฟิก Network Interface Backup" ในหน้า 403
- **8023ad**: อ็อพชันนี้เปิดใช้งารการใช้ IEEE 802.3ad Link Aggregation Control Protocol (LACP) สำหรับการรวมลิงก์โดยอัตโนมัติ สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับคุณลักษณะนี้ดูที่ "การตั้งค่า IEEE 802.3ad Link Aggregation" ในหน้า 410

- **IEEE 802.3ad Interval**: คุณสามารถเลือกจากค่าต่อไปนี้ :
  - **long**: เป็นค่าดีฟอลต์ของช่วงเวลา เมื่อถูกเลือก EtherChannel จะร้องขอแพ็กเก็ต LACP จากคู่ของมันที่ค่าของช่วงเวลาที่ยาวที่ถูกระบุโดยโปรโตคอล
  - **short**: เมื่อถูกเลือก EtherChannel จะร้องขอแพ็กเก็ต LACP จากคู่ของมันที่ค่าของช่วงเวลาที่สั้นที่ถูกระบุโดยโปรโตคอล

    หมายเหตุ: ค่าช่วงเวลาจะถูกใช้เฉพาะเมื่อ EtherChannel ทำงานในโหมด IEEE 802.3ad ไม่เช่นนั้น ค่านี้จะถูกข้ามไป

    หมายเหตุ: AIX ยอมรับทั้งคำร้องขอช่วงเวลาสั้นและยาว จากคู่
- **Hash Mode**: เลือกจากโหมด hash ต่อไปนี้ ซึ่งจะกำหนดว่าค่าของข้อมูลที่จะถูกใช้โดยอัลกอริทึมที่กำหนดอะแด็ปเตอร์ขาออก :
  - **default**: IP แอดเดรสปลายทางของแพ็กเก็ตที่จะถูกใช้เพื่อกำหนดอะแด็ปเตอร์ขาออก สำหรับทราฟิกที่ไม่ใช่ IP (เช่น ARP) ไบต์สุดท้ายของ MAC แอดเดรสปลายทางถูกใช้เพื่อทำการคำนวณ โหมดนี้จะรับประกันว่าแพ็กเก็ตจะถูกส่งออกไปบน EtherChannel ตามลำดับที่มันถูกได้รับมา แต่แบนด์วิดธ์อาจจะไม่ถูกใช้อย่างเต็มที่
  - **src_port**: UDP ต้นทางหรือค่าของพอร์ต TCP ของแพ็กเก็ตที่จะถูกใช้เพื่อกำหนดอะแด็ปเตอร์ขาออก ถ้าแพ็กเก็ตไม่ใช่ UDP หรือทราฟิก TCP ไบต์สุดท้ายของ IP แอดเดรสปลายทางจะถูกใช้ ถ้าแพ็กเก็ตไม่ใช่ทราฟิก IP ไบต์สุดท้ายของ MAC แอดเดรสปลายทางจะถูกใช้
  - **dst_port**: UDP ปลายทางหรือค่าของพอร์ต TCP ของแพ็กเก็ตที่จะถูกใช้เพื่อกำหนดอะแด็ปเตอร์ขาออก ถ้าแพ็กเก็ตไม่ใช่ UDP หรือทราฟิก TCP ไบต์สุดท้ายของ IP แอดเดรสปลายทางจะถูกใช้ ถ้าแพ็กเก็ตไม่ใช่ทราฟิก IP ไบต์สุดท้ายของ MAC แอดเดรสปลายทางจะถูกใช้
  - **src_dst_port**: UDP ต้นทางหรือปลายทาง หรือค่าของพอร์ต TCP ของแพ็กเก็ตที่จะถูกใช้เพื่อกำหนดอะแด็ปเตอร์ขาออก (โดยเฉพาะอย่างยิ่ง พอร์ตต้นทางและปลายทางจะถูกเพิ่มและจากนั้นจะถูกหารด้วยสองก่อนที่จะถูกป้อนให้กับอัลกอริทึม) ถ้าแพ็กเก็ตไม่ใช่ UDP หรือทราฟิก TCP ไบต์สุดท้ายของ IP จะถูกใช้ ถ้าแพ็กเก็ตไม่ใช่ทราฟิก IP ไบต์สุดท้ายของ MAC แอดเดรสปลายทางจะถูกใช้ในโหมดนี้สามารถให้การกระจายแพ็กเก็ตที่ดีในหลายสถานการณ์ ทั้งสำหรับไคลเอ็นต์และเซิร์ฟเวอร์

    หมายเหตุ: มันเป็นการรวมที่ไม่ถูกต้องที่จะเลือกโหมด Hash แทนที่จะเป็น `default` กับโหมด`round_robin`

เพื่อเรียนรู้เพิ่มเติมเกี่ยวกับการกระจายแพ็กเก็ตและ load balancing ดูที่ “อ็อพชัน EtherChannel load-balancing” ในหน้า 404

- **Backup Adapter:** ฟิลด์นี้เป็นอ็อพชัน ใส่อะแด็ปเตอร์ที่คุณต้องการใช้เป็น EtherChannel แบ็กอัพ
- **Internet Address to Ping:** ฟิลด์นี้เป็นอ็อพชัน และจะมีผลเมื่อคุณรันโหมด **Network Interface Backup** หรือถ้าคุณมีหนึ่งอะแด็ปเตอร์หรือมากกว่าใน EtherChannel และแบ็กอัพอะแด็ปเตอร์ EtherChannel จะ ping IP แอดเดรสหรือชื่อโฮสต์ที่คุณระบุที่นี่ ถ้า EtherChannel ไม่สามารถ ping แอดเดรสนี้เป็นจำนวนครั้งที่ถูกระบุในฟิลด์ **Number of Retries** และใช้ช่วงเวลาที่ถูกระบุในฟิลด์ **Retry Timeout** EtherChannel จะเปลี่ยนอะแด็ปเตอร์
- **Number of Retries:** ใส่จำนวนความล้มเหลวของการตอบสนองการ ping ที่ยอมให้ก่อนที่ EtherChannel จะสลับอะแด็ปเตอร์ ค่าดีฟอลต์คือ 3 ฟิลด์นี้เป็นอ็อพชันและจะใช้ได้ถ้าคุณตั้ง **Internet Address to Ping**
- **Retry Timeout:** ใส่จำนวนวินาทีระหว่างเวลาที่ EtherChannel จะ ping **Internet Address to Ping** ค่าดีฟอลต์คือ 1 วินาที ฟิลด์นี้เป็นอ็อพชันและจะใช้ได้ถ้าคุณตั้ง **Internet Address to Ping**

5. กด Enter หลังจากเปลี่ยนฟิลด์ที่ต้องการเพื่อสร้าง EtherChannel
6. ตั้งค่า IP บนอุปกรณ์ EtherChannel ที่ถูกสร้างใหม่โดยพิมพ์ `smitty chinet` ที่บรรทัดรับคำสั่ง
7. เลือก EtherChannel อินเตอร์เฟสใหม่ของคุณจากลิสต์
8. เติมฟิลด์ที่ต้องการทั้งหมดและกด Enter

สำหรับงานเพิ่มเติมที่สามารถถูกกระทำหลังจาก EtherChannel ถูกตั้งค่า ดูที่ “การลิสต์ EtherChannels หรือ Link Aggregations” ในหน้า 407

อ็อพชัน Recovery และ failover

คุณลักษณะ Recovery และ failover จะมีสำหรับ EtherChannel หรือ IEEE 802.3ad Link Aggregation

ด้วยคุณลักษณะเหล่านี้ จะทำให้มีการปรับปรุงดังต่อไปนี้:

- สามารถป้องกันแพ็กเก็ตที่หายระหว่างการกู้คืน
- failover สามารถถูกตั้งให้เกิดขึ้นทันที
- การกู้คืนโดยอัตโนมัติสามารถถูกปิดเพื่อที่แบ็กอัพอะแด็ปเตอร์จะทำงานต่อ
- Link Aggregations สามารถถูกบังคับให้ failover จากแชนแนลหลักไปยังแบ็กอัพ หรือกลับกัน

Lossless recovery:

คุณลักษณะ lossless recovery ทำให้แน่ใจว่าการกู้คืนจากแบ็กอัพอะแด็ปเตอร์ไปยังแชนแนลหลักจะสูญเสียแพ็กเก็ตน้อยที่สุดเท่าที่จะเป็นไปได้

ก่อน lossless recovery EtherChannel หรือ IEEE 802.3ad จะกู้คืนไปยังแชนเนลหลักในจังหวะเดียวกับที่มันตรวจพบการกู้คืนของหนึ่งในอะแด็ปเตอร์หลัก ในบางกรณี สวิตช์ของอะแด็ปเตอร์ไม่ได้อยู่ในสถานะที่มันจะจะส่งหรือรับข้อมูล และบางแพ็กเก็ตจะสูญหายทันทีหลังจากการกู้คืน

โดยใช้ lossless recovery EtherChannel หรือ IEEE 802.3ad อะแด็ปเตอร์จะกู้คืนไปยังแชนเนลหลักเฉพาะเมื่อมันสามารถรับทราฟิกบนมันแล้วจริงๆ ซึ่งจะทำให้แน่ใจว่าพอร์ตของสวิตช์จะเริ่มต้นอย่างเต็มที่และไม่มีแพ็กเก็ตสูญหาย

Lossless failover:

คุณลักษณะ lossless failover จะแก้ไขพฤติกรรมของคุณลักษณะ lossless recovery

เมื่อความล้มเหลวของการ ping ทำให้เกิด a failover lossless recovery จะถูกสังเกตโดยดีฟอลต์ ซึ่งจะเกี่ยวข้องกับช่วงเวลาการรอจนกว่าสวิตช์ของอะแด็ปเตอร์ที่ไม่แอ็คทีฟ ได้รับทราฟิกก่อนที่จะสรุปว่า failover อย่างไรก็ตาม ถ้าแอ็ตทริบิวต์ **noloss_failover** ถูกตั้งเป็น `no` การ failover ของ ping จะเกิดขึ้นทันที

การกู้คืนโดยอัตโนมัติ:

หลังจาก failover จากแชนแนลลำดับแรกไปยังอะแด็ปเตอร์สำรอง EtherChannel และ IEEE 802.3ad Link Aggregation จะสตาร์ทการกู้คืนโดยอัตโนมัติไปยังแชนแนลลำดับแรกที่ล้มเหลวเมื่ออย่างน้อยหนึ่งในอะแด็ปเตอร์ของมันถูกกู้คืน

อ็อพชันการกู้คืนนี้ไม่ได้รับการสนับสนุนในโหมด IEEE 802.3ad และ failover ในแบ็คอัพอะแด็ปเตอร์ เนื่องจากความล้มเหลว Link Aggregation Control Protocol (LACP) ความล้มเหลว LACP จะเกิดขึ้นเมื่ออะแด็ปเตอร์ ทั้งหมดในแชนเนลหลักไม่ได้รับหน่วยข้อมูล LACP (LACPDU) ภายในช่วงการหมดเวลา ช่วงการหมดเวลาถูกกำหนด โดยมาตรฐาน IEEE ซึ่งขึ้นอยู่กับช่วงเวลาที่กำหนดคอนฟิกไว้ สำหรับโหมด IEEE 802.3ad

พฤติกรรมดีฟอลต์นี้สามารถถูกดัดแปลงโดยการตั้งแอ็ตทริบิวต์ **auto_recovery** เป็น`no` โดยการตั้งค่านี้ EtherChannel หรือ IEEE 802.3ad Link Aggregation จะทำงานต่อบนอะแด็ปเตอร์สำรองหลังจาก failover การทำงานบนอะแด็ปเตอร์สำรองจะทำต่อไปจนกว่าหนึ่งในเหตุการณ์ต่อไปนี้จะเกิดขึ้น :

- บังคับให้เกิด failover
- แบ็คอัพอะแด็ปเตอร์ล้มเหลว
- ตรวจพบการ ping ที่ล้มเหลวบนแบ็คอัพอะแด็ปเตอร์

การบังคับ failovers:

EtherChannel หรือ IEEE 802.3ad Link Aggregation สามารถถูกบังคับเพื่อ fail over จากแชนแนลหลักไปยังแบ็กอัพอะแด็ปเตอร์ หรือจากแบ็กอัพอะแด็ปเตอร์ไปยังแชนแนลหลัก

การบังคับ failovers จะทำงานเฉพาะถ้ามีการกำหนดแบ็กอัพอะแด็ปเตอร์ และถ้าแชนแนลที่ไม่แอ็คทีฟทำงานอยู่ ตัวอย่างเช่น เพื่อบังคับ failover จากแชนแนลหลักไปยังแบ็กอัพอะแด็ปเตอร์ แบ็กอัพอะแด็ปเตอร์ต้องรันอยู่

เพื่อใช้คุณลักษณะนี้ ใส่ `smitty etherchannel` และเลือกอ็อพชัน **Force A Failover In An EtherChannel / Link Aggregation** จากหน้าจอ จากนั้นเลือก EtherChannel หรือ IEEE 802.3ad Link Aggregation ที่ต้องการบังคับ failover

การตั้งค่า Network Interface Backup

Network Interface Backup จะป้องกันจุดเดียวของเน็ตเวิร์กที่ล้มเหลวโดยจัดเตรียมการตรวจจับความล้มเหลว และ failover ที่ไม่มีการขัดจังหวะการเชื่อมต่อของผู้ใช้ เมื่อทำงานในโหมดนี้ จะมีเพียงอะแด็ปเตอร์เดียวที่แอ็คทีฟในเวลาหนึ่ง

ถ้าแอ็คทีฟอะแด็ปเตอร์ล้มเหลว อะแด็ปเตอร์อื่นใน EtherChannel จะถูกใช้สำหรับทราฟิกทั้งหมด เมื่อทำงานในโหมด Network Interface Backup มันไม่จำเป็นต้องเชื่อมต่อกับสวิตช์แบบ EtherChannel

การตั้งค่า Network Interface Backup จะมีประสิทธิภาพที่สุดเมื่ออะแด็ปเตอร์เชื่อมต่ออยู่กับสวิตช์ของเน็ตเวิร์กอื่น เนื่องจากมันจะให้ redundancy ที่มากกว่าการเชื่อมต่ออะแด็ปเตอร์ทั้งหมดกับสวิตช์เดียว เมื่อเชื่อมต่อกับสวิตช์อื่น ต้องแน่ใจว่ามีการเชื่อมต่อระหว่างสวิตช์ นี่จะให้ความสามารถของ failover อะแด็ปเตอร์หนึ่งไปยังอะแด็ปเตอร์อื่นโดยการทำให้แน่ใจว่าจะมีเส้นทางไปยังอะแด็ปเตอร์ที่กำลังแอ็คทีฟเสมอ

ให้ลำดับความสำคัญกับอะแด็ปเตอร์ที่กำหนดคอนฟิกใน EtherChannel หลัก ผ่านแบ็คอัพอะแด็ปเตอร์ ตราบใดที่อะแด็ปเตอร์หลักยังทำงาน มันจะถูกใช้ นี่ตรงข้ามกับลักษณะการทำงานของโหมด Network Interface Backup ในรีลีสก่อนหน้า ซึ่งมีการใช้แบ็คอัพอะแด็ปเตอร์ จนกว่าจะล้มเหลวด้วย โดยไม่คำนึงว่าอะแด็ปเตอร์หลักมีการ กู้คืนแล้วหรือไม่

ตัวอย่างเช่น `ent0` สามารถถูกตั้งค่าเป็นอะแด็ปเตอร์หลัก และ `ent2` เป็นแบ็กอัพอะแด็ปเตอร์ สร้าง EtherChannel เรียกว่า `ent3` ตามแนวคิด `ent0` และ `ent2` จะถูกเชื่อมต่อกับสวิตช์ 2 ตัวที่แตกต่างกัน ในตัวอย่างนี้ ทราฟิกทั้งหมดถูกส่งผ่าน `ent3` (อินเตอร์เฟสของ EtherChannel) ถูกส่งผ่าน `ent0` โดยดีฟอลต์ โดยที่ `ent2` จะไม่ทำงาน ในเวลาใดๆที่ `ent0` ล้มเหลว ทราฟิกทั้งหมดจะถูกส่งผ่านแบ็กอัพอะแด็ปเตอร์ `ent2` เมื่อ `ent0` ถูกแก้ไข มันจะถูกใช้สำหรับทราฟิกทั้งหมดอีกครั้ง

ตอนนี้เป็นไปได้ที่จะตั้งค่า EtherChannel เพื่อตรวจจับความล้มเหลวของลิงก์ และการที่เน็ตเวิร์กที่ไม่สามารถเข้าถึงได้สำหรับหลาย EtherChannels ที่มีแบ็กอัพอะแด็ปเตอร์ ในการทำดังกล่าว ใช้แอ็ตทริบิวต์ **netaddr** เพื่อระบุ IP แอดเดรสหรือชื่อโฮสต์ของรีโมตโฮสต์ ที่การเชื่อมต่อจะมีอยู่เสมอ EtherChannel จะทำการ ping host นี้เป็นระยะๆเพื่อระบุว่ายังคงมีพาธของเน็ตเวิร์กไปยังมัน ถ้าจำนวนของ ping ที่ระบุไม่ได้รับการตอบ EtherChannel จะ failover ไปยังอะแด็ปเตอร์อื่น โดยหวังว่าจะมีพาธของเน็ตเวิร์กไปยังรีโมตโฮสต์ผ่านอะแด็ปเตอร์อื่น ในการตั้งค่านี้ ไม่ใช่เฉพาะทุกอะแด็ปเตอร์ที่ควรเชื่อมต่อกับสวิตช์อื่น แต่แต่ละสวิตช์ยังมีเส้นทางอื่นไปยังโฮสต์ที่ถูก ping

คุณลักษณะการ ping นี้มีบนหนึ่ง EtherChannels หรือมากกว่าพร้อมกับแบ็กอัพอะแด็ปเตอร์ อย่างไรก็ตาม ถ้ามี failover เนื่องจาก pings ที่ไม่ได้ตอบบนอะแด็ปเตอร์หลัก แบ็คอัพอะแด็ปเตอร์จะยังคงเป็นแชนเนลที่แอ็คทีฟ ตราบใดที่อะแด็ปเตอร์ทำงานอยู่ จะไม่มีทางรู้เลยว่า ขณะทำงานบนแบ็กอัพอะแด็ปเตอร์ มันจะสามารถเข้าถึงโฮสต์ที่ถูก ping จากอะแด็ปเตอร์หลักหรือไม่ เพื่อหลีกเลี่ยงการ failover กลับไปกลับมาระหว่างอะแด็ปเตอร์หลักและแบ็กอัพ มันจะยังคงทำงานบนแบ็กอัพ (แม้ว่าการ ping จะไม่ได้รับคำตอบบนแบ็กอัพอะแด็ปเตอร์ หรือแบ็กอัพอะแด็ปเตอร์ล้มเหลวเอง ในกรณีนี้มันอาจ fail over ไปยังอะแด็ปเตอร์หลัก) อย่างไรก็ตาม ถ้า failover เกิดขึ้นเนื่องจากอะแด็ปเตอร์หลักล้มเหลว (ไม่ใช่เนื่องจาก ping ไม่ได้รับการตอบ) จากนั้น EtherChannel จะกลับไปที่อะแด็ปเตอร์หลักทันทีที่มันกลับเป็นปกติ

เมื่อต้องการกำหนดคอนฟิก Network Interface Backup ในเวอร์ชันที่ใหม่ขึ้น โปรดดู “การกำหนดคอนฟิก Network Interface Backup”

การกำหนดคอนฟิก Network Interface Backup:

ใช้โพรซีเดอร์นี้เพื่อกำหนดคอนฟิกแบ็คอัพอินเตอร์เฟสเครือข่าย ในเวอร์ชันที่ใหม่ขึ้น

1. โดยใช้สิทธิของ root พิมพ์ `smitty etherchannel` บนบรรทัดรับคำสั่ง
2. เลือก **Add an EtherChannel / Link Aggregation** จากลิสต์และกด Enter
3. เลือก Ethernet อะแด็ปเตอร์หลักและกด Enter นี่เป็นอะแด็ปเตอร์ที่จะถูกใช้จนกระทั่งมันล้มเหลว

   หมายเหตุ: ฟิลด์ **Available Network Adapters** จะแสดง Ethernet อะแด็ปเตอร์ทั้งหมด ถ้าคุณเลือก Ethernet อะแด็ปเตอร์ที่ถูกใช้แล้ว คุณจะได้รับข้อความแสดงข้อผิดพลาด และต้องถอดอินเตอร์เฟสนี้ก่อนที่คุณจะสามารถใช้มัน อ้างถึง “การเปลี่ยนแปลงใน EtherChannel ด้วย 5200-01 และก่อนหน้านี้” ในหน้า 409 สำหรับข้อมูลเกี่ยวกับวิธีการถอดอินเตอร์เฟส

4. ใส่ข้อมูลในฟิลด์ต่อไปนี้ตามแนวทางต่อไปนี้ :
   - **Parent Adapter**: ฟิลด์นี้จะให้ข้อมูลของอุปกรณ์พาเรนต์ของ EtherChannel (ตัวอย่างเช่น เมื่อ EtherChannel เป็นของ Shared Ethernet Adapter) ฟิลด์นี้จะแสดงค่าของ NONE ถ้า EtherChannel ไม่ได้อยู่ในอะแด็ปเตอร์อื่น (โดยดีฟอลต์) ถ้า EtherChannel อยู่ในอะแด็ปเตอร์อื่น ฟิลด์นี้จะแสดงชื่อของพาเรนต์อะแด็ปเตอร์ (ตัวอย่างเช่น ent6) ฟิลด์นี้จะให้ข้อมูลเท่านั้นและไม่สามารถแก้ไขได้ อ็อพชันแบ็คอัพอะแด็ปเตอร์พร้อมใช้งานใน ระบบปฏิบัติการ AIX
   - **EtherChannel / Link Aggregation Adapters**: คุณควรเห็นอะแด็ปเตอร์หลักที่คุณเลือกในขั้นตอนก่อนหน้านี้
   - **Enable Alternate Address**: ฟิลด์นี้เป็นอ็อพชัน ตั้งเป็น yes จะให้คุณสามารถระบุ MAC แอดเดรสที่คุณต้องการใช้ EtherChannel ถ้าคุณตั้งอ็อพชันนี้เป็น no EtherChannel จะใช้ MAC แอดเดรสของอะแด็ปเตอร์แรกที่ถูกระบุ
   - **Alternate Address**: ถ้าคุณตั้ง **Enable Alternate Address** เป็น yes ระบุ MAC แอดเดรสที่คุณต้องการใช้ที่นี่ แอดเดรสที่คุณระบุต้องเริ่มต้นด้วย 0x และเป็นแอดเดรสเลขฐานสิบหก 12-หลัก (ตัวอย่างเช่น 0x001122334455)
   - **Enable Gigabit Ethernet Jumbo Frames**: ฟิลด์นี้เป็นอ็อพชัน เพื่อที่จะใช้อ็อพชันนี้ การสวิตช์ของคุณต้องสนับสนุนเฟรมขนาดใหญ่ นี่จะใช้ได้กับอินเตอร์เฟส Standard Ethernet (en) เท่านั้น ไม่ใช่กับอินเตอร์เฟส IEEE 802.3 (et) ตั้งค่านี้เป็น yes ถ้าคุณต้องการใช้มัน
   - **Mode**: นี่ไม่เข้ากับโหมดการทำงานใดที่คุณเลือก เนื่องจากจะมีเพียงอะแด็ปเตอร์เดียวใน EtherChannel หลัก แพ็กเก็ตทั้งหมดที่ถูกส่งผ่านอะแด็ปเตอร์นั้นจนกว่ามันจะล้มเหลว จะไม่มีโหมด netif_backup เนื่องจากโหมดนั้นสามารถถูกอีมูเลตโดยใช้แบ็กอัพอะแด็ปเตอร์
   - **Mode**: นี่ไม่เข้ากับโหมด hash ที่คุณเลือก เนื่องจากจะมีเพียงอะแด็ปเตอร์เดียวใน EtherChannel หลัก แพ็กเก็ตทั้งหมดที่ถูกส่งผ่านอะแด็ปเตอร์นั้นจนกว่ามันจะล้มเหลว
   - **Backup Adapter**: ใส่อะแด็ปเตอร์ที่คุณต้องการให้เป็นแบ็กอัพอะแด็ปเตอร์ หลังจาก failover อะแด็ปเตอร์นี้จะถูกใช้ขนกว่าอะแด็ปเตอร์หลักจะถูกแก้ไข แนะนำให้ใช้อะแด็ปเตอร์ที่ต้องการเป็นอะแด็ปเตอร์หลัก
   - **Internet Address to Ping**: ฟิลด์นี้เป็นอ็อพชัน EtherChannel จะ ping IP แอดเดรส หรือชื่อโฮสต์ที่คุณระบุที่นี่ ถ้า EtherChannel ไม่สามารถ ping แอดเดรสนี้เป็นจำนวนครั้งที่ถูกระบุในฟิลด์ **Number of Retries** และใช้ช่วงเวลาที่ถูกระบุในฟิลด์ **Retry Timeout** EtherChannel จะเปลี่ยนอะแด็ปเตอร์
   - **Number of Retries**: ใส่จำนวนความล้มเหลวของการตอบสนองการ ping ที่ยอมให้ก่อนที่ EtherChannel จะสลับอะแด็ปเตอร์ ค่าดีฟอลต์คือ 3 ฟิลด์นี้เป็นอ็อพชันและจะใช้ได้ถ้าคุณตั้ง **Internet Address to Ping**
   - **Retry Timeout**: ใส่จำนวนวินาทีระหว่างเวลาที่ EtherChannel จะ ping **Internet Address to Ping** ค่าดีฟอลต์คือ 1 วินาที ฟิลด์นี้เป็นอ็อพชันและจะใช้ได้ถ้าคุณตั้ง **Internet Address to Ping**
5. กด Enter หลังจากเปลี่ยนฟิลด์ที่ต้องการเพื่อสร้าง EtherChannel
6. ตั้งค่า IP บนอินเตอร์เฟสใหม่โดยการพิมพ์ smitty chinet ที่บรรทัดรับคำสั่ง
7. เลือก EtherChannel อินเตอร์เฟสใหม่ของคุณจากลิสต์
8. เติมฟิลด์ที่ต้องการทั้งหมดและกด Enter

network interface backup ของคุณถูกตั้งค่าแล้ว

อ็อพชัน EtherChannel load-balancing

วิธีการการทำ load balancing อยู่ 2 วิธีสำหรับทราฟิกขาออกใน EtherChannel ดังต่อไปนี้: วิธี round-robin ซึ่งจะกระจายทราฟิกขาออกเท่าๆกันไปยังอะแด็ปเตอร์ทั้งหมดใน EtherChannel และวิธีมาตรฐาน ซึ่งจะเลือกอะแด็ปเตอร์โดยใช้อัลกอริทึม

พารามิเตอร์ Hash Mode จะกำหนดค่าเป็นตัวเลขที่จะถูกป้อนให้กับอัลกอริทึม

ตารางสรุปต่อไปนี้ให้การรวมอ็อพชัน load-balancing ที่ใช้ได้

ตารางที่ 82. การรวม Mode และ Hash Mode และการกระจายทราฟิกขาออกที่แต่ละตัวสร้างขึ้น

โหมด	Hash Mode	Outgoing Traffic Distribution
มาตรฐานหรือ 8023ad	default	พฤติกรรมของ AIX แบบดั้งเดิม อัลกอริทึมการเลือกอะแด็ปเตอร์ใช้ไบต์สุดท้ายของ IP แอดเดรสปลายทาง (สำหรับทราฟิก **TCP/IP**) หรือ MAC อะแด็ปเตอร์ปลายทาง (สำหรับ ARP และทราฟิกอื่นที่ไม่ใช่ IP) โดยทั่วไปโหมดนี้จะเป็นตัวเลือกเริ่มต้นที่ดีสำหรับเซิร์ฟเวอร์ที่มีไคลเอ็นต์จำนวนมาก
มาตรฐานหรือ 8023ad	src_dst_port	พาธของอะแด็ปเตอร์ขาออกถูกเลือกโดยอัลกอริทึมโดยการ TCP ต้นทางและปลายทาง หรือค่าของพอร์ต UDP เนื่องจากแต่ละการเชื่อมต่อจะมีพอร์ต TCP หรือ UDP เฉพาะ hash โหมดแบบ 3 พอร์ตให้ความยืดหยุ่นในการกระจายไปยังอะแด็ปเตอร์เพิ่มเติม เมื่อมีหลายการเชื่อมต่อ TCP หรือ UDP ที่แยกกันระหว่างคู่ของ IP แอดเดรส
มาตรฐานหรือ 8023ad	src_port	อัลกอริทึมการเลือกอะแด็ปเตอร์จะใช้ TCP ต้นทางหรือค่าพอร์ต UDP ในเอาต์พุตของคำสั่ง **netstat -an** พอร์ตจะเป็นค่าของส่วนที่ต่อท้าย TCP/IP แอดเดรส ในคอลัมน์ `Local`
มาตรฐานหรือ 8023ad	dst_port	พาธของอะแด็ปเตอร์ขาออกถูกเลือกโดยอัลกอริทึมโดยใช้ค่าของพอร์ตของระบบปลายทาง ในเอาต์พุตของคำสั่ง **netstat -an**ส่วนต่อท้ายของ TCP/IP คอลัมน์ในคอลัมน์ `Foreign` จะเป็นค่าของพอร์ต TCP หรือ UDP ปลายทาง
round-robin	default	ทราฟิกขาออกจะถูกกระจายเท่าๆกันบนพอร์ตทั้งหมดของอะแด็ปเตอร์ใน EtherChannel โดยทั่วไปโหมดนี้จะเป็นตัวเลือกของโฮสต์ 2 ตัวที่เชื่อมต่อกันแบบ back-to-back (โดยไม่ใช้สวิตช์)

การกระจายแบบ Round-Robin:

ทราฟิกขาออกทั้งหมดจะถูกกระจายโดยเท่าๆกันในอะแด็ปเตอร์ทั้งหมดใน EtherChannel มันจะให้การใช้งานแบนด์วิดธ์ที่เหมาะสมที่สุดสำหรับระบบ AIX เซิร์ฟเวอร์ ขณะที่การกระจายแบบ round-robin เป็นวิธีที่เป็นแนวคิดที่จะใช้ลิงก์ทั้งหมดเท่าๆกัน พิจารณาว่ามันยังแนะนำความเป็นไปได้สำหรับแพ็กเก็ตที่ไม่เป็นไปตามลำดับที่ระบบที่รับ

โดยทั่วไปโหมดเป็นแนวคิดสำหรับการเชื่อมต่อแบบ back-to-back ที่รันเฟรมขนาดใหญ่ ในสภาวะแวดล้อมนี้ จะไม่มีการใช้สวิตช์ดังนั้นจะไม่มีโอกาสที่กระบวนการที่สวิตช์จะเปลี่ยนเวลาของการส่งแพ็กเก็ต ลำดับ หรือพาธของอะแด็ปเตอร์ บนเน็ตเวิร์กพาธแบบใช้สายตรงนี้ แพ็กเก็ตจะถูกได้รับเหมือนกับที่มันส่ง เฟรมแบบ Jumbo (MTU ขนาด 9000 ไบต์) จะให้ประสิทธิภาพของการถ่ายโอนไฟล์ที่ดีกว่าขนาดของ MTU 1500 ไบต์แบบดั้งเดิม อย่างไรก็ตาม ในกรณีนี้ ยังมีประโยชน์อื่นเพิ่มเติม แพ็กเก็ตขนาดใหญ่เหล่านี้ใช้เวลานานในการส่ง ดังนั้นมันสามารถเป็นไปได้น้อยที่โฮสต์ที่เป็นตัวรับจะถูกขัดจังหวะโดยแพ็กเก็ตที่ไม่เป็นไปตามลำดับ

โหมดแบบ Round-robin สามารถถูกใช้ในสภาวะแวดล้อมอื่นแต่จะเพิ่มความเสี่ยงที่จะมีแพ็กเก็ตที่ไม่เป็นไปตามลำดับที่ระบบที่เป็นตัวรับ ความเสี่ยงนี้จะสูงขึ้น เมื่อมีสตรีมของการเชื่อมต่อ TCP ที่น้อย และนาน เมื่อมีการเชื่อมต่อดังกล่าวจำนวนมากระหว่างคู่ของโฮสต์ แพ็กเก็ตจากการเชื่อมต่ออื่นสามารถแทรกเข้ามา ดังนั้นจะลดโอกาสที่แพ็กเก็ตสำหรับการเชื่อมต่อเดียวกันจะมาถึงไม่เป็นไปตามลำดับ ตรวจสอบสถิติสำหรับแพ็กเก็ตที่ไม่เป็นไปตามลำดับในส่วน `tcp` ของเอาต์พุตของคำสั่ง **netstat -s** ค่าที่เพิ่มขึ้นเรื่อยๆจะระบุว่ามีความเป็นไปได้ของปัญหาในทราฟิกที่ส่งจาก EtherChannel

ถ้าแพ็กเก็ตที่ไม่เป็นไปตามลำดับเป็นปัญหาบนระบบที่ต้องใช้ Ethernet MTU แบบดั้งเดิม และต้องเชื่อมต่อผ่านสวิตช์ให้ใช้การทำงานในโหมดมาตรฐาน แต่ละโหมดมีจุดแข็งที่ต่างกัน แต่โหมดมาตรฐานและ src_dst_port เป็นจุดเริ่มต้นแบบโลจิคัลที่สามารถใช้ได้อย่างกว้างขวาง

อัลกอริทึมมาตรฐาน หรือ 8032ad:

มีข้อดีในการใช้อัลกอริทึม EtherChannel มาตรฐาน

อัลกอริทึมมาตรฐานถูกใช้สำหรับทั้งการรวมลิงก์แบบมาตรฐาน และ IEEE 802.3ad-style AIX หารไบต์สุดท้ายของ "ค่าที่เป็นตัวเลข" ด้วยจำนวนของอะแด็ปเตอร์ใน EtherChannel และใช้เศษที่เหลือเพื่อระบุลิงก์ขาออก ถ้าเศษที่เหลือเป็นศูนย์ อะแด็ปเตอร์แรกใน EtherChannel จะถูกเลือก เศษที่เหลือของหนึ่งจะหมายถึงอะแด็ปเตอร์ที่สอง และอื่นๆ (อะแด็ปเตอร์ถูกเลือกตามลำดับที่มันถูกลิสต์ในแอ็ตทริบิวต์ **adapter_names**)

การเลือก Hash Mode จะกำหนดค่าที่เป็นตัวเลขที่ใช้ในการคำนวณ โดยดีฟอลต์ ไบต์สุดท้ายของ IP แอดเดรส หรือ MAC แอดเดรสปลายทาง ถูกใช้ในการคำนวณ แต่ค่าพอร์ต TCP หรือ UDP ต้นทางและปลายทางยังสามารถถูกใช้ด้วย วิธีเหล่านี้ทำให้คุณสามารถปรับแต่งการกระจายของทราฟิกขาออกข้ามอะแด็ปเตอร์จริงใน EtherChannel อย่างละเอียด

ในโหมด hash แบบดีฟอลต์ อัลกอริทึมการเลือกอะแด็ปเตอร์ถูกใช้กับไบต์สุดท้ายของ IP แอดเดรสปลายทางสำหรับทราฟิก IP สำหรับ ARP และทราฟิกอื่นที่ไม่ใช่ IP สูตรเดียวกันจะถูกใช้บนไบต์สุดท้ายของ MAC แอดเดรสปลายทาง ยกเว้นจะมีอะแด็ปเตอร์ที่ล้มเหลวที่ทำให้เกิด failover ทราฟิกทั้งหมดระหว่างคู่ของโฮสต์ในโหมดมาตรฐานแบบดีฟอลต์จะออกไปที่อะแด็ปเตอร์เดียวกัน โหมด hash แบบดีฟอลต์อาจเป็นแนวคิดเมื่อโลคัลโฮสต์สร้างการเชื่อมต่อกับหลาย IP แอดเดรสที่ต่างกัน

ถ้าโลคัลโฮสต์สร้างการเชื่อมต่อแบบยาวไปยัง IP แอดเดรสจพนวนเล็กน้อย คุณจะสังเกตว่าบางอะแด็ปเตอร์จะมีโหลดมากกว่าอะแด็ปเตอร์อื่น เนื่องจากทราฟิกทั้งหมดถูกส่งไปยังปลายทางที่ระบุถูกส่งบนอะแด็ปเตอร์เดียวกัน โดยป้องกันไม่ให้แพ็กเก็ตถูกส่งมาถึงแบบไม่เป็นไปตามลำดับ มันอาจไม่ได้ใช้แบนด์วิดธ์อย่างมีประสิทธิภาพในทุกกรณี โหมด hash แบบ port-based ยังคงส่งแพ็กเก็ตตามลำดับ แต่มันจะยอมให้แพ็กเก็ตที่เป็นของการเชื่อมต่อ UDP หรือ TCP อื่น ถูกส่งบนอะแด็ปเตอร์อื่นแม้ว่ามันถูกส่งไปยังปลายทางเดียวกัน ซึ่งจะเป็นการใช้แบนด์วิดธ์ของอะแด็ปเตอร์ทั้งหมด

ใน **src_dst_port** hash โหมด ค่าพอร์ตของ TCP หรือ UDP ต้นทางและปลายทางของแพ็กเก็ตขาออกจะถูกเพิ่ม จากนั้นหารด้วยสอง ผลลัพธ์ของจำนวนทั้งหมด (ไม่ใช่ทศนิยม) จะถูกใช้กับอัลกอริทึมมาตรฐาน ทราฟิก TCP หรือ UDP ถูกส่งบนอะแด็ปเตอร์ที่ถูกเลือกโดยอัลกอริทึมมาตรฐานและค่าโหมด hash ที่ถูกเลือก ทราฟิกที่ไม่ใช่ TCP หรือ UDP จะถูกใช้โหมด hash แบบดีฟอลต์ หมายความว่าไบต์สุดท้ายของ IP แอดเดรส หรือ MAC แอดเดรสปบายทาง อ็อพชันของ **src_dst_port** hash โหมดจะพิจารณาทั้งค่าของพอร์ต TCP หรือ UDP ปลายทาง ในโหมดนี้ แพ็กเก็ตทั้งหมดในการเชื่อมต่อ TCP หรือ UDP เดียวจะถูกส่งบนอะแด็ปเตอร์เดียว ดังนั้นจึงรับประกันได้ว่ามันจะมาถึงตามลำดับ แต่ทราฟิกยังควถูกกระจายออกเนื่องจากการเชื่อมต่อ (แม้จะไปยังโฮสต์เดียวกัน) อาจถูกส่งผ่านอะแด็ปเตอร์ที่ต่างกัน ผลของโหมด hash นี้จะไม่เยื้องกันโดยทิศทางการการสร้างการเชื่อมต่อ เนื่องจากมันใช้ทั้งค่าพอร์ตของ TCP หรือ UDP ต้นทางและปลายทาง

ใน **src_port** hash โหมด ค่าพอร์ตของ TCP หรือ UDP ต้นทางของแพ็กเก็ตขอออกจะถูกใช้ใน **dst_port** hash โหมด ค่าพอร์ตของ TCP หรือ UDP ปลายทางของแพ็กเก็ตขอออกจะถูกใช้ใช้อ็อพชันของ **src_port** หรือ **dst_port** hash โหมด ถ้าค่าของพอร์ตเปลี่ยนจากการเชื่อมต่อหนึ่งไปยังการเชื่อมต่ออื่น และถ้าอ็อพชัน **src_dst_port** ไม่ย้อมให้มีการกระจายที่ต้องการ

การลิสต์ EtherChannels หรือ Link Aggregations

ใช้โพรซีเดอร์นี้เพื่อลิสต์ EtherChannels หรือ Link Aggregations

1. บนบรรทัดคำสั่ง พิมพ์ `smitty etherchannel`
2. เลือก **List All EtherChannels / Link Aggregations** และกด Enter

การเปลี่ยน alternate address

เพื่อระบุ MAC แอดเดรสสำหรับ EtherChannel หรือ Link Aggregation ของคุณ ทำตามขั้นตอนเหล่านี้

1. ขึ้นอยู่กับเวอร์ชันของ AIX ที่คุณรันอยู่ คุณอาจต้องถอดอินเตอร์เฟส :
   - บน AIX 5.2 ที่มี 5200-01 และก่อนหน้านี้ พิมพ์ `smitty chinet` และเลือกอินเตอร์เฟสที่เป็นของ EtherChannel ของคุณ เปลี่ยนแอ็ตทริบิวต์ **Current STATE** เป็น **detach** และกด Enter
   - บน AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03 และหลังจากนั้น คุณสามารถเปลี่ยน alternate address ของ EtherChannel ดดยไม่ต้องถอดอินเตอร์เฟส
2. บนบรรทัดคำสั่ง พิมพ์ `smitty etherchannel`
3. เลือก **Change / Show Characteristics of an EtherChannel** และกด Enter
4. ถ้าคุณมีหลาย EtherChannels เลือก EtherChannel ที่คุณต้องการสร้างแอดเดรสอื่น
5. เปลี่ยนค่าใน **Enable Alternate EtherChannel Address** เป็น `yes`
6. ใส่แอดเดรสอื่นในฟิลด์ **Alternate EtherChannel Address** แอดเดรสต้องเริ่มด้วย `0x` และเป็นแอดเดรสเลขฐานสิบหก 12-หลัก (ตัวอย่างเช่น 0x001122334455)
7. กด Enter เพื่อทำกระบวนการให้สำเร็จ

หมายเหตุ: การเปลี่ยน MAC แอดเดรสของ EtherChannel ที่รันไทม์สามารถทำให้การเชื่อมต่อขาดหายชั่วคราว เนื่องจากอะแด็ปเตอร์ต้องถูกรีเซ็ตเพื่อที่มันจะเรียนรู้อะแด็ปเตอร์แอดเดรสใหม่ และบางอะแด็ปเตอร์ใช้ไม่กี่วินาทีเพื่อที่จะเริ่มต้น

Dynamic Adapter Membership

ก่อนหน้า AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03 เพื่อที่จะเพิ่มหรือลบอะแด็ปเตอร์จาก EtherChannel อินเตอร์เฟสของมันต้องถูกถอดออกก่อน ซึ่งจะขัดจังหวะทราฟิกทั้งหมดของผู้ใช้ เพื่อข้ามข้อจำกัดนี้ Dynamic Adapter Membership (DAM) ได้ถูกเพิ่มใน AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03

มันทำให้อะแด็ปเตอร์ถูกเพิ่มหรือลบจาก EtherChannel โดยไม่ต้องขัดจังหวะการเชื่อมต่อของผู้ใช้ แบ็กอัพอะแด็ปเตอร์ยังสามารถถูกเพิ่มหรือลบ EtherChannel สามารถถูกสร้างโดยไม่ต้องมีแบ็กอัพอะแด็ปเตอร์โดยสามารถถูกเพิ่มเข้าไปภายหลังถ้าต้องการ

ไม่ใช้เฉพาะอะแด็ปเตอร์สามารถถูกเพิ่มและลบโดยไม่ขัดจังหวะการเชื่อมต่อของผู้ใช้ แอ็ตทริบิวต์ส่วนใหญ่ของ EtherChannel ยังสามารถแก้ไขที่รันไทม์ ตัวอย่างเช่น คุณอาจเริ่มต้นใช้คุณลักษณะ "ping" ของ Network Interface Backup ขณะที่ EtherChannel กำลังถูกใช้งาน หรือเปลี่ยนรีโมตโฮสต์ที่กำลังถูก ping เมื่อใดก็ได้

คุณยังสามารถเปลี่ยน EtherChannel ธรรมดาเป็น IEEE 802.3ad Link Aggregation (หรือกลับกัน) ทำให้ผู้ใช้สามารถใช้คุณลักษณะนี้โดยไม่ต้องลบและสร้าง EtherChannel ใหม่

นอกจากนี้โดยใช้ DAM คุณมีทางเลือกที่จะสร้าง EtherChannel แบบอะแด็ปเตอร์เดียว EtherChannel แบบอะแด็ปเตอร์เดียวทำหน้าที่เหมือนกับอะแด็ปเตอร์ธรรมดา อย่างไรก็ตาม เมื่ออะแด็ปเตอร์นี้ล้มเหลวมันจะสามารถถูกแทนที่ได้ที่รันไทม์โดยที่การเชื่อมต่อไม่ขาด ในการทำดังกล่าว คุณจะเพิ่มอะแด็ปเตอร์ชั่วคราวเข้ากับ EtherChannel ลบอะแด็ปเตอร์ที่มีปัญหาจาก EtherChannel เปลี่ยนอะแด็ปเตอร์ที่มีปัญหาโดยใช้ Hot Plug เพิ่มอะแด็ปเตอร์ใหม่เข้ากับ EtherChannel และจากนั้นลบอะแด็ปเตอร์ชั่วคราวออก ระหว่างกระบวนการนี้คุณจะไม่สังเกตเลยว่าการเชื่อมต่อขาดไป อย่างไรก็ตาม ถ้าอะแด็ปเตอร์ถูกใช้งานเป็นอะแด็ปเตอร์แบบสแตนด์อะโลน มันควรถูกถอดออกก่อนที่จะถูกลบโดยใช้ Hot Plug และระหว่างเวลานั้นทราฟิกจะหายไป

การเพิ่มอะแด็ปเตอร์ ลบ หรือเปลี่ยนใน EtherChannel หรือ Link Aggregation

มี 2 วิธีที่จะเพิ่ม ลบ หรือเปลี่ยนอะแด็ปเตอร์ใน EtherChannel หรือ Link Aggregation

หนึ่งวิธีที่ต้องการให้ถอด EtherChannel หรือ Link Aggregation interface ขณะที่วิธธอื่นไม่ต้อง (โดยใช้ Dynamic Adapter Membership ซึ่งมีใน AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03 และหลังจากนั้น)

การทำการเปลี่ยนแปลงกับ EtherChannel โดยใช้ Dynamic Adapter Membership:

การทำการเปลี่ยนแปลงโดยใช้ Dynamic Adapter Membership คุณจะไม่ต้องหยุดทราฟิกที่วิ่งผ่าน EtherChannel โดยการถอดอินเตอร์เฟสของมัน

พิจารณาต่อไปนี้ก่อนที่จะทำต่อ :

1. เมื่อเพิ่มอะแด็ปเตอร์ที่รันไทม์โปรดสังเกตว่า Ethernet อะแด็ปเตอร์ที่ต่างกันจะมีความสามารถที่ต่างกัน (ตัวอย่างเช่น ความสามารถในการลดโหลดของ checksum การใช้เซ็กเมนต์ส่วนตัว ส่งทำการส่งขนาดใหญ่ และอื่นๆ) ถ้าชนิดของอะแด็ปเตอร์ที่ต่างกันถูกใช้ใน EtherChannel เดียวกัน ความสามารถที่รายงานไปยังเลเยอร์ของอินเตอร์เฟสจะเป็นที่ได้รับการสนับสนุนโดยอะแด็ปเตอร์ทั้งหมด (ตัวอย่างเช่น ถ้าอะแด็ปเตอร์ทั้งหมดสนับสนุนการใช้เซกเมนต์ส่วนตัวยกเว้นหนึ่งอะแด็ปเตอร์ EtherChannel จะบอกว่ามันไม่สนับสนุนเซกเมนต์ส่วนตัว ถ้าอะแด็ปเตอร์ทั้งหมดสนับสนุนการส่งขนาดใหญ่ แชนแนลจะบอกว่ามันสนับสนุนการส่งขนาดใหญ่) เมื่อเพิ่มอะแด็ปเตอร์เข้ากับ EtherChannel ที่รันไทม์ ต้องแน่ใจว่ามันสนันสนุนอย่างน้อยที่สุดมันมีความสามารถเหมือนกับอะแด็ปเตอร์อื่นที่อยู่ใน EtherChannel ถ้าคุณพยายามจะเพิ่มอะแด็ปเตอร์ที่ไม่มีสนับสนุนความสามารถทั้งหมดที่ EtherChannel สนับสนุน การเพิ่มทั้งหมดจะล้มเหลว อย่างไรก็ตาม โปรดสังเกตว่าถ้าอินเตอร์เฟสของ EtherChannel ถูกถอดออก คุณอาจจะเพิ่มอะแด็ปเตอร์ใดๆ (โดยไม่ต้องสนใจความสามารถที่มันสนับสนุน) และเมื่ออินเตอร์เฟสถูกเปิดใช้งานอีกครั้ง EtherChannel จะคำนวณความสามารถที่มันสนับสนุนใหม่โดยขึ้นอยู่กับลิสต์ของอะแด็ปเตอร์ใหม่
2. ถ้าคุณไม่ได้ใช้แอดเดรสอื่น และคุณวางแผนที่จะลบอะแด็ปเตอร์ที่ MAC แอดเดรสถูกใช้สำหรับ EtherChannel (MAC แอดเดรสที่ถูกใช้สำหรับ "ถูกเป็นเจ้าของ" โดยหนึ่งในอะแด็ปเตอร์เหล่านั้น) EtherChannel จะใช้ MAC แอดเดรสของอะแด็ปเตอร์ถัดไปที่ว่าง อีกนัยหนึ่ง อะแด็ปเตอร์ที่กลายเป็นอะแด็ปเตอร์แรกหลังจากการลบ หรือแบ็กอัพอะแด็ปเตอร์ในกรณีที่อะแด็ปเตอร์หลักทั้งหมดถูกลบ ตัวอย่างเช่น ถ้า EtherChannel มีอะแด็ปเตอร์หลัก `ent0` และ `ent1` และแบ็กอัพอะแด็ปเตอร์ `ent2` โดยดีฟอลต์ มันจะใช้ MAC แอดเดรสของ `ent0` (โดยมันจะบอกว่า `ent0` "เป็นเจ้าของ" MAC แอดเดรส) ถ้า `ent0` ถูกลบ ดังนั้น EtherChannel จะใช้ MAC แอดเดรสของ `ent1` จากนั้น ถ้า `ent1` ถูกลบ EtherChannel จะใช้ MAC แอดเดรสของ `ent2` ถ้า `ent0` ถูกเพิ่มเข้ากับ EtherChannel ภายหลัง มันจะยังคงใช้ MAC แอดเดรส-v' `ent2` เนื่องจากตอนนี้ `ent2` เป็นเจ้าของ MAC แอดเดรส จากนั้น ถ้า `ent2` ถูกลบจาก EtherChannel มันจะเริ่มใช้ MAC แอดเดรสของ `ent0` อีกครั้ง

การลบอะแด็ปเตอร์ที่ MAC แอดเดรส ถูกใช้สำหรับ EtherChannel อาจทำให้การเชื่อมต่อขาดหายชั่วคราว เนื่องจากอะแด็ปเตอร์ทั้งหมดใน EtherChannel ต้องถูกรีเซ็ตใหม่เพื่อที่มันจะเรียนรู้ฮาร์ดแวร์แอดเดรสใหม่ของมัน บางอะแด็ปเตอร์ใช้ไม่กี่วินาทีในการเริ่มต้น

ถ้า EtherChannel ของคุณใช้ alternate address (MAC แอดเดรสที่คุณระบุ) มันจะยังคงใช้ MAC แอดเดรสนี้ดดยไม่สนใจว่าอะแด็ปเตอร์ถูกเพิ่มหรือถูกลบ นอกจากนั้น มันยังหมายความว่าจะการขาดหายของการเชื่อมต่อชั่วคราวจะไม่เกิดขึ้นเมื่อเพิ่มหรือลบอะแด็ปเตอร์ เนื่องจากไม่มีอะแด็ปเตอร์ที่ "เป็นเจ้าของ" MAC แอดเดรสของ EtherChannel

3. ดังนั้น แอ็ตทริบิวต์ของ EtherChannel ส่วนใหญ่ทั้งหมดสามารถถูกแก้ไขที่รันไทม์ ยกเว้น **Enable Gigabit Ethernet Jumbo Frames** เพื่อแก้ไขแอ็ตทริบิวต์ **Enable Gigabit Ethernet Jumbo Frames** คุณต้องถอดอินเตอร์เฟสของ EtherChannel ออกก่อนที่จะทำการแก้ไขค่านี้
4. สำหรับแอ็ตทริบิวต์ใดๆที่ไม่สามารถถูกแก้ไขที่รันไทม์ (ปัจจุบัน เฉพาะ **Enable Gigabit Ethernet Jumbo Frames** เท่านั้น) จะมีฟอลด์ชื่อ **Apply change to DATABASE only** ถ้าแอ็ตทริบิวต์นี้ถูกตั้งเป็น `yes` เป็นไปได้ที่จะเปลี่ยนที่รันไทม์ค่าของแอ็ตทริบิวต์ที่โดยทั่วไปจะไม่สามารถถูกเปลี่ยนที่รันไทม์โดยที่ฟิลด์ **Apply change to DATABASE only** ถูกตั้งเป็น `yes` แอ็ตทริบิวต์จะถูกเปลี่ยนใน ODM เท่านั้น และจะไม่สามารถถูกแสดงใน EtherChannel จนกว่ามันจะถูกโหลดใหม่เข้าไปในหน่วยความจำ (โดยการถอดอินเตอร์เฟสของมัน โดยใช้คำสั่ง `rmdev -l EtherChannel_device` และจากนั้น `mkdev -l EtherChannel_device`) หรือจนกว่าเครื่องจะถูกรีบูต นี่จะเป็นวิธีที่สะดวกที่สุดที่จะทำให้แน่ใจว่าแอ็ตทริบิวต์ถูกแก้ไขครั้งต่อไปที่เครื่องบูต โดยไม่ต้องขัดจังหวะการทำงานของ EtherChannel

เพื่อทำการเปลี่ยนแปลงกับ EtherChannel หรือ Link Aggregation โดยใช้ Dynamic Adapter Membership ทำตามขั้นตอนเหล่านี้:

1. ที่บรรทัดรับคำสั่ง พิมพ์ `smitty etherchannel`
2. เลือก **Change / Show Characteristics of an EtherChannel / Link Aggregation**
3. เลือก EtherChannel หรือ Link Aggregation ที่คุณต้องการแก้ไข
4. เติมฟิลด์ที่ต้องการตามแนวทางต่อไปนี้:
   - ในฟิลด์ **Add adapter** หรือ **Remove adapter** เลือก Ethernet อะแด็ปเตอร์ที่คุณต้องการเพิ่มหรือลบ
   - ในฟิลด์ **Add backup adapter** หรือ **Remove backup adapter** เลือก Ethernet อะแด็ปเตอร์ที่คุณต้องการสตาร์ทหรือหยุดใช้เป็นแบ็กอัพ
   - แอ็ตทริบิวต์ของ EtherChannel ทั้งหมดส่วนใหญ่ สามารถแก้ไขได้ที่รันไทม์ แม้ว่าแอ็ตทริบิวต์ **Enable Gigabit Ethernet Jumbo Frames** จะไม่สามารถทำได้
   - เพื่อเปลี่ยน EtherChannel ธรรมดาให้เป็น IEEE 802.3ad Link Aggregation เปลี่ยนแอ็ตทริบิวต์ **Mode** เป็น `8023ad` เพื่อเปลี่ยน IEEE 802.3ad Link Aggregation เป็น EtherChannel เปลี่ยนแอ็ตทริบิวต์ **Mode** เป็น `standard` หรือ `round_robin`
5. เติมข้อมูลที่จำเป็น และกด Enter

การเปลี่ยนแปลงใน EtherChannel ด้วย 5200-01 และก่อนหน้านี้:

ใช้โพรซีเดอร์นี้เพื่อถอดอินเตอร์เฟส และเปลี่ยนแปลง ใน EtherChannel ด้วย 5200-01 และก่อนหน้านี้

1. พิมพ์ `smitty chinet` และเลือกอินเตอร์เฟสที่เป็นของ EtherChannel ของคุณ เปลี่ยนแอ็ตทริบิวต์ **Current STATE** เป็น **detach** และกด Enter
2. บนบรรทัดคำสั่ง พิมพ์ `smitty etherchannel`
3. เลือก **Change / Show Characteristics of an EtherChannel / Link Aggregation** และกด Enter

4. เลือก EtherChannel หรือ Link Aggregation ที่คุณต้องการแก้ไข
5. แก้ไขแอ็ตทริบิวต์ที่คุณต้องการเปลี่ยนใน EtherChannel ของคุณหรือ Link Aggregation และกด Enter
6. เติมฟิลด์ที่จำเป็นและกด Enter

การลบ EtherChannel หรือ Link Aggregation:

ใช้โพรซีเดอร์นี้เพื่อลบ EtherChannel หรือ Link Aggregation

1. พิมพ์ `smitty chinet` และเลือกอินเตอร์เฟสที่เป็นของ EtherChannel ของคุณ เปลี่ยนแอ็ตทริบิวต์ **Current STATE** เป็น **detach** และกด Enter
2. บนบรรทัดคำสั่ง พิมพ์ `smitty etherchannel`
3. เลือก **Remove an EtherChannel** และกด Enter
4. เลือก EtherChannel ที่คุณต้องการลบและกด Enter

การตั้งค่าหรือลบแบ็กอัพอะแด็ปเตอร์บน EtherChannel หรือ Link Aggregation ที่มีอยู่:

ทำตามโพรซีเดอร์ต่อไปนี้เพื่อตั้งค่าหรือลบแบ็กอัพอะแด็ปเตอร์บน EtherChannel หรือ Link Aggregation

1. พิมพ์ `smitty chinet` และเลือกอินเตอร์เฟสที่เป็นของ EtherChannel ของคุณ เปลี่ยนแอ็ตทริบิวต์ **Current STATE** เป็น **detach** และกด Enter
2. บนบรรทัดคำสั่ง พิมพ์ `smitty etherchannel`
3. เลือก **Change / Show Characteristics of an EtherChannel / Link Aggregation**
4. เลือก EtherChannel หรือ Link Aggregation ที่คุณกำลังเพิ่มหรือแก้ไขแบ็กอัพอะแด็ปเตอร์
5. ใส่อะแด็ปเตอร์ที่คุณต้องการใช้เป็นแบ็กอัพอะแด็ปเตอร์ในฟิลด์ **Backup Adapter** หรือเลือก **NONE** ถ้าคุณต้องการหยุดการใช้แบ็กอัพอะแด็ปเตอร์

การตั้งค่า IEEE 802.3ad Link Aggregation

IEEE 802.3ad เป็นวิธีมาตรฐานของการรวมลิงก์ ตามแนวคิด มันทำงานในแบบเดียวกับ EtherChannel ที่หลาย Ethernet อะแด็ปเตอร์ถูกรวมเข้ากับอะแด็ปเตอร์เสมือนเดียว เพื่อให้ได้แบนด์วิดธ์ที่กว้างและป้องกันความล้มเหลว

ตัวอย่างเช่น `ent0` และ `ent1` สามารถถูกรวมเข้ากับ IEEE 802.3ad Link Aggregation ที่เรียกว่า `ent3` จากนั้นอินเตอร์เฟส `ent3` จะถูกตั้งค่าด้วย IP แอดแดรส ระบบจะพิจารณาว่าอะแด็ปเตอร์ที่ถูกรวมเหล่านี้เป็นอะแด็ปเตอร์เดียว ดังนั้น IP จะถูกตั้งค่าบนมันเหมือนกับบน Ethernet อะแด็ปเตอร์ใดๆ

IEEE 802.3ad ต้องการการสนับสนุนในสวิตช์

ข้อดีของการใช้ IEEE 802.3ad Link Aggregation แทน EtherChannel คือคุณสามารถใช้สวิตช์ที่สนับสนุน มาตรฐาน IEEE 802.3ad แต่ไม่สนับสนุน EtherChannel และให้ การป้องกันความล้มเหลวของอะแด็ปเตอร์

เมื่อมีการกำหนดคอนฟิกการรวม IEEE 802.3ad link aggregation control protocol data units (LACPDUs) จะมีการแลกเปลี่ยน ระหว่างเครื่องเซิร์ฟเวอร์ (ระบบโฮสต์) และสวิตช์ที่ติดกัน เฉพาะแชนเนลที่แอ็คทีฟ ซึ่งสามารถเป็นแชนเนลหลักหรือแบ็คอัพอะแด็ปเตอร์ จะแลกเปลี่ยน LACPDU กับสวิตช์ที่ติดกัน

เพื่อให้สามารถรวมอะแด็ปเตอร์ (หมายความว่าสวิตช์ยอมให้ อะแด็ปเตอร์เป็นสมาชิกของการรวมเดียวกัน) อะแด็ปเตอร์ต้องมีความเร็ว เท่ากัน (ตัวอย่างเช่น 100 Mbps ทั้งหมด หรือ 1 Gbps ทั้งหมด) และอะแด็ปเตอร์ต้องเป็น full duplex ทั้งหมด ถ้าคุณพยายามใส่อะแด็ปเตอร์ที่มีความเร็วแตกต่าง หรือโหมด duplex ที่แตกต่าง การสร้างการรวม บนระบบ AIX จะสำเร็จ แต่สวิตช์จะไม่รวมอะแด็ปเตอร์เข้าด้วยกัน ถ้าสวิตช์ไม่สามารถรวมอะแด็ปเตอร์เข้าด้วยกัน คุณอาจจะสังเกตเห็นประสิทธิภาพของเน็ตเวิร์กที่ลดลง สำหรับข้อมูลเกี่ยวกับวิธีการกำหนด ว่าการรวมบนสวิตช์สำเร็จหรือไม่ โปรดดู “การแก้ปัญหา IEEE 802.3ad Link Aggregation” ในหน้า 413

ตามข้อมูลจำเพาะของ IEEE 802.3ad แพ็กเก็ตที่ส่ง ไปยัง IP แอดเดรสเดียวกันจะถูกส่งผ่านอะแด็ปเตอร์เดียวกัน ดังนั้น เมื่อทำงานในโหมด 802.3ad แพ็กเก็ต จะถูกกระจายในวิธีมาตรฐานเสมอ ไม่ใช่วิธี วนรอบ

คุณลักษณะของแบ็กอัพอะแด็ปเตอร์ที่มีสำหรับ IEEE 802.3ad Link Aggregationsจะเหมือนกับที่มีสำหรับ EtherChannel แบ็คอัพ อะแด็ปเตอร์ยังปฏิบัติตาม IEEE 802.3ad LACP ด้วย พอร์ตของสวิตช์ที่เชื่อมต่อ กับแบ็คอัพอะแด็ปเตอร์ยังต้องมีการเปิดใช้งาน IEEE 802.3ad ด้วย

หมายเหตุ: ขั้นตอนนี้จะเป็นการใช้ IEEE 802.3ad บนสวิตช์ที่ต่างกัน คุณต้องศึกษาเอกสารคู่มือสำหรับสวิตช์ของคุณเพื่อให้ทราบ ขั้นตอนเริ่มต้น ถ้ามี ที่ต้องทำเพื่อเปิดใช้งาน LACP ใน สวิตช์

สำหรับข้อมูลเกี่ยวกับวิธีการตั้งค่าการรวม IEEE 802.3ad ดูที่ “การกำหนดคอนฟิก IEEE 802.3ad Link Aggregation”

พิจารณาข้อมูลต่อไปนี้ก่อนคุณกำหนดคอนฟิก IEEE 802.3ad Link Aggregation:

- แม้ว่าไม่ได้รับการสนับสนุนอย่างเป็นทางการ การใช้ AIX ของ IEEE 802.3ad อนุญาตให้ การรวมลิงก์มีอะแด็ปเตอร์ที่มีความเร็วต่างกัน อย่างไรก็ตาม คุณต้องรวมเฉพาะอะแด็ปเตอร์ที่มีการตั้งค่าความเร็วเดียวกัน และตั้งค่าเป็น full duplex การทำเช่นนี้จะช่วยหลีกเลี่ยงปัญหาที่อาจเกิดขึ้นได้ในการกำหนดคอนฟิกการรวมลิงก์บนสวิตช์ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับชนิดของการรวมที่อนุญาตโดยสวิตช์ของคุณ โปรดดูเอกสารคู่มือ ของสวิตช์ของคุณ
- ถ้าคุณกำลังใช้อะแด็ปเตอร์อีเทอร์เน็ต 10/100 ในการรวมลิงก์ คุณต้องเปิดใช้งานการโพลลิงก์บนอะแด็ปเตอร์ดังกล่าวก่อนคุณเพิ่มอะแด็ปเตอร์ ลงในการรวม พิมพ์ smitty chgenet ที่บรรทัดรับคำสั่ง เปลี่ยนค่า **Enable Link Polling** เป็น yes และกด Enter ทำแอ็คชันนี้สำหรับ ทุกอะแด็ปเตอร์อีเทอร์เน็ต 10/100 ที่คุณกำลังเพิ่มลงในการรวมลิงก์

  หมายเหตุ: ใน AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03 และหลังจากนั้น ไม่มีความจำเป็นต้องเปิดใช้งานกลไก link polling โปรแกรมโพลลิงก์มีการเริ่มต้น โดยอัตโนมัติ

การกำหนดคอนฟิก IEEE 802.3ad Link Aggregation:

ปฏิบัติตามขั้นตอนต่อไปนี้เพื่อกำหนดคอนฟิก IEEE 802.3ad Link Aggregation

1. พิมพ์ smitty etherchannel บนบรรทัดรับคำสั่ง
2. เลือก **Add an EtherChannel / Link Aggregation** จากรายการและกด Enter
3. เลือกอะแด็ปเตอร์อีเทอร์เน็ตหลักที่คุณต้องการบน Link Aggregation ของคุณและกด Enter ถ้าคุณวางแผนที่จะใช้อะแด็ปเตอร์สำรอง อย่าเลือกอะแด็ปเตอร์ที่คุณวางแผนที่จะใช้สำหรับแบ็กอัพ ณ จุดนี้

   หมายเหตุ: **Available Network Adapters** จะแสดง Ethernet อะแด็ปเตอร์ทั้งหมด ถ้าคุณเลือก Ethernet อะแด็ปเตอร์ที่ถูกใช้อยู่ (มีการกำหนดอินเตอร์เฟส) คุณจะได้รับข้อความแสดงข้อผิดพลาด คุณต้องถอดอินเตอร์เฟสเหล่านี้ออกก่อน ถ้าคุณต้องการใช้มัน
4. ป้อนข้อมูลในฟิลด์ตามแนวทางต่อไปนี้ :

- **Parent Adapter:** ให้ข้อมูลเกี่ยวกับอุปกรณ์พาเรนต์ของ EtherChannel (ตัวอย่างเช่น เมื่อ EtherChannel เป็นของ Shared Ethernet Adapter) ฟิลด์นี้จะแสดงค่าของ NONE ถ้า EtherChannel ไม่ได้อยู่ในอะแด็ปเตอร์อื่น (โดยดีฟอลต์) ถ้า EtherChannel อยู่ในอะแด็ปเตอร์อื่น ฟิลด์นี้จะแสดงชื่อของพาเรนต์อะแด็ปเตอร์ (ตัวอย่างเช่น ent6) ฟิลด์นี้จะให้ข้อมูลเท่านั้นและไม่สามารถแก้ไขได้ อ็อพชัน parent adapter จะมีใน AIX 5.3 และหลังจากนั้น
- **EtherChannel / Link Aggregation Adapters:** คุณควรเห็นอะแด็ปเตอร์หลักทั้งหมดที่คุณใช้ใน Link ของคุณ คุณเลือกอะแด็ปเตอร์เหล่านี้ในขั้นตอนก่อนหน้านี้
- **Enable Alternate Address:** ฟิลด์นี้เป็นอ็อพชัน ตั้งค่าเป็น yes จะให้คุณสามารถระบุ MAC แอดเดรสที่คุณต้องการใช้ Link Aggregation ถ้าคุณตั้งค่าอ็อพชันนี้เป็น no, Link Aggregation จะใช้ MAC แอดเดรสของอะแด็ปเตอร์แรก
- **Alternate Address**: ถ้าคุณตั้ง **Enable Alternate Address** เป็น yes ระบุ MAC แอดเดรสที่คุณต้องการใช้ที่นี่ แอดเดรสที่คุณระบุต้องเริ่มต้นด้วย 0x และเป็นแอดเดรสเลขฐานสิบหก 12-หลัก (ตัวอย่างเช่น 0x001122334455)
- **Enable Gigabit Ethernet Jumbo Frames:** ฟิลด์นี้เป็นอ็อพชัน เพื่อที่จะใช้อ็อพชันนี้ การสวิตช์ของคุณต้องสนับสนุนเฟรมขนาดใหญ่ นี่จะใช้ได้กับอินเตอร์เฟส Standard Ethernet (en) เท่านั้น ไม่ใช่กับอินเตอร์เฟส IEEE 802.3 (et) ตั้งค่านี้เป็น yes ถ้าคุณต้องการใช้มัน
- **Mode:** ป้อน 8023ad
- **Hash Mode**: คุณสามารถเลือกจากโหมด hash ต่อไปนี้ ซึ่งจะกำหนดว่าค่าของข้อมูลที่จะถูกใช้โดยอัลกอริทึมที่กำหนดอะแด็ปเตอร์ขาออก:
  - **default**: ในโหมด hash นี้ IP แอดเดรสปลายทาง ของแพ็กเก็ตจะถูกใช้เพื่อกำหนดอะแด็ปเตอร์ขาออก สำหรับทราฟิกที่ไม่ใช่ IP (เช่น ARP) ไบต์สุดท้ายของ MAC แอดเดรสปลายทางถูกใช้เพื่อทำการคำนวณ โหมดนี้จะรับประกันว่าแพ็กเก็ตจะถูกส่งออกไปบน EtherChannel ตามลำดับที่มันถูกได้รับมา แต่แบนด์วิดธ์อาจจะไม่ถูกใช้อย่างเต็มที่
  - **src_port**: UDP ต้นทางหรือค่าของพอร์ต TCP ของแพ็กเก็ตที่จะถูกใช้เพื่อกำหนดอะแด็ปเตอร์ขาออก ถ้าแพ็กเก็ตไม่ใช่ UDP หรือทราฟิก TCP, ไบต์สุดท้ายของ IP แอดเดรสปลายทางจะถูกใช้ ถ้าแพ็กเก็ตไม่ใช่ทราฟิก IP ไบต์สุดท้ายของ MAC แอดเดรสปลายทางจะถูกใช้
  - **dst_port**: UDP ปลายทางหรือค่าของพอร์ต TCP ของแพ็กเก็ตที่จะถูกใช้เพื่อกำหนดอะแด็ปเตอร์ขาออก ถ้าแพ็กเก็ตไม่ใช่ UDP หรือทราฟิก TCP ไบต์สุดท้ายของ IP จะถูกใช้ ถ้าแพ็กเก็ตไม่ใช่ทราฟิก IP ไบต์สุดท้ายของ MAC แอดเดรสปลายทางจะถูกใช้
  - **src_dst_port**: UDP ต้นทางหรือปลายทาง หรือค่าของพอร์ต TCP ของแพ็กเก็ตที่จะถูกใช้เพื่อกำหนดอะแด็ปเตอร์ขาออก (โดยเฉพาะอย่างยิ่ง พอร์ตต้นทางและปลายทางจะถูกเพิ่ม และจากนั้นจะถูกหารด้วยสองก่อนที่จะถูกป้อนให้กับอัลกอริทึม) ถ้าแพ็กเก็ตไม่ใช่ UDP หรือทราฟิก TCP ไบต์สุดท้ายของ IP จะถูกใช้ ถ้าแพ็กเก็ตไม่ใช่ทราฟิก IP ไบต์สุดท้ายของ MAC แอดเดรสปลายทางจะถูกใช้ในโหมดนี้สามารถให้การกระจายแพ็กเก็ตที่ดีในหลายสถานการณ์ ทั้งสำหรับไคลเอ็นต์และเซิร์ฟเวอร์

  เพื่อเรียนรู้เพิ่มเติมเกี่ยวกับการกระจายแพ็กเก็ตและ load balancing ดูที่ “อ็อพชัน EtherChannel load-balancing” ในหน้า 404
- **Backup Adapter:** ฟิลด์นี้เป็นอ็อพชัน ป้อนอะแด็ปเตอร์ ที่คุณต้องการใช้เป็นแบ็กอัพ
- **Internet Address to Ping:** ฟิลด์นี้เป็นอ็อพชัน และพร้อมใช้งานเมื่อคุณมีอะแด็ปเตอร์หนึ่งอะแด็ปเตอร์หรือมากกว่าใน aggregation หลัก และอะแด็ปเตอร์แบ็กอัพ Link Aggregation จะ ping IP แอดเดรสหรือชื่อโฮสต์ ที่คุณระบุที่นี่ ถ้า Link Aggregation ไม่สามารถ ping แอดเดรสนี้ ตามจำนวนครั้งที่ระบุโดยฟิลด์ **Number of Retries** และตามช่วงเวลาที่ระบุโดยฟิลด์ **Retry Timeout** Link Aggregation จะเปลี่ยนอะแด็ปเตอร์

- **Number of Retries:** ป้อนจำนวนความล้มเหลว ในการตอบสนอง ping ที่อนุญาตก่อนเปลี่ยนอะแด็ปเตอร์ Link Aggregation ค่าดีฟอลต์คือ 3 ฟิลด์นี้เป็นอ็อพชันและจะใช้ได้ถ้าคุณตั้ง **Internet Address to Ping**
- **Retry Timeout:** ป้อนจำนวนวินาทีระหว่าง เวลาที่ Link Aggregation จะ ping **Internet Address to Ping** ค่าดีฟอลต์คือ 1 วินาที ฟิลด์นี้เป็นอ็อพชันและจะใช้ได้ถ้าคุณตั้ง **Internet Address to Ping**

5. กด Enter หลังการเปลี่ยนฟิลด์ที่ต้องการเพื่อสร้าง Link Aggregation
6. กำหนดคอนฟิก IP บนอุปกรณ์ Link Aggregation ที่ถูกสร้างใหม่ โดยพิมพ์ `smitty chinet` ที่บรรทัดรับคำสั่ง
7. เลือกอินเตอร์เฟส Link Aggregation ใหม่จากรายการ
8. กรอกฟิลด์ที่จำเป็นทั้งหมด และกด Enter

การแก้ปัญหา IEEE 802.3ad Link Aggregation:

ใช้คำสั่ง **entstat** เพื่อแก้ไขปัญหา IEEE 802.3ad Link Aggregation

ถ้าคุณมีปัญหากับ IEEE 802.3ad Link Aggregation ของคุณ ใช้คำสั่งต่อไปนี้เพื่อตรวจสอบการทำงานของ Link Aggregation:

```
entstat -d device
```

โดยที่ *device* เป็นอุปกรณ์ Link Aggregation

นี่ยังจะเป็นความพยายามที่ดีที่สุดที่จะกำหนดสถานะของความคืบหน้าของ LACP โดยขึ้นอยู่กับ LACPDU ที่ได้รับจากสวิตช์ ต่อไปนี้เป็นค่าที่เป็นไปได้ของสถานะ :

- `Inactive`: LACP ยังไม่ถูกเริ่มต้น นี่เป็นสถานะเมื่อ Link Aggregation ยังไม่ถูกตั้งค่า เนื่องจากมันยังไม่ถูกกำหนด IP แอดเดรส หรือเนื่องจากอินเตอร์เฟสของมันถูกถอดออก
- `Negotiating`: LACP กำลังดำเนินการ แต่สวิตช์ยังไม่รวมอะแด็ปเตอร์ ถ้า Link Aggregation ยังคงเป็นสถานะนี้นานกว่าหนึ่งนาที ตรวจสอบว่าสวิตช์ถูกตั้งค่าอย่างถูกต้องหรือไม่ ตัวอย่างเช่น คุณควรตรวจสอบว่า LACP ถูกเปิดใช้งานบนพอร์ต
- `Aggregated`: LACP ทำเสร็จแล้วและสวิตช์รวมอะแด็ปเตอร์เข้าด้วยกันแล้ว
- `Failed`: LACP ล้มเหลว สาเหตุที่เป็นไปได้บางอย่างคือ อะแด็ปเตอร์ในการรวมกันถูกตั้งความเร็วสาย หรือโหมดของ duplex ที่แตกต่างกัน หรือมันถูกเสียบเข้ากับสวิตช์อื่น ตรวจสอบการตั้งค่าของอะแด็ปเตอร์
  นอกจากนี้บางสวิตช์ยอมให้เฉพาะพอร์ตที่อยู่ติดกันที่จะรวมกันได้ และอาจมีข้อจำกัดเกี่ยวกับจำนวนของอะแด็ปเตอร์ที่สามารถถูกรวม ศึกษาเอกสารของสวิตช์เพื่อดูว่ามีข้อจำกัดที่สวิตช์อาจมี จากนั้นตรวจสอบการตั้งค่าสวิตช์

หมายเหตุ: สถานะของ Link Aggregation เป็นค่าการวินิจฉัยและจะไม่มีผลกับด้าน AIX ของการตั้งค่า ค่าของสถานะนี้ถูกได้มาจากความพยายามที่ดีที่สุด เพื่อดีบักปัญหาที่เกี่ยวกับการรวมกัน จะดีที่สุดที่จะตรวจสอบการตั้งค่าของสวิตช์

สถิติของ IEEE 802.3ad Link Aggregation ต่อไปนี้แสดงสถานะของ LACP บนแต่ละพอร์ตของการรวมกัน

มันจะถูกแสดงสำหรับทั้ง Actor (the IEEE 802.3ad Link Aggregation) และสำหรับ Partner (พอร์ตของสวิตช์)

```
System Priority: ค่าลำดับความสำคัญของระบบนี้
System: ค่าที่จะระบุระบบนี้โดยเฉพาะ
Operational Key: ค่าที่จะแสดงว่าพอร์ตใดที่อาจจะ
        ถูกรวมเข้าด้วยกัน
Port Priority: ค่าลำดับความสำคัญของพอร์ตนี้
```

Port: ค่าที่เป็นหนึ่งเดียวที่ระบุว่าพอร์ตนี้อยู่ในสถานะการรวมกัน
:

LACP activity: Active หรือ Passive - ว่าถูกเริ่มต้น
ส่ง LACPDUs เสมอ หรือเฉพาะเมื่อตอบสนองกับ
LACPDU อื่น : IEEE 802.3ad Link Aggregation
จะทำงานในโหมด Active เสมอ

LACP timeout: ยาว หรือ สั้น - เวลาที่รอก่อนที่
จะส่ง LACPDUs: the IEEE 802.3ad Link Aggregation
จะใช้ timeout แบบนาน

Aggregation: Individual หรือ Aggregatable - พอร์ตนี้
สามารถรวมกับพอร์ตอื่นหรือไม่ หรือ
มันสามารถรวมกับตัวมันเองเท่านั้น :
พอร์ตในหนึ่งอะแด็ปเตอร์ IEEE 802.3ad Link Aggregation
จะถูกกำหนดเป็น Individual หรือ Aggregatable
ถ้ามีมากกว่าหนึ่งพอร์ต

Synchronization: IN_SYNC or OUT_OF_SYNC - การรวมกัน
ถูกกำหนดว่ามันเข้าถึง
การซิงโครไนซ์กับพาเรนต์

Collecting: Enabled หรือ Disabled - IEEE 802.3ad
Link Aggregation กำลังรวมรวม (ได้รับ) แพ็กเก็ต

Distributing: Enabled หรือ Disabled - IEEE 802.3ad
Link Aggregation กำลังกระจาย (ส่ง) แพ็กเก็ต

Defaulted: True หรือ False - IEEE 802.3ad
Link Aggregation ใช้ค่าดีฟอลต์สำหรับ
ข้อมูลของพาร์ตเนอร์

Expired: True หรือ False - IEEE 802.3ad
Link Aggregation กำลังทำงานในโหมด ที่หมดอายุแล้ว

สถิติต่อไปนี้ถูกแสดงบนทั้งบนพื้นฐานแบบพอร์ตต่อพอร์ต หรือการรวม:

Received LACPDUs: ได้รับ LACPDU แพ็กเก็ต
Transmitted LACPDUs: แพ็กเก็ต LACPDU ถูกส่งออกไป
Received marker PDUs: ได้รับ marker PDU
Transmitted marker PDUs: marker PDU ถูกส่ง:
โปรโตคอลเวอร์ชันนี้ไม่ได้ใช้
โปรโตคอล marker ดังนั้นสถิตินี้จะเป็นศูนย์เสมอ
Received marker response PDUs: ได้รับ marker PDU
Transmitted marker response PDUs: ส่ง marker response PDU:
โปรโตคอลเวอร์ชันนี้ไม่ได้ใช้
โปรโตคอล marker ดังนั้นสถิตินี้จะเป็นศูนย์เสมอ
Received unknown PDUs: ได้รับ PDUs ชนิดที่ไม่รู้จัก
Received illegal PDUs: ได้รับ PDUs ชนิดที่ไม่รู้จักแต่
ถูก malformed ความยาวที่ไม่คาดคิด หรือชนิดย่อยที่ไม่รู้จัก

สถานการณ์จำลองความสามารถในการทำงานข้ามระบบ

พิจารณาสถานการณ์จำลองความสามารถในการทำงานข้ามระบบต่อไปนี้เมื่อตั้งค่า EtherChannel หรือ E 802.3ad Link Aggregation ของคุณ

คำอธิบายเพิ่มเติมของแต่ละสถานการณ์จำลองถูกอธิบายหลังจากตาราง

ตารางที่ 83. การรวมการตั้งค่า AIX และสวิตช์ที่แตกต่างกันและผลที่แต่ละการรวมกันจะสร้างขึ้น

โหมด EtherChannel	การตั้งค่าสวิตช์	ผลลัพธ์
8023ad	IEEE 802.3ad LACP	OK – AIX จะเริ่มต้น LACPDUs ซึ่งจะทริกเกอร์ IEEE 802.3ad Link Aggregation บนสวิตช์
มาตรฐาน หรือ round_robin	EtherChannel	OK – ผลลัพธ์ในพฤติกรรมของ EtherChannel แบบดั้งเดิม
8023ad	EtherChannel	ไม่ต้องการ – AIX และ สวิตช์ไม่สามารถรวม AIX เริ่มต้น LACPDUs แต่สวิตช์ละเว้นและไม่ส่ง LACPDUs ไปยัง AIX เนื่องจาก LACPDU ไม่มีอยู่ AIX จึงไม่ได้แจกจ่ายแพ็กเก็ต บนลิงก์/พอร์ต ผลลัพธ์คือสูญเสียภาวะเชื่อมต่อเครือข่าย
มาตรฐาน หรือ round_robin	IEEE 802.3ad LACP	ไม่สามารถตัดสินใจ – สวิตช์ไม่สามารถรวม ผลอาจเกิดจากประสิทธิภาพที่ต่ำเมื่อสวิตช์ย้าย MAC แอดเดรสระหว่างพอร์ตของสวิตช์

คำอธิบายแบบสั้นของแต่ละการรวมกันของการตั้งค่า เป็นดังต่อไปนี้:

- 8023ad พร้อมด้วย EtherChannel:
  ในกรณีนี้ AIX จะส่ง LACPDU แต่มันจำไม่ได้รับการตอบเนื่องจากสวิตช์ทำงานเป็น EtherChannel ตามผลลัพธ์ที่ได้ เนื่องจาก LACPDU ไม่มีอยู่ AIX จึงไม่ใช้ลิงก์/พอร์ต สำหรับการแจกจ่ายแพ็กเก็ต ซึ่งเป็นสาเหตุทำให้สูญเสียภาวะเชื่อมต่อเครือข่าย

  หมายเหตุ: ในกรณีนี้ คำสั่ง `entstat -d` จะรายงานการรวมที่อยู่ในสถานะ การเจรจา เสมอ และ ในเอาต์พุต entstat ส่วนของ IEEE 802.3ad Port Statistics จะแสดง **การแจกจ่าย** ที่ปิดใช้งาน **Actor**
- มาตรฐาน หรือ round_robin พร้อมด้วย EtherChannel:
  นี่เป็นการตั้งค่า EtherChannel แบบทั่วไปที่สุด
- มาตรฐาน หรือ round_robin พร้อมด้วย IEEE 802.3ad LACP:
  การตั้งค่านี้ใช้ไม่ได้ ถ้าสวิตช์ใช้ LACP เพื่อสร้างการรวมกัน การรวมกันจะไม่เคยเกิดขึ้นเนื่องจาก AIX จะไม่ตอบกับ LACPDUs เพื่อให้ทำงานอย่างถูกต้อง 8023ad ควรถูกตั้งโหมดเป็น AIX

อะแด็ปเตอร์ที่ได้รับการสนับสนุน

EtherChannel และ IEEE 802.3ad Link Aggregation ได้รับการสนับสนุนบนอะแด็ปเตอร์ IBM Power Systems™ Peripheral Component Interconnect-X (PCI-X) และ PCI Express (PCIe) Ethernet

ข้อควรพิจารณาเพิ่มเติมมีดังต่อไปนี้:

- Virtual I/O Ethernet Adapter

Virtual I/O Ethernet Adapters ได้รับการสนับสนุนในเฉพาะ 2 การตั้งค่า EtherChannel ที่เป็นไปได้:

- หนึ่ง Virtual I/O Ethernet Adapter เป็นหลัก หนึ่ง Virtual I/O Ethernet Adapter ไว้สำรอง ในการตั้งค่านี้ แอ็ตทริบิวต์ **Internet Address to Ping** ต้องถูกเปิดใช้งานเพื่อที่ EtherChannel สามารถตรวจจับความล้มเหลวของการเชื่อมต่อแบบรีโมต Virtual I/O Ethernet Adapter แต่ละตัวต้องถูกตั้งค่าด้วย Port VLAN ID (PVID) ที่แตกต่างกัน นอกจากนี้ แต่ละตัวต้องถูกตั้งค่าเพื่อที่จะถูกบริดจ์โดย Virtual I/O Servers (VIOSs) ที่ต่างกัน
- หนึ่งฟิสิคัล Ethernet อะแด็ปเตอร์ที่ได้รับการสนับสนุนเป็นหลัก หนึ่ง Virtual I/O Ethernet Adapter ไว้สำรอง ในการตั้งค่านี้ แอ็ตทริบิวต์ **Internet Address to Ping** ต้องถูกเปิดใช้งานเพื่อที่ EtherChannel สามารถตรวจจับความล้มเหลวของการเชื่อมต่อแบบรีโมต

- Host Ethernet Adapter (HEA)

  โลจิคัลพอร์ตของ HEA ได้รับการสนับสนุนภายใต้ EtherChannel ถ้าอะแด็ปเตอร์ทั้งหมดภายใน EtherChannel เป็นโลจิคัลพอร์ตของ HEA สำหรับพอร์ตเฉพาะของ HEA การรวมลิงก์กับ PCI/PCI-E อะแด็ปเตอร์ได้รับการสนับสนุน นอกจากนี้ PCI/PCI-E และอะแด็ปเตอร์อีเทอร์เน็ตเสมือนเป็นอะแด็ปเตอร์สำรอง ได้รับการสนับสนุน (เมื่ออะแด็ปเตอร์หลักมี HEA)

  เมื่อใช้หลายโลจิคัลพอร์ตของ HEA เป็นอะแด็ปเตอร์หลักใน EtherChannel ฟิสิคัลพอร์ตที่เชื่อมโยงกับโลจิคัลพอร์ตของ HEA ต้องถูกวางใน EtherChannel ในสวิตช์ Ethernet ดังนั้น พาร์ติชันทั้งหมดที่ใช้โลจิคัลพอร์ตของ HEA ที่ไปยังฟิสิคัลพอร์ตเดี่ยวกันของ HEA ยังต้องถูกวางใน EtherChannel

  ตัวอย่างเช่น สมมุติว่า Partition 1 ถูกตั้งค่าดังต่อไปนี้ :

  - โลจิคัลพอร์ตของ HEA จากฟิสิคัลพอร์ต 0 ของ HEA
  - โลจิคัลพอร์ตของ HEA จากฟิสิคัลพอร์ต 1 ของ HEA
  - EtherChannel จะถูกสร้างโดยใช้โลจิคัลพอร์ตของ HEA ที่ถูกลิสต์ข้างบน

  ถ้าพาร์ติชันอื่นบนระบบเดียวกันต้องการใช้โลจิคัลพอร์ตของ HEA จากฟิสิคัลพอร์ต 0 ของ HEA หรือจาก ฟิสิคัลพอร์ต 1 ของ HEA คุณต้องสร้าง EtherChannel สำหรับพาร์ติชันบนโลจิคัลพอร์ตทั้งสองของ HEA เหมือนกับการตั้งค่าของ Partition 1 การพยายามใช้โลจิคัลพอร์ตของ HEA เหล่านั้นเป็นพอร์ตแบบ stand-alone ในพาร์ติชันอื่นอาจทำให้มีปัญหาเกี่ยวกับการเชื่อมต่อ เนื่องจากแพ็กเก็ตอาจไม่สามารถถูกส่งไปถึงโลจิคัลพอร์ตของ HEA ที่ถูกต้อง

  ไม่มีข้อจำกัดเมื่อใช้พอร์ต HEA โลจิคัลในการกำหนดคอนฟิกสำรองอินเตอร์เฟสเครือข่าย (1 ตัวหลัก และ 1 ตัวสำรอง) เนื่องจากพอร์ต HEA ฟิสิคัลไม่ต้องการการกำหนดคอนฟิกเฉพาะ บนสวิตช์ Ethernet

  หมายเหตุ: ถ้าโลจิคัลพอร์ตจากฟิสิคัลพอร์ตของ HEA ถูกตั้งค่าเป็นส่วนของการรวม LACP (802.3ad) ดังนั้น ฟิสิคัลพอร์ตเหล่านั้นต้องเป็นพิเศษสำหรับ LPAR HMC ไม่ได้ป้องกันพอร์ตจากการถูกกำหนดให้กับ LPAR อื่น แต่ไม่สนับสนุนการกระทำดังกล่าว

- Fibre Channel บน Ethernet เหมือนกับเน็ตเวิร์กอะแด็ปเตอร์

  การรวมลิงก์ระหว่าง พอร์ตที่แบ่งใช้ (พอร์ตที่ใช้สำหรับทั้งทราฟิก Ethernet และ Fiber Channel) และอะแด็ปเตอร์ที่สนับสนุนอื่นๆ ได้รับการสนับสนุนหากสวิตช์ที่เชื่อมต่อกับพอร์ตที่แบ่งใช้สามารถสนับสนุนการรวมลิงก์โดยไม่กระทบกับทราฟิก Fiber Channel

- อะแด็ปเตอร์ Single Root I/O Virtualization (SR-IOV)

  โลจิคัลพอร์ต SR-IOV แบ่งใช้ข้อจำกัดเดียวกันกับโลจิคัลพอร์ต HEA ถ้าโลจิคัลพอร์ต SR-IOV ถูกใช้ภายใต้ EtherChannel อะแด็ปเตอร์ทั้งหมดใน Etherchannel ต้องเป็นโลจิคัลพอร์ต SR-IOV ยิ่งไปกว่านั้น หาก EtherChannel รวมโลจิคัลพอร์ต SR-IOV สองตัวไว้ (เตรียมไว้จากฟิสิคัลพอร์ต SR-IOV สองพอร์ต) คอนฟิกูเรชันนี้ จะรวมฟิสิคัลพอร์ตสองพอร์ตได้อย่างมีประสิทธิผล

ดังนั้น โลจิคัลพอร์ตอื่นใดที่สร้างจาก ฟิสิคัลพอร์ตสองพอร์ตต้องถูกรวมไว้ด้วยวิธีเดียวกับโลจิคัลพอร์ตสองพอร์ตแรก
เนื่องจากข้อจำกัดการใช้งานนี้ โลจิคัลพอร์ต EtherChannel บน SR-IOV จะถูกกีดกันไม่ให้ใช้ ยกเว้นว่า โลจิคัลพอร์ตที่ถูกรวมไว้จะได้รับสิทธิ์การใช้งาน 100% ของความสามารถของฟิสิคัลพอร์ต SR-IOV ตามลำดับ

สำหรับข้อมูลรีลีสเพิ่มเติมเกี่ยวกับอะแด็ปเตอร์ใหม่ ดูที่ AIX Release Notes ที่สอดคล้องกับระดับของ AIX ของคุณ

ข้อสำคัญ: การผสมอะแด็ปเตอร์ที่มีความเร็วต่างกันใน EtherChannel เดียวกันไม่ได้รับการสนับสนุน แม้ว่าหนึ่งในนั้นจำทำงานเป็นแบ็กอัพ ซึ่ง*ไม่*ได้หมายความว่าการกระทำดังกล่าวจะใช้ไม่ได้ ไดรเวอร์ของ EtherChannel ทำให้ความพยายามทุกความพยายามที่มีเหตุผลสมารถทำงานได้ แม้ในสถานการณ์จำลองที่มีความเร็วผสมกัน

การแก้ปัญหา EtherChannel

ถ้าคุณมีปัญหากับ EtherChannel ของคุณ จะมีหลายสถานการณ์จำลองที่จะพิจารณา

คุณสามารถใช้การติดตามและสถิติเพื่อช่วยในการวินิจฉัยปัญหา ซึ่งสามารถเกี่ยวข้องกับปัญหากับ failover และ jumbo frames

การติดตาม EtherChannel:

ใช้ **tcpdump** และ **iptrace** เพื่อแก้ไขปัญหา EtherChannel.

trace hook ID สำหรับการส่งแพ็กเก็ตคือ 2FA และเหตุการณ์อื่นคือ 2FB คุณไม่สามารถติดตามแพ็กเก็ตที่ได้รับบน EtherChannel ทั้งหมด แต่คุณสามารถติดตามแต่ละ trace hooks ที่ได้รับของอะแด็ปเตอร์

สถิติ EtherChannel:

ใช้คำสั่ง **entstat** เพื่อให้ได้การรวบรวมสถิติของอะแด็ปเตอร์ทั้งหมดใน EtherChannel

ตัวอย่างเช่น `entstat ent3` จะแสดงการรวบรวมสถิติของ ent3 การเพิ่มแฟล็ก **-d** ยังจะแสดงสถิติของแต่ละอะแด็ปเตอร์ ตัวอย่างเช่น การพิมพ์ `entstat -d ent3` จะแสดงการรวบรวมสถิติของ EtherChannel พร้อมด้วยสถิติของแต่ละอะแด็ปเตอร์ใน EtherChannel

หมายเหตุ: ในส่วน `General Statistics` ตัวเลขที่แสดงใน `Adapter Reset Count` คือจำนวนของ failovers ใน EtherChannel แบ็กอัพ การกลับไปใช้ EtherChannel หลักจากแบ็กอัพอะแด็ปเตอร์ จะไม่ถูกนับเป็น failover เฉพาะการ failover จากแชนแนลหลักไปยังแบ็กอัพที่จะถูกนับ

ในฟิลด์ `Number of Adapters` แบ็กอัพอะแด็ปเตอร์จะถูกนับในจำนวนที่ถูกแสดง

Slow failover:

ถ้าเวลาของการ failover เมื่อคุณใช้โหมด network interface backup หรือ EtherChannel แบ็กอัพมีความช้า ตรวจสอบว่าสวิตช์ของคุณไม่ได้ใช้ Spanning Tree Protocol (STP)

เมื่อสวิตช์ตรวจพบการเปลี่ยนแปลงในการแม็พพอร์ตของสวิตช์กับ MAC แอดเดรส มันจะรันอัลกอริทึม spanning tree เพื่อดูว่ามี loop ในเน็ตเวิร์กหรือไม่ Network Interface Backup และ EtherChannel แบ็กอัพอาจเป็นสาเหตุของการเปลี่ยนการแม็พพอร์ตกับ MAC แอดเดรส

พอร์ตของสวิตช์จะมีตัวนับการหน่วงเวลาการฟอร์เวิร์ดที่ระบุว่าเร็วแค่ไหนหลังจากการเริ่มต้นที่แต่ละพอร์ตควรเริ่มการฟอร์เวิร์ดหรือส่งแพ็กเก็ต สำหรับสาเหตุนี้ เมื่อแชนแนลหลักถูกเปิดใช้งานอีกครั้ง จะมีการหน่วงเวลาก่อนที่การเชื่อมต่อจะเกิดขึ้น ที่การ failover ไปยังแบ็กอัพอะแด็ปเตอร์จะเร็วกว่า ตรวจสอบตัวนับการหน่วงเวลาการฟอร์เวิร์ดบนสวิตช์ของคุณและทำให้มันมีค่าน้อยเท่าที่จะเป็นไปได้เพื่อที่การกลับไปใช้แชนแนลหลักจะเกิดขึ้นเร็วที่สุด

เพื่อให้การทำงานของ EtherChannel แบ็กอัพเป็นไปอย่างถูกต้อง ตัวนับการหน่วงเวลาการฟอร์เวิร์ดต้องไม่มากกว่า 10 วินาที ถ้าไม่การกลับไปใช้ EtherChannel หลักอาจทำงานไม่ถูกต้อง แนะนำให้ตั้งค่าตัวนับการหน่วงเวลาการฟอร์เวิร์ดเป็นค่าน้อยที่สุดที่สวิตช์ยอมรับได้

อะแด็ปเตอร์ไม่ fail over:

ถ้าการที่อะแด็ปเตอร์ล้มเหลวไม่ทริกเกอร์ failover และคุณรัน AIX 5.2 ที่มี 5200-01 หรือก่อนหน้านี้ ตรวจสอบเพื่อดูว่าอะแด็ปเตอร์การ์ดของคุณต้องเปิดใช้งาน link polling เพื่อตรวจจับความล้มเหลวของลิงก์หรือไม่

บางอะแด็ปเตอร์ไม่สามารถตรวจจับสถานะของลิงก์ของมันโดยอัตโนมัติ เพื่อตรวจจับเงื่อนไขนี้ อะแด็ปเตอร์เหล่านี้ต้องเปิดใช้งานกลไก link polling ที่จะสตาร์ทตัวจับเวลาที่จะตรวจสอบสถานะลองลิงก์เป็นช่วงเวลา Link polling จะถูกปิดการใช้งานโดยดีฟอลต์ อย่างไรก็ตาม เพื่อให้ EtherChannel ทำงานอย่างถูกต้องกับอะแด็ปเตอร์เหล่านี้ กลไก link polling ต้องถูกเปิดใช้งานบนแต่ละอะแด็ปเตอร์ก่อนที่ EtherChannel จะถูกสร้าง ถ้าคุณรัน AIX 5L เวอร์ชัน 5.2 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5200-03 หลังหลังจากนั้น link polling จะถูกสตาร์ทโดยอัตโนมัติและนี่ไม่น่าจะเป็นปัญหา

อะแด็ปเตอร์ที่มีกลไก link polling จะมีแอ็ตทริบิวต์ ODM ที่เรียกว่า **poll_link** ซึ่งต้องถูกตั้งเป็น yes สำหรับการเปิดใช้งาน link polling ก่อนที่จะสร้าง EtherChannel ใช้คำสั่งต่อไปนี้บนทุกอะแด็ปเตอร์เพื่อที่จะถูกรวมใน :

```
smitty chgenet
```

เปลี่ยนค่า **Enable Link Polling** เป็น yes และกด Enter

Jumbo frames:

นอกเหนือจากการเปิดใช้งานแอ็ตทริบิวต์ **use_jumbo_frame** บน EtherChannel แล้ว คุณยังต้องเปิดใช้งานเฟรม jumbo บนแต่ละอะแด็ปเตอร์ ก่อนจะสร้าง EtherChannel ด้วย

ในการทำดังกล่าว รันคำสั่งต่อไปนี้ :

```
smitty chgenet
```

เฟรม jumbo มีการเปิดใช้งานโดยอัตโนมัติในทุกอะแด็ปเตอร์ที่สำคัญ เมื่อแอ็ตทริบิวต์ **use_jumbo_frame** ของ EtherChannel มีการตั้งค่าเป็น ใช่

Remote dump:

Remote dump ไม่ได้รับการสนับสนุนบน EtherChannel

อินเตอร์เน็ตโปรโตคอลบน InfiniBand (IPoIB)

Internet protocol (IP) อินเตอร์เน็ตโปรโตคอลสามารถถูกส่งบน InfiniBand (IB) อินเตอร์เฟส การขนส่งนี้สามารถทำโดยการ encapsulate IP แพ็กเก็ตของ IB แพ็กเก็ตโดยใช้เน็ตเวิร์กอินเตอร์เฟส

เพื่อที่จะใช้ IP บน IB คุณต้องติดตั้งและตั้งค่าไดรเวอร์ InfiniBand connection manager (ICM) และมีอย่างน้อยหนึ่งอุปกรณ์ IB ในระบบ เพื่อดูว่าอุปกรณ์ IB ถูกติดตั้งแล้วหรือไม่ ใช้คำสั่ง **lsdev -C | grep iba** ชื่อของ fileset ที่ประกอบด้วยอินเตอร์เฟส IB คือ : `devices.common.IBM.ib` `devices.chrp.IBM.lhca` fileset เป็นตัวอย่างของ fileset ของอะแด็ปเตอร์ที่ได้รับการสนับสนุนในปัจจุบัน

เพื่อตั้งค่าไดรเวอร์ ICM อ้างถึง “การตั้งค่าไดรเวอร์ InfiniBand Communication Manager” ในหน้า 421

เพื่อที่จะสร้าง InfiniBand interface (IB IF) IB IF ต้องสามารถถูกรวมเข้ากับกลุ่มบรอดคาสต์-มัลติคาสต์ ด้วย PKEY ที่ผู้ใช้จัดเตรียม (หรือดีฟอลต์ PKEY = 0xFFFF ถูกใช้ถ้าผู้ใช้ไม่ได้จัดเตรียม) และ Q_Key ที่ผู้ใช้จัดเตรียม (หรือดีฟอลต์ Q_Key = 0x1E ถูกใช้ถ้าผู้ใช้ไม่ได้จัดเตรียม) กลุ่มบรอดคาสต์-มัลติคาสต์เป็นกลุ่มมัลติคาสต์ที่อินเตอร์เฟสต้องรวมด้วยเพื่อที่จะส่งบรอดคาสต์ และ **ARP** แพ็กเก็ต ถ้ากลุ่มบรอดคาสต์-มัลติคาสต์นั้นไม่มีอยู่ หรือไม่สามารถถูกสร้างโดยอินเตอร์เฟส การสร้าง IB IF จะล้มเหลว

คุณสามารถสร้างหรือเปลี่ยน IB IF โดยใช้อินเตอร์เฟสบรรทัดรับคำสั่ง หรือส่วนติดต่อผู้ใช้ของ SMIT พารามิเตอร์ที่ต้องการสำหรับสร้าง IB IF มีดังต่อไปนี้ :

- *ชื่ออินเตอร์เฟส*
- *ชื่ออะแด็ปเตอร์*
- *หมายเลขพอร์ต*
- *อินเตอร์เฟส IP แอดเดรส*

พารามิเตอร์ต่อไปนี้สำหรับการเปลี่ยนแปลง IB IF:

- *อินเตอร์เน็ตแอดเดรส*
- *เน็ตเวิร์กมาสก์*
- *ขนาดของ MTU* (เท่ากับ MTU ที่ต้องการ น้อยกว่า 4 ไบต์สำหรับส่วนหัวของ IB)
- *สถานะ*
- *ขนาดของคิวการส่งและรับ (ดีฟอลต์คือ 4000)*
- *Multicast Queue Key*
- *ซุปเปอร์แพ็กเก็ต เปิด หรือ ปิด*

ต่อไปนี้เป็นตัวอย่างของคำสั่งที่ใช้เพื่อสร้าง IB IF จากบรรทัดรับคำสั่ง:

```
$ /usr/sbin/mkiba -i ib0 -p 1 -A iba0 -a 1.2.3.8 [-P -1 -S "up" -m "255.255.254.0" -M 2044]
```

โดยที่:

ไอเท็ม	คำอธิบาย
`-M 2044`	หน่วยการส่งผ่านสูงสุด
`-m "255.255.254.0"`	เน็ตมาสก์
`-p 1`	หมายเลขพอร์ต (ค่าดีฟอลต์ *1* ถ้าไม่ถูกกำหนด)
`-A iba0`	ชื่ออุปกรณ์ IB
`-a 1.2.3.8`	IF IP แอดเดรส
`-i ib0`	ชื่ออินเตอร์เฟส
`-P -1`	คีย์พาร์ติชัน (ค่าดีฟอลต์คือ *PKEY* ถ้าไม่ถูกกำหนด หลังจากอินเตอร์เฟสถูกสร้าง PKEY จะไม่สามารถถูกเปลี่ยน ผู้ใช้ต้องได้รับ PKEY ที่ไม่ใช่ดีฟอลต์จากผู้บริหารเน็ตเวิร์ก)
`-S "up"`	สถานะของอินเตอร์เฟส
`-q 8000`	ขนาดของคิวการรับและส่ง (แต่ละคิว)

ไอเท็ม	คำอธิบาย
`-Q 0x1E`	คีย์ของดิวมัลติคาสต์ถูกกำหนดให้กับกลุ่มมัลติคาสต์ (ค่าดีฟอลต์คือ Q_KEY = 0x1E ถ้าไม่ถูกกำหนด)
`-k "on"`	ซุปเปอร์แพ็กเก็ตจะยอมให้ TCP/IP MTU ของอินเตอร์เฟสเป็น 64K มันต้องถูกเปิดใช้งานในรีโมตโฮสต์ด้วยเพื่อที่จะสามารถทำงานได้

ต่อไปนี้เป็นตัวอย่างของคำสั่งที่ใช้เพื่อสร้าง IB IF จากส่วนติดต่อผู้ใช้ของ SMIT :

```
$ smitty inet
```

หลังจากเมนู Network Interface Selection ถูกแสดง ทำตามโพรซีเดอร์ต่อไปนี้ :

1. เลือก **Add a Network Interface** หรือ **Change / Show Characteristics of a Network Interface.** เมนู Add a Network Interface จะถูกแสดง
2. ในเมนู Add a Network Interface เลือก **Add an IB Network Interface** เมนู Add an IB Network Interface จะถูกแสดง
3. ในเมนู Add an IB Network Interface ทำการเปลี่ยนแปลงที่จำเป็น และกด Enter

การสร้าง การแสดง การเพิ่ม การลบ entry ของ ARP และการแก้ไขตัวจับเวลา ARP

entry ของ **Address Resolution Protocol (ARP)** จะยอมให้อินเตอร์เฟสสื่อสารกับอินเตอร์เฟสอื่นแม้ว่ามันจะไม่ได้อยู่ในกลุ่มมัลติคาสต์เดียวกัน

entry ของ **ARP** สามารถถูกสร้างแบบแมนวลโดยใช้คำสั่ง **arp -t ib**

เพื่อแสดง entry ของ **ARP** ทั้งหมด รันคำสั่ง **$ arp -t ib -a** ถ้าคุณต้องการแสดงจำนวนของ entry ของ **ARP** ที่ระบุ คุณสามารถระบุจำนวน ตัวอย่างเช่น `$ arp -t ib -a 5` จะแสดง 5 entry ของ **ARP**

คำสั่งต่อไปนี้จะเพิ่ม entry ของ **ARP** :

```
$ arp -t ib -s IB interface name dlid <16 bits DLID> dqp
16 bits hex Destination Queue Pair Number
ipaddr <Destination IP Address>
```

โดยที่:

ไอเท็ม	คำอธิบาย
DLID	เป็นปลายทาง ID ของปลายทาง
DGID	เป็นโกลบอล ID ของปลายทาง

คำสั่งต่อไปนี้จะลบ entry ของ **ARP** :

```
$ arp -t ib -d IP Address
```

ต่อไปนี้จะแก้ไขค่าของตัวจับเวลาของ entry ของ ARP สำหรับ entry ของ ARP ที่สมบูรณ์ และไม่สมบูรณ์ ค่าเหล่านี้ถูกใช้เพื่อลบ entry ของ ARP หลังจากช่วงเวลาหนึ่ง :

```
arp -t ib -i <number in complete minutes to remove incomplete ARP entries>
-c <number in complete minutes to remove complete ARP entries>
```

ค่าดีฟอลต์ของเวลาปัจจุบันสำหรับ entry ของ ARP ที่ไม่สมบูรณ์ที่จะถูกลบ คือ 3 นาที สำหรับ entry ของ ARP ที่สมบูรณ์ เวลาดีฟอลต์คือ 24 ชั่วโมง ถ้าค่าต้องถูกเปลี่ยน การใช้คำสั่งจะเปลี่ยนเฉพาะค่าสำหรับอินเตอร์เฟสปัจจุบันทั้งหมดที่ถูกตั้งค่า (หรือมีสถานะที่ถูกกำหนด) ถ้าอินเตอร์เฟสถูกตั้งค่า คำสั่งต้องถูกเรียกใช้งานอีกครั้ง ค่ายังถูกเปลี่ยนใน ODM

ค่าสามารถถูกเปลี่ยนแบบไดนามิกเป็นอินเตอร์เฟสหนึ่งที่ถูกระบุโดยการใช้คำสั่ง **ifconfig** :

```
เพื่อเปลี่ยนตัวจับเวลา entry ของ ARP ที่ไม่สมบูรณ์
ifconfig ib0 inc_timer 4
ifconfig ib0 com_timer 60
```

การเปลี่ยนพารามิเตอร์ของ InfiniBand อินเตอร์เฟส

เพื่อเปลี่ยนพารามิเตอร์ของ IB IF ใช้ส่วนติดต่อผู้ใช้ของ SMIT หรือบรรทัดรับคำสั่ง

เพื่อเปลี่ยนพารามิเตอร์ของ IB IF โดยใช้ SMIT:

1. รันคำสั่ง **$ smitty inet** เมนู Network Interface Selection จะถูกแสดง
2. ในเมนู Network Interface Selection เลือก **Change / Show Characteristics of a Network Interface** เมนู Available Network Interfaces จะถูกแสดง
3. ในเมนู Available Network Interfaces ให้เลือก **InfiniBand Interface** เมนู Change / Show an IB Interface จะถูกแสดง
4. เปลี่ยนพารามิเตอร์ที่ต้องการ

เพื่อเปลี่ยนพารามิเตอร์ IB IF ที่บรรทัดรับคำสั่ง รันคำสั่ง **$ ifconfig** คำสั่งต่อไปนี้จะเปลี่ยนพารามิเตอร์ของ IB IF จากบรรทัดรับคำสั่ง :

```
$ ifconfig ib0 [ib_port port number mtu maximum transmission unit p_key
16 bits hex partition key  ib_adapter InfiniBand adapter name netmask
dotted decimals]

$ ifconfig ib0 inc_timer  3  com_timer 60
```

- *inc_timer* เป็นเวลาหน่วยเป็นนาทีที่ entry ของ ARP ที่ไม่สมบูรณ์จะหมดอายุ ค่าดีฟอลต์คือ 2 นาที
- *com_timer* เป็นเวลาหน่วยเป็นนาทีที่ entry ของ ARP ที่สมบูรณ์จะหมดอายุ ค่าดีฟอลต์คือ 24 ชั่วโมง

การตั้งค่าไดรเวอร์ InfiniBand Communication Manager

ใช้โพรซีเดอร์นี้เพื่อตั้งค่า InfiniBand Communication Manager

1. รันคำสั่ง **$ smitty icm** เมนูของ InfiniBand Communication Manager จะถูกแสดง
2. ในเมนูของ InfiniBand Communication Manager เลือก **Add an InfiniBand Communication Manager**
3. ในเมนูของ Add an InfiniBand Communication Manager เลือก **Add an InfiniBand Communication Manager** เมนู The Name of IB Communication Manager to Add จะถูกแสดง
4. ในเมนูของ Name of IB Communication Manager to Add เลือก **management icm InfiniBand**
5. ใช้ค่าดีฟอลต์ หรือเปลี่ยนพารามิเตอร์ที่จำเป็น และจากนั้นกด Enter

ตัวเริ่มต้นซอฟต์แวร์ iSCSI และซอฟต์แวร์เป้าหมาย

ตัวเริ่มต้นซอฟต์แวร์ iSCSI จะเปิดใช้งาน AIX เพื่อเข้าถึงอุปกรณ์หน่วยความจำโดยใช้ TCP/IP บน Ethernet เน็ตเวิร์กอะแด็ปเตอร์ ซอฟต์แวร์ iSCSI เป้าหมายจะเปิดใช้งาน AIX เพื่อเอ็กซ์พอร์ตหน่วยเก็บแบบโลคัลเพื่อให้สามารถถูกเข้าถึงโดยตัวเริ่มต้น iSCSI อื่น โดยใช้โปรโตคอล iSCSI ที่ถูกกำหนดใน RFC 3720

การใช้เทคโนโลยี iSCSI บ่อยครั้งจะถูกอ้างถึงเป็น SAN บนเทคโนโลยี IP ที่ยอมให้ใช้ storage area networking บน IP เน็ตเวิร์ก iSCSI เป็นวิธีแบบมาตรฐานแบบเปิด ที่ข้อมูล SCSI จะถูก encapsulate โดย **TCP/IP** เพื่อให้มันสามารถส่งบน Ethernet และ

gigabit Ethernet เน็ตเวิร์ก iSCSI ทำให้ Ethernet เน็ตเวิร์กที่มีอยู่สามารถส่งคำสั่งหรือข้อมูล SCSI โดยเป็นอิสระจากตำแหน่งโดนสิ้นเชิง โซลูชัน iSCSI ใช้ส่วนประกอบที่แตกต่าง แต่มีความสัมพันธ์กัน ต่อไปนี้:

- ตัวเริ่มต้น

  เป็นไดรเวอร์อุปกรณ์ที่อยู่บนไคลเอ็นต์ มันจะ encapsulate คำสั่ง SCSI และส่งมันบน IP เน็ตเวิร์กไปยังอุปกรณ์เป้าหมาย
- ซอฟต์แวร์เป้าหมาย

  ซอฟต์แวร์จะได้รับคำสั่ง SCSI ที่ถูก encapsulate บน IP เน็ตเวิร์ก ซอฟต์แวร์ยังสามารถให้การสนับสนุนการตั้งค่าและสนับสนุนการจักการหน่วยเก็บข้อมูล
- ฮาร์ดแวร์เป้าหมาย

  ฮาร์ดแวร์สามารถเป็นอุปกรณ์หน่วยเก็บข้อมูลที่ประกอบด้วยหน่วยเก็บข้อมูลแบบฝัง ฮาร์ดแวร์ยังสามารถเป็นผลิตภัณฑ์เกตเวย์หรือบริดจ์ที่ไม่มีหน่วยเก็บข้อมูลภายในของมันเอง

การตั้งค่าผู้เริ่มต้นซอฟต์แวร์ iSCSI

ผู้เริ่มต้นซอฟต์แวร์ถูกตั้งค่าโดยใช้ SMIT ดังแสดงในโพรซีเดอร์ต่อไปนี้

1. เลือก **Devices**
2. เลือก **iSCSI**
3. เลือก **Configure iSCSI Protocol Device**
4. เลือก **Change / Show Characteristics of an iSCSI Protocol Device**
5. ตรวจสอบว่าค่า **Initiator Name** ถูกต้อง ค่า **Initiator Name** ถูกใช้โดย iSCSI เป้าหมายระหว่างล็อกอิน

   หมายเหตุ: ชื่อดีฟอลต์ของตัวเริ่มต้นจะถูกกำหนดเมื่อซอฟต์แวร์ถูกติดตั้ง ชื่อผู้เริ่มต้นสามารถถูกเปลี่ยนโดยผู้ใช้เพื่อให้ตรงกับหลักการตั้งชื่อของโลคัลเน็ตเวิร์ก
6. ฟิลด์ **Maximum Targets Allowed** จะสอดคล้องกับจำนวนสูงสุดของ iSCSI เป้าหมายที่สามารถถูกตั้งค่า ถ้าคุณลดจำนวนนี้ คุณยังลดจำนวนของหน่วยความจำของเน็ตเวิร์กที่ถูกจัดสรรล่วงหน้าสำหรับไดรเวอร์โปรโตคอลของ iSCSI ระหว่างการตั้งค่า
7. กำหนดคอนฟิกเมธอดการค้นพบ iSCSI โดยใช้ฟิลด์ **Discovery Policy** เพื่อค้นพบ iSCSI เป้าหมาย ซอฟต์แวร์ iSCSI initiator สนับสนุนเมธอดการค้นพบ 4 เมธอดต่อไปนี้:

   ไฟล์ ข้อมูลเกี่ยวกับเป้าหมายถูกเก็บอยู่ในไฟล์คอนฟิกูเรชัน

   odm ข้อมูลเกี่ยวกับเป้าหมายถูกเก็บอยู่ในอ็อบเจ็กต์ Object Data Manager (ODM) เมื่อใช้ดิสก์ iSCSI เป็นบูตดิสก์หรือเป็นส่วนหนึ่งของบูต **rootvg** เมธอดการค้นพบ **odm** ต้องถูกนำมาใช้ โปรดดู การเพิ่ม iSCSI เป้าหมายที่ค้นพบเชิงสถิติลงใน ODM

   isns ข้อมูลเกี่ยวกับเป้าหมายถูกเก็บอยู่บนเซิร์ฟเวอร์ Internet Storage Name Service (iSNS) และถูกดึงในระหว่างคอนฟิกูเรชัน iSCSI initiator

   slp ข้อมูลเกี่ยวกับเป้าหมายถูกเก็บอยู่บนเซอร์วิสเอเจนต์ Service Location Protocol (SLP) หรือไดเร็กทอรีเอเจนต์และดึงข้อมูลโดยอัตโนมัติ ในระหว่างคอนฟิกูเรชัน iSCSI initiator

หลังจากตัวเริ่มต้นซอฟต์แวร์ถูกตั้งค่า ให้ทำต่อไปนี้:

1. หากนโยบายการค้นพบคือ **ไฟล์** ให้แก้ไขไฟล์ `/etc/iscsi/targets` เพื่อสอดแทรก iSCSI เป้าหมายที่ต้องการในระหว่างคอนฟิกูเรชันอุปกรณ์

แต่ละบรรทัดในไฟล์ที่ถูกยกเลิกหมายเหตุจะแทน iSCSI เป้าหมาย สำหรับข้อมูลเพิ่มเติม ดูที่ ไฟล์เป้าหมาย ใน *การอ้างอิงไฟล์*

หากนโยบายการค้นพบคือ **odm** ให้ใช้คำสั่ง **mkiscsi** หรือพาเนล smit เพื่อสร้างนิยามเป้าหมายใน ODM สำหรับข้อมูลเพิ่มเติม โปรดดู การเพิ่ม iSCSI เป้าหมายที่ค้นพบเชิงสถิติลงใน ODM

หากนโยบายการค้นพบ คือ **isns** หรือ **slp** ตรวจสอบให้แน่ใจว่า เซิร์ฟเวอร์ iSNS หรือ SLP ถูกกำหนดคอนฟิกไว้อย่างถูกต้องและสามารถเข้าถึงได้โดย iSCSI initiator

การตั้งค่าอุปกรณ์ iSCSI ต้องการให้ iSCSI เป้าหมายสามารถถูกเข้าถึงผ่านเน็ตเวิร์กอินเตอร์เฟสที่ถูกตั้งค่าอย่างถูกต้อง แม้ว่าซอฟต์แวร์ iSCSI initiator สามารถทำงานได้โดยใช้ 10/100 Ethernet LAN ซึ่งถูกออกแบบมาสำหรับใช้กับเครือข่ายกิกะไบต์ Ethernet ซึ่งแยกออกจาก ทราฟิกเครือข่ายอื่นๆ

2. หลังจากนิยามเป้าหมาย ให้พิมพ์คำสั่งต่อไปนี้:

   ```
   cfgmgr -l iscsi0
   ```

   คำสั่งนี้ กำหนดคอนฟิกซอฟต์แวร์ไดรฟ์ initiator

   คำสั่งนี้จะทำให้ไดรเวอร์พยายามสื่อสารกับเป้าหมายที่ลิสต์ในไฟล์ `/etc/iscsi/targets` และเพื่อกำหนด hdisk ใหม่สำหรับแต่ละ LUN บนเป้าหมายที่ถูกพบ สำหรับข้อมูลเพิ่มเติม ดูที่ คำสั่ง **cfgmgr** ที่ถูกอธิบายใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1*

   หมายเหตุ: ถ้าดิสก์ที่เหมาะสมไม่ถูกกำหนด ตรวจสอบการตั้งค่าของตัวเริ่มต้น เป้าหมาย และ iSCSI เกตเวย์เพื่อให้แน่ใจถึงความถูกต้อง และจากนั้นรันคำสั่ง **cfgmgr** ใหม่

ถ้าคุณต้องการตั้งค่าพารามิเตอร์เพิ่มเติมสำหรับอุปกรณ์ตัวเริ่มต้นซอฟต์แวร์ iSCSI ใช้ SMIT ดังต่อไปนี้ :

1. เลือก **Devices**
2. เลือก **Fixed Disk**

โดยทั่วไปอุปกรณ์ตัวเริ่มต้นซอฟต์แวร์จะดูเหมือนต่อไปนี้ :

```
hdisk2   Available              Other iSCSI Disk Drive
```

ถ้าดิสก์ iSCSI สนับสนุนคำสั่งการเข้าคิวแท็ก และ NACA=1 ในไบต์ควบคุม พิจารณาเปลี่ยนค่า queue depth ของดิสก์เป็นค่าที่มากขึ้น ค่าที่มากขึ้นจะช่วยปรับปรุงประสิทธิภาพของอุปกรณ์ การตั้งค่าของ queue depth ที่เหมาะสมไม่ควรเกินขนาดของคิวที่แท้จริงบนไดร์ฟ ตั้งค่า queue depth เพื่อให้มีค่าที่มากขึ้นจากนั้นขนาดของคิวของไดร์ฟอาจทำให้ประสิทธิภาพลดลงได้ เพื่อกำหนดขนาดของคิวของไดร์ฟให้ศึกษาจากเอกสารของไดร์ฟ

การตั้งค่าซอฟต์แวร์ iSCSI เป้าหมาย

ไดรเวอร์ของซอฟต์แวร์ iSCSI เป้าหมายจะทำให้ AIX ทำหน้าที่เป็นอุปกรณ์ iSCSI เป้าหมายหนึ่งตัว หรือหลายตัว ไดรเวอร์ iSCSI เป้าหมายจะเอ็กซ์พอร์ตโลคัลดิสก์ โลจิคัลวอลุ่ม หรือไฟล์แบบโลคัลไปยัง iSCSI ที่เริ่มต้นที่เชื่อมต่อกับ AIX โดยใช้โปรโตคอล iSCSI และ TCP/IP

อุปกรณ์เป้าหมายแต่ละตัวจะมี iSCSI Qualified Name และชุดของ logical unit numbers (LUNs) ที่พร้อมใช้กับตัวเริ่มต้นที่เชื่อมต่อกับ iSCSI เสมือนเป้าหมาย สำหรับอุปกรณ์เป้าหมายแต่ละตัว คุณสามารถระบุว่าเน็ตเวิร์กอินเตอร์เฟสใดและหมายเลขพอร์ต TCP/IP ใดที่ไดรเวอร์เป้าหมายสามารถใช้เพื่อรับการเชื่อมต่อขาเข้า

หมายเหตุ: คุณต้องมีชุดของไฟล์ iSCSI เป้าหมายติดตั้งอยู่ ชื่อของชุดไฟล์ คือ devices.tmiscsw.rte และชุดไฟล์มีอยู่บน AIX Expansion pack

เพื่อตั้งค่าไดรเวอร์ iSCSI เป้าหมาย ทำตามขั้นตอนต่อไปนี้ :

1. สร้างอินสแตนซ์เดียวของไดรเวอร์ iSCSI เป้าหมายโดยใช้พาธ SMIT ต่อไปนี้ อินสแตนซ์จะทำหน้าที่เป็น container สำหรับอ็อบเจ็กต์ของ iSCSI อื่น

   Devices > iSCSI > iSCSI Target Device > iSCSI Target Protocol Device > Add an iSCSI Target Protocol Device

2. สร้างอุปกรณ์ iSCSI เป้าหมายหนึ่งตัวสำหรับแต่ละ iSCSI เสมือนเป้าหมายที่ถูกจัดสรรโดยไดรเวอร์ iSCSI เป้าหมาย ใช้พาธ SMIT ต่อไปนี้เพื่อสร้างอุปกรณ์ iSCSI เป้าหมายแต่ละตัว :

   Devices > iSCSI > iSCSI Target Device > iSCSI Targets > Add an iSCSI Targets

3. กำหนดหนึ่ง LUN หรือมากกว่าสำหรับอุปกรณ์เป้าหมายแต่ละตัวโดยใช้พาธ SMIT ต่อไปนี้ :

   หมายเหตุ: LUN สามารถเข้าถึงได้โดยตัวเริ่มต้นที่เชื่อมต่อกับเป้าหมายเสมือน บน iSCSI เป้าหมาย แต่ละ LUN สามารถถูกเชื่อมโยงกับโลจิคัลวอลุ่มที่ถูกกำหนดก่อนหน้านี้ กับวอลุ่มแบบฟิสิคัล หรือกับไฟล์ที่ถูกสร้างก่อนหน้านี้บนระบบไฟล์แบบโลคัล วอลุ่มแบบฟิสิคัลใดๆที่ถูกเชื่อมโยงกับหน่วยของ iSCSI เป้าหมายแบบโลจิคัลไม่สามารถถูกใช้ในวิธีอื่นโดยระบบ AIX ที่รันไดรเวอร์ iSCSI เป้าหมาย

   Devices > iSCSI > iSCSI Target Device > iSCSI Target LUNs

   ขั้นตอนนี้จะทำการตั้งค่า อย่างไรก็ตาม ถ้าคุณใช้ Challenge Handshake Authentication Protocol (CHAP) หรือถ้าคุณใช้ Access Control Lists (ACLs) เพื่อระบุว่าตัวเริ่มต้นใดสามารถเข้าถึง LUN ใด อาจต้องการขั้นตอนเพิ่มเติมเพื่อทำการตั้งค่าเป้าหมาย

   - ถ้าคุณใช้การพิสูจน์ตัวตน CHAP ของตัวเริ่มต้น แก้ไขไฟล์ `/etc/tmiscsi/autosecrets` และเพิ่มรหัสลับที่ถูกใช้โดยตัวเริ่มต้นเพื่อล็อกอิน ไฟล์ `/etc/tmiscsi/autosecrets` ประกอบด้วยหนึ่ง entry ต่อเป้าหมาย แต่ละ entry มีรูปแบบต่อไปนี้ :

     target_name chap_name chap_secret

   - ถ้าคุณใช้ ACLs เพื่อระบุว่าตัวเริ่มต้นใดสามารถเข้าถึง LUN ใด แก้ไขไฟล์ `/etc/tmiscsi/access_lists` เพื่อเพิ่มหนึ่ง entry ต่อเป้าหมาย แต่ละ entry มีรูปแบบต่อไปนี้ :

     target_namelun_name iSCSI_name, *iSCSI_name,...*

ข้อมูลที่เกี่ยวข้อง:

`/etc/tmiscsi/autosecrets`

`/etc/tmiscsi/access_lists`

`/etc/tmiscsi/isns_servers`

ข้อควรพิจารณาเกี่ยวกับผู้เริ่มต้นซอฟต์แวร์ iSCSI

พิจารณาต่อไปนี้เมื่อจัดการกับผู้เริ่มต้นซอฟต์แวร์ iSCSI

- การค้นหาเป้าหมาย

  ซอฟต์แวร์ iSCSI initiator สนับสนุนแบบฟอร์ม 4 แบบฟอร์มต่อไปนี้ของการค้นพบเป้าหมาย:

  ไฟล์ เท็กซ์ไฟล์ถูกใช้เพื่อตั้งค่าแต่ละเป้าหมาย

  odm อ็อบเจ็กต์ ODM ถูกใช้เพื่อกำหนดคอนฟิกให้กับแต่ละเป้าหมาย เมื่อใช้ดิสก์ iSCSI เป็นบูตดิสก์หรือเป็นส่วนหนึ่งของบูต rootvg เมธอดการค้นพบ **odm** ต้องถูกนำมาใช้

isns แต่ละเป้าหมายถูกลงทะเบียนในเซิร์ฟเวอร์ Internet Storage Name Service (iSNS) หนึ่งเครื่องหรือมากกว่า

slp แต่ละเป้าหมายถูกลงทะเบียนในเซอร์วิสเอเจนต์ Service Location Protocol (SLP) หรือไดเร็กทอรีเอเจนต์ตั้งแต่หนึ่งตัวขึ้นไป

- การพิสูจน์ตัวจริง iSCSI

  เฉพาะ CHAP(MD5) สามารถถูกใช้เพื่อตั้งค่าการพิสูจน์ตัวตนผู้เริ่มต้น การพิสูจน์ตัวจริงเป้าหมายไม่ได้ใช้

- จำนวนของ LUN ที่ถูกตั้งค่า

  จำนวนสูงสุดของ LUN ที่ถูกตั้งค่าถูกทดสอบโดยใช้ตัวเริ่มต้นซอฟต์แวร์ iSCSI คือ 128 ต่อ iSCSI เป้าหมาย ตัวเริ่มต้นซอฟต์แวร์จะใช้การเชื่อมต่อ TCP เดียวสำหรับแต่ละ iSCSI เป้าหมาย (หนึ่งการเชื่อมต่อต่อ iSCSI เซสชัน) การเชื่อมต่อ TCP นี้ถูกแบ่งใช้ระหว่าง LUNs ทั้งหมดที่ถูกตั้งค่าสำหรับเป้าหมาย พื้นที่การส่งและรับของ TCP ซ็อกเก็ตของตัวเริ่มต้นซอฟต์แวร์จะถูกตั้งเป็นบัฟเฟอร์ซ็อกเก็ตสูงสุดของระบบ ค่าสูงสุดถูกตั้งโดยเน็ตเวิร์กอ็อพชัน **sb_max** ค่าดีฟอลต์คือ 1 MB

- กลุ่มวอลุ่ม

  เพื่อหลีกเลี่ยงปัญหาของการตั้งค่าและ entry ของบันทึกข้อผิดพลาดเมื่อคุณสร้างกลุ่มวอลุ่มโดยใช้อุปกรณ์ iSCSI ให้ทำตามแนวทางเหล่านี้ :

  – ตั้งค่ากลุ่มวอลุ่มที่ถูกสร้างโดยใช้อุปกรณ์ iSCSI ให้อยู่ในสถานะ inactive หลังจากรีบูต หลังจากอุปกรณ์ iSCSI ถูกตั้งค่า เปิดใช้งานกลุ่มวอลุ่ม iSCSI-backed แบบแมนวล จากนั้น เมาต์ที่เกี่ยวข้อง

    กลุ่มวอลุ่มจะถูกเปิดใช้งานระหว่างเฟสของการบูตที่แตกต่างกันเมื่อเทียบกับไดรเวอร์ซอฟต์แวร์ iSCSI ด้วยสาเหตุนี้ จึงเป็นไปไม่ได้ที่จะเปิดใช้งานกลุ่มวอลุ่ม iSCSI ระหว่างกระบวนการบูต

  – ห้ามแตกกลุ่มวอลุ่มข้ามอุปกรณ์ที่ไม่ใช่ iSCSI

- ความล้มเหลวของ I/O

  ถ้าการเชื่อมต่อไปยังอุปกรณ์ iSCSI เป้าหมายขาดไป ความล้มเหลวของ I/O จะเกิดขึ้น เพื่อป้องกันความล้มเหลวของ I/O และระบบไฟล์ความล้มเหลว ให้หยุดกิจกรรมทั้งหมดของ I/O และถอดระบบไฟล์ของ iSCSI backed ออกก่อนที่จะทำอะไรที่ทำให้การเชื่อมต่อไปยัง iSCSI เป้าหมายขาดหายเป็นเวลานาน

  ถ้าการเชื่อมต่อไปยัง iSCSI เป้าหมายขาดหายไประหว่างที่แอ็พพลิเคชันพยายามทำกิจกรรม I/O กับอุปกรณ์ iSCSI ข้อผิดพลาด I/O จะเกิดขึ้นได้เช่นกัน มันอาจเป็นไปไม่ได้ที่จะถอดออกระบบไฟล์ของ iSCSI backed เนื่องจากอุปกรณ์ iSCSI นั้นกำลังทำงานอยู่

  การบำรุงรักษาระบบไฟล์ต้องถูกกระทำ ถ้าความล้มเหลวของ I/O เกิดขึ้นเนื่องจากการเชื่อมต่อไปยัง iSCSI เป้าหมายที่แอ็คทีฟขาดหายไป เพื่อทำการบำรุงรักษาระบบไฟล์ รันคำสั่ง **fsck**

- ห้ามใช้โปรแกรมเริ่มต้นซอฟต์แวร์ AIX iSCSI หรือซอฟต์แวร์ AIX iSCSI เป้าหมายที่มีอินเตอร์เฟส loopback (lo0) การประมวลผลของการรบกวนอินเตอร์เฟส loopback แตกต่างจากการประมวลผล ของการรบกวนอีเทอร์เน็ตอะแด็ปเตอร์เน็ตเวิร์กอินเตอร์เฟสแบบฟิสิคัลหรือแบบเสมือน ระบบปฏิบัติการ AIX อาจหยุดทำงานหากอินเตอร์เฟส loopback ถูกใช้กับไดร์เวอร์ซอฟต์แวร์ iSCSI

ข้อมูลที่เกี่ยวข้อง:

การเพิ่มเป้าหมาย iSCSI ที่ค้นพบแบบสแตติกเข้าใน ODM

ข้อควรพิจารณาด้านความปลอดภัยของ iSCSI:

ไดเร็กทอรี /etc/iscsi, ไดเร็กทอรี /etc/tmiscsi, และไฟล์ในไดเร็กทอรีเหล่านี้ถูกป้องกันไว้จากผู้ใช้ที่ไม่ได้รับอนุญาตผ่านสิทธิใช้งานไฟล์และความเป็นเจ้าของ

ความลับของ CHAP ถูกเก็บไว้ในไฟล์ /etc/iscsi/targets และไฟล์ /etc/tmiscsi/autosecrets ในข้อความเคลียร์

หมายเหตุ: ห้ามเปลี่ยนสิทธิใช้งานไฟล์ และความเป็นเจ้าของเดิมของไฟล์เหล่านี้

ข้อควรพิจารณาเกี่ยวกับประสิทธิภาพของ iSCSI:

ตั้งการตั้งค่าต่อไปนี้เพื่อให้ได้ประสิทธิภาพที่ดีที่สุดจาก iSCSI

เพื่อให้แน่ใจว่าได้ประสิทธิภาพที่ดีที่สุด :

- เปิดใช้งาน TCP Large Send, โฟลว์คอนโทรลการส่งและรับ TCP และคุณลักษณะ Jumbo Frame ของ AIX Gigabit Ethernet อะแด็ปเตอร์ และอินเตอร์เฟส iSCSI เป้าหมาย
- ปรับแต่งเน็ตเวิร์กอ็อพชันและอินเตอร์เฟสพารามิเตอร์สำหรับทรูพุตของ iSCSI I/O สูงสุดบนระบบ AIX ดังต่อไปนี้ :
  - เปิดใช้งาน RFC 1323 เน็ตเวิร์กออ็อพชัน
  - ตั้งค่าเน็ตเวิร์กอ็อพชัน **tcp_sendspace, tcp_recvspace, sb_max** และ **mtu_size** เน็ตเวิร์กอินเตอร์เฟสอ็อพชันเป็นค่าที่เหมาะสม

    ขนาดการถ่ายโอนสูงสุดของตัวเริ่มต้นซอฟต์แวร์ iSCSI คือ 256 KB สมมุติว่าค่าสูงสุดของระบบสำหรับ **tcp_sendspace** และ **tcp_recvspace** ถูกตั้งเป็น 262144 ไบต์ คำสั่ง **ifconfig** ที่ใช้ตั้งค่า gigabit Ethernet อินเตอร์เฟสจะเหมือนดังต่อไปนี้ :

    ```
    ifconfig en2 10.1.2.216 mtu 9000 tcp_sendspace 262144 tcp_recvspace 262144
    ```
  - ตั้งเน็ตเวิร์ก **sb_max** เป็นอย่างน้อย 524288 โดยค่าที่ต้องการเป็น 1048576
  - ตั้ง **mtu_size** เป็น 9000
  - สำหรับบาง iSCSI เป้าหมาย อัลกอริทึม TCP Nagle ต้องถูกปิดการทำงานเพื่อให้ได้ประสิทธิภาพที่ดีที่สุด ใช้คำสั่ง **no** เพื่อตั้งพารามิเตอร์ **tcp_nagle_limit** เป็น 0 ซึ่งจะปิดการใช้งานอัลกอริทึม Nagle

หมายเหตุ: สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการตั้งเน็ตเวิร์กอ็อพชัน ดูที่ คำอธิบายคำสั่ง **no** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*

สำหรับข้อมูลเพิ่มเติมและการปรับแต่งพารามิเตอร์เพิ่มเติม ดูที่ การปรับแต่งประสิทธิภาพ TCP และ UDP

ข้อควรพิจารณาเกี่ยวกับตัวเริ่มต้นซอฟต์แวร์ iSCSI

พิจารณาต่อไปนี้เมื่อคุณกำหนดซอฟต์แวร์ iSCSI เป้าหมายและการเอ็กซ์พอร์ต logical unit numbers (LUNs):

- iSCSI Qualified Name (IQN) ของแต่ละเป้าหมายเสมือนจะถูกระบุ SMIT เมื่อซอฟต์แวร์เป้าหมายถูกกำหนด พาเนลของ SMIT ไม่ได้จำกัดรูปแบบของชื่อ อย่างไรก็ตาม ตัวเริ่มต้น iSCSI บางตัวต้องการให้ IQN ถูกระบุในรูปแบบที่ถูกกำหนดโดยโปรโตคอล iSCSI การใช้รูปแบบของชื่อที่ไม่ถูกต้องอาจทำให้ตัวเริ่มต้นไม่สามารถล็อกอินกับเป้าหมายและไม่สามารถเข้าถึงดิสก์ที่ถูกเอ็กซ์ปอร์ตโดยเป้าหมาย

  เพื่อแสเงชื่อปัจจุบันของอุปกรณ์ iSCSI เป้าหมาย ให้ทำขั้นตอนต่อไปนี้ :

  1. รันคำสั่งเหมือนดังต่อไปนี้ สำหรับตัวอย่างนี้ สมมุติว่าอุปกรณ์ iSCSI เป้าหมาย คือ target0

     ```
     lsattr -E -l target0
     ```
  2. ตรวจสอบแอ็ตทริบิวต์ iscsi_name
- การสอบถามข้อมูลที่ถูกคืนกลับมาสำหรับ LUN ที่ถูกเอ็กซ์พอร์ต มีค่าดังต่อไปนี้ :
  - Vendor ID: AIX
  - Product ID: iSCSI_VDASD

– ANSI version number: 3

- ห้ามใช้โปรแกรมเริ่มต้นซอฟต์แวร์ AIX iSCSI หรือซอฟต์แวร์ AIX iSCSI เป้าหมายที่มีอินเตอร์เฟส loopback (lo0) การประมวลผลของการรบกวนอินเตอร์เฟส loopback แตกต่างจากการประมวลผล ของการรบกวนอีเทอร์เน็ตอะแด็ปเตอร์เน็ตเวิร์กอินเตอร์เฟสแบบฟิสิคัลหรือแบบเสมือน ระบบปฏิบัติการ AIX อาจหยุดทำงานหากอินเตอร์เฟส loopback ถูกใช้กับไดร์เวอร์ซอฟต์แวร์ iSCSI

Stream Control Transmission Protocol

Stream Transmission Control Protocol (SCTP) คือโปรโตคอลแบบ connection-oriented ซึ่งคล้ายกับ TCP แต่จัดเตรียมการถ่ายโอนข้อมูลแบบ message-oriented คล้ายกับ **UDP** ระบบปฏิบัติการ AIX เข้ากันได้ กับ RFC 4960

ตารางต่อไปนี้ไฮไลต์ความแตกต่างทั่วไปในลักษณะการทำงานระหว่าง SCTP และการส่งผ่านโปรโตคอลที่มีอยู่ TCP และ UDP

ตารางที่ 84. ความแตกต่างระหว่าง TCP, UDP และ SCTP

แอ็ตทริบิวต์	TCP	UDP	SCTP
ความเชื่อถือได้	ความไว้วางใจได้	ไม่สามารถไว้วางใจได้	ไว้วางใจได้
การจัดการกับการเชื่อมต่อ	Connection-oriented	Connectionless	Connection-oriented
การส่งข้อมูล	Byte-oriented	Message-oriented	Message-oriented
Flow Control	Yes	ไม่	Yes
การควบคุมความแออัด	Yes	ไม่	Yes
Fault Tolerance	ไม่	ไม่	Yes
การส่งมอบข้อมูล	การเรียงลำดับที่จำกัด	ไม่เรียงลำดับ	การเรียงลำดับบางส่วน
การรักษาความปลอดภัย	Yes	Yes	ปรับปรุงแล้ว

โดยทั่วไป **SCTP** อาจจัดเตรียมความยืดหยุ่นสำหรับแอ็พพลิเคชันบางตัว เช่น **Voice over IP (VoIP)** ที่ต้องการไว้วางใจแต่การถ่ายโอนข้อมูล message-oriented สำหรับหมวดหมู่นี้ของแอ็พพลิเคชัน **SCTP** จะเหมาะสมกว่า **TCP** หรือ **UDP**

- **TCP** จัดเตรียมความไว้วางใจและจำกัดการส่งมองข้อมูลการส่งผ่านที่จำกัด สำหรับแอ็พพลิเคชันที่ต้องการความเชื่อถือได้แต่สามารถทนต่อการส่งมอบข้อมูลที่ไม่เรียงลำดับ หรือเรียงลำดับเป็นบางส่วน **TCP** อาจเป็นสาเหตุทำให้เกิดการหน่วงเวลาที่ไม่จำเป็น เนื่องจากการบล็อกส่วนหัวของบรรทัด ด้วยแนวคิดของสตรีมจำนวนมากภายในการเชื่อมต่อ **SCTP** สามารถจัดเตรียมการส่งมอบที่เรียงลำดับแล้วอย่างจำกัดภายในสตรีม ขณะที่แยกข้อมูลออกจากสตรีมที่แตกต่างกัน
- **SCTP** คือ message-oriented ซึ่งไม่เหมือนกับ **TCP** ที่เป็น byte-oriented เนื่องจากลักษณะการทำงานแบบ byte-oriented ของ **TCP** แอ็พพลิเคชัน จะเพิ่มการทำเครื่องหมายที่เร็กคอร์ดของตนเพื่อรักษาขอบเขตของข้อความ
- **SCTP** จัดเตรียมดีกรีของ fault tolerance โดยใช้คุณลักษณะแบบ Multihoming โฮสต์ถูกพิจารณาว่าเป็น multihomed เมื่อมีมากกว่าหนึ่งเน็ตเวิร์กอินเตอร์เฟส ที่พ่วงต่อบนเน็ตเวิร์กเดียวกันหรือต่างเน็ตเวิร์ก อย่างใดอย่างหนึ่ง การเชื่อมโยง **SCTP** สามารถสร้างขึ้นได้ระหว่างโฮสต์แบบ multihomed สองโฮสต์ ในกรณีนี้ IP แอดเดรส ทั้งหมดของส่วนปลายทั้งสองด้านถูกแลกเปลี่ยนเมื่อเริ่มต้นการเชื่อมโยงซึ่งอนุญาตให้แต่ละส่วนปลาย ใช้แอดเดรสเหล่านี้ผ่านการเชื่อมต่อที่ใช้งานได้ หากหนึ่งในอินเตอร์เฟสหยุดทำงานด้วยเหตุผลใดๆ ตราบเท่าที่เพียร์ยังสามารถเข้าถึงได้ผ่าน อินเตอร์เฟสสำรอง
- **SCTP** จัดเตรียมคุณลักษณะความปลอดภัยเพิ่มเติมซึ่ง **TCP** และ **UDP** ไม่ได้ทำ ใน **SCTP** การจัดสรรรีซอร์สในระหว่างการติดตั้งการเชื่อมโยงถูกหน่วงเวลา จนกว่า identity ของไคลเอ็นต์สามารถตรวจสอบการใช้กลไกการแลกเปลี่ยนคุกกี้ ดังนั้นจึงลดความน่าจะเป็นของการต่อสู้แบบ Denial of Service

การเริ่มต้นการทำงานและปิดระบบการเชื่อมโยง SCTP

คำแนะนำในการเริ่มต้นและการปิดการเชื่อมโยง **SCTP** ถูกกล่าวถึง ไว้ในที่นี้

การเชื่อมโยง **SCTP** ประกอบขึ้นเป็น handshake สี่วิธีที่เข้าแทนที่ ในลำดับต่อไปนี้:

1. ไคลเอ็นต์ส่งสัญญาณ **INIT** ไปยังเซิร์ฟเวอร์ เพิ่มเริ่มต้นการเชื่อมโยง
2. สำหรับการรับสัญญาณ **INIT** เซิร์ฟเวอร์ส่งการตอบกลับ **INIT-ACK** ไปยังไคลเอ็นต์ สัญญาณ **INIT-ACK** มีสถานะคุกกี้ สถานะคุกกี้นี้ ต้องมี Message Authentication Code (MAC) พร้อมกับการประทับเวลา ที่สอดคล้องกับการสร้างคุกกี้ การขยายเวลาของสถานะคุกกี้ และข้อมูลที่จำเป็นต่อการสร้างการเชื่อมโยง MAC ถูกคำนวณโดยเซิร์ฟเวอร์แบบอิงคีย์ลับที่รู้จักเท่านั้น
3. สำหรับการรับสัญญาณ **INIT-ACK** นี้ ไคลเอ็นต์ส่งการตอบกลับ **COOKIE-ECHO** ที่เพิ่ง echo สถานะคุกกี้
4. หลังจากที่ตรวจสอบการพิสูจน์ตัวตนของสถานะคุกกี้โดยใช้คีย์ลับแล้ว เซิร์ฟเวอร์จะจัดสรรรีซอร์สสำหรับการเชื่อมโยง ส่งการตอบกลับ **COOKIE-ACK** ที่ตอบรับสัญญาณ **COOKIE-ECHO** และย้ายการเชื่อมโยงไปเป็นสถานะ **ESTABLISHED**

SCTP ยังสนับสนุนการผ่อนผันการปิดการเชื่อมโยงที่แอ็คทีฟ ตามคำร้องขอจากผู้ใช้ **SCTP** ลำดับต่อไปนี้ของเหตุการณ์จะเกิดขึ้น:

1. ไคลเอ็นต์ส่งสัญญาณ **SHUTDOWN** ไปยังเซิร์ฟเวอร์ ซึ่งจะแจ้งให้เซิร์ฟเวอร์ทราบว่า ไคลเอ็นต์พร้อมที่จะปิดการเชื่อมต่อแล้ว
2. ไคลเอ็นต์ตอบกลับโดยส่งการตอบกลับ **SHUTDOWN-ACK**
3. จากนั้น ไคลเอ็นต์ส่งสัญญาณ **SHUTDOWN-COMPLETE** กลับไปยังเซิร์ฟเวอร์

SCTP ยังสนับสนุนการปิดในทันที (สัญญาณ **ABORT**) ของการเชื่อมโยงที่แอ็คทีฟตามคำร้องขอจากไคลเอ็นต์ **SCTP** หรือเนื่องจาก เกิดข้อผิดพลาดในสแต็ก **SCTP** อย่างไรก็ตาม **SCTP** ไม่สนับสนุนการเชื่อมต่อแบบเปิด ครึ่งหนึ่ง ข้อมูลเพิ่มเติมเกี่ยวกับโปรโตคอล และส่วนภายในของโปรโตคอลสามารถดูได้ใน RFC 4960

นอกจากนี้ ความแตกต่างที่กล่าวถึงข้างต้นระหว่าง **SCTP** และโปรโตคอนการส่งผ่านที่มีอยู่ นั่นคือ **SCTP** จัดเตรียมคุณลักษณะดังต่อไปนี้:

- **การส่งมอบตามลำดับภายในสตรีม**: สตรีมในคอนเท็กซ์ **SCTP** อ้างถึงลำดับของข้อความผู้ใช้ที่ถูกส่งผ่านระหว่างจุดปลาย การเชื่อมโยง **SCTP** สามารถสนับสนุนสตรีมจำนวนมาก ณ เวลาของการติดตั้งการเชื่อมโยง ผู้ใช้สามารถระบุจำนวนของสตรีมได้ ค่าที่มีผลกระทบของจำนวนสตรีม ถูกแก้ไขหลังจากต่อรองกับเพียร์ ภายในสตรีมแต่ละตัว ลำดับของการส่งมอบข้อมูลจะถูกรักษาไว้อย่างจำกัด อย่างไรก็ตาม ระหว่างการส่งมอบข้อมูลสตรีม จะเป็นอิสระ ดังนั้น การสูญเสียข้อมูลจากหนึ่งสตรีมไม่ได้ปกป้องข้อมูลจาก การส่งมอบในสตรีมอื่น ซึ่งอนุญาตให้ใช้แอ็พพลิเคชันสำหรับผู้ใช้ เพื่อใช้สตรีมอื่นๆ สำหรับข้อมูลที่เป็นอิสระในเชิงโลจิคัล ข้อมูลยังสามารถ ส่งมองในแบบที่ไม่เรียงลำดับโดยใช้อ็อพชันพิเศษ ซึ่งมีประโยชน์ต่อการส่งข้อมูลแบบเร่งด่วน
- **การแตกแฟรกเมนต์ข้อมูลของผู้ใช้**: **SCTP** สามารถแตกแฟรกเมนต์ข้อความผู้ใช้ เพื่อมั่นใจว่า ขนาดของแพ็กเก็ตถูกส่งผ่านไปยังเลเยอร์ที่ต่ำกว่าซึ่งไม่เกินพาธ MTU ณ เวลารับ แฟรกเมนต์ถูกรวบรวมเป็นข้อความที่สมบูรณ์ และส่งผ่านไปยังผู้ใช้ แม้ว่าแฟรกเมนต์ยังสามารถดำเนินการได้ที่ระดับเน็ตเวิร์ก แฟรกเมนต์ในเลเยอร์ของการส่งผ่านจัดเตรียมประโยชน์จำนวนมาก ผ่านแฟรกเมนต์เลเยอร์ IP ประโยชน์เหล่านี้บางข้อที่สอดแทรกไว้ไม่มีอยู่ เพื่อส่งข้อความทั้งหมดอีกครั้ง เมื่อแฟรกเมนต์สูญหายในเน็ตเวิร์ก และลดภาระของเราเตอร์ ซึ่งจะมีการดำเนินการแตกแฟรกเมนต์ IP
- **การตอบรับและการควบคุมความคับคั่ง**: การตอบรับแพ็กเก็ต เป็นสิ่งจำเป็นสำหรับการส่งมอบข้อมูลที่เชื่อถือได้ เมื่อ **SCTP** ไม่ได้ขอรับการตอบรับสำหรับแพ็กเก็ต ที่ส่งภายในเวลาที่ระบุไว้ SCTP จะทริกเกอร์การส่งข้อมูลของ แพ็กเก็ตเดียว

กัน **SCTP** ทำตามอัลกอริทึมการควบคุมความคับคั่ง ซึ่งคล้ายกับที่ใช้โดย **TCP** นอกจากนี้ การใช้การตอบรับแบบสะสม เช่น **TCP SCTP** จะใช้กลไก Selective Acknowledgment (SACK) ซึ่งอนุญาตให้ตอบรับแพ็กเก็ตที่เลือกไว้

- **การบันเดิลแบบเป็นชิ้น**: ชิ้นข้อมูลอาจมีข้อมูลของผู้ใช้หรือข้อมูลการควบคุม **SCTP** ชิ้นข้อมูลจำนวนมากถูกบันเดิลพร้อมกันภายใต้ส่วนหัว **SCTP** เดียวกัน การบันเดิลชิ้นข้อมูลจำเป็นต้องมีการรวบรวมชิ้นข้อมูลลงในแพ็กเก็ต **SCTP** ที่การส่งส่วนปลายและแยกส่วนของแพ็กเก็ตตามลำดับไปเป็นชิ้นข้อมูล ที่ receiver ส่วนปลาย
- **การตรวจสอบความถูกต้องของแพ็กเก็ต**: แต่ละแพ็กเก็ต **SCTP** มีฟิลด์แท็กการตรวจสอบความถูกต้อง ซึ่งตั้งค่าในระหว่างการเชื่อมโยงการเริ่มต้นทำงานด้วยจุดปลายแต่ละจุด แพ็กเก็ตทั้งหมดถูกส่งพร้อมกับแท็กการตรวจสอบความถูกต้องผ่านช่วงเวลาที่ใช้งานได้ของ การเชื่อมโยง ในระหว่างช่วงเวลาที่ใช้งานได้ของการเชื่อมโยง หากแพ็กเก็ตได้รับ พร้อมกับแท็กการตรวจสอบความถูกต้องที่ไม่ได้คาดคิดไว้ แพ็กเก็ตจะถูกละทิ้ง และเช็กซัม CRC-32 ควรตั้งค่าโดยผู้ส่งแพ็กเก็ต **SCTP** แต่ละรายเพื่อจัดเตรียมการปกป้องที่เพิ่มขึ้น สำหรับความล้มเหลวของข้อมูลในเน็ตเวิร์ก แพ็กเก็ตใดๆ ที่ได้รับพร้อมกับเช็กซัม CRC-32 ที่ไม่ถูกต้องจะถูกละทิ้ง
- **การจัดการกับพาธ**: ณ เวลาของการเชื่อมโยงการติดตั้ง จุดปลายแต่ละจุด อาจนำเนอรายการของแอดเดรสการส่งผ่านที่มี อย่างไรก็ตาม เฉพาะหนึ่งพาธหลักเท่านั้น ที่ถูกกำหนดไว้สำหรับการเชื่อมโยง **SCTP** และถูกใช้สำหรับ การถ่ายโอนข้อมูลปกติ ในกรณีที่พาธหยุดทำงาน แอดเดรสการส่งผ่านอื่น จะถูกใช้ในระหว่างช่วงอายุของการเชื่อมโยง สัญญาณ heartbeat จะส่งไปที่ช่วงเวลาปกติ ผ่านพาธทั้งหมดเพื่อมอนิเตอร์สถานะของพาธ

SCTP socket APIs

คุณลักษณะของ **SCTP** socket APIs สอดแทรกความสอดคล้องกัน ความสามารถในการเข้าถึง และความเข้ากันได้

SCTP Socket APIs ถูกออกแบบมาเพื่อจัดเตรียมคุณลักษณะต่อไปนี้:

- รักษาความสอดคล้องกันกับซ็อกเก็ต API ที่มีอยู่
- จัดเตรียมข้อมูลพื้นฐานสำหรับการเข้าถึงคุณลักษณะ **SCTP** ใหม่
- จัดเตรียมความเข้ากันได้ ดังนั้น แอ็พพลิเคชัน **TCP** และ **UDP** ส่วนใหญ่ที่มีอยู่สามารถถ่ายโอนไปยัง **SCTP** พร้อมกับการเปลี่ยนแปลงเพียงเล็กน้อย

หากสิ่งอำนวยความสะดวกง่ายต่อการถ่ายโอนแอ็พพลิเคชัน **TCP** และ **UDP** ที่มีอยู่ ลักษณะที่แตกต่างกันสองลักษณะของ **SCTP** APIs ได้ถูกแปลงเป็นสูตร:

- **UDP**-Style API – ซีแมนทิกส์คล้ายกับที่นิยามไว้สำหรับ โปรโตคอลที่มีการเชื่อมต่อแบบ connectionless เช่น **UDP**
- **TCP**-Style API – ซีแมนทิกส์คล้ายกับที่นิยามไว้สำหรับโปรโตคอลที่มีการเชื่อมต่อแบบ connection-oriented เช่น **TCP**

ตลอดทั้ง **SCTP** อนุญาตให้ใช้สำหรับลักษณะของ **TCP** และ **UDP** ของซ็อกเก็ต API ที่ต้องถูกนิยามและใช้ใน AIX 5.3 เฉพาะส่วนสนับสนุนสำหรับไวยากรณ์ซ็อกเก็ตที่มีลักษณะ **UDP** ถูกจัดเตรียมไว้ เนื่องจาก API ที่มีลักษณะ **UDP** จัดเตรียมความยืดหยุ่นมากกว่าในการเข้าถึงคุณลักษณะใหม่ของ **SCTP** การใช้ API ลักษณะ **UDP** เซิร์ฟเวอร์ใช้ลำดับต่อไปนี้ของการเรียก ในระหว่างช่วงเวลาใช้งานของการเชื่อมโยง

1. **socket**()
2. **bind**()
3. **listen**()
4. **recvmsg**()
5. **sendmsg**()
6. **close**()

ไคลเอ็นต์ใช้ลำดับต่อไปนี้ของการเรียกซ็อกเก็ต API:

1. **socket**()
2. **sendmsg**()
3. **recvmsg**()
4. **close**()

การเชื่อมโยงที่สร้างขึ้นโดยใช้ลำดับการเรียกข้างต้นถูกเรียกการเชื่อมโยง ที่สร้างขึ้น การเชื่อมโยงสามารถถูกสร้างขึ้นโดยนัย หลังจากสร้างซ็อกเก็ต โดยเรียก **sendmsg**(), **recvmsg**() หรือ **sendto**() and **recvto**() ในกรณีของการเชื่อมโยงโดยนัย การเรียก **bind**() และ **listen**() ไม่จำเป็นต้องมี ไวยากรณ์ของการเรียกระบบ เหล่านี้คล้ายกับที่ใช้กับซ็อกเก็ต **UDP** สำหรับรูทีนย่อยซ็อกเก็ต ฟิลด์ **Type** ควรถูกตั้งค่าเป็น SOCK_SEQPACKET และฟิลด์ **Protocol** ควรเป็น IPPROTO_SCTP นอกจากนี้ ซ็อกเก็ต API มาตรฐานเหล่านี้ APIs **SCTP** จัดเตรียม API ใหม่สองตัว: **sctp_peeloff**() และ **sctp_opt_info**() ข้อมูลเพิ่มเติมเกี่ยวกับการใช้ Socket API สำหรับ **SCTP** สามารถพบได้ใน SCTP Socket API Draft **SCTP** ได้ถูกนำไปใช้เป็นส่วนขยายเคอร์เนลใน AIX 5.3 ผู้ใช้สามารถใช้คำสั่ง **sctpctrl** เพื่อโหลดและยกเลิกการโหลด ส่วนขยายเคอร์เนล SCTP

นอกจากนี้ คำสั่งนี้ยังสามารถใช้เพื่อดูและเปลี่ยนข้อมูลสถิติต่างๆ และปรับเปลี่ยนส่วนขยายเคอร์เนล **SCTP** โดยใช้อ็อพชันอื่นๆ เช่น get และ set สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง **sctpctrl** โปรดดูคำอธิบายคำสั่ง **sctpctrl** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 5*

รูทีนย่อย sctp_bindx:

เพิ่มหรือลบแอดเดรสที่เชื่อมไว้บนซ็อกเก็ต

ไลบรารี

/usr/lib/libsctp.a

ไวยากรณ์

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/sctp.h>

int sctp_bindx(int sd, struct sockaddr * addrs, int addrcnt, int flags);
```

คำอธิบาย

รูทีนย่อย **sctp_bindx** เพิ่มหรือลบชุดของแอดเดรสที่เชื่อมกัน ซึ่งส่งผ่านในอาร์เรย์ **addrs** ไปยังหรือจากซ็อกเก็ต **sd** พารามิเตอร์ **addrcnt** คือจำนวนของแอดเดรสในอาร์เรย์ และพารามิเตอร์ **flags** ระบุว่าแอดเดรสต้องการ เพิ่มหรือลบออก

หากซ็อกเก็ต **sd** คือซ็อกเก็ต IPv4 แอดเดรสที่ส่งผ่าน ต้องเป็นแอดดรส IPv4 หากซ็อกเก็ต **sd** คือซ็อกเก็ต IPv6 แอดเดรสที่ส่งผ่านสามารถเป็นแอดเดรส IPv4 หรือ IPv6

พารามิเตอร์ **addrs** คือตัวชี้ไปยังอาร์เรย์ของซ็อกเก็ตแอดเดรส ตั้งแต่หนึ่งตัวขึ้นไป แต่ละแอดเดรสมีอยู่ในโครงสร้างที่เหมาะสม นั่นคือ **struct sockaddr_in** หรือ **struct sockaddr_in6** ตระกูลของชนิดแอดเดรสต้องใช้เพื่อแยก ความยาวของแอดเดรส ตัวเรียกระบุจำนวนของแอดเดรสในอาร์เรย์ พร้อมกับ **addrcnt**

พารามิเตอร์ **flags** สามารถเป็น **SCTP_BINDX_ADD_ADDR** or **SCTP_BINDX_REM_ADDR** แอ็พพลิเคชันสามารถใช้ SCTP_BINDX_ADD_ADDR เพื่อเชื่อมโยงกับแอดเดรสเพิ่มเติม ด้วยจุดปลายหลังการเรียกคำสั่ง **bind** พารามิเตอร์ **SCTP_BINDX_REM_ADDR** สั่งให้ SCTP ลบแอดเดรสที่กำหนดไว้จากการเชื่อมโยง ผู้เรียกต้องไม่ลบ แอดเดรสทั้งหมดออกจากการาเชื่อมโยง คำสั่งล้มเหลวมีผลต่อโค้ดระบุความผิดพลาด **EINVAL**

ค่าส่งคืน

เนื่องจากความสมบูณ์ที่ไม่สำเร็จ คำสั่ง **sctp_bindx( )** จะส่งคืน 0 สำหรับความล้มเหลว คำสั่ง **sctp_bindx( )** จะส่งคืน -1 และตั้งค่าพารามิเตอร์ **errno** ให้เป็น โค้ดระบุความผิดพลาดที่เหมาะสม

โค้ดระบุความผิดพลาด

ข้อผิดพลาด	คำอธิบาย
EINVAL	โค้ดระบุความผิดพลาด **EINVAL** บ่งชี้ว่าพอร์ตหรือแอดเดรสไม่ถูกต้องหรือคำสั่งกำลังลองลบ แอดเดรสทั้งหมดออกจากการเชื่อมโยง
EOPNOTSUPP	โค้ดระบุความผิดพลาด **EOPNOTSUPP** บ่งชี้ว่าคำสั่งกำลังพิมพ์เพื่อเพิ่มแอดเดรสจาก การเชื่อมต่อที่เชื่อมต่อ

รูทีนย่อย sctp_getladdrs และ sctp_freeladdrs:

ส่งคืนแอดเดรสที่เชื่อมไว้บนซ็อกเก็ต

ไลบรารี

```
/usr/lib/libsctp.a
```

ไวยากรณ์

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/sctp.h>

int sctp_getladdrs(int sd, sctp_assoc_t assoc_id, struct sockaddr **addrs);
void sctp_freeladdrs(struct sockaddr *addrs);
```

คำอธิบาย

รูทีนย่อย **sctp_getladdrs** ส่งคืนแอดเดรสที่เชื่อมไว้แบบโลคัล บนซ็อกเก็ต ในทางกลับกัน พารามิเตอร์ **addrs** จะชี้ไปยังอาร์เรย์ของโครงสร้าง **sockaddr** ของชนิดที่เหมาะสมที่แพ็กและจัดสรรแบบไดนามิกของชนิดที่เหมาะสมสำหรับแต่ละโลคัลแอดเดรส คุณต้องใช้พารามิเตอร์ **sctp_freeladdrs** เพื่อล้างหน่วยความจำ

หมายเหตุ: พารามิเตอร์ in หรือ out **addrs** ต้องไม่มีค่า NULL

หากพารามิเตอร์ **sd** คือซ็อกเก็ต IPv4 แอดเดรสที่ส่งคืน จะเป็นแอดเดรส IPv4 ทั้งหมด หากพารามิเตอร์ **sd** คือซ็อกเก็ต IPv6 แอดเดสที่ส่งคืนสามารถเป็นค่าผสมของแอดเดรส IPv4 หรือ IPv6

สำหรับซ็อกเก็ตที่มีลักษณะ one-to-many ฟิลด์ **id** จะระบุความสัมพันธ์ในการเคียวรี สำหรับลักษณะซ็อกเก็ต one-to-one ฟิลด์ **id** จะถูกละเว้น หากฟิลด์ **id** ตั้งค่าเป็น 0 แอดเดรสที่เชื่อมไว้แบบโลคัล จะถูกส่งคืนโดยไม่พิจารณาถึงความสัมพันธ์เฉพาะ

รูทีนย่อย **sctp_freeladdrs** ล้างข้อมูลรีซอร์สทั้งหมด ที่จัดสรรไว้โดยรูทีนย่อย **sctp_getladdrs**

ค่าส่งคืน

สำหรับความสำเร็จ รูทีนย่อย **sctp_getladdrs** จะส่งคืนจำนวนของแอดเดรสโลคัลที่เชื่อมกับซ็อกเก็ต หากซ็อกเก็ต ไม่ได้เชื่อมไว้ ค่า 0 จะถูกส่งคืนและค่าของฟิลด์ ***addrs** ไม่สามารถนิยามได้ สำหรับข้อผิดพลาด รูทีนย่อย **sctp_getladdrs** ส่งคืนค่า -1 และค่าของฟิลด์ ***addrs** ที่ไม่ได้นิยามไว้

รูทีนย่อย sctp_getpaddrs และ sctp_freepaddrs:

ส่งคืนแอดเดรสเพียร์ทั้งหมดในการเชื่อมโยง

ไลบรารี

```
/usr/lib/libsctp.a
```

ไวยากรณ์

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/sctp.h>

int sctp_getpaddrs(int sd, sctp_assoc_t assoc_id, struct sockaddr **addrs);
void sctp_freepaddrs(struct sockaddr *addrs);
```

คำอธิบาย

รูทีนย่อย **sctp_getpaddrs** ส่งคืนแอดเดรสเพียร์ทั้งหมดในการเชื่อมโยง ในทางกลับกัน พารามิเตอร์ **addrs** จะชี้ไปยังอาร์เรย์ของโครงสร้าง **sockaddr** ของชนิดที่เหมาะสมที่แพ็กและจัดสรรแบบไดนามิกของชนิดที่เหมาะสมสำหรับแต่ละแอดเดรส คุณต้องใช้รูทีนย่อย **sctp_freepaddrs** เพื่อล้างหน่วยความจำ

หมายเหตุ: พารามิเตอร์ in หรือ out **addrs** ต้องไม่มีค่า NULL

หากพารามิเตอร์ **sd** คือซ็อกเก็ต IPv4 แอดเดรสที่ส่งคืน จะเป็นแอดเดรส IPv4 ทั้งหมด หากพารามิเตอร์ **sd** คือซ็อกเก็ต IPv6 แอดเดรสที่ส่งคืนสามารถเป็นค่าผสมของแอดเดรส IPv4 หรือ IPv6 สำหรับซ็อกเก็ตที่มีลักษณะ one-to-many ฟิลด์ **id** จะระบุความสัมพันธ์ในการเคียวรี สำหรับลักษณะซ็อกเก็ต one-to-one ฟิลด์ **id** จะถูกละเว้น

รูทีนย่อย **sctp_freepaddrs** ล้างข้อมูลรีซอร์สทั้งหมดที่จัดสรรไว้โดยรูทีนย่อย **sctp_getpaddrs**

ค่าส่งคืน

สำหรับความสำเร็จ รูทีนย่อย **sctp_getpaddrs** ส่งคืนจำนวนของแอดเดรสเพียร์ในการเชื่อมโยง หากไม่มี การเชื่อมมโยงบนซ็อกเก็ตนี้ ค่า 0 จะถูกส่งคืนและค่าของฟิลด์ ***addrs** ไม่สามารถนิยามได้ สำหรับข้อผิดพลาด รูทีนย่อย **sctp_getpaddrs** ส่งคืนค่า -1 และค่าของฟิลด์ ***addrs** ไม่ได้นิยามไว้

การค้นพบพาธ MTU

สำหรับโฮสต์สองรายการที่สื่อสารระหว่างกันบนพาธที่ประกอบด้วยหลายเครือข่าย แพ็กเก็ตที่ส่งผ่านจะมีการแบ่งแฟรกเมนต์ ถ้าขนาดของแพ็กเก็ตนั้นใหญ่กว่า MTU ที่เล็กที่สุดของเครือข่ายใดๆ ในพาธ เนื่องจากการแบ่งแฟรกเมนต์แพ็กเก็ตอาจส่งผลให้ประสิทธิภาพของ เครือข่ายลดลง จึงควรหลีกเลี่ยงการแบ่งแฟรกเมนต์โดยการส่งผ่านแพ็กเก็ต ที่มีขนาดไม่ใหญ่เกินกว่า MTU ที่เล็กที่สุดในพาธเครือข่าย ขนาดนี้เรียกว่าพาธ MTU

ระบบปฏิบัติการสนับสนุนอัลกอริทึมการค้นพบพาธ MTU ตามที่กล่าวไว้ใน RFC 1191 การค้นพบพาธ MTU สามารถเปิดใช้งานสำหรับแอ็พพลิเคชัน **TCP** และ **UDP** โดยแก้ไขอ็อพชัน **tcp_pmtu_discover** และ **udp_pmtu_discover** ของคำสั่ง **no** เมื่อเปิดใช้งานสำหรับ **TCP** การค้นพบพาธ MTU จะบังคับให้ใช้ขนาดของแพ็กเก็ตทั้งหมดแบบอัตโนมัติที่ส่งผ่านโดย **TCP** ซึ่งไม่เกินกว่าค่าของพาธ MTU เนื่องจากแอ็พพลิเคชัน **UDP** พิจารณาขนาดของแพ็กเก็ตที่ส่งผ่าน แอ็พพลิเคชัน **UDP** ต้องเขียนเพื่อใช้ข้อมูลพาธ MTU โดยใช้อ็อพชันซ็อกเก็ต **IP_FINDPMTU** แม้ว่า อ็อพชัน **udp_pmtu_discover no** ถูกเปิดใช้งาน โดยดีฟอลต์ **tcp_pmtu_discover** และ **udp_pmtu_discover** มีการ เปิดใช้งาน

เมื่อมีความพยายามค้นหา Path MTU สำหรับปลายทาง รายการ pmtu จะถูกสร้างขึ้นในตาราง Path MTU (PMTU) ตารางนี้สามารถแสดงได้โดยใช้คำสั่งที่แสดง **pmtu** รายการ pmtu ที่สะสมสามารถหลีกเลี่ยงได้โดยอนุญาตให้ใช้รายการ pmtu ที่ยังไม่ใช่เพื่อให้หมดอายุและถูกลบทิ้ง รายการ PMTU ที่หมดอายุถูกควบคุมโดยอ็อพชัน pmtu_expire ของคำสั่ง **no** pmtu_expire ถูกตั้งค่าเป็น 10 นาทีตามค่าดีฟอลต์

เนื่องจากเราต์สามารถเปลี่ยนแปลงแบบไดนามิก ค่าพาธ MTU สำหรับพาธอาจเปลี่ยนแปลง ผ่านช่วงเวลา ลดจำนวนลงในค่าพาธ MTU จะส่งผลให้การแตกแฟรกเมนต์แพ็กเก็ต ดังนั้น ค่าพาธ MTU ที่ค้นพบถูกตรวจสอบสำหรับ การลดจำนวนลง ตามค่าดีฟอลต์ การลดลงตรวจสอบสำหรับ 10 นาที และค่นี้สามารถเปลี่ยนไปเป็นการแก้ไขค่าของอ็อพชัน **pmtu_default_age** ของคำสั่ง **no**

แอ็พพลิเคชัน **UDP** จำเป็นต้องตั้งค่าอ็อพชันซ็อกเก็ต **IP_DONTFRAG** เสมอ เพื่อให้พบการลดลงใน PMTU ซึ่งจะเปิดใช้งานการตรวจพบในทันทีของการลดจำนวนลงใน Path MTU แทนการตรวจสอบการลดจำนวนลงทุกๆ pmtu_default_age นาที

การเพิ่มในค่าพาธ MTU สามารถส่งผลทำให้การเพิ่มในผลการทำงานของเน็ตเวิร์ก ดังนั้น ค่าพาธ MTU ที่ค้นพบจะถูกตรวจสอบสำหรับการเพิ่มเติม ตามค่าดีฟอลต์ การเพิ่มตรวจสอบสำหรับ 30 นาที และค่นี้สามารถเปลี่ยนไปเป็นการแก้ไขค่าของอ็อพชัน **pmtu_rediscover_interval** ของคำสั่ง **no**

หากไม่ใช่เราต์ทั้งหมดในพาธของเน็ตเวิร์กสนับสนุน RFC 1191 เราต์นั้นจะไม่ถูกพิจารณาเป็นค่าพาธ MTU ในกรณีเหล่านี้ คำสั่ง **mmtu** สามารถใช้เพื่อเพิ่มหรือลบค่าพาธ MTU ที่พยายาม

หมายเหตุ:

1. การค้นพบพาธ MTU ไม่สามารถใช้บนเราต์ที่ซ้ำกัน ซึ่งประกอบด้วยการตั้งค่าเหล่านี้ สำหรับการเราต์ของกลุ่ม (โปรดดู “ข้อจำกัดการใช้เส้นทาง” ในหน้า 383) การค้นหา Path MTU สามารถใช้บนเราต์ที่ซ้ำกันได้
2. การเปิดใช้งานการค้นพบพาธ MTU ตั้งค่าของอ็อพชัน **arpqsize** ของคำสั่ง **no** ไปเป็นค่าต่ำสุดของ 5 ค่านี้ไม่ได้ลดจำนวนลง หากพาธไม่ใช่การค้นพบพาธ MTU ถูกปิดใช้งาน

TCP/IP Quality of Service

Quality of Service (QoS) คือ family ของการพัฒนา Internet มาตรฐาน ที่จัดเตรียมวิธีที่กำหนดการใช้เป็นพิเศษกับชนิดของทราฟิก IP

ด้วยส่วนสนับสนุนสำหรับ QoS พร้อมกับเราต์ สามารถทำให้ดีขึ้น ซึ่งเป็นผลกระทบของตัวแปรที่จัดคิวหน่วงเวลาและล้มเหลว ซึ่งสร้างผลการทำงาน เน็ตเวิร์กที่ช้า ระบบปฏิบัติการจัดเตรียมส่วนสนับสนุนโฮสต์สำหรับ QoS เพื่อจัดหมวดหมู่ทราฟิกเป็นคลาสของเซอร์วิสและประกาศ และสร้างการจองรีซอร์สตามคำร้องขอโดยไคลเอ็นต์แอ็พพลิเคชัน

QoS สามารถใช้โดยองค์กรเพื่อนำไปใช้งานและบังคับนโยบายเน็ตเวิร์กที่กำหนดการใช้ ของแบนด์วิดธ์ของเน็ตเวิร์ก ด้วย QoS โฮสต์สามารถ:

- ควบคุมจำนวนของทราฟิกของชนิดที่ติดกับเน็ตเวิร์ก
- ทำเครื่องหมายแพ็กเก็ตที่เลือกไว้ตามนโยบายบางส่วน ดังนั้นเราเตอร์ตามลำดับ ที่สามารถส่งผ่านเซอร์วิสที่บ่งชี้ได้
- สนับสนุนเซอร์วิสเช่น เซอร์วิสสายเช่าเสมือนด้วยส่วนสนับสนุน QoS ที่ถูกต้องพร้อมกับเราต์ และ
- ทำงานร่วมกันในคำร้องขอการจองรีซอร์สจาก receivers และประกาศ เซสชันผู้ส่งที่พร้อมใช้งานสำหรับคำร้องขอการจองรีซอร์ส

ส่วนสนับสนุน QoS จัดเตรียมฟังก์ชันต่อไปนี้:

- เซอร์วิสที่แตกต่างกันตามที่กำหนดไว้ใน RFC 2474
- นโยบายทราฟิก
- การทำเครื่องหมายแพ็กเก็ต In-profile และ out-of-profile
- การจัดรูปแบบทราฟิก
- การวัด
- เซอร์วิสรวมสำหรับแอ็พพลิเคชันไคลเอ็นต์และเซิร์ฟเวอร์ตามที่นิยามอยู่ใน RFC 1633
- การส่งสัญญาณ RSVP (RFC 2205)
- การรับประกันเซอร์วิส (RFC 2212)
- เซอร์วิส Controlled-Load (RFC 2211)
- การวางเน็ตเวิร์กแบบอิงนโยบาย
- ไลบรารี RAPI ที่แบ่งใช้สำหรับแอ็พพลิเคชัน

ระบบย่อย QoS ที่ประกอบด้วยสี่คอมโพเนนต์:

ส่วนขยายเคอร์เนล QoS (`/usr/lib/drivers/qos`)
ส่วนขยายเคอร์เนล QoS ที่ตั้งอยู่ใน `/usr/lib/drivers/qos` และถูกโหลดและยกเลิกการโหลดโดยใช้เมธอด **cfgqos** and **ucfgqos** ส่วนขยายเคอร์เนลนี้เปิดใช้งานการสนับสนุน QoS

เอเจนต์นโยบาย (`/usr/sbin/policyd`)
เอเจนต์นโยบายคือ daemon ระดับผู้ใช้ที่ตั้งอยู่ใน `/usr/sbin/policyd` ซึ่งจัดเตรียมส่วนสนับสนุนสำหรับการจัดการกับนโยบายและอินเตอร์เฟสกับส่วนขยายเคอร์เนล QoS เพื่อติดตั้ง แก้ไข และลบกฎของนโยบาย กฎของนโยบายสามารถนิยามอยู่ใน ไฟล์คอนฟิกูเรชันบนโลคัล (`/etc/policyd.conf`) ซึ่งเรียกคืนจาก central network policy server โดยใช้ LDAP หรือทั้งสอง

เอเจนต์ RSVP (`/usr/sbin/rsvpd`)
เอเจนต์ RSVP คือ daemon ระดับผู้ใช้ที่ตั้งอยู่ใน `/usr/sbin/rsvpd` ซึ่งใช้ซีแมนทิกส์โปรโตคอลการส่งสัญญาณ RSVP

ไลบรารี RAPI ที่แบ่งใช้ (`/usr/lib/librapi.a`)
แอ็พพลิเคชันสามารถใช้ RSVP API (RAPI) เพื่อร้องขอ quality of service ที่ปรับปรุงแล้วตามที่นิยามโดยโมเดล

Integrated Services Internet QoS ไลบรารีนี้โต้ตอบกลับเอเจนต์ RSVP บนโลคัลเพื่อกระจายคำร้องขอ QoS พร้อมกับพาธของการไหลของข้อมูลโดยใช้โปรโตคอล RSVP API นี้คือการเปิดมาตรฐาน

หมายเหตุ: การนำไปใช้งานของ QoS นี้อ้างอิงตามชุดของอินเตอร์เน็ตมาตรฐานที่ได้รับการพัฒนา และเป็นแบบร่างมาตรฐานที่อยู่ภายใต้การพัฒนาโดย Internet Engineering Task Force (IETF) และกลุ่มการทำงานต่างๆ เทคโนโลยีนี้ สอดคล้องกันมากและนิยามไว้เป็นมาตรฐานในการดำเนินการภายใน IETF และเป็นสิ่งสำคัญในการจดบันทึกว่า QoS คือจุดกำเนิดของเทคโนโลยีอินเตอร์เน็ต ที่เพิ่งเริ่มต้นนำไปใช้งานภายในอินเตอร์เน็ต มีประโยชน์มากมายของ QoS ที่ทุกขั้นตอนของการนำไปใช้งาน อย่างไรก็ตาม เป็นจริงที่ว่า เซอร์วิส end-to-end สามารถจำแนกได้ เมื่อส่วนสนับสนุน QoS มีอยู่เฉพาะเราต์

โมเดล QoS

โมเดล QoS สำหรับอินเตอร์เน็ตเป็นมาตรฐานเปิด ซึ่งกำหนดโดย IETF

มีโมเดล QoS อินเตอร์เน็ตสองแบบเป็นมาตรฐานปัจจุบันภายใน IETF: *integrated services* และ *differentiated services* โมเดล QoS อินเตอร์เน็ตสองแบบเหล่านี้เพิ่มเซอร์วิสโมเดลที่มีความพยายามดี ที่อธิบายใน RFC 1812

เซอร์วิสแบบรวม:

Integrated Services (IS) คือโมเดลการสำรองรีซอร์สไดนามิก สำหรับอินเตอร์เน็ตที่กล่าวถึงใน RFC 1633

โฮสต์ใช้การส่งสัญญาณเรียกว่า Resource ReSerVation Protocol (RSVP) เพื่อร้องขอ quality of service ที่ระบุไว้จากเน็ตเวิร์ก พารามิเตอร์ QoS ถูกใช้ในข้อความ RSVP เหล่านี้และโหนดของเน็ตเวิร์กแต่ละวง พร้อมกับพาธการติดตั้งพารามิเตอร์เพื่อขอรับ quality of service ที่ร้องขอ พารามิเตอร์ QoS กล่าวถึงหนึ่งในสองเซอร์วิสที่กำหนดไว้ในปัจจุบัน ซึ่งเซอร์วิสรับประกัน และเซอร์วิสการโหลดที่ควบคุม คุณสมบัติที่สำคัญของ IS นั่นคือ การส่งสัญญาณนี้ถูกทำสำหรับการไฟลของทราฟิกและการสำรองถูกติดตั้งอยู่ที่ hop แต่ละ hop พร้อมกับเราต์ แม้ว่า โมเดลนี้เหมาะสมเป็นอย่างดีสำหรับการประชุม การเปลี่ยนแปลงความต้องการแบบไดนามิกของแอ็พพลิเคชัน ซึ่งมีการวัดที่สำคัญอยู่ ที่หมายความว่า ไม่สามารถนำไปใช้ในเน็ตเวิร์กในเราเตอร์เดี่ยว ที่จัดการกับการไหลแบบพร้อมเพียงกัน

เซอร์วิสที่แตกต่างกัน:

Differentiated Services (DS) ลบปัญหาเกี่ยวกับความสามารถในการวัดต่อการไหลหรือต่อ hop การวางปัญหาต่างๆ กับกลไกแบบง่ายๆ ของการแบ่งประเภทแพ็กเก็ต

ด้วยการใช้วิธีการส่งสัญญาณแบบไดนามิกแทน DS ใช้บิตในชนิด IP ของ เซอร์วิส (TOS) ไบต์ เพื่อแยกแพ็กเก็ตลงในคลาสต่างๆ รูปแบบบิตเฉพาะ ใน IP TOS ไบต์ถูกเรียกใช้ DS codepoint และถูกใช้เราเตอร์เพื่อนิยาม quality of service ที่จัดส่งไปที่ hop เฉพาะ ซึ่งมากเท่ากับวิธีของเราเตอร์ เพื่อทำการส่งต่อ IP โดยใช้การมองหาตารางการเราเตอร์ การใช้ที่กำหนดให้กับแพ็กเก็ตด้วย DS codepoint โดยเฉพาะถูกเรียกใช้ต่อ per-hop behavior (PHB) และถูกดูแลระบบอย่างเป็นอิสระที่โหนดเน็ตเวิร์กแต่ละครั้ง เมื่อกระทบกับสิ่งเหล่านี้ PHB อิสระถูกเชื่อมต่อเข้าด้วยกัน ผลลัพธ์นี้ในเซอร์วิส end-to-end

เซอร์วิสที่แตกต่างกันจะเป็นมาตรฐานโดยกลุ่มการทำงาน IETF ซึ่งถูกนิยามไว้เป็น PHB สามตัว: the Expedited Forwarding (EF) PHB, กลุ่ม Assured Forwarding (AF) PHB group และ Default (DE) PHB EF PHB สามารถใช้ เพื่อนำเวลาแฝงต่ำสุดไปใช้ การสูญหายต่ำสุด เซอร์วิสแบบ end-to-end เช่น virtual leased line (VLL) AF คือ family ของ PHBs เรียกใช้กลุ่ม PHB ซึ่งถูกใช้เพื่อจัดหมวดหมู่แพ็กเก็ตลงในระดับของการมาก่อนปล่อย การปล่อยการนำหน้า ที่กำหนดให้กับแพ็กเก็ตพิจารณาความสำคัญที่เกี่ยวข้องของแพ็กเก็ต ภายในคลาส AF ซึ่งสามารถใช้เพื่อนำมาใช้ ซึ่งเรียกว่าเซอร์วิส *Olympic* ซึ่งประกอบด้วยคลาสสามคลาสคือ: bronze, silver และ gold DE PHB คือโมเดลเซอร์วิสที่สนับสนุนที่ดีที่สุดเป็นมาตรฐานใน RFC 1812

มาตรฐานที่ได้รับการสนับสนุนและมาตรฐานแบบร่าง

RFCs เหล่านี้และอินเตอร์เน็ตแบบร่างจะอธิบายมาตรฐานเกี่ยวกับพื้นฐานของการใช้ QoS นี้

ไอเท็ม	คำอธิบาย
RFC 2474	นิยามของบ Differentiated Services Field (DS Field) ในส่วนหัว IPv4 และ IPv6
RFC 2475	สถาปัตยกรรมสำหรับ Differentiated Services
RFC 1633	Integrated Services ใน Internet Architecture: ภาพรวม
RFC 2205	Resource ReSerVation Protocol (RSVP)
RFC 2210	การใช้ RSVP พร้อมกับ IETF Integrated Services
RFC 2211	ข้อกำหนดคุณสมบัติของ Controlled-Load Network Element Service
RFC 2212	ข้อกำหนดคุณสมบัติของ Guaranteed Quality of Service
RFC 2215	General Characterization Parameters สำหรับ Integrated Service Network Elements

ไอเท็ม	คำอธิบาย
draft-ietf-diffserv-framework-01.txt, ตุลาคม 1998	กรอบงานสำหรับ Differentiated Services
draft-ietf-diffserv-rsvp-01.txt, พฤศจิกายน 1998	กรอบงานการใช้ RSVP พร้อมกับ DIFF-serv Networks
draft-ietf-diffserv-phb-ef-01.txt	Expedited Forwarding PHB
draft-ietf-diffserv-af-04.txt	Assured Forwarding PHB Group
draft-rajan-policy-qosschema-00.txt, ตุลาคม 1998	Schema สำหรับ Differentiated Services และ Integrated Services in Networks
draft-ietf-rap-framework-01.txt, พฤศจิกายน 1998	กรอบงานสำหรับ Policy-based Admission Control[25]
draft-ietf-rap-rsvp-ext-01.txt, พฤศจิกายน 1998	RSVP Extensions for Policy Control

หมายเหตุ: QoS เป็นอินเตอร์เน็ตเทคโนโลยีที่เกิดใหม่ มีประโยชน์หลายอย่างของการใช้ QoS อย่างไรก็ตาม เซอร์วิสแบบจุดต่อจุดที่แท้จริง สามารถทราบได้เมื่อใช้ QoS กับเส้นทางทั้งหมด

การติดตั้ง QoS

QoS ถูกจัดแพ็กเกจด้วย `bos.net.tcp.server` ชุดไฟล์นี้ต้องถูกติดตั้งเพื่อใช้งาน QoS

เมื่อต้องการใช้ไลบรารีที่แบ่งใช้ RAPI `bos.adt.include` ต้องถูกติดตั้งไว้

การหยุดและการสตาร์ทระบบย่อย QoS

QoS สามารถสตาร์ทหรือหยุดทำงานผ่าน SMIT พร้อมกับวิธีลัด `smit qos` หรือพร้อมกับคำสั่ง **mkqos** และ **rmqos**

1. หากต้องการปิดใช้งานระบบย่อย QoS เดี๋ยวนี้และบนระบบถัดไปที่รีสตาร์ท:

   ```
   /usr/sbin/rmqos -B
   ```

2. หากต้องการเปิดใช้งานระบบย่อย QoS เดี๋ยวนี้เท่านั้น:

   ```
   /usr/sbin/mkqos -N
   ```

โปรดดูคำอธิบายคำสั่งสำหรับ **mkqos** และ **rmqos** สำหรับการเริ่มต้นทำงาน และลบแฟล็กคำสั่ง

policyd และ **rsvpd** daemon ถูกตั้งค่าไว้ผ่านไฟล์คอนฟิกูเรชันที่ชื่อ `/etc/policyd.conf` and `/etc/rsvpd.conf` ตามลำดับ ไฟล์คอนฟิกูเรชันเหล่านี้ *ต้อง* ถูกแก้ไข เพื่อปรับแต่งระบบย่อย QoS กับสภาพแวดล้อมโลคัล QoS ไม่ได้ทำงานอย่างถูกต้องกับคอนฟิกูเรชันตัวอย่างที่จัดหา

คอนฟิกูเรชันเอเจนต์ RSVP

เอเจนต์ RSVP จำเป็นต้องมีหากโฮสต์คือการสนับสนุนโปรโตคอล RSVP

ไฟล์คอนฟิกูเรชันของ /etc/rsvpd.conf ถูกใช้เพื่อตั้งค่าเอเจนต์ RSVP ไวยากรณ์ของไฟล์คอนฟิกูเรชัน ถูกกล่าวถึงในตัวอย่างไฟล์คอนฟิกูเรชันที่ติดตั้งอยู่ใน /etc/rsvpd.conf

ตัวอย่างต่อไปนี้แสดงให้เห็นถึงคอนฟิกูเรชัน RSVP ที่เป็นไปได้ซึ่ง โฮสต์มี 4 อินเตอร์เฟส (เสมือนหรือฟิสิคัล) ที่กำหนดโดย 4 IP แอดเดรส, 1.2.3.1, 1.2.3.2, 1.2.3.3, และ 1.2.3.4

```
interface 1.2.3.1
interface 1.2.3.2 disabled
interface 1.2.3.3 disabled
interface 1.2.3.4
{
  trafficControl
}

rsvp 1.2.3.1
{
  maxFlows 64
}

rsvp 1.2.3.4
{
  maxFlows 100
}
```

อินเตอร์เฟส 1.2.3.1 เปิดใช้งานสำหรับ RSVP อย่างไรก็ตาม การควบคุมทราฟิก ไม่ได้ระบุไว้และข้อความ RSVP RESV ขาเข้าไม่ได้เป็นสาเหตุ ทำให้จองรีซอร์สภายในระบบย่อย TCP อินเตอร์เฟสนี้สามารถสนับสนุน ได้สูงสุด 64 RSVP เซสชันแบบพร้อมเพียงกัน

อินเตอร์เฟส 1.2.3.2 และ 1.2.3.3 ปิดใช้งาน เอเจนต์ RSVP ไม่สามารถใช้อินเตอร์เฟสนี้เพื่อส่งผ่านหรือรับข้อความ RSVP

อินเตอร์เฟส 1.2.3.4 เปิดใช้งานสำหรับ RSVP นอกจากนี้ ยังสามารถติดตั้งการจองรีซอร์สลงในระบบย่อย **TCP** ในการตอบกลับไปยัง ข้อความ RSVP RESV อินเตอร์เฟสนี้สามารถสนับสนุนได้สูงสุด 100 RSVP เซสชัน

อินเตอร์เฟสอื่นใดๆ ที่แสดงอยู่บนโฮสต์ไม่ได้กล่าวถึงใน /etc/rsvpd.conf ถูกปิดใช้งาน

การตั้งค่าเอเจนต์นโยบาย

เอเจนต์นโยบายเป็นส่วนประกอบที่ต้องการของระบบย่อย QoS

ไฟล์คอนฟิกูเรชัน /etc/policyd.conf ถูกใช้เพื่อตั้งค่าเอเจนต์นโยบาย ไวยากรณ์ของไฟล์คอนฟิกูเรชันถูกอธิบายในไฟล์คอนฟิกูเรชันตัวอย่างที่ถูกติดตั้งใน /etc/policyd.conf

เอเจนต์นโยบายสามารถถูกกำหนดคอนฟิกโดยการแก้ไข /etc/policyd.conf นอกจากนี้ คำสั่งต่อไปนี้ถูกจัดเตรียมเพื่อช่วยเหลือในการกำหนดคอนฟิกนโยบาย :

- **qosadd**
- **qosmod**

- **qoslist**
- **qosremove**

ในตัวอย่างต่อไปนี้ หมวดหมู่ของเซอร์วิสคุณภาพสูง ถูกสร้างและใช้ในกฎ tcptraffic policy หมวดหมู่ของเซอร์วิสนี้มีอัตราสูงสุดเป็น 110000 Kbps, token bucket depth เป็น 10000 bits และค่า outgoing IP TOS เป็น 11100000 ในไบนารี กฎ tcptraffic policy ในเซอร์วิสคุณภาพสูงกับทราฟิกทั้งหมดด้วย IP แอดเดรสต้นทางที่ถูกให้โดย 1.2.3.6, แอดเดรสปลายทาง 1.2.3.3 และพอร์ตปลายทางในช่วง 0 ถึง 1024

```
ServiceCategories  premium
{
   PolicyScope     DataTraffic
   MaxRate         110000
   MaxTokenBucket  10000
   OutgoingTOS     11100000
}

ServicePolicyRules   tcptraffic
{
   PolicyScope   DataTraffic
   ProtocolNumber 6  # tcp
   SourceAddressRange      1.2.3.6-1.2.3.6
   DestinationAddressRange 1.2.3.3-1.2.3.3
   DestinationPortRange    0-1024
   ServiceReference        premium
}
```

คำสั่งต่อไปนี้จะตั้งหมวดหมู่เซอร์วิสดีฟอลต์และใช้มันเพื่อจำกัดการไหลของ UDP ทราฟิกจากอินเตอร์เฟส 1.2.3.1 ผ่าน 1.2.3.4 ไปยัง IP แอดเดรส 1.2.3.6 ถึง 1.2.3.10 พอร์ต 8000

```
ServiceCategories  default
{
   MaxRate         110000
   MaxTokenBucket  10000
   OutgoingTOS     00000000
}

ServicePolicyRules   udptraffic
{
   ProtocolNumber 17  # udp
   SourceAddressRange      1.2.3.1-1.2.3.4
   DestinationAddressRange 1.2.3.6-1.2.3.10
   DestinationPortRange    8000-8000
   ServiceReference        default
}
```

ตัวอย่างการตั้งค่าต่อไปนี้สามารถถูกใช้เพื่อดาวน์โหลดกฎจาก LDAP เซิร์ฟเวอร์โดยใช้ชื่อทรีย่อยที่แตกต่าง เพื่อค้นดูนโยบายบนโฮสต์ LDAP เซิร์ฟเวอร์

```
ReadFromDirectory
{
  LDAP_Server      1.2.3.27
  Base             ou=NetworkPolicies,o=myhost.mydomain.com,c=us
}
```

การแก้ปัญหา QoS

คำสั่ง **qosstat** อาจถูกใช้เพื่อแสดงข้อมูลสถานะ เกี่ยวกับการติดตั้งและนโยบายที่แอ็คทีฟในระบบย่อย QoS ข้อมูลนี้อาจมีประโยชน์ต่อคุณในการพิจารณาโดยที่ปัญหามีอยู่ หากคุณกำลังแก้ไขคอนฟิกูเรชัน QoS ของคุณ

qosstat สามารถใช้เพื่อสร้างรายงานต่อไปนี้

```
Action:
  Token bucket rate (B/sec): 10240
  Token bucket depth (B): 1024
  Peak rate (B/sec): 10240
  Min policied unit (B): 20
  Max packet size (B): 1452
  Type: IS-CL
  Flags: 0x00001001 (POLICE,SHAPE)

  Statistics:
    Compliant packets: 1423 (440538 bytes)

  Conditions:
    Source address        Dest address          Protocol
    192.168.127.39:8000   192.168.256.29:35049  tcp       (1 connection)

Action:
  Token bucket rate (B/sec): 10240
  Token bucket depth (B): 1024
  Peak rate (B/sec): 10240
  Outgoing TOS (compliant): 0xc0
  Outgoing TOS (non-compliant): 0x00
  Flags: 0x00001011 (POLICE,MARK)
  Type: DS

  Statistics:
    Compliant packets: 335172 (20721355 bytes)
    Non-compliant packets: 5629 (187719 bytes)

  Conditions:
    Source address        Dest address          Protocol
    192.168.127.39:80     *:*                   tcp       (1 connection)
    192.168.127.40:80     *:*                   tcp       (5 connections)
```

ข้อกำหนดคุณสมบัติของนโยบาย QoS

คลาสอ็อบเจ็กต์และแอ็ตทริบิวต์ถูกใช้โดยนโยบายเอเจนต์ เพื่อระบุนโยบายสำหรับ quality of service (QoS) บนทราฟิกขาออกตามที่กล่าวไว้ที่นี่

คลาสอ็อบเจ็กต์และแอ็ตทริบิวต์ได้ถูกนิยามไว้ ตามด้วยคำแนะนำในการเปิดใช้งานการทำเครื่องหมาย การสร้างนโยบาย และการจัดแต่งรูปร่าง

ระเบียบเหล่านี้ถูกใช้เพื่ออธิบายให้เห็นภาพดังนี้

```
p  : choose one in the allowed parameter set
B  : integer value of a byte (i.e., 0 =< B =< 255)
b  : bit string starting with left most bit (e.g., 101 is
     equivalent 10100000 in a byte field)
```

```
 i  : integer value
 s  : a character string
 a  : IP address format B.B.B.B
(R) : Required parameter
(O) : Optional parameter
```

คำสั่ง ReadFromDirectory:

คำสั่งนี้ระบุพารามิเตอร์สำหรับการสร้างเซสชัน LDAP

คำสั่ง `ReadFromDirectory` ถูกใช้ในไฟล์ `/etc/policyd.conf` เพื่อสร้างเซสชัน LDAP

```
ReadFromDirectory
{
  LDAP_Server    a   # IP address of directory server running LDAP
  LDAP_Port      i   # Port number LDAP server is listening to
  Base     s         # Distinguished Name for LDAP usage
  LDAP_SelectedTag s # Tag to match SelectorTag in object classes
}
```

where

```
LDAP_Server (R): IP address of LDAP server
LDAP_Port   (O): Unique port number, default port is 389
Base        (R): Example is o=ibm, c=us where o is your organization and c is country
LDAP_SelectedTag (R): Unique string matching SelectorTag attribute in the object class
```

คำสั่ง ServiceCategories:

คำสั่งนี้ระบุชนิดของเซอร์วิสที่การไหลของแพ็กเก็ต IP (ตัวอย่างเช่น จากการเชื่อมต่อ **TCP** หรือข้อมูล **UDP**) ควรรับแบบปลายต่อปลายตามที่ข้ามผ่านเน็ตเวิร์ก

`ServiceCategories` สามารถทำซ้ำด้วยชื่อที่แตกต่างกัน เพื่อให้สามารถอ้างอิงได้ในภายหลัง อ็อบเจ็กต์ `ServiceCategories` จำเป็นต้องมี `ServicePolicyRules` เพื่อให้เสร็จสิ้นนิยามของนโยบาย

```
ServiceCategories   s
{
   SelectorTag    s     # Required tag for LDAP Search
   MaxRate        i     # Target rate for traffic in this service class
   MaxTokenBucket i     # The bucket depth
   OutgoingTOS    b     # TOS value of outbound traffic for this service class
   FlowServiceType   p  # Type of traffic

}
```

where

```
s           (R)     : is the name of this service category
SelectorTag (R)     : Required only for LDAP to Search object classes
MaxRate     (O)     : in Kbps (K bits per second), default is 0
MaxTokenBucket(O)   : in Kb, default is system defined maximum
OutgoingTOS (O)     : default is 0
FlowServiceType  (O): ControlledLoad | Guaranteed, default is ControlledLoad
```

คำสั่ง ServicePolicyRules:

คำสั่งนี้ระบุคุณสมบัติของ IP แพ็กเก็ตที่ถูกใช้เพื่อให้จับคู่กับ หมวหมู่การให้บริการที่สอดคล้องกัน

อีกนัยหนึ่ง ซึ่งนิยามชุดของ IP datagram ที่ควรรับเซอร์วิส เฉพาะ แอ็ตทริบิวต์ ServicePolicyRules ถูกเชื่อมโยงกับ ServiceCategories ผ่านแอ็ตทริบิวต์ ServiceReference หากกฎสองกฎอ้างถึง ServiceCategory เดี่ยวกัน กฎแต่ละกฎเชื่อมโยงกับอินสแตนซ์เฉพาะของ ServiceCategory

```
ServicePolicyRules   s
{
   SelectorTag       s   # Required tag for LDAP Search
   ProtocolNumber    i   # Transport protocol id for the policy rule
   SourceAddressRange      a1-a2
   DestinationAddressRange a1-a2
   SourcePortRange         i1-i2
   DestinationPortRange    i1-i2
   PolicyRulePriority  i       # Highest value is enforced first
   ServiceReference        s  # Service category name which for this policy rule
}
```

where

```
s           (R): is the name of this policy rule
SelectorTag (R): required only for LDAP to Search object class
ProtocolNumber     (R): default is 0 which causes no match, must explicity specify
SourceAddressRange (O): from a1 to a2 where a2 >= a1, default is 0, any source address
SourcePortRange    (O): from i1 to i2 where i2 >= i1, default is 0, any source port
DestinationAddressRange (O): same as SourceAddressRange
DestinationPortRange    (O): same as SourcePortRange
PolicyRulePriority (O): Important to specify when ovelapping policies exist
ServiceReference   (R): service category this rule uses
```

คำแนะนำสำหรับสภาพแวดล้อม DiffServ

ต่อไปนี้คือคำแนะนำในการระบุนโยบายสำหรับการทำเครื่องหมาย การเปลี่ยนรูปร่าง และ/หรือการสร้างนโยบายในสภาพแวดล้อมแบบ DiffServ

1. **การทำเครื่องหมายเท่านั้น**

   ```
   OutgoingTOS     : Desired Type Of Service
   FlowServiceType : ControlledLoad
   MaxRate         : Take default of 0
   ```

2. **การจัดรูปร่างเท่านั้น**

   ```
   OutgoingTOS     : Take default of 0
   FlowServiceType : Guaranteed
   MaxRate         : Target rate desired for traffic as a positive integer
   ```

3. **การทำเครื่องหมายและนโยบาย (โปรดดูหมายเหตุ)**

   ```
   OutgoingTOS     : Desired Type of Service
   FlowServiceType : ControlledLoad
   MaxRate         : Target rate desired for traffic as a positive integer
   ```

4. **การทำเครื่องหมายและการจัดรูปแบบ**

```
        OutgoingTOS     : Desired Type of Service
        FlowServiceType : Guaranteed
        MaxRate         : Target rate desired for traffic as a positive integer
```

หมายเหตุ: ชนิดของชุดเซอร์วิสสำหรับแพ็กเก็ตโปรไฟล์ที่ไม่มีอยู่ถูกตั้งค่าให้เป็นศูนย์ในกรณีของการสร้างนโยบาย

ตัวอย่างไฟล์คอนฟิกูเรชัน policyd

นี่เป็นตัวอย่างที่สมบูรณ์ของไฟล์คอนฟิกูเรชัน /etc/policyd.conf

```
#loglevel  511    # Verbose logging

##############################################################################
#
# Mark rsh traffic on TCP source ports 513 and 514.
ServiceCategories       tcp_513_514_svc
{
        MaxRate                 0               # Mark only
        OutgoingTOS             00011100        # binary
        FlowServiceType         ControlledLoad
}

ServicePolicyRules      tcp_513_514_flt
{
        ProtocolNumber          6  # TCP
        SourceAddressRange      0.0.0.0-0.0.0.0 # Any IP src addr
        DestinationAddressRange 0.0.0.0-0.0.0.0 # Any IP dst addr
        SourcePortRange         513-514
        DestinationPortRange    0-0             # Any dst port
        ServiceReference        tcp_513_514_svc
}
#
##############################################################################
#
# Shape connected UDP traffic on source port 9000.
ServiceCategories       udp_9000_svc
{
        MaxRate                 8192     # kilobits
        MaxTokenBucket          64       # kilobits
        FlowServiceType         Guaranteed
}

ServicePolicyRules      udp_9000_flt
{
        ProtocolNumber          17  # UDP
        SourceAddressRange      0.0.0.0-0.0.0.0 # Any IP src addr
        DestinationAddressRange 0.0.0.0-0.0.0.0 # Any IP dst addr
        SourcePortRange         9000-9000
        DestinationPortRange    0-0             # Any dst port
        ServiceReference        udp_9000_svc
}
#
##############################################################################
#
# Mark and police finger traffic on TCP source port 79.
ServiceCategories       tcp_79_svc
```

```
{
        MaxRate                 8         # kilobits
        MaxTokenBucket          32        # kilobits
        OutgoingTOS             00011100  # binary
        FlowServiceType         ControlledLoad
}

ServicePolicyRules      tcp_79_flt
{
        ProtocolNumber          6  # TCP
        SourceAddressRange      0.0.0.0-0.0.0.0 # Any IP src addr
        DestinationAddressRange 0.0.0.0-0.0.0.0 # Any IP dst addr
        SourcePortRange         79-79
        DestinationPortRange    0-0             # Any dst port
        ServiceReference        tcp_79_svc
}
#
##############################################################################
#
# Mark and shape ftp-data traffic on TCP source port 20.
ServiceCategories       tcp_20_svc
{
        MaxRate                 81920        # kilobits
        MaxTokenBucket          128          # kilobits
        OutgoingTOS             00011101     # binary
        FlowServiceType         Guaranteed
}

ServicePolicyRules      tcp_20_flt
{
        ProtocolNumber          6  # TCP
        SourceAddressRange      0.0.0.0-0.0.0.0 # Any IP src addr
        DestinationAddressRange 0.0.0.0-0.0.0.0 # Any IP dst addr
        SourcePortRange         20-20
        DestinationPortRange    0-0             # Any dst port
        ServiceReference        tcp_20_svc
}
#
##############################################################################
#
# LDAP server entry.
#ReadFromDirectory
#{
#   LDAP_Server             9.3.33.138  # IP address of LDAP server
#   Base                    o=ibm,c=us  # Base distinguished name
#   LDAP_SelectedTag        myhost      # Typically client hostname
#}
#
##############################################################################
```

โหลดนโยบาย IBM SecureWay Directory Server

ถ้า policy daemon ถูกใช้กับ IBM SecureWay Directory LDAP Server ใช้ schema นี้เพื่อเป็นแนวทางเพื่ออัพเดต `/etc/ldapschema/V3.modifiedschema` ก่อนที่จะสตาร์ท LDAP เซิร์ฟเวอร์

อ้างถึง “การวางแผนและการกำหนดคอนฟิกความละเอียดของชื่อ LDAP (สกีมา IBM SecureWay Directory)” ในหน้า 231 สำหรับรายละเอียด

```
objectClasses {
( ServiceCategories-OID NAME 'ServiceCategories' SUP top MUST
( objectClass $ SelectorTag $ serviceName ) MAY
( description $ FlowServiceType $ MaxRate $ MaxTokenBucket $ OutgoingTos ) )
( ServicePolicyRules-OID NAME 'ServicePolicyRules' SUP top MUST
( objectClass $ PolicyName $ SelectorTag ) MAY
( description $ DestinationAddressRange $ DestinationPortRange $
ProtocolNumber $ ServiceReference $ SourceAddressRange $ SourcePortRange ) )
}
attributeTypes {
( DestinationAddressRange-OID NAME 'DestinationAddressRange' SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( DestinationPortRange-OID NAME 'DestinationPortRange' SYNTAX
 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( FlowServiceType-OID NAME 'FlowServiceType'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( MaxRate-OID NAME 'MaxRate' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( MaxTokenBucket-OID NAME 'MaxTokenBucket' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( OutgoingTos-OID NAME 'OutgoingTos' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( PolicyName-OID NAME 'PolicyName' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( ProtocolNumber-OID NAME 'ProtocolNumber' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( SelectorTag-OID NAME 'SelectorTag' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( ServiceReference-OID NAME 'ServiceReference' SYNTAX
 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( SourceAddressRange-OID NAME 'SourceAddressRange' SYNTAX
 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
( SourcePortRange-OID NAME 'SourcePortRange' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
}

IBMattributeTypes {
( DestinationAddressRange-OID DBNAME ( 'DestinationAddressRange' 'DestinationAddressRange' ) )
( DestinationPortRange-OID DBNAME ( 'DestinationPortRange' 'DestinationPortRange' ) )
( FlowServiceType-OID DBNAME ( 'FlowServiceType' 'FlowServiceType' ) )
( MaxRate-OID DBNAME ( 'MaxRate' 'MaxRate' ) )
( MaxTokenBucket-OID DBNAME ( 'MaxTokenBucket' 'MaxTokenBucket' ) )
( OutgoingTos-OID DBNAME ( 'OutgoingTos' 'OutgoingTos' ) )
( PolicyName-OID DBNAME ( 'PolicyName' 'PolicyName' ) )
( ProtocolNumber-OID DBNAME ( 'ProtocolNumber' 'ProtocolNumber' ) )
( SelectorTag-OID DBNAME ( 'SelectorTag' 'SelectorTag' ) )
( ServiceReference-OID DBNAME ( 'ServiceReference' 'ServiceReference' ) )
( SourceAddressRange-OID DBNAME ( 'SourceAddressRange' 'SourceAddressRange' ) )
( SourcePortRange-OID DBNAME ( 'SourcePortRange' 'SourcePortRange' ) )
}

ldapSyntaxes {
}

matchingRules {
}
```

คอนฟิกูเรชันระบบ QoS

นโยบายที่ซ้อนทับกันถูกติดตั้งอยู่ใน QoS Manager ในการเรียงลำดับ แบบไม่ถูกนำมาพิจารณา ในกรณีของการซ้อนทับนโยบาย แอ็ตทริบิวต์ `PolicyRulePriority` ของ `ServicePolicyRules` ควรถูกระบุไว้เพื่อพิจารณา การเรียงลำดับการบังคับใช้นโยบาย แอ็ตทริบิวต์ `PolicyRulePriority` ใช้เลขจำนวนเต็มเป็นพารามิเตอร์ ในกรณีของการซ้อนทับนโยบาย กฎที่มีค่าตัวเลขสูงสุดถูกบังคับใช้

เฉพาะซ็อกเก็ต **UDP** ที่เชื่อมต่อได้รับการสนับสนุนสำหรับ QoS

นโยบายและเอเจนต์ RSVP เป็นอิสระจากที่ใช้งานร่วมกัน ดังนั้น จึงควรใช้อย่างระมัดระวังเพื่อไม่ให้ระบุนโยบายที่ขัดแย้งหรือครอบคลุมโดยการสำรอง RSVP ที่มีอยู่ ในการมีอยู่ของความขัดแย้งบางข้อ ระบบยอมรับนโยบายแรก หรือการสำรองขณะที่สร้างแฟล็กการละเมิดสำหรับข้ออื่นๆ

สำหรับการดำเนินการที่ถูกต้อง แอ็ตทริบิวต์ `MaxTokenBucket` ต้องถูกตั้งค่าอย่างน้อยที่สุดคือ MTU สูงสุดของอินเตอร์เฟสทั้งหมดที่ตั้งค่าอยู่ในระบบ

การแก้ไขนโยบายถูกจัดการโดยเอเจนต์ของนโยบายด้วยการลบนโยบาย ที่มีอยู่ในปัจจุบันและติดตั้งเอเจนต์ใหม่ ซึ่งอาจส่งผลทำให้หน้าต่างเวลาชั่วคราว สั้นลงในระหว่างที่ทราฟิกที่สอดคล้องกันได้รับเซอร์วิสที่เป็นค่าดีฟอลต์ (โดยปกติคือการสนับสนุนที่ดีที่สุด)

ข้อตกลง IETF มาตรฐานสำหรับโมเดล IntServ และ DiffServ

รีลีสนี้ทำงานร่วมกันกับการพัฒนา Internet Engineering Task Force (IETF) มาตรฐานสำหรับ Differentiated (DiffServ) และ Integrated Services (IntServ) บนอินเตอร์เน็ต

RFC ต่อไปนี้อธิบายถึงคอมโพเนนต์ที่หลากหลายของโมเดล IntServ:

- การใช้ RSVP พร้อมกับ IETF Integrated Services (RFC 2210)
- ข้อกำหนดคุณสมบัติของ Controlled-Load Network Element Service (RFC 2211)
- ข้อกำหนดคุณสมบัติของ Guaranteed Quality of Service (RFC 2212)

RFC ต่อไปนี้อธิบายถึงคอมโพเนนต์ที่หลากหลายของโมเดล DiffServ:

- นิยามของ Differentiated Services Field (DS Field) ในส่วนหัว IPv4 และ IPv6 (RFC 2474)
- สถาปัตยกรรมสำหรับ Differentiated Services (RFC 2475)

RFC ต่อไปนี้มีโครงร่างการใช้งานปัจจุบันของ IP TOS octet:

- ชนิดของเซอร์วิสใน Internet Protocol Suite (RFC 1349)

RFC ต่อไปนี้มีโครงร่างการฝึกปฏิบัติในอนาคตที่กำหนดการใช้ของ IP TOS octet:

- นิยามของ Differentiated Services Field (DS Field) ในส่วนหัว IPv4 และ IPv6 (RFC 2474)
- Assured Forwarding PHB Group (RFC 2597)
- An Expedited Forwarding PHB (RFC 2598)

การสนับสนุน IPv6

QoS สนับสนุนเฉพาะ IPv4 ส่วน IPv6 ไม่สนับสนุน

นโยบายควบคุม daemon

คุณสามารถควบคุมนโยบาย daemon โดยใช้ system resource controller (SRC)

ตัวอย่างเช่นคำสั่ง:

```
startsrc -s policyd -a "-i 60"
```

สตาร์ทนโยบายเอเจนต์ด้วยการรีเฟรชช่วงเวลา 60 วินาที

คำสั่ง

```
stopsrc -s policyd
```

หยุดนโยบาย daemon

หมายเหตุ: การหยุดนโยบาย daemon ไม่ได้ถอนนโยบายที่ติดตั้งไว้ในเคอร์เนล เมื่อคุณเริ่มต้นนโยบาย daemon อีกครั้ง นโยบายเก่า (ติดตั้งไว้ในเคอร์เนลก่อนหน้านี้) ถูกลบทิ้ง และนโยบายที่ นิยามไว้ในไฟล์ `/etc/policyd.conf` ที่ได้ติดตั้งไว้

คำสั่ง **refresh** SRC ไม่ได้ รับการสนับสนุนในปัจจุบัน

คำสั่ง QoS และเมธอด

คำสั่ง **TCP/IP** quality of service และเมธอดถูกแสดง ไว้ที่นี่

สำหรับอัพเดตที่สำคัญกับเอกสารคู่มือนี้ ให้ศึกษาไฟล์ README ใน `/usr/samples/tcpip/qos`

คำสั่ง QoS ต่อไปนี้ไม่ได้รับการสนับสนุน:

- **qosadd**
- **qoslist**
- **qosmod**
- **qosremove**
- **qosstat**
- **mkqos**
- **rmqos**

เมธอด QoS ต่อไปนี้ได้รับการสนับสนุน:

- **cfgqos**
- **ucfgqos**

การแก้ปัญหา TCP/IP

คำสั่ง **netstat** เป็นทูลที่ดีสำหรับการวินิจฉัย ปัญหาทั่วไปในสภาวะแวดล้อม **Transmission Control Protocol/Internet Protocol (TCP/IP)** เน็ตเวิร์ก

คำสั่ง **netstat** ช่วยให้คุณกำหนดพื้นที่ เน็ตเวิร์กที่มีปัญหา หลังจากคุณได้แยกปัญหาเป็นส่วนๆ คุณสามารถใช้เครื่องมือที่เฉพาะด้าน เพื่อดำเนินการต่อ ตัวอย่าง คุณ อาจใช้ **netstat –i** และ **netstat –v** เพื่อ กำหนดว่าคุณมีปัญหากับฮาร์ดแวร์อินเตอร์เฟสใด แล้ว รันการวินิจฉัยเพื่อแยกปัญหาต่อไป หรือ ถ้าคำสั่ง **netstat –s** แสดงว่ามีข้อผิดพลาดโปรโตคอล จากนั้นคุณสามารถ ใช้คำสั่ง **trpt** หรือ **iptrace**

ปัญหาการสื่อสาร

ปัญหาการสื่อสาร **TCP/IP** ทั่วไปคือการไม่สามารถ สื่อสารกับโฮสต์บนเครือข่ายและปัญหาการเราต์ วิธีแก้ไขบางส่วน มีดังนี้

หากคุณไม่สามารถสื่อสารกับเครือข่าย:

- ลองติดต่อกับโฮสต์โดยใช้คำสั่ง **ping** รัน คำสั่ง **ping** บนโลคัลโฮสต์เพื่อตรวจสอบว่าโลคัล อินเตอร์เฟสสำหรับเครือข่ายอัพและกำลังรันอยู่
- ลองแก้ไขชื่อของโฮสต์โดยใช้คำสั่ง **host** หากชื่อไม่ได้รับการแก้ไข แสดงว่าคุณมีปัญหาในการแก้ไขชื่อ โปรดดู “ปัญหาการระบุชื่อ”สำหรับข้อมูลเพิ่มเติม

หากชื่อได้รับการแก้ไขและคุณกำลังพยายามติดต่อกับโฮสต์บนเครือข่ายอื่น แสดงว่าคุณอาจมีปัญหาการเราต์ โปรดดู “ปัญหาการจัดเส้นทาง TCP/IP” ในหน้า 448 สำหรับข้อมูลเพิ่มเติม

- ถ้าเครือข่ายเป็นเครือข่าย token-ring ให้ตรวจสอบเพื่อดูว่าโฮสต์เป้าหมาย อยู่บน ring อื่นหรือไม่ ถ้าใช่ ฟิลด์ **allcast** อาจมีการตั้งค่า ไม่ถูกต้อง ใช้ System Management Interface Tool (SMIT) fast path `smit chinet` เพื่อเข้าถึง เมนู Network Interfaces จากนั้น ตั้งค่าฟิลด์ จำกัดการแพร่สัญญาณไปยังโลคัล Ring เป็น **ไม่ใช่** ในไดอะล็อก token-ring
- ถ้ามีแพ็กเก็ต **Address Resolution Protocol (ARP)** เป็นจำนวนมาก บนเครือข่าย ให้ตรวจสอบว่า subnet mask มีการตั้งค่าอย่างถูกต้อง สภาพนี้รู้จักในชื่อ ว่า broadcast storm และสามารถกระทบต่อประสิทธิภาพของระบบ

ปัญหาการระบุชื่อ

รูทีนตัวแก้ปัญหาบนโฮสต์ที่กำลังรัน **TCP/IP** พยายามแก้ไข ชื่อ โดยใช้แหล่งที่มาตามลำดับที่แสดงรายการ

1. โดเมนเนมเซิร์ฟเวอร์ (**named**)
2. Network Information Service (NIS)
3. ไฟล์ `/etc/hosts` โลคัล

การแก้ไขปัญหาไคลเอ็นต์โฮสต์:

หากคุณไม่สามารถแก้ไขชื่อโฮสต์ และคุณกำลังใช้วิธีแก้ไขชื่อ flat (โดยใช้ไฟล์ `/etc/hosts`) ให้ตรวจสอบว่ามี ชื่อโฮสต์และข้อมูล Internet Protocol (IP) แอดเดรสที่ถูกต้องอยู่ใน ไฟล์ `/etc/hosts`

หากคุณไม่สามารถแก้ไขชื่อโฮสต์ และคุณกำลังใช้เนมเซิร์ฟเวอร์ ให้ปฏิบัติตามขั้นตอนเหล่านี้:

1. ตรวจสอบว่าคุณมีไฟล์ `resolv.conf` ที่ระบุ ชื่อโดเมนและอินเทอร์เน็ตแอดเดรสของเนมเซิร์ฟเวอร์
2. ตรวจสอบว่าเนมเซิร์ฟเวอร์โลคัลอัพโดยการออกใช้คำสั่ง **ping** พร้อมกับ IP แอดเดรสของเนมเซิร์ฟเวอร์ (ที่พบในไฟล์ `resolv.conf` โลคัล)
3. หากเนมเซิร์ฟเวอร์โลคัลอัพ ให้ตรวจสอบว่า **named** daemon บนเนมเซิร์ฟเวอร์โลคัลของคุณใช้งานอยู่โดยการออกใช้คำสั่ง **lssrc –s named** บนเนมเซิร์ฟเวอร์
4. ถ้าคุณกำลังรัน **syslogd** ให้ตรวจสอบข้อความ ที่บันทึกไว้ เอาต์พุตสำหรับข้อความเหล่านี้มีการกำหนดไว้ในไฟล์ `/etc/syslog.conf`

หากขั้นตอนเหล่านี้ไม่ช่วยระบุปัญหา ให้ตรวจสอบเซิร์ฟเวอร์โฮสต์ ชื่อ

การแก้ไขปัญหาเซิร์ฟเวอร์โฮสต์ name:

ใช้ขั้นตอนนี้เพื่อแก้ไขปัญหาเนมเซิร์ฟเวอร์โฮสต์

ถ้าคุณไม่สามารถแก้ไขชื่อโฮสต์ได้:

1. ตรวจสอบว่า **named** daemon แอ็คทีฟโดยเรียก คำสั่งต่อไปนี้:

   ```
   lssrc -s named
   ```

2. ตรวจสอบว่าแอดเดรสของโฮสต์ปลายทางมีอยู่และถูกต้อง ในฐานข้อมูลเนมเซิร์ฟเวอร์ ส่งสัญญาณ **SIGINT** ไปที่ **named** daemon เพื่อดัมพ์ฐานข้อมูลและแคชไปที่ไฟล์ `/var/tmp/named_dump.db` ตรวจสอบว่าแอดเดรสที่คุณพยายามแก้ไขมีอยู่และถูกต้อง

   เพิ่ม หรือแก้ไขข้อมูลการกำหนด name-to-address ในไฟล์ข้อมูลโฮสต์ **named** สำหรับเซิร์ฟเวอร์ master name ของโดเมน จากนั้นเรียกคำสั่ง **SRC** ต่อไปนี้ เพื่ออ่านซ้ำไฟล์ข้อมูล:

   ```
   refresh -s named
   ```

3. ตรวจสอบว่าการร้องขอการกำหนดชื่อถูกประมวลผลอยู่ ทำได้โดย ป้อน **named** daemon จากบรรทัดคำสั่ง และระบุระดับการดีบัก ระดับดีบักที่ใช้ได้คือ 1 ถึง 9 ยิ่งระดับ สูงเท่าไหร่ จะมีการบันทึกข้อมูลกลไกดีบักมากขึ้น

   ```
   startsrc -s named -a "-d DebugLevel"
   ```

4. ตรวจสอบปัญหาคอนฟิกูเรชันในไฟล์ข้อมูล **named** สำหรับข้อมูลเพิ่มเติม ดูที่ “การระบุเนมเซิร์ฟเวอร์” ในหน้า 208 นอกจากนี้ ดูที่ ″DOMAIN Data File Format,″ ″DOMAIN Reverse Data File Format,″ ″DOMAIN Cache File Format,″ และ ″DOMAIN Local Data File Format″ ใน *การอ้างอิงไฟล์*

   หมายเหตุ: ข้อผิดพลาด ทั่วไปคือการใช้ . (จุด) และ @ (เครื่องหมาย at) ไม่ถูกต้องในไฟล์ข้อมูล DOMAIN

ถ้าผู้ใช้ภายนอกไม่สามารถเข้าถึงโดเมนของคุณ ตรวจสอบว่าเซิร์ฟเวอร์ non-master name ของคุณทั้งหมด (slave, hint) มีข้อมูล time-to-live (TTL) เท่ากันในไฟล์ข้อมูล DOMAIN

ถ้า resolvers ภายนอกเคียวรีเซิร์ฟเวอร์ ของคุณอย่างคงที่ ตรวจสอบว่าเซิร์ฟเวอร์ของคุณได้แยกไฟล์ข้อมูล DOMAIN ด้วยค่า TTL ที่เหมาะสม ถ้า TTL เป็นศูนย์หรือน้อยกว่านี้ ข้อมูล ที่คุณส่งจะหมดอายุการใช้งานอย่างรวดเร็ว ให้เซ็ตค่าน้อยที่สุดในเร็กคอร์ด start of authority (SOA) ของคุณเป็นหนึ่งสัปดาห์หรือมากกว่านั้นเพื่อแก้ไขปัญหานี้

ปัญหาการจัดเส้นทาง TCP/IP

ถ้าคุณไม่สามารถเชื่อมต่อโฮสต์ปลายทาง, ให้พิจารณาวิธีแก้ปัญหากับสถานการณ์ ต่อไปนี้

- ถ้าคุณได้รับข้อความแสดงความผิดพลาด `Network Unreachable` ตรวจสอบว่า เส้นทางไปที่เกตเวย์โฮสต์ได้ถูกกำหนดแล้วและถูกต้อง ตรวจสอบได้โดย ใช้คำสั่ง **netstat -r** เพื่อแสดงตารางการจัดเส้นทางเคอร์เนล
- ถ้าคุณได้รับข้อความแสดงความผิดพลาด `ไม่มีเส้นทางไปที่โฮสต์` ตรวจสอบว่า โลคัลเน็ตเวิร์กอินเตอร์เฟสทำงานอยู่ โดยเรียกคำสั่ง **ifconfig** interface_name เอาต์พุตจะแสดงว่าอินเตอร์เฟสทำงานอยู่หรือไม่ ใช้คำสั่ง **ping** เพื่อพยายามและติดต่อกับโฮสต์อื่นบนเน็ตเวิร์กของคุณ
- ถ้าคุณได้รับข้อความแสดงความผิดพลาด `หมดเวลาใช้งานการเชื่อมต่อ`:
  - ตรวจสอบว่าโลคัลเกตเวย์ทำงานอยู่โดยใช้คำสั่ง **ping** ด้วยชื่อ หรืออินเตอร์เน็ตแอดเดรสของเกตเวย์

- ตรวจสอบว่าเส้นทางไปที่เกตเวย์โฮสต์ได้ถูกกำหนดแล้วและถูกต้อง ตรวจสอบได้โดยใช้คำสั่ง **netstat -r** เพื่อแสดงตารางการจัดเส้นทางเคอร์เนล
- ตรวจสอบว่าโฮสต์ที่คุณต้องการสื่อสารด้วยมีรายการตารางการจัดเส้นทาง กลับมาที่เครื่องของคุณ

- ถ้าคุณใช้การจัดเส้นทางสแตติก, ตรวจสอบว่าเส้นทางไปที่โฮสต์ และเกตเวย์โฮสต์เป้าหมายได้ถูกกำหนดไว้ ตรวจสอบได้โดย ใช้คำสั่ง **netstat -r** เพื่อแสดงตารางการจัดเส้นทางเคอร์เนล

  หมายเหตุ: ตรวจสอบว่า โฮสต์ที่คุณต้องการสื่อสารด้วยมีรายการตารางการจัดเส้นทางไปที่เครื่องของคุณ
- ถ้าคุณกำลังใช้การจัดเส้นทางไดนามิกอยู่ ตรวจสอบว่าเกตเวย์ถูกแสดงและ ถูกต้องในตารางการจัดเส้นทางเคอร์เนลโดยใช้คำสั่ง **netstat -r**
- ถ้าเกตเวย์โฮสต์ใช้ **Routing Information Protocol** (RIP) อยู่กับ **routed** daemon, ตรวจสอบว่าเส้นทางสแตติกไปที่โฮสต์ปลายทางถูกเซ็ตอัพในไฟล์ `/etc/gateways`

  หมายเหตุ: คุณ จำเป็นต้องกระทำดังนี้เฉพาะถ้า routing daemon ไม่สามารถระบุเส้นทางไปที่ โฮสต์ระยะไกลผ่านเคียวรีไปที่เกตเวย์อื่น
- ถ้าเกตเวย์โฮสต์ใช้ **RIP** อยู่กับ **gated** daemon, ตรวจสอบว่าเส้นทางสแตติกไปที่โฮสต์ปลายทางถูกเซ็ตอัพในไฟล์ `gated.conf`
- ถ้าคุณใช้การจัดเส้นทางไดนามิกกับ **routed** daemon:
  - ถ้า **routed** ไม่สามารถระบุเส้นทางผ่านเคียวรี (ตัวอย่าง, ถ้าโฮสต์ปลายทางไม่ได้รัน **RIP**, ตรวจไฟล์ `/etc/gateways` เพื่อตรวจสอบว่าเส้นทางไปที่โฮสต์ปลายทางถูกกำหนดไว้
  - ตรวจสอบว่าเกตเวย์ที่มีหน้าที่ในการส่งต่อแพ็กเก็ตไปที่โฮสต์ ทำงานอยู่และรัน **RIP** มิฉะนั้น คุณจะจำเป็นต้องกำหนดเส้นทาง สแตติก
  - รัน **routed** daemon โดยใช้อ็อพชันดีบักเพื่อบันทึกข้อมูลดังกล่าว เป็นการรับแพ็กเก็ตที่ไม่ถูกต้อง เรียก daemon จากบรรทัดคำสั่งโดยใช้ คำสั่งต่อไปนี้:

    ```
    startsrc -s routed -a "-d"
    ```

  - รัน **routed** daemon โดยใช้แฟล็ก **-t** ซึ่งทำให้แพ็กเก็ต ทั้งหมดที่ส่งหรือรับถูกเขียนไปที่เอาต์พุตมาตรฐาน เมื่อ **routed** ถูก รันในโหมดนี้ จะยังคงอยู่ใต้การควบคุมของเทอร์มินัลที่สตาร์ท routed ดังนั้น อินเตอร์รัปต์จากเทอร์มินัลการควบคุมจะหยุด daemon
- ถ้าคุณใช้การจัดเส้นทางไดนามิกกับ **gated** daemon:
  - ตรวจสอบว่าไฟล์ `/etc/gated.conf` ถูกต้องค่าอย่างถูกต้อง และคุณรันโปรโตคอลที่ถูกต้อง
  - ตรวจสอบว่าเกตเวย์บนซอร์สเน็ตเวิร์กใช้โปรโตคอลเดียวกันกับ เกตเวย์บนเน็ตเวิร์กปลายทาง
  - ตรวจสอบว่าเครื่องที่คุณพยายามสื่อสารมี เส้นทางกับมาที่เครื่องโฮสต์ของคุณ
  - ตรวจสอบว่าชื่อเกตเวย์ในไฟล์ `gated.conf` ตรงกับชื่อเกตเวย์ที่แสดงในไฟล์ `/etc/networks`
- ถ้าคุณใช้โปรโตคอล **RIP** หรือ **HELLO** และไม่สามารถระบุเส้นทาง ไปที่ปลายทางผ่านเคียวรีการจัดเส้นทางได้ให้ตรวจสอบไฟล์ `gated.conf` เพื่อตรวจสอบว่าเส้นทางไปที่โฮสต์ปลายทางถูกกำหนดไว้ เซ็ตเส้นทางสแตติกภายใต้ เงื่อนไขต่อไปนี้:
  - โฮสต์ปลายทางไม่ได้รันโปรโตคอลเหมือนกับโฮสต์ต้นทาง ดังนั้นจึงไม่สามารถแลกเปลี่ยนข้อมูลการจัดเส้นทาง
  - โฮสต์ต้องถูกติดต่อโดยเกตเวย์ระยะไกล (เกตเวย์ที่อยู่บน ระบบ autonomous อื่นที่ไม่ใช่โฮสต์ต้นทาง) **RIP** สามารถใช้ได้ เฉพาะกับโฮสต์บนระบบ autonomous เดียวกัน

ถ้านอกเหนือจากนี้ล้มเหลว คุณอาจต้องการเปิดการติดตามสำหรับ daemon การจัดเส้นทางของคุณ (**routed** หรือ **gated**) ใช้คำสั่ง SRC **traceson** จากบรรทัดคำสั่ง หรือส่งสัญญาณไปที่ daemon เพื่อระบุระดับ การติดตามอื่น ดูที่ **gated** daemon หรือ **routed** daemon สำหรับข้อมูลจำเพาะเกี่ยวกับการส่งสัญญาณไปที่ daemons เหล่านี้

การแก้ไขปัญหาด้วยการสนับสนุน SRC

ใช้ข้อเสนอแนะเหล่านี้เพื่อแก้ไขปัญหาทั่วไปกับ System Resource Controller

- ถ้าเปลี่ยนเป็นไฟล์ `/etc/inetd.conf` ไม่มี ผล:

  อัพเดต **inetd** daemon โดยเรียกคำสั่ง **refresh -s inetd** หรือคำสั่ง **kill -1** *InetdPID*
- ถ้า **startsrc -s** [*subsystem name*] ส่งกลับข้อความแสดงความผิดพลาดต่อไปนี้:

  ```
  0513-00  System Resource Controller ไม่แอ็คทีฟ
  ```

  ระบบย่อย System Resource Controller ไม่ถูกเรียกทำงาน เรียกคำสั่ง **srcmstr &** เพื่อสตาร์ท SRC และวเรียกซ้ำคำสั่ง **startsrc**

  คุณ อาจต้องการพยายามสตาร์ท daemon จากบรรทัดคำสั่งโดยไม่มีการสนับสนุน SRC
- ถ้า **refresh -s** [*subsystem name*] หรือ **lssrc -ls** [*subsystem name*] ส่งคือข้อความแสดงความผิดพลาดต่อไปนี้:

  ```
  [subsystem name] ไม่สนับสนุนอ็อพชันนี้
  ```

  ระบบย่อย ไม่สนับสนุนอ็อพชัน SRC ที่เรียก ตรวจสอบเอกสารคู่มือระบบย่อย เพื่อตรวจสอบอ็อพชันที่ระบบย่อยสนับสนุน
- ถ้าข้อความต่อไปนี้ถูกแสดง:

  ```
  ไม่พบ SRC, ดำเนินการต่อโดยไม่มีการสนับสนุน SRC
  ```

  daemon ถูกเรียกโดยตรงจากบรรทัดคำสั่ง แทนการใช้คำสั่ง **startsrc** นี่ไม่เป็นปัญหา อย่างไรก็ตาม คำสั่ง SRC เช่น **stopsrc** และ **refresh** จะ ไม่ควบคุมระบบย่อยที่ถูกเรียกโดยตรง

ถ้า **inetd** daemon ทำงานอยู่อย่างถูกต้อง แบะเซอร์วิสที่เหมาะสมดูเหมือนว่าจะถูกต้องแต่คุณยังคงไม่สามารถเชื่อมต่อได้ให้ลองรันกระบวนการ **inetd** daemon ผ่านดีบักเกอร์

1. หยุด **inetd** daemon ชั่วคราว:

   ```
   stopsrc -s inetd
   ```

   stopsrc command หยุดระบบย่อยเช่น **inetd** daemon
2. แก้ไขไฟล์ `syslog.conf` เพื่อเพิ่มบรรทัด การดีบักที่ด้านล่าง ตัวอย่างเช่น:

   ```
   vi /etc/syslog.conf
   ```

   a. เพิ่มบรรทัด `*.debug /tmp/myfile` ที่ด้านล่างของ ไฟล์และออก

   b. ไฟล์ที่คุณระบุต้องมีอยู่ (`/tmp/myfile` ใน ตัวอย่างนี้) คุณสามารถใช้คำสั่ง **touch** เพื่อสร้าง ไฟล์ของคุณขึ้น
3. รีเฟรชไฟล์:

   - ถ้าคุณใช้ SRC อยู่ให้ป้อน:

     ```
     refresh -s syslogd
     ```
   - ถ้าคุณไม่ใช้ SRC, ให้ kill **syslogd** daemon:

     ```
     kill -1 `ps -e | grep /etc/syslogd | cut -c1-7`
     ```

4. เริ่มต้น **inetd** daemon backup โดยมีการเปิดใช้ การดีบัก:

   ```
   startsrc -s inetd -a "-d"
   ```

   แฟล็ก **–d** เปิดใช้การดีบัก

5. พยายามสร้างการเชื่อมต่อเพื่อบันทึกข้อผิดพลาดในไฟล์การดีบัก `/tmp/myfile` ตัวอย่างเช่น:

   ```
   tn bastet
   Trying...
   connected to bastet
   login:>
   Connection closed
   ```

6. ดูว่ามีข้อมูลใดแสดงขึ้นมาเนื่องจากปัญหาในไฟล์การดีบัก ตัวอย่างเช่น:

   ```
   tail -f /tmp/myfile
   ```

การแก้ไขปัญหา telnet หรือ rlogin

คำอธิบายเหล่านี้อาจมีประโยชน์ในการแก้ไขปัญหากับคำสั่ง **telnet** หรือ **rlogin**

ถ้าคุณมีปัญหากับจอภาพแสดงผลไม่ถูกต้องใน แอ็พพลิเคชันแบบเต็มจอ:

1. ตรวจสอบตัวแปรสภาวะแวดล้อม **TERM** โดยการ เรียกหนึ่งในคำสั่งต่อไปนี้:

   ```
   env
   ```

   ```
   echo $TERM
   ```

2. ตรวจสอบว่าตัวแปร **TERM** ถูกเซ็ต เป็นค่าที่ตรงกับชนิดของจอแสดงผลเทอร์มินัลที่คุณใช้อยู่

คำสั่งย่อย **telnet** ที่สามารถช่วยเหลือในการดีบัก ปัญหารวมถึง:

ไอเท็ม	คำอธิบาย
display	แสดงเซ็ตและสลับค่า
toggle	สลับการแสดงผลของข้อมูลเน็ตเวิร์กทั้งหมดเป็น hex
อ็อพชัน toggle	สลับการแสดงผลของอ็อพชันกระบวนการ **telnet** ภายใน

ถ้า **inetd** daemon สามารถเรียกใช้เซอร์วิส **telnet** แต่คุณยังคงไม่สามารถเชื่อมต่อโดยใช้คำสั่ง **telnet,** อาจมีปัญหากับอินเตอร์เฟส **telnet**

1. ตรวจสอบว่า **telnet** ใช้ชนิดเทอร์มินัลที่ถูกต้อง
   a. ตรวจสอบตัวแปร **$TERM** บนเครื่องของคุณ:

      ```
      echo $TERM
      ```

   b. ล็อกอินเข้าสู่เครื่องซึ่งคุณพยายามเชื่อมต่อและตรวจสอบตัวแปร **$TERM**:

      ```
      echo $TERM
      ```

2. ใช้ความสามารถการดีบักของอินเตอร์เฟส **telnet** โดย การป้อนคำสั่ง **telnet** โดยไม่มีแฟล็ก

   ```
   telnet
   tn>
   ```

   a. พิมพ์ `open` *`host`* โดยที่ *host* คือ ชื่อเครื่อง
   b. กด Ctrl–T เพื่อไปที่พร้อมต์ `tn%gt;`
   c. ที่พร้อมต์ `tn>` พิมพ์ `debug` สำหรับ โหมดการดีบัก

3. พยายามเชื่อมต่อไปที่เครื่องอื่นโดยใช้อินเตอร์เฟส **telnet**:

```
telnet bastet
Trying...
Connected to bastet
Escape character is '^T'.
```

ดูที่จอแสดงผลขณะที่คำสั่งต่างๆ เลื่อนขึ้นบนจอภาพ ตัวอย่างเช่น:

```
SENT do ECHO
SENT do SUPPRESS GO AHEAD
SENT will TERMINAL TYPE (reply)
SENT do SUPPORT SAK
SENT will SUPPORT SAK (reply)
RCVD do TERMINAL TYPE (don't reply)
RCVD will ECHO (don't reply)
RCVD will SUPPRESS GO AHEAD (don't reply)
RCVD wont SUPPORT SAK (reply)
SENT dont SUPPORT SAK (reply)
RCVD do SUPPORT SAK (don't reply)
SENT suboption TELOPT_NAWS Width 80, Height 25
RCVD suboption TELOPT_TTYPE SEND
RCVD suboption TELOPT_TTYPE aixterm
...
```

4. ตรวจสอบ `/etc/termcap` หรือ `/usr/lib/terminfo` สำหรับ นิยาม `aixterm` ตัวอย่างเช่น:

```
ls -a /usr/lib/terminfo
```

5. ถ้านิยาม `aixterm` ไม่มีอยู่ ให้เพิ่มโดยสร้าง ไฟล์ `ibm.ti` ตัวอย่างเช่น:

```
tic ibm.ti
```

คำสั่ง **tic** เป็นคอมไพลเลอร์ข้อมูลเทอร์มินัล

ปัญหากับฟังก์ชันและปุ่มลูกศรอาจเกิดขึ้นได้ เมื่อใช้คำสั่ง **rlogin** และ **telnet** กับโปรแกรมที่ใช้ extended curses ฟังก์ชันและปุ่มลูกศรสร้าง escape sequences, ซึ่งถูกแยก ถ้ามีเวลาน้อยเกินไปถูกกำหนดให้สำหรับลำดับคีย์ ทั้งหมด Curses รอเป็นระยะเวลาหนึ่งเพื่อตัดสินว่า Esc บ่งชี้ถึงคีย์ escape เท่านั้นหรือเป็นการเริ่มต้นของ escape sequence มัลติไบต์ ที่สร้างโดยคีย์อื่น เช่นเคอร์เซอร์คีย์ คีย์การดำเนินการ และ ฟังก์ชันคีย์

ถ้าไม่มีข้อมูล หรือข้อมูลที่ไม่ถูกต้อง ตามด้วย Esc ใน ระยะเวลาที่กำหนดให้, curses ตัดสินว่า Esc เป็นคีย์ escape และ ลำดับคีย์จะถูกแยก การหน่วงเป็นผลมาจากคำสั่ง **rlogin** หรือ **telnet** มีความสัมพันธ์กับเน็ตเวิร์ก บางครั้งคีย์ ลูกศรและฟังก์ชันคีย์ ทำงานแต่บางครั้ง ก็ไม่ ขึ้นกับความเร็วของเน็ตเวิร์กซึ่งคุณเชื่อมต่ออยู่ การตั้งค่าตัวแปรสภาวะแวดล้อม **ESCDELAY** เป็นค่าขนาดใหญ่ (1000 ถึง 1500) แก้ปัญหานี้ได้อย่างมีประสิทธิภาพ

ปัญหาคอนฟิกูเรชัน TCP/IP

อินเตอร์เฟสเครือข่ายมีการกำหนดคอนฟิกโดยอัตโนมัติในระหว่าง สตาร์ทอัพระบบครั้งแรกหลังจากติดตั้งอะแด็ปเตอร์การ์ด อย่างไรก็ตาม คุณยังคงต้อง ตั้งค่าแรกเริ่มบางอย่างสำหรับ **TCP/IP** รวมถึงชื่อโฮสต์ อินเทอร์เน็ตแอดเดรส และ subnet mask

เมื่อต้องการทำเช่นนี้ คุณสามารถใช้อินเตอร์เฟส SMIT ด้วยวิธีต่อไปนี้:

- ใช้พาธด่วน `smit mktcpip` เพื่อตั้งค่าแรกเริ่ม สำหรับชื่อโฮสต์ อินเทอร์เน็ตแอดเดรส และ subnet mask

- ใช้พาธด่วน smit mktcpip เพื่อระบุเนมเซิร์ฟเวอร์ ที่จะนำเสนอเซอร์วิสการแก้ไขชื่อ (โปรดทราบว่า smit mktcpip กำหนดคอนฟิก อินเตอร์เฟสเครือข่ายเพียงรายการเดียวเท่านั้น)
- ใช้พาธด่วน smit chinet เพื่อตั้งค่าแอ็ตทริบิวต์เครือข่าย อื่น

คุณยังอาจต้องการตั้งค่าสแตติกเราต์ที่โฮสต์ต้องการสำหรับการส่ง ข้อมูลที่ส่งผ่าน เช่น เราต์ไปยังโลคัลเกตเวย์ ใช้ SMIT fast path, smit mkroute เพื่อตั้งค่า เหล่านี้อย่างถาวรในฐานข้อมูลการกำหนดคอนฟิก

หากคุณมีปัญหาอื่นเกี่ยวกับคอนฟิกูเรชัน ให้ดูที่ “การกำหนดค่าเน็ตเวิร์ก TCP/IP” ในหน้า 115 สำหรับข้อมูลเพิ่มเติม

ปัญหา TCP/IP ทั่วไปกับเน็ตเวิร์กอินเตอร์เฟส

เน็ตเวิร์กอินเตอร์เฟสถูกกำหนดค่าโดยอัตโนมัติระหว่างการเริ่มทำงาน ระบบครั้งแรกหลังจากติดตั้งการ์ดอะแด็ปเตอร์ อย่างไรก็ตาม มีค่าที่แน่นอน บางค่าที่ต้องถูกตั้งค่าเพื่อให้ **TCP/IP** เริ่มทำงาน เหล่านี้รวมชื่อโฮสต์ และอินเตอร์เน็ตแอดเดรส และสามารถ ตั้งค่าโดยใช้ SMIT fast path, smit mktcpip

ถ้าคุณเลือกใช้วิธี SMIT ให้ใช้พาธด่วน smit mktcpip เพื่อตั้งค่าเหล่านี้ถาวรในฐานข้อมูลการกำหนดค่า ใช้พาธด่วน smit chinet และ smit hostname เพื่อ เปลี่ยนค่าในระบบที่กำลังดำเนินงาน พาธด่วน smit mktcpip อย่างน้อยที่สุดจะกำหนดค่า **TCP/IP** เมื่อต้องการเพิ่มอะแด็ปเตอร์ ให้ใช้เมนู Further Configuration ซึ่งสามารถเข้าถึงโดยใช้พาธด่วน smit tcpip

ถ้าคุณตรวจสอบค่าเหล่านี้เพื่อยืนยันความถูกต้องแล้ว และคุณยังคง มีปัญหาในการส่ง และรับข้อมูล ให้ตรวจสอบต่อไปนี้:

- ตรวจสอบว่าเน็ตเวิร์กอะแด็ปเตอร์ของคุณมีเน็ตเวิร์กอินเตอร์เฟสอยู่โดยการเรียกใช้ คำสั่ง **netstat -i** เอาต์พุตควรแสดงรายการอินเตอร์เฟส เช่น **tr0** ในคอลัมน์ Name ถ้าไม่ใช่ ให้สร้างอินเตอร์เฟสเครือข่ายโดยการป้อน SMIT fast path smit mkinet
- ตรวจสอบว่า IP address สำหรับอินเตอร์เฟสถูกต้องโดยการเรียกใช้คำสั่ง **netstat -i** เอาต์พุตควรแสดงรายการ IP address ในคอลัมน์ Network ถ้าไม่ถูกต้อง ให้ตั้งค่า IP แอดเดรส โดยการป้อน SMIT fast path smit chinet
- ใช้คำสั่ง **arp** เพื่อให้แน่ใจว่าคุณมี IP address สมบูรณ์สำหรับเครื่องปลายทาง ตัวอย่างเช่น:

  ```
  arp -a
  ```

  คำสั่ง **arp** ค้นหาแอดเดรสอะแด็ปเตอร์ฟิสิคัล คำสั่งนี้อาจแสดงแอดเดรส ที่ไม่สมบูรณ์ ตัวอย่าง เช่น:

  ```
  ? (192.100.61.210) at (incomplete)
  ```

  ซึ่ง อาจเกิดขึ้นเนื่องจากเครื่องไม่ได้เสียบปลั๊ก แอดเดรสที่มีปัญหาที่ไม่มีเครื่องอยู่ที่ แอดเดรสที่ระบุ หรือมีปัญหาด้านฮาร์ดแวร์ (เช่นเครื่องที่เชื่อมต่อ และได้รับแพ็กเก็ต แต่ไม่สามารถส่งแพ็กเก็ตกลับได้)
- ค้นหาข้อผิดพลาดบนการ์ดอะแด็ปเตอร์ ตัวอย่างเช่น:

  ```
  netstat -v
  ```

  คำสั่ง **netstat -v** แสดงสถิติสำหรับไดรเวอร์อุปกรณ์อะแด็ปเตอร์ Ethernet, Token Ring, X.25 และ 802.3 คำสั่งยังแสดงเน็ตเวิร์กและ ข้อมูลบันทึกข้อผิดพลาดสำหรับไดรเวอร์อุปกรณ์ทั้งหมดบนอินเตอร์เฟสได้แก่: No Mbufs Errors, No Mbuf Extension Errors และ Packets Transmitted and Adapter Errors Detected
- ตรวจสอบบันทึกข้อผิดพลาดโดยการรันคำสั่ง **errpt** เพื่อ ให้แน่ใจว่าไม่มีปัญหาในอะแด็ปเตอร์
- ตรวจสอบว่าการ์ดอะแด็ปเตอร์ใช้งานได้ปกติ โดยการรันการตรวจวินิจฉัย ใช้ smit diag fast path หรือคำสั่ง **diag**

ปัญหา TCP/IP กับ SLIP เน็ตเวิร์กอินเตอร์เฟส:

โดยทั่วไป วิธีที่มีประสิทธิภาพที่สุดสำหรับการดีบักปัญหากับ **Serial Line Interface Protocol (SLIP)** อินเตอร์เฟสคือตรวจสอบซ้ำ การคอนฟิกูเรชันของคุณ, โดยตรวจสอบแต่ละขั้นตอน

อย่างไรก็ตาม คุณยังสามารถ:

- ตรวจสอบว่ากระบวนการ **slattach** รันอยู่และใช้ พอร์ต tty ที่ถูกต้องโดยการเรียกคำสั่ง **ps -ef** ถ้า ไม่ให้รันคำสั่ง **slattach** (ดูที่ "การตั้งค่า SLIP ผ่านโมเด็ม" ในหน้า 666 หรือ "การตั้งค่า SLIP ผ่านสายเคเบิลแบบ null โมเด็ม" ในหน้า 668 สำหรับไวยากรณ์ ที่ถูกต้องที่คุณควรใช้)
- ตรวจสอบว่าแอดเดรส point-to-point ถูกระบุอย่างถูกต้องโดยการ ป้อน `smit chinet` fast path

  เลือกอินเตอร์เฟส **SLIP** ตรวจสอบว่าฟิลด์ **INTERNET ADDRESS** และ **DESTINATION ADDRESS** ถูกต้อง

ถ้าโมเด็มไม่ได้ทำงานอย่างถูกต้อง:

- ตรวจสอบว่าโมเด็มถูกติดตั้งอย่างถูกต้อง ดูคู่มือการติดตั้ง โมเด็ม
- ตรวจสอบว่าการควบคุมสายงานของโมเด็มไม่ถูกปิด

ถ้า tty ทำงานไม่ถูกต้อง ตรวจสอบว่าอัตรา tty baud และคุณสมบัติโมเด็มถูกเซ็ตอย่างถูกต้องในฐานข้อมูลคอนฟิกูเรชัน โดยการป้อน `smit tty` fast path

ปัญหาของ TCP/IP กับการ์ดเครือข่ายอีเทอร์เน็ต:

อ้างถึงรายการตรวจสอบนี่เมื่อปัญหาด้าน **TCP/IP** กับการ์ดเครือข่ายอีเทอร์เน็ตยังคงอยู่

ถ้าการ์ดเครือข่ายถูกเตรียมข้อมูลแล้ว แอดเดรสที่ถูกต้องถูกระบุไว้แล้ว และคุณได้ตรวจสอบการ์ดแล้วว่าทำงานได้:

- ให้ตรวจสอบคุณใช้หัวต่อแบบ T เสียบโดยตรงกับตัวรับส่ง inboard/outboard
- ตรวจดูให้แน่ใจว่าคุณใช้สายเคเบิลอีเทอร์เน็ต (สายเคเบิลอีเทอร์เน็ตคือ 50 OHM)
- ตรวจดูให้แน่ใจว่าคุณใช้เทอรมิเนเตอร์ของอีเทอร์เน็ต (เทอรมิเนเตอร์ของอีเทอร์เน็ต คือ 50 OHM)
- การ์ดอีเทอร์เน็ตสามารถใช้กับตัวรับส่งที่อยู่บนการ์ด หรือ กับตัวรับส่งแบบภายนอก ซึ่งมีจัมเปอร์บนการ์ดให้ระบุ ว่าคุณใช้งานแบบใด ตรวจสอบว่าจัมเปอร์ของคุณตั้งค่าไว้อย่างถูกต้อง (โปรดดู คู่มือการ์ดของคุณสำหรับคำแนะนำ)
- ตรวจสอบว่าคุณใช้ชนิดหัวต่ออีเทอร์เน็ตที่ถูกต้อง (แบบ thin ใช้ BNC; แบบ thick ใช้ DIX) ถ้าคุณเปลี่ยนแปลงชนิดตัวเชื่อมต่อนี้ ใช้ **smit chgenet** SMIT fast path เพื่อตั้งค่าฟิลด์ Apply Change to Database Only (ตั้งค่าเป็น Yes ใน SMIT) รีสตาร์ทเครื่องเพื่อประยุกต์ใช้คอนฟิกูเรชันที่เปลี่ยนแปลง (โปรดดู "การจัดการและตั้งค่าอะแด็ปเตอร์" ในหน้า 174)

ปัญหา TCP/IP กับเน็ตเวิร์กอินเตอร์เฟส Token-Ring:

ใช้คำแนะนำต่อไปนี้เพื่อแก้ไขปัญหาการสื่อสารกับเน็ตเวิร์กอินเตอร์เฟส ของคุณ

ถ้าคุณไม่สามารถสื่อสารกับบางเครื่องบนเน็ตเวิร์กของคุณ แม้ว่าเน็ตเวิร์กอินเตอร์เฟสได้ถูกเตรียมข้อมูลเบื้องต้นแล้ว แอดเดรสถูกระบุ อย่างถูกต้อง และคุณได้ตรวจสอบแล้วว่าการ์ดทำงานได้ดี:

- ตรวจสอบเพื่อดูว่าโฮสต์ซึ่งมีผู้ที่คุณไม่สามารถสื่อสารได้ อยู่บนวงแหวนที่ ต่างกันหรือไม่ ถ้ามีอยู่ ให้ใช้ SMIT fast path `smit chinet` เพื่อตรวจสอบฟิลด์ Confine BROADCAST to Local Token-Ring *ห้าม* ตั้งค่าเป็น No ใน SMIT

- ตรวจสอบเพื่อดูว่าโทเค็นริงอะแด็ปเตอร์ถูกตั้งค่าให้รันที่ความเร็ว วงแหวนที่ถูกต้อง ถ้าถูกกำหนดคอนฟิกไม่ถูกต้อง ให้ใช้ SMIT เพื่อเปลี่ยนแปลงแอ็ตทริบิวต์ความเร็วอะแด็ปเตอร์ริง (ดูที่ "การจัดการและตั้งค่าอะแด็ปเตอร์" ในหน้า 174) เมื่อ **TCP/IP** ถูก รีสตาร์ท, โทเค็นริงอะแด็ปเตอร์จะมีความเร็ววงแหวนเท่ากับส่วนที่เหลือ ของเน็ตเวิร์ก

ปัญหา TCP/IP กับบริดจ์ Token-Ring/Ethernet:

ถ้าคุณไม่สามารถสื่อสารระหว่างโทเค็นริงและเน็ตเวิร์ก Ethernet โดยใช้บริดจ์ และคุณได้ตรวจสอบแล้วว่าบริดจ์ทำงานถูกต้อง, อะแด็ปเตอร์ Ethernet อาจมีการดรอปของแพ็กเก็ต

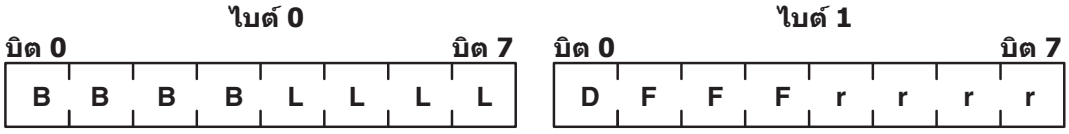
เครื่องจะดรอปแพ็กเก็ต ถ้าแพ็กเก็ตขาเข้า (รวมทั้งส่วนหัว) มีขนาดมากกว่าค่า maximum transmission unit (MTU) ของเน็ตเวิร์กอะแด็ปเตอร์ ตัวอย่าง แพ็กเก็ต 1500-ไบต์ที่ส่งโดยอะแด็ปเตอร์โทเค็นริงผ่านบริดจ์ รวบรวมส่วนหัวขนาด 8-ไบต์ logical link control (LLC), ขนาดแพ็กเก็ตรวม จะเป็น 1508 ถ้าการรับ Ethernet adapter MTU ถูกเซ็ตเป็น 1500, แพ็กเก็ตจะถูกดรอป

ตรวจสอบค่า MTU ของเน็ตเวิร์กอะแด็ปเตอร์ทั้งสอง เพื่ออนุญาตค่าส่วนหัว แปดไบต์ LLL, โทเค็นริงอะแด็ปเตอร์ผนวกกับแพ็กเก็ตขาออก ให้เซ็ตค่า MTU สำหรับโทเค็นริงอะแด็ปเตอร์อย่างน้อยแปดไบต์ต่ำกว่า ค่า MTU สำหรับอะแด็ปเตอร์ Ethernet ตัวอย่าง เซ็ต MTU สำหรับโทเค็นริงอะแด็ปเตอร์ เป็น 1492 ในการสื่อสารกับอะแด็ปเตอร์ Ethernet ที่มี MTU เป็น 1500

ปัญหา TCP/IP กับบริดจ์ Token-Ring/Token-Ring:

เมื่อดำเนินการผ่านบริดจ์ให้เปลี่ยนค่าดีฟอลต์ เป็น 1500 สำหรับ maximum transmission unit (MTU) เป็นค่าที่น้อยกว่าฟิลด์ข้อมูลสูงสุดอยู่แปด (I-frame สูงสุด) ที่ประกาศโดย บริดจ์ในฟิลด์ควบคุมการจัดเส้นทาง

เมื่อต้องการค้นหาค่าฟิลด์ควบคุมการจัดเส้นทาง ให้ใช้ **iptrace** daemon เพื่อดูแพ็กเก็ตขาเข้า บิต 1, 2 และ 3 ของ Byte 1 เป็น Largest Frame Bits, ซึ่งระบุฟิลด์ข้อมูลสูงสุดที่สามารถส่งได้ ระหว่างสถานีการสื่อสารบนเส้นทางจำเพาะ ดูที่ข้อมูลต่อไปนี้สำหรับรูปแบบของฟิลด์ควบคุมการจัดเส้นทาง:



รูปที่ 27. ฟิลด์ควบคุมการจัดเส้นทาง

ข้อมูลนี้แสดงไบต์ 0 และไบต์ 1 ของฟิลด์ควบคุม การจัดเส้นทาง แปดบิตของไบต์หนึ่งคือ B, B, B, B, L, L, L, L แปดบิตของไบต์ 1 คือ D, F, F, F, r, r, r, r

ค่าสำหรับ Largest Frame Bits เป็นดังนี้:

ไอเท็ม	คำอธิบาย
000	ระบุค่าสูงสุด 516 ไบต์ในฟิลด์ข้อมูล
001	ระบุค่าสูงสุด 1500 ไบต์ในฟิลด์ข้อมูล
010	ระบุค่าสูงสุด 2052 ไบต์ในฟิลด์ข้อมูล
011	ระบุค่าสูงสุด 4472 ไบต์ในฟิลด์ข้อมูล
100	ระบุค่าสูงสุด 8144 ไบต์ในฟิลด์ข้อมูล
101	สงวนไว้
110	สงวนไว้
111	ใช้ในเฟรม all-routes broadcast

ตัวอย่าง ถ้าค่า I-frame สูงสุดคือ 2052 ในฟิลด์ ควบคุมการจัดเส้นทาง ขนาด MTU ควรถูกเซ็ตเป็น 2044 นี่สำหรับ โทเค็นริงเน็ตเวิร์กอินเตอร์เฟสเท่านั้น

หมายเหตุ: เมื่อใช้ **iptrace,** เอาต์พุตไฟล์ *ไม่* อยู่บน Network File System (NFS)

ปัญหา TCP/IP ในการสื่อสารกับโฮสต์รีโมต

ถ้าคุณไม่สามารถสื่อสารกับโฮสต์รีโมตได้ ให้ลองทำตามข้อเสนอแนะเหล่านี้

- รันคำสั่ง **ping** บนโฮสต์โลคัลเพื่อตรวจสอบว่า อินเตอร์เฟสโลคัลไปยังเน็ตเวิร์กนั้นทำงาน และกำลังรันอยู่
- ใช้คำสั่ง **ping** สำหรับโฮสต์และเกตเวย์ที่ มีจำนวน hops เพิ่มมากขึ้นอย่างต่อเนื่องจากโฮสต์โลคัลเพื่อพิจารณาจุดที่ จะทำให้การสื่อสารล้มเหลว

ถ้าคุณมีปัญหาแพ็กเก็ตสูญหาย หรือประสบปัญหา ล่าช้าในการนำส่งแพ็กเก็ต ให้ลองต่อไปนี้:

- ใช้คำสั่ง **trpt** เพื่อติดตามแพ็กเก็ตที่ระดับ ซ็อกเก็ต
- ใช้คำสั่ง **iptrace** เพื่อติดตามเลเยอร์โปรโตคอล ทั้งหมด

ถ้าคุณไม่สามารถสื่อสารระหว่างเน็ตเวิร์กที่เป็นแบบโทเค็นริง และ อีเทอร์เน็ตโดยใช้บริดจ์ได้ และคุณได้ตรวจสอบว่าบริดจ์ทำงานถูกต้องแล้ว:

- ตรวจสอบค่า MTU ของอะแด็ปเตอร์ทั้งสอง ค่า MTU ต้องใช้ร่วมกันได้ เพื่ออนุญาตให้มีการสื่อสาร เครื่องที่จะตัดแพ็กเก็ตทิ้ง ถ้าแพ็กเก็ตขาเข้า (รวมถึง ส่วนหัว) มีขนาดใหญ่กว่าค่า MTU ของอะแด็ปเตอร์ ตัวอย่างเช่น แพ็กเก็ตขนาด 1500 ไบต์ที่ส่งผ่านบริดจ์จะรวมส่วนหัว LLC 8 ไบต์ ทำให้ขนาด แพ็กเก็ตรวมเป็น 1508 ถ้า MTU ของเครื่องรับถูกตั้งค่าเป็น 1500 แพ็กเก็ต ที่มีขนาด 1508 ไบต์จะถูกตัดทิ้ง

ปัญหา TCP/IP กับการตอบกลับ snmpd ไปที่เคียวรี

ถ้า **snmpd** ไม่ตอบสนองกับเคียวรีและ ไม่มีการรับข้อความไฟล์บันทึก, แพ็กเก็ตอาจใหญ่เกินไปสำหรับเคอร์เนล **User Datagram Protocol (UDP)** แพ็กเก็ต handler

ถ้าเป็นกรณีนี้ ให้เพิ่มค่าตัวแปรเคอร์เนล, **udp_sendspace** และ **udp_recvspace** โดยเรียกคำสั่งต่อไปนี้:

```
no -o udp_sendspace=64000
no -o udp_recvspace=64000
```

ขนาดสูงสุดสำหรับแพ็กเก็ต **UDP** หนึ่งคือ 64K ถ้าเคียวรีของคุณ ใหญ่กว่า 64K, เคียวรีจะถูกปฏิเสธ แยกแพ็กเก็ตออกเป็นแพ็กเก็ตขนาดเล็กลง เพื่อหลีกเลี่ยงปัญหานี้

ปัญหา TCP/IP เกี่ยวกับ Dynamic Host Configuration Protocol

ในกรณีที่คุณไม่สามารถหาข้อมูลคอนฟิกูเรชัน ให้ลอง วิธีแก้ไขต่อไปนี้

หากคุณไม่สามารถเรียกใช้ IP แอดเดรสหรือพารามิเตอร์คอนฟิกูเรชันอื่น:

- ตรวจสอบเพื่อดูว่าคุณได้ระบุอินเตอร์เฟสที่จะกำหนดคอนฟิกแล้วหรือไม่ ซึ่งสามารถทำได้โดยใช้ SMIT fast path `smit dhcp`
- ตรวจสอบเพื่อดูว่ามีเซิร์ฟเวอร์บนเครือข่ายโลคัล หรือรีเลย์เอเจนต์ ที่กำหนดคอนฟิกเพื่อเรียกใช้คำร้องขอของคุณนอกเครือข่ายโลคัลหรือไม่
- ตรวจสอบเพื่อดูว่าโปรแกรม **dhcpcd** กำลังรันหรือไม่ ถ้า ไม่ให้ใช้คำสั่ง **startsrc -s dhcpcd**

คำสั่ง TCP/IP

TCP/IP เป็นส่วนหนึ่งของโครงสร้างสำคัญของระบบ TCP/IP ช่วยให้คุณสามารถสื่อสารกับเทอร์มินัลหรือระบบอื่นโดยเพียงแค่ ดำเนินการคำสั่งหรือโปรแกรม

TCP/IP เป็นส่วนหนึ่งของโครงสร้างสำคัญของระบบ TCP/IP ช่วยให้คุณสามารถสื่อสารกับเทอร์มินัลหรือระบบอื่นโดยเพียงแค่ ดำเนินการคำสั่งหรือโปรแกรม จากนั้น ระบบจะรับผิดชอบงานส่วนที่เหลือ

ไอเท็ม	คำอธิบาย
chnamsv	เปลี่ยนคอนฟิกูเรชันเซอร์วิสชื่อตาม **Transmission Control Protocol/Internet Protocol (TCP/IP)** บนโฮสต์
chprtsv	เปลี่ยนการกำหนดค่าเซอร์วิสการพิมพ์บนเครื่องไคลเอ็นต์หรือ เซิร์ฟเวอร์
hostent	จัดการรายการการแม็พแอดเดรสโดยตรงใน ฐานข้อมูลการกำหนดค่าระบบ
ifconfig	กำหนดคอนฟิกหรือแสดงพารามิเตอร์อินเตอร์เฟสเครือข่ายสำหรับเครือข่าย ที่ใช้ **TCP/IP**
mknamsv	กำหนดคอนฟิกเซอร์วิสชื่อตาม **TCP/IP** บนโฮสต์สำหรับไคลเอ็นต์
mkprtsv	กำหนดคอนฟิกเซอร์วิสการพิมพ์ตาม **TCP/IP** บนโฮสต์
mktcpip	ตั้งค่าที่จำเป็นสำหรับการเริ่มต้น **TCP/IP** บนโฮสต์
no	กำหนดคอนฟิกอ็อพชันเครือข่าย
rmnamsv	ยกเลิกกำหนดคอนฟิกเซอร์วิสชื่อตาม **TCP/IP** บนโฮสต์
rmprtsv	ยกเลิกกำหนดคอนฟิกเซอร์วิสการพิมพ์บนเครื่องไคลเอ็นต์หรือเซิร์ฟเวอร์
slattach	แนบบรรทัดอนุกรมเป็นอินเตอร์เฟสเครือข่าย
arp	แสดงหรือเปลี่ยนอินเทอร์เน็ตแอดเดรสในตารางการแปลฮาร์ดแวร์แอดเดรส ที่ใช้โดย **Address Resolution Protocol (ARP)**
gettable	รับค่าตารางโฮสต์รูปแบบ Network Information Center (NIC) จากโฮสต์
hostid	ตั้งค่าหรือแสดงตัวระบุของโลคัลโฮสต์ปัจจุบัน
hostname	ตัวค่าหรือแสดงชื่อของระบบโฮสต์ปัจจุบัน
htable	แปลงโฮสต์ไฟล์ไปเป็นรูปแบบที่ใช้โดยรูทีนไลบรารีเครือข่าย
ipreport	สร้างรายการการติดตามแพ็กเก็ตจากไฟล์ การติดตามแพ็กเก็ตที่ระบุ
iptrace	จัดให้มีการติดตามแพ็กเก็ตระดับอินเตอร์เน็ตสำหรับอินเทอร์เน็ตโปรโตคอล
lsnamsv	แสดงข้อมูลเซอร์วิสชื่อที่เก็บในฐานข้อมูล
lsprtsv	แสดงข้อมูลเซอร์วิสการพิมพ์ที่เก็บในฐานข้อมูล
mkhosts	สร้างไฟล์ตารางโฮสต์
namerslv	จัดการรายการเซิร์ฟเวอร์โดเมนเนมโดยตรง สำหรับรูทีน local resolver ในฐานข้อมูลการกำหนดค่าระบบ
netstat	แสดงสถานะเครือข่าย
route	จัดการตารางการจัดเส้นทางด้วยตนเอง
ruser	จัดการกับรายการในฐานข้อมูลระบบที่แยกจากกันสามฐานข้อมูลระบบโดยตรง ซึ่งควบคุมการโฮสต์อื่นๆ เข้าถึงโปรแกรมต่างๆ
ruptime	แสดงสถานะของโฮสต์แต่ละตัวที่อยู่บนเครือข่าย
securetcpip	เปิดใช้งานคุณลักษณะการรักษาความปลอดภัยเครือข่าย
setclock	ตั้งค่าเวลาและวันที่สำหรับโฮสต์บนเครือข่าย
timedc	ส่งกลับข้อมูลเกี่ยวกับ **timed** daemon
trpt	ทำการติดตามโปรโตคอลบนซ็อกเก็ต **Transmission Control Protocol (TCP)**

คำสั่ง SRC

คำสั่ง SRC สามารถมีผลกับหนึ่ง daemon, กลุ่มของ daemons หรือ daemon และ daemons ที่ควบคุม (ระบบย่อยที่มีเซิร์ฟเวอร์ย่อย)

นอกจากนั้น บาง **TCP/IP** daemons ไม่ตอบกลับคำสั่ง SRC ทั้งหมด ต่อไปนี้คือรายการลิสต์ SRC ที่สามารถใช้ ควบคุม **TCP/IP** daemons รวมถึงข้อยกเว้น

ไอเท็ม	คำอธิบาย
startsrc	เริ่มทำงานระบบย่อย **TCP/IP** และเซิร์ฟเวอร์ย่อย **inetd** ทั้งหมด คำสั่ง **startsrc** ทำงานกับระบบย่อย **TCP/IP**P และเซิร์ฟเวอร์ย่อย **inetd** ทั้งหมด
stopsrc	หยุดทำงานระบบย่อย **TCP/IP** และเซิร์ฟเวอร์ย่อย **inetd** ทั้งหมด คำสั่งนี้ยังถูกเรียกว่าคำสั่ง **stop normal** คำสั่ง **stop normal** อนุญาตให้ระบบย่อยประมวลผลงานที่คงเหลืออยู่ทั้งหมด และสิ้นสุดการทำงานโดยไม่มีปัญหา สำหรับเซิร์ฟเวอร์ย่อย **inetd** การเชื่อมต่อ ที่ค้างอยู่ทั้งหมดจะได้รับอนุญาตให้เริ่มทำงานและการเชื่อมต่อที่มีอยู่ทั้งหมดได้รับอนุญาต ให้ดำเนินการจนเสร็จ คำสั่ง **stop normal** ทำงานกับระบบย่อย **TCP/IP** และ เซิร์ฟเวอร์ย่อย **inetd** ทั้งหมด
stopsrc –f	หยุดทำงานระบบย่อย **TCP/IP** และเซิร์ฟเวอร์ย่อย **inetd** ทั้งหมด คำสั่งนี้ยังถูกเรียกว่า **stop force** คำสั่ง **stop force** ยุติการทำงานของระบบย่อยทั้งหมดในทันที สำหรับเซิร์ฟเวอร์ย่อย **inetd** การเชื่อมต่อที่ค้างอยู่ทั้งหมดและการเชื่อมต่อที่มีอยู่แล้วจะถูกยุติการทำงานในทันที
refresh	รีเฟรชระบบย่อยและเซิร์ฟเวอร์ย่อยต่อไปนี้: เซิร์ฟเวอร์ย่อย **inetd, syslogd, named, dhcpsd** และ **gated**
lssrc	แสดงสถานะแบบย่อสำหรับระบบย่อย ซึ่งเป็นสถานะของระบบย่อย ที่ระบุ (แอ็คทีฟ หรือไม่มีการทำงาน) รวมทั้งแสดงสถานะแบบย่อสำหรับเซิร์ฟเวอร์ย่อย **inetd** สถานะแบบย่อสำหรับเซิร์ฟเวอร์ย่อย **inetd** ได้แก่: ชื่อ เซิร์ฟเวอร์ย่อย, สถานะ, รายละเอียดเซิร์ฟเวอร์ย่อย, ชื่อคำสั่ง และอาร์กิวเมนต์ที่ถูก เรียกใช้พร้อมคำสั่ง
lssrc –l	แสดงสถานะแบบย่อ รวมถึงข้อมูลเพิ่มเติม (สถานะแบบยาว) สำหรับระบบย่อยต่อไปนี้: **gated** ดีบักของการดีบัก หรือการติดตาม โปรโตคอลการจัดเส้นทางที่ถูกเรียกทำงาน ตารางการจัดเส้นทาง สัญญาณที่รับ และฟังก์ชันของสัญญาณ **inetd** สถานะของการดีบัก รายการเซิร์ฟเวอร์ย่อยที่แอ็คทีฟพร้อมสถานะแบบย่อ สัญญาณ ที่รับและฟังก์ชันของสัญญาณ **named** สถานะของการดีบัก, ข้อมูลไฟล์ `named.conf` **dhcpsd** สถานะของการดีบัก, IP addresses ที่ควบคุมทั้งหมดและสถานะปัจจุบัน **routed** สถานะของการดีบักและการติดตาม, สถานะของการระบุข้อมูลการจัดเส้นทาง, ตาราง การจัดเส้นทาง **syslogd** ข้อมูลการกำหนดค่า **syslogd** คำสั่ง **lssrc –l** ยังแสดง สถานะแบบยาวสำหรับเซิร์ฟเวอร์ย่อย **inetd** สถานะแบบยาวประกอบด้วย ข้อมูลสถานะแบบย่อ และข้อมูลการเชื่อมต่อที่แอ็คทีฟ บางเซิร์ฟเวอร์ย่อย จะมีข้อมูลเพิ่มเติม ข้อมูลเพิ่มเติมโดยเซิร์ฟเวอร์ย่อย ได้แก่: **ftpd** สถานะของการดีบักและการบันทึกการทำงาน **telnetd** ชนิดของการอีมูเลตเทอร์มินัล **rlogind** สถานะของการดีบัก **fingerd** สถานะของการดีบักและการบันทึกการทำงาน เซิร์ฟเวอร์ย่อย **rwhod** และ**timed** ไม่มีข้อมูลสถานะแบบยาวให้
traceson	เปิดใช้งานการดีบักระดับของซ็อกเก็ต ใช้คำสั่ง **trpt** เพื่อจัดรูปแบบเอาต์พุต ระบบย่อย **timed** และ **iptraced** ไม่สนับสนุนการใช้คำสั่ง **traceson**
tracesoff	ปิดการดีบักระดับซ็อกเก็ต ใช้คำสั่ง **trpt** เพื่อจัดรูปแบบเอาต์พุต ระบบย่อย **timed** และ **iptraced** ไม่สนับสนุนการใช้คำสั่ง **tracesoff**

สำหรับตัวอย่างวิธีการใช้คำสั่งเหล่านี้ โปรดดูหัวข้อ เกี่ยวกับแต่ละคำสั่ง สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ System Resource Controller ดูที่ System Resource Controller ใน *การจัดการระบบปฏิบัติการและอุปกรณ์*

คำสั่งโอนย้ายไฟล์

คำอธิบายย่อของคำสั่งโอนย้ายไฟล์แสดงอยู่ที่นี่

ไอเท็ม	คำอธิบาย
ftp *ชื่อโฮสต์*	โอนย้ายไฟล์ระหว่างโลคัลและรีโมตโฮสต์
rcp*file host:file*	โอนย้ายไฟล์ระหว่างโลคัลและรีโมตโฮสต์ หรือระหว่างสองรีโมตโฮสต์
tftp	โอนย้ายไฟล์ระหว่างโฮสต์

คำสั่งล็อกอินรีโมต

คำอธิบายย่อของคำสั่งล็อกอินรีโมต **TCP/IP** แสดงอยู่ ที่นี่

ไอเท็ม	คำอธิบาย
rexec *คำสั่งโฮสต์*	เรียกใช้งานคำสั่งทีละคำสั่งบนรีโมตโฮสต์
rlogin *remotehost*	เชื่อมต่อโฮสต์บนโลคัลกับโฮสต์แบบรีโมต
คำสั่ง **rsh** และ **remsh** *remotehost*	ดำเนินการคำสั่งที่ระบุที่รีโมตโฮสต์ หรือล็อกอินเข้าสู่รีโมตโฮสต์
telnet, tn และ **tn3270** *hostname*	เชื่อมต่อโลคัลโฮสต์กับรีโมตโฮสต์โดยใช้อินเตอร์เฟส **TELNET**

คำสั่ง Status

คำอธิบายย่อของคำสั่งสถานะ **TCP/IP** แสดงไว้ที่นี่

ไอเท็ม	คำอธิบาย
finger หรือ **f** *user@host*	แสดงข้อมูลผู้ใช้
host *hostname*	เปลี่ยนชื่อโฮสต์ให้เป็นอินเตอร์เน็ตแอดเดรส หรืออินเตอร์เน็ตแอดเดรสเป็น ชื่อโฮสต์
ping *hostname*	ส่งการร้องขอเอกโคไปยังเน็ตเวิร์กโฮสต์
rwho	แสดงผู้ใช้ที่ล็อกอินเข้าสู่โฮสต์บนโลคัลเน็ตเวิร์ก
whois *name*	ระบุผู้ใช้โดยใช้ ID ผู้ใช้ หรือ alias

คำสั่งการสื่อสารรีโมต

คำสั่งการสื่อสารรีโมต **TCP/IP**, **talk** *User@Host*, ช่วยให้คุณสามารถสนทนากับผู้ใช้รายอื่นได้

ไอเท็ม	คำอธิบาย
talk *User@Host*	สนทนากับผู้ใช้รายอื่น

คำสั่งการพิมพ์

คำอธิบายย่อของคำสั่งการพิมพ์ **TCP/IP** แสดงอยู่ ที่นี่

ไอเท็ม	คำอธิบาย
enq *ไฟล์*	จัดคิวไฟล์
refresh	ร้องขอให้รีเฟรชระบบย่อย หรือกลุ่มของระบบย่อย
smit	ดำเนินการจัดการระบบ

TCP/IP daemons

ระบบย่อย คือ daemon หรือเซิร์ฟเวอร์ที่ควบคุมโดย SRC *เซิร์ฟเวอร์ย่อย* คือ daemon ที่ควบคุมโดยระบบย่อย (โดยปกติ คำสั่ง daemon และชื่อ daemon มีการบ่งชี้โดย **d** ที่ ตอนท้ายของชื่อ)

หมวดหมู่ของระบบย่อยและเซิร์ฟเวอร์ย่อยไม่มีความเกี่ยวข้องซึ่งกันและกัน นั่นคือ daemons ไม่มีการแสดงรายการเป็นทั้งระบบย่อยและเป็นเซิร์ฟเวอร์ย่อย ระบบย่อย **TCP/IP** เดียวที่ควบคุม daemons อื่นคือ **inetd** daemon เซิร์ฟเวอร์ย่อย **TCP/IP** ทั้งหมดยังเป็นเซิร์ฟเวอร์ย่อย **inetd** ด้วย

ข้อมูลต่อไปนี้เป็น **TCP/IP** daemons ที่ควบคุมโดย SRC:

ระบบย่อย

ไอเท็ม	คำอธิบาย
gated	นำเสนอฟังก์ชันการเราต์เกตเวย์และสนับสนุนโปรโตคอล **Routing Information Protocol (RIP)**, **Routing Information Protocol Next Generation (RIPng)**, **Exterior Gateway Protocol (EGP)**, **Border Gateway Protocol (BGP)** และ **BGP4+**, **Defense Communications Network Local-Network Protocol (HELLO)**, **Open Shortest Path First (OSPF)**, Intermediate System to Intermediate System (IS-IS), และ **Internet Control Message Protocol (ICMP** และ **ICMPv6)/ Router Discovery routing** นอกจากนั้น **gated** daemon ยังสนับสนุน **Simple Network Management Protocol (SNMP) gated** daemon เป็นหนึ่งในสอง daemons การเราต์ที่มีอยู่สำหรับการเราต์ไปยังแอดเดรสเครือข่าย และเป็น daemon การเราต์ที่แนะนำ แนะนำให้ใช้ **gated** daemon มากกว่า **routed** daemon เนื่องจาก **gated** daemon สนับสนุนเกตเวย์โปรโตคอลมากกว่า
inetd	เรียกใช้และจัดตารางเวลา daemons อื่นเมื่อได้รับคำร้องขอเซอร์วิส daemon Daemon นี้ยังสามารถเริ่มต้น daemons อื่นได้ด้วย **inetd** daemon รู้จักกันในอีกชื่อหนึ่งว่า super daemon
iptrace	จัดเตรียมฟังก์ชันการติดตามแพ็กเก็ตระดับอินเตอร์เฟสสำหรับอินเทอร์เน็ตโปรโตคอล
named	จัดเตรียมฟังก์ชันการตั้งชื่อสำหรับโปรโตคอล **Domain Name Server** (DOMAIN)
routed	จัดการตารางการเราต์เครือข่ายและสนับสนุน **Routing Information Protocol (RIP)** แนะนำให้ใช้ **gated** daemon มากกว่า **routed** daemon เนื่องจาก **gated** daemon สนับสนุนเกตเวย์โปรโตคอลมากกว่า
rwhod	ส่งการแพร่สัญญาณไปยังโฮสต์อื่นทั้งหมดทุกสามนาที และจัดเก็บ ข้อมูลเกี่ยวกับผู้ใช้ที่ล็อกอินและสถานะเครือข่าย ควรใช้ **rwhod** daemon ด้วยความระมัดระวังอย่างยิ่ง เนื่องจากอาจใช้รีซอร์สของเครื่องเป็นจำนวนมาก
timed	จัดเตรียมฟังก์ชันเซิร์ฟเวอร์เวลา

หมายเหตุ: ทั้ง **routed** และ **gated** daemons มีการแสดงรายการเป็นระบบย่อย TCP/IP อย่ารันคำสั่ง **startsrc -g tcpip** ซึ่งเริ่มต้น daemons การเราต์ทั้งสองรายการเหล่านี้ ควบคู่ไปกับระบบย่อย **TCP/IP** อื่นทั้งหมด การรันทั้งสอง daemons พร้อมกันบนเครื่องเดียวอาจทำให้เกิดผลลัพธ์ ที่คาดการณ์ไม่ได้

TCP/IP daemons ที่ควบคุมโดยระบบย่อย **inetd** มีดังต่อไปนี้:

เซิร์ฟเวอร์ย่อย inetd

ไอเท็ม	คำอธิบาย
comsat	แจ้งให้ผู้ใช้ทราบว่ามีเมลเข้า
fingerd	จัดเตรียมรายงานสถานะของผู้ใช้ที่ล็อกอินทั้งหมดและสถานะเครือข่าย ที่รีโมตโฮสต์ที่ระบุ Daemon นี้ใช้โปรโตคอล **Finger**
ftpd	จัดเตรียมฟังก์ชันการโอนย้ายไฟล์สำหรับไคลเอ็นต์โปรเซสโดยใช้ **File Transfer Protocol (FTP)**
rexecd	จัดเตรียมฟังก์ชันโฮสต์เซิร์ฟเวอร์ต่างประเทศสำหรับคำสั่ง **rexec**
rlogind	จัดเตรียมฟังก์ชันสิ่งอำนวยความสะดวกรีโมตล็อกอินสำหรับคำสั่ง **rlogin**
rshd	จัดเตรียมฟังก์ชันเซิร์ฟเวอร์การดำเนินการคำสั่งรีโมตสำหรับคำสั่ง **rcp** และ **rsh**
talkd	จัดเตรียมฟังก์ชันการสนทนาสำหรับคำสั่ง **talk**
syslogd	อ่านและบันทึกข้อความระบบ Daemon นี้อยู่ในกลุ่ม **Remote Access Service** (RAS) ของระบบย่อย
telnetd	จัดเตรียมฟังก์ชันเซิร์ฟเวอร์สำหรับโปรโตคอล **TELNET**
tftpd	จัดเตรียมฟังก์ชันเซิร์ฟเวอร์สำหรับ **Trivial File Transfer Protocol (TFTP)**
uucpd	จัดการการสื่อสารระหว่าง Basic Network Utilities (BNU) และ **TCP/IP**

เมธอด Device

เมธอด Device คือโปรแกรมที่เชื่อมโยงกับอุปกรณ์ที่ดำเนิน การกำหนดค่าอุปกรณ์พื้นฐาน

ดูที่ รายชื่อ TCP/IP Programming References ใน *หลักการเขียนโปรแกรมการสื่อสาร* สำหรับ ดูข้อมูลเกี่ยวกับเมธอด **TCP/IP**

การร้องขอความคิดเห็น

TCP/IP Request for Comments (RFCs) ต่อไปนี้ได้รับการสนับสนุน ในระบบ AIX

สำหรับรายการ RFCs (Request for Comments) ที่ระบบปฏิบัติการ นี้สนับสนุน ดูที่ List of TCP/IP Programming References ใน *หลักการเขียนโปรแกรมการสื่อสาร*

- RFC 1359 *การเชื่อมต่ออินเตอร์เน็ต: ข้อปฏิบัติการเชื่อมต่อใดที่ควร คาดหวัง*
- RFC 1325 *FYI ของคำถามและคำตอบ: ตอบคำถาม 'ผู้ใช้ อินเตอร์เน็ตมือใหม่' ทั่วไป*
- RFC 1244 *Site Security Handbook*
- RFC 1178 *การเลือกชื่อสำหรับคอมพิวเตอร์ของคุณ*
- RFC 1173 *ความรับผิดชอบของผู้จัดการโฮสต์และเน็ตเวิร์ก: สรุป `ประวัติที่เล่าต่อๆ กันมา' เกี่ยวกับอินเตอร์เน็ต*

Basic Networking Utilities

BNUs เป็นกลุ่มของโปรแกรม ไดเร็กทอรี และไฟล์ที่สร้างการสื่อสารระหว่างระบบคอมพิวเตอร์บนเน็ตเวิร์กแบบโลคัลและรีโมต มันสามารถใช้เพื่อสื่อสารกับระบบ UNIX ใดๆที่เวอร์ชันของ UNIX-to-UNIX Copy Program (UUCP) รันอยู่ BNU เป็นหนึ่งในโปรแกรมการบริการที่ถูกขยายที่สามารถถูกติดตั้งกับระบบปฏิบัติการฐาน

กลุ่มของคำสั่งที่สัมพันธ์กับโปรแกรมการสื่อสาร UUCP UNIX-to-UNIX ที่ถูกพัฒนาโดย AT&T และถูกแก้ไขโดยเป็นส่วนของ Berkeley Software Distribution (BSD) ที่อยู่ใน BNU BNU จัดเตรียมคำสั่ง กระบวนการ และฐานข้อมูลการสนับสนุนสำหรับเชื่อมต่อกับระบบโลคัลและรีโมต เน็ตเวิร์กการสื่อสาร เช่น Token-Ring และ Ethernet ถูกใช้เพื่อเชื่อมต่อระบบบนเน็ตเวิร์กแบบโลคัล เน็ตเวิร์กแบบโลคัลสามารถเชื่อมต่อกับระบบรีโมตโดยการเชื่อมต่อแบบ hardwire หรือโมเด็ม จากนั้นคำสั่งและไฟล์สามารถถูกแลกเปลี่ยนระหว่างระบบแบบโลคัลและระบบแบบรีโมต

ก่อนที่ผู้ใช้บนระบบของคุณสามารถรันโปรแกรม BNU BNU ต้องถูกติดตั้งและตั้งค่า

BNU ถูกควบคุมโดยชุดของไฟล์ที่กำหนดว่าระบบรีโมตสามารถล็อกอินยังระบบโลคัลและสามารถทำอะไรหลังจากล็อกอินแล้ว ไฟล์คอนฟิกูเรชันเหล่านี้ต้องถูกตั้งค่าตามข้อกำหนดและรีซอร์สของระบบของคุณ

เพื่อรักษา BNU ไว้คุณต้องการและลบล็อกไฟล์เป็นระยะๆ และตรวจสอบคิวของ BNU เพื่อให้แน่ใจว่างานถูกถ่ายโอนไปบยังระบบรีโมตอย่างถูกต้อง คุณยังต้องอัพเดตไฟล์คอนฟิกูเรชันเป็นระยะๆ เพื่อแสดงถึงการเปลี่ยนแปลงในระบบของคุณหรือระบบรีโมต

BNU ทำงานอย่างไร

BNU ใช้ชุดของฮาร์ดแวร์การเชื่อมต่อและซอฟต์แวร์โปรแกรมเพื่อสื่อสารระหว่างระบบ

โครงสร้างของไดเร็กทอรีและไฟล์จะติดตามกิจกรรมของ BNU โครงสร้างนี้จะรวมชุดของพับลิกไดเร็กทอรี กลุ่มของไดเร็กทอรี และไฟล์การจัดการ ไฟล์คอนฟิกูเรชัน และไฟล์การล็อก ไดเร็กทอรีส่วนใหญ่สำหรับ BNU ถูกสร้างระหว่างกระบวนการติดตั้ง บางไดเร็กทอรีและไฟล์การจัดการถูกสร้างโดยโปรแกรม BNU ต่างๆ

โดยเป็นข้อยกเว้นของคำสั่งการล็อกอินแบบรีโมต BNU จะทำงานเป็นระบบแบบแบ็ตช์ เมื่อผู้ใช้ร้องขอให้ส่งงานไปยังระบบรีโมต BNU จะเก็บข้อมูลที่ต้องการเพื่อทำงานให้สำเร็จ นี่รู้จักว่าเป็นการ*เข้าคิว*งาน เมื่อเวลาที่ถูกกำหนด หรือเมื่อผู้ใช้บอกให้มันทำ BNU จะติดต่อกับระบบรีโมตต่างๆ ถ่ายโอนงานที่ถูกเข้าคิว และยอมรับงาน การถ่ายโอนเหล่านี้ถูกควบคุมโดยไฟล์คอนฟิกูเรชันบนระบบของคุณ และระบบรีโมตเหล่านั้น

การสนับสนุนภาษาประจำชาติสำหรับคำสั่ง BNU

คำสั่ง BNU ทั้งหมด ยกเว้น **uucpadm** มีให้ใช้งานสำหรับการสนับสนุนภาษาประจำชาติ

ชื่อผู้ใช้ไม่ต้องเป็นอักขระ ASCII อย่างไรก็ตามชื่อระบบทั้งหมดต้องเป็นอักขระ ASCII ถ้าผู้ใช้พยายามกำหนดเวลาการถ่ายโอน หรือการประมวลผลคำสั่งแบบรีโมตเกี่ยวข้องกับชื่อระบบทที่ไม่ใช่ ASCII BNU จะส่งข้อความข้อผิดพลาดกลับคืนมา

โครงสร้างไฟล์และไดเร็กทอรีของ BNU

BNU ใช้โครงสร้างของไดเร็กทอรีและไฟล์เพื่อติดตามกิจกรรม

โครงสร้างนี้รวมพับลิกไดเร็กทอรี ไฟล์คอนฟิกูเรชัน ไดเร็กทอรีการจัดการ และล็อกไฟล์

ไดเร็กทอรีส่วนใหญ่สำหรับ BNU ถูกสร้างระหว่างกระบวนการติดตั้ง บางไดเร็กทอรีและไฟล์การจัดการถูกสร้างโดยโปรแกรม BNU ต่างๆ เมื่อมันรัน

พับลิกไดเร็กทอรี BNU

เมื่อถูกระบุ พับลิกไดเร็กทอรี BNU (`/var/spool/uucppublic`) จะเก็บไฟล์ที่ถูกถ่ายโอนมายังระบบโลคัลจากระบบอื่น

ไฟล์จะรอในพับลิกไดเร็กทอรีจนกว่าผู้ใช้จะเรียกใช้มัน พับลิกไดเร็กทอรีถูกสร้างเมื่อ BNU ถูกติดตั้ง ภายใน พับลิกไดเร็กทอรี BNU จะสร้างไดเร็กทอรีย่อยสำหรับแต่ระระบบรีโมตที่ส่งไฟล์ มายังระบบโลคัล

ไฟล์คอนฟิกูเรชัน BNU

ไฟล์คอนฟิกูเรชัน BNU ยังถูกรู้จักว่าเป็นฐานข้อมูลการสนับสนุน BNU ที่อยู่ในไดเร็กทอรี `/etc/uucp` ไฟล์ต้องถูกตั้งค่าเป็นพิเศษสำหรับระบบของคุณ

มันถูกเป็นเจ้าของโดยล็อกอิน ID uucp และสามารถถูกแก้ไขโดยใช้สิทธิของ root ไฟล์คอนฟิกูเรชันประกอบด้วยข้อมูลเกี่ยวกับ:

- ระบบรีโมตที่สามารถเข้าถึงได้
- อุปกรณ์สำหรับติดต่อกับระบบรีโมต
- เวลาที่ติดต่อกับระบบรีโมต
- อะไรที่ระบบรีโมตยอมให้ทำบนระบบของคุณ

ไฟล์คอนฟิกูเรชันบางไฟล์ยังระบุข้อจำกัดบนกิจกรรมของ BNU ที่ป้องกันไม่ให้รับของคุณทำงานมากเกินไป

ไฟล์คอนฟิกูเรชัน BNU ประกอบด้วย:

ไอเท็ม	คำอธิบาย
Devices	ประกอบด้วยข้อมูลเกี่ยวกับอุปกรณ์ที่พร้อมใช้งาน รวมถึงโมเด็มและการเชื่อมต่อโดยตรง
Dialcodes	ประกอบด้วยตัวย่อของโค้ดการหมุนโทรศัพท์ ซึ่งให้คุณสามารถย่อหมายเลขโทรศัพท์ในไฟล์ Systems
Dialers	ระบุ syntax ของคำสั่งการโทรสำหรับชนิดของโมเด็มที่ระบุ ("dialer")
Maxuuscheds	จำกัดงานที่ถูกกำหนดเวลาให้ทำพร้อมกัน
Maxuuxqts	จำกัดการประมวลผลคำสั่งรีโมตที่ทำพร้อมกัน
Permissions	มีโค้ดสิทธิการเข้าถึง ไฟล์นี้เป็นไฟล์ลำดับแรกที่ระบุความปลอดภัยสำหรับ BNU
Poll	ระบุว่าเมื่อใดที่โปรแกรม BNU ควรโพลระบบรีโมตเพื่อเริ่มงาน
Sysfiles	จะลิสต์ไฟล์ที่ทำหน้าที่เป็นไฟล์ Systems, Devices, และ Dialers สำหรับการตั้งค่า BNU ถ้าไฟล์นี้ไม่ถูกใช้ ไฟล์ดีฟอลต์คือ /etc/uucp/Systems, /etc/uucp/Devices, และ /etc/uucp/Dialers
Systems	จะลิสต์ระบบรีโมตที่สามารถเข้าถึงและข้อมูลที่ต้องการเพื่อติดต่อกับมัน รวมถึงอุปกรณ์ที่จะใช้ และชื่อผู้ใช้และรหัสผ่านที่คุณต้องการเพื่อล็อกอิน นอกจากนี้ยังระบุเวลาที่ระบบสามารถถูกติดต่อ

ไฟล์คอนฟิกูเรชันจะอ้างอิงถึงกันเมื่อ BNU ถูกใช้ ตัวอย่างเช่น:

- ไฟล์ Devices ประกอบด้วยฟิลด์ *Token* ที่อ้างถึง entry ในไฟล์ Dialers
- ไฟล์ Systems ประกอบด้วย entry สำหรับ *คลาส* ของอุปกรณ์ อุปกรณ์ของแต่ละ *คลาส* ที่ถูกอ้างถึงในไฟล์ Systems ต้องถูกกำหนดในไฟล์ Devices
- ไฟล์ Poll ประกอบด้วย entry สำหรับระบบที่ระบบของคุณจะเรียกไป แต่ละระบบเหล่านี้ต้องถูกกำหนดในไฟล์ Systems

Entry ในไฟล์คอนฟิกูเรชัน BNU ขึ้นอยู่กับชนิดของการเชื่อมต่อระหว่างระบบของคุณและระบบรีโมตแต่ละระบบ ตัวอย่างเช่น entry พิเศษต้องถูกทำถ้าใช้ Transmission Control Protocol/Internet Protocol (TCP/IP) หรือการเชื่อมต่อโดยตรงเพื่อติดต่อกับระบบอื่น ถ้าโมเด็มถูกใช้เพื่อติดต่อกับระบบอื่น โมเด็มต้องถูกกำหนดในไฟล์ Dialers

ไฟล์ Systems, Devices และ Permissions ต้องถูกตั้งค่าบนระบบของคุณก่อนที่คุณจะสามารถติดต่อระบบรีโมตโดยใช้ BNU ไฟล์คอนฟิกูเรชันอื่นให้คุณสามารถใช้ความสามารถของ BNU เช่นการโพลโดยอัตโนมัติ ไฟล์คอนฟิกูเรชันหลายไฟล์ต้องถูกแก้ไขบ่อยๆ เพื่อแสดงถึงการเปลี่ยนแปลงของระบบของคุณหรือระบบที่คุณติดต่อ ไฟล์ Sysfiles สามารถถูกใช้เพื่อระบุไฟล์อื่นที่ไม่ใช่ไฟล์ดีฟอลต์ Systems, Devices และ Dialersเพื่อให้ได้บทบาทเดิม

ไดเร็กทอรีการจัดการและไฟล์ BNU

ไดเร็กทอรีการจัดการและไฟล์ BNU จะอยู่ในไดเร็กทอรีย่อยของไดเร็กทอรี /var/spool/uucp

ไดเร็กทอรีและไฟล์เหล่านี้ประกอบด้วยข้อมูล 2 ชนิด :

- ข้อมูลที่รอเพื่อถูกถ่ายโอนไปยังระบบอื่น
- ล็อกและข้อมูลข้อผิดพลาดเกี่ยวกับกิจกรรมของ BNU

ภายใต้ไดเร็กทอรี /var/spool/uucp BNU จะสร้างไดเร็กทอรีต่อไปนี้ :

ไอเท็ม	คำอธิบาย
.Admin	ประกอบด้วยไฟล์การจัดการ 4 ไฟล์ : • audit • Foreign • errors • xferstats ไฟล์เหล่านี้ประกอบด้วยล็อกและข้อมูลข้อผิดพลาดเกี่ยวกับกิจกรรมของ BNU
.Corrupt	ประกอบด้วยคัดลอกของไฟล์ที่ไม่สามารถถ๔กประมวลผลโดยโปรแกรม BNU
.Log และ .Old	ประกอบด้วยล็อกไฟล์จากรายการของ BNU
.Status	เก็บเวลาล่าสุดที่ **uucico** daemon พยายามติดต่อระบบรีโมต
.Workspace	เก็บไฟล์ชั่วคราวที่โปรแกรมการถ่ายโอนไฟล์ใช้ภายใน
.Xqtdir	ประกอบด้วยไฟล์ที่เรียกทำงานกับลิสต์ของคำสั่งที่ระบบรีโมตสามารถรัน
SystemName	ประกอบด้วยไฟล์ที่ถูกใช้โดยโปรแกรมการถ่ายโอนไฟล์ ไฟล์เหล่านี้คือ: • คำสั่ง (**C.***) • ข้อมูล (**D.***) • เรียกใช้งาน (**X.***) • ชั่วคราว (**TM.***) BNU จะสร้างไดเร็กทอรี *SystemName* สำหรับแต่ละระบบรีโมตที่มันติดต่อ

ไดเร็กทอรีที่ชื่อขึ้นต้นด้วยจุด คือ *ถูกซ่อน* มันจะไม่ถูกพบโดยคำสั่ง **ls** หรือ **li** นอกจากใช้แพล็ก **–a** เมื่อ **uucico** daemon ถูกสตาร์ท มันจะค้นหาไดเร็กทอรี /var/spool/uucp สำหรับไฟล์ที่ทำงานและถ่ายโอนไฟล์จากไดเร็กทอรีใดๆที่ไม่ถูกซ่อน **uucico** daemon จะเห็นเฉพาะไดเร็กทอรี *SystemName* ไม่ใช่ไดเร็กทอรีการจัดการอื่น

ไฟล์ในไดเร็กทอรีที่ถูกซ่อนจะเป็นเจ้าของโดยล็อกอิน ID uucp ไฟล์เหล่านี้สามารถถูกเข้าถึงได้เฉพาะสิทธิที่เป็น root หรือด้วยล็อกอิน ID ที่มี UID เป็น 5

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการบำรุงรักษาไดเร็กทอรีการจัดการ BNU ดูที่ “การบำรุงรักษา BNU” ในหน้า 478

ล็อกไฟล์ BNU

ล็อกไฟล์ BNU ถูกเก็บในไดเร็กทอรี /var/locks เมื่อ BNU ใช้อุปกรณ์เพื่อเชื่อมต่อกับคอมพิวเตอร์แบบรีโมต มันจะใส่ล็อกไฟล์สำหรับอุปกรณ์ในไดเร็กทอรี /var/locks

เมื่อโปรแกรม BNU อื่น หรือโปรแกรมอื่นใดๆต้องการอุปกรณ์โปรแกรมนั้นจะตรวจสอบไดเร็กทอรี /var/locks สำหรับล็อกไฟล์ ถ้าล็อกไฟล์มีอยู่โปรแกรมจะรอจนกว่าอุปกรณ์จะพร้อมใช้งาน หรือใช้อุปกรณ์อื่นสำหรับการสื่อสาร

นอกจากนี้ **uucico** daemon จะใส่ล็อกไฟล์สำหรับระบบรีโมตในไดเร็กทอรี /var/locks ก่อนที่จะติดต่อระบบรีโมต **uucico** daemon จะตรวจสอบไดเร็กทอรี /var/locks สำหรับล็อกไฟล์สำหรับระบบนั้น ไฟล์เหล่านี้ป้องกันอินสแตนซ์อื่นของ **uucico** daemon จากการสร้างการเชื่อมต่อไปยังระบบรีโมตเดียวกันซ้ำ

หมายเหตุ: ซอฟต์แวร์อื่นนอกจาก BNU เช่น Asynchronous Terminal Emulation (ATE) และ TCP/IP ใช้ไดเร็กทอรี /var/locks

การตั้งค่า BNU

โพรซีเดอร์นี้อธิบายวิธีการกำหนดคอนฟิก Basic Network Utilities (BNU) สำหรับการเชื่อมต่อชนิดต่างๆ เช่น โดยตรง โมเด็ม และการเชื่อมต่อ Transmission Control Protocol/Internet Protocol (TCP/IP)

ข้อกำหนดเบื้องต้น

- BNU ต้องถูกติดตั้งบนระบบของคุณ
- คุณต้องมีสิทธิผู้ใช้ root เพื่อแก้ไขไฟล์คอนฟิกูเรชัน BNU
- ถ้าคุณใช้การเชื่อมต่อโดยตรงสำหรับการสื่อสาร BNU ต้องตั้งค่า การเชื่อมต่อที่เหมาะสมระหว่างระบบของคุณและระบบรีโมต
- ถ้าคุณใช้โมเด็มสำหรับการสื่อสาร BNU คุณต้องติดตั้งและ กำหนดคอนฟิกแต่ละโมเด็ม
- ถ้าการเชื่อมต่อของคุณหนึ่งรายการขึ้นไปใช้ TCP/IP TCP/IP ต้อง รันอยู่ระหว่างระบบของคุณและระบบรีโมตที่เหมาะสม
- รวบรวมข้อมูลที่คุณต้องการเพื่อกำหนดคอนฟิก BNU (โปรดดู รายการต่อไปนี้) ข้อมูลนี้มีรายการของระบบรีโมต และรายการของอุปกรณ์และโมเด็มที่จะใช้เชื่อมต่อกับระบบ

การรวบรวมข้อมูลระบบที่ต้องการ

ก่อน คุณกำหนดคอนฟิก BNU ให้รวบรวมข้อมูลต่อไปนี้:

- สำหรับแต่ละ *ระบบรีโมต* ที่ระบบของคุณเรียก ให้รวบรวม ข้อมูลต่อไปนี้:
  - ชื่อระบบ
  - ชื่อล็อกอินที่ระบบของคุณจะใช้บนระบบรีโมต
  - รหัสผ่านสำหรับชื่อล็อกอิน
  - พร้อมต์ล็อกอินและรหัสผ่านบนระบบรีโมต
  - ชนิดของการเชื่อมต่อที่คุณใช้เพื่อเข้าถึงระบบรีโมต (โดยตรง โมเด็ม หรือ TCP/IP)

  ถ้าการเชื่อมต่อเป็นแบบโดยตรง ให้รวบรวมข้อมูลต่อไปนี้:
  - อัตราบิตของการเชื่อมต่อ
  - พอร์ตบนระบบโลคัลที่การเชื่อมต่อถูกเชื่อมอยู่

  ถ้าการเชื่อมต่อเป็นแบบผ่านโมเด็ม (การเชื่อมต่อโทรศัพท์) ให้รวบรวมข้อมูลต่อไปนี้:
  - หมายเลขโทรศัพท์ของระบบรีโมต
  - ความเร็วของโมเด็มของคุณเข้ากันได้กับความเร็วของ ระบบรีโมต

  หมายเหตุ: ถ้าระบบรีโมตเรียกระบบของคุณ ตรวจสอบให้แน่ใจว่า ผู้ดูแลระบบ BNU บนระบบรีโมตแต่ละระบบมีข้อมูลก่อนหน้าทั้งหมด เกี่ยวกับระบบของคุณ
- สำหรับแต่ละ *โลคัลโมเด็ม* ที่คุณใช้สำหรับการเชื่อมต่อ BNU ให้รวบรวมข้อมูลต่อไปนี้:
  - สคริปต์การพูดคุยสำหรับโมเด็ม (ศึกษาเอกสารคู่มือของโมเด็ม)

    หมายเหตุ: สำหรับ บางโมเด็ม สคริปต์การพูดคุยมีอยู่ในไฟล์ `/etc/uucp/Dialers`
  - พอร์ตแบบโลคัลของโมเด็ม

การสร้างรายการของอุปกรณ์ระบบ

ใช้ข้อมูล ที่คุณรวบรวมเพื่อจัดทำรายการของแต่ละอุปกรณ์ระบบ ซึ่งคุณต้องการ เชื่อมต่อกับระบบรีโมต ต่อไปนี้เป็นรายการตัวอย่างสำหรับ ระบบโลคัล morgan:

```
direct:
hera 9600 tty5
zeus& 2400 tty2
ariadne 2400 tty1
hayes modem (tty3):  apollo, athena
TCP/IP:  merlin, arthur, percy
```

ใน ตัวอย่างก่อนหน้านี้ เพื่อเชื่อมต่อกับระบบ hera มีการใช้การเชื่อมต่อ direct ที่ความเร็ว 9600 จาก พอร์ต tty5 เพื่อเชื่อมต่อกับระบบ apollo มีการใช้โมเด็ม hayes ซึ่งเชื่อมต่อกับพอร์ต tty3 ระบบใช้ TCP/IP เพื่อเชื่อมต่อกับระบบ merlin, arthur และ percy

การกำหนดคอนฟิกสิ่งอำนวยความสะดวกสำหรับการสื่อสารแบบรีโมต

เพื่อให้ BNU ทำงานได้อย่างถูกต้องที่ไซต์ของคุณ คุณต้องกำหนดคอนฟิกสิ่งอำนวยความสะดวกสำหรับ การสื่อสารแบบรีโมต ดังนี้:

- แสดงรายการอุปกรณ์ที่ใช้เพื่อสร้างลิงก์การสื่อสารแบบโดยตรง ผ่านโทรศัพท์ หรือผ่านโมเด็ม
- แสดงรายการโมเด็มที่ใช้เพื่อติดต่อระบบรีโมตผ่าน เครือข่ายโทรศัพท์
- ลิสต์ของระบบรีโมตที่สามารถเข้าถึงได้
- แสดงรายการตัวย่อที่ใช้แสดงแทนคำเติมหน้าของหมายเลขโทรศัพท์ ซึ่งใช้เพื่อติดต่อระบบรีโมตที่ระบุ (ทางเลือก)
- ตั้งค่าสิทธิเข้าถึงโดยระบุวิธีที่ระบบโลคัลและ รีโมตใช้สื่อสารกัน
- กำหนดเวลาการมอนิเตอร์สำหรับระบบรีโมตที่เป็นเน็ตเวิร์ก (อ็อพชัน)

เมื่อต้องการสร้างรายการเหล่านี้ สิทธิ และตารางเวลา ให้ทำ ขั้นตอนต่อไปนี้:

- เปลี่ยนไฟล์คอนฟิกูเรชัน BNU
- แก้ไขไฟล์ /var/spool/cron/crontabs/uucp เพื่อลบอักขระข้อคิดเห็น (#) ออกจากตอนต้นของบรรทัด ที่จัดตารางเวลารูทีนการบำรุงรักษาแบบอัตโนมัติ

หมายเหตุ: คุณต้องกำหนดคอนฟิกไฟล์ *ระบบ*, *อุปกรณ์* และ *สิทธิ* เพื่อให้แน่ใจว่า BNU รัน อย่างถูกต้องที่ไซต์ของคุณ อย่างไรก็ตาม ไม่จำเป็นต้องเปลี่ยน ไฟล์คอนฟิกูเรชัน BNU ในลำดับเฉพาะใดๆ

หลังจาก คุณทำโพรซีเดอร์ก่อนหน้าเสร็จสมบูรณ์แล้ว คุณสามารถกำหนดคอนฟิก BNU บนระบบ ของคุณ

การกำหนดคอนฟิก BNU บนระบบของคุณ

เมื่อต้องการกำหนดคอนฟิก BNU ให้ทำขั้นตอนต่อไปนี้:

1. ตรวจสอบให้แน่ใจว่า BNU มีการติดตั้งไว้บนระบบของคุณโดยรัน คำสั่งต่อไปนี้:

   ```
   lslpp -h bos.net.uucp
   ```

   ถ้า BNU มีการติดตั้งไว้ bos.net.uucp จะแสดงขึ้นใน เอาต์พุต ถ้าคุณไม่เห็น ให้ติดตั้ง BNU จากเทปการติดตั้ง

2. ตั้งค่า IDs และรหัสผ่านล็อกอินที่เหมาะสมสำหรับระบบรีโมต ซึ่งเรียกระบบของคุณ และจัดเตรียมล็อกอินและรหัสผ่านให้แก่บุคคลที่รับผิดชอบในการ ควบคุมดูแล BNU หรือ UNIX-to-UNIX Copy Program (UUCP) บนแต่ละ ระบบรีโมต ขั้นตอนนี้เสร็จสมบูรณ์ได้โดยแก้ไขไฟล์ /etc/passwd, /etc/group, /etc/security/login.cfg และ /etc/security/passwd

   ข้อควรสนใจ: ถ้าคุณอนุญาตให้ระบบรีโมตล็อกอินเข้าสู่ ระบบโลคัลโดยใช้ ID ล็อกอิน UUCP ความปลอดภัยของระบบจะ มีความเสี่ยง ระบบรีโมตที่ล็อกอินด้วย UUCP ID สามารถแสดง และอาจจะเปลี่ยนไฟล์ ระบบ และ สิทธิ แบบโลคัล แอ็คชันเหล่านี้ของระบบรีโมตขึ้นอยู่กับสิทธิที่ ระบุในรายการ LOGNAME ของไฟล์ สิทธิ ขอ แนะนำให้คุณสร้าง IDs ล็อกอิน BNU อื่นสำหรับระบบรีโมต และสงวน ID ล็อกอิน UUCP ไว้สำหรับบุคคลที่ควบคุมดูแล BNU บน ระบบโลคัล เพื่อให้ได้ความปลอดภัยที่ดีที่สุด แต่ละระบบรีโมตที่ติดต่อกับ ระบบโลคัลต้องมี ID ล็อกอินซึ่งไม่ซ้ำกันที่มีหมายเลข ID ผู้ใช้ (UID) ซึ่งไม่ซ้ำกัน IDs ล็อกอินเหล่านี้ต้องมี ID กลุ่ม (GIDs) เป็น 5 โดยดีฟอลต์ ระบบปฏิบัติการมี ID ล็อกอิน NUUCP สำหรับการโอนย้าย ไฟล์

   a. ถ้าคุณต้องการรักษาการควบคุมที่สมบูรณ์ในการเข้าถึง ของแต่ละระบบ คุณต้องสร้าง ID ล็อกอินที่แยกต่างหาก และรวมรายการ MACHINE และ LOGNAME ไว้ในไฟล์ สิทธิ คุณมีอ็อพชันในการเก็บล็อกอินที่แยกต่างหาก หรือมีหนึ่งล็อกอินสำหรับการเชื่อมต่อ BNU ทั้งหมด ตัวอย่างรายการ /etc/passwd มีดังนี้:

      ```
      Umicrtk:!:105:5:micrtk uucp:/usr/spool/uucppublic:/usr/sbin/uucp/uucico
      Ufloydl:!:106:5:floydl uucp:/usr/spool/uucppublic:/usr/sbin/uucp/uucico
      Uicus:!:107:5:icus uucp:/usr/spool/uucppublic:/usr/sbin/uucp/uucico
      Urisctkr:!:108:5::/usr/spool/uucppublic:/usr/sbin/uucp/uucico
      ```

   b. ถ้าคุณต้องการมีชุดของสิทธิหนึ่งชุด และไม่ต้องการ มีการควบคุมที่แยกต่างหากสำหรับการเชื่อมต่อ UUCP ใดๆ คุณสามารถมีล็อกอินเดียวสำหรับระบบทั้งหมด รายการตัวอย่าง สำหรับสถานการณ์จำลองดังกล่าวมีดังนี้:

      ```
      nuucp:!:6:5::/usr/spool/uucppublic:/usr/sbin/uucp/uucico
      ```

      หมายเหตุ:

      - UID ซึ่งเป็นฟิลด์ลำดับสามที่คั่นด้วยโคลอน ต้องไม่ซ้ำกัน เพื่อหลีกเลี่ยงความเสี่ยงด้านความปลอดภัย
      - GID ซึ่งเป็นฟิลด์ลำดับสี่ที่คั่นด้วยโคลอน ต้องเป็น 5 เพื่อให้แน่ใจว่าอยู่ในกลุ่มเดียวกันกับ UUCP
      - ไดเร็กทอรีโฮม ซึ่งเป็นฟิลด์ลำดับหกที่คั่นด้วยโคลอน สามารถเปลี่ยนเป็นไดเร็กทอรีที่ถูกต้องใดๆ
      - ล็อกอินเชลล์ ซึ่งเป็นฟิลด์ลำดับเจ็ดที่คั่นด้วยโคลอน ต้อง เป็น /usr/sbin/uucp/uucico เสมอ

   c. ตรวจสอบให้แน่ใจว่าไฟล์ /etc/group มีผู้ใช้ใหม่ ตัวอย่างของรายการดังกล่าวมีดังนี้:

      ```
      uucp:!:5:uucp,uucpadm,nuucp,Umicrtk,Uicus,Urisctakr
      ```

   d. เพิ่มผู้ใช้ลงในกลุ่ม UUCP ที่ใช้โมเด็มเพื่อเชื่อมต่อ กับโปรแกรมแทนที่จะใช้คำสั่ง **cu**

   e. หลังจากคุณแก้ไขไฟล์เหล่านี้เป็น root แล้ว ให้ตั้งรหัสผ่านสำหรับ ผู้ใช้ใหม่โดยใช้คำสั่ง **passwd** *UserName*

      หมายเหตุ: ถ้าคุณเปลี่ยนรหัสผ่านจาก root ล็อกอิน รายการ แฟล็กใน stanza สำหรับผู้ใช้ในไฟล์ /etc/security/passwd จะมีบรรทัดต่อไปนี้:

      ```
      flags = ADMCHG
      ```

      คุณ ต้องเปลี่ยนบรรทัดก่อนหน้านี้ ดังแสดงในตัวอย่างต่อไปนี้:

      ```
      flags =
      ```

      ไม่เช่นนั้น เมื่อกระบวนการ **uucico** แบบรีโมต ล็อกอินเข้าสู่ระบบของคุณ ระบบจะพร้อมต์ให้ป้อนรหัสผ่านใหม่ แอ็คชันนี้ไม่สามารถเป็นไปได้ ดังนั้น ล็อกอินจึงล้มเหลว

f. เพื่อหลีกเลี่ยงการอินเตอร์รัปต์ในกระบวนการล็อกอินที่มีสาเหตุ มาจากกระบวนการ **uucico** ซึ่งอาจเริ่มต้นขึ้น โดยดีฟอลต์ herald ด้วย Ctrl-J's ทั้งหมด ให้แสดงข้อคิดเห็นเกี่ยวกับดีฟอลต์ stanza (ด้วยดอกจัน) และกำหนด stanza สำหรับ tty ของคุณ ดังแสดง ในตัวอย่างต่อไปนี้:

```
/dev/tty0:
            herald = "\nrisc001 login:"
```

g. ใช้เท็กซ์เอดิเตอร์ ASCII หรือคำสั่ง **uucpadm** เพื่อแก้ไขไฟล์ `Poll` เพิ่มรายการสำหรับแต่ละ ระบบที่ระบบของคุณจะโพล

หมายเหตุ: ระบบที่ แสดงรายการในไฟล์ `Poll` ยังต้องแสดงรายการ ในไฟล์ `/etc/uucp/Systems` ด้วย

h. ใช้เท็กซ์เอดิเตอร์ ASCII เพื่อแก้ไขไฟล์ `/var/spool/cron/crontabs/uucp` ลบอักขระหมายเหตุ (#) จากบรรทัดที่รันคำสั่ง **uudemon.hour** และ **uudemon.poll** คุณสามารถเปลี่ยนจำนวนครั้งที่รันคำสั่งเหล่านี้ อย่างไรก็ตาม ต้องแน่ใจว่าจัดตารางเวลาคำสั่ง **uudemon.poll** ประมาณ 5 นาทีก่อนคุณจัดตารางเวลาคำสั่ง **uudemon.hour**

i. ตรวจสอบให้แน่ใจว่าการเปลี่ยนแปลงของคุณมีผลบังคับใช้แล้วโดยรัน คำสั่งต่อไปนี้:

```
crontab -l uucp
```

j. ตั้งค่าไฟล์ข้อมูล BNU ต่อไปนี้: `Systems`, `Permissions`, `Devices`, `Dialers` และ `Sysfiles` คุณอาจใช้คำสั่ง `/usr/sbin/uucp/uucpadm` เพื่อตั้งค่าไฟล์ในครั้งแรก จากนั้น แก้ไขให้เหมาะสมกับความต้องการของคุณ ใช้ไฟล์ `Sysfiles` เพื่อระบุไฟล์อื่นที่ ไม่ใช่ `/etc/uucp/Systems`, `/etc/uucp/Devices` และ `/etc/uucp/Dialers` สำหรับคอนฟิกูเรชัน BNU สำหรับข้อมูลเพิ่มเติม โปรดดู `Sysfiles`

3. ถ้าคุณตัดสินใจที่จะใช้ตัวย่อโค้ดการหมุนสำหรับหมายเลขโทรศัพท์ในไฟล์ ระบบ ให้ตั้งค่ารายการ `Dialcodes` สำหรับแต่ละตัวย่อ สำหรับรายละเอียด โปรดดู รูปแบบไฟล์โค้ดการหมุนสำหรับ BNU

   ถ้า คุณใช้ TCP/IP สำหรับการเชื่อมต่อ BNU ของคุณ ให้ใช้คำสั่ง **netstat** เพื่อดูว่า **uucpd** daemon กำลังทำงานอยู่หรือไม่ โดยป้อนคำสั่งต่อไปนี้:

   ```
   netstat -a
   ```

   uucpd daemon ถูกสตาร์ทโดย **inetd** daemon ถ้า **uucpd** daemon ไม่ได้รัน ให้กำหนดคอนฟิก **inetd** daemon อีกครั้งเพื่อเริ่มต้น **uucpd** daemon สำหรับข้อมูลเพิ่มเติม โปรดดู "การกำหนดคอนฟิก inetd daemon" ในหน้า 377

4. ใช้รายการอุปกรณ์ที่คุณรวบรวมไว้ ก่อนคุณ เริ่มต้นโพรซีเดอร์นี้ เพื่อเปลี่ยนไฟล์ อุปกรณ์ บนระบบของคุณ จัดทำรายการสำหรับแต่ละ modem และแต่ละการเชื่อมต่อโดยตรง ถ้าคุณใช้ TCP/IP ให้ยกเลิกการแสดงข้อคิดเห็นเกี่ยวกับรายการ TCP/IP ในไฟล์ อุปกรณ์ คุณสามารถกำหนดคอนฟิกไฟล์ `/etc/uucp/Sysfiles` เพื่อระบุไฟล์อื่นที่จะใช้สำหรับคอนฟิกูเรชันของอุปกรณ์ สำหรับ รายละเอียดเกี่ยวกับไฟล์อุปกรณ์ โปรดดู รูปแบบ ไฟล์อุปกรณ์สำหรับ BNU

   นอกจากนี้ ถ้าคุณใช้ TCP/IP ให้ตรวจสอบว่าไฟล์ `/etc/services` มี บรรทัดต่อไปนี้:

   ```
   uucp        540/tcp          uucpd
   ```

   ถ้า ไม่มี ให้เพิ่มบรรทัดนี้ลงในไฟล์

5. ใช้ข้อมูลเกี่ยวกับแต่ละระบบรีโมตที่คุณรวบรวมไว้ ก่อน คุณเริ่มต้นโพรซีเดอร์นี้ เพื่อเปลี่ยนไฟล์ ระบบ บนระบบของคุณ ใช้ตัวอย่างที่แสดงข้อคิดเห็นในไฟล์ ระบบ เป็น แนวทางเมื่อคุณระบุคอนฟิกูเรชันของคุณ ถ้าคุณใช้ TCP/IP ตรวจสอบให้แน่ใจว่าตารางชื่อโฮสต์ในไฟล์ `/etc/hosts` มีชื่อของคอมพิวเตอร์แบบรีโมตซึ่งคุณต้องการเชื่อมต่อ คุณสามารถกำหนดคอนฟิกไฟล์ `/etc/uucp/Sysfiles` เพื่อระบุไฟล์อื่นที่จะใช้สำหรับคอนฟิกูเรชันของระบบ

6. ใช้ข้อมูลเกี่ยวกับอุปกรณ์และโมเด็มที่คุณรวบรวมไว้ ก่อน คุณเริ่มต้นโพรซีเดอร์นี้ เพื่อให้แน่ใจว่าไฟล์ Dialers บนระบบของคุณมีรายการสำหรับแต่ละโมเด็ม ถ้าคุณใช้ TCP/IP และการเชื่อมต่อโดยตรง ตรวจสอบให้แน่ใจว่ารายการ TCP/IP และรายการโดยตรง มีอยู่ในไฟล์ คุณสามารถกำหนดคอนฟิกไฟล์ /etc/uucp/Sysfiles เพื่อระบุไฟล์อื่นที่จะใช้สำหรับคอนฟิกูเรชันของ dialers
7. ตัดสินใจเกี่ยวกับระดับการเข้าถึงระบบของคุณที่คุณต้องการให้ กับแต่ละระบบรีโมตซึ่งคุณเรียก และให้กับแต่ละระบบรีโมตที่ เรียกคุณ ตั้ง entry ที่เหมาะสมสำหรับแต่ละระบบและแต่ละชื่อล็อกอินในไฟล์ Permissions
8. ใช้คำสั่ง **uucheck** เพื่อตรวจสอบว่า ไดเร็กทอรี โปรแกรม และไฟล์สนับสนุนมีการตั้งค่าอย่างถูกต้อง:

   ```
   /usr/sbin/uucp/uucheck -v
   ```

   คำสั่ง **uucheck** จะตรวจสอบว่า ไดเร็กทอรี โปรแกรม และไฟล์สนับสนุนถูกตั้งค่าอย่างถูกต้อง และ entry ของไฟล์ Permissions เชื่อถือได้ ถ้าคำสั่ง **uucheck** รายงานข้อผิดพลาด ให้แก้ไขข้อผิดพลาด
9. ทางเลือก: ตั้งค่าการมอนิเตอร์อัตโนมัติของการดำเนินการ BNU และการโพลอัตโนมัติของระบบรีโมต สำหรับข้อมูลเพิ่มเติม โปรดดู “การตั้งค่าการมอนิเตอร์ BNU โดยอัตโนัติ”และ “การตั้งค่า BNU เพื่อโพลระบบรีโมต”

การตั้งค่าการมอนิเตอร์ BNU โดยอัตโนัติ

BNU ใช้ **cron** daemon เพื่อสตาร์ท BNU daemons และเพื่อมอนิเตอร์กิจกรรมของ BNU

ข้อกำหนดเบื้องต้น

- ทำขั้นตอนที่ลิสต์ใน “การตั้งค่า BNU” ในหน้า 465
- คุณต้องมีสิทธิผู้ใช้ root เพื่อแก้ไขไฟล์ /var/spool/cron/crontabs/uucp

cron daemon จะอ่านไฟล์ /var/spool/cron/crontabs/uucp สำหรับวิธีการเกี่ยวกับเมื่อใดที่จะเริ่มโพรซีเดอร์ BNU

เมื่อต้องการตั้งค่า การมอนิเตอร์อัตโนมัติของ BNU ให้ทำขั้นตอนต่อไปนี้:

1. ล็อกอินเป็นผู้ใช้ด้วยสิทธิของ root
2. ใช้เท็กซ์เอดิเตอร์ ASCII เพื่อแก้ไขไฟล์ /var/spool/cron/crontabs/uucp
3. ยกเลิกหมายเหตุบรรทัดสำหรับโพรซีเดอร์การบำรุงรักษา BNU uudemon.admin และ uudemon.cleanup คุณสามารถเปลี่ยน เวลาที่โพรซีเดอร์เหล่านี้รัน ถ้าระบบของคุณต้องการการบำรุงรักษา ในช่วงเวลาที่บ่อยมากขึ้นหรือน้อยลง อย่างไรก็ตาม มันจะดีที่สุดที่จะรันคำสั่ง uudemon.admin อย่างน้อยวันละครั้ง และคำสั่ง uudemon.cleanup อย่างน้อยอาทิตย์ละครั้ง
4. ใช้ไฟล์ crontabs/uucp เพื่อจัดตารางเวลา คำสั่งการบำรุงรักษา BNU อื่น เช่น คำสั่ง **uulog, uuclean** หรือ **uucleanup** นอกจากนี้ คุณสามารถใช้ไฟล์ crontabs/uucp เพื่อบอกให้ **cron** daemon สตาร์ท **uucico, uuxqt** หรือ **uusched** daemons ที่เวลาที่กำหนด

การตั้งค่า BNU เพื่อโพลระบบรีโมต

เพื่อเปิดใช้งาน BNU เพื่อโพลระบบรีโมตสำหรับงาน ลิสต์ระบบในไฟล์ /etc/uucp/Poll

ข้อกำหนดเบื้องต้น

- ทำขั้นตอนที่แสดงรายการใน “การตั้งค่า BNU” ในหน้า 465
- คุณต้องมีสิทธิ root เพื่อแก้ไขไฟล์ /var/spool/cron/crontabs/uucp และไฟล์ /etc/uucp/Poll

นอกเหนือจากลิสต์ระบบในไฟล์ /etc/uucp/Poll รันคำสั่ง **uudemon.hour** และ **uudemon.poll** เป็นครั้งคราว

เมื่อต้องการตั้งค่าการโพล BNU ของระบบรีโมต ให้ทำ ขั้นตอนต่อไปนี้:

1. ตัดสินใจว่าระบบรีโมตใดที่จะถูกโพลโดยอัตโนมัติ ตัดสินใจว่า คุณต้องการโพลแต่ละระบบบ่อยเพียงใด ระบุเวลาสำหรับแต่ละ ระบบด้วยไฟล์ `โพล` ซึ่งอาจน้อยเพียง วันละหนึ่งครั้ง หรือบ่อยตามที่คุณต้องการ
2. ล็อกอินเป็นผู้ใช้ด้วยสิทธิของ root
3. ใช้เท็กซ์เอดิเตอร์ ASCII หรือคำสั่ง **uucpadm** แก้ไขไฟล์ `Poll` เพิ่มรายการสำหรับแต่ละระบบ ซึ่งระบบของคุณมีการตั้งค่าให้โพล

   หมายเหตุ: ระบบที่ แสดงรายการในไฟล์ `Poll` ยังต้องแสดงรายการ ในไฟล์ `/etc/uucp/Systems` ด้วย
4. การใช้เท็กซ์เอดิเตอร์ ASCII แก้ไขไฟล์ `/var/spool/cron/crontabs/uucp` ลบอักขระหมายเหตุ (`#`) จากบรรทัดที่รันคำสั่ง **uudemon.hour** และ **uudemon.poll** คุณสามารถเปลี่ยนเวลาที่คำสั่งเหล่านี้รัน อย่างไรก็ตาม ต้องแน่ใจว่าจัดตารางเวลาคำสั่ง **uudemon.poll** ประมาณ 5 นาทีก่อนคุณจัดตารางเวลาคำสั่ง **uudemon.hour**

ตอนนี้ BNU มีการตั้งค่าให้โพลระบบที่ แสดงรายการในไฟล์ `โพล` โดยอัตโนมัติตามเวลาที่ คุณระบุ

ไฟล์ /etc/uucp/Systems

ระบบรีโมตจะถูกลิสต์ในไฟล์ `/etc/uucp/Systems`

ไฟล์ `/etc/uucp/Systems` เป็นไฟล์ `Systems` แบบดีฟอลต์ ผู้ดูแลระบบสามารถระบุไฟล์เพิ่มเติมในไฟล์ **/etc/uucp/Sysfiles**

แต่ละรายการในไฟล์ `ระบบ` มี ไอเท็มต่อไปนี้:

- ชื่อของระบบรีโมต
- เวลาที่ผู้ใช้สามารถเชื่อมต่อกับระบบรีโมต
- ชนิดของลิงก์ (สายตรงหรือโมเด็ม)
- ความเร็วของการส่งข้อมูลบนลิงก์
- ข้อมูลที่ต้องใช้เพื่อล็อกอินเข้าสู่ระบบรีโมต

แต่ละ entry ในไฟล์ `Systems` จะแทนระบบรีโมตหนึ่งระบบ เพื่อเริ่มการสื่อสาร ระบบรีโมตต้องถูกลิสต์ในไฟล์ `Systems` แบบโลคัล ไฟล์ `Systems` ต้องถูกแสดงบนทุกระบบที่ใช้ BNU โดยปกติ เฉพาะผู้ใช้ root สามารถอ่านไฟล์ `Systems` อย่างไรก็ตาม ผู้ใช้ใดๆ สามารถแสดงรายการชื่อของระบบ BNU แบบรีโมตโดยใช้ คำสั่ง **uuname**

การแก้ไขไฟล์อุปกรณ์สำหรับการเชื่อมต่อโดยตรง

เพื่อแก้ไขไฟล์ `อุปกรณ์` สำหรับการเชื่อมต่อ โดยตรง คุณต้องมีสิทธิ root เพื่อแก้ไขไฟล์ `/etc/uucp/Devices` หรือไฟล์อื่นที่ระบุในไฟล์ `/etc/uucp/Sysfiles` เป็นไฟล์ `อุปกรณ์`

เมื่อต้องการตั้งค่าการเชื่อมต่อโดยตรงที่ระบุพอร์ตและ ระบบรีโมต ให้จัดทำรายการดังนี้:

1. ป้อนชื่อของระบบรีโมตซึ่งคุณต้องการเชื่อมต่อ คอมพิวเตอร์แบบโลคัลผ่านสายโดยตรงในฟิลด์ **ชนิด** ในบรรทัดที่สองของรายการ
2. ป้อนชื่ออุปกรณ์ที่เหมาะสมสำหรับการเชื่อมต่อโดยตรง ซึ่ง ใช้ที่ไซต์ของคุณในฟิลด์ **สาย** ในทั้งสอง บรรทัดของรายการ
3. ป้อนขีดกลาง (`-`) เป็นตัวยึดตำแหน่งในฟิลด์ **สาย2** ในทั้งสองบรรทัดของรายการ
4. ป้อนอัตราการส่งข้อมูลที่เหมาะสมสำหรับการเชื่อมต่อโดยตรง ซึ่งใช้ที่ไซต์ของคุณในฟิลด์ **ความเร็ว** ในทั้งสองบรรทัดของรายการ

5. ใส่ direct (ตัวพิมพ์เล็กทั้งหมด) ในฟิลด์ **Dialer-Token Pairs** ในทั้งสองบรรทัดของ entry

   ตัวอย่าง เช่น:

   ```
   type device - speed direct
   ```

เพิ่มรายการลงในไฟล์ อุปกรณ์ ต่อไป จนกว่าคุณแสดงรายการแต่ละอุปกรณ์ที่เชื่อมต่อระบบโลคัล กับระบบรีโมตโดยตรง

เมื่อต้องการตั้งค่าการเชื่อมต่อโดยตรงระหว่างสองระบบที่ ใช้การเชื่อมต่อซีเรียลอะซิงโครนัสแบบถาวร ให้จัดทำรายการแบบหนึ่งบรรทัด ดังนี้:

1. ป้อนชื่อของระบบรีโมตในฟิลด์ **ชนิด**
2. ป้อนชื่อของอุปกรณ์ tty ในฟิลด์ **สาย**
3. ป้อนขีดกลาง (-) เป็นตัวยึดตำแหน่งใน ฟิลด์ **สาย2**
4. ป้อนอัตราการส่งข้อมูลที่เหมาะสมสำหรับการเชื่อมต่อโดยตรง ซึ่งใช้ที่ไซต์ของคุณในฟิลด์ **คลาส**
5. ป้อน direct (ตัวพิมพ์เล็กทั้งหมด) ในฟิลด์ **Dialer-Token Pairs** ตัวอย่างเช่น:

   ```
   type device - speed direct
   ```

เพิ่มรายการลงในไฟล์ อุปกรณ์ ต่อไป จนกว่าคุณแสดงรายการแต่ละอุปกรณ์โดยตรงที่เชื่อมต่อระบบโลคัล กับระบบรีโมต

การแก้ไขไฟล์ อุปกรณ์ สำหรับการเชื่อมต่อแบบ autodialer

ปฏิบัติตามขั้นตอนเหล่านี้เมื่อคุณแก้ไขไฟล์ /etc/uucp/Devices

คุณต้องมีสิทธิ root เพื่อแก้ไขไฟล์ /etc/uucp/Devices หรือไฟล์อื่นที่ระบุในไฟล์ /etc/uucp/Sysfiles เป็นไฟล์ อุปกรณ์

ใน entry ของ telephone-connection ฟิลด์ **Type** ถูกระบุเป็น automatic calling unit (ACU) ป้อน ACU เป็นรายการฟิลด์ **ชนิด** ในการเชื่อมต่อแบบรีโมตทั้งหมดที่สร้างขึ้นบนสาย โทรศัพท์ เพื่อตั้งค่าไฟล์ Devices entry สำหรับการเชื่อมต่อแบบ autodialer สร้าง entry แบบบรรทัดเดียวสำหรับแต่ละโมเด็ม :

1. ในฟิลด์ **ชนิด** ป้อน ACU
2. ในฟิลด์ **สาย** ป้อนชื่ออุปกรณ์ ที่ต่อพ่วงกับโมเด็ม
3. ในฟิลด์ **สาย2** ป้อนขีดกลาง (-) เป็นตัวยึดตำแหน่ง ยกเว้นว่า autodialer เป็น 801 dialer มาตรฐาน ถ้า autodialer เป็น 801 dialer มาตรฐาน ใส่ 801
4. ในฟิลด์ **ความเร็ว** ป้อนอัตรา baud ที่เหมาะสมสำหรับโมเด็มและสายของคุณ หรือคลาสของโมเด็มของคุณ (ตัวอย่างเช่น D2400) ค่าของอัตรา baud สามารถเป็น 300, 1200, 2400 หรือสูงกว่า ขึ้นอยู่กับโมเด็ม

   หมายเหตุ: ถ้าโมเด็ม สามารถใช้ได้ที่อัตรา baud ซึ่งระบุมากกว่าหนึ่งอัตรา ให้ทำรายการแยกต่างหาก ในไฟล์ อุปกรณ์ สำหรับแต่ละอัตรา ถ้า โมเด็มสามารถใช้ได้ที่อัตรา baud ใดๆ ให้ป้อนคำว่า Any ใน ฟิลด์ **ความเร็ว**
5. ป้อนชื่อโมเด็มเป็นรายการของฟิลด์ **Dialer** ในฟิลด์ **Dialer-Token Pair** ถ้าคุณวางแผนจะรวมหมายเลขโทรศัพท์ทั้งหมดในไฟล์ /etc/uucp/Systems หรือ ไฟล์ ระบบ อื่น ซึ่งระบุในไฟล์ /etc/uucp/Sysfiles ให้ปล่อยฟิลด์ **โทเค็น** ว่างไว้ พื้นที่ว่างเปล่าจะสั่งให้ โปรแกรม BNU ใช้โทเค็น \D ดีฟอลต์ ถ้าคุณ วางแผนจะใช้ตัวย่อโค้ดการหมุนที่ระบุ ในไฟล์ /etc/uucp/Dialcodes ให้ป้อนโทเค็น \T

   ตัวอย่าง เช่น:

   ```
   พิมพ์คู่ของ สาย - ความเร็วของ dialer - โทเค็น
   ```

เพิ่ม entry เข้ากับไฟล์ Devices จนกระทั่งคุณลิสต์แต่ละการเชื่อมต่อระหว่างระบบโลคัลและระบบรีโมตที่ใช้สายโทรศัพท์และโมเด็ม

การแก้ไขไฟล์อุปกรณ์สำหรับ TCP/IP

ปฏิบัติตามขั้นตอนเหล่านี้เมื่อคุณแก้ไขไฟล์ /etc/uucp/Devices

คุณต้องมีสิทธิ root เพื่อแก้ไขไฟล์ /etc/uucp/Devices หรือไฟล์อื่นที่ระบุในไฟล์ /etc/uucp/Sysfiles เป็นไฟล์ อุปกรณ์

ถ้าไซต์ของคุณใช้ TCP/IP เพื่อเชื่อมต่อระบบ รวมถึงรายการ TCP/IP ที่เกี่ยวข้องใน ไฟล์ อุปกรณ์ เมื่อต้องการตั้งค่าไฟล์สำหรับใช้ กับ TCP/IP ให้ป้อนบรรทัดต่อไปนี้ในไฟล์ อุปกรณ์:

```
TCP - - - TCP
```

ตัวอย่าง: คอนฟิกูเรชัน BNU สำหรับการเชื่อมต่อ TCP/IP

ตัวอย่างกลุ่มนี้กำหนดคอนฟิก BNU สำหรับการเชื่อมต่อ TCP/IP

ไฟล์ต่อไปนี้มีการตั้งค่าสำหรับการเชื่อมต่อ TCP/IP ระหว่าง ระบบ zeus และ hera โดยที่ zeus คือ ระบบโลคัลและ hera คือ ระบบรีโมต

ไฟล์ BNU สำหรับ entry ของการเชื่อมต่อ TCP/IP ในไฟล์ของระบบโลคัล:

ไฟล์ BNU เหล่านี้เป็นรายการบนระบบโลคัล zeus

- **ไฟล์ระบบ:** ไฟล์ ระบบ บน ระบบ zeus มีรายการต่อไปนี้เพื่อให้ zeus สามารถ ติดต่อกับระบบ hera:

  ```
  hera Any TCP,t - - in:--in: uzeus word: birthday
  ```

  ตัวอย่างนี้ ระบุว่าระบบ zeus สามารถเรียกระบบ hera ได้ ทุกเวลา โดยใช้โปรโตคอล **t** สำหรับการสื่อสารกับระบบ hera ระบบzeus ล็อกอินกับระบบ hera เป็น uzeus ด้วยรหัสผ่าน birthday

  หมายเหตุ: โปรโตคอล **t** สนับสนุน **TCP** ดังนั้น ใช้โปรโตคอล **t** เสมอ สำหรับการสื่อสาร BNU ผ่านการเชื่อมต่อ TCP/IP อย่างไรก็ตาม ไม่สามารถใช้โปรโตคอล **t** เมื่อฟิลด์ **ชนิด** เป็น ACU (automatic calling unit) หรือเมื่อใช้การเชื่อมต่อโมเด็ม BNU จะใช้ฟิลด์ **Type** และ **Class** ในไฟล์ Systems เพื่อหาอุปกรณ์ที่เหมาะสมสำหรับการเชื่อมต่อ BNU จะตรวจสอบไฟล์ อุปกรณ์ เพื่อหารายการชนิด TCP

- **ไฟล์อุปกรณ์:** ไฟล์ อุปกรณ์ ที่ ใช้โดย **uucico** daemon บนระบบ zeus มี รายการต่อไปนี้สำหรับการเชื่อมต่อ TCP/IP:

  ```
  TCP - - - TCP
  ```

  เนื่องจากชนิดของอุปกรณ์เป็น TCP จะไม่มี entry *Class*, *Line* หรือ *Line2 Dialer* ยังถูกระบุเป็น TCP BNU จะค้นหาในไฟล์ Dialers เพื่อหารายการ TCP

- **ไฟล์ Dialers:** ไฟล์ Dialers ที่ ใช้โดย **uucico** daemon บนระบบ zeus มี รายการ TCP/IP ดังนี้:

  ```
  TCP
  ```

  entry จะระบุว่าไม่ต้องการการตั้งค่า dialer

  หมายเหตุ: การตั้งค่า Dialer ไม่เคยถูกต้องการบนการเชื่อมต่อ TCP/IP

- **ไฟล์สิทธิ:** ไฟล์ สิทธิ บนระบบ zeus มีรายการต่อไปนี้ ซึ่งให้สิทธิกับระบบ hera ในการเข้าถึงระบบ zeus:

```
LOGNAME=uhera SENDFILES=yes REQUEST=yes \
MACHINE=zeus:hera VALIDATE=uhera \
READ=/var/spool/uucppublic:/home/hera \
WRITE=/var/spool/uucppublic:/home/hera COMMANDS=ALL
```

รายการ LOGNAME และ MACHINE ที่รวมเข้าด้วยกันให้สิทธิต่อไปนี้ กับระบบ hera เมื่อระบบ zeus และ ระบบ hera มีการเชื่อมต่อกัน:

- ระบบ hera สามารถร้องขอและส่งไฟล์โดยไม่สนใจว่าใช้เป็นผู้ทำการเรียก
- ระบบ hera สามารถอ่านและเขียนในไดเร็กทอรี พับลิกและในไดเร็กทอรี /home/hera บน ระบบ zeus
- ระบบ hera สามารถรันคำสั่งทั้งหมดบนระบบ zeus
- ระบบ hera ต้องล็อกอินเข้าสู่ระบบ zeus เป็น ผู้ใช้ uhera และระบบ hera ไม่สามารถ ใช้ ID ล็อกอินอื่นสำหรับธุรกรรม BNU

  หมายเหตุ: เนื่องจากการอนุญาตเหมือนกันโดยไม่สนใจว่าระบบใดเป็นผู้ทำการเรียก entry LOGNAME และ MACHINE ที่มาก่อนหน้าจะถูกรวม ถ้าสิทธิสำหรับระบบ hera และระบบ zeus ไม่เหมือนกัน รายการ LOGNAME และ MACHINE จะเป็นดังนี้:

  ```
  LOGNAME=uhera VALIDATE=hera SENDFILES=yes REQUEST=yes \
  READ=/var/spool/uucppublic:/home/hera \
  WRITE=/var/spool/uucppublic:/home/hera

  MACHINE=zeus:hera REQUEST=yes COMMANDS=ALL\
  READ=/var/spool/uucppublic:/home/hera \
  WRITE=/var/spool/uucppublic:/home/hera
  ```

ไฟล์ BNU สำหรับ entry ของการเชื่อมต่อ TCP/IP ในไฟล์ของระบบรีโมต:

ไฟล์เหล่านี้อยู่บนระบบรีโมต hera

- **ไฟล์ระบบ:** ไฟล์ *ระบบ* บนระบบ hera มี รายการต่อไปนี้เพื่ออนุญาตให้ hera เชื่อมต่อกับระบบ zeus:

  ```
  zeus Any TCP,t - - ogin:--ogin: uhera ord: lightning
  ```

  ตัวอย่าง นี้ระบุว่าระบบ hera สามารถเรียกระบบ zeus ได้ ทุกเวลา โดยใช้โปรโตคอล **t** สำหรับการสื่อสารกับ ระบบ zeus ระบบhera ล็อกอินกับระบบ zeus เป็นผู้ใช้ uhera ด้วยรหัสผ่าน lightning อีกครั้ง ต่อไป BNU จะตรวจสอบไฟล์ Devices สำหรับ entry ของชนิด TCP

  หมายเหตุ: โปรโตคอล **t** สนับสนุน **TCP** ดังนั้น ใช้โปรโตคอล **t** เสมอ สำหรับการสื่อสาร BNU ผ่านการเชื่อมต่อ TCP/IP อย่างไรก็ตาม โปรโตคอล **t** ไม่สามารถใช้เมื่อฟิลด์ *Type* เป็น ACU หรือเมื่อการเชื่อมต่อโดยใช้โมเด็มถูกใช้

- **ไฟล์อุปกรณ์:** ไฟล์ *อุปกรณ์* ที่ ใช้โดย **uucico** daemon บนระบบ hera มี รายการต่อไปนี้สำหรับการเชื่อมต่อ TCP/IP:

  ```
  TCP  -  -  -  TCP
  ```

  เนื่องจากชนิดของอุปกรณ์เป็น TCP จะไม่มี entry *Type*, *Line* หรือ *Line2 Dialer* ยังถูกระบุเป็น TCP BNU จะค้นหาในไฟล์ Dialers เพื่อหารายการ TCP

- **ไฟล์ Dialers:** ไฟล์ Dialers ที่ ใช้โดย **uucico** daemon บนระบบ hera มี รายการ TCP/IP ดังนี้:

  ```
  TCP
  ```

  entry จะระบุว่าไม่ต้องการการตั้งค่า dialer

  หมายเหตุ: การตั้งค่า Dialer ไม่เคยถูกต้องการบนการเชื่อมต่อ TCP/IP

- **ไฟล์สิทธิ:** ไฟล์ สิทธิ บนระบบ hera มีรายการต่อไปนี้ ซึ่ง ให้สิทธิกับระบบ zeus ในการเข้าถึงระบบ hera:

  ```
  LOGNAME=uzeus SENDFILES=yes REQUEST=yes \
  MACHINE=hera:zeus VALIDATE=zeus COMMANDS=rmail:who:uucp
  ```

  The combined LOGNAME and MACHINE entries provide the following permissions to system zeus, when system zeus and system hera are connected:

  - ระบบ zeus สามารถร้องขอและส่งไฟล์โดยไม่สนใจว่าใช้เป็นผู้ทำการเรียก
  - ระบบ zeus สามารอ่านและเขียนไปยังพับลิกไดเร็กทอรีเท่านั้น (ดีฟอลต์)
  - ระบบ zeus สามารถรันคำสั่ง **rmail, who,** และ **uucp** เท่านั้น
  - ระบบ zeus ต้องล็อกอินเข้าสู่ระบบ hera เป็น ผู้ใช้ uzeus และระบบ zeus ไม่สามารถ ใช้ ID ล็อกอินอื่นสำหรับธุรกรรม BNU

    หมายเหตุ: ถ้าสิทธิสำหรับระบบ hera และระบบ zeus ไม่เหมือนกัน รายการ LOGNAME และ MACHINE จะเป็นดังนี้:

    ```
    LOGNAME=uzeus VALIDATE=zeus SENDFILES=yes REQUEST=yes
    MACHINE=hera:zeus COMMANDS=rmail:who:uucp REQUEST=yes
    ```

ตัวอย่าง: คอนฟิกูเรชัน BNU สำหรับการเชื่อมต่อผ่านโทรศัพท์

มีการตั้งค่าไฟล์ตัวอย่างเพื่อเชื่อมต่อระบบ venus และ merlin ผ่าน สายโทรศัพท์โดยใช้โมเด็ม

ระบบ venus เป็นระบบโลคัล และ ระบบ merlin เป็นระบบรีโมต

บนทั้งสองระบบ อุปกรณ์ tty1 มีการเชื่อมต่อกับ โมเด็ม Hayes ที่ความเร็ว 1200 baud ID ล็อกอินที่ใช้สำหรับระบบ venus เพื่อล็อก อินเข้าสู่ระบบ merlin คือ uvenus และ รหัสผ่านที่เชื่อมโยงคือ mirror ID ล็อกอินสำหรับ ระบบ merlin เพื่อล็อกอินเข้าสู่ระบบ venus คือ umerlin และรหัสผ่านที่เชื่อมโยงคือ oaktree หมายเลข โทรศัพท์สำหรับโมเด็มที่ต่อพ่วงกับ venus คือ 9=3251436 และ หมายเลขของโมเด็ม merlin คือ 9=4458784 คอมพิวเตอร์ทั้งสองเครื่องรวมหมายเลขโทรศัพท์บางส่วนไว้ในไฟล์ ระบบ และรวมโค้ดการหมุนไว้ในไฟล์ Dialcodes

ไฟล์ตัวอย่างต่อไปนี้มีการตั้งค่าเพื่อเชื่อมต่อระบบ venus และ merlin:

- **ไฟล์ระบบ:** ไฟล์ ระบบ บน ระบบ venus มีรายการต่อไปนี้สำหรับระบบ merlin ซึ่งรวมถึงหมายเลขโทรศัพท์และคำเติมหน้าการหมุน:

  ```
  merlin Any ACU 1200 local8784 "" in:--in: uvenus word: mirror
  ```

  ระบบ venus สามารถ เรียกระบบ merlin ได้ทุกเวลา โดยใช้อุปกรณ์ ACU ที่ความเร็ว 1200 baud และล็อกอินเป็น uvenus โดยใช้ รหัสผ่าน mirror หมายเลขโทรศัพท์ถูกขยายโดยขึ้นอยู่กับโค้ด local ในไฟล์ Dialcodes และอุปกรณ์ที่จะใช้ถูกกำหนดโดยขึ้นอยู่กับ entry *Type* และ *Class* BNU ตรวจสอบไฟล์ อุปกรณ์ เพื่อหาอุปกรณ์ ชนิด ACU และคลาส 1200
- **ไฟล์ Dialcodes :** ไฟล์ Dialcodes บนระบบ venus ประกอบด้วยส่วนนำหน้า dial-code สำหรับใช้กับหมายเลขในไฟล์ Systems :

  ```
  local 9=445
  ```

  โค้ดนี้ หมายเลขโทรศัพท์สำหรับระบบmerlin ในไฟล์ Systems จะถูกขยายเป็น 9=4458784
- **ไฟล์อุปกรณ์:** ไฟล์ อุปกรณ์ บน ระบบ venus มีรายการต่อไปนี้สำหรับ การเชื่อมต่อกับระบบ merlin:

  ```
  ACU tty1  -  1200  hayes \T
  ```

พอร์ตที่จะใช้คือ tty1 และ entry *Dialer* ในฟิลด์ *Dialer-Token Pairs* คือ hayes รายการ *โทเค็น* \T บ่งชี้ ว่าหมายเลขโทรศัพท์จะถูกขยายโดยใช้โค้ดจากไฟล์ Dialcodes BNU จะตรวจสอบไฟล์ Dialers เพื่อดูชนิดของ hayes dialer

- **ไฟล์ Dialers:** ไฟล์ Dialers ที่ใช้โดย **uucico** daemon บนระบบ venus มี รายการต่อไปนี้สำหรับ hayes modem:

  ```
  hayes =,-, "" \dAT\r\c OK \pATDT\T\r\c CONNECT
  ```

  หมายเหตุ: อักขระที่ต้องถูกส่งจะถูกกำหนดในรูปแบบของไฟล์ Dialers

- **ไฟล์สิทธิ:** ไฟล์ สิทธิ บนระบบ venus มีรายการต่อไปนี้ ซึ่ง ระบุวิธีซึ่งระบบ merlin สามารถทำธุรกรรม **uucico** และ **uuxqt** กับระบบ venus:

  ```
  LOGNAME=umerlin REQUEST=yes SENDFILES=yes \
  READ=/var/spool/uucppublic:/home/merlin \
  WRITE=/var/spool/uucppublic:/home/merlin
  MACHINE=venus:merlin VALIDATE=umerlin REQUEST=yes SENDFILES=yes    \
  COMMANDS=ALL \
  READ=/var/spool/uucppublic:/home/merlin \
  WRITE=/var/spool/uucppublic:/home/merlin
  ```

  ระบบ merlin จะล็อกอินกับระบบ venus เป็น umerlin ซึ่งเป็นล็อกอินที่เป็นหนึ่งเดียวสำหรับระบบ merlin ระบบ merlin สามารถ ร้องขอและส่งไฟล์โดยไม่คำนึงว่าใครเริ่มต้นการเรียก นอกจากนี้ ระบบ merlin ยังสามารถอ่านและเขียนในไดเร็กทอรี /var/spool/uucppublic และในไดเร็กทอรี /home/merlin บนระบบ venus ระบบ merlin สามารถออกใช้คำสั่งทั้งหมดในชุดคำสั่งดีฟอลต์ บนระบบ venus

ไฟล์ BNU ที่มีรายการการเชื่อมต่อผ่านโทรศัพท์บนระบบโลคัล:

ไฟล์เหล่านี้มีรายการการเชื่อมต่อผ่านโทรศัพท์บน ระบบโลคัล venus

- **ไฟล์ระบบ:** ไฟล์ ระบบ บน ระบบ venus มีรายการต่อไปนี้สำหรับระบบ merlin ซึ่งรวมถึงหมายเลขโทรศัพท์และคำเติมหน้าการหมุน:

  ```
  merlin Any ACU 1200 local8784 "" in:--in: uvenus word: mirror
  ```

  ระบบ venus สามารถ เรียกระบบ merlin ได้ทุกเวลา โดยใช้อุปกรณ์ ACU ที่ความเร็ว 1200 baud และล็อกอินเป็นผู้ใช้ uvenus โดยใช้ รหัสผ่าน mirror หมายเลขโทรศัพท์ถูกขยายโดยขึ้นอยู่กับโค้ด local ในไฟล์ Dialcodes และอุปกรณ์ที่จะใช้ถูกกำหนดโดยขึ้นอยู่กับ entry *Type* และ *Class* BNU ตรวจสอบไฟล์ อุปกรณ์ เพื่อหาอุปกรณ์ ชนิด ACU และคลาส 1200

- **ไฟล์ Dialcodes :** ไฟล์ Dialcodes บนระบบ venus ประกอบด้วยส่วนนำหน้า dial-code สำหรับใช้กับหมายเลขในไฟล์ Systems :

  ```
  local 9=445
  ```

  โค้ดนี้ หมายเลขโทรศัพท์สำหรับระบบmerlin ในไฟล์ Systems จะถูกขยายเป็น 9=4458784

- **ไฟล์อุปกรณ์:** ไฟล์ อุปกรณ์ บน ระบบ venus มีรายการต่อไปนี้สำหรับ การเชื่อมต่อกับระบบ merlin:

  ```
  ACU  tty1  -  1200  hayes \T
  ```

  พอร์ตที่จะใช้คือ tty1 และ entry *Dialer* ในฟิลด์ **Dialer-Token Pairs** คือ hayes รายการ *โทเค็น* \T บ่งชี้ ว่าหมายเลขโทรศัพท์จะถูกขยายโดยใช้โค้ดจากไฟล์ Dialcodes BNU ตรวจสอบไฟล์ Dialers เพื่อหารายการชนิด dialer hayes

- **ไฟล์ Dialers:** ไฟล์ Dialers ที่ใช้โดย **uucico** daemon บนระบบ venus มี รายการต่อไปนี้สำหรับ hayes modem:

  ```
  hayes  =,-,  ""  \dAT\r\c OK \pATDT\T\r\c CONNECT
  ```

  หมายเหตุ: อักขระที่ต้องถูกส่งจะถูกกำหนดในรูปแบบของไฟล์ Dialers

- **ไฟล์สิทธิ:** ไฟล์ สิทธิ บนระบบ venus มีรายการต่อไปนี้ ซึ่ง ระบุวิธีซึ่งระบบ merlin สามารถทำธุรกรรม **uucico** และ **uuxqt** กับระบบ venus:

  ```
  LOGNAME=umerlin REQUEST=yes SENDFILES=yes \
  READ=/var/spool/uucppublic:/home/merlin \
  WRITE=/var/spool/uucppublic:/home/merlin
  MACHINE=venus:merlin VALIDATE=umerlin REQUEST=yes SENDFILES=yes    \
  COMMANDS=ALL \
  READ=/var/spool/uucppublic:/home/merlin \
  WRITE=/var/spool/uucppublic:/home/merlin
  ```

  ระบบ merlin จะล็อกอินกับระบบ venus เป็น umerlin ซึ่งเป็นล็อกอินที่เป็นหนึ่งเดียวสำหรับระบบ merlin ระบบ merlin สามารถ ร้องขอและส่งไฟล์โดยไม่คำนึงว่าใครเริ่มต้นการเรียก นอกจากนี้ ระบบ merlin ยังสามารถอ่านและเขียนในไดเร็กทอรี /var/spool/uucppublic และในไดเร็กทอรี /home/merlin บนระบบ venus ระบบ merlin สามารถออกใช้คำสั่งทั้งหมดในชุดคำสั่งดีฟอลต์ บนระบบ venus

ไฟล์ BNU พร้อมกับ entry ของการเชื่อมต่อโดยใช้โทรศัพท์บนระบบรีโมต:

ไฟล์เหล่านี้มีรายการการเชื่อมต่อผ่านโทรศัพท์บน ระบบรีโมต merlin

- **ไฟล์ระบบ:** ไฟล์ ระบบ บน ระบบ merlin มีรายการต่อไปนี้สำหรับระบบ venus ซึ่งรวมถึงหมายเลขโทรศัพท์และคำเติมหน้าการหมุน:

  ```
  venus Any ACU 1200 intown4362 "" in:--in: umerlin word: oaktree
  ```

  ระบบ merlin สามารถ เรียกระบบ venus ได้ทุกเวลา โดยใช้อุปกรณ์ ACU ที่ความเร็ว 1200 baud และล็อกอินเป็นผู้ใช้ umerlin โดยใช้ รหัสผ่าน oaktree หมายเลขโทรศัพท์ถูกขยายโดยขึ้นอยู่กับโค้ด intown ในไฟล์ Dialcodes และอุปกรณ์ที่จะใช้ถูกกำหนดโดยขึ้นอยู่กับ entry *Type* และ *Class* BNU ตรวจสอบไฟล์ อุปกรณ์ เพื่อหาอุปกรณ์ ชนิด ACU และคลาส 1200

- **ไฟล์ Dialcodes :** ไฟล์ Dialcodes บนระบบ merlin ประกอบด้วยส่วนนำหน้า dial-code สำหรับใช้กับหมายเลขในไฟล์ Systems :

  ```
  intown 9=325
  ```

  ดังนั้น หมายเลขโทรศัพท์ที่ถูกขยายเพื่อเข้าถึงระบบ venus คือ 9=3254362

- **ไฟล์อุปกรณ์:** ไฟล์ อุปกรณ์ บน ระบบ merlin มีรายการต่อไปนี้สำหรับ การเชื่อมต่อกับระบบ venus:

  ```
  ACU  tty1  -  1200  hayes \T
  ```

  ACU ถูกเชื่อมกับพอร์ต tty1 และ dialer คือhayes หมายเลขโทรศัพท์จะถูกขยายด้วยข้อมูลจากไฟล์ Dialcodes BNU ตรวจสอบไฟล์ Dialers เพื่อหารายการของ โมเด็ม hayes

- **ไฟล์ Dialers:** ไฟล์ Dialers ที่ ใช้โดย **uucico** daemon บนระบบ merlin มี รายการต่อไปนี้สำหรับโมเด็ม:

  ```
  hayes  =,-,  ""  \dAT\r\c OK \pATDT\T\r\c CONNECT
  ```

- **ไฟล์สิทธิ:** ไฟล์ สิทธิ บนระบบ merlin มีรายการต่อไปนี้ ซึ่งให้สิทธิกับระบบ venus ในการเข้าถึงระบบ merlin:

  ```
  LOGNAME=uvenus SENDFILES=call REQUEST=no \
  WRITE=/var/spool/uucppublic:/home/venus \
  READ=/var/spool/uucppublic:/home/venus
  MACHINE=merlin:venus VALIDATE=uvenus  \
  READ=/ WRITE=/ COMMANDS=ALL REQUEST=yes \
  NOREAD=/etc/uucp:/usr/etc/secure \
  NOWRITE=/etc/uucp:/usr/etc/secure
  ```

ตัวอย่าง: คอนฟิกูเรชัน BNU สำหรับการเชื่อมต่อโดยตรง

ไฟล์ตัวอย่างต่อไปนี้มีการตั้งค่าสำหรับการเชื่อมต่อโดยตรง ระหว่างระบบ zeus และ hera โดยที่ zeus คือ ระบบโลคัลและ hera คือระบบรีโมต

อุปกรณ์โดยตรงบนระบบ zeus คือ tty5 บน ระบบ hera อุปกรณ์โดยตรงคือ tty1 ความเร็วของการเชื่อมต่อคือ 1200 bps ล็อกอิน ID สำหรับระบบ zeus บนระบบ hera คือ uzeus และรหัสผ่านที่เกี่ยวข้องคือ thunder ล็อกอิน ID สำหรับ hera บนระบบ zeus คือ uhera และรหัสผ่านที่เกี่ยวข้องคือ portent

ไฟล์ BNU ที่มีการเชื่อมต่อโดยตรงในไฟล์ของระบบโลคัล:

ไฟล์เหล่านี้มีรายการการเชื่อมต่อผ่านโทรศัพท์บน ระบบโลคัล zeus

- **ไฟล์ระบบ:** ไฟล์ ระบบ บน ระบบ zeus มีรายการต่อไปนี้สำหรับ ระบบรีโมต hera:

  ```
  hera Any hera 1200 - "" \r\d\r\d\r in:--in: uzeus word:  thunder
  ```

  รายการนี้ ระบุว่าระบบ hera สามารถล็อกอินเข้าสู่ระบบ zeus ได้ตลอดเวลาโดยใช้การเชื่อมต่อโดยตรง ซึ่งระบุในไฟล์ อุปกรณ์ เพื่อหา entry ในไฟล์ Devices BNU จะใช้ฟิลด์ที่สามและสี่ของ entry Systems ดังนั้น BNU จะหา entry ในไฟล์ Devices ที่มี *Type* เป็น hera และ *Class* เป็น 1200 ระบบzeus ล็อกอินกับระบบ hera เป็นผู้ใช้ uzeus ด้วยรหัสผ่าน thunder

- **ไฟล์อุปกรณ์:** ไฟล์ อุปกรณ์ บน ระบบ zeus มีรายการต่อไปนี้เพื่อ เชื่อมต่อกับระบบรีโมต hera:

  ```
  hera    tty5  -  1200  direct
  ```

  รายการนี้ ระบุว่าระบบ zeus จะใช้อุปกรณ์ tty5 ที่ความเร็ว 1200 bps เพื่อสื่อสารกับระบบ hera โปรดสังเกตว่า *Dialer* ในฟิลด์ **Dialer-Token Pairs** คือ direct เมื่อคุณเชื่อมต่อ กับระบบ hera BNU จะตรวจสอบไฟล์ Dialers เพื่อหารายการโดยตรง

- **ไฟล์ Dialers:** ไฟล์ Dialers บน ระบบ zeus มีรายการต่อไปนี้สำหรับการเชื่อมต่อ โดยตรง:

  ```
  โดยตรง
  ```

  รายการนี้ระบุ ว่าไม่ต้องใช้ handshaking สำหรับการเชื่อมต่อโดยตรง

- **ไฟล์สิทธิ:** ไฟล์ สิทธิ บนระบบโลคัล zeus มีรายการต่อไปนี้ ซึ่งระบุวิธีที่ระบบรีโมต hera สามารถ ทำธุรกรรม **uucico** และ **uuxqt** กับระบบ zeus:

  ```
  LOGNAME=uhera MACHINE=hera VALIDATE=uhera REQUEST=yes \
  SENDFILES=yes MACHINE=zeus READ=/ WRITE=/ COMMANDS=ALL
  ```

  entry นี้จะระบุว่าระบบ hera ล็อกอินเป็น uhera เนื่องจากอ็อพชัน VALIDATE=uhera ถูกรวมด้วย ระบบ hera จะไม่สามารถล็อกอินยังระบบ zeus โดยใช้ล็อกอิน ID อื่น หรือระบบรีโมตอื่นไม่สามารถใช้ uhera ID ระบบ hera สามารถอ่านและเขียนไปยังไดเร็กทอรีใดๆบนระบบ zeus และสามารถส่งและร้องขอไฟล์โดยไม่สนใจว่าใครเป็นผู้เริ่มการติดต่อ ระบบ hera ยังสามารถใช้คำสั่งบนระบบ zeus

  หมายเหตุ: เนื่องจาก สิทธิที่ได้รับมอบเหมือนกันโดยไม่คำนึงว่าระบบใด เริ่มต้นการเชื่อมต่อ รายการ LOGNAME และ MACHINE จึงถูกรวมเข้าด้วยกัน ถ้าสิทธิสำหรับระบบ hera และระบบ zeus ไม่เหมือนกัน รายการ LOGNAME และ MACHINE จะเป็น ดังนี้:

  ```
  LOGNAME=uhera REQUEST=yes SENDFILES=yes READ=/ WRITE=/
  MACHINE=zeus:hera VALIDATE=uhera READ=/ WRITE=/ REQUEST=yes \
  COMMANDS=ALL
  ```

ข้อควรสนใจ: การให้การอนุญาตในตัวอย่างก่อนหน้านี้ เหมือนกับการให้ล็อกอิน ID กับผู้ใช้ใดๆบนระบบรีโมต บนระบบโลคัล สิทธิ ที่เป็นอิสระดังกล่าวอาจเป็นอันตรายต่อความปลอดภัย และมีการมอบให้กับ ระบบรีโมตที่เชื่อถือได้สูงที่ไซต์เดียวกันเท่านั้น

ไฟล์ BNU ที่มีการเชื่อมต่อโดยตรงในไฟล์ของระบบรีโมต:

ไฟล์เหล่านี้มีรายการการเชื่อมต่อผ่านโทรศัพท์บน ระบบรีโมต hera

- **ไฟล์ระบบ:** ไฟล์ ระบบ บน ระบบ hera มีรายการต่อไปนี้สำหรับระบบ zeus:

  ```
  zeus Any zeus 1200 - "" \r\d\r\d\r in:--in: uhera word: portent
  ```

  รายการนี้ ระบุว่าระบบ hera สามารถล็อกอินเข้าสู่ระบบ zeus ได้ ตลอดเวลาโดยใช้การเชื่อมต่อโดยตรง ซึ่งระบุในไฟล์ อุปกรณ์ เพื่อหา entry ในไฟล์ Devices BNU จะใช้ฟิลด์ที่สามและสี่ของ entry Systems ดังนั้น BNU จะหารายการในไฟล์ อุปกรณ์ ที่มีฟิลด์ **ชนิด** เป็นค่า zeus และ ฟิลด์ **คลาส** เป็นค่า 1200 ระบบhera ล็อกอินกับระบบ zeus เป็นผู้ใช้ uhera ด้วยรหัสผ่าน portent
- **ไฟล์อุปกรณ์:** ไฟล์ อุปกรณ์ บน ระบบ hera มีรายการต่อไปนี้สำหรับ การสื่อสารกับระบบ zeus:

  ```
  zeus    tty1  -  1200  direct
  ```

  รายการนี้ ระบุว่าระบบ hera จะใช้อุปกรณ์ tty1 ที่ความเร็ว 1200 bps เพื่อสื่อสารกับระบบ zeus เนื่องจาก *Dialer* มีการระบุเป็น direct BNU จะตรวจสอบไฟล์ Dialers เพื่อหารายการ direct
- **ไฟล์ Dialers:** ไฟล์ Dialers บน ระบบ hera มีรายการต่อไปนี้สำหรับการเชื่อมต่อ โดยตรง:

  ```
  direct
  ```

  รายการนี้ระบุว่า ไม่ต้องการคอนฟิกูเรชัน dialer บนการเชื่อมต่อโดยตรง
- **ไฟล์สิทธิ:** ไฟล์ สิทธิ บนระบบ hera มีรายการต่อไปนี้ ซึ่ง ระบุวิธีซึ่งระบบ zeus สามารถทำธุรกรรม **uucico** และ **uuxqt** กับระบบ hera:

  ```
  LOGNAME=uzeus REQUEST=yes SENDFILES=yes READ=/ WRITE=/
  MACHINE=hera:zeus VALIDATE=uzeus REQUEST=yes COMMANDS=ALL READ=/\
  WRITE=/
  ```

  entry เหล่านี้ระบุว่าระบบ zeus ล็อกอินกับระบบ hera เป็น uzeus เนื่องจาก มีพารามิเตอร์ VALIDATE=uzeus ระบบ zeus จึงไม่สามารถ ล็อกอินเข้าสู่ระบบ hera ด้วย ID ล็อกอินอื่นใดๆ และ ระบบรีโมตอื่นไม่สามารถใช้ uzeus ID ระบบ zeus สามารถอ่านและเขียนไปยังไดเร็กทอรีใดๆบนระบบ hera และสามารถส่งและร้องขอไฟล์โดยไม่สนใจว่าใครเป็นผู้เริ่มการติดต่อ ระบบ zeus ยังสามารถใช้คำสั่งบนระบบ hera

  ข้อควรสนใจ: ถ้าคุณให้สิทธิทั้งหมดในตัวอย่าง ก่อนหน้า นั่นเท่ากับว่าคุณให้ ID ล็อกอินบนระบบโลคัลกับ ผู้ใช้ทุกรายบนระบบรีโมต สิทธิที่เป็นอิสระดังกล่าวอาจเป็นอันตราย ต่อความปลอดภัย และมีการมอบให้กับระบบรีโมตที่ไซต์เดียวกันเท่านั้น

การบำรุงรักษา BNU

BNU ต้องถูกบำรุงรักษาเพื่อให้ทำงานได้อย่างถูกต้องบนระบบของคุณ

เมื่อต้องการบำรุงรักษา BNU:

- อ่านและลบไฟล์ล็อกเป็นระยะๆ
- ใช้คำสั่ง **uuq** และ **uustat** เพื่อตรวจสอบคิวของ BNU เพื่อให้แน่ใจว่างานถูกถ่ายโอนไปยังระบบรีโมตอย่างถูกต้อง

- กำหนดเวลาคำสั่งแบบอัตโนมัติที่โพลระบบรีโมตสำหรับงาน ส่งไฟล์ที่ยังไม่ได้ส่งคืนให้แก่ผู้ใช้ และส่งข้อความให้คุณเป็นระยะๆเกี่ยวกับสถานะของ BNU
- อัพเดตไฟล์คอนฟิกูเรชันเป็นระยะๆ เพื่อแสดงถึงการเปลี่ยนแปลงในระบบของคุณ

นอกจากนี้ ตรวจสอบกับผู้ดูแลระบบของระบบรีโมตเป็นครั้งคราวเพื่อติดตามความเปลี่ยนแปลงบนระบบรีโมตที่อาจมีผลกับการตั้งค่าของคุณ ตัวอย่างเช่น ถ้าผู้ดูแลระบบ `venus` เปลี่ยนรหัสผ่านของระบบของคุณ คุณต้องใส่รหัสผ่านใหม่ในไฟล์ `/etc/uucp/Systems` (หรือไฟล์ Systems ที่เหมาะสมที่ถูกระบุโดย `/etc/uucp/Sysfiles`) ก่อนที่ระบบของคุณสามารถจะล็อกอินกับระบบ `venus`

ล็อกไฟล์ของ BNU

BNU จะสร้างล็อกไฟล์และไฟล์ข้อผิดพลาดเพื่อติดตามกิจกรรมของมัน

ไฟล์เหล่านี้ต้องถูกตรวจสอบและลบเป็นช่วงเวลาเพื่อป้องกันไม่ให้มันใช้พื้นที่หน่วยเก็บข้อมูลบนระบบของคุณ BNU จัดเตรียมคำสั่งหลายคำสั่งสำหรับใช้ในการลบล็อกไฟล์ :

- `uulog`
- `uuclean`
- `uucleanup`
- `uudemon.cleanu.`

รันคำสั่งเหล่านี้แบบแมนวล หรือใช้ entry ในไฟล์ `/var/spool/cron/crontabs/uucp` เพื่อรันคำสั่งโดย **cron** daemon

ล็อกไฟล์ในไดเร็กทอรี .Log และ .Old:

BNU จะสร้างแต่ละล็อกไฟล์ในไดเร็กทอรี `/var/spool/uucp/.Log`

BNU จะสร้างล็อกไฟล์เหล่านี้สำหรับแต่ละระบบรีโมตที่สามารถเข้าถึงได้โดยใช้คำสั่ง **uucp, uucico, uux** และ**uuxqt** BNU ใส่ข้อมูลสถานะเกี่ยวกับแต่ละรายการในล็อกไฟล์ที่เหมาะสมแต่ละครั้งที่บางคนบนระบบใช้ BNU เมื่อกระบวนการ BNU มากกว่าหนึ่งกระบวนการกำลังรัน ระบบจะไม่สามารถเข้าถึงล็อกไฟล์ มันจะใส่ข้อมูลสถานะในไฟล์ที่แยกต่างหากด้วยส่วนที่นำหน้าว่า `.LOG` แทน

คำสั่ง **uulog** จะแสดงการสรุปของคำร้องขอ **uucp** หรือ **uux** โดยผู้ใช้หรือโดยระบบ คำสั่ง **uulog**จะแสดงไฟล์ อย่างไรก็ตาม คุณยังสามารถให้ BNU รวมล็อกไฟล์ในล็อกไฟล์ลำดับแรกโดยอัตโนมัติ นี่เรียกว่า *compacting* ล็อกไฟล์และสามารถทำโดยคำสั่ง **uudemon.cleanu** โดยทั่วไปรันโดย **cron** daemon

cron daemon จะรันคำสั่ง **uudemon.cleanu** คำสั่ง **uudemon.cleanu** ระรวมล็อกไฟล์ **uucico** และ **uuxqt** บนระบบโลคัล และเก็บมันในไดเร็กทอรี `/var/spool/uucp/.Old` ในเวลาเดียวกัน คำสั่งจะลบล็อกไฟล์เก่าที่ก่อนหน้านี้ถูกเก็บในไดเร็กทอรี `.Old` โดยดีฟอลต์ คำสั่ง **uudemon.cleanu** จะบันทึกล็อกไฟล์ที่อายุ 2 วัน

ถ้าพื้นที่หน่วยเก็บข้อมูลเป็นปัญหา พิจารณาลดจำนวนของวันที่ไฟล์ถูกเก็บ เพื่อติดตามรายการของ BNU เป็นช่วงเวลานาน พิจารณาเพิ่มจำนวนวันที่ไฟล์จะถูกเก็บ เพื่อเปลี่ยนเวลาดีฟอลต์สำหรับการบันทึกล็อกไฟล์ แก้ไขเซลล์โพรซีเดอร์สำหรับคำสั่ง **uudemon.cleanu** สคริปต์นี้ถูกเก็บในไดเร็กทอรี `/usr/sbin/uucp` และสามารถถูกแก้ไขด้วยสิทธิของ root

ล็อกไฟล์ BNU /.Admin:

BNU ยังรวบรวมข้อมูลและเก็บมันในไดเร็กทอรี /var/spool/uucp/.Admin ไดเร็กทอรีนี้ประกอบด้วยไฟล์ errors, xferstats, Foreign และ audit

ไฟล์เหล่านี้ต้องถูกตรวจสอบและลบเป็นครั้งคราวเพื่อประหยัดพื้นที่หน่วยเก็บข้อมูล BNU จะสร้างแต่ละไฟล์เมื่อมันถูกต้องการ

เมื่อระบบอื่นติดต่อระบบของคุณด้วยการเปิดโหมดการดีบัก **uucico** daemon มันจะใช้ **uucico** daemon บนระบบของคุณโดยเปิดระบบการดีบัก ข้อความการดีบักที่ถูกสร้างโดย daemon บนระบบโลคัลจะถูกเก็บในไฟล์ audit ไฟล์นี้อาจมีขนาดใหญ่ ตรวจสอบและลบไฟล์ audit บ่อยๆ

ไฟล์ errors จะบันทึกข้อผิดพลาดที่พบโดย **uucico** daemon การตรวจสอบไฟล์นี้สามารถช่วยคุณแก้ปัญหา เช่น การอนุญาตที่ไม่ถูกต้องบนไฟล์การทำงาน BNU

ไฟล์ xferstats ประกอบด้วยข้อมูลเกี่ยวกับสถานะของการถ่ายโอนไฟล์ทั้งหมด ตรวจสอบและลบไฟล์นี้เป็นครั้งคราว

ไฟล์ Foreign มีความสำคัญกับความปลอดภัยของระบบของคุณ เมื่อไรก็ตามที่ระบบที่ไม่รู้จักพยายามล็อกอินกับระบบโลคัล BNU จะเรียกใช้เชลล์โพรซีเดอร์ remote.unknown เชลล์โพรซีเดอร์นี้จะล็อกความพยายามในไฟล์ Foreign ไฟล์ Foreign ประกอบด้วยชื่อของระบบที่พยายามเรียกระบบโลคัลและถูกปฏิเสธ ถ้าระบบพยายามเรียกหลายครั้ง ใช้ข้อมูลนี้เมื่อพิจารณาว่ายอมให้ระบบนั้นเข้าถึงหรือไม่

ล็อกไฟล์ของระบบที่ใช้โดย BNU:

เนื่องจากกระบวนการ BNU จำนวนมากต้องการสิทธิของ root เพื่อทำงานของมัน BNU จะสร้าง entry บ่อยๆในไฟล์ /var/spool/sulog log

เช่นเดียวกัน การใช้ **cron** daemon เพื่อกำหนดเวลางาน BNU จะสร้างหลาย entry ในไฟล์ /var/spool/cron/log เมื่อใช้ BNU ตรวจสอบและลบไฟล์เหล่านี้

คำสั่งการบำรุงรักษาของ BNU

Basic Networking Utilities ประกอบด้วยหลายคำสั่งสำหรับมอนิเตอร์กิจกรรมของ BNU และการลบไดเร็กทอรีและไฟล์ BNU

คำสั่ง cleanup ของ BNU:

BNU ประกอบด้วยคำสั่ง 3 คำสั่งที่ลบไดเร็กทอรีและลบไฟล์ที่ยังไม่ถูกส่ง

ไอเท็ม	คำอธิบาย
uuclean	ลบไฟล์ทั้งหมดที่เก่ากว่าจำนวนชั่วโมงที่ระบุ จากไดเร็กทอรีการดูแลระบบ BNU ใช้คำสั่ง **uuclean** เพื่อระลุไดเร็กทอรีที่จะถูกลบ หรือชนิดของไฟล์ที่จะถูกลบ คุณยังสามารถบอกให้คำสั่งเพื่อแจ้งเจ้าของไฟล์ที่ถูกลบ คำสั่ง **uuclean** จะเหมือนกับคำสั่ง Berkeley ของคำสั่ง **uucleanup**
uucleanup	ทำงานเหมือนกับคำสั่ง **uuclean** อย่างไรก็ตาม คำสั่ง **uucleanup** จะตรวจสอบอายุของไฟล์โดยขึ้นอยู่กับ *วัน* แทนที่จะเป็นชั่วโมง ใช้คำสั่ง **uucleanup** เพื่อส่งข้อความเตือนไปยังผู้ใช้ที่ไฟล์ยังไม่ถูกถ่ายโอน แจ้งว่าไฟล์ยังคงอยู่ในคิว คำสั่ง **uucleanup** ยังลบไฟล์ที่เกี่ยวข้องกับระบบบรีโมตที่ระบุ
uudemon.cleanu	เชลล์โพรซีเดอร์ที่ใช้คำสั่ง **uulog** และ **uucleanup** เพื่อบีบอัดล็อกไฟล์ของ BNU และลบล็อกไฟล์และไฟล์ทำงานที่เก่ากว่า 3 วัน คำสั่ง **uudemon.cleanu** ถูกรันโดย **cron** daemon

คำสั่ง status-checking ของ BNU:

BNU ยังจัดเตรียมคำสั่งสำหรับการตรวจสอบสถานะของการถ่ายโอนและการล็อกไฟล์

ไอเท็ม	คำอธิบาย
uuq	แสดงงานที่อยู่ในคิวงานของ BNU ในปัจจุบัน ใช้คำสั่ง **uuq** เพื่อแสดงสถานะของงานที่ระบุของงานทั้งหมด ด้วยสิทธิของ root คุณสามารถใช้คำสั่ง **uuq** เพื่อลบงานจากคิว
uustat	จะให้ข้อมูลที่เหมือนกับที่ถูกให้โดยคำสั่ง **uuq** แต่ในรูปแบบที่ต่างกัน ใช้คำสั่ง **uustat** เพื่อตรวจสอบสถานะของงานและลบงานที่คุณเป็นเจ้าของ ด้วยสิทธิของ root คุณยังสามารถลบงานที่เป็นของผู้ใช้อื่น
uulog	แสดงการสรุปของคำร้องขอ **uucp** หรือ **uux** โดบผู้ใช้หรือโดยระบบ คำสั่ง **uulog** จะแสดงชื่อไฟล์ โปรดดู “ล็อกไฟล์ของ BNU” ในหน้า 479
uupoll	บังคับการโพลระบบรีโมต นี่จะมีประโยชน์เมื่องานสำหรับระบบนั้นรออยู่ในคิวและต้องการถูกถ่ายโอน ก่อนที่ระบบจะถูกกำหนดเวลาให้ถูกเรียกโดยอัตโนมัติ
uusnap	แสดงการสรุปแบบสั้นมากของสถานะของ BNU สำหรับแต่ละระบบรีโมต คำสั่งนี้จะแสดงจำนวนของไฟล์ที่รอที่จะถ่ายโอน อย่างไรก็ตามมันจะไม่แสดงว่ามันไฟล์รอมานานแค่ไหนแล้ว คำสั่ง **uusnap** เป็นเวอร์ชัน Berkeley ของคำสั่ง **uustat**

เชลล์โพรซีเดอร์ของ BNU:

BNU จะมี 2 เชลล์โพรซีเดอร์ที่ถูกใช้สำหรับการบำรุงรักษา

ไอเท็ม	คำอธิบาย
uudemon.cleanu	กล่าวถึงใน “คำสั่ง cleanup ของ BNU” ในหน้า 480
uudemon.admin	ใช้คำสั่ง **uustat** คำสั่ง **uustat** จะรายงานสถานะงานของ BNU มันจะส่งผลลัพธ์เป็นเมลไปยังล็อกอิน ID uucp คุณสามารถแก้ไขเชลล์โพรซีเดอร์ **uudemon.admin** เพื่อส่งเมลไปยังที่อื่น หรือใช้โปรแกรมเมลเพื่อส่งเมลทั้งหมดสำหรับล็อกอิน ID uucp ไปยังผู้ใช้ที่รับผิดชองการดูแลระบบ BNU

เชลล์โพรซีเดอร์เหล่านี้ถูกเก็บในไดเร็กทอรี `/usr/sbin/uucp` คัดลอกโพรซีเดอร์และแก้ไขชุดคัดลอก ถ้าคุณต้องการเปลี่ยนสิ่งที่มันทำ รันโพรซีเดอร์จากบรรทัดรับคำสั่ง หรือกำหนดเวลาให้มันถูกรันโดย **cron** daemon

เพื่อรันคำสั่ง **uudemon.cleanu** และ **uudemon.admin** โดยอัตโนมัติ ลบอักขระหมายเหตุ (#) จากต้นของบรรทัดที่เกี่ยวข้องในไฟล์ `/var/spool/cron/crontabs/uucp`

ชื่อพาธ BNU

ชื่อพาธที่ถูกใช้กับคำสั่ง Basic Networking Utilities (BNU) สามารถถูกใส่ในวิธีที่แตกต่างกันหลายวิธี

ชื่อพาธประกอบด้วย root ไดเร็กทอรี หรือ พาธแบบซ็อตคัตไปยังเป้าหมาย ซึ่งเป็นชื่อของระบบรีโมตหรือระบบ การแตกต่างกันของแต่ละพาธจะเป็นไปตามแนวทางที่ระบุ

ชื่อพาธแบบเต็ม

ชื่อพาธแบบเต็มเริ่มที่ root และติดตามไดเร็กทอรีทั้งหมดลงไปถึงไดเร็กทอรีและไฟล์เป้าหมาย

ตัวอย่างเช่น `/etc/uucp/Devices` อ้างถึงไฟล์ `Devices` ในไดเร็กทอรี `uucp` ในไดเร็กทอรี root `etc`

พิมพ์สแลชนำหน้าเสมอ (`/`) เพื่อแสดงถึงไดเร็กทอรี root แยกแต่ละส่วนในพาธด้วยสแลชเสมอ (`/`)

ชื่อพาธแบบสัมพันธ์

ชื่อพาธแบบสัมพันธ์จะลิสต์เฉพาะไดเร็กทอรีที่สัมพันธ์กับไดเร็กทอรีปัจจุบัน

ตัวอย่างเช่น ถ้าไดเร็กทอรีปัจจุบันคือ `/usr/bin` และไดเร็กทอรีเป้าหมายคือ `/usr/bin/reports` พิมพ์ชื่อพาธแบบสัมพันธ์ `reports` (โดยไม่มีสแลชนำหน้า)

ชื่อพาธแบบสัมพันธ์สามารถใช้กับคำสั่ง **cu, uucp,** and **uux** และกับชื่อของไฟล์ต้นทางในคำสั่ง **uuto**

> **หมายเหตุ :** ชื่อพาธแบบสัมพันธ์อาจใช้ไม่ได้กับคำสั่ง BNU ทั้งหมด ถ้าณมีปัญหาเกี่ยวกับการใช้ชื่อพาธแบบสัมพันธ์ พิมพ์คำสั่งอีกครั้งด้วยชื่อพาธแบบเต็ม

~[อ็อพชัน]ชื่อพาธ

~`[อ็อพชัน]` ชื่อพาธจะแทนไดเร็กทอรีหลักของผู้ใช้ที่ระบุ

tilde (~) สามารถถูกใช้เป็นช็อตคัตไปนังไดเร็กทอรีนั้นๆ

ตัวอย่างเช่น ~`jane` อ้างถึงไดเร็กทอรีหลักของผู้ใช้ `jane` entry ~`uucp` หรือ ~ (tilde อย่างเดียว) อ้างถึงพับลิกไดเร็กทอรี BNU บนระบบรีโมต ชื่อพาธแบบเต็มสำหรับพับลิกไดเร็กทอรี BNU คือ `/var/spool/uucppublic`

> **หมายเหตุ:** การใช้ tilde ไม่ควรทำให้สับสนกับการใช้ tilde อื่นใน BNU tilde ยังถูกใช้เป็นตัวนำหน้าคำสั่งสำหรับการประมวลผลบนระบบโลคัล เมื่อล็อกอินกับระบบรีโมตเมื่อใช้คำสั่ง **cu**

system_name!ชื่อพาธ

system_name! ชื่อพาธจะระบุพาธไปยังไฟล์บนระบบอื่น

ตัวอย่างเช่น `distant!/account/march` จะอ้างถึงไฟล์ `march` ในไดเร็กทอรี `account` บนระบบรีโมต `distant`

system_name!system_name!ชื่อพาธ

system_name!*system_name*! ชื่อพาธจะระบุเส้นทางไปยังหลายระบบ

ตัวอย่างเช่น ถ้าระบบชื่อ `distant` สามารถถูกเข้าถึงผ่านระบบอื่นที่ชื่อ `near` ดังนั้นชื่อพาธจะเป็น `near!distant!/account/march`

แยกชื่อระบบด้วยเครื่องหมายตกใจ (!). ในกรณีของชื่อพาธแบบหลายระบบ กฎของการแยกส่วนต่างๆด้วยเครื่องหมายสแลช (/) จะไม่สามารถใช้ได้กับชื่อระบบ อย่างไรก็ตาม กฎนี้ไม่ได้ใช้กับระบบที่เป็นตัวสุดท้าย ที่ระบุไดเร็กทอรีและไฟล์อย่างชัดเจน

หมายเหตุ: เมื่อคุณใช้ bourne เชลล์ให้แยกชื่อระบบด้วยเครื่องหมายตกใจ (!). เมื่อคุณใช้ BNU in ใน C หรือ korn เชลล์ ให้นำหน้าเครื่องหมายตกใจด้วยแบ็คสแลช (\) แบ็คสแลชเป็นอักขระ escape ที่จำเป็นสำหรับการแปลตัวอักขระต่อไป ว่าเป็นตัวอักษรแทนที่จะเป็นอักขระพิเศษ

BNU daemons

ซอร์ฟแวร์ BNU จะประกอบด้วย 4 daemon ที่ถูกเก็บในไดเร็กทอรี `/usr/sbin/uucp`

ไอเท็ม	คำอธิบาย
uucico	อำนวยความสะดวกในการถ่ายโอนไฟล์ (ดูที่ "uucico daemon")
uusched	อำนวยความสะดวกการกำหนดเวลาคำร้องขอการทำงานของไฟล์ที่ถูกเข้าคิวในไดเร็กทอรี spooling แบบโลคัล (ดูที่ "uusched daemon" ในหน้า 484)
uuxqt	อำนวยความสะดวกในการประมวลผลคำสั่งรีโมต (ดูที่ "uuxqt daemon" ในหน้า 484)
uucpd	อำนวยความสะดวกในการสื่อสารโดยใช้ TCP/IP (ดูที่ "uucpd daemon" ในหน้า 484)

uucico, uusched และ **uuxqt** daemons ถูกสตาร์ทโดย **cron** daemon โดยขึ้นอยู่กับกำหนดเวลาที่ถูกตั้งโดยผู้ดูแลระบบ BNU ด้วยสิทธิของ root คุณสามารถสตาร์ท daemons เหล่านี้แบบแมนวล **uucpd** daemon ควรถูกสตาร์ทโดย TCP/IP **inetd** daemon

uucico daemon

uucico daemon ส่งลำเรียงไฟล์ที่ต้องการเพื่อส่งข้อมูลจากระบบหนึ่งไปยังระบบอื่น

คำสั่ง **uucp** และ **uux** สตาร์ท **uucico** daemon เพื่อถ่ายโอนคำสั่ง ข้อมูล และประมวลผลไฟล์ไปยังระบบที่กำหนด **uucico** daemon ยังถูกสตาร์ทเป็นระยะๆโดย BNU scheduler **uusched** daemon. เมื่อถูกสตาร์ทโดย **uusched** daemon **uucico** daemon จะพยายามติดต่อระบบอื่นและประมวลผลวิธีการในไฟล์คำสั่ง

เพื่อรันวิธีการในไฟล์คำสั่ง ลำดับแรก **uucico** daemon จะตรวจสอบไฟล์ `/etc/uucp/Systems` (หรือไฟล์อื่นที่ระบุโดย `/etc/uucp/Sysfiles`) สำหรับระบบที่จะถูกเรียก จากนั้น daemon จะตรวจสอบ entry ของไฟล์ `Systems` สำหรับเวลาที่ถูกต้องที่จะเรียก ถ้าเวลาถูกต้อง **uucico** daemon จะตรวจสอบฟิลด์ *Type* และ *Class* และเข้าถึงไฟล์ `/etc/uucp/Devices` (หรือไฟล์อื่นที่ถูกระบุโดย `/etc/uucp/Sysfiles`) สำหรับอุปกรณ์ที่ตรง

หลังจากหาอุปกรณ์ **uucico** daemon จะตรวจสอบไดเร็กทอรี `/var/locks` เพื่อล็อกไฟล์สำหรับอุปกรณ์ ถ้ามีอยู่ daemon จะตรวจสอบอุปกรณ์อื่นของชนิดที่ถูกร้องขอและความเร็ว

เมื่ออุปกรณ์ไม่พร้อมใช้งาน daemon จะกลับไปที่ไฟล์ `Systems` สำหรับ entry อื่นสำหรับระบบรีโมต ถ้ามีอยู่ daemon จะทำกระบวนการค้นหาอุปกรณ์ซ้ำ ถ้าไม่พบ entry อื่น daemon จะสร้าง entry ในไฟล์ `/var/spool/uucp/.Status/`*SystemName* สำหรับระบบรีโมตนั้นและไปที่คำร้องขอถัดไป ไฟล์คำสั่งจะยังคงอยู่ในคิว **uucico** daemon พยายามถ่ายโอนอีกครั้งภายหลัง ความพยายามครั้งล่าสุดเรียกว่า *retry*

เมื่อ **uucico** daemon เข้าถึงระบบรีโมต มันใช้วิธีการในไฟล์ `Systems` เพื่อล็อกอิน นี่ทำให้อินสแตนซ์ของ **uucico** daemon ถูกใช้บนระบบรีโมตด้วย

uucico daemons จำนวน 2 ตัว หนึ่งตัวบนแต่ละระบบทำงานร่วมกันเพื่อทำการถ่ายโอน **uucico** daemon บนระบบที่ทำการเรียกจะควบคุมลิงก์ จะระบุคำร้องขอที่จะทำ **uucico** daemon บนระบบรีโมตจะตรวจสอบการอนุญาตบนโลคัลเพื่อดูว่าทำยอมให้คำร้องขอถูกกระทำหรือไม่ ถ้าใช่ การถ่ายโอนจะเริ่มต้น

หลังจาก **uucico** daemon บนระบบที่ทำการเรียกทำการถ่ายโอยคำร้องขอทั้งหมดที่มันมีสำหรับระบบรีโมตเสร็จ มันจะส่งคำร้องขอให้วางหู เมื่อรีโมต **uucico** daemon มีรายการที่จะส่งไปยังระบบที่เป็นผู้เรียก มันจะปฏิเสธคำร้องขอเพื่อวางหู และ daemons สองตัวจะทำบทบาทกลับกัน

หมายเหตุ: ทั้งไฟล์ `/etc/uucp/Permissions` บนระบบโลคัล หรือไฟล์ `/etc/uucp/Permissions` บนระบบรีโมตสามารถปฏิเสธการกลับบทบาทของ daemons ในกรณีนี้ ระบบรีโมตต้องรอเพื่อจะถ่ายโอนไฟล์จนกระทั่งมันเรียกไปยังระบบโลคัล

เมื่อไม่มีสิ่งที่จะถ่ายโอนในทั้งสองทิศทาง **uucico** daemons ทั้งสองจะวางหู ที่เวลานี้ **uuxqt** daemon ("uuxqt daemon") ถูกเรียกเพื่อประมวลผลคำร้องขอคำสั่งของรีโมต

ตลอดกระบวนการถ่ายโอน **uucico** daemons บนทั้งสองระบบจะล็อกข้อความในล็อกของ BNU และไฟล์ข้อผิดพลาด

uusched daemon

uusched daemon จะกำหนดเวลาการถ่ายโอนไฟล์ที่ถูกเข้าคิวในไดเร็กทอรี spooling บนระบบโลคัล

ไดเร็กทอรี spooling คือ `/var/spool/uucppublic` เมื่อ **uusched** daemon ถูกใช้ มันจะสแกนไดเร็กทอรี spooling เพื่อหาไฟล์คำสั่ง จากนั้นสุ่มไฟล์และสตาร์ท **uucico** daemon **uucico** daemon จะถ่ายโอนไฟล์

uuxqt daemon

เมื่อผู้ใช้ใช้คำสั่ง **uux** เพื่อรันคำสั่งที่ถูกระบุบนระบบที่กำหนด **uuxqt** daemon จะรันคำสั่ง

หลังจากสร้างไฟล์ที่จำเป็นคำสั่ง **uux** จะสตาร์ท **uucico** daemon ซึ่งถ่ายโอนไฟล์เหล่านั้นไปยังพับลิกสพูลไดเร็กทอรีบนระบบที่ระบุ

uuxqt daemon จะค้นหาไดเร็กทอรีเป็นระยะๆ สำหรับคำร้องขอการประมวลผลคำสั่งบนทุกระบบที่เชื่อมต่ออยู่ เมื่อมันหาคำร้องขอนั้น **uuxqt** daemon จะตรวจสอบไฟล์และการอนุญาตที่จำเป็น จากนั้น ถ้าได้รับอนุญาต daemon จะรันคำสั่งที่ระบุ

uucpd daemon

uucpd daemon ต้องการสามารถรันบนระบบรีโมตก่อนที่ BNU สามารถเริ่มต้นการสื่อสารกับรีโมตคอมพิวเตอร์ด้วย **Transmission Control Protocol/Internet Protocol (TCP/IP)**

uucpd daemon เป็นเซิร์ฟเวอร์ย่อยของ TCP/IP **inetd** daemon และถูกสตาร์ทโดย **inetd** daemon

โดยดีฟอลต์ **uucpd** daemon ถูกทำหมายเหตุในไฟล์ `inetd.conf` เพื่อใช้มัน คุณต้องลบอักขระหมายเหตุและรีสตาร์ท **inetd** อย่างไรก็ตาม ถ้ามันถูกเปลี่ยนบนระบบของคุณ คุณอาจต้องตั้งค่า **inetd** daemon ใหม่เพื่อสตาร์ท**uucpd** daemon

ความปลอดภัยของ BNU

เนื่องจากระบบอื่นติดต่อระบบของคุณเพื่อล็อกอิน ถ่ายโอนไฟล์ และใช้คำสั่ง BNU ได้จัดเตรียมเกี่ยวกับการสร้างความปลอดภัย

ความปลอดภัยของ BNU ให้คุณสามารถจำกัดว่าผู้ใช้ของระบบรีโมตสามารถทำอะไรบนระบบโลคัล (ผู้ใช้ของระบบรีโมตยังสามารถจำกัดว่าคุณสามารถทำอะไรบนระบบรีโมตได้เช่นกัน) BNU จะรันหลาย daemons เพื่อทำกิจกรรมนี้และใช้ไดเร็กทอรีการจัดการเพื่อเก็บไฟล์ที่มันต้องการ BNU ยังเก็บล็อกของกิจกรรมของมันเอง

ความปลอดภัยของ BNU ทำงานบนหลายระดับ เมื่อคุณตั้งค่า BNU คุณสามารถกำหนด :

- ใครบนระบบของคุณที่สามารถเข้าถึงไฟล์ BNU
- ระบบรีโมตใดที่ระบบของคุณสามารถติดต่อ
- ผู้ใช้บนระบบรีโมตจะล็อกอินระบบของคุณอย่างไร
- ผู้ใช้บนระบบรีโมตสามารถทำอะไรบนระบบของคุณเมื่อล็อกอินแล้ว

ล็อกอิน ID uucp

เมื่อ BNU ถูกติดตั้ง ไฟล์คอนฟิกูเรชันทั้งหมด daemons คำสั่งจำนวนมาก และเชลล์โพรซีเดอร์ จะถูกเป็นเจ้าของล็อกอิน ID

ID ล็อกอิน uucp มี ID ผู้ใช้ (UID) เป็น 5 และ ID กลุ่ม (GID) เป็น 5 **cron** daemon จะอ่านไฟล์ `/var/spool/cron/crontabs/uucp` เพื่อจัดตารางเวลางานอัตโนมัติสำหรับ BNU

โดยทั่วไป การล็อกอินเป็นผู้ใช้ uucp ไม่สามารถทำได้ เพื่อเปลี่ยนไฟล์ที่เป็นเจ้าของโดยล็อกอิน ID uucp ให้ล็อกอินด้วยสิทธิของ root

ข้อควรสนใจ: การยอมให้ระบบรีโมตพื่อล็อกอินกับระบบโลคัลด้วยล็อกอิน uucp จะทำลายความปลอดภัยของระบบโลคัลอย่างรุนแรง ระบบรีโมตที่ล็อกอินด้วย uucp ID สามารถแสดงและอาจแก้ไขไฟล์ `Systems` and `Permissions` โดยขึ้นอยู่กับการอนุญาตอื่นที่ถูกระบุใน entry LOGNAME แนะนำว่าคุณสร้าง ID การล็อกอินของ BNU อื่นสำหรับระบบรีโมต และ เก็บล็อกอิน ID ของ uucp สำหรับผู้ดูแลระบบ BNU บนระบบโลคัล เพื่อให้ได้ความปลอดภัยที่ดีที่สุด แต่ละระบบรีโมตที่ติดต่อกับระบบโลคัลต้องมีล็อกอิน ID ที่เป็นหนึ่งเดียว กับตัวเลข UID ที่เป็นหนึ่งเดียว

ระบบปฏิบัติการจะจัดเตรียมล็อกอิน ID nuucp ดีฟอลต์ สำหรับถ่ายโอนไฟล์

ล็อกอิน ID ของ BNU

เชลล์การเริ่มต้นสำหรับล็อกอิน ID ของ BNU คือ **uucico** daemon (`/usr/sbin/uucp/uucico`)

เมื่อระบบรีโมตเรียกมายังระบบโลคัล มันจะสตาร์ท **uucico** daemon บนระบบของคุณโดยอัตโนมัติ ล็อกอิน IDs สำหรับ BNU มี ID ของกลุ่ม uucp เป็น 5

ล็อกอิน ID ถูกใช้โดยระบบรีโมตที่ต้องการรหัสผ่าน เพื่อที่จะป้องกันความปลอดภัยจากการพร้อมต์ล็อกอิน ID ของ BNU สำหรับรหัสผ่านใหม่ เมื่อระบบรีโมตล็อกอิน คุณต้องตั้งรหัสผ่านทันทีที่คุณสร้างบัญชีผู้ใช้ ในการทำดังกล่าว ใช้คำสั่ง **passwd** ตามด้วยคำสั่ง **pwdadm** ตัวอย่างเช่น เพื่อตั้งรหัสผ่านสำหรับล็อกอิน ID `nuucp` ล็อกอินด้วยผู้ใช้ root และใช้คำสั่งต่อไปนี้ :

```
passwd nuucp
```

```
pwadm -f NOCHECK
nuucp
```

ระบบจะพร้อมต์ให้คุณใส่รหัสผ่านสำหรับล็อกอิน ID `nuucp` การทำขั้นตอนเหล่านี้ทำให้ระบบรีโมตล็อกอินโดยไม่ถูกพร้อมต์ให้ใส่รหัสผ่านใหม่ทันที (ซึ่งล็อกอิน ID `nuucp` แบบ batch-oriented ไม่สามารถทำได้)

หลังจากสร้างล็อกอิน ID สำหรับระบบรีโมต แจ้งล็อกอิน ID และรหัสผ่านให้ผู้ดูแลระบบ BNU เพื่อเข้าถึงระบบของคุณ

ผู้ใช้ที่มีสิทธิของ root สามารถตั้งค่าล็อกอิน ID ของผู้ดูแลระบบ BNU นี่จะมีประโยชน์ถ้าคุณต้องการมอบหมายงานผู้ดูแลระบบ BNU ให้กับผู้ใช้ที่ไม่มีสิทธิของ root ล็อกอิน ID ของผู้ดูแลระบบ BNU ควรมีความปลอดภัยของรหัสผ่าน UID เป็น 5 และอยู่ใน ID ของกลุ่มของ uucp เป็น 5 เชลล์การล็อกอินสำหรับล็อกอินของผู้ดูแลระบบควรเป็นโปรแกรม `/usr/bin/sh`

(แทนที่จะเป็น **uucico** daemon) การให้ล็อกอินผู้ดูแลระบบ BNU UID เป็น 5 จะทำให้ได้สิทธิเหมือนกับล็อกอิน ID **uucp** ดังนั้น สำหรับความปลอดภัย ระบบรีโมตไม่ควรได้รับอนุญาตให้ล็อกอินเป็นผู้ดูแลระบบ BNU

ความปลอดภัยและไฟล์ Systems และ remote.unknown

บนระบบ BNU ส่วนใหญ่ เฉพาะระบบรีโมตที่ถูกลิสต์ในไฟล์ /etc/uucp/Systems หรือหนึ่งในที่ใช้แทนที่มัน (ถูกระบุในไฟล์ Sysfiles) สามารถล็อกอินกับระบบโลคัล

สคริปต์ /usr/sbin/uucp/remote.unknown จะถูกประมวลผลเมื่อระบบที่ไม่รู้จักพยายามเรียกไปยังระบบโลคัล สคริปต์นี้จะปฏิเสธที่จะให้ระบบที่ไม่รู้จักล็อกอิน และสร้าง entry ในไฟล์ /var/spool/uucp/.Admin/Foreign บันทึกเวลาของการพยายามล็อกอิน

โดยใช้สิทธิของ root หรือผู้ดูแลระบบ BNU คุณสามารถแก้ไขเชลล์โพรซีเดอร์ remote.unknown เพื่อล็อกข้อมูลเพิ่มเติมเกี่ยวกับระบบรีโมต หรือเพื่อเก็บข้อมูลในไฟล์อื่น ตัวอย่างเช่น คุณสามารถแก้ไขเชลล์โพรซีเดอร์เพื่อส่งเมลไปยังผู้ดูแลระบบ BNU เมื่อระบบที่ไม่รู้จักพยายามที่จะล็อกอิน

โดยการยกเลิกสิทธิในการประมวลผลบนเชลล์โพรซีเดอร์ remote.unknown คุณสามารถให้เครื่องที่ไม่รู้จักล็อกอิน ในกรณีนี้ คุณควรเพิ่ม entry MACHINE=OTHER เข้ากับไฟล์ /etc/uucp/Permissions เพื่อให้การอนุญาตสำหรับเครื่องที่ไม่รู้จัก

ระบบของคุณสามารถติดต่อเฉพาะระบบรีโมตที่ถูกลิสต์ในไฟล์ Systems นี่จะป้องกันผู้ใช้บนระบบของคุณไม่ให้ติดต่อระบบที่ไม่รู้จัก

ความปลอดภัยและไฟล์ Permissions

พิจารณาปัญหาเกี่ยวกับความปลอดภัยต่อไปนี้เมื่อใช้ไฟล์ Permissions

ไฟล์ /etc/uucp/Permissions จะระบุ

- ชื่อผู้ใช้ล็อกอินแบบรีโมต สำหรับล็อกอินกับระบบโลคัล
- อนุมัติคำสั่งและสิทธิสำหรับระบบรีโมตล็อกอินกับระบบโลคัล

ไฟล์ /etc/uucp/Permissions ประกอบด้วย entry สองชนิด :

ไอเท็ม	คำอธิบาย
LOGNAME	กำหนดชื่อล็อกอิน และสิทธิที่เกี่ยวข้องกับมัน entry LOGNAME จะมีผลเมื่อระบบรีโมตเรียกมายังระบบโลคัลและพยายามล็อกอิน
MACHINE	กำหนดชื่อเครื่อง และสิทธิที่เกี่ยวข้องกับมัน entry MACHINE จะมีผลเมื่อระบบรีโมตพยายามใช้คำสั่งบนระบบโลคัล

อ็อพชันในไฟล์ Permissions ให้คุณสามารถสร้างความปลอดภัยระดับต่างๆสำหรับแต่ละระบบรีโมต ตัวอย่างเช่น ถ้าระบบรีโมตหลายระบบใช้ล็อกอิน ID เดียวบนระบบโลคัล ใช้อ็อพชัน VALIDATE เพื่อให้แต่ละระบบรีโมตใช้ล็อกอิน ID ที่เป็นหนึ่งเดียว อ็อพชัน SENDFILES, REQUEST และ CALLBACK ระบุว่าระบบใดที่ควบคุม ทำให้ระบบโลคัลควบคุมรายการ ถ้าจำเป็น

อ็อพชัน READ, WRITE, NOREAD และ NOWRITE กำหนดการเข้าถึงไดเร็กทอรีที่ระบุบนระบบโลคัล อ็อพชันเหล่านี้ยังควบคุมที่ที่ผู้ใช้ระบบรีโมตสามารถเก็บข้อมูล อ็อพชัน COMMANDS จำกัดจำนวนของคำสั่งของผู้ใช้บนระบบรีโมตที่สามารถประมวลผลบนระบบโลคัล อ็อพชัน COMMANDS=ALL ยอมให้สิทธิทั้งหมดกับระบบที่เชื่อมโยงใกล้ชิดกับระบบของคุณ

ข้อควรสนใจ: อ็อพชัน COMMANDS=ALL สามารถทำลายความปลอดภัยของระบบของคุณ

การสื่อสารระหว่างระบบโลคัลและรีโมต

เพื่อที่จะสื่อสารระหว่างระบบรีโมตและระบบโลคัล ระบบรีโมตต้องมีลิงก์ hardwire หรือโมเด็มกับระบบโลคัล ติดตั้งระบบปฏิบัติการ UNIX-based และ BNU หรือเวอร์ชันอื่นของ UNIX-to-UNIX Copy Program (UUCP) รันอยู่

> **หมายเหตุ:** คุณสามารถใช้ BNU เพื่อสื่อสารกับระบบที่ไม่ใช่ UIX แต่การเชื่อมต่อนั้นอาจต้องการฮาร์ดแวร์หรือซอฟต์แวร์เพิ่มเติม

BNU มีสองคำสั่งที่ให้คุณสามารถสื่อสารกับระบบรีโมต คำสั่ง **cu** เชื่อมต่อระบบบน hardwire หรือสายโทรศัพท์ คำสั่ง **ct** เชื่อมต่อระบบผ่านสายโทรศัพท์เท่านั้น โดยใช้โมเด็ม

ใช้คำสั่ง **cu** เพื่อเริ่มการสื่อสารระหว่างเน็ตเวิร์กเมื่อคุณรู้หมายเลขโทรศัพท์ หรือชื่อของระบบเป้าหมาย เพื่อใช้คำสั่ง **ct** คุณ *ต้อง* มีหมายเลขโทรศัพท์ของระบบเป้าหมาย

> **หมายเหตุ:** คำสั่งที่สาม **tip** ทำงานเหมือนกับคำสั่ง **cu** อย่างไรก็ตาม คำสั่ง **tip** เป็นส่วนประกอบของ Berkeley Software Distribution (BSD) เวอร์ชันของโปรแกรม UCP การติดตั้งมันกับ BNU ต้องการการตั้งค่าพิเศษ

การสื่อสารกับระบบอื่นโดยใช้ hardwire หรือโมเด็ม

ใช้คำสั่ง **cu** จากระบบโลคัลของคุณเพื่อทำงานการสื่อสารเหล่านี้ :

- เริ่มการเชื่อมต่อกับระบบรีโมตที่ระบุ
- ล็อกอินกับระบบรีโมต
- ทำงานบนระบบรีโมต
- สลับกลับไปกลับมา ทำงานพร้อมกันบนทั้งสองระบบ

ถ้าระบบรีโมตรันภายใต้ระบบปฏิบัติงานเดียวกัน คุณสามารถใช้คำสั่งธรรมดาจากระบบโลคัลของคุณ ตัวอย่างเช่น คุณสามารถใช้คำสั่งเพื่อเปลี่ยนไดเร็กทอรี ลิสต์เนื้อหาของไดเร็กทอรี ดูไฟล์ หรือส่งไฟล์ไปยังคิวการพิมพ์บนระบบรีโมต เพื่อใช้คำสั่งสำหรับใช้บนระบบโลคัล หรือเพื่อใช้คำสั่งรีโมต และการแลกเปลี่ยนไฟล์ ใช้คำสั่งโลคัล **cu** แบบพิเศษที่นำหน้าด้วย tilde (~)

การสื่อสารกับระบบอื่นโดยใช้โมเด็ม

ใช้คำสั่ง **ct** เพื่อสื่อสารโดยกับระบบอื่นโดยใช้โมเด็ม

ใส่คำสั่ง **ct** ตามด้วยหมายเลขโทรศัพท์ เพื่อเรียกไปยังระบบรีโมต เมื่อทำการเชื่อมต่อแล้ว พร็อมต์ล็อกอินของรีโมตจะถูกแสดงบนหน้าจอของคุณ

คำสั่ง **ct** จะมีประโยชน์ภายใต้เงื่อนไขเฉพาะ สำหรับรายละเอียดเกี่ยวกับการใช้คำสั่ง BNU **ct** ดูที่ :

- “หมุนหมายเลขโทรศัพท์จนกว่าจะสามารถเชื่อมต่อได้”
- “หมุนหมายเลขโทรศัพท์หลายหมายเลขจนกว่าจะสามารถเชื่อมต่อได้” ในหน้า 488

หมุนหมายเลขโทรศัพท์จนกว่าจะสามารถเชื่อมต่อได้

โพรซีเดอร์นี้จะอธิบายวิธีใช้คำสั่ง **ct** เพื่อให้หมุนหมายเลขโมเด็มรีโมตจนกว่าจะสามารถเชื่อมต่อได้ หรือจนกว่าจะผ่านเวลาที่ถูกกำหนดไว้

ระบบที่จะถูกเรียกต้องรัน Basic Networking Utilities (BNU) หรือเวอร์ชันของ UNIX-to-UNIX Copy Program (UUCP) บางเวอร์ชัน

ที่บรรทัดรับคำสั่งบนระบบโลคัล พิมพ์ :

```
ct -w3 5550990
```

ซึ่งจะหมุนไปยังรีโมตโมเด็มที่หมายเลข 555-0990 แฟล็ก **-w3** และหมายเลข จะบอกคำสั่ง **ct** ให้หมุนไปยังรีโมตโมเด็มเป็นช่วงเวลาหนึ่งนาทีจนกว่าจะสามารถติดต่อได้ หรือจนกว่าจะครบ 3 นาที

หมายเหตุ: พิมพ์หมายเลขโทรศัพท์ของรีโมตโมเด็มบนบรรทัดรับคำสั่ง **ct** ได้ทั้งก่อนหรือหลังแฟล็ก

หมุนหมายเลขโทรศัพท์หลายหมายเลขจนกว่าจะสามารถเชื่อมต่อได้

โพรซีเดอร์นี้จะอธิบายวิธีใช้คำสั่ง **ct** เพื่อให้หมุนหมายเลขรีโมตโมเด็มหลายหมายเลขจนกว่าจะสามารถเชื่อมต่อได้ หรือจนกว่าจะผ่านเวลาที่ถูกกำหนดไว้

ระบบที่จะถูกเรียกต้องรัน Basic Networking Utilities (BNU) หรือเวอร์ชันของ UNIX-to-UNIX Copy Program (UUCP) บางเวอร์ชัน

ที่บรรทัดรับคำสั่งบนระบบโลคัล พิมพ์ :

```
ct -w6 5550990 5550991 5550992 5550993
```

ซึ่งจะหมุนหมายเลขของรีโมตโมเด็มที่หมายเลข 555-0990, 555-0991, 555-0992 และ 555-0993 แฟล็ก **-w6** และหมายเลขจะบอกคำสั่ง **ct** ให้หมุนไปยังรีโมตโมเด็มเป็นช่วงเวลาหนึ่งนาทีจนกว่าจะสามารถติดต่อได้ หรือจนกว่าจะครบ 6 นาที

หมายเหตุ: พิมพ์หมายเลขโทรศัพท์ของรีโมตโมเด็มบนบรรทัดรับคำสั่ง **ct** ได้ทั้งก่อนหรือหลังแฟล็ก

การแลกเปลี่ยนไฟล์ระหว่างระบบโลคัลและระบบรีโมต

ถารถ่ายโอนไฟล์ระหว่างระบบเป็นแอ็พพลิเคชันทั่วไปส่วนมากของ Basic Networking Utilities (BNU) BNU ใช้คำสั่ง 4 คำสั่ง **uucp, uusend, uuto** และ **uupick** เพื่อแลกเปลี่ยนไฟล์ระหว่างระบบโลคัลและระบบรีโมต

คำสั่ง **uucp** เป็นยูทิลิตีการถ่ายโอยข้อมูล BNU ลำดับแรก คำสั่ง **uusend** เป็นคำสั่งการถ่ายโอยของ Berkeley Software Distribution (BSD) ที่ถูกรวมเข้าใน BNU คำสั่ง **uuto** และ **uupick** เป็นคำสั่งส่งและรับพิเศษที่ทำงานกับคำสั่ง **uucp**

คำสั่ง BNU **uuencode** และ **uudecode** จะช่วยเหลือการถ่ายโอนไฟล์ คำสั่งเหล่านี้เข้ารหัสและถอดรหัสไฟล์ไบนารีที่ถูกส่งโดยเครื่องมือ BNU mail

การส่งและรับไฟล์

คำสั่งที่ใช้สำหรับการส่งและรับไฟล์ผ่านการเชื่อมต่อ BNU จะรวมคำสั่ง **uucp** และ **uusend**

ใช้คำสั่ง **uucp** และอ็อพชันเพื่อแลกเปลี่ยนไฟล์ภายในระบบโลคัลของคุณ ระหว่างระบบโลคัลและระบบรีโมต และระหว่างระบบรีโมต ตัวอย่างเช่น อ็อพชัน **uucp** สามารถสร้างไดเร็กทอรีเพื่อเก็บไฟล์บนด้านที่เป็นผู้รับ หรือส่งข้อความอีเมลเกี่ยวกับความสำเร็จหรือความล้มเหลวของการถ่ายโอนไฟล์

ใช้คำสั่ง **uusend** เพื่อส่งไฟล์ไปยังระบบรีโมตที่ไม่ได้ถูกเชื่อมต่อโดยตรงกับระบบผู้ส่ง แต่สามารถเข้าถึงได้ผ่านการเชื่อมต่อ BNU ด้วยการใช้อ็อพชันที่น้อยกว่าคำสั่ง **uucp uusend** จะถูกรวมกับยูทิลิตี BNU เพื่อสร้างความพอใจให้กับการกำหนดค่าตามความชอบของผู้ใช้ BSD UNIX-to-UNIX Copy Program (UUCP)

การส่งไฟล์ไปยังผู้ใช้ที่ระบุ

เพื่อที่จะส่งไฟล์ไปยังผู้ใช้ที่ระบุ ระบบที่ส่ง และรับต้องรัน Basic Networking Utilities (BNU) หรือบางเวอร์ชันของ UNIX-to-UNIX Copy Program (UUCP)

ใช้คำสั่ง **uuto** เพื่อส่งไฟล์บางไฟล์ จากระบบหนึ่งไปยังระบบอื่น มันเป็นส่วนของคำสั่ง **uucp** และทำให้กระบวนการแลกเปลี่ยนไฟล์ง่ายขึ้นสำหรับผู้ส่งและผู้รับ คำสั่ง **uuto** จะส่งไฟล์ไปยังผู้ใช้ที่ระบุ และเก็บมันในไดเร็กทอรีของผู้ใช้โดยตรงภายใต้ พับลิกไดเร็กทอรี BNU ของระบบนั้น ในจะแจ้งผู้รับว่า ไฟล์มาถึงแล้ว ผู้รับจะใช้คำสัง **uupick** เพื่อจัดการกับไฟล์ใหม่

การส่งไฟล์โดยใช้คำสั่ง uuto:

เมื่อคุณใช้คำสั่ง **uuto** เพื่อส่งไฟล์ รวมไฟล์ที่จะถูกส่ง ระบบรีโมตปลายทาง และผู้ใช้ที่ปลายทาง

ตัวอย่างเช่น:

```
uuto /home/bin/file1 distant!joe
```

นี่จะส่ง `file1` จากไดเร็กทอรีโลคัล `/home/bin` ไปยังผู้ใช้ `joe` บนระบบรีโมต `distant`

คำสั่ง **uuto** จะรันภายใต้คำสั่ง **uucp** ไฟล์จะถูกถ่ายโอนไปยังระบบรีโมต ใน `/var/spool/uucppublic` ไฟล์จะถูกเก็บในไดเร็กทอรี `/var/spool/uucppublic/receive/`*user*`/System` บนระบบรีโมต ถ้าไดเร็กทอรีเป้าหมายไม่มีอยู่ มันจะถูกสร้างระหว่างการถ่ายโอนไฟล์

คำสั่ง BNU **rmail** จะแจ้งผู้รับว่าไฟล์มาถึงแล้ว

> **หมายเหตุ:** เพื่อส่งไฟล์ไปยังผู้ใช้บนระบบ *local* ใส่คำสั่ง **uuto** และรวมไฟล์ที่จะส่ง ระบบโลคัลปลายทาง และผู้ใช้ที่ระบบโลคัลปลายทาง ตัวอย่างเช่น:
>
> ```
> uuto /home/bin/file2 near!nick
> ```
>
> นี่จะส่ง `file2` จากไดเร็กทอรีโลคัล `/home/bin` ไปยังผู้ใช้ `nick` บนระบบโลคัล `near`

การรับไฟล์

เพื่อที่จะรับและจัดการกับไฟล์ ระบบที่ส่งและรับต้องรัน Basic Networking Utilities (BNU) หรือบางเวอร์ชันของ UNIX-to-UNIX Copy Program (UUCP)

ใช้คำสั่ง **uupick** เพื่อรับและจัดการกับไฟล์ที่ส่งด้วยคำสั่ง **uuto** มันมีอ็อพชัน file-handling ที่ยอมให้ผู้รับหาไฟล์ที่ส่ง ย้ายไฟล์ไปยังไดเร็กทอรีที่ระบุ ประมวลผลคำสั่ง หรือลบไฟล์

การรับไฟล์ด้วยคำสั่ง uupick:

ใช้คำสั่ง **uupick** เพื่อรับไฟล์

ตัวอย่างเช่น:

```
uupick
```

คำสั่ง **uupick** จะค้นหาไดเร็กทอรีสำหรับไฟล์ที่มี ID ผู้ใช้แบบรีโมต ในชื่อของพาธ จากนั้นคำสั่ง **uupick** จะแสดงข้อความต่อไปนี้ บนหน้าจอของระบบรีโมต:

```
from system base: file file1?
```

คำสั่งย่อย ? (เครื่องหมายคำถาม) บนบรรทัดที่สองของหน้าจอการแจ้งเตือนจะพร้อมต์ผู้รับ ให้ใช้อ็อพชัน **uupick** สำหรับจัดการกับไฟล์ในพับลิกไดเร็กทอรีของ BNU

สำหรับลิสต์ของอ็อพชันที่มี พิมพ์เครื่องหมายดอกจัน (*) บนบรรทัดใต้เครื่องหมายคำถาม (?) อ็อพชัน display, save และ quit คือ:

ไอเท็ม	คำอธิบาย
p	แสดงเนื้อหาของไฟล์
m *[Directory]*	บันทึกไฟล์ในไดเร็กทอรีที่ถูกระบุดยตัวแปร *[Directory]* ถ้าไม่มีปลายทางทีถูกระบุในอ็อพชัน **m** ไฟล์จะถูกย้ายไปยังไดเร็กทอรีการทำงานปัจจุบัน
q	Quits (ออกจาก) จากกระบวนการ **uupick** file-handling

การเข้ารหัสและถอดรหัสไฟล์สำหรับถ่ายโอน

ใช้คำสั่ง **uuencode** และ **uudecode** เพื่อเตรียมไฟล์สำหรับการส่งโดยโมเด็ม

คำสั่งทำงานเรียงตามลำดับ คำสั่ง **uuencode** จะแปลงไบนารีไฟล์เป็นไฟล์ ASCII ไฟล์เหล่านี้สามารถถูกส่งโดยเครื่องมืออีเมลไปยังระบบรีโมต

โดยใช้คำสั่ง **uudecode** ผู้ใช้ที่รับจะแปลงไฟล์ที่ถูกเข้ารหัสแบบ ASCII กลับเป็นรูปแบบไบนารี

คำสั่งและรายงานสถานะการแลกเปลี่ยนไฟล์

คุณสามารถดูรายงานสถานะการแลกเปลี่ยนไฟล์โดยใช้คำสั่ง **uusnap**, **uuq** และ **uustat**

การแสดงสถานะของระบบที่ถูกเชื่อมต่อโดย BNU

คำสั่ง **uusnap** จะแสดงตารางของข้อมูลเกี่ยวกับระบบทั้งหมดที่ถูกเชื่อมต่อโดย BNU

ตารางจะแสดงบรรทัดสำหรับแต่ละระบบ รายงานชื่อ และจำนวนของไฟล์คำสั่ง ไฟล์ข้อมูล และการประมวลผลคำสั่งรีโมตที่เก็บอยู่ในคิวของระบบ ไอเท็มสุดท้ายบนแต่ละบรรทัดคือข้อความสถานะ ข้อความนี้จะบอกว่าการเชื่อมต่อของ BNU สำเร็จหรือไม่ หรืออธิบายว่าทำไม BNU ไม่สามารถสร้างลิงก์ได้

อ้างถึง คำสั่ง **uusnap**

การแสดงคิวงานของ BNU

คำสั่ง **uuq** จะลิสต์ entry ใดๆในคิวงานของ BNU

รูปแบบของลิสต์จะเหมือนกับรูปแบบที่ถูกแสดงโดยคำสั่ง **ls** การแสดงแต่ละ entry จะรวมหมายเลขงาน ติดกันบนบรรทัดเดียวโดยสรุป รวมถึงชื่อระบบ และจำนวนของงานสำหรับระบบ และจำนวนทั้งหมดของไบต์ที่ส่ง ผู้ใช้ที่มีสิทธิของ root สามารถใช้คำสั่ง **uuq** เพื่อระบุงานที่ถูกเข้าคิวโดยหมายเลขงาน

อ้างถึงคำสั่ง **uuq** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 5*

สถานะการทำงานของ BNU

คำสั่ง **uustat** จะแสดงสถานะของคำสั่งนั้นๆ หรือการแลกเปลี่ยนไฟล์ในระบบ BNU

เมื่อถูกป้อนโดยไม่มีอ็อพชันแฟล็ก คำสั่ง **uustat** จะแสดงแต่ละงานที่ถูกร้องขอโดยผู้ใช้ปัจจุบันเป็นบรรทัดเดียว รวมถึง :

- หมายเลข ID ของงาน
- วันและเวลา
- สถานะ (ส่งหรือรับ)
- ชื่อระบบ
- ID ผู้ใช้ของบุคคลที่ใช้คำสั่ง
- ขนาดและชื่อของไฟล์งาน

เมื่อใช้กับหลายๆ แฟล็ก คำสั่ง **uustat** สามารถรายงานงานทั้งหมด โดยผู้ใช้ทั้งหมด ในคิว หรือบนงานที่ถูกร้องขอโดยระบบอื่นบนเน็ตเวิร์ก

คำสั่ง **uustat** จะให้ผู้ใช้สามารถควบคุมคิวของงานแบบจำกัด เพื่อรันบนรีโมตคอมพิวเตอร์ คุณสามารถตรวจสอบสถานะการเชื่อมต่อของ BNU ไปยังระบบอื่น ติดตามไฟล์ และการแลกเปลี่ยนคำสั่ง ตัวอย่างเช่น คุณสามารถคัดลอกคำร้องขอที่ถูกเริ่มโดยคำสั่ง **uucp**

อ้างถึง คำสั่ง**uustat**

การแลกเปลี่ยนคำสั่งระหว่างระบบโลคัลและรับบรีโมต

Basic Networking Utilities (BNU) ทำให้ผู้ใช้สามารถแลกเปลี่ยนคำสั่งระหว่างระบบโลคัลและระบบรีโมต

คำสั่ง **uux** จะรันคำสั่งบนระบบรีโมต คำสั่ง **uupoll** จะควบคุมจังหวะสำหรับการประมวลผลคำสั่ง

คำร้องขอการประมวลผลคำสั่งบนระบบรีโมต

ใช้คำสั่ง **uux** เพื่อร้องขอการประมวลผล ของคำสั่งบนระบบรีโมต

คำสั่ง **uux** จะไม่ประมวลผลคำสั่ง บนระบบรีโมต มันจะเตรียมการควบคุมที่จำเป็น และไฟล์ข้อมูลใน `/var/spool/uucp` แทน **uucico** daemon ถูกใช้เพื่อทำการถ่ายโอน หลังจากการถ่ายโอนเสร็จเรียบร้อย **uucico** ของระบบรีโมตจะสร้างไฟล์ที่เรียกทำงานได้ในสพูลไดเร็กทอรีของมัน

เมื่อ **uucico** daemons 2 ตัวตกลงที่จะวางหู **uuxt** daemon จะสแกนสพูลไดเร็กทอรีสำหรับคำร้องขอการประมวลผลที่ยังเหลืออยู่ ตรวจสอบการอนุญาต และตรวจสอบเพื่อดูว่ามีข้อมูลเพิ่มเติมที่ถูกต้องการอีกหรือไม่ จากนั้นจะแยกคำสั่งเพื่อทำ สิ่งที่ร้องขอ

> **หมายเหตุ:** คุณสามารถใช้คำสั่ง **uux** ในระบบใดๆ ที่ถูกกำหนดคอนฟิกเพื่อรันคำสั่งที่ถูกระบุ อย่างไรก็ตาม นโยบายของบางไซต์ อาจจำกัดการใช้คำสั่งบางอย่างสำหรับเหตุผลเกี่ยวกับความปลอดภัย ตัวอย่างเช่น บางไซต์อาจอนุญาตให้ประมวลผลคำสั่ง **mail**

หลังจากไฟล์ถูกได้รับบนระบบรีโมต **uuxqt** daemon จะรันคำสั่งที่ถูกระบุบนระบบนั้น **uuxqt** daemon จะสแกนพับลิกสพูลไดเร็กทอรีของระบบรีโมตเป็นระยะๆ เพื่อหาไฟล์ที่ได้รับในการส่งของ **uux uuxqt** daemon จะตรวจสอบว่าข้อมูลนั้นจะถูกเข้าถึงโดยไฟล์ที่ถูกส่งจะแสดง บนระบบรีโมต มันยังตรวจสอบว่าระบบที่ส่งมีการอนุญาตให้เข้าถึง ข้อมูล จากนั้น **uuxqt** daemon จะประมวลผลคำสั่ง หรือแจ้งระบบที่ส่งว่าคำสั่งไม่ได้รัน

การมอนิเตอร์การเชื่อมต่อ BNU แบบรีโมต

ใช้โพรซีเดอร์ต่อไปนี้สำหรับมอนิเตอร์การเชื่อมต่อ BNU แบบรีโมต

- โปรแกรม BNU ต้องถูกติดตั้งบนระบบของคุณ
- ลิงก์ (hardwired โมเด็ม หรือ TCP/IP) ต้องถูกตั้งค่าระหว่างระบบของคุณและระบบรีโมต
- ไฟล์คอนฟิกูเรชัน BNU รวมถึงไฟล์ `Systems` ไฟล์ `Permissions` ไฟล์ `Devices` และไฟล์ `Dialers` (และไฟล์ `Sysfiles` ถ้าใช้ได้) ต้องถูกตั้งค่าสำหรับการสื่อสารระหว่างระบบของคุณและระบบรีโมต

  หมายเหตุ: คุณต้องมีสิทธิของผู้ใช้ root เพื่อแก้ไขไฟล์คอนฟิกูเรชัน BNU

คำสั่ง **Uutry** สามารถช่วยคุณมอนิเตอร์กระบวนการ **uucico** daemon ถ้าผู้ใช้ที่ไซต์ของคุณรายงานปัญหาเกี่ยวกับการถ่ายโอนไฟล์

1. ใช้คำสั่ง **uustat** เพื่อระบุสถานะของงานการถ่ายโอนทั้งหมดในคิวปัจจุบันดังต่อไปนี้ :

   ```
   uustat -q
   ```

   ระบบจะแสดงรายงานสถานะเหมือนดังต่อไปนี้ :

   ```
   venus 3C (2) 05/09-11:02 CAN'T ACCESS DEVICE
   hera 1C 05/09-11:12 SUCCESSFUL
   merlin 2C 5/09-10:54 NO DEVICES AVAILABLE
   ```

   รายงานนี้จะระบุว่าไฟล์คำสั่ง `C.*`) 3 ไฟล์ที่ตั้งใจสำหรับระบบรีโมต `venus` อยู่ในคิวมาสองวัน อาจมีหลายสาเหตุสำหรับความล่าช้านี้ ตัวอย่างเช่น บางทีระบบ `venus` ถูกปิดการทำงานสำหรับการบำรุงรักษา หรือโมเด็มถูกปิด
2. ก่อนที่คุณจะเริ่มกิจกรรมการแก้ปัญหาอย่างจริงจัง ใช้คำสั่ง **Uutry** ดังต่อไปนี้เพื่อระบุว่าระบบโลคัลของคุณสามารถเชื่อมต่อกับระบบ `venus` ได้ในตอนนี้ :

   ```
   /usr/sbin/uucp/Uutry -r venus
   ```

   คำสั่งนี้จะสตาร์ท **uucico** daemon ด้วยจำนวนของการดีบักปานกลาง และบอกให้ทับเวลาการลองใหม่แบบดีฟอลต์ คำสั่ง **Uutry** จะส่งเอาต์พุตของการดีบักไปที่ไฟล์ชั่วคราว `/tmp/venus`
3. ถ้าระบบโลคัลของคุณทำการเชื่อมต่อกับระบบ `venus` สำเร็จ เอาต์พุตของการดีบักจะประกอบด้วยข้อมูลที่เป็นประโยชน์ อย่างไรก็ตาม บรรทัดสุดท้ายในสคริปต์นี้ เป็นส่วนที่สำคัญที่สุด :

   ```
   Conversation Complete: Status SUCCEEDED
   ```

   ถ้าการเชื่อมต่อสำเร็จ สันนิษฐานว่าเป็นปัญหาการถ่ายโอนไฟล์ชั่วคราวและถูกแก้ไขแล้ว ใช้คำสั่ง **uustat** อีกครั้งเพื่อความแน่นอนว่าไฟล์ในไดเร็กทอรี spooling ถูกถ่ายโอนไปยังระบบรีโมตสำเร็จ ถ้าไม่ใช้ขั้นตอนใน “การมอนิเตอร์การถ่ายโอนไฟล์ของ BNU” ในหน้า 493 เพื่อตรวจสอบสำหรับปัญหาของการถ่ายโอนไฟล์ ระหว่างระบบของคุณและระบบรีโมต
4. ถ้าระบบโลคัลของคุณไม่สามารถติดต่อกับระบบรีโมต เอาต์พุตของการดีบักที่ถูกสร้างโดยคำสั่ง **Uutry** จะประกอบด้วยชนิดของข้อมูลต่อไปนี้ (รูปแบบที่แท้จริงของเอาต์พุตอาจแตกต่างออกไป):

```
mchFind called (venus)
conn (venus)
getto ret -1
Call Failed: CAN'T ACCESS DEVICE
exit code 101
Conversation Complete: Status FAILED
```

ลำดับแรก ให้ตรวจสอบการเชื่อมต่อแบบฟิสิคัลระหว่างระบบโลคัลและระบบรีโมต ต้องแน่ใจว่ารีโมตคอมพิวเตอร์เปิดอยู่ และสายเคเบิลทั้งหมดถูกเชื่อมต่ออย่างถูกต้อง พอร์ตถูกเปิดใช้งานหรือปิดใช้งาน (ตามความเหมาะสม) บนทั้งสองระบบ และโมเด็มทำงานอยู่ (ถ้าใช้)

ถ้าการเชื่อมต่อแบบฟิสิคัลถูกต้องและแน่หนาดีแล้ว จากนั้นตรวจสอบว่าไฟล์คอนฟิกูเรชันที่เกี่ยวข้องบนทั้งระบบโลคัลและรีโมต มีสิ่งต่อไปนี้ :

- แน่ใจว่า entry ในไฟล์ `Devices`, `Systems` และ `Permissions` (และไฟล์ `Sysfiles` ถ้าใช้) ในไดเร็กทอรี **/etc/uucp** ถูกต้องบนทั้งสองระบบ
- ถ้าคุณใช้โมเด็ม ต้องแน่ใจว่าไฟล์ `/etc/uucp/Dialers` (หรือไฟล์อื่นที่ระบุใน **/etc/uucp/Sysfiles**) ประกอบด้วย entry ที่ถูกต้อง ถ้าคุณใช้ dial-code ตัวย่อ ต้องแน่ใจว่าตัวย่อถูกกำหนดในไฟล์ `/etc/uucp/Dialcodes`
- ถ้าคุณใช้การเชื่อมต่อ TCP/IP ต้องแน่ใจว่า **uucpd** daemon สามารถรันบนระบบรีโมต และไฟล์คอนฟิกูเรชันประกอบด้วย entry ของ TCP ที่ถูกต้อง

5. หลังจากคุณตรวจสอบการเชื่อมต่อแบบฟิสิคัล และไฟล์คอนฟิกูเรชัน ใช้คำสั่ง **Uutry** อีกครั้ง ถ้าเอาต์พุตของการดีบักยังคงรายงานว่าการเชื่อมต่อล้มเหลว คุณอาจต้องปรึกษากับสมาชิกของทีมสนับสนุนของระบบของคุณ บันทึกเอาต์พุตของการดีบักที่สร้างโดยคำสั่ง **Uutry** นี่อาจมีประโยชน์ในการวินิจฉัยปัญหา

การถ่ายโอนไฟล์ไปยังระบบรีโมตสำหรับพิมพ์

ใช้คำสั่ง **uux** เพื่อถ่ายโอนไฟล์ไปยังระบบรีโมตสำหรับพิมพ์

เพื่อถ่ายโอนไฟลไปยังระบบรีโมตสำหรับพิมพ์ สิ่งที่ต้องการก่อนต่อไปนี้ต้องมีอยู่แล้ว :

- การเชื่อมต่อกับ Basic Networking Utilities (BNU) ต้องถูกสร้างไปยังระบบรีโมต
- คุณต้องได้รับอนุญาตให้สามารถทำการประมวลผลบนระบบรีโมต

บนทรรทัดรับคำสั่งบนระบบโลคัล พิมพ์ :

```
uux remote!/usr/bin/lpr local!filename
```

นี่จะพิมพ์ไฟล์โลคัล *filename* บนระบบรีโมต

การมอนิเตอร์การถ่ายโอนไฟล์ของ BNU:

ใช้โพรซีเดอร์นี้เพื่อมอนิเตอร์การถ่ายโอนไฟล์ไปยังระบบรีโมต

- โปรแกรม BNU ต้องถูกติดตั้งและตั้งค่าบนระบบของคุณ
- สร้างการเชื่อมต่อกับระบบรีโมตโดยใช้ขั้นตอนที่ให้ใน “การมอนิเตอร์การเชื่อมต่อ BNU แบบรีโมต” ในหน้า 492

การมอนิเตอร์การถ่ายโอนไฟล์มีประโยชน์เมื่อการถ่ายโอนไฟล์ไปยังระบบรีโมตล้มเหลวโดยไม่ทราบสาเหตุ ข้อมูลการดีบักที่ถูกสร้างโดย **uucico** daemon (ถูกเรียกโดยคำสั่ง **Uutry**) สามารถช่วยคุณหาสาเหตุว่าอะไรไม่ถูกต้อง

คำสั่ง **Uutry** ให้คุณสามารถมอนิเตอร์การถ่ายโอนไฟล์ ดังต่อไปนี้ :

1. เตรียมไฟล์สำหรับถ่ายโอนโดยใช้คำสั่ง **uucp** พร้อมกับแฟล็ก **–r** โดยการใส่ :

   ```
   uucp -r test1 venus!~/test2
   ```

   แฟล็ก **–r** จะบอกโปรแกรม UUCP ให้สร้างและคิวไฟล์ที่ถ่ายโอนที่จำเป็น แต่*ไม่* สตาร์ท **uucico** daemon
2. ใช้คำสั่ง **Uutry** กับแฟล็ก **–r** เพื่อสตาร์ท **uucico** daemon พร้อมกับเปิดการดีบักโดยใส่ :

   ```
   /usr/sbin/uucp/Uutry -r venus
   ```

   นี่จะบอกให้ **uucico** daemon เพื่อติดต่อกับระบบรีโมต `venus` ทับเวลาการลองใหม่แบบดีฟอลต์ daemon จะติดต่อกับระบบ `venus` ล็อกอิน และถ่ายโอนไฟล์ ขณะที่คำสั่ง **Uutry** จะสร้างเอาต์พุตของการดีบักที่ให้คุณสามารถมอนิเตอร์กระบวนการ **uucico** กดลำดับคีย์ Interrupt เพื่อหยุดเอาต์พุตของการดีบักและกลับไปที่จุดรับคำสั่ง
   คำสั่ง**Uutry** ยังเก็บเอาต์พุตของการดีบักในไฟล์ `/tmp/`*`SystemName`* ถ้าคุณหยุดเอาต์พุตของการดีบักก่อนที่การเชื่อมต่อจะสำเร็จ คุณสามารถดูที่เอาต์พุตไฟล์เพื่อดูผลของการเชื่อมต่อ

การส่งงานที่ถูก Spool:

ใช้คำสั่ง **uupoll** เพื่อเริ่มการส่งงานที่ถูกเก็บในพับลิกสพูลลิ้งไดเร็กทอรีของระบบโลคัล

คำสั่ง **uupoll** จะสร้างงานแบบ null ในพับลิกไดเร็กทอรีสำหรับระบบรีโมต และเริ่มต้นคำสั่ง **uucico** นี่จะบังคับให้ **uucico** daemon ติดต่อระบบรีโมต และถ่ายโอนงานที่ถูกเข้าคิวทันที

ระบุระบบที่เข้ากันได้

ใช้คำสั่ง **uuname** เพื่อแสดงลิสต์ของระบบทั้งหมดที่สามารถเข้าถึงระบบโลคัล

ตัวอย่างเช่น เมื่อคุณพิมพ์ :

```
uuname
```

ที่บรรทัดรับคำสั่ง ระบบจะแสดงลิสต์ เช่น :

```
arthur
hera
merlin
zeus
```

ข้อมูลนี้ถูกใช้เพื่อระบุชื่อของระบบที่สามารถเข้าถึงได้ก่อนที่จะคัดลอกไฟล์ไปที่มัน คำสั่ง **uuname** ยังถูกใช้เพื่อสร้างเอกลักษณ์ของระบบโลคัล คำสั่ง**uuname** ได้ข้อมูลของมันโดยการอ่านไฟล์ `/etc/uucp/systems`

การสื่อสารกับระบบ UNIX ที่เชื่อมต่อ โดยใช้คำสั่ง tip

ใช้คำสั่ง **tip** เพื่อติดต่อระบบที่ถูกเชื่อมต่อใดๆที่รันระบบปฏิบัติการ UNIX

คำสั่ง **tip** ถูกติดตั้งกับ Basic Networking Utilities (BNU) และสามารถใช้การเชื่อมต่ออะซิงโครนัสเดียวกันกับที่ใช้โดย BNU

คำสั่ง **tip** ใช้ตัวแปรและสัญญาณ escape พร้อมกับแฟล็ก เพื่อควบคุมการทำงานของมัน แฟล็กสามารถถูกใส่ที่บรรทัดรับคำสั่ง สัญญาณ escape สามารถถูกใช้บนการเชื่อมต่อกับระบบรีโมตเพื่อเริ่มและหยุดการถ่ายโอนข้อมูล เปลี่ยนทิศทางของการถ่ายโอนข้อมูล และออกจากเชลล์ย่อย

ตัวแปรของคำสั่ง tip:

ตัวแปรของคำสั่ง **tip** จะกำหนดการตั้งค่า เช่น อักขระสิ้นสุดบรรทัด สัญญาณ break และโหมดของการถ่ายโอนไฟล์

การตั้งค่าตัวแปรสามารถถูกเริ่มต้นที่รันไทม์โดยใช้ไฟล์ .tiprc การตั้งค่าตัวแปรยังสามารถถูกเปลี่ยนระหว่างการประมวลผลโดยใช้สัญญาณ escape ~s บางตัวแปร เช่นอักขระสิ้นสุดบรรทัด สามารถถูกตั้งสำหรับแต่ละระบบใน entry ของระบบในไฟล์ remote

คำสั่ง **tip** จะอ่านไฟล์ 3 ไฟล์ คือ ไฟล์ phones ไฟล์ remote และไฟล์ .tiprc เพื่อระบุค่าเริ่มต้นสำหรับตัวแปรของมัน ไฟล์ .tiprc ต้องอยู่ในไดเร็กทอรีหลักของผู้ใช้เสมอ ชื่อและตำแหน่งของไฟล์ remote และ phones อาจแตกต่างกัน ชื่อของไฟล์ remote file และไฟล์ phones สามารถถูกกำหนดโดยตัวแปรสภาวะแวดล้อม :

ไอเท็ม	คำอธิบาย
PHONES	ระบุชื่อของผู้ใช้ไฟล์ phone ไฟล์สามารถมีชื่อไฟล์ที่ถูกต้องใดๆ และต้องถูกตั้งในรูปแบบของไฟล์ /usr/lib/phones-file ไฟล์ดีฟอลต์คือ etc/phones ถ้าไฟล์ถูกระบุพร้อมกับตัวแปร PHONES มันจะถูกใช้แทน (ไม่ใช่เพิ่มเติมจาก) ไฟล์ /etc/phones
REMOTE	ระบุชื่อของผู้ใช้ไฟล์คำจำกัดความของระบบรีโมต ไฟล์สามารถมีชื่อไฟล์ที่ถูกต้องใดๆ และต้องถูกตั้งในรูปแบบของไฟล์ /usr/lib/remote-file ไฟล์ดีฟอลต์ คือ /etc/remote ถ้าไฟล์ถูกระบุพร้อมกับตัวแปร **REMOTE** มันจะถูกใช้แทน (ไม่ใช่เพิ่มเติมจาก) ไฟล์ /etc/remote

เพื่อใช้ตัวแปรสภาวะแวดล้อม ตั้งค่ามันก่อนที่จะใช้คำสั่ง **tip** นอกจากนั้น ชื่อของไฟล์ phones และ remote สามารถถูกกำหนดโดยใช้คำสั่ง **tip** ตัวแปร phones และตัวแปร remote ตางลำดับ ในไฟล์ .tiprc

หมายเหตุ: คำสั่ง **tip** จะอ่านเฉพาะไฟล์ *last* remote หรือ phones ที่ถูกระบุ ดังนั้น ถ้าคุณระบุไฟล์ remote หรือ phones พร้อมกับตัวแปร ไฟล์ใหม่จะถูกใช้แทน (ไม่ใช่เพิ่มเติมจาก) ไฟล์ก่อนหน้านี้ที่คุณระบุ

คำสั่ง **tip** ใช้การตั้งค่าตัวแปรในลำดับต่อไปนี้ :

1. คำสั่งจะตรวจสอบการตั้งค่าของตัวแปรสภาวะแวดล้อม **PHONES** และ**REMOTE** สำหรับไฟล์ที่ใช้เป็นไฟล์ phones และ remote
2. คำสั่งจะอ่านไฟล์ .tiprc และตั้งตัวแปรทั้งหมดตาม ถ้าตัวแปร phones หรือ remote ถูกตั้งในไฟล์ .tiprc ค่าที่ตั้งนี้จะทับการตั้งค่าตัวแปรสภาวะแวดล้อม
3. เมื่อเริ่มต้นการเชื่อมต่อกับระบบรีโมต คำสั่งจะอ่าน entry ของไฟล์ remote จากระบบนั้น การตั้งค่าใน entry ของไฟล์ remote จะทับการตั้งค่าที่ถูกทำในไฟล์ .tiprc
4. ถ้าแฟล็ก - BaudRate ถูกใช้กับคำสั่ง **tip** อัตราที่ถูกระบุจะทับการตั้งค่าอัตรา baud ก่อนหน้านี้ทั้งหมด
5. การตั้งค่าที่ทำกับสัญญาณ escape ~s จะทับการตั้งค่าก่อนหน้านี้ของตัวแปร

   หมายเหตุ: ผู้ใช้ **tip** สามารถสร้างไฟล์ .tiprc และใช้ไฟล์นี้เพื่อระบุการตั้งค่าเริ่มต้นสำหรับตัวแปร **tip** ไฟล์ .tiprc ต้องอยู่ในไดเร็กทอรี $HOME ของผู้ใช้

ไฟล์คอนฟิกูเรชันคำสั่ง tip:

ก่อนที่คำสั่ง **tip** จะสามารถเชื่อมต่อกับระบบรีโมต ไฟล์ /etc/remote และ /etc/phones ต้องถูกเริ่มก่อน

ไอเท็ม	คำอธิบาย
/etc/remote	จะกำหนดแอ็ตทริบิวต์ของระบบรีโมต เช่น พอร์ต และชนิดของอุปกรณ์ที่จะถูกใช้เพื่อเข้าถึงระบบ พร้อมกับสัญญาณที่ใช้เพื่อระบุการเริ่มต้นและการสิ้นสุดของการส่งข้อมูล
/etc/phones	ลิสต์หมายเลขโทรศัพท์ที่ใช้เพื่อติดต่อระบบรีโมตผ่านสายโมเด็ม

ตัวอย่างไฟล์ remote และ phones จะมากับแพ็กเกจของ bos.net.uucp ตัวอย่างไฟล์ remote จะชื่อ /usr/lib/remote-file ตัวอย่างไฟล์ phones จะชื่อ /usr/lib/phones-file คัดลอก /usr/lib/remote-file ไปยัง /etc/remote และแก้ไข /etc/remote เพื่อเริ่มต้นหนึ่งในไฟล์เหล่านี้ คัดลอกตัวอย่างไฟล์ไปยังชื่อที่ถูกต้องและแก้ไขมันเพื่อให้เหมาะกับไซต์ของคุณ

ผู้ใช้ **tip** ยังสามารถสร้างไฟล์ remote และ phones ที่ถูกปรับแต่ง แต่ละไฟล์ remote ต้องอยู่ในรูปแบบของไฟล์ /usr/lib/remote-file และถูกระบุด้วยตัวแปร remote หรือตัวแปรสภาวะแวดล้อม REMOTE ไฟล์ phones เพิ่มเติมต้องอยู่ในรูปแบบของไฟล์ /usr/lib/phones-file และถูกระบุด้วยตัวแปร phones หรือตัวแปรสภาวะแวดล้อม PHONES ถ้าแต่ละไฟล์ phones หรือ remote ถูกระบุด้วยหนึ่งในตัวแปร ไฟล์นั้นจะถูกอ่านแทน (ไม่ใช่เพิ่มเติมกับ) ไฟล์ /etc/phones หรือ /etc/remote

ผู้ใช้ของ **tip** สามารถใช้แต่ละไฟล์ phones และ remote รวมกัน ตัวอย่างเช่น ผู้ใช้สามารถใช้ไฟล์ remote แบบดีฟอลต์ /etc/remote แต่ใช้แต่ละไฟล์ phones ที่ตั้งชื่อกับตัวแปร phones

การยกเลิกงานแบบรีโมต

ใช้คำสั่ง **uustat** เพื่อยกเลิกกระบวนการ BNU ที่ถูกทำกับระบบรีโมต

เมื่อต้องการยกเลิกงานแบบรีโมต สิ่งต้องการก่อนต่อไปนี้ต้องพร้อมแล้ว :

- การเชื่อมต่อ Basic Networking Utilities (BNU) ต้องถูกสร้างกับระบบรีโมตเป้าหมาย
- งานแบบรีโมตต้องถูกใช้จากระบบโลคัล

1. ระบุหมายเลข ID ของงานของกระบวนการ ถูกลิสต์ในคิวของรีโมต บนทรรทัดรับคำสั่งบนระบบโลคัล พิมพ์ :

   ```
   uustat -a
   ```

   อ็อพชัน **–a** จะแสดงงานทั้งหมดในคิวที่เก็บของระบบรีโมตและคำร้องขอของงานของผู้ใช้ BNU อื่นใดๆบนระบบ

   BNU จะตอบสนองด้วยข้อความคล้ายดังนี้ :

   ```
   heraC3113   11/06-17:47 S hera   you  289 D.venus471afd8
   merlinC3119 11/06-17:49 S merlin jane 338 D.venus471bc0a
   ```

2. จากนั้น พิมพ์:

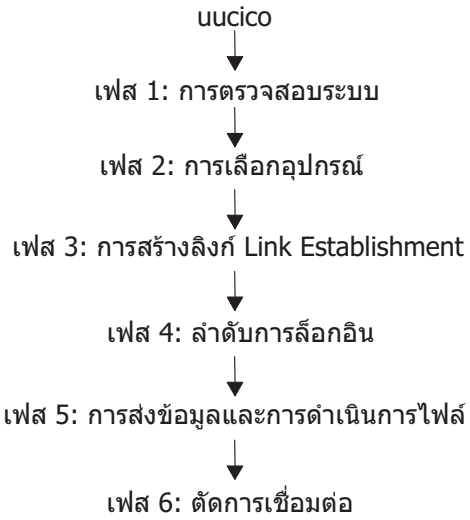
   ```
   uustat -k heraC3113
   ```

   อ็อพชัน **–k** จะยกเลิกคำร้องขอของงาน heraC3113

การแก้ปัญหา BNU

ข้อความแสดงข้อผิดพลาดของ BNU สามารถถูกลิงก์กับเฟสที่ระบุในโฟลว์การคุย กัน ใช้ ″ไดอะแกรมการคุยกันของ BNU″ และคำอธิบายข้อผิดพลาดต่อไปนี้เพื่อช่วยในการวินิจฉัยปัญหา BNU ของคุณ

บางข้อความต่อไปนี้อาจไม่ถูกส่งจาก BNU แต่ถูกรวมในกรณีที่เวอร์ชันของ UUCP อื่นถูกใช้งาน



รูปที่ 28. ไดอะแกรมการคุยกันของ BNU

นี่แสดงไฟลว์และเฟสต่างๆของการคุยกันของ BNU จาก uucico ที่ด้านบน ข้อมูลจะถูกผ่านไปยัง Phase 1–System Verification จากนั้น Phase 2–Device Selection และ Phase 3–Link Establishment จากนั้น Phase 4–Login Sequence ต่อไป Phase 5–Data Transfer และ File Execution สุดท้าย Phase 6–Disconnect

ข้อความสถานะของ BNU PHASE 1

จะมีข้อความสถานะของ BNU PHASE 1 ห้าข้อความ ที่อธิบายในตารางต่อไปนี้

ไอเท็ม	คำอธิบาย
`Assert Error`	หน่วยของระบบโลคัลมีปัญหา ตรวจสอบรายงานข้อผิดพลาดสำหรับสาเหตุที่เป็นไปได้โดยการใช้คำสั่ง `errpt -a | pg`
`System not in Systems`	ถ้าคุณใส่ชื่อระบบรีโมตที่ไม่มีในไฟล์ `Systems` ข้อีวามสถานะนี้จะถูกสร้าง BNU จะยกเลิก ใช้คำสั่ง **uuname** เพื่อตรวจสอบชื่อระบบอีกครั้ง
`Wrong time to call`	ไฟล์ `Systems` มีข้อจำกัดเกี่ยวกับเวลาที่ยอมให้โทรออก BNU จะพยายามจนกว่าเวลาจะถูกต้อง ตรวจสอบไฟล์ `Systems`
`Callback required`	เน็ตเวิร์กถูกจำกัดการใช้งานทั้งสำหรับความปลอดภัยหรือเพื่อความประหยัด และการเข้าถึงถูกปฏิเสธในเวลานี้
`Cannot call No Call`	ข้อผิดพลาดเหล่านี้หมายความว่าล่าสุด BNU พยายามเรียกไปยังระบบรีโมตและล้มเหลว มันจะไม่ลองใหม่ทันที มันยังสามารถมีสาเหตุจากไฟล์สถานะเก่าของระบบถูกเก็บไว้ ทำให้ **uucico** daemon พยายามลองใหม่

ข้อความสถานะของ BNU PHASE 2

จะมีข้อความสถานะของ BNU PHASE 2 สี่ข้อความ ที่อธิบายในตารางต่อไปนี้

ไอเท็ม	คำอธิบาย
Dialer Script Failed	สคริปต์ของไฟล์ Dialers ของคุณทำไม่สมบูรณ์
No Device Available Can't Access Device	โมเด็มหรือวานโทรศัพท์ที่โทรออกจากระบบของคุณไม่ว่าง ตรวจสอบข้อผิดพลาดใน entry ของอุปกรณ์ของไฟล์ Systems นอกจากนี้ ตรวจสอบไฟล์ Devices และ Dialers ต้องแน่ใจว่าอุปกรณ์แบบโลจิคัลมีอุปกรณ์แบบฟิสิคัลที่เชื่อมโยงกับมัน ไฟล์ /etc/uucp/Sysfiles อาจถูกระบบเป็นไฟล์ Systems, Devices หรือ Dialers อื่นที่ถูกตั้งค่าไม่ถูกต้อง อุปกรณ์ถูกใช้โดยโปรแกรมอื่นหรือไม่ ? ตรวจสอบไดเร็กทอรี /var/locks สำหรับการล็อกพอร์ต ถ้ามีล็อกไฟล์ (ตัวอย่างเช่น LCK..TTY0) ตรวจสอบเพื่อดูถ้ากระบวนการที่ถูกระบุโดยตัวเลขในล็อกไฟล์ยังแอ็คทีฟอยู่ ถ้าไม่ คุณสามารถลบมัน (ตัวอย่างเช่น rm /var/locks/LCK..TTY0) ยังต้องตรวจสอบการอนุญาตบนพอร์ต
Dial Failed Failed (call to system)	ข้อผิดพลาดเหล่านี้จะแสดงเมื่อระบบของคุณสามารถโทรไปยังระบบอื่น แต่ระบบนั้นไม่ตอบรับ มันอาจบอกเกี่ยวกับปัญหาในไฟล์ Devices ใส่คำสั่ง uucico -r1 -x6 -s *SystemName* อาจเป็นไปได้ว่า BNU ต้องการสตริงบางสตริงที่มันไม่ได้รับ ทำการเชื่อมต่อด้วยมือเพื่อหาว่าอะไรที่ต้องรวมเข้ากับ entry ของไฟล์ Systems เพื่อการร้องขอ อย่าลืมเกี่ยวกับ "จังหวะ" บางครั้งอาจต้องมีการหน่วงเวลาในสตริงของการหมุนโมเด็ม นี่อาจหมายความว่าพอร์ตไม่ว่าง คุณหมุนเบอร์ไม่ถูกต้อง หรือ BNU ไม่ได้เป็นเจ้าของพอร์ตแล้ว
OK Auto Dial	นี่เป็นข้อความที่เป็นข้อมูลเท่านั้น ไม่ได้แสดงถึงข้อผิดพลาด

ข้อความสถานะของ BNU PHASE 3

จะมีข้อความสถานะของ BNU PHASE 3 ห้าข้อความ ที่อธิบายในตารางต่อไปนี้

ไอเท็ม	คำอธิบาย
Handshake Failed (LCK)	อุปกรณ์ถูกใช้โดยผู้อื่น กระบวนการไม่สามารถสร้างไฟล์ LCK บางครั้งไฟล์ LCK ต้องถูกลบแบบแมนวลโดยผู้ดูแลระบบ หลังจากพยายามหลายครั้ง ให้ติดต่อผู้ดูแลระบบของคุณ ดูว่ามีกระบวนการอื่นควบคุมพอร์ตอยู่หรือไม่ (ตัวอย่างเช่น อินสแตนซ์อื่นของ **uucico** daemon)
Login Failed	การล็อกอินล้มเหลวเนื่องจากการเชื่อมต่อไม่ดี หรือเครื่องทำงานช้า
Timeout	ระบบรีโมตไม่ตอบสนองภายในช่วงเวลาที่กำหนด นี่สามารถระบุได้ถึงปัญหากับ chat สคริปต์
Succeeded (Call to System)	การโทรสำเร็จแล้ว
BNU (continued)	นี่เป็นข้อความที่เป็นข้อมูลเท่านั้น ไม่ได้แสดงถึงข้อผิดพลาด

ข้อความสถานะของ BNU PHASE 4

จะมีข้อความสถานะของ BNU PHASE 4 หกข้อความ ที่อธิบายในตารางต่อไปนี้

ไอเท็ม	คำอธิบาย
Startup Failed Remote reject after login	หลังจากล็อกอิน **uucico** daemon ถูกสตาร์ทบนระบบรีโมต ถ้ามีปัญหาในการเริ่มการพูดคุยระหว่างสองระบบ ข้อความเหล่านี้จะถูกสร้างขึ้น คุณยังอาจล็อกอินยังบัญชีผู้ใช้ BNU ที่ไม่ถูกต้อง หรือการเริ่มต้นการแฮนด์เช็กล้มเหลว
Wrong machine name	เครื่องถูกเรียกไม่ถูกต้อง หรือชื่อเครื่องถูกเปลี่ยน
Bad login/machine combination	การล็อกอินไปยังระบบรีโมตล้มเหลว ปัญหาอาจเป็นหมายเลขโทรศัพท์ที่ไม่ถูกต้อง ล็อกอินหรือรหัสผ่านไม่ถูกต้อง หรือมีข้อผิดพลาดใน chat สคริปต์
Remote has a LCK file for me	ทั้งสองระบบพยายามโทรหากันในเวลาเดียวกัน คำร้องขอโลคัลจะล้มเหลวชั่วคราว
OK Talking	นี่เป็นข้อความที่เป็นข้อมูลเท่านั้น ไม่ได้แสดงถึงข้อผิดพลาด

ไอเท็ม	คำอธิบาย
LOGIN: PASSWORD:	ถ้าพร้อมต์ล็อกอินหรือรหัสผ่านเป็นตัวพิมพ์ใหญ่ทั้งหมด โมเด็มอาจเข้าสู่โหมด echo (E1 บน Hayes compatibles) นี่ทำให้โมเด็มเอ็คโค่กลับ หรือส่ง RING ไปยังระบบของคุณเมื่อได้รับการโทรเข้ามา คำสั่ง **getty** ได้รับสตริงและเปลี่ยน `login:` หรือ `password:` เป็นตัวพิมพ์ใหญ่ทั้งหมด เปลี่ยนโหมด echo บนโมเด็มเป็น off (ใช้ `ATE0` สำหรับ Hayes compatibles) **หมายเหตุ:** โปรดจำว่า เมื่อทำการเปลี่ยน คุณควรใช้ `ATE1` ใน chat สคริปต์ของไฟล์ `Dialers` ของคุณ หรือคุณจะไม่ได้รับ `OK` ที่ต้องการกลับจากโมเด็ม ถ้าพอร์ตของรีโมตถูกตั้งสำหรับ `delay` or `getty -r` และ chat สคริปต์ต้องการคีย์อินพุต ดังนั้นพอร์ตที่ถูกตั้งสำหรับ `delay` จะต้องการการขึ้นบรรทัดใหม่หนึ่งหรือสองครั้ง ก่อนที่จะดำเนินการกับล็อกอิน ลองเริ่มต้น chat สคริปต์บนระบบที่ทำการโทรด้วยต่อไปนี้ : `"" \r\d\r\d\r\d\r in:--in: ...` เมื่อแปล chat สคริปต์จะอ่านได้ดังต่อไปนี้: ไม่ต้องการอะไร ส่งรีเทิร์น หน่วงเวลา รีเทิร์น หน่วงเวลา รีเทิร์น หน่วงเวลา รีเทิร์น

ข้อความสถานะของ BNU PHASE 5

จะมีข้อความสถานะของ BNU PHASE 5 ห้าข้อความ ที่อธิบายในตารางต่อไปนี้

ไอเท็ม	คำอธิบาย
`Alarm`	**uucico** daemon มีปัญหากับการเชื่อมต่อ อาจเป็นการเชื่อมต่อไม่ดี หรือ "xon/xoff" ถูกตั้งเป็น yes บนโมเด็ม
`Remote access to path/file denied copy (failed)`	ข้อความเหล่านี้ระบุว่าเป็นปัญหาเกี่ยวกับการอนุญาต ตรวจสอบไฟล์และพาธ permissions
`Bad read`	ระบบรีโมตไม่มีพื้นที่เหลือ โดยมากเป็นพื้นที่ของ spool หรือ **uucico** daemon ไม่สามารถอ่านหรือเขียนไปยังอุปกรณ์
`Conversation failed`	Carrier detect ของโมเด็มหายไป เป็นไปได้ว่าโมเด็มถูกปิด สายเคเบิลหลวมหรือหลุด หรือระบบรีโมตมีปัญหา หรือถูกปิดระบบ การตัดการเชื่อมต่อของโทรศัพท์อาจเป็นสาเหตุของปัญหา
`Requested Copy (succeeded)`	นี่เป็นข้อความที่เป็นข้อมูลเท่านั้น ไม่ได้แสดงถึงข้อผิดพลาด

ข้อความสถานะของ BNU PHASE 6

จะมีข้อความสถานะของ BNU PHASE 6 สองข้อความ ที่อธิบายในตารางต่อไปนี้

ไอเท็ม	คำอธิบาย
`OK (Conversation Complete)`	ระบบรีโมตสามารถปฏิเสธหรือวางหูคำร้องขอและสลับบทบาท (หมายความว่าระบบรีโมตทำงานที่ระบบโลคัลทำ) หลังจากที่ 2 **uucico** daemons ตกลงว่าไม่มีงานเหลือแล้ว มันจะวางหู
`Conversation succeeded`	นี่เป็นข้อความที่เป็นข้อมูลเท่านั้นและไม่ได้แสดงถึงข้อผิดพลาด

การดีบักการล็อกอิน BNU ที่ล้มเหลวโดยใช้ uucico daemon

ใช้ **uucico** daemon เพื่อแสดงการล็อกอิน BNU ที่ล้มเหลว

- BNU ต้องถูกติดตั้งบนระบบของคุณ
- ลิงก์ (hardwired โมเด็ม หรือ TCP/IP) ต้องถูกตั้งค่าระหว่างระบบของคุณและระบบรีโมต
- ไฟล์คอนฟิกูเรชัน BNU จะรวมไฟล์ `Sysfiles` (ถ้าใช้ได้) ไฟล์ `Systems` ไฟล์ `Permissions` ไฟล์ `Devices` และไฟล์ `Dialers` ต้องถูกตั้งค่าสำหรับการสื่อสารระหว่างระบบของคุณและระบบรีโมต

  หมายเหตุ: คุณต้องมีสิทธิของผู้ใช้ root เพื่อแก้ไขไฟล์คอนฟิกูเรชัน BNU
- คุณต้องมีสิทธิผู้ใช้ root เพื่อใช้ **uucico** daemon ในโหมดการดีบัก

1. เพื่อสร้างข้อมูลการดีบักเกี่ยวกับการเชื่อมต่อระบบโลคัลกับระบบรีโมต สตาร์ท **uucico** daemon ด้วยแฟล็ก **–x** ดังต่อไปนี้ :

   ```
   /usr/sbin/uucp/uucico -r 1 -s venus -x 9
   ```

   โดยที่ `-r 1` จะระบุตัวหลัก หรือโหมดผู้เรียก `-s venus` ชื่อของระบบรีโมตที่คุณต้องการเชื่อมต่อ และ `-x 9` ระดับของการดีบักที่สร้างข้อมูลการดีบักแบบละเอียดที่สุด
2. ถ้าลำดับของ entry expect–send ในไฟล์ `Systems` อยู่ในรูปแบบของ `/etc/uucp/Systems` คือ:

   ```
   venus Any venus 1200 - "" \n in:--in: uucp1 word:
    mirror
   ```

   uucico จะเชื่อมต่อระบบโลคัลกับระบบรีโมต `venus` เอาต์พุตของการดีบักจะเหมือนกับ :

   ```
   expect: ""
   got it
   sendthem (^J^M)
   expect (in:)^
   M^Jlogin:got it
   sendthem (uucp1^M)
   expect (word:)^
   M^JPassword:got it
   sendthem (mirror^M)
   imsg >^M^J^PShere^@Login Successful: System=venus
   ```

   โดยที่:

ไอเท็ม	คำอธิบาย
`expect: ""`	จะระบุว่าระบบโลคัลจะไม่รอข้อมูลใดๆจากระบบรีโมต
`got it`	ตอบรับว่าข้อความข้อความได้รับแล้ว
`sendthem (^J^M)`	ระบุว่าระบบโลคัลจะส่ง carriage return และการขึ้นบรรทัดใหม่ให้กับระบบรีโมต
`expect (in:)`	ระบุว่าระบบโลคัลต้องการพร้อมต์ล็อกอินของระบบรีโมต ซึ่งสิ้นสุดด้วยสตริงอักขระ `in:`
`^M^Jlogin:got it`	ยืนยันว่าระบบโลคัลได้รับพร้อมต์ล็อกอินของรีโมตแล้ว
`sendthem (uucp1^M)`	ระบุว่าระบบโลคัลจะส่งล็อกอิน ID `uucp1` ไปยังระบบรีโมต
`expect (word:)`	ระบุว่าระบบโลคัลต้องการได้รับพร้อมต์รหัสผ่านของระบบรีโมต ซึ่งสิ้นสุดด้วยสตริงอักขระ `word:`
`^M^JPassword:got it`	ยืนยันว่าระบบโลคัลได้รับพร้อมต์รหัสผ่านของรีโมตแล้ว
`sendthem (mirror^M)`	ระบุว่าระบบโลคัลจะส่งรหัสผ่านสำหรับ `uucp1` ไปยังระบบรีโมต
`imsg >^M^J^PShere^@Login Successful: System=venus`	ยืนยันว่าระบบโลคัลล็อกอินกับระบบรีโมต `venus` สำเร็จแล้ว

หมายเหตุ:

1. เอาต์พุตของการดีบัก expect–send ที่ถูกสร้างโดยคำสั่ง **uucico** สามารถมาจากข้อมูลในไฟล์ `/etc/uucp/Dialers` หรือจากข้อมูลในไฟล์ `/etc/uucp/Systems` ข้อมูลเกี่ยวกับการสื่อสารกับโมเด็มมาจากไฟล์ `Dialers` ขณะที่ข้อมูลเกี่ยวกับการสื่อสารกับระบบรีโมตมาจากไฟล์ `Systems` (โปรดสังเกตว่า `/etc/uucp/Systems` และ `/etc/uucp/Dialers` เป็นไฟล์คอนฟิกูเรชัน BNU แบบดีฟอลต์ ไฟล์อื่นที่ถูกระบุใน `/etc/uucp/Sysfiles` เพื่อทำงานเป็นบทบาทเดียวกัน)
2. เพื่อตั้งค่าเชื่อมต่อกับระบบรีโมต คุณต้องคุ้นเคยกับลำดับของการล็อกอินของระบบนั้น

SNMP สำหรับการจัดการกับเน็ตเวิร์ก

สิ่งอำนวยความสะดวกของ Network Management จัดเตรียมความเข้าใจในการจัดการกับระบบเน็ตเวิร์ก ผ่านการใช้ **Simple Network Management Protocol (SNMP)** ซึ่งเปิดใช้งานโฮสต์สำหรับเน็ตเวิร์กเพื่อแลกเปลี่ยนข้อมูลการจัดการ

SNMP คือโปรโตคอลการทำงานกับอินเตอร์เน็ตที่ออกแบบมาเพื่อใช้กับอินเตอร์เน็ตแบบอิง **TCP/IP**

เมื่อติดตั้งระบบปฏิบัติการ AIX ไว้ เวอร์ชันที่ไม่เข้ารหัสของ **SNMPv3** จะมีการติดตั้งไว้โดยดีฟอลต์ และเริ่มต้นในเวลาที่บูตระบบ หากคุณมี community แทร็ป และรายการ SMUX ของคุณเองที่ตั้งค่าไว้ในไฟล์ `/etc/snmpd.conf` ของคุณ คุณอาจต้องการถ่ายโอนสิ่งเหล่านี้ไปยังไฟล์ `/etc/snmpdv3.conf` ด้วยตนเอง สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการถ่ายโอน community โปรดดู “การย้ายจาก SNMPv1 ไปยัง SNMPv3” ในหน้า 511

คุณอาจต้องการศึกษาข้อมูลใน ภาพรวมของ SNMP สำหรับโปรแกรมเมอร์ใน *หลักการเขียนโปรแกรมการสื่อสาร*

การจัดการกับเน็ตเวิร์ก **SNMP** อ้างอิงตามโมเลไคลเอ็นต์/เซิร์ฟเวอร์ที่คุ้นเคย ที่ถูกใช้แบบกว้างขวางในแอ็พพลิเคชันเน็ตเวิร์กแบบอิง **TCP/IP** แต่ละโฮสต์ ที่ถูกจัดการจะรันกระบวนการที่เรียกว่า *เอเจนต์* เอเจนต์คือกระบวนการของเซิร์ฟเวอร์ ที่รักษาฐานข้อมูล Management Information Base (MIB) สำหรับโฮสต์โฮสต์ที่เกี่ยวข้องกับการตัดสินใจในเรื่องของการจัดการกับเน็ตเวิร์ก สามารถรันกระบวนการที่เรียกว่าตัวจัดการ *ตัวจัดการ* คือไคลเอ็นต์แอ็พพลิเคชัน ที่สร้างคำร้องขอสำหรับข้อมูล MIB และการประมวลผลการตอบกลับ นอกจากนี้ ตัวจัดการสามารถส่งคำร้องขอไปยังเอเจนต์เซิร์ฟเวอร์เพื่อแก้ไขข้อมูล MIB

SNMP ใน AIX จัดการกับส่วนสนับสนุนสำหรับ RFC ต่อไปนี้:

ไอเท็ม	คำอธิบาย
RFC 1155	Sturcture ฟีก Identification ของof Management Information for **TCP/IP**-based Internets
RFC 1157	A **Simple Network Management Protocol (SNMP)**
RFC 1213	Management Information Base for Network Management of **TCP/IP**-based internets: MIB-II
RFC 1227	**Simple Network Management Protocol (SNMP)** single multiplexer (SMUX) protocol and Management Information Base (MIB)
RFC 1229	ส่วนขยายของอินเตอร์เฟสทั่วไปสำหรับ Management Information Base (MIB)
RFC 1231	IEEE 802.5 token-ring Management Information Base (MIB)
RFC 1398	Definitions of Managed Objects for the Ethernet-like Interface Types
RFC 1512	FDDI Management Information Base
RFC 1514	Host Resources MIB
RFC 1592	**Simple Network Management Protocol**-Distributed Program Interface Version 2
RFC 1905	Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC 1907	Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC 2572	Message Processing and Dispatching for the **Simple Network Management Protocol (SNMP)**
RFC 2573	**SNMP** Applications
RFC 2574	User-based Security Model (USM) for version 3 of the **Simple Network Management Protocol (SNMPv3)**
RFC 2575	View-based Access Control Model (VACM) for the **Simple Network Management Protocol (SNMP)**

SNMPv3

ในเวอร์ชันก่อนหน้าของระบบปฏิบัติการ AIX, **SNMPv1** เป็น เวอร์ชันเดียวที่พร้อมใช้งานของ **SNMP** สำหรับ AIX **SNMPv3** ที่จัดให้ในระบบปฏิบัติการ AIX มีกรอบงานที่ทรงพลังและยืดหยุ่น สำหรับความปลอดภัยของข้อความและการควบคุมการเข้าถึง

ข้อมูลนี้ในส่วนนี้ใช้กับ **SNMPv3** เท่านั้น

ข้อความแสดงความปลอดภัยเกี่ยวข้องกับการจัดเตรียมสิ่งต่อไปนี้:

- การตรวจสอบ Data integrity เพื่อตรวจสอบให้แน่ใจว่า ข้อมูลไม่ได้ถูกเลือกในการเปลี่ยน
- การตรวจสอบข้อมูลต้นทางเพื่อตรวจสอบให้แน่ใจว่า คำร้องขอหรือการตอบกลับที่สร้างขึ้นจาก แหล่งที่มาที่มีการเรียกคืน
- การตรวจสอบข้อความเวลาและข้อมูลที่เป็นความลับเพื่อป้องกัน การแอบฟัง

สถาปัตยกรรม **SNMPv3** แนะนำ User-based Security Model (USM) สำหรับข้อความด้านความปลอดภัยและ View-based Access Control Model (VACM) สำหรับการควบคุมการเข้าถึง สถาปัตยกรรมนี้สนับสนุนการใช้ความปลอดภัยที่แตกต่างกันอย่างพร้อมเพียงกัน การควบคุมการเข้าถึง และโมเดลกระบวนการข้อความ ตัวอย่างเช่น ความปลอดภัยแบบอิง community สามารถใช้พร้อมกับ USM ได้ หากต้องการ

USM ใช้แนวคิดของผู้ใช้ที่มีพารามิเตอร์ความปลอดภัย (ระดับของความปลอดภัย การพิสูจน์ตัวตนและโปรโตคอลความเป็นส่วนตัว และคีย์) ถูกตั้งค่าไว้ทั้งที่เอเจนต์ และตัวจัดการ ข้อความที่ส่งโดยใช้ USM คือการป้องกันที่ดีกว่า ข้อความที่ส่งไปพร้อมกับความปลอดภัยแบบอิง community โดยที่รหัสผ่านถูกส่ง ด้วยความชัดเจนและแสดงอยู่ในการติดตาม ด้วย USM ข้อความที่แลกเปลี่ยนระหว่างตัวจัดการ และเอเจนต์ที่มีการตรวจสอบ data integrity และการพิสูจน์ตัวตนดั้งเดิม ข้อความหน่วงเวลาและข้อความที่แสดงอีกครั้ง (ใกล้กับสิ่งที่เกิดขึ้นเนื่องจากโปรโตคอลการส่งผ่านแบบ connectionless) ถูกปกป้องโดยการใช้ตัวบ่งชี้เวลา และ ID คำร้องขอ ข้อมูลที่เป็นความลับ หรือเข้ารหัสลับยังพร้อมใช้งาน โดยที่ได้รับอนุญาต เป็นผลิตภัณฑ์ที่สามารถติดตั้งแยกจากกันได้ เวอร์ชัน **SNMP** ที่เข้ารหัสแล้วสามารถพบได้บน AIX Expansion Pack

การใช้ VACM เกี่ยวข้องกับการเก็บรวบรวมข้อมูล (เรียกว่า มุมมอง) กลุ่มของผู้ใช้ของข้อมูล และเข้าถึงคำสั่งที่นิยามไว้ซึ่งดูกลุ่มของผู้ใช้โดยเฉพาะ สามารถใช้เพื่ออ่าน เขียน หรือรับในแทร็บ

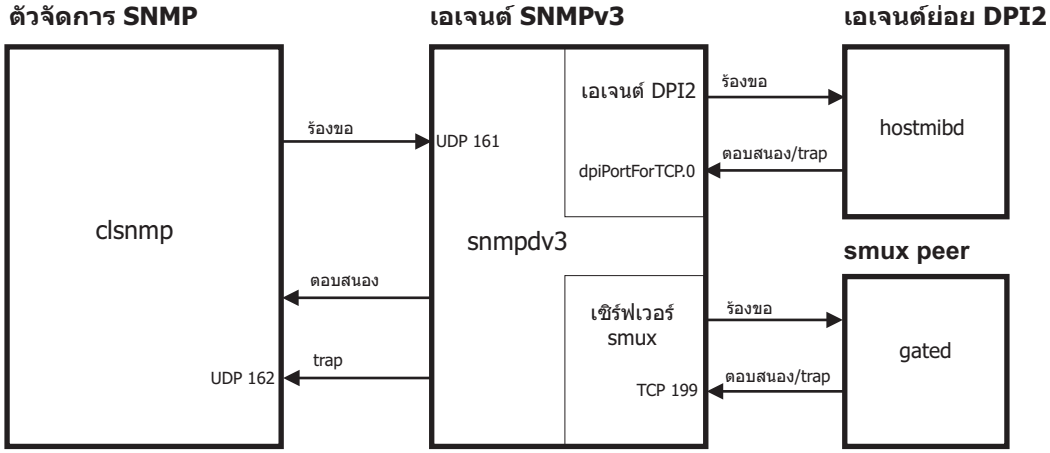
SNMPv3 ยังแนะนำความสามารถในการตั้งค่าเอเจนต์ **SNMP** แบบไดนามิกโดยใช้คำสั่ง **SNMP** SET กับอ็อบเจ็กต์ MIB ที่แสดงถึงคอนฟิกูเรชัน ของเอเจนต์ คอนฟิกูเรชันแบบไดนามิกนี้สนับสนุนการเปิดใช้งานการเพิ่ม การลบ และการแก้ไขรายการคอนฟิกูเรชันแบบโลคัลหรือแบบรีโมต

SNMPv3 เข้าถึงนโยบายและพารามิเตอร์ความปลอดภัยถูกระบุไว้ในไฟล์ `/etc/snmpdv3.conf` บนเอเจนต์ **SNMP** และไฟล์ `/etc/clsnmp.conf` บนตัวจัดการ **SNMP** สำหรับสถานการณ์จำลองเกี่ยวกับวิธีการตั้งค่าไฟล์เหล่านี้ โปรดดู “การสร้างผู้ใช้ใน SNMPv3” ในหน้า 515 คุณยังสามารถอ้างถึงรูปแบบไฟล์ `/etc/snmpdv3.conf` และ `/etc/clsnmp.conf` ใน *การอ้างอิงไฟล์*

สถาปัตยกรรมแบบ SNMPv3

มีสี่ส่วนหลักสำหรับสถาปัตยกรรมแบบ **SNMPv3**

การโต้ตอบระหว่างระบบเหล่านี้เพื่อจัดเตรียมข้อมูลที่จำเป็นซึ่งร้องขอ ถูกอธิบายถึงในรูปภาพประกอบต่อไปนี้:



รูปที่ 29. ส่วนหลักของสถาปัตยกรรม SNMPv3

รูปภาพประกอบนี้แสดงและเป็นตัวอย่างของสถาปัตยกรรม **SNMPv3** เอเจนต์ย่อย DPI2 เพียร์ smux ตัวจัดการ **SNMP** และเอเจนต์ **SNMP** ถูกแสดง นอกจากนี้ วิธีการสื่อสารกับแต่ละบุคคลถูกแสดง

เอเจนต์ SNMP:

เอเจนต์ **SNMP** รับคำร้องขอและสร้างการตอบกลับไปยังตัวจัดการ **SNMP**

นอกจากนี้ เอเจนต์ **SNMP** สื่อสารกับเอเจนต์ย่อย DPI2 และเพียร์ SMUX บนระบบ เอเจนต์ **SNMP** จัดการกับตัวแปร MIB บางตัวและเอเจนต์ย่อย DPI2 และการลงทะเบียนเพียร์ SMUX กับตัวแปร MIB พร้อมกับเอเจนต์ **SNMP**

เมื่อ **clsnmp** (ตัวจัดการ **SNMP**) ออกคำร้องขอ คำสั่งจะส่งไปยัง UDP 161 บนเอเจนต์ **SNMP** หากคำร้องขอคือคำร้องขอ **SNMPv1** หรือ **SNMPv2c** เอเจนต์ **SNMP** จะตรวจสอบชื่อ community และประมวลผลคำร้องขอ หากคำร้องคือคำร้องขอ **SNMPv3** เอเจนต์ **SNMP** จะพยายาม พิสูจน์ตัวตนกับการร้องขอข้อมูลและตรวจสอบว่า ผู้ใช้มีการเข้าถึงสิทธิ ที่จำเป็นต่อการเติมเต็มคำร้องขอโดยใช้คีย์การพิสูจน์ตัวตน และหากเวอร์ชันของการเข้ารหัสกำลังรันอยู่ นั่นคือคีย์ส่วนบุคคล หากเอเจนต์ **SNMP** ไม่สามารถพิสูจน์ตัวตนผู้ใช้ หรือหากผู้ใช้ไม่มีสิทธิในการเข้าถึงที่ถูกต้อง เพื่อเติมเต็มคำร้องขอ เอเจนต์ **SNMP** จะไม่ใช้ คำร้องขอนั้น สำหรับข้อมูลเกี่ยวกับการสร้างผู้ใช้ใน **SNMPv3** โปรดดู “การสร้างผู้ใช้ใน SNMPv3” ในหน้า 515

หากผู้ใช้พิสูจน์ตัวตนและมีสิทธิในการเข้าถึงที่ถูกต้อง เอเจนต์ **SNMP** จะเติมเต็มคำร้องขอ เอเจนต์ **SNMP** จะวางตัวแปร MIB ที่ถูกร้องขอ หากเอเจนต์ **SNMP** เองกำลังจัดการกับตัวแปร MIB ที่ร้องขอ ซึ่งจะประมวลผลคำร้องขอและส่งการตอบกลับไปยังตัวจัดการ **SNMP** หากเอเจนต์ย่อย DPI2 หรือเพียร์ SMUX กำลังจัดการกับตัวแปร MIB ที่ร้องขอ เอเจนต์ **SNMP** จะส่งต่อคำร้องขอไปยังเอเจนต์ย่อย DPI2 หรือเพียร์ SMUX ที่ตัวแปร MIB และถูกจัดการ ให้อนุญาตให้ประมวลผลคำร้องขอ และจากน้นจะตอบกลับไปยังตัวจัดการ **SNMP**

เอเจนต์ย่อย DPI2:

เอเจนต์ย่อย DPI2 เช่น **hostmibd**, สื่อสารกับเอเจนต์ DPI2 ใน **SNMPv3** เป็นส่วนของเอเจนต์ SNMP

เอเจนต์ย่อย DPI2 ส่งการตอบสนองและ traps ไปยังเอเจนต์ DPI2 ผ่าน dpiPortForTCP.0 เนื่องจากนี่ไม่ใช่ well-known พอร์ต ลำดับแรกเอเจนต์ย่อย DPI2 ต้องส่งคำร้องขอสำหรับหมายเลขพอร์ตสำหรับ dpiPortForTCP.0 คำร้องขอนี้จะถูกส่งไปยัง UDP 161 บน **SNMP** เอเจนต์ หลังจากนั้น **SNMP** เอเจนต์จะตอบสนองกับเอเจนต์ย่อย DPI2 ด้วยหมายเลขพอร์ตสำหรับ

dpiPortForTCP.0 หลังจากได้รับหมายเลขพอร์ต เอเจนต์ย่อย DPI2 จะสร้างการเชื่อมต่อกับเอเจนต์ DPI2 โดยใช้หมายเลขพอร์ตที่ได้ จากนั้น เอเจนต์ย่อย DPI2 จะลงทะเบียนทรีย่อยของ MIB ของมันกับเอเจนต์ DPI2

หมายเหตุ: เพื่อให้เอเจนต์ **SNMP** รับฟังบนพอร์ตอื่นที่ไม่ใช่ UDP 161 คุณต้องตั้งสภาวะแวดล้อม **SNMP_PORT** มี 2 วิธีในการตั้งตัวแปรนี้ :

- **วิธีที่ 1**: หยุดเอเจนต์ย่อย DPI2 และพิมพ์คำสั่งต่อไปนี้ :
  - `SNMP_PORT=<port_number> /usr/sbin/aixmibd -d 128`
  - `SNMP_PORT=<port_number> /usr/sbin/hostmibd -d 128`
  - `SNMP_PORT=<port_number> /usr/sbin/snmpmibd -d 128`

  โดยที่ *port_number* เป็นหมายเลขพอร์ตที่คุณต้องการใช้

  หลังจากที่คำสั่งถูกประมวลผล สตาร์ทเอเจนต์ DPI2
- **วิธีที่ 2**: รวมตัแปร **SNMP_PORT** ในไฟล์ `/etc/environment` และกำหนดค่าพอร์ตใหม่ให้มัน ยอมให้ **aixmibd, hostmibd, snmpmibd,** และ **snmpd** daemons รันจาก `/etc/rc.tcpip` ในวิธีนี้ คุณไม่จำเป็นต้องรันคำสั่ง **aixmibd, hostmibd** และ **snmpmibd** จากบรรทัดรับคำสั่ง

หลังจากที่การเชื่อมต่อถูกสร้างและทรีย่อยของ MIB ถูกลงทะเบียน เอเจนต์ย่อย DPI2 จะพร้อมที่จะตอบสนองกับคำร้องขอจากเอเจนต์ DPI2 เมื่อได้รับคำร้องขอ เอเจนต์ย่อย DPI2 จะประมวลผลคำร้องขอและตอบสนองด้วยข้อมูลที่จำเป็น

เอเจนต์ย่อย DPI2 ยังพร้อมที่จะส่ง traps ถ้าจำเป็น เมื่อ trap ถูกส่ง เอเจนต์ **SNMP** จะตรวจสอบไฟล์ `/etc/snmpdv3.conf` ของมันเพื่อระบุ IP แอดเดรสหรือแอดเดรสที่ trap ควรถูกฟอร์เวิร์ดไป และมันจะส่ง trap ไปยังแอดเดรสเหล่านั้น

SMUX peers:

SNMP Multiplexing (SMUX) peer เช่น **gated** เมื่อเริ่มต้นขึ้นจะสร้างการเชื่อมต่อกับ **TCP** 199 และจะเริ่มต้น การเชื่อมโยง SMUX

หลังจากการเริ่มต้น SMUX peer จะลงทะเบียนแผนผังย่อย MIB ที่จะจัดการ

หลังจากการลงทะเบียน SMUX peer พร้อมที่จะยอมรับคำร้องขอขาเข้าใดๆ จากเซิร์ฟเวอร์ SMUX และส่งการตอบกลับกลับไป เมื่อ SMUX peer ได้รับ คำร้องขอ peer จะประมวลผลคำร้องขอและส่งการตอบกลับกลับไปยังเซิร์ฟเวอร์ SMUX

SMUX peer ยังสามารถส่งการดักจับไปยังเซิร์ฟเวอร์ SMUX ด้วย หากมีการส่งการดักจับ เอเจนต์ **SNMP** จะตรวจสอบไฟล์ `/etc/snmpdv3.conf` เพื่อ กำหนด IP แอดเดรสซึ่งต้องส่งต่อการดักจับ และจะส่งการดักจับไปยังแอดเดรสเหล่านั้น

SNMP manager:

SNMP manager รัน **clsnmp** ซึ่ง เข้ากันได้กับ **SNMPv1**, **SNMPv2c**, และ **SNMPv3**

ใช้คำสั่ง **clsnmp** เพื่อออกใช้คำร้องขอ เช่น คำร้องขอ `get`, `get-next`, `get-bulk`, หรือ `set` คำร้องขอถูกส่งไปยัง UDP 161 บนเอเจนต์ **SNMP** ซึ่ง หลังจากนั้นจะรอการตอบกลับจากเอเจนต์ **SNMP**

หมายเหตุ: เมื่อต้องการให้ **SNMP** Manager สามารถใช้พอร์ตอื่นนอกเหนือจาก UDP 161 คุณต้อง ประกาศหมายเลขพอร์ตซึ่งคุณต้องการใช้และ IP แอดเดรสในฟิลด์ **targetAgent** ของไฟล์ `/etc/clsnmp.conf` สำหรับข้อมูลเพิ่มเติมเกี่ยวกับไฟล์ `/etc/clsnmp.conf` ให้ดูที่ ไฟล์ clsnmp.conf ใน *การอ้างอิงไฟล์*

และยังสามารถฟังการดักจับ **SNMP** บน UDP 162 ได้ด้วย **SNMP** manager จะได้รับการดักจับถ้ามีการระบุ IP แอดเดรสใน ไฟล์ /etc/snmpdv3.conf บนเอเจนต์ **SNMP**

ตัวแปร MIB:

ข้อมูลเกี่ยวกับตัวแปร MIB สามารถพบได้ในที่ตั้งต่อไปนี้

หากต้องการข้อมูลเกี่ยวกับตัวแปร MIB ให้ดูที่ ฐานข้อมูลการจัดการ, คำศัพท์เกี่ยวกับตัวแปรฐานข้อมูลการจัดการ, การทำงานกับ ตัวแปรฐานข้อมูลการจัดการ, และ ฐานข้อมูลของฐานข้อมูลการจัดการ ใน *หลักการเขียนโปรแกรมการสื่อสาร*

ถ้าคุณต้องการกำหนดคอนฟิกเอเจนต์ย่อย DPI2 หรือ smux peer ให้ดูที่ไดเร็กทอรี /usr/samples/snmpd/smux และ /usr/samples/snmpd/dpi2

คีย์การพิสูจน์ตัวตน SNMPv3

การพิสูจน์ตัวตนโดยทั่วไปจะถูกต้องการเพื่อให้คำร้องขอ **SNMPv3** ถูกประมวลผล (ยกเว้นระดับความปลอดภัยที่ร้องขอคือ noAuth)

เมื่อพิสูจน์ตัวตนคำร้องขอ เอเจนต์ **SNMP** จะตรวจสอบว่าคีย์การพิสูจน์ตัวตนที่ส่งในคำร้องขอ **SNMPv3** สามารถถูกใช้เพื่อสร้างข้อความแบบย่อที่ตรงกับข้อความแบบย่อที่ถูกสร้างจากคีย์การพิสูจน์ตัวตนที่ถูกกำหนดโดยผู้ใช้

เมื่อคำร้องขอถูกส่งจากตัวจัดการ **SNMP** คำสั่ง **clsnmp** จะใช้คีย์การพิสูจน์ตัวตนที่พบบน entry ในไฟล์ /etc/clsnmp.conf บนตัวจัดการ **SNMP** มันต้องสัมพันธ์กับคีย์การพิสูจน์ตัวตนที่ระบุบน entry ของ USM_USER สำหรับผู้ใช้นั้นใน **SNMP** เอเจนต์ ไฟล์ /etc/snmpdv3.conf คีย์การพิสูจน์ตัวตนจะถูกสร้างโดยใช้คำสั่ง **pwtokey**

คีย์การพิสูจน์ตัวตนถูกสร้างจากข้อมูล 2 ส่วน :

- รหัสผ่านที่ถูกระบุ
- identification ของ **SNMP** เอเจนต์ที่คีย์จะถูกใช้ ถ้าเอเจนต์เป็น IBM เอเจนต์ engineID ของมันจะถูกสร้างโดยใช้สูตร engineID ของผู้ขาย เอเจนต์อาจถูกระบุโดย IP แอดเดรสหรือชื่อโฮสต์ ไม่เช่นนั้น engineID ต้องถูกจัดเตรียมเป็น identification ของเอเจนต์

คีย์ที่รวม identification ของเอเจนต์ที่มันจะถูกใช้เรียกว่าคีย์แบบ localized มันสามารถใช้เฉพาะกับเอเจนต์นั้น คีย์ที่ไม่รวม engineID ของเอเจนต์ที่มันจะถูกใช้เรียกว่า non-localized

คีย์จะถูกเก็บในไฟล์คอนฟิกูเรชันของคำสั่ง **clsnmp** /etc/clsnmp.conf คาดหวังว่าจะเป็นคีย์แบบ non-localized คีย์ที่ถูกเก็บในไฟล์คอนฟิกูเรชันของ **SNMP** เอเจนต์ /etc/snmpdv3.conf สามารถเป็นทั้ง localized หรือ non-localized โดยการใช้คีย์แบบ localized จะถูกพิจารณาว่ามีความปลอดภัยมากกว่า

ทางเลือกอื่นในการเก็ยคีย์การพิสูจน์ตัวตนในไฟล์คอนฟิกูเรชัน คำสั่ง **clsnmp** ยอมให้เก็บรหัสผ่านของผู้ใช้ ถ้าคำสั่ง **clsnmp** ถูกตั้งค่าด้วยรหัสผ่าน โค้ดจะสร้างคีย์การพิสูจน์ตัวตน (และคีย์ที่เป็นส่วนตัวถ้าต้องการ และถ้าเวอร์ชันที่ถูกเข้ารหัสถูกติดตั้ง) สำหรับผู้ใช้ คีย์เหล่านี้ต้องสร้างค่าการพิสูจน์ตัวตนที่เหมือนกันเป็นคีย์ที่ถูกตั้งค่าสำหรับ USM_USER ในเอเจนต์ ไฟล์ /etc/snmpdv3.conf หรือถูกตั้งค่าแบบไดนามิกด้วยคำสั่ง **SNMP** SET อย่างไรก็ตาม การใช้รหัสผ่านในไฟล์คอนฟิกูเรชันของไคลเอ็นต์ถูกพิจารณาว่ามีความปลอดภัยน้อยที่ใช้คีย์ในไฟล์คอนฟิกูเรชัน

คีย์ความเป็นส่วนตัว SNMPv3

การเข้ารหัสมีอยู่เป็นผลิตภัณฑ์แยกต่างหากบน AIX Expansion Pack ซึ่งใช้กฎหมายเอ็กซ์พอร์ตได้ คีย์ที่ใช้สำหรับการเข้ารหัสมีการสร้างขึ้นโดยใช้ อัลกอริทึมเดียวกันกับที่ใช้สำหรับการพิสูจน์ตัวตน

อย่างไรก็ตาม ความยาวคีย์อาจแตกต่างกัน ตัวอย่างเช่น คีย์การพิสูจน์ตัวตน HMAC-SHA ยาว 20 ไบต์ แต่คีย์การเข้ารหัสที่โลคัลไลซ์ซึ่งใช้กับ HMAC-SHA ยาวเพียง 16 ไบต์เท่านั้น

เวอร์ชันที่เข้ารหัสมีการเรียกใช้โดยอัตโนมัติหลังจากการติดตั้ง เมื่อต้องการ สลับกลับไปยังเวอร์ชันที่ไม่ได้เข้ารหัส ให้ใช้คำสั่ง **snmpv3_ssw**

คีย์การสร้าง SNMPv3

AIX ใช้คำสั่ง **pwtokey** เพื่อสร้างการพิสูจน์ตัวตนและคีย์ส่วนบุคคล เมื่อเรียกใช้งานได้

คำสั่ง **pwtokey** เปิดใช้การแปลงของรหัสผ่าน ลงในการพิสูจน์ตัวตนแบบ non-localized และ localized และคีย์ส่วนบุคคล โพรซีเดอร์ **pwtokey** ใช้รหัสผ่านและตัวระบุเป็นเอเจนต์และสร้างการพิสูจน์ตัวตน และคีย์ส่วนตัว เนื่องจากโพรซีเดอร์ถูกใช้โดยคำสั่ง **pwtokey** มีอัลกอริธคมที่ใช้โดยคำสั่ง **clsnmp** บุคคลที่ตั้งค่าเอเจนต์ **SNMP** สามารถสร้างคีย์การพิสูจน์ตัวตนที่เหมาะสม (และความเป็นส่วนตัว) เพื่อวางลงในไฟล์ `/etc/clsnmp.conf` บนตัวจัดการ **SNMP** สำหรับผู้ใช้ กำหนดรหัสผ่านและ IP แอดเดรสที่เป้าหมายรันอยู่

หลังจากที่คุณได้สร้างคีย์การพิสูจน์ตัวตนแล้ว (และคีย์ส่วนบุคคลหากคุณกำลังรันเวอร์ชัน ที่เข้ารหัส) คุณจำเป็นต้องป้อนคีย์เหล่านี้ลงในไฟล์ `/etc/snmpdv3.conf` บนเอเจนต์ **SNMP** และในไฟล์ `/etc/clsnmp.conf` บนตัวจัดการ **SNMP**

ใน **SNMPv3** มีเก้าคอนฟิกูเรชันของผู้ใช้ที่เป็นไปได้ แต่ละคอนฟิกูเรชัน พร้อมกับตัวอย่าง ถูกกำหนดไว้ด้านล่าง คีย์เฉพาะเหล่านี้ ถูกสร้างโดยใช้ `defaultpassword` สำหรับรหัสผ่านและ `9.3.149.49` เป็น IP address คำสั่งต่อไปนี้ ถูกใช้:

```
pwtokey -u all -p all defaultpassword 9.3.149.49
```

การพิสูจน์ตัวตนและคีย์ส่วนบุคคลต่อไปนี้ถูกสร้างขึ้น:

```
Display of 16 byte HMAC-MD5 authKey:
  18a2c7b78f3df552367383eef9db2e9f

Display of 16 byte HMAC-MD5 localized authKey:
  a59fa9783c04bcbe00359fb1e181a4b4

Display of 16 byte HMAC-MD5 privKey:
  18a2c7b78f3df552367383eef9db2e9f

Display of 16 byte HMAC-MD5 localized privKey:
  a59fa9783c04bcbe00359fb1e181a4b4

Display of 20 byte HMAC-SHA authKey:
  754ebf6ab740556be9f0930b2a2256ca40e76ef9

Display of 20 byte HMAC-SHA localized authKey:
  cd988a098b4b627a0e8adc24b8f8cd02550463e3

Display of 20 byte HMAC-SHA privKey:
```

```
  754ebf6ab740556be9f0930b2a2256ca40e76ef9

Display of 16 byte HMAC-SHA localized privKey:
  cd988a098b4b627a0e8adc24b8f8cd02
```

รายการเหล่านี้จะปรากฏขึ้นในไฟล์ /etc/snmpdv3.conf คอนฟิกูเรชันเก้าแบบต่อไปนี้อาจเป็นไปได้:

- การพิสูจน์ตัวตนแบบ Localized และคีย์ส่วนบุคคลโดยใช้โปรโตคอล HMAC-MD5:

  ```
  USM_USER user1 - HMAC-MD5 a59fa9783c04bcbe00359fb1e181a4b4 DES a59fa9783c04bcbe00359fb1e181a4b4 L - -
  ```

- การพิสูจน์ตัวตนแบบ localized และคีย์ส่วนบุคคลโดยใช้โปรโตคอล HMAC-MD5:

  ```
  USM_USER user2 - HMAC-MD5 18a2c7b78f3df552367383eef9db2e9f DES 18a2c7b78f3df552367383eef9db2e9f N  - -
  ```

- คีย์การพิสูจน์ตัวตนแบบ Localized ใช้โปรโตคอล HMAC-MD5:

  ```
  USM_USER user3 - HMAC-MD5 a59fa9783c04bcbe00359fb1e181a4b4 - - L -
  ```

- คีย์การพิสูจน์ตัวตนแบบ Non-localized ใช้โปรโตคอล HMAC-MD5:

  ```
  USM_USER user4 - HMAC-MD5 18a2c7b78f3df552367383eef9db2e9f - - N -
  ```

- การพิสูจน์ตัวตนแบบ Localized และคีย์ส่วนบุคคลโดยใช้โปรโตคอล HMAC-SHA:

  ```
  USM_USER user5 - HMAC-SHA cd988a098b4b627a0e8adc24b8f8cd02550463e3 DES
  cd988a098b4b627a0e8adc24b8f8cd02 L -
  ```

- การพิสูจน์ตัวตนแบบ Non-localized และคีย์ส่วนบุคคลโดยใช้โปรโตคอล HMAC-SHA:

  ```
  USM_USER user6 - HMAC-SHA 754ebf6ab740556be9f0930b2a2256ca40e76ef9 DES
  754ebf6ab740556be9f0930b2a2256ca40e76ef9 N -
  ```

- คีย์การพิสูจน์ตัวตนแบบ Localized ใช้โปรโตคอล HMAC-SHA:

  ```
  USM_USER user7 - HMAC-SHA cd988a098b4b627a0e8adc24b8f8cd02550463e3 - - L -
  ```

- คีย์การพิสูจน์ตัวตนแบบ Non-localized ใช้โปรโตคอล HMAC-SHA:

  ```
  USM_USER user8 - HMAC-SHA 754ebf6ab740556be9f0930b2a2256ca40e76ef9 - - N -
  ```

- ไม่ใช่ทั้งการพิสูจน์ตัวตนและคีย์ส่วนบุคคลที่ใช้ (**SNMPv1**)

  ```
  USM_USER user9 - none - none - - -
  ```

การตั้งค่าผู้ใช้ใน **SNMPv3** ต้องการคอนฟิกูเรชันทั้งไฟล์ /etc/snmpdv3.conf และไฟล์ /etc/clsnmp.conf สำหรับสถานการณ์จำลองเกี่ยวกับการสร้างคีย์ผู้ใช้ และแก้ไขไฟล์คอนฟิกูเรชันที่จำเป็น โปรดดู “การสร้างผู้ใช้ใน SNMPv3” ในหน้า 515 นอกจากนี้ โปรดดูคำสั่ง **pwtokey** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4* และคำสั่ง **clsnmp** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1* และรูปแบบไฟล์สำหรับไฟล์ /etc/clsnmp.conf และไฟล์ /etc/snmpdv3.conf ใน *การอ้างอิงไฟล์* คุณยังสามารถอ้างถึงไฟล์คอนฟิกูเรชันตัวอย่าง snmpdv3.conf และไฟล์คอนฟิกูเรชัน clsnmp.conf ที่อยู่ในไดเร็กทอรี /usr/samples/snmpdv3

คีย์การอัพเดต SNMPv3

SNMPv3 นำเสนอความสามารถของคีย์ผู้ใช้ที่อัพเดตตามข้อมูล รหัสผ่านใหม่แบบไดนามิก

การดำเนินการนี้ทำโดยใช้คำสั่ง **pwchange** เพื่อสร้าง คีย์ผู้ใช้ใหม่ตามข้อมูลรหัสผ่านที่อัพเดต โดยใช้คำสั่ง **clsnmp** เพื่ออัพเดตคีย์ผู้ใช้แบบไดนามิกในไฟล์ /etc/snmpdv3.conf และการแก้ไขไฟล์ /etc/clsnmp.conf ด้วยคีย์ใหม่ ในระหว่างโปรเซสนี้ไม่เคยมีการสื่อสารรหัสผ่านใหม่ระหว่างเครื่องต่างๆ

สำหรับคำแนะนำแบบทีละขั้นตอนเกี่ยวกับการอัพเดตคีย์ผู้ใช้ ให้ดูที่ "การอัพเดตคีย์การอนุญาตและคีย์ความเป็นส่วนตัวแบบไดนามิกใน SNMPv3"นอกจากนี้ ให้อ้างอิงคำสั่ง **pwchange** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4* และ คำสั่ง **clsnmp** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1* และรูปแบบไฟล์ /etc/clsnmp.conf และรูปแบบไฟล์ /etc/snmpdv3.conf ใน *การอ้างอิงไฟล์*

การอัพเดตคีย์การอนุญาตและคีย์ความเป็นส่วนตัวแบบไดนามิกใน SNMPv3

สถานการณ์จำลองนี้แสดงวิธีการอัพเดตคีย์การอนุญาตแบบไดนามิก สำหรับผู้ใช้ใน **SNMPv3**

ในสถานการณ์จำลองนี้ ผู้ใช้ u4 จะอัพเดตคีย์การอนุญาต สำหรับผู้ใช้ u8 ทั้งผู้ใช้ u4 และ u8 มี คีย์การอนุญาตที่สร้างขึ้นจากข้อมูลรหัสผ่าน defaultpassword และ IP แอดเดรส 9.3.149.49 อยู่แล้ว และทุกอย่างทำงานได้

ในระหว่าง สถานการณ์จำลองนี้ คีย์ใหม่จะถูกสร้างขึ้นสำหรับผู้ใช้ u8 และไฟล์ /etc/snmpdv3.conf จะมีการอัพเดตแบบไดนามิก จากนั้น จะต้องแก้ไขคีย์การอนุญาตสำหรับผู้ใช้ u8 ใน ไฟล์ /etc/clsnmp.conf ของด้านผู้จัดการด้วยตนเอง เพื่อสะท้อนถึงคีย์ใหม่

จัดทำสำเนาสำรองของไฟล์ /etc/snmpdv3.conf บนเอเจนต์ **SNMP** และสำเนาสำรองของไฟล์ /etc/clsnmp.conf บนผู้จัดการ **SNMP** ก่อนคุณเริ่มต้นโพรซีเดอร์นี้

ข้างล่างนี้คือ ไฟล์ /etc/snmpdv3.conf ที่จะมีการอัพเดตแบบไดนามิก:

```
USM_USER u4 - HMAC-MD5 18a2c7b78f3df552367383eef9db2e9f - - N -
USM_USER u8 - HMAC-SHA 754ebf6ab740556be9f0930b2a2256ca40e76ef9 - - N -

VACM_GROUP group1 SNMPv1  public  -
VACM_GROUP group2 USM u4 -
VACM_GROUP group2 USM u8 -

VACM_VIEW defaultView        internet                    - included -

VACM_ACCESS  group1 - - noAuthNoPriv SNMPv1  defaultView - defaultView -
VACM_ACCESS  group2 - - noAuthNoPriv USM defaultView defaultView defaultView -
VACM_ACCESS  group2 - - AuthNoPriv USM defaultView defaultView defaultView -
VACM_ACCESS  group2 - - AuthPriv USM defaultView defaultView defaultView -

NOTIFY notify1 traptag trap -

TARGET_ADDRESS Target1 UDP 127.0.0.1       traptag trapparms1 - - -
TARGET_ADDRESS Target2 UDP 9.3.149.49       traptag trapparms2 - - -
TARGET_ADDRESS Target3 UDP 9.3.149.49       traptag trapparms3 - - -
TARGET_ADDRESS Target4 UDP 9.3.149.49       traptag trapparms4 - - -

TARGET_PARAMETERS trapparms1 SNMPv1  SNMPv1  public  noAuthNoPriv -
TARGET_PARAMETERS trapparms3 SNMPv2c  SNMPv2c  publicv2c  noAuthNoPriv -
TARGET_PARAMETERS trapparms4 SNMPv3  USM     u4 AuthNoPriv -
```

ข้างล่างนี้คือไฟล์ /etc/clsnmp.conf ที่จะมีการอัพเดตสำหรับผู้ใช้ u8:

```
testu4 9.3.149.49 snmpv3 u4 - - AuthNoPriv HMAC-MD5 18a2c7b78f3df552367383eef9db2e9f - -
testu8 9.3.149.49 snmpv3 u8 - - AuthNoPriv HMAC-SHA 754ebf6ab740556be9f0930b2a2256ca40e76ef9 - -
```

สิ่งที่ต้องพิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ได้ผ่านการทดสอบโดยใช้เวอร์ชันเฉพาะของ AIX ผลลัพธ์ที่คุณได้อาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชันและระดับ AIX ของคุณ

อัพเดตรหัสผ่านและคีย์การอนุญาตของคุณ

ชื่อชุมชน ในไฟล์ /etc/snmpd.conf กลายเป็นส่วนหนึ่งของรายการ VACM_GROUP ในไฟล์ /etc/snmpdv3.conf ต้องวางแต่ละชุมชนไว้ในกลุ่ม จากนั้น คุณจะมอบสิทธิการดูและการเข้าถึงที่ต้อง การให้กับกลุ่ม

1. บนด้านผู้จัดการ **SNMP** ให้รันคำสั่ง **pwchange** ในสถานการณ์จำลองนี้ เรารันคำสั่งต่อไปนี้:

   ```
   pwchange -u auth -p HMAC-SHA defaultpassword newpassword 9.3.149.49
   ```

   คำสั่งนี้ จะสร้างคีย์การอนุญาตใหม่

   - -u auth ระบุว่าจะสร้างเฉพาะคีย์การอนุญาต เท่านั้น ถ้าคุณกำลังอัพเดตคีย์ความเป็นส่วนตัวด้วย ให้ใช้ -u all
   - -p HMAC-SHA ระบุโปรโตคอลที่จะใช้ เพื่อสร้างคีย์การอนุญาต ถ้าคุณกำลังอัพเดตคีย์ความเป็นส่วนตัวด้วย ให้ใช้ -p all
   - *defaultpassword* คือรหัสผ่าน ที่ใช้เพื่อสร้างคีย์การอนุญาตล่าสุด (ตัวอย่างเช่น ถ้ามีการใช้ bluepen เพื่อสร้างคีย์การอนุญาตล่าสุด ควรจะใช้ bluepen ที่นี่ด้วย)
   - *newpassword* คือรหัสผ่านใหม่ ที่จะใช้เพื่อสร้างคีย์การอนุญาต โปรดเก็บรหัสผ่านนี้ไว้สำหรับ การอ้างอิงในอนาคต
   - 9.3.149.49 คือ IP แอดเดรสที่เอเจนต์ **SNMP** กำลังรัน

   คำสั่งนี้ทำให้เกิดเอาต์พุตต่อไปนี้:

   ```
   Dump of 40 byte HMAC-SHA authKey keyChange value:
     8173701d7c00913af002a3379d4b150a
     f9566f56a4dbde21dd778bb166a86249
     4aa3a477e3b96e7d
   ```

   คุณจะใช้คีย์การอนุญาตนี้ใน ขั้นตอนถัดไป

   หมายเหตุ: โปรดเก็บรหัสผ่านใหม่ที่คุณใช้ไว้ในสถานที่ปลอดภัย คุณจะต้องใช้ รหัสผ่านนั้นอีกครั้งเมื่อทำการเปลี่ยนแปลงในอนาคต

2. บนผู้จัดการ **SNMP** ผู้ใช้ u4 จะเปลี่ยนคีย์การอนุญาต ของผู้ใช้ u8 โดยการป้อนคำสั่งต่อไปนี้:

   ```
   clsnmp -h testu4 set usmUserAuthKeyChange.12.0.0.0.2.0.0.0.0.9.3.149.49.2.117.56
    \'8173701d7c00913af002a3379d4b150af9566f56a4dbde21dd778bb166a862494aa3a477e3b96e7d\'h
   ```

   - testu4 มีการใช้เนื่องจากถูกแม็พเข้ากับผู้ใช้ u4 ใน ไฟล์ /etc/clsnmp.conf
   - ID อินสแตนซ์ของ *usmUserAuthKeyChange* ประกอบด้วย ID เอ็นจินของเอเจนต์ SNMP ซึ่งการอัพเดตเกิดขึ้นและชื่อผู้ใช้ ที่เป็นเจ้าของคีย์การอนุญาตซึ่งกำลังอัพเดต โดยค่าเหล่านี้เป็นค่าฐานสิบ ID เอ็นจิน สามารถพบได้ในไฟล์ /etc/snmpd.boots (ไฟล์ /etc/snmpd.boots ประกอบด้วยตัวเลขสองสตริง ID เอ็นจินคือสตริงแรก ละเว้น สตริงที่สองของตัวเลข)

     จะต้องแปลง ID เอ็นจิน จากค่าฐานสิบหกเป็นค่าฐานสิบเพื่อที่จะใช้ที่นี่ ตัวเลขสองตัว ใน ID เอ็นจินฐานสิบหกแปลงเป็นค่าฐานสิบ หนึ่งตัวต่อหนึ่งค่า ตัวอย่างเช่น ID เอ็นจิน 000000020000000009039531 จะถูกอ่านเป็น 00 00 00 02 00 00 00 00 09 03 95 31 แต่ละจำนวนเหล่านี้ต้องถูกแปลงเป็นค่าทศนิยม ซึ่งทำให้เกิด 0.0.0.2.0.0.0.0.9.3.

149.49 (สำหรับตารางการแปลง ให้ดู ASCII ฐานสิบ ฐานสิบหก ฐานแปด และตารางการแปลงไบนารี). ตัวเลขแรกในสตริงคือจำนวนไบต์ในสตริงฐานสิบ ในกรณีนี้ จำนวนไบต์คือ 12 ส่งผลให้ได้ค่า 12.0.0.0.2.0.0.0.0.9.3.149.49

ตัวเลข ต่อไปนี้คือจำนวนไบต์ในชื่อผู้ใช้ ตามด้วยค่าฐานสิบ ของชื่อผู้ใช้เอง ในกรณีชื่อ ชื่อผู้ใช้คือ u8 เมื่อแปลงเป็นค่าฐานสิบ u8 กลายเป็น 117.56 เนื่องจากชื่อผู้ใช้ยาว 2 ไบต์ ค่าที่แสดงถึงชื่อผู้ใช้จึง กลายเป็น 2.117.56 เพิ่มไปยังส่วนท้ายของ ID เอ็นจินเลขฐานสิบ (สำหรับตารางการแปลง โปรดดู ASCII, ฐานสิบ, ฐานสิบหก, ฐานแปด, และตารางการแปลงไบนารี).

ใน กรณีนี้ ผลลัพธ์คือ 12.0.0.0.2.0.0.0.0.9.3.149.49.2.117.56

- ค่าถัดไปในคำสั่งคือคีย์การอนุญาตใหม่ที่สร้างขึ้น โดยใช้คำสั่ง **pwchange** ในขั้นตอนก่อนหน้านี้

  หมายเหตุ: หาก ผู้ใช้มีคีย์ความเป็นส่วนตัวที่ตั้งค่าคอนฟิกด้วย ต้องทำซ้ำโพรซีเดอร์นี้ เพื่ออัพเดตคีย์ความเป็นส่วนตัว เมื่ออัพเดตคีย์ความเป็นส่วนตัว ให้ใช้ค่า usmUserPrivKeyChange แทนค่า usmUserAuthKeyChange

  การใช้ usmUserOwnAuthKeyChange แทน usmUserAuthKeyChange จะช่วยให้ผู้ใช้สามารถเปลี่ยนคีย์ การอนุญาตของตนเองได้ ตัวอย่างเช่น ผู้ใช้ u4 สามารถเปลี่ยน คีย์การอนุญาตของตนโดยใช้ usmUserOwnAuthKeyChange

เอาต์พุตของคำสั่งเป็นดังนี้:

```
1.3.6.1.6.3.15.1.2.2.1.6.12.0.0.0.2.0.0.0.0.9.3.149.49.2.117.56 = '8173701d7c00913af002a3379
d4b150af9566f56a4dbde21dd778bb166a862494aa3a477e3b96e7d'h
```

หลังจาก คำสั่งนี้เสร็จสมบูรณ์ไฟล์ /etc/snmpdv3.conf จะมีการ อัพเดตโดยอัตโนมัติหลังผ่านไปห้านาทีบนด้านเอเจนต์ **SNMP** คุณยังสามารถหยุดและเริ่มต้น **SNMP** daemon เพื่ออัพเดตไฟล์ได้ด้วย รายการต่อไปนี้ของผู้ใช้ u8 จะมีการอัพเดตแบบไดนามิก ในไฟล์ /etc/snmpdv3.conf:

```
USM_USER u8 00000002000000009039531 HMAC-SHA 4be657b3ae92beee322ee5eaeef665b338caf2d9
None - L nonVolatile
```

3. บนด้านผู้จัดการ **SNMP** ให้รันคำสั่ง **pwtokey** เพื่อ สร้างคีย์การอนุญาตใหม่จากข้อมูลรหัสผ่านใหม่เพื่อวาง ในไฟล์ /etc/clsnmp.conf ในสถานการณ์จำลองนี้ เรารันคำสั่งต่อไปนี้:

   ```
   pwtokey -u auth -p HMAC-SHA  newpassword 9.3.149.49
   ```

   - -u auth ระบุว่าจะสร้างเฉพาะคีย์การอนุญาต เท่านั้น ถ้าคุณกำลังอัพเดตคีย์ความเป็นส่วนตัวด้วย ให้ใช้ -u all
   - -p HMAC-SHA ระบุโปรโตคอลที่จะใช้ในการสร้างคีย์การอนุญาต ถ้าคุณกำลังอัพเดตคีย์ความเป็นส่วนตัวด้วย ให้ใช้ -p all
   - รหัสผ่านที่ใช้ (ในกรณีนี้ *newpassword*) ต้องเหมือนกับรหัสผ่านที่ใช้เมื่อสร้างคีย์การอนุญาตใหม่ ด้วยคำสั่ง **pwchange**
   - IP แอดเดรสที่ใช้ (ในกรณีนี้ 9.3.149.49) ต้องเป็น IP แอดเดรสที่เอเจนต์กำลังรัน

   ผลลัพธ์ให้คีย์การอนุญาตที่โลคัลไลซ์และไม่ได้โลคัลไลซ์:

   ```
   Display of 20 byte HMAC-SHA authKey:
     79ce23370c820332a7f2c7840c3439d12826c10d

   Display of 20 byte HMAC-SHA localized authKey:
     b07086b278163a4b873aace53a1a9ca250913f91
   ```

4. เปิดไฟล์ /etc/clsnmp.conf ด้วยโปรแกรมแก้ไขข้อความที่ โปรดปรานของคุณ และวางคีย์การอนุญาตที่ไม่ได้โลคัลไลซ์ไว้ในบรรทัดของ ผู้ใช้ที่เป็นเจ้าของคีย์ซึ่งกำลังอัพเดตอยู่ ในสถานการณ์จำลองนี้ รายการเป็นดังนี้:

   ```
   testu8 9.3.149.49 snmpv3 u8 - - AuthNoPriv HMAC-SHA 79ce23370c820332a7f2c7840c3439d12826c10d - -
   ```

บันทึกและปิดไฟล์

5. ทดสอบการตั้งค่าคอนฟิกที่อัพเดตโดยการรันคำสั่งต่อไปนี้:

```
clsnmp -v -h testu8 walk mib
```

โดยที่ *mib* คือตัวแปร MIB ซึ่งผู้ใช้ u8 มี สิทธิการอ่าน ในกรณีนี้ ผู้ใช้ u8 มีสิทธิเข้าถึง internet

คำร้องขอ SNMPv3

คำสั่ง **clsnmp** ใช้เพื่อส่งคำร้องขอ **SNMP** ไปยัง เอเจนต์ **SNMP** บนโลคัลหรือรีโมตโฮสต์

คำร้องขอสามารถเป็นคำร้องขอ **SNMPv1**, **SNMPv2c**, หรือ **SNMPv3** เพื่อประมวลผลคำร้องขอ ต้องกำหนดคอนฟิกไฟล์ /etc/clsnmp.conf

คำสั่ง **clsnmp** สามารถออกใช้คำร้องขอ get, getnext, getbulk, set, walk, และ findname แต่ละคำร้องขอเหล่านี้มีการอธิบายโดยย่อ ข้างล่าง:

get ช่วยให้ผู้ใช้สามารถรวบรวมข้อมูลจากหนึ่งตัวแปร MIB

getnext ให้ตัวแปร MIB ถัดไปในแผนผังย่อย MIB

getbulk ให้ตัวแปร MIB ทั้งหมดจากหลายแผนผังย่อย MIB

set ช่วยให้ผู้ใช้สามารถตั้งค่าตัวแปร MIB

walk ให้ตัวแปร MIB ทั้งหมดจากหนึ่งแผนผังย่อย

findname
แม็พ OID เข้ากับชื่อตัวแปร

trap ช่วยให้ **clsnmp** สามารถฟังการดักจับบนพอร์ต 162

สำหรับข้อมูลรายละเอียดเกี่ยวกับการออกใช้คำร้องขอ **clsnmp** ให้ดูที่คำสั่ง **clsnmp** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1*

การย้ายจาก SNMPv1 ไปยัง SNMPv3

สถานการณ์จำลองนี้แสดงการย้ายแบบปกติจาก **SNMPv1** ไปยัง **SNMPv3**

ในระบบปฏิบัติการ AIX เอเจนต์ **SNMP** ดีฟอลต์ซึ่งกำลังรันที่เวลาบูตของระบบ คือเวอร์ชันที่ไม่ได้เข้ารหัสของ **SNMPv3** **SNMPv3** ใช้ไฟล์ /etc/snmpdv3.conf เป็นไฟล์คอนฟิกูเรชัน พารามิเตอร์ใดๆ ที่คุณกำหนดคอนฟิกไว้ในไฟล์ /etc/snmpd.conf ซึ่งใช้โดย **SNMPv1** ในเวอร์ชันก่อนหน้าของระบบปฏิบัติการ AIX จะต้อง มีการย้ายไปยังไฟล์ /etc/snmpdv3.conf ด้วยตนเอง

ใน สถานการณ์จำลองนี้ ชุมชนและการดักจับที่ตั้งค่าคอนฟิกไว้ในไฟล์ /etc/snmpd.conf จะถูกย้ายไปยังไฟล์ /etc/snmpdv3.conf ที่ตอนท้าย ของสถานการณ์จำลอง **SNMPv3** จะนำเสนอฟังก์ชันการทำงานเหมือนกับที่ **SNMPv1** นำเสนอ หากคุณไม่ได้ตั้งค่าคอนฟิกชุมชนหรือการดักจับ **SNMPv1** ของคุณเอง คุณไม่จำเป็นต้องทำโพรซีเดอร์นี้ให้สมบูรณ์

ไฟล์นี้ ไม่มีข้อมูลใดๆ เกี่ยวกับคุณลักษณะที่มีอยู่ใน **SNMPv3** สำหรับข้อมูลเกี่ยวกับการสร้างผู้ใช้โดยใช้คุณลักษณะ **SNMPv3** ที่ไม่มี อยู่ใน **SNMPv1** ให้ดูที่ “การสร้างผู้ใช้ใน SNMPv3” ในหน้า 515

ไฟล์ ต่อไปนี้เป็นตัวอย่างไฟล์ /etc/snmpd.conf ที่ กำลังจะถูกย้าย มีการตั้งค่าคอนฟิกชุมชนต่อไปนี้: daniel, vasu, และ david ต้องย้ายชุมชนเหล่านี้ด้วยตนเอง

```
logging         file=/usr/tmp/snmpd.log         enabled
logging         size=0                          level=0

community       daniel       0.0.0.0     0.0.0.0         readWrite  1.17.35
community       vasu         9.3.149.49  255.255.255.255 readOnly   10.3.5
community       david        9.53.150.67 255.255.255.255 readWrite  1.17.35

view 1.17.35    udp  icmp  snmp 1.3.6.1.2.1.25
view 10.3.5     system interfaces tcp icmp

trap            daniel          9.3.149.49      1.17.35   fe
trap            vasu            9.3.149.49      10.3.5   fe
trap            david           9.53.150.67     1.17.35   fe

smux            1.3.6.1.4.1.2.3.1.2.3.1.1         sampled_password  # sampled
```

เพื่อทำขั้นตอนในสถานการณ์จำลองนี้ให้สมบูรณ์โปรดอ้างอิงไฟล์ /etc/snmpd.conf ของคุณ เตรียมสำเนาของไฟล์ดังกล่าวไว้ให้พร้อมเมื่อคุณเริ่มต้นโพรซีเดอร์นี้

สิ่งที่ต้องพิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ได้ผ่านการทดสอบโดยใช้เวอร์ชันเฉพาะของ AIX ผลลัพธ์ที่คุณได้อาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชันและระดับ AIX ของคุณ

ขั้นตอนที่ 1. ย้ายข้อมูลชุมชน

ชื่อชุมชน ในไฟล์ /etc/snmpd.conf กลายเป็นส่วนหนึ่งของรายการ VACM_GROUP ในไฟล์ /etc/snmpdv3.conf ต้องวางแต่ละชุมชนไว้ในกลุ่ม จากนั้น คุณจะมอบสิทธิการดูและการเข้าถึงที่ต้องการให้กับ กลุ่ม

1. ด้วยสิทธิ root ให้เปิดไฟล์ /etc/snmpdv3.conf ด้วยเท็กซ์เอดิเตอร์ที่คุณชอบ ระบุตำแหน่งรายการ VACM_GROUP ในไฟล์
2. สร้างรายการ VACM_GROUP สำหรับแต่ละชุมชนซึ่งคุณ ต้องการย้าย หากหลายชุมชนจะแบ่งใช้มุมมองและสิทธิการเข้าถึงเดียวกัน คุณต้องสร้างกลุ่มเพียงหนึ่งกลุ่มเท่านั้นสำหรับชุมชนดังกล่าว ชื่อชุมชน ในไฟล์ /etc/snmpd.conf กลายเป็นค่า *securityName* ของรายการ VACM_GROUP ในสถานการณ์จำลองนี้ มีการเพิ่มรายการต่อไปนี้ สำหรับ vasu, daniel, และ david:

   ```
   #-------------------------------------------------------------------
   # VACM_GROUP entries
   #    Defines a security group (made up of users or communities)
   #    for the View-based Access Control Model (VACM).
   # Format is:
   #  groupName securityModel securityName storageType
   VACM_GROUP group2 SNMPv1  vasu -
   VACM_GROUP group3 SNMPv1  daniel -
   VACM_GROUP group3 SNMPv1  david -
   #-------------------------------------------------------------------
   ```

   - *groupName* สามารถเป็นค่าใดๆ ที่คุณเลือก ยกเว้น group1
   - *securityModel* ยังคงเป็น SNMPv1 เนื่อง จากเรากำลังจะย้ายชุมชน SNMPv1

- ในสถานการณ์จำลองนี้ daniel และ david แบ่งใช้ มุมมองและสิทธิการเข้าถึงเดียวกันในไฟล์ /etc/snmpd.conf ดังนั้น ทั้งสองจึงเป็นสมาชิกของ group3 ในไฟล์ /etc/snmpdv3.conf ชุมชน vasu ถูกวางไว้ในกลุ่มอื่น เนื่องจากสิทธิการดูและการเข้าถึง แตกต่างจากสิทธิของ david และ daniel

ขณะนี้ ชุมชนต่างๆ มีการวางไว้ในกลุ่ม

ขั้นตอนที่ 2. ย้ายข้อมูลมุมมอง

ข้อมูลมุมมอง ในไฟล์ /etc/snmpd.conf จะกลายเป็นรายการ COMMUNITY, VACM_VIEW, และ VACM_ACCESS ในไฟล์ /etc/snmpdv3.conf รายการเหล่านี้จะกำหนดสิทธิการดูและการเข้าถึงสำหรับแต่ละกลุ่ม

1. สร้างรายการ COMMUNITY สำหรับ daniel, vasu, และ david โดยรักษา IP แอดเดรสเดียวกันของ netAddr และ netMask ตาม ที่ระบุไว้ในไฟล์ /etc/snmpd.conf

```
#-------------------------------------------------------------------
# COMMUNITY
#    Defines a community for community-based security.
# Format is:
#  communityName securityName securityLevel netAddr netMask storageType
COMMUNITY public    public     noAuthNoPriv 0.0.0.0     0.0.0.0         -
COMMUNITY daniel    daniel     noAuthNoPriv 0.0.0.0     0.0.0.0         -
COMMUNITY vasu      vasu       noAuthNoPriv 9.3.149.49  255.255.255.255 -
COMMUNITY david     david      noAuthNoPriv 9.53.150.67 255.255.255.255 -
#-------------------------------------------------------------------
```

2. สร้างรายการ VACM_VIEW สำหรับอ็อบเจ็กต์ MIB หรือตัวแปรทุก รายการที่แต่ละกลุ่มมีสิทธิเข้าถึงได้ ตามข้อมูลไฟล์ /etc/snmpd.conf daniel และ david มี สิทธิเข้าถึง udp, icmp, snmp, และ 1.3.6.1.2.1.25 (แผนผังย่อยโฮสต์ตามที่กำหนดไว้ใน RFC 1514) และ vasu มีสิทธิเข้าถึง system, interfaces, tcp, และ icmp รายการมุมมองเหล่านี้ถูกย้ายไปยังไฟล์ /etc/snmpdv3.conf ดังนี้:

```
#-------------------------------------------------------------------
# VACM_VIEW entries
#    Defines a particular set of MIB data, called a view, for the
#    View-based Access Control Model.
# Format is:
#  viewName viewSubtree viewMask viewType storageType

VACM_VIEW group2View       system                    - included -
VACM_VIEW group2View       interfaces                - included -
VACM_VIEW group2View       tcp                       - included -
VACM_VIEW group2View       icmp                      - included -

VACM_VIEW group3View       udp                       - included -
VACM_VIEW group3View       icmp                      - included -
VACM_VIEW group3View       snmp                      - included -
VACM_VIEW group3View       1.3.6.1.2.1.25            - included -
#-------------------------------------------------------------------
```

3. กำหนดสิทธิการเข้าถึงให้กับตัวแปร MIB ที่กำหนดไว้ในรายการ VACM_VIEW โดยการเพิ่มรายการ VACM_ACCESS ในไฟล์ /etc/snmpd.conf ทั้ง daniel และ david มีสิทธิ readWrite ในตัวแปร MIB ของตน ในขณะที่ vasu มีสิทธิ readOnly

กำหนดสิทธิเหล่านี้โดยการเพิ่มรายการ VACM_ACCESS ในสถานการณ์จำลองนี้ เราให้ group2 (vasu) group2View สำหรับ readView แต่ให้ - สำหรับ writeView เนื่องจาก vasu มีสิทธิ readOnly ใน ไฟล์ /etc/snmpd.conf เราให้ group3 (daniel และ david) group3View สำหรับทั้ง readView และ writeView เนื่องจาก กลุ่มเหล่านั้นมีสิทธิ readWrite ใน /etc/snmpd.conf ให้ดูที่ตัวอย่างต่อไปนี้

```
#--------------------------------------------------------------------------------------------------------
# VACM_ACCESS entries
#    Identifies the access permitted to different security groups
#    for the View-based Access Control Model.
# Format is:
# groupName contextPrefix contextMatch securityLevel securityModel readView writeView notifyView storageType
VACM_ACCESS  group1 - - noAuthNoPriv SNMPv1  defaultView - defaultView -
VACM_ACCESS  group2 - - noAuthNoPriv SNMPv1  group2View - group2View -
VACM_ACCESS  group3 - - noAuthNoPriv SNMPv1  group3View group3View group3View -
#--------------------------------------------------------------------------------------------------------
```

ขั้นตอนที่ 3. ย้ายข้อมูลดักจับ

รายการดักจับ ในไฟล์ /etc/snmpd.conf จะกลายเป็นรายการ NOTIFY, TARGET_ADDRESS, และ TARGET_PARAMETERS ใน ไฟล์ /etc/snmpdv3.conf อย่างไรก็ตาม จะต้องย้ายเฉพาะ TARGET_ADDRESS และ TARGET_PARAMETERS เท่านั้น

1. IP แอดเดรสที่แสดงรายการในรายการดักจับในไฟล์ /etc/snmpd.conf กลายเป็นส่วนหนึ่งของรายการ TARGET_ADDRESS ในไฟล์ /etc/snmpdv3.conf บรรทัดนี้ระบุโฮสต์ที่จะส่งการดักจับไป คุณสามารถกำหนดรายการ targetParams ในสถานการณ์จำลองนี้ เราใช้ trapparms1, trapparms2, trapparms3, และ trapparms4 ซึ่งจะมีการ กำหนดในรายการ TARGET_PARAMETERS

   ```
   #-------------------------------------------------------------------------------
   # TARGET_ADDRESS
   #    Defines a management application's address and parameters
   #    to be used in sending  notifications.
   # Format is:
   #  targetAddrName tDomain tAddress tagList targetParams timeout retryCount storageType
   TARGET_ADDRESS Target1 UDP 127.0.0.1    traptag trapparms1 - - -
   TARGET_ADDRESS Target2 UDP 9.3.149.49   traptag trapparms2 - - -
   TARGET_ADDRESS Target3 UDP 9.3.149.49   traptag trapparms3 - - -
   TARGET_ADDRESS Target4 UDP 9.53.150.67  traptag trapparms4 - - -
   #-------------------------------------------------------------------------------
   ```

2. ชื่อชุมชนที่ระบุไว้ในรายการดักจับในไฟล์ /etc/snmpd.conf กลายเป็นส่วนหนึ่งของรายการ TARGET_PARAMETERS ใน ไฟล์ /etc/snmpdv3.conf ชื่อชุมชนต้องมีการแม็พเข้ากับรายการ TARGET_ADDRESS เฉพาะ โดยใช้ค่า targetParams ตัวอย่างเช่น ชุมชน daniel มีการ แม็พเข้ากับ trapparms2 ซึ่งภายใต้รายการ TARGET_ADDRESS จะแม็พเข้ากับ IP แอดเดรส 9.3.149.49 โดยดั้งเดิม ชุมชน daniel และ IP แอดเดรส 9.3.149.49 เป็นรายการ trap ในไฟล์ /etc/snmpd.conf ให้ดูที่ตัวอย่างต่อไปนี้:

   ```
   #-----------------------------------------------------------------------
   # TARGET_PARAMETERS
   #    Defines the message processing and security parameters
   #    to be used in sending notifications to a particular management target.
   # Format is:
   #  paramsName mpModel securityModel securityName securityLevel storageType
   TARGET_PARAMETERS trapparms1 SNMPv1  SNMPv1  public  noAuthNoPriv -
   TARGET_PARAMETERS trapparms2 SNMPv1  SNMPv1  daniel  noAuthNoPriv -
   TARGET_PARAMETERS trapparms3 SNMPv1  SNMPv1  vasu    noAuthNoPriv -
   TARGET_PARAMETERS trapparms4 SNMPv1  SNMPv1  david   noAuthNoPriv -
   #-----------------------------------------------------------------------
   ```

3. ข้อมูล trapmask ในไฟล์ /etc/snmpd.conf ไม่ได้ย้ายไปยังไฟล์ /etc/snmpdv3.conf

ขั้นตอนที่ 4. ย้ายข้อมูล smux

หากคุณมีข้อมูล smux ซึ่งคุณต้องการย้าย คุณสามารถคัดลอกบรรทัดเหล่านั้นไปยังไฟล์ใหม่ ได้โดยตรง ในสถานการณ์จำลองนี้ รายการ sampled smux มีการตั้งค่าคอนฟิกในไฟล์ /etc/snmpd.conf ต้องคัดลอก บรรทัดนั้นไปยังไฟล์ /etc/snmpdv3.conf

```
#--------------------------------------------------------------------------------------------
#       smux <client OIdentifier> <password> <address> <netmask>
smux            1.3.6.1.4.1.2.3.1.2.3.1.1         sampled_password  # sampled
#--------------------------------------------------------------------------------------------
```

ขั้นตอนที่ 5. หยุดและเริ่มต้น snmpd daemon

หลังจาก การย้ายไฟล์ /etc/snmpd.conf ไปยังไฟล์ /etc/snmpdv3.conf เสร็จสมบูรณ์แล้ว ให้หยุด แล้วเริ่มต้น **snmpd** daemon คุณ จะต้องหยุดและเริ่มต้น **snmpd** daemon ในทุกครั้ง ที่คุณทำการเปลี่ยนแปลงในไฟล์ /etc/snmpdv3.conf

1. พิมพ์คำสั่งต่อไปนี้เพื่อหยุด daemon:

   ```
   stopsrc -s snmpd
   ```

2. พิมพ์คำสั่งต่อไปนี้เพื่อเริ่มต้น daemon:

   ```
   startsrc -s snmpd
   ```

หมายเหตุ: การรีเฟรชเอเจนต์ **SNMPv3** เพียงอย่างเดียวจะใช้ไม่ได้ เหมือนใน **SNMPv1** หากคุณทำการเปลี่ยนแปลงในไฟล์ /etc/snmpdv3.conf คุณต้องหยุดและเริ่มต้น daemon ตามที่แนะนำข้างบน ฟังก์ชันการตั้งค่าคอนฟิกแบบไดนามิก ที่สนับสนุนใน **SNMPv3** จะไม่อนุญาตให้คุณรีเฟรช

การสร้างผู้ใช้ใน SNMPv3

สถานการณ์จำลองนี้แสดงวิธีการสร้างผู้ใช้ใน SNMPv3 ด้วยตนเองโดย การแก้ไขไฟล์ /etc/snmpdv3.conf และ /etc/clsnmp.conf

ผู้ใช้ u1 จะถูกสร้างขึ้นในสถานการณ์จำลองนี้ ผู้ใช้ u1 จะได้รับ มอบคีย์การอนุญาต แต่จะไม่ได้รับมอบคีย์ความเป็นส่วนตัว (ซึ่งมีอยู่ เฉพาะถ้าคุณได้ติดตั้งชุดไฟล์ snmp.crypto แล้วเท่านั้น) จะมีการใช้โปรโตคอล HMAC-MD5 เพื่อสร้างคีย์การอนุญาตของ u1 หลังจากตั้งค่าคอนฟิก u1 แล้ว u1 จะถูกวางเข้าในกลุ่มซึ่งจะมีการกำหนดสิทธิการดูและการเข้าถึง กลุ่มนั้นในเวลาต่อมา สุดท้าย จะมีการสร้างรายการดักจับ สำหรับ u1

แต่ละค่าที่ใช้ในไฟล์ /etc/snmpdv3.conf และ /etc/clsnmp.conf ต้องไม่เกินกว่า 32 ไบต์

สิ่งที่ต้องพิจารณา

- ข้อมูลในสถานการณ์จำลองวิธีการนี้ได้ผ่านการทดสอบโดยใช้เวอร์ชันเฉพาะของ AIX ผลลัพธ์ที่คุณได้อาจแตกต่างไปอย่างมาก ขึ้นอยู่กับเวอร์ชันและระดับ AIX ของคุณ

ขั้นตอนที่ 1. สร้างผู้ใช้

1. ตัดสินใจเลือกโปรโตคอลความปลอดภัยซึ่งคุณต้องการใช้ ระหว่าง HMAC-MD5 หรือ HMAC-SHA อย่างใดอย่างหนึ่ง ในสถานการณ์จำลองนี้ จะใช้ HMAC-MD5

2. สร้างคีย์การอนุญาตโดยใช้คำสั่ง pwtokey เอาต์พุตของคุณอาจมีลักษณะแตกต่างออกไป ทั้งนี้ขึ้นอยู่กับโปรโตคอลการอนุญาตที่ คุณใช้อยู่และคุณกำลังใช้คีย์ความเป็นส่วนตัวอยู่หรือไม่ คีย์เหล่านี้จะมีการใช้ในไฟล์ /etc/snmpdv3.conf และ /etc/clsnmp.conf คำสั่งที่ใช้สำหรับผู้ใช้ u1 มีดังนี้:

   ```
   pwtokey -p HMAC-MD5 -u auth anypassword 9.3.230.119
   ```

   IP แอดเดรสที่ระบุคือ IP แอดเดรสที่เอเจนต์กำลังรัน รหัสผ่าน สามารถเป็นค่าใดๆ ก็ได้ แต่ต้องแน่ใจว่าบันทึกรหัสผ่านไว้ในสถานที่ที่ปลอดภัย สำหรับการใช้งานในอนาคต เอาต์พุตควรจะมีลักษณะคล้ายกับตัวอย่างต่อไปนี้:

   ```
   Display of 16 byte HMAC-MD5 authKey:
     63960c12520dc8829d27f7fbaf5a0470

   Display of 16 byte HMAC-MD5 localized authKey:
     b3b6c6306d67e9c6f8e7e664a47ef9a0
   ```

3. ด้วยสิทธิ root ให้เปิดไฟล์ /etc/snmpdv3.conf ด้วยเท็กซ์เอดิเตอร์ที่คุณชอบ
4. สร้างผู้ใช้โดยการเพิ่มรายการ USM_USER ตามรูปแบบ ที่กำหนดในไฟล์ ค่า *authKey* จะเป็นคีย์การอนุญาตที่โลคัลไลซ์และถูกสร้างขึ้นโดยใช้คำสั่ง **pwtokey** รายการสำหรับผู้ใช้ u1 มีดังนี้:

   ```
   #-------------------------------------------------------------------------
   # USM_USER entries
   #    Defines a user for the User-based Security Model (USM).
   # Format is:
   #  userName engineID authProto authKey privProto privKey keyType storageType
   #
   USM_USER u1 - HMAC-MD5 b3b6c6306d67e9c6f8e7e664a47ef9a0 - - L -
   #-------------------------------------------------------------------------
   ```

   - *userName* คือชื่อของ ผู้ใช้ในกรณีนี้ ชื่อคือ u1
   - *authProto* ต้องเป็นโปรโตคอล ที่คุณใช้เมื่อคุณสร้างคีย์ ในกรณีนี้ โปรโตคอลคือ HMAC-MD5
   - *authKey* คือคีย์การอนุญาตที่โลคัลไลซ์ และถูกสร้างขึ้นโดยใช้คำสั่ง **pwtokey**
   - *privProto* และ *privkey* ไม่มี การระบุเนื่องจากเราจะไม่ใช้คีย์ความเป็นส่วนตัวในสถานการณ์จำลองนี้
   - *keyType* คือ L เนื่องจาก เรากำลังใช้คีย์การอนุญาตที่โลคัลไลซ์
5. บันทึกและปิดไฟล์ /etc/snmpdv3.conf
6. เปิดไฟล์ /etc/clsnmp.conf บน SNMP manager ด้วยโปรแกรมแก้ไขข้อความที่โปรดปราน
7. เพิ่มผู้ใช้ใหม่ตามรูปแบบที่กำหนดในไฟล์ รายการ สำหรับ u1 มีดังนี้:

   ```
   #-----------------------------------------------------------------------------------------------
   #
   # Format of entries:
   # winSnmpName targetAgent admin secName password context secLevel authProto authKey privProto privKey
   #
   user1  9.3.230.119  SNMPv3  u1  -  -  AuthNoPriv  HMAC-MD5  63960c12520dc8829d27f7fbaf5a0470  -  -
   #-----------------------------------------------------------------------------------------------
   ```

   - *winSnmpName* สามารถเป็นค่าใดๆ ก็ได้ ค่านี้จะใช้เมื่อจัดทำคำร้องขอ SNMP โดยใช้คำสั่ง **clsnmp**
   - *targetAgent* คือ IP แอดเดรส ที่เอเจนต์กำลังรัน และใช้ในการสร้างคีย์การอนุญาต ด้วย
   - *admin* มีการตั้งค่าเป็น SNMPv3 เนื่องจาก เราจะส่งคำร้องขอ SNMPv3
   - *secName* คือชื่อของผู้ใช้ ซึ่งคุณกำลังสร้าง ในกรณีนี้ ชื่อคือ u1

- *seclevel* มีการตั้งค่าเป็น AuthNoPriv เนื่องจากกำลังตั้งค่าคอนฟิกเพื่อใช้การอนุญาตไม่ใช่ความเป็นส่วนตัว (ส่งผลให้ไม่มีค่าสำหรับ *privProto* และ *privKey*)
- *authproto* มีการตั้งค่าเป็นโปรโตคอลการอนุญาต ที่ใช้ในการสร้างคีย์การอนุญาต
- *authKey* คือคีย์ที่ไม่ได้โลคัลไลซ์ซึ่งถูกสร้างขึ้นโดยคำสั่ง **pwtokey**

8. บันทึกและปิดไฟล์ /etc/clsnmp.conf

ขั้นตอนที่ 2. ตั้งค่าคอนฟิกกลุ่ม

ขณะนี้ ต้องมีการวาง ผู้ใช้ไว้ในกลุ่ม หากคุณมีกลุ่มที่ตั้งค่าคอนฟิกไว้แล้วด้วยสิทธิการดูและการเข้าถึง ทั้งหมดซึ่งคุณต้องการมอบให้แก่ผู้ใช้รายนี้ คุณสามารถ วางผู้ใช้นี้ในกลุ่มนั้นได้ ถ้าคุณต้องการมอบสิทธิการดูและการเข้าถึงซึ่งไม่มีในกลุ่มอื่นๆ ให้แก่ผู้ใช้รายนี้ หรือถ้าคุณไม่มีกลุ่มที่ตั้งค่าคอนฟิก ให้สร้างกลุ่มและเพิ่มผู้ใช้นี้ลงในกลุ่มที่สร้างขึ้น

เพื่อเพิ่มผู้ใช้ลงในกลุ่มใหม่ ให้สร้างรายการ VACM_GROUP ใหม่ในไฟล์ /etc/snmpdv3.conf รายการกลุ่มสำหรับ u1 มีดังนี้:

```
#--------------------------------------------------------------
# VACM_GROUP entries
#    Defines a security group (made up of users or communities)
#    for the View-based Access Control Model (VACM).
# Format is:
#  groupName securityModel securityName storageType
VACM_GROUP group1 USM u1 -
#--------------------------------------------------------------
```

- *groupName* สามารถเป็นชื่อใดๆ ก็ได้ ชื่อนี้จะกลายเป็นชื่อกลุ่มของคุณ ในกรณีนี้ ชื่อคือ group1
- *securityModel* มีการตั้งค่าเป็น USM ซึ่งใช้ข้อดีของคุณลักษณะความปลอดภัย SNMPv3
- *securityName* คือชื่อของ ผู้ใช้ ในกรณีนี้ ชื่อคือ u1

ขั้นตอนที่ 3. ตั้งค่าคอนฟิกสิทธิการดูและการเข้าถึง

ต้องตั้งค่า สิทธิการดูและการเข้าถึงสำหรับกลุ่มใหม่ที่เพิ่งสร้างขึ้น สิทธิเหล่านี้มีการตั้งค่าโดยการเพิ่มรายการ VACM_VIEW และ VACM_ACCESS ลงในไฟล์ /etc/snmpdv3.conf

1. ตัดสินใจเลือกสิทธิการดูและการเข้าถึงซึ่งคุณต้องการให้กลุ่มใหม่มี
2. เพิ่มรายการ VACM_VIEW ลงในไฟล์ /etc/snmpdv3.conf เพื่อกำหนดอ็อบเจ็กต์ MIB ที่กลุ่มสามารถเข้าถึงได้ ในสถานการณ์จำลองนี้ group1 จะมี สิทธิเข้าถึง interfaces, tcp, icmp, และแผนผังย่อย MIB system อย่างไรก็ตาม เราจะจำกัดสิทธิการเข้าถึงของ group1 เป็นตัวแปร MIB sysObjectID ภายในแผนผังย่อย MIB ของระบบ

   ```
   #--------------------------------------------------------------
   # VACM_VIEW entries
   #    Defines a particular set of MIB data, called a view, for the
   #    View-based Access Control Model.
   # Format is:
   #  viewName viewSubtree viewMask viewType storageType
   VACM_VIEW group1View  interfaces    - included -
   VACM_VIEW group1View  tcp        - included -
   VACM_VIEW group1View  icmp        - included -
   VACM_VIEW group1View  system     - included -
   VACM_VIEW group1View  sysObjectID    - excluded -
   #--------------------------------------------------------------
   ```

   - *viewName* คือชื่อของ มุมมอง ในสถานการณ์จำลองนี้ ชื่อคือ group1View

- *viewSubtree* คือแผนผังย่อย MIB ซึ่งคุณต้องการให้สิทธิเข้าถึง
- *viewType* กำหนดว่าจะรวมแผนผังย่อย MIB ที่กำหนดไว้ในมุมมองหรือไม่ ในกรณีนี้ มีการรวมแผนผังย่อย ทั้งหมด แต่ไม่รวมตัวแปร MIB sysObjectID ซึ่งเป็น ส่วนหนึ่งของแผนผังย่อย system

3. เพิ่มรายการ VACM_ACCESS ลงในไฟล์ /etc/snmpdv3.conf เพื่อกำหนดสิทธิการอนุญาตที่กลุ่มมีในอ็อบเจ็กต์ MIB ซึ่งระบุ ข้างบน สำหรับ group1 มีการให้สิทธิอ่านอย่างเดียว

```
#--------------------------------------------------------------------------------------------------
# VACM_ACCESS entries
#    Identifies the access permitted to different security groups
#    for the View-based Access Control Model.
# Format is:
# groupName contextPrefix contextMatch securityLevel securityModel readView writeView notifyView storageType
VACM_ACCESS  group1 - - AuthNoPriv USM group1View - group1View -
#--------------------------------------------------------------------------------------------------
```

- *groupName* คือชื่อของ กลุ่ม ในกรณีนี้ ชื่อคือ group1
- *securityLevel* คือระดับ ความปลอดภัยที่ใช้อยู่ ในสถานการณ์จำลองนี้ ใช้คีย์การอนุญาต ไม่ใช่คีย์ความเป็นส่วนตัว ดังนั้นจึงมีการตั้งค่าเป็น AuthNoPriv
- *securityModel* คือโมเดล ความปลอดภัยที่คุณใช้อยู่ (SNMPv1, SNMPv2c, หรือ USM) ในสถานการณ์จำลองนี้ มีการตั้งค่า เป็น USM เพื่อให้สามารถใช้คุณลักษณะความปลอดภัย SNMPv3 ได้
- *readView* กำหนด VACM_VIEWs ซึ่งกลุ่มมีสิทธิการอ่าน ในสถานการณ์จำลองนี้ มีการกำหนด group1View ให้ ซึ่งทำให้ group1 มีสิทธิการอ่านในรายการ group1View VACM_VIEW
- *writeView* กำหนด VACM_VIEWs ซึ่งกลุ่มมีสิทธิการเขียน ในสถานการณ์จำลองนี้ ไม่มีการให้สิทธิการเขียน แก่ group1
- *notifyView* ระบุชื่อของ มุมมองที่จะใช้เมื่อทำการดักจับภายใต้การควบคุมของรายการ ในตารางการเข้าถึง

หมายเหตุ: ในบางกรณี อาจจำเป็นต้องใช้หลายรายการ VACM_ACCESS สำหรับหนึ่งกลุ่ม หากผู้ใช้ในกลุ่มมีค่าติดตั้งการอนุญาตและความเป็นส่วนตัวที่แตกต่างกัน (noAuthNoPriv, AuthNoPriv, หรือ AuthPriv) จำเป็นต้องใช้หลายรายการ VACM_ACCESS พร้อมกับพารามิเตอร์ securityLevel ที่ตั้งค่าตามนั้น

ขั้นตอนที่ 4. ตั้งค่าคอนฟิกรายการดักจับสำหรับผู้ใช้

รายการ ดักจับใน SNMPv3 มีการสร้างขึ้นโดยการเพิ่มรายการ NOTIFY, TARGET_ADDRESS และ TARGET_PARAMETERS ลงในไฟล์ /etc/snmpdv3.conf รายการ TARGET_ADDRESS จะระบุตำแหน่งซึ่งคุณต้องการให้ส่งการดักจับ และรายการ TARGET_PARAMETERS จะแม็พข้อมูล TARGET_ADDRESS เข้ากับ group1

รายการ NOTIFY มีการตั้งค่าคอนฟิกแล้วโดยค่าเริ่มต้น ข้อมูลต่อไปนี้เป็นรายการ NOTIFY ดีฟอลต์:

```
NOTIFY notify1 traptag trap -
```

ใน สถานการณ์จำลองนี้ เราใช้ค่าที่ระบุในรายการดีฟอลต์คือ traptag

1. เพิ่มรายการ TARGET_ADDRESS เพื่อระบุตำแหน่งซึ่งคุณต้องการ ให้ส่งการดักจับ

```
#-------------------------------------------------------------------------------
# TARGET_ADDRESS
#    Defines a management application's address and parameters
#    to be used in sending  notifications.
# Format is:
```

```
#  targetAddrName tDomain tAddress tagList targetParams timeout retryCount storageType
#-------------------------------------------------------------------------------------
TARGET_ADDRESS Target1 UDP 9.3.207.107     traptag trapparms1 - - -
```

- *targetAddrName* สามารถเป็นชื่อใดๆ ก็ได้ในสถานการณ์จำลองนี้ เราใช้ Target1
- *tAddress* คือ IP แอดเดรส ที่ควรจะส่งการดักจับสำหรับกลุ่ม
- *tagList* คือชื่อที่ตั้งค่าคอนฟิกใน รายการ NOTIFY ในสถานการณ์จำลองนี้ ชื่อคือ traptag
- *targetParams* สามารถเป็นค่าใดๆ ก็ได้ ค่าที่เราใช้คือ trapparms1 ซึ่งจะถูกใช้ในรายการ TARGET_PARAMETERS

2. เพิ่มรายการ TARGET_PARAMETERS

```
#-----------------------------------------------------------------------
# TARGET_PARAMETERS
#    Defines the message processing and security parameters
#    to be used in sending notifications to a particular management target.
# Format is:
#  paramsName mpModel securityModel securityName securityLevel storageType
#-----------------------------------------------------------------------
TARGET_PARAMETERS trapparms1 SNMPv3  USM     u1        AuthNoPriv -
```

- *paramsName* เป็นค่าเดียวกันกับค่า targetParams ในรายการ TARGET_ADDRESS ซึ่งในกรณีนี้คือ trapparms1
- *mpModel* คือเวอร์ชันของ SNMP ที่ใช้อยู่
- *securityModel* คือโมเดล ความปลอดภัยที่คุณใช้อยู่ (SNMPv1, SNMPv3, หรือ USM) ในสถานการณ์จำลองนี้ มีการตั้งค่า เป็น USM เพื่อให้สามารถใช้คุณลักษณะความปลอดภัย SNMPv3 ได้
- *securityName* คือชื่อผู้ใช้ ที่ระบุในรายการ USM_USER ซึ่งในกรณีนี้คือ u1
- *securityLevel* มีการตั้งค่าเป็น AuthNoPriv เนื่องจาก เรากำลังใช้คีย์การอนุญาต ไม่ใช่คีย์ความเป็นส่วนตัว

ขั้นตอนที่ 5. หยุดและเริ่มต้น snmpd daemon

หลังจากทำการ เปลี่ยนแปลงไฟล์ /etc/snmpdv3.conf แล้ว ให้หยุดและ เริ่มต้น **snmpd** daemon

1. พิมพ์คำสั่งต่อไปนี้เพื่อหยุด **snmpd** daemon:

```
stopsrc -s snmpd
```

2. พิมพ์คำสั่งต่อไปนี้เพื่อเริ่มต้น **snmpd** daemon:

```
startsrc -s snmpd
```

ขณะนี้ ค่าติดตั้งใหม่จะมีผลบังคับใช้

หมายเหตุ: การรีเฟรชเอเจนต์ SNMPv3 โดยใช้ refresh -s snmpd เพียงอย่างเดียวจะใช้ไม่ได้ เหมือนใน SNMPv1 หากคุณทำการเปลี่ยนแปลงในไฟล์ /etc/snmpdv3.conf คุณต้องหยุดและเริ่มต้น daemon ตามที่แนะนำข้างบน ฟังก์ชันการตั้งค่าคอนฟิกแบบไดนามิก ที่สนับสนุนใน SNMPv3 จะไม่อนุญาตให้คุณรีเฟรช

ขั้นตอนที่ 6. ทดสอบการตั้งค่าคอนฟิกของคุณ

เพื่อตรวจสอบว่าการตั้งค่าคอนฟิก ของคุณถูกต้อง คุณสามารถรันคำสั่งต่อไปนี้บน SNMP manager

```
clsnmp -h user1 walk mib
```

โดยที่ *mib* คือ แผนผังย่อย MIB ซึ่งผู้ใช้มีสิทธิเข้าถึง ในสถานการณ์จำลองนี้ แผนผังย่อยอาจเป็น `interfaces`, `tcp`, `icmp`, หรือ `system` ถ้าการตั้งค่าคอนฟิกถูกต้อง คุณจะเห็น ข้อมูลจากแผนผังย่อยที่ระบุ

ถ้าคุณไม่ได้รับเอาต์พุต ที่ถูกต้อง ให้ตรวจทานขั้นตอนในเอกสารนี้และตรวจสอบว่าคุณได้ป้อน ข้อมูลทั้งหมดอย่างถูกต้อง

การแก้ไขปัญหา SNMPv3

อาจพบปัญหาเหล่านี้เมื่อใช้ **SNMPv3**

- ขณะย้าย คุณต้องย้ายชุมชนและรายการ SMUX ที่ กำหนดไว้ในไฟล์ `/etc/snmpd.conf` ไปยังไฟล์ `/etc/snmpdv3.conf` สำหรับข้อมูลเกี่ยวกับการย้าย ข้อมูลนี้ให้ดูที่ “การย้ายจาก SNMPv1 ไปยัง SNMPv3” ในหน้า 511
- คำร้องขอไม่ได้สร้างการตอบกลับใดๆ

  สาเหตุที่เป็นไปได้มากที่สุดของ ปัญหานี้คือ ข้อผิดพลาดคอนฟิกูเรชันในไฟล์ `/etc/snmpdv3.conf` หรือ ไฟล์ `/etc/clsnmp.conf` หรือทั้งสองไฟล์ ตรวจทานไฟล์เหล่านี้ อย่างรอบคอบเพื่อให้แน่ใจว่ามีการป้อนข้อมูลครบทั้งหมดอย่างถูกต้อง สำหรับข้อมูลเกี่ยวกับ การแก้ไขไฟล์เหล่านี้เมื่อสร้างผู้ใช้ใหม่ ให้ดูที่ “การสร้างผู้ใช้ใน SNMPv3” ในหน้า 515
- มีการกำหนดคอนฟิกผู้ใช้ใหม่โดยใช้ทั้งการพิสูจน์ตัวตนและคีย์ความเป็นส่วนตัว แต่มีการส่งคืนข้อความแสดงข้อผิดพลาดเมื่อใช้ผู้ใช้รายนี้

  สาเหตุที่เป็นไปได้มากที่สุด คือ คุณไม่ได้กำลังรันเวอร์ชันที่เข้ารหัสของ **SNMPv3** ปฏิบัติตาม ขั้นตอนเหล่านี้เพื่อกำหนดเวอร์ชันที่คุณกำลังรันอยู่:

  1. รัน `ps -e|grep snmpd`
     - หากคุณไม่ได้รับเอาต์พุต คุณอาจต้องเริ่มต้น **snmpd** daemon รัน `startsrc -s snmpd`
     - หากเอาต์พุตมีคำว่า `snmpdv1` แสดงว่าคุณกำลังรัน **SNMPv1** คุณจะสามารถทำคำร้องขอ **SNMPv1** เมื่อกำลังรันเวอร์ชันนี้
     - หากเอาต์พุตมีคำว่า `snmpdv3ne` แสดงว่าคุณกำลังรันเวอร์ชันที่ไม่ได้เข้ารหัสของ **SNMPv3** หลังจากติดตั้งระบบปฏิบัติการ AIX เวอร์ชันนี้ จะรันอยู่โดยดีฟอลต์ เวอร์ชันนี้ไม่อนุญาตให้คุณใช้คีย์ ความเป็นส่วนตัว
     - หากเอาต์พุตมีคำว่า `snmpdv3e` แสดงว่าคุณกำลังรันเวอร์ชันที่เข้ารหัสของ **SNMPv3** ซึ่งเป็นผลิตภัณฑ์ที่สามารถติดตั้งได้แยกต่างหาก เวอร์ชันที่เข้ารหัสของ **SNMPv3** มีอยู่บน AIX Expansion Pack ในตำแหน่งที่ใช้ได้ เวอร์ชันที่เข้ารหัสของ **SNMPv3** อนุญาตให้ใช้ คีย์ความเป็นส่วนตัวได้
  2. พิจารณาว่าเวอร์ชันซึ่งคุณกำลังรันเป็นเวอร์ชันที่ต้องการหรือไม่ ถ้าไม่ ให้ใช้คำสั่ง **snmpv3_ssw** เพื่อเปลี่ยน เวอร์ชันดังนี้:
     - `snmpv3_ssw -1` จะสลับไปยัง **SNMPv1**
     - `snmpv3_ssw -n` จะสลับไปยัง **SNMPv3** ที่ไม่เข้ารหัส
     - `snmpv3_ssw -e` จะสลับไปยัง **SNMPv3** ที่เข้ารหัส ถ้ามีการติดตั้งไว้
- หลังจากทำการเปลี่ยนแปลงในไฟล์ `/etc/snmpdv3.conf` และ รีเฟรช daemon แล้ว การเปลี่ยนแปลงของฉันไม่มีผลบังคับใช้

  หลังจากทำ การเปลี่ยนแปลงในไฟล์ `/etc/snmpdv3.conf` แล้ว ต้องหยุดและเริ่มต้น **SNMP** daemon การรีเฟรช daemon จะไม่ทำงาน ใช้โพรซีเดอร์ต่อไปนี้:

  1. หยุด **SNMP** daemon โดยรัน `stopsrc -s snmpd`
  2. เริ่มต้น **SNMP** daemon โดยรัน `startsrc -s snmpd`
- เอเจนต์ย่อย DPI2 เริ่มต้นขึ้น แต่ไม่สามารถเคียวรีตัวแปร MIB จากเอเจนต์ย่อย นั้นได้

สาเหตุที่เป็นไปได้มากที่สุดคือ ไม่ได้กำหนดคอนฟิกชุมชน public ในไฟล์ /etc/snmpdv3.conf โดยค่าดีฟอลต์ เอเจนต์ย่อย DPI2 ที่จัดส่งมาพร้อมกับ AIX ใช้ ชุมชนชื่อ public เพื่อเชื่อมต่อตัวเองกับเอเจนต์ **SNMP** ชุมชน public มีการกำหนดคอนฟิกอยู่ในไฟล์ /etc/snmpdv3.conf โดยค่าดีฟอลต์ หากคุณลบชุมชน public ออกจาก ไฟล์ /etc/snmpd.conf ให้เพิ่มบรรทัดต่อไปนี้ลงใน ไฟล์:

```
VACM_GROUP group1 SNMPv1 public -
VACM_VIEW defaultView 1.3.6.1.4.1.2.2.1.1.1.0 - included -
VACM_ACCESS  group1 - - noAuthNoPriv SNMPv1  defaultView - defaultView -
COMMUNITY public    public     noAuthNoPriv 0.0.0.0     0.0.0.0         -
```

1.3.6.1.4.1.2.2.1.1.1.0 คือ OID สำหรับ dpiPortForTCP.0

- ตัวแปร MIB มีการจัดการโดย SMUX peer ที่สามารถเคียวรีได้ก่อนที่จะไม่ สามารถเคียวรีการย้ายได้อีก

  ตรวจสอบให้แน่ใจว่ารายการ SMUX มีอยู่ในไฟล์ /etc/snmpdv3.conf และไฟล์ /etc/snmpd.peers หากคุณกำหนดคอนฟิก SMUX peers ใหม่ ตรวจสอบให้แน่ใจว่ามีการป้อน peer ใหม่ในทั้งสองไฟล์เหล่านี้
- มีการนำชุดส่วนบุคคลของตัวแปร MIB ไปใช้ แต่ไม่สามารถรวมหรือแยกตัวแปร จากการดูโดยผู้ใช้รายอื่น

  ในรายการ VACM_VIEW ในไฟล์ /etc/snmpdv3.conf คุณต้องระบุ OID ของตัวแปร MIB แทนชื่อตัวแปร MIB
- ไม่ได้รับการดักจับที่ได้รับ

  ตรวจสอบให้แน่ใจว่าคุณกำหนดคอนฟิก รายการการดักจับอย่างถูกต้องในไฟล์ /etc/snmpdv3.conf นอกจากนั้น ถ้าการดักจับเป็นการดักจับ **SNMPv3** ต้องกำหนดคอนฟิกไฟล์ /etc/clsnmp.conf ด้วย สำหรับคำแนะนำเกี่ยวกับการกำหนดคอนฟิกการดักจับ ให้ดูที่ “การสร้างผู้ใช้ใน SNMPv3” ในหน้า 515

  นอก จากนี้ ต้องแน่ใจว่าเครื่องที่ระบุให้รับการดักจับ (ในไฟล์ /etc/snmpdv3.conf) กำลังฟังการดักจับอยู่ คุณสามารถเริ่มต้นโปรเซสนี้โดยรัน clsnmp trap บนบรรทัดรับคำสั่งของเครื่องที่ได้รับ
- เพราะเหตุใดเซิร์ฟเวอร์ DPI2 จึงไม่รันในสภาวะแวดล้อม **SNMPv3**?

  ในสถาปัตยกรรม **SNMPv3** เอเจนต์ **SNMPv3** รันเซิร์ฟเวอร์ DPI2 ด้วยตัวเอง โปรดดู “สถาปัตยกรรมแบบ SNMPv3” ในหน้า 502 สำหรับข้อมูลเพิ่มเติม

SNMPv1

ข้อมูลนี้เป็นข้อมูลเฉพาะกับ **SNMPv1** เมื่อใช้ **SNMPv1** เอเจนต์ **snmpd** ใช้ scheme การพิสูจน์ตัวตนแบบปกติ เพื่อกำหนดตำแหน่งตัวจัดการ **Simple Network Management Protocol (SNMP)** ที่สามารถเข้าถึงตัวแปร Management Information Base (MIB)

scheme การพิสูจน์ตัวตนนี้เกี่ยวข้องกับข้อกำหนดคุณสมบัติของ **SNMP** ที่เข้าถึง **SNMPv1** นโยบายการเข้าถึง **SNMP** คือความสัมพันธ์ของการดูแล ที่เกี่ยวข้องกับการเชื่อมโยงระหว่าง **SNMP** community โหมดการเข้าถึง และมุมมอง MIB

SNMP community คือกลุ่มของโฮสต์ตั้งแต่หนึ่งโฮสต์ขึ้นไป และชื่อ community ชื่อ community คือสตริง octets ที่ตัวจัดการ **SNMP** ต้องฝังอยู่ในแพ็กเก็ตคำร้องขอ **SNMP** สำหรับวัตถุประสงค์ในการพิสูจน์ตัวตน

โหมดการเข้าถึง ระบุการเข้าถึงโฮสต์ใน community ที่ได้รับอนุญาตให้เรียกคืนและแก้ไขตัวแปร MIB จากเอเจนต์ **SNMP** ที่ระบุเฉพาะ โหมดการเข้าถึงต้องเป็นหนึ่งในโหมดต่อไปนี้: *none*, *read-only*, *read-write* หรือ *write-only*

มุมมอง MIB view กำหนดแผนผังย่อย MIB ตั้งแต่หนึ่งแบบขึ้นไปซึ่ง **SNMP** community ที่ระบุเฉพาะสามารถเข้าถึงได้ มุมมอง MIB สามารถเป็นแผนผัง MIB ทั้งหมดหรือชุดย่อยที่จำกัดของ แผนผัง MIB ทั้งหมด

เมื่อเอเจนต์ **SNMP** ได้รับคำร้องขอ เอเจนต์ตรวจสอบชื่อ community ด้วย IP แอดเดรสของโฮสต์ที่ร้องขอเพื่อพิจารณาว่าโฮสต์ที่ร้องขอ คือสมาชิกของ **SNMP** community ที่ระบุโดยชื่อ community หากโฮสต์ที่ร้องขอคือสมาชิกของ **SNMP** community เอเจนต์ **SNMP** จะกำหนดว่า โฮสต์ที่ร้องขอได้รับอนุญาตให้เข้าถึงตามที่ระบุไว้ สำหรับตัวแปร MIB ที่ระบุไว้ตามที่กำหนดไว้ในนโยบายการเข้าถึงที่เชื่อมโยงกับ community นั้น หากตรงตามเงื่อนไข เอเจนต์ **SNMP** จะพยายาม ให้คำร้องขอมิฉะนั้น เอเจนต์ **SNMP** จะสร้างแทร็บ *authenticationFailure* หรือส่งคืนข้อความแสดงความผิดพลาดที่เหมาะสมกับโฮสต์ที่ร้องขอ

นโยบายการเข้าถึง **SNMPv1** สำหรับเอเจนต์ **snmpd** คือนโยบายแบบผู้ใช้ตั้งค่าเอง และถูกระบุอยู่ในไฟล์ `/etc/snmpd.conf` To configure the **SNMP** access policies for the **snmpd** agent, see the `/etc/snmpd.conf` file ใน *ข้อมูลอ้างอิงไฟล์*.

คอนฟิกูเรชัน SNMP daemon

Simple Network Management Protocol (SNMP) daemon คือกระบวนการเซิร์ฟเวอร์พื้นหลังที่สามารถรันอยู่บนโฮสต์เวิร์กสเตชัน **Transmission Control Protocol/Internet Protocol (TCP/IP)** ใดๆ

daemon ทำหน้าที่เป็นเอเจนต์ **SNMP** การรับ การพิสูจน์ตัวตน และประมวลผลคำร้องขอ **SNMP** จากตัวจัดการแอ็พพลิเคชัน โปรดดู Simple Network Management Protocol, วิธีที่ตัวจัดการทำงาน และ วิธีที่เอเจนต์ทำงาน ใน *หลักการเขียนโปรแกรมการสื่อสาร* สำหรับข้อมูลเพิ่มเติมโดยละเอียดบนเอเจนต์และตัวจัดการการทำงาน

หมายเหตุ: คำว่า **SNMP** daemon, เอเจนต์ **SNMP** และเอเจนต์ ถูกใช้เพื่อแลกเปลี่ยนระหว่างกัน

snmpd daemon ต้องการวนกลับอินเตอร์เฟส **TCP/IP** ที่ต้องแอ็คทีฟสำหรับคอนฟิกูเรชันต่ำ ป้อนคำสั่งต่อไปนี้ ก่อนเริ่มต้น **TCP/IP**:

```
ifconfig lo0 loopback up
```

SNMP daemon จะพยายามซ่อนซ็อกเก็ตเพื่อขอรับ **User Datagram Protocol (UDP)** และพอร์ต **Transmission Control Protocol (TCP)** ที่ใช้งานได้ ซึ่งต้องถูกนิยามไว้ในไฟล์ `/etc/services` ดังต่อไปนี้:

```
snmp            161/udp
snmp-trap       162/udp
smux            199/tcp
```

เซอร์วิส `snmp` ต้องถูกกำหนดเป็นพอร์ต `161` ตามที่ต้องการโดย RFC `1157` ไฟล์ `/etc/services` กำหนดพอร์ต `161`, `162` และ `199` ให้กับเซอร์วิสเหล่านี้ หากไฟล์ `/etc/services` ปิดการให้บริการเครื่องอื่น พอร์ตที่กำหนดไว้เหล่านี้ต้องถูกทำให้พร้อมใช้งาน ในการใช้ไฟล์ `/etc/services` บนเซิร์ฟเวอร์ก่อนที่ **SNMP** daemon สามารถรัน

SNMP daemon อ่านไฟล์คอนฟิกูเรชันที่รัน **SNMP** เวอร์ชันการเริ่มต้นทำงานเมื่อคำสั่ง **refresh** (หาก **snmpd** daemon ถูกเรียกใช้ภายใต้การควบคุม System Resource Controller) หรือการส่งสัญญาณ **kill -1** ถูกเรียกใช้งาน

ไฟล์ /etc/snmpd.conf:

ไฟล์คอนฟิกูเรชันที่ชื่อ `/etc/snmpd.conf` ระบุชื่อ community และสิทธิพิเศษในการเชื่อมโยงและมุมมอง โฮสต์สำหรับการแจ้งเตือนแทร็บ แอ็ตทริบิวต์บันทึกการทำงาน คอนฟิกูเรชันที่ระบุเฉพาะพารามิเตอร์ **snmpd** และคอนฟิกูเรชันมัลติเพล็กเซอร์เดี่ยว (SMUX) สำหรับ **SNMP** daemon สำหรับ **SNMPv1**

โปรดดูไฟล์ /etc/snmpd.conf ใน *ข้อมูลอ้างอิงไฟล์* สำหรับข้อมูลเพิ่มเติม

การประมวลผล SNMP daemon

Simple Network Management Protocol (SNMP) daemon ประมวลผลคำร้องขอ **SNMP** จากแอ็พพลิเคชันตัวจัดการ

โปรดอ่าน Simple Network Management Protocol (SNMP), วิธีที่ตัวจัดการการทำงาน และ วิธีที่เอเจนต์ทำงาน ใน *หลักการเขียนโปรแกรมการสื่อสาร* สำหรับข้อมูลโดยละเอียดเกี่ยวกับเอเจนต์และฟังก์ชันตัวจัดการ

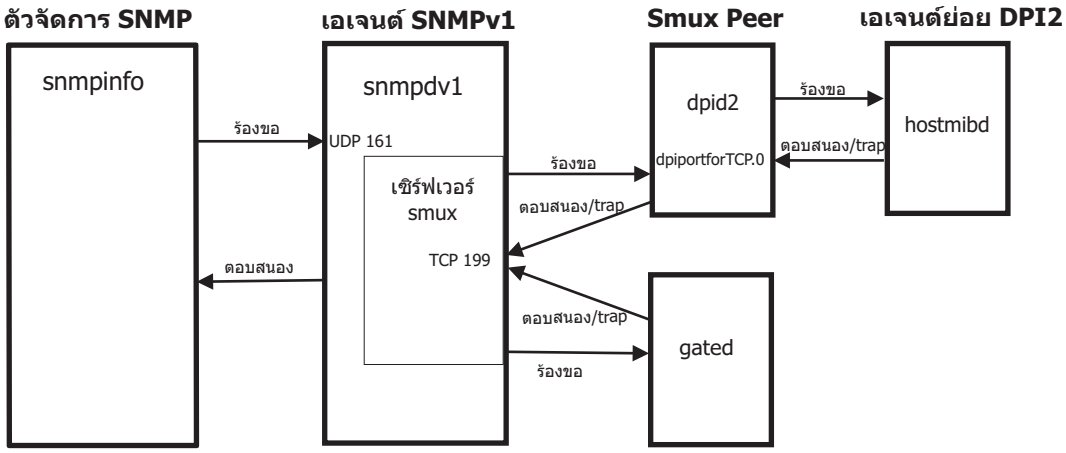
การประมวลผลข้อความ SNMP และการพิสูจน์ตัวตน:

คำร้องขอ, traps และการตอบสนองทั้งหมดถูกส่งในรูปแบบของข้อความที่ถูกเข้ารหัสแบบ ASN.1

ข้อความ ดังที่กำหนดโดย RFC 1157 มีโครงสร้างต่อไปนี้ :

Version Community PDU

โดยที่ *Version* เป็นเวอร์ชันของ SNMP (ปัจจุบันเป็นเวอร์ชัน 1) *Community* เป็นชื่อ community และ *PDU* เป็น protocol data unit ที่ประกอบด้วยคำร้องขอ การตอบสนอง และข้อมูล trap ของ **SNMP** PDU ยังถูกเข้ารหัสตามกฎของ ASN.1



รูปที่ 30. ส่วนหลักของสถาปัตยกรรม SNMPv1

รูปประกอบนี้แสดงและเป็นตัวอย่างของสถาปัตยกรรม **SNMPv1** เอเจนต์ย่อย DPI2 เพียร์ smux ตัวจัดการ **SNMP** และเอเจนต์ **SNMP** ถูกแสดง นอกจากนี้ วิธีการสื่อสารกับแต่ละบุคคลถูกแสดง

SNMP daemon รับและส่งข้อความโปรโตคอล **SNMP** ทั้งหมดผ่าน **Transmission Control Protocol/Internet Protocol (TCP/IP) User Datagram Protocol (UDP)** คำร้องขอจะถูกยอมรับบน well-known พอร์ต 161 Traps จะถูกส่งไปยังโฮสต์ที่ถูกลิสต์ใน entry ของ trap ในไฟล์ `/etc/snmpd.conf` ที่รับฟังบน well-known พอร์ต 162

เมื่อได้รับคำร้องขอ IP แอดเดรสต้นทางและชื่อ community name จะถูกตรวจสอบกับลิสต์ที่ประกอบด้วย IP แอดเดรส ชื่อ community การอนุญาต และมุมมอง ดังที่ระบุใน community และ entry ของมุมมองในไฟล์ `/etc/snmpd.conf` **snmpd** เอเจนต์จะอ่านไฟล์นี้เมื่อตอนเริ่มทำงานและบนคำสั่ง **refresh** หรือสัญญาณ **kill -1** ถ้าไม่พบ entry ที่ตรงกัน คำร้องขอจะถูกข้ามไป ถ้าพบ entry ที่ตรงกัน การยอมให้มีการเข้าถึงจากสิทธิที่ถูกกำหนดใน entry ของ community และมุมมองสำหรับ IP แอดเดรสนั้น community และชื่อของมุมมองที่เชื่อมโยงในไฟล์ `/etc/snmpd.conf` ทั้งข้อความและ PDU ต้องถูกเข้ารหัสตามกฎของ ASN.1

scheme การพิสูจน์ตัวตนนี้ไม่ได้มีวัตถุประสงค์ที่จะให้ความปลอดภัยเต็มที่ ถ้า **SNMP** daemon ถูกใช้เฉพาะสำหรับคำร้องขอ get และ get-next ความปลอดภัยอาจไม่เป็นปัญหา ถ้าชุดของคำร้องขอได้รับอนุญาต ชุดของสิทธิสามารถถูกจำกัด

See the `/etc/snmpd.conf` file ใน *ข้อมูลอ้างอิงไฟล์* for further information. ดูที่ Management Information Base (MIB) ใน *หลักการเขียนโปรแกรมการสื่อสาร* สำหรับข้อมูลเพิ่มเติม

กระบวนการร้องขอ SNMP:

มีสามชนิดของคำร้องขอ PDU ที่สามารถรับได้โดย **SNMP** daemon

ชนิดของคำร้องขอที่นิยามไว้ใน RFC 1157 และ PDU มีรูปแบบ ทั้งหมดต่อไปนี้:

ตารางที่ 85. รูปแบบคำร้องขอ PDU

request-ID	error-status	error-index	variable-bindings
GET	0	0	*VarBindList*
GET-NEXT	0	0	*VarBindList*
SET	0	0	*VarBindList*

ฟิลด์ request-ID ระบุลักษณะของคำร้องขอ ฟิลด์ error-status และฟิลด์ error-index ถูกยกเลิกการใช้งานและต้องมีค่าศูนย์ 0 (ศูนย์) และฟิลด์ variable-bindings มีความยาวตัวแปรที่แสดง ID อินสแตนซ์รูปแบบตัวเลข ที่มีคร้องขอ หากค่าของฟิลด์ request-ID ถูกตั้งค่า SET ไว้ ฟิลด์ variable-bindings คือรายการของคู่ของ ID อินสแตนซ์ และค่า

โปรดอ่าน การใช้ฐานข้อมูล Management Information Base (MIB) ใน *หลักการเขียนโปรแกรมการสื่อสาร* สำหรับคำกล่าวของสามชนิดคำร้องขอ

กระบวนการตอบกลับ SNMP:

ตอบกลับ PDUs ที่ใกล้เคียงกับรูปแบบเดียวกันกับคำร้องขอ PDU

ตารางที่ 86. ตอบกลับรูปแบบ PDU

request-ID	error-status	error-index	variable-bindings
GET-RESPONSE	*ErrorStatus*	*ErrorIndex*	*VarBindList*

หากคำร้องขอที่ได้ประมวลผลแล้วเป็นผลสำเร็จ ค่าสำหรับทั้งฟิลด์ error-status และ error-index คือ 0 (ศูนย์) และฟิลด์การเชื่อมตัวแปรมีรายการที่สมบูรณ์ของคู่ของ ID อินสแตนซ์และค่า

หาก ID อินสแตนซ์ใดๆ ในฟิลด์ variable-bindings ของคำร้องขอ PDU ไม่ได้ประมวลผลเป็นผลสำเร็จ เอเจนต์ SNMP จะหยุดการประมวลผล และเขียนดัชนี ของ ID อินสแตนซ์ที่ล้มเหลวลงในฟิลด์ error-index บันทึกโค้ดระบุความผิดพลาดลงในฟิลด์ error-status และคัดลอกรายการผลลัพธ์สำหรับส่วนที่สมบูรณ์ ลงในฟิลด์ variable-binding

RFC 1157 นิยามค่าต่อไปนี้สำหรับฟิลด์ error-status:

ตารางที่ 87. ค่าสำหรับฟิลด์ error-status

Value	Value	คำอธิบาย
noError	0	การประมวลผลที่เสร็จสิ้นเป็นผลสำเร็จ (error-index คือ 0)
tooBig	1	ขนาดของการตอบกลับ PDU จะมีค่าเกินกว่าข้อจำกัดของ implementation-defined (error-index คือ 0)
noSuchName	2	ID อินสแตนซ์ไม่มีอยู่ในมุมมอง MIB ที่เกี่ยวข้องสำหรับชนิดคำร้องขอ GET และ SET หรือไม่มีตัวระบุผลสำเร็จในแผนผัง MIB ในมุมมอง MIB ที่เกี่ยวข้องสำหรับคำร้องขอ GET-NEXT (error-index ไม่ใช่ค่าศูนย์)
badValue	3	สำหรับคำร้องขอ SET เท่านั้น ค่าที่ระบุเข้ากันไม่ได้กับ ชนิดแอ็ตทริบิวต์ของ ID อินสแตนซ์การตอบกลับ (error-index ไม่ใช่ค่าศูนย์)
readOnly	4	ไม่ได้นิยามไว้
genErr	5	ข้อผิดพลาด implementation-defined เกิดขึ้น (error- index ไม่ใช่ค่าศูนย์) ตัวอย่างเช่น ความพยายามในการกำหนดค่าที่มีค่าเกินกว่าข้อจำกัดที่นำไปปฏิบัติ

การประมวลผลการแทร็บ SNMP:

แทร็บ PDU ถูกนิยามโดย RFC 1157 เพื่อให้มีรูปแบบตามที่แสดงอยู่ในตาราง นี้

ตารางที่ 88. รูปแบบแทร็บ PDU

enterprise	agent-address	generic-trap	specific-trap	time-stamp	variable-bindings
Object ID	*Integer*	*Integer*	*Integer*	*TimeTicks*	*VarBindList*

ฟิลด์ถูกใช้ดังต่อไปนี้:

ไอเท็ม **คำอธิบาย**

enterprise
ตัวระบุอ็อบเจ็กต์ที่กำหนดให้กับเวนเดอร์ผู้ใช้เอเจนต์ นี่คือค่าของตัวแปร **sysObjectID** และ เป็นค่าเฉพาะสำหรับผู้นำไปใช้แต่ละรายของเอเจนต์ **SNMP** ค่าได้ถูกกำหนดให้กับ การนำไปใช้งานนี้ของเอเจนต์คือ **1.3.6.1.4.1.2.3.1.2.1.1.3** หรือ **risc6000snmpd.3**

agent-address
IP แอดเดรสของอ็อบเจ็กต์ที่สร้างแทร็บ

ไอเท็ม	คำอธิบาย
generic-trap	เลขจำนวนเต็ม ดังต่อไปนี้: **0** *coldStart* **1** *warmStart* **2** *linkDown* **3** *linkUp* **4** *authenticationFailure* **5** *egpNeighborLoss* **6** *enterpriseSpecific*
specific-trap	ไม่ได้ใช้ สงวนไว้สำหรับการพัฒนาในอนาคต
time-stamp	เวลาที่ผ่านไปในหน่วยที่หนึ่งร้อยของวินาทีจากการ reinitialization ครั้งล่าสุดของเอเจนต์กับเหตุการณ์ที่สร้างแทร็บ
variable-bindings	ข้อมูลพิเศษขึ้นอยู่กับชนิดของ *generic-trap*

ค่า generic-trap ต่อไปนี้บ่งชี้ถึงเหตุการณ์ของระบบที่ได้ถูก ตรวจพบ:

ไอเท็ม	คำอธิบาย
coldStart	เอเจนต์ที่กำลัง reinitialize ข้อมูลคอนฟิกูเรชันหรือค่าตัวแปร MIB หรือทั้งสองอาจเปลี่ยนแปลงไป รีสตาร์ทการวัด epoch
warmStart	เอเจนต์กำลัง reinitialize แต่คอนฟิกูเรชันหรือตัวแปร MIB ไม่ได้เปลี่ยนแปลง ในการนำไปใช้งานของเอเจนต์ **SNMP** แทร็บ *warmStart* ถูกสร้างขึ้นเมื่อไฟล์ `/etc/snmpd.conf` ถูกอ่านอีกครั้ง ข้อมูลคอนฟิกูเรชันในไฟล์ `/etc/snmpd.conf` ใช้สำหรับคอนฟิกูเรชันเอเจนต์ที่ไม่มีผลข้างเคียงเกี่ยวกับฐานข้อมูลตัวจัดการ SNMP การวัด epoch ไม่ควรถูกรีสตาร์ท
linkDown	เอเจนต์ถูกตรวจพบที่อินเตอร์เฟสการสื่อสารที่รู้จัก ได้ถูกปิดใช้งาน
linkUp	เอเจนต์ถูกตรวจพบที่อินเตอร์เฟสการสื่อสารที่รู้จัก ได้ถูกเปิดใช้งาน
authenticationFailure	ข้อความที่ได้รับซึ่งไม่สามารถพิสูจน์ตัวตนได้
egpNeighborLoss	**Exterior Gateway Protocol (EGP)** neighbor หายไป ค่านี้ ถูกสร้างขึ้นเมื่อเอเจนต์กำลังรันอยู่บนโฮสต์ที่รัน **gated** daemon โดยใช้ **EGP**
enterpriseSpecific	ไม่ได้ถูกนำมาใช้ สงวนไว้สำหรับการใช้ในอนาคต

แทร็บ *linkDown* และ *linkUp* มี ID อินสแตนซ์เดี่ยว/คู่ของค่าในรายการที่เชื่อมกับตัวแปร ID อินสแตนซ์ ระบุ **ifIndex** ของอะแด็ปเตอร์ที่ปิดใช้งานหรือเปิดใช้งาน และค่าคือค่า **ifIndex** แทร็บสำหรับ *egpNeighborLoss* ยังมีการเชื่อมที่ประกอบด้วย ID อินสแตนซ์ ID และค่าของ *egpNeighAddr* สำหรับ neighbor ที่หายไป

ส่วนสนับสนุน SNMP daemon สำหรับ EGP family ของตัวแปร MIB

หากเอเจนต์โฮสต์กำลังรัน **gated** daemon พร้อมกับ **Exterior Gateway Protocol (EGP)** ที่เปิดใช้งาน ซึ่งมีตัวแปร Management Information Base (MIB) จำนวนมากในกลุ่ม **EGP** สนับสนุนโดย **gated** daemon ซึ่งเอเจนต์ **snmpd** agent สามารถเข้าถึงได้

ตัวแปร **EGP** MIB ต่อไปนี้เป็นอินสแตนซ์เดี่ยวเฉพาะ:

ไอเท็ม	คำอธิบาย
egpInMsgs	จำนวนข้อความ **EGP** ที่ได้รับโดยไม่มีข้อผิดพลาด
egpInErrors	จำนวนข้อความ **EGP** ที่ได้รับด้วยความผิดพลาด
egpOutMsgs	จำนวนข้อความ **EGP** ทั้งหมดโดย **gated** daemon ที่รันอยู่บนโฮสต์ของเอเจนต์
egpOutErrors	จำนวนข้อความ **EGP** ที่ไม่สามารถส่งได้โดยเอเจนต์โฮสต์ **gated** daemon เนื่องจากข้อจำกัดด้านรีซอร์ส
egpAs	ระบบแบบอัตโนมัติที่มีจำนวนของเอเจนต์โฮสต์ **gated** daemon

ตัวแปร **EGP** MIB ต่อไปนี้มีอินสแตนซ์สำหรับเพียร์ **EGP** และ neighbor ที่ได้รับโดยเอเจนต์โฮสต์ **gated**:

ไอเท็ม	คำอธิบาย
egpNeighState	สถานะของเพียร์ **EGP** นี้คือ: **1** idle **2** ขอรับ **3** ไม่ทำงาน **4** ทำงาน **5** หยุด
egpNeighAddr	IP แอดเดรสของเพียร์ **EGP** นี้
egpNeighAs	ระบบแบบอัตโนมัติที่มีจำนวนของเพียร์ **EGP** ศูนย์ (0) บ่งชี้ถึงระบบแบบอัตโนมัติของเพียร์นี้ที่ไม่ยังไม่รู้จัก
egpInNeighMsgs	จำนวนข้อความ **EGP** ที่ได้รับโดยไม่มีข้อผิดพลาดจากเพียร์ **EGP** นี้
egpNeighInErrs	จำนวนข้อความ **EGP** ที่ได้รับด้วยความผิดพลาดจากเพียร์ **EGP** นี้
egpNeighOutMsgs	จำนวนของข้อความ **EGP** ที่ถูกสร้างขึ้นแบบโลคัลกับเพียร์ **EGP** นี้
egpNeighOutErrs	จำนวนของข้อความ **EGP** ที่ถูกสร้างขึ้นโดยไม่ส่งไปยังยังเพียร์ **EGP** นี้เนื่องจากมีข้อจำกัดด้านรีซอร์ส
egpNeighInErrMsgs	จำนวนของข้อความแสดงความผิดพลาด **EGP** ที่ถูกกำหนดไว้ซึ่งได้รับจากเพียร์ **EGP** นี้
egpNeighOutErrMsgs	จำนวนของข้อความแสดงความผิดพลาด **EGP** ที่ถูกกำหนดไว้ซึ่งส่งไปยังเพียร์ **EGP** นี้
egpNeighStateUp	จำนวนของการเปลี่ยนสถานะ **EGP** ไปเป็นสถานะ UP กับเพียร์ **EGP** นี้
egpNeighStateDowns	จำนวนการเปลี่ยนสถานะ **EGP** จากสถานะ UP ไปเป็นสถานะอื่น กับเพียร์ **EGP** นี้
egpNeighIntervalHello	ช่วงระหว่างการส่งข้อมูลคำสั่ง **EGP** Hello ใหม่อีกครั้งในเวลาที่ หนึ่งร้อยของวินาที
egpNeighIntervalPoll	ช่วงระหว่างการส่งข้อมูลคำสั่ง **EGP** poll ใหม่อีกครั้งในเวลาที่ หนึ่งร้อยของวินาที
egpNeighMode	โหมดการหยั่งสัญญาณของเพียร์ **EGP** นี้ โหมดสามารถแอ็คทีฟ (1) หรือพาสซีฟ (2) อย่างใดอย่างหนึ่ง
egpNeighEventTrigger	ตัวแปรควบคุมทริกเกอร์เหตุการณ์เริ่มต้นและเหตุการณ์สิ้นสุดที่เริ่มต้นด้วยตัวดำเนินการ กับเพียร์ **EGP** นี้ ตัวแปร MIB นี้สามารถตั้งค่าเพื่อเริ่มต้น (1) หรือหยุดทำงาน (2)

ถ้า **gated** daemon ไม่รันอยู่ หรือถ้า **gated** daemon กำลังรันอยู่ แต่ไม่ถูกกำหนดคอนฟิกให้สื่อสารกับเอเจนต์ **snmpd**, หรือถ้า **gated** daemon ไม่ถูกกำหนดคอนฟิกสำหรับ **EGP**, คำร้องขอ get และ set สำหรับค่าของตัวแปรเหล่านี้จะคืนค่าโค้ดการตอบสนองข้อผิดพลาด *noSuchName*

ไฟล์คอนฟิกูเรชันสำหรับ **gated** daemon ที่ชื่อ `/etc/gated.conf` ควรมีคำสั่งต่อไปนี้:

```
snmp    yes;
```

gated daemon ถูกตั้งค่าภายในให้เป็นโปรโตคอลเพียร์ Simple Network Management Protocol (SNMP) single multiplexer (SMUX) หรือพร็อกซีเอเจนต์ของ **snmpd** daemon เมื่อ **gated** daemon เริ่มต้นทำงาน ซึ่งลงทะเบียนแผนผังตัวแปร *ipRouteTable* MIB ที่มีเอเจนต์ **snmpd** หาก **gated** daemon ถูกตั้งค่าไว้สำหรับ **EGP** ดังนั้น **gated** daemon ยังลงทะเบียนแผนผังตัวแปร **EGP** MIB หลังจากที่การลงทะเบียนเสร็จสิ้นแล้ว ตัวจัดการ SNMP ยังสามารถทำการร้องขอไปยังเอเจนต์ **snmpd** สำหรับ *ipRouteTable* ซึ่งตัวแปร **EGP** MIB ได้รับการสนับสนุนโดยเอเจนต์โฮสต์ **gated** daemon นี้ เมื่อ **gated** daemon กำลังรันอยู่ ข้อมูลการเราต์ MIB ทั้งหมดจะถูกขอรับโดยใช้ **gated** daemon ในกรณีนี้ การตั้งค่าคำร้องขอไปเป็น *ipRouteTable* ไม่ได้รับอนุญาต

การสื่อสาร SMUX ระหว่าง **gated** daemon และ **snmpd** daemon เข้าแทนที่ผ่าน Transmission Control Protocol (TCP) พอร์ต 199 ที่รู้จักเป็นอย่างดี หาก **gated** daemon ยกเลิก **snmpd** ไม่ได้ลงทะเบียนแผนผัง **gated** daemon ที่ลงทะเบียนไว้ก่อนหน้านี้ หาก **gated** daemon ถูกเริ่มต้นก่อน **snmpd** daemon แล้ว **gated** daemon ตรวจสอบ **snmpd** daemon เป็นระยะๆ จนกว่าการเชื่อมโยง SMUX สามารถถูกสร้างขึ้นได้

หากต้องการตั้งค่าเอเจนต์ **snmpd** เพื่อจดจำ และอนุญาตให้การเชื่อมโยง SMUX กับไคลเอ็นต์ **gated** daemon แล้ว ผู้ใช้ต้องเพิ่มรายการ SMUX ไปยังไฟล์ `/etc/snmpd.conf` ตัวระบุอ็อบเจ็กต์ไคลเอ็นต์และรหัสผ่านที่ระบุในรายการ SMUX นี้ สำหรับ **gated** daemon ต้องตรงกับที่ระบุอยู่ในไฟล์ `/etc/snmpd.peers`

เอเจนต์ **snmpd** สนับสนุนชุดคำร้องขอสำหรับตัวแปร MIB I และ MIB II แบบอ่าน-เขียนต่อไปนี้:

sysContact

identification เชิงข้อความของบุคคลผู้ติดต่อสำหรับเอเจนต์โฮสต์นี้ ข้อความนี้สอดแทรกชื่อของบุคคลนี้และวิธีการติดต่อกับบุคคลนี้ : ตัวอย่างเช่น ″Bob Smith, 555-5555, ext 5″ ค่าถูกจำกัดความยาวอยู่ที่ 256 ตัวอักษร สำหรับการตั้งค่าคำร้องขอ หากสตริงสำหรับตัวแปร MIB นี้ มีความยาวมากกว่า 256 ตัวอักษร เอเจนต์ **snmpd** จะส่งคืนข้อผิดพลาด *badValue* และการตั้งค่าการดำเนินการ ไม่ได้ถูกดำเนินการ ค่าเริ่มต้นของ *sysContact* ถูกกำหนดไว้ใน `/etc.snmp.conf` หากไม่ได้กำหนดไว้ ค่าจะเป็นสตริง null

อินสแตนซ์	ค่า	แอ็คชัน
0	″DefaultKeyBindings″	ตัวแปร MIB ถูกตั้งค่าเป็น ″string″

sysName

ชื่อโฮสต์สำหรับเอเจนต์โฮสต์นี้ โดยทั่วไป นี่คือชื่อโดเมนที่ผ่านการรับรอง โดยสมบูรณ์ของโหนด ค่าถูกจำกัดความยาวอยู่ที่ 256 ตัวอักษร สำหรับการตั้งค่าคำร้อง หากสตริงสำหรับตัวแปร MIB นี้มีความยาวมากกว่า 256 ตัวอักษร เอเจนต์ **snmpd** จะส่งคืนข้อผิดพลาด *badValue* และการตั้งค่าการดำเนินการ ไม่ได้ถูกดำเนินการ

อินสแตนซ์	ค่า	แอ็คชัน
0	″DefaultKeyBindings″	ตัวแปร MIB ถูกตั้งค่าเป็น ″string″

sysLocation

สตริงเชิงข้อความกล่าวถึงตำแหน่งฟิสิคัลของเครื่อง ที่เอเจนต์ **snmpd** นี้ตั้งอยู่: ตัวอย่างเช่น ″Austin site, building 802, lab 3C-23″ ค่าถูกจำกัดความยาวอยู่ที่ 256 ตัวอักษร สำหรับการตั้งค่าคำร้อง หากสตริงสำหรับตัวแปร MIB นี้มีความยาวมากกว่า 256 ตัวอักษร เอเจนต์ **snmpd** จะส่งคืนข้อผิดพลาด *badValue* และการตั้งค่าการดำเนินการ ไม่ได้ถูกดำเนินการ ค่าเริ่มต้นของ *sysLocation* ถูกกำหนดอยู่ใน `/etc/snmp.conf` หากไม่ได้กำหนดไว้ ค่าจะเป็นสตริง null

อินสแตนซ์	ค่า	แอ็คชัน
0	″DefaultKeyBindings″	ตัวแปร MIB ถูกตั้งค่าเป็น ″string″

ifAdminStatus

สถานะที่ต้องการของอินเตอร์เฟสอะแด็ปเตอร์บนโฮสต์ของเอเจนต์ สถานะที่สนับสนุน คือ ขึ้นและลง สถานะที่สามารถตั้งค่าเพื่อการทดสอบแต่การดำเนินการ ไม่กระทบกับสถานะการดำเนินการของอินเตอร์เฟส

อินสแตนซ์	ค่า	แอ็คชัน
f	1	อินเตอร์เฟสอะแด็ปเตอร์ที่มี **ifIndex f** ถูกเปิดใช้งาน

หมายเหตุ: มาความเป็นไปได้ที่ค่า *ifAdminStatus* สามารถตั้งค่าขึ้นหรือลงได้ ซึ่งการเปลี่ยนแปลงการดำเนินการจริงของอินเตอร์เฟส จะล้มเหลว ในบางกรณี คำร้องขอรับของ *ifAdminStatus* อาจส่งผลต่อ *up* ขณะที่ *ifOperStatus* สำหรับอินเตอร์เฟสนั้นที่อาจมีผลต่อ *down* หากมีบางสถานการณ์เกิดขึ้น ผู้ดูแลเน็ตเวิร์กจะออกคำร้องขอการตั้งค่าเพื่อตั้งค่า *ifAdminStatus* ให้เป็น up เพื่อพยายามเปลี่ยนแปลงการดำเนินการอีกครั้ง

atPhysAddress

ส่วนของแอดเดรสของฮาร์ดแวร์ของการเชื่อมตารางแอดเดรสบนเอเจนต์โฮสต์ (รายการในตาราง Address Resolution Protocol) ซึ่งเหมือนกับตัวแปร MIB ที่เป็น *ipNetToMediaPhysAddress*

อินสแตนซ์	ค่า	แอ็คชัน
f.1.n.n.n.n	hh:hh:hh:hh:hh:hh	สำหรับอินเตอร์เฟสที่มี **ifIndex f** การเชื่อมตาราง ARP ที่มีอยู่ใดๆ สำหรับ IP แอดเดรส n.n.n.n ถูกแทนที่ด้วยการเชื่อม (n.n.n.n, hh:hh:hh:hh:hh:hh) หากการเชื่อมไม่มีอยู่ การเชื่อมใหม่จะถูกเพิ่มไว้ hh:hh:hh:hh:hh:hh คือแอดเดรสของฮาร์ดแวร์ที่มีดิจิตเลขฐานสิบหกสิบสองตัว

atN0etAddress

IP แอดเดรสที่สอดคล้องกับแอดเรดสของฮาร์ดแวร์และฟิสิคัลแอดเดรสที่ระบุไว้ใน *atPhysAddress* ซึ่งเหมือนกับตัวแปร MIB ที่เป็น *ipNetToMediaNetAddress*

อินสแตนซ์	ค่า	แอ็คชัน
f.1.n.n.n.n	m.m.m.m	สำหรับอินเตอร์เฟสที่มี **ifIndex f** รายการตาราง ARP ที่มีอยู่สำหรับ IP แอดเดรส n.n.n.n ถูกแทนที่ด้วย IP แอดเดรส m.m.m.m

ipForwarding

บ่งชี้ว่า เอเจนต์โฮสต์นี้กำลังส่งต่อดาตาแกรม

ตารางที่ 89. ipforwarding

อินสแตนซ์	ค่า	แอ็คชัน
0	1	หากเอเจนต์โฮสต์มีมากกว่าหนึ่งอินเตอร์เฟสที่แอ็คทีฟ ดังนั้น เคอร์เนล **TCP/IP** จะถูกตั้งค่าเพื่อส่งต่อแพ็กเก็ต หากเอเจนต์โฮสต์มีเพียงอินเตอร์เฟสเดียวที่แอ็คทีฟ คำร้องขอการตั้งค่าจะล้มเหลว
0	2	เคอร์เนล **TCP/IP** บนเอเจนต์โฮสต์ถูกกำหนดคอนฟิกเพื่อไม่ให้ ส่งต่อแพ็กเก็ต

ipDefaultTTL

ค่า time-to-live (TTL) ดีฟอลต์ที่แทรกอยู่ในส่วนหัวของ IP ของดาตาแกรม จะถูกสร้างขึ้นโดยเอเจนต์โฮสต์

อินสแตนซ์	ค่า	แอ็คชัน
0	n	ค่า time-to-live ดีฟอลต์ที่ถูกใช้โดยส่วนสนับสนุน IP โปรโตคอล ถูกตั้งค่าเป็นเลขจำนวนเต็ม n

ipRouteDest

IP แอดเดรสปลายทางของเราต์ในตารางเราต์

อินสแตนซ์	ค่า	แอ็คชัน
n.n.n.n	m.m.m.m	เราต์ปลายทางสำหรับเราต์ n.n.n.n ถูกตั้งค่าเป็น IP แอดเดรส m.m.m.m

ipRouteNextHop

เกตเวย์ที่ IP แอดเดรสเป้าหมายสามารถเข้าถึงได้จาก เอเจนต์โฮสต์ (รายการในตารางเราต์)

อินสแตนซ์	ค่า	แอ็คชัน
n.n.n.n	m.m.m.m	รายการตารางเราต์เพื่อเข้าถึงเน็ตเวิร์ก n.n.n.n โดยใช้เกตเวย์ m.m.m.m ถูกเพิ่มให้กับตารางเราต์ ส่วนของโฮสต์ของ IP แอดเดรส n.n.n.n ต้องเป็น 0 เพื่อบ่งชี้เน็ตเวิร์กแอดเดรส

ipRouteType

สถานะของรายการตารางเราต์บนเอเจนต์โฮสต์ (ถูกใช้เพื่อลบรายการ)

อินสแตนซ์	ค่า	แอ็คชัน
h.h.h.h	1	เราต์ไปยัง IP แอดเดรสของโฮสต์ h.h.h.h ถูกลบทิ้ง
n.n.n.n	2	เราต์ใดๆ ไปยัง IP แอดเดรสของโฮสต์ n.n.n.n ถูกลบทิ้ง

ipNetToMediaPhysAddress

ส่วนของแอดเดรสของฮาร์ดแวร์ของการเชื่อมตารางแอดเดรสบนเอเจนต์โฮสต์ (รายการในตาราง ARP) ซึ่งเหมือนกับตัวแปร MIB ที่เป็น *atPhysAddress*

อินสแตนซ์	ค่า	แอ็คชัน
f.1.n.n.n.n	hh:hh:hh:hh:hh:hh	สำหรับอินเตอร์เฟสที่มี **ifIndex f** การเชื่อมตาราง ARP ที่มีอยู่ใดๆ สำหรับ IP แอดเดรส n.n.n.n ถูกแทนที่ด้วยการเชื่อม (n.n.n.n, hh:hh:hh:hh:hh:hh) หากการเชื่อมไม่มีอยู่ การเชื่อมใหม่จะถูกเพิ่มไว้ hh:hh:hh:hh:hh:hh is คือแอดเดรสของฮาร์ดแวร์ที่มีดิจิตเลขฐานสิบหก 12 ตัว

ipNetToMediaNetAddress

IP แอดเดรสที่สอดคล้องกับแอดเดรสของฮาร์ดแวร์และฟิสิคัลแอดเดรสที่ระบุไว้ใน *ipNetToMediaPhysAddress* ซึ่งเหมือนกับตัวแปร MIB ที่เป็น *atNetAddress*

อินสแตนซ์	ค่า	แอ็คชัน
f.1.n.n.n.n	m.m.m.m	สำหรับอินเตอร์เฟสที่มี **ifIndex f** รายการตาราง ARP ที่มีอยู่สำหรับ IP แอดเดรส n.n.n.n ถูกแทนที่ด้วย IP แอดเดรส m.m.m.m

ipNetToMediaType

ชนิดของการแม็พจาก IP แอดเดรสกับฟิสิคัลแอดเดรส

อินสแตนซ์	ค่า	แอ็คชัน
f.1.n.n.n.n	1	สำหรับอินเตอร์เฟสที่มี **ifIndex f** ตัวอย่างเช่น การเชื่อม ARP จาก IP แอดเดรสกับฟิสิคัลแอดเดรส จะมีชนิดของการแม็พเป็น 1 หรือเป็นอย่างอื่น
f.1.n.n.n.n	2	สำหรับอินเตอร์เฟสที่มี **ifIndex f** ตัวอย่างเช่น การเชื่อม ARP จาก IP แอดเดรสกับฟิสิคัลแอดเดรส จะมีชนิดของการแม็พเป็น 2 หรือไม่ถูกต้อง เนื่องจากมีผลกระทบข้างเคียงเกิดขึ้น รายการที่สอดคล้องกันใน **ipNetMediaTable** อาจไม่ถูกต้อง อินเตอร์เฟสไม่ได้ถูกเชื่อมโยงจากรายการ **ipNetToMediaTable** นี้
f.1.n.n.n.n	3	สำหรับอินเตอร์เฟสที่มี **ifIndex f** ตัวอย่างเช่น การเชื่อม ARP จาก IP แอดเดรสกับฟิสิคัลแอดเดรส จะมีชนิดของการแม็พเป็น 3 หรือเป็นแบบไดนามิก
f.1.n.n.n.n	4	สำหรับอินเตอร์เฟสที่มี **ifIndex f** ตัวอย่างเช่น การเชื่อม ARP จาก IP แอดเดรสกับฟิสิคัลแอดเดรส จะมีชนิดของการแม็พเป็น 4 หรือเป็นแบบสแตติก

snmpEnableAuthenTraps

บ่งชี้ว่า เอเจนต์ **snmpd** ถูกตั้งค่าไว้เพื่อสร้างแทร็บ *authenticationFailure*

อินสแตนซ์	ค่า	แอ็คชัน
0	1	เอเจนต์ **snmpd** จะสร้างพิสูจน์ตัวตนแทร็บ ที่ล้มเหลว
0	2	เอเจนต์ **snmpd** จะไม่สร้างพิสูจน์ตัวตนแทร็บ ที่ล้มเหลว

smuxPstatus

สถานะของเพียร์สำหรับโปรโตคอล SMUX (ใช้เพื่อลบเพียร์ SMUX)

อินสแตนซ์	ค่า	แอ็คชัน
n	1	เอเจนต์ **snmpd** ไม่ได้ทำสิ่งใด
n	2	เอเจนต์ **snmpd** หยุดการสื่อสารกับเพียร์ SMUX n

smuxTstatus

สถานะของแผนผัง SMUX MIB (ใช้เพื่อลบแผนผังที่ MIB mount)

อินสแตนซ์	ค่า	แอ็คชัน
l.m.m.m._ _ _ .p	1	เอเจนต์ **snmpd** ไม่ได้ทำสิ่งใด
l.m.m.m._ _ _ .p	2	Unmount SMUX ที่เป็นการ mount ของแผนผัง MIB m.m.m... โดยที่ *l* คือความยาวของอินสแตนซ์ของแผนผัง MIB และ *p* คือ smuxTpriority

ตัวแปรต่อไปนี้คือตัวแปรที่สามารถตั้งค่าได้ ตามที่กำหนดไว้ใน RFC 1229 **snmpd** daemon อนุญาตให้ผู้ใช้ตั้งค่าตัวแปรเหล่านี้ อุปกรณ์ที่อยู่ในตำแหน่งที่ต่ำกว่าอาจไม่อนุญาตให้ใช้การตั้งค่าของตัวแปรบางตัว ตรวจสอบกับ อุปกรณ์แต่ละตัวเพื่อดูส่วนที่สนับสนุนและส่วนที่ไม่สนับสนุน

ifExtnsPromiscuous

สถานะของโหมดที่มีหลายองค์ประกอบบนอุปกรณ์ที่กำหนดไว้ซึ่งถูกใช้เพื่อเปิดใช้งาน และปิดใช้งานโหมดที่มีหลายองค์ประกอบบนอุปกรณ์ที่กำหนดไว้ แอ็คชัน **snmpd** คือแอ็คชันสุดท้ายและสมบูรณ์ เมื่อ **snmpd** ถูกกล่าวให้ปิดโหมดที่มีหลายองค์ประกอบจะถูกปิดอย่างสมบูรณ์โดยไม่พิจารณาถึงแอ็พพลิเคชันใดๆ บนเครื่อง

อินสแตนซ์	ค่า	แอ็คชัน
n	1	เปิดโหมดที่มีหลายองค์ประกอบสำหรับอุปกรณ์ n
n	2	ปิดโหมดที่มีหลายองค์ประกอบสำหรับอุปกรณ์ n

ifExtnsTestType

ตัวแปรที่เริ่มต้นการทดสอบ เมื่อตัวแปรนี้ถูกตั้งค่า การทดสอบที่เหมาะสม ถูกรันสำหรับอุปกรณ์นั้น ตัวระบุอ็อบเจ็กต์คือค่าของตัวแปร ค่าที่ระบุไว้ถูกพึ่งพาบนชนิดของอุปกรณ์ และทดสอบว่า กำลังรันอยู่ ในปัจจุบันนี้ เฉพาะการทดสอบที่นิยามไว้ว่า **snmpd** รู้ว่าต้องรันคือการทดสอบ testFullDuplexLoopBack

อินสแตนซ์	ค่า	แอ็คชัน
n	oid	เริ่มต้นการทดสอบที่ระบุโดย oid

ifExtnsRcvAddrStatus

ตัวแปรสถานะของแอดเดรส เมื่อตัวแปรนี้ถูกตั้งค่าไว้ แอดเดรสที่ระบุไว้อยู่ในการมีอยู่กับระดับของช่วงระยะเวลาที่เหมาะสม **snmpd** อนุญาตให้การตั้งค่าของแอดเดรสชั่วคราว เนื่องจากไม่สามารถตั้งค่าอุปกรณ์เร็กคอร์ด Object Data Manager (ODM) และไม่อนุญาตให้ตั้งค่ามัลติคาสต์ หรือแอดเดรสการกระจายสัญญาณ

อินสแตนซ์	ค่า	แอ็คชัน
n.m.m.m.m.m.m	1	เพิ่มแอดเดรสเป็นบางสิ่งที่นอกเหนือจากแอดเดรสชั่วคราวหรือถาวร
n.m.m.m.m.m.m	2	ลบแอดเดรสจากการใช้งาน
n.m.m.m.m.m.m	3	เพิ่มแอดเดรสเป็นแอดเดรสชั่วคราว
n.m.m.m.m.m.m	4	เพิ่มแอดเดรสเป็นแอดเดรสถาวร

ตัวแปรที่แสดงด้านล่างคือตัวแปรที่สามารถตั้งค่าได้ตามที่กำหนดไว้ใน RFC 1231 **snmpd** daemon อนุญาตให้ผู้ใช้ตั้งค่าตัวแปรเหล่านี้ อุปกรณ์ที่อยู่ในตำแหน่งที่ต่ำกว่าอาจไม่อนุญาตให้ใช้การตั้งค่าของตัวแปรบางตัว คุณควรตรวจสอบ กับอุปกรณ์เพื่อดูส่วนที่ได้รับการสนับสนุน

dot5Commands

คำสั่งที่อุปกรณ์โทเค็นริงควรถูกดำเนินการ

อินสแตนซ์	ค่า	แอ็คชัน
n	1	ไม่ดำเนินการ ส่งคืนแล้ว
n	2	แจ้งให้อุปกรณ์โทเค็นริงเปิด
n	3	แจ้งให้โทเค็นริงรีเซ็ต
n	4	แจ้งให้อุปกรณ์โทเค็นริงปิด

dot5RingSpeed

ความเร็ววงแหวนปัจจุบันหรือแบนด์วิดธ์

อินสแตนซ์	ค่า	แอ็คชัน
n	1	ความเร็วที่ไม่รู้จัก
n	2	ความเร็ววงแหวน 1 เมกะบิต
n	3	ความเร็ววงแหวน 4 เมกะบิต
n	4	ความเร็ววงแหวน 16 เมกะบิต

dot5ActMonParticipate

อ็อบเจ็กต์ที่ระบุไม่ว่าอุปกรณ์ทำงานร่วมกันในกระบวนการเลือกมอนิเตอร์ แอ็คทีฟหรือไม่ก็ตาม

อินสแตนซ์	ค่า	แอ็คชัน
n	1	ทำงานร่วมกัน
n	2	ไม่ทำงานร่วมกัน

dot5Functional

ตัวพรางด้านการทำงานที่อนุญาตให้อุปกรณ์โทเค็นริงระบุแอดเดรสที่ได้รับ กรอบ

อินสแตนซ์	ค่า	แอ็คชัน
n	m.m.m.m.m.m	ตัวพรางด้านการทำงานที่ต้องถูกตั้งค่า

ตัวแปรการจัดการกับตัวจับเวลาแบบซับซ้อนต่อไปนี้ถูกนิยามอยู่ใน RFC เป็นแบบอ่านอย่างเดียวแต่คุณถูกสนับสนุนเพื่อทำให้การจัดการแบบอ่าน–เขียน ตรวจทาน RFC เพื่อรับความเข้าใจของการโต้ตอบแบบเต็ม **snmpd** อนุญาตให้คำร้องขอเพื่อตั้งค่า แต่อุปกรณ์ไม่ใช่ ตรวจสอบไดรเวอร์อุปกรณ์ เอกสารคู่มือสำหรับข้อมูลเพิ่มเติม ตัวแปรคือ:

- dot5TimerReturnRepeat
- dot5TimerHolding
- dot5TimerQueuePDU
- dot5TimerValidTransmit
- dot5TimerNoToken

- dot5TimerActiveMon
- dot5TimerStandbyMon
- dot5TimerErrorReport
- dot5TimerBeaconTransmit
- dot5TimerBeaconReceive

SNMP daemon อนุญาตให้ผู้ใช้เพื่อตั้งค่าตัวแปรต่อไปนี้ daemon ใช้โปรโตคอล FDDI Station Management (SMT) 7.2 มาตรฐานเพื่อขอรับข้อมูล และถูกพิจารณาที่ระดับของไมโครโค้ด ตรวจสอบไมโครโค้ดบนเอกสารคู่มือ FDDI เพื่อตรวจสอบว่าไมโครโค้ด SMT 7.2 ถูกใช้

fddimibSMTUserData

ตัวแปรที่พัก 32 ไบต์ของข้อมูลผู้ใช้

อินสแตนซ์	ค่า	แอ็คชัน
n	string	เก็บรายละเอียดผู้ใช้ขนาด 32 ไบต์

fddimibSMTConfigPolicy

สถานะของนโยบายคอนฟิกูเรชัน โดยเฉพาะการใช้นโยบาย การพักเป็นพิเศษ

อินสแตนซ์	ค่า	แอ็คชัน
n	0	ห้ามใช้นโยบายการพัก
n	1	ใช้นโยบายการพัก

fddimibSMTConnectionPolicy

สถานะของนโยบายการเชื่อมต่อในโหนด FDDI โปรดดู RFC 1512 สำหรับข้อมูลเพิ่มเติมเกี่ยวกับค่าที่สามารถตั้งค่าเฉพาะ

อินสแตนซ์	ค่า	แอ็คชัน
n	k	นิยามนโยบายการเชื่อมต่อ

fddimibSMTTNotify

ตัวจับเวลาที่แสดงอยู่ในวินาทีถูกใช้ในโปรโตคอล Neighbor Notification ซึ่งมีช่วงเวลา 2 วินาทีถึง 30 วินาที และค่าดีฟอลต์คือ 30 วินาที

อินสแตนซ์	ค่า	แอ็คชัน
n	k	นิยามค่าตัวจับเวลา

fddimibSMTStatRptPolicy

สถานะของการสร้างกรอบการรายงานสถานะ

อินสแตนซ์	ค่า	แอ็คชัน
n	1	บ่งชี้ว่าโหนดสร้างกรอบการรายงานสถานะสำหรับเหตุการณ์ถูกนำมาปฏิบัติ
n	2	บ่งชี้ว่าโหนดไม่ได้สร้างกรอบการรายงานสถานะ

fddimibSMTTraceMaxExpiration

ตัวแปรนี้นิยามค่าการหมดอายุตัวจับเวลาสูงสุดสำหรับการติดตาม

อินสแตนซ์	ค่า	แอ็คชัน
n	k	นิยามการหมดอายุตัวจับเวลาสูงสุดในหน่วยมิลลิวินาที

fddimibSMTStationAction

ตัวแปรนี้เป็นสาเหตุที่เอ็นทิตี SMT เพื่อใช้แอ็คชันที่ระบุเฉพาะ โปรดดู RFC เพื่อขอรับข้อมูลที่ระบุเฉพาะเกี่ยวกับตัวแปรนี้

อินสแตนซ์	ค่า	แอ็คชัน
n	k	นิยามแอ็คชันบนเอ็นทิตี SMT ค่าของช่วงตั้งแต่ 1 ถึง 8

fddimibMACRequestedPaths

นิยามพาธ medium access control (MAC) ควรถูกแทรก

อินสแตนซ์	ค่า	แอ็คชัน
n.n	k	นิยามพาธที่ร้องขอสำหรับ MAC

fddimibMACFrameErrorThreshold

Threshold เมื่อรายงานสถานะ MAC ถูกสร้างขึ้น นิยามจำนวนข้อผิดพลาด ที่ต้องเกิดขึ้นก่อนที่รายงานจะถูกสร้างขึ้น

อินสแตนซ์	ค่า	แอ็คชัน
n.n	k	นิยามจำนวนข้อผิดพลาดที่ต้องถูกสังเกตก่อนที่รายงานสถานะ MAC จะถูกสร้างขึ้น

fddimibMACMAUnitdataEnable

ตัวแปรนี้พิจารณาค่าของแฟล็ก **MA_UNITDATA_Enable** ใน RMT ค่าดีฟอลต์และค่าเริ่มต้นของแฟล็กนี้เป็นจริง (1)

อินสแตนซ์	ค่า	แอ็คชัน
n.n	1	ทำเครื่องหมายแฟล็ก MA_UNITDATA_Enable ว่าเป็นจริง
n.n	2	ทำเครื่องหมายแฟล็ก MA_UNITDATA_Enable ว่าเป็นเท็จ

fddimibMACNotCopiedThreshold

threshold สำหรับการพิจารณาเมื่อรายงานเงื่อนไข MAC ถูกสร้าง

อินสแตนซ์	ค่า	แอ็คชัน
n.n	k	นิยามจำนวนของข้อผิดพลาดที่ต้องถูกสังเกตก่อนที่รายงานเงื่อนไข MAC ถูกสร้างขึ้น

ตัวแปรสามตัวต่อไปนี้คือตัวแปรจับเวลาที่เป็นแบบโต้ตอบ ท่ามกลางตัวแปรเหล่านี้ ก่อนที่เปลี่ยนแปลงตัวแปรเหล่านี้ใดๆ คุณควรมีความเข้าใจเป็นอย่างดี เกี่ยวกับความหมายเป็นนิยามใน **RFC 1512**

- fddimibPATHTVXLowerBound
- fddimibPATHTMaxLowerBound
- fddimibPATHMaxTReq

fddimibPORTConnectionPolicies

ระบุนโยบายการเชื่อมต่อสำหรับพอร์ตที่ระบุ

อินสแตนซ์	ค่า	แอ็คชัน
n.n	k	นิยามนโยบายการเชื่อมต่อสำหรับพอร์ตที่ระบุไว้

fddimibPORTRequestedPaths

ตัวแปรนี้คือรายการของพาธที่อนุญาตที่อิลิเมนต์รายการแต่ละตัว นิยามพอร์ตที่มีพาธที่ได้รับอนุญาต octet แรกสอดคล้องกับ `none' octet ที่สองสอดคล้องกับ `tree' และ octet ที่สามสอดคล้องกับ `peer'

อินสแตนซ์	ค่า	การดำเนินการ
n.n	ccc	นิยามพาธของพอร์ต

fddimibPORTLerCutoff

อัตราความผิดพลาดของลิงก์ประมาณการที่การเชื่อมต่อลิงก์ขาดสัญญาณ ซึ่งเป็นช่วงจาก 10**-4 ถึง 10**-15 และถูกรายงานเป็นค่าสัมบูรณ์ของ ลอการิทึมฐาน 10 (ค่าดีฟอลต์คือ 7)

ไอเท็ม	คำอธิบาย	
อินสแตนซ์	ค่า	แอ็คชัน
n.n	k	นิยามพอร์ต LerCutoff

fddimibPORTLerAlarm

อัตราความผิดพลาดของลิงก์ประมาณการที่การเชื่อมต่อลิงก์ที่สร้าง เสียงเตือน ซึ่งเป็นช่วงจาก 10**-4 ถึง 10**-15 และถูกรายงานเป็นค่าสัมบูรณ์ของ ลอการิทึมฐาน 10 ของการประมาณการ (ค่าดีฟอลต์คือ 8)

อินสแตนซ์	ค่า	แอ็คชัน
n.n	k	นิยามพอร์ต LerAlarm

fddimibPORTAction

ตัวแปรนี้เป็นสาเหตุทำให้พอร์ตใช้แอ็คชันที่ระบุเฉพาะ โปรดดู RFC เพื่อขอรับข้อมูลที่ระบุเฉพาะเกี่ยวกับตัวแปรนี้

อินสแตนซ์	ค่า	แอ็คชัน
n	k	นิยามแอ็คชันบนพอร์ตที่นิยามไว้ ค่าของช่วงจาก 1 ถึง 6

หมายเหตุ: RFC 1213 อธิบายถึงตัวแปรทั้งหมดในตาราง *atEntry* และ *ipNetToMediaEntry* แบบอ่าน-เขียน การตั้งค่าส่วนสนับสนุนถูกใช้เฉพาะสำหรับตัวแปร *atEntry* ตัวแปร *atPhysAddress* และตัวแปร *atNetAddress* และตัวแปร *ipNetToMediaEntry*, *ipNetToMediaPhysAddress*, *ipNetToMediaNetAddress* และ *ipNetToMediaType* หากต้องการยอมรับ คำร้องขอการตั้งค่าที่อาจระบุแอ็ตทริบิวต์ที่ไม่ได้รับการสนับสุนนที่เหลืออยู่ในตารางทั้งสองตาราง คำร้องขอการตั้งค่าสำหรับตัวแปรที่เหลืออยู่จะถูกยอมรับใน *atIfIndex* และ *ipNetToMediaIfIndex* การตอบกลับ ข้อผิดพลาดถูกส่งคืนไปยังตัวสร้างคำร้องขอการตั้งค่า แต่คำร้องขอการรับลำดับถัดมา จะแสดงค่าเริ่มต้นที่เก็บไว้

ในตาราง *ipRouteEntry* RFC 1213 อธิบายตัวแปรทั้งหมดยกเว้น *ipRouteProto* ที่เป็นอ่าน-เขียน ตามที่ได้กล่าวข้างต้น ส่วนสนับสนุนการตั้งค่าถูกนำมาใช้เฉพาะสำหรับตัวแปร *ipRouteDest*, *ipRouteNextHop* และ *ipRouteType* หากต้องการยอมรับคำร้องขอการตั้งค่า ที่อาจระบุแอ็ตทริบิวต์เราต์ที่ไม่ได้รับการสนับสนุนต่างๆ คำร้องขอการตั้งค่าสำหรับตัวแปรที่เหลืออยู่ในตาราง *ipRouteEntry* ถูกยอมรับ: *ipRouteIfIndex*, *ipRouteMetric1*, *ipRouteMetric2*, *ipRouteMetric3*, *ipRouteMetric4*, *ipRouteMetric5*, *ipRouteAge* และ *ipRouteMask* การตอบกลับ ข้อผิดพลาดถูกส่งคืนไปยังตัวสร้างคำร้องขอการตั้งค่า แต่คำร้องขอการรับลำดับถัดมา จะแสดงค่าเริ่มต้นที่เก็บไว้ **snmpd** daemon ไม่ได้ประสานการเราต์กับ **routed** daemon หาก **gated** daemon กำลังรันและได้ลงทะเบียน *ipRouteTable* ด้วย **snmpd** daemon คำร้องขอการตั้งค่ากับ *ipRouteTable* ไม่ได้รับอนุญาต

RFC 1229 อธิบายถึงตัวแปรที่สามารถตั้งค่าที่ **snmpd** ได้รับอนุญาต โปรดดูรายการก่อนหน้านี้สำหรับความเบี่ยงเบนจริง

ตัวอย่างต่อไปนี้ใช้คำสั่ง **snmpinfo** ซึ่งถูก สมมติขึ้นว่า ชื่อ community **snmpinfo** ที่เป็นค่าดีฟอลต์ พับลิก มีการเข้าถึงการอ่าน-เขียนสำหรับแผนผังย่อย MIB ตามลำดับ

```
snmpinfo -m set sysContact.0="Primary contact: Bob Smith, office phone: 555-5555,
beeper: 9-123-4567. Secondary contact: John Harris, phone: 555-1234."
```

คำสั่งนี้ตั้งค่า `sysContact.0` ไปเป็นสตริง ที่ระบุไว้ หากรายการสำหรับ `sysContact.0` มีอยู่แล้ว ซึ่งถูกแทนที่

```
snmpinfo -m set sysName.0="bears.austin.ibm.com"
```

คำสั่งนี้ตั้งค่า `sysName.0` ไปเป็น สตริงที่ระบุ หากรายการสำหรับ `sysName.0` มีอยู่แล้ว ซึ่งถูกแทนที่

```
snmpinfo -m set sysLocation.0="Austin site, building 802, lab 3C-23, southeast
corner of the room."
```

คำสั่งนี้ตั้งค่า `sysLocation.0` ไปเป็นสตริง ที่ระบุไว้ หากรายการสำหรับ `sysLocation.0` มีอยู่แล้ว ซึ่งถูกแทนที่

```
snmpinfo -m set ifAdminStatus.2=2
```

คำสั่งนี้ปิดใช้งานเน็ตเวิร์กอินเตอร์เฟสอะแด็ปเตอร์ซึ่งมี ifIndex 2 หากค่าที่กำหนดไว้คือ 1 อินเตอร์เฟสอะแด็ปเตอร์ถูกเปิดใช้งาน

```
snmpinfo -m set atPhysAddress.2.1.192.100.154.2=02:60:8c:2e:c2:00
snmpinfo -m set ipNetToMediaPhysAddress.2.1.192.100.154.2=02:60:8c:2e:c2:00
```

คำสั่งสองคำสั่งเหล่านี้เปลี่ยนแอดเดรสของฮาร์ดแวร์ในรายการตาราง ARP สำหรับ 192.100.154.2 ไปเป็น 02:60:8c:2e:c2:00 คำสั่งสองคำสั่งเหล่านี้มีผลต่อรายการตาราง ARP เหมือนกัน ตัวแปร MIB *atPhysAddress* คือตัวแปรที่ไม่คัดค้านและถูกแทนที่ด้วยตัวแปร MIB *ipNetToMediaPhysAddress* ดังนั้น *atPhysAddress* และ *ipNetToMediaPhysAddress* เข้าถึงโครงสร้างเดียวกันในตาราง TCP/IP kernel ARP

```
snmpinfo -m set atNetAddress.2.1.192.100.154.2=192.100.154.3
snmpinfo -m set ipNetToMediaNetAddress.2.1.192.100.154.2=192.100.154.3
```

คำสั่งเหล่านี้เปลี่ยนแปลง IP แอดเดรสในรายการตาราง ARP สำหรับ 192.100.154.2 กับ 192.100.154.3 คำสั่งสองคำสั่งเหล่านี้มีผลต่อรายการตาราง ARP เหมือนกัน ตัวแปร MIB *atNetAddress* คือตัวแปรที่ไม่คัดค้านและถูกแทนที่ด้วยตัวแปร MIB *ipNetToMediaNetAddress* ดังนั้น *atNetAddress* และ *ipNetToMediaNetAddress* เข้าถึงโครงสร้างเดียวกันในตาราง TCP/IP kernel ARP

```
snmpinfo -m set ipForwarding.0=1
```

คำสั่งนี้ตั้งค่าเคอร์เนล **TCP/IP** ดังนั้น จึงสามารถส่งต่อแพ็กเก็ต หากเอเจนต์มีอินเตอร์เฟสมากกว่าหนึ่งที่เริ่มทำงาน หากโฮสต์มีเพียงหนึ่งอินเตอร์เฟส ที่แอ็คทีฟ ดังนั้นคำร้องขอการตั้งค่าล้มเหลว และเอเจนต์ **snmpd** ส่งคืนข้อผิดพลาด *badValue*

```
snmpinfo -m set ipDefaultTTL=50
```

คำสั่งนี้อนุญาตให้ IP ดาตาแกรมที่ใช้ time-to-live (TTL) ที่เป็นค่าดีฟอลต์ เพื่อส่งผ่านได้ถึง 50 เกตเวย์ก่อนที่จะละเว้น เมื่อเกตเวย์แต่ละตัวประมวลผลดาตาแกรม เกตเวย์ลบ 1 จากฟิลด์ time-to-live นอกจากนี้ เกตเวย์แต่ละตัวลดจำนวนฟิลด์ time-to-live ด้วยจำนวนวินาที ที่ดาตาแกรมที่รอสำหรับเซอร์วิสที่เกตเวย์ก่อนที่จะส่งผ่านดาตาแกรมไปยัง ปลายทางถัดไป

```
snmpinfo -m set ipRouteDest.192.100.154.0=192.100.154.5
```

คำสั่งนี้ตั้งค่า IP แอดเดรสปลายทางของเราต์ที่เชื่อมโยงกับ 192.100.154.0 ไปเป็น IP แอดเดรส 192.100.154.5 ซึ่งสมมติเราต์ 192.100.154 ที่มีอยู่

```
snmpinfo -m set ipRouteNextHop.192.100.154.1=129.35.38.47
```

คำสั่งนี้ตั้งค่าเราต์ไปเป็นโฮสต์ 192.100.154.1 ที่ใช้เกตเวย์โฮสต์ 129.35.38.47 ซึ่งสมมติเราต์ 192.100.154.1 ที่มีอยู่

```
snmpinfo -m set ipRouteNextHop.192.100.154.0=192.100.154.7
```

คำสั่งนี้ตั้งค่าเราต์ไปเป็นเน็ตเวิร์กคลาส C 192.100.154 โดยใช้เกตเวย์โฮสต์ 192.100.154.7 ซึ่งสมมติเราต์ 192.100.154.0 ที่มีอยู่ หมายเหตุ ส่วนของโฮสต์ของแอดเดรสต้องมีค่า 0 เพื่อบ่งชี้ เน็ตเวิร์กแอดเดรส

```
snmpinfo -m set ipRouteType.192.100.154.5=2
```

คำสั่งนี้ลบเราต์ใดๆ ไปยังโฮสต์ 192.100.154.5

```
snmpinfo -m set ipRouteDest.129.35.128.1=129.35.128.1
               ipRouteType.129.35.128.1=3
            ipRouteNextHop.129.35.128.1=129.35.128.90
```

คำสั่งนี้สร้างเราต์ใหม่จากโฮสต์ 129.35.128.90 ไปยัง 129.35.128.1 เป็นเกตเวย์

```
snmpinfo -m set ipNetToMediaType.2.1.192.100.154.11=4
```

คำสั่งนี้ตั้งค่ารายการตาราง ARP สำหรับ 192.100.154.11 เป็นแบบสแตติก

```
snmpinfo -m set snmpEnableAuthenTraps=2
```

คำสั่งนี้เป็นสาเหตุทำให้เอเจนต์ **snmpd** บนโฮสต์ที่ระบุไว้ไม่ให้สร้างแทร็บ *authenticationFailure*

```
snmpinfo -m set smuxPstatus.1=2
```

คำสั่งนี้ไม่ได้ตรวจสอบความถูกต้อง SMUX เพียร์ 1 ผลลัพธ์คือ การเชื่อมต่อระหว่างเอเจนต์ **snmpd** และเพียร์ SMUX นี้ถูกยกเลิก

```
snmpinfo -m set smuxTstatus.8.1.3.6.1.2.1.4.21.0=2
```

คำสั่งนี้ไม่ได้ตรวจสอบความถูกต้องหรือลบการเมาต์ของแผนผัง SMUX 1.3.6.1.2.1.4.21 นั่นคือ ตาราง *ipRoute* หมายเลขแรกในอินสแตนซ์ บ่งชี้ถึงจำนวนของระดับในตัวระบุแผนผัง SMUX หมายเลขตัวสุดท้าย ในอินสแตนซ์บ่งชี้ถึง smuxTpriority ในตัวอย่างนี้ มี 8 ระดับในตัวระบุแผนผัง SMUX: 1.3.6.1.2.1.4.21 ระดับความสำคัญ 0 คือระดับความสำคัญสูงสุด

```
snmpinfo -m set ifExtnsPromiscuous.1=1 ifExtnsPromiscuous.2=2
```

คำสั่งนี้เปิดโหมดที่มีหลายองค์ประกอบสำหรับอุปกรณ์แรกในตารางอินเตอร์เฟส และปิดโหมดที่มีหลายองค์ประกอบในตาราง อินเตอร์เฟส

```
snmpinfo -m set ifExtnsTestType.1=testFullDuplexLoopBack
```

คำสั่งนี้เริ่มต้นการทดสอบ testFullDuplexLoopBack บนอินเตอร์เฟส 1

```
snmpinfo -m set ifExtnsRcvAddrStatus.1.129.35.128.1.3.2=2
```

คำสั่งนี้แจ้งให้อินเตอร์เฟส 1 ทราบเพื่อลบฟิสิคัลแอดเดรส 129.35.128.1.3.2 ออกจารายการของแอดเดรสที่สามารถยอมรับได้

```
snmpinfo -m set dot5Commands.1=2
```

คำสั่งนี้แจ้งให้อินเตอร์เฟสแรกเพื่อทำการเปิด

```
snmpinfo -m set dot5RingSpeed.1=2
```

คำสั่งนี้แจ้งให้อินเตอร์เฟสแรกเพื่อตั้งค่าความเร็ววงแหวนไปเป็น 1 เมกะบิต

```
snmpinfo -m set dot5ActMonParticipate.1=1
```

คำสั่งนี้แจ้งให้อินเตอร์เฟสแรกเพื่อสื่อสารในกระบวนการเลือกการมอนิเตอร์ ที่แอ็คทีฟ

```
snmpinfo -m set dot5Functional.1=255.255.255.255.255.255
```

คำสั่งนี้ตั้งค่าตัวพรางแอดเดรสด้านการทำงานเพื่ออนุญาตให้ใช้ทุกสิ่ง

```
snmpinfo -m set fddimibSMTUserData.1="Greg's Data"
```

คำสั่งนี้ตั้งค่าข้อมูลผู้ใช้บนเอ็นทิตี SMT แรกไปเป็น "Greg's Data"

```
snmpinfo -m set fddimibMACFrameErrorThreshold.1.1=345
```

คำสั่งนี้ตั้งค่า threshold สำหรับข้อผิดพลาดของกรอบไปยัง 345 บน MAC แรกของเอ็นทิตี SMT แรก

หมายเหตุ: ตัวแปรทั้งหมดอธิบายถึงการตกลงในหนึ่งในเมธอดก่อนหน้านี้ที่แสดงรายการ ที่ถูกใช้เพื่อตั้งค่าตัวแปร

โปรดดู “Address Resolution Protocol” ในหน้า 154 และ “อินเตอร์เน็ตแอดเดรส” ในหน้า 193 สำหรับข้อมูลเพิ่มเติม เกี่ยวกับโปรโตคอลและอินเตอร์เน็ตแอดเดรส

การแก้ไขปัญหา SNMP daemon

เทคนิคการแก้ปัญหาสำหรับ **SNMP** daemon สอดแทรกการยกเลิกการแก้ปัญหา ตัวแปร MIB เข้าถึงปัญหารายการ community ปัญหาเกี่ยวกับ noSuchName ดังนั้นจึงไม่มีการตอบกลับจากปัญหาเกี่ยวกับเอเจนต์ และความล้มเหลวของ daemon

ปัญหาเกี่ยวกับการยกเลิก Daemon

หากเอเจนต์ **snmpd** ไม่ได้ทำหน้าที่ตามที่คาดการณ์ไว้ ต่อไปนี้คือคำแนะนำบางข้อเพื่อช่วยในการกำหนดและแก้ไขปัญหา ซึ่งให้คำแนะนำว่า คุณต้องเริ่มต้นทำงานเอเจนต์ **snmpd** ที่มีชนิดของการบันทึกการทำงานบางอย่าง หากเรียกทำงาน **snmpd** daemon ที่เป็นสาเหตุของปัญหา มีข้อแนะนำว่า **syslogd** daemon จะตั้งค่าไว้สำหรับการบันทึกการทำงานที่สิ่งอำนวยความสะดวก daemon และระดับความสำคัญ DEBUG See the **snmpd** command in *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 5* and the **snmpd.conf** file ใน *ข้อมูลอ้างอิงไฟล์*for more information on **snmpd** logging.

หาก **snmpd** daemon ยกเลิกเมื่อใดก็ตามที่เรียกใช้ ซึ่งต่อไปนี้คือเหตุผลที่เป็นไปได้สำหรับความล้มเหลวและโซลูชันที่อาจเป็นไปได้:

- เหตุผล **snmpd** daemon ที่ยกเลิกแล้วจะถูกบันทึกการทำงาน ในไฟล์บันทึกการทำงาน **snmpd** หรือไฟล์บันทึกการทำงาน **syslogd** ที่ตั้งค่าไว้ โปรดตรวจสอบไฟล์บันทึกการทำงานเพื่อดูข้อความแสดงความผิดพลาด **FATAL**

  Solution: แก้ไขปัญหาและรีสตาร์ท **snmpd** daemon
- การใช้งานบรรทัดรับคำสั่ง **snmpd** ไม่ถูกแก้ไข หากคำสั่ง **snmpd** ถูกเรียกใช้โดยไม่มี System Resource Controller (SRC) ข้อความการใช้งานที่จำเป็นต้องมี ต้องถูก echo กับหน้าจอ หาก **snmpd** daemon ถูกเรียกใช้งานภายใต้การควบคุม SRC แล้ว ข้อความการใช้งานไม่ถูก echo กับหน้าจอ ตรวจสอบไฟล์บันทึกการทำงานเพื่อดูข้อความการใช้งาน

  Solution: เรียกใช้คำสั่ง **snmpd** ด้วยคำสั่งการใช้งานที่ถูกต้อง
- **snmpd** daemon ต้องถูกเรียกใช้งานโดยผู้ใช้ root

  Solution: สับเปลี่ยนไปยังผู้ใช้ root และรีสตาร์ท **snmpd** daemon
- ไฟล์ **snmpd.conf** ต้องเป็นเจ้าของโดยผู้ใช้ root เอเจนต์ **snmpd** ตรวจสอบความเป็นเจ้าของไฟล์ คอนฟิกูเรชัน หากไฟล์ไม่ได้เป็นเจ้าของโดยผู้ใช้ root เอเจนต์ **snmpd** ยกเลิกพร้อมกับข้อความรุนแรง

  Solution: ตรวจสอบให้แน่ใจว่า คุณคือผู้ใช้ root เปลี่ยนความเป็นสมาชิกของไฟล์คอนฟิกูเรชัน กับผู้ใช้ root และรีสตาร์ท **snmpd** daemon
- ไฟล์ **snmpd.conf** ต้องมีอยู่ หากไม่ได้ระบุแฟล็ก **–c** ไว้ในไฟล์คอนฟิกูเรชันบนบรรทัดรับคำสั่ง **snmpd** ไฟล์ **/etc/snmpd.conf** ไม่มีอยู่ หากไฟล์ **/etc/snmpd.conf** ถูกลบทิ้งโดยบังเอิญ ให้ติดตั้งอิมเมจ **bos.net.tcp.client** อีกครั้ง หรือสร้างไฟล์อีกครั้งด้วยรายการคอนฟิกูเรชันที่เหมาะสม ตามที่นิยามไว้ในเพจการจัดการของไฟล์ **snmpd.conf** หากระบุ ไฟล์คอนฟิกูเรชันด้วยแฟล็ก **–c** บนบรรทัดรับคำสั่ง **snmpd** ตรวจสอบให้แน่ใจว่า ไฟล์ที่มีอยู่ และไฟล์ที่เป็นเจ้าของโดย ผู้ใช้ root พาธแบบเต็มและชื่อไฟล์ของไฟล์คอนฟิกูเรชันต้องถูกระบุ หรือไฟล์ **/etc/snmpd.conf** ดีฟอลต์จะถูกใช้

  Solution: ตรวจสอบไฟล์คอนฟิกูเรชันที่มีอยู่ซึ่งระบุไว้ และไฟล์นี้เป็นเจ้าของโดยผู้ใช้ root รีสตาร์ท **snmpd** daemon

- `udp port 161` ถูกจัดขอบเขตแล้ว ตรวจสอบว่า **snmpd** daemon ยังไม่ได้รันอยู่ ออกคำสั่ง **ps -eaf | grep snmpd** เพื่อกำหนดว่า กระบวนการ **snmpd** daemon ถูกเรียกใช้งานแล้ว เฉพาะเอเจนต์ **snmpd** หนึ่งตัวเท่านั้นที่สามารถเชื่อมกับ `udp port 161` ได้

  Solution: หยุดทำงานเอเจนต์ **snmpd** ที่มีอยู่ หรือห้ามพยายามเริ่มต้นทำงานกับกระบวนการ **snmpd** daemon อื่นๆ

ปัญหาเกี่ยวกับความล้มเหลวของ Daemon

หาก **snmpd** daemon ล้มเหลวเมื่อคุณออกคำสั่ง **refresh** หรือส่งสัญญาณ **kill -1** แล้ว ต่อไปนี้อาจเป็นเหตุผลที่อาจเป็นไปได้สำหรับ ความล้มเหลวและโซลูชันที่อาจเป็นไปได้:

- เหตุผล **snmpd** daemon ที่ยกเลิกแล้ว จะถูกล็อกอินเข้าสู่ไฟล์บันทึกการทำงาน **snmpd** หรือไฟล์บันทึกการทำงาน **syslogd** ที่ตั้งค่าไว้ ตรวจสอบไฟล์บันทึกการทำงานเพื่อดูว่า ข้อความแสดงความผิดพลาด FATAL

  Solution: แก้ไขปัญหาและรีสตาร์ท **snmpd** daemon
- ตรวจสอบให้แน่ใจว่า พาธและชื่อไฟล์ที่สมบูรณ์ของไฟล์คอนฟิกูเรชัน ถูกระบุ เมื่อ **snmpd** daemon ถูกเรียกใช้ **snmpd** daemon แยกและเปลี่ยนไปเป็นไดเร็กทอรี root เวลาที่เรียกใช้ หากชื่อพาธเต็มของ ไฟล์คอนฟิกูเรชันไม่ได้ถูกระบุไว้ เอเจนต์ **snmpd** ไม่สามารถค้นหาไฟล์สำหรับรีเฟรช นี่คือข้อผิดพลาดรุนแรงและจะเป็นสาเหตุเอเจนต์ **snmpd** เพื่อยกเลิก

  Solution: ระบุพาธและชื่อไฟล์เต็มของ ไฟล์คอนฟิกูเรชัน **snmpd** ตรวจสอบให้แน่ใจว่า ไฟล์คอนฟิกูเรชันเป็นเจ้าของไฟล์โดยผู้ใช้ root รีสตาร์ท **snmpd** daemon
- ตรวจสอบให้แน่ใจว่าไฟล์คอนฟิกูเรชันสำหรับ **snmpd** ยังคงมีอยู่ ไฟล์อาจถูกลบทิ้งโดยบังเอิญหลังจากที่เรียกใช้เอเจนต์ **snmpd** หากเอเจนต์ **snmpd** ไม่สามารถเปิดไฟล์คอนฟิกูเรชัน ได้ เอเจนต์ **snmpd** จะถูกยกเลิก

  Solution: สร้างไฟล์คอนฟิกูเรชัน **snmpd** อีกครั้ง โปรดตรวจสอบให้แน่ใจว่าไฟล์คอนฟิกูเรชัน ถูกเป็นเจ้าของโดยผู้ใช้ root และรีสตาร์ท **snmpd** daemon

ปัญหาเกี่ยวกับการเข้าถึงตัวแปร MIB

หากตัวแปร Management Information Base (MIB) ไม่สามารถเข้าถึงได้จากเอเจนต์ **snmpd** หากเอเจนต์ **snmpd** กำลังรันอยู่ แอ็พพลิเคชันตัวจัดการ **Simple Network Management Protocol (SNMP)** หมดเวลารอการตอบกลับจากเอเจนต์ **snmpd** ให้ลองใช้วิธีการต่อไปนี้:

- ตรวจสอบคอนฟิกูเรชันของเน็ตเวิร์กของโฮสต์ที่เอเจนต์ **snmpd** กำลังรันอยู่โดยใช้คำสั่ง **netstat -in** ตรวจสอบว่า lo0 ซึ่งเป็นการวนกลับ คืออุปกรณ์ที่กำลังทำงาน หากอุปกรณ์หยุดทำงานแล้ว * (เครื่องดอกจัน) แสดงอยู่ทางซ้ายของ lo0 lo0 ต้องเริ่มทำงานสำหรับเอเจนต์ **snmpd** เพื่อร้องขอเซอร์วิส

  Solution: ออกคำสั่งต่อไปนี้ เพื่อเริ่มทำงานอินเตอร์เฟสการวนกลับ:

  ```
  ifconfig lo0 inet up
  ```
- ตรวจสอบว่า **snmpd** daemon มีเราต์ไปยังโฮสต์ ที่คำร้องขอถูกเรียกใช้

  Solution: สำหรับโฮสต์ที่ **snmpd** daemon กำลังรันอยู่ เพิ่มเราต์ไปยังโฮสต์ที่คำสั่ง **route add** เรียกใช้งาน โปรดดูคำสั่ง **route** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4* สำหรับข้อมูลเพิ่มเติม
- ตรวจสอบว่า ชื่อโฮสต์และ IP แอดเดรสของโฮสต์คือค่าเดียวกัน

  Solution: รีเซ็ตชื่อโฮสต์เพื่อตอบกลับไปยัง IP แอดเดรสของโฮสต์
- ตรวจสอบว่า *localhost* ถูกนิยามไว้เป็น lo0 IP แอดเดรส

  Solution: นิยาม *localhost* ที่เป็นแอดเดรสเดียวกับที่ใช้โดย lo0 IP แอดเดรส (โดยปกติคือ `127.0.0.1`)

ตัวแปร MIB เข้าถึงในปัญหาเกี่ยวกับรายการ community

หากรายการ community ที่ระบุในไฟล์คอนฟิกูเรชันพร้อมกับชื่อมุมมอง MIB แต่ตัวแปร MIB ไม่สามารถเข้าถึงได้ให้ตรวจสอบต่อไปนี้:

- ตรวจสอบว่า คุณได้ระบุรายการ community ที่ถูกต้อง หากคุณระบุชื่อมุมมอง ในรายการ community ฟิลด์ทั้งหมดใน community จำเป็นต้องมีอย่างแน่นอน

  Solution: ระบุฟิลด์ทั้งหมดในรายการ community ในไฟล์คอนฟิกูเรชัน รีเฟรชเอเจนต์ **snmpd** และลองคำร้องขอของคุณอีกครั้ง
- ตรวจสอบว่า โหมดเข้าถึงในรายการ community สอดคล้องกับ ชนิดคำร้องขอของคุณ หากคุณกำลังออกคำร้องขอ **get** หรือ **get-next** ให้ตรวจสอบว่า community มีสิทธิในการอ่านอย่างเดียวหรืออ่าน-เขียน หากคุณกำลังออกคำร้องขอ **set** โปรดตรวจสอบให้แน่ใจว่า community มีสิทธิอ่าน-เขียน

  Solution: ระบุโหมดการเข้าถึงที่ถูกต้องในรายการ community รีเฟรชเอเจนต์ **snmpd** และลองคำร้องขอของคุณอีกครั้ง
- ตรวจสอบให้แน่ใจว่า รายการมุมมองสำหรับชื่อมุมมองที่ระบุถูกระบุอยู่ในรายการ community ในไฟล์คอนฟิกูเรชัน หากมีชื่อมุมมองที่ระบุไว้ในรายการ community แต่ไม่มีรายการมุมมองที่สอดคล้องกัน เอเจนต์ **snmpd** ไม่ได้อนุญาตให้เข้าถึง community นั้น รายการมุมมองจำเป็นต้องมีสำหรับชื่อมุมมอง ที่ระบุอยู่ในไฟล์คอนฟิกูเรชัน

  Solution: ระบุรายการมุมมองสำหรับชื่อมุมมองที่ระบุอยู่ใน รายการ community รีเฟรชเอเจนต์ **snmpd** และลองคำร้องขอของคุณอีกครั้ง
- หากระบุ `iso` ไว้เป็นแผนผังย่อย MIB สำหรับรายการมุมมอง ตรวจสอบว่า `iso.3` ถูกระบุไว้ อินสแตนซ์ 3 จำเป็นต้องมีสำหรับเอเจนต์ **snmpd** เพื่อเข้าถึงส่วนของ `org` ของแผนผัง `iso`

  Solution: ระบุแผนผัง MIB เป็น `iso.3` ในรายการมุมมอง รีเฟรชเอเจนต์ **snmpd** และลองคำร้องขอของคุณอีกครั้ง
- ตรวจสอบ *IP address* และ *network mask* ในรายการ community ตรวจสอบว่า โฮสต์ที่ออกคำร้องขอ SNMP ถูกสอดแทรกใน community ถูกระบุไว้ด้วยชื่อ community

  Solution: เปลี่ยนฟิลด์ *IP address* และ *network mask* ในรายการ community ในไฟล์คอนฟิกูเรชัน เพื่อรวมโฮสต์ที่กำลังเรียกใช้คำร้องขอ SNMP

ไม่ได้ตอบกลับจากปัญหาของเอเจนต์

หาก *IP address* ใน community ถูกระบุไว้เป็น 0.0.0.0 แต่ไม่มีการตอบกลับจากเอเจนต์ **snmpd** ให้ลองวิธีการต่อไปนี้:

- ตรวจสอบฟิลด์ *network mask* ในรายการ community สำหรับการเข้าถึงทั่วไปกับชื่อ community นี้ *network mask* ต้องเป็น **0.0.0.0** หาก *network mask* ถูกระบุไว้เป็น `255.255.255.255` เอเจนต์ **snmpd** ถูกตั้งค่าเพื่อไม่อนุญาตให้คำร้องขอใดๆ ด้วยชื่อ community ระบุไว้

  Solution: ระบุ *network mask* ในรายการ community กับ 0.0.0.0 รีเฟรชเอเจนต์ **snmpd** และลองคำร้องขออีกครั้ง
- ตรวจสอบว่า โหมดการเข้าถึงในรายการ community สอดคล้องกับ ชนิดคำร้อง เมื่อออกคำร้องขอ **get** หรือ **get-next** ตรวจสอบว่า community มีสิทธิในการอ่านอย่างเดียวหรืออ่าน-เขียน หากคุณกำลังออกคำร้องขอ **set** โปรดตรวจสอบให้แน่ใจว่า community มีสิทธิอ่าน-เขียน

  Solution: ระบุโหมดเข้าถึง ที่ถูกต้องในรายการ community รีเฟรชเอเจนต์ **snmpd** และลองคำร้องขอของคุณอีกครั้ง

ปัญหาเกี่ยวกับ noSuchName

หากมีความพยายามในการตั้งค่าตัวแปร MIB ที่เอเจนต์ **snmpd** ถูกสมมติเพื่อสนับสนุน ข้อความแสดงความผิดพลาด `noSuchName` จะถูกส่งคืน ต่อไปนี้อาจคือสาเหตุ:

คำร้องขอการตั้งค่าที่เรียกใช้ไม่ได้สอดแทรกชื่อ community สำหรับ community ที่ถูกต้องด้วยสิทธิในการเข้าถึงการเขียน โปรโตคอล **SNMP** คาดการณ์ว่า คำร้องขอในการตั้งค่า ด้วย community ด้วยสิทธิพิเศษในการเข้าถึงที่ไม่เหมาะสมถูกตอบคำถามด้วยข้อความแสดงความผิดพลาด `noSuchName`

Solution: ออกคำร้องขอในการตั้งค่าชื่อ community สำหรับ community ที่มีสิทธิในการเขียนและสอดแทรกโฮสต์ ที่เรียกใช้คำร้องขอการตั้งค่า

ระบบไฟล์เครือข่าย

Network File System (NFS) คือกลไกสำหรับการเก็บไฟล์ บนเน็ตเวิร์ก ซึ่งเป็นระบบไฟล์ที่แจกจ่ายที่อนุญาตให้ผู้ใช้เข้าถึงไฟล์ และไดเร็กทอรีที่วางอยู่บนคอมพิวเตอร์แบบรีโมตและใช้ไฟล์เหล่านั้น และไดเร็กทอรีหากอยู่บนโลคัล

ตัวอย่างเช่น ผู้ใช้สามารถใช้คำสั่งระบบปฏิบัติการ เพื่อสร้าง ลบ อ่าน เขียน และตั้งค่าแอ็ตทริบิวต์ไฟล์สำหรับไฟล์และไดเร็กทอรี รีโมต

ซอฟต์แวร์แพ็กเกจ NFS สอดแทรกคำสั่งและ daemon สำหรับ NFS, Network Information Service (NIS) และเซอร์วิสอื่นๆ แม้ว่า NFS และ NIS ถูกติดตั้งพร้อมกับหนึ่งแพ็กเกจ แต่ละแพ็กเกจเป็นอิสระจากกัน และแต่ละแพ็กเกจถูกตั้งค่าและถูกดูแลระบบ อย่างเป็นอิสระ

AIX 5.3 และเวอร์ชันถัดมา สนับสนุนโปรโตคอล NFS เวอร์ชัน 2, 3 และ 4 NFS เวอร์ชัน 4 คือ NFS เวอร์ชันที่กำหนดไว้ล่าสุด และถูกอธิบายโดย RFC 3530 รายละเอียดเพิ่มเติมเกี่ยวกับส่วนสนับสนุน AIX ของ NFS เวอร์ชัน 4 จะถูกกล่าวถึงในภายหลังในส่วนนี้ ไคลเอ็นต์ NFS ใช้โปรโตคอล NFS เวอร์ชัน 3 ตามค่าดีฟอลต์

เซอร์วิส NFS

NFS จัดเตรียมเซอร์วิสผ่านความสัมพันธ์แบบไคลเอ็นต์–เซิร์ฟเวอร์

คอมพิวเตอร์ที่สร้าง *ระบบไฟล์* หรือ *ไดเร็กทอรี* และรีซอร์สอื่นๆ พร้อมกับการเข้าถึงแบบรีโมตถูกเรียกว่า *เซิร์ฟเวอร์* การกระทำของการสร้างระบบไฟล์ที่พร้อมใช้งานถูกเรียกว่า *การเอ็กซ์พอร์ต* คอมพิวเตอร์ และกระบวนการที่ใช้รีซอร์สของเซิร์ฟเวอร์พิจารณาถึง *ไคลเอ็นต์* หลังจากที่ไคลเอ็นต์เมาต์ระบบไฟล์ที่เซิร์ฟเวอร์เอ็กซ์พอร์ต ไคลเอ็นต์สามารถเข้าถึงไฟล์เซิร์ฟเวอร์แบบเดี่ยว (เข้าถึงไดเร็กทอรีที่เอ็กซ์พอร์ต สามารถจำกัดไคลเอ็นต์ที่ระบุได้)

เซอร์วิสหลักที่จัดเตรียมไว้โดย NFS คือ:

ตารางที่ 90. เซอร์วิส NFS

Service	คำอธิบาย
เซอร์วิสการเมาต์	เมาต์จาก **/usr/sbin/rpc.mountd** daemon บนเซิร์ฟเวอร์และคำสั่ง **/usr/sbin/mount** บนไคลเอ็นต์ เซอร์วิสนี้พร้อมใช้งานบน NFS เวอร์ชัน 2 และเวอร์ชัน 3
การเข้าถึงไฟล์แบบรีโมต	เข้าถึงจาก **/usr/sbin/nfsd** daemon บนเซิร์ฟเวอร์ และ **/usr/sbin/biod** daemon บนไคลเอ็นต์
เซอร์วิสการประมวลผลแบบรีโมต	เรียกใช้จาก **/usr/sbin/rpc.rexd** daemon บนเซิร์ฟเวอร์และคำสั่ง **/usr/bin/on** บนไคลเอ็นต์
เซอร์วิสสถิติระบบรีโมต	คอมไพล์จาก **/usr/sbin/rpc.rstatd** daemon บนเซิร์ฟเวอร์และคำสั่ง **/usr/bin/rup** บนไคลเอ็นต์
เซอร์วิสการแสดงรายชื่อผู้ใช้แบบรีโมต	แสดงรายการจาก **/usr/lib/netsvc/rusers/rpc.rusersd** daemon บนเซิร์ฟเวอร์และคำสั่ง **/usr/bin/rusers** บนไคลเอ็นต์

ตารางที่ 90. เซอร์วิส NFS (ต่อ)

Service	คำอธิบาย
เซอร์วิสบูตพารามิเตอร์	จัดเตรียมพารามิเตอร์เริ่มต้นทำงานกับไคลเอ็นต์แบบ diskless ของระบบปฏิบัติการ Sun จาก **/usr/sbin/rpc.bootparamd** daemon บนเซิร์ฟเวอร์
เซอร์วิส Remote Wall	ปกป้องจาก **/usr/lib/netsvc/rwall/rpc.rwalld** daemon บนเซิร์ฟเวอร์และคำสั่ง **/usr/sbin/rwall** บนไคลเอ็นต์
เซอร์วิสแบบ Spray	ส่งสตรีมแบบหนึ่งทิศทางของแพ็กเก็ต Remote Procedure Call (RPC) จาก **/usr/lib/netsvc/spray/rpc.sprayd** daemon บนเซิร์ฟเวอร์และคำสั่ง **/usr/sbin/spray** บนไคลเอ็นต์
เซอร์วิสการพิสูจน์ตัวตนของ PC	จัดเตรียมเซอร์วิสการพิสูจน์ตัวตนของผู้ใช้สำหรับ PC-NFS จาก **/usr/sbin/rpc.pcnfsd** daemon บนเซิร์ฟเวอร์
เซอร์วิสความปลอดภัยที่ปรับปรุงแล้ว	จัดเตรียมการเข้าถึงทั้งบนไคลเอ็นต์และเซิร์ฟเวอร์ให้กับเซอร์วิสความปลอดภัย ระดับสูง เช่น Kerberos 5 ซึ่ง **/usr/sbin/gssd** daemon จัดเตรียม NFS พร้อมกับการเข้าถึงเซอร์วิสความปลอดภัยที่จัดเตรียมโดย Network Authentication Service ชุดไฟล์ Network Authentication Service และ Cryptographic Library (`krb5.client.rte`, **krb5.server.rte** และ **modcrypt.base**) ต้องถูกติดตั้ง ชุดไฟล์เหล่านี้ ต้องถูกติดตั้งจาก AIX Expansion Pack
เซอร์วิสการแปลงที่มีลักษณะเฉพาะ	ดำเนินการแปลระหว่างหลักการแสดงความปลอดภัย ซึ่ง NFS เวอร์ชัน 4 มีลักษณะเป็นสตริง และสอดคล้องกับ ID ของระบบแบบตัวเลข นอกจากนี้ การแม็พข้อมูลที่มีลักษณะเฉพาะจาก NFS เวอร์ชัน 4 ภายนอกที่โดเมนถูกจัดการ เซอร์วิสเหล่านี้ ถูกจัดเตรียมไว้โดย **/usr/sbin/nfsrgyd** daemon

หมายเหตุ: คอมพิวเตอร์สามารถเป็นได้ทั้งเซิร์ฟเวอร์ NFS และไคลเอ็นต์ NFS พร้อมกันได้

เซิร์ฟเวอร์ NFS เวอร์ชัน 2 และ 3 เป็นเซิร์ฟเวอร์แบบ *stateless* ซึ่งหมายความว่า เซิร์ฟเวอร์ไม่ได้เก็บข้อมูลรายการดำเนินการเกี่ยวกับไคลเอ็นต์ รายการดำเนินการ NFS สอดคล้องกับการดำเนินการกับไฟล์เดี่ยวและสมบูรณ์ NFS จำเป็นต้องมีไคลเอ็นต์ที่จำข้อมูลใดๆ ที่จำเป็นสำหรับการใช้ NFS ในภายหลัง

เซิร์ฟเวอร์ NFS เวอร์ชัน 4 แบบมีสถานะเนื่องจากไฟล์เปิดและไฟล์ล็อกการดำเนินการ ที่นิยามอยู่ในโปรโตคอล NFS เวอร์ชัน 4

การสนับสนุนNFS แอ็คเซสคอนโทรล

การใช้ AIX NFS เวอร์ชัน 4 สนับสนุน ACL 2 ชนิด : NFS4 และ AIXC

ซอร์สของการกำหนดสิทธิสำหรับการตรวจสอบการเข้าถึงขึ้นอยู่กับระบบไฟล์ที่ถูกเอ็กซ์พอร์ตโดย NFS เซิร์ฟเวอร์ ระบบไฟล์จะพิจารณาแอ็คเซสคอนโทรลของไฟล์ (ACL หรือบิตการอนุญาต) สิทธิของผู้เรียก และข้อจำกัดของระบบโลคัลที่อาจใช้ แอ็พพลิเคชันและผู้ใช้ไม่ควรสันนิษฐานว่าการตรวจสอบของ UNIX โหมดบิต หรือ ACLs ลำพังสามารถถูกใช้เพื่อสรุปการคาดการณ์การเข้าถึง

คำสั่ง **aclget, aclput** และ **acledit** สามารถถูกใช้บนไคลเอ็นต์เพื่อดำเนินการกับ NFS หรือ AIX ACLs สำหรับข้อมูลเพิ่มเติม ดูที่ข้อมูลลิสต์ใน *การรักษาความปลอดภัย*

NFS RBAC

NFS จัดเตรียมการสนับสนุน Role Based Access Control (RBAC) คำสั่งไคลเอ็นต์และเซิร์ฟเวอร์ NFS เปิดใช้งาน RBAC

สิ่งนี้ช่วยให้ผู้ใช้ที่ไม่ใช่ root เรียกใช้คำสั่ง NFS เมื่อผู้ดูแลระบบ กำหนดบทบาท RBAC ของคำสั่งให้กับผู้ใช้ เมื่อต้องการดูรายการ ของสตริงการอนุญาตและสิทธิที่เกี่ยวข้องกับคำสั่ง NFS โปรดอ้างอิงไฟล์ /etc/security/privcmds บนระบบ

NFS4 ACL

NFS4 ACL คือ ACL ที่นิยามไว้โดยโปรโตคอล NFS เวอร์ชัน 4

NFS4 ACL คือแพล็ตอิสระ ดังนั้นจึงสามารถสนับสนุนโดย ไคลเอ็นต์หรือเซิร์ฟเวอร์ของเวนเดอร์ ไคลเอ็นต์ NFS เวอร์ชัน 4 และเซิร์ฟเวอร์ไม่จำเป็นต้องมี เพื่อสนับสนุน NFS4 ACL

ในเซิร์ฟเวอร์ AIX หากอินสแตนซ์ของระบบไฟล์แบบฟิสิคัลที่อยู่ใต้สนับสนุน NFS4 ACL ดังนั้น เซิร์ฟเวอร์ AIX NFS4 สนับสนุน NFS4 ACL สำหรับอินสแตนซ์ระบบไฟล์ ชนิดของระบบไฟล์แบบฟิสิคัลส่วนใหญ่ AIX ไม่สนับสนุน NFS4 ACL ชนิดของระบบไฟล์เหล่านี้รวมไว้ แต่ถูกจำกัดไว้กับ CFS, UDF, JFS และ JFS2 ด้วยแอ็ตทริบิวต์ขยายเพิ่มในเวอร์ชัน 1 อินสแตนซ์ของ JFS2 ทั้งหมดด้วยแอ็ตทริบิวต์ที่ขยายเพิ่มในเวอร์ชัน 2 ที่สนับสนุน NFS4 ACL

ระบบไฟล์สำหรับไคลเอ็นต์ NFS เวอร์ชัน 4 ไม่สามารถอ่านและเขียน NFS4 ACL หากอินสแตนซ์ระบบไฟล์ NFS เวอร์ชัน 4 บนเซิร์ฟเวอร์ที่สนับสนุน NFS4 ACL

AIX ACL

AIXC ACL เป็นแอ็คเซสคอนโทรลลิสต์เฉพาะของ AIX เซิร์ฟเวอร์

ไม่ได้ถูกกำหนดโดยโปรโตคอล NFS เวอร์ชัน 4 และเข้าใจเฉพาะโดย AIX เซิร์ฟเวอร์และไคลเอ็นต์

บนเซิร์ฟเวอร์ NFS เวอร์ชัน 4 AIXC ACL ได้รับการสนับสนุนเมื่ออินสแตนซ์ของระบบไฟล์ที่สำคัญสนับสนุน AIXC ACL อินสแตนซ์ทั้งหมดของ JFS และ JFS2 สนับสนุน AIXC ACL

ไคลเอ็นต์ NFS เวอร์ชัน 4 มีอ็อพชันการเมาต์ที่เปิดการใช้งานและปิดการใช้งานการสนับสนุนสำหรับ AIX ACL ดีฟอลต์คือไม่สนับสนุน AIXC ACL ผู้ใช้ระบบไฟล์บนไคลเอ็นต์ NFS เวอร์ชัน 4 สามารถอ่านและเขียน AIXC ACL เมื่อทั้งไคลเอ็นต์และเซิร์ฟเวอร์รัน AIX และอินสแตนซ์ของระบบไฟล์ฟิสิคัลที่สำคัญบนเซิร์ฟเวอร์สนับสนุน AIXC ACL และ AIX ไคลเอ็นต์ที่เมาต์อินสแตนซ์ของระบบไฟล์กับ AIXC ACL ถูกเปิดใช้งาน การสนับสนุน AIXC ACL ใน NFS เวอร์ชัน 4 จะเหมือนกับการสนับสนุนการใช้ AIXC ACL ใน AIX NFS เวอร์ชัน 2 และ NFS เวอร์ชัน 3

อินสแตนซ์ทั้งหมดของระบบไฟล์ JFS2 พร้อมกับแอ็ตทริบิวต์ส่วนขยายเวอร์ชัน 2 สนับสนุนทั้ง AIXC ACL และ NFS4 ACL ไฟล์ในระบบไฟล์ชนิดนี้อาจมีโหมดบิตเฉพาะ (ไม่มี ACL), NFS4 ACL หรือ AIXC ACL แต่มันไม่สามารถมี NFS4 ACL และ AIXC ACL ในเวลาเดียวกัน

คำสั่ง **aclgettypes** สามารถถูกใช้เพื่อกำหนดชนิดของ ACL ที่สามารถถูกอ่าน และเขียนบนอินสแตนซ์ของระบบไฟล์ คำสั่งนี้อาจให้เอาต์พุตที่แตกต่างจากเมื่อรันกับระบบไฟล์แบบฟิสิคัลบนเซิร์ฟเวอร์ NFS เวอร์ชัน 4 แบบโลคัล กับเมื่อรันกับระบบไฟล์เดียวกันบน NFS เวอร์ชัน 4 ไคลเอ็นต์ ตัวอย่างเช่น อินสแตนซ์ของระบบไฟล์ NFS เวอร์ชัน 4 บน และเซิร์ฟเวอร์ NFS เวอร์ชัน 4 อาจสนับสนุน NFS4 ACL และ AIXC ACL แต่ไคลเอ็นต์ถูกตั้งค่าเฉพาะส่งและรับ NFS4 ACL ในกรณีนี้ เมื่อคำสั่ง **aclgettypes** ถูกเรียกใช้งานจากระบบไฟล์ NFS เวอร์ชัน 4 ไคลเอ็นต์ เฉพาะ NFS4 จะถูกคืนกลับไป นอกจากนี้ ถ้าผู้ใช้บนไคลเอ็นต์ร้องขอ AIXC ACL ข้อผิดพลาดจะถูกส่งคืนกลับไป

ส่วนสนับสนุนระบบไฟล์แคช

Cache File System (CacheFS) คือกลไกการแคชระบบไฟล์ ที่มีวัตถุประสงค์ทั่วไปเพื่อปรับปรุงสมรรถนะของเซิร์ฟเวอร์ NFS และความสามารถในการวัด โดยลดโหลดของเซิร์ฟเวอร์และเน็ตเวิร์ก

ด้วยการออกแบบตามระบบไฟล์ที่ทำเป็นเลเยอร์ CacheFS จัดเตรียมความสามารถในการแคช หนึ่งระบบไฟล์บนระบบไฟล์อื่น ในสภาพแวดล้อมแบบ NFS CacheFS เพิ่มอัตราส่วนไคลเอ็นต์ต่อเซิร์ฟเวอร์ ลดโหลดของเซิร์ฟเวอร์และเน็ตเวิร์ก และปรับปรุงผลการทำงานสำหรับไคลเอ็นต์ สำหรับลิงก์ที่ช้า เช่น Point-to-Point Protocol (PPP)

แคชถูกสร้างขึ้นบนเครื่องไคลเอ็นต์ ดังนั้น ระบบไฟล์ที่ระบุไว้เพื่อเมาต์ในแคชสามารถเข้าถึงแบบโลคัลได้แทนการเข้าถึงระหว่างเน็ตเวิร์ก ไฟล์ถูกวางอยู่ในแคช เมื่อผู้ใช้ร้องขอการเข้าถึงเป็นอันดับแรก แคชจะไม่ ขอรับการเติมจนกว่าผู้ใช้จะร้องขอการเข้าถึงไฟล์หรือไฟล์ต่างๆ คำร้องขอไฟล์เริ่มต้นอาจดูช้าลง แต่ลำดับการใช้ของไฟล์เดียวกัน อาจเร็วกว่า

หมายเหตุ:

1. คุณไม่สามารถแคช / (root) หรือระบบไฟล์ /usr
2. คุณสามารถเมาต์ได้เฉพาะระบบไฟล์ที่ถูกแบ่งใช้เท่านั้น (โปรดดูคำสั่ง **exportfs** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2*)
3. ไม่มีผลการทำงานที่ได้รับในการแคชระบบไฟล์ดิสก์ Journaled File System (JFS) บนโลคัล
4. คุณต้องมีสิทธิแบบผู้ใช้ root หรือสิทธิในระบบเพื่อทำภารกิจในตาราง ต่อไปนี้

ตารางที่ 91. CacheFS tasks

ภารกิจ	วิธีลัด SMIT	คำสั่งหรือไฟล์
ตั้งค่าแคช	cachefs_admin_create	**cfsadmin -c** *MountDirectoryName*[1]
การระบุไฟล์สำหรับการเมาต์	cachefs_mount	**mount -F cachefs -o backfstype=***FileSysType*, **cachedir=***CacheDirectory*[,*options*] BackFileSystem *MountDirectoryName*[2] or edit /etc/**filesystems.**
แก้ไขแคช	cachefs_admin_change	ลบแคช จากนั้นสร้างแคชโดยใช้อ็อพชันคำสั่ง **mount** ตามความเหมาะสม
แสดงข้อมูลแคช	cachefs_admin_change	**cfsadmin -l** *MountDirectoryName.*
ลบแคช	cachefs_admin_remove	1. ยกเลิกการเมาต์ระบบไฟล์: **umount** *MountDirectoryName* 2. พิจารณา ID แคช: **cfsadmin -l** *MountDirectoryName* 3. ลบระบบไฟล์: **cfsadmin -d** *CacheID CacheDirectory*
ตรวจสอบ Integrity ของระบบไฟล์	cachefs_admin_check	**fsck_cachefs***CacheDirectory*[3]

Notes:

1. หลังจากที่คุณได้สร้างแคชแล้ว ห้ามดำเนินการกับการดำเนินการใดๆ ภายในไดเร็กทอรีแคช (cachedir) เอง ซึ่งอาจเป็นสาเหตุของความขัดแย้ง ภายในซอฟต์แวร์ CacheFS
2. หากคุณใช้อ็อพชันคำสั่ง **mount** เพื่อระบุไฟล์สำหรับการเมาต์ คำสั่งต้องถูกเรียใช้อีกครั้งในแต่ละครั้งที่ระบบรีสตาร์ท
3. ใช้อ็อพชัน **-m** หรือ **-o** ของคำสั่ง **fsck_cachefs** เพื่อตรวจสอบระบบไฟล์ที่ไม่มีการทำการซ่อมแซมใดๆ
4. หลังจากการย้ายระบบไปยัง AIX เวอร์ชัน 6.1 หรือสูงกว่าจากเวอร์ชันก่อนหน้านี้ของ AIX ระบบไฟล์แคชเก่าที่สร้างขึ้นในเวอร์ชันที่เก่ากว่าของ AIX ต้องถูกลบออกและสร้างขึ้นใหม่

ส่วนสนับสนุนไฟล์ NFS ที่แม็พไว้

ส่วนสนับสนุนไฟล์ NFS ที่แม็พไว้อนุญาตให้โปรแกรมบนไคลเอ็นต์เข้าถึงไฟล์ ตลอดเวลาที่อยู่ในหน่วยความจำ

ด้วยการใช้รูทีนย่อย **shmat** ผู้ใช้สามารถแม็พ พื้นที่ของไฟล์ลงในพื้นที่แอดเดรสของผู้ใช้เอง เนื่องจากเป็นการอ่านและเขียนโปรแกรมลงใน ส่วนของหน่วยความจำนี้ ไฟล์จะถูกอ่านลงในหน่วยความจำจากเซิร์ฟเวอร์ หรือ ถูกอัพเดตตามต้องการบนเซิร์ฟเวอร์

การแม็พไฟล์ผ่าน NFS ถูกจำกัดในสามวิธี:

- ไฟล์ไม่ได้แบ่งใช้ข้อมูลระหว่างไคลเอ็นต์
- เปลี่ยนแปลงไฟล์บนหนึ่งไคลเอ็นต์โดยใช้ไฟล์ที่แม็พซึ่งมองไม่เห็นบน ไคลเอ็นต์อื่น
- ขอบเขตการล็อกและปลดล็อกไฟล์ไม่ใช่วิธีที่ได้ประสิทธิผลเพื่อทำงานร่วมกับ ข้อมูลระหว่างไคลเอ็นต์

หากไฟล์ NFS ต้องถูกใช้สำหรับการแบ่งใช้ข้อมูลระหว่างโปรแกรมบนไคลเอ็นต์ ที่แตกต่างกัน ให้ใช้การล็อกเร็กคอร์ดและรูทีนย่อย **read** และ **write** ปกติ

โปรแกรมจำนวนมากบนไคลเอ็นต์เดียวกันสามารถแบ่งใช้ข้อมูลได้อย่างมีประสิทธิผลโดยใช้ไฟล์ที่แม็พ คำแนะนำเกี่ยวกับการล็อกเร็กคอร์ดสามารถทำงานร่วมกับอัพเดตกับไฟล์บนไคลเอ็นต์ ซึ่งจัดเตียมไฟล์ทั้งหมดที่ถูกล็อกไว้ไคลเอ็นต์จำนวนมากสามารถแบ่งใช้ไฟล์ที่แม็พตามการใช้ข้อมูลหากข้อมูลไม่เคยมีการเปลี่ยนแปลงเท่านั้น ตามที่อยู่ในฐานข้อมูล แบบสแตติก

การให้บริการ NFS พร็อกซี

AIX สนับสนุนการให้บริการ Network File System (NFS) พร็อกซี AIX เซิร์ฟเวอร์สามารถเอ็กซ์พอร์ตระบบไฟล์ที่สามารถเข้าถึงได้แบบโลคัลและเอ็กซ์พอร์ตพร็อกซี มุมมองพร็อกซีที่ถูกเอ็กซ์พอร์ตสามารถถูกประกอบเข้าโดย NFS ไคลเอ็นต์

AIX การให้บริการพร็อกซี NFS ใช้การแคชของดิสก์ของข้อมูลที่ถูกเข้าถึงเพื่อให้บริการกับคำร้องขอแบบโลคัลที่เหมือนกันที่ตามมาโดยการลดเน็ตเวิร์กทราฟิกกับเซิร์ฟเวอร์ด้านหลัง การให้บริการพร็อกซีมีโอกาสที่จะขยายการเข้าถึงข้อมูล NFS บนเน็ตเวิร์กที่ช้าและมีความน่าเชื่อถือต่ำ ที่มีประสิทธิภาพที่ถูกปรับปรุงและเน็ตเวิร์กทราฟิกที่ลดลงกับเซิร์ฟเวอร์หลักที่มีข้อมูลอยู่ภายใน ขึ้นอยู่กับสภาพพร้อมใช้งานและข้อกำหนดการจัดการเนื้อหา การให้บริการพร็อกซีสามารถให้โซลูชันสำหรับการขยายการเข้าถึง NFS กับขอบของเน็ตเวิร์กโดยไม่ต้องมีการคัดลอกข้อมูล คุณสามารถตั้งค่า AIX การให้บริการพร็อกซี NFS โดยใช้ **mknfsproxy**

การแคชพร็อกซีสามารถถูกใช้กับทั้ง NFS v3 และ NFS v4 โปรโตคอล โปรโตคอลระหว่างพร็อกซีและไคลเอ็นต์ที่เชื่อมต่ออยู่สามารถเป็น NFS v3 หรือ NFS v4 เมื่อ NFS v4 โปรโตคอลถูกใช้ระหว่างพร็อกซีและเซิร์ฟเวอร์ด้านหลัง อย่างไรก็ตาม เมื่อใช้ NFS v3 โปรโตคอล โปรโตคอลระหว่างพร็อกซีและไคลเอ็นต์ที่เชื่อมต่ออยู่ต้องเป็น NFS v3 โปรโตคอล ทั้งการอ่านและเขียนได้รับการสนับสนุนเพิ่มเติมจาก byte range advisory locks

วิธีการรักษาความปลอดภัย krb5, krb5i และ krb5p สามารถถูกใช้ระหว่างพร็อกซีเซิร์ฟเวอร์และไคลเอ็นต์ที่เชื่อมต่อของมัน วิธีเหล่านี้ยังสามารถถูกใช้ระหว่างพร็อกซีเซิร์ฟเวอร์และเซิร์ฟเวอร์หลัก โดยใช้เทคโนโลยีการฟอร์เวิร์ดตั๋วผ่านพร็อกซี คุณสามารถพิสูจน์ตัวตนบนไคลเอ็นต์และถูกพิสูจน์ตัวตนกับเซิร์ฟเวอร์หลัก เพื่อใช้ประโยชน์ของเทคโนโลยีนี้ ใช้คำสั่ง **kinit** พร้อมกับอ็อพชัน **-f** เมื่อคุณทำการพิสูจน์ตัวจริงโดยใช้ Kerberos ถ้าความปลอดภัย **auth_sys** ถูกใช้ระหว่างพร็อกซีและเซิร์ฟเวอร์ด้านหลัง เมื่อคุณเข้าถึงเซิร์ฟเวอร์ด้านหลัง พร็อกซีเซิร์ฟเวอร์จะแม็พการเข้าถึง Kerberos ไคลเอ็นต์กับแอ็ตทริบิวต์ **auth_sys** เพื่อให้ได้ผลที่ดีที่สุด พร็อกซีเซิร์ฟเวอร์และเซิร์ฟเวอร์ด้านหลังควรแบ่งใช้ผู้ใช้และคำจำกัดความเอกลักษณ์ของกลุ่มเดียวกัน

ข้อจำกัดต่อไปนี้ใช้กับการให้บริการ NFS พร็อกซี :

- การให้บริการพร็อกซีต้องการไคลเอ็นต์ที่เชื่อมต่อโดยใช้ TCP

- เนื่องจากการให้บริการพร็อกซีจัดเตรียมวิธีสำหรับ NFS v3 ไคลเอ็นต์เพื่อเรียกดูผ่านเนมสเปซของ NFS v4 ที่ถูกเอ็กซ์พอร์ต โดยไม่ได้ใช้คำสั่ง **mount** และ **unmount** คุณต้องใช้คำสั่ง **mknfsproxy** พร้อมกับอ็อพชัน **mfsid** เมื่อคุณสร้างระบบไฟล์ของพร็อกซี
- ระบบไฟล์แคชใช้กับการให้บริการพร็อกซีต้องเป็นระบบไฟล์ Enhanced JFS (JFS2)
- การให้บริการพร็อกซีจะรัน CacheFS ด้านบนของ AIX ไคลเอ็นต์ที่ถูกประกอบเข้ากับเซิร์ฟเวอร์ NFS ด้านหลัง คุณลักษณะ I/O (CIO) ที่ใช้พร้อมกัน มีให้ใช้กับ AIX NFS ไคลเอ็นต์ จะปรับปรุงประสิทธิภาพของ CacheFS พยายามเข้าถึงการประกอบเข้ากับ NFS ไคลเอ็นต์อาจล้มเหลวเนื่องจากขัดแย้งกับการพยายามเปิดของ CIO

ชนิดของการเมาต์ NFS

มีสามชนิดของการเมาต์ NFS: predefined, explicit และ automatic

การเมาต์แบบ *Predefined* ระบุอยู่ในไฟล์ `/etc/filesystems` แต่ละ stanza (หรือรายการ) ในไฟล์นี้นิยามคุณสมบัติของการเมาต์ ข้อมูล เช่น ชื่อโฮสต์ พาธรีโมต พาธโลคัล และอ็อพชันการเมาต์ใดๆ ถูกแสดงอยู่ใน stanza นี้ การเมาต์แบบ Predefined ถูกใช้เมื่อเมาต์ ต้องการการดำเนินการที่ถูกต้องของไคลเอ็นต์

การเมาต์แบบ *Explicit* ตอบสนองความต้องการของผู้ใช้ root การเมาต์แบบ Explicit จะทำสำหรับระยะเวลาสั้นๆ เมื่อมีความต้องการสำหรับการเมาต์ ที่ไม่ได้วางแผนไว้เป็นครั้งคราว การเมาต์แบบ Explicit ยังสามารถถูกใช้หากการเมาต์ จำเป็นสำหรับภารกิจพิเศษ และเมาต์นั้นพร้อมใช้งานบนไคลเอ็นต์ NFS เมาต์เหล่านี้ผ่านการรับรองบนบรรทัดรับคำสั่ง โดยใช้คำสั่ง **mount** พร้อมกับข้อมูลที่จำเป็นทั้งหมด การเมาต์แบบ Explicit ไม่ได้ต้องการอัพเดตไฟล์ `/etc/filesystems` ระบบไฟล์เมาต์แบบ explicitly ยังคงเมาต์อยู่ยกเว้นการยกเลิกการเมาต์แบบ explicitly พร้อมกับคำสั่ง **umount** หรือจนกระทั่งระบบรีสตาร์ท

การเมาต์แบบ *Automatic* ควบคุมโดยคำสั่ง **automount** ซึ่งเป็นสาเหตุทำให้ส่วนขยายเคอร์เนล **AutoFS** มอนิเตอร์ไดเร็กทอรีที่ระบุ สำหรับกิจกรรม หากโปรแกรมหรือผู้ใช้พยายามเข้าถึงไดเร็กทอรี ที่ไม่ได้เมาต์ในปัจจุบัน จากนั้น **AutoFS** กั้นคำร้องขอ จัดเรียงการเมาต์ของระบบไฟล์ จากนั้น ให้บริการคำร้องขอ

การเอ็กซ์พอร์ตและติดตั้ง NFS

การเอ็กซ์พอร์ตและติดตั้งไดเร็กทอรีต้องทำความเข้าใจสำหรับ NFS ผู้ดูแลระบบ

เซิร์ฟเวอร์ NFS ต้องเอ็กซ์พอร์ตไฟล์หรือไดเร็กทอรี หลังจาก ไคลเอ็นต์ NFS ติดตั้งไฟล์หรือไดเร็กทอรี รายละเอียดเพิ่มเติมเกี่ยวกับแนวคิดเหล่านี้ ถูกรวมไว้ในส่วนนี้

การเอ็กซ์พอร์ตไดเร็กทอรี NFS

การเอ็กซ์พอร์ตไดเร็กทอรีถูกทำบนเซิร์ฟเวอร์ การเอ็กซ์พอร์ตไดเร็กทอรี ประกาศว่า ไดเร็กทอรีใน namespace ของเซิร์ฟเวอร์พร้อมใช้งานกับ เครื่องไคลเอ็นต์

ไดเร็กทอรีที่เอ็กซ์พอร์ตแล้วถูกอ้างอิงกับ *export* และสอดแทรก ไฟล์ทั้งหมดภายในไดเร็กทอรีนั้น ที่ตั้งอยู่บนระบบไฟล์ ของไดเร็กทอรี

การเอ็กซ์พอร์ตแต่ละครั้งยังกำหนดข้อจำกัดในการเข้าถึง ตัวอย่างเช่น ข้อจำกัดต่อไปนี้ อาจถูกนิยามไว้:
- ไคลเอ็นต์ที่อาจเข้าถึงไดเร็กทอรีที่เอ็กซ์พอร์ต
- เวอร์ชัน NFS ที่ไคลเอ็นต์ต้องใช้เพื่อเข้าถึงไดเร็กทอรี
- ไม่ว่าไคลเอ็นต์จะสามารถเขียนไฟล์ลงในเอ็กซ์พอร์ตหรือไม่ก็ตาม

- เมธอดความปลอดภัยที่ไคลเอ็นต์ต้องใช้เพื่อเข้าถึงไดเร็กทอรีและไฟล์ในเอ็กซ์พอร์ต

สำหรับคำอธิบายแบบเต็มของข้อจำกัดที่ต้องเอ็กซ์พอร์ตซึ่งได้รับอนุญาตและซีแมนทิกส์เอ็กซ์พอร์ต โปรดดู คำอธิบายคำสั่ง **exportfs** ในคำอธิบายไฟล์ *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2* และ /etc/exports ใน *การอ้างอิงไฟล์*

หมายเหตุ: เมื่อแอ็ตทริบิวต์ของเอ็กซ์พอร์ตเปลี่ยนแปลงไป ไดเร็กทอรีต้องถูกเอ็กซ์พอร์ตอีกครั้ง หากต้องการให้มีผลบังคับใช้ไดเร็กทอรีอาจจำเป็นต้องเอ็กซ์พอร์ตอีกครั้ง เนื่องจากการเปลี่ยนแปลงไปเป็นไฟล์อื่นหรือเปลี่ยนเป็นเซิร์ฟเวอร์ภายนอก ตัวอย่างเช่น หากชื่อไคลเอ็นต์ที่ระบุในรายการเข้าถึงคือ netgroup ที่นิยามอยู่ในไฟล์ /etc/netgroup และนิยามของกลุ่มไคลเอ็นต์เปลี่ยนแปลงไป เอ็กซ์พอร์ตทั้งหมดที่ใช้ netgroup นั้นในรายการเข้าถึงต้องถูกเอ็กซ์พอร์ตอีกครั้งสำหรับการเปลี่ยนแปลงเพื่อให้มีผลบังคับใช้

เช่นเดียวกัน หาก IP แอดเดรส ของไคลเอ็นต์เปลี่ยนแปลงไป เอ็กซ์พอร์ตทั้งหมดที่ระบุไว้ที่ ไคลเอ็นต์ในรายการเข้าถึงต้องถูกเอ็กซ์พอร์ตอีกครั้ง เหตุผลคือ เนื่องจากเซิร์ฟเวอร์ NFS รักษาแคชของสิทธิในการเข้าถึงไคลเอ็นต์ในแต่ละการเอ็กซ์พอร์ต แคชจะถูกล้างข้อมูลสำหรับข้อมูลที่ไม่ได้เอ็กซ์พอร์ตหรือเอ็กซ์พอร์ตใหม่อีกครั้งในแต่ละครั้ง หากสิทธิในการเข้าถึง เอ็กซ์พอร์ตถูกแก้ไข โดยเฉพาะหาก IP แอดเดรส ของไคลเอ็นต์เปลี่ยนแปลงไป หรือหากไคลเอ็นต์ถูกย้ายออกจากรายการเข้าถึงข้อมูลที่ไม่ได้เอ็กซ์พอร์ตหรือเอ็กซ์พอร์ตใหม่อีกครั้ง ต้องถูกทำ ดังนั้น การเข้าถึงของไคลเอ็นต์มีผลกระทบอย่างถูกต้องในแคช เซิร์ฟเวอร์ NFS เรียก **rpc.mountd** daemon เพื่อขอรับสิทธิในการเข้าถึงแต่ละไคลเอ็นต์ ดังนั้น **rpc.mountd** daemon ต้องรันอยู่บนเซิร์ฟเวอร์ แม้ว่าเซิร์ฟเวอร์จะเอ็กซ์พอร์ตระบบไฟล์สำหรับการเข้าถึง NFS เวอร์ชัน 4 เท่านั้น

การเมาต์ไดเร็กทอรี NFS

ไคลเอ็นต์ NFS สามารถเมาต์ไดเร็กทอรีที่ได้ถูกเอ็กซ์พอร์ตโดยเซิร์ฟเวอร์ NFS การเมาต์ไดเร็กทอรีทำให้ไฟล์ที่ตั้งอยู่บนเซิร์ฟเวอร์ NFS พร้อมใช้งานกับไคลเอ็นต์ NFS

ไคลเอ็นต์สามารถเข้าถึงไฟล์บนเซิร์ฟเวอร์หากไฟล์ถูกเอ็กซ์พอร์ตโดยเซิร์ฟเวอร์ และข้อจำกัดในการเอ็กซ์พอร์ตที่อนุญาตให้ไคลเอ็นต์มีสิทธิเข้าถึงไฟล์ของ การเอ็กซ์พอร์ต หากไคลเอ็นต์ถูกเมาต์กับการเอ็กซ์พอร์ตของเซิร์ฟเวอร์ที่เป็นผลสำเร็จแล้วบนจุดเมาต์ใน namespace ไฟล์ของเซิร์ฟเวอร์สำหรับการเอ็กซ์พอร์ตนั้นจะมีอยู่ใน namespace ของไคลเอ็นต์และปรากฏขึ้นเป็นไฟล์บนระบบไฟล์โลคัล

ตัวอย่างเช่น สมมติว่าคุณต้องการเอ็กซ์พอร์ตไดเร็กทอรี /tmp บนเซิร์ฟเวอร์ diamond และเมาต์ไดเร็กทอรีนั้นบนไคลเอ็นต์ clip เป็นไดเร็กทอรี /mnt บนเซิร์ฟเวอร์ให้พิมพ์ คำสั่งต่อไปนี้:

```
exportfs -i -o access=clip /tmp
```

คำสั่งนี้ทำให้ไดเร็กทอรี /tmp พร้อมใช้งานกับไคลเอ็นต์

สำหรับไคลเอ็นต์ ให้พิมพ์คำสั่งต่อไปนี้:

```
mount diamond:/tmp /mnt
```

ไดเร็กทอรีและไฟล์ที่อยู่ในไดเร็กทอรี /tmp ของเซิร์ฟเวอร์จะปรากฏขึ้นในไดเร็กทอรี /mnt ของไคลเอ็นต์

หมายเหตุ:

1. มีความแตกต่างกันบางประการระหว่าง NFS เวอร์ชัน 2 และ 3 และ NFS เวอร์ชัน 4 ในเรื่องของวิธีการจัดการกับการเมาต์ ใน NFS เวอร์ชัน 2 และ 3 เซิร์ฟเวอร์ที่เอ็กซ์พอร์ต ไดเร็กทอรีต้องการทำให้พร้อมใช้งานสำหรับการเมาต์ ไคลเอ็นต์ NFS เวอร์ชัน 2 หรือ 3 ต้องถูกเมาต์แต่ละการเอ็กซ์พอร์ตที่ต้องการ เข้าถึง

ด้วย NFS เวอร์ชัน 4 เซิร์ฟเวอร์ยังคงระบุการควบคุมการเอ็กซ์พอร์ต สำหรับแต่ละไดเร็กทอรีเซิร์ฟเวอร์หรือระบบไฟล์ที่ต้องการเอ็กซ์พอร์ตสำหรับการเข้าถึง NFS จากการควบคุม การเอ็กซ์พอร์ตเหล่านี้ เซิร์ฟเวอร์ที่สร้างการแสดงผลแผนผังไดเร็กทอรีเดี่ยวของการกรอกข้อมูล ที่เอ็กซ์พอร์ตซึ่งมีช่องว่างระหว่างไดเร็กทอรีที่ถูกเอ็กซ์พอร์ต แผนผังนี้ รู้จักการว่าเป็นระบบไฟล์แบบ pseudo และเริ่มต้นที่ pseudo root ของ NFS เวอร์ชัน 4 โมเดลระบบไฟล์ pseudo สำหรับ NFS เวอร์ชัน 4 อนุญาตให้ไคลเอ็นต์ NFS เวอร์ชัน 4 ดำเนินการเมาต์เดี่ยวของ pseudo root ของเซิร์ฟเวอร์ตามลำดับการเข้าถึง ข้อมูลที่เอ็กซ์พอร์ตของเซิร์ฟเวอร์ทั้งหมด ขึ้นอยู่กับการนำไปปฏิบัติ ไคลเอ็นต์ AIX NFS สนับสนุนคุณลักษณะนี้ เนื้อหาจริงที่มองเห็นโดยไคลเอ็นต์ขึ้นอยู่กับการควบคุมการเอ็กซ์พอร์ตของเซิร์ฟเวอร์

2. NFS เวอร์ชัน 4 ไม่ได้อนุญาตให้มีการเมาต์แบบไฟล์ต่อไฟล์

การเมาต์ NFS

ไคลเอ็นต์เข้าถึงไฟล์บนเซิร์ฟเวอร์โดยเมาต์เซิร์ฟเวอร์ไดเร็กทอรีที่เอ็กซ์พอร์ต เป็นอันดับแรก เมื่อไคลเอ็นต์เมาต์ไดเร็กทอรี ซึ่งไม่ได้ทำสำเนาของไดเร็กทอรี นั้นไว้ แต่ กระบวนการเมาต์ใช้ชุดของการเรียกโพรซีเดอร์แบบรีโมต เพื่อเปิดใช้งานไคลเอ็นต์ในการเข้าถึงไดเร็กทอรีบนเซิร์ฟเวอร์

ต่อไปนี้อธิบายถึงกระบวนการเมาต์:

1. เมื่อเซิร์ฟเวอร์เริ่มต้นขึ้น สคริปต์ `/etc/rc.nfs` จะรันคำสั่ง **exportfs** ซึ่งอ่านไฟล์เซิร์ฟเวอร์ `/etc/exports` จากนั้นแจ้งให้เคอร์เนลทราบว่า ไดเร็กทอรีถูกเอ็กซ์พอร์ต และข้อจำกัดในการเข้าถึง ที่ต้องการ
2. **rpc.mountd** daemon และ **nfsd** daemons จำนวนมากสตาร์ทโดยสคริปต์ `/etc/rc.nfs`
3. จากนั้น สคริปต์ `/etc/rc.nfs` เรียกใช้งานคำสั่ง **mount** ที่อ่านระบบไฟล์ที่แสดงอยู่ในไฟล์ `/etc/filesystems`
4. คำสั่ง **mount** วางเซิร์ฟเวอร์ตั้งแต่หนึ่งเครื่องขึ้นไป ที่เอ็กซ์พอร์ตข้อมูลที่ไคลเอ็นต์ต้องการและตั้งค่าการสื่อสารระหว่างตัวเองและเซิร์ฟเวอร์นั้น กระบวนการนี้เรียกว่า *การเชื่อม*
5. คำสั่ง **mount** ร้องขอเซิร์ฟเวอร์ ตั้งแต่หนึ่งเครื่องขึ้นไป ซึ่งอนุญาตให้ไคลเอ็นต์เข้าถึงไดเร็กทอรีในไฟล์ `/etc/filesystems`
6. เซิร์ฟเวอร์ daemon รับไคลเอ็นต์เมาต์ร้องขอ และให้สิทธิหรือปฏิเสธ หากไดเร็กทอรีที่ร้องขอพร้อมใช้งานที่ไคลเอ็นต์นั้น เซิร์ฟเวอร์ daemon ส่งไคลเอ็นต์เคอร์เนลที่มี identifier ที่เรียกว่า *การจัดการไฟล์*
7. เคอร์เนลไคลเอ็นต์จะผูกการจัดการกับไฟล์กับจุดเมาต์ (ไดเร็กทอรี) โดยบันทึกข้อมูลบางอย่างใน *เร็กคอร์ดเมาต์*

การสื่อสารกับไคลเอ็นต์ด้วย **rpc.mountd** daemon ไม่ได้เกิดขึ้นกับ NFS เวอร์ชัน 4 ที่ประมวลผลการเมาต์ การดำเนินการในโปรโตคอลหลัก NFS เวอร์ชัน 4 ถูกใช้เพื่อบริการในฝั่งไคลเอ็นต์ที่ดำเนินการเมาต์ การใช้เซิร์ฟเวอร์ NFS เวอร์ชัน 4 ใช้ส่วนสนับสนุนใน **rpc.mountd** daemon เป็นส่วนหนึ่งของการจัดการกับ NFS เวอร์ชัน 4

ไฟล์ /etc/exports

ไฟล์ `/etc/exports` บ่งชี้ไดเร็กทอรีทั้งหมดที่เซิร์ฟเวอร์เอ็กซ์พอร์ตไปยัง ไคลเอ็นต์

แต่ละบรรทัดในไฟล์ที่ระบุไดเร็กทอรีเดี่ยว ไดเร็กทอรีสามารถระบุได้สองครั้งในไฟล์ `/etc/exports`: หนึ่งสำหรับ NFS เวอร์ชัน 2 หรือ NFS เวอร์ชัน 3 และอีกหนึ่งสำหรับ NFS เวอร์ชัน 4 เซิร์ฟเวอร์เอ็กซ์พอร์ตไดเร็กทอรี แต่ละครั้งที่เริ่มต้นเซิร์ฟเวอร์ NFS ไดเร็กทอรีที่เอ็กซ์พอร์ต สามารถเมาต์โดยไคลเอ็นต์ ไวยากรณ์ของบรรทัดใน ไฟล์ `/etc/exports` คือ:

```
directory    -option[,option]
```

directory คือชื่อพาธของไดเร็กทอรี อ็อพชันสามารถกำหนดแฟล็กปกติได้ เช่น **ro** หรือรายการของชื่อโฮสต์โปรดดูเอกสารคู่มือเฉพาะของไฟล์ /etc/exports คำสั่ง ใน *ข้อมูลอ้างอิงไฟล์* และ **exportfs** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2* สำหรับรายการอ็อพชันและคำอธิบาย สคริปต์ /etc/rc.nfs ไม่ได้สตาร์ท **nfsd** daemons หรือไฟล์ **rpc.mountd** daemon if the /etc/exports ไม่มีอยู่

ต่อไปนี้คือตัวอย่างรายการจากไฟล์ /etc/exports:

```
/usr/games    -ro,access=ballet:jazz:tap
/home         -root=ballet,access=ballet
/var/tmp
/usr/lib      -access=clients
/accounts/database -vers=4,sec=krb5,access=accmachines,root=accmachine1
/tmp          -vers=3,ro
/tmp          -vers=4,sec=krb5,access=accmachines,root=accmachine1
```

รายการแรกในตัวอย่างนี้ระบุว่าไดเร็กทอรี /usr/games สามารถเมาต์โดยระบบที่ชื่อ ballet, jazz และ tap ระบบเหล่านี้สามารถอ่านข้อมูลและรันโปรแกรมจากไดเร็กทอรี แต่ไม่สามารถเขียนลงในไดเร็กทอรีได้

รายการที่สองในตัวอย่างนี้ระบุว่าไดเร็กทอรี /home สามารถเมาต์โดยระบบ ballet และการเข้าถึง root ที่อนุญาตให้ใช้สำหรับไดเร็กทอรี

รายการที่สามในตัวอย่างนี้ ระบุว่า ไคลเอ็นต์ใดๆ สามารถเมาต์ไดเร็กทอรี /var/tmp (สังเกตว่า รายการเข้าถึงหายไป)

รายการที่สี่ในตัวอย่างนี้ระบุรายการเข้าถึงที่ออกแบบโดย netgroup clients อีกนัยหนึ่งคือ เครื่องเหล่านี้ที่ออกแบบมรเป็นของ netgroup clients สามารถเมาต์ไดเร็กทอรี /usr/lib จากเซิร์ฟเวอร์นี้ (*netgroup* คือกลุ่มเน็ตเวิร์กแบบกว้างๆ ที่อนุญาตให้เข้าถึง รีซอร์สเน็ตเวิร์กสำหรับความปลอดภัยหรือวัตถุประสงค์ในการจัดการ Netgroups ถูกควบคุมโดยใช้ NIS

รายการที่ห้าอนุญาตให้เข้าถึงไดเร็กทอรี /accounts/database กับไคลเอ็นต์ใน accmachines netgroup โดยใช้โปรโตคอล NFS เวอร์ชัน 4 และการเข้าถึงไดเร็กทอรีโดยใช้การพิสูจน์ตัวตนแบบ Kerberos 5 การเข้าถึงแบบ root อนุญาตให้ใช้จาก accmachine1 เท่านั้น

รายการที่หกและรายการที่เจ็ดเอ็กซ์พอร์ตไดเร็กทอรี /tmp โดยใช้เวอร์ชันและอ็อพชันอื่นๆ หากรายการสองรายการสำหรับไดเร็กทอรีเดียวกันกับเวอร์ชัน NFS อื่นๆ ที่มีอยู่ในไฟล์ /etc/exports คำสั่ง **exportfs** จะเอ็กซ์พอร์ตทั้งสองไฟล์ หากไดเร็กทอรีมีอ็อพชันเดียวกันสำหรับ NFS เวอร์ชัน 4 และ NFS เวอร์ชัน 3 คุณสามารถมีได้หนึ่งรายการในไฟล์ /etc/exports ที่ระบุโดย -vers=3:4

ไฟล์ /etc/xtab

ไฟล์ /etc/xtab มีรูปแบบที่คล้ายกับไฟล์ /etc/exports และแสดงไดเร็กทอรี ที่เอ็กซ์พอร์ตในปัจจุบัน

เมื่อใดก็ตามที่คำสั่ง **exportfs** รัน ไฟล์ /etc/xtab จะเปลี่ยนแปลงไป ซึ่งอนุญาตให้คุณเอ็กซ์พอร์ตไดเร็กทอรีชั่วคราวโดยไม่ต้อง เปลี่ยนแปลงไฟล์ /etc/exports หากไดเร็กทอรีที่เอ็กซ์พอร์ตแบบชั่วคราว ไม่ได้ถูกเอ็กซ์พอร์ต ไดเร็กทอรีจะถูกลบออกจากไฟล์ /etc/xtab

หมายเหตุ: ไฟล์ /etc/xtab จะถูกอัพเดตโดยอัตโนมัติ และไม่สามารถแก้ไขได้

ไฟล์ /etc/nfs/hostkey

ไฟล์ถูกใช้โดยเซิร์ฟเวอร์ NFS เพื่อระบุโฮสต์หลัก Kerberos และตำแหน่งของไฟล์ keytab

สำหรับคำสั่งเกี่ยวกับวิธีกำหนดคอนฟิกและการควบคุมดูแลไฟล์นี้ โปรดดูคำอธิบายคำสั่ง **nfshostkey** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*

ไฟล์ /etc/nfs/local_domain

ไฟล์นี้มีโดเมน NFS โลคัลของระบบ

ซึ่งเห็นได้ว่าระบบที่แบ่งใช้โดเมน NFS โลคัลเดียวกันจะแบ่งใช้รีจีสทรีของผู้ใช้และกลุ่มเดียวกัน สำหรับคำสั่งเกี่ยวกับวิธีกำหนดคอนฟิก และควบคุมดูแลไฟล์นี้ โปรดดูคำอธิบายคำสั่ง **chnfsdom** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1*

ไฟล์ /etc/nfs/realm.map

ไฟล์นี้ถูกใช้โดยการลงทะเบียน NFS เพื่อแม็พหลักการ Kerberos ขากเข้าของรูปแบบ *name@kerberos-realm* กับรูปแบบ *name@nfs-domain*

จากนั้นสามารถแก้ไขปัญหา *name@nfs-domain* กับหนังสือรับรอง UNIX บนโลคัล ไฟล์นี้จัดเตรียมวิธีการแบบปกติเพื่อแม็พหลักการ Kerberos กับการลงทะเบียนผู้ใช้ เซิร์ฟเวอร์ ซึ่งเหมาะสมกัน เมื่อไคลเอ็นต์ในขอบเขต Kerberos ต่างๆ จะถูกเข้าถึงเซิร์ฟเวอร์ แต่ namespace ของผู้ใช้แบบโกลบอล ไฟล์ควรมีบรรทัด ในรูปแบบต่อไปนี้:

```
realm1 nfs-domain
realm2 nfs-domain
```

สำหรับขอบเขต Kerberos ทั้งหมดที่เซิร์ฟเวอร์สนับสนุน หากชื่อขอบเขต Kerberos คือชื่อเดียวกับโดเมน NFS ของเซิร์ฟเวอร์ ไฟล์นี้ไม่จำเป็นต้องมี หากคุณต้องการความสามารถทั่วไปเพิ่มเติม ของการแม็พ *userA@kerberos-realm* กับ *userB@nfs-domain* ใช้เซอร์วิส Enterprise Identity Mapping (EIM) หากต้องการข้อมูลเพิ่มเติม โปรดดู “การแม็พลักษณะเฉพาะ” ในหน้า 571

หากต้องการเพิ่ม แก้ไข หรือลบรายการในไฟล์นี้ ให้ใช้คำสั่ง **chnfsrtd** โปรดดูคำอธิบายคำสั่ง **chnfsrtd** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1* สำหรับรายละเอียดเพิ่มเติม

ไฟล์ /etc/nfs/princmap

ไฟล์นี้แม็พชื่อโฮสต์กับ Kerberos หลักเมื่อหลักการ ไม่ใช่ชื่อโดเมนที่ผ่านการรับรองโดยสมบูรณ์ของเซิร์ฟเวอร์

ไฟล์ประกอบด้วยจำนวนบรรทัดต่างๆ ในรูปแบบต่อไปนี้:

```
<host part of principal> alias1 alias2 ...
```

หากต้องการเพิ่ม แก้ไข หรือลบรายการออกจากไฟล์นี้ ให้ใช้คำสั่ง **nfshostmap** โปรดดูคำอธิบายคำสั่ง **nfshostmap** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4* สำหรับข้อมูลเพิ่มเติม

ไฟล์ /etc/nfs/security_default

ไฟล์ `/etc/nfs/security_default` มีรายการความปลอดภัยที่คุ้นเคยซึ่งอาจใช้โดยไคลเอ็นต์ NFS หาก สมควรนำมาใช้

ใช้คำสั่ง **chnfssec** เพื่อจัดการกับไฟล์นี้ โปรดดูคำอธิบายคำสั่ง **chnfssec** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1* สำหรับข้อมูลเพิ่มเติม

Remote Procedure Call Protocol

NFS ถูกนำไปใช้กับชนิดของเครื่องอย่างกว้างขวาง ระบบปฏิบัติการ และสถาปัตยกรรมด้านเน็ตเวิร์ก NFS บรรลุถึงความเป็นอิสระนี้โดยใช้โปรโตคอล **Remote Procedure Call (RPC)**

RPC คือไลบรารีของโพรซีเดอร์ โพรซีเดอร์อนุญาตให้หนึ่งกระบวนการ (กระบวนการของไคลเอ็นต์) ส่งตรงไปยังกระบวนการอื่น (กระบวนการเซิร์ฟเวอร์) เพื่อรันการเรียกโพรซีเดอร์หากกระบวนการของไคลเอ็นต์ได้รันการเรียกในพื้นที่แอดเดรสของตนเอง เนื่องจากไคลเอ็นต์และเซิร์ฟเวอร์คือกระบวนการสองขั้นตอนที่แบ่งแยก กระบวนการเหล่านี้ต้องไม่มีอยู่บนระบบฟิสิคัลเดียวกัน (แม้ว่าจะสามารถอยู่บนระบบเดียวกันได้ก็ตาม)

NFS ถูกนำไปใช้เป็นชุดของการเรียก **RPC** ที่เซิร์ฟเวอร์ให้บริการชนิดของการเรียกที่ทำขึ้นโดยไคลเอ็นต์ ไคลเอ็นต์ต้องทำการเรียกอ้างอิงตาม การดำเนินการกับระบบไฟล์ที่ถูกทำโดยกระบวนการของไคลเอ็นต์ NFS ที่อยู่ในการรับรู้นี้ คือแอ็พพลิเคชัน RPC

เนื่องจากกระบวนการเซิร์ฟเวอร์และไคลเอ็นต์สามารถตั้งอยู่บนระบบฟิสิคัลที่แตกต่างกันสองระบบ ซึ่งอาจมีสถาปัตยกรรมที่แตกต่างกัน **RPC** ต้องแอดเดรสความเป็นไปได้ที่สองระบบอาจไม่แสดงข้อมูลอยู่ใน วิธีเดียวกัน สำหรับเหตุผลนี้ **RPC** ใช้ชนิดข้อมูลที่นิยามไว้โดยโปรโตคอล **eXternal Data Representation (XDR)**

โปรโตคอล eXternal Data Representation

โปรโตคอล **eXternal Data Representation (XDR)** คือข้อกำหนดคุณสมบัติ สำหรับการแทนค่ามาตรฐานของชนิดข้อมูลที่หลากหลาย

ด้วยการใช้การแทนค่าชนิดข้อมูลมาตรฐาน โปรแกรมสามารถทำให้มั่นใจได้ว่า กำลังตีความข้อมูลอย่างถูกต้อง แม้ว่า แหล่งข้อมูลคือเครื่อง ที่มีสถาปัตยกรรมที่แตกต่างกันโดยสมบูรณ์

ในการฝึกปฏิบัติ โปรแกรมส่วนใหญ่ไม่ได้ใช้ **XDR** ภายใน แต่ ใช้การแทนค่าชนิดข้อมูลที่ระบุเฉพาะกับสถาปัตยกรรมของคอมพิวเตอร์ ที่โปรแกรมกำลังรันอยู่ เมื่อโปรแกรมต้องการสื่อสารกับโปรแกรมอื่น ให้แปลงข้อมูลไปเป็นรูปแบบ **XDR** ก่อนที่จะส่ง ข้อมูล ในทางกลับกัน เมื่อได้รับข้อมูล โปรแกรมจะแปลงข้อมูลจากรูปแบบ **XDR** ไปเป็นการแทนค่าชนิดข้อมูลเฉพาะของตนเอง

portmap daemon

portmap daemon ช่วยให้หมายเลขโปรแกรมที่แม็พไคลเอ็นต์ และหมายเลขเวอร์ชันที่คู่กับหมายเลขพอร์ตของเซิร์ฟเวอร์

แอ็พพลิเคชัน RPC แต่ละตัวที่เชื่อมโยงกับหมายเลขโปรแกรมและ หมายเลขเวอร์ชัน หมายเลขเหล่านี้เพื่อสื่อสารกับเซิร์ฟเวอร์แอ็พพลิเคชันบน ระบบ เมื่อทำการร้องขอจากเซิร์ฟเวอร์ ไคลเอ็นต์จำเป็นต้องทราบหมายเลขพอร์ต ที่เซิร์ฟเวอร์กำลังยอมรับคำร้องขอ หมายเลขพอร์ตนี้เชื่อมโยงกับ กับ **User Datagram Protocol (UDP)** หรือ **Transmission Control Protocol**

(TCP) ที่ถูกใช้โดยเซอร์วิส ไคลเอ็นต์รู้จักหมายเลขโปรแกรม หมายเลขเวอร์ชัน และชื่อระบบหรือชื่อโฮสต์ที่เซอร์วิสนั้น ตั้งอยู่ ไคลเอ็นต์ต้องการวิธีการแม็พหมายเลขโปรแกรมและหมายเลขเวอร์ชัน ที่คู่กับหมายเลขพอร์ตของแอ็พพลิเคชันเซิร์ฟเวอร์ ซึ่งทำกับวิธีใช้ **portmap** daemon

portmap daemon รันบนระบบที่เป็นแอ็พพลิเคชัน NFS เมื่อเซิร์ฟเวอร์เริ่มรัน เซิร์ฟเวอร์จะลงทะเบียนด้วย **portmap** daemon เนื่องจากการลงทะเบียนฟังก์ชันนี้ เซิร์ฟเวอร์จัดหาหมายเลขโปรแกรม หมายเลขเวอร์ชัน และหมายเลขพอร์ต **UDP** หรือ **TCP** **portmap** daemon เก็บตารางของแอ็พพลิคเชันเซิร์ฟเวอร์ เมื่อไคลเอ็นต์ลองทำการร้องขอเซิร์ฟเวอร์ ซึ่งติดต่อกับ **portmap** daemon เพื่อค้นหาพอร์ตที่เซิร์ฟเวอร์ใช้ **portmap** daemon ตอบกลับไคลเอ็นต์ด้วยพอร์ตของเซิร์ฟเวอร์ที่ไคลเอ็นต์กำลังร้องขอ ตามการรับของหมายเลขพอร์ต ไคลเอ็นต์สามารถทำการร้องขอในอนาคต โดยตรงกับเซิร์ฟเวอร์แอ็พพลิเคชัน

NFS แอ็พพลิเคชันและการควบคุม

NFS และ NIS daemons ถูกควบคุมโดย System Resource Controller (SRC)

นั่นหมายความว่า คุณต้องใช้คำสั่ง SRC เช่น **startsrc, stopsrc** และ **lssrc** เพื่อเริ่มต้นทำงาน หยุดทำงาน และตรวจสอบสถานะของ NFS และ NIS daemons

บาง NFS daemons จะไม่ถูกควบคุมโดย SRC โดยเฉพาะ **rpc.rexd, rpc.rusersd, rpc.rwalld** และ **rpc.rsprayd** จะไม่ถูกควบคุมโดย SRC daemons เหล่านี้ถูกสตาร์ท และหยุดโดย **inetd** daemon

ตารางต่อไปนี้จะลิสต์ daemon ที่ถูกควบคุมโดย SRC และชื่อระบบย่อยของมัน

ตารางที่ 92. Daemons และระบบย่อยของมัน

พาธไฟล์	ชื่อระบบย่อย	ชื่อกลุ่ม
/usr/sbin/nfsd	**nfsd**	nfs
/usr/sbin/biod	**biod**	nfs
/usr/sbin/rpc.lockd	**rpc.lockd**	nfs
/usr/sbin/rpc.statd	**rpc.statd**	nfs
/usr/sbin/rpc.mountd	**rpc.mountd**	nfs
/usr/sbin/nfsrgyd	**nfsrgyd**	nfs
/usr/sbin/gssd	**gssd**	nfs
/usr/lib/netsvc/yp/ypserv	**ypserv**	yp
/usr/lib/netsvc/yp/ypbind	**ypbind**	yp
/usr/lib/netsvc/rpc.yppasswdd	**yppasswdd**	yp
/usr/lib/netsvc/rpc.ypupdated	**ypupdated**	yp
/usr/sbin/keyserv	**keyserv**	keyserv
/usr/sbin/portmap	**portmap**	portmap

ข้อมูลที่เกี่ยวข้อง:

ภาพรวมของ System Resource Controller

การเปลี่ยนจำนวนของ biod และ nfsd daemons

คำสั่ง **chnfs** สามารถใช้เพื่อเปลี่ยนจำนวนสูงสุดของ **biod** หรือ **nfsd** daemons ที่จะรันบนระบบ

ตัวอย่างเช่น หากต้องการตั้งค่าจำนวนของสูงสุดของ **nfsd** daemon ให้มีค่า 1000 และจำนวนสูงสุดของ **biod** daemon ให้มีค่า 4 ให้รันคำสั่งต่อไปนี้:

```
chnfs -n 1000 -b 4
```

หมายเหตุ: คำสั่งนี้ จะหยุดการรัน daemon ในปัจจุบัน อัพเดตข้อมูลคอนฟิกูเรชัน SRC จากนั้น รีสตาร์ท daemon ตามผลลัพธ์ที่ได้ เซอร์วิส NFS จะไม่พร้อมใช้งานชั่วคราว
จำนวนสูงสุดของ **biod** daemons ยังถูกระบุต่อการเมาต์โดยใช้อ็อพชันการเมาต์ `biods=`*n*

หมายเหตุ: หากจำนวนของ **nfsd** ไม่ได้เพียงพอต่อการให้ไคลเอ็นต์ใช้ ข้อผิดพลาดการดำเนินการที่ไม่เหมือนกันถูกส่งคืนกลับไปยังไคลเอ็นต์ ตัวอย่างเช่น หากไคลเอ็นต์ลบไดเร็กทอรี ข้อผิดพลาด `ENOENT` ถูกส่งคืน แม้ว่า ไดเร็กทอรีบนเซิร์ฟเวอร์ที่ถูกลบทิ้ง

การเปลี่ยนอาร์กิวเมนต์บรรทัดรับคำสั่งสำหรับ daemons ที่ควบคุมโดย SRC

NFS และ NIS daemons จำนวนมากมีอาร์กิวเมนต์บรรทัดรับคำสั่งที่สามารถระบุได้ เมื่อเริ่มต้นทำงานกับ daemon เนื่องจาก daemon เหล่านี้ไม่ได้สตาร์ทโดยตรงจาก บรรทัดรับคำสั่ง ซึ่งคุณต้องอัพเดตฐานข้อมูล SRC เพื่อให้ daemons สามารถสตาร์ทได้อย่างถูกต้อง

หากต้องการทำสิ่งนี้ ให้ใช้คำสั่ง **chssys** คำสั่ง **chssys** มีรูปแบบ:

```
chssys -s Daemon -a 'NewParameter'
```

ตัวอย่าง เช่น:

```
chssys -s nfsd -a '10'
```

เปลี่ยนระบบย่อย **nfsd** ดังนั้น เมื่อสตาร์ท daemon แล้ว บรรทัดรับคำสั่งจะดูคล้ายกับ `nfsd 10` การเปลี่ยนแปลงที่ทำไว้โดยคำสั่ง **chssys** ไม่มีผลกระทบจนกว่าระบบย่อย ถูกหยุดทำงานและรีสตาร์ท

การสตาร์ท NFS daemons

ข้อจำกัดของขนาดไฟล์สำหรับไฟล์ที่วางอยู่บนเซิร์ฟเวอร์ NFS ถูกนิยามโดยสภาพแวดล้อมการประมวลผลเมื่อ **nfsd** ถูกสตาร์ท

หากต้องการใช้ค่าเฉพาะ ให้แก้ไขไฟล์ `/etc/rc.nfs` ใช้คำสั่ง **ulimit** พร้อมกับข้อจำกัดที่ต้องการก่อนคำสั่ง **startsrc** สำหรับ **nfsd** daemon

NFS daemons สามารถสตาร์ท แบบเดี่ยวๆ หรือทั้งหมดเพียงหนึ่งครั้ง หากต้องการสตาร์ท NFS daemons แบบเดี่ยวๆ ให้รัน:

```
startsrc -s Daemon
```

โดยที่ *Daemon* คือหนึ่งใน daemon ที่ควบคุม SRC ตัวอย่างเช่น หากต้องการสตาร์ท **nfsd** daemons ให้รัน:

```
startsrc -s nfsd
```

หากต้องการสตาร์ท NFS daemon ทั้งหมด ให้รัน:

```
startsrc -g nfs
```

หมายเหตุ: หากไฟล์ /etc/exports ไม่มีอยู่ **nfsd** และ **rpc.mountd** daemons จะไม่ถูกสตาร์ท คุณสามารถสร้างไฟล์ /etc/exports ที่ว่างเปล่าโดยรันคำสั่ง touch /etc/exports ซึ่งจะอนุญาตให้ **nfsd** และ **rpc.mountd** daemon สตาร์ท แม้ว่า ไม่มีระบบไฟล์จะถูกเอ็กซ์พอร์ต

การหยุด NFS daemons

NFS daemon สามารถหยุดแบบเดี่ยวๆ หรือหยุดทั้งหมดในหนึ่งครั้ง

หากต้องการหยุด NFS daemon แบบเดี่ยวๆ ให้รัน:

```
stopsrc -s Daemon
```

โดยที่ *Daemon* คือ SRC-controlled daemons ใดๆ ตัวอย่างเช่น หากต้องการหยุด **rpc.lockd** daemon ให้รัน:

```
stopsrc -s rpc.lockd
```

หากต้องการหยุด NFS daemon ทั้งหมด ให้รัน:

```
stopsrc -g nfs
```

การขอกรับสถานะปัจจุบันของ NFS daemons

คุณสามารถขอรับสถานะปัจจุบันของ NFS daemons เดี่ยวๆ ได้ หรือทั้งหมดเพียงครั้งเดียวได้

หากต้องการขอรับสถานะปัจจุบันของ NFS daemon แบบเดี่ยว ให้รัน:

```
lssrc -s Daemon
```

โดยที่ *Daemon* คือหนึ่งใน daemon ที่ควบคุม SRC ตัวอย่างเช่น หากต้องการขอรับสถานะปัจจุบันของ **rpc.lockd** daemon ให้รัน:

```
lssrc -s rpc.lockd
```

หากต้องการขอรับสถานะปัจจุบันของ NFS daemon ทั้งหมดเพียงครั้งเดียว ให้รัน:

```
lssrc
-a
```

ส่วนสนับสนุน NFS เวอร์ชัน 4

เริ่มต้นด้วย AIX 5.3 ส่วนสนับสนุนสำหรับโปรโตคอล NFS เวอร์ชัน 4 ที่สอดแทรกไว้

คุณลักษณะบังคับของโปรโตคอลถูกสนับสนุนเป็นการกล่าวถึงใน RFC 3530 พร้อมกับข้อยกเว้นต่อไปนี้:

- กลไกความปลอดภัย LIPKEY และ SPKM-3 ไม่สนับสนุนด้วยการพิสูจน์ตัวตน RPCSEC-GSS RPC เฉพาะกลไก Kerberos V5 ถูกสนับสนุน
- ข้อกำหนด UTF-8 ถูกสนับสนุนแบบเต็ม โดยเฉพาะอย่างยิ่ง การส่งข้อมูลของชื่อไฟล์แบะระบบไฟล์สตริง เช่น ลิงก์สัญลักษณ์เนื้อหาและชื่อรายการไดเร็กทอรี ถูกรับประกันที่ต้องอยู่ในรูปแบบ UTF-8 การส่งข้อมูลของสตริงแอ็ตทริบิวต์ NFS เช่น เจ้าของ และกลุ่มเจ้าของ อยู่ในรูปแบบ UTF-8 เซิร์ฟเวอร์ NFS และไคลเอ็นต์ดำเนินการกับการตรวจสอบความถูกต้อง UTF-8 บนข้อมูลสตริงขาเข้า ตามที่นิยาม RFC 3530 การตรวจสอบนี้สามารถปิดใช้งานโดยใช้คำสั่ง **nfso** การปิดใช้งานการตรวจสอบ UTF-8 อาจจำเป็น เพื่อใช้ NFS เวอร์ชัน 4 ในสภาพแวดล้อมด้วยคอนฟิกูเรชันและข้อมูล UTF-8
- ไคลเอ็นต์ Diskless, NIM และ UDP ไม่สนับสนุนผ่าน NFS เวอร์ชัน 4

คุณลักษณะอ็อพชันต่อไปนี้ของ NFS เวอร์ชัน 4 ถูกสนับสนุน:

- NFS เวอร์ชัน 4 ACLs ถูกสนับสนุนโดยไคลเอ็นต์และเซิร์ฟเวอร์ NFS ไคลเอ็นต์ NFS สนับสนุนการจัดการของ NFS เวอร์ชัน 4 ACL โดยใช้ยูทิลิตี **acledit, aclget** และ **aclput** เซิร์ฟเวอร์ NFS คือความสามารถของการเก็บ และการเรียก NFS เวอร์ชัน 4 ACLs ภายใต้ระบบไฟล์ที่สนับสนุน โมเดล NFS เวอร์ชัน 4 ACL สำหรับข้อมูลเพิ่มเติม ดูที่ “การสนับสนุนNFS แอ็คเซสคอนโทรล” ในหน้า 544
- ส่วนสนับสนุนถูกจัดเตรียมเพื่อแม็พหลักการและแอ็ตทริบิวต์เจ้าของไฟล์จากหนึ่งโดเมน NFS เวอร์ชัน 4 ลงในโดเมนอื่น ส่วนสนับสนุนนี้ถูกเจตนาสำหรับการใช้ ที่เซิร์ฟเวอร์ AIX NFS ซึ่งต้องการนำไปใช้งาน LDAP การแม็พ NFS ถูกจัดการโดยใช้ยูทิลิตี **chnfsim**

มีข้อควรพิจารณาต่างๆ เมื่อใช้การเข้าถึงด้วย NFS เวอร์ชัน 2 และ 3 และ NFS เวอร์ชัน 4 การเข้าถึง NFS เวอร์ชัน 3 อาจได้รับข้อผิดพลาด เนื่องจากสถานะ NFS เวอร์ชัน 4 ที่ได้รับอนุญาต และ ผลการทำงาน NFS เวอร์ชัน 3 อาจถูกปรับปรุงเมื่อข้อมูลถูกเอ็กซ์พอร์ตสำหรับการเข้าถึง NFS เวอร์ชัน 4

ช่วงเวลาผ่อนผันของเซิร์ฟเวอร์ NFS

โปรโตคอล NFS เวอร์ชัน 4 (NFSv4) จัดเตรียมการทำงานที่อนุญาตให้ ผู้ดูแลระบบเปิดใช้งานช่วงเวลาผ่อนผันบนเซิร์ฟเวอร์ NFSv4 สำหรับการจัดการของ การดำเนินการโดยเฉพาะ

ภายในช่วงเวลาผ่อนผันนี้ ผู้ดูแลระบบสามารถจัดการกับการล็อก การดำเนินการอ่าน และการดำเนินการเขียนสำหรับช่วงระยะเวลาทั้งหมดของสัญญาเช่าเซิร์ฟเวอร์ การล็อก และสถานะที่เชื่อมโยงสามารถกู้คืนได้โดยไคลเอ็นต์ผ่านคำร้องขอล็อกชนิดที่เรียกคืนได้

หมายเหตุ: ไม่ใช่สถานะทั้งหมดที่เรียกคืนได้โดยไคลเอ็นต์ในช่วงเวลาผ่อนผันสามารถรับประกัน เป็นสถานะที่ถูกพักโดยเซิร์ฟเวอร์ในอินสแตนซ์ก่อนหน้านี้ สถานะที่ถูกเรียกคืนในระหว่างช่วงเวลาผ่อนผันถูกรับประกันให้เป็นสถานะที่ถูกต้อง ตามที่นิยามไว้โดย NFSv4 RFC

การเริ่มต้นด้วย AIX 5L เวอร์ชัน 5.3 ที่มีระดับเทคโนโลยี 5300-05 ผู้ดูแลระบบสามารถใช้ช่วงเวลาผ่อนผันบนเซิร์ฟเวอร์ NFSv4 ช่วงเวลาผ่อนผันถูกปิดใช้งาน ตามค่าดีฟอลต์ หากต้องการเปิดใช้งานช่วงเวลาผ่อนผันบนเซิร์ฟเวอร์ ให้ใช้เมนู SMIT หรืออินเตอร์เฟสบรรทัดรับคำสั่ง **chnfs**

เมื่อเปิดใช้งานช่วงเวลาผ่อนผัน เซิร์ฟเวอร์ NFSv4 บันทึกข้อมูลสถานะ ลงในดิสก์ที่อยู่ในไฟล์ `/var` สถานะที่บันทึกไว้ถูกเรียกคืนแบบอัตโนมัติ เมื่อเซิร์ฟเวอร์รีสตาร์ท

ส่วนสนับสนุน NFS DIO และ CIO

AIX 5L เวอร์ชัน 5.3 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5300-03 สนับสนุน I/O และ I/O แบบพร้อมเพียงกันในไคลเอ็นต์ NFS สำหรับโปรโตคอลเวอร์ชัน 3 และ 4 DIO และ CIO เท่านั้นที่เกี่ยวข้องกับไคลเอ็นต์

การใช้ DIO และ CIO ศูนย์ข้อมูลเวิร์กโหลด เช่น ฐานข้อมูล และแอ็พพลิเคชันการคำนวณผลการทำงานสามารถพบกับระดับที่สูงกว่าของ ผลการทำงานควบคู่กันไปกับระบบ CPU และรีซอร์สหน่วยความจำที่ลดลง ขณะที่รักษาผลประโยชน์ของการจัดศูนย์กลางของหน่วยเก็บแบบอิงไฟล์ และการจัดการที่เชื่อมโยงกับระบบส่วนหลัง

I/O ไม่เป็นลำดับ และแอ็พพลิเคชันไม่มีประโยชน์ จากการทำแคชข้อมูลที่ไคลเอ็นต์ NFS หรือแอ็พพลิเคชันทำการแคช ระดับสูงทั้งหมด แอ็พพลิเคชันเหล่านี้มีประโยชน์เมื่อ NFS ไม่ได้แคช ให้ดำเนินการคาดการณ์การอ่านหรือใช้กลไกหลังการเขียน

นอกจากนี้ แอ็พพลิเคชันบางอย่าง เช่น ฐานข้อมูลไม่ได้ขึ้นอยู่กับซีแมนทิกส์ POSIX แบบไซต์เดียว ซึ่ง serialize การอ่านกับการเขียน แอ็พพลิเคชันเหล่านี้ใช้การอ่านและการเขียน แบบพร้อมเพียงกัน แต่แอ็พพลิเคชันเหล่านี้รับผิดชอบสำหรับความสอดคล้องกัน และการประสานงานของการดำเนินการ

I/O โดยตรงสำหรับ NFS

DIO อนุญาตให้แอ็พพลิเคชันเพื่อดำเนินการอ่านและเขียนไปยังเซิร์ฟเวอร์ NFS โดยตรงโดยไม่ได้ผ่านไปยังเลเยอร์การแคชไคลเอ็นต์ NFS (Virtual Memory Manager) หรือก่อให้เกิดค่าใช้จ่ายที่เชื่อมโยงของการแคชข้อมูล

ภายใต้ DIO คำร้องขอแอ็พพลิเคชัน I/O ถูกให้บริการโดยใช้ Remote Procedure Calls (RPC) ไปยังเซิร์ฟเวอร์ NFS โดยตรง คุณสามารถตั้งค่า DIO ได้โดยใช้อ็อพชันการเมาต์ AIX *dio* หากไม่มีอ็อพชันการเมาต์ คุณยังสามารถเปิดใช้งาน DIO ต่อไฟล์โดยใช้แฟล็ก **AIX O_DIRECT open( )**

การให้บริการ NFS direct I/Os อาจต้องการ RPC จำนวนมากกับเซิร์ฟเวอร์ขึ้นอยู่กับขนาดของคำร้องขอ I/O และขนาดการถ่ายโอนสูงสุดที่อนุญาตโดย เซิร์ฟเวอร์และไคลเอ็นต์ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ DIO โปรดดูอ็อพชัน **-o** สำหรับคำสั่ง **mount**

I/O แบบพร้อมเพียงกันสำหรับ NFS

ด้วย CIO การอ่านและเขียนแอ็พพลิเคชันอ่านและเขียนที่เรียกใช้พร้อมกันกับ การรันแบบพร้อมเพียงกันโดยไม่มีบล็อกการอ่านสำหรับช่วงระยะเวลาของการเขียน หรือกลับกัน

การเขียนจำนวนมากยังรันแบบพร้อมเพียงกัน การความเป็นอันหนึ่งอันเดียวกันของ POSIX ไม่ได้จัดเตรียมไว้ เมื่อ CIO มีผลบังคับใช้ I/O โดยตรงจะมีความหมาย ใช้อ็อพชันการเมาต์ AIX *cio* หรือแฟล็ก **O_CIO open( )** เพื่อตั้งค่า CIO สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ CIO โปรดดูอ็อพชัน **-o** สำหรับคำสั่ง **mount**

ใน AIX เวอร์ชัน 6.1 ที่มีระดับเทคโนโลยี 6100-04 และภายหลัง คุณสามารถรันคำสั่ง **mount** คำสั่ง **nfs4cl** หรือรูทีนย่อย **open( )** เพื่ออนุญาตให้คุณเปิดการอ่านไฟล์เท่านั้น เมื่อไฟล์เหล่านั้นถูกเปิดแล้วใน CIOR อ็อพชันการเมาต์ **cior** และแฟล็ก **O_CIOR open ( )** สามารถใช้ได้ในการเชื่อมกับ CIO

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง mount

การโต้ตอบของ DIO, CIO, regular opens และไฟล์ที่แม็พ สำหรับ NFS

ลักษณะการทำงานต่อไปนี้มีอยู่ระหว่างโหมดการเข้าถึงที่แตกต่างกัน ซึ่งสามารถเกิดขึ้นได้ด้วย DIO และ CIO

เมื่อ DIO เปิดที่มีอยู่มีผลบังคับใช้:

- การเปิดแบบปกติเป็นสาเหตุทำให้ DIO ถูกปิดจนกว่าจะไม่มีการเปิดแบบปกติ อีกต่อไป เมื่อการปิดลดการเปิดแบบปกติไปเป็น 0 DIO จะถูกเปิดใช้งานอีกครั้งหากยังคงมี DIO ที่เปิดอยู่ค้างอยู่
- การแม็พไฟล์ด้วย shmat( ) หรือ mmap( ) จะหยุดทำงาน DIO บนไฟล์ จนกว่าจำนวนของการแม็พจะตกลงที่ 0 ดังนั้น หากยังคงมี DIO เปิดอยู่ DIO จะถูกปิดใช้งาน
- ความพยายามในการเปิดไฟล์สำหรับ CIO จะล้มเหลวด้วยข้อผิดพลาด **EINVAL**

เมื่อมีการเปิดแบบปกติ (ไม่มี CIO หรือ DIO) ที่มีผลบังคับใช้:

- DIO เปิดมีความพยายามในการทำให้สำเร็จ แต่ DIO ไม่ได้ถูกเรียกใช้งานจนกว่า จำนวนของการเปิดแบบปกติตกอยู่ที่ 0
- การเปิดสำหรับ CIO จะล้มเหลวพร้อมกับข้อผิดพลาด **EINVAL**

เมื่อ CIO เปิดมีผลบังคับใช้:

- DIO ปกติและความพยายามในการแม็พไฟล์จะล้มเหลวทั้งหมด พร้อมกับข้อผิดพลาด **EINVAL**

เมื่อ CIO|CIOR เปิดมีผลบังคับใช้:

- DIO ปกติและความพยายามในการแม็พไฟล์จะล้มเหลวพร้อมกับข้อผิดพลาด **EINVAL** ยกเว้นการอ่านอย่างเดียวและ CIO|CIOR เปิด

หมายเหตุ: เมื่อมีการส่งผ่านไปยัง DIO หรือ CIO การแก้ไขแคชของไคลเอ็นต์ จะเขียนกลับไปยังเซิร์ฟเวอร์ NFS เป็นอันดับแรกก่อนที่จะลบ ข้อมูลที่แคชทั้งหมด

การจำลอง NFS และ namespace แบบโกลบอล

โปรโตคอล NFS เวอร์ชัน 4 (NFSv4) จัดเตรียมฟังก์ชันที่อนุญาตให้คุณ ซึ่งเป็นผู้ดูแลระบบแจกจ่ายข้อมูลระหว่างเซิร์ฟเวอร์จำนวนมาก ด้วยวิธีที่โปร่งใสต่อผู้ใช้ข้อมูลนั้น

คุณสามารถใช้สองคุณลักษณะที่จัดเตรียมไว้เพื่อเริ่มต้นด้วย AIX 5L เวอร์ชัน 5.3 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5300-03 อันดับแรกคือ คุณลักษณะ namespace แบบโกลบอลที่เรียกว่า *การอ้างอิง* อันดับที่สองคือ ความหมายของการระบุตำแหน่งที่ทำสำเนาของข้อมูลที่สามารถพบได้ ซึ่งเรียกว่า *เรพลิกา*

การอ้างอิง คืออ็อบเจ็กต์พิเศษที่คุณสามารถสร้างขึ้นใน namespace ของเซิร์ฟเวอร์กับข้อมูลตำแหน่งที่พ่วงต่อ เซิร์ฟเวอร์ใช้คุณลักษณะของโปรโตคอล NFSv4 เพื่อเปลี่ยนทิศทางไคลเอ็นต์ไปยังเซิร์ฟเวอร์ที่ระบุอยู่ในข้อมูล ตำแหน่ง การอ้างอิงจะจัดรูปแบบการสร้างบล็อกสำหรับข้อมูลที่รวบรวมไว้เกี่ยวกับเซิร์ฟเวอร์ NFS จำนวนมากไปเป็นแผนผังไฟล์ namespace เดี่ยวที่ไคลเอ็นต์ NFSv4 รับรู้การอ้างอิง ที่สามารถนำทางได้

เรพลิกา คือสำเนาของระบบไฟล์บนหนึ่งเซิร์ฟเวอร์ NFS ที่ถูกวางอยู่บนเซิร์ฟเวอร์ NFS อื่น (หรือที่ตำแหน่งที่เลือกไว้ เช่น ดิสก์อื่นๆ บนเซิร์ฟเวอร์เดียวกัน) หากกำหนดตำแหน่งเรพลิกาที่ใช้งานอยู่ โดยไคลเอ็นต์ NFSv4 ที่รับรู้ถึงเรพลิกาไม่พร้อมใช้งาน ไคลเอ็นต์จะสับเปลี่ยนไปเป็นเรพลิกาที่พร้อมใช้งาน ตัวอื่น สำหรับข้อมูลเพิ่มเติมเกี่ยวกับเรพลิกา โปรดอ้างถึง “เรพลิกา NFS” ในหน้า 561

การอ้างอิง NFS

ตัวอย่างต่อไปนี้จัดเตรียมสถานการณ์จำลองเพื่อช่วยคุณเข้าใจถึง การอ้างอิง

ในตัวอย่างต่อไปนี้ มีสี่เซิร์ฟเวอร์คือ:

- เซิร์ฟเวอร์ที่เรียกว่า `publications` มีไฟล์เอกสารคู่มือ
- เซิร์ฟเวอร์ที่เรียกว่า `projects` มีไดเร็กทอรีการทำงานของผู้ใช้
- เซิร์ฟเวอร์ที่เรียกว่า `data` มีฐานข้อมูลรายละเอียด
- เซิร์ฟเวอร์ที่เรียกว่า `account1` คือเซิร์ฟเวอร์หลัก NFS ที่เอ็กซ์พอร์ตไฟล์อื่นๆ ทั้งหมดและคือเซิร์ฟเวอร์ที่ไคลเอ็นต์ทั้งหมดรับทราบ

การอนุญาตให้ไคลเอ็นต์ทั้งหมดเข้าถึงไฟล์บนเซิร์ฟเวอร์หลัก NFS

เซิร์ฟเวอร์ `account1` เอ็กซ์พอร์ตไดเร็กทอรี `/work` ไปยังไคลเอ็นต์ทั้งหมดที่ใช้คำสั่งต่อไปนี้ในไฟล์ `/etc/exports`:

```
/work -vers=4
```

ไคลเอ็นต์ทั้งหมดสามารถเข้าถึงไฟล์ในรีโมตไดเร็กทอรี /work โดยเมาต์ / จากเซิร์ฟเวอร์ account1 บนไดเร็กทอรี /mnt โดยใช้คำสั่งต่อไปนี้:

```
mount -o vers=4 account1:/ /mnt
```

เมื่อผู้ใช้บนไคลเอ็นต์แสดงรายการเนื้อหาของไดเร็กทอรี /mnt ผู้ใช้จะมองเห็นรีโมตไดเร็กทอรี work ที่พาธ /mnt/work เนื้อหาของไดเร็กทอรี /mnt/work บนไคลเอ็นต์ ที่เหมือนกับเนื้อหาของไดเร็กทอรี /work บนเซิร์ฟเวอร์ account1

การอนุญาตให้ไคลเอ็นต์เข้าถึงไฟล์บนเซิร์ฟเวอร์ที่ระบุเฉพาะ

ผู้ใช้บนไคลเอ็นต์ยังต้องการเข้าถึงไดเร็กทอรี /usr/doc บนเซิร์ฟเวอร์ publications

ในรีลีสก่อนหน้านี้ คุณต้องเอ็กซ์พอร์ตไดเร็กทอรีจากเซิร์ฟเวอร์ และเมาต์ไดเร็กทอรีที่ไคลเอ็นต์

การใช้การอ้างอิงเพื่อสร้าง namespace ที่แจกจ่าย

คุณสามารถตั้งค่าเซิร์ฟเวอร์เพื่อให้ไคลเอ็นต์สามารถเข้าถึงข้อมูลบนเซิร์ฟเวอร์อื่นๆ โดยไคลเอ็นต์ไม่รู้ว่า ข้อมูลอยู่ที่ใด เฉพาะผู้ดูแลระบบบนเซิร์ฟเวอร์การอ้างอิงเท่านั้น ที่จำเป็นต้องรู้ว่า ข้อมูลอยู่ที่ใด เซิร์ฟเวอร์การอ้างอิงสามารถเปลี่ยนทิศทางไคลเอ็นต์ ไปยังตำแหน่งของไดเร็กทอรี /usr/doc โดยใช้การอ้างอิง บนเซิร์ฟเวอร์ publications ไดเร็กทอรี /usr/doc สามารถเอ็กซ์พอร์ตได้โดยเพิ่มคำสั่งต่อไปนี้ลงในไฟล์เอ็กซ์พอร์ต:

```
/usr/doc -vers=4
```

ซึ่ง ทำให้ไดเร็กทอรีพร้อมใช้งานกับไคลเอ็นต์ NFSv4

เซิร์ฟเวอร์ account1 สามารถใช้การอ้างอิงได้ในตอนนี้เพื่อทำให้ไดเร็กทอรี เหล่านั้นพร้อมใช้งานกับไคลเอ็นต์โดยเพิ่มคำสั่งต่อไปนี้ลงใน ไฟล์เอ็กซ์พอร์ต:

```
/usr/doc -vers=4,refer=/usr/doc@publications
```

จากนั้น คุณเอ็กซ์พอร์ตไดเร็กทอรี ที่จุดนี้ ไคลเอ็นต์ที่เมาต์กับไดเร็กทอรี /mnt จากไดเร็กทอรี / บนเซิร์ฟเวอร์ account1 มีสิทธิในการเข้าถึงไดเร็กทอรี usr เมื่อไคลเอ็นต์แสดงไดเร็กทอรี /mnt ไคลเอ็นต์ไม่จำเป็นต้องดำเนินการเมาต์ใดๆ กับเซิร์ฟเวอร์อื่นๆ ผู้ใช้บนไคลเอ็นต์ไม่จำเป็นต้องระวังว่า ไฟล์ไม่ได้ถูกจัดเตรียมไว้โดยเซิร์ฟเวอร์ account1 ตัวอย่างเช่น คุณสามารถทำให้ไดเร็กทอรี /databases/db บนเซิร์ฟเวอร์ data และ /home/accts บนเซิร์ฟเวอร์ projects ให้พร้อมใช้งานผ่าน account1 โดยเอ็กซ์พอร์ตไดเร็กทอรีจาก data และเซิร์ฟเวอร์ projects และการสร้างการอ้างอิงบน account1 กับไดเร็กทอรีเหล่านี้

เนื่องจากผู้ใช้บนไคลเอ็นต์ไม่ได้รู้ถึงตำแหน่งที่แน่นอนของข้อมูล ผู้ดูแลระบบสามารถเปลี่ยนทิศทางไคลเอ็นต์จากเซิร์ฟเวอร์หนึ่งไปยังอีกเซิร์ฟเวอร์หนึ่งแบบง่ายๆ โดยเปลี่ยนคำสั่งการอ้างอิงในเอ็กซ์พอร์ตไฟล์บนเซิร์ฟเวอร์ ผู้ดูแลระบบ รับผิดชอบต่อการหาตำแหน่งและความถูกต้องของข้อมูลที่อ้างอิงกับ ข้อกำหนดคุณสมบัติของตำแหน่ง

ผู้ดูแลระบบควรแน่ใจว่า เซิร์ฟเวอร์สำรองไม่อ้างอิงคำร้องขอ ที่ย้อนกลับไปยังเซิร์ฟเวอร์แรก ซึ่งจะสร้างการอ้างอิงแบบวน ในตัวอย่างข้างต้น หากผู้ดูแลระบบได้สร้างการอ้างอิงบนเซิร์ฟเวอร์ publications ที่ /usr/doc ซึ่งอ้างถึง /usr/doc บนเซิร์ฟเวอร์ account1 ผลลัพธ์ของการอ้างอิงแบบวน จะไม่เป็นที่ต้องการ

แม้ว่าการอ้างอิงถูกสร้างขึ้นโดยใช้ exportfs การอ้างอิงเหล่านั้นแตกต่างจากการเอ็กซ์พอร์ตข้อมูล ตำแหน่งที่ระบุสำหรับการอ้างอิง ควรตอบกลับไปยังไดเร็กทอรี root ของระบบไฟล์ NFSv4 ที่เอ็กซ์พอร์ต คุณสามารถสร้าง การอ้างอิงภายในเนมสเปซที่เอ็กซ์พอร์ต หรือในเนมสเปซที่ไม่ได้เอ็กซ์พอร์ต ในตัวอย่างข้างต้น การอ้างอิง /usr/doc สามารถถูกสร้างขึ้นบนเซิร์ฟเวอร์

account1 แม้ว่า /usr ไม่ได้ถูกเอ็กซ์พอร์ตก็ตาม ซึ่งจะวางการอ้างอิงอยู่ภายใน NFSv4 pseudospace หาก account1 ได้เอ็กซ์พอร์ต /usr แล้ว การเอ็กซ์พอร์ตการอ้างอิงยังได้รับอนุญาตให้ทำ ซึ่งต่างจากการเอ็กซ์พอร์ตไดเร็กทอรีที่เรียกว่า doc ซึ่งจะมีความล้มเหลว หากอยู่ในระบบไฟล์เดียวกัน ในกรณีใดๆ เหล่านี้ การเอ็กซ์พอร์ตการอ้างอิงจะล้มเหลว หากไฟล์หรือไดเร็กทอรีมีอยู่แล้วที่ /usr/doc ไม่มีข้อจำกัดเกี่ยวกับจำนวนของการอ้างอิงที่สามารถสร้างขึ้นภายใน NFSv4 pseudospace ของเซิร์ฟเวอร์หรือภายในระบบไฟล์ที่เอ็กซ์พอร์ต อย่างใดอย่างหนึ่ง

เนื่องจาก การอ้างอิงไม่ได้เอ็กซ์พอร์ตข้อมูลใดๆ และมีความหมายต่อโปรโตคอล NFSv4 เท่านั้น การอ้างอิงจะพร้อมใช้งานใน NSFv4 เท่านั้น การเอ็กซ์พอร์ตที่อ้างอิงโดยไม่มีอ็อพชัน vers=4 จะล้มเหลว แม้ว่าตัวอย่างนี้ระบุเพียงหนึ่งตำแหน่งเท่านั้น แต่สามารถระบุได้สูงสุด 8 ตำแหน่ง

การสร้างการอ้างอิงจะสร้างอ็อบเจ็กต์การอ้างอิงพิเศษที่ตำแหน่งที่ระบุไว้โดย พารามิเตอร์ไดเร็กทอรี เนื่องจากการเข้าถึงไคลเอ็นต์กับอ็อบเจ็กต์ถูกพิจารณาโดย การเข้าถึงไคลเอ็นต์กับไดเร็กทอรีหลักของอ็อบเจ็กต์ ซึ่งอ็อพชันการเอ็กซ์พอร์ตอื่นๆ โดยส่วนใหญ่ไม่มีความหมาย และอนุญาตให้ใช้และละเว้นได้ มีเพียงข้อยกเว้น exname เท่านั้นซึ่งจะมีลักษณะการทำงานตามที่คาดการณ์ไว้ ตัวอย่างเช่น หากเซิร์ฟเวอร์สร้างการอ้างอิง /n4root/special/users -vers=4,exname=/exported/users,refer=/restricted/users@secrethost แล้ว ไคลเอ็นต์ที่เมาต์ / จากเซิร์ฟเวอร์จะมองเห็นพาธ /mnt/exported/users ซึ่งจะเปลี่ยนทิศทางไคลเอ็นต์ไปเป็น ไดเร็กทอรี /restricted/users บน secrethost สำหรับการเอ็กซ์พอร์ตเซิร์ฟเวอร์ อ็อบเจ็กต์ที่อ้างอิงจะถูกสร้างขึ้นใน namespace บนโลคัลที่ /n4root/special/users ดังนั้น จึงไม่มีไฟล์หรือไดเร็กทอรีที่สามารถมีอยู่ที่นั้น เมื่อเอ็กซ์พอร์ตเสร็จสิ้น อ็อบเจ็กต์พิเศษ ที่สร้างบนเซิร์ฟเวอร์เพื่อพักข้อมูลตำแหน่งการอ้างอิง ไดเร็กทอรีใดๆ พร้อมกับพาธไปยังการอ้างอิงจะยังถูกสร้างขึ้น หากไม่มีอยู่ หากการอ้างอิงไม่ได้ถูกเอ็กซ์พอร์ต ข้อมูลการอ้างอิงจะถูกลบออกจากอ็อบเจ็กต์ แต่ตัวอ็อบเจ็กต์เองจะไม่ถูกลบทิ้ง เซิร์ฟเวอร์ NSFv4 จะไม่อนุญาตให้ไคลเอ็นต์เข้าถึงผลลัพธ์ของอ็อบเจ็กต์การอ้างอิง *stale* หรือ *orphan* ซึ่งจะส่งคืนข้อผิดพลาดในการเข้าถึงไคลเอ็นต์ที่พยายามเข้าถึง อ็อบเจ็กต์ อ็อบเจ็กต์สามารถลบทิ้งได้โดยใช้ **rm** หากต้องการ การอ้างอิงสามารถเอ็กซ์พอร์ตได้อีกครั้งด้วยข้อมูลการอ้างอิงใหม่ ซึ่งไม่เป็นข้อแนะนำ เนื่องจากความถี่ในการฝึกปฏิบัติ เพราะอาจใช้เวลาบางส่วนสำหรับไคลเอ็นต์ เพื่อให้เข้าถึงการอ้างอิงที่จดจำได้ว่า ข้อมูลตำแหน่ง มีการเปลี่ยนแปลง เซิร์ฟเวอร์ติดต่อกับไดเร็กทอรีหลักของการอ้างอิง เพื่อบ่งชี้ถึงข้อมูลในไดเร็กทอรีที่ได้ถูกเปลี่ยนแปลง ซึ่งจะช่วยให้ไคลเอ็นต์ จดจำข้อมูลใดๆ ที่ไคลเอ็นต์ได้ถูกแคชเกี่ยวกับไดเร็กทอรี (และการอ้างอิงภายในไดเร็กทอรี) เปลี่ยนแปลงไป และต้องการให้ล้างข้อมูลอีกครั้ง แต่ไม่มีการรับประกันถึงระยะเวลาที่ใช้สำหรับไคลเอ็นต์เพื่อสังเกต

สำหรับข้อมูลเกี่ยวกับการใช้อ็อพชัน **refer** เพื่อเปลี่ยนลำดับของตำแหน่ง ที่ระบุอยู่ในรายการตำแหน่งระบบไฟล์ โปรดดู การเรียงลำดับตำแหน่งระบบไฟล์ โดยใช้อ็อพชัน **scatter**

เรพลิกา NFS

เรพลิเคชันอนุญาตให้คุณในฐานะผู้ดูแลระบบวางสำเนาของข้อมูล บนเซิร์ฟเวอร์ NFSv4 จำนวนมากและแจ้งให้ไคลเอ็นต์ NFSv4 ทราบถึงตำแหน่งที่เรพลิกา ตั้งอยู่

ในเหตุการณ์ที่เซิร์ฟเวอร์ข้อมูลหลักกลับกลายไม่สามารถเข้าถึงไคลเอ็นต์ได้ ไคลเอ็นต์สามารถใช้หนึ่งในเซิร์ฟเวอร์เรพลิกาดำเนินการต่อบน ระบบไฟล์ที่ทำเรพลิเคทแล้ว ระบบไฟล์ที่ทำเรพลิเคทแล้วถูกสมมติขึ้นเป็นสำเนาของข้อมูลที่แน่นอน บนเซิร์ฟเวอร์หลัก คุณสามารถตั้งค่าได้ถึง 8 ตำแหน่งของเรพลิกา เซิร์ฟเวอร์ AIX ไม่สามารถระบุระบบไฟล์ที่เรพลิกาซึ่งถูกสร้างจากระบบไฟล์หลัก หรือวิธีการเก็บข้อมูลเป็นกลุ่มก้อนได้ หากคุณกำลังระบุเรพลิกาเป็นการอ่าน–เขียน คุณต้องเก็บข้อมูลไว้บนเรพลิกาที่เป็นกลุ่มก้อนด้วยระบบไฟล์หลัก

เรพลิกาคือเซิร์ฟเวอร์ที่มีสำเนาของไดเร็กทอรี หรือไดเร็กทอรีต่างๆ ของเซิร์ฟเวอร์ หากเซิร์ฟเวอร์หลักกลับกลายเป็นว่าไม่พร้อมใช้งานกับไคลเอ็นต์ ไคลเอ็นต์สามารถเข้าถึงไฟล์เดียวกันได้จากตำแหน่งของเรพลิกา สถานการณ์จำลองต่อไปนี้ คือตัวอย่าง:

หากไฟล์อยู่ในไดเร็กทอรี /data บนเซิร์ฟเวอร์ account1 ยังพร้อมใช้งานในไดเร็กทอรี /backup/data บนเซิร์ฟเวอร์ inreserve ไคลเอ็นต์ NSFv4 สามารถรับรู้ถึงสิ่งนี้ได้โดย ระบุตำแหน่งเรพลิกาบนเอ็กซ์พอร์ต ด้วยการเพิ่มคำสั่ง ที่คล้ายกับคำสั่งที่อยู่ในเอ็กซ์พอร์ตไฟล์ คุณสามารถเอ็กซ์พอร์ตไดเร็กทอรี /data และระบุตำแหน่งของสำเนาเรพลิกา:

```
/data -vers=4,replicas=/data@account1:/backup/data@inreserve
```

หาก เซิร์ฟเวอร์ account1 ไม่พร้อมใช้งาน ผู้ใช้ ไคลเอ็นต์ที่ใช้ไฟล์ภายใต้ไดเร็กทอรี /data ของเซิร์ฟเวอร์ account1 สามารถเริ่มต้นใช้ไฟล์ในไดเร็กทอรี /backup/data บนเซิร์ฟเวอร์ inreserve ได้โดยไม่ต้องรับทราบว่า ไคลเอ็นต์ได้สับเปลี่ยนไปเป็นเซิร์ฟเวอร์อื่นแล้ว

สำหรับข้อมูลเกี่ยวกับการใช้อ็อพชัน **replicas** เพื่อเปลี่ยนลำดับของตำแหน่ง ที่ระบุอยู่ในรายการตำแหน่งระบบไฟล์โปรดดู การเรียงลำดับตำแหน่งระบบไฟล์โดยใช้อ็อพชัน **scatter**

ข้อกำหนดคอนฟิกูเรชัน NFS เพื่ออนุญาตให้ข้อกำหนดคุณสมบัติของเรพลิกา:

คุณต้องเป็นผู้ดูแลระบบเพื่อเปิดใช้งาน ปิดใช้งาน หรือระบุเรพลิกา สำหรับ root

หากต้องการเปิดใช้งาน ปิดใช้งาน และระบุเรพลิกา root ให้ใช้คำสั่งต่อไปนี้:

```
chnfs -R {on|off|host[+host]}
```

หากต้องการ ระบุเรพลิกา เซิร์ฟเวอร์ต้องถูกตั้งค่าไว้ด้วย **chnfs -R** (**chnfs -R** on) เพื่อเรียกใช้การจัดการไฟล์ NFSv4 ที่ลบเลือนได้ การจัดการไฟล์คือตัวระบุที่เซิร์ฟเวอร์ NFS เรียกใช้ไคลเอ็นต์ เพื่อระบุไฟล์หรือไดเร็กทอรีบนเซิร์ฟเวอร์ ตามค่าดีฟอลต์ เซิร์ฟเวอร์ออกใช้ การจัดการไฟล์แบบถาวร การสับเปลี่ยนระหว่างชนิดของการจัดการไฟล์ สามารถส่งผลให้เกิดข้อผิดพลาดกับแอ็พพลิเคชันที่ไคลเอ็นต์ NFSv4 ซึ่งกำลังใช้เซิร์ฟเวอร์ที่แอ็คทีฟ เมื่อการสับเปลี่ยนถูกดำเนินการ เมื่อต้องการเปลี่ยนโหมดการจัดการไฟล์ด้วย **chnfs -R** ไม่มีระบบไฟล์ที่สามารถเอ็กซ์พอร์ตสำหรับการเข้าถึง NFSv4 ค่าติดตั้งการควบคุมการจัดการไฟล์ ควรทำขึ้นกับเซิร์ฟเวอร์ NFS ที่ถูกจัดเตรียมไว้ใหม่ หรือทำขึ้นเมื่อกิจกรรม NFS สามารถลดจำนวนลงหรือหยุดทำงาน สำหรับไคลเอ็นต์ที่เชื่อมต่อกับเซิร์ฟเวอร์ ที่แอ็คทีฟ เมื่อโหมดเปลี่ยนแปลงไป ไคลเอ็นต์นั้นอาจจำเป็นต้องยกเลิกการเมาต์ และเมาต์ NFSv4 อีกครั้งบนไคลเอ็นต์เหล่านั้น หากต้องการทำให้การดำเนินการนี้มีขนาดเล็กลง จำนวนของไคลเอ็นต์ที่เมาต์สามารถลดลงให้เป็นจำนวนของการเมาต์ขนาดเล็กที่เมาต์ ไดเร็กทอรีระดับบนสุดของพื้นที่ไฟล์เอ็กซ์พอร์ตของเซิร์ฟเวอร์ NFSv4

ไคลเอ็นต์ NFSv4 ไม่สามารถล้มเหลวในกับเรพลิกาที่มีคุณสมบัติการเข้าถึง ที่เอ็กซ์พอร์ตต่างกัน ผู้ดูแลระบบต้องตรวจสอบให้แน่ใจว่า เรพลิกาทั้งหมดถูกระบุไว้ด้วย การควบคุมการเข้าถึงแบบเอ็กซ์พอร์ตเดียวกันและโหมดการเข้าถึง (อ่านอย่างเดียวหรืออ่าน-เขียน) ด้วยข้อยกเว้นที่อาจเกิดขึ้นได้ของ GPFS™ ที่ถูกเอ็กซ์พอร์ต ข้อยกเว้นถูกคาดการณ์ว่าข้อมูลที่ถูกเรพลิกาจะถูกเอ็กซ์พอร์ตไว้แบบอ่านอย่างเดียว ซึ่งยังเป็นความรับผิดชอบของผู้ดูแลระบบ เพื่อคงไว้ซึ่งเนื้อหาข้อมูลที่ตำแหน่งเรพลิกาทั้งหมด แผนผังไดเร็กทอรีและเนื้อหาข้อมูลทั้งหมด ควรเก็บไว้โดยเฉพาะ อัพเดตไปยังเนื้อหาข้อมูลจำเป็นต้อง ถูกดำเนินการด้วยวิธีที่เข้ากันได้กับแอ็พพลิเคชัน ที่จะใช้ข้อมูล

ด้วยเรพลิกา คุณสามารถใช้อ็อพชันของการเอ็กซ์พอร์ต **exname** เพื่อซ่อนรายละเอียดของ namespace ระบบไฟล์บนโลคัลของเซิร์ฟเวอร์จากไคลเอ็นต์ NFSv4 สำหรับรายละเอียดเพิ่มเติม โปรดดูคำอธิบายคำสั่ง **exportfs** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2* และคำอธิบายไฟล์ /etc/exports ใน *การอ้างอิงไฟล์*

คุณสามารถใช้อ็อพชัน **replicas** พร้อมกับระบบไฟล์คลัสเตอร์ที่เอ็กซ์พอร์ต เช่น General Parallel File System (GPFS) เพื่อระบุเซิร์ฟเวอร์โหนด NFS จำนวนมากที่ดูมุมมอง GPFS เดียวกัน นี่คือคอนฟิกูเรชันที่เอ็กซ์พอร์ตข้อมูลสำหรับการเข้าถึงการอ่าน-เขียน ซึ่งอาจใช้งานได้ อย่างไรก็ตาม ด้วยเรพลิกาการอ่าน-เขียน หากความล้มเหลวของเรพลิกาเกิดขึ้นขณะที่การดำเนิน

การเขียนกำลังดำเนินอยู่ แอ็พพลิเคชันที่ดำเนินการเขียนอาจพบกับข้อผิดพลาดที่ไม่สามารถกู้คืนได้ เช่นเดียวกัน **mkdir** หรือไฟล์เฉพาะที่สร้างการดำเนินการในการรัน ในระหว่างความล้มเหลวอาจพบกับข้อผิดพลาด **EXISTS**

การเอ็กซ์พอร์ตที่ทำเรพลิเคทแล้วต้องเอ็กซ์พอร์ตระบบไฟล์ทั้งหมด ซึ่งหมายความว่า ไดเร็กทอรีที่เอ็กซ์พอร์ตต้องเป็น root ของระบบไฟล์แบบโลคัล เซิร์ฟเวอร์ ที่เอ็กซ์พอร์ตระบบไฟล์ที่ทำเรพลิเคทแล้วควรระบุตัวเองเป็นหนึ่งในตำแหน่ง สำหรับเอ็กซ์พอร์ต สำหรับเซิร์ฟเวอร์ที่มีอินเตอร์เฟสจำนวนมาก สิ่งนี้ต้องสอดแทรกชื่อโฮสต์หลักของเซิร์ฟเวอร์ หากเซิร์ฟเวอร์ที่เอ็กซ์พอร์ตระบบไฟล์ที่ทำเรพลิเคทแล้ว ไม่ระบุตัวเองเป็นหนึ่งในตำแหน่งสำหรับเอ็กซ์พอร์ต เซิร์ฟเวอร์การเอ็กซ์พอร์ตจะถูกเพิ่มไปยังรายการของตำแหน่งเรพลิกาแบบ silent เป็นตำแหน่งเรพลิกาตำแหน่งแรก การเรียงลำดับของตำแหน่งเรพลิกาในรายการเรพลิกา จะระบุเการเรียงลำดับของการมาก่อนของการกำหนดค่าตามความชอบของไคลเอ็นต์ที่ควรใช้ เมื่อเกิดความล้มเหลว ตัวอย่างเช่น หากผู้ใช้ที่ `serverA` ต้องการเอ็กซ์พอร์ต `/webpages` และมีเรพลิกาของ `/webpages` บน `serverB` ในไดเร็กทอรี `/backup/webpages` รายการต่อไปนี้ในไฟล์ `/etc/exports` จะเอ็กซ์พอร์ต `/webpages` จาก `serverA` และแจ้งให้ไคลเอ็นต์ทราบที่มีสำเนาของระบบไฟล์บน `serverB` ที่ `/backup/webpages`:

```
/webpages -vers=4,ro,replicas=/webpages@serverA:
/backup/webpages@serverB
```

ทั้ง `/webpages` บน `serverA` และ `/backup/webpages` บน `serverB` ถูกสันนิษฐานว่าต้องเป็นไดเร็กทอรี root ของระบบไฟล์ หาก `serverA` ไม่ได้แสดงอยู่ในเอ็กซ์พอร์ต เซิร์ฟเวอร์จะ ถูกเพิ่มแบบไม่โต้ตอบไปยังตำแหน่งเรพลิกาเป็นอันดับแรก ซึ่งเป็นเพราะ เซิร์ฟเวอร์ที่เอ็กซ์พอร์ตข้อมูลถูกสมมติว่าเป็นเซิร์ฟเวอร์ที่ต้องการสำหรับข้อมูล ที่กำลังเอ็กซ์พอร์ต

เรพลิกาจะถูกใช้โดยโปรโตคอล NFSv4 เท่านั้น การเอ็กซ์พอร์ตข้างต้นสามารถระบุ NFSv3 (`vers=3:4`) ได้ แต่ข้อมูลการจำลอง จะไม่มีอยู่ในไคลเอ็นต์ NFSv3 อย่างไรก็ตาม ไคลเอ็นต์ที่กำลังใช้ NFSv3 เข้าถึงข้อมูลใน `/webpages` บน `serverA` แต่จะไม่ล้มเหลวในการเรพลิกา หาก `serverA` ไม่พร้อมใช้งาน

ส่วนสนับสนุนในฝั่งไคลเอ็นต์ NFS สำหรับตำแหน่งจำนวนมาก:

เมื่อไคลเอ็นต์ไม่สามารถเข้าถึงข้อมูลที่เรพลิเคทได้ยาวนานกว่าจากเซิร์ฟเวอร์ปัจจุบัน ไคลเอ็นต์จะพยายามเข้าถึงข้อมูลจากเซิร์ฟเวอร์แบบ next-most-favored

ลำดับที่ระบุเรพลิกาในรายการเรพลิกาถูกใช้โดยไคลเอ็นต์ เพื่อเรียงลำดับการกำหนดค่าตามความชอบ

ผู้ดูแลระบบไคลเอ็นต์สามารถแทนทับการกำหนดค่าตามความชอบของเรพลิกาโดยใช้คำสั่งย่อย **prefer** ของคำสั่ง **nfs4cl** คำสั่ง **nfs4cl** แสดงข้อมูลระบบไฟล์ทั้งหมดบนไคลเอ็นต์หรือแก้ไขอ็อพชันระบบไฟล์ ของระบบไฟล์และแสดงหรือแก้ไขสถิติและคุณสมบัติ NFSv4 ปัจจุบัน

ข้อควรพิจารณาทั่วไปเกี่ยวกับ NFS สำหรับเรพลิกาและการอ้างอิง:

หากไคลเอ็นต์พบกับพาธที่แตกต่างกันสองพาธซึ่งนำไปสู่ข้อมูล (ระบบไฟล์) เดียวกัน ไคลเอ็นต์จะใช้ทั้งสองพาธเป็นลิงก์สัญลักษณ์ไปยัง ไฟล์

ตัวอย่างเช่น `server A` เอ็กซ์พอร์ต:

```
/tmp/a              -vers=4,replicas=/tmp/a@B:/tmp/a@A
/tmp/b              -vers=4,refer=/tmp/a/b@B
```

และ `server B` exports:

```
/tmp/a                -vers=4
/tmp/a/b              -vers=4
```

ในตัวอย่างนี้ ไคลเอ็นต์เมาต์ / บน server A กับ /mnt โดยใช้คำสั่ง mount -o vers=4 A:/ /mnt ผู้ใช้ไคลเอ็นต์เข้าถึง /tmp/a/b บน server B ผ่าน cd /mnt/tmp/a/b or cd /mnt/tmp/b หากผู้ใช้เปลี่ยนไดเร็กทอรีไปเป็น cd /mnt/tmp/a/b ในครั้งแรก จากนั้น พาธ /mnt/tmp/b ทำหน้าที่เป็นลิงก์สัญลักษณ์ไปยัง /mnt/tmp/a/b ในสถานการณ์จำลองนี้ หากผู้ใช้อยู่ใน /mnt/tmp/b และใช้คำสั่ง /bin/pwd, /bin/pwd > จะส่งคืน /mnt/tmp/a/b

หมายเหตุ: การฝึกปฏิบัติข้างต้น ไม่ใช่ข้อแนะนำ ผู้ดูแลระบบควรตั้งค่าข้อกำหนดคุณสมบัติของการเอ็กซ์พอร์ต ที่ส่งผลทำให้เกิดหนึ่งพาธของ namespace ที่อาจเป็นไปได้ไปยังข้อมูลที่เอ็กซ์พอร์ต

คุณสามารถแสดงรายการตำแหน่งจำนวนมากในการอ้างอิงหากข้อมูลเป้าหมายของการอ้างอิง ถูกเรพลิเคท ไคลเอ็นต์จะใช้เฉพาะตำแหน่งที่อ้างอิงเท่านั้น เพื่อค้นหาเป้าหมายการอ้างอิงที่เซิร์ฟเวอร์ที่มีอยู่ หากไคลเอ็นต์สร้างการเข้าถึงการอ้างอิงเป้าหมายแล้ว ไคลเอ็นต์จะขอรับข้อมูลตำแหน่งใหม่สำหรับ ข้อมูลที่พบ

เนื่องจากไคลเอ็นต์อาจไม่ได้ตรวจพบการเปลี่ยนแปลงในทันทีเกี่ยวกับข้อมูลตำแหน่งการอ้างอิง การลบหรือการเปลี่ยนตำแหน่งบ่อยๆ ไม่เป็นข้อแนะนำให้กระทำ เมื่อเปลี่ยนตำแหน่ง เป้าหมายของตำแหน่งการอ้างอิง ขอแนะนำว่า ตำแหน่งใหม่จะถูกใส่ข้อมูลพร้อมกับการเปลี่ยนข้อมูลตำแหน่งใน ข้อกำหนดคุณลักษณะของการอ้างอิงของเอ็กซ์พอร์ต ข้อมูลที่ตำแหน่งเก่าควรถูกเก็บไว้ เป็นเวลาหลายชั่วโมงหรือหลายวันเพื่อกำหนดเวลาให้ไคลเอ็นต์ดูหรือใช้ ตำแหน่งใหม่

ทั้งการจำลองและการอ้างอิงสามารถรันบนเซิร์ฟเวอร์ที่รันเคอร์เนลแบบ 64 บิต ไคลเอ็นต์สามารถรันบนเคอร์เนลแบบ 32 และ 64 บิตได้

หากคุณกำลังระบุเรพลิกาเป็นการอ่าน-เขียน คุณต้องเก็บข้อมูลบนกลุ่มของเรพลิกา พร้อมกับชุดของไฟล์หลัก

ไคลเอ็นต์ NFS ล้มเหลวได้อย่างไร:

fail-over คือ เมื่อไคลเอ็นต์สับเปลี่ยนจาก ตำแหน่งเรพลิกาหนึ่งไปยังอีกตำแหน่งหนึ่งหลังจากที่พิจารณาว่า เซิร์ฟเวอร์ปัจจุบันที่กำลังสื่อสารไม่สามารถเข้าถึงได้

การปรับเปลี่ยนต่อไปนี้นำไปสู่ความล้มเหลวของไคลเอ็นต์ NFS:

อ็อพชันการเมาต์ NFS *timeo*
: อ็อพชันการเมาต์นี้ระบุเวลาที่เลเยอร์ TCP/IP ต้องรอ ก่อนที่จะส่งคืนด้วยการตอบกลับการหมดเวลาใช้งาน

อ็อพชันการเมาต์ NFS *retrans*
: อ็อพชันการเมาต์นี้ระบุจำนวนครั้งที่เลเยอร์ NFS RPC ควรลองคำร้องขอของไคลเอ็นต์ก่อนที่จะส่งคืนข้อผิดพลาดการหมดเวลาใช้งาน RPC (**ETIMEDOUT**)

อ็อพชัน nfso *nfs_v4_fail_over_timeout*
: คุณสามารถใช้อ็อพชันนี้ **nfso** เพื่อระบุจำนวนต่ำสุดของเวลา ที่ต้องรอก่อนที่จะเกิดความล้มเหลวกับเรพลิกา อ็อพชันนี้คือโกลบอลกับไคลเอ็นต์ NFS และแทนทับค่าดีฟอลต์ต่อลักษณะของ การเมาต์ ตามค่าดีฟอลต์แล้ว **nfs_v4_fail_over_timeout** ไม่แอ็คทีฟ ซึ่งค่าคือ 0

เมื่อ **nfs_v4_fail_over_timeout** ไม่แอ็คทีฟ ความล้มเหลวของ threshold จะตั้งค่าเป็นการเมาต์ค่าอ็อพชัน **timeo** สองครั้ง เมื่อไม่มีการเรียก RPC ที่เป็นผลสำเร็จเกิดขึ้นสำหรับระยะเวลานี้ ไคลเอ็นต์จะเริ่มต้นประมวลผลความล้มเหลวเพื่อค้นหาเรพลิกาที่พร้อมใช้งาน อย่างไรก็ตาม เวลาจริงที่ไคลเอ็นต์จะรอมีอิทธิพลต่ออ็อพชัน **retrans** หาก **retrans** มีค่ามากกว่า 2 ไคลเอ็นต์จะ

รอ จนกระทั่งรับค่าการหมดเวลาใช้งาน RPC อ้างอิงตามค่า **retrans** คูณด้วยค่า **timeo** (`retrans × timeo`) ดังนั้น การรวมกันของอ็อพชัน **timeo** และ **retrans** สามารถปรับเปลี่ยนเพื่อควบคุมลักษณะความล้มเหลวต่อการเมาต์ NFS คุณยังสามารถตั้งค่าอ็อพชันเหล่านี้ที่ระดับเล็กๆ โดยใช้คำสั่ง **nfs4cl**

เมื่อ **nfso nfs_v4_fail_over_timeout** ถูกตั้งค่าที่ไม่ใช่ศูนย์ ค่านั้นจะแสดงจำนวนวินาทีที่ไคลเอ็นต์จะรอ อยู่บนเซิร์ฟเวอร์ที่ไม่พร้อมใช้งานก่อนที่จะพิจารณาความล้มเหลวของเรพลิกา หากอ็อพชัน **timeo** และ **retrans** ส่งผลให้ลักษณะการหมดเวลาใช้งาน RPC เข้าใกล้ค่าติดตั้ง **nfso** การประมวลผลความล้มเหลวอาจไม่สตาร์ทจนกว่าจะสร้างการหมดเวลาใช้ RPC

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับอ็อพชัน **retrans, timeo** และ **nfs_v4_fail_over_timeout** โปรดอ้างถึง อ็อพชันที่ระบุเฉพาะ NFS ของคำสั่ง **mount, nfs4cl** และ **nfso**

นอกจากความล้มเหลวของเรพลิกาในเหตุการณ์ของเซิร์ฟเวอร์ที่ไม่พร้อมใช้งานแล้ว มีกรณีที่ไคลเอ็นต์จะสับเปลี่ยนจากหนึ่งตำแหน่งเรพลิกา ไปเป็นตำแหน่งอื่น หนึ่งกรณีคือ เมื่อคุณใช้คำสั่ง **nfs4cl** เพื่อสร้างเรพลิกาที่ต้องการ ในกรณีนี้ ไคลเอ็นต์จะเริ่มต้นสับเปลี่ยนเซิร์ฟเวอร์ ที่ต้องการ หากไม่ใช่เซิร์ฟเวอร์ปัจจุบันที่ไคลเอ็นต์ใช้ ไคลเอ็นต์ยังล้างข้อมูลตำแหน่งเรพลิกาอีกครั้งจากเซิร์ฟเวอร์ NFS ประมาณ 30 นาทีตามความเหมาะสม เมื่อมีกิจกรรมล่าสุดบน ข้อมูลที่เชื่อมโยง หากการเรียงลำดับตำแหน่งเปลี่ยนแปลงไป ไคลเอ็นต์ จะพยายามสับเปลี่ยนตำแหน่งแรก หากแตกต่างจากเซิร์ฟเวอร์ปัจจุบัน ที่ไคลเอ็นต์ใช้และคุณไม่ได้ตั้งค่าการกำหนดค่าตามความชอบของเรพลิกาด้วยคำสั่ง **nfs4cl**

การเมาต์ NFS แบบชั่วคราวกับลักษณะการทำงานที่ล้มเหลว:

โมเดลการเมาต์แบบดีฟอลต์สำหรับ NFS คือการเมาต์แบบถาวรและลักษณะการทำงานที่ความล้มเหลว ของเรพลิกาใช้กับการเมาต์แบบถาวร ลักษณะการทำงานที่ล้มเหลวแตกต่างกัน หากการเมาต์ NFS แบบชั่วคราวถูกนำมาใช้

หากค่าติดตั้งการเมาต์แบบชั่วคราวส่งผลทำให้การหมดเวลาใช้งาน RPC ก่อนระยะเวลารอที่สร้างไว้ สำหรับความล้มเหลวของเรพลิกา ดังนั้น การหมดเวลาใช้งานจะส่งผลทำให้เกิดข้อผิดพลาด **ETIMEDOUT** กับการเรียกแอ็พพลิเคชัน การใช้เมาต์แบบชั่วคราวพร้อมกับข้อมูลที่เรพลิเคทไม่ได้ถูกแนะนำไว้ หากการเมาต์แบบชั่วคราว ถูกใช้และค่า **nfso nfs_v4_fail_over_timeout** ได้ถูกตั้งค่าไว้จึงถูกแนะนำไว้ว่า อ็อพชันการเมาต์ **retrans** และ **timeo** ถูกตั้งค่าให้มีค่าติดตั้ง **nfso** มากเกินไป ซึ่งจะหลีกเลี่ยง **ETIMEDOUT** ที่ส่งคืนกลับไปยังแอ็พพลิเคชันสำหรับ ข้อมูลที่เรพลิเคท

การเรียงลำดับรายการตำแหน่งระบบไฟล์โดยใช้อ็อพชัน scatter

อ็อพชัน **scatter** ของคำสั่ง **exportfs** อนุญาตให้คุณเปลี่ยนลำดับของรายการตำแหน่งที่ระบุไว้ในระบบไฟล์ ที่ถูกตั้งค่าด้วยอ็อพชัน **refer** หรืออ็อพชัน **replicas** ของคำสั่ง **exportfs**

การใช้อ็อพชันนี้สร้างชุดที่แตกต่างของตำแหน่งเซิร์ฟเวอร์ ดังนั้น รายการที่แตกต่างมีเซิร์ฟเวอร์ที่แตกต่างในลำดับของการกำหนดค่าตามความชอบ ในลำดับต่อมา ไคลเอ็นต์อื่นมีตำแหน่งเซิร์ฟเวอร์รายการที่แตกต่างกัน การเรียงลำดับนี้ช่วยให้สร้างดุลยภาพในการโหลด เนื่องจากเซิร์ฟเวอร์แรกในรายการตำแหน่ง ของไคลเอ็นต์อื่นๆ คือเซิร์ฟเวอร์ที่แตกต่าง และ หากเซิร์ฟเวอร์หยุดทำงาน ความล้มเหลวในการโหลดถูกแจกจ่ายระหว่างเซิร์ฟเวอร์จำนวนมาก เนื่องจากเซิร์ฟเวอร์ ในตำแหน่งถัดไปในตำแหน่งเซิร์ฟเวอร์อื่นๆ อ็อพชัน **scatter** ใช้เฉพาะกับไดเร็กทอรีที่เอ็กซ์พอร์ตสำหรับการเข้าถึงโดยโปรโตคอล NFS เวอร์ชัน 4

อ็อพชัน **scatter** สามารถมีค่าต่อไปนี้:

- **full** – เซิร์ฟเวอร์ทั้งหมดถูกเรียงลำดับใหม่เพื่อจัดรูปแบบชุดของ ตำแหน่งที่เลือก จำนวนทั้งหมดของชุดที่ถูกจำกัดไว้ที่ 12 หรือจำนวนของเซิร์ฟเวอร์ซึ่งเป็นจำนวนที่สูงกว่า

- **partial** – ตำแหน่งแรกสำหรับชุดของเซิร์ฟเวอร์ทั้งหมด ที่สร้างขึ้นถูกแก้ไขไปเป็นเซิร์ฟเวอร์แรกในรายการเซิร์ฟเวอร์ ส่วนที่เหลือของตำแหน่ง ถูกแสดงรายการหากการเรียงลำดับอีกครั้งถูกใช้
- **none** – ไม่มีการเรียงลำดับใหม่ของรายการตำแหน่งระบบไฟล์ที่เสร็จสิ้น นี่คือค่าดีฟอลต์สำหรับ scatter **option** ใช้ค่านี้เพื่อปิดใช้งานการเรียงลำดับใหม่ก่อนหน้านี้ของรายการตำแหน่ง

หมายเหตุ: หากแฟล็ก **noauto** ไม่ได้ระบุไว้ เมื่อคุณกำลังใช้คำสั่ง **exportfs** จากนั้น รายการตำแหน่งสอดแทรก ชื่อโฮสต์หลักเป็นหนึ่งในตำแหน่งเรพลิกา สำหรับข้อมูลเพิ่มเติมเกี่ยวกับแฟล็ก **noauto** โปรดดูคำสั่ง **exportfs** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2*

เมื่อต้องการระบุการอ้างอิงสำหรับไดเร็กทอรี `/common/documents` ที่โฮสต์ s1, s2 และ s3 จากนั้น จัดลำดับอีกครั้งโดยใช้ออ็อพชัน **เต็ม** ให้เพิ่ม บรรทัดต่อไปนี้ลงในไฟล์ `/etc/exports` จากนั้น เอ็กซ์พอร์ตไดเร็กทอรี `/common/documents`:

```
/common/documents -ver=4, refer=/common/documents@s1:/common/document@s2a:/common/
documents@s3,scatter=full
```

เมื่อต้องการระบุเรพพลิกาสำหรับไดเร็กทอรี `/common/documents` ที่โฮสต์ s1, s2, s3 และ s4 และจัดลำดับอีกครั้งเป็นบางส่วน (เซิร์ฟเวอร์ fail-over แรกคือ s1 สำหรับ ชุดทั้งหมด) ให้เพิ่มบรรทัดต่อไปนี้ลงในไฟล์ `/etc/exports` จากนั้น เอ็กซ์พอร์ตไดเร็กทอรี `/common/documents`:

```
/common/documents -vers=4, replicas=/common/documents@s1:/common/documents@s2:/common/
   documents@s3:/common/documents@s4,scatter=partial
```

การแต่งตั้งตัวแทนเซิร์ฟเวอร์-ไคลเอ็นต์ NFS

การแต่งตั้งตัวแทน คือความสามารถของเซิร์ฟเวอร์ในการมอบอำนาจ ความรับผิดชอบบางอย่างให้กับไคลเอ็นต์

เริ่มต้นด้วย AIX 5L เวอร์ชัน 5.3 ที่มีแพ็กเกจการดูแลรักษาที่แนะนำ 5300-03 คุณสามารถใช้การแต่งตั้งตัวแทนได้ เมื่อเซิร์ฟเวอร์ให้สิทธิในการแต่งตั้งตัวแทนสำหรับไฟล์ไปยังไคลเอ็นต์แล้ว ไคลเอ็นต์จะไม่รับประกันซีแมนทิกส์บางอย่างตามที่แบ่งใช้ไฟล์นั้น กับไคลเอ็นต์อื่น เมื่อไฟล์ถูกเปิด เซิร์ฟเวอร์สามารถจัดเตรียมการแต่งตั้งตัวแทนการอ่าน ไฟล์ให้กับไคลเอ็นต์ หากไคลเอ็นต์ถูกให้สิทธิในการแต่งตั้งตัวแทนการอ่านแล้ว จึงมั่นใจได้ว่า ไม่มีไคลเอ็นต์อื่นที่มีความสามารถในการเขียนลงไฟล์ สำหรับช่วงเวลาของการแต่งตั้งตัวแทน หากไคลเอ็นต์ได้รับสิทธิในการแต่งตั้งตัวแทนการเขียน ไคลเอ็นต์จะมั่นใจได้ว่าไม่มีไคลเอ็นต์อื่นที่มีสิทธิในการอ่านหรือเขียน ลงในไฟล์ เซิร์ฟเวอร์ AIX จะให้สิทธิในการแต่งตั้งตัวแทนการอ่านเท่านั้น เซิร์ฟเวอร์ AIX สนับสนุนการแต่งตั้งตัวแทนด้วยเคอร์เนล AIX แบบ 64 บิตเท่านั้น ไคลเอ็นต์ AIX สนับสนุนทั้งการแต่งตั้งตัวแทนการอ่านและเขียน

หากเซิร์ฟเวอร์ให้สิทธิในการแต่งตั้งตัวแทนการอ่านให้กับไคลเอ็นต์แล้ว ไคลเอ็นต์ต้องจัดเตรียมแอดเดรส callback ให้กับเซิร์ฟเวอร์ เมื่อการแต่งตั้งตัวแทนถูกเรียกอีกครั้ง เซิร์ฟเวอร์จะส่งคำร้องขอการเรียกกลับไปยังแอดเดรสนี้ ตามค่าดีฟอลต์ไคลเอ็นต์จะบ่งชี้ IP แอดเดรสที่ถูกใช้สำหรับการสื่อสารตามปกติกับ เซิร์ฟเวอร์ สำหรับไคลเอ็นต์ที่มีเน็ตเวิร์กอินเตอร์เฟสจำนวนมาก แอดเดรสที่ระบุไว้สามารถระบุอยู่ในไฟล์ `/etc/nfs/nfs4_callback.conf` ได้ รูปแบบของรายการในไฟล์นี้คือ:

server-host client-ip-address

โดยที่ *server-host* คือชื่อหรือแอดเดรสของเซิร์ฟเวอร์ NFSv4 และ *client-ip-address* คือไคลเอ็นต์แอดเดรส ที่ต้องถูกใช้เมื่อจัดเตรียมข้อมูลการ callback เซิร์ฟเวอร์ หากชื่อ *server-host* คือ IPv4 แอดเดรส 0.0.0.0 หรือ IPv6 แอดเดรส 0::0 แล้ว *client-ip-address* ระบุไว้จะถูกใช้สำหรับเซิร์ฟเวอร์ทั้งหมดที่ไม่ได้แสดงอยู่ในไฟล์ หากไฟล์นี้ไม่มีอยู่ หรือหากไม่พบรายการสำหรับเซิร์ฟเวอร์ (หรือรายการดีฟอลต์) ไคลเอ็นต์จะเลือกแอดเดรสที่อ้างอิงตามการเชื่อมต่อที่มีอยู่ในเซิร์ฟเวอร์

การแต่งตั้งตัวแทนสามารถเรียกอีกครั้งได้ด้วยเซิร์ฟเวอร์ หากไคลเอ็นต์อื่นร้องขอ การเข้าถึงไฟล์ในวิธีที่การเข้าถึงขัดแย้งกับการแต่งตั้งตัวแทนที่ให้สิทธิ เซิร์ฟเวอร์จะสามารถแจ้งเตือนไคลเอ็นต์เริ่มต้นและเรียกการแต่งตั้งตัวแทนอีกครั้ง ซึ่งจำเป็นต้องมีพาธ callback ระหว่างเซิร์ฟเวอร์และไคลเอ็นต์ หากพาธ callback นี้ไม่มีอยู่ การแต่งตั้งตัวแทนไม่สามารถได้รับสิทธิ หากการแต่งตั้งตัวแทนไฟล์ ถูกให้สิทธิ สิทธิในการเข้าถึงจากไคลเอ็นต์ NFSv4 ไคลเอ็นต์ NFS เวอร์ชัน 2 และ 3 และการเข้าถึงแบบโลคัลไปยังไฟล์ที่ไฟล์เซิร์ฟเวอร์ อาจเป็นสาเหตุทำให้การแต่งตั้งตัวแทนถูกเรียกอีกครั้ง หาก GPFS คือ NFSv4 ที่ถูกเอ็กซ์พอร์ต การเข้าถึงที่โหนด GPFS ในเน็ตเวิร์กอาจเป็นสาเหตุทำให้การแต่งตั้งตัวแทนต้องถูกเรียกอีกครั้ง

สิ่งจำเป็นของการแต่งตั้งตัวแทนคือ สิ่งที่อนุญาตให้ไคลเอ็นต์ใช้การดำเนินการเซอร์วิสแบบโลคัล เช่น OPEN, CLOSE, LOCK, LOCKU, READ และ WRITE โดยไม่โต้ตอบกับเซิร์ฟเวอร์ในทันที

การแต่งตั้งตัวแทนเซิร์ฟเวอร์ถูกเปิดใช้งานตามค่าดีฟอลต์ การแต่งตังตัวแทนเซิร์ฟเวอร์สามารถปิดใช้งานด้วย คำสั่ง `nfso -o server_delegation=0` ผู้ดูแลระบบสามารถใช้ อ็อพชัน **exportfs deleg=yes | no** เพื่อปิดใช้งานหรือเปิดใช้งานการให้สิทธิในการแต่งตั้งตัวแทนบนพื้นฐานของระบบไฟล์ ที่จะเขียนทับค่าติดตั้ง **nfso**

การแต่งตั้งตัวแทนไคลเอ็นต์สามารถปิดใช้งานได้ด้วยคำสั่ง `nfso -o client_delegation=0` การแต่งตั้งตัวแทนไคลเอ็นต์ต้องถูกตั้งค่าก่อนที่การเมาต์ใดๆ เข้าแทนที่บนไคลเอ็นต์

หากผู้ดูแลระบบกำลังเอ็กซ์พอร์ตระบบไฟล์ที่ไคลเอ็นต์จำนวนมากจะเขียนลงในไฟล์ทั่วไปจำนวนมาก ผู้ดูแลระบบอาจต้องการปิดใช้งานการแต่งตั้งตัวแทนสำหรับระบบไฟล์ นั้น

หากไคลเอ็นต์ไม่สามารถติดต่อได้ (ตัวอย่างเช่น หากเน็ตเวิร์กหรือไคลเอ็นต์ พบกับปัญหาเรื่องสัญญาณขาดหาย) ไคลเอ็นต์อื่นอาจหน่วงเวลาในการเข้าถึงข้อมูลได้

การสร้างโฮสต์หลักทั่วไปสำหรับพาธของ Kerberos-protected callback

คุณสามารถตั้งพาธการ callback สำหรับ IBM Network Authentication Service (Kerberos)

ไคลเอ็นต์ที่ได้รับการมอบหมายต้องเป็นไคลเอ็นต์แบบเต็มที่มีโฮสต์หลักของมันเอง อย่างไรก็ตาม คุณสามารถสร้างโฮสต์หลักทั่วไปสำหรับไคลเอ็นต์ทั้งหมดเพื่อใช้สำหรับ callbacks

เพื่อสร้างโฮสต์หลักทั่วไปสำหรับไคลเอ็นต์ทั้งหมดเพื่อใช้สำหรับ callbacks ทำขั้นตอนเหล่านี้ :

1. เพื่อสร้างเซอร์วิสหลัก (ตัวอย่างเช่น `nfs/client`) ใช้วิธีเดียวกับที่ใช้เพื่อสร้างโฮสต์หลัก อ้างถึง การสร้าง Kerberos หลัก ใน *การรักษาความปลอดภัย*
2. สร้าง entry ของ keytab สำหรับเซอร์วิสหลักนั้น ตัวอย่างเช่น เพื่อสร้าง keytab ชื่อ slapd_krb5.keytab ทำต่อไปนี้ :

```
kadmin.local: ktadd -k /etc/security/slapd_krb5.keytab ldap/plankton.austin.ibm.com
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type Triple DES cbc mode with HMAC/sha1 added to keytab
WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type ArcFour with HMAC/md5 added to keytab WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type AES-256 CTS mode with 96-bit SHA-1 HMAC added to keytab
WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type DES cbc mode with RSA-MD5 added to keytab WRFILE:/etc/security/slapd_krb5.keytab.
kadmin.local:
```

3. กระจาย keytab นี้ไปยังไคลเอ็นต์ทั้งหมดที่จะใช้มัน

4. ตั้งค่าไคลเอ็นต์ด้วยคำสั่ง **nfshostkey**

กระบวนการนี้จะเหมือนกับกระบวนการสำหรับตั้งค่าเซิร์ฟเวอร์เพื่อใช้กับ Kerberos แต่หลักทั่วไปไม่สามารถใช้สำหรับเซิร์ฟเวอร์ แต่ละเซิร์ฟเวอร์ต้องมีหลักของรูปแบบ `nfs/`*`hostname`* ของมันเอง

ระบบไฟล์เครือข่ายระยะสั้น STNFS

ระบบไฟล์ Short Term Network File Systems (STNFS) คือระบบไฟล์ที่สำรองข้อมูลโดย Network File System (NFS) และระบบไฟล์ STNFS อนุญาตให้ทำการแก้ไขแบบโลคัลกับไฟล์ การแก้ไขไม่ได้บันทึกไว้บน เซิร์ฟเวอร์

หมายเหตุ:

1 หลายไคลเอ็นต์ STNFS สามารถแบ่งใช้อิมเมจระบบไฟล์เดียวกันจาก เซิร์ฟเวอร์ แต่โมดิฟิเคชันใดๆ เห็นได้โดยไคลเอ็นต์ที่ทำโมดิฟิเคชัน เท่านั้น

2 โมดิฟิเคชันใดๆ ที่ทำโดยไคลเอ็นต์จะสูญหายเมื่อระบบไฟล์ ถูกเลิกเมาท์ หรือเมื่อไคลเอ็นต์รีบู๊ต

3 การดำเนินการเขียนจาก STNFS ล้มเหลวเมื่อหน่วยความจำ ระบบอยู่ต่ำกว่าขนาดจำกัดที่กำหนดไว้ล่วงหน้า ขีดจำกัดนี้เป็น ค่าภายในสำหรับ STNFS และไม่สามารถกำหนดค่าจากภายนอก

การเมาท์ระบบไฟล์ระยะสั้น NFS

คำสั่ง **mount** ถูกใช้เพื่อเมาท์ระบบไฟล์ NFS ใน แบบระยะสั้น ตัวอย่างเช่น พิมพ์คำสั่งต่อไปนี้ :

```
mount -v stnfs -o options server:/remote-path /local-path
```

อ็อพชันที่มีให้ใช้ คือ :

vers=3 ใช้ NFS เวอร์ชัน 3 เพื่อสื่อสารกับเซิร์ฟเวอร์

vers=4 ใช้ NFS เวอร์ชัน 4 เพื่อสื่อสารกับเซิร์ฟเวอร์

rsize=size
ตั้งค่าไบต์ของขนาดการอ่าน

proto=udp
ใช้ UDP เพื่อสื่อสารกับ NFS เซิร์ฟเวอร์

proto=tcp
ใช้ TCP เพื่อสื่อสารกับ NFS เซิร์ฟเวอร์

ฮาร์ด ใช้การฮาร์ดเมาต์ NFS

soft ใช้การซอฟเมาต์ NFS

sec ใช้ประเภทความปลอดภัยที่ระบุ

ดีฟอลต์อ็อพชัน คือ:

vers=3

rsize=32768

proto=tcp

ฮาร์ด

sec=sys

รายการตรวจสอบสำหรับการตั้งค่า NFS

หลังจากที่ติดตั้งซอฟต์แวร์ NFS บนระบบของคุณ คุณพร้อมที่จะตั้งค่า NFS ให้ทำตามขั้นตอนเหล่านี้เพื่อตั้งค่า NFS

CryptoLite ในไลบรารีเคอร์เนล C (CLiC) ต้องถูกติดตั้งไว้ ก่อนการตั้งค่า NFS เพื่อใช้ชนิดของความปลอดภัยต่อไปนี้:

- krb5
- krb5i
- krb5p

แต่ละขั้นตอนถูกกล่าวถึงในรายละเอียดเพิ่มเติมด้านล่างนี้

1. พิจารณาว่า ระบบที่อยู่ในเน็ตเวิร์กคือเซิร์ฟเวอร์ และคือไคลเอ็นต์ (ระบบที่สามารถตั้งค่าเป็นได้ทั้งเซิร์ฟเวอร์และไคลเอ็นต์)
2. พิจารณาเวอร์ชันของ NFS ที่คุณกำลังใช้
3. ตัดสินใจว่า คุณกำลังใช้ความปลอดภัย RPCSEC-GSS หรือไม่ หากใช่ โปรดอ้างถึงข้อควรพิจารณาใน "การตั้งค่าเน็ตเวิร์กสำหรับ RPCSEC-GSS" ในหน้า 572
4. สำหรับแต่ละระบบ (ไม่ว่าจะเป็นไคลเอ็นต์หรือเซิร์ฟเวอร์) ให้ทำตามคำสั่งใน "เริ่มต้น NFS daemons ที่การเริ่มต้นทำงานกับระบบ"
5. สำหรับแต่ละเซิร์ฟเวอร์ NFS ให้ทำตามคำสั่งใน "การตั้งค่าเซิร์ฟเวอร์ NFS"
6. สำหรับแต่ละไคลเอ็นต์ NFS ให้ทำตามคำสั่งใน "การตั้งค่าไคลเอ็นต์ NFS" ในหน้า 570
7. หากคุณต้องการให้คอมพิวเตอร์ส่วนบุคคลอยู่บนเน็ตเวิร์กของคุณเพื่อให้เข้าถึงเซิร์ฟเวอร์ NFS ของคุณ (ใกล้เคียงกับความสามารถในการเมาต์ระบบไฟล์) ให้ตั้งค่า PC-NFS โดยทำตามคำสั่งต่อไปนี้ใน "PC-NFS" ในหน้า 585
8. หากใช้ NFS เวอร์ชัน 4 ตามที่วางแผนไว้ โปรดอ้างถึงข้อควรพิจารณาใน "ส่วนสนับสนุน NFS เวอร์ชัน 4" ในหน้า 556

เริ่มต้น NFS daemons ที่การเริ่มต้นทำงานกับระบบ

ตามค่าดีฟอลต์แล้ว NFS daemon ไม่ได้เริ่มต้นในระหว่างการติดตั้ง

เมื่อติดตั้งไว้แล้ว ไฟล์ทั้งหมดถูกวางอยู่บนระบบ แต่ขั้นตอนการเรียกใช้ NFS จะไม่ถูกใช้ คุณสามารถเริ่มต้น NFS daemon ณ เวลาเริ่มต้น ระบบผ่าน:

- วิธีลัด SMIT `smit mknfs`
- คำสั่ง **mknfs**

เมธอดเหล่านี้ทั้งหมดวางรายการอยู่ในไฟล์ `inittab` ดังนั้น สคริปต์ `/etc/rc.nfs` ถูกรันในแต่ละครั้งที่ระบบ รีสตาร์ท ในทางกลับ สคริปต์นี้สตาร์ท NFS daemon ทั้งหมดที่จำเป็นต้องมีสำหรับระบบ เฉพาะ

การตั้งค่าเซิร์ฟเวอร์ NFS

ใช้โพรซีเดอร์นี้เพื่อตั้งค่าเซิร์ฟเวอร์ NFS

หากต้องการตั้งค่าเซิร์ฟเวอร์ NFS:

1. สร้างไฟล์ /etc/exports โปรดดู "ไฟล์ /etc/exports" ในหน้า 550
2. หากคุณกำลังใช้ Kerberos ให้ตั้งค่าเซิร์ฟเวอร์ NFS เป็นไคลเอ็นต์ เวอร์ชัน 4 โปรดดู "การตั้งค่าเน็ตเวิร์กสำหรับ RPCSEC-GSS" ในหน้า 572
3. หากคุณกำลังใช้ NFS เวอร์ชัน 4 ให้สร้างโดเมน NFS เวอร์ชัน 4 ให้ใช้คำสั่ง **chnfsdom** โปรดดูคำอธิบายคำสั่ง **chnfsdom** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1* สำหรับรายละเอียด

   เมื่อเริ่มต้น คุณสามารถระบุโดเมนของอินเตอร์เน็ตของเซิร์ฟเวอร์ในไฟล์ อย่างไรก็ตาม อาจเป็นไปได้ที่จะนิยามโดเมน NFS เวอร์ชัน 4 ที่แตกต่างจาก โดเมนของอินเตอร์เน็ตของเซิร์ฟเวอร์ สำหรับความชัดเจนเกี่ยวกับสิ่งนี้ โปรดดูเอกสารคู่มือ สำหรับ NFS registry daemon **nfsrgyd** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*
4. หากคุณกำลังใช้ NFS เวอร์ชัน 4 พร้อมกับ Kerberos คุณอาจจำเป็นต้องสร้าง ไฟล์ /etc/nfs/realm.map โปรดดู "ไฟล์ /etc/nfs/realm.map" ในหน้า 552
5. หากคุณต้องการใช้การพิสูจน์ตัวตน Kerberos บนเซิร์ฟเวอร์ คุณต้องเปิดใช้งานความปลอดภัยที่พัฒนาแล้วบนเซิร์ฟเวอร์ คุณสามารถเปิดใช้งานความปลอดภัยที่พัฒนาแล้วโดยผ่านการใช้ SMIT หรือโดยใช้คำสั่ง **chnfs -S -B** สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ **chnfs** โปรดอ้างอิงคำอธิบายคำสั่ง **chnfs** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1*

การตั้งค่าไคลเอ็นต์ NFS

ใช้โพรซีเดอร์นี้เพื่อตั้งค่าไคลเอ็นต์ NFS

1. เริ่มต้น NFS โดยใช้คำสั่งใน "การสตาร์ท NFS daemons" ในหน้า 555
2. สร้างจุดเมาต์บนโลคัลโดยใช้คำสั่ง **mkdir** สำหรับ NFS ที่ต้องการเมาต์ให้สำเร็จ ไดเร็กทอรีที่ทำหน้าที่คล้ายกับจุดเมาต์ (หรือตัวยึดตำแหน่ง) ของการเมาต์ NFS ต้องถูกแสดง ไดเร็กทอรีนี้ ควรว่างเปล่า จุดเมาต์นี้สามารถสร้างไดเร็กทอรีอื่นใดๆ ที่เหมือนกันได้ และไม่มีแอ็ตทริบิวต์พิเศษที่จำเป็น

   หมายเหตุ: ด้วยหนึ่งข้อยกเว้น จุดเมาต์สำหรับการเมาต์ NFS ทั้งหมดต้องมีอยู่บนระบบของคุณก่อนที่จะเมาต์ ระบบไฟล์ หากใช้ **automount** daemon จึงไม่จำเป็นต้องสร้างจุดเมาต์ โปรดดูคำอธิบาย **automount** daemon ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1* สำหรับรายละเอียด
3. หากกำลังใช้ Kerberos ให้ทำตามขั้นตอนเหล่านี้:
   a. ตั้งค่าไคลเอ็นต์ NFS ลงในขอบเขต Kerberos ซึ่งถูกทำ พร้อมกับคำสั่ง **config.krb5** โปรดอ้างถึง *IBM Network Authentication Service Administrator's and User's Guide* สำหรับรายละเอียดคอนฟิกูเรชัน
   b. สร้างหลักการของ Kerberos สำหรับผู้ใช้ทั้งหมดบนไคลเอ็นต์ ผู้ที่จะเข้าถึงไฟล์ผ่านการเมาต์ Kerberos ซึ่งจะทำด้วยคำสั่ง **kadmin** โปรดอ้างถึง *Network Authentication Service Administrator's and User's Guide* สำหรับคำอธิบายเกี่ยวกับวิธีการสร้างหลักการ Kerberos
   c. การสร้างหลักการ Kerberos สำหรับเครื่องไคลเอ็นต์เอง เป็นตัวเลือก ไคลเอ็นต์ที่ไม่มีหลักการจะรู้จักว่าเป็น *slim client* และไคลเอ็นต์ที่มีหลักการถูกอ้างถึงเป็น *full client* Slim client ใช้ความปลอดภัย NFS RPC ที่มีจุดอ่อนมากกว่าเมื่อดำเนินการกับ NFS เวอร์ชัน 4 การดำเนินการจัดการคอนเท็กซ์แบบไคลเอ็นต์ต่อเซิร์ฟเวอร์ที่ใช้สำหรับการจัดการกับสถานะ ไคลเอ็นต์แบบเต็ม ซึ่งขึ้นอยู่กับคอนฟิกูเรชัน สามารถใช้ความปลอดภัย Kerberos-based RPC ที่แข็งแรงกว่า คอนฟิกูเรชัน Slim client จำเป็นต้องมีค่าใช้จ่ายในการดูแลน้อยกว่า และอาจเพียงพอสำหรับสภาพแวดล้อมจำนวนมาก การนำไปใช้งานจำเป็นต้องมีระดับที่สูงที่สุด ของความปลอดภัยที่อาจเลือกเพื่อรันคอนฟิกูเรชันไคลเอ็นต์แบบเต็ม

4. หากคุณกำลังใช้ NFS เวอร์ชัน 4 คุณต้องสร้างโดเมน NFS เวอร์ชัน 4 โดยใช้คำสั่ง **chnfsdom** เมื่อเริ่มต้น คุณสามารถระบุโดเมนของอินเตอร์เน็ตของไคลเอ็นต์ในไฟล์ อย่างไรก็ตาม เป็นไปได้ที่จะนิยามโดเมน NFS เวอร์ชัน 4 ซึ่งแตกต่างจากโดเมนของอินเตอร์เน็ต ของไคลเอ็นต์ สำหรับความชัดเจนเกี่ยวกับสิ่งนี้ โปรดดูเอกสารคู่มือสำหรับ NFS registry daemon **nfsrgyd**
5. หากคุณต้องการใช้การพิสูจน์ตัวตนของ Kerberos เกี่ยวกับไคลเอ็นต์ คุณต้องเปิดใช้งานความปลอดภัยที่พัฒนาแล้วบนไคลเอ็นต์ คุณสามารถเปิดใช้งานความปลอดภัยที่พัฒนาแล้วโดยใช้ SMIT หรือโดยใช้คำสั่ง **chnfs -S -B** สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ **chnfs** โปรดอ้างหน้าที่อ้างอิงถึงคำสั่ง **chnfs**
6. สร้างและเมาต์จุดเมาต์ที่กำหนดไว้ก่อนโดยทำตามคำสั่งใน “การสร้างเมาต์ NFS ที่กำหนดไว้ล่วงหน้า” ในหน้า 580

การแม็พลักษณะเฉพาะ

ลักษณะเฉพาะการแม็พจัดเตรียมเมธอดสำหรับเซิร์ฟเวอร์ NFS และไคลเอ็นต์เพื่อแปลผู้ใช้ภายนอกและกลุ่มไปยังผู้ใช้และกลุ่มโลคัล

AIX ใช้เทคโนโลยี EIM ที่อ้างอิงบน LDAP เพื่อดำเนินการระบุการแม็พ ลักษณะเฉพาะการแม็พ NFS ทั้งหมดถูกเก็บอยู่บนเซิร์ฟเวอร์ LDAP

In order to set up an EIM client, the `bos.eim.rte` and `ldap.client` filesets must be installed. เซิร์ฟเวอร์ EIM ยังต้องการชุดไฟล์ `ldap.server` After the appropriate filesets are installed, the `/usr/sbin/chnfsim` is used to configure EIM. อ็อพชันต่ำสุดของการติดตั้งมีดังต่อไปนี้:

```
/usr/sbin/chnfsim -c -a -t [type] -h [EIM server] -e [LDAP/EIM domain] -f [LDAP suffix] -w [administrator password]
```

ซึ่ง ตั้งค่าทั้งไคลเอ็นต์และเซิร์ฟเวอร์ EIM เพื่อใช้เซิร์ฟเวอร์ EIM ที่ระบุสำหรับการแม็พ ลักษณะเฉพาะ หากชื่อโฮสต์ระบุอยู่ในคำสั่งคือชื่อโฮสต์โลคัล จากนั้น เซิร์ฟเวอร์ LDAP จะไม่ถูกตั้งค่าไว้

หลังจากที่ขั้นตอนของการคอนฟิกูเรชันเสร็จสิ้นแล้ว ผู้ดูแลระบบ EIM สามารถบรรจุเซิร์ฟเวอร์ LDAP พร้อมกับข้อมูลการแม็พลักษณะเฉพาะของ NFS ผู้ใช้หรือกลุ่มเดี่ยว เช่น John Doe รู้จักลักษณะเฉพาะของการแม็พนี้ เจ้าของ NFS ของผู้ใช้นั้น *johndoe@austin.ibm.com* รู้จักลักษณะของการแม็พ หากต้องการอินพุตเซิร์ฟเวอร์ LDAP พร้อมกับข้อมูลนี้ คำสั่งต่อไปนี้ควรรัน:

```
/usr/sbin/chnfsim -a -u -i "John Doe" -n johndoe -d austin.ibm.com
```

ลักษณะเฉพาะของการแม็พคือชื่อเชิงอธิบายของผู้ใช้หรือกลุ่ม และลักษณะเฉพาะของการแม็พคือ *name@domain* ซึ่งเป็นเจ้าของ NFS ขอบเขตของการแม็พโดเมนยังถูกเก็บอยู่ในเซิร์ฟเวอร์ LDAP หากต้องการอินพุตว่า Kerberos realm `kerb.austin.ibm.com` แม็พกับโดเมน NFS `austin.ibm.com` คำสั่งต่อไปนี้ควรถูกรัน:

```
/usr/sbin/chnfsim -a -r kerb.austin.ibm.com -d austin.ibm.com
```

หาก ต้องการตั้งค่า NFS เพื่อใช้ข้อมูลการแม็พใน EIM NFS registry daemon จำเป็นต้องรีสตาร์ท การลงทะเบียน NFS registry daemon ตรวจสอบสภาพพร้อมใช้งานของ เซิร์ฟวเอร์ EIM ตามการเริ่มต้นทำงานและหากพบ ฟังก์ชันการแม็พทั้งหมด จะไปยัง EIM และการแม็พโลคัลทั้งหมดจะไม่ถูกใช้อีกต่อไป

สำหรับข้อมูลเกี่ยวกับ EIM โปรดดู Enterprise identity mapping ใน *การรักษาความปลอดภัย*

การเอ็กซ์พอร์ตระบบไฟล์ NFS

คุณสามารถเอ็กซ์พอร์ตระบบไฟล์ NFS โดยใช้โพรซีเดอร์ต่อไปนี้

- หากต้องการเอ็กซ์พอร์ตระบบไฟล์ NFS โดยใช้ SMIT:
  1. ตรวจสอบว่า NFS กำลังรันโดยพิมพ์คำสั่ง `lssrc -g nfs` เอาต์พุตควรบ่งชี้ว่า **nfsd** และ **rpc.mountd** daemon แอ็คทีฟอยู่ หากไม่มีอยู่ ให้สตาร์ท NFS โดยใช้คำสั่งใน “การสตาร์ท NFS daemons” ในหน้า 555
  2. ที่บรรทัดรับคำสั่ง ให้พิมพ์คำสั่งต่อไปนี้และกด Enter:

     ```
     smit mknfsexp
     ```
  3. ระบุค่าที่เหมาะสมใน PATHNAME ของไดเร็กทอรีที่เอ็กซ์พอร์ต MODE ที่ต้องการเอ็กซ์พอร์ตไดเร็กทอรี และไดเร็กทอรี EXPORT เดี๋ยวนี้ ระบบจะรีสตาร์ทหรือทั้งสองฟิลด์
  4. ระบุคุณสมบัติเผื่อเลือกที่คุณต้องการ หรือยอมรับค่าดีฟอลต์โดยออกจากฟิลด์ที่เหลืออยู่ตามที่เป็น
  5. เมื่อคุณเสร็จสิ้นการเปลี่ยนแปลงแล้ว SMIT จะอัพเดตไฟล์ `/etc/exports` หากไฟล์ `/etc/exports` ไม่มีอยู่ ระบบจะสร้างขึ้นใหม่
  6. ทำซ้ำขั้นตอนที่ 3 ถึง 5 สำหรับไดเร็กทอรีทั้งหมดที่คุณต้องการเอ็กซ์พอร์ต
- หากต้องเอ็กซ์พอร์ตระบบไฟล์ NFS โดยใช้เท็กซ์เอดิเตอร์:
  1. เปิดไฟล์ `/etc/exports` กับเท็กซ์เอดิเตอร์ ที่คุณคุ้นเคย
  2. สร้างรายการสำหรับไดเร็กทอรีที่ต้องการเอ็กซ์พอร์ตโดยใช้ชื่อพาธเต็มของ ไดเร็กทอรี แสดงไดเร็กทอรีแต่ละตัวที่ต้องการเอ็กซ์พอร์ตซึ่งเริ่มต้นใน ระยะขอบซ้าย ไม่มีไดเร็กทอรีที่ควรสอดแทรกไดเร็กทอรีอื่นที่พร้อม เอ็กซ์พอร์ต โปรดดูไฟล์ `/etc/exports` ใน *การอ้างอิงไฟล์* สำหรับคำอธิบายของไวยากรณ์แบบเต็มสำหรับรายการในไฟล์ `/etc/exports`
  3. บันทึกและปิดไฟล์ `/etc/exports`
  4. หาก NFS กำลังรันอยู่ ให้พิมพ์คำสั่งต่อไปนี้และกด Enter:

     ```
     /usr/sbin/exportfs -a
     ```

     อ็อพชัน **–a** แจ้งให้คุณทราบถึงคำสั่ง **exportfs** ที่ต้องการส่งข้อมูลทั้งหมดใน ไฟล์ `/etc/exports` ไปยังเคอร์เนล หาก NFS ไม่ได้รัน ให้สตาร์ท NFS โดยใช้คำสั่งใน “การสตาร์ท NFS daemons” ในหน้า 555
- หากต้องการเอ็กซ์พอร์ตระบบไฟล์ NFS ชั่วคราว (โดยไม่เปลี่ยนไฟล์ `/etc/exports`) ให้พิมพ์คำสั่งต่อไปนี้และกด Enter:

  ```
  exportfs -i /dirname
  ```

  โดยที่ *dirname* คือชื่อระบบไฟล์ที่คุณต้องการเอ็กซ์พอร์ต คำสั่ง **exportfs –i** ระบุว่าไฟล์ `/etc/exports` ไม่สามารถตรวจสอบได้สำหรับไดเร็กทอรีที่ระบุ และอ็อพชันทั้งหมดที่ใช้จาก บรรทัดรับคำสั่งโดยตรง

  AIX NFS เวอร์ชัน 4 อนุญาตให้ผู้ดูแลระบบสร้างและควบคุม namespace ที่เลือกไว้โดยสร้างการแสดงผลโดยเซิร์ฟเวอร์ NFS กับไคลเอ็นต์ ซึ่งทำได้โดยใช้อ็อพชันเอ็กซ์พอร์ต **exname** ส่วนสนับสนุนนี้ยังใช้เพื่อซ่อนรายละเอียด ของ namespace ของระบบไฟล์โลคัลของเซิร์ฟเวอร์จากไคลเอ็นต์ NFS สำหรับรายละเอียดเพิ่มเติม โปรดดูคำอธิบายคำสั่ง **exportfs** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 2* และคำอธิบายไฟล์ `/etc/exports` ใน *การอ้างอิงไฟล์*

การตั้งค่าเน็ตเวิร์กสำหรับ RPCSEC-GSS

เน็ตเวิร์กที่กำลังตั้งค่าอยู่ในสถานการณ์จำลองนี้มีห้าเซิร์ฟเวอร์ และตั้งค่าสำหรับ RPCSEC–GSS

ห้าเซิร์ฟเวอร์บนเน็ตเวิร์กมีดังต่อไปนี้:

- kdc.austin.ibm.com
- alpha.austin.ibm.com
- beta.austin.ibm.com
- gamma.austin.ibm.com
- zeta.austin.ibm.com

ระบบ kdc.austin.ibm.com จะถูกตั้งค่าเป็น เซิร์ฟเวอร์ Key Distribution Center (KDC) และขอบเขตของ Kerberos AUSTIN.IBM.COM จะถูกสร้างขึ้น ซึ่งระบบทั้งหมดยกเว้น kdc.austin.ibm.com และ zeta.austin.ibm.com จะเป็นเซิร์ฟเวอร์ NFS ที่นำเสนอระบบไฟล์ที่เอ็กซ์พอร์ตด้วย RPCSEC-GSS

ระบบ alpha.austin.ibm.com และ beta.austin.ibm.com มีลิงก์เพิ่มเติมระหว่างระบบข้ามผ่านลิงก์นั้น ซึ่งปรากฏขึ้นกับแต่ละระบบ เป็น fast_alpha.test.austin.com และ fast_beta.test.austin.ibm.com สำหรับเหตุผลนี้ ขั้นตอนของการคอนฟิกูเรชันเพิ่มเติมอาจจำเป็นต้องมี

นอกจากนี้ เน็ตเวิร์กนี้มีผู้ใช้ต่อไปนี้ ซึ่งได้ถูกตั้งค่าไว้ บนระบบอื่นๆ:

- adam
- brian
- charlie
- dave
- eric

หมายเหตุ: การตั้งค่าการติดตามถูกจัดเตรียมไว้ตามตัวอย่าง และอาจไม่เหมาะสมกับ สภาพแวดล้อมทั้งหมด โปรดดูคำแนะนำในการดูแลระบบและผู้ใช้ สำหรับ Network Authentication Service ก่อนที่จะพยายามตั้งค่าขอบเขต Kerberos ใหม่

หมายเหตุ: Kerberos ต้องการให้เวลาระบบใกล้เคียงกับ เน็ตเวิร์กทั้งหมดอย่างสมเหตุสมผล ก่อนที่จะเริ่มต้นโพรซีเดอร์นี้ คุณควรตั้งค่ากลไกเพื่อซิงโครไนซ์เวลาแบบอัตโนมัติตลอดทั้งเน็ตเวิร์ก เช่น AIX **timed** daemon หรือการตั้งค่า NTP

1. ตั้งค่าเซิร์ฟเวอร์ KDC

   หมายเหตุ: เซิร์ฟเวอร์ KDC ไม่ควรใช้สำหรับวัตถุประสงค์อื่นใด หาก KDC ถูกประนีประนอม Kerberos ที่สำคัญจะถูกประนีประนอมด้วยเช่นกัน

   ในสถานการณ์จำลองนี้ kdc.austin.ibm.com จะถูกตั้งค่าเป็นเซิร์ฟเวอร์ KDC คอนฟิกูเรชันต่อไปนี้ใช้สำหรับ **des3** หาก **des** จะถูกนำเสนอสำหรับเหตุผลสำหรับผลการทำงาน ให้เพิ่มอาร์กิวเมนต์ -e des-cbc-crc:normal ให้กับการเรียก addprinc และ ktadd สำหรับ **kadmin** ด้านล่าง

   หากตั้งค่าเน็ตเวิร์กของคุณกับการเข้ารหัส **aes** ให้เพิ่มอาร์กิวเมนต์ -e aes256-cts:normal ให้กับการเรียก addprinc และการเรียก ktadd สำหรับคำสั่ง **kadmin**

   a. ติดตั้งชุดไฟล์ krb5.server.rte บน kdc.austin.ibm.com
   b. ตั้งค่าเซิร์ฟเวอร์ KDC ในสถานการณ์นี้ คำสั่งต่อไปนี้ ถูกนำมาใช้:

      ```
      config.krb5 -S -d austin.ibm.com -r AUSTIN.IBM.COM
      ```

      หลังจากที่รัน คำสั่งนี้ ระบบจำถามรหัสผ่าน Master Database และ รหัสผ่านสำหรับหลักการดูแลระบบ

c. สร้างหลักการสำหรับแต่ละผู้ใช้และโฮสต์ที่รันด้วยคำสั่ง `/usr/krb5/sbin/kadmin.local` บนเซิร์ฟเวอร์ KDC ตัวอย่างนี้สร้างหลักการ Kerberos ที่ตรงกับชื่อผู้ใช้ UNIX ของผู้ใช้ที่เชื่อมโยง ชื่อหลักการจะถูกแม็พกับชื่อผู้ใช้โดย NFS เพื่อกำหนดหนังสือรับรอง UNIX ที่เชื่อมโยงกับหลักการ สำหรับคำอธิบายเกี่ยวกับวิธีการใช้การแม็พทั่วไประหว่างหลักการและชื่อผู้ใช้ให้ป้อน "การแม็พลักษณะเฉพาะ" ในหน้า 571 สำหรับเน็ตเวิร์กนี้ เราได้สร้างหลักการต่อไปนี้:

   - `adam`
   - `brian`
   - `charlie`
   - `dave`
   - `eric`
   - `nfs/alpha.austin.ibm.com`
   - `nfs/beta.austin.ibm.com`
   - `nfs/gamma.austin.ibm.com`

   หมายเหตุ: ชื่อหลักการสำหรับผู้ใช้ที่เลือกต้องตรงกับชื่อผู้ใช้ที่สอดคล้องกัน ในการลงทะเบียนผู้ใช้ที่ตั้งค่าไว้ของระบบ (`/etc/passwd`, **LDAP**, **NIS** และอื่นๆ) NFS ใช้ชื่อหลักการเป็นชื่อผู้ใช้เพื่อขอรับผู้ใช้ และ ID กลุ่มบนบนระบบโลคัล หากชื่อไม่ตรงกัน การเข้าถึงจะถูกใช้เป็นการเข้าถึงแบบไม่ระบุชื่อ

KDC ถูกตั้งค่าไว้แล้วในเวลานี้

2. ไคลเอ็นต์ NFS และเซิร์ฟเวอร์แต่ละเครื่องจะถูกตั้งค่าไว้เป็นไคลเอ็นต์ Kerberos โดยใช้คำสั่ง **config.krb5** วิธีการที่ใช้จะขึ้นอยู่กับวิธีการตั้งค่า KDC ในสถานการณ์นี้ เราจะรันคำสั่ง ต่อไปนี้บนระบบ NFS แต่ละระบบ:

   ```
   config.krb5 -C -d austin.ibm.com -r AUSTIN.IBM.COM -c kdc.austin.ibm.com -s kdc.austin.ibm.com
   ```

   อาจเป็นไปได้ที่ **kinit** เป็นหลักการผู้ใช้ใดๆ บนระบบที่ถูกตั้งค่าไว้ใดๆ ตัวอย่างเช่น หากต้องการให้ **kinit** เป็นผู้ใช้ `adam` ให้รันคำสั่งต่อไปนี้:

   ```
   /usr/krb5/bin/kinit adam
   ```

   คุณอาจต้องการ ระบุรหัสผ่าน Kerberos ของ `adam` ซึ่งไม่ใช่ AIX

   ตัวอย่างนี้ใช้ **kinit** เพื่อพิสูจน์ตัวตนผู้ใช้ซึ่งเป็นไปได้ ที่จะตั้งค่า AIX เพื่อใช้การพิสูจน์ตัวตนของ Kerberos ระหว่างการล็อกอินของระบบ สำหรับข้อมูลเพิ่มเติม โปรดดู การพิสูจน์ตัวตนกับ AIX การใช้ Kerberos ใน *การรักษาความปลอดภัย*

3. แต่ละเซิร์ฟเวอร์ NFS จะถูกตั้งค่าด้วยรายการ `keytab` ที่เหมาะสม ในสถานการณ์นี้ เราตั้งค่ารายการ `keytab` สำหรับ `alpha.austin.ibm.com` ดังตัวอย่าง กระบวนการเดียวกัน จะถูกใช้บน `beta.austin.ibm.com` และ `gamma.austin.ibm.com`

   a. จากนั้น `alpha.austin.ibm.com` ให้รัน **kadmin** ดังนั้น ให้รันคำสั่งต่อไปนี้:

      ```
      ktadd nfs/alpha.austin.ibm.com
      ```

      คำสั่งนี้ สร้างไฟล์ `keytab`

   b. จากนั้น ให้ตั้งค่า **gssd** daemon เพื่อใช้ไฟล์ `keytab` ที่คุณเพิ่งสร้างขึ้นด้วยคำสั่ง **nfshostkey** ในสถานการณ์นี้ เราจะรันคำสั่งต่อไปนี้:

      ```
      nfshostkey -p nfs/alpha.austin.ibm.com -f /etc/krb5/krb5.keytab
      ```

   c. ตั้งค่า **gssd** daemon เพื่อเริ่มต้นโดยอัตโนมัติ โดยรันคำสั่งต่อไปนี้:

```
chnfs -S -B
```

ทำซ้ำการติดตั้งนี้สำหรับแต่ละระบบ

4. ที่จุดนี้ เซิร์ฟเวอร์ NFS จะทำงาน แม้ว่า ผู้ใช้ทั้งหมดจะมา ระหว่าง nobody ซึ่งสามารถแนะนำให้ผู้ใช้ทั้งหมด มีอยู่บนเซิร์ฟเวอร์ทั้งหมดที่มี uid และ gid เดียวกัน ผู้ใช้ใดๆ ที่ไม่มีอยู่ จะมีการเข้าถึงไดเร็กทอรีที่เอ็กซ์พอร์ตเฉพาะ nobody เท่านั้น หากต้องการขอรับชื่อผู้ใช้เพื่อแม็พอย่างถูกต้อง คุณต้องตั้งค่าการลงทะเบียน NFS daemon
   a. ตั้งค่าโดเมนโดยใช้คำสั่ง **chnfsdom** ในสถานการณ์จำลองนี้ คำสั่งต่อไปนี้จะรันบนเซิร์ฟเวอร์ NFS ทั้งหมดเพื่อตั้งค่า เพื่อตั้งค่า austin.ibm.com เป็นโดเมน:

      ```
      chnfsdom austin.ibm.com
      ```

   b. ตั้งค่าไฟล์ /etc/nfs/realm.map ไฟล์นี้ควรมีเพียงหนึ่งบรรทัดที่มีชื่อขอบเขตที่ตามด้วยโลคัลโดเมน สำหรับตัวอย่างเน็ตเวิร์กของเรา ไฟล์สองไฟล์เหล่านี้ควรดูคล้ายกับที่แสดงอยู่บน เซิร์ฟเวอร์ NFS ทั้งหมด:

      ```
      realm.map AUSTIN.IBM.COM   austin.ibm.com
      ```

      รายการ ขอบเขตในไฟล์นี้ไม่คำนึงถึงขนาดตัวพิมพ์ ดังนั้น ในเชิงเทคนิค รายการนี้ไม่จำเป็นต้องมี

   c. สำหรับ zeta.austin.ibm.com ซึ่งจะไม่ใช่เซิร์ฟเวอร์ NFS ให้เริ่มต้นทำงาน **gssd** daemon โดยใช้คำสั่ง chnfs -S -B ก่อนที่พยายามการดำเนินการกับไคลเอ็นต์ Kerberos ใดๆ ผู้ใช้ต้องใช้ **kinit** ขอรับหนังสือรับรองที่ถูกต้อง
5. ในสถานการณ์จำลองนี้ มีลิงก์เน็ตเวิร์กแบบด่วนที่ตั้งค่าระหว่าง alpha.austin.ibm.com และ beta.austin.ibm.com ระหว่างลิงก์นี้ beta.austin.ibm.com จะดู alpha.austin.ibm.com เป็น fast_alpha.test.austin.ibm.com และ alpha.autsin.ibm.com จะดู beta.austin.ibm.com เป็น fast_beta.test.austin.ibm.com เนื่องจากทั้ง nfs/fast_alpha.test.austin.ibm.com และ nfs/fast_beta.test.austin.ibm.com ไม่ใช่หลักการที่ถูกต้อง ดังนั้นจึงไม่สามารถใช้ลิงก์นี้สำหรับเมาต์

   หากต้องการแก้ไขนี้ คำสั่ง **nfshostmap** จะถูกใช้ ซึ่งจะแม็พหลักการเพื่อจัดการสถานการณ์นี้
   a. บน alpha.austin.ibm.com เรารันคำสั่ง ต่อไปนี้:

      ```
      nfshostmap -a beta.austin.ibm.com fast_beta.test.austin.ibm.com
      ```

      คำสั่งนี้ แจ้งให้ alpha.austin.ibm.com ทราบว่า หลักการของ fast_beta.test.austin.ibm.com ใช้สำหรับ beta.austin.ibm.com
   b. สำหรับเบต้า เราจะรันคำสั่งต่อไปนี้:

      ```
      nfshostmap -a alpha.austin.ibm.com fast_alpha.test.austin.ibm.com
      ```

   เซิร์ฟเวอร์สามารถมีหลักการเกี่ยวกับโฮสต์จำนวนมาก ซึ่งสมมติขึ้นว่า IP แอดเดรส สำหรับ fast_alpha คือ 10.0.0.1 และ IP แอดเดรส สำหรับ fast_beta คือ 10.0.0.2 ให้ดำเนินการตามขั้นตอนนี้ให้เสร็จสิ้น เพื่อเพิ่มหลักการของโฮสต์จำนวนมาก:
   a. เพิ่มหลักการ nfs/fast_alpha.test.austin.ibm.com และ nfs/fast_beta.test.austin.ibm.com ให้กับไฟล์คีย์แท็บตามความเหมาะสม
   b. รันคำสั่ง **nfshostkey** บนเซิร์ฟเวอร์ alpha ดังต่อไปนี้:

      ```
      nfshostkey -a -p nfs/fast_alpha.test.austin.ibm.com -i 10.0.0.1
      ```

   c. รันคำสั่ง **nfshostkey** บนเซิร์ฟเวอร์ beta ดังต่อไปนี้:

      ```
      nfshostkey -a -p nfs/fast_beta.test.austin.ibm.com -i 10.0.0.2
      ```

การยกเลิกเอ็กซ์พอร์ตระบบไฟล์ NFS

คุณสามารถยกเลิกการเอ็กซ์พอร์ตไดเร็กทอรี NFS โดยใช้โพรซีเดอร์ต่อไปนี้

- หากต้องการยกเลิกการเอ็กซ์พอร์ตไดเร็กทอรี NFS โดยใช้ SMIT:
  1. พิมพ์คำสั่งต่อไปนี้ที่จุดรับคำสั่งและกด Enter:

     ```
     smit rmnfsexp
     ```

  2. ป้อนชื่อพาธที่เหมาะสมใน PATHNAME ของไดเร็กทอรีที่เอ็กซ์พอร์ตที่ต้องลบฟิลด์

     ไดเร็กทอรีจะถูกลบออกจากไฟล์ /etc/exports และยกเลิกการเอ็กซ์พอร์ต

     หากไดเร็กทอรีถูกเอ็กซ์พอร์ตไปยังไคลเอ็นต์โดยใช้ NFS เวอร์ชัน 4 การยกเลิกการเอ็กซ์พอร์ตอาจล้มเหลวเนื่องจากสถานะของไฟล์บนเซิร์ฟเวอร์ สถานะของไฟล์หมายถึง ไฟล์ในไดเร็กทอรีเอ็กซ์พอร์ตที่เปิดโดยไคลเอ็นต์ คุณสามารถใช้การดำเนินการ เพื่อหยุดแอ็พพลิคเชันโดยใช้ข้อมูลนั้น หรือคุณสามารถยกเลิกการเอ็กซ์พอร์ต (**exportfs -F**) ข้อมูล ซึ่งอาจส่งผลทำให้เกิดความล้มเหลวสำหรับแอ็พพลิเคชันที่ใช้ข้อมูล
- หากต้องการยกเลิกการเอ็กซ์พอร์ตไดเร็กทอรี NFS โดยใช้เท็กซ์เอดิเตอร์:
  1. เปิดไฟล์ /etc/exports กับเท็กซ์เอดิเตอร์ ที่คุณคุ้นเคย
  2. ค้นหารายการสำหรับไดเร็กทอรีที่คุณต้องการยกเลิกการเอ็กซ์พอร์ต และลบบรรทัดนั้น
  3. บันทึกและปิดไฟล์ /etc/exports
  4. หาก NFS กำลังรันอยู่ ให้ป้อน:

     ```
     exportfs -u dirname
     ```

     โดยที่ *dirname* คือชื่อพาธเต็มของไดเร็กทอรีที่คุณเพิ่งลบออกจากไฟล์ /etc/exports หาก การยกเลิกการเอ็กซ์พอร์ตล้มเหลวเนื่องจากการเข้าถึงโดยไคลเอ็นต์ NFS V4 คุณสามารถเพิ่มอ็อพชัน -F เพื่อบังคับให้ไดเร็กทอรีถูกยกเลิกการเอ็กซ์พอร์ต

การเปลี่ยนระบบไฟล์ที่เอ็กซ์พอร์ต

เปลี่ยนระบบไฟล์ NFS ที่เอ็กซ์พอร์ตโดยใช้โพรซีเดอร์ ต่อไปนี้

- หากต้องการเปลี่ยนแปลงระบบไฟล์ NFS ที่เอ็กซ์พอร์ตโดยใช้ SMIT:
  1. หากต้องการยกเลิกการเอ็กซ์พอร์ตระบบไฟล์ ให้พิมพ์:

     ```
     exportfs -u /dirname
     ```

     โดยที่ *dirname* คือชื่อของระบบไฟล์ที่คุณต้องการเปลี่ยนแปลง
  2. พิมพ์:

     ```
     smit chnfsexp
     ```

  3. ป้อนชื่อพาธที่เหมาะสมใน PATHNAME ของฟิลด์ไดเร็กทอรี ที่เอ็กซ์พอร์ต
  4. ทำการเปลี่ยนแปลงใดๆ ที่คุณต้องการ
  5. ออกจาก SMIT
  6. เอ็กซ์พอร์ตระบบไฟล์อีกครั้งโดยป้อน:

     ```
     exportfs /dirname
     ```

     โดยที่ *dirname* คือชื่อของระบบไฟล์ที่คุณเพิ่งเปลี่ยนแปลง

- หากต้องการเปลี่ยนแปลงระบบไฟล์ NFS ที่เอ็กซ์พอร์ตโดยใช้เท็กซ์เอดิเตอร์:
  1. หากต้องการยกเลิกการเอ็กซ์พอร์ตระบบไฟล์ ให้พิมพ์:

     ```
     exportfs -u /dirname
     ```

     โดยที่ *dirname* คือชื่อของระบบไฟล์ที่คุณต้องการเปลี่ยนแปลง
  2. เปิดไฟล์ /etc/exports กับเท็กซ์เอดิเตอร์ ที่คุณคุ้นเคย
  3. ทำการเปลี่ยนแปลงใดๆ ที่คุณต้องการ
  4. บันทึกและปิดไฟล์ /etc/exports
  5. เอ็กซ์พอร์ตระบบไฟล์อีกครั้งโดยป้อน:

     ```
     exportfs /dirname
     ```

     โดยที่ *dirname* คือชื่อของระบบไฟล์ที่คุณเพิ่งเปลี่ยนแปลง

ผู้ใช้ root เข้าถึงระบบไฟล์ที่ถูกเอ็กซ์พอร์ต

เมื่อระบบไฟล์ถูกเอ็กซ์พอร์ต โดยดีฟอลต์ ผู้ใช้ root จะไม่ถูกให้สิทธิของ root เพื่อระบบไฟล์ที่ถูกเอ็กซ์พอร์ตนั้น

เมื่อผู้ใช้ root บนโฮสต์หนึ่งร้องขอการเข้าถึงไฟล์นั้นๆ จาก NFS ID ID ผู้ใช้ของผู้ร้องขอจะถูกแม็พโดย NFS กับ ID ผู้ใช้ของผู้ใช้ nobody (nobody เป็นหนึ่งในชื่อผู้ใช้ที่ถูกวางในไฟล์ /etc/password โดยดีฟอลต์) สิทธิการเข้าถึงของผู้ใช้ nobody จะเหมือนกับที่ให้กับ (*others*) พับลิกของไฟล์นั้นๆ ตัวอย่างเช่น ถ้า *others* มีเฉพาะสิทธิการรันไฟล์ ดังนั้นผู้ใช้ nobody สามารถรันไฟล์เท่านั้น

เพื่อให้ผู้ใช้ root เข้าถึงระบบไฟล์ที่ถูก ทำตามวิธีใน “การเปลี่ยนระบบไฟล์ที่เอ็กซ์พอร์ต” ในหน้า 576ถ้าคุณใช้เมธอด SMIT ระบุในฟิลด์ HOSTS allowed root access ด้วยชื่อของโฮสต์ซึ่งคุณต้องการให้สิทธิ์เข้าถึงเป็น root ถ้าคุณแก้ไขไฟล์ด้วยเท็กซ์เอดิเตอร์ เพิ่ม qualifier -root=hostname กับ entry ของระบบไฟล์ ตัวอย่างเช่น

```
/usr/tps -root=hermes
```

ระบุว่าผู้ใช้ root บนโฮสต์ hermes อาจเข้าถึงไดเร็กทอรี /usr/tps ด้วยสิทธิของ root

การเมาต์ระบบไฟล์ NFS แบบ explicitly

เมื่อต้องการเมาท์ไดเร็กทอรี NFS โดยชัดแจ้ง ใช้โพรซีเดอร์ ต่อไปนี้:

1. ตรวจสอบว่า เซิร์ฟเวอร์ NFS ได้เอ็กซ์พอร์ตไดเร็กทอรี:

   ```
   showmount -e ServerName
   ```

   โดยที่ *ServerName* คือชื่อของเซิร์ฟเวอร์ NFS คำสั่งนี้แสดงชื่อของไดเร็กทอรีที่เอ็กซ์พอร์ตจากเซิร์ฟเวอร์ NFS หากไดเร็กทอรีที่คุณต้องการเมาต์ไม่ได้แสดงอยู่ เอ็กซ์พอร์ต ไดเร็กทอรีจากเซิร์ฟเวอร์

   หมายเหตุ: คำสั่ง **showmount** จะไม่ทำงานสำหรับระบบไฟล์ที่ถูกเอ็กซ์พอร์ตเป็นระบบไฟล์ NFS เวอร์ชัน 4 สำหรับ NFS เวอร์ชัน 4 ไคลเอ็นต์สามารถเมาต์ระบบไฟล์ root สำหรับเซิร์ฟเวอร์และส่งผ่านโครงสร้างไดเร็กทอรีที่เอ็กซ์พอร์ตระบบไฟล์ที่เอ็กซ์พอร์ตเดี่ยว ไม่ได้เมาต์เพื่อเข้าถึงโดยไคลเอ็นต์

2. สร้างจุดเมาต์บนโลคัลโดยใช้คำสั่ง **mkdir** ไดเร็กทอรี null (ว่าง) ที่ทำหน้าที่เป็นจุดเมาต์ (หรือตัวพักวาง) ของ NFS เมาต์ ต้องแสดงไว้สำหรับ NFS เพื่อเสร็จสิ้นการเมาต์ให้เป็นผลสำเร็จ จุดเมาต์นี้สามารถสร้างไดเร็กทอรีอื่นใดๆ ที่เหมือนกันได้ และไม่มีแอ็ตทริบิวต์พิเศษที่จำเป็น
3. โดยพิมพ์:

   ```
   mount ServerName:/remote/directory /local/directory
   ```

   โดยที่ *ServerName* คือชื่อของเซิร์ฟเวอร์ NFS `/remote/directory` คือไดเร็กทอรีบนเซิร์ฟเวอร์ NFS คุณต้องเมาต์ และ `/local/directory` คือจุดเมาต์บนไคลเอ็นต์ NFS
4. บนเครื่องไคลเอ็นต์ให้พิมพ์วิธีลัด SMIT ต่อไปนี้:

   ```
   smit mknfsmnt
   ```
5. ทำการเปลี่ยนแปลงฟิลด์ต่อไปนี้ตามความเหมาะสมสำหรับคอนฟิกูเรัน ของเน็ตเวิร์ก คอนฟิกูเรชันอาจต้องการความสมบูรณ์ ทั้งหมดของรายการบนหน้าจอนี้

   หมายเหตุ: หากอินเตอร์เฟส SMIT ถูกใช้ให้กดคีย์ Tab เพื่อเปลี่ยนค่าที่ถูกต้องสำหรับฟิลด์แต่ละฟิลด์ แต่ *ห้าม* กด Enter จนกว่าจะเสร็จสิ้นขั้นตอนที่ 7

   - PATHNAME ของจุดเมาต์
   - PATHNAME ของไดเร็กทอรีแบบรีโมต
   - HOST ที่ไดเร็กทอรีแบบรีโมตตั้งอยู่
   - MOUNT เดี๋ยวนี้ เพิ่มรายการให้กับ `/etc/filesystems` หรือทั้งสอง?
   - รายการ `/etc/filesystems` จะเมาต์ไดเร็กทอรีบนระบบ RESTART
   - MODE สำหรับระบบไฟล์ NFS นี้
6. เปลี่ยนหรือใช้ค่าดีฟอลต์สำหรับรายการที่เหลืออยู่ ขึ้นอยู่กับ คอนฟิกูเรชัน NFS ของคุณ
7. เมื่อคุณเสร็จสิ้นการเปลี่ยนแปลงบนหน้าจอนี้ SMIT เมาต์ระบบไฟล์ NFS
8. เมื่อฟิลด์ **Command:** แสดง OK ให้ออก SMIT

ระบบไฟล์ NFS พร้อมใช้งาน

ระบบย่อยสำหรับการเมาต์อัตโนมัติ

ระบบย่อย **automount** อนุญาตให้ผู้ใช้ที่ไม่ใช่ root เมาต์ระบบไฟล์แบบรีโมตหากจุดเมาต์เริ่มต้นถูกระบุไว้โดยผู้ใช้ root

ไฟล์ `/etc/auto_master` ระบุข้อมูลนี้ จุดเมาต์เหล่านี้ซึ่งรู้จักกันว่าเป็นคีย์มีความสอดคล้องกับแม็พที่พิจารณาว่า ระบบไฟล์รีโมตถูกเมาต์ผ่านจุดนั้น รูปแบบของไฟล์ `/etc/auto_master` มีดังต่อไปนี้:

```
/key map
```

หมายเหตุ: ไฟล์ `/etc/auto_master` ถูกอ่านเมื่อคำสั่ง **automount** ถูกเรียกใช้งานในตอนต้น และการเปลี่ยนแปลงไม่มีผลบังคับใช้จนกว่า **automount** จะรันอีกครั้ง

แม็พส่วนใหญ่เป็นแม็พโดยตรง แม็พโดยอ้อม และโฮสต์แม็พ

แม็พการนำทาง

แม็พการนำทางต้องการคีย์พิเศษ (/-) ในไฟล์ /etc/auto_master

แม็พคือไฟล์ที่มีรูปแบบต่อไปนี้:

/directkey [*-options*] *server*:*/dir*

เมื่อผู้ใช้เข้าถึงไดเร็กทอรี */directkey* **automount** daemon จะเมาต์ *server*:*/dir* ผ่าน */directkey*

การแม็พโดยอ้อม

ชนิดอื่นๆ ของการแม็พที่พิจารณาว่า ระบบไฟล์แบบรีโมต ที่เมาต์ผ่านจุดเมาต์คือการแม็พโดยอ้อม

แม็พโดยอ้อมมีรูปแบบต่อไปนี้:

indirectkey [*-options*] *server*:*/dir*

เมื่อผู้ใช้เข้าถึงไดเร็กทอรี /key/*indirectkey* **automount** daemon จะเมาต์ *server*:*/dir* ผ่าน /key/*indirectkey*

โฮสต์แม็พ

โฮสต์แม็พจำเป็นต้องใช้แม็พพิเศษ (-hosts) ในไฟล์ /etc/auto_master

automount daemon จะสร้างไดเร็กทอรีย่อย ภายใต้ไดเร็กทอรี /key สำหรับทุกเซิร์ฟเวอร์ที่แสดงรายการอยู่ในไฟล์ /etc/hosts เมื่อผู้ใช้เข้าถึงไดเร็กทอรี /key/server แล้ว **automount** daemon จะเมาต์ไดเร็กทอรีที่เอ็กซ์พอร์ตของเซิร์ฟเวอร์ไปยังไดเร็กทอรี /key/server

การใช้ AutoFS เพื่อเมาต์ระบบไฟล์โดยอัตโนมัติ

AutoFS ขึ้นอยู่กับการใช้คำสั่ง **automount** เพื่อถ่ายทอดข้อมูลการตั้งค่าการเมาต์แบบอัตโนมัติไปยัง **AutoFS** ส่วนขยายเคอร์เนล และสตาร์ท **automountd** daemon

โดยผ่านการเผยแพร่การตั้งค่านี้ ส่วนขยายจะเมาต์ระบบไฟล์โดยอัตโนมัติและโปร่งใสเมื่อใดก็ตามที่ไฟล์หรือไดเร็กทอรีภายในไฟล์นั้นถูกเปิด ส่วนขยายจะบอก **automountd** daemon เกี่ยวกับการร้องขอการเมาต์และการยกเลิกการเมาต์ และ **automountd** daemon จะดำเนินการจริงๆกับเซอร์วิสที่ถูกร้องขอ

เนื่องจากการเชื่อมชื่อกับตำแหน่งเป็นแบบไดนามิกภายใน **automountd** daemon การอัพเดตกับการแม็พ Network Information Service (NIS) จะถูกใช้โดย **automountd** daemon จะไม่เห็นโดยผู้ใช้ นอกจากนี้ ไม่จำเป็นต้องเมาต์ระบบใช้ไฟล์ร่วมกันล่วงหน้าสำหรับแอ็พพลิเคชันที่มีการอ้างถึงแบบฮาร์ดโค้ดกับไฟล์และไดเร็กทอรี และไม่จำเป็นต้องเก็บรักษาเร็กคอร์ดของโฮสต์ที่ต้องถูกเมาต์สำหรับแอ็พพลิเคชันนั้นๆ

AutoFS ยอมให้ระบบไฟล์ถูกเมาต์เมื่อต้องการ ด้วยวิธีการเมาต์ไดเร็กทอรีนี้ ระบบไฟล์ทั้งหมดไม่จำเป็นต้องถูกเมาต์ตลอดเวลา เฉพาะระบบไฟล์ที่ถูกใช้จะถูกเมาต์

ตัวอย่างเช่น เพื่อเมาต์ไดเร็กทอรี NFS โดยอัตโนมัติ:

1. ตรวจสอบว่า NFS เซิร์ฟเวอร์เอ็กซ์พอร์ตไดเร็กทอรีแล้ว โดยการใส่ :

   ```
   showmount -e ServerName
   ```

โดยที่ *ServerName* เป็นชื่อของ NFS เซิร์ฟเวอร์ คำสั่งนี้จะแสดงชื่อของไดเร็กทอรีที่ถูกเอ็กซ์พอร์ตจาก NFS เซิร์ฟเวอร์ในปัจจุบัน

2. สร้างไฟล์ **AutoFS** หลัก และแม็พไฟล์ **AutoFS** จะเมาต์และยกเลิกการเมาต์ไดเร็กทอรีที่ถูกระบุในแม็พไฟล์เหล่านี้ ตัวอย่างเช่น สมมุติว่าคุณต้องการ **AutoFS** ให้เมาต์ไดเร็กทอรี `/local/dir1` และ `/local/dir2` ดังที่ถูกต้องการจาก **serve1** เซิร์ฟเวอร์ บนไดเร็กทอรี `/remote/dir1` และ `/remote/dir2` ตามลำดับ entry ของไฟล์ `auto_master` เป็นดังนี้ :

   ```
   /remote  /tmp/mount.map
   ```

   entry ของไฟล์ `/tmp/mount.map` เป็นดังนี้ :

   ```
   dir1 -rw serve1:/local/dir1
   dir2 -rw serve1:/local/dir2
   ```

3. ต้องแน่ใจว่าส่วนขยายของ **AutoFS**ถูกโหลดและ **automountd** daemon รันอยู่ ซึ่งสามารถทำได้ 2 วิธี:
   a. การใช้คำสั่ง **automount** : ใช้ `/usr/bin/automount -v`
   b. การใช้ **SRC**: ใช้ `lssrc -s automountd` ถ้าระบบย่อย **automountd** ไม่รัน ใช้ `startsrc -s automountd`

      หมายเหตุ: การสตาร์ท **automountd** daemon ด้วยคำสั่ง **startsrc** จะไม่สนใจการเปลี่ยนแปลงใดๆที่ถูกทำกับไฟล์ `auto_master` file.

4. เพื่อหยุด **automount** daemon ใช้คำสั่ง `stopsrc -s automountd`
   ถ้า ในบางกรณี **automountd** daemon ถูกสตาร์ทโดยไม่ได้ใช้ **SRC** ใช้ :

   ```
   kill automountd_PID
   ```

   โดยที่ *automountd_PID* เป็น ID ของกระบวนการของ **automountd** daemon (การรันคำสั่ง **ps –e** จะแสดง ID ของกระบวนการของ **automountd** daemon) คำสั่ง **kill** จะส่งสัญญาณ SIGTERM ไปยัง **automountd** daemon

การสร้างเมาต์ NFS ที่กำหนดไว้ล่วงหน้า

คุณสามารถสร้างการเมาท์ NFS ที่กำหนดไว้ล่วงหน้าโดยใช้หนึ่งใน โพรซีเดอร์ต่อไปนี้

หมายเหตุ: นิยามอ็อพชัน **bg** (พื้นหลัง) และ **intr** (ความสามารถในการอินเตอร์รัปต์) ในไฟล์ `/etc/filesystems` เมื่อสร้างเมาต์ ที่ได้กำหนดไว้ล่วงหน้าซึ่งถูกเมาต์ในระหว่างการเริ่มต้นทำงานกับระบบ เมาต์ที่ไม่สามารถ อินเตอร์รัปต์ได้ และรันอยู่ในพื้นหน้าที่สามารถหยุดไคลเอ็นต์ได้ หากเน็ตเวิร์กหรือเซิร์ฟเวอร์หยุดทำงานเมื่อระบบไคลเอ็นต์เริ่มต้นทำงาน หากไคลเอ็นต์ไม่สามารถ เข้าถึงเน็ตเวิร์กหรือเซิร์ฟเวอร์ ผู้ใช้ต้องสตาร์ทเครื่องอีกครั้งในโหมดการดูแลรักษา และแก้ไขคำร้องของเมาต์ตามความเหมาะสม

- หากต้องการสร้างเมาต์ที่ได้ถูกกำหนดไว้ก่อนผ่าน SMIT:
  1. โดยพิมพ์:

     ```
     smit mknfsmnt
     ```

  2. ระบุค่าในหน้าจอนี้สำหรับแต่ละการเมาต์ที่คุณต้องการกำหนดไว้ก่อน ระบุค่าสำหรับฟิลด์ที่ร้องขอแต่ละฟิลด์ (ฟิลด์ที่ทำเครื่องหมายด้วยเครื่องหมายดอกจัน (*) ในระยะขอบซ้าย) ยังระบุค่าสำหรับฟิลด์อื่นๆ หรือยอมรับ ค่าดีฟอลต์ เมธอดนี้สร้างรายการในไฟล์ `/etc/filesystems` สำหรับเมาต์ที่ต้องการและพยายามเมาต์
- หากต้องการสร้าง NFS ดีฟอลต์เมาต์โดยแก้ไขไฟล์ `/etc/filesystems`:
  1. เปิดไฟล์ `/etc/filesystems` ด้วยเท็กซ์เอดิเตอร์
  2. เพิ่มรายการสำหรับระบบไฟล์รีโมตแต่ละระบบที่จะเมาต์ เมื่อระบบถูกสตาร์ท ตัวอย่าง เช่น:

```
/home/jdoe:
dev = /home/jdoe
mount = false
vfs = nfs
nodename = mach2
options = ro,soft
type = nfs_mount
```

stanza นี้สั่งให้ระบบเมาต์ไดเร็กทอรี /home/jdoe แบบรีโมตผ่านจุดเมาต์บนโลคัล ของชื่อเดียวกัน ระบบไฟล์ถูกเมาต์เป็นแบบอ่านอย่างเดียวกัน (ro) เนื่องจากว่า ระบบไฟล์ยังถูกเมาต์กับ soft ข้อผิดพลาดถูกส่งคืนใน เหตุการณ์ที่เซิร์ฟเวอร์ไม่ได้ตอบกลับ โดยระบุพารามิเตอร์ *type* เป็น nfs_mount ระบบพยายามเมาต์ไฟล์ /home/jdoe (พร้อมกับระบบไฟล์อื่นใดๆ ที่ถูกระบุไว้ในกลุ่ม type = nfs_mount) เมื่อคำสั่ง **mount -t nfs_mount** ถูกเรียกใช้

stanza ตัวอย่างด้านล่างสั่งให้ระบบเมาต์กับระบบไฟล์ /usr/games ณ เวลาที่เริ่มต้นทำงานกับระบบ หากเมาต์ล้มเหลว ระบบดำเนินการเพื่อพยายาม เมาต์ในพื้นหลัง

```
/usr/games:
dev = /usr/games
mount = true
vfs             = nfs
nodename = gameserver
options = ro,soft,bg
type = nfs_mount
```

พารามิเตอร์ต่อไปนี้ จำเป็นต้องมีสำหรับ stanzas ที่เกี่ยวข้องกับ NFS ที่เมาต์:

ไอเท็ม	คำอธิบาย
dev=*filesystem_name*	ระบุชื่อพาธของระบบไฟล์รีโมต ที่เมาต์
mount=[true|false]	หาก true ระบบไฟล์ NFS จะถูกเมาต์เมื่อระบบบูต หาก false ระบบไฟล์ NFS ไม่ได้ถูกเมาต์เมื่อระบบบูต
nodename=*hostname*	ระบุเครื่องโฮสต์ที่รบบไฟล์รีโมต ตั้งอยู่
vfs=nfs	ระบุว่าระบบไฟล์เสมือนที่เมาต์ คือระบบไฟล์ NFS

พารามิเตอร์ต่อไปนี้คือพารามิเตอร์เผื่อเลือกสำหรับ stanzas ที่เกี่ยวข้องกับ ที่เมาต์:

ไอเท็ม	คำอธิบาย
type=*type_name*	นิยามระบบไฟล์ที่เมาต์เป็นส่วนหนึ่งของกลุ่มเมาต์ *type_name* พารามิเตอร์นี้ถูกใช้พร้อมกับคำสั่ง **mount -t** ที่เมาต์กลุ่มของระบบไฟล์ที่ระบุในเวลาเดียวกัน
options=*options*	ระบุพารามิเตอร์ *options* ต่อไปนี้ตั้งแต่หนึ่งตัวขึ้นไป: biods=*N* ระบุจำนวนสูงสุดของ **biod** daemons ที่ต้องการใช้ ค่าดีฟอลต์คือเจ็ดสำหรับ NFS เวอร์ชัน 2 และคือสี่สำหรับ NFS เวอร์ชัน 3 และเวอร์ชัน 4 bg ระบุความพยายามในการเมาต์อีกครั้งในพื้นหลังหากความพยายามในการเมาต์ครั้งแรกล้มเหลว

ไอเท็ม	คำอธิบาย
	`fg` ระบุความพยายามในการเมาต์อีกครั้งในพื้นหน้า หากความพยายามในการเมาต์ครั้งแรกล้มเหลว `noacl` ปิดใช้งานสำหรับการเมาต์นี้เท่านั้น ซึ่งส่วนสนับสนุน Access Control List (ACL) ถูกจัดเตรียมไว้โดยระบบไฟล์ NFS ที่เจอร์นัลแล้ว เมื่อใช้ระหว่างสองระบบ NFS สนับสนุน access control lists หากใช้อ็อพชัน `noacl` เมื่อเมาต์กับระบบไฟล์ NFS ไม่ได้ใช้ ACL ผลกระทบของอ็อพชัน `noacl` เท่ากับสิ่งที่เกิดขึ้นเมื่อไคลเอ็นต์ NFS บนระบบเมาต์จากเซิร์ฟเวอร์ NFS ที่ไม่สนับสนุน ACL สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ ACL โปรดอ้างถึง “การสนับสนุนNFS แอ็คเซสคอนโทรล” ในหน้า 544 `retry=n` ตั้งค่าจำนวนครั้งที่พยายามเมาต์ `rsize=n` ตั้งค่าขนาดบัฟเฟอร์การอ่านเป็นจำนวนไบต์ที่ระบุโดย *n* `wsize=n` ตั้งค่าขนาดบัฟเฟอร์การเขียนเป็นจำนวนไบต์ที่ระบุโดย *n* `timeo=n` ตั้งค่าการหมดเวลา NFS ให้เป็นวินาทีที่สิบซึ่งระบุโดย *n* ใช้เป็นตัวแปรนี้เพื่อหลีกเลี่ยงสถานการณ์ที่สามารถเกิดขึ้นในเน็ตเวิร์กที่การโหลดเซิร์ฟเวอร์ สามารถเป็นสาเหตุให้เวลาตอบสนองไม่เพียงพอ `retrans=n` ตั้งค่าจำนวนของการส่งข้อมูล NFS ให้เป็นจำนวนที่ระบุโดย *n* `port=n` ตั้งค่าเซิร์ฟเวอร์พอร์ตให้เป็นจำนวนที่ระบุโดย *n* `soft` ส่งคืนค่าข้อผิดพลาดหากเซิร์ฟเวอร์ไม่ตอบกลับ `ฮาร์ด` ให้ลองร้องขอต่อจนกว่าเซิร์ฟเวอร์ที่ตอบกลับ **หมายเหตุ:** เมื่อคุณระบุการเมาต์แบบ `hard` ไว้ อาจเป็นไปได้ว่า กระบวนการสามารถหยุดทำงาน ขณะที่รอการตอบกลับ หากต้องการให้สามารถอินเตอร์รัปต์กระบวนการ และจบกระบวนการจากคีย์บอร์ด ให้ใช้ตัวแปร `intr` ในตัวแปรการเมาต์ `intr` อนุญาตการอินเตอร์รัปต์คีย์บอร์ดกับการฮาร์ดเมาต์ `ro` ตั้งค่าตัวแปรแบบอ่านอย่างเดียว

ไอเท็ม	คำอธิบาย
	`rw` ตั้งค่าตัวแปรอ่าน-เขียน ใช้ตัวแปร `hard` พร้อมกับตัวแปรนี้เพื่อหลีกเลี่ยงเงื่อนไขข้อผิดพลาดที่สามารถขัดแย้งกับแอ็พพลิเคชันได้ หากการเมาต์แบบ `soft` มีความพยายามเป็นแบบอ่าน-เขียน โปรดดู “การแก้ปัญหา NFS” ในหน้า 592 สำหรับข้อมูลเพิ่มเติมเกี่ยวกับปัญหาในการเมาต์แบบ hard หรือ soft `secure` ระบุเพื่อใช้โปรโตคอลความปลอดภัยเพิ่มเติมสำหรับรายการดำเนินการ NFS `sec` ตัวเลือก **sec** ระบุรายการการรักษาความปลอดภัยที่ต้องการสำหรับ การเมาต์ NFS การใช้งานที่มีคือ **des, unix, sys, krb5, krb5i** และ **krb5p** ตัวเลือกนี้ใช้กับ AIX 5.3 หรือใหม่กว่าเท่านั้น `actimeo=`*n* ขยายเวลาการล้างข้อมูล *n* วินาทีสำหรับทั้งไฟล์ปกติ และไดเร็กทอรี **หมายเหตุ:** แคชแอ็ตทริบิวต์เก็บไฟล์แอ็ตทริบิวต์ บนไคลเอ็นต์ แอ็ตทริบิวต์สำหรับไฟล์ถูกกำหนดเวลาที่ต้องถูกลบ หากไฟล์ ถูกเมาต์ก่อนเวลาล้างข้อมูล จากนั้นเวลาล้างข้อมูลถูกขยายตามเวลา เนื่องจากการแก้ไขก่อนหน้านี้ (ภายใต้ข้อสรุปที่ไฟล์ที่เปลี่ยนแปลงล่าสุด คือการเปลี่ยนแปลงอีกครั้ง) มีส่วนขยายเวลาล้างข้อมูลต่ำสุด และสูงสุดสำหรับไฟล์ปกติและสำหรับไดเร็กทอรี `vers` ระบุเวอร์ชัน NFS ค่าดีฟอลต์คือเวอร์ชันของโปรโตคอล NFS ที่ใช้ระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ และเป็นค่าสูงสุดที่มีอยู่ บนทั้งสองระบบ หากเซิร์ฟเวอร์ NFS ไม่สนับสนุน NFS เวอร์ชัน 3 แล้ว การเมาต์ NFS จะใช้ NFS เวอร์ชัน 2 ให้ใช้อ็อพชัน **vers** เพื่อเลือกเวอร์ชัน NFS โดยค่าดีฟอลต์ การเมาต์ NFS จะไม่ใช้ NFS เวอร์ชัน 4 ยกเว้นว่าระบุไว้ `acregmin=`*n* คงค่าแอ็ตทริบิวต์ที่แคชไว้อย่างน้อย *n* วินาที หลังการแก้ไขไฟล์ `acregmax=`*n* พักแอ็ตทริบิวต์ที่แคชไม่เกิน *n* วินาที หลังจากการแก้ไขไฟล์ `acdirmin=`*n* คงค่าแอ็ตทริบิวต์ที่แคชไว้อย่างน้อย *n* วินาที หลังการอัพเดตไดเร็กทอรี `acdirmax=`*n* เก็บแอ็ตทริบิวต์ที่แคชเป็นเวลาไม่เกิน *n* วินาที หลังจากอัพเดตไดเร็กทอรี **cio** ระบุระบบไฟล์ที่จะถูก เมาต์สำหรับ ตัวอ่านและตัวเขียน I/O บนไฟล์ในระบบไฟล์ นี้จะทำงานเหมือนกับว่าไฟล์ถูกเปิดด้วย **O_CIO** ที่ระบุ ในการเรียกของระบบ **open( )** การใช้ตัวเลือกนี้จะป้องกัน การเข้าถึงในลักษณะอื่นๆ ที่นอกเหนือจาก CIO ซึ่งเป็นไปไม่ได้ที่จะใช้ I/O ที่แคชแล้วบนระบบไฟล์ที่เมาต์กับอ็อพชัน **cio** นั่นหมายความว่า คำสั่งการแม็พ เช่น **mmap( )** และ **shmat( )** จะล้มเหลวพร้อมกับ **EINVAL** เมื่อถูกใช้กับไฟล์ใดๆ ในระบบไฟล์ ที่เมาต์กับอ็อพชัน **cio** ผลข้างเคียงคือ เป็นไปไม่ได้ที่จะรันไบนารีจากระบบไฟล์ที่เมาต์กับ **cio** เนื่องจากตัวโหลดอาจใช้ **mmap( )** **dio** ระบุว่า I/O บนระบบไฟล์ จะทำงานเหมือนกับว่าไฟล์ทั้งหมดถูกเปิดด้วย **O_DIRECT** ที่ระบุ ในการเรียกของระบบ **open( )** **หมายเหตุ:** การใช้ แฟล็ก **-odio** หรือ **-ocio** สามารถช่วยเรื่องผลการทำงานในเวิร์กโหลดที่แน่นอน แต่ผู้ใช้ควรทราบว่า การใช้แฟล็กเหล่านี้จะป้องกันการแคชไฟล์สำหรับระบบไฟล์เหล่านี้ เนื่องจากปิดใช้งานการอ่านล่วงหน้าสำหรับระบบไฟล์เหล่านี้ อาจทำให้ลด ผลการทำงานสำหรับการอ่านเรียงลำดับขนาดใหญ่

ไอเท็ม	คำอธิบาย

	`maxpout=`*n*
	ระบุระดับของ page-out สำหรับไฟล์บนระบบไฟล์นี้ซึ่ง thread ควรถูก sleep หากระบุ **maxpout** ไว้ คุณยังต้องระบุ **minpout** ค่านี้ต้องไม่เป็นค่าติดลบและมากกว่า **minpout** ค่าดีฟอลต์คือระดับ **maxpout** เคอร์เนล
	`minpout=`*n*
	ระบุระดับ page-out สำหรับไฟล์ระบบไฟล์นี้ที่ thread ควรพร้อมใช้งาน หากระบุ **minpout** ไว้ คุณยังต้องระบุ **maxpout** ไว้ ค่านี้ต้องเป็นค่าติดลบ ค่าดีฟอลต์คือระดับ **minpout** เคอร์เนล
	`rbr` ใช้ความสามารถแบบ release-behind-when-reading เมื่อตรวจพบการอ่านไฟล์ตามลำดับ ในระบบไฟล์นี้ เพจหน่วยความจำจริงถูกใช้โดยไฟล์ ถูกรีลีสแล้วเมื่อเพจถูกคัดลอกไปยังบัฟเฟอร์ภายใน
	หมายเหตุ: หากคุณไม่ได้ตั้งค่าอ็อพชันต่อไปนี้ไว้ เคอร์เนลจะตั้งค่าอ็อพชันเหล่านั้นแบบอัตโนมัติไปยังค่าดีฟอลต์เหล่านี้:
	`fg`
	`retry=10000`
	`rsize=8192`
	`wsize=8192`
	`timeo=7`
	`retrans=5`
	`port=NFS_PORT`
	`ฮาร์ด`
	`secure=off`
	`acregmin=3`
	`acregmax=60`
	`acdirmin=30`
	`acdirmax=60`

3. ลบรายการไดเร็กทอรีใดๆ ที่คุณไม่ต้องการเมาต์แบบอัตโนมัติ ในเวลาเริ่มต้นระบบ
4. บันทึกและปิดไฟล์
5. รันคำสั่ง **mount -a** เพื่อเมาต์ไดเร็กทอรีทั้งหมด ที่ระบุไว้ในไฟล์ `/etc/filesystems`

การยกเลิกการเมาต์อย่างชัดเจนหรือเมาต์ระบบไฟล์แบบอัตโนมัติ

โพรซีเดอร์ต่อไปนี้สามารถใช้เพื่อยกเลิกการเมาต์อย่างชัดเจนหรือเมาต์ไดเร็กทอรี NFS แบบอัตโนมัติ

หากต้องการยกเลิกการเมาต์อย่างชัดเจนหรือเมาต์ไดเร็กทอรี NFS แบบอัตโนมัติ ให้พิมพ์:

```
umount /directory/to/unmount
```

การลบการเมาต์ NFS ที่ได้ถูกกำหนดไว้ก่อน

คุณสามารถลบการเมาท์ NFS ที่กำหนดไว้ล่วงหน้าโดยใช้โพรซีเดอร์ ต่อไปนี้

- หากต้องการลบการเมาต์ NFS ที่ได้ถูกกำหนดไว้ก่อนผ่าน SMIT:
  1. พิมพ์:

     ```
     smit rmnfsmnt
     ```
- หากต้องการลบการเมาต์ NFS ที่ได้ถูกกำหนดไว้ก่อนโดยแก้ไขไฟล์ `/etc/filesystems`:

1. ให้ป้อนคำสั่งต่อไปนี้: `umount /directory/to/unmount`
2. เปิดไฟล์ `/etc/filesystems` กับเอดิเตอร์ที่ คุ้นเคย
3. ค้นหารายการสำหรับไดเร็กทอรีที่คุณเพิ่งยกเลิกการเมาต์ จากนั้นลบ
4. บันทึกและปิดไฟล์

PC-NFS

PC-NFS คือโปรแกรมสำหรับคอมพิวเตอร์ส่วนบุคคลที่เปิดใช้งานคอมพิวเตอร์ส่วนบุคคล เพื่อเมาต์ระบบไฟล์ที่เอ็กซ์พอร์ตโดยเซิร์ฟเวอร์ Network File System (NFS)

คอมพิวเตอร์ส่วนบุคคลยังสามารถร้องขอเน็ตเวิร์กแอดเดรสและชื่อโฮสต์ จากเซิร์ฟเวอร์ NFS นอกจากนี้ หากเซิร์ฟเวอร์ NFS กำลังรัน **rpc.pcnfsd** daemon อยู่ คอมพิวเตอร์ส่วนบุคคลสามารถเข้าถึงการพิสูจน์ตัวตนและเซอร์วิสการสพูลงานพิมพ์

คุณอาจต้องการตั้งค่า **rpc.pcnfsd** daemon บนข้อความต่อไปนี้:

- ระบบที่ดำเนินการกับเซอร์วิสการพิสูจน์ตัวตนของผู้ใช้
- ระบบที่นำเสนอการสพูลงานพิมพ์
- Network Information Service (NIS) ต้นแบบและเซิร์ฟเวอร์แบบ slave ทั้งหมด

หมายเหตุ: เนื่องจากเน็ตเวิร์ก NIS ถูกตั้งค่าตามปกติ ดังนั้น PC-NFS สามารถเลือกเซิร์ฟเวอร์ NIS ใดๆ เป็นเซิร์ฟเวอร์ดีฟอลต์ ซึ่งเป็นสิ่งจำเป็นที่เซิร์ฟเวอร์ทั้งหมด มี **rpc.pcnfsd** daemon ที่รันอยู่ หากรัน daemon นี้บนเซิร์ฟเวอร์ NIS ทั้งหมดไม่ใช้การฝึกปฏิบัติ หรือหากคุณต้องการจำกัดคำร้องขอไปยังเซิร์ฟเวอร์ที่ระบุเฉพาะ ให้เพิ่มคำสั่ง **net pcnfsd** ไปยังไฟล์ `autoexec.bat` บนคอมพิวเตอร์ส่วนบุคคลแต่ละเครื่องเพื่อบังคับให้ใช้เซิร์ฟเวอร์ NIS ที่ระบุเฉพาะ

ข้อมูลที่เกี่ยวข้อง:

Network Information Services (NIS)

เซอร์วิสการพิสูจน์ตัวตน PC-NFS

ตามดีฟอลต์แล้ว PC-NFS แสดงตัวเองกับเซิร์ฟเวอร์ NFS เป็นผู้ใช้ `nobody` ด้วยสิทธิพิเศษ `nobody` ไฟล์ผู้ใช้คอมพิวเตอร์ส่วนบุคคล ปรากฏขึ้นเป็นเจ้าของโดย `nobody` และคุณไม่สามารถแบ่งแยก ระหว่างผู้ใช้คอมพิวเตอร์ส่วนบุคคลอื่นๆ ในลำดับถัดมา

ความสามารถในการพิสูจน์ตัวตนของ **rpc.pcnfsd** daemon อนุญาตให้คุณมอนิเตอร์รีซอร์สของระบบ และความปลอดภัยโดยจดจำผู้ใช้เดี่ยว และกำหนดสิทธิพิเศษอื่นๆ ให้กับผู้ใช้

ด้วย **rpc.pcnfsd** daemon ที่รันอยู่ ผู้ใช้ PC-NFS สามารถออกคำสั่ง **net name** จากคอมพิวเตอร์ส่วนบุคคล เพื่อล็อกอินเข้าสู่ PC-NFS ในวิธีเดียวกับที่ผู้ใช้สามารถล็อกอินเข้าสู่ ระบบปฏิบัติการนี้ ชื่อผู้ใช้และรหัสผ่านถูกตรวจสอบโดย **rpc.pcnfsd** daemon โพรซีเดอร์การพิสูจน์ตัวตนนี้ไม่ได้ทำให้เซิร์ฟเวอร์มีความปลอดภัย แต่ยังจัดเตรียมการควบคุมเพิ่มเติมผ่านการเข้าถึงไฟล์ที่พร้อมใช้งานผ่าน NFS

เซอร์วิสเกี่ยวกับสพูลการพิมพ์ PC-NFS

เซอร์วิสเกี่ยวกับสพูลการพิมพ์ของ **rpc.pcnfsd** daemon เปิดใช้งานคอมพิวเตอร์ส่วนบุคคลที่รัน PC-NFS เพื่อพิมพ์ไปยังพรินเตอร์ที่ไม่ได้พ่วงต่อโดยตรงกับ คอมพิวเตอร์ส่วนบุคคล

โดยเฉพาะอย่างยิ่ง PC-NFS เปลี่ยนทิศทางไฟล์ที่มีเจตนาสำหรับพรินเตอร์แบบคอมพิวเตอร์ส่วนบุคคล ไปยังไฟล์บนเซิร์ฟเวอร์ NFS ไฟล์นี้ถูกวางในไดเร็กทอรีสพูลบนเซิร์ฟเวอร์ NFS **rpc.pcnfsd** daemon เรียกใช้งานสิ่งอำนวยความสะดวก ในการพิมพ์ของเซิร์ฟเวอร์ (ไดเร็กทอรีการสพูลต้องอยู่ในระบบไฟล์ที่เอ็กซ์พอร์ต ดังนั้น ไคลเอ็นต์ PC-NFS สามารถเมาต์ได้) เมื่อ PC-NFS ร้องขอว่า **rpc.pcnfsd** daemon พิมพ์ไฟล์ ซึ่งจัดเตรียมข้อมูลต่อไปนี้:

- ชื่อของไฟล์ที่ต้องถูกพิมพ์
- ID ล็อกอินของผู้ใช้บนไคลเอ็นต์
- ชื่อพรินเตอร์ที่ต้องถูกใช้

การตั้งค่า rpc.pcnfsd daemon

สำหรับผลการทำงานที่ดีที่สุด ให้ตั้งค่า **rpc.pcnfsd** daemon โดยใช้ขั้นตอนเหล่านี้

หากตั้งค่า **rpc.pcnfsd** daemon:

1. ติดตั้งโปรแกรม PC-NFS บนคอมพิวเตอร์ส่วนบุคคลของคุณ
2. เลือกตำแหน่งสำหรับไดเร็กทอรีสพูลบนเซิร์ฟเวอร์ NFS ไดเร็กทอรีสพูลที่เป็นค่าดีฟอลต์คือ `/var/tmp` ไดเร็กทอรีสพูลต้องมีอย่างน้อย 100K ไบต์ของพื้นที่อิสระ
3. เอ็กซ์พอร์ตไดเร็กทอรีสพูล ห้ามวางข้อจำกัดในการเข้าถึงบนไดเร็กทอรี ที่เอ็กซ์พอร์ตที่สามารถเป็นสาเหตุของการเข้าถึงปัญหาในเน็ตเวิร์กของคุณ สำหรับรายละเอียดของโพรซีเดอร์นี้ โปรดดู “การเอ็กซ์พอร์ตระบบไฟล์ NFS” ในหน้า 572
4. สตาร์ท **rpc.pcnfsd** daemon โดยทำตามคำสั่งใน “การสตาร์ท rpc.pcnfsd daemon”
5. ตรวจสอบว่า **rpc.pcnfsd** daemon สามารถเข้าถึงได้โดยคำสั่งใน “การตรวจสอบว่า rpc.pcnfsd daemon สามารถเข้าถึงได้” ในหน้า 587

หมายเหตุ: เนื่องจากคำร้องขอเปลี่ยนทิศทางพรินเตอร์ในบางครั้งเป็นสาเหตุของไฟล์ที่แสดงรายการ ของความยาวที่เป็นศูนย์ให้อยู่ในไดเร็กทอรีสพูล PC-NFS ให้ล้างข้อมูลไดเร็กทอรีการสพูล ของรายการเหล่านี้

การสตาร์ท rpc.pcnfsd daemon

หากต้องการสตาร์ท **rpc.pcnfsd** daemon โดยใช้ไดเร็กทอรีการสพูลดีฟอลต์ ให้ใช้โพรซีเดอร์ต่อไปนี้

1. ด้วยเท็กซ์เอดิเตอร์ให้ยกเลิกคอมเมนต์รายการต่อไปนี้ในไฟล์ `/etc/inetd.conf`:

   ```
   pcnfsd sunrpc_udp udp wait root /usr/sbin/rpc.pcnfsd pcnfsd 150001 1
   ```

2. บันทึกไฟล์และออกจากเท็กซ์เอดิเตอร์

หากต้องการสตาร์ท **rpc.pcnfsd** daemon โดยใช้ไดเร็กทอรี ที่แตกต่างจากค่าดีฟอลต์:

1. Use a text editor to add the following entry to the `/etc/rc.nfs` file:

   ```
   if [ -f /usr/sbin/rpc.pcnfsd ] ; then
   /usr/sbin/rpc.pcnfsd -s spooldir ; echo ' rpc.pcnfsd\c'
   fi
   ```

   โดยที่ *spooldir* ระบุชื่อพาธเต็ม ของไดเร็กทอรีสพูล

2. บันทึกไฟล์และออกจากเท็กซ์เอดิเตอร์
3. การใช้เท็กซ์เอดิเตอร์ให้คอมเมนต์รายการต่อไปนี้ในไฟล์ `/etc/inetd.conf`:

   ```
   #pcnfsd sunrpc_udp udp wait root /usr/sbin/rpc.pcnfsd pcnfsd 150001 1
   ```

การวางเครื่องหมาย pound (#) ที่จุดเริ่มต้นของบรรทัดป้องกัน **inetd** daemon จากการเริ่มต้น **rpc.pcnfsd** daemon การใช้ไดเร็กทอรีสพูล ค่าดีฟอลต์

4. สตาร์ท **rpc.pcnfsd** daemon พรินเตอร์สพูลเลอร์โดยพิมพ์ต่อไปนี้ที่บรรทัดรับคำสั่ง:

   ```
   /usr/sbin/rpc.pcnfsd -s spooldir
   ```

   โดยที่ *spooldir* ระบุชื่อพาธเต็มของไดเร็กทอรีสพูล

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการอัพเดตฐานข้อมูลคอนฟิกูเรชัน **inetd** โปรดดู "การกำหนดคอนฟิก inetd daemon" ในหน้า 377

หมายเหตุ: ไดเร็กทอรีดีฟอลต์ ที่ **rpc.pcnfsd** daemon ไม่สามารถเปลี่ยนจากไฟล์ `inetd.conf`

การตรวจสอบว่า rpc.pcnfsd daemon สามารถเข้าถึงได้

ทำตามโพรซีเดอร์นี้เพื่อพิจารณาว่า **rpc.pcnfsd** daemon สามารถเข้าถึงได้

หากต้องการตรวจสอบว่า **rpc.pcnfsd** daemon สามารถเข้าไปได้ ให้พิมพ์:

```
rpcinfo -u host 150001
```

โดยที่ *host* ระบุชื่อโฮสต์ของระบบที่คุณกำลังตั้งค่า **rpc.pcnfsd** และ 15001 คือหมายเลขโปรแกรม RPC ของ **rpc.pcnfsd** daemon หลังจากที่คุณป้อนคำสั่ง คุณจะได้รับข้อความที่โปรแกรมพร้อมใช้งาน และกำลังรอ

แม็พการเมาต์ LDAP แบบอัตโนมัติ

คุณสามารถตั้งค่าระบบย่อยการเมาต์แบบอัตโนมัติเพื่อเรียกคืนแม็พจากเซิร์ฟเวอร์ LDAP

หากต้องการดูแลแม็พการเมาต์แบบอัตโนมัติใน LDAP ให้เพิ่มบรรทัดต่อไปนี้ลงในไฟล์ `/etc/irs.conf`:

```
automount nis_ldap
```

หากต้องการดูแลแม็พการเมาต์แบบอัตโนมัติใน LDAP คุณจำเป็นต้องสร้างไฟล์ LDIF ที่เหมาะสม คุณสามารถแปลงไฟล์แม็พแบบอัตโนมัติบนโลคัลกับรูปแบบ LDIF โดยที่คำสั่ง **nistoldif** ตัวอย่างเช่น หากเซิร์ฟเวอร์ LDAP มีชื่อว่า `ldapserver` คำลงท้ายแม็พไฟล์ที่มีค่า`dc=suffix` และ `/etc/auto_home` มีบรรทัดต่อไปนี้:

```
user1   server1:/home/user1
user2   server1:/home/user2
user3   server1:/home/user3
```

ใช้คำสั่งต่อไปนี้เพื่อสร้างไฟล์ LDIF สำหรับไฟล์แม็พ `/etc/auto_home` และเพิ่มไปยังเซิร์ฟเวอร์ LDAP:

```
nistoldif -d dc=suffix -sa -f /etc/auto_home > /tmp/auto_home.ldif
ldapadd -D cn=admin -w passwd -h ldapserver -f /tmp/auto_home.ldif
```

หากต้องการแก้ไขหรือลบรายการเมาต์แบบอัตโนมัติที่มีอยู่ออกจากเซิร์ฟเวอร์ LDAP ไฟล์ LDIF ต้องถูกสร้างขึ้นแบบแมนวล ตัวอย่างเช่น ไดเร็กทอรีหลักของ `user2` ซึ่งอยู่บน `server2` LDIF ต่อไปนี้ควรถูกสร้างขึ้น:

```
# cat /tmp/ch_user2.ldif
dn: automountKey=user2,automountMapName=auto_home,dc=suffix
changetype: modify
replace: automountInformation
automountInformation: server2:/home/user2
```

หลังจากที่สร้าง LDIF ข้างต้นแล้ว ให้รันคำสั่งต่อไปนี้:

```
ldapmodify -D cn=admin -w passwd -h ldapserver -f /tmp/ch_user2.ldif
```

คุณต้องสร้างไฟล์ LDIF เพื่อลบผู้ใช้ ตัวอย่างเช่น หากต้องการลบ user3 ให้สร้าง LDIF ต่อไปนี้:

```
# cat /tmp/rm_user3.ldif
dn: automountKey=user3,automountMapName=auto_home,dc=suffix
changetype: delete
```

หลังจากที่สร้าง LDIF ข้างต้นแล้ว ให้รันคำสั่งต่อไปนี้:

```
ldapmodify -D cn=admin -w passwd -h ldapserver -f /tmp/rm_user3.ldif
```

WebNFS

ระบบปฏิบัติการจัดเตรียมความสามารถในเซิร์ฟเวอร์ NFS สำหรับ WebNFS

Defined by Oracle, WebNFS is a simple extension of the NFS protocol that allows easier access to servers and clients through Internet firewalls.

เว็บเบราว์เซอร์ที่เปิดใช้ WebNFS สามารถใช้ NFS universal resource locator (URL) เพื่อเข้าถึงข้อมูลได้โดยตรงจากเซิร์ฟเวอร์ ตัวอย่าง NFS URL คือ:

```
nfs://www.YourCompany.com/
```

WebNFS ทำงานที่เรียงลำดับตามกันมาซึ่งมีโปรโตคอลแบบอิงเว็บเพื่อจัดเตรียมข้อมูล ให้กับไคลเอ็นต์

WebNFS ยังใช้ประโยชน์ของความสามารถในการวัดของเซิร์ฟเวอร์ NFS

ตัวจัดการล็อกของเน็ตเวิร์ก

ตัวจัดการล็อกของเน็ตเวิร์กคือสิ่งอำนวยความสะดวกที่ทำงานในการทำงานร่วมกับ Network File System (NFS) เพื่อจัดการกับลักษณะของ System V ของไฟล์คำแนะนำ และเร็กคอร์ดการล็อกผ่านเน็ตเวิร์ก

ตัวจัดการล็อกของเน็ตเวิร์ก (**rpc.lockd**) และการมอนิเตอร์สถานะ ของเน็ตเวิร์ก (**rpc.statd**) คือ daemon การให้บริการเน็ตเวิร์ก **rpc.statd** daemon คือกระบวนการระดับผู้ใช้ ขณะที่ **rpc.lockd** daemon ถูกนำมาใช้เป็นชุดของ thread เคอร์เนล (คล้ายกับเซิร์ฟเวอร์ NFS) ทั้ง daemons ที่จำเป็นต่อความสามารถของเคอร์เนล เพื่อจัดเตรียมเซอร์วิสของเน็ตเวิร์กพื้นฐาน

หมายเหตุ:

1. การล็อกที่จำเป็นต้องมีหรือการล็อกแบบบังคับไม่ได้สนับสนุนผ่าน NFS
2. ตัวจัดการล็อกของเน็ตเวิร์กคือตัวจัดการเฉพาะกับ NFS เวอร์ชัน 2 และเวอร์ชัน 3

สถาปัตยกรรมตัวจัดการล็อกของเน็ตเวิร์ก

ตัวจัดการล็อกของเน็ตเวิร์กมีทั้งฟังก์ชันเซิร์ฟเวอร์และไคลเอ็นต์

ฟังก์ชันไคลเอ็นต์รับผิดชอบต่อคำร้องขอการประมวลผลจากแอ็พพลิเคชัน และส่งคำร้องขอไปยังตัวจัดการล็อกของเน็ตเวิร์กที่เซิร์ฟเวอร์ ฟังก์ชันเซิร์ฟเวอร์ รับผิดชอบต่อการยอมรับคำร้องขอการล็อกจากไคลเอ็นต์ และสร้างการเรียกของการล็อกที่เหมาะสมที่ระดับของเซิร์ฟเวอร์ จากนั้น เซิร์ฟเวอร์จะตอบกลับคำร้องขอการล็อกของไคลเอ็นต์

ในทางตรงกันข้าม NFS ซึ่งเป็นแบบ stateless ซึ่งตัวจัดการล็อกของเน็ตเวิร์กมี สถานะโดยนัย อีกนัยหนึ่ง ตัวจัดการล็อกของเน็ตเวิร์กต้องจำได้ว่า ไคลเอ็นต์มีล็อกอยู่ในปัจจุบัน สถานะของเน็ตเวิร์กที่มอนิเตอร์ นั่นคือ **rpc.statd** นำโปรโตคอลปกติที่อนุญาตให้ตัวจัดการล็อกของเน็ตเวิร์กเพื่อมอนิเตอร์ สถานะของเครื่องอื่นๆ บนเน็ตเวิร์ก โดยมีข้อมูลสถานะที่แม่นยำ ตัวจัดการล็อกของเน็ตเวิร์กสามารถรักษาสถานะที่สอดคล้องกันภายในสภาพแวดล้อม NFS แบบ stateless

กระบวนการล็อกเน็ตเวิร์กไฟล์

เมื่อแอ็พพลิเคชันต้องการขอรับการล็อกบนโลคัลไฟล์ ซึ่งส่งคำร้องขอไปยังเคอร์เนลโดยใช้ **lockf, fcntl** หรือรูทีนย่อย **flock**

เคอร์เนลประมวลผลคำร้องขอการล็อก อย่างไรก็ตาม หากแอ็พพลิเคชันบนไคลเอ็นต์ NFS สร้างคำร้องขอการล็อกสำหรับรีโมตไฟล์ ไคลเอ็นต์ Network Lock Manager สร้าง Remote Procedure Call (RPC) ไปยังเซิร์ฟเวอร์เพื่อจัดการกับ คำร้องขอ

เมื่อไคลเอ็นต์รับคำร้องขอล็อกแบบรีโมตเริ่มต้น ซึ่งลงทะเบียนสิ่งที่น่าสนใจในเซิร์ฟเวอร์ ด้วย **rpc.statd** ของไคลเอ็นต์ ซึ่งจะเป็นจริง สำหรับตัวจัดการล็อกเน็ตเวิร์กที่เซิร์ฟเวอร์ สำหรับคำร้องขอเริ่มต้นจากไคลเอ็นต์ ซึ่งลงทะเบียนสิ่งที่น่าสนใจในไคลเอ็นต์ด้วยการมอนิเตอร์สถานะ ของเน็ตเวิร์กแบบโลคัล

กระบวนการกู้คืนจากการหยุดทำงาน

rpc.statd daemon บนแต่ละเครื่องแจ้งเตือน **rpc.statd** daemon บนเครื่องอื่นๆ ทุกเครื่องของกิจกรรม เมื่อ **rpc.statd** daemon ได้รับการแจ้งเตือนที่เครื่องอื่นหยุดทำงานหรือถูกกู้คืน ซึ่งแจ้งเตือน **rpc.lockd** daemon

หากเซิร์ฟเวอร์หยุดทำงาน ไคลเอ็นต์ที่มีไฟล์ที่ถูกล็อกต้องสามารถกู้คืน ล็อกของตนเอง หากไคลเอ็นต์หยุดทำงาน ซึ่งเซิร์ฟเวอร์ต้องพักการล็อกไคลเอ็นต์ ขณะที่เรียกคืน นอกจากนี้ หากต้องการสงวน NFS แบบโปร่งใสทั้งหมด การกู้คืนจากการหยุดทำงานต้องเกิดขึ้นโดยไม่ต้องมีการแทรกแซง ของแอ็พพลิเคชันเอง

กระบวนการกู้คืนจากการหยุดทำงานเป็นวิธีการแบบง่ายๆ หากความล้มเหลวของไคลเอ็นต์ถูกตรวจพบ เซิร์ฟเวอร์จะปล่อยให้การล็อกไคลเอ็นต์เกิดความล้มเหลวสำหรับข้อสรุปที่ไคลเอ็นต์แอ็พพลิเคชัน จะร้องขอการล็อกอีกครั้งตามความต้องการ หากการหยุดทำงานและการกู้คืน เซิร์ฟเวอร์ถูกตรวจพบ ตัวจัดการล็อกสำหรับไคลเอ็นต์จะส่งผ่านคำร้องขอการล็อกทั้งหมดก่อนหน้านี้ได้รับอนุญาตโดยเซิร์ฟเวอร์ ข้อมูลที่ถูกส่งผ่านถูกใช้โดยเซิร์ฟเวอร์ เพื่อสร้างสถานะการล็อกอีกครั้งในระหว่างช่วงเวลาผ่อนผัน (ช่วงเวลาผ่อนผัน 45 วินาทีตามค่าดฟอลต์คือระยะเวลาภายในเซิร์ฟเวอร์ที่อนุญาตให้ไคลเอ็นต์ เรียกคืนการล็อกของตนเอง)

rpc.statd daemon ใช้ชื่อโฮสต์ที่เก็บอยู่ใน `/var/statmon/sm` และ `/var/statmon/sm.bak` เพื่อเก็บการติดตามว่าโฮสต์ใดต้องถูกแจ้งเมื่อเครื่องต้องการ กู้คืนการดำเนินการ

การเริ่มต้นตัวจัดการล็อกของเน็ตเวิร์ก

ตามค่าดีฟอลต์ สคริปต์ `/etc/rc.nfs` เริ่มต้น **rpc.lockd** และ **rpc.statd** daemon พร้อมกับ NFS daemon อื่นๆ

หาก NFS กำลังรันอยู่ คุณสามารถตรวจสอบว่า **rpc.lockd** และ **rpc.statd** daemons กำลังรันอยู่โดยใช้คำสั่งต่อไปนี้ใน"การขอรับสถานะปัจจุบันของ NFS daemons" ในหน้า 556 สถานะของทั้งสอง daemon เหล่านี้ควรมีสถานะเป็น *active* หาก **rpc.lockd** และ **rpc.statd** daemons ไม่แอ็คทีฟ และไม่รันอยู่ให้ทำสิ่งต่อไปนี้:

1. การใช้เท็กซ์เอดิเตอร์ที่คุ้นเคย ให้เปิดไฟล์ `/etc/rc.nfs`
2. ค้นหาบรรทัดต่อไปนี้:

```
if [ -x /usr/sbin/rpc.statd ]; then
              startsrc -s rpc.statd
fi
if [ -x /usr/sbin/rpc.lockd ]; then
       startsrc -s rpc.lockd
fi
```

3. หากมีเครื่องหมาย pound (#) ที่จุดเริ่มต้นของบรรทัดเหล่านี้ ให้ลบอักขระทิ้ง จากนั้น บันทึกและออกจากไฟล์ จากนั้นรีสตาร์ท **rpc.statd** และ **rpc.lockd** daemons ทำตามคำสั่งต่อไปนี้ใน "การสตาร์ท NFS daemons" ในหน้า 555

   หมายเหตุ: ลำดับมีความสำคัญ ซึ่งจะสตาร์ท **statd** daemon เป็นอันดับแรกเสมอ

4. หาก NFS กำลังรันและรายการในไฟล์ /etc/rc.nfs ถูกต้อง หยุดและรีสตาร์ท **rpc.statd** และ **rpc.lockd** daemons โดยทำตามคำสั่งใน "การหยุด NFS daemons" ในหน้า 556 และ "การสตาร์ท NFS daemons" ในหน้า 555

   หมายเหตุ: ลำดับมีความสำคัญ ซึ่งจะสตาร์ท **statd** daemon เป็นอันดับแรกเสมอ

หาก **rpc.statd** และ **rpc.lockd** daemons ยังไม่รันอยู่ โปรดดู "การแก้ปัญหาตัวจัดการล็อกของเน็ตเวิร์ก"

การแก้ปัญหาตัวจัดการล็อกของเน็ตเวิร์ก

ปัญหาเกี่ยวกับตัวจัดการล็อกของเน็ตเวิร์กบางข้อที่คุณพบสามารถแก้ไขได้โดยใช้คำแนะนำต่อไปนี้

หากคุณได้รับข้อความบนไคลเอ็นต์ที่คล้ายกับ:

```
clnttcp_create: RPC: Remote System error - Connection refused
rpc.statd:cannot talk to statd at {server}
```

เครื่อง จะคิดว่ามีเครื่องอื่นที่จำเป็นต้องแจ้งให้ทราบว่า อาจมีการใช้การวัดที่กู้คืน เมื่อเครื่องรีสตาร์ท หรือเมื่อ **rpc.lockd** และ **rpc.statd** daemons หยุดทำงานหรือรีสตาร์ท ชื่อเครื่องจะถูกย้ายจาก /var/statmon/sm ไปยัง /var/statmon/sm.bak และ **rpc.statd** daemon พยายามแจ้งให้แต่ละเครื่อง ที่สอดคล้องกันกับแต่ละรายการใน /var/statmon/sm.bak ที่ต้องการโพรซีเดอร์การกู้คืน

หาก **rpc.statd** daemon สามารถเข้าถึงเครื่องได้ รายการของเครื่องใน /var/statmon/sm.bak จะถูกลบออก หาก **rpc.statd** daemon ไม่สามารถเข้าถึงเครื่องได้ daemon นั้นจะลองพยายามอีกครั้งที่ช่วงเวลาปกติ แต่ละครั้งที่เครื่องล้มเหลวในการตอบกลับ การหมดเวลาใช้งานจะสร้างข้อความข้างต้น ในความสนใจของการล็อก integrity daemon จะยังคงพยายามอยู่ อย่างไรก็ตาม ความพยายามนี้สามารถมีผลต่อผลการทำงาน ของการล็อกในทางตรงกันข้าม การจัดการจึงแตกต่างกัน ขึ้นอยู่กับว่าเครื่องเป้าหมายไม่ตอบกลับหรือออกจากระบบที่ใช้งานจริงกึ่งถาวร หากต้องการจัดการกับข้อความ:

1. ตรวจสอบว่า **statd** และ **lockd** daemons บนเซิร์ฟเวอร์กำลังรันโดยคำสั่งต่อไปนี้ใน "การขอกรับสถานะปัจจุบันของ NFS daemons" ในหน้า 556 (สถานะของ daemon สองตัวเหล่านี้ควรมีสถานะเป็น *active*)
2. หาก daemon เหล่านี้ไม่ได้รันอยู่ ให้สตาร์ท **rpc.statd** และ **rpc.lockd** daemon บนเซิร์ฟเวอร์โดยติดตามคำสั่งใน "การสตาร์ท NFS daemons" ในหน้า 555

   หมายเหตุ: ลำดับมีความสำคัญ ซึ่งจะสตาร์ท **statd** daemon เป็นอันดับแรกเสมอ
   หลังจากที่คุณได้รีสตาร์ท daemon แล้ว โปรดจำไว้ว่า มีช่วงเวลาผ่อนผันเกิดขึ้น ในระหว่างเวลานี้ **lockd** daemons อนุญาตให้เรียกคืนคำร้องขอที่มาจากไคลเอ็นต์อื่นๆ ที่จัดการกับล็อกก่อนหน้านี้ พร้อมกับเซิร์ฟเวอร์ ดังนั้น คุณอาจไม่ขอรับล็อกใหม่โดยทันทีหลังจากที่สตาร์ท daemon

หรือ กำจัดข้อความโดย:

1. หยุด **rpc.statd** และ **rpc.lockd** daemon บนไคลเอ็นต์โดยทำตามคำสั่งต่อไปนี้ใน “การหยุด NFS daemons” ในหน้า 556
2. บนไคลเอ็นต์ให้ลบรายการเครื่องเป้าหมายออกจากไฟล์ /var/statmon/sm.bak โดยป้อน:

   ```
   rm /var/statmon/sm.bak/TargetMachineName
   ```

   การดำเนินการนี้ เก็บเครื่องเป้าหมายจากที่ได้พึ่งระวังไว้ว่า อาจจำเป็นต้องทำงานร่วมกันในการล็อก การกู้คืน ซึ่งควรถูกใช้เมื่อสามารถพิจารณาได้ว่า เครื่องไม่มีแอ็พพลิเคชันใดๆ ที่รันอยู่ ซึ่งกำลังทำงานร่วมกันในการล็อก เน็ตเวิร์กด้วยเครื่องที่มีผลกระทบ
3. สตาร์ท **rpc.statd** และ **rpc.lockd** daemons บนไคลเอ็นต์โดยทำตามคำสั่งใน “การสตาร์ท NFS daemons” ในหน้า 555

หากคุณไม่สามารถขอรับล็อกจากไคลเอ็นต์ให้ทำตามวิธีการต่อไปนี้:

1. ใช้คำสั่ง **ping** เพื่อตรวจสอบว่า ไคลเอ็นต์และเซิร์ฟเวอร์สามารถเข้าถึงและจดจำแต่ละเครื่องได้ หากเครื่อง กำลังรันและเครือข่ายที่ไม่เปลี่ยนแปลง ให้ตรวจสอบชื่อโฮสต์ที่แสดงอยู่ในไฟล์ /var/statmon/hosts สำหรับแต่ละเครื่อง ชื่อโฮสต์ต้องตรงกันระหว่างเซิร์ฟเวอร์และไคลเอ็นต์ สำหรับการจดจำเครื่อง หากเซิร์ฟเวอร์ชื่อถูกใช้สำหรับการแก้ไขชื่อโฮสต์ ให้ตรวจสอบว่า ข้อมูลโฮสต์เป็นข้อมูลเดียวกันกับที่อยู่ในไฟล์ /var/statmon/hosts
2. ตรวจสอบว่า **rpc.lockd** และ **rpc.statd** daemons กำลังรันอยู่ทั้งบนไคลเอ็นต์และเซิร์ฟเวอร์โดยทำตามคำสั่งใน “การขอกรับสถานะปัจจุบันของ NFS daemons” ในหน้า 556 สถานะของ ทั้งสอง daemon เหล่านี้ควรมีสถานะเป็น *active*
3. หาก daemon เหล่านี้ไม่แอ็คทีฟ ให้สตาร์ท **rpc.statd** และ **rpc.lockd** daemons โดยทำตามคำสั่งใน “การสตาร์ท NFS daemons” ในหน้า 555
4. หาก daemon เหล่านั้นแอ็คทีฟ คุณอาจจำเป็นต้องรีเซ็ต daemon เหล่านั้นทั้งบนไคลเอ็นต์และเซิร์ฟเวอร์ หากต้องการทำสิ่งนี้ ให้หยุดแอ็พพลิเคชันทั้งหมดที่กำลังร้องขอการล็อก
5. ถัดไป ให้หยุด **rpc.statd** และ **rpc.lockd** daemon ทั้งบนไคลเอ็นต์และเซิร์ฟเวอร์โดยทำตามคำสั่งต่อไปนี้ใน “การหยุด NFS daemons” ในหน้า 556
6. ถึงตอนนี้ให้รีสตาร์ท **rpc.statd** และ **rpc.lockd** daemon บนเซิร์ฟเวอร์เป็นอันดับแรก จากนั้นตามด้วยบนไคลเอ็นต์โดยทำตามคำสั่งใน “การสตาร์ท NFS daemons” ในหน้า 555

   หมายเหตุ: ลำดับ มีความสำคัญ ซึ่งจะสตาร์ท **statd** daemon อันดับแรกเสมอ

หากโพรซีเดอร์ไม่สามารถช่วยแก้ปัญหาการล็อกได้ให้รัน **lockd** daemon ในโหมดการดีบัก โดยทำสิ่งต่อไปนี้:

1. หยุด **rpc.statd** และ **rpc.lockd** daemons ทั้งบนไคลเอ็นต์และเซิร์ฟเวอร์โดยทำตามคำสั่งใน “การหยุด NFS daemons” ในหน้า 556
2. สตาร์ท **rpc.statd** daemon บนไคลเอ็นต์และเซิร์ฟเวอร์โดยทำตามคำสั่งใน “การสตาร์ท NFS daemons” ในหน้า 555 ต่อไปนี้
3. สตาร์ท **rpc.lockd** daemon บนไคลเอ็นต์และเซิร์ฟเวอร์โดยพิมพ์:

   ```
   /usr/sbin/rpc.lockd -d1
   ```

   เมื่อเรียกใช้งานพร้อมกับแฟล็ก **-d1** แล้ว **lockd** daemon จัดเตรียมการวินิจฉัยข้อความ ให้กับ syslog ในครั้งแรก จะมีจำนวนของข้อความที่ทำงานกับช่วงเวลาผ่อนผัน ซึ่งรอให้จำนวนข้อความเหล่านี้หมดเวลาใช้งาน หลังจากที่ช่วงเวลาผ่อนผันหมดเวลาใช้งาน ทั้งบนเซิร์ฟเวอร์และไคลเอ็นต์ใดๆ ให้รันแอ็พพลิเคชันที่ล็อกปัญหา และตรวจสอบว่า คำร้องขอการล็อกถูกส่งผ่านจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์ และจากเซิร์ฟเวอร์ไปยังไคลเอ็นต์

คุณสามารถจำกัดจำนวนช่วงของพอร์ต IP ที่ใช้โดยไคลเอ็นต์ NFS สำหรับสื่อสารกับเซิร์ฟเวอร์ NFS โดยตั้งค่าตัวแปร **NFS_PORT_RANGE** ในไฟล์ /var/statmon/environment

ช่วงของพอร์ต NFS

ตัวแปรสภาวะแวดล้อม **NFS_PORT_RANGE** สามารถใช้เพื่อจำกัดพอร์ตต้นทางของเน็ตเวิร์กที่เรียกไคลเอ็นต์ ที่ทำกับเซิร์ฟเวอร์

ถ้าใช้ ตัวแปรสภาวะแวดล้อมนี้ต้องถูกเพิ่มไปยังไฟล์ /etc/environment รูปแบบของตัวแปรสภาวะแวดล้อมมีดังต่อไปนี้:

```
NFS_PORT_RANGE=udp[4000-5000]:tcp[7000-8000]
```

ในตัวอย่างนี้ แพ็กเก็ต UDP ที่ส่งโดย ไคลเอ็นต์มีพอร์ตต้นทางในช่วง 4000 – 5000 และการเชื่อมต่อ TCP มีพอร์ตต้นทางในช่วง 7000 – 8000 เพื่อหลีกเลี่ยงปัญหาในการนำพอร์ตกลับมาใช้ใหม่ หมายเลขพอร์ต ที่ระบุไว้ในช่วงนี้ต้องไม่ได้ใช้เป็นหมายเลขพอร์ตแบบคงที่สำหรับ Network File System (NFS) daemons ในไฟล์ /etc/services

ความปลอดภัย NFS

ข้อมูลเกี่ยวกับความปลอดภัย NFS สามารถดูได้ในหลาย ตำแหน่ง

หัวข้อ ความปลอดภัยของระบบไฟล์เครือข่าย ใน *การรักษาความปลอดภัย* อธิบาย รายละเอียดเกี่ยวกับความปลอดภัยของ DES สำหรับข้อมูลเพิ่มเติมเกี่ยวกับความปลอดภัยสำหรับ Kerberos โปรดดู “การตั้งค่าเน็ตเวิร์กสำหรับ RPCSEC-GSS” ในหน้า 572

การแก้ปัญหา NFS

เนื่องด้วยเน็ตเวิร์กเซอร์วิส ปัญหาสามารถเกิดขึ้นได้บนเครื่อง ที่ใช้ Network File System (NFS) การแก้ปัญหาสำหรับปัญหาเหล่านี้ เกี่ยวข้องกับความเข้าใจถึงยุทธวิธีสำหรับการติดตามปัญหา NFS การจดจำข้อความแสดงความผิดพลาด ที่เกี่ยวข้องกับ NFS และการเลือกโซลูชันที่เหมาะสม

เมื่อการติดตามปัญหา NFS หยุดลง ให้แยกจุดหลักของความล้มเหลวสามจุดออกจากกัน เพื่อพิจารณาว่าไม่มีการทำงาน: เซิร์ฟเวอร์ ไคลเอ็นต์ หรือตัวเน็ตเวิร์กเอง

หมายเหตุ: โปรดดู “การแก้ปัญหาตัวจัดการล็อกของเน็ตเวิร์ก” ในหน้า 590 สำหรับปัญหาในการล็อกไฟล์

ปัยหาเกี่ยวกับไฟล์แบบ Hard-mounted และ soft-mounted

เมื่อเน็ตเวิร์กหรือเซิร์ฟเวอร์มีปัญหา โปรแกรมที่เข้าถึงรีโมตไฟล์แบบ hard-mounted ล้มเหลวจะแตกต่างจากที่เข้าถึงรีโมตไฟล์แบบ soft-mounted

หากเซิร์ฟเวอร์ล้มเหลวในการตอบกลับไปยังคำร้องขอแบบ hard-mount NFS จะพิมพ์ข้อความ:

```
NFS server hostname not responding, still trying
```

ระบบไฟล์แบบรีโมตแบบ Hard-mounted เป็นสาเหตุทำให้โปรแกรมหยุดทำงานจนกว่าเซิร์ฟเวอร์จะตอบกลับ เนื่องจากไคลเอ็นต์พยายามเมาต์คำร้องขอจนกว่าจะประสบผลสำเร็จ ใช้แฟล็ก **-bg** พร้อมกับคำสั่ง **mount** เมื่อดำเนินการกับการเมาต์แบบ hard หากเซิร์ฟเวอร์ไม่ตอบกลับ ไคลเอ็นต์จะลองเมาต์ใน แบ็กกราวน

หากเซิร์ฟเวอร์ล้มเหลวในการตอบกลับไปยังคำร้องขอแบบ soft-mount NFS จะพิมพ์ข้อความ:

```
Connection timed out
```

ระบบไฟล์แบบรีโมตแบบ Soft-mounted ส่งคืนข้อผิดพลาดหลังจากพยายาม ไม่เป็นผลสำเร็จ โปรแกรมจำนวนมากไม่ได้ถูกตรวจสอบเพื่อส่งคืนเงื่อนไขของการดำเนินการกับระบบไฟล์ ดังนั้น คุณจึงมองไม่เห็นข้อความแสดงความผิดพลาดนี้ เมื่อเข้าถึงไฟล์แบบ soft-mounted อย่างไรก็ตาม ข้อความแสดงความผิดพลาด NFS นี้ พิมพ์อยู่บนคอนโซล

การระบุปัญหา NFS

หากคุณกำลังพบกับปัญหา NFS ให้ทำตามขั้นตอนเหล่านี้

หากไคลเอ็นต์มีปัญหา NFS ให้ทำตามขั้นตอนต่อไปนี้:

1. ตรวจสอบว่าการเชื่อมต่อเน็ตเวิร์กดีอยู่
2. ตรวจสอบว่า **inetd, portmap** และ **biod** daemons กำลังรันอยู่บนไคลเอ็นต์ โดยทำตามคำสั่งใน “การขอกรับสถานะปัจจุบันของ NFS daemons” ในหน้า 556
3. ตรวจสอบว่า จุดเมาต์ที่ถูกต้องมีอยู่สำหรับระบบไฟล์ที่กำลัง เมาต์ สำหรับข้อมูล เพิ่มเติม ให้ดูที่ “การตั้งค่าไคลเอ็นต์ NFS” ในหน้า 570
4. ตรวจสอบว่า เซิร์ฟเวอร์ทำงานอยู่และรันอยู่โดยรันคำสั่งต่อไปนี้ ที่พร้อมต์ของเชลล์ของไคลเอ็นต์:

   ```
   /usr/bin/rpcinfo  -p  server_name
   ```

   หากเซิร์ฟเวอร์ ทำงานอยู่ รายการของโปรแกรม เวอร์ชัน โปรโตคอล และหมายเลขพอร์ต ถูกพิมพ์ ซึ่งคล้ายกับที่แสดงต่อไปนี้:

   ```
   program    vers  proto      port
   100000        2      tcp       111    portmapper
   100000        2      udp       111    portmapper
   100005        1      udp      1025    mountd
   100001        1      udp      1030    rstatd
   100001        2      udp      1030    rstatd
   100001        3      udp      1030    rstatd
   100002        1      udp      1036    rusersd
   100002        2      udp      1036    rusersd
   100008        1      udp      1040    walld
   100012        1      udp      1043    sprayd
   100005        1      tcp       694    mountd
   100003        2      udp      2049    nfs
   100024        1      udp       713    status
   100024        1      tcp       715    status
   100021        1      tcp       716    nlockmgr
   100021        1      udp       718    nlockmgr
   100021        3      tcp       721    nlockmgr
   100021        3      udp       723    nlockmgr
   100020        1      udp       726    llockmgr
   100020        1      tcp       728    llockmgr
   100021        2      tcp       731    nlockmgr
   ```

   หากการตอบกลับไม่ส่งคืน ให้ล็อกอินเข้าสู่เซิร์ฟเวอร์คอนโซลและตรวจสอบสถานะของ **inetd** daemon ด้วยคำสั่งต่อไปนี้ ใน “การขอกรับสถานะปัจจุบันของ NFS daemons” ในหน้า 556
5. ตรวจสอบว่า **mountd, portmap** และ **nfsd** daemons กำลังรันอยู่บนเซิร์ฟเวอร์ NFS โดยป้อนคำสั่งต่อไปนี้ที่พร้อมต์ของเชลล์ ไคลเอ็นต์:

```
/usr/bin/rpcinfo  -u  server_name  mount
/usr/bin/rpcinfo  -u  server_name  portmap
/usr/bin/rpcinfo  -u  server_name  nfs
```

หาก daemons กำลังรันที่เซิร์ฟเวอร์ การตอบกลับต่อไปนี้จะถูกส่งคืน:

```
program  100005  version  1  ready  and  waiting
program  100000  version  2  ready  and  waiting
program  100003  version  2  ready  and  waiting
```

หมายเลขโปรแกรม สอดคล้องกับคำสั่ง ตามที่แสดงอยู่ในตัวอย่างก่อนหน้านี้ หากการตอบไม่ได้ถูกส่งคืน ให้ล็อกอินเข้าสู่เซิร์ฟเวอร์ที่เซิร์ฟเวอร์ คอนโซลและตรวจสอบว่า คำสั่งอยู่ใน “การขอกรับสถานะปัจจุบันของ NFS daemons” ในหน้า 556.

6. ตรวจสอบว่า ไฟล์ `/etc/exports` บนรายการเซิร์ฟเวอร์แสดงชื่อของระบบไฟล์ที่ไคลเอ็นต์ต้องการเมาต์ และที่ระบบไฟล์ถูกเอ็กซ์พอร์ต ทำสิ่งนี้เพื่อป้อนคำสั่ง:

   ```
   showmount -e server_name
   ```

   คำสั่งนี้แสดงระบบไฟล์ที่เอ็กซ์พอร์ตโดย *server_name*
7. สำหรับ NFS เวอร์ชัน 4 ตรวจสอบว่า โดเมน NFSv4 ถูกตั้งค่าไว้อย่างถูกต้อง
8. สำหรับ NFS เวอร์ชัน 4 ตรวจสอบว่า **nfsrgyd** daemon กำลังรันอยู่
9. หากคุณกำลังใช้ความปลอดภัยที่พัฒนาแล้ว โปรดดู “การกำหนดปัญหา RPCSEC-GSS” ในหน้า 600

ข้อผิดพลาดในการเขียนแบบอะซิงโครนัส

เมื่อแอ็พพลิเคชันโปรแกรมเขียนข้อมูลลงในไฟล์ที่อยู่ในระบบไฟล์ที่เมาต์กับ NFS การดำเนินการเขียนตามกำหนดการสำหรับกระบวนการแบบอะซิงโครนัสโดย **biod** daemon

หากเกิดข้อผิดพลาดที่เซิร์ฟเวอร์ NFS ในเวลาเดียวกับที่ข้อมูลถูกเขียนลงในดิสก์จริง ข้อผิดพลาดถูกส่งคืนไปยังไคลเอ็นต์ NFS และ **biod** daemon บันทึกข้อผิดพลาดภายในในโครงสร้างข้อมูล NFS ข้อผิดพลาดที่เก็ฐไว้ถูกส่งคืนตามลำดับ ไปยังแอ็พพลิเคชันโปรแกรมในครั้งถัดไปที่เรียกฟังก์ชัน **fsync** หรือ **close** เนื่องจากมีข้อผิดพลาดในลำดับถัดมา แอ็พพลิเคชันไม่ถูกแจ้งเตือนข้อผิดพลาดสำหรับการเขียน จนกว่าโปรแกรมจะปิดไฟล์ ตัวอย่างทั่วไปของเหตุการณ์นี้คือ เมื่อระบบไฟล์ที่อยู่บนเซิร์ฟเวอร์เต็ม จึงเป็นสาเหตุทำให้เกิดความพยายามในการเขียน โดยไคลเอ็นต์ล้มเหลว

ข้อความแสดงความผิดพลาดของ nfs_server

เมื่อบัฟเฟอร์ที่ส่งผ่านของคุณ มีขนาดเล็กเกินไป ข้อความแสดงความผิดพลาดจะถูกส่งคืน

บัฟเฟอร์ที่ส่งผ่านไม่เพียงพอบนเน็ตเวิร์กของคุณสามารถเป็นสาเหตุทำให้เกิดข้อความแสดงความผิดพลาด ต่อไปนี้ได้:

```
nfs_server:  bad  sendreply
```

เมื่อต้องการเพิ่มบัฟเฟอร์การส่ง ใช้ System Management Interface Tool (SMIT) fast path, `smit commodev` ดังนั้น เลือกชนิดของอะแด็ปเตอร์ของคุณ และเพิ่มจำนวนของ บัฟเฟอร์ที่ส่งผ่าน

ข้อความแสดงความผิดพลาดในการเมาต์

กระบวนการเมาต์แบบรีโมตสามารถล้มเหลวในหลายๆ วิธี ข้อความแสดงความผิดพลาด เชื่อมโยงกับความล้มเหลวในการเมาต์ถูกอธิบายถึงที่นี่

`mount: ... already mounted`
ระบบไฟล์ที่คุณกำลังพยายามเมาต์ได้ถูกเมาต์แล้ว

`mount: ... not found in /etc/filesystems`
ระบบไฟล์ที่ระบุไว้หรือชื่อไดเร็กทอรีไม่สามารถจับคู่ได้

หากคุณออกคำสั่ง **mount** กับไดเร็กทอรีหรือชื่อระบบไฟล์ แต่ไม่ใช่ทั้งสอง คำสั่งจะมองหาในไฟล์ `/etc/filesystems` สำหรับรายการที่มีระบบไฟล์หรือฟิลด์ไดเร็กทอรีที่ตรงกับอาร์กิวเมนต์ หากคำสั่ง **mount** ค้นหารายการ ดังเช่นต่อไปนี้:

```
/dancer.src:
          dev=/usr/src
          nodename  =  d61server
          type  =  nfs
          mount           = false
```

ดังนั้น จึงดำเนินการเมาต์ หากคุณได้ป้อนคำสั่งต่อไปนี้ที่บรรทัดรับคำสั่ง:

```
/usr/sbin/mount -n dancer -o rw,hard /usr/src /dancer.src
```

`... not in hosts database`
บนเน็ตเวิร์กที่ไม่มี Network Information Service ข้อความนี้บ่งชี้ว่า โฮสต์ที่ระบุในคำสั่ง **mount** ไม่ได้อยู่ในไฟล์ `/etc/hosts` บนเน็ตเวิร์กที่รัน NIS ข้อความบ่งชี้ว่า NIS ไม่สามารถค้นหาชื่อโฮสต์ในฐานข้อมูล `/etc/hosts` หรือที่ NIS **ypbind** daemon บนเครื่องของคุณเองได้ตายแล้ว หากไฟล์ `/etc/resolv.conf` มีอยู่แล้ว ดังนั้น เซิร์ฟเวอร์รายชื่อ ถูกใช้สำหรับการแก้ไขรายชื่อโฮสต์ ซึ่งอาจเป็นปัญหาในฐานข้อมูล **named** โปรดดู “การแก้ไขปัญหาเรื่องชื่อโฮสต์บนเซิร์ฟเวอร์ NFS” ในหน้า 599

ตรวจสอบการสะกดทำและไวยากรณ์ในคำสั่ง **mount** ของคุณ หากคำสั่งนั้นถูกต้อง เน็ตเวิร์กของคุณจะไม่รัน NIS และคุณขอรับข้อความนี้ สำหรับชื่อโฮสต์นี้ ให้ตรวจสอบรายการในไฟล์ `/etc/hosts`

หากเน็ตเวิร์กของคุณ กำลังรัน NIS ตรวจสอบว่า **ypbind** daemon กำลังรันโดยป้อนต่อไปนี้ที่บรรทัดรับคำสั่ง:

```
ps -ef
```

คุณควรดู **ypbind** daemon ในรายการ ให้พยายามใช้คำสั่ง **rlogin** เพื่อล็อกอินแบบรีโมตเข้าสู่เครื่องอื่น หรือใช้คำสั่ง **rcp** กับบางสิ่งที่คัดลอกแบบรีโมตกับเครื่องอื่น หากสิ่งนี้ล้มเหลว **ypbind** daemon ของคุณหยุดทำงานหรือติดขัด

หากคุณขอรับข้อความนี้สำหรับชื่อโฮสต์นี้ ตรวจสอบรายการ `/etc/hosts` บนเซิร์ฟเวอร์ NIS

`mount: ... server not responding: port mapper failure - RPC timed out`
เซิร์ฟเวอร์ที่คุณกำลังเมาต์หยุดทำงานหรือตัวแม็พพอร์ต หยุดทำงานหรือติดขัด อย่างใดอย่างหนึ่ง ให้ลองรีสตาร์ทเซิร์ฟเวอร์เพื่อเรียกใช้งาน **inetd, portmap** และ **ypbind** daemons

หากคุณไม่สามารถล็อกอินเข้าสู่เซิร์ฟเวอร์ แบบรีโมตด้วยคำสั่ง **rlogin** แต่เซิร์ฟเวอร์เริ่มทำงาน ให้ตรวจสอบการเชื่อมต่อเน็ตเวิร์กเพื่อล็อกอินแบบรีโมตไปยังเครื่องอื่นๆ บางเครื่อง และตรวจสอบการเชื่อมต่อเน็ตเวิร์กเซิร์ฟเวอร์

`mount: ... server not responding: program not registered`
ซึ่งหมายความว่า คำสั่ง **mount** ได้รับผ่านไปยังตัวแม็พพอร์ต แต่ **rpc.mountd** NFS mount daemon ไม่ได้ถูกลงทะเบียน

`mount: access denied ...`
ชื่อเครื่องของคุณไม่ได้อยู่ในรายการเอ็กซ์พอร์ตสำหรับระบบไฟล์ ที่คุณกำลังลองพยายามเมาต์จากเซิร์ฟเวอร์

คุณสามารถขอรับรายการของเซิร์ฟเวอร์ ซ่งเป็นระบบไฟล์ที่เอ็กซ์พอร์ตโดยรันคำสั่งต่อไปนี้ที่บรรทัดคำสั่ง:

`showmount -e hostname`

หากระบบไฟล์ที่คุณต้องการไม่ให้อยู่ในรายการ หรือชื่อเครื่องของคุณหรือชื่อ netgroup ไม่ได้อยู่ในรายการของคุณ สำหรับระบบไฟล์ให้ล็อกอินเข้าสู่เซิร์ฟเวอร์ และตรวจสอบไฟล์ `/etc/exports` สำหรับรายการระบบไฟล์ที่ถูกต้อง ชื่อระบบไฟล์ที่ปรากฏขึ้นในไฟล์ `/etc/exports` แต่ไม่ได้อยู่ในเอาต์พุตจากคำสั่ง **showmount** บ่งชี้ถึงความล้มเหลวใน **mountd** daemon daemon ไม่สามารถวิเคราะห์บรรทัดนั้นในไฟล์ ซึ่งไม่สามารถค้นหาไดเร็กทอรี หรือชื่อไดเร็กทอรีไม่ได้เมาต์ไดเร็กทอรีบนโลคัล อย่างใดอย่างหนึ่ง หากไฟล์ `/etc/exports` ค้นหาความถูกต้องและเน็ตเวิร์กของคุณรัน NIS ให้ตรวจสอบ **ypbind** daemon บนเซิร์ฟเวอร์ ซึ่งอาจหยุดทำงานหรือติดขัด

`mount: ...: Permission denied`

ข้อความนี้คือการบ่งชี้ทั่วไปที่ส่วนของการพิสูจน์ตัวตนบางส่วน ล้มเหลวบนเซิร์ฟเวอร์ ซึ่งในตัวอย่างข้างต้น คุณไม่ได้อยู่ในรายการเอ็กซ์พอร์ต เซิร์ฟเวอร์ไม่สามารถจดจำเครื่อง **ypbind** daemon ของคุณ หรือเซิร์ฟเวอร์ไม่ได้ยอมรับการระบุที่คุณจัดเตรียมไว้

ตรวจสอบไฟล์ `/etc/exports` บนเซิร์ฟเวอร์และ **ypbind** daemon หากสามารถเรียกใช้งานได้ ในกรณีนี้ คุณสามารถเปลี่ยนชื่อโฮสต์ของคุณด้วยคำสั่ง **hostname** และลองคำสั่ง **mount**

`mount: ...: Not a directory`

พาธแบบรีโมตหรือพาธแบบโลคัลอย่างใดอย่างหนึ่งไม่ใช่ไดเร็กทอรี ตรวจสอบ การสะกดคำในคำสั่งของคุณ และลองรันทั้งสองไดเร็กทอรี

`mount: ...: You are not allowed`

คุณต้องมีสิทธิแบบผู้ใช้ root หรือไม่ได้เป็นสมาชิกของกลุ่มของระบบเพื่อรันคำสั่ง **mount** บนเครื่องของคุณ เนื่องจากมีผลต่อระบบไฟล์ สำหรับผู้ใช้ทั้งหมดบนเครื่องนั้น NFS เมาต์และยกเลิกการเมาต์ที่ได้รับอนุญาตให้ใช้โดย ผู้ใช้ root และสมาชิกของกลุ่มของระบบเท่านั้น

ข้อมูลที่เกี่ยวข้อง:

Network Information Services (NIS)

สาเหตุของเวลาเข้าถึงช้าสำหรับ NFS

ถ้าการเข้าถึงไฟล์รีโมตช้าผิดปกติ ตรวจสอบให้แน่ใจว่าเวลาเข้าถึงไม่ถูกยับยั้งโดย daemon ที่หลุดไป, บรรทัด **tty** ที่ไม่ถูกต้อง หรือปัญหาเดียวกัน

การเชื่อมต่อเน็ตเวิร์ก:

ใช้คำสั่ง **nfsstat** เพื่อรวบรวมข้อมูล เกี่ยวกับการเชื่อมต่อเน็ตเวิร์กของคุณ

คำสั่ง **nfsstat** กำหนดว่า คุณกำลังปล่อยแพ็กเก็ต ใช้คำสั่ง **nfsstat -c** และ **nfsstat -s** เพื่อพิจารณาว่า ไคลเอ็นต์หรือเซิร์ฟเวอร์ที่กำลังข้อมูลบล็อก ขนาดใหญ่ การส่งข้อมูลอีกครั้งยังมีความเป็นไปได้ เนื่องจาก แพ็กเก็ตสูญหายหรือเซิร์ฟเวอร์ไม่ว่าง อัตราการส่งผ่านข้อมูลใหม่ห้าเปอร์เซ็นต์ หรือมากกว่าจะถูกนำมาพิจารณาสูง

ความน่าจะเป็นของการส่งข้อมูลใหม่สามารถลดจำนวนลงโดยเปลี่ยนอะแด็ปเตอร์ การสื่อสารสำหรับพารามิเตอร์คิวการส่งข้อมูล เมนู SMIT สามารถใช้เพื่อเปลี่ยน พารามิเตอร์เหล่านี้ ตัวอย่างเช่น โปรดอ้างถึง อินเตอร์เฟสของระบบที่มีอยู่ใน *การจัดการระบบปฏิบัติการและอุปกรณ์*

ค่าต่อไปนี้ถูกแนะนำให้ใช้สำหรับเซิร์ฟเวอร์ NFS

หมายเหตุ:

1. ใช้ค่าเหล่านี้กับไคลเอ็นต์ NFS หากการส่งผ่านข้อมูลใหม่ยังคงอยู่
2. โหนดทั้งหมดบนเน็ตเวิร์กต้องใช้ขนาด MTU ที่มีขนาดเดียวกัน

ตารางที่ 93. ขนาดของ Communication Adapter Maximum Transmission Unit (MTU) และ Transmit Queue

Adapter	MTU	Transmit queue
Token Ring		
4 Mb	1500	50
	3900	40 (เพิ่มขึ้นหากคำสั่ง **nfsstat** หมดเวลาใช้งาน)
16 Mb	1500	40 (เพิ่มขึ้นหากคำสั่ง **nfsstat** หมดเวลาใช้งาน)
	8500	40 (เพิ่มขึ้นหากคำสั่ง **nfsstat** หมดเวลาใช้งาน)
Ethernet	1500	40 (เพิ่มขึ้นหากคำสั่ง **nfsstat** หมดเวลาใช้งาน)

ขนาด MTU ที่ใหญ่กว่าสำหรับแต่ละตัวประมวลผลที่ลดความเร็วโทเค็นริงแต่ละตัว ที่ใช้และปรับปรุงการดำเนินการอ่าน/เขียน

การตั้งค่าขนาดของ MTU:

เมื่อต้องการตั้งค่าขนาด MTU ใช้ SMIT fast path, `smit chif`

เลือกอะแด็ปเตอร์ให้เหมาะสม และป้อนค่า MTU ในฟิลด์ Maximum IP Packet Size

คำสั่ง **ifconfig** สามารถใช้เพื่อตั้งค่าขนาด MTU (และ *ต้อง* ถูกใช้เพื่อตั้งค่าขนาด MTU ที่ 8500) รูปแบบสำหรับคำสั่ง **ifconfig** คือ:

```
ifconfig trn NodeName up mtu MTUSize
```

โดยที่ `trn` คือชื่ออะแด็ปเตอร์ ตัวอย่างเช่น `tr0`

เมธอดอื่นๆ ของการตั้งค่าขนาด MTU รวมเข้ากับคำสั่ง **ifconfig** ด้วย SMIT

1. เพิ่มคำสั่ง **ifconfig** สำหรับโทเค็นริง ตามที่แสดงในตัวอย่างก่อนหน้านี้ให้กับไฟล์ `/etc/rc.bsdnet`
2. ป้อนวิธีลัด `smit setbootup_option` สลับฟิลด์ **ใช้ลักษณะของ BSD** ให้มีค่า **yes**

ขนาดคิวในการส่งผ่าน:

ขนาดคิวในการส่งผ่านของอะแด็ปเตอร์การสื่อสารถูกตั้งค่าด้วย SMIT

ป้อนพาธด่วน `smit chgtok` เลือกอะแด็ปเตอร์ที่เหมาะสม และป้อนขนาดคิวในฟิลด์ Transmit

โปรแกรมหยุดทำงาน:

หากโปรแกรมหยุดทำงานในระห่างการทำงานที่เกี่ยวข้องกับไฟล์ เซิร์ฟเวอร์ NFS ไม่สามารถหยุดทำงานได้

ในกรณีนี้ ข้อความแสดงความผิดพลาดต่อไปนี้อาจถูกแสดง:

`NFS server hostname not responding, still trying`

เซิร์ฟเวอร์ NFS (*hostname*) หยุดทำาน ซึ่งบ่งชี้ว่า เกิดปัญหากับเซิร์ฟเวอร์ NFS การเชื่อมต่อเน็ตเวิร์ก หรือเซิร์ฟเวอร์ NIS

ตรวจสอบเซิร์ฟเวอร์จากที่คุณได้เมาต์ระบบไฟล์หากเครื่องของคุณ หยุดทำงานอย่างสมบูรณ์ หากมีหนึ่งในเซิร์ฟเวอร์หยุดทำงาน ปัญหานี้จะไม่เกี่ยวข้อง เมื่อ เซิร์ฟเวอร์กลับมาทำงาน โปรแกรมของคุณจะยังคงเป็นอัตโนมัติ ไม่มีไฟล์ที่ ถูกทำลาย

หากเซิร์ฟเวอร์แบบ soft-mounted หยุดทำงาน งานอื่นจะไม่มีผลกระทบ โปรแกรมที่หมดเวลาใช้งาน ขณะที่พยายามเข้าถึงรีโมตไฟล์แบบ soft-mounted ล้มเหลวพร้อมกับข้อความ `errno` แต่คุณยังคงสามารถเข้าถึงระบบไฟล์อื่นๆ ของคุณได้

หากเซิร์ฟเวอร์ทั้งหมดกำลังรันอยู่ ให้พิจารณาบุคคลอื่นที่กำลังใช้เซิร์ฟเวอร์เดียวกัน นั้นอาจมีปัญหา มากกว่าหนึ่งเครื่องที่มีปัญหาด้านการให้บริการ ซึ่งบ่งชี้ปัญหากับ **nfsd** daemons บนเซิร์ฟเวอร์ ในกรณีนี้ ให้ล็อกอินเข้าสู่เซิร์ฟเวอร์และรันคำสั่ง **ps** เพื่อดูว่า **nfsd** daemon กำลังรันและสะสมเวลา CPU หากไม่ คุณอาจสามารถหยุด และรีสตาร์ท **nfsd** daemon หากไม่เป็นผล คุณอาจต้องรีสตาร์ทเซิร์ฟเวอร์

ตรวจสอบการเชื่อมต่อเน็ตเวิร์กและเชื่อมต่อเซิร์ฟเวอร์ หากระบบยังคงทำงานและรันอยู่

scheme สิทธิและการพิสูจน์ตัวตน:

ในบางครั้ง หลังจากการเมาต์ถูกสร้างขึ้น มีปัญหาในการอ่าน เขียน หรือสร้างรีโมตไฟล์หรือไดเร็กทอรี ซึ่งเป็นไปได้ยากตามปกติเนื่องจากปัญหาเกี่ยวกับสิทธิหรือการพิสูจน์ตัวตน

ปัญหาเกี่ยวกับสิทธิและการพิสูจน์ตัวตนสามารถเปลี่ยนแปลงเป็นสาเหตุที่ขึ้นอยู่กับ NIS ที่ต้องถูกใช้และป้องกันความปลอดภัยของการเมาต์ที่ระบุไว้

กรณีที่ง่ายที่สุดเกิดขึ้นเมื่อ เมาต์ที่ไม่มีความปลอดภัยจะถูกระบุไว้ และ NIS ไม่ได้ถูกใช้ในกรณีนี้ ID ผู้ใช้ (UIDs) และ ID กลุ่ม (GIDs) ถูกแม็พ ผ่านเซิร์ฟเวอร์ไฟล์ `/etc/passwd` และไฟล์ไคลเอ็นต์ `/etc/group` ใน scheme นี้ สำหรับผู้ใช้ที่ชื่อ `B` ที่ต้องถูกระบุทั้งบนไคลเอ็นต์ และเซิร์ฟเวอร์เป็น `B` ผู้ใช้ `B` ต้องมีหมายเลข UID เดียวกันในไฟล์ `/etc/passwd` ต่อไปนี้คือตัวอย่างของวิธีการที่อาจทำให้เกิดปัญหา:

```
User  B  is  uid  200  on  client  foo.
User  B  is  uid  250  on  server  bar.
User  G  is  uid  200  on  server  bar.
```

ไดเร็อทอรี `/home/bar` ถูกเมาต์จากเซิร์ฟเวอร์ `bar` บนไคลเอ็นต์ `foo` หากผู้ใช้ `B` กำลังแก้ไขไฟล์ บนระบบรีโมตไฟล์ `/home/bar` บนไคลเอ็นต์ `foo` ความสับสนของผลลัพธ์อาจเกิดขึ้นได้เมื่อบันทึกไฟล์

เซิร์ฟเวอร์ `bar` คิดว่าไฟล์เป็นของ `user G` เนื่องจาก `G` คือ UID `200` บน `bar` หาก `B` ล็อกออนเข้าสู่ `bar` ไดเร็กทอรีโดยใช้คำสั่ง **rlogin** ซึ่งอาจไม่สามารถเข้าถึงไฟล์ ที่เพิ่งสร้างขณะการทำงานบนระบบไฟล์เมาต์รีโมต `G` อย่างไรก็ตาม มีความสามารถที่จะทำเนื่องจากกลไกของสิทธิโดย UID ไม่ใช่ชื่อ

โซลูชันถาวรเฉพาะกับคือ กำหนด UID ที่สอดคล้องกันอีกครั้งบน สองเครื่อง ตัวอย่างเช่น กำหนด `B` UID `200` บนเซิร์ฟเวอร์ `bar` หรือ `250` บนไคลเอ็นต์ `foo` ไฟล์ที่เป็นเจ้าของ `B` อาจต้องการให้มีคำสั่ง **chown** รันอีกครั้ง เพื่อทำให้ตรงกับ ID ใหม่บนเครื่องที่เหมาะสม

เนื่องจากปัญหาที่เกิดขึ้นกับการแม็พ UID และ GID ที่สอดคล้องกันบนเครื่องทั้งหมดในเครือข่าย NIS ถูกใช้เพื่อดำเนินการแม็พตามความเหมาะสม เพื่อหลีกเลี่ยง ชนิดของปัญหานี้

การแก้ไขปัญหาเรื่องชื่อโฮสต์บนเซิร์ฟเวอร์ NFS:

เมื่อเซิร์ฟเวอร์ NFS ให้บริการคำร้องขอเมาต์ เซิร์ฟเวอร์จะมองหาชื่อของไคลเอ็นต์ ที่สร้างคำร้องขอ เซิร์ฟเวอร์ใช้ Internet Protocol (IP) address ของไคลเอ็นต์และค้นหาชื่อโฮสต์ที่สอดคล้องกันซึ่งตรงกับแอดเดรสนั้น

หลังจากที่พบชื่อโฮสต์แล้ว เซิร์ฟเวอร์จะมองหารายการเอ็กซ์พอร์ต สำหรับไดเร็กทอรีที่ร้องขอและตรวจสอบการมีอยู่ของชื่อไคลเอ็นต์ใน รายการเข้าถึงสำหรับไดเร็กทอรี ในรายการที่มีอยู่สำหรับไคลเอ็นต์ และรายการที่ตรงกับที่ส่งคืนสำหรับการแก้ปัญหาเรื่องชื่อ ส่วนนั้นของการพิสูจน์ตัวตนเมาต์จะผ่าน

หากเซิร์ฟเวอร์ไม่สามารถดำเนินการกับการแก้ไขปัญหาเรื่องชื่อ IP address-to-host-name เซิร์ฟเวอร์จะปฏิเสธคำร้องขอการเมาต์ เซิร์ฟเวอร์ต้องสามารถค้นหาการจับคู่ สำหรับ IP แอดเดรสของไคลเอ็นต์เพื่อสร้างคำร้องขอเมาต์ หากไดเร็กทอรีที่เอ็กซ์พอร์ต ด้วยการเข้าถึงไคลเอ็นต์ เซิร์ฟเวอร์ยังคงต้องสามารถแปลงชื่อกลับ เพื่อค้นหาซึ่งอนุญาตให้ใช้คำร้องขอเมาต์

เซิร์ฟเวอร์ต้องสามารถค้นหาชื่อที่ถูกต้องสำหรับไคลเอ็นต์ ตัวอย่างเช่น หากชื่อมีรายการอยู่ในไฟล์ `/etc/exports` ดังเช่นที่แสดงต่อไปนี้:

```
/tmp      -access=silly:funny
```

รายการที่สอดคล้องกันต่อไปนี้จะมีอยู่ในไฟล์ `/etc/hosts`:

```
150.102.23.21        silly.domain.name.com
150.102.23.52        funny.domain.name.com
```

โปรดสังเกตว่า ชื่อไม่สอดคล้องกันอย่างชัดเจน เมื่อเซิร์ฟเวอร์มองหา IP address-to-host-name ที่ตรงกับกับโฮสต์ `silly` และ `funny` ชื่อสตริงไม่ตรงกับรายการอย่างชัดเจนในรายการเข้าถึงของการ เอ็กซ์พอร์ต ปัญหาในแก้ไขปัญหาเรื่องชื่อชนิดนี้เกิดขึ้นเมื่อ้ **named** daemon สำหรับการแก้ไขปัญหาเรื่องชื่อ ฐานข้อมูล **named** daemon โดยส่วนใหญ่มี alias สำหรับชื่อโดเมนเต็มของโฮสต์ ดังนั้น ผู้ใช้ไม่ต้องป้อน ชื่อเต็มเมื่ออ้างถึงโฮสต์ แม้ว่า รายการแอดเดรส host-name-to-IP เหล่านี้มีอยู่สำหรับ alias การมองหาย้อนกลับอาจไม่มีอยู่ ฐานข้อมูลสำหรับการค้นหาชื่อย้อนกลับ (IP แอดเดรสไปยังชื่อโฮสต์) มีรายการที่มี IP แอดเดรสและชื่อโดเมนเต็ม (ไม่ใช่ alias) ของโฮสต์นั้น บางครั้งรายการเอ็กซ์พอร์ตถูกสร้างขึ้นด้วย alias ที่สั้นกว่า จึงเป็นสาเหตุทำให้เกิดปัญหาเมื่อไคลเอ็นต์พยายามเมาต์

ข้อจำกัดเกี่ยวกับจำนวนของกลุ่มในโครงสร้าง NFS:

บนระบบที่ใช้ NFS เวอร์ชัน 2 หรือ 3 ผู้ใช้ไม่สามารถเป็น สมาชิกได้มากกว่า 16 กลุ่มโดยไม่มีความซับซ้อน

กลุ่มถูกนิยามไว้โดยคำสั่ง **groups** หากผู้ใช้คือสมาชิกของกลุ่ม 17 กลุ่มหรือมากกว่า และผู้ใช้พยายามเข้าถึงไฟล์ ที่เป็นเจ้าของโดยกลุ่มที่ 17 (หรือมากกว่า) ระบบจะไม่อนุญาตให้อ่านหรือ คัดลอกไฟล์ หากต้องการอนุญาตให้ผู้ใช้เข้าถึงไฟล์ต่างๆ ให้จัดเรียงลำดับตามกลุ่ม ใหม่

ข้อมูลข้างต้นกล่าวถึงลักษณะการทำงานดีฟอลต์ โปรดดูพารามิเตอร์ **maxgroups** ของคำสั่ง **mount** สำหรับรายละเอียดเพิ่มเติม

เซิร์ฟเวอร์ NFS กับเวอร์ชันก่อนหน้าของ NFS:

ไคลเอ็นต์ NFS เวอร์ชัน 3 ไม่สามารถเมาต์กับเซิร์ฟเวอร์ NFS เวอร์ชัน 4 ได้

เมื่อเมาต์ระบบไฟล์จากเซิร์ฟเวอร์ NFS ก่อนหน้าเวอร์ชันกับไคลเอ็นต์ NFS เวอร์ชัน 3 ปัญหาจะเกิดขึ้น เมื่อผู้ใช้บนไคลเอ็นต์ที่เรียกใช้การเมาต์คือ สมาชิกของที่มีมากกว่าแปดกลุ่ม เซิร์ฟเวอร์บางตัวไม่สามารถทำงานกับสถานการณ์นี้ได้อย่างถูกต้อง และปฏิเสธคำร้องขอสำหรับเมาต์โซลูชันคือ การเปลี่ยนความเป็นสมาชิกกลุ่มของผู้ใช้กับหมายเลขที่น้อยกว่าแปด จากนั้นลองเมาต์อีกครั้ง ข้อความแสดงตัวอย่างต่อไปนี้ คือคุณสมบัติของปัญหาในกลุ่มนี้:

```
RPC:    Authentication  error;  why=Invalid  client  credential
```

การกำหนดปัญหา RPCSEC-GSS:

พิจารณาโซลูชันต่อไปนี้เมื่อคุณพบกับปัญหาเกี่ยวกับ RPCSEC-GSS

- ใช้คำสั่ง **klist** บนไคลเอ็นต์เพื่อตรวจสอบให้แน่ใจว่า คุณมีหนังสือรับรองปัจจุบันที่ถูกต้อง
- ตรวจสอบนาฬิกาบนไคลเอ็นต์ เซิร์ฟเวอร์ และ KDC ซิงโครไนซ์ซึ่งขอแนะนำว่า NTP หรือการติดตั้งที่เทียบเท่าถูกใช้เพื่อตรวจสอบให้มั่นใจถึงความสอดคล้องกันของเวลา ระหว่างขอบเขตของ Kerberos ทั้งหมด
- ตรวจสอบให้แน่ใจว่า เซิร์ฟเวอร์มีไฟล์ `keytab` และหลักการของโฮสต์ที่ถูกต้อง หากคำสั่งต่อไปนี้ล้มเหลว เซิร์ฟเวอร์จะไม่ทำงาน:

  ```
  kinit -kt 'tail -n 1 /etc/nfs/hostkey' 'head -n 1 /etc/nfs/hostkey'
  ```

- ตรวจสอบให้แน่ใจว่า **gssd** daemon กำลังรันและตอบกลับไคลเอ็นต์ และเซิร์ฟเวอร์ด้วยคำสั่งต่อไปนี้:

  ```
  rpcinfo -u localhost 400234
  ```

  หาก **gssd** daemon ไม่ตอบกลับ RPCSEC-GSS จะล้มเหลว หยุดทำงาน และการรีสตาร์ท **gssd** daemon อาจแก้ไข ปัญหานี้
- หากคุณขอรับข้อผิดพลาดในการเขียนพร้อมกับความเป็นส่วนตัวและ integrity ให้ตรวจสอบให้มั่นใจว่า คุณกำลังใช้โมดูลเคอร์เนล Integrity และความเป็นส่วนตัวไม่ได้รับการสนับสนุน โดยไม่มีโมดูลเคอร์เนล (โมดูลเคอร์เนลคือ โมดูลเคอร์เนล Kerberos `/usr/lib/drivers/nfs.ext` ซึ่งถูกติดตั้งพร้อมกับชุดไฟล์ `modcrypt.base` จากแพ็กส่วนขยาย)
- หากผู้ใช้ที่ระบุเฉพาะมีประสบกาณณ์กับการถูกปฏิเสธ denial เมื่อเข้าถึงข้อมูลที่ควรได้รับการเข้าถึง ให้ตรวจสอบว่า หลักการที่เกี่ยวข้องใน KDC ถูกซิงโครไนซ์ อย่างถูกต้องพร้อมกับชื่อบัญชีผู้ใช้ AIX ของผู้ใช้
- เรียกใช้งานบันทึกการทำงานของระบบ ข้อผิดพลาด RPCSEC-GSS ส่วนใหญ่จะถูกบันทึกการทำงานไว้ ข้อผิดพลาดมีสองส่วนคือ : ส่วนแรกคือโค้ดระบุความผิดพลาด GSS (โปรดดู RFC 2744 สำหรับรายละเอียด) และส่วนที่สองคือโค้ดระบุความผิดพลาด Kerberos

  หมายเหตุ: การเรียกใช้บันทึกการทำงานของระบบ อาจมีผลต่อผลการทำงานของระบบ ดังนั้น บันทึกการทำงานควรหยุดการทำงาน หลังจากที่การกำหนดปัญหาเสร็จสิ้น
  โค้ดระบุความผิดพลาดทั่วไปบางข้อ และโซลูชันอาจมีดังต่อไปนี้:

  `KRB5_CC_NOTFOUND`
  : หนังสือรับรอง Kerberos ที่ถูกต้องไม่สามารถพบได้ คำสั่ง **kinit** อาจแก้ไขปัญหานี้ได้

  `KRB5_KDC_UNREACH`
  : KDC ไม่สามารถเข้าถึงได้ ตรวจสอบให้แน่ใจว่า KDC ทำงานและไม่มีปัญหาเรื่องเน็ตเวิร์กระหว่างไคลเอ็นต์หรือเซิร์ฟเวอร์และ KDC

  `KRB5_KT_NOTFOUND`
  : รายการ `keytab` สำหรับหลักการของเซิร์ฟเวอร์ของคุณ ไม่พบ ใช้คำสั่ง **nfshostkey -l** เพื่อตรวจสอบว่า คุณกำลังใช้หลักการที่ถูกต้อง (ซึ่งควรเป็นชื่อโดเมน `nfs/`*<ที่ผ่านการรับรองโดยสมบูรณ์>*) และไฟล์ `keytab` ใช้ **klist -ke** เพื่อตรวจสอบเซิร์ฟเวอร์ไฟล์ `keytab` สำหรับรายการที่เหมาะสม

KRB5KRB_AP_ERR_TKT_NYV
: โดยส่วนใหญ่จะบ่งชี้ปัญหาเรื่องของนาฬิกา

KRB5KRB_AP_WRONG_PRINC และ KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN
: ข้อผิดพลาดทั้งสองนี้บ่งชี้ว่า หลักการที่ไคลเอ็นต์กำลังใช้สำหรับไคลเอ็นต์ ไม่ตรงกับหลักการของโฮสต์ของเซิร์ฟเวอร์

KRB5KRB_AP_WRONG_PRINC
: บ่งชี้ว่า ไคลเอ็นต์ดำเนินการเป็นผลสำเร็จในการแก้ไขชื่อโฮสต์ของเซิร์ฟเวอร์ไปเป็นหลักการที่มีอยู่ของรูปแบบชื่อโดเมน nfs/<*ที่ผ่านการรับรองโดยสมบูรณ์*> แต่หลักการของโฮสต์ของเซิร์ฟเวอร์ไม่ตรงกับหลักการนี้

KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN
: บ่งชี้ว่า ไคลเอ็นต์ไม่สามารถแก้ปัญหาชื่อโฮสต์ของเซิร์ฟเวอร์ไปเป็นหลักการที่มีอยู่ได้ ใช้คำสั่ง **nfshostkey -l** เพื่อตรวจสอบเซิร์ฟเวอร์เพื่อมั่นใจว่ามีหลักการที่ถูกต้อง หากเป็นเช่นนี้ ตารางการแม็พโฮสต์ของไคลเอ็นต์จะจำเป็นต้องถูกอัพเดต โปรดดูคำสั่ง **nfshostmap** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4* สำหรับรายละเอียดเพิ่มเติม

การกำหนดปัญหา EIM:

เมื่อแก้ปัญหา EIM ให้พิจารณาคำแนะนำต่อไปนี้

พิจารณาต่อไปนี้เมื่อคุณมีปัญหากับ EIM:

- ถ้าคำสั่ง **nfsrgyd** หรือ **chnfsim** ไม่สามารถเชื่อมต่อกับ EIM LDAP เซิร์ฟเวอร์ ให้แน่ใจว่ากระบวนการ **ibmslapd** รันอยู่บน EIM LDAP เซิร์ฟเวอร์โดยพิมพ์คำสั่งต่อไปนี้:

  ```
  ps -ef | grep ibmslapd
  ```

  ถ้ากระบวนการ **ibmslapd** ไม่ได้รันอยู่ พิมพ์คำสั่งต่อไปนี้เพื่อเปิดใช้งานมัน :

  ```
  /usr/sbin/ibmslapd
  ```

- ถ้าคำสั่ง **nfsrgyd** หรือ **chnfsim** สามารถเชื่อมต่อกับ EIM LDAP เซิร์ฟเวอร์ แต่ไม่สามารถทำการแม็พเอกลักษณ์ใดๆ ต้องแน่ใจว่ากระบวนการ **ibmslapd** ไม่ได้รันอยู่ในโหมดการตั้งค่าเท่านั้น นี่สามารถเกิดขึ้นได้ถ้าฐานข้อมูล **ldapdb2** ไม่ได้รันอยู่เมื่อ **ibmslapd** เซิร์ฟเวอร์ถูกสตาร์ท โดยปฏิบัติตามขั้นตอนเหล่านี้:
  1. ล็อกอินยัง EIM LDAP เซิร์ฟเวอร์เป็นผู้ใช้ root
  2. ดูที่ไฟล์ /var/ldap/ibmslapd.log ตรวจสอบว่าเมื่อใดที่กระบวนการ **ibmslapd** ถูกสตาร์ทครั้งล่าสุด ยังต้องตรวจสอบว่าเซิร์ฟเวอร์ถูกสตาร์ทในโหมดการตั้งค่าเท่านั้นหรือไม่ เนื่องจากมันไม่สามารถเชื่อมต่อกับฐานข้อมูล **ldapdb2**

  ถ้าเซิร์ฟเวอร์ไม่สามารถเชื่อมต่อกับฐานข้อมูล **ldapdb2** ฐานข้อมูลต้องถูกสตาร์ท ทำตามขั้นตอนเหล่านี้เพื่อสตาร์ทฐานข้อมูล **ldapdb2** :
  1. ล็อกอินยัง EIM LDAP เซิร์ฟเวอร์เป็นผู้ใช้ root
  2. พิมพ์คำสั่งต่อไปนี้เพื่อตรวจสอบว่ากระบวนการ **ibmslapd** แอ็คทีฟหรือไม่ :

     ```
     ps -ef | grep ibmslapd
     ```

     ถ้ามันแอ็คทีฟ ปิดการใช้งานมันโดยการรันคำสั่งต่อไปนี้ :

     ```
     kill ibmslapd pid
     ```

     โดยที่ *pid* เป็น ID ของกระบวนการที่ถูกส่งกลับมาจากคำสั่ง **ps -ef**

3. หลังจากกระบวนการ **ibmslapd** ถูกปิดใช้งาน สตาร์ทฐานข้อมูล **ldapdb2** :
   a. ล็อกอินยัง EIM LDAP เซิร์ฟเวอร์เป็นผู้ใช้ **ldapdb2**
   b. พิมพ์ `db2start`
4. หลังจากฐานข้อมูล **ldapdb2** ถูกสตาร์ท เปิดใช้งานกระบวนการ **ibmslapd** :
   a. ล็อกอินยัง EIM LDAP เซิร์ฟเวอร์เป็นผู้ใช้ root
   b. พิมพ์ `ibmslapd`

ปัญหาที่เกิดขึ้นหากส่วนขยายเคอร์เนล NFS ไม่ได้โหลด:

คำสั่ง NFS บางคำสั่งไม่ได้รันอย่างถูกต้องหากส่วนขยายเคอร์เนล NFS ไม่ได้ถูกโหลดไว้ บางคำสั่งที่มีการพึ่งพานี้คือ: **nfsstat, exportfs, mountd, nfsd** และ **biod**

เมื่อติดตั้ง NFS บนระบบ ส่วนขยายเคอร์เนลถูกวางอยู่ในไฟล์ `/usr/lib/drivers/nfs.ext` ไฟล์นี้ถูกโหลดเป็นส่วนขยายเคอร์เนล NFS เมื่อระบบถูกตั้งค่าไว้ สคริปต์ที่ทำให้ส่วนขยายเคอร์เนลนี้โหลดไฟล์ `/etc/rc.net` มีบางสิ่งที่ต้องทำในสคริปต์นี้ หนึ่งในนั้นคือการโหลดส่วนขยายเคอร์เนล NFS ซึ่งเป็นสิ่งสำคัญที่ต้องจดจำว่า ส่วนขยายเคอร์เนล **Transmission Control Protocol/Internet Protocol (TCP/IP)** และไฟล์ **nfs_kdes_null.ext** ควรถูกโหลดก่อนที่ส่วนขยายเคอร์เนล NFS จะถูกโหลด

หมายเหตุ: คำสั่ง **gfsinstall** ถูกใช้เพื่อโหลดส่วนขยายเคอร์เนล NFS ลงในเคอร์เนลเมื่อระบบสตาร์ท คำสั่งนี้สามารถรันได้มากกว่าหนึ่งต่อการเริ่มต้น ระบบและจะไม่เป็นสาเหตุของปัญหา ระบบถูกจัดส่งมาพร้อมกับคำสั่ง **gfsinstall** ที่ใช้ในไฟล์ `/etc/rc.net` และ `/etc/rc.nfs` ไม่มีการลบการเรียกเหล่านี้ใดๆ

ปัญหาที่เกิดขึ้นเมื่อไม่ได้ติดตั้ง kerberos ที่สนับสนุน:

ถ้าไม่ติดตั้ง kerberos ที่สนับสนุนไว้ **gssd** daemon จะไม่เริ่มต้น

ให้แน่ใจว่าชุดไฟล์ `krb5.client.rte` และ `modcrypt.base` ถูกติดตั้งไว้แล้ว ถ้าไม่ได้ติดตั้ง **gssd** daemon จะไม่รัน

ไอเท็มที่ตรวจสอบเมื่อรีจีสทรี daemon ไม่รัน:

nfsrgyd daemon จะไม่รันเมื่อโดเมน NFS เวอร์ชัน 4 ยังไม่ถูกกำหนดคอนฟิก

สำหรับข้อมูลเกี่ยวกับการกำหนดคอนฟิกโดเมน NFS เวอร์ชัน 4 โปรดดู “ไฟล์ /etc/nfs/local_domain” ในหน้า 552

ไฟล์ NFS

ไฟล์ NFS และคำอธิบายสามารถอ้างอิงได้ที่นี่

ไอเท็ม	คำอธิบาย
bootparams	แสดงไคลเอ็นต์ที่ไคลเอ็นต์ที่ไม่มีดิสก์สามารถใช้เพื่อการบูต
exports	แสดงรายการไดเร็กทอรีที่สามารถเอ็กซ์พอร์ตไปยังไคลเอ็นต์ NFS
filesystems	แสดงรายการระบบไฟล์ทั้งหมดที่ถูกเมาต์ ณ เวลาที่ระบบรีสตาร์ท
hostkey	ระบุหลักการของโฮสต์ Kerberos และตำแหน่งของไฟล์ keytab
local_domain	มีโดเมน NFS บนโลคัลของระบบ
เครือข่าย	มีข้อมูลเกี่ยวกับเน็ตเวิร์กบนอินเตอร์เน็ตเน็ตเวิร์ก
pcnfsd.conf	จัดเตรียมอ็อพชันคอนฟิกูเรชันสำหรับ **rpc.pcnfsd** daemon
princmap	แม็พชื่อโฮสต์กับหลักการ Kerberos เมื่อหลักการไม่ใช่ชื่อโดเมนที่ผ่านการรับรอง ของเซิร์ฟเวอร์
realm.map	ใช้โดย NFS registry daemon เพื่อแม็พกับหลักการของ Kerberos ขาเข้า
rpc	มีรายละเอียดฐานข้อมูลสำหรับโปรแกรม Remote Procedure Call (RPC)
security_default	มีความปลอดภัย NFS ที่เป็นค่าดีฟอลต์
xtab	แสดงไดเร็กทอรีที่ถูกเอ็กซ์พอร์ตในปัจจุบัน

คำสั่ง NFS

คำสั่ง NFS และคำอธิบายที่สามารถอ้างอิงได้ที่นี่

ไอเท็ม	คำอธิบาย
chnfs	เริ่มต้นจำนวนที่ระบุของ **biod** และ **nfsd** daemons
chnfsdom	เปลี่ยนโดเมน NFS บนโลคัล
chnfsim	เปลี่ยนการแม็พ NFS foreign identity
chnfssec	เปลี่ยน flavor การรักษาความปลอดภัยดีฟอลต์ที่ใช้โดยไคลเอ็นต์ NFS
chnfsrtd	เปลี่ยนการแม็พ realm-to-domain ของ NFS โลคัล
mknfs	ตั้งค่าระบบเพื่อรัน NFS และสตาร์ท NFS daemon
nfso	ตั้งค่าอ็อพชันเน็ตเวิร์ก NFS
automount	เมาต์ระบบไฟล์ NFS แบบอัตโนมัติ
chnfsexp	เปลี่ยนค่าแอ็ตทริบิวต์ของไดเร็กทอรี NFS ที่เอ็กซ์พอร์ต
chnfsmnt	เปลี่ยนค่าแอ็ตทริบิวต์ของไดเร็กทอรีที่เมาต์กับ NFS
exportfs	เอ็กซ์พอร์ตและยกเลิกเอ็กซ์พอร์ต ไดเร็กทอรีกับไคลเอ็นต์ NFS
lsnfsexp	แสดงคุณสมบัติของไดเร็กทอรีที่เอ็กซ์พอร์ตด้วย NFS
lsnfsmnt	แสดงคุณสมบัติของระบบ NFS ที่เมาต์
mknfsexp	เอ็กซ์พอร์ตไดเร็กทอรีโดยใช้ NFS
mknfsmnt	เมาต์ไดเร็กทอรีโดยใช้ NFS
nfshostkey	ตั้งค่าคีย์โฮสต์สำหรับเซิร์ฟเวอร์ NFS
nfs4cl	แสดงข้อมูลเกี่ยวกับระบบไฟล์ไคลเอ็นต์กำลังเข้าถึงโดยใช้ NFS เวอร์ชัน 4
nfs4smctl	การดูแลการเพิกถอนสถานะของ NFS เวอรัน 4
rmnfs	หยุดทำงาน NFS daemon
rmnfsexp	ลบไดเร็กทอรีที่เอ็กซ์พอร์ต NFS จากรายการเอ็กซ์พอร์ตของเซิร์ฟเวอร์
rmnfsmnt	ลบระบบไฟล์ที่เมาต์กับ NFS ออกจากรายการของการเมาต์ของไคลเอ็นต์

NFS daemons

NFS daemons และคำอธิบายสามารถอ้างอิงได้ที่นี่

การล็อก daemons

ไอเท็ม	คำอธิบาย
lockd	ประมวลผลคำร้องขอการล็อกผ่านแพ็กเกจ RPC
statd	จัดเตรียมฟังก์ชัน crash-and-recovery สำหรับการล็อกเซอร์วิสบน NFS

เน็ตเวิร์กเซอร์วิสสำหรับ daemon และยูทิลิตี

ไอเท็ม	คำอธิบาย
biod	ส่งคำร้องขอการอ่านและเขียนสำหรับไคลเอ็นต์ไปยังเซิร์ฟเวอร์
mountd	ตอบการร้องขอจากไคลเอ็นต์การเมาต์ระบบไฟล์
nfsrgyd	ดำเนินการแปลระหว่างหลักการแสดงความปลอดภัย ซึ่ง NFS เวอร์ชัน 4 มีลักษณะเป็นสตริง และสอดคล้องกับ ID ของระบบแบบตัวเลข นอกจากนี้ การแม็พข้อมูลที่มีลักษณะเฉพาะจาก NFS เวอร์ชัน 4 ภายนอกที่โดเมนถูกจัดการ
nfsd	สตาร์ท daemon ที่จัดการกับคำร้องขอไคลเอ็นต์สำหรับ การดำเนินการกับระบบไฟล์
nfsstat	แสดงข้อมูลเกี่ยวกับความสามารถในการรับการเรียกสำหรับ เครื่องเฉพาะ
on	เรียกใช้งานคำสั่งเกี่ยวกับเครื่องแบบรีโมต
pcnfsd	จัดการกับคำร้องขอเซอร์วิสจากไคลเอ็นต์ PC-NFS
portmap	แม็พหมายเลขโปรแกรม RPC กับหมายเลขอินเตอร์เน็ตพอร์ต
rexd	ยอมรับคำร้องขอเพื่อรันโปรแกรมจากเครื่องแบบรีโมต
rpcgen	สร้างโค้ดภาษา C เพื่อใช้โปรโตคอล RPC
rpcinfo	รายงานสถานะของเซิร์ฟเวอร์ RPC
rstatd	ส่งกลับสถิติผลการทำงานที่ได้จากเคอร์เนล
rup	แสดงสถานะของรีโมตโฮสต์บนโลคัลเน็ตเวิร์ก
rusers	รายงานรายการของผู้ใช้ที่ล็อกออนเข้าสู่เครื่องแบบรีโมต
rusersd	ตอบกลับเคียวรีจากคำสั่ง **rusers**
rwall	ส่งข้อความไปยังผู้ใช้ทั้งหมดบนเน็ตเวิร์ก
rwalld	จัดการการร้องขอจากคำสั่ง **rwall**
showmount	แสดงรายการของไคลเอ็นต์ทั้งหมดที่เมาต์ระบบไฟล์แบบรีโมต
spray	ส่งหมายเลขของแพ็กเก็ตที่ระบุไว้ไปยังโฮสต์
sprayd	รับแพ็กเก็ตที่ส่งโดยคำสั่ง **spray**

การรักษาความปลอดภัยให้กับ daemon การวางเน็ตเวิร์กและยูทิลิตี

ไอเท็ม	คำอธิบาย
chkey	เปลี่ยนคีย์การเข้ารหัสลบ
gssd	จัดเตรียม NFS กับการเข้าถึงเซอร์วิสการรักษาความปลอดภัยที่จัดเตรียมไว้โดย Network Authentication Services
keyenvoy	จัดเตรียมตัวกลางระหว่างกระบวนการผู้ใช้และคีย์เซิร์ฟเวอร์
keylogin	ถอดรหัสและเก็บคีย์ลับของผู้ใช้
keyserv	เก็บพับลิกและไพรเวตคีย์
mkkeyserv	สตาร์ท **keyserv** daemon และยกเลิคอมเมนต์รายการที่เหมาะสม ในไฟล์ `/etc/rc.nfs`
newkey	สร้างคีย์ใหม่ในไฟล์ `publickey`
rmkeyserv	หยุดทำงาน **keyserv** daemon และคอมเมนต์รายการสำหรับ **keyserv** daemon ในไฟล์ `/etc/rc.nfs`
ypupdated	อัพเดตข้อมูลในแม็พ Network Information Service (NIS)

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับความปลอดภัย NFS โปรดดู Network File System security ใน *การรักษาความปลอดภัย*

ส่วนสนับสนุนไคลเอ็นต์ Sun แบบ diskless

ไอเท็ม	คำอธิบาย
bootparamd	จัดเตรียมข้อมูลที่จำเป็นสำหรับการบูตให้กับไคลเอ็นต์ diskless

รูทีนย่อย NFS

รูทีนย่อย NFS ถูกอธิบายถึงในที่นี

ไอเท็ม	คำอธิบาย
cbc_crypt, des_setparity, or **ecb_crypt**	นำรูทีน Data Encryption Standard (DES) ไปใช้งาน

Server Message Block file system

Server Message Block Filesystem (SMBFS) อนุญาตให้เข้าถึงการแบ่งใช้บนเซิร์ฟเวอร์ SMB เป็นระบบไฟล์โลคัลบน AIX

ในระบบไฟล์นี้ ผู้ใช้สามารถสร้าง ลบ อ่าน เขียน และแก้ไข เวลาที่เข้าถึงไฟล์และไดเร็กทอรี เจ้าของหรือโหมดการเข้าถึงไฟล์และไดเร็กทอรีไม่สามารถเปลี่ยนแปลงได้

SMBFS สามารถใช้เพื่อเข้าถึงไฟล์บนเซิร์ฟเวอร์ SMB เซิร์ฟเวอร์ SMB คือเซิร์ฟเวอร์ที่รัน Samba หรือเซิร์ฟเวอร์ Windows XP, Windows NT หรือ Windows 2000 หรือเวิร์กสเตชัน แต่ละชนิดของเซิร์ฟเวอร์เหล่านี้ อนุญาตให้ไดเร็กทอรีถูกเอ็กซ์พอร์ตเป็นแบบแบ่งใช้ซึ่งการแบ่งใช้สามารถเมาต์บนระบบ AIX โดยใช้ SMBFS

การติดตั้ง SMBFS

เมื่อต้องการติดตั้ง SMBFS บนระบบ AIX ติดตั้งแพ็คเกจ `bos.cifs_fs`

เมื่อแพ็กเกจ `bos.cifs_fs` ติดตั้งแล้ว อุปกรณ์ `nsmb0` จะถูกสร้างขึ้น อุปกรณ์นี้อนุญาตให้คำสั่ง **mount** สร้างการเชื่อมต่อระหว่างเซิร์ฟเวอร์ SMB กับไคลเอ็นต์

การเมาต์ SMBFS

Server Message Block Filesystem (SMBFS) สามารถเมาต์ได้ด้วย หนึ่งในสองวิธี

ซึ่งสามารถดำเนินการผ่านคำสั่ง AIX **mount** ตัวอย่างเช่น:

```
mount -v cifs -n pezman/user1/pass1 -o uid=201,fmode=750 /home /mnt
```

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง **mount** และคำอธิบายของแฟล็กที่ใช้ โปรดดู คำสั่ง **mount** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 3*

คุณสามารถระบุอ็อพชันการเมาต์โดยใช้

```
-o
```

แฟล็ก อ็อพชันบรรทัดรับคำสั่งควรถูกคั่นด้วยเครื่องหมายจุลภาค ไม่ใช่เครื่องหมายจุลภาคและช่องว่าง อ็อพชันสำหรับระบบไฟล์คือ:

ไอเท็ม	คำอธิบาย
fmode	ตั้งค่าไฟล์หรือไดเร็กทอรีให้เป็นโหมดเลขฐานแปด ค่าดีฟอลต์คือ 755
uid	กำหนด UID ให้กับไฟล์ในระหว่างที่เมาต์ ค่าดีฟอลต์คือ **root**
gid	กำหนด GID ให้กับไฟล์ในระหว่างการเมาต์ ค่าดีฟอลต์คือ **system**
wrkgrp	กลุ่มงานที่เซิร์ฟเวอร์ SMB เป็นเจ้าของ
op	ตั้งค่า 1 หากใช้การล็อกที่มีโอกาสกระทำได้ ตั้งค่าเป็น 0 หากการล็อกที่มีโอกาสกระทำได้ไม่ได้ถูกใช้
opfs	ชื่อของระบบไฟล์ที่แคชเพื่อใช้สำหรับเก็บไฟล์แคช ล็อก
opsz	ขนาดของแคชไฟล์เดี่ยวที่ใช้สำหรับการล็อก ที่มีโอกาสกระทำได้
opfssz	ขนาดของระบบไฟล์ที่แคชแล้วซึ่งใช้ในการล็อก ที่มีโอกาสกระทำได้

คุณยังสามารถเมาต์ระบบไฟล์โดยใช้ยูทิลิตี SMIT นั่นคือ smit cifs_fs ซึ่งรันคำสั่ง **mount** หลังจากรวบรวมข้อมูลที่จำเป็นทั้งหมด

หากต้องการเมาต์ระบบไฟล์ SMBFS คุณจำเป็นต้องจัดเตรียมชื่อผู้ใช้ และรหัสผ่านเพื่อพิสูจน์ตัวตนเซิร์ฟเวอร์ ชื่อผู้ใช้และรหัสผ่านนี้ ถูกใช้เพื่อดำเนินการกับการดำเนินการกับระบบไฟล์ที่จำเป็นทั้งหมด บนเซิร์ฟเวอร์ ฟิลด์ **Password** ในพาเนล smit ไม่ไดทำเครื่องหมายว่าจำเป็นต้องมี หากฟิลด์รหัสผ่านไม่ได้ถูกรอกข้อมูลไว้ ไฟล์ cifscred ถูกค้นหาสำหรับการจับคู่หนังสือรับรอง สำหรับผู้ใช้หรือเซิร์ฟเวอร์ที่ถูกจัดเตรียมไว้ หากมีการจับคู่กัน รหัสผ่านที่เก็บไว้จากไฟล์ cifscred ถูกใช้ และผู้ใช้ที่ได้รับพร้อมต์สำหรับรหัสผ่านผ่านไปยังพร้อมต์รหัสผ่าน AIX มาตรฐาน วิธีนี้ ผู้ใช้สามารถจัดเตรียมรหัสผ่านได้โดยไม่ต้องสร้างความสามารถ ในการดู

หมายเหตุ: รหัสผ่านที่ใช้สำหรับการเมาต์ SMBFS สามารถมีความยาวได้สูงสุด 14 ตัวอักษรและรหัสผ่านสามารถมีอักขระพิเศษได้

เมื่อใดก็ตามที่ระบบเรียกใช้คำสั่ง เช่น อ่านไฟล์ ภายในจุดเมาต์ SMBFS คำร้องขอจะถูกส่งไปยังเซิร์ฟเวอร์ เพื่ออ่านไฟล์ ชื่อผู้ใช้และรหัสผ่านถูกส่งเป็นส่วนหนึ่งของคำร้องขอนี้ ดังนั้น เซิร์ฟเวอร์สามารถกำหนดผู้ใช้ที่มีสิทธิบนเซิร์ฟเวอร์ เพื่อดำเนินการอ่านบนไฟล์นั้น ดังนั้น จึงไม่จำกัดสิทธิ ที่วางอยู่กับเซิร์ฟเวอร์เป็นการดำเนินการบนไฟล์ ที่สามารถให้สิทธิในการใช้งาน

อย่างไรก็ตาม อ็อพชัน fmode ของคำสั่ง **mount** จัดเตรียมวิธีการสำหรับผู้ใช้ root บนระบบไคลเอ็นต์เพื่อควบคุมสิทธิในการเข้าถึงไฟล์ บนเซิร์ฟเวอร์ก่อนที่เคียวรีเซิร์ฟเวอร์ หากอ็อพชัน fmode ไม่ได้จัดเตรียมโดยผู้ใช้ ดีฟอลต์คือ 755 ตารางต่อไปนี้แสดงให้เห็นวิธีที่อ็อพชัน fmode ทำงานโดยใช้คำร้องขอการเขียน:

ตารางที่ 94. ห้ากรณีที่ผู้ใช้ถูกอนุญาต หรือปฏิเสธการเข้าถึงโดยอ้างอิงสิทธิการใช้งานที่กำหนดไว้

หมายเลขกรณี	ผู้ใช้ที่พิสูจน์ตัวตนกับเซิร์ฟเวอร์	ผู้ใช้บนฝั่งไคลเอ็นต์ที่ต้องการสิทธิในการเขียน	เมาต์เจ้าของ กลุ่ม และโหมด	เจ้าของ กลุ่ม และโหมดบนเซิร์ฟเวอร์	การเข้าถึงที่อนุญาต
กรณีที่ 1	user1	user2	user1, staff rwxr-xr-x	user1, staff rwxrwxr-x	ไม่

ตารางที่ 94. ห้ากรณีที่ผู้ใช้ถูกอนุญาต หรือปฏิเสธการเข้าถึงโดยอ้างอิงสิทธิการใช้งานที่กำหนดไว้ (ต่อ)

หมายเลขกรณี	ผู้ใช้ที่พิสูจน์ตัวตนกับเซิร์ฟเวอร์	ผู้ใช้บนฝั่งไคลเอ็นต์ที่ต้องการสิทธิในการเขียน	เมาต์เจ้าของ กลุ่ม และโหมด	เจ้าของ กลุ่ม และโหมดบนเซิร์ฟเวอร์	การเข้าถึงที่อนุญาต
กรณีที่ 2	user1	root	user1, staff rwxr-xr-x	user2, staff rwxr-xr-x	ไม่
กรณีที่ 3	user1	user1	user1, staff rwxr-xr-x	user2, staff rwxrwxr-x	yes
กรณีที่ 4	user1	user1	user, staff rwxr-xr-x	root, system rwx------	ไม่
กรณีที่ 5	user1	user1	user1, staff rwxr-xr-x	root, system rwxrwxrwx	yes

ในกรณีที่ 1 การเข้าถึงถูกนิยามไว้เนื่องจากเจ้าของ กลุ่ม และโหมดที่เมาต์บนไคลเอ็นต์ ไม่ได้อนุญาตให้เข้าถึงการเขียนกับ `user2`

ในกรณีที่ 2 การเข้าถึงถูกปฏิเสธ แม้ว่า `root` มีสิทธิเข้าถึงทุกสิ่งบนฝั่งไคลเอ็นต์ ผู้ใช้ที่พิสูจน์ตัวตนของเซิร์ฟเวอร์ `user1` ไม่ได้มีสิทธิในการเข้าถึงไฟล์ บนเซิร์ฟเวอร์

ในกรณีที่ 3 การเข้าถึงถูกให้สิทธิ เนื่องจาก `user1` เป็นเจ้าของการเมาต์ และ `user1` เป็นสมาชกของกลุ่ม `staff` บนเซิร์ฟเวอร์ ซึ่งมีสิทธิเข้าถึงไฟล์ บนเซิร์ฟเวอร์

ในกรณีที่ 4 การเข้าถึงถูกปฏิเสธ แม้ว่า `user1` เป็นเจ้าของการเมาต์ ไฟล์คือเจ้าของโดย `root` บนเซิร์ฟเวอร์ ซึ่งไม่มีการเข้าถึงโดยกลุ่มหรืออื่นๆ

ในกรณีที่ 5 การเข้าถึงถูกให้สิทธิเนื่องจาก `user1` เป็นเจ้าของการเมาต์ และ `user1` มีสิทธิเข้าถึงไฟล์ บนเซิร์ฟเวอร์ผ่านสิทธิในการใช้งานอื่นๆ

Notes:

1. บนระบบไฟล์ที่เมาท์ การดำเนินการคัดลอกไฟล์หนึ่งไปยังอีกไฟล์ ทำได้สำเร็จสำหรับไฟล์ขนาด 4 GB + 4096 ไบต์หรือน้อยกว่า สำหรับไฟล์ ที่เกินขนาดนี้ จะมีข้อความเตือนถูกพิมพื และ 4 GB + 4096 ไบต์ ของไฟล์ต้นฉบับถูกคัดลอกไปยังปลายทาง
2. บนระบบไฟล์ที่เมาท์ อักขระต่อไปนี้ไม่สามารถ ใช้ได้ในชื่อไฟล์: คีย์แบคสแลช {\}, คีย์สแลช {/}, โคลอน {:}, เครื่องหมายดอกจัน {*}, เครื่องหมายคำถาม {?}, คีย์น้อยกว่า {<}, คีย์ มากกว่า {>}, คีย์แถบแนวดิ่ง {|}

รหัสผ่านที่จัดเก็บไว้

SMBFS สามารถจัดเก็บหลักฐาน `server/user/password` ในไฟล์ `/etc/cifs_fs/cifscred` เพื่อให้สามารถดึงข้อมูล รหัสผ่านได้โดยอัตโนมัติเมื่อติดตั้ง SMBFS

สามารถเพิ่ม เปลี่ยน และลบหลักฐานออกจากไฟล์นี้ได้โดยใช้คำสั่ง **mkcifscred, chcifscred,** และ **rmcifscred** (ที่อยู่ในไฟล์ `/usr/sbin`) รหัสผ่านที่เพิ่มลงในไฟล์นี้จะมีการเข้ารหัส เมื่อพยายามติดตั้งโดยไม่ได้ระบุ รหัสผ่าน ระบบจะค้นหาไฟล์ `cifscred` เพื่อหาหลักฐาน ที่ตรงกัน หากมีรายการที่ตรงกัน จะใช้รหัสผ่านซึ่งจัดเก็บไว้จากไฟล์ `cifscred` มิฉะนั้น ผู้ใช้จะได้รับการพร้อมต์ให้ป้อนรหัสผ่านผ่านทางพร้อมต์รหัสผ่าน AIX มาตรฐาน

การสนับสนุนรหัสผ่านที่จัดเก็บไว้มีข้อจำกัดต่อไปนี้:

- เพื่อให้การดึงข้อมูลรหัสผ่านที่จัดเก็บไว้สามารถทำงานได้อย่างถูกต้อง ระเบียบ การตั้งชื่อเซิร์ฟเวอร์ต้องสอดคล้องกัน ตัวอย่างเช่น ถ้าเพิ่มหลักฐานที่มี IP แอดเดรส แทนชื่อโฮสต์หรือชื่อโดเมนแบบเต็มที่ถูกต้อง (FQDN) รหัสผ่าน จะถูกดึงข้อมูลเมื่อติดตั้งโดยใช้ IP แอดเดรสเท่านั้น
- เมธอดการดึงข้อมูลรหัสผ่านที่จัดเก็บไว้ไม่สนับสนุนการพิสูจน์ตัวตนรหัสผ่าน ข้อความปกติ หากเซิร์ฟเวอร์ต้องการรหัสผ่านข้อความปกติ การพิสูจน์ตัวตนจะล้มเหลว

การสนับสนุน /etc/filesystems

SMBFS สนับสนุน `/etc/filesystems` เพื่อยอมให้มีการเมาต์แบบอัตโนมัติเมื่อระบบเริ่มต้น

การสับสนุนสำหรับ `/etc/filesystems` ยังให้การเข้าถึงเซิร์ฟเวอร์ ผู้ใช้ รหัสผ่าน และข้อมูลอ็อพชันที่ถูกเก็บ เมื่อเมาต์ ใช้คำสั่ง **mkcifsmnt, chcifsmnt, rmcifsmnt** และ **lscifsmnt** (อยู่ใน `/usr/sbin`) เพื่อเพิ่ม แก้ไข ลบ และลิสต์ ตามลำดับ cifs stanzas ใน`/etc/filesystems` สิทธิต้องถูกเก็บในไฟล์ `cifscred`

การแก้ปัญหา SMBFS

ทำตามขั้นตอนเหล่านี้เมื่อพบกับปัญหา SMBFS

หากคำสั่ง **mount** หรือวิธีลัด `smit cifs_fs` ส่งคืนข้อผิดพลาด ให้พิจารณาต่อไปนี้:

1. ตรวจสอบว่า ชื่อผู้ใช้และรหัสผ่านถูกต้อง ชื่อผู้ใช้ และรหัสผ่านจำเป็นต้องได้รับอนุญาตให้เข้าถึงการแบ่งใช้บนเซิร์ฟเวอร์
2. ตรวจสอบว่าชื่อเซิร์ฟเวอร์ถูกต้อง หากชื่อเซิร์ฟเวอร์ถูกต้อง ให้ใช้ชื่อโฮสต์ที่ผ่านการรับรองในกรณีที่เซิร์ฟเวอร์ไม่ใช่ส่วนหนึ่งของ subnet เดียวกับไคลเอ็นต์ คุณยังสามารถลองใช้ IP แอดเดรสของเซิร์ฟเวอร์
3. ตรวจสอบว่า คำสั่ง `lsdev -L|grep nsmb` ส่งคืนชื่ออุปกรณ์ หากอุปกรณ์ `nsmb` ไม่พร้อมใช้งาน ไคลเอ็นต์ AIX จะไม่สามารถสร้างการเชื่อมต่อกับเซิร์ฟเวอร์ SMB ได้
4. ตรวจสอบว่า ชื่อที่แบ่งใช้ถูกต้อง หากการแบ่งใช้ไม่มีอยู่บนเซิร์ฟเวอร์ หรือไม่สามารถเข้าถึงได้ด้วยชื่อผู้ใช้และรหัสผ่านที่กำหนดไว้ เซิร์ฟเวอร์ SMB จะปฏิเสธคำร้องขอการเชื่อมต่อ
5. ใช้ event ID 525 เพื่อรวบรวมข้อมูลการติดตามระบบสำหรับ SMBFS
6. ตรวจสอบว่า เซิร์ฟเวอร์กำหนดคอนฟิกเพื่อยอมรับรหัสผ่าน NTLM, LM หรือข้อความล้วน เหล่านี้คือชนิดของรหัสผ่านที่เข้ารหัสซึ่งสนับสนุนโดย SMBFS
7. หากคุณต้องการพิสูจน์ตัวตนกับโดเมน ชื่อโดเมนต้องถูกระบุไว้ด้วยอ็อพชัน **wrkgrp** หากไม่มีอ็อพชันนี้ การพิสูจน์ตัวตนจะถูกจัดการแบบโลคัลโดยเซิร์ฟเวอร์

การสื่อสารอะซิงโครนัส

AIX จัดเตรียมประเภทต่อไปนี้ของไดรเวอร์ของอุปกรณ์แบบอะซิงโครนัส ยังเรียกว่าไดรเวอร์อุปกรณ์ tty :

- ไดรเวอร์สำหรับซีเรียลพอร์ตบนระนาบของระบบ
- ไดรเวอร์สำหรับซีเรียลพอร์ตที่เชื่อมต่อกับระบบผ่านอะแด็ปเตอร์
- ไดรเวอร์ของ Pseudo-tty

ไดรเวอร์ในหมวดหมู่แรกคืออะแด็ปเตอร์ PCI ซึ่งรวมถึงอะแด็ปเตอร์ 2-พอร์ต, 8-พอร์ต และ 128-พอร์ต

ในหมวดหมู่ที่สอง อะแด็ปเตอร์ PCI 8-พอร์ตและ 128-พอร์ตมีการ เรียกว่าอะแด็ปเตอร์แบบฉลาด เนื่องจากอะแด็ปเตอร์ใช้ตัวประมวลผล Intel 8086 เพื่อลดโหลดการประมวลผลอักขระจากโฮสต์ CPU อย่างมาก อะแด็ปเตอร์เหล่านี้จะถูกขับโดย 20 ms poller แทนฮาร์ดแวร์อินเตอร์รัปต์และให้คุณลักษณะของประสิทธิภาพที่เหมาะสมกับอุปกรณ์ซีเรียลและแอ็พพลิเคชันส่วนใหญ่ เมื่ออุปกรณ์เพิ่มเติมถูกเพิ่มเข้ากับระบบ เวิร์กโหลดของระบบจะสูงขึ้นเล็กน้อย โดยมีเหตุผลที่อะแด็ปเตอร์เหล่านี้สามารถสนับสนุนอุปกรณ์ซีเรียลจำนวนมาก มากกว่าที่ใช้ฮาร์ดแวร์อินเตอร์รัปต์ นอกจากนี้ เนื่องจากอะแด็ปเตอร์เหล่านี้ใช้ซอฟต์แวร์การปรับปรุงประสิทธิภาพที่มีสิทธิบัตร มันสามารถส่งและรับข้อมูลจำนวนมากได้รวดเร็วและมีประสิทธิภาพกว่าพอร์ตของระบบดั้งเดิม ตราบใดที่ข้อมูลถูกย้ายเป็นบล็อกขนาดใหญ่ สำหรับข้อมูลเพิ่มเติม ดูที่คำอธิบายในไฟล์ `/usr/include/sys/pse/README.pse`

หมายเหตุ: POWER5 *พอร์ตของระบบ* แบบรวมจะเหมือนกับซีเรียลพอร์ตยกเว้นพอร์ตของระบบมีให้ใช้งานเฉพาะสำหรับฟังก์ชันที่ได้รับการสนับสนุนโดยเฉพาะ โปรดอ้างถึง “ความแตกต่างของฟังก์ชันการทำงานระหว่างพอร์ตของระบบและซีเรียลพอร์ต” ในหน้า 618 สำหรับข้อมูลเพิ่มเติม

อย่างไรก็ตาม บางอุปกรณ์และแอ็พพลิเคชันต้องการเวลาแฝงที่น้อยมากในการประมวลผลตัวอักขระเดียว ดังนั้นคุณจะพบปัญหาเกี่ยวกับเวลาเมื่อเชื่อมต่อกับอะแด็ปเตอร์ที่ฉลาดเหล่านี้ เวลาแฝงของตัวอักขระ หรือการแสดงตัวอักขระอาจถูกกำหนดเป็นเวลาที่มันใช้เพื่อรับอักขระเดียวบนซีเรียลพอร์ต ส่งอักขระนั้นไปยังแอ็พพลิเคชัน จากนั้นแสดงอักขระนั้นกลับไปที่ซีเรียลพอร์ตเดิม

เนื่องจากมันใช้อินเตอร์รัปต์ที่มีลำดับความสำคัญสูงสุดบนระบบ (INTCLASS0) พอร์ตแบบ interrupt-driven จะให้ค่าเวลาแฝงในช่วง 0.10 ถึง 0.20 ms บนระบบที่ไม่ได้ทำงาน อะแด็ปเตอร์ PCI 8-พอร์ตให้ค่าเวลาแฝง เฉลี่ยประมาณ 10 ถึง 12 ms โดยที่แต่ละเวลาจะแตกต่าง โดยบวกหรือลบ 10 ms เนื่องจาก 20 ms poller อะแด็ปเตอร์ PCI 128 พอร์ต มี 20 ms poller เดียวกัน ซึ่งจะสื่อสารบนลิงก์การสื่อสารที่โพล กับ Remote Access Nodes (RANs) RANs ยอมให้ไดรเวอร์ที่โพลเพื่อควบคุมซีเรียลพอร์ต ค่าเวลาแฝงบนพอร์ตเหล่านี้เฉลี่ยประมาณ 30 ms แต่อาจเกิน 60 ms

ค่าเวลาแฝงบน 8-พอร์ต PCI และ 128-พอร์ต PCI อะแด็ปเตอร์สามารถถูกปรับแต่งสำหรับแอ็พพลิเคชันพิเศษโดยใช้พารามิเตอร์ "event delay" (EDELAY) สำหรับความตอบสนองสูงสุดเมื่อได้รับอักขระเดียว ลดค่าของพารามิเตอร์ EDELAY นี่จะลดเวลาที่ต้องการเพื่อให้ได้อักขระเดียวจากซีเรียลพอร์ตไปยังแอ็พพลิเคชัน แต่สามารถส่งผลในการลดประมาณงานและประสิทธิภาพของระบบโดยรวมเมื่อหลายอักขระถูกได้รับใน burst

2-พอร์ต PCI EIA-232 อะแด็ปเตอร์เป็นการสื่อสารซีเรียลแบบอะซิงโครนัสอะแด็ปเตอร์โดยมีพื้นฐานบน Exar 17D152 Universal PC Dual UART 2-พอร์ต อะแด็ปเตอร์สนับสนุน ตัวเชื่อมต่อ DB-9 2 ตัว และให้การเชื่อมต่อกับอุปกรณ์อะซิงโครนัส EIA-232 เช่น โมเด็ม และเทอร์มินัล tty

บนแพล็ตฟอร์ม IBM eServer™ p5 จะไม่มีพอร์ตของระบบดังเดิมที่พร้อมใช้งานกับ AIX แม้ว่าอินเตอร์เฟสของเทอร์มินัลเสมือนถูกปรับปรุงเพื่อสนับสนุนซีเรียลพอร์ตแบบฟิสิคัลที่อยู่บน FSP ผ่าน hypervisor อินเตอร์เฟสนี้จะสนับสนุนชุดที่ระบุของอุปกรณ์แบบซีเรียล และไม่เหมาะที่จะใช้แทนซีเรียลพอร์ตแบบฟิสิคัลที่ใช้งานทั่วไป 2-พอร์ตอะแด็ปเตอร์ทำงานเหมือนกับซีเรียลพอร์ตดั้งเดิม ไดรเวอร์อุปกรณ์ของอะแด็ปเตอร์เป็นแบบ interrupt-driven และ สนับสนุนระดับการทริกเกอร์ FIFO transit และได้รับที่สามารถโปรแกรมได้ ดังนั้น มันเป็น PCI อะแด็ปเตอร์ ไดรเวอร์อุปกรณ์ที่สนับสนุน EEH, hot-plug และเคียวรี VPD 2-พอร์ตอะแด็ปเตอร์ไม่สนับสนุนคุณลักษณะพอร์ตของระบบดังเดิมสำหรับเมื่อใดที่เทอร์มินัลเสมือนถูกใช้ เช่นระหว่างบูต ติดตั้ง และสนับสนุน KDB

ไดรเวอร์ Pseudo-tty ถูกใช้เมื่อเข้าถึงระบบผ่านเน็ตเวิร์กโดยใช้คำสั่ง **rlogin** หรือ **telnet** หรือเมื่อเข้าถึงระบบโดยใช้ระบบหน้าต่างบนมอนิเตอร์แบบกราฟิก ไดรเวอร์ pseudo-tty จัดเตรียมการรันเวลาแฝงของแอ็พพลิเคชันแบบ character-based เช่น เท็กซ์เอดิเตอร์ **vi** บนสื่อการสื่อสารที่ไม่ใช่ซีเรียล สิ่งที่สำคัญที่ต้องจำเกี่ยวกับไดรเวอร์ pseudo-tty คือมันไม่สมมาตรา The slave end provides a POSIX-standard-compliant interface to earlier applications. ด้านที่เป็นหลักจะถูกควบคุมโดย entity เช่น **rlogin** หรือ **telnet** daemon หรือ X-windows ซึ่งจัดเตรียมอีมูเลชันของอุปกรณ์เทอร์มินัลแบบซีเรียล กับไดรเวอร์ pseudo-tty AIX สามารถสนับสนุนอุปกรณ์ pseudo-tty จำนวนมากอย่างมีประสิทธิภาพ

ความเร็วของสายที่ไม่ใช่ POSIX

อินเตอร์เฟสไปยังอุกปรณ์ซีเรียลที่ถูกระบุโดย POSIX และมาตรฐาน UNIX ที่ตามมา เช่น X/OPEN ขึ้นอยู่กับโครงสร้างข้อมูล **termios** ที่ถูกกำหนดใน `/usr/include/termios.h` โชคร้าย ที่โครงสร้างข้อมูลนั้นไม่สามารถถูกใช้เพื่อระบุความเร็วสายเกินกว่า 38,400 บิตต่อวินาที ฮาร์ดแวร์ซีเรียลส่วนใหญ่ที่ถูกใช้ในปัจจุบันสามารถสนับสนุนความเร็วมากถึง 230,000 bps เมื่อต้องการใช้ความเร็วสาย สูงสขึ้นบน AIX ความเร็วที่ต้องการควรถูกระบุเมื่อกำหนดค่าพอร์ตโดยใช้ SMIT ถ้าฮาร์ดแวร์ซีเรียลพอร์ต (UART) สามารถสนับสนุนความเร็วของสายที่คุณระบุ พอร์ตจะสามารถถูกตั้งค่า

ได้รับแอ็ตทริบิวต์ ioctls โดยใช้โครงสร้าง **termio** หรือ **termios** จะรายงานความเร็วของสายเป็น 50 bps ความเร็วของสายแบบ non-POSIX จะถูกใช้โดยพอร์ตจนกว่าจะถูกเปลี่ยน ดังนั้นแอ็พพลิเคชันที่ใช้ตั้งแอ็ตทริบิวต์ ioctls กับโครงสร้าง **termio** และ **termios** ไม่ควรแก้ไขแฟล็ก CBAUD ยกเว้นมันต้องการเปลี่ยนความเร็วของสายจริงๆ ถ้าฮาร์ดแวร์ซีเรียลพอร์ต (UART) ไม่สามารถสนับสนุนความเร็วของสายที่ต้องการ การตั้งค่าพอร์ตจะล้มเหลวและได้รับข้อผิดพลาดกลับมา

หมายเหตุ: 8- และ 128-พอร์ต PCI อะแด็ปเตอร์สนับสนุนความเร็วของสาย non-POSIX ที่ 115,200 และ 230,000 bps เท่านั้น 128-พอร์ต PCI อะแด็ปเตอร์มีข้อจำกัดเพิ่มเติมของการรวมแบนด์วิดธ์ 2.5 Mbps (ด้วยสายเคเบิลแบบ 8-เส้น) ซึ่งสามารถถูกใช้โดย 11 อุปกรณ์แต่ละตัวสามารถที่ถ่ายโอนที่ 230,000 bps ข้อจำกัดนี้อยู่บนสายที่เชื่อมอะแด็ปเตอร์กับ RAN ดังนั้นอะแด็ปเตอร์เดียวสามารถถูกใช้โดย 22 อุปกรณ์

อะซิงโครนัสอะแด็ปเตอร์

ผลิตภัณฑ์การสื่อสารแบบอะซิงโครนัสมีข้อได้เปรียบ คือ ราคาถูก มีผู้ใช้ได้หลายคน มีเทอร์มินัลและอุปกรณ์การสื่อสารที่มีประสิทธิภาพปานกลางถึงสูง

AIX ยอมให้ผู้ใช้หลายคนเข้าถึงรีซอร์สของระบบและแอ็พพลิเคชัน แต่ละผู้ใช้ต้องเชื่อมต่อผ่านเทอริมินัลเซสชัน การเชื่อมต่อสามารถเป็นแบบโลคัลหรือรีโมตผ่านซีเรียลพอร์ต

แต่ละหน่วยของระบบมีอย่างน้อยหนึ่งซีเรียลพอร์ตแบบมาตรฐาน (บางระบบมี 3 ซีเรียลพอร์ต) พอร์ตเหล่านี้สามารถสนับสนุนการสื่อสารและอุปกรณ์เชื่อมต่อแบบอะซิงโครนัส

พอร์ตแบบอะซิงโครนัสยอมให้การเชื่อต่ออุปกรณ์แบบอะซิงโครนัสที่มีมาตรฐาน EIA 232, EIA 422 หรือ RS-423 เช่น :

- อะซิงโครนัสโมเด็ม
- เครื่องสแกนบาร์โค้ด
- เครื่องพิมพ์แบบกราฟิกและแบบตัวอักษร
- คีย์บอร์ดและหน้าจอของเทอร์มินัล
- เครื่องคอมพิวเตอร์ส่วนบุคคล
- พล็อตเตอร์และเครื่องพิมพ์
- เทอร์มินัล Point-of-sale
- เซ็นเซอร์และอุปกรณ์ควบคุม
- เครื่องสแกนข้อความ
- นาฬิกาจับเวลา

อ็อพชันของการสื่อสารแบบอะซิงโครนัส

ความสามารถอะซิงโครนัสที่ขยายสามารถมีการเพิ่มในหน่วยของ ระบบด้วยอะแด็ปเตอร์โดยใช้บัส Peripheral Component Interconnect (PCI)

มีหลายปัจจัยที่อาจมีผลกับชนิดของการเชื่อมต่อแบบอะซิงโครนัสที่คุณเลือก ตารางต่อไปนี้จะสรุปผลิตภัณฑ์เหล่านี้

ตารางที่ 95. ผลิตภัณฑ์การสื่อสารแบบอะซิงโครนัส

สิ่งที่แนบแบบอะซิงโครนัส	ใช้ตัวประมวลผล POWER®	ใช้ Itanium	ชนิดของบัส	โค้ดคุณลักษณะหรือชนิดของเครื่อง (โมเดล)	อัตราของข้อมูลสูงสุดต่อพอร์ต (kbps)	คุณลักษณะที่ไม่มีการโต้ตอบ
ซีเรียลพอร์ตแบบมาตรฐาน	X	X	ระนาบของระบบ	n/a	สามารถเลือกได้โดยขึ้นอยู่กับความเร็วของสัญญาณนาฬิกาของตัวสร้าง baud rate ของ universal asynchronous receiver and transmitter (UART)	คุณลักษณะมาตรฐาน
232 RAN	X	X		8130	57.6	ความสามารถแบบรีโมต
232 RAN ที่ได้รับการปรับปรุง	X	X		8137	230	ความสามารถแบบรีโมต
16พอร์ต RAN EIA 422	X	X		8138	230	ความสามารถแบบรีโมต
คอนโทรลเลอร์ 128-พอร์ต	X			8128	230	ประสิทธิภาพ สูงกว่าจำนวนอุปกรณ์
คอนโทรลเลอร์ 128-พอร์ต	X			2933	230	ประสิทธิภาพ สูงกว่าจำนวนอุปกรณ์
คอนโทรลเลอร์ 128-พอร์ต	X	X	PCI	2944	230	ประสิทธิภาพ สูงกว่าจำนวนอุปกรณ์

หมายเหตุ: Rack Mount RAN FC คือ 8136

หมายเหตุ: อัตราของข้อมูลสูงสุดต่อพอร์ตถูกจำกัดโดยแบนด์วิดท์ของสาย (1.2 Mbps สำหรับ RAN มาตรฐาน หรือ 2.4 Mbps สำหรับ RAN ที่ปรับปรุง)

คุณลักษณะแรกในตารางนี้แทนซีเรียลพอร์ตแบบเชื่อมกับระนาบ ที่เป็นมาตรฐานกับทุกหน่วยของระบบ คุณลักษณะต่อไปคืออะแด็ปเตอร์ ระบบย่อย 128-พอร์ตอะซิงโครนัส จะรวม remote asynchronous nodes (RANs) ที่เชื่อมต่อกับมัน

พอร์ตอะซิงโครนัสที่เชื่อมต่อกับระนาบ

หน่วยของระบบส่วนใหญ่มีสองซีเรียลพอร์ตแบบอะซิงโครนัสแบบรวม (มาตรฐาน) EIA 232 อุปกรณ์ EIA 232 อะซิงโครนัสแบบซีเรียลสามารถเชื่อมต่อโดยตรงกับซีเรียลพอร์ตแบบมาตรฐานโดยใช้สายซีเรียลแบบมาตรฐานโดยตัวเชื่อมต่อแบบ D-shell 9-pin

ระบบหลายโพรเซสเซอร์บางระบบมีซีเรียลพอร์ตพอร์ตที่สามที่ใช้สำหรับสื่อสารกับศูนย์บริการแบบรีโมต

หมายเหตุ: ระบบ Itanium-based มีซีเรียลพอร์ตแบบรวม หนึ่งหรือสองพอร์ต โมเดลของเวิร์กสเตชันเริ่มต้นจะมีหนึ่งพอร์ต ขณะที่โมเดลของเซิร์ฟเวอร์คลาสเริ่มต้นจะมี 2 พอร์ต

พอร์ตอะซิงโครนัสที่มีพอร์ตต่ออยู่

แต่ละอะแด็ปเตอร์ต้องการบัสสล็อตและสามารถถูกใช้ในระบบที่สนับสนุนชนิดของบัสที่ต้องการ

128-พอร์ต ISA 8-พอร์ตอะแด็ปเตอร์ และ PCI 8-พอร์ต อะแด็ปเตอร์เป็นอะแด็ปเตอร์ที่มีความฉลาดที่ช่วยลดโหลดของตัวประมวลผลหลักของระบบ

EIA 232 เป็นมาตรฐานการสื่อสารทั่วไป แต่ EIA 422A (ถูกใช้เมื่อต้องการใช้สายเคเบิลที่ยาวกว่า) ก็ได้รับการสนับสนุน การใช้งาน EIA 422A จะไม่รวมความสามารถในการตรวจจับสถานะของอุปกรณ์ หรือ สัญญาณควบคุมโมเด็ม RS 232

หมายเหตุ: แพล็ตพอร์ม Itanium-based สนับสนุนเฉพาะ 8- และ 128-พอร์ต PCI อะแด็ปเตอร์

พอร์ตอะซิงโครนัสที่ถูกเชื่อมต่อกับโหนด

อะแด็ปเตอร์แบบ 128-พอร์ต จะพร้อมใช้งานสำหรับ Micro Channel, ISA หรือ PCI บัส สามารถใช้เชื่อมต่อกับ หนึ่งถึงแปด remote asynchronous nodes (RANs)

แต่ละ RAN มีพอร์ตอะซิงโครนัส 16 พอร์ตสำหรับเชื่อมต่อกับอุปกรณ์และมีหน่วยจ่ายกำลังไฟที่แยกต่างหาก มากถึง 4 RAN สามารถเชื่อมต่อแบบ daisy-chain จากแต่ละ 2 การเชื่อมต่อบน 128-พอร์ต อะแด็ปเตอร์การ์ด RANs สามารถสนับสนุนอุปกรณ์ EIA 232 16 อุปกรณ์ EIA 422 16 อุปกรณ์ ตัวควบคุม 128-พอร์ต เป็นอะแด็ปเตอร์แบบฉลาดที่เพิ่มจำนวนของเซสชันอะซิงโครนัสที่ระดับของการใช้งาน CPU นั้นๆ

ต่อไปนี้เป็นคุณลักษณะเพิ่มเติมของคุณลักษณะ 128-พอร์ต :

- RANs สามารถอยู่ห่างออกไป 300 เมตรจากตัวประมวลผลของระบบ โดยใช้สายเคเบิลแบบซีลด์ที่มี 8 เส้น โดยยังคงอัตราของประสิทธิภาพได้อย่างเต็มที่
- ระยะทางสามารถขยายออกไปถึง 1200 เมตรโดยลดอัตราของข้อมูลระหว่าง RANs และตัวประมวลผลของระบบ
- RANs สามารถอยู่ห่างจากตัวประมวลผลของระบบโดยใช้โมเด็มอะซิงโครนัส EIA 232 และ EIA 422 แต่ละชุดของ RAN daisy chain 4 ชุด จะมีเพียงหนึ่งคู่ของโมเด็มที่บางจุดของ chain
- ประสิทธิภาพของระบบจะถูกปรับปรุงโดยการลกโหลดการประมวลผลอักขระของ tty จากตัวประมวลผลของระบบ

การพิจารณาเลือกผลิตภัณฑ์

ผลิตภัณฑ์อะซิงโครนัสที่เหมาะสมโดยมากขึ้นอยู่กับสถานการณ์นั้นๆ

คำถามต่อไปนี้จะช่วยคุณเลือกว่าผลิตภัณฑ์ใดที่คุณต้องการติดตั้ง

ความสามารถในการขยาย

ต้องการพอร์ตอะซิงโครนัสกี่พอร์ต ?
จะต้องการใช้พอร์ตที่พอร์ตในอนาคต ?

โทโปโลยี

อุปกรณ์จะอยู่ที่ตึกอื่นหรือตำแหน่งที่อยู่ห่างไกลหรือไม่ ?
จะทำการดูแลระบบ/เน็ตเวิร์กจากที่ไหน ?
จะมี HACMP™ คลัสเตอร์ หรือไม่ ?
ชนิดของสายเคเบิลที่ต้องการ หรือมีอยู่แล้ว?

ประสิทธิภาพ

แอ็พพลิเคชันของคุณใช้งาน CPU มากหรือไม่ ?
ชนิดของอุปกรณ์ที่จะต่อพ่วง ?
แบนด์วิดธ์อะซิงโครนัสที่ต้องการสำหรับอุปกรณ์ทั้งหมด ?

ตารางที่ 96. ความต้องการแบนด์วิดธ์ของอุปกรณ์ที่สัมพันธ์กัน

ความต้องการต่ำ	ความต้องการปานกลาง	ความต้องการสูง
ASCII เทอร์มินัล Point-of-sale เทอร์มินัล อะซิงโครนัสโมเด็ม	เครื่องพิมพ์ FAX/โมเด็มความเร็วต่ำ เครื่องสแกนบาร์โค้ด	X-terminal แบบซีเรียล FAX/โมเด็มความเร็วสูง เครื่องพิมพ์ความเร็วสูง แอ็พพลิเคชันการถ่ายโอนไฟล์

ข้อกำหนดของอินเตอร์เฟสของอุปกรณ์

อินเตอร์เฟสอะซิงโครนัสอะไรที่ต้องการ ตัวอย่างเช่น EIA 232, EIA 422A, EIA 423?
อุปกรณ์หรือแอ็พพลิเคชันต้องการอินเตอร์เฟส EIA 232 แบบเต็มหรือไม่ ?

การรักษาความปลอดภัย

ต้องการ system assurance kernel (SAK) หรือไม่ ? (พอร์ตแบบติดกับระนาบเท่านั้น)

ตารางต่อไปนี้แสดงคุณลักษณะของผลิตภัณฑ์แบบละเอียด

ตารางที่ 97. คุณลักษณะของผลิตภัณฑ์ที่เชื่อมต่อแบบอะซิงโครนัส

คุณสมบัติ	ซีเรียลพอร์ตแบบดั้งเดิม	2-พอร์ต PCI	8-พอร์ต PCI	128-พอร์ต PCI พร้อมกับ RAN
จำนวนของพอร์ตแบบอะซิงโครนัสต่ออะแด็ปเตอร์	n/a	2	8	128
จำนวนของอะแด็ปเตอร์สูงสุด	n/a	ไม่จำกัด	20	20
จำนวนของพอร์ตแบบอะซิงโครนัสสูงสุด	2 หรือ 3	2	160	2560
จำนวนของพอร์ตแบบอะซิงโครนัสต่อ RAN	n/a	n/a	n/a	16
จำนวนของ RAN สูงสุด	n/a	n/a	n/a	160
ความเร็วสูงสุด (KBits/วินาที)	สามารถเลือกได้โดยขึ้นอยู่กับความเร็วของสัญญาณนาฬิกาของตัวสร้าง baud rate ของ UART	230	230	230
วิธีการเชื่อมต่อ	planar	โดยตรง	โดยตรง	โหนด
การสนับสนุนอินเตอร์เฟสอะซิงโครนัสแบบไฟฟ้า	EIA 232	EIA 232	EIA 232 EIA 422A	EIA 232 EIA 422
มาตรฐานตัวเชื่อมต่อ	DB9	DB9	DB25M	RJ-45 (10-ขา หรือ 8-ขา)
อ็อพชันของสายเคเบิล DB25	n/a	n/a	n/a	RJ-45-DB25
อ็อพชันแบบยึด Rack	n/a	n/a	n/a	yes
แหล่งจ่ายไฟ	n/a	n/a	n/a	external
สัญญาณที่ได้รับการสนับสนุน (EIA 232)	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI

แอ็พพลิเคชันอะซิงโครนัสอะแด็ปเตอร์

การเสนอของแต่ละผลิตภัณฑ์ถูกทำให้เป็นคุณลักษณะโดยการแสดงสถานการณ์จำลองสำหรับความแข็งแกร่งของมัน อะแด็ปเตอร์ในหัวข้อนี้จะถูกลิสต์ตามด้วยข้อกำหนดของมันเพื่อที่คุณจะสามารถเลือกสำหรับแต่ละสถานการณ์จำลองที่ระบุ

ไอเท็ม	คำอธิบาย
2-พอร์ต PCI Bus EIA 232	• มี PCI สล็อต • มีมากถึง 2 พอร์ตต่ออะแด็ปเตอร์ • ต้องการพอร์ต EIA 232 ทั้งหมด • ความเร็วอะซิงโครนัสถึง 230 Kbps

ไอเท็ม	คำอธิบาย
8-พอร์ต PCI Bus EIA 232/EIA 422	• มี PCI สล็อต • ต้องการน้อยกว่า 8 พอร์ตโดยมีการขยายน้อยหรือไม่มี • ต้องการพอร์ต EIA 232 ทั้งหมด EIA 422 ทั้งหมด หรือผสมกันระหว่างพอร์ต EIA 232 และ EIA 422 • ลดโหลดการอินเตอร์รัปต์ตัวอักษร และการประมวลผลเทอร์มินัล I/O จาก CPU หลัก • ความเร็วอะซิงโครนัสถึง 230 Kbps • ประสิทธิภาพสูงสุดสำหรับโมเด็มความเร็วสูง (33.6 Kbps) พร้อมกับการบีบอัดข้อมูล
128-พอร์ต Adapter (PCI)	• มี Micro Channel, ISA หรือ PCI บัสสล็อตให้ใช้งาน (สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ Micro Channel หรือ ISA ดูที่ “128-พอร์ต อะแด็ปเตอร์ (Micro Channel, ISA)” ในหน้า 711) • ตอนนี้ 16 พอร์ตที่ขยายได้มากถึง 128 พอร์ตโดยไม่มีสล็อตเพิ่มเติม • ระยะของเทอร์มินัลส่วนใหญ่จะอยู่ที่ 90 เมตร (300 ฟุต) จากระบบ ที่อัตราของข้อมูลสูงสุด 230 Kbps • มีวางแผนสำหรับเทอร์มินัล: ใกล้ๆหรือในสถานที่ ระยะไกลหรือในสถานที่ และรีโมต • ต้องการปริมาณงานของอะซิงโครนัสสูงโดยต้องการตัวประมวลผลต่ำ • ต้องการความสามารถเชื่อมเครื่องพิมพ์กับเทอร์มินัล • ต้องการเชื่อมต่อสถานที่แบบรีโมตผ่าน fiber-optic หรืออะซิงโครนัสโมเด็ม

สถานการณ์จำลองสำหรับโซลูชันแบบอะซิงโครนัส

สถานการณ์จำลองของลูกคThe customer scenarios that follow were solved with an 8-port PCI and a 128-port asynchronous controller.

ไอเท็ม	คำอธิบาย
สำนักงานอสังหาริมทรัพย์	• ความง่ายและต้นทุนมีความสำคัญสูงสุด • ระบบปฏิบัติการและเซิร์ฟเวอร์ • อุปกรณ์ 6 ถึง 10 ตัวเชื่อมอยู่กับเซิร์ฟเวอร์เพื่อเข้าถึงฐานข้อมูล • หนึ่งสล็อตพร้อมใช้งานสำหรับการสื่อสารแบบอะซิงโครนัส • อุปกรณ์อยู่ห่างจากเซิร์ฟเวอร์น้อยกว่า 61 เมตร (200 ฟุต) **โซลูชัน:** 8-พอร์ต PCI

ไอเท็ม	คำอธิบาย
ร้านขายปลีก	• ต้นทุนต่อหน่วยมีความสำคัญสูงสุด • ระบบปฏิบัติการและเซิร์ฟเวอร์ • เทอร์มินัลแบบ ASCII 20 ตัวหรือมากกว่า ตัวอย่างเช่น เครื่องคิดเงิน • หนึ่งสล็อตพร้อมใช้งานสำหรับการสื่อสารแบบอะซิงโครนัส • มีการวางแผนในการขยายเทอร์มินัลเพิ่มเติมในอนาคต **โซลูชัน:** ตัวควบคุมแบบอะซิงโครนัส 128-พอร์ต พร้อมกับ 2 RAN จะมีการเพิ่ม RAN ในอนาคต

การพิจารณาเกี่ยวกับโทโปโลยี

อะแด็ปเตอร์ตระกูลอะซิงโครนัสให้ทางเลือกมากมายโดยพิจารณาถึงโทโปโลยีของระยะทาง

ความยาวของสายเคเบิลสูงสุดจากระนาบและอะแด็ปเตอร์ที่ต่ออยู่โดยตรงโดยทั่วไปจะเป็นความยาวระหว่างพอร์ตและอุปกรณ์อะซิงโครนัส ที่ทำงานที่อัตราของข้อมูลที่ถูกระบุสูงสุด 128-พอร์ต อะแด็ปเตอร์ถูกวัดจากอะแด็ปเตอร์การ์ดถึง RAN แบบ daisy-chained ที่ต่ออยู่กับมัน โดยใช้ 128-พอร์ต สามารถมีความยาวที่ไม่จำกัดโดยการใช้โมเด็มแบบซิงโครนัส EIA 422 เพื่อต่อ RAN กับอะแด็ปเตอร์

การใช้สายเคเบิลที่เหมาะสมมีความสำคัญมากและมีความเป็นหนึ่งเดียวสำหรับแต่ละสภาวะแวดล้อม

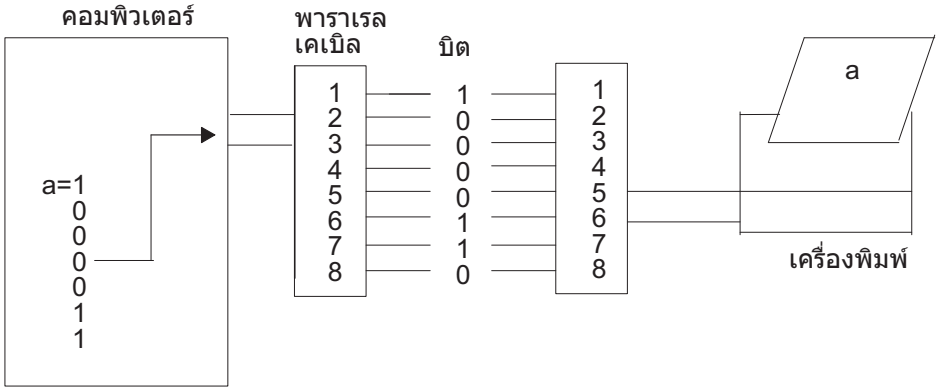
การสื่อสารแบบซีเรียล

มาตรฐานการสื่อสารแบบอะซิงโครนัส ฮาร์ดแวร์ คำศัพท์ และแนวคิด จะถูกอธิบายที่นี่

ซีเรียลพอร์ตถูกใช้เพื่อการเชื่อมต่อแบบฟิสิคัลอุปกรณ์อะซิงโครนัสกับคอมพิวเตอร์ มันจะอยู่ที่ด้านหลังของระบบ ทั้งเป็นแบบถูกรวมหรือใช้อะแด็ปเตอร์แบบหลายพอร์ต เช่นอะแด็ปเตอร์อะซิงโครนัสแบบ 2-พอร์ต 8-พอร์ต 16-พอร์ต และ 128-พอร์ต

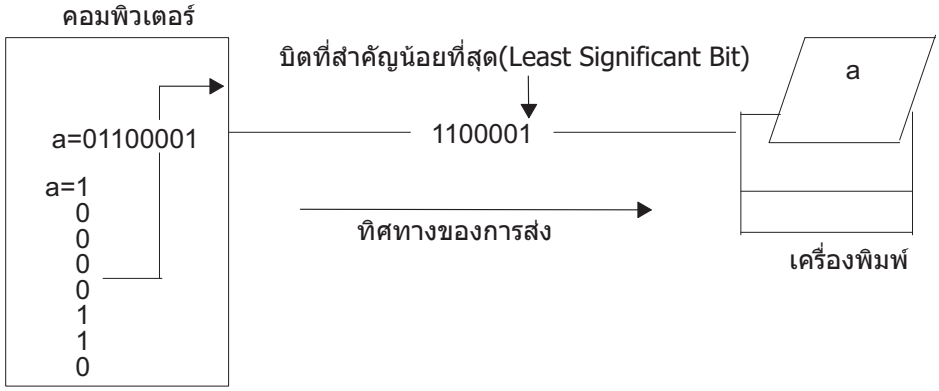
หมายเหตุ: พอร์ตแบบรวมของระบบ POWER5 จะไม่ใช่ซีเรียลพอร์ตแบบทำงานเต็มรูปแบบที่ใช้งานทั่วไป โปรดดู “ความแตกต่างของฟังก์ชันการทำงานระหว่างพอร์ตของระบบและซีเรียลพอร์ต” ในหน้า 618 สำหรับข้อมูลเพิ่มเติม

เพื่อเข้าใจการทำงานของซีเรียลพอร์ต จำเป็นต้องศึกษาการสื่อสารแบบขนานก่อน พอร์ตแบบขนานมาตรฐานจะใช้แปดขา หรือ สาย เพื่อส่งบิตของข้อมูลพร้อมกัน ซึ่งจะรวมเป็นอักขระเดียว ภาพต่อไปนี้จะแสดงการส่งแบบขนานของอักษร a



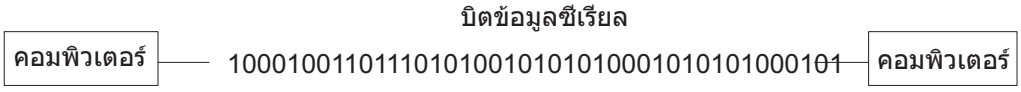
รูปที่ 31. พอร์ตการสื่อสารแบบขนาน

ซีเรียลพอร์ตต้องการเพียงขา หรือสายเดียวเพื่อส่งข้อมูลของอักขระเดียวกันไปยังอุปกรณ์ เพื่อทำดังกล่าว ข้อมูลจะถูกแปลงจากรูปแบบขนาน (ถูกส่งโดยคอมพิวเตอร์) เป็นรูปแบบที่เรียงตามลำดับ โดยที่บิตจะถูกจัดระเบียบให้แต่ละบิตเรียงกันเป็นชุด จากนั้นข้อมูลจะถูกส่งไปยังอุปกรณ์โดยส่งบิตขวาสุด (หรือบิตศูนย์) หลังจากอุปกรณ์รีโมตได้รับข้อมูล ข้อมูลจะถูกแปลงกลับเป็นรูปแบบขนาน ต่อไปนี้จะแสดงการส่งแบบซีเรียลของอักษร a



รูปที่ 32. พอร์ตการสื่อสารแบบซีเรียล

การส่งอักขระเดียวแบบซีเรียลจะเป็นแบบง่ายๆและตรงไปตรงมา อย่างไรก็ตาม ความยุ่งยากจะเพิ่มขึ้นเมื่อจำนวนของอักขระที่ต้องการส่งมีมากขึ้น ดังแสดงต่อไปนี้ ระบบที่ได้รับจะไม่รู้ว่าจุดสิ้นสุดของอักขระแรกอยู่ที่ไหนและอักขระอื่นเริ่มตรงไหน เพื่อแก้ไขปัญหานี้ทั้งสองด้านของลิงก์การสื่อสารต้องถูกซิงโครไนซ์หรือตั้งเวลา



รูปที่ 33. การสื่อสารแบบซีเรียล

ความแตกต่างของฟังก์ชันการทำงานระหว่างพอร์ตของระบบและซีเรียลพอร์ต

POWER5 *พอร์ตของระบบ* แบบรวมจะเหมือนกับซีเรียลพอร์ตยกเว้นพอร์ตของระบบมีให้ใช้งานเฉพาะสำหรับฟังก์ชันที่ได้รับการสนับสนุนโดยเฉพาะ

พอร์ตของระบบถูกปิดใช้งานเมื่อพอร์ต Hardware Management Console (HMC) ถูกเชื่อมต่อเข้ากับ HMC โดยที่สามารถเลือกใช้พอร์ต HMC หรือพอร์ตของระบบ แต่ไม่ใช่ทั้งสอง

แม้ว่าเมื่อไม่มี HMC ต่ออยู่ พอร์ตของระบบแบบรวมจะถูกจำกัดกับ TTY ที่ถูกต่อแบบซีเรียลเพื่อทำหน้าที่คอนโซล มันจะทำงานได้ถูกต้องเฉพาะกับโมเด็มแบบ call-home เทอร์มินัลแบบ async และ UPS เฉพาะที่ได้รับการอนุมัติเท่านั้น การเชื่ออุปกรณ์แบบซีเรียลอื่น (รวมถึงการเชื่อมต่อแบบระบบกับระบบสำหรับ HACMP) ต้องใช้อะแด็ปเตอร์ซีเรียลพอร์ตในสล็อต PCI

การซิงโครไนซ์

การซิงโครไนซ์เป็นกระบวนการของการให้จังหวะการส่งข้อมูลแบบซีเรียลเพื่อให้ระบุข้อมูลที่ถูกส่งได้อย่างถูกต้อง

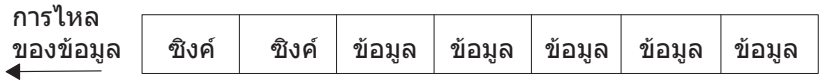
โหมดที่ใช้ทั่วไป 2 โหมดคือซิงโครนัสและอะซิงโครนัส

การส่งข้อมูลแบบซิงโครนัส:

คำว่า *ซิงโครนัส* ถูกใช้เพื่ออธิบายความต่อเนื่องและความแน่นอนของที่ถ่ายโอนบล็อกของข้อมูล

ชนิดของการเชื่อมต่อเหล่านี้ถูกใช้เมื่อต้องถ่ายโอนข้อมูลจำนวนมากอย่างรวดเร็วจากตำแหน่งหนึ่งไปยังตำแหน่งอื่น ความเร็วของการเชื่อมต่อแบบซิงโครนัสสามารถทำได้โดยการถ่ายโอนข้อมูลเป็นบล็อกขนาดใหญ่แทนที่จะเป็นทีละอักขระ

บล็อกของข้อมูลจะถูกรวมกลุ่มและเว้นในช่วงเวลาแบบธรรมดาและถูกนำหน้าโดยอักขระพิเศษที่เรียกว่า syn หรือ synchronous idle character ดูภาพต่อไปนี้



รูปที่ 34. การส่งข้อมูลแบบซิงโครนัส

หลังจากอักขระ syn ได้รับโดยอุปกรณ์รีโมต มันจะถูกถอดรหัสและถูกใช้เพื่อซิงโครไนซ์การเชื่อมต่อ หลังจากการเชื่อมต่อถูกซิงโครไนซ์อย่างถูกต้อง การส่งข้อมูลจะเริ่มต้น

ความคล้ายคลึงของการเชื่อมต่อชนิดนี้จะเป็นการส่งข้อมูลเอกสารข้อความขนาดใหญ่ ก่อนที่เอกสารจะถูกถ่ายโอนข้ามสายซิงโครนัส มันจะถูกแตกออกเป็นบล็อกของประโยคและย่อหน้า จากนั้นบล็อกจะถูกส่งข้างลิงก์การสื่อสารไปยังรีโมตไซต์ โดยใช้การส่งข้อมูลโหมดอื่น ข้อความจะถูกจัดเป็นสตริงยาวๆของตัวอักษร (หรืออักขระ) ที่ประกอบเป็นคำภายในประโยคหรือย่อหน้า อักขระเหล่านี้จะถูกส่งข้ามลิงก์การสื่อสารทีละตัวและถูกประกอบที่ตำแหน่งรีโมต

จังหวะที่ต้องการสำหรับซิงโครไนซ์การเชื่อมต่อได้มาจากอุปกรณ์ที่อยู่บนลิงก์การสื่อสาร อุปกรณ์ทั้งหมดบนลิงก์แบบซิงโครนัสต้องถูกตั้งด้วยสัญญาณนาฬิกาเดียวกัน

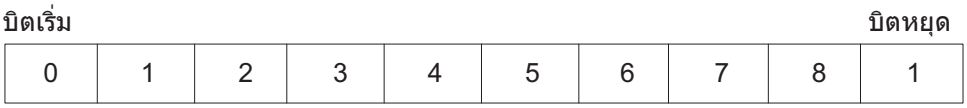
ต่อไปนี้เป็นลิสต์ของคุณลักษณะที่ระบุกับการสื่อสารซิงโครนัส :

- ไม่มีช่องว่างระหว่างอักขระที่ถูกส่ง
- จังหวะเวลาถูกให้โดยโมเด็มหรืออุปกรณ์อื่นที่แต่ละด้านของการเชื่อมต่อ
- ใช้อักขระพิเศษ syn นำหน้าข้อมูลที่ถูกส่ง
- อักขระ syn ถูกใช้ระหว่างบล็อกของข้อมูลสำหรับการให้จังหวะเวลา

การส่งข้อมูลแบบอะซิงโครนัส:

คำว่า *อะซิงโครนัส* ถูกใช้เพื่ออธิบายกระบวนการที่ข้อมูลที่ถูกส่งถูกเข้ารหัสด้วย start และ stop บิต การระบุการเริ่มต้นและสิ้นสุดของแต่ละอักขระ

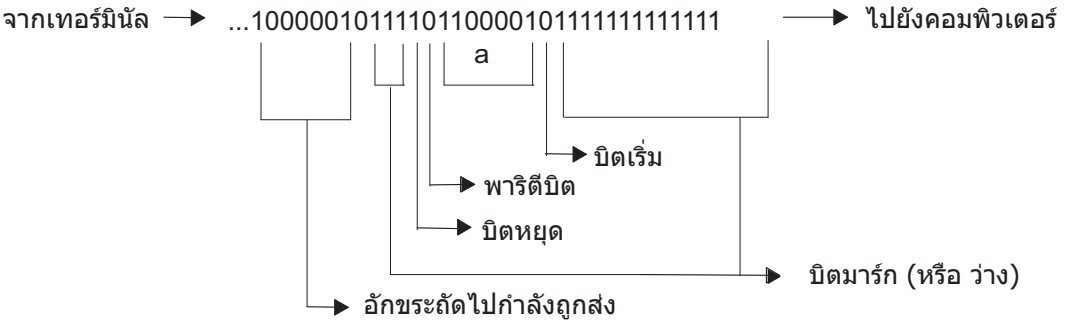
ตัวอย่างของการส่งแบบอะซิงโครนัสแสดงในรูป ต่อไปนี้



รูปที่ 35. การส่งข้อมูลแบบอะซิงโครนัส

บิตเพิ่มเติมเหล่านี้ให้จังหวะหรือการซิงโครไนซ์สำหรับการเชื่อมต่อ โดยการระบุว่าเมื่อใดที่ตัวอักขระทั้งหมดถูกส่งหรือรับ ดังนั้นจังหวะเวลาของแต่ละอักขระจะเริ่มด้วย start บิตและสิ้นสุดด้วย stop บิต

เมื่อมีช่วงว่างระหว่างการส่งตัวอักขระสายการส่งอะซิงโครนัสจะบอกว่ามันอยู่ในสถานะ mark mark คือไบนารีที่เป็น 1 (หรือโวลท์เตจลบ) ที่ถูกส่งระหว่างช่วงเวลาที่ไม่มีกิจกรรมบนสายดังแสดงในรูปต่อไปนี้



รูปที่ 36. Mark (idle) บิตในสตรีมของข้อมูล

เมื่อสถานะ mark ถูกอินเตอร์รัปต์โดยโวลท์เตจบวก (ไบนารีที่เป็น 0) ระบบที่เป็นตัวรับจะรู้ว่าข้อมูลของตัวอักขระกำลังจะตามมา สาเหตุนี้ที่ start บิต ที่นำหน้าข้อมูลตัวอักขระจะเป็น space บิตเสมอ (ไบนารี 0) และ stop บิต ซึ่งบอกจุดสิ้นสุดของตัวอักขระ จะเป็น mark บิต (ไบนารี 1)

ต่อไปนี้เป็นลิสต์ของคุณลักษณะที่ระบุกับการสื่อสารอะซิงโครนัส :

- แต่ละอักขระจะนำหน้าโดย start บิตและตามด้วยหนึ่ง stop บิตหรือมากกว่า
- อาจมีช่องว่างระหว่างตัวอักขระ

พารามิเตอร์การสื่อสารแบบซีเรียล

พารามิเตอร์ที่ใช้ระหว่างการสื่อสารแบบซีเรียลจะรวม บิตต่ออักขระ บิตต่อวินาที (bps) อัตรา baud พาริตี และ start stop และ mark บิต

Bits-per-character:

จำนวนของ bits-per-character (bpc) จะระบุจำนวนของบิตที่ถูกใช้เพื่อแทนอักขระข้อมูลเดียวระหว่างการสื่อสารแบบซีเรียล

ตัวเลขนี้ไม่ได้แสดงถึงจำนวนทั้งหมดของพาริตี stop หรือ start บิตที่ถูกรวมกับอักขระ การตั้งค่าที่เป็นไปได้ 2 อย่างสำหรับ bpc คือ 7 และ 8

เมื่อใช้การตั้งค่า 7 bits-per-character มันจะส่งเฉพาะ 128 อักขระแรก (0-127) ของชุดอักขระมาตรฐานของ ASCII แต่ละอักขระเหล่านี้ถูกแทนด้วย 7 บิตข้อมูล การตั้งค่า 8 bits-per-character ควรถูกใช้เพื่อส่งชุดอักขระที่เป็นส่วนขยายของ ASCII (128-255) แต่ละอักขระเหล่านี้จะถูกแทนโดยใช้ 8 บิตข้อมูล

Bits-per-second (bps):

ให้คำอธิบายสถิติของ bits-per-second

Bits-per-second (bps) เป็นจำนวนของบิตข้อมูล (ไบนารีที่เป็น 1และ 0) ที่ถูกส่งต่อวินาทีผ่านสายการสื่อสาร

อัตราบอด (Baud rate):

อัตรา baud เป็นจำนวนครั้งต่อวินาทีที่สัญญาณการสื่อสารแบบซีเรียลเปลี่ยนสถานะ สถานะจะเป็นระดับโวลท์เตจ ความถี่ หรือมุมของเฟสของความถี่

ถ้าสัญญาณเปลี่ยนหนึ่งครั้งสำหรับแต้ละบิตข้อมูล ดังนั้นหนึ่ง bps จะเท่ากับหนึ่ง baud ตัวอย่างเช่น โมเด็มแบบ 300 เปลี่ยนสถานะของมัน 300 ครั้งต่อวินาที

พาริตีบิต:

พาริตีบิต ไม่เหมือนกับ start และ stop บิต เป็นพารามิเตอร์ที่เป็นอ็อพชัน ที่ถูกใช้ในการสื่อสารเพื่อกำหนดว่า อักขระของข้อมูลที่ถูกส่งถูกได้รับอย่างถูกต้องโดยอุปกรณ์ปลายทางหรือไม่

บิตเริ่ม									**บิตหยุด**
0	1	2	3	4	5	6	7	**8 หรือ พาริตี**	1

รูปที่ 37. พาริตี

พาริตีบิตสามารถมีหนึ่งในห้าของข้อกำหนดต่อไปนี้ :

ไอเท็ม	คำอธิบาย
none	ระบุว่าระบบโลคัลไม่ควรสร้างพาริตีบิตสำหรับอักขระของข้อมูลที่จะถูกส่ง มันยังระบุว่าระบบโลคัลไม่ต้องตรวจสอบพาริตีบิตในข้อมูลที่ได้รับจากรีโมตโฮสต์
คู่	ระบุว่าจำนวนทั้งหมดของไบนารีที่เป็น 1 ในอักขระเดียว รวมกันเป็นจำนวนคู่ ถ้าไม่ พาริตีบิตต้องเป็น 1 เพื่อให้แน่ใจว่าจำนวนทั้งหมดของไบนารีที่เป็น 1 เป็นจำนวนคู่ ตัวอย่างเช่น ถ้าตัวอักษร a (ไบนารี 1100001) ถูกส่งภายใต้พาริตีคู่ ระบบที่ส่งจะเพิ่มจำนวนของไบนารีที่เป็น 1 ในกรณีนี้คือสาม เพื่อทำให้พาริตีบิตเป็น 1 เพื่อให้จำนวนของไบนารี 1 เป็นจำนวนคู่ ถ้าตัวอักษร A (ไบนารี 1000001) ถูกส่งภายใต้สถานการณ์เดียวกัน พาริตีบิตควรเป็น 0 เพื่อให้จำนวนของไบนารี 1 เป็นจำนวนคู่i
คี่	ทำงานภายใต้แนวทางเดียวกันกับพาริตีคู่ ยกเว้นจำนวนไบนารีที่เป็น 1 ทั้งหมดต้องเป็นจำนวนคี่
space	จะระบุว่าพาริตีบิตจะเป็นไบนารี 0 เสมอ คำอื่นถูกใช้สำหรับพาริตี space คือ bit filling ซึ่งมาจากมันถูกใช้เป็นตัวเติมสำหรับข้อมูลแบบ 7 บิตที่ถูกส่งไปยังอุปกรณ์ที่สามารถรับได้เฉพาะข้อมูล 8 บิต อุปกรณ์นั้นจะเป็น space พาริตีบิตเป็นขิตข้อมูลเพิ่มเติมสำหรับการส่งอักขระ
mark	ทำงานภายใต้แนวทางเดียวกันกับ space พาริตี ยกเว้นพาริตีบิตจะเป็นไบนาที่เป็น 1 เสมอ mark พาริตีบิตทำหน้าที่เป็นตัวเติมเท่านั้น

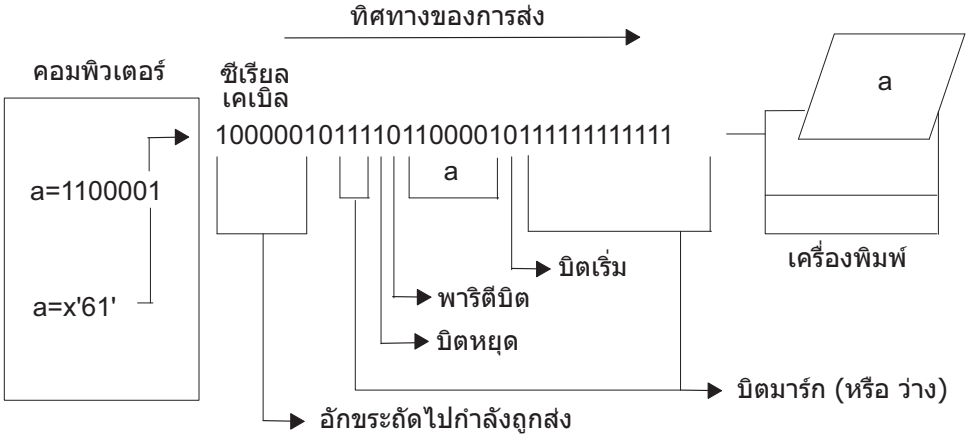
Start, stop และ mark บิต:

start และ stop บิตถูกใช้ในการสื่อสารแบบอะซิงโครนัสเพื่อเป็นการให้จังหวะหรือการซิงโครไนซ์อักขระข้อมูลที่ถูกส่ง

ถ้าไม่มีการใช้บิตเหล่านี้ ระบบที่ส่งและรับจะไม่รู้ว่าอักขระตัวแรกสิ้นสุดที่ไหนและตัวอื่นเริ่มที่ไหน

บิตอื่นๆถูกใช้เพื่อแยกอักขระข้อมูลระหว่างการส่ง คือ mark (หรือ ว่าง) RS บิต บิตนี้ ไบนารีที่เป็น 1 ถูกส่งเมื่อสายการสื่อสารว่าง หรือไม่มีอักขระถูกส่งหรับรับ

เมื่อระบบได้รับบิตเริ่มต้น (ไบนารี 0) ระบบเข้าใจว่าบิตอักขระจำนวนคงที่ (กำหนดโดย พารามิเตอร์ **bits per character**) และบิตคู่ (กำหนดโดยพารามิเตอร์ **parity**) จะตามหลังบิตเริ่มต้น จากนั้น ระบบได้รับบิตหยุด (ไบนารี 1) ในตัวอย่างต่อไปนี้ มีบิต **parity** และ **bits per character** คือ 7



รูปที่ 38. Start, stop และ mark บิต

มาตรฐาน EIA 232D

มาตรฐาน EIA 232D ถูกพัฒนาในปี 1969 เพื่อระบุการเชื่อมต่อระหว่างคอมพิวเตอร์และโมเด็ม

คำของมันเองเป็นตัวย่อซึ่งสามารถอย่างได้ดังต่อไปนี้:

Electronics Industry Association (EIA) ยอมรับมาตรฐาน หมายเลข ID 232 revision D

EIA 232D จะระบุคุณลักษณะของการเชื่อมต่อแบบฟิสิคัลและทางไฟฟ้าระหว่าง 2 อุปกรณ์ ชื่อและคำย่อถูกกำหนดให้กับแต่ละขาหรือสายที่มีความจำเป็นสำหรับการสื่อสารแบบซีเรียล ตัวอย่างเช่น :

ตารางที่ 98. การเชื่อมต่อ EIA 232D

Signal	ติดตั้ง ชนิด	สัญลักษณ์	Pin
ส่งข้อมูล	DCE	TxD	2
รับข้อมูล	DTE	RxD	3
Request to Send	DCE	RTS	4
Clear to Send	DTE	CTS	5
Data Set Ready	DTE	DSR	6
สัญญาณกราวด์		SG	7
Carrier Detect	DTE	CD	8
Data Terminal Ready	DCE	DTR	20
Ring Indicator	DTE	RI	22

ใน EIA 232D อุปกรณ์จะใช้ขา 2 (TxD) สำหรับเอาต์พุต (ตัวอย่างเช่น คอมพิวเตอร์และเทอร์มินัล) และตั้งชื่อว่า data terminal equipment (DTE) อุปกรณ์จะใช้ขา 2 (TxD) สำหรับอินพุต (ตัวอย่างเช่น โมเด็ม) และตั้งชื่อว่า data communication equipment (DCE)

EIA 232D ยังระบุตัวเชื่อมต่อ อุปกรณ์ DTE โดยมากจะมีตัวเชื่อมต่อตัวผู้ ขณะที่อุปกรณ์ DCE จะมีตัวเชื่อมต่อตัวเมีย มาตรฐานนี้จะไม่ยึดติดกับผู้ผลิตเสมอไป ดังนั้นผู้ใช้ควรตรวจสอบเอกสารของอุปกรณ์ก่อนการเชื่อมต่อสายเคเบิล

วิธีการสื่อสารแบบอะซิงโครนัส

นี่จะอธิบายรูปแบบ 2 รูปแบบของการสื่อสารแบบอะซิงโครนัส แบบทางเดียว และแบบสองทาง (ซึ่งจะรวมแบบ half-duplex และ full-duplex)

แบบ Simplex หรือการสื่อสารแบบทางเดียว เป็นรูปแบบการเชื่อมต่อระหว่างสองอุปกรณ์ที่ง่ายที่สุด โหมดของการสื่อสารนี้จะยอมให้ข้อมูลถูกส่งได้ทางเดียวเท่านั้น และต้องการแค่สองสายเพื่อที่จะเชื่อมต่อ ตัวอย่างเช่น TxD (หรือ RxD) และ SG

มีรูปแบบ 2 รูปแบบสำหรับการสื่อสารแบบ 2 ทาง : half-duplex และ full-duplex การเชื่อมต่อในโหมด half-duplex จะยอมให้ข้อมูลถูกส่งได้ 2 ทิศทางแต่ไม่พร้อมกัน การเปรียบเทียบของ half-duplex จะเป็นการใช้วิทยุ CB โดยที่การสื่อสารแบบ 2 ทางเป็นไปได้แต่จะมีเพียงคนเดียวที่สามารถพูดในเวลาหนึ่งๆ

ใน full-duplex หรือโหมด duplex การสื่อสารข้อมูลสามารถทำใน 2 ทิศทางพร้อมกัน การเปรียบเทียบสำหรับ full-คือการคุยโทรศัพท์เมื่อคนสองคนคุยกันได้ในเวลาเดียวกัน

โฟลว์คอนโทรล

โฟลว์คอนโทนรลบางชนิดถูกต้องการโดยอุปกรณ์แบบซีเรียลเพื่อจำกัดจำนวนของข้อมูลที่ถูกส่งโดยระบบ

อุปกรณ์แบบซีเรียล เช่น เครื่องพิมพ์ และโมเด็ม ไม่สามารถประมวลผลข้อมูลเร็ว หรือมีประสิทธิภาพเท่ากับคอมพิวเตอร์ที่มันเชื่อมต่อ

คำว่า *โฟลว์คอนโทรล* ถูกใช้เพื่ออธิบายวิธีที่อุปกรณ์แบบซีเรียลควบคุมจำนวนของข้อมูลที่ถูกส่งไปที่มัน

RTS/CTS ฮาร์ดแวร์โฟลว์:

Request to send/clear to send (RTS/CTS) บางครั้งเรียกว่า pacing หรือฮาร์ดแวร์แฮนด์เชคกิ้งแทนที่จะเป็นโฟลว์คอนโทรล

คำว่าฮาร์ดแวร์แฮนด์เชคกิ้งมาจากการใช้สายเคเบิลและโวลท์เตจเป็นวิธีควบคุมการส่งข้อมูล ไม่เหมือนกับ XON/XOFF ซึ่งจะส่งอักขระควบคุมในสตรีมของข้อมูล RTS/CTS จะใช้โวลท์เตจบวกและลบของขาหรือสายที่แยกต่างหากในอุปกรณ์สายเคเบิล

โวลต์เตจที่เป็นบวกหมายถึงยอมให้มีการส่งข้อมูลขณะที่โวลต์เตจที่เป็นลบส่งสัญญาณว่าควรระงับการส่งสัญญาณ

DTR/DSR ฮาร์ดแวร์โฟลว์:

Data terminal ready (DTR) เป็นอีกรูปแบบหนึ่งของฮาร์ดแว์โฟลว์คอนโทรล ที่โดยทั่วไปจะถูกสร้างโดยอุปกรณ์ เช่น เครื่องพิมพ์ เพื่อระบุว่ามันพร้อมที่จะสื่อสารกับระบบ สัญญาณนี้จะใช้ร่วมกับ data set ready (DSR) ที่ถูกสร้างโดยระบบเพื่อควบคุมการโฟลว์ของข้อมูล

โวลต์เตจที่เป็นบวกหมายถึงยอมให้มีการส่งข้อมูลขณะที่โวลต์เตจที่เป็นลบส่งสัญญาณว่าควรระงับการส่งสัญญาณ

XON/XOFF ซอฟต์แวร์โฟลว์:

Transmitter on/transmitter off (XON/XOFF) โฟลว์คอนโทรลจะเกี่ยวข้องกับการส่งของตัวอักขระควบคุมการส่งข้อมูลในสตรีมของข้อมูล (TxD และ RxD) ด้วยเหตุนี้มันจะถูกอ้างถึงถึงเป็นซอฟต์แวร์โฟลว์คอนโทรล

เมื่อข้อมูลถูกส่งไปยังโมเต็ม มันจะถูกใส่ไว้ในบัฟเฟอร์ ก่อนที่บัฟเฟอร์จะเต็ม โมเด็มจะส่งอักขระ XOFF ไปยังระบบและระบบจะหยุดส่งข้อมูล เมื่อบัฟเฟอร์ของโมเด็มว่างและพร้อมที่จะรับข้อมูลเพิ่มเติม มันจะส่งอักขระ XON กลับไปยังระบบเพื่อให้ส่งข้อมูลเพิ่มเติม

การตั้งค่าพอร์ตสำหรับ RTS/CTS ฮาร์ดแวร์แฮนด์เชคกิ้ง:

โมเด็มที่เชื่อมต่อกับเซิร์ฟเวอร์จะทำงานที่ความเร็ว 9600 หรือสูงกว่า ได้รับคำแนะนำให้ใช้ RTS/CTS ฮาร์ดแวร์แฮนด์เชคกิ้งแทนที่จะใช้ XON/XOFF โฟลว์คอนโทรล

ซึ่งจะเป็นการหลีกเลี่ยง buffer overrun ในระบบที่มีรีซอร์สจำกัด RTS ไม่ได้เป็นค่าดีฟอลต์บนพอร์ต tty ใดๆและต้องถูกตั้งโดยผู้ดูแลระบบ

ข้อกำหนดเบื้องต้น

อย่างน้อยคุณต้องใช้สายเคเบิลที่มี 5 เส้นเพื่อสนับสนุน RTS/CTS

เพื่อเปิดใช้งาน RTS/CTS สำหรับพอร์ต ใช้ขั้นตอนต่อไปนี้ :

1. ใช้ `smit tty` fast path
2. เลือก **Change / Show Characteristics of a TTY**
3. เลือก tty ที่ RTS/CTS จะถูกเปิดใช้งาน
4. ตั้ง FLOW CONTROL เพื่อใช้ฟิลด์ **rts**
5. เลือก **Do**
6. ออกจาก SMIT

อุปกรณ์เทอร์มินัล TTY

อุปกรณ์เทอร์มินัล tty เป็นอุปกรณ์อักขระที่ดำเนินการอินพุตเอาต์พุตบนพื้นฐานของอักขระต่ออักขระ

การสื่อสารระหว่างอุปกรณ์เทอร์มินัล และโปรแกรมที่อ่านและเขียนลงในอุปกรณ์เหล่านี้ที่ถูกควบคุมโดยอินเตอร์เฟส tty ตัวอย่างของอุปกรณ์ tty คือ:

- โมเด็ม
- เทอร์มินัล ASCII
- คอนโซลระบบ (LFT)
- **aixterm** ภายใต้ AIXwindows

อุปกรณ์ tty สามารถเพิ่ม ลบ แสดงรายการ และเปลี่ยนแปลงเช่นเดียว กับอุปกรณ์อื่นๆ บนระบบของคุณโดยใช้เครื่องมือ SMIT หรือคำสั่งเฉพาะ อุปกรณ์

ค่า TERM สำหรับหน้าจอและเทอร์มินัลที่ต่างกัน

ข้อมูลเกี่ยวกับความสามารถของเทอร์มินัลจะถูกเก็บในฐานข้อมูล `terminfo`

ค่าของตัวแปรสภาวะแวดล้อม **TERM** จะระบุคำอธิบายเทอร์มินัลที่ระบุในฐานข้อมูล `terminfo` ซึ่งจะให้ข้อมูลทั้งหมดที่โปรแกรมต้องการสำหรับการสื่อสารอย่างมีประสิทธิภาพกับอุปกรณ์ tty ปัจจุบัน

ตารางที่ 99. ค่าของ TERM สำหรับเทอร์มินัลต่างๆ

หน้าจอ/เทอร์มินัล	ค่า
เทอร์มินัล 3161 ASCII	ibm3161
เทอร์มินัล 3163 ASCII	ibm3161
DEC VT100 (เทอร์มินัล)	vt100
DECVT220	vt220
3151 ASCII Display Station พร้อมกับ Cartridge หรือ 3161 ASCII Display Station พร้อมกับ Cartridge	ibm3161-C
3162 ASCII Display Station	ibm3161
3162 ASCII Display Station พร้อมกับ Cartridge	ibm3162

ตารางที่ 99. ค่าของ TERM สำหรับเทอร์มินัลต่างๆ (ต่อ)

หน้าจอ/เทอร์มินัล	ค่า
6091 หน้าจอ	lft
AIXwindows	aixterm

สำหรับข้อมูลเกี่ยวกับ entry ในฐานข้อมูล `terminfo` ดูที่ `terminfo` รูปแบบไฟล์ใน *ข้อมูลอ้างอิงไฟล์* เพื่อแปลง entry `termcap` เป็น entry `terminfo` ดูที่คำสั่ง **captoinfo** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1* (ไฟล์ `termcap` ประกอบด้วยคำอธิบายเทอร์มินัลสำหรับระบบ Berkeley แบบเก่า)

คุณลักษณะของ TTY

line discipline จัดเตรียมอินเตอร์เฟสผู้ใช้แบบไม่ขึ้นอยู่กับฮาร์ดแวร์สำหรับการสื่อสารระหว่างคอมพิวเตอร์และอุปกรณ์อะซิงโครนัส

ตัวอย่างเช่น ผู้ใช้สามารถลบสายเดียว หรืออินเตอร์รัปต์กระบวนการที่กำลังรันอยู่โดยการพิมพ์ลำดับของอักขระนั้นๆ คุณสามารถกำหนดความหมายของลำดับอักขระเหล่านี้รวมถึง ตั้งค่าอักขระเทอร์มินัลอื่นๆ เช่นความเร็วการสื่อสาร โดยใช้คำสั่ง **chdev,** System Management Interface Tool (SMIT) หรือคำสั่ง **stty**

แอ็ตทริบิวต์ที่ต้องการบนอุปกรณ์ TTY ที่ต่ออยู่

การสื่อสารที่ถูกต้องระหว่างโฮสต์และอุปกรณ์ tty ที่ต่ออยู่ต้องมีข้อกำหนดเหล่านี้

- สายการสื่อสารถูกเชื่อมต่ออย่างถูกต้อง
- ค่าการสื่อสาร (ความเร็วของสาย ขนาดของอักขระ พาริตี stop บิต และอินเตอร์เฟส) ระหว่างโฮสต์และอุปกรณ์ tty ที่ต่ออยู่ตรงกัน

การจัดการอุปกรณ์ TTY

งานการจัดการอุปกรณ์และ SMIT fast paths ที่เกี่ยวข้องของมัน และคำสั่งสามารถถูกอ้างอิงที่นี่

ตารางที่ 100. งานการจัดการอุปกรณ์ TTY

งาน	วิธีลัด SMIT	คำสั่งหรือไฟล์
List Defined TTY Devices	`smit lsdtty`	**lsdev -C -c tty -H**
Add a TTY	`smit mktty`	**mkdev -t tty**[1,2]
Move a TTY to Another Port[3]	`smit movtty`	**chdev -l** *Name* **-p** *ParentName* **-w** *ConnectionLocation*[2,4]
Change / Show Characteristics of a TTY	`smit chtty`	**lsattr -l***Name* **-E** (to show); **chdev -l** *Name* (to change)[4,5]
Remove a TTY[3]	`smit rmtty`	**rmdev -l** *Name*
Configure a Defined TTY (Make Available for Use)	`smit mktty`	**mkdev -l** *Name*

หมายเหตุ:

1. แฟล็กอื่นอาจถูกใช้เพื่อระบุอุปกรณ์ tty ใหม่เพิ่มเติม ตัวอย่างเช่น เพื่อกำหนดและตั้งค่าอุปกรณ์ tty RS-232 ที่ถูกเชื่อมกับพอร์ต 0 บน 8-พอร์ต อะซิงโครนัส อะแด็ปเตอร์ sa3 ด้วยแอ็ตทริบิวต์ speed ถูกตั้งเป็น 19200 และแอ็ตทริบิวต์อื่นถูกตั้งเป็นค่าที่ถูกดึงจากไฟล์ foo :

   ```
   mkdev -t tty -s rs232 -p sa3 -w 0 -a speed=19200 -f foo
   ```

2. คำสั่ง **mkdev** และ **chdev** สนับสนุนอ็อพชันที่ไม่สามารถทำได้ด้วย SMIT
3. ปิดการใช้งาน tty ก่อนที่จะทำงานนี้ ดูที่ **pdisable** คำสั่งใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*
4. ใช้แฟล็กเพื่อเปลี่ยนอักขระที่ระบุเกี่ยวกับ tty จากบรรทัดรับคำสั่ง
5. คุณสามารถเลือก Posix baud rate ฟังก์ชัน List หรือคุณสามารถพิมพ์ non-Posix baud rate ในฟิลด์โดยตรง ถ้าอัตรา baud ที่เลือกไม่สามารถถูกใช้โดยฮาร์ดแวร์ของโมเด็ม ระบบจะแสดงข้อความแสดงข้อผิดพลาด

ถ้าเพิ่มหรือเปลี่ยน tty จากบรรทัดรับคำสั่ง ศึกษาลิสต์ต่อไปนี้เพื่อดูว่าชื่อของ *Attribute* ที่จะระบุในแฟล็ก
-a *Attribute*=*Value* สำหรับอักขระที่คุณต้องการตั้ง ตัวอย่างเช่น ระบุ `-a speed=Value` เพื่อตั้งอัตรา baud ของอุปกรณ์ tty

ตารางที่ 101. แอ็ตทริบิวต์ TTY

คุณสมบัติ	ชื่อแอ็ตทริบิวต์
Enable LOGIN	ล็อกอิน
ความเร็วอัตรา BAUD	ความเร็ว
PARITY	พาริตี
BITS ต่ออักขระ	bpc
จำนวนของ STOP BITS	stops
TIME ก่อนที่จะไปยังการตั้งพอร์ตถัดไป	timeout
XON-XOFF handshaking	xon
TERMINAL type	term
FLOW CONTROL ที่จะถูกใช้	flow_disp
OPEN DISCIPLINE ที่จะถูกใช้	open_disp
แอ็ตทริบิวต์ STTY สำหรับ RUN time	runmodes
แอ็ตทริบิวต์ STTY สำหรับ LOGIN	logmodes
ตัวจัดการกิจกรรมของ RUN เชลล์	เชลล์
ชื่อ LOGGER	logger
STATUS ของอุปกรณ์ที่ BOOT time	autoconfig
จำนวนของ TRANSMIT บัฟเฟอร์	tbc
ระดับการทริกเกอร์ RECEIVE	rtrig
STREAMS modules to be pushed at open time	โมดูล
ไฟล์การแม็พ INPUT	imap
ไฟล์การแม็พ OUTPUT	omap
ไฟล์การแม็พ CODESET	csmap

ตารางที่ 101. แอ็ตทริบิวต์ TTY (ต่อ)

คุณสมบัติ	ชื่อแอ็ตทริบิวต์
อักขระ INTERRUPT	intr
อักขระ QUIT	quit
อักขระ ERASE	erase
อักขระ KILL	kill
อักขระ END OF FILE	eof
อักขระ END OF LINE	eol
อักขระ END OF LINE ลำดับที่สอง	eol2
อักขระ DELAY SUSPEND PROCESS	dsusp
อักขระ SUSPEND PROCESS	susp
อักขระ LITERAL NEXT	lnext
อักขระ START	start
อักขระ STOP	stop
อักขระ WORD ERASE	werase
อักขระ REPRINT LINE	reprint
อักขระ DISCARD	discard

การแก้ปัญหา TTY

มีหลายสถานการณ์จำลองการแก้ปัญหา TTY ทั่วไป

สถานการณ์จำลองการแก้ปัญหา TTY ทั่วไปรวมถึงข้อผิดพลาด `Respawning Too Rapidly` พอร์ต TTY หยุดทำงาน และไฟล์การล็อกข้อผิดพลาดทั่วไป คำสั่ง และข้อความรายงานข้อผิดพลาด

ข้อผิดพลาด Respawning Too Rapidly:

The system records the number of **getty** processes created for a particular tty in a short time period. If the number of **getty** processes created in this time frame exceeds five, then the `Respawning Too Rapidly` error is displayed on the console and the port is disabled by the system.

tty จะยังคงถูกปิดใช้งานประมาณ 19 นาที หรือจนกว่าผู้ดูแลระบบจะเปิดใช้งานพอร์ตอีกครั้ง At the end of the 19 minutes, the system automatically enables the port, resulting in the creation of a new **getty** process.

สาเหตุที่เป็นไปได้รวมถึงต่อไปนี้ :

- การตั้งค่าโมเด็มไม่ถูกต้อง
- พอร์ตถูกระบุและถูกเปิดใช้งานแต่ไม่มีสายเคเบิลหรืออุปกรณ์ต่ออยู่กับมัน
- สายเคเบิลไม่ดีหรือการเชื่อมต่อหลวม
- สัญญาณรบกวนบนสายการสื่อสาร

- ไฟล์ /etc/environment หรือ /etc/inittab ที่ไม่ดี
- การตั้งค่า tty เสีย
- ฮาร์ดแวร์ไม่สมบูรณ์

ใช้โพรซีเดอร์ต่อไปนี้สำหรับกู้คืน เลือกใช้ที่เหมาะกับสถานการณ์ของคุณ

- การตั้งค่าโมเด็มไม่ถูกต้อง

  ต้องแน่ใจว่า carrier detect ของโมเด็ม *ไม่* ถูกบังคับเป็น high

  หมายเหตุ: ต่อไปนี้ใช้ได้กับโมเด็มแบบ Hayes-compatible

  1. เชื่อมต่อกับโมเด็มและตรวจสอบโปรไฟล์ที่แอ็คทีฟ
  2. ตั้ง carrier detect ของโมเด็มเป็น **&C1** แทนที่จะเป็น **&C0** (ถูกบังคับให้เป็น high) ใช้คำสั่ง AT ต่อไปนี้ของโมเด็มเพื่อตั้งและเปลี่ยนแอ็ตทริบิวต์ attribute:

     ```
     AT&C1
     AT&W
     ```

     หมายเหตุ:

     a. ดูที่ "การส่งคำสั่ง AT โดยใช้คำสั่ง cu" ในหน้า 640

     b. ดูเอกสารของโมเด็มของคุณสำหรับข้อมูลเพิ่มเติม
- ปิดการใช้งาน tty ลบคำจำกัดความของ tty หรือเชื่อมอุปกรณ์กับพอร์ต :
  - เพื่อปิดใช้งานคำจำกัดความของ tty ใช้คำสั่ง **chdev** ดังต่อไปนี้ :

    ```
    chdev -l ttyName -a Login=disable
    ```

    หลังจากรันคำสั่งนี้ tty จะ*ไม่* กลายเป็น enabled หลังจากที่ระบบรีสตาร์ท
  - เพื่อลบคำจำกัดความของ tty :
    1. ปิดการใช้งานพอร์ต tty ใช้คำสั่ง **pdisable** และ enter:

       ```
       pdisable ttyName
       ```
    2. ลบคำจำกัดความของ tty จากระบบ ดูที่ "การจัดการอุปกรณ์ TTY" ในหน้า 625 สำหรับข้อมูลเพิ่มเติม
- ตรวจสอบหาสายที่ไม่ดีหรือการเชื่อมต่อที่หลวม:
  1. ตรวจสอบการเดินสายเคเบิล ทำการเชื่อมต่อที่หลวมให้แน่น และเปลี่ยนตัวเชื่อมต่อที่เสียหายหรือไม่เหมาะสม
  2. ตรวจสอบว่าสายเคเบิลที่สงสัย คือสายซีเรียล IBM P/N 6323741 หรือเป็นสายที่เป็นไปตามมาตรฐาน เปลี่ยนสายที่เสียหายหรือไม่เหมาะสม
- กำจัดสัญญาณรบกวนบนสายการสื่อสาร:
  1. ตรวจสอบว่าสายเคเบิลมีความยาวและอิมพีแดนซ์ที่ถูกต้อง
  2. ต้องแน่ใจว่าวง toroid อยู่ในที่ที่ต้องการบนสายที่ยาว
  3. ตรวจสอบเส้นทางของสายเคเบิล มันไม่ควรใกล้กับไฟฟลูออเรสเซนต์หรือมอเตอร์
- ตรวจสอบหาไฟล์ /etc/environment หรือ the /etc/inittab ที่ไม่ดี:
  1. ถ้าเป็นไปได้ เปรียบเทียบไฟล์เหล่านี้กับคัดลอกที่รู้ว่าดี
  2. คัดลอกไฟล์นี้ไว้สำรองและทำการเปลี่ยนแปลงถ้าต้องการ

3. ในไฟล์ /etc/environment ลบบรรทัดที่ *ไม่ใช่*:
   - บรรทัดว่าง
   - บรรทัดที่เป็นหมายเหตุ
   - *variable=value*
4. ในไฟล์ /etc/inittab ตรวจสอบบรรทัดของอุปกรณ์ tty ถ้า tty ถูกตั้งเป็น off เป็นไปได้ว่าพอร์ต tty ไม่ถูกใช้ ถ้ามันไม่ถูกใช้ ลบคำจำกัดความของ tty หรือต่ออุปกรณ์เข้ากับพอร์ต

- ลบการตั้งค่า tty ที่เสียหาย:
  1. ลบคำจำกัดความของ tty ดูที่ "การจัดการอุปกรณ์ TTY" ในหน้า 625 สำหรับข้อมูลเพิ่มเติม
  2. ถ้าคุณต้องการบันทึกที่เป็นฮาร์ดคัดลอกของคำจำกัดความของ tty ก่อนที่จะลบมัน กดคีย์ Image (F8 หรือ Esc+8) นี่จะดักจับรูปของหน้าจอปัจจุบัน และคัดลอกมันไปยังไฟล์ smit.log ในไดเร็กทอรี $HOME ของคุณ
  3. อ่านคำจำกัดความของ tty ดูวิธีการสำหรับ Adding a TTY ภายใต้ "การจัดการอุปกรณ์ TTY" ในหน้า 625
- หาฮาร์ดแวร์ที่ไม่สมบูรณ์ :
  1. รันการวินิจฉัยโดยใช้คำสั่ง **diag**
  2. ถ้าตรวจพบปัญหาของฮาร์ดแวร์ ทำตามโพรซีเดอร์การแก้ปัญหาแบบโลคัล

ข้อมูลล็อกข้อผิดพลาด และตัวระบุล็อก TTY:

คำสั่งต่อไปนี้และไฟล์การล็อกจะเกี่ยวข้องกับ TTY

คำสั่ง : **errclear**

คำสั่งนี้จะลบ entry จากล็อกข้อผิดพลาด ล็อกทั้งหมดสามารถถูกลบด้วย errclear 0 หรือ entry ที่มีหมายเลข ID ข้อผิดพลาดที่ระบุ คลาส หรือชนิดสามารถถูกลบ

คำสั่ง: **errpt**

คำสั่งนี้จะสร้างรายงานข้อผิดพลาดจาก entry ในล็อกข้อผิดพลาดของระบบ รูปแบบที่ถูกใช้มากที่สุดสำหรับคำสั่งนี้ คือ errpt -a | pg ซึ่งจะสร้างรายงานแบบละเอียดเริ่มต้นด้วยข้อผิดพลาดล่าสุด

ไฟล์ : /var/adm/ras/errlog

ไฟล์นี้จะเก็บอินสแตนซ์ของข้อผิดพลาดและความล้มเหลวที่พบโดยระบบ ไฟล์ errlog มีแนวโน้มว่าจะมีความยาวมาก ถ้าไม่ถูกลบเป็นประจำ มันจะใช้พื้นที่บนฮาร์ดดิสก์ของคุณค่อนข้างมาก ใช้คำสั่ง **errclear** ที่กล่าวถึงก่อนหน้านี้เพื่อลบไฟล์นี้

ไฟล์ : /usr/include/sys/errids.h

ไฟล์ส่วนหัว errids.h ที่เชื่อม ID ของข้อผิดพลาดกับเลเบลของข้อผิดพลาด

ข้อความของรายงานข้อผิดพลาดทั่วไปต่อไปนี้จะเกี่ยวข้องกับ TTY:

ตารางที่ 102. ข้อความแสดงข้อผิดพลาดของ TTY

ข้อความ	รายละเอียด	หมายเหตุ
Core Dump	โปรแกรมซอฟต์แวร์ถูกหยุดแบบไม่ปกติ	ข้อผิดพลาดนี้จะถูกล็อกเมื่อโปรแกรมซอฟต์แวร์ถูกหยุดแบบไม่ปกติและทำให้เกิด core dump ผู้ใช้อาจไม่ได้ออกจากโปรแกรมอย่างถูกต้อง ระบบอาจถูกปิดระบบขณะที่ผู้ใช้กำลังใช้งานแอ็พพลิเคชัน หรือเทอร์มินัลของผู้ใช้ถูกล็อกและแอ็พพลิเคชันถูกหยุด
Errlog On	Errdaemon ถูกเปิด	ข้อผิดพลาดนี้ถูกล็อกโดย **error** daemon เมื่อการล็อกข้อผิดพลาดถูกสตาร์ท ระบบจะปิดการล็อกข้อผิดพลาดระหว่างการปิดระบบ
Lion Box Died	ขาดการสื่อสารกับ 64-พอร์ต คอนเซนเตรเตอร์	ข้อผิดพลาดนี้จะถูกล็อกโดยไดรเวอร์ของ 64-พอร์ต คอนเซนเตรเตอร์ ถ้าการสื่อสารกับคอนเซนเตรเตอร์หายไป ถ้าคุณได้รับข้อผิดพลาดนี้ ตรวจสอบวันที่และเวลาประทับเพื่อดูว่า ถ้าผู้ใช้อาจเป็นสาเหตุให้ข้อความนี้เกิดขึ้น ชุดของข้อผิดพลาดเหล่านี้สามารถระบุปัญหากับ 64-พอร์ต อะแด็ปเตอร์ หรือฮาร์ดแวร์ที่เกี่ยวข้องกับมัน
Lion Buffero	Buffer overrun: 64-พอร์ต คอนเซนเตรเตอร์	ข้อผิดพลาดนี้จะเกิดขึ้นเมื่อบัฟเฟอร์ของฮาร์ดแวร์ใน 64-พอร์ต คอนเซนเตรเตอร์ overrun ถ้าอุปกรณ์และสายเคเบิลอนุญาต ลองเพิ่ม request to send (RTS) แฮนเช็กกิ้งเข้ากับพอร์ตและอุปกรณ์ นอกจากนี้ลองลดอัตรา baud
Lion Chunknumc	Bad chunk count: ตัวควบคุม 64-พอร์ต	ข้อผิดพลาดนี้จะเกิดเมื่อค่าของจำนวนของอักขระใน chunk ไม่ตรงกับค่าในบัฟเฟอร์ ข้อผิดพลาดนี้ยังระบุถึงปัญหากับฮาร์ดแวร์ ลองรันการวินิจฉัยบนอุปกรณ์
Lion Hrdwre	ไม่สามารถเข้าถึงหน่วยความจำบนตัวควบคุม 64-พอร์ต	ข้อผิดพลาดนี้จะถูกล็อกโดยไดรเวอร์ของ 64-พอร์ต คอนเซนเตรเตอร์ ถ้ามันไม่สามารถเข้าถึงหน่วยความจำบนตัวควบคุม 64-พอร์ต
Lion Mem ADAP	ไม่สามารถจัดสรรหน่วยความจำ : โครงสร้าง ADAP	ข้อผิดพลาดนี้ถูกล็อกโดยไดรเวอร์ของ 64-พอร์ต คอนเซนเตรเตอร์ถ้ารูทีน **malloc** สำหรับโครงสร้าง adap ล้มเหลว
Lion Mem List	ไม่สามารถจัดสรรหน่วยความจำ : ลิสต์ TTYP_T	ข้อผิดพลาดนี้ถูกล็อกโดยไดรเวอร์ของ 64-พอร์ต คอนเซนเตรเตอร์ ถ้ารูทีน **malloc** สำหรับโครงสร้างลิสต์ *ttyp_t* ล้มเหลว
Lion Pin ADAP	ไม่สามารถจัดสรรหน่วยความจำ : โครงสร้าง ADAP	ข้อผิดพลาดนี้ถูกล็อกโดยไดรเวอร์ของ 64-พอร์ต คอนเซนเตรเตอร์ ถ้ารูทีน **pin** สำหรับโครงสร้าง adap ล้มเหลว
SRC	โปรแกรมซอฟต์แวร์มีข้อผิดพลาด	ข้อผิดพลาดนี้ถูกล็อกโดย System Resource Controller (SRC) daemon ในเหตุการณ์ที่มีเงื่อนไขที่ไม่ปกติบางอย่าง เงื่อนไขที่ไม่ปกติจะถูกแบ่งออกเป็น 3 พื้นที่ : ระบบย่อยล้มเหลว การสื่อสารล้มเหลว และความล้มเหลวอื่น
Lion Unkchunk	โค้ดระบุความผิดพลาดที่ไม่รู้จักจาก 64-พอร์ต คอนเซนเตรเตอร์	โค้ดระบุความผิดพลาด: ได้รับจำนวนของอักขระใน chunk

ตารางที่ 102. ข้อความแสดงข้อผิดพลาดของ TTY (ต่อ)

ข้อความ	รายละเอียด	หมายเหตุ
TTY Badinput	สายเคเบิลหรือการเชื่อมต่อไม่ดี	พอร์ตสร้างอินพุตเร็วกว่าที่ระบบจะสามารถใช้มัน และบางอินพุตนั้นถูกทิ้ง โดยทั่วไป อินพุตที่ไม่ดีมีสาเหตุจากสัญญาณ RS-232 หนึ่งหรือมากกว่าเปลี่ยนสถานะของมันอย่างรวดเร็วและซ้ำๆในช่วงเวลาที่สั้น ทำให้ระบบของคุณใช้เวลาส่วนใหญ่กับตัวจัดการอินเตอร์รัปต์ โดยทั่วไปข้อผิดพลาดของสัญญาณมีสาเหตุจากตัวเชื่อมต่อที่หลวมหรือเสียหาย กราวด์ไม่ดี หรือชีลด์ไม่ดี หรือมีสัญญาณในลิงก์การสื่อสาร
TTY Overrun	ตัวรับ overrun บนอินพุต	พอร์ต TTY ส่วนใหญ่มี 16-อักขระ อินพุต FIFO และค่าตั้งแบบดีฟอลต์ที่ระบุว่าจะถูกอินเตอร์รัปต์หลังจากได้รับ 14 อักขระ ข้อผิดพลาดนี้ถูกรายงานเมื่อไดรเวอร์ของตัวจัดการอินเตอร์รัปต์เคลียร์อินพุต FIFO และข้อมูลหายไป วิธีการแก้ไขที่เป็นไปได้ขึ้นอยู่กับฮาร์ดแวร์ที่คุณใช้ : • อะแด็ปเตอร์ 8-พอร์ต และ 128-พอร์ต ตรวจสอบว่าโฟลว์คอนโทรลถูกตั้งค่าอย่างถูกต้อง ถ้าเป็นเช่นนั้น รันการวิเคราะห์ และเปลี่ยนฮาร์ดแวร์ถ้าจำเป็น • พอร์ตดั้งเดิม ถ้าปัญหาเกิดขึ้นบนระบบที่ไม่ได้ทำงาน ย้ายเวิร์กโหลดไปยังพอร์ตอื่น ถ้าสามารถแก้ปัญหาให้อัพเกรด firmware ของระบบ • การแก้ปัญหาทั่วไป – ลดพารามิเตอร์ "RECEIVE trigger level" สำหรับพอร์ตนี้จาก 3 เป็น 2 หรือ 1 – ลดความเร็วของสายบนพอร์ตนี้ – ตรวจสอบอุปกรณ์และกระบวนการอื่นเพื่อลองลดเวลาที่ระบบใช้ดับการปิดใช้งานอินเตอร์รัปต์
TTY TTYHOG	TTYHOG overrun	โดยทั่วไปข้อผิดพลาดนี้เกิดจากวิธีของโฟลว์คอนโทรลที่ใช้ระหว่างตัวส่งและตัวรับที่ไม่ตรงกัน ไดรเวอร์ TTY พยายามบอกให้ตัวส่งหยุดชั่วคราว แต่อินพุตยังไม่หยุด ทำให้ข้อมูลถูกทิ้งไป ตรวจสอบวิธีของโฟลว์คอนโทรลที่ถูกตั้งค่าบนแต่ละด้านที่ให้แน่ใจว่าแต่ละตัวใช้วิธีเดียวกัน
TTY Parerr	ข้อผิดพลาด Parity/Framing บนอินพุต	ข้อผิดพลาดนี้จะระบุข้อผิดพลาดของพาริตีบนข้อมูลขาเข้าไปยังพอร์ตอะซิงโครนัสบนพื้นฐานของอักขระต่ออักขระ โดยทั่วไปพารามิเตอร์นี้เกิดจากพารามิเตอร์ในการควบคุมสายไม่ตรงกัน (พาริตี ความเร็วของสาย ขนาดของอักขระ หรือจำนวนของ stop บิต) ระหว่างตัวส่งและตัวรับ พารามิเตอร์การควบคุมสายต้องถูกตั้งเหมือนกันทั้งสองด้านเพื่อที่จะทำการสื่อสาร

ตารางที่ 102. ข้อความแสดงข้อผิดพลาดของ TTY (ต่อ)

ข้อความ	รายละเอียด	หมายเหตุ
TTY Prog PTR	ข้อผิดพลาดภายในของไดรเวอร์	ข้อผิดพลาดนี้ถูกล็อกโดยไดรเวอร์ tty ถ้าตัวชี้ *t_hptr* เป็น null

การเคลียร์พอร์ต TTY ที่ค้าง:

ในตัวอย่างนี้ จะเคลียร์พอร์ตที่ค้าง สมมุติว่าพอร์ต tty ที่คางคือ `tty0`

คุณต้องมีสิทธิของ root เพื่อที่จะสามารถทำโพรซีเดอร์นี้

1. กำหนดว่า tty กำลังจัดการกระบวนการใดๆโดยการพิมพ์ต่อไปนี้ :

   ```
   ps -lt tty0
   ```

   ซึ่งควรจะให้ผลลัพธ์เหมือนดังต่อไปนี้ :

   ```
        F S UID   PID  PPID   C PRI NI ADDR    SZ    WCHAN    TTY  TIME CMD
   240001 S 202 22566  3608   0  60 20 781a   444 70201e44   tty0  0:00 ksh
   ```

   Process ID (PID) ที่นี่คือ 22566 เพื่อ kill กระบวนการนี้ พิมพ์ต่อไปนี้ :

   ```
   kill 22566
   ```

   ต้องแน่ใจว่ากระบวนการถูกเคลียร์แล้ว โดยการใช้คำสั่ง `ps -lt tty0` ถ้ากระบวนการยังอยู่ เพิ่มแฟล็ก `-9` เข้ากับคำสั่ง kill ดังแสดงในตัวอย่างด้านล่าง

   หมายเหตุ: ห้ามใช้อ็อพชัน `-9` เพื่อ kill กระบวนการ slattach การ kill กระบวนการ slattach ด้วยแฟล็ก `-9` อาจทำให้ slip lock ยังคงอยู่ในไฟล์ `/etc/locks` ลบล็อกไฟล์นี้เพื่อลบหลังจาก slattach

   ```
   kill -9 22566
   ```

2. ระบุว่ากระบวนการใดๆพยายามใช้ tty โดยการพิมพ์ต่อไปนี้:

   ```
   ps -ef | grep tty0
   ```

   หมายเหตุ: ถ้าคำสั่ง **ps –ef | grep tty** ให้ผลลัพธ์เหมือนดังต่อไปนี้ :

   ```
   root 19050      1    0    Mar 06       -  0:00 /usr/sbin/getty /dev/tty
   ```

   โดยที่ `"-"` is ถูกแสดงระหว่างวันที่ (`Mar 06`) และเวลา (`0:00`) แสดงว่า tty ไม่ได้ใช้สายเคเบิลที่ถูกต้อง สถานะนี้ระบุว่ากระบวนการลอกอินของระบบ (getty) พยายามเปิด tty นี้ และกระบวนการการเปิดหยุดทำงานเนื่องจากสัญญาณ RS-232 Data Carrier Detect (DCD) ไม่ถูกใช้ คุณสามารถแก้ไขได้โดยการใช้อะแด็ปเตอร์ null โมเด็มในการเดินสายเคเบิล เมื่อ getty สามารถเปิดพอร์ต tty `"-"` จะถูกแทนที่โดยหมายเลขของ tty สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสายเคเบิล ดูที่ “เชื่อมต่อโมเด็มกับสายเคเบิลที่เหมาะสม” ในหน้า 639

   หมายเหตุ: คำสั่งต่อไปนี้สามารถถูกใช้เพื่อปิดการใช้งานกระบวนการล็อกอินบน `tty0`

   ```
   pdisable tty0
   ```

   ถ้ากระบวนนี้ถูกเคลียร์เรียบร้อยแล้ว แต่ tty ยังคงไม่ตอบสนอง ให้ทำขั้นตอนต่อไป

3. พิมพ์คำสั่งต่อไปนี้:

   ```
   fuser -k /dev/tty0
   ```

ซึ่งจะเคลียร์กระบวนการใดๆที่สามารถพบว่ารันอยู่บนพอร์ตและแสดง PID ถ้า tty ยังใช้ไม่ได้ ทำขั้นตอนต่อไป

4. ใช้คำสั่ง **strreset** เพื่อ flush ข้อมูลขาออกจากพอร์ตที่หยุดทำงานเนื่องจากข้อมูลไม่สามารถถูกส่งเนื่องจากการเชื่อมต่อไปยังด้านรีโมตหายไป

   หมายเหตุ: ถ้าคำสั่ง **strreset** สามารถแก้ไขพอร์ตที่ไม่ทำงาน แสดงว่าพอร์ตมีปัญหาเกี่ยวกับสายเคเบิลหรือการตั้งค่า เนื่องจากการเชื่อมต่อไปยังด้านรีโมตที่หายไปควรทำให้ข้อมูลที่ถูกบัฟเฟอร์ถูก flush โดยอัตโนมัติ

   คุณต้องตรวจสอบหมายเลขหลักหรือรองของอุปกรณ์สำหรับ tty ก่อนโดยการพิมพ์ต่อไปนี้ :

   ```
   ls -al /dev/tty0
   ```

   ผลลัพธ์ของคุณ ควรมีลักษณะคล้ายกับตัวอย่างต่อไปนี้:

   ```
   crw-rw-rw-    1 root     system    18,   0 Nov  7  06:19 /dev/tty0
   ```

   นี่จะระบุว่า tty0 มีหมายเลขหลักของอุปกรณ์เป็น 18 และหมายเลขรองของอุปกรณ์เป็น 0 ระบุหมายเลขเหล่านี้เมื่อใช้คำสั่ง **strreset** ดังต่อไปนี้ :

   ```
   /usr/sbin/strreset -M 18 -m 0
   ```

   ถ้า tty ยังใช้ไม่ได้ ทำขั้นตอนต่อไป

5. ถอดและต่อสายเคเบิลใหม่จากพอร์ต tty ที่หยุดทำงาน AIX จะใช้สัญญาณ Data Carrier Detect (DCD) เพื่อระบุการมีอยู่ของอุปกรณ์ที่ต่ออยู่กับพอร์ต โดยการดร็อบ DCD ถอดและใส่สายเคเบิลจะมีหลายกรณีที่จะช่วยเคลียร์กระบวนการที่หยุดทำงาน

   เพื่อกำหนดตำแหน่งของพอร์ตที่ tty ถูกตั้งค่า พิมพ์คำสั่งต่อไป :

   ```
   lsdev -Cl tty0
   ```

   ผลที่ได้ควรดูเหมือนดังต่อไปนี้ :

   ```
   tty0    Available   00-00-S1-00  Asynchronous Terminal
   ```

   คอลัมน์ที่สามในเอาต์พุตข้างบนระบุโค้ดของตำแหน่งของ tty ในตัวอย่างนี้ S1 จะระบุว่าซีเรียลพอร์ตถูกตั้งค่าสำหรับซีเรียลพอร์ 1 แบบดั้งเดิม สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการแปลโค้ดตำแหน่ง ดูที่ โค้ดตำแหน่งของอุปกรณ์ ใน *การจัดการระบบปฏิบัติการและอุปกรณ์*

   ถ้า tty ยังใช้ไม่ได้ ทำขั้นตอนต่อไป

6. Flush พอร์ตโดยใช้ **stty-cxma** พิมพ์ดังต่อไปนี้:

   ```
   /usr/lbin/tty/stty-cxma flush tty0
   ```

   คำสั่งนี้ใช้กับ tty ที่ถูกตั้งค่าบนพอร์ตของอะแด็ปเตอร์ 8-พอร์ต และ 128-พอร์ต อย่างไรก็ตาม ในหลายกรณีมันสามารถถูกใช้เพื่อ flush พอร์ต tty อื่นด้วย

   ถ้า tty ยังใช้ไม่ได้ ทำขั้นตอนต่อไป

7. บนคีย์บอร์ดของเทอร์มินัลที่หยุดทำงาน กดคีย์ Ctrl ค้างไว้และกด Q นี่จะเริ่มใช้งานเอาต์พุตที่ถูกหยุดชั่วคราวโดยการส่งอักขระ **Xon**

   ถ้า tty ยังใช้ไม่ได้ ทำขั้นตอนต่อไป

8. บางครั้งโปรแกรมจะเปิดพอร์ต tty แก้ไขบางแอ็ตทริบิวต์ และปิดพอร์ตโดยไม่รีเซ็ตแอ็ตทริบิวต์เป็นสถานะแรกเริ่มของมัน เพื่อแก้ไข ทำให้ tty อยู่ในสถานะ DEFINED และจากนั้นทำให้มันพร้อมใช้งานโดยการพิมพ์ต่อไปนี้ :

   ```
   rmdev -l tty0
   ```

คำสั่งนี้จะปล่อยให้ข้อมูลที่เกี่ยวข้องกับ tty อยู่ในฐานข้อมูล แต่ทำให้ tty ไม่พร้อมใช้งานบนระบบ
คำสั่งต่อไปนี้จะเปิดใช้งาน tty อีกครั้ง:

```
mkdev -l tty0
```

ถ้า tty ยังใช้ไม่ได้ พิจารณาย้ายอุปกรณ์ไปยังพอร์ตอื่นและตั้งค่า tty ที่ตำแหน่งนั้นจนกว่าระบบจะสามารถถูกรีบูต ถ้าการรีบูตไม่สามารถเคลียร์พอร์ต เป็นไปได้ว่าคุณมีปัญหาเกี่ยวกับฮาร์ดแวร์ ตรวจสอบรายงานข้อผิดพลาดสำหรับปัญหาฮาร์ดแวร์ของพอร์ต โดยการใช้คำสั่งต่อไปนี้ :

```
errpt -a | pg
```

บางคำสั่งก่อนหน้านี้อาจใช้ไม่ได้ และมันจะให้ข้อผิดพลาดของวิธีที่ระบุว่าอุปกรณ์ไม่ว่าง ซึ่งเนื่องมาจากกระบวนการรันอยู่บน tty ถ้าไม่มีขั้นตอนใดข้างบนสามารถแก้ไขพอร์ตที่ไม่ทำงาน วิธีสุดท้ายคือ รีบูตระบบ AIX และ flush เคอร์เนลที่เพื่อกระบวนการจะได้หายไป

โมเด็ม

โมเด็มให้การสื่อสารแบบซีเรียลผ่านสายโทรศัพท์แบบธรรมดา แนวคิดของโมเด็มจะรวมมาตรฐาน การตั้งค่าโมเด็มทั่วไป และคำแนะนำการตั้งค่าที่ระบุสำหรับโมเด็มที่เป็นที่นิยม

โมเด็ม เป็นอุปกรณ์ที่ให้คุณสามารถเชื่อมต่อคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องผ่านสายโทรศัพท์ธรรมดา ระบบโทรศัพท์ปัจจุบันไม่สามารถส่งการเปลี่ยนแปลงของโวลท์เตจที่ต้องการสำหรับการเชื่อมต่อแบบดิจิตอลโดยตรง โมเด็มสามารถข้ามข้อจำกัดนี้โดยการเปลี่ยนข้อมูลแบบดิจิตอลเป็นโทนของเสียงเพื่อจะส่งข้อมูลผ่านสายโทรศัพท์ และโดยการแปลงโทนเหล่านั้นกลับเป็นข้อมูลแบบดิจิตอลเมื่อได้รับแล้ว โมเด็มมักจะถูกใช้กับ Basic Network Utilities (BNU) หรือการใช้งานของ UNIX-to-UNIX Copy Program (UUCP) โมเด็มความเร็วสูง (14,400 bps หรือมากกว่า) ยังสามารถใช้กับ Serial Line Interface Protocol (SLIP) เพื่อให้การเชื่อมต่อแบบ Transmission Control Protocol/Internet Protocol (TCP/IP) ด้วย

โดยมาก คำว่า *baud* ถูกใช้เพื่ออ้างถึงความเร็วของโมเด็มแทน bps Baud โดยทั่วไปเป็นการวัดอัตราการแปลงข้อมูล ในโมเด็มแบบเก่า จะมีเพียง 1 บิตที่ถูกเข้ารหัสในแต่ละการเปลี่ยนแปลงของสัญญาณ ดังนั้นอัตรา baud ของโมเด็มจะเท่ากับความเร็วของโมเด็ม โมเด็มที่ทำงานที่ความเร็วที่สูงกว่า โดยทั่วไปยังทำงานที่ 2,400 (หรือแม้แต่ 1,200) baud และเข้ารหัส 2 บิตหรือมากกว่าต่อการเปลี่ยนแปลงของสัญญาณ อัตรา bps ของโมเด็มถูกคำนวณโดยการคูณจำนวนของบิตข้อมูลต่อสัญญาณด้วย baud (ตัวอย่างเช่น 2,400 baud x 6 บิตต่อสัญญาณที่เปลี่ยนแปลง = 14,400 บิตต่อวินาที) โมเด็มสมัยใหม่ส่วนมากสามารถสื่อสารที่ความเร็วต่างๆ (ตัวอย่างเช่น 28,800, 14,400, 9,600, 7,800, 4,800, และ 2,400 bps)

มาตรฐานการสื่อสารโทรคมนาคม

ความเร็วแบบเก่าที่ 300, 1,200 และ 2,400 bps ถูกกำหนดไว้อย่างดี อย่างไรก็ตามผู้ผลิตโมเด็มเริ่มประดิษฐ์วิธีเพื่อให้ได้ความเร็วที่สูงขึ้น แต่ละผู้ผลิตโมเด็มเริ่มใช้วิธีเฉพาะที่ไม่สอดคล้องกับโมเด็มจากผู้ผลิตอื่น วันนี้ ITU-TSS (ซึ่งเดิมคือ the United Nations Consultative Committee for International Telephony and Telegraphy, abbreviated CCITT) ได้กำหนดมาตรฐานสำหรับการสื่อสารความเร็วสูงส่วนใหญ่

แม้ว่าโมเด็มความเร็วสูงจะช้ากว่าวิธีอื่นของการสื่อสารของคอมพิวเตอร์มาก โมเด็มความเร็วสูงสามารถทำงานที่ 28,800 bps แต่การเชื่อมต่อ Ethernet ทำงานที่ 10,000,000 bps เพื่อเพิ่มประมาณงานของข้อมูล โมเด็มความเร็วสูงโดยทั่วไปจะใช้อัลกอริทึมการบีบอัดข้อมูลหนึ่งวิธีหรือมากกว่า อัลกอริทึมเหล่านี้สามารถเพิ่มประมาณงานของโมเด็มความเร็วสูงได้ถึง 57,600 bps (ถ้าอัตราของข้อมูลคือ 14,400 bps) หรือ 115,200 bps (ถ้าอัตราของข้อมูลคือ 28,800 bps) โปรดสังเกตว่าอัลกอริทึมการบีบอัดเหล่านี้จะไวต่อข้อมูลที่ถูกส่ง ถ้าข้อมูลถูกบีบอัดแล้ว (ตัวอย่างเช่น โดยคำสั่ง **compress**) วิธีการบีบอัดข้อมูลของโมเด็มความเร็วสูงจะไม่มีประโยชน์หรือมีน้อย และอาจลดปริมาณงานของข้อมูล เมื่อใช้โมเด็มที่ใช้เทคโนโลยีการบีบอัดข้อมูล

ความเร็วของการเชื่อมต่อ data terminal equipment/data circuit-terminating equipment (DTE/DCE) ระหว่างคอมพิวเตอร์และโมเด็มจะเท่ากับหรือมากกว่าอัตราของข้อมูลปกติของการเชื่อมต่อระหว่างโมเด็ม ตัวอย่างเช่น ด้วย V.32*bis* โมเด็ม ด้วยการบีบอัดข้อมูล V.42*bis* อัตราของข้อมูลของโมเด็ม (ความเร็วที่โมเด็มสื่อสารโดยใช้สายโทรศัพท์) คือ 14,400 bps เมื่อการบีบอัด V.42*bis* แอ็คทีฟ ปริมาณงานของข้อมูลจริงสามารถสูงถึง 57,600 bps เพื่อให้ได้ปริมาณงานที่มากขึ้นที่ให้โดยการบีบอัดข้อมูล ความเร็วของลิงก์ระหว่างคอมพิวเตอร์และโมเด็มควรตั้งเป็น 57,600 bps

ITU-TSS กำหนดมาตรฐานสำหรับการสื่อสารความเร็วสูง รวมถึงอัลกอริทึมการบีบอัดข้อมูล มาตรฐาน ITU-TSS โดยทั่วไปจะชื่อ V.*nn* โดยที่ *nn* เป็นตัวเลข นอกจากนี้ ที่แตกต่างจากมากตราฐานเพียงเล็กน้อยคือ Microcom Networking Protocol (MNP) มีให้ใช้ในเวอร์ชัน (เรียกว่าคลาส) 1-9 MNP เป็นโปรโตคอลที่มีประสิทธิภาพสูง และความเร็วสูง ที่มีให้ใช้งานเร็วๆนี้และกลายเป็นมาตรฐานโดยนัยก่อนที่จะเป็นมาตรฐานของ ITU-TSS

การส่งข้อมูลแบบ Full และ half duplex:

เมื่อเรียนรู้เกี่ยวกับมาตรฐานการสื่อสารโทรคมนาคม มความสำคัญที่จะต้องเข้าใจความแตกต่างระหว่างการส่งข้อมูลแบบ half duplex และ full duplex

ในการส่งข้อมูลแบบ *half duplex* (HDX) แพ็กเก็ตของข้อมูลจะถูกส่งโดยหนึ่งระบบและถูกรับบโดยระบบอื่น Another data packet cannot be sent until the receiving system sends an acknowledgment back to the sender.

ในการส่งข้อมูลแบบ *full duplex* (FDX) ทั้งระบบผู้ส่งและระบบผู้รับจะสื่อสารกันแบบพร้อมกัน อีกนัยหนึ่ง ทั้งสองโมเด็มสามารถส่งและรับข้อมูลในเวลาเดียวกัน นี่หมายความว่าโมเด็มสามารถรับแพ็กเก็ตของข้อมูลขณะที่รับการตอบรับจากผู้อื่น

มาตรฐานการสื่อสาร ITU-TSS:

บางมาตรฐานการสื่อสารทั่วไปถูกกำหนดโดย ITU-TSS ดังอธิบายในที่นี้

โปรดสังเกตว่าเฉพาะลิสต์บางส่วน สำหรับลิสต์ที่สมบูรณ์ อ้างถึงอินเตอร์เน็ตเว็บไซต์สำหรับ the International Telecommunication Union

ไอเท็ม	คำอธิบาย
V.29	มาตรฐาน ITU-TSS สำหรับการสื่อสาร half-duplex 9600 bps
V.32	มาตรฐาน ITU-TSS สำหรับการสื่อสาร full-duplex 9600 bps
V.32*bis*	มาตรฐาน ITU-TSS สำหรับการสื่อสาร 14,400 V.32*bis* เป็นเวอร์ชันที่ปรับปรุงของมาตรฐาน V.32
V.34	มาตรฐาน ITU-TSS สำหรับการสื่อสาร 33,600 โปรดสังเกตว่ามาตรฐานนี้จะได้อัตราข้อมูลถึง 33,600 bpsโดยใช้การเข้ารหัสหลายบิตแทนที่จะใช้การบีบอัดข้อมูลที่ถูกใช้โดย MNP Class 9 มาตรฐานนี้เคยถูกอ้างถึงเป็น V.*fast*.
V.42	โพรซีเดอร์การแก้ไขข้อผิดพลาด ITU-TSS สำหรับ DCEs โดยใช้การแปลงอะซิงโครนัสเป็นซิงโครนัส
V.42*bis*	มาตรฐานการบีบอัดข้อมูล ITU-TSS ที่ถูกปรับปรุง

Microcom Networking Protocol:

มาตรฐานที่อื่นโดยพฤตินัย คือ **Microcom Networking Protocol (MNP)** ซึ่งถูกพัฒนาโดย Microcom, Inc.

มีในเวอร์ชัน (เรียกว่าคลาส) 1-9, **MNP** เป็นโปรโตคอลที่มีประสิทธิภาพสูง และความเร็วสูง ที่มีให้ใช้ก่อนที่จะเป็นมาตรฐานของ ITU-TSS โดยใช้ **MNP** ข้อผิดพลาดในแพ็กเก็ตของข้อมูลที่ถูกส่งจะถูกตรวจจับโดยรีโมตโมเด็ม ทำให้มันร้องขอให้ส่งแพ็กเก็ตของข้อมูลที่มีข้อผิดพลาดอีกครั้ง ความสามารถในการตรวจพบและแก้ไขข้อผิดพลาดของข้อมูลได้อย่างรวดเร็วทำให้ **MNP** เป็นหนึ่งในโปรโตคอลที่เป็นที่นิยมมากที่สุดในวันนี้

ตารางต่อไปนี้ระบุมาตรฐานการสื่อสาร **MNP**

ไอเท็ม	คำอธิบาย
MNP คลาส 1	วิธีการถ่ายโอนข้อมูลแบบอะซิงโครนัส half-duplex และ byte-oriented ที่มีประสิทธิภาพที่น่าเชื่อถือประมาณ 70% มาตรฐานนี้ไม่ค่อยมีอยู่ในโมเด็มสมัยใหม่
MNP คลาส 2	full-duplex ที่เสริมกับ **MNP คลาส 1** ซึ่งไม่ค่อยมีอยู่ในโมเด็มสมัยใหม่เช่นเดียวกัน
MNP คลาส 3	วิธีการถ่ายโอนข้อมูลแบบซิงโครนัส bit-oriented และ full-duplex ที่มีประสิทธิภาพที่น่าเชื่อถือประมาณ 108% ประสิทธิภาพที่มากกว่า 100% ถูกยอมรับได้ เนื่องจาก start/stop บิตที่ถูกต้องการสำหรับการเชื่อมต่อแบบอะซิงโครนัสจะถูกตัดออก DTE/DCE ระหว่างโมเด็มและระบบยังคงเป็นแบบอะซิงโครนัส
MNP คลาส 4	เป็นการปรับปรุง **MNP คลาส 3** เพื่อรวมกลไกสำหรับเปลี่ยนขนาดของแพ็กเก็ต (การประกอบแพ็ตเก็ตที่เปลี่ยนแปลงได้) และการกำจัดโอเวอร์เฮดของผู้ดูแลระบบที่ซ้ำซ้อน (การเพิ่มประสิทธิภาพของเฟสข้อมูล) โมเด็ม **MNP คลาส 4** ให้ประสิทธิภาพประมาณ 120%
MNP คลาส 5	จะรวมการบีบอัดข้อมูลกับคุณลักษณะของคลาส 4 โมเด็ม **MNP คลาส 5** ให้ประสิทธิภาพ 200%
MNP คลาส 6	ยอมให้การรวมกันของเทคนิคการแปลงข้อมูลที่ไม่สามารถเข้ากันได้อยู่ในโมเด็มเดียว (universal link negotiation) นี่ทำให้โมเด็ม **MNP Class 6** สามารถเริ่มการสื่อสารที่ความเร็วที่ต่ำกว่า และต่อรองการเปลี่ยนแปลงเป็นความเร็วที่สูงกว่า **Class 6** ยังรวม statistical duplexing scheme ซึ่งจะจัดสรรการแปลงเซอร์วิส half-duplex เป็นการจำลอง full-duplex แบบไดนามิก สนับสนุนคุณลักษณะทั้งหมดของ **MNP Class 5**
MNP คลาส 7	รวมการบีบอัดข้อมูลที่ได้รับการปรับปรุง ถูกรวมกับคลาส 4 มีประสิทธิภาพ 300% ที่สามารถเชื่อถือได้
MNP คลาส 8	ไม่ได้ใช้
MNP คลาส 9	รวมการบีบอัดข้อมูลที่ได้รับการปรับปรุงกับเทคโนโลยี V.32 เพื่อให้ได้อัตราของข้อมูลสูงถึง 28,800 bps

ข้อพิจารณาเกี่ยวกับโมเด็ม

ข้อกำหนดของโมเด็มอินเตอร์เฟสสำหรับผู้ใช้ทั่วไปอาจแตกต่างกัน

คอนฟิกูเรชันของโมเด็มี้มากับระบบปฏิบัติการนี้จะแตกต่างจากของคอมพิวเตอร์ส่วนบุคคล (PC) หรือเวิร์กสเตชัน

โมเด็มที่ได้รับการสนับสนุน:

โมเด็มใดๆที่สอดคล้องกับ EIA 232 และสามารถให้ผลลัพธ์เพื่อตอบสนองต่อคำสั่งสามารถถูกเชื่อมต่อกับระบบปฏิบัติการนี้

การจัดการกับ Data Carrier Detect:

เซิร์ฟเวอร์ใช้สัญญาณ Data Carrier Detect (DCD) เพื่อมอนิเตอร์สถานะที่เป็นจริงของโมเด็ม

ถ้าสัญญาณ DCD บนพอร์ตของโมเด็มเป็น "high" เซิร์ฟเวอร์จะเข้าใจว่าโมเด็ทกำหลังถูกใช้ ดังนั้นมีความจำเป็นต้องรู้ว่าสถานการณ์ใดที่สัญญาณนี้ถูกบังคับให้อยู่ในสถานะ "high" สัญญาณ DCD สามารถถูกทำให้เป็น high ด้วยสาเหตุต่อไปนี้ :

- การใช้ **clocal** ในแอ็ตทริบิวต์ stty สำหรับฟิลด์ runtime บนพาเนล SMIT **TTY Configuration**
- ฟิลด์ Ignore Carrier Detect ถูกตั้งเป็น **enable** บนพาเนล SMIT **TTY Configuration** สำหรับ ttys ที่เชื่อมกับ 128-พอร์ตอะแด็ปเตอร์
- โมเด็มจะบังคับ DCD เป็น high โดยคำสั่ง AT หรือสวิตช์
- พอร์ต tty พร้อมใช้งานโดยแอ็พพลิเคชัน

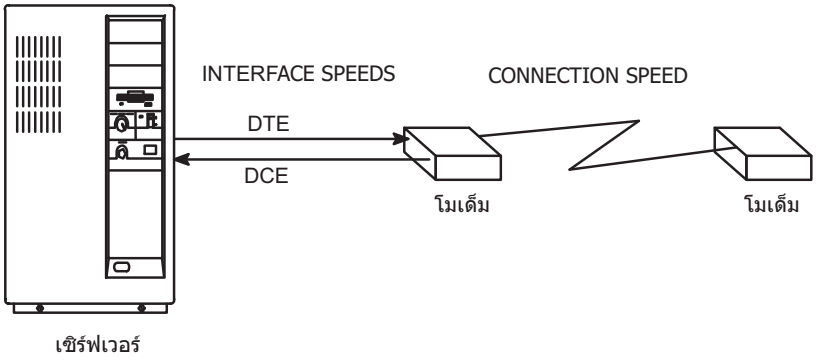
หมายเหตุ: เมื่อโมเด็มทำการเชื่อมต่อกับโมเด็มอื่น โมเด็มจะยกระดับสัญญาณ CD ค่าดีฟอลต์ส่วนใหญ่ของโมเด็มจะตั้งสัญญาณนี้เป็น "high" ตลอดเวลาแม้เมื่อโมเด็มไม่ถูกใช้งาน CD ไม่ควรถูกบังคับให้เป็น "high"

ความเร็วของ Data Terminating Equipment หรือ Data Circuit-Terminating Equipment:

Data Terminating Equipment (DTE) และ Data Communication Equipment (DCE) ถูกใช้เพื่ออธิบายกลุ่มของฮาร์ดแวร์ที่แตกต่างกัน 2 กลุ่ม

คำว่า DTE ถูกใช้เป็นหลักสำหรับอุปกรณ์ที่แสดงข้อมูลของผู้ใช้ มันยังรวมอุปกรณ์ใดๆที่เก็บหรือสร้างข้อมูลสำหรับผู้ใช้ หน่วยของระบบ เทอร์มินัล และดครื่องพิมพ์ จัดอยู่ในประเภทของ DTE

DCE จะรวมอุปกรณ์ใดๆที่สามารถถูกใช้เพื่อเข้าถึงระบบผ่านสายการสื่อสารโทรคมนาคม รูปแบบทั่วไปของ DCE ส่วนใหญ่คือ โมเด็มและมัลติเพล็กเซอร์



รูปที่ 39. ข้อพิจารณาเกี่ยวกับความเร็วของโมเด็ม

เมื่อการสื่อสารแบบซีเรียลบนระบบปฏิบัติการนี้เกี่ยวข้องกับโมเด็ม ดังแสดงขางบน จะมีข้อพิจารณาหลัก 3 ข้อ :

- ความเร็วของอินเตอร์เฟสของ DTE (เซิร์ฟเวอร์ถึงโมเด็ม) นี่เป็นความเร็วที่เซิร์ฟเวอร์สื่อสารกับโมเด็ม
- ความเร็วของอินเตอร์เฟสของ DCE (โมเด็มถึงเซิร์ฟเวอร์) บางครั้งเรียกว่า "ความเร็วของอินเตอร์เฟสของซีเรียลพอร์ต" นี่เป็นความเร็วที่โมเด็มสื่อสารกับเซิร์ฟเวอร์
- ความเร็วการเชื่อมต่อ (โมเด็มถึงโมเด็ม) นี่เป็นความเร็วที่โมเด็มสื่อสาร (หรือคุย) กับโมเด็มอื่น

โมเด็มสมัยใหม่ ความเร็วสูงยอมให้ความเร็วอินเตอร์เฟสของ DCE แตกต่างจากความเร็วของการเชื่อมต่อ นี่จะยอมให้ความเร็วของ DTE ถูกล็อกที่อัตรา baud เดียวขณะที่ยอมให้ความเร็วของการเชื่อมต่อเปลี่ยนแปลง ขึ้น หรือ ลง ตามต้องการสำหรับการสื่อสารระหว่างโมเด็มที่เหมาะสม

โมเด็มสมัยใหม่ ความเร็วสูงเก็บข้อมูลที่จะถูกส่งไปยังเซิร์ฟเวอร์ในบัฟเฟอร์และส่งมันเมื่อระบบสามารถรับมัน มันยังสามารถเก็บข้อมูลที่จะถูกส่งไปยังโมเด็มอื่นในบัฟเฟอร์และส่งมันเมื่อรีโมตสามารถที่จะรับมัน วิธีการส่งข้อมูลชนิดนี้ต้องการให้โมเด็มและเซิร์ฟเวอร์ใช้ *โฟลว์คอนโทรล*

สัญญาณควบคุมโมเด็ม:

โมเด็มมักถูกใช้เพื่อเริ่มต้นการรับการโทร ดังนั้นมันจะมีความสำคัญที่จะโปรแกรมโมเด็มเพื่อจะสื่อสารที่ความเร็วสูงสุดที่จะเป็นไปได้ และรีเซ็ตตัวเองเป็นสถานะที่รู้จักหลังจากการเชื่อมต่อถูกหยุด

เซิร์ฟเวอร์จะปิดเปิดสัญญาณ Data Terminal Ready (DTR) จาก on เป็น off เพื่อบอกให้โมเด็มยกเลิกการเชื่อมต่อ โมเด็มส่วนใหญ่สามารถถูกตั้งค่าเพื่อให้รีเซ็ตตัวเองเมื่อการเปลี่ยนแปลงของ DTR จาก on เป็น off เกิดขึ้น

หมายเหตุ: tty สามารถถูกตั้งค่าเพื่อที่จะไม่ดร็อบ DTR โดยการปิดการใช้งานแฟล็ก **hupcl** ในแอ็ตทริบิวต์ stty run-time

สำหรับการเชื่อมต่อระหว่างเซิร์ฟเวอร์และโมเด็มเพื่อให้ทำงานอย่างเต็มที่ สายเคเบิลต้องมีคุณสมบัติดังต่อไปนี้ :

- มันต้องต้องเป็นไปตามข้อกำหนด
- มันควรมีการชิลด์ที่เหมาะสม
- ควรมีสัญญาณต่อไปนี้ต่อไปนี้ : RxD, TxD, RTS, CTS, SG, DCD และ DTR

หมายเหตุ: 16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์ไม่สนับสนุนสัญญาณ RTS และ CTS ดังนั้นจำเป็นต้องใช้ RTS/CTS ฮาร์ดแวร์โฟลว์คอนโทรลกับอะแด็ปเตอร์นี้

ถ้าข้อมูลไบนารีจะถูกถ่ายโอนโดยใช้โมเด็มบนอะแด็ปเตอร์นี้ โปรโตคอลการถ่ายโอนไฟล์ที่ตรวจพบข้อมูลที่ไม่ถูกต้องและส่งข้อมูลที่หายไปใหม่ (ตัวอย่างเช่น Xmodem, zmodem, Kermit และ UUCP) ควรถูกใช้

ต่อไปนี้จะคำอธิบายสัญญาณที่ถูกใช้โดยเซิร์ฟเวอร์ :

Signal	รายละเอียด
FG	Frame Ground ขาที่ 1 ของข้อกำหนด EIA 232D ที่มีให้สำหรับการชีลด์สายเคเบิล การใช้งานอย่างถูกต้อง สัญญาณจะต่อกับขาที่ 1 บนด้านหนึ่งของสายเคเบิลเท่านั้น และถูกเชื่อมกับปลอกโลหะรอบๆสาย
TxD	ส่งข้อมูล ขาที่ 2 ของข้อกำหนดของ EIA 232D ข้อมูลจะถูกส่งบนสัญญาณนี้ ถูกควบคุมโดยเซิร์ฟเวอร์
RxD	รับข้อมูล ขาที่ 3 ของข้อกำหนดของ EIA 232D ข้อมูลถูกได้รับบนสัญญาณนี้ ถูกควบคุมโดยโมเด็ม ซึ่งจะถูกส่งโดยโมเด็ม
RTS	Request To Send ขาที่ 4 ของข้อกำหนดของ EIA 232D ถูกใช้เมื่อ RTS/CTS โฟลว์คอนโทรลถูกเปิดใช้งาน สัญญาณนี้จะเป็น high เมื่อระบบพร้อมที่จะส่งข้อมูลและถูกดร็อบเมื่อระบบต้องการให้โมเด็มหยุดส่งข้อมูล
CTS	Clear To Send ขาที่ 5 ของข้อกำหนดของ EIA 232D ถูกใช้เมื่อ RTS/CTS โฟลว์คอนโทรลถูกเปิดใช้งาน สัญญาณนี้จะเป็น high เมื่อโมเด็มพร้อมที่จะส่งหรือรับข้อมูล มันจะถูกดร็อบเมื่อโมเด็มต้องการให้เซิร์ฟเวอร์หยุดส่งข้อมูล ถูกควบคุมโดยโมเด็ม
DSR	Data Set Ready ขาที่ 6 ของข้อกำหนดของ EIA 232D ส่งสัญญาณให้เซิร์ฟเวอร์ว่าโมเด็มอยู่ในสถานะที่พร้อมใช้งาน ถูกควบคุมโดยโมเด็ม
SG	สัญญาณกราวด์ ขาที่ 7 ของข้อกำหนดของ EIA 232D สัญญาณนี้ให้โวลท์เตจเพื่ออ้างอิงสำหรับสัญญาณอื่นๆ
DCD	Data Carrier Detect ขาที่ 8 ของข้อกำหนดของ EIA 232D จะให้สัญญาณกับเซิร์ฟเวอร์ว่าโมเด็มกำลังเชื่อมต่อกับโมเด็มอื่น เมื่อสัญญาณนี้เป็น high โปรแกรมที่รันบนเซิร์ฟเวอร์จะสามารถเปิดพอร์ต ถูกควบคุมโดยโมเด็ม
DTR	Data Terminal Ready ขาที่ 20 ของข้อกำหนดของ EIA 232D จะให้สัญญาณกับโมเด็มว่าเซิร์ฟเวอร์เปิดอยู่และพร้อมที่จะรับการเชื่อมต่อ สัญญาณนี้จะถูกดร็อบเมื่อเซิร์ฟเวอร์ต้องการให้โมเด็มดร็อบการเชื่อมต่อกับโมเด็มอื่น มันจะเป็น high เมื่อพอร์ตถูกเปิด ถูกควบคุมโดยเซิร์ฟเวอร์
RI	Ring Indicate ขาที่ 22 ของข้อกำหนดของ EIA 232D จะให้สัญญาณกับเซิร์ฟเวอร์ว่าโมเด็มกำลังได้รับการเรียก มันถูกใช้น้อย และไม่ต้องการสำหรับการทำงานทั่วไป ถูกควบคุมโดยโมเด็ม

การติดตั้งสายโมเด็ม

ตารางเหล่านี้แสดงสรุปของข้อมูลของสายเคเบิลที่ต้องการเพื่อเชื่อมต่อกับโมเด็มที่ถูกต้องกับคอนโทรลเลอร์แบบซีเรียลใดๆ

อะแด็ปเตอร์/คอนโทรลเลอร์	IBM หมายเลขชิ้นส่วน
หมายเลขเดิม (S1 หรือ S2)	00G0943*, 6326741
คอนโทรลเลอร์แบบ 2-พอร์ต	00G0943*, 6326741
คอนโทรลเลอร์แบบ 8-พอร์ต	6323741
คอนโทรลเลอร์แบบ 128-พอร์ต	43G0935, 6323741

IBM หมายเลขชิ้นส่วน	รายละเอียด	ความยาวเป็นฟุต
00G0943*	Serial Port Jumper (pigtail)	.33
6323741	อะซิงโครนัส	10
43G0935	สายเคเบิลแปลง RJ-45 เป็น DB25	2

*หมายเลขชิ้นส่วนนี้ไม่ต้องการสำหรับบางชนิดของเครื่อง

การตั้งค่าอุปกรณ์ TTY บนระบบปฏิบัติการ

ใช้ System Management Interface Tool (SMIT) เพื่อกำหนดพอร์ตของ tty สำหรับการเชื่อมกับอุปกรณ์

ฟิลด์ส่วนใหญ่จะใช้สำหรับชนิดของอุปกรณ์ทั่วไป ฟิลด์เดียวที่สามารถมีผลกับโมเด็มคือ Enable LOGIN โดยมีค่าต่อไปนี้ :

ไอเท็ม	คำอธิบาย
DISABLE	ไม่มีกระบวนการ getty ถูกรันบนพอร์ต ใช้การตั้งค่านี้สำหรับพอร์ตที่โทรออกอย่างเดียวของโมเด็ม
ENABLE	กระบวนการ getty ถูกรันบนพอร์ต ใช้การตั้งค่านี้สำหรับโมเด็มที่โทรเข้าเท่านั้น
SHARE	กระบวนการ getty จะรันบนพอร์ต แต่กระบวนการ getty จะยอมให้โปรแกรมสามารถโทรเข้าและออกบนพอร์ตนี้โดยไม่ต้องมีการเปลี่ยนจากปิดใช้งานเป็นเปิดใช้งานแบบแมนวล ใช้การตั้งค่านี้สำหรับการใช้พอร์ตแบบสองทิศทาง
DELAY	getty ถูกรับบนพอร์ตในโหมดสองทิศทาง แต่ไม่มีข่าวสารถูกส่งจนกว่ากระบวนการ getty จะได้รับการกดคีย์จากผู้ใช้

ฟิลดที่ระบุสำหรับ 128-พอร์ต อะซิงโครนัส อะแด็ปเตอร์ :

ไอเท็ม	คำอธิบาย
บังคับ Carrier หรือไม่สนใจ Carrier Detect	disable*
ทำการประมวลผล Cooked ในอะแด็ปเตอร์	disable

หมายเหตุ: การตั้งค่านี้จะถูกระบุโดยเครื่องหมายดอกจัน (*) ถูกตั้งเป็นถูกปิดใช้งานถ้าใช้ตัวเชื่อมต่อ RJ-45 แบบ 10 ขา การตั้งค่านี้ควรถูกเปิดใช้งานถ้าใช้ตัวเชื่อมต่อ RJ-45 แบบ 8 ขา

เชื่อมต่อโมเด็มกับสายเคเบิลที่เหมาะสม

ขั้นตอนแรกตั้งการติดตั้งโมเด็มคือการเชื่อมต่อโมเด็มกับสายเคเบิลที่เหมาะสม

หมายเลขชิ้นส่วนและคำอธิบายของมันจะถูกลิสต์ด้ายล่าง

6323741

สายเคเบิลแบบอะซิงค์ EIA-232 ใช้เพื่อเชื่อมอุปกรณ์แบบอะซิงโครนัสทั้งหมด บางครั้งถูกใช้กับการประกอบสายเคเบิลอื่น

59F3740

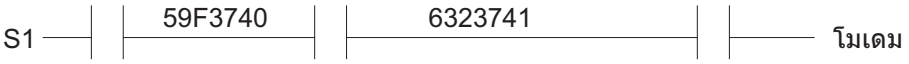
ตัวเชื่อมต่อ D-shell แบบ 10 เป็น 25-pin ใช้เพื่อเชื่อมสายเคเบิลแบบอะซิงโครนัส 6323741 กับพอร์ตซีเรียลแบบเดิม S1 และ S2 ดังแสดงในรูปต่อไปนี้

รูปที่ 40. ตัวเชื่อมต่อ 10 เป็น 25-pin

ภาพนี้แสดงตัวเชื่อมต่อ 10 เป็น 25-pin

ต่อไปนี้เป็นตัวอย่างของการเชื่อมสายเคเบิล :

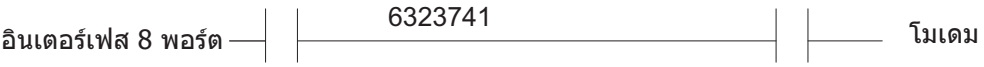
1. เพื่อเชื่อมโมเด็มกับซีเรียลพอร์ตแบบเดิม S1 ใช้สายเคเบิลต่อไปนี้ :



รูปที่ 41. การประกอบสายเคเบิลจากโมเด็มไปยังซีเรียลพอร์ตแบบเดิม

ภาพนี้แสดงสายเคเบิล 59F3740 บนด้านของซีเรียลพอร์ต และ 6323741 บนด้านของโมเด็ม

2. เพื่อเชื่อมต่อโมเด็มกับการประกอบสายเคเบิลอะแด็ปเตอร์อะซิงค์อะแด็ปเตอร์แบบ 8-พอร์ต (EIA-232) ใช้สายเคเบิลต่อไปนี้ :



รูปที่ 42. อินเตอร์เฟส 8-พอร์ตกับการประกอบสายเคเบิลของโมเด็ม

ภาพนี้แสดงอินเตอร์เฟส 8-พอร์ต เชื่อมต่อกับโมเด็มโดยใช้สายเคเบิล 6323741

การเพิ่ม TTY สำหรับโมเด็ม

ใช้ข้อมูลนี้เมื่อเพิ่ม tty สำหรับโมเด็ม

ขั้นแรก ต้องแน่ใจว่าระบบถูกเปิดและโมเด็มถูกปิด ใช้ SMIT fast path `smit mktty`

การตั้งค่าโมเด็ม

ใช้หนึ่งใน 2 วิธีที่แสดงนี้สำหรับการตั้งค่าโมเด็ม

ถ้าคุณมี Basic Networking Utilities (BNU) ติดตั้งอยู่ ดูที่ "การส่งคำสั่ง AT โดยใช้คำสั่ง cu"ถ้าคุณไม่ได้ติดตั้ง BNU ดูที่ "การส่งคำสั่ง AT โดยใช้โปรแกรมภาษา C" ในหน้า 641 สำหรับข้อมูลเกี่ยวกับการติดตั้ง BNU ดูที่ "Basic Networking Utilities" ในหน้า 461

การส่งคำสั่ง AT โดยใช้คำสั่ง cu:

ถ้าคุณมี Basic Networking Utilities (BNU) ติดตั้งอยู่ ใช้คำสั่ง **cu** เพื่อตั้งค่าโมเด็มดังต่อไปนี้

คำสั่งและการตั้งค่าที่อธิบายในส่วนนี้ตั้งค่าโมเด็มที่ใช้ได้กับ Hayes ด้วยพารามิเตอร์พื้นฐานที่ต้องการสำหรับทำงานบนซีเรียลพอร์ตของเซิร์ฟเวอร์

1. เพิ่มบรรทัดต่อไปนี้ในไฟล์ **/usr/lib/uucp/Devices** ของคุณ ไม่ต้องเพิ่มบรรทัดถ้ามันมีอยู่แล้ว (แทนที่ # ด้วยจำนวนพอร์ตของคุณ)

```
Direct tty# - Any direct
```

2. ตรวจสอบว่า tty ถูกปิดการใช้งานโดยพิมพ์ต่อไปนี้ :

   ```
   pdisable tty#
   ```

3. พิมพ์คำสั่งต่อไปนี้:

   ```
   cu -ml tty#
   ```

   คุณควรเป็นข้อความที่บอกว่า Connected

4. ตรวจสอบว่าคุณมีโมเด็มที่ต้องการโดยการพิมพ์ต่อไปนี้:

   ```
   AT
   ```

   โมเด็มควรตอบสนองด้วย OK ถ้าไม่ อ้างถึง “การแก้ไขปัญหาโมเด็ม” ในหน้า 644

   สำหรับ คำสั่ง **AT** เพิ่มเติมและคำอธิบายของมัน ดูที่ “คำสั่ง AT” ในหน้า 646

5. ขึ้นอยู่กับอ็อพชันของ getty ที่คุณเลือก ใส่หนึ่งในคำสั่งต่อไปนี้ แทนค่าอุปกรณ์ tty ด้วย *n*

   - penable tty*n*
   - pshare tty*n*
   - pdelay tty*n*
   - pdisplay tty*n*

ตอนนี้โมเด็มถูกตั้งค่าด้วยคำสั่งพื้นฐานที่ต้องการเพื่อทำงานที่การสื่อสารแบบซีเรียลของระบบปฏิบัติการส่วนใหญ่ต้องการ ถ้าคุณมีปัญหา ใช้คำสั่ง **cu –dl** เพื่อสตาร์ทการติดตามการวินิจฉัยบนการเชื่อมต่อ

การส่งคำสั่ง AT โดยใช้โปรแกรมภาษา C:

ถ้าคุณไม่สามารถตั้งค่าโมเด็มโดยใช้คำสั่ง **cu** หรือถ้าคุณไม่มี BNU ติดตั้งอยู่ ลองรันโปรแกรมภาษา C ต่อไปนี้

สร้างไฟล์ชื่อ motalk.c ที่ประกอบด้วยโค้ดต่อไปนี้ บันทึกไฟล์ คอมไพล์และรันมันตามวิธีในหมายเหตุของคำสั่ง

```
/*************************************************************/
/*  MoTalk - A "C" program for modem setup.                  */
/*           This program is meant as an aid only and is     */
/*           not supported by IBM.                          */
/*                  compile:  cc -o motalk motalk.c          */
/*                  Usage:  motalk /dev/tty? [speed]         */
/*************************************************************/
#include <errno.h>
#include <stdio.h>
#include <signal.h>
#include <fcntl.h>
#include <termio.h>
FILE *fdr, *fdw;
int fd;
struct termio term_save, stdin_save;
void Exit(int sig)
{
   if (fdr) fclose(fdr);
   if (fdw) fclose(fdw);
   ioctl(fd, TCSETA, &term_save);
```

```
   close(fd);
   ioctl(fileno(stdin), TCSETA, &stdin_save);
   exit(sig);
}
main(int argc, char *argv[])
{
   char *b, buffer[80];
   int baud=0, num;
   struct termio term, tstdin;
   if (argc < 2 || !strcmp(argv[1], "-?"))
   {
      fprintf(stderr, "Usage: motalk /dev/tty? [speed]\n");
      exit(1);
   }
   if ((fd = open(argv[1], O_RDWR | O_NDELAY)) < 0)
   {
      perror(argv[1]);
      exit(errno);
   }
   if (argc > 2)
   {
      switch(atoi(argv[2]))
      {
         case   300: baud = B300;
                     break;
         case  1200: baud = B1200;
                     break;
         case  2400: baud = B2400;
                     break;
         case  4800: baud = B4800;
                     break;
         case  9600: baud = B9600;
                     break;
         case 19200: baud = B19200;
                     break;
         case 38400: baud = B38400;
                     break;
         default:    baud = 0;
                     fprintf(stderr, "%s: %s is an unsupported baud\n", argv[0],argv[2]);
                     exit(1);
         }
      }
   /* Save stdin and tty state and trap some signals */
   ioctl(fd, TCGETA, &term_save);
   ioctl(fileno(stdin), TCGETA, &stdin_save);
   signal(SIGHUP, Exit);
   signal(SIGINT, Exit);
   signal(SIGQUIT, Exit);
   signal(SIGTERM, Exit);
   /*  Set stdin to raw mode, no echo */
   ioctl(fileno(stdin), TCGETA, &tstdin);
   tstdin.c_iflag = 0;
   tstdin.c_lflag &= ~(ICANON | ECHO);
   tstdin.c_cc[VMIN] = 0;
   tstdin.c_cc[VTIME] = 0;
```

```
   ioctl(fileno(stdin), TCSETA, &tstdin);
   /*  Set tty state */
   ioctl(fd, TCGETA, &term);
   term.c_cflag |= CLOCAL|HUPCL;
   if (baud > 0)
   {
      term.c_cflag &= ~CBAUD;
      term.c_cflag |= baud;
   }
   term.c_lflag &= ~(ICANON | ECHO); /* to force raw mode */
   term.c_iflag &= ~ICRNL; /* to avoid non-needed blank lines */
   term.c_cc[VMIN] = 0;
   term.c_cc[VTIME] = 10;
   ioctl(fd, TCSETA, &term);
   fcntl(fd, F_SETFL, fcntl(fd, F_GETFL, 0) & ~O_NDELAY);
   /*  Open tty for read and write */
   if ((fdr = fopen(argv[1], "r")) == NULL )
   {
      perror(argv[1]);
      exit(errno);
   }
   if ((fdw = fopen(argv[1], "w")) == NULL )
   {
      perror(argv[1]);
      exit(errno);
   }
   /*  Talk to the modem */
   puts("Ready... ^C to exit");
   while (1)
   {
      if ((num = read(fileno(stdin), buffer, 80)) > 0)
         write(fileno(fdw), buffer, num);
      if ((num = read(fileno(fdr), buffer, 80)) > 0)
         write(fileno(stdout), buffer, num);
      Exit (0);
   }
}
```

การใช้โมเด็ม Hayes และ Hayes-compatible

ใช้โพรซีเดอร์นี้สำหรับโมเด็ม Hayes และ Hayes-compatible

1. เปลี่ยนการตั้งค่า tty ถ้าจำเป็น โดยใช้ SMIT fast path, `smit chtty` ตัวอย่างเช่น คุณอาจต้องการเปลี่ยนฟิลด์ Enable LOGIN เป็น **Share** หรือ **Enable**
2. เพิ่มบรรทัดต่อไปนี้ในไฟล์ `/usr/lib/uucp/Systems`:

   ```
   hayes Nvr HAYESPROG 2400
   ```

3. เพิ่มบรรทัดนี้ในไฟล์ `/usr/lib/uucp/Devices`:

   ```
   # For programming the  hayes modem only:
   HAYESPROG tty0 - 2400 HayesProgrm2400
   #regular ACU entry:
   ACU tty0 - Any hayes
   ```

4. เพิ่มบรรทัดนี้ในไฟล์ `/usr/lib/uucp/Dialers`:

```
# This Entry is used to PROGRAM the modem ONLY:
# the next 3 lines should be made into one:
HayesProgrm2400     =,-,     "" \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OK AT&K3&C1\r\c OK ATL0E0Q2\r\c OK ATS0=1\r\c OK AT&W\r\c
OK
hayes     =,-,     "" \dAT\r\c OK ATDT\T\d\r\c CONNECT
```

5. เพื่อโปรแกรมโมเด็ม ใช้คำสั่ง `cu -d hayes` คำสั่งนี้ใช้คำสั่ง **cu** เพื่อโปรแกรมโมเด็ม เนื่องจากไม่มีการเชื่อมต่อไปยังระบบอื่น คำสั่งจะล้มเหลว โมเด็มจะถูกโปรแกรมถ้า `sendthem AT&W` และจากนั้น `OK got it` ถูกแสดงในเอาต์พุต

   ถ้าคุณไม่ได้ทำการถ่ายโอนไฟล์ไบนารี หรือใช้ BNU ไม่ต้องใช้คำสั่ง **&K3** และตั้ง XON เป็นโฟลว์คอนโทรลที่จะถูกใช้ อย่างไรก็ตาม มันจะมีประสิทธิภาพมากกว่าถ้าใช้ฮาร์ดแวร์โฟลว์คอนโทรล (ตรงกันข้ามกับ XON-XOFF แฮนด์เชคกิ้ง) ในการทำดังกล่าว ใช้การตั้งค่าและ entry `Dialers` จากขั้นตอนต่อไป
6. หลังจากโมเด็มถูกโปรแกรม คุณสามารถตั้งค่าไดรเวอร์อุปกรณ์ของระบบเพื่อใช้ฮาร์ดแวร์โฟลว์คอนโทรล การใช้ SMIT (`smit chtty` fast path) เปลี่ยนตัวควบคุมโฟลว์ไปยัง RTS ตรวจสอบคู่มือของโมเด็มเพื่อหาว่าโมเด็มของคุณสนับสนุนฮาร์ดแวร์โฟลว์คอนโทรลหรือไม่

การแก้ไขปัญหาโมเด็ม

เมื่อพบปัญหาเมื่อคุณใช้โมเด็มกับคอมพิวเตอร์ ของคุณ โปรดจำจุดต่อไปนี้

- บางโมเด็มสนใจขนาดตัวพิมพ์ใช้ตัวพิมพ์ใหญ่สำหรับคำสั่ง **AT**
- ในการทำงานปกติ ต้องการให้โมเด็มถูกรีเซ็ตเมื่อ DTR ถูกดร็อบ (&การตั้งค่า D3) อย่างไรก็ตามเมื่อโมเด็มถูกตั้งค่าเป็นครั้งแรก แนะนำว่าไม่ให้รีเซ็ตโมเด็ม ถ้า DTR ถูกดร็อบ (&การตั้งค่า D2) ถ้าโมเด็มรีเซ็ตตัวเอง การตั้งค่าที่ถูกโปรแกรมไว้ทั้งหมดที่ยังไม่ถูกบันทึกในหน่วยความจำของโมเด็มจะหายไป

  การที่ไม่ให้โมเด็มรีเซ็ตยังเป็นการป้องกันการเปลี่ยนแปลงเมื่อ &C1 ถูกตั้ง การเปลี่ยนสถานะ Carrier Detect อาจทำให้สาย carrier detect ถูกเปิดปิดบนบางโมเด็ม ซึ่งส่งผลให้คำสั่ง **cu** ดร็อปสาย คุณอาจต้องการตั้งค่าโมเด็มเป็น &D3 หลังจากทำการตั้งค่าครั้งสุดท้าย
- แม้ว่าคำสั่งที่ให้ในคอลเล็กชันหัวข้อนี้เป็นคำสั่งมาตรฐาน สำหรับโมเด็มที่เข้ากันได้กับ Hayes ส่วนใหญ่ แต่ไม่ได้รับประกันว่าเป็นคำสั่งมาตรฐาน สำหรับโมเด็มของคุณ เปรียบเทียบคำสั่งกับเอกสารคู่มือ โมเด็มของคุณก่อนทำต่อไป

วิธีการที่สะดวกในการดีบักปัญหาของโมเด็มคือการถอดโมเด็มออกและต่อ ASCII เทอร์มินัล (พร้อมกับ interposer หรือ null modem) กับพอร์ตและสายเคเบิลเดียวกับโมเด็ม ตั้งค่าเทอร์มินัลด้วยความเร็วของสาย บิตต่อตัวอักขระ และพาริตีเดียวกับโมเด็ม ถ้าพอร์ต มีการเปิดใช้งานสำหรับล็อกอิน หน้าล็อกอินต้องแสดงขึ้นบน หน้าจอ ถ้าหน้าล็อกอินถูกแสดงบนหน้าจอของเทอร์มินัล ดังนั้นปัญหาเกี่ยวกับคอนฟิกูเรชันของโมเด็มจะถูกแยกออกมาอย่างรวดเร็ว

เคล็ดลับต่อไปนี้ช่วยคุณในการแยกแยะปัญหาที่เชื่อมโยง กับการเชื่อมต่อโมเด็ม:

ตารางที่ 103. ปัญหาของโมเด็มและการแก้ไข

ปัญหา	การแก้ไข
Respawning เร็วเกินไป	โปรแกรม **getty** ถูก respawn โดย **init**

ตารางที่ 103. ปัญหาของโมเด็มและการแก้ไข (ต่อ)

ปัญหา	การแก้ไข
ข้อความบนคอนโซล หรือ errpt	ถ้า **init** เห็นว่ามันจะต้อง respawn โปรแกรมใดๆมากกว่า 5 ครั้งใน 225 วินาที มันจะแสดงข้อความบนคอนโซลและไม่ respawn มันสำหรับช่วงเวลาหนึ่ง วิธีการแก้ไขคือการหาว่าทำไม **getty** ถึงตาย ซึ่งเป็นได้หลายกรณี: • ตั้งค่าโมเด็มไม่ถูกต้อง โดยทั่วไปเป็นผลมาจากมี CD เป็น high บนโมเด็มหรือสายเคเบิลและยังมี "echo" หรือ "command response" ที่ถูกเปิด (CD ยังสามารถเป็น high โดยการเพิ่ม clocal ใน runmodes และ/หรือ logmodes ในคอนฟิกูเรชันของพอร์ตหรือยังถูกบังคับบน 128 พอร์ต) • การปิดเปิดของสัญญาณ CD กระบวนการ **getty**จะตายทุกครั้งที่ CD ถูกปิดเปิดจากสถานะ on เป็น off (แอ็คชันนี้อาจเกิดได้จากหลายสาเหตุ ต้องแน่ใจว่าตรวจสอบว่าสายเคเบิลถูกชิลด์อย่างเหมาะสม การล็อกอิน และล็อกเอาต์หลายๆครั้งในการเข้าถึงอย่างรวดเร็วสามารถทำให้เกิดปัญหานี้ได้)
ไม่มีพร้อมต์สำหรับล็อกอินถูกแสดงหลังจากการเชื่อมต่อกับโมเด็ม	ต้องแน่ใจว่า **getty** กำลังรันบนพอร์ต ถ้ามันรัน ตรวจสอบว่าการเชื่อมต่อ carrier detect กับสัญญาณของโมเด็มสูงขึ้นหลังจากที่ด้านรีโมตเชื่อมเข้ากับโมเด็ม ถ้า CD ถูกยืนยันว่าถูกต้องแล้ว ดังนั้นตรวจสอบว่าโมเด็มเชื่อมกับพอร์ตที่ถูกต้อง ถ้าคุณยังไม่เป็นล็อกอิน ดังนั้นเชื่อมเทอร์มีนัลพร้อมกับ interposer เข้ากับสายเคเบิลแทนที่โมเด็มและตรวจสอบว่าพร้อมต์สำหรับล็อกอินปรากฏหรือไม่ ถ้าคุณยังไม่เห็นพร้อมต์ พยายามให้แสดงตัวอักขระกับหน้าจอของเทอร์มินัลเพื่อตรวจสอบสายเคเบิลและฮาร์ดแวร์ทำงานอย่างถูกต้อง
เมื่อเชื่อมต่อรีโมตโมเด็ม มันจะตัดการเชื่อมต่อทันที	ตรวจสอบว่าโมเด็มคุยกับเซิร์ฟเวอร์ด้วยความเร็วเดียวกับที่เซิร์ฟเวอร์ฟังโมเด็ม ลองอัตรา baud อื่นสำหรับ tty หรือโปรแกรมโมเด็มเพื่อล็อกความเร็ว DTE เพื่อให้ตรงกับความเร็วของพอร์ต tty ตรวจสอบว่าโมเด็มและพอร์ตไม่มีสัญญาณ carrier detect ที่สูง หรือพอร์ตนั้นถูกใช้โดยกระบวนการอื่น
ได้รับอักขระขยะแทนที่จะได้รับพร้อมต์สำหรับล็อกอิน	เกิดเนื่องจากใช้โปรโตคอลต่างกัน ต้องแน่ใจว่าโมเด็มและพอร์ต tty ใช้พาริตี อัตรา baud โฟลว์คอนโทรล และขนาดของตัวตักขระเดียวกัน
บางครั้ง หลังจากเซสชันทำสำเร็จ ไม่มีใครสามารถล็อกอิน	อาจเป็นไปได้ว่าโมเด็มไม่ถูกรีเซ็ตหลังจากตัดการเชื่อมต่อ ดูที่คู่มือของโมเด็มเพื่อดูวิธีการตั้งโมเด็มเพื่อให้รีเซ็ตหลังจากการเปลี่ยนแปลงของ DTR จาก on เป็น off
บัฟเฟอร์ของตัวรับ overruns ใน errpt	บัฟเฟอร์ของชิป UART overrun ลดค่าของทริกเกอร์ของตัวรับใน SMIT สำหรับ tty วิธีการแก้ปัญหานี้ใช้ได้กับอะแด็ปเตอร์อะซิงโครนัส 8-พอร์ต หรือ 16-พอร์ตแบบดั้งเดิมเท่านั้น ตรวจสอบว่าโมเด็มและพอร์ต tty ใช้โฟลว์คอนโทรลเหมือนกัน
ข้อผิดพลาด ttyhog ใน errpt	โมเด็มและ tty ไม่ได้ใช้โฟลว์คอนโทรลเหมือนกัน หรือไม่ได้ใช้โฟลว์คอนโทรล

คำถามสำหรับการบริการซอร์ฟแวร์ของโมเด็ม:

ก่อนที่คุณจะโทรเพื่อขอความช่วยเหลือเกี่ยวกับปัญหาของโมเด็ม ให้รวบรวมข้อมูลพื้นฐานบางอย่างเพื่อทำให้คุณได้รับบริการที่เร็วขึ้น

ข้อมูลที่คุณควรมีรวมถึงต่อไปนี้ :

- ระดับของระบบปฏิบัติการ คุณใช้ระบบปฏิบัติการนี้มานานแค่ไหนแล้ว ?
- โมเด็มเคยใช้งานได้มาก่อนหรือไม่ ?
- โมเด็มที่คุณใช้เป็นชนิดไหน ? โมเด็มที่อยู่อีกด้านของการเชื่อมต่อเป็นชนิดอะไร ?
- ชนิดของอะแด็ปเตอร์ที่โมเด็มต่ออยู่คืออะไร ?
- หมายเลขพอร์ตที่โมเด็มเชื่อมต่อ ?
- หมายเลข tty ที่โมเด็มเชื่อมต่อ ?
- ชนิดของสายเคเบิลที่คุณใช้ ?
- การตั้งค่าการล็อกอินคืออะไร (share, delay, enable)?
- โมเด็มสามารถเชื่อมต่อกับโมเด็มอื่นหรือไม่ ?

- โมเด็มอื่นสามารถเชื่อมต่อกับโมเด็มของคุณหรือไม่ ?
- ค่าต่อไปนี้คืออะไรใน SMIT, โมเด็ม หรือพอร์ต?
  - XON/XOFF?
  - RTS/CTS?
  - อัตรา BPS ?
- รวมสิ่งต่อไปนี้ในคำอธิบายปัญหาของคุณ :
  - พอร์ตถูกล็อกเป็นบางครั้งหรือไม่ ?
  - คุณสามารถโทรออกหรือไม่ ? ผู้อื่นสามารถโทรเข้ามาได้หรือไม่ ?
  - เงื่อนไขอื่นหรือเงื่อนไขข้อผิดพลาดที่สามารถอธิบายได้
- มีข้อผิดพลาดบนคอนโซลหรือไม่ ? มันคืออะไร ?
- มีข้อผิดพลาดในรายงานข้อผิดพลาดหรือไม่ ? (**errpt** หรือ **errpt -a**)
- คุณใช้คำสั่งอะไรในการโทรออก ?
- ซอฟต์แวร์อะไรที่ถูกใช้บนระบบ ?

คำสั่ง AT:

ชุดของคำสั่ง Hayes Smartmodem จะรวมชุดคำสั่ง AT ที่ถูกใช้โดยโมเด็มที่เป็นที่นิยมจำนวนมาก

ข้อมูลนี้มาจาก Hayes Smartmodem 2400 *Quick Reference Card* ที่ถูกเผยแพร่โดย Hayes Microcomputer Products, Inc. ศึกษาเอกสารของโมเด็มสำหรับลิสต์ของคำสั่ง AT ที่เกี่ยวข้อง

ไอเท็ม	คำอธิบาย
AT	ส่วนนำหน้าของคำสั่ง - นำหน้าบรรทัดรับคำสั่ง
<CR>	ตัวอักขระขึ้นบรรทัดใหม่ (บรรทัดใหม่) - ยกเลิกบรรทัดรับคำสั่ง
A	Go off-hook ยังคงอยู่ในโหมดคำสั่ง
A/	บรรทัดรับคำสั่งก่อนหน้านั้นซ้ำอีกครั้ง คำสั่งนี้ไม่นำหน้าด้วย **AT** หรือตามด้วย **<CR>** /
B0	เลือกมาตรฐาน CCITT V.22 สำหรับการสื่อสาร 1200 bps
B1	เลือกมาตรฐาน Bell 212A สำหรับการสื่อสาร 1200 bps
D	เข้าสู่โหมด originate หมุนหมายเลขที่ตามมา และพยายามออนไลน์ D โดยมากจะตามด้วย T สำหรับ tone P สำหรับ pulse อาจถูกใช้
DS=*n*	หมุนหมายเลขที่ถูกเก็บในตำแหน่ง *n*
E0	ปิดการใช้งานการ echo อักขระในสถานะคำสั่ง
E1	เปิดการใช้งานการ echo อักขระในสถานะคำสั่ง
H0	Go on-hook (วางหูโทรศัพท์)
H1	ดำเนินการ switch-hook และ auxiliary relay.
I0	คืนโค้ดการระบุผลิตภัณฑ์
I1	ดำเนินการ checksum บน firmware ROM คืนค่า checksum
I2	ดำเนินการ checksum บน firmware ROM คืนค่า OK หรือ ERROR เป็นผลลัพธ์
L0	ปิดลำโพง
L1	วอลุ่มของลำโพงระดับต่ำ
L2	วอลุ่มของลำโพงระดับกลาง
L3	วอลุ่มของลำโพงระดับสูง
M0	ปิดลำโพง
M1	เปิดลำโพงจนกว่าจะตรวจพบ carrier
M2	ลำโพงเปิดตลอด
M3	เปิดลำโพงจนกว่าจะตรวจพบ carrier ยกเว้นระหว่างหมุนหมายเลข
O0	เข้าสู่สถานะออนไลน์
O1	เข้าสู่สถานะออนไลน์และเริ่มต้น equalizer retrain
Q0	โมเด็มคืนค่าโค้ดผลลัพธ์

ไอเท็ม	คำอธิบาย
Q1	โมเด็มไม่คืนค่าโค้ดผลลัพธ์
Sr	ตั้งตัวชี้ไปยัง register r
Sr=*n*	ตั้ง register r เป็นค่า *n*
V0	แสดงโค้ดผลลัพธ์ในรูปแบบตัวเลข
V1	แสดงโค้ดผลลัพธ์ในรูปแบบ verbose (เป็นคำ)
X0	เปิดใช้งานคุณลักษณะที่ถูกแทนโดยโค้ดผลลัพธ์ 0–4
X1	เปิดใช้งานคุณลักษณะที่ถูกแทนโดยโค้ดผลลัพธ์ 0–5, 10
X2	เปิดใช้งานคุณลักษณะที่ถูกแทนโดยโค้ดผลลัพธ์ 0–6, 10
X3	เปิดใช้งานคุณลักษณะที่ถูกแทนโดยโค้ดผลลัพธ์ 0–5, 7, 10
X4	เปิดใช้งานคุณลักษณะที่ถูกแทนโดยโค้ดผลลัพธ์ 0–7, 10
Y0	ปิดใช้งาน long space disconnect.
Y1	เปิดใช้งาน long space disconnect.
Z	รีเซ็ตโมเด็ม
&C0	สันนิษฐานว่า carrier มีอยู่เสมอ
&C1	ติดตามการมีอยู่ของ data carrier
&D0	ไม่สนใจสัญญาณ DTR
&D1	สันนิษฐานว่าเป็นสถานะคำสั่งเมื่อมีการเปลี่ยนแปลงจาก on เป็น off ของ DTR เกิดขึ้น
&D2	วางหูและสันนิษฐานว่าเป็นสถานะคำสั่งเมื่อมีการเปลี่ยนแปลงจาก on เป็น off ของ DTR เกิดขึ้น
&D3	รีเซ็ตเมื่อมีการเปลี่ยนแปลงจาก on เป็น off ของ DTR เกิดขึ้น
&F	เรียกใช้ค่าที่ตั้งจากโรงงานเป็นการตั้งค่าที่แอ็คทีฟ
&G0	ไม่มี guard tone
&G1	500 Hz guard tone
&G2	1800 Hz guard tone
&J0	RJ-11/RJ41/RJ45S telco jack.
&J1	RJ-11/RJ-13 telco jack
&P0	การหมุนโทรศัพท์แบบ Pulse พร้อมกับอัตรา make/break เป็น 39/61
&P1	การหมุนโทรศัพท์แบบ Pulse พร้อมกับอัตรา make/break เป็น 33/67
&Q0	ทำงานในโหมดอะซิงโครนัส
&Qn	ทำงานในโหมดซิงโครนัส *n*
&R0	ติดตาม CTS โดยขึ้นอยู่กับ RTS
&R1	ไม่สนใจ RTS สันนิษฐานว่ามี CTS อยู่ตลอด
&S0	สันนิษฐานว่ามีสัญญาณ DSR
&S1	ติดตามการมีอยู่ของสัญญาณ DSR
&T0	กำลังยกเลิกการทดสอบ
&T1	เริ่มต้น analog loopback แบบโลคัล
&T3	เริ่มต้น ดิจิทัล loopback
&T4	สร้างคำร้องขอจากโมเด็มแบบรีโมตจาก remote data link (RDL)
&T5	ปฏิเสธคำร้องขอ RDL จากโมเด็มแบบรีโมตสำหรับ
&T6	เริ่มต้น ดิจิทัล loopback แบบรีโมต
&T7	เริ่มต้น ดิจิทัล loopback แบบรีโมตพร้อมกับ self-test
&T8	เริ่มต้น analog loopback แบบโลคัลพร้อมกับ self-test
&V	ดูการตั้งค่าที่แอ็คทีฟโปรไฟล์ผู้ใช้ และจำนวนที่เก็บ
&Wn	บันทึกพารามิเตอร์ที่สามารถเก็บได้ของการตั้งค่าที่แอ็คทีฟเป็นโปรไฟล์ผู้ใช้ *n*
&X0	โมเด็มให้สัญญาณนาฬิกาการส่งข้อมูล
&X1	เทอร์มินัลข้อมูลให้สัญญาณนาฬิกาการส่งข้อมูล
&X2	carrier ที่รับให้สัญญาณนาฬิกาการส่งข้อมูล
&Yn	เรียกคืนโปรไฟล์ผู้ใช้ *n*
&Zn=*x*	เก็บหมายเลขโทรศัพท์ *x* ในตำแหน่ง *n*

การสรุป S-register:

S-registers ช่วงของมัน และคำอธิบายของมันถูกแสดงในตารางต่อไปนี้

ตารางที่ 104. คำอธิบายของ S-register

ลงทะเบียน	Range	รายละเอียด
S0	0-255	เลือกจำนวน ring ก่อนที่จะตอบ
S1	0-255	จำนวนของ Ring (ถูกเพิ่มโดยแต่ละ ring)
S2	0-127	กำหนดลำดับของอักขระ escape (ASCII)
S3	0-127	กำหนดอักขระ carriage return (ASCII)
S4	0-127	กำหนดอักขระขึ้นบรรทัดใหม่ (ASCII)
S5	0-32, 127	กำหนดอักขระแบ็กสเปช (ASCII)
S6	2-255	เลือก wait-time เป็นวินาทีก่อน blind dialing
S7	1-55	เลือก wait-time เป็นวินาทีสำหรับ carrier/dial tone.
S8	0-255	เลือกช่วงเวลาเป็นวินาทีของคอมม่า
S9	1-255	เวลาการตอบสนอง Carrier detect โดยเพิ่มขึ้นทีละ .1 วินาที (10 = 1 วินาที)
S10	1-255	การหน่วงเวลาระหว่าง carrier loss และการวางหู โดยเพิ่มขึ้นทีละ .1 วินาที
S11	50-255	ช่วงเวลา/ช่องว่างของ tone หน่วยเป็นมิลลิวินาที
S12	50-255	ช่วงเวลา Escape sequence guard time ใน .02 วินาที
S13	—	สงวนไว้
S14	—	สงวนไว้
S15	—	สงวนไว้
S16	—	สงวนไว้ - ฟังก์ชันสำหรับ register นี้ถูกควบคุมโดยคำสั่ง **&T**)
S17	—	สงวนไว้
S18	0-255	ช่วงเวลาการจับเวลาการทดสอบหน่วยเป็นวินาที
S19	—	สงวนไว้
S20	—	สงวนไว้
S21	—	สงวนไว้
S22	—	สงวนไว้
S23	—	สงวนไว้
S24	—	สงวนไว้
S25	0-255	เลือกช่วงเวลา DTR change detect time ใน .01 วินาที
S26	0-255	ช่วงเวลา RTS to CTS delay ใน .01 วินาที

ตารางที่ 104. คำอธิบายของ S-register (ต่อ)

ลงทะเบียน	Range	รายละเอียด
S27	—	สงวนไว้

โค้ดผลลัพธ์สำหรับอะแด็ปเตอร์อะซิงโครนัส:

โค้ดผลลัพธ์ที่ส่งคืนมาจากอะแด็ปเตอร์อะซิงโครนัส รวมถึงตัวเลขของมัน คำ และคำอธิบาย ถูกระบุในตารางต่อไปนี้

ตารางที่ 105. โค้ดผลลัพธ์ของอะแด็ปเตอร์อะซิงโครนัส

ตัวเลข	คำ	รายละเอียด
0	OK	คำสั่งที่ถูกประมวลผล
1	CONNECT	การเชื่อมต่อถูกทำที่ 0-300 bps
2	RING	ตรวจพบสัญญาณ Ring
3	NO CARRIER	สัญญาณ Carrier หายไปหรือตรวจไม่พบ
4	ERROR	คำสั่งไม่ถูกต้อง checksum ข้อผิดพลาดในบรรทัดรับคำสั่ง หรือบรรทัดรับคำสั่งยาวเกินไป
5	CONNECT 1200	การเชื่อมต่อถูกทำที่ 1200 bps
6	NO DIALTONE	ตรวจไม่พบ dial tone
7	BUSY	ตรวจพบสัญญาณ Busy
8	NO ANSWER	ไม่มีการตอบสนองเมื่อโทรไปยังระบบ
9	CONNECT 2400	การเชื่อมต่อถูกทำที่ 2400 bps

ตัวแก้ไขการหมุนโทรศัพท์:

ตัวแก้ไขการหมุนโทรศัพท์และคำอธิบายของมันสามารถอ้างถึงในตารางต่อไปนี้

ไอเท็ม	คำอธิบาย
0-9 # * A-D	ตัวเลขและตัวอักขระสำหรับการหมุนโทรศัพท์
P	การหมุนโทรศัพท์แบบ Pulse
T	การหมุนโทรศัพท์แบบ Tone
,	การหน่วงเวลาสำหรับตัวอักขระถัดไป
!	Hookflash
@	รอเพื่อให้เงียบ
W	รอโทนการหมุนโทรศัพท์
;	กลับไปยังสถานะคำสั่งหลังจากที่หมุนโทรศัพท์
R	โหมดตรงข่าม
S=*n*	หมุนหมายเลขที่ถูกเก็บในตำแหน่ง*n*

ความช่วยเหลือเกี่ยวกับโมเด็ม:

เมื่อพบปัญหากับโมเด็มของคุณ ความช่วยเหลือจะมีให้ในที่ต่อไปนี้

- ผู้แทนในพื้นที่ของคุณสามารถช่วยเหลือในการตั้งค่าโมเด็ม

- มีอ็อพชันการสนับสนุนที่แตกต่างกันหลายอ็อพชันพร้อมสำหรับลูกค้าในการเสนอ Support Services รวมถึงการช่วยเหลือแบบ on-site หรือการสนับสนุนโดยโทรศัพท์ ติดต่อตัวแทนบริการของคุณสำหรับความช่วยเหลือ
- บางทีรีซอร์สของความช่วยเหลือที่มักจะมองข้ามคือผู้ผลิตโมเด็มเอง ผู้ผลิตส่วนใหญ่จะมีการช่วยเหลือแบบออนไลน์บางอย่างสำหรับผลิตภัณฑ์ของเขา

entry ของไฟล์ /usr/lib/uucp/Dialers.samples:

ตัวอย่างของ entry ของไฟล์เหล่านี้ถูกจัดเตรียมโดยไม่มีการรับประกันใดๆและจะทำงานตามที่เป็นอยู่สำหรับโมเดลที่กล่าวถึง แต่อาจไม่ตรงกับความต้องการของคุณ

การแก้ไขบางอย่างอาจต้องการเพื่อให้ตรงกับความต้องการของคุณ ศึกษาคู่มือของโมเด็มของคุณสำหรับคำอธิบายการตั้งค่าในรายละเอียดเพิ่มเติม

เพื่อใช้การตั้งค่าเพื่อโปรแกรมโมเด็ม คุณต้องใช้ entry ในไฟล์ /usr/lib/uucp/Systems เช่น :

```
hayes Nvr HayesPRGM Any
```

ไฟล์ /usr/lib/uucp/Devices ควรมี entry เช่น :

```
HayesPRGM tty0 - 2400 HayesProgrm2400
```

โดยการใช้ 2 entry ด้านบน ใช้คำสั่ง **cu** เพื่อโปรแกรมโมเด็ม :

```
cu -d hayes
```

```
# COMPONENT_NAME: cmduucp
#
#
# (C) COPYRIGHT International Business Machines Corp. 1994
# Licensed Materials - Property of IBM
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM
# Corp.
#########################################################################
# Motorola UDS Modem
#
# Use udsmodemPROGRAM to program the modem.
# Port needs to have rts/cts set.
# Use uds or hayes dialer.
#
# The "udsmodemPROGRAM" line should be a single, continuous line
#
#########################################################################
udsmodemPROGRAM =,-, "" \dAT&FQ2\r\c OK
ATE0Y0&C1&D2&S1%B5%E0*LC\r\c OKAT&K3&W\r\c OK

uds =,-, "" \dAT\r\c OK\r ATDT\T\d\r\c CONNECT

#########################################################################
#
# IBM 7855 Model 10
# Use IBMProgrm to program the modem.
# This sets rts/cts flow control, turns
```

```
# off xon/xoff, and sets the DTE speed at 19,200 bps.
# The modem will connect at the appropriate speed and
# flow control with the server.
# Port needs to have rts/cts set.
#
# The "IBMProgrm" line should be a single, continuous line
#
#########################################################################
IBMProgrm =,-, "" \dATQ0\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OK AT&C1\R2\Q2\M14\r\c OK AT&B8N1L0E0\A0\r\c OK
ATS0=1\r\c OK ATQ1&W0&Y0\r\c ""

#########################################################
# The following are used for Dialing out on a 7855
# regular ACU device. We have to turn on result
# codes (Q0) because they are turned off when we
# programmed it. (Keeps all upper case login from
# happening on dial in attempts.)
# We have to have an extra "\" before "\N" because
# the BNU programs strips it if it's before an "N".
#########################################################
ibm =,-, "" \dATQ0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 ECL (No Compression)
ibmecl =,-, "" \dAT\\N3%C0Q0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 ECLC (Compression)
ibmeclc =,-, "" \dAT\\N3%C1Q0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 ECLC Compression with 256 byte block size
ibmeclc256 =,-, "" \dAT\\N3%C1Q0\A3\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 1200bps
ibm_ne12 =,-, "" \dATQ0\\N0&A2%C0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 2400bps
ibm_ne24 =,-, "" \dATQ0\\N0&A3%C0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 9600bps
ibm_ne96 =,-, "" \dATQ0\\N0&A6%C0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 19200bps
ibm_ne192 =,-, "" \dATQ0\\N0%C0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 12000bps
ibm_ne120 =,-, "" \dATQ0\\N3%C0&AL8\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 1200bps (Dial Quietly)
ibmq12 =,-, "" \dATQ0\r\c OK AT&A2M0DT\T\d\r\c CONNECT

# IBM 7855 No Compression 2400bps (Dial Quietly)
ibmq24 =,-, "" \dATQ0\r\c OK AT&A3M0DT\T\d\r\c CONNECT

# IBM 7855 No Compression 9600bps (Dial Quietly)
ibmq96 =,-, "" \dATQ0\r\c OK AT&A6M0DT\T\d\r\c CONNECT
```

```
# IBM 7855 No Compression 19200bps (Dial Quietly)
ibmq192 =,-, "" \dATQ0\r\c OK ATM0DT\T\d\r\c CONNECT

##########################################################################
#
# Intel 9600EX Modem
# Use IntelProgram to program the modem.
# This sets rts/cts flow control, and turns
# off xon/xoff.
# Port needs to have rts/cts set. (Use hayes dialer)
#
# The "IntelProgram" line should be a single, continuous line
#
##########################################################################
#IntelProgram =,-, "" \d\dAT\r\c OK AT&F\r\c OK AT&S1M1\r\c OK
AT&D3\r\c OKAT&C1\r\c OK ATL0E0Y0&Y0\X1\r\c OK ATS0=1\r\c OK
AT&W\r\c OK

##########################################################################
# Practical Peripherals 14400FXMT Modem
# Use PracPerProgram144 to program the modem.
# This sets rts/cts flow control, and turns
# off xon/xoff. (Use hayes dialer)
# DTE speed will be locked at connect speed when
# the modem is programmed. (Suggestion: 38400 baud)
#
# The "PracPerProgram144" line should be a single, continuous
# line
##########################################################################
PracPerProgram144 =,-, "" \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OKAT&C1&K3\r\c OK ATQ2E1&Q9\r\c OK ATS0=1S9=20\r\c OK
AT&W\r\c OK

##########################################################################
# Practical Peripherals 9600 bps Modem
# Use PracPerProgram9600 to program the modem.
# This sets rts/cts flow control, and turns
# off xon/xoff. (Use hayes dialer)
#
# The "PracPerProgram144" line should be a single, continuous
# line
##########################################################################
PracPerProgram9600 =,-, "" \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OKAT&C1&K3\r\c OK ATL0E0\r\c OK ATS0=1S9=20\r\c OK
AT&W\r\c OK

##########################################################################
# Practical Peripherals 2400 bps Modem
# Use PracPerProgram to program the modem
#
# The "PracPerProgram2400" line should be a single, continuous
# line
##########################################################################
PracPerProgram2400 =,-, "" \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
```

```
AT&D3\r\c OKAT&C1\r\c OK ATL0E0\r\c OK ATS0=1S9=20\r\c OK AT&W\r\c OK

#########################################################################
# Hayes 2400 bps Modem
# Use HayesProgrm2400 to program the modem.
# (Use hayes dialer to dial)
#
# The "HayesProgrm2400" line should be a single, continuous line
#
#########################################################################
HayesProgrm2400 =,-, "" \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OKAT&C1\r\c OK ATL0E0\r\c OK AT S0=1\r\c OK AT&W\r\c OK

#########################################################################
# Telebit t2000 Trailblazer Plus
# Use TelebitProgrm to program the modem
# This sets rts/cts flow control, and turns
# off xon/xoff and sets the Default DTE speed at
# 19,200 bps.
# Port needs to have rts/cts set.
# This sets modem to send PEP tones last as they can
# can confuse some other modems.
#
# The "TelebitProgram" line should be a single, continuous line
#
#########################################################################
TelebitProgram =,-, "" \dAT&F\r\c OK
ats2=255s7=60s11=50s41=2s45=255s51=254s52=2s54=3s58=2s64=1s66=1\r\c OK
ATs69=1s92=1s96=0s105=0s110=1s111=30s130=3s131=1F1M0Q6TV1W0X3Y0\r\c OK
ATE0&W\r\c OK
# Telebit T2000 dialers Entries:
# Forces a PEP connection:
tbfast =,-, "" \dATs50=255s7=60\r\c OK\r ATDT\T\r\c
CONNECT-\d\c-CONNECT

# 2400bps connection:

#tb2400 =,-, "" \dATs50=3\r\c OK\r ATDT\T\r\c CONNECT

# 2400 MNP:
tb24mnp =,-, "" \dAT\r\c OK ATS0=0S95=2S50=3S41=0\r\c OK
ATDT\T\r\c CONNECT

# 1200bps connection:#tb1200 =,-, "" \dATs50=2\r\c OK\r
ATDT\T\r\c CONNECT

# 1200 MNP:
tb12mnp =,-, "" \dAT\r\c OK ATS0=0S95=2S50=2S41=0\r\c OK
ATDT\T\r\c CONNECT

#########################################################################
# Telebit WorldBlazer
# WORLDBLAZERProgram sets the DTE speed at 38400, but
# you could set it higher if the DTE connection can
# handle it. We answer with PEP tones last so as not
```

```
# to confuse other modems. This turns off xon/xoff
# and turns on RTS/CTS flow control. The port should
# be locked to 38400 with these settings, and needs
# to have RTS/CTS turned on.
#
# The "WORLDBLAZERProgram" line should be a single, continuous
# line
##########################################################################
WORLDBLAZERProgram =,-, "" \dAT\r\c AT AT&F3M0\r\c AT
ATs51=253s92=1\r\c ATAT&W\r\c AT

##########################################################
# ACU Dialers for various BAUD rates for the
# WorldBlazer - each sets the modem to attempt to
# connect at a specific speed and lower. # WBlazer will accept whatever the remote modem can
# do. You will want to use PEP for other Telebits,
# so use WBlazer38400 or WBlazer19200 for those
##########################################################
# WBlazer =,-, "" \dAT\r\c OK ATDT\T\d\r\c CONNECT
WBlazer38400 =,-, "" \dATs50=255\r\c OK ATDT\T\d\r\c CONNECT
WBlazer19200 =,-, "" \dATs50=255\r\c OK ATDT\T\d\r\c CONNECT
# WBlazer14400 attempts to negotiate a V.42bis connection.
WBlazer14400 =,-, "" \dATs50=7\r\c OK ATDT\T\d\r\c CONNECT

# For a V.32 connection:
WBlazer9600 =,-, "" \dATs50=6\r\c OK ATDT\T\d\r\c CONNECT

# For a V.22 connection:
WBlazer2400 =,-, "" \dATs50=3\r\c OK ATDT\T\d\r\c CONNECT

# For a 1200 bps connection:
WBlazer1200 =,-, "" \dATs50=2\r\c OK ATDT\T\d\r\c CONNECT
```

ข้อพิจารณาการเดินสายเคเบิลสำหรับ 128-พอร์ต โมเด็ม:

ระบบปฏิบัติการนี้ไม่ต้องการ DSR ในแอ็พพลิเคชันสำหรับควบคุมโมเด็ม และเนื่องจากโมเด็มเกือบทั้งหมดทุกวันนี้มีความสามารถในการตอบรับแบบอัตโนมัติ สัญญาณ Ring Indicator โดยทั่วไปจะไม่จำเป็นต้องใช้

ปลั๊ก RJ-45 แบบ 10-ขา ไม่ได้เป็นระบบย่อยของสายเคเบิลที่มีความโดดเด่น และอาจจะไม่ได้รับความนิยมในตลาดค้าปลีก ระบบย่อย TTY ของระบบปฏิบัติการนี้จัดเตรียมคุณลักษณะที่เป็นอ็อพชันที่เรียกว่า ALTPIN ซึ่งสลับฟังก์ชันแบบโลจิคัลของ DSR (Data Set Ready) กับ DCD (Data Carrier Detect) สำหรับพอร์ต เมื่อ ALTPIN ถูกเปิดใช้งาน DCD จะมีให้ใช้งานบนขาที่ 1 ของตัวเชื่อมต่อ RJ-45 แบบ 8-ขา (เทียบเท่ากับ ขาที่ 2 ของตัวเชื่อมต่อแบบ 10-ขา)

ถ้าคุณต้องการทำ สายเคเบิลของโมเด็มแบบ 8 สายสำหรับ 128-พอร์ต RAN ใช้การเดินสายปลั๊ก RJ-45 แบบ 8-ขา :

ตารางที่ 106. การเดินสายเคเบิลสำหรับ 128-พอร์ต โมเด็ม

ไอเท็ม	คำอธิบาย	โมเด็ม
SYSTEM END CONNECTOR RJ-45c แบบ 8 ขา	**DEVICE ENDRI**	22
1	DSR	6
2	RTS	4
3 (Chassis)	GND	SHELL
4	TxD	2
5	TxD	3
6 (สัญญาณ)	GND	7
7	CTS	5
8	DTR	20
	CD	8

หมายเหตุ: ตำแหน่งฟิสิคัลของ DSR และ CD อาจถูกสลับกันโดยพารามิเตอร์ ALTPIN เมื่อเปิดใช้งานโดยใช้คำสั่ง stty-cmxa

ตารางต่อไปนี้แสดงสัญญาณการสื่อสารแบบอะซิงโครนัสระหว่างหน่วยของระบบและโมเด็มที่เชื่อมอยู่ ในที่นี้ ข้อมูลถูกส่งจากหน่วยของระบบไปยังระบบแบบรีโมต

ตารางที่ 107. สัญญาณการสื่อสารแบบอะซิงโครนัส

DEVICE	SIGNAL	ON/OFF	MEANING
คอมพิวเตอร์	DTR	+	นี่ โมเด็ม คุณพร้อมที่จะเชื่อมต่อกับระบบอื่นหรือไม่ ?
โมเด็ม	DSR	+	ใช่ ฉันพร้อมแล้ว ทำต่อ และหมุนเบอร์
โมเด็ม	DCD	+	ฉันมีระบบอื่นอยู่ในสาย
คอมพิวเตอร์	RTS	+	OK ฉันสามารถส่งข้อมูลได้หรือยัง ?
โมเด็ม	CTS	+	แน่นอน เริ่มเลย
คอมพิวเตอร์	TxD		กำลังส่งข้อมูลออกไปยังโมเด็ม
โมเด็ม	RxD		ฉันได้รับข้อมูลแล้ว
โมเด็ม	CTS	–	อย่าเพิ่งส่งข้อมูลเพิ่มเติม ฉันกำลังส่งมันออกไป...
โมเด็ม	CTS	+	OK ฉันพร้อมจะรับข้อมูลต่อแล้ว ส่งมาได้เลย !
ขั้นตอนการส่งข้อมูลอาจ ถูกทำซ้ำจนกระทั่ง... คอมพิวเตอร์	DTR	–	เรียบร้อยแล้ว! ทำต่อและวางหู
โมเด็ม	DCD	–	OK

ตารางที่ 107. สัญญาณการสื่อสารแบบอะซิงโครนัส (ต่อ)

DEVICE	SIGNAL	ON/OFF	MEANING
นี่เป็นสัญญาณการสื่อสารระหว่าง RS/6000 และโมเด็มเกี่ยวกับการได้รับสายเรียกเข้าจากระบบอื่นคอมพิวเตอร์	DTR	+	ฉันพร้อมแล้วและได้ "เปิดใช้งาน" พอร์ตสำหรับการหมุนโทรศัพท์
โมเด็ม	DSR	+	ฉันพร้อมแล้ว แต่ฉันกำลังรอการเรียกอยู่
บางคนเรียกเข้ามา! โมเด็ม	DCD	+	บางคนเรียกเข้ามา และอยู่ในสาย
โมเด็ม	CTS	+	ฉันได้รับข้อมูลจากกล่องอื่น ฉันสามารถส่งข้อมูลให้คุณได้หรือยัง ?
คอมพิวเตอร์	RTS	+	ฉันพร้อมที่จะรับแล้ว ทำต่อ และส่งได้
โมเด็ม	RxD		ส่งแล้ว!
โมเด็มยังคงส่งข้อมูลจนกระทั่ง... คอมพิวเตอร์	RTS	–	รอสักครู่! บัฟเฟอร์ฉันเต็ม อย่าเพิ่งส่งข้อมูลให้ฉัน
คอมพิวเตอร์	RTS	+	ตอนนี้ฉัน OK แล้ว ส่งข้อมูลให้ฉันต่อได้
โมเด็ม	DCD	–	การโทรจบแล้ว
คอมพิวเตอร์	DTR	–	OK กรุณาวางสาย

อ็อพชันของเทอร์มินัล stty-cxma

stty-cxma เป็นยูทิลิตีโปรแกรมที่ตั้งและแสดงอ็อพชันของเทอร์มินัลสำหรับ PCI 2-, 8- และ 128-พอร์ตอะแด็ปเตอร์ และอยู่ในไดเร็กทอรี `/usr/lbin/tty`

รูปแบบคือ:

```
stty-cxma [-a] [option(s)] [ttyname]
```

โดยไม่มีอ็อพชัน **stty-cxma** จะแสดงการตั้งค่าไดรเวอร์พิเศษทั้งหมด สัญญาณโมเด็ม และพารามิเตอร์มาตรฐานทั้งหมดที่ถูกแสดงโดย **stty(1)** สำหรับอุปกรณ์ tty ที่ถูกอ้างถึงโดยอินพุตมาตรฐาน อ็อพชันของคำสั่งถูกจัดเตรียมเพื่อเปลี่ยนการตั้งค่าโฟลว์คอนโทรล ตั้งอ็อพชันการพิมพ์แบบ transparent บังคับสายควบคุมโมเด็ม และแสดงการตั้งค่า tty ทั้งหมด อ็อพชันที่ไม่รู้จักใดๆจะถูกผ่านไปยัง **stty(1)** สำหรับการแปล อ็อพชันได้แก่:

-a จะแสดงการตั้งอ็อพชันของอะแด็ปเตอร์ที่เป็นหนึ่งเดียว พร้อมกับการตั้งค่า tty มาตรฐานที่ถูกรายงานโดยคำสั่ง **stty -a**

ttyname
ตั้งและแสดงอ็อพชันสำหรับอุปกรณ์ tty ที่ระบุ แทนที่จะเป็นอินพุตมาตรฐาน ฟอร์มนี้สามารถถูกใช้กับชื่อพาธของ tty ที่นำหน้าโดย **/dev/** หรือชื่อ tty แบบง่ายที่เริ่มต้นด้วย tty อ็อพชันนี้อาจถูกใช้บนสายควบคุมของโมเด็มเมื่อไม่มี carrier

อ็อพชันต่อไปนี้ระบุแอ็คชันชั่วคราวที่จะถูกทำทันที :

break ส่งสัญญาณ break 250 ms ออกไปบนสาย tty

flush ระบุการ flush (ลบ) ของ tty อินพุตและเอาต์พุตโดยทันที

flushin Flush tty อินพุตเท่านั้น

flushout
Flush tty เอาต์พุตเท่านั้น

อ็อพชันต่อไปนี้ระบุแอ็คชันที่รีเซ็ตเมื่ออุปกรณ์ถูกปิด อุปกรณ์จะใช้ค่าดีฟอลต์ครั้งต่อไปที่มันเปิด

stopout หยุดเอาต์พุตเหมือนกับว่าได้รับอักขระ XOFF

startout
รีสตาร์ทเอาต์พุตที่ถูกหยุดเหมือนกับว่าได้รับอักขระ XON

stopin เปิดใช้งานโฟลว์คอนโทรลเพื่อหยุดอินพุต

startin ปล่อยโฟลว์คอนโทรลเพื่อเริ่มต้นอินพุตที่ถูกหยุดใหม่

[−]dtr [drop]
ยกสายควบคุมโมเด็ม DTR แม้ว่า DTR ฮาร์ดแวร์โฟลว์คอนโทรลจะถูกเลือก

[−]rts [drop]
ยกสายควบคุมโมเด็ม RTS แม้ว่า RTS ฮาร์ดแวร์โฟลว์คอนโทรลจะถูกเลือก

อ็อพชันต่อไปนี้จะยังคงมีผลจนกว่าระบบจะถูกรีบูต หรือจนกว่าอ็อพชันจะถูกเปลี่ยน

[−]fastcook
ทำกระประมวลผล cooked output บนการ์ดแบบฉลาดเพื่อลดการใช้ CPU ของโฮสต์ และเพิ่มประสิทธิภาพของอินพุตโหมด raw

[−]fastbaud
สลับตารางอัตรา baud ดังนั้น 50 baud จะกลายเป็น 57,600 baud 75 baud จะกลายเป็น 76,800 baud 110 baud จะกลายเป็น 115,200 baud และ 200 baud จะกลายเป็น 230,000 baud สำหรับอุปกรณ์ที่สนับสนุน

[−]rtspace
ปิดใช้งาน/ปิดใช้งาน RTS ฮาร์ดแวร์อินพุตโฟลว์คอนโทรล ดังนั้น RTS จะดร็อบเพื่อหยุดการส่งของรีโมตชั่วคราว

[−]ctspace
เปิดใช้งาน/ปิดใช้งาน CTS ฮาร์ดแวร์เอาต์พุตโฟลว์คอนโทรล ดังนั้นการส่งแบบโลคัลจะถูกหยุดชั่วคราวเมื่อ CTS ดร็อบ

[−]dsrpace
เปิดใช้งาน/ปิดใช้งาน DSR ฮาร์ดแวร์เอาต์พุตโฟลว์คอนโทรล ดังนั้นการส่งแบบโลคัลจะถูกหยุดชั่วคราวเมื่อ DSR ดร็อบ

[−]dcdpace
เปิดใช้งาน/ปิดใช้งาน DCD ฮาร์ดแวร์เอาต์พุตโฟลว์คอนโทรล ดังนั้นการส่งแบบโลคัลจะถูกหยุดชั่วคราวเมื่อ DCD ดร็อบ

[-]dtrpace

ปิดใช้งาน/ปิดใช้งาน DTR ฮาร์ดแวร์อินพุตโฟลว์คอนโทรล ดังนั้น DTR จะตร็อบเพื่อหยุดการส่งของรีโมตชั่วคราว

[-]forcedcd

ปิดใช้งาน [เปิดใช้งานใหม่] carrier sense ดังนั้น tty อาจถูกเปิดและถูกใช้แม้ว่าเมื่อไม่มี carrier

[-]altpin

แม็พขาออกของตัวเชื่อมต่อ RJ-45 กับค่าของตัวเชื่อมต่อ 10-ขาแบบดีฟอลต์ หรือค่าตัวเชื่อมต่อ 8-ขา เมื่อพารามิเตอร์นี้เป็น **enabled** ตำแหน่งของ DSR และ DCD จะสลับกัน ดังนั้น DCD จะมีให้ใช้งานเมื่อใช้ตัวเชื่อมต่อ RJ-45 8-ขาแทนตัวเชื่อมต่อ RJ-45 10-ขา (ดีฟอลต์=**disable**)

ค่าที่เป็นไปได้:

enabled (จะระบุค่าตัวเชื่อมต่อ 8-ขา)

disable (จะระบุค่าตัวเชื่อมต่อ 10-ขา)

startc *c* ตั้งอักขระ XON โฟลว์คอนโทรล อักขระอาจถูกระบุเป็นเลขฐานสิบ เลขฐานแปด หรือเลขฐานสิบหก เลขฐานแปดจะถูกรู้จักโดยนำหน้าด้วยศูนย์ และเลขฐานสิบหกจะถูกนำหน้าด้วย 0x ตัวอย่างเช่น อักขระ XON มาตรฐาน CTRL-Q สามารถถูกใส่เป็น 17 (เลขฐานสิบ) 021 (เลขฐานแปด) หรือ 0x11 (เลขฐานสิบหก)

stopc*c* ตั้งอักขระ XOFF โฟลว์คอนโทรล อักขระอาจถูกระบุเป็นเลขฐานสิบ เลขฐานแปด หรือเลขฐานสิบหก (ดูที่ **startc** สำหรับรูปแบบของเลขฐานแปดและเลขฐานสิบหก)

astartc*c*

ตั้งอักขระ auxiliary XON โฟลว์คอนโทรล อักขระอาจถูกระบุเป็นเลขฐานสิบ เลขฐานแปด หรือเลขฐานสิบหก (ดูที่ **startc** สำหรับรูปแบบของเลขฐานแปดและเลขฐานสิบหก)

astopc*c* ตั้งอักขระ auxiliary XOFF โฟลว์คอนโทรล อักขระอาจถูกระบุเป็นเลขฐานสิบ เลขฐานแปด หรือเลขฐานสิบหก (ดูที่ **startc** สำหรับรูปแบบของเลขฐานแปดและเลขฐานสิบหก)

[-]aixon

เปิดใช้งาน auxiliary flow control ดังนั้น 2 อักขระที่เหมือนกันจะถูกใช้สำหรับ XON และ XOFF ถ้าได้รับอักขระ XOFF ทั้งสอง การส่งจะไม่ถูกเริ่มใหม่จนกว่าจะได้รับอักขระ XON ทั้งสอง

[-]2200flow

ใช้ 2200 style โฟลว์คอนโทรลบนพอร์ต เทอร์มินัล 2200 สนับสนุนเครื่องพิมพ์ที่ต่ออยู่ และใช้อักขระโฟลว์คอนโทรล 4 ตัว: terminal XON (0xF8), printer XON (0xF9), terminal XOFF (0xFA) และ XOFF (0xFB).

[-]2200print

กำหนดว่าอักขระโฟลว์คอนโทรลเหล่านี้จะถูกแปลอย่างไร ถ้าถูกตั้งเป็น 2200print รันโฟลว์คอนโทรลแบบอิสระสำหรับเทอร์มินัลและอุปกรณ์การพิมพ์แบบ transparent ไม่เช่นนั้น โฟลว์คอนโทรลของเทอร์มินัลและเครื่องพิมพ์จะเชื่อมกันแบบโลจิคัล ถ้าได้รับอักขระ XOFF เอาต์พุตทั้งหมดจะถูกหยุดชั่วคราวจนกว่าอักขระ XON ที่ตรงกันจะได้รับ

maxcps*n*

ตั้งอัตราอักขระสูงสุดต่อวินาที (cps) ที่อักขระจะถูกเอาต์พุตไปที่อุปกรณ์การพิมพ์แบบ transparent อัตราที่ถูกเลือกควรต่ำกว่าความเร็วเฉลี่ยของเครื่องพิมพ์เล็กน้อย ถ้าตัวเลขน้อยเกินไป ความเร็วของเครื่องพิมพ์จะถูกลดลง ถ้าตัวเลขสูงเกินไป เครื่องพิมพ์จะใช้โฟลว์คอนโทรล และเวลาที่ลองใหม่ของผู้ใช้จะลดลง ค่าดีฟอลต์คือ 100 cps

maxchar*n*

ตั้งจำนวนสูงสุดของอักขระการพิมพ์แบบ transparent ที่ไดรเวอร์ใส่ลงไปในเอาต์พุตคิว การลดจำนวนนี้จะเพิ่มโอเวอร์เฮดของระบบ การเพิ่มจำนวนนี้จะหน่วงเวลาการทำงานการ echo การกดคีย์เมื่อเครื่องพิมพ์แบบ transparent ค่าดีฟอลต์คือ 50 อักขระ

bufsize*n*

ตั้งค่าขนาดโดยประมาณของอินพุตบัฟเฟอร์ของไดรเวอร์ของเครื่องพิมพ์แบบ transparent หลังจากช่วงเวลาที่ไม่มีการทำงาน ไดรเวอร์จะส่งอักขระจำนวนมากไปยังเครื่องพิมพ์แบบ transparent ก่อนที่จะลดลงเป็นอัตราของ maxcps ค่าดีฟอลต์คือ 100

onstr*s* ตั้งลำดับของ escape ของเทอร์มินัลเพื่อเปิดเครื่องพิมพ์แบบ transparent สตริงสามารถประกอบด้วยอักขระการพิมพ์ ASCII มาตรฐานและอักขระที่ไม่สามารถพิมพ์ได้ อักขระควบคุม (ไม่สามารถพิมพ์ได้) ต้องถูกใส่โดยใช้ค่าเลขฐานแปดของมัน และต้องมี 3 หลักที่นำหน้าด้วยแบ็กสแลช ตัวอย่างเช่น อักขระ Escape 33 เลขฐานแปด ควรใส่เป็น \033 ถ้าเครื่องพิมพ์แบบ transparent ถูกเปิดโดยสตริง <Esc>[5i (มาตรฐาน ANSI) มันจะถูกใส่เป็น :\033[5i

offstr*s* ตั้งลำดับของ escape ของเทอร์มินัลเพื่อปิดเครื่องพิมพ์แบบ transparent อ้างถึง **onstr***s* สำหรับรูปแบบของสตริง

term*t* ตั้งสตริงการเปิด/ปิด เครื่องพิมพ์แบบ transparent เป็นค่าที่พบในตารางแบบดีฟอลต์ภายใน ตารางดีฟอลต์ภายในถูกใช้สำหรับเทอร์มินัลต่อไปนี้ : adm31, ansi, dg200, dg210, hz1500, mc5, microterm, multiterm, pcterm, tvi, vp-a2, vp-60, vt52, vt100, vt220, wyse30, wyse50, wyse60 หรือ wyse75 ถ้าชนิดของเทอร์มินัลไม่พบในตารางดีฟอลต์ภายใน ditty จะอ่าน entry ของ terminfo สำหรับชนิดของเทอร์มินัลและตั้งสตริงการเปิด/ปิด เครื่องพิมพ์แบบ transparent เป็นค่าที่ให้โดยแอ็ตทริบิวต์ mc5/mc4 ที่พบใน entry ของ terminfo

ระบบย่อยของ Asynchronous Point-to-Point Protocol

ระบบย่อยของ**Asynchronous Point-to-Point Protocol (PPP)** จัดเตรียมทางเลือกให้กับ SLIP

PPP จัดเตรียมวิธีมาตรฐานสำหรับการส่งดาตาแกรมแบบหลายโปรโตคอลผ่านสื่อแบบจุดต่อจุด **PPP** ประกอบด้วย 3 เลเยอร์หลัก:

1. วิธีสำหรับการ encapsulate ดาตาแกรมแบบหลายโปรโตคอล **PPP** สนับสนุนโปรโตคอล TCP/IP network layer
2. **Link Control Protocol (LCP)** สำหรับการเริ่มต้น ตั้งค่า และทดสอบการเชื่อมต่อ data-link **PPP** ใช้สิ่งนี้ผ่านสตรีมของส่วนขยายของเคอร์เนล
3. ตระกูลของ **Network Control Protocols (NCPs)** สำหรับการเริ่ม และตั้งค่าโปรโตคอล network layer อื่น **PPP** สนับสนุน **Internet Protocol Control Protocol (IPCP/IPv6CP)** สำหรับการต่อรองการเชื่อมต่อ TCP/IP

การใช้งาน **PPP** สนับสนุน Request for Comments (RFCs) ต่อไปนี้:

- RFC 1661, *Point-to-Point Protocol, LCP*
- RFC 1332, *PPP Internet Protocol Control Protocol (IPCP)*
- RFC 1662, *PPP in HDLC-like Framing*
- RFC 1334, *PPP Authentication Protocols*
- RFC 1990, *PPP Multilink*
- RFC 2472, *IP Version 6 over PPP*

PPP แตกต่างกันระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ ระบบปฏิบัติการนี้ทำหน้าที่เป็นทั้งไคลเอ็นต์และเซิร์ฟเวอร์ ความแตกต่างกันทำให้ง่ายในการตั้งค่า **PPP** เซิร์ฟเวอร์จะจัดสรรพูลของแอดเดรส IP/IPv6CP ในจำนวนของการเชื่อมต่อที่มี มีความสัมพันธ์บางอย่างระหว่างอุปกรณ์สื่อ การใช้ **PPP** นี้จะทำลายความสัมพันธ์นี้ การเชื่อมต่อ **PPP** ของเซิร์ฟเวอร์ทั้งหมดจะถูกจัดสรรบนพื้นฐานของการพร้อมใช้งานก่อน นี่ทำให้สะดวกในการแยก **PPP** จากสื่อ กระบวนการ attachment ต้องร้องขอเพื่อจะถูกลิงก์กับชนิดของลิงก์ที่ถูกต้อง

กระบวนการ PPP ระดับผู้ใช้

โปรโตคอลอะซิงโครนัสแบบจุดต่อจุด บนระบบปฏิบัติการนี้ใช้กระบวนการระดับผู้ใช้ 3 กระบวนการ

1. daemon การควบคุม (**pppcontrold**) รันโดย root ภายใต้ System Resource Controller (**startsrc -s pppcontrold**) ฟังก์ชันของ daemon การควบคุมจะรวมการโหลดและการตั้งค่าส่วนขยายของเคอร์เนลทั้งหมดที่เกี่ยวข้องกับระบบย่อย มันจะยังคงรันตราบใดที่ฟังก์ชัน **PPP** ยังถูกใช้โดยระบบปฏิบัติการ
2. กระบวนการ attachment (**pppattachd**) ที่เชื่อมสตรีมของ TTY เข้ากับอินสแตนซ์ของโปรโตคอล **Link Control Protocol Network Control Protocol** และ datagram อินสแตนซ์ **pppattachd** จะมีอยู่สำหรับแต่ละการเชื่อมต่อของ **PPP** ที่แอ็คทีฟในระบบ ผู้ใช้ของกระบวนการ attachment process ต้องเป็นของกลุ่ม **uucp** และประกอบด้วย `/usr/sbin` ภายในตัวแปรสภาวะแวดล้อม **PATH**
3. กระบวนการ dialer (**pppdial**) ที่สร้างการเชื่อมต่อขาออก dialer ถูกดำเนินการโดย **pppattachd** เป็นโปรแกรมตัวเชื่อมต่อวัตถุประสงค์ของมันคือทำงานร่วมกับอุปกรณ์อะซิงโครนัสก่อนที่จะมีการต่อรองแบบ **PPP** การทำงานร่วมกันนี้ถูกกำหนดเหมือนกับรูปแบบของไดอะล็อกของ UUCP chat dialer สามารถให้ความช่วยเหลือในการสร้างการเชื่อมต่อกับระบบรีโมต การสร้างเซสชันจริงๆอยู่นอกเหนือขอบเขตของ **PPP**

การตั้งค่า Asynchronous Point-to-Point Protocol

คุณสามารถใช้ SMIT เพื่อกำหนดค่า **โปรโตคอล Point-to-Point อะซิงโครนัส**

ตารางต่อไปนี้แสดงงานทั้งหมดที่คุณอาจต้องใช้เมื่อตั้งค่าระบบของคุณ คุณต้องมีสิทธิของ root เพื่อทำงานในตารางนี้

อย่าน้อยที่สุด เมื่อคุณเริ่มต้นตั้งค่าระบบของคุณ คุณต้องเลือกงานต่อไปนี้จากตาราง :

- เพิ่ม Link Configuration
- เพื่อ Server Interface (ถ้าคุณตั้งเครื่องเป็น PPP เซิร์ฟเวอร์)
- เพื่อม Demand Interface (ถ้าคุณต้องการให้เครื่องสนับสนุนการเชื่อมต่อแบบตามต้องการ)
- จัดการกับ ผู้ใช้/รหัสผ่านของ PAP or CHAP (ถ้าคุณต้องการให้เครื่องสนับสนุนการพิสูจน์ตัวตนแบบ **PPP**)
- สตาร์ท **PPP** เพื่อให้การเปลี่ยนแปลงมีผลใช้งาน (หรือหยุดการทำงาน จากนั้น สตาร์ท **PPP** ถ้า **PPP** กำลังทำงานอยู่)

ตารางที่ 108. งานการตั้งค่าอะซิงโครนัส PPP

งาน	วิธีลัด SMIT
สร้าง Link Control Configuration	`smit ppplcp`
เพิ่ม Link Configuration	`smit addlcp`
แก้ไข/แสดง Link Configuration	`smit chglcp`
ลบ Link Configuration[1]	`smit rmlcp`
สร้าง PPP IP Interfaces	`smit pppip`
เพิ่ม Server Interface	`smit addpppserver`

ตารางที่ 108. งานการตั้งค่าอะซิงโครนัส PPP (ต่อ)

งาน	วิธีลัด SMIT
แก้ไข/แสดง Server Interface	smit listserver
ลบ Server Interface[1]	smit rmlistserver
เพิ่ม Demand Interface	smit addpppdemand
แก้ไข/แสดง Demand Interface	smit listdemand
ลบ Demand Interface[1]	smit rmlistdemand
จัดการกับผู้ใช้/รหัสผ่านของ PAP	smit ppppap
เพิ่ม PAP User	smit addpapuser
แก้ไข/แสดง PAP User	smit listpapuser
ลบ PAP User	smit rmpapuser
จัดการกับผู้ใช้/รหัสผ่านของ CHAP	smit pppchap
เพิ่ม CHAP User	smit addchapuser
แก้ไข/แสดง CHAP User	smit listchapuser
ลบ CHAP User	smit rmchapuser
สตาร์ท **PPP**[2]	smit startppp
หยุดการทำงาน **PPP**[3]	smit stopppp
PPP IPv6 Interfaces	smit pppipv6
เพิ่ม **PPP** IPv6 Server Interface	smit addpppv6server
แสดง หรือแก้ไข **PPP** IPv6 interface	smit listv6server
ลบ **PPP** IPv6 interface	smit rmlistv6server
เพิ่ม **PPP** IPv6 client interface	smit addpppv6client
แสดง หรือแก้ไข **PPP** IPv6 client interface	smit listpppv6client
ลบ **PPP** IPv6 client interface	smit rmlistpppv6client
เพิ่ม **PPP** IPv6 demand interface	smit addpppv6demand
แสดง หรือแก้ไข **PPP** IPv6 demand interface	smit listpppv6demand
ลบ **PPP** IPv6 demand interface	smit rmlistpppv6demand
PPP IP and IPv6 Interfaces	smit pppipv4_6
เพิ่ม **PPP** IP/IPv6 Server Interface	smit addpppv4_6server
แสดง หรือแก้ไข **PPP** IP/IPv6 interface	smit listv4_6server
ลบ **PPP** IP/IPv6 interface	smit rmlistv4_6server
เพิ่ม **PPP** IP/IPv6 client interface	smit addpppv4_6client
แสดง หรือแก้ไข **PPP** IP/IPv6 client interface	smit listpppv4_6client

ตารางที่ 108. งานการตั้งค่าอะซิงโครนัส PPP (ต่อ)

งาน	วิธีลัด SMIT
ลบ **PPP** IP/IPv6 client interface.	`smit rmlistpppv4_6client`
เพิ่ม **PPP** IP/IPv6 demand interface	`smit addpppv4_6demand`
แสดง หรือแก้ไข **PPP** IP/IPv6 demand interface	`smit listpppv4_6demand`
ลบ **PPP** IP/IPv6 demand interface	`smit rmlistpppv4_6demand`

หมายเหตุ:

1. เลือกงานนี้จะทำลายข้อมูลที่มีอยู่
2. วิธีอื่นที่จะสตาร์ท **PPP** คือการใช้คำสั่ง **startsrc -s pppcontrold** อย่างไรก็ตาม SMIT อินเตอร์เฟสยังยอมให้คุณตั้ง **PPP** เพื่อสตาร์ทเมื่อเวลาบูต
3. วิธีอื่นที่จะหยุด **PPP** คือการใช้คำสั่ง **stopsrc -s pppcontrold** อย่างไรก็ตาม SMIT อินเตอร์เฟสยังยอมให้คุณตั้งไม่ให้ **PPP** สตาร์ทเมื่อเวลาบูต

การเปิดใช้งาน PPP SNMP

PPP สามารถทำงานกับ TCP/IP SNMP daemon เพื่อรายงานข้อมูลการตั้งค่า PPP link layer cพร้อมกับข้อมูลเกี่ยวกับอินเตอร์เฟส **Link Control Protocol (LCP)** ที่แอ็คทีฟ

ทำให้ทั้ง TCP/IP SNMP และซอร์ฟแวร์การจัดการ SNMP ถูกตั้งค่าอย่างถูกต้อง **PPP** เปิดใช้งาน SNMP :

- การดึงข้อมูล **PPP** Link Configuration (เช่น Maximum Receive Unit size และ Asynchronous Character Mapping)
- การตั้งค่าข้อมูล **PPP** Link Configuration
- การดึงข้อมูลของ LCP อินเตอร์เฟสสำหรับ LCP ลิงก์ที่แอ็คทีฟ
- การเปลี่ยนสถานะของ LCP ลิงก์ที่แอ็คทีฟ สามารถถูกเปลี่ยนเป็น "down" โดยการตั้งอ็อบเจ็กต์ **ifAdminStatus** Management Information Base (MIB) ที่เหมาะสม

ไม่ใช่อ็อบเจ็กต์ทั้งหมดที่ถูกกำหนดโดย RFC1471 สำหรับ **PPP** MIB ที่จะได้รับการสนับสนุน เฉพาะตาราง **pppLink** ที่ใช้ได้กับระบบย่อย **PPP** ดังนั้นส่วนของ **pppLqr** และ **pppTests** ไม่ได้รับการสนับสนุน ส่วนของ **pppLink** ได้รับการสนับสนุนโดยมีข้อยกเว้นดังต่อไปนี้ :

- อ็อบเจ็กต์ **pppLinkConfigMagicNumber** จะอ่านได้อย่างเดียว ใน **PPP** การต่อรองโดยตัวเลขมหัศจรรย์จะถูกทำเสมอและไม่สามารถปิดใช้งานได้
- อ็อบเจ็กต์ **pppLinkConfigFcsSize** จะอ่านได้อย่างเดียว **PPP** สนับสนุน FCS ขนาด 16 กับระบบปฏิบัติการนี้เท่านั้น

โดยดีฟอลต์ SNMP สำหรับ **PPP** จะถูกปิดใช้งาน เมื่อต้องการเปิดใช้งาน **PPP** SNMP คุณสามารถใช้โพรซีเดอร์ต่อไปนี้ คุณต้องมีสิทธิของ root เพื่อทำโพรซีเดอร์นี้

หมายเหตุ: โพรซีเดอร์ต่อไปนี้สันนิษฐานว่า **PPP** Link Configuration ถูกตั้งค่าเรียบร้อยแล้ว ถ้าไม่ ทำโพรซีเดอร์ที่อธิบายใน "การตั้งค่า Asynchronous Point-to-Point Protocol" ในหน้า 660 ก่อนที่จะเปิดใช้งาน **PPP** SNMP

1. สตาร์ท SMIT อินเตอร์เฟสและแสดงหน้าจอ Change/Show a Link Configuration โดยใส่ :

   ```
   smit chglcp
   ```

2. เปลี่ยนฟิลด์ Enable **PPP** SNMP subagent เป็น yes
3. ยอมรับการเปลี่ยนแปลงของคุณและออกจาก SMIT

PPP SNMP จะไม่ถูกเปิดใช้งานจนกว่า **PPP** จะถูกรีสตาร์ท

- ถ้า **PPP** กำลังทำงานอยู่
  1. หยุดการทำงาน **PPP** โดยใช้ `smit stopppp` fast path (ดูตารางใน "การตั้งค่า Asynchronous Point-to-Point Protocol" ในหน้า 660)
  2. ตรวจสอบบ่อยๆเพื่อดูว่าระบบย่อยถูกปิดการทำงานเรียบร้อยหรือยังโดยการใช้ :

     ```
     lssrc -s pppcontrold
     ```

     จำนวนเวลาที่ใช้เพื่อหยุดการทำงานของระบบย่อยขึ้นอยู่กับจำนวนของลิงก์ที่ถูกกำหนดในการตั้งค่า **PPP** ระบบย่อยจะถูกปิดการทำงานสมบูรณ์เมื่อเอาต์พุตของคำสั่งนี้แสดงสถานะเป็น `inoperative`
  3. สตาร์ท **PPP** โดยใช้ `smit startppp` fast path (ดูตารางใน "การตั้งค่า Asynchronous Point-to-Point Protocol" ในหน้า 660)
- ถ้า **PPP** ไม่ได้ทำงานอยู่ สตาร์ท **PPP** โดยใช้ `smit startppp` fast path (ดูตารางใน "การตั้งค่า Asynchronous Point-to-Point Protocol" ในหน้า 660)

Serial Line Internet Protocol

Serial Line Internet Protocol (SLIP) เป็นโปรโตคอลที่ TCP/IP ใช้เมื่อทำงานผ่านการเชื่อมต่อแบบซีเรียล

มันถูกใช้ทั่วไปบนซีเรียลลิงก์เฉพาะและการเชื่อมต่อแบบหมุนโทรศัพท์ที่ทำงานที่ความเร็วระหว่าง 1200bps และ 19.2Kbps หรือสูงกว่า

หมายเหตุ: เพื่อที่จะใช้อัตรา baud ที่สูงกว่า 38400 ระบุอัตรา baud เป็น 50 ในไฟล์ `/etc/uucp/Devices` สำหรับ tty ที่ต้องการ จากนั้นเปลี่ยนการตั้งค่า SMIT สำหรับ tty นั้นเพื่อแสดงอัตรา baud ที่แท้จริงที่ต้องการ

ตัวอย่างเช่น เพื่อรันคำสั่ง **cu** บน tty0 ด้วยอัตรา baud เป็น 115200 ใช้โพรซีเดออร์ต่อไปนี้ :

1. ต้องแน่ใจว่าฮาร์ดแวร์สนับสนุนอัตรา baud นั้น
2. แก้ไขไฟล์ `/etc/uucp/Devices` เพื่อเพิ่มบรรทัดต่อไปนี้ :

   ```
   Direct tty0 - 50 direct
   ```
3. ใส่ `smit chtty` fast path
4. เลือก tty0
5. เปลี่ยนอัตรา baud เป็น 115200
6. ออกจาก SMIT

การตั้งค่า SLIP

มีขั้นตอนที่แนะนำ / ขั้นตอนสำระหว่างการตั้งค่า **SLIP**

การใช้วิธีแบบ 2 ขั้นตอนจะแยกข้อกำหนดของฮาร์ดแวร์และการตั้งค่าที่ขึ้นอยู่กับเครื่องจากซอฟต์แวร์ SLIP และปัญหาของ syntax ของคำสั่ง

1. ใช้ ATE หรือยูทิลิตี **cu** เพื่อล็อกอินที่ระบบรีโมต นี่จะพิสูจน์ความสามารถใช้ได้และความถูกต้องของฟิสิคัลลิงก์

ข้อสำคัญ คือตรวจสอบการทำงานได้ของโมเด็มที่เกี่ยวข้องในลิงก์ **SLIP** เนื่องจากเป็นสาเหตุของปัญหาซึ่งพบบ่อยที่สุดระหว่างระยะเซ็ตอัพ

2. หลังจากคุณล็อกอินเข้าสู่ระบบรีโมตโดยไม่มีข้อผิดพลาด โดยใช้ ATE หรือคำสั่ง **cu** คุณสามารถเริ่มต้นคอนฟิกูเรชัน **SLIP**

ข้อพิจารณาเกี่ยวกับ SLIP โมเด็ม

เมื่อตั้งค่าโมเด็มสำหรับ **SLIP** จำเป็นที่การเปลี่ยนแปลงเหล่านี้ต้องถูกทำบนทั้งสองด้านของลิงก์การสื่อสาร

ทั้งโมเด็มแบบโลคัลและรีโมตต้องถูกตั้งค่าเหมือนกัน

1. โมเด็มต้องรับรู้ถึงการมีอยู่ของ DTR

   เมื่อกล่าวถึงโมเด็มที่โลคัล ถ้า DTR ถูกสันนิษฐานเอา หรือไม่ได้รับการสนใจ โมเด็มจะไม่สามารถวางสายได้ มันจะสามารถยกเลิกสายหรือวางหูเฉพาะเมื่อมันรู้ว่า carrier หายไปจากด้านอื่น นี่หมายความว่าการวางหูจะต้องทำจากด้านอื่นเท่านั้น คำสั่ง AT &D2 หรือ &D3 จะถูกตั้งค่าอย่างถูกต้องสำหรับโมเด็มแบบ Hayes-compatible ส่วนใหญ่

2. โมเด็มต้องไม่ถูกบังคับ สันนิษฐาน หรือไม่สนใจ data carrier detect (DCD)

   DCD ต้องทำตาม หรือ ติดตามสภาวะที่เป็นจริง นี่หมายความว่า carrier จะมีอยู่หลังจากการเชื่อมต่อที่แท้จริงไปยังด้านอื่น (โมเด็ม) ข้ามชุมสายโทรศัพท์ นี่ยังใช้ได้กับสายโดยเฉพาะ &C1 เป็นค่าตั้งที่แนะนำสำหรับโมเด็มแบบ Hayes-compatible ส่วนใหญ่

3. โมเด็มต้องไม่ถูกบังคับ สันนิษฐาน หรือไม่สนใจสัญญาณ clear to send (CTS)

   CTS ต้องติดตามหรือตาม request To send (RTS) CTS ถูกบังคับให้เป็น true พอร์ตที่เปิดจะล้มเหลวเมื่อ **getty** ถูกใส่ไปที่พอร์ต หรือเมื่อโปรโตคอล RTS โฟลว์คอนโทรลถูกเพิ่มเข้ากับพอร์ต

4. โมเด็มควรถูกตั้งค่าเพื่อปิดโค้ด automatic repeat request (ARQ) ถ้าปัญหาเกิดขึ้นระหว่างความพยายามหมุน **slattach**

   ถ้าโมเด็มยังคงล้มเหลวในการทำการเชื่อมต่อระหว่างความพยายามหมุน **slattach** ผู้ใช้ควรตรวจสอบการตั้งค่าโมเด็มและปิดโค้ด ARQ ถ้ามันยังเปิดอยู่ ในโมเด็มแบบ Hayes-compatible ส่วนใหญ่ นี่จะเป็นการตั้งค่า &A0

   การปิดใช้งานโค้ดผลลัพธ์ของ ARQ จะไม่มีผลกับการเชื่อมต่อที่มีการควบคุมข้อผิดพลาด หรือทำให้โมเด็มไม่ส่งข้อความมาตรฐาน CONNECT (ถ้าโค้ดผลลัพธ์ถูกเปิดใช้งาน) ที่ต้องการสำหรับสตริงการหมุน **slattach**

5. ECL (Error Checking on the Link) มีความวิกฤต

   ทั้งสองหรือไม่มีโมเด็มที่สามารถใช้มัน โดยทั่วไปโมเด็มทั้งสองต้องตกลงที่จะใช้มันระหว่างเซสชันการเชื่อมต่อ ถ้า ECL ถูกเลือก สายโทรศัพท์แบบฟิสิคัลต้องดีพอที่จะให้การกู้ข้อมูลที่มีข้อผิดพลาดก่อนที่การจับเวลา TCP/IP จะหมดเวลาขณะที่รอแพ็กเก็ตการตอบรับสำหรับข้อมูลล่าสุดที่ถูกส่งข้าม **SLIP** ลิงก์

6. การบีบอัดข้อมูลข้ามลิงก์

   มันสามารถยอมรับได้ที่จะใช้บีบอัดข้อมูลข้างลิงก์ตราบใดที่มันสามารถได้รับการจัดการทั้งหมดโดยโมเด็ม **SLIP** จะไม่ทำการบีบอัดข้อมูลชนิดใดๆ ถ้าการบีบอัดข้อมูลถูกใช้ มันจะดีกว่าที่มีโมเด็มสองตัวที่เป็นชนิดเดียวกัน นี่จะทำให้แน่ใจว่าแต่ละตัวจะทำการบีบอัดโดยวิธีเดียวกันและใช้กรอบเวลาเดียวกัน

การโปรแกรมโมเด็มแบบแมนวลโดยใช้คำสั่ง cu

ใช้โพรซีเดอร์ต่อไปนี้เพื่อโปรแกรมโมเด็มที่ต่ออยู่กับระบบแบบแมนวล

- UNIX-to-UNIX Copy Program (UUCP) ต้องถูกติดตั้งบนระบบ ใช้คำสั่ง **lslpp -f | grep bos.net.UUCP** เพื่อตรวจสอบการติดตั้ง
- โมเด็มต้องต่ออยู่กับระบบและเปิดอยู่

- ผู้ใช้ที่มีสิทธิเป็น root ต้องใช้สำหรับแก้ไขไฟล์ที่เหมาะสม

1. เพิ่มบรรทัดต่อไปนี้เข้ากับไฟล์ /etc/uucp/Devices ถ้ามันไม่มีอยู่ (แทนที่ # ด้วยหมายเลขของพอร์ตของคุณ)

   ```
   Direct tty# - Any direct
   ```

   หมายเหตุ: บรรทัดใดๆในไฟล์ Devices ที่เริ่มต้นด้วย # ในคอลัมน์ซ้ายสุดคือหมายเหตุ
2. บันทึกและออกจากไฟล์
3. พิมพ์คำสั่งต่อไปนี้บนบรรทัดรับคำสั่ง :

   ```
   cu -ml tty#
   ```
4. ข้อความว่าเชื่อมต่อแล้วควรจะแสดงบนหน้าจอระบุว่าโมเด็มถูกเชื่อมต่อและพร้อมที่จะถูกโปรแกรม
5. พิมพ์ AT และกด Enter โมเด็มจะตอบสนองด้วย OK ถ้าไม่มีการตอบสนองจากโมเด็ม หรือถ้าอักขระที่ถูกพิมพ์ไม่ปรากฏบนหน้าจอ ให้ตรวจสอบต่อไปนี้ :
   - ตรวจสอบการเชื่อมต่อสายเคเบิล
   - ตรวจสอบว่าโมเด็มเปิดอยู่
   - สังเกตไฟด้านหน้าของโมเด็มเมื่อคุณกด Enter ถ้าไฟ Receive Data (RD) และ Send Data (SD) กระพริบ ดังนั้นโมเด็มกำลังสื่อสารกับระบบและปัญหาอาจเกิดจากการตั้งค่าโมเด็มปัจจุบัน ถ้าไฟไม่กระพริบ ดังนั้นปัญหามาจากการเชื่อมต่อของโมเด็ม
   - พิมพ์ต่อไปนี้ และดูว่าสถานะเปลี่ยนแปลงหรือไม่ :

     ```
     ATE1 <enter>
     ATQ0 <enter>
     ```

     ATE1 จะกลายเป็นโหมด echo ที่จะแสดงอักขระที่ถูกพิมพ์บนหน้าจอ ATQ0 จะเปิดใช้งานการแสดงโค้ดของผลลัพธ์
6. โปรแกรมโมเด็มโดยใช้การตั้งค่าที่แสดงในส่วนก่อนหน้านี้ "ข้อพิจารณาเกี่ยวกับโมเด็ม" ตัวอย่างต่อไปนี้แสดงวิธีโปรแกรมและบันทึกการตั้งค่าพื้นฐานสำหรับโมเด็ม Hayes-compatible ให้ป้อน:

   ```
   AT&F    <enter>
   AT&D2   <enter>
   ATS0=1  <enter>
   ATS9=12 <enter>
   AT&C1   <enter>
   AT&W    <enter>
   ~.      <enter>
   ```

   โดยที่ &F ใช้เพื่อรีเซ็ตโมเด็มเป็นค่าดีฟอลต์จากโรงงาน &D2 ตั้ง DTR, S0 และ S9 ตั้งค่า register, &C1 ตั้ง carrier และ &W จะเขียนค่าที่ตั้งไปยังโมเด็ม tilde-period จะสิ้นสุดการเชื่อมต่อ

การตั้งค่าโมเด็มแบบอัตโนมัติ

ผู้ใช้สามารถปรับแต่งโมเด็มแบบแมนวล หรือใช้ยูทิลิตี **cu** พร้อมกับไฟล์ที่เกี่ยวข้องของมันเพื่อสร้างสคริปต์การตั้งค่าโมเด็มแบบอัตโนมัติ

- UUCP ต้องถูกติดตั้งบนระบบ ใช้คำสั่ง **lslpp -f | grep bos.net.UUCP** เพื่อตรวจสอบการติดตั้ง
- โมเด็มต้องต่ออยู่กับระบบและเปิดอยู่
- สตริงคำสั่ง AT ของโมเด็มต้องมีอยู่แล้ว (ตัวอย่างเช่น at&f&c1&d3) ผู้ใช้ไม่ควรพยายามตั้งค่าโมเด็มแบบอัตโนมัติ จนกว่าสตริงคำสั่งได้ถูกลองใช้แบบแมนวลโดยใช้คำสั่ง **cu**
- ผู้ใช้ที่มีสิทธิเป็น root ต้องใช้สำหรับแก้ไขไฟล์ที่เหมาะสม

ตัวอย่างต่อไปนี้แสดงวิธีการกำหนดคอนฟิกโมเด็ม Telebit T3000 ที่เชื่อมอยู่กับ tty0 โดยอัตโนมัติ

1. แก้ไขไฟล์ /etc/uucp/Systems
2. เพิ่มบรรทัดต่อไปนี้ที่ท้ายของไฟล์ entry ควรเริ่มที่คอลัมน์ซ้ายสุดของไฟล์

   ```
   telebit Nvr TELEPROG 19200
   ```

3. บันทึกและออกจากไฟล์
4. แก้ไขไฟล์ /etc/uucp/Devices
5. เพิ่มบรรทัดต่อไปนี้ที่ท้ายของไฟล์ entry ควรเริ่มที่คอลัมน์ซ้ายสุดของไฟล์

   ```
   TELEPROG tty0 - 19200 TelebitProgram
   ```

6. บันทึกและออกจากไฟล์
7. แก้ไขไฟล์ /etc/uucp/Dialers
8. เพิ่มบรรทัดต่อไปนี้ที่ท้ายของไฟล์ entries ควรเริ่มที่คอลัมน์ซ้ายสุดของไฟล์

   หมายเหตุ: 4 บรรทัดต่อไปนี้ควรทำเป็นบรรทัดเดียวแบบยาว :

   ```
   TelebitProgram  =,-,    "" \dAT&F\r\c OK
   ats0=1s2=255s7=60s11=50s41=2s45=255s51=252s63=1s58=2s64=1\r\c OK
   ATs69=2s105=0s111=30s255=0M0&C1Q2&D3&Q0&R3&S1&T5\r\c OK
   ATE0X12&W\r\c OK
   ```

9. บันทึกและออกจากไฟล์
10. เพื่อเริ่มการตั้งค่าแบบอัตโนมัติ พิมพ์คำสั่งต่อไปนี้:

    ```
    cu -d telebit
    ```

    คำสั่งจะล้มเหลวถ้าคุณไม่ได้เชื่อมต่อกับระบบ ดูที่เอาต์พุตของการดีบักของคำสั่งเพื่อดูว่า ATE0X12&W ถูกส่งไปยังโมเด็ม และได้รับการตอบรับด้วย OK ถ้าเป็นเช่นนั้น โมเด็มได้ถูกโปรแกรมเรียบร้อยแล้ว

อาจเกิดปัญหาขึ้นได้เนื่องจากใส่ค่าที่ไม่ถูกต้องในไฟล์ Dialers หรือเนื่องจากคอนฟิกูเรชันที่มีอยู่แล้วของโมเด็ม ถ้าสิ่งนี้เกิดขึ้น พยายามโปรแกรมแบบแมนวล และใส่สตริง dialers (ในขั้นตอนที่ 8) ทีละขั้นตอน

การตั้งค่า SLIP ผ่านโมเด็ม

เมื่อต้องการกำหนดคอนฟิก **Serial Line Interface Protocol (SLIP)** ระหว่าง สองระบบที่สื่อสารผ่านโมเด็ม คุณสามารถใช้โพรซีเดอร์นี้ ซึ่งเป็นทางเลือกระหว่างอินเตอร์เฟส System Management Interface Tool (SMIT) และบรรทัดรับคำสั่งเพื่อดำเนินการกำหนดคอนฟิกให้สมบูรณ์

เพื่อความชัดเจน วิธีการต่อไปนี้จะใช้ชื่อ bronze และ gold สำหรับโฮสต์ 2 ตัว

1. เชื่อมต่อโมเด็มเข้ากับ bronze และ gold
2. เพื่อสร้าง tty บนโดยใช้ SMIT ทำตามขั้นตอนเหล่านี้ :
   a. พิมพ์:

      ```
      smit maktty
      ```

   b. เลือก **rs232** เป็นชนิดของ tty ที่คุณต้องการสร้าง
   c. เลือกซีเรียลพอร์ตที่พร้อมใช้งาน ตัวอย่างเช่น sa0 (ซีเรียลพอร์ต 1 ของระบบ)
   d. เลือกหมายเลขพอร์ตสำหรับ tty นี้จากลิสต์
   e. ตั้งอัตรา BAUD เป็นอัตรา baud ของโมเด็มของคุณ

f. ตั้ง Enable LOGIN เป็น disable

g. ออกจาก SMIT

3. สร้าง tty บน gold

   ทำโพรซีเดอร์เดียวกันที่คุณใช้กับ bronze (ในขั้นตอนที่ 2) ยกเว้นตั้ง Enable LOGIN เป็น **enable**

   ส่วนที่เหลือของวิธีการเหล่านี้สันนิษฐานว่าหมายเลขของ tty บนทั้ง bronze และ gold คือ tty1

4. ทดสอบการเชื่อมต่อแบบฟิสิคัลด้วย ATE

   a. บน bronze **พิมพ์** :

      ```
      ate
      ```

   b. ที่เมนูหลัก Unconnected เลือกคำสั่งย่อย **Alter** ตั้ง Rate เป็นอัตรา baud ของโมเด็มของคุณและ Device เป็น tty1

   c. ที่เมนูหลัก Unconnected เลือกคำสั่งย่อย **Connect** เมื่อได้รับพร้อมต์ของ ATE เพื่อให้ใส่หมายเลขโทรศัพท์ ใส่หมายเลขโทรศัพท์ของ gold และกด Enter

   d. ตอนนี้ คุณควรได้รับพร้อมต์สำหรับล็อกอินสำหรับ gold ล็อกอิน

   e. กลับไปที่หน้า connected ล็อกเอาต์จาก gold กด Ctrl-v (เพื่อให้ได้รับ ATE CONNECTED MAIN MENU) กดคีย์ T เพื่อยกเลิกการเชื่อมต่อ และกดคีย์ Q เพื่อออกจาก ATE

   หมายเหตุ: ถ้าคุณไม่ได้รับพร้อมต์สำหรับล็อกอิน กลับไปที่ขั้นตอนที่ 1 และตรวจสอบว่าการตั้งค่าของคุณถูกต้อง ห้ามทำต่อจนกว่าคุณจะล็อกอินกับ gold

5. เนื่องจากการตั้งค่า tty สำหรับใช้กับ ATE จะแตกต่างกับการตั้งค่าสำหรับใช้กับ SLIP เล็กน้อย คุณต้องทำการเปลี่ยนแปลงต่อไปนี้ :

   a. บน bronze **พิมพ์** :

      ```
      smit chgtty
      ```

   b. บน gold **พิมพ์** :

      ```
      smit chgtty-pdisable tty1
      ```

   c. เลือก **tty1** จากนั้นเลือก **Change/Show TTY Program**

   d. ตั้ง Enable LOGIN เป็น disable จากนั้นออกจาก SMIT

6. เพิ่มบรรทัดต่อไปนี้เข้ากับไฟล์ `/usr/lib/uucp/Devices` บนทั้ง bronze และ gold:

   ```
   Direct tty1 - 9600 direct
   ```

   หรือ แทนที่ `9600` ด้วยความเร็วของโมเด็มของคุณ

7. สร้าง **SLIP** เน็ตเวิร์กอินเตอร์เฟสบน bronze

   a. **พิมพ์**:

      ```
      smit mkinet1sl
      ```

   b. สำหรับ TTY PORT สำหรับ SLIP Network Interface เลือก **tty1**

   c. ระบุ INTERNET ADDRESS ตัวอย่างเช่น 130.130.130.1

   d. ระบุ DESTINATION แอดเดรส (ของ gold) ตัวอย่างเช่น 130.130.130.2

   e. ระบุ BAUD RATE ของโมเด็มของคุณ

   f. ระบุ DIAL STRING ตัวอย่างเช่น :

      - "" AT OK ATDT555-1234 CONNECT ""

- ความหมายของคำสั่งคือ : ใช้ **tty1** ที่ 9600 baud ส่ง AT ไปยังโมเด็ม โมเด็มควรตอบด้วย OK หมุนหมายเลขโทรศัพท์ 555-1234 โมเด็มควรตอบด้วย CONNECT ต้องมีช่องว่างก่อนและหลังอักขระ ""

g. ออกจาก SMIT

8. สร้าง **SLIP** เน็ตเวิร์กอินเตอร์เฟสบน gold ทำโพรซีเดอร์เดียวกับที่คุณใช้กับ bronze (ในขั้นตอนที่ 5) ยกเว้นสลับ INTERNET ADDRESS และ DESTINATION แอดเดรส
9. เพิ่ม 2 entr ต่อไปนี้เข้ากับไฟล์ **/etc/hosts** บนทั้ง bronze และ gold:

```
130.130.130.1   bronze
130.130.130.2   gold
```

ชื่อที่คุณกำหนดต้องเป็นหนึ่งเดียว อีกนัยหนึ่ง ถ้าอินเตอร์เฟส Token-Ring บน bronze ถูกกำหนดชื่อเป็นbronze กำหนด **SLIP** อินเตอร์เฟสเป็นชื่อ เช่น bronze_slip

หมายเหตุ: เพื่อทำให้อินเตอร์เฟสมีความง่ายสำหรับคำสั่ง**slattach** คุณอาจใช้สคริปต์/usr/sbin/slipcall

10. ทดสอบการเชื่อมต่อ **SLIP**
    a. บน bronze **พิมพ์** :

```
ping gold
```

    b. บน gold **พิมพ์** :

```
ping bronze
```

ถ้าการทดสอบทั้งสอลผ่าน การเชื่อมต่อ **SLIP** จะพร้อมใช้งาน ถ้าไม่ กลับไปที่ขั้นตอนที่ 5 และตรวจสอบว่าการตั้งค่าบนทั้ง bronze และ gold ถูกต้อง

การตั้งค่า SLIP ผ่านสายเคเบิลแบบ null โมเด็ม

เมื่อต้องการกำหนดคอนฟิก **SLIP** ระหว่างสองระบบที่เชื่อมต่อกัน โดยใช้สายเคเบิลโมเด็ม null คุณสามารถใช้โพรซีเดอร์นี้ ซึ่งเป็นทางเลือก ระหว่างอินเตอร์เฟส System Management Interface Tool (SMIT) และ บรรทัดรับคำสั่งเพื่อดำเนินการกำหนดคอนฟิกให้สมบูรณ์

เพื่อความชัดเจน วิธีการเหล่านี้จะใช้ชื่อ bronze และ gold สำหรับโฮสต์ 2 ตัว

1. เชื่อมต่อ bronze และ goldโดยใช้สายเคเบิลแบบ null โมเด็ม ต้องการสายเคเบิลต่อไปนี้ (สายเคเบิลจะถูกลิสต์ตามลำดับที่มันจะถูกเชื่อมต่อจาก bronze กับ gold)
   a. Cable B (หมายเลขชิ้นส่วน 00G0943) Serial Port Jumper Cable; ถูกจัดเตรียมให้ 2 ชุดสำหรับแต่ละระบบ ยกเว้นรุ่น 220, 340 และ 350 ที่ไม่ต้องการมัน
   b. Cable D (หมายเลขชิ้นส่วน 6323741 โค้ดคุณลักษณะ 2936) สายเคเบิลอะซิงโครนัส EIA-232/V.24
   c. Cable E (หมายเลขชิ้นส่วน 59F2861 โค้ดคุณลักษณะ) Printer/Terminal Interposer EIA-232 (สาย null โมเด็ม)
   d. Changer Adapter (ทั้งสองด้านของอะแด็ปเตอร์เป็นซ็อกเก็ต)
2. สร้าง tty บน bronze
   a. **พิมพ์:**

```
smit maktty
```

   b. เลือก **rs232** เป็นชนิดของ tty ที่คุณต้องการสร้าง
   c. เลือกซีเรียลพอร์ตที่พร้อมใช้งาน ตัวอย่างเช่น **sa0** (ซีเรียลพอร์ต 1 ของระบบ)

d. เลือกหมายเลขพอร์ตสำหรับ tty นี้จากลิสต์

e. ตั้งอัตรา BAUD เป็น 19200 (คุณจะเปลี่ยนมันเป็น 38400 ภายหลัง แต่สำหรับตอนนี้ ใช้ 19200)

f. ตั้ง Enable LOGIN เป็น disable จากนั้นออกจาก SMIT

3. สร้าง tty บน gold ทำโพรซีเดอร์เดียวกันที่คุณใช้กับ bronze (ในขั้นตอนที่ 2) ยกเว้นตั้ง Enable LOGIN เป็น **enable**

   หมายเหตุ: ส่วนที่เหลือของวิธีการเหล่านี้สันนิษฐานว่าหมายเลขของ tty บนทั้ง bronze และ gold คือ tty1

4. ทดสอบการเชื่อมต่อแบบฟิสิคัลด้วย ATE

   a. บน bronze **พิมพ์ :**

      ```
      ate
      ```

   b. ที่เมนูหลัก Unconnected เลือกคำสั่งย่อย **Alter** ตั้ง Rate เป็น 19200 และ Device เป็น tty1

   c. ที่เมนูหลัก Unconnected เลือกคำสั่งย่อย **Connect** เมื่อได้รับพร้อมต์ของ ATE เพื่อให้ใส่หมายเลขโทรศัพท์ กด Enter คุณควรได้รับข้อความ :

      ```
      ate: 0828-010 The Connect command has made a connection through port tty1
      ```

   d. กด Enter คุณควรได้รับล็อกอินพร้อมต์สำหรับ gold ล็อกอินไปยัง gold

   e. สุดท้าย กลับไปที่หน้า connected ล็อกเอาต์จาก gold กด Ctrl-v (เพื่อให้ได้รับ ATE CONNECTED MAIN MENU) กดคีย์ T เพื่อยกเลิกการเชื่อมต่อ (สิ้นสุด) และกดคีย์ Q เพื่อออกจาก ATE

      หมายเหตุ: ถ้าคุณไม่ได้รับพร้อมต์สำหรับล็อกอิน กลับไปที่ขั้นตอนที่ 1 และตรวจสอบว่าการตั้งค่าของคุณถูกต้อง ห้ามทำต่อจนกว่าคุณจะล็อกอินกับ gold

5. เนื่องจากการตั้งค่า tty สำหรับใช้กับ ATE จะแตกต่างกับการตั้งค่าสำหรับใช้กับ **SLIP** เล็กน้อย คุณต้องทำการเปลี่ยนแปลงต่อไปนี้ :

   a. บน bronze **พิมพ์ :**

      ```
      smit chgtty
      ```

   b. เลือก **tty1** ตั้งอัตรา BAUD เป็น 38400 และออกจาก SMIT

   c. บน gold **พิมพ์ :**

      ```
      pdisable tty1
      ```

   d. บน gold **พิมพ์ :**

      ```
      smit chgtty
      ```

   e. เลือก **tty1** ตั้ง Enable LOGIN เป็น disable ตั้ง อัตรา BAUD เป็น 38400 และจากนั้นออกจาก SMIT

6. เพิ่มบรรทัดต่อไปนี้เข้ากับไฟล์ /usr/lib/uucp/Devices บนทั้ง bronze และ gold:

   ```
   Direct tty1 - 38400 direct
   ```

7. สร้าง **SLIP** เน็ตเวิร์กอินเตอร์เฟสบน **bronze**

   a. **พิมพ์:**

      ```
      smit mkinet1sl
      ```

   b. สำหรับ TTY PORT สำหรับ SLIP Network Interface เลือก **tty1**

   c. ระบุ INTERNET ADDRESS ตัวอย่างเช่น 130.130.130.1

   d. ระบุ DESTINATION แอดเดรส (ของ gold) ตัวอย่างเช่น 130.130.130.2 และจากนั้นเลือก OK หรือ Enter

   e.

8. สร้าง **SLIP** เน็ตเวิร์กอินเตอร์เฟสบน gold ทำโพรซีเดอร์เดียวกับที่คุณใช้กับ bronze (ในขั้นตอนที่ 5) ยกเว้นสลับ INTERNET ADDRESS และ DESTINATION แอดเดรส
9. เพิ่ม 2 entry ต่อไปนี้เข้ากับไฟล์ `/etc/hosts` บนทั้ง bronze และ gold:

   ```
   130.130.130.1   bronze
   130.130.130.2   gold
   ```

   ชื่อที่คุณกำหนดต้องเป็นหนึ่งเดียว อีกนัยหนึ่ง ถ้าอินเตอร์เฟส Token-Ring บน bronze ถูกกำหนดชื่อเป็น`bronze` กำหนด **SLIP** อินเตอร์เฟสเป็นชื่อ เช่น `bronze_slip`
10. สตาร์ท **SLIP** บนทั้ง bronze และ gold พิมพ์:

    ```
    slattach tty1
    ```
11. ทดสอบการเชื่อมต่อ **SLIP**
    a. บน bronze พิมพ์ :

       ```
       ping gold
       ```
    b. บน gold พิมพ์ :

       ```
       ping bronze
       ```

ถ้าการทดสอบทั้งสอลผ่าน การเชื่อมต่อ **SLIP** จะพร้อมใช้งาน ถ้าไม่ กลับไปที่ขั้นตอนที่ 5 และตรวจสอบว่าการตั้งค่าบนทั้ง bronze และ gold ถูกต้อง

การปิดการใช้งานการเชื่อมต่อ SLIP

เพื่อปิดการใช้งานการเชื่อมต่อ **SLIP** ใช้โพรซีเดอร์นี้

1. พิมพ์:

   ```
   ps -ef | grep slatt
   ```

   บันทึกหมายเลขกระบวนการของกระบวนการที่เกี่ยวข้องกับคำสั่ง **slattach**
2. สำหรับแต่ละหมายเลขกระบวนการ พิมพ์ :

   ```
   kill process_number
   ```

   ห้ามใช้แฟล็ก **-9** ของคำสั่ง **kill**

   ถ้า **slattach** ถูก kill โดยไม่ได้ตั้งใจด้วยแฟล็ก **-9** slip lock อาจยังอยู่ใน `/etc/locks` ลบล็อกไฟล์นี้เพื่อลบหลังจาก **slattach**

ในการปิดใช้งานการเชื่อมต่อ **SLIP** ชั่วคราว ทำดังต่อไปนี้บนทั้งระบบโลคัลและระบบรีโมต :

1. พิมพ์:

   ```
   ifconfig sl# down
   ```
2. ลิสต์กระบวนการ **slattach** ที่รันอยู่ในปัจจุบันโดยใช้คำสั่ง:

   ```
   ps -ef | grep slat
   ```

   เอาต์พุตจะคล้ายดังต่อไปนี้:

   ```
   root  1269  1    0   Jun 25  ... slattach
   ```
3. Kill กระบวน **slattach** โดยใช้ ID ของกระบวนการของมัน ตัวอย่างเช่น เพื่อ kill กระบวนการ **slattach** ที่ถูกแสดงในตัวอย่างก่อนหน้านี้ ใส่ :

   ```
   kill 1269
   ```

โดยที่ 1269 คือ ID ของกระบวนการของ **slattach** ต้อง *ไม่*ลบกระบวนการ **slattach** โดยใช้แฟล็ก **-9** ของคำสั่ง **kill**

การเชื่อมต่อ **SLIP** ถูกปิดใช้งานแล้ว

การเปิดใช้งานการเชื่อมต่อ SLIP

ใช้วิธีเหล่านี้เพื่อเปิดใช้งานการเชื่อมต่อ **SLIP** ที่ถูกปิดใช้งานชั่วคราว

รันคำสั่งเหล่านี้บนทั้งระบบโลคัลและรีโมต

1. พิมพ์:

   ```
   ifconfig sl# up
   ```

2. ใช้คำสั่ง **slattach** ที่ใช้เมื่อเริ่มต้นอีกครั้ง

การลบอินเตอร์เฟส SLIP

ใช้วิธีการเหล่านี้เพื่อลบอินเตอร์เฟส **SLIP**

หลังจากวิธีการเหล่านี้ถูกดำเนินการทั้ง sl# อินเตอร์เฟส และกระบวนการ **slattach** ที่เกี่ยวข้องจะถูกลบ entry ใดๆที่ถูกทำกับไฟล์ /etc/hosts จะยังคงอยู่และควบถูกลบแบบแมนวล

1. เพื่อลบอินเตอร์เฟส **SLIP** และกระบวนการ **slattach** ที่เกี่ยวข้อง ใช้`smit rminet` fast path เพื่อเข้าถึงหน้าจอ **Available Network Interfaces**
2. เลือก entry ที่เหมาะสมจากหน้าจอ **Available Network Interfaces** และเลือก **Do**

หมายเหตุ: entry ใดๆที่ถูกทำกับไฟล์ /etc/hosts จะยังคงอยู่และควบถูกลบแบบแมนวล

การแก้ปัญหา SLIP

คำสั่งเหล่านี้ถูกต้องการเพื่อดีบักปัญหาเกี่ยวกับ **SLIP**

แต่ละคำสั่งที่มาพร้อมกับตัวอย่างของวิธีที่คำสั่งถูกใช้เพื่อแก้ไขปัณหา **SLIP**

นอกจากนี้ ลิสต์ของปัญหาทั่วไปและข้อความแสดงข้อผิดพลาดถูกจัดเตรียมสำหรับให้คุณอ้างอิง

คำสั่ง netstat:

คำสั่ง **netstat** ทำงานร่วมกับคำสั่ง **ifconfig** เพื่อให้เงื่อนไขของสถานะของ TCP/IP เน็ตเวิร์กอินเตอร์เฟส

ตัวอย่างเช่น คำสั่ง **netstat -in** ใช้แฟล็ก **-i** เพื่อแสดงข้อมูลบนเน็ตเวิร์กอินเตอร์เฟส ขณะที่แฟล็ก **-n** จะพิมพ์ IP แอดเดรสแทนที่จะเป็นชื่อโฮสต์ ใช้คำสั่งนี้เพื่อตรวจสอบ SLIP อินเตอร์เฟส แอดเดรส และชื่อโฮสต์ ส่วนต่อไปนี้จะอธิบายเอาต์พุตของ **netstat -in**

โปรแกรมโมเด็มโดยใช้การตั้งค่าที่แสดงในส่วน “ข้อพิจารณาเกี่ยวกับ SLIP โมเด็ม” ในหน้า 664 ตัวอย่างต่อไปนี้แสดงวิธีโปรแกรมและบันทึกการตั้งค่าพื้นฐานสำหรับโมเด็ม Hayes-compatible ป้อน:

```
Name  Mtu   Network     Address            Ipkts     Ierrs Opkts Oerrs  Col
lo0   1536  <Link>                         2462       0    2462   0      0
lo0   1536  127         localhost.austi    2462       0    2462   0      0
tr0   1492  <Link>                         1914560    0   21000   0      0
```

```
tr0   1492  129.35.16  glad.austin.ibm     1914560    0   21000   0     0
sl0   552   1.1.1.0    1.1.1.1             48035      0   54963   0     0
sl1*  552   140.252.1  140.252.1.5         48035      0   54963   0     0
```

สังเกต * ข้างๆอินเตอร์เฟส sl1 นี่จะแสดงว่าเน็ตเวิร์กอินเตอร์เฟสไม่ทำงานหรือไม่พร้อมใช้งาน ผู้ใช้สามารถแก้ไขได้โดยการใช้คำสั่ง **ifconfig sl1 up** ถ้ามันเป็นอินเตอร์เฟส **SLIP** ที่ถูกต้อง

netstat ให้สถิติที่เกี่ยวกับจำนวนแพ็กเก็ตของอินพุตและเอาต์พุต พร้อมด้วยข้อผิดพลาดอินพุตและเอาต์พุต ที่มีประโยชน์เมื่อแก้ไขปัญหาการเชื่อมต่อ **SLIP**

ตัวอย่างเช่น ผู้ใช้ใส่ **ping** กับระบบรีโมตข้าม **SLIP** ลิงก์ และคำสั่ง **ping** ดูเหมือนว่าจะหยุดทำงาน ผู้ใช้จะรันคำสั่ง **netstat -in** ทันทีจากเชลล์คำสั่งอื่น และสังเกตว่า Opkts เพิ่มขึ้น แต่ไม่มี Ipkts จากรีโมตโฮสต์ นี่จะบอกว่าระบบรีโมตไม่ได้ส่งข้อมูลกลับ (หรือไม่ได้รับ) ผู้ใช้ต้องใช้คำสั่ง **netstat** เดิมบนระบบรีโมตเพื่อตรวจสอบการได้รับแพ็กเก็ต **ping** หรือจำนวนข้อผิดพลาดเพิ่มขึ้น

การแปลงชื่อโฮสต์ กับ หมายเลขอินเตอร์เน็ตมีความสัมพันธกับการแปลงชื่อ และดังนั้นจะมีความสำคัญกับการทำงานที่ถูกต้องของสาย **SLIP** เพื่อดีบักชื่อโฮสต์ alias และปัญหาของการหาเส้นทาง ใช้คำสั่ง **netstat -rn** ชื่อพื้นฐานของโฮสต์หรือชื่อโฮสต์จะเป็นชื่อเดียวที่ควรส่งกลับมาจากไฟล์ **/etc/hosts** ถ้าเครื่องได้รับเซอร์วิสโดยเนมเซิร์ฟเวอร์ (ซึ่งคือมี `/etc/resolv.conf`) ดังนั้น name-server จะให้ชื่อโดเมนที่ถูกต้องแบบเต็มในคำสั่งนี้

คำสั่ง ifconfig:

คำสั่ง **ifconfig** เป็นเครื่องมือการตั้งค่าเน็ตเวิร์กอินเตอร์เฟส ที่ทำให้เน็ตเวิร์กอินเตอร์เฟส STRUCTURE ถูกสร้างหรือตรวจพบแบบไดนามิกจากหน่วยความจำเคอร์เนล

คำสั่งนี้ยอมรับข้อมูลจากบรรทัดรับคำสั่ง จากนั้นสร้างโครงสร้างของหน่วยความจำที่สอดคล้องกับพารามิเตอร์ สำหรับจุดประสงค์ในการดีบัก คำสั่ง **ifconfig** ถูกใช้เพื่อตรวจสอบสถานะของอินเตอร์เฟสการสื่อสาร

หมายเหตุ: การเปลี่ยนแปลงใดๆ ที่ทำกับแอ็ตทริบิวต์ของอินเตอร์เฟส โดยใช้คำสั่ง **ifconfig** จะสูญหายเมื่อ ระบบรีบูต

ตัวอย่างเช่น เพื่อตรวจสอบสถานะปัจจุบันของอินเตอร์เฟส sl1 :

1. ใส่คำสั่ง **netstat -i** และตรวจสอบเอาต์พุตที่เลือกอินเตอร์เฟส sl# ที่เหมาะสม ตัวอย่างเช่น sl0, sl1, sl2 และอื่นๆ
2. ใส่คำสั่ง **ifconfig sl#** และตรวจสอบเอาต์พุต ifconfig สำหรับฟิลด์ของคีย์ต่อไปนี้ :

ไอเท็ม	คำอธิบาย
POINTTOPOINT flag	แฟล็กนี้ควรมีเสมอบนการทำงาน SLIP ลิงก์ ถ้าไม่ ควรอยู่ในสถานะ down หรือ disconnected ลองใช้คำสั่ง **ifconfig sl# up** และ **ifconfig sl#** อีกครั้งเพื่อดูว่าสถานะของมันเปลี่ยนหรือไม่
แฟล็ก UP	ระบุว่าเน็ตเวิร์ก sl# อินเตอร์เฟสถูกเปิดใช้งานและควรทำงานอยู่
แฟล็ก RUNNING	ระบุว่าคำสั่ง **slattach** ทำสำเร็จ ในความเป็นจริง ลิงก์จะถูกเข้าถึง การหมุนโทรศัพท์ทำสำเร็จ อีกด้านตอบรับ และด้านรีโมตส่งค่าสถานะ CARRIER DETECT เมื่อสถานะ CD เกิดขึ้นแฟล็กจะถูกอัพเดตด้วย running บิต

คำสั่ง pdisable และ lsdev:

พอร์ต tty ใดๆที่ถูกใช้สำหรับการเชื่อมต่อ **SLIP** ต้องถูกปิดใช้งาน หรือมีสถานะไม่พร้อมใช้งาน

เพื่อตรวจสอบว่าพอร์ตสำหรับ tty1 ถูกปิดใช้งาน ใช้สิทธิของผู้ใช้ root และใส่หนึ่งในคำสั่งต่อไปนี้ :

- `lsattr -El tty1 -a login`

คำสั่งนี้จะแสดงสถานะถาวรของพอร์ต tty ที่ถูกบันทึกใน Object Database Manager (ODM) ของระบบ ถ้าเอาต์พุตเป็นอย่างอื่นที่ไม่ใช่ `login disable` ใช้ SMIT เพื่อเปลี่ยนฟิลด์ enable LOGIN เป็น **disable**

- `pdisable | grep tty1`
  คำสั่งนี้ เมื่อถูกใช้โดยไม่มีพารามิเตอร์ จะแสดงพอร์ต tty ทั้งหมดที่อยู่ในสถานะถูกปิดใช้งาน ในตัวอย่างนี้ **pdisable** ถูกไพพ์ไปยังคำสั่ง **grep** เพื่อกำจัดเอาต์พุตที่ไม่จำเป็น ถ้า tty1 ไม่ถูกแสดงหลังจากรันคำสั่งนี้ พอร์ตยังไม่ถูกปิดใช้งาน

คำสั่ง ps:

คำสั่ง **ps** จะแสดงข้อมูลเกี่ยวกับกระบวนการที่แอ็คทีฟไปยังเอาต์พุตมาตรฐาน

ใช้คำสั่งนี้เพื่อตรวจสอบการมีอยู่ (หรือไม่มีอยู่) ของกระบวนการ **slattach**ที่ใช้เพื่อกำหนดสาย tty กับเน็ตเวิร์กอินเตอร์เฟส

ถ้า **netstat -in** แสดงว่าอินเตอร์เฟสไม่ทำงาน ผู้ใช้ควรรันคำสั่ง **ps -ef | grep slat** เพื่อดูว่า กระบวนการ **slattach** ยังรันอยู่บนพอร์ต tty ที่เกี่ยวข้องหรือไม่ โปรดสังเกตว่า สำหรับการเชื่อมอินเตอร์เฟส **SLIP** ที่เชื่อมต่อโดยตรง การเชื่อมต่อที่ถูกตัดจะถูกทำใหม่โดยอัตโนมัติโดยไม่ต้องทำแบบแมนวล สำหรับอินเตอร์เฟส **SLIP** ที่ถูกเชื่อมต่อโดยโมเด็ม การเชื่อมต่อที่เสียหายต้องถูกหมุนใหม่แบบแมนวล ถ้าผู้ใช้ใส่สตริงการหมุนโทรศัพท์ในบรรทัดรับคำสั่ง **slattach** ผู้ใช้ต้องใส่คำสั่งและสตริงการหมุนโทรศัพท์ใหม่เพื่อเรียกคืนการเชื่อมต่อ

คำสั่ง ping และไฟของโมเด็ม:

คำสั่ง **ping** และไฟของโมเด็มถูกใช้เพื่อดีบักปัญหาของการสื่อสารแบบ **SLIP**

ping เป็นแพ็กเก็ตข้องขอ echo โดยส่งออกจากเครื่องและแพ็กเก็ต echo ที่ตอบสนองจะถูกส่งกลับมา ลำดับของเหตุการณ์เหล่านี้จะมีประโยชน์ถ้าผู้ดูแลระบบสามารถเห็นไฟของโมเด็ม

ตัวอย่างเช่น ระบบโลคัลสร้างแพ็กเก็ตคำร้องขอ echo และส่งไปยังระบบรีโมต ไฟ Send Data (SD) บนโมเด็มของโลคัลจะสว่าง นี่หมายความว่าโลคัล TCP/IP **slattach** และ tty สามารถรวบรวมข้อมูลและส่งมันออกไปจากโมเด็มไปยังระบบรีโมต

ระบบรีโมตจะรับแพ็กเก็ตและไฟ receive data จะกระพริบแต่ไฟล์ SD ของมนไม่ติด นี่หมายความว่าระบบรีโมตไม่สามารถส่ง (หรือส่งคืน) คำร้องขอ ping ของระบบโลคัล ดังนั้น ผู้ใช้บนระบบโลคัลอาจจะเห็นว่าคำสั่ง **ping** หยุดทำงาน และต้องกด Ctrl-C เพื่อออกจากการเชื่อมต่อ

สาเหตุทั่วไปของปัญหานี้คือการใช้ XON/XOFF โฟลว์คอนโทรล ในโมเด็มตัวหนึ่งหรือทั้งสองตัว อย่างไรก็ตาม ผู้ใช้ไม่ควรมองข้ามความเป็นไปได้ของการหาเส้นทางหรือแอดเดรสบนระบบชนกัน

ปัญหาทั่วไปของ SLIP และข้อความแสดงข้อผิดพลาด:

ปัญหาทั่วไปของ **SLIP** และข้อความแสดงข้อผิดพลาด สาเหตุที่เป็นไปได้ และแอ็คชันของผู้ใช้ที่แนะนำสามารถอ้างอิงได้ที่นี่

ข้อความ: 0821-296 Cannot set line discipline for /dev/tty# to slip.ioctl(TXSETLD) การเรียกใช้ของระบบได้รับพารามิเตอร์ที่ไม่ถูกต้อง

สาเหตุที่เป็นไปได้ : ข้อผิดพลาดชนิดนี้โดยมากเกิดขึ้นเมื่อสตาร์ทกระบวนการ **slattach** และเนื่องมาจากการตั้งค่า SLIP ที่ไม่ถูกต้อง ปัญหาอาจมีสาเหตุมาจากความไม่สอดคล้องกันระหว่างหมายเลขของอุปกรณ์ tty และหมายเลขอินเตอร์เฟส sl นี่ยังอธิบายได้ว่าทำไม ifconfig ไม่ถูกรันก่อน **slattach**

ปัญหานี้ยังอาจเกิดเมื่อกระบวนการ **slattach** ถูกดร็อบหรือถูก kill ไม่ถูกต้อง หรือเมื่อผู้ใช้พยายามย้ายการเชื่อมต่อ **SLIP** ไปยังพอร์ต tty อื่นและลืมตั้งค่าอินเตอร์เฟส sl# ให้ตรงกับ tty ตรวจสอบกระบวนการ **slattach** ที่รัน ที่ยังคงรันอยู่ (ตัวอย่างเช่น `ps -ef | grep slat`)

แอ็คชัน : อุปกรณ์ tty สำหรับ SLIP คือ **/dev/tty24** และผู้ใช้ได้สร้างอินเตอร์เฟส sl0 นี่ไม่ถูกต้อง ผู้ใช้ควรสร้างอินเตอร์เฟส sl24 ซึ่งตรงกับหมายเลข tty (tty24 และ sl24) ถ้าปัญหายังมีอยู่ ผู้ใช้ควรปิดอินเตอร์เฟส sl (ดูที่ ″การปิดอินเตอร์เฟส SLIP″) และตั้งค่าการเชื่อมต่อใหม่โดยใช้คำสั่งต่อไปนี้ :

```
lsdev  -Cc if  -s SL
lsattr -El sl0
```

ข้อความ:

เน็ตเวิร์กไม่พร้อมใช้งานในปัจจุบัน

เส้นทางไปยังรีโมตโฮสต์ไม่พร้อมใช้งาน

สาเหตุที่เป็นไปได้: ข้อผิดพลาดเหล่านี้เกิดขึ้นบ่อยเมื่อผู้ใช้พยายาม ping โฮสต์ผ่าน **SLIP** ลิงก์และลิงก์ไม่ถูกสร้างอย่างเหมาะสม ปัญหาส่วนใหญ่คือพอร์ต tty หนึ่งพอร์ตหรือทั้งสองพอร์ตที่เกี่ยวข้องกับอินเตอร์เฟส sl# อยู่ในสถานะถูกปิดใช้งาน มันยังเป็นไปได้ที่มีแอดเดรสหรือเส้นทางที่ขัดกันระหว่างระบบโฮสต์

แอ็คชัน:

- ลบอินเตอร์เฟส sl# โดยใช้ `smit rminet` fast path นี้ต้องถูกทำบนทั้งโลคัลและรีโมตโฮสต์ของ SLIP
- ทำต่อไปนี้สำหรับแต่ละโฮสต์ของ SLIP:
  1. ใส่ `pdisable | grep tty#`
  2. ถ้าอุปกรณ์ tty ไม่ถูกลิสต์ในเอาต์พุตของคำสั่งก่อนหน้านี้ tty ไม่ถูกปิดใช้งาน ปิดใช้งาน tty ผ่าน SMIT หรือบรรทัดรับคำสั่ง โดยที่พอร์ต tty ถูกปิดใช้งาน ใช้ SMIT เพื่อสร้างอินเตอร์เฟส **SLIP** บนทั้งสองระบบใหม่ ถ้าปัญหายังมีอยู่ ตรวจสอบเน็ตเวิร์กแอดเดรสและเส้นทาง ใช้คำสั่ง **netstat –ir** เพื่อดูแอดเดรส การหาเส้นทาง และข้อมูลอินเตอร์เฟสอย่างรวดเร็ว

ปัญหา : เมื่อรีโมตไซต์หมุนโทรศัพท์มาที่โลคัลโฮสต์ โมเด็มบนโลคัลโฮสต์จะเชื่อมต่อแต่ไม่สามารถทำกระบวนการล็อกอิน

สาเหตุที่เป็นไปได้ : ถ้าโมเด็มทั้งสองเชื่อมต่อกันและเริ่มการแฮนด์เช็ก หรือแลกเปลี่ยนข้อมูลการเชื่อมต่อ แต่จากนั้นตัดการเชื่อมต่อ ปัญหาอาจเกิดจากโค้ดผลลัพธ์ของโมเด็ม ปัญหานี้ยังสามารถมีสาเหตุจากสตริงการหมุนโทรศัพท์ของ **slattach** ที่ไม่ถูกต้อง ถ้าโมเด็มทั้งสองดังแต่ไม่เริ่มกระบวนการแฮนด์เช็ก ปัญหาอาจเกิดจากโมเด็มไม่ถูกตั้งเป็น auto–answer

แอ็คชัน:

1. ทดสอบการเชื่อมต่อของโมเด็มก่อนด้วยคำสั่ง **cu** โมเด็มบนรีโมตโฮสต์ควรยอมให้ผู้ใช้ล็อกอินกับระบบ ไม่ควรมีขยะบนหน้าจอระหว่างการล็อกอิน ถ้าเป็นเช่นนั้น อาจบอกได้ว่ามีสัญญาณรบกวนในสายโทรศัพท์ซึ่งอาจเป็นส่วนหนึ่งของปัญหาระหว่างการล็อกอิน การแจ้งเกี่ยวกับหลายล็อกอิน *ไม่* ควรเลื่อนข้ามหน้าจอ ถ้าเกิดขึ้น นี่อาจบอกได้ว่าเป็นปัญหาเกี่ยวกับสายโทรศัพท์ หรือการตั้งค่าโมเด็มที่ไม่ถูกต้อง
2. ตรวจสอบการตั้งค่าโมเด็ม และพยายามปิดโค้ด ARQ ถ้ามันเปิดอยู่ ในโมเด็มแบบ Hayes–compatible นี่คือการตั้งค่า &A0 การปิดใช้งานโค้ดผลลัพธ์ของ ARQ จะไม่มีผลกับการเชื่อมต่อที่มีการควบคุมข้อผิดพลาด หรือทำให้โมเด็มไม่ส่งข้อความมาตรฐาน CONNECT (ถ้าโค้ดผลลัพธ์ถูกเปิดใช้งาน) ที่ต้องการสำหรับสตริงการหมุน **slattach**

ปัญหา: ผู้ใช้ไม่สามารถ **ping** ข้ามการเชื่อมต่อโมเด็มแบบ **SLIP** คำสั่ง **ping** อาจหยุดทำงาน หรือให้ข้อความแสดงข้อผิดพลาด

สาเหตุที่เป็นไปได้:

1. โมเด็มและ/หรือพอร์ต tty อาจถูกตั้งค่าเพื่อใช้ XON/XOFF โฟลว์คอนโทรล
2. กระบวนการ **slattach** อาจถูกยกเลิกบนรีโมตโฮสต์หรือการเชื่อมต่อของโมเด็มถูกดร็อบ
3. แอดเดรสที่กำหนดให้กับโฮสต์ **SLIP** อาจไม่ถูกต้อง

แอ็คชัน:

1. ตรวจสอบการตั้งค่าของโมเด็มของทั้งโลคัลและรีโมต มันควรถูกตั้งเพื่อใช้ RTS/CTS (ฮาร์ดแวร์) โฟลว์คอนโทรล หรือไม่ใช้โฟลว์คอนโทรลเลย ผู้ใช้ควรพยายาม ping จากแต่ละระบบ Ping systemA ไปยัง systemB
2. ตรวจสอบว่ากระบวนการ **slattach** ยังคงรันอยู่บนทั้งระบบโลคัลและระบบรีโมต ใช้คำสั่ง: `ps -ef |grep slat`. ตรวจสอบว่า sl# อินเตอร์เฟสอยู่ในสถานะ running ใช้คำสั่ง : `ifconfig sl#`
3. ตรวจสอบว่าไม่มีการขัดกันระหว่าง **SLIP** แอดเดรส และที่เกี่ยวข้องกับเน็ตเวิร์กอินเตอร์เฟสอื่น (ถ้ามี) ใช้คำสั่ง : **netstat -ir** ถ้าไม่แน่ใจเกี่ยวกับแอดเดรสและคลาสของแอดเดรส ตั้งค่า **SLIP** โดยใช้ scheme ของแอดเดรสแบบง่าย เป็น 1.1.1.1 สำหรับโลคัลโฮสต์และ 1.1.1.2 สำหรับรีโมตโฮสต์

แบบสอบถามเกี่ยวกับ SLIP

ใช้แบบสอบถามนี้เพื่อเก็บข้อมูลเกี่ยวกับการตั้งค่า **SLIP**

ข้อมูลที่ถูกเก็บรวบรวมบนชีตเหล่านี้สามารถถูกแฟ็กซ์ไปยังตัวแทนผู้บริการเมื่อต้องการความช่วยเหลือเพิ่มเติมกับการตั้งค่า **SLIP**

1. คอนฟิกูเรชัน **SLIP** นี้เคยทำงานได้ก่อนหน้านี้หรือไม่? (Y/N) ____
2. ชนิดของเครื่องคืออะไร ? (ตัวอย่างเช่น : UNIX/PC, DOS/PC เป็นต้น)
   ระบบโลคัล : _____ ระบบรีโมต: _____
   ถ้า โฮสต์ไม่ใช่ระบบ IBM UNIX กรุณาระบุชนิดของซอฟต์แวร์ที่ถูกใช้เพื่อสร้างการเชื่อมต่อ **SLIP**

   _____
3. เวอร์ชันของระบบปฏิบัติการ IBM UNIX ที่อยู่บนแต่ระบบคืออะไร ? ใช้คำสั่ง `/bin/oslevel` ถ้าคำสั่งนี้ใช้ไม่ได้ ใช้วิธีต่อไปนี้ :

   ```
   lslpp -h bos.rte
   ```

   หาบรรทัดของระดับของรีลีส *active commit*
   ระบบโลคัล : _____ ระบบรีโมต: _____
4. ลิสต์อินเตอร์เฟสทั้งหมดที่มีอยู่บนทั้งสองระบบ (ตัวอย่างเช่น sl0, sl1) ทำได้โดย ใช้คำสั่ง : `lsdev -Cc if`
   ระบบโลคัล: _____ ระบบรีโมต: _____

   _____ _____

   _____ _____

   _____ _____

   หมายเลขอินเตอร์เฟสของ **SLIP** ควรตรงกับหมายเลขของอุปกรณ์ tty ตัวอย่างเช่น `/dev/tty53` ควรถูกใช้กับ sl53

5. **SLIP** ถูกตั้งค่าผ่าน SMIT หรือด้วยคำสั่งหรือไม่ ? การตั้งค่า SLIP โดยใช้คำสั่งจะไม่ถาวรและจะไม่อยู่หลังจากระบบถูกรีบูต

   _____

6. **SLIP** ถูกตั้งค่าผ่านโมเด็ม หรือสายซีเรียลโดยตรงหรือไม่?

   _____

7. ถ้าโมเด็มถูกใช้ ลิสต์ผู้ผลิตและชนิดของโมเด็มสำหรับทั้งระบบโลคัลและรีโมต

   ```
           TYPE       BAUD RATE  IBM CABLING if not IBM cable,
                                 (Yes/No)    what type?
   local:  _________  _________    ___       _____________________
   Remote: _________  _________    ___       _____________________
   ```

8. ถ้าโมเด็มถูกใช้ ชนิดของ carrier ของโทรศัพท์คืออะไร ? (leased-line หรือสวิตช์แบบธรรมดา)

   _____

9. ฮาร์ดแวร์ที่สาย **SLIP** ใช้อยู่คืออะไร ?

   128-พอร์ต อะแด็ปเตอร์ (กับ 16-พอร์ต RAN(s)): ___

   2-พอร์ต อะแด็ปเตอร์: ___

   8-พอร์ต อะแด็ปเตอร์: ___

   Native, ซีเรียลพอร์ต S1 หรือ S2 แบบดังเดิม: ___

10. สามารถ ping จากระบบโลคัลไปยังระบบรีโมตหรือไม่ ?

    ```
    (Y/N) _____    (on the local system enter: ping <remote address> )
    ```

11. สามารถ ping จากระบบรีโมตมายังระบบโลคัลหรือไม่ ?

    ```
    (Y/N) _____    (on the remote system enter: ping <local address> )
    ```

12. พอร์ต tty ถูกปิดใช้งานบนทั้งระบบโลคัลและระบบรีโมตหรือไม่ ?

    ```
    (Y/N) _____
    ```

    ใช้คำสั่ง : `pdisable | grep tty#` เฉพาะหมายเลขของ tty ที่ถูกปิดใช้งานจะถูกแสดงเป็นเอาต์พุตของคำสั่งนี้

13. มีข้อความแสดงข้อผิดพลาดถูกแสดงหรือไม่? ถ้ามีกรุณาลิสต์มันด้านล่าง :

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

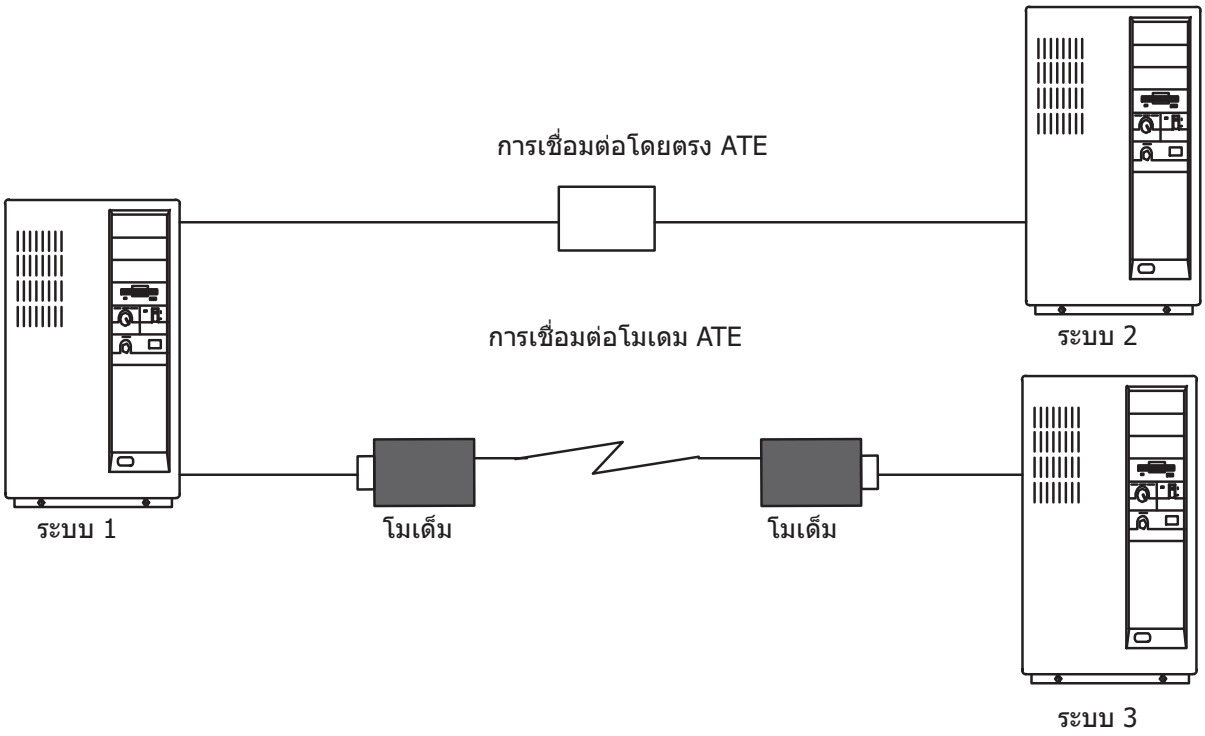
    _____

Asynchronous Terminal Emulation

โปรแกรม Asynchronous Terminal Emulation (ATE) ทำให้เทอร์มินัลบนระบบปฏิบัติการเพื่ออีมูเลตเทอร์มินัล ดังนั้นจะยอมให้ผู้ใช้เชื่อมต่อกับระบบอื่นส่วนใหญ่ที่สนับสนุนเทอร์มินัลแบบอะซิงโครนัส

ATE ทำสิ่งนี้โดยการทำให้ระบบรีโมตเห็นเทอร์มินัลเป็นหน้าจอของระบบ หรือเป็นเทอร์มินัล DEC VT100 อ็อพชัน VT100 ยอมให้ผู้ใช้ล็อกอินกับระบบที่ไม่สนับสนุนเทอร์มินัลของมัน แต่สนับสนุนเทอร์มินัล VT100

ATE ใช้ทั้งการเชื่อมต่อโดยตรง (สายเคเบิล) และการเชื่อมต่อโดยใช้โมเด็มเพื่อสื่อสารระหว่างระบบของผู้ใช้และระบบรีโมต ดังแสดงต่อไปนี้



รูปที่ 43. ชนิดของการเชื่อมต่อ ATE

ขึ้นอยู่กับชนิดของการเชื่อมต่อที่ใช้ ผู้ใช้สามารถตั้งค่า ATE เพื่อเชื่อมต่อกับระบบในห้องถัดไป หรือระบบข้ามประเทศ สำหรับการเชื่อมต่อโดยตรง ผู้ใช้ต้องรู้พอร์ตที่จะใช้บนระบบ สำหรับการเชื่อมต่อโดยใช้โมเด็ม ผู้ใช้ต้องรู้พอร์ตที่ใช้บนระบบและหมายเลขโทรศัพท์ของระบบรีโมต ผู้ใช้ยังต้องรู้ล็อกอิน ID และรหัสผ่านบนระบบรีโมต

ATE ให้ผู้ใช้สามารถรันคำสั่งบนระบบรีโมต ส่งและรับไฟล์ และใช้โปรโตคอล **xmodem** เพื่อตรวจสอบความถูกต้องของข้อมูลในไฟล์ที่ถูกถ่ายโอนระหว่างระบบ ผู้ใช้ยังสามารถดักจับไฟล์ข้อมูลขาเข้าจากระบบรีโมต

> **หมายเหตุ:** ผู้ใช้ต้องเป็น สมาชิกของกลุ่ม UNIX-to-UNIX Copy Program (UUCP) เพื่อที่จะใช้ ATE ผู้ใช้ที่มีสิทธิ root ใช้ System Management Interface Tool (SMIT) เพื่อ ติดตั้งผู้ใช้แต่ละคนในกลุ่ม

การตั้งค่า ATE

ก่อนที่จะรัน ATE ผู้ดูแลระบบต้องติดตั้งซอฟต์แวร์ที่เหมาะสม (ถ้าต้องการ) และตั้งค่าพอร์ต tty และการเชื่อมต่อ

- ATE เป็นผลิตภัณฑ์ที่เป็นอ็อพชันโปรแกรม ไฟล์ทั้งหมดที่จำเป็นสำหรับการทำงานของ ATE จะอยู่ในผลิตภัณฑ์โปรแกรม **bos.net.ate** ที่มีอยู่บนสื่อบันทึกการติดตั้ง ใช้คำสั่งต่อไปนี้เพื่อตรวจสอบว่า ATE พร้อมใช้งานบนระบบของคุณ:

  ```
  lslpp -h | more   <return>
  /bos.net.ate      <return>
  ```

  ATE ไม่พร้อมใช้งานบนระบบของคุณ ติดตั้งอิมเมจ **bos.net.ate** จากสื่อบันทึกการติดตั้ง (เทป ดิสก์เก็ต หรือเน็ตเวิร์กเซิร์ฟเวอร์)
- ATE ถูกติดตั้งบนระบบ ลิสต์ของไฟล์ที่เกี่ยวข้องกับโปรแกรมนี้จะถูกแสดงโดยใช้คำสั่งต่อไปนี้ :

  ```
  lslpp -f | more   <return>
  /bos.net.ate      <return>
  ```
- ผู้ใช้ต้องมีสิทธิของ root เพื่อตั้งค่าพอร์ตสำหรับอุปกรณ์การสื่อสาร

ATE ใช้ทั้งการเชื่อมต่อโดยตรง (สายเคเบิล) และการเชื่อมต่อโดยใช้โมเด็ม การเชื่อมต่อ RS-232C แบบโลคัลยอมให้ใช้ความยาวสูงสุด ที่ 15 เมตร (50 ฟุต) ระหว่างเครื่อง และการเชื่อมต่อแบบ RS-422A ยอมให้ใช้ ความยาวถึง 1200 เมตร (4000 ฟุต) ระหว่างเครื่อง

ก่อนที่จะใช้ ATE เพื่อเรียกไปยังระบบแบบรีโมต ตรวจสอบว่าอุปกรณ์ tty ของระบบแบบรีโมตพร้อมที่จะรับการเรียก

เพื่อเตรียม ATE เพื่อรันบนระบบ ทำขั้นตอนต่อไปนี้:

1. ติดตั้งการ์ดอะซิงโครนัสอะแด็ปเตอร์ ในสล็อตที่เหมาะสมในหน่วยของระบบ ยกเว้นระบบมีซีเรียลพอร์ต ที่ติดตั้งมาอยู่แล้ว
2. เสียบสายเคเบิล RS-232C หรือ RS-422A เข้ากับอะแด็ปเตอร์การ์ด หรือซีเรียลพอร์ตที่มีมาด้วย
3. เพิ่มอุปกรณ์ tty สำหรับพอร์ตการสื่อสาร โดยใช้ `smit mkdev` fast path
4. เลือกชนิดของเทอร์มินัลที่จะอีมูเลตกับ ATE และทำการปรับแต่งที่จำเป็นกับสภาวะแวดล้อม การเปลี่ยนแปลงทั่วไปคือความเร็วของสาย การตั้งค่าพาริตี จำนวนบิตต่อตัวอักขระ และสายจะถูกขับเป็นแบบรีโมตหรือโลคัล ใช้ bpc 8 หรือไม่มีพาริตี ถ้าต้องการใช้ National Language Support (NLS)
5. ตั้งค่าพอร์ตสำหรับอุปกรณ์ เพื่อตั้งค่าพอร์ตที่จะเรียกออกด้วย ATE ใช้คำสั่ง **pdisable** ตัวอย่างเช่น เพื่อตั้งค่าพอร์ต tty1 ใส่:

   ```
   pdisable tty1
   ```

   เพื่อตั้งค่าพอร์ตเพื่อที่ผู้อื่นสามารถเรียกเข้า ใช้คำสั่ง **penable** ตัวอย่างเช่น เพื่อให้ระบบอื่นเรียกเข้ามาที่พอร์ต tty2 ใส่:

   ```
   penable tty2
   ```
6. ต้องแน่ใจว่าอุปกรณ์ถูกระบุ กับระบบรีโมตก่อนหน้านี้แล้ว หลังจากที่อุปกรณ์ถูกระบุ โปรแกรม ATE ต้องถูกปรับแต่งเพื่อแสดงค่าติดตั้งอุปกรณ์ บนระบบรีโมต ปรับแต่งค่าดีฟอลต์โดยการปรับเปลี่ยนคำสั่งย่อย หรือโดยการแก้ไขไฟล์ `ate.def` ดีฟอลต์ เพื่อเปลี่ยนค่าติดตั้งดีฟอลต์สำหรับการเชื่อมต่อแบบโทรศัพท์ ใช้ entry ไฟล์ของไดเร็กทอรีการหมุนโทรศัพท์

เมนูหลักของ ATE

ATE แสดงเมนูตามคำสั่งย่อยที่ใช้

การสตาร์ท ATE ด้วยคำสั่ง **ate** จะแสดงเมนูหลักของ Unconnected ซึ่งให้คุณ :

- เปลี่ยนคุณลักษณะของ ATE (**modify, alter**) ชั่วคราว

- เชื่อมต่อกับระบบอื่น (**directory, connect**)
- ได้รับการช่วยเหลือ (**help**)
- ใช้คำสั่งของระบบปฏิบัติการของเวิร์กสเตชันบนระบบ (**perform**)
- ออกจาก ATE (**quit**)

ขึ้นอยู่กับคำสั่งย่อยที่ใช้จากเมนูหลักของ Unconnected ATE จะแสดงเมนูย่อยต่างกัน:

ตารางที่ 109. เมนูย่อยของ ATE

เมื่อคุณใช้	**ATE จะแสดง**
คำสั่งย่อย **modify**	เมนู Modify (สำหรับข้อมูล ดูที่คำสั่ง **ate** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1*)
คำสั่งย่อย **alter**	เมนู Alter (สำหรับข้อมูล ดูที่คำสั่ง **ate** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1*)
คำสั่งย่อย **connect** หรือ **directory** เพื่อเชื่อมต่อกับระบบแบบรีโมต	เมนูหลักของ Connected
คำสั่งย่อย **directory**	ไดเร็กทอรีการหมุนโทรศัพท์ (ลิสต์ของหมายเลขโทรศัพท์)

จากเมนูหลักของ Connected คุณสามารถใช้คำสั่งย่อยเพื่อ :

- ส่งไฟล์และรับไฟล์จากระบบแบบรีโมต (**send, receive**)
- ส่งสัญญาณ break ไปยังระบบแบบรีโมต (**break**)
- สิ้นสุดการเชื่อมต่อกับระบบแบบรีโมต (**terminate**)

นอกจากนี้ คำสั่งย่อย **modify, alter, help, perform,** และ **quit** จะทำหน้าที่เดียวกับที่มีในเมนูหลักของ Unconnected

คุณสามารถควบคุมแอ็คชันของ ATE ด้วยลำดับของคีย์ควบคุม ลำดับของคีย์เหล่านี้ถูกรู้จักว่าเป็น CAPTURE_KEY, MAINMENU_KEY และ PREVIOUS_KEY ลำดับของคีย์ถูกอธิบายใน “ลำดับของคีย์ควบคุมของ ATE” ในหน้า 680 ATE ถูกติดตั้งพร้อมกับการรวมกันของคีย์แบบดีฟอลต์สำหรับคีย์เหล่านี้ แต่คุณสามารถเปลี่ยนการรวมกันโดยการแก้ไขไฟล์ ATE ดีฟอลต์ `ate.def`

เมนูหลักของ ATE Unconnected:

ใช้คำสั่ง**ate** เพื่อแสดงเมนูหลักของ ATE Unconnected

หลักจากทำการเชื่อมต่อแล้ว ใช้คำสั่งย่อย ATE **connect** เพื่อแสดงเมนูหลักของ Unconnected

คุณสามารถใช้คำสั่งย่อยต่อไปนี้จากเมนูหลักของ ATE Unconnected ในการใช้คำสั่งย่อย พิมพ์ตัวอักษรแรกของคำสั่งย่อยที่จุดรับคำสั่งบนเมนู ตัวอย่างเช่น พิมพ์ **d** เพื่อใช้คำสั่งย่อย **directory**

ไอเท็ม	คำอธิบาย
alter	เปลี่ยนคุณลักษณะของการส่งข้อมูลเป็นการชั่วคราว เช่นความเร็วในการส่ง
connect	ทำการเชื่อมต่อ
directory	แสดงไดเร็กทอรีการหมุนโทรศัพท์
help	แสดงข้อมูลวิธีใช้
modify	แก้ไขค่าที่ตั้งแบบโลคัลเป็นการชั่วคราว เช่นการดักจับไฟล์สำหรับข้อมูลที่เข้ามา
perform	ยอมให้คุณใช้คำสั่งของระบบปฏิบัติการของเวิร์กสเตชันภายใน ATE
quit	ออกจากโปรแกรม ATE

หมายเหตุ: จากลำดับของคีย์ควบคุม CAPTURE_KEY, MAINMENU_KEY และ PREVIOUS_KEY เฉพาะ PREVIOUS_KEY ที่สามารถใช้จากเมนูหลักของ ATE Unconnected

เมนูหลักของ ATE ที่ถูกเชื่อมต่อ:

ใช้คำสั่งย่อย **connect** จากเมนูหลักของ ATE Unconnected เพื่อแสดงเมนูหลักของการเชื่อมต่อ

ทางเลือกอื่น กด MAINMENU_KEY ขณะที่เชื่อมต่อกับระบบแบบรีโมต

คุณสามารถใช้คำสั่งย่อยต่อไปนี้จากเมนูหลักของ ATE ที่เชื่อมต่อ สำหรับคำจำกัดความของคำสั่งย่อยเหล่านี้ อ้างถึง คำสั่ง**ate** ใน*ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 1* ในการใช้คำสั่งย่อย พิมพ์ตัวอักษรแรกของคำสั่งย่อยที่จุดรับคำสั่งบนเมนู ตัวอย่างเช่น พิมพ์ a เพื่อใช้คำสั่งย่อย **alter**

ไอเท็ม	คำอธิบาย
alter	เปลี่ยนคุณลักษณะของการส่งข้อมูลเป็นการชั่วคราว เช่นความเร็วในการส่ง
break	ส่งสัญญาณ break ไปยังระบบแบบรีโมต
help	แสดงข้อมูลวิธีใช้
modify	แก้ไขค่าที่ตั้งแบบโลคัลที่ถูกใช้โดยอีมูเลเตอร์เป็นการชั่วคราว เช่นการดักจับไฟล์สำหรับข้อมูลที่เข้ามา
perform	ยอมให้คุณใช้คำสั่งของระบบปฏิบัติการของเวิร์กสเตชันภายใน ATE
quit	ออกจากโปรแกรม ATE
receive	รับไฟล์จากระบบแบบรีโมต
send	ส่งไฟล์ไปยังระบบแบบรีโมต
terminate	ยกเลิกการเชื่อมต่อ ATE

ลำดับคีย์การควบคุม ATE ทั้งสามนี้สามารถถูกใช้จากเมนูหลักของ ATE Connected

ลำดับของคีย์ควบคุมของ ATE

ใช้คีย์การควบคุมต่อไปนี้กับ ATE เปลี่ยนลำดับของคีย์สำหรับแต่ละฟังก์ชันโดยการแก้ไขไฟล์ `ate.def`

ไอเท็ม	คำอธิบาย
CAPTURE_KEY	เริ่มหรือหยุดการบันทึกข้อมูลที่ถูกแสดงบนหน้าจอระหว่างการเชื่อมต่อ ลำดับของคีย์แบบดีฟอลต์สำหรับ CAPTURE_KEY คือ Ctrl-B CAPTURE_KEY มีผลกับการสวิตช์หรือปิดเปิด การกดคีย์การควบคุมนี้จะเริ่มต้นการบันทึกข้อมูล การกดคีย์การควบคุมนี้เป็นครั้งที่สอง จะหยุดการบันทึกข้อมูล ข้อมูลจะถูกบันทึกในไฟล์การดักจับที่ถูกกำหนดในไฟล์ `ate.def` ชื่อดีฟอลต์ของไฟล์ดักจับคือไฟล์ `$HOME/kapture` ใช้คำสั่งย่อย **modify** เพื่อเปลี่ยนชื่อไฟล์การดักจับชั่วคราว แก้ไขไฟล์ ATE ดีฟอลต์เพื่อเปลี่ยนชื่อของไฟล์การดักจับแบบถาวร โปรดดู “การแก้ไขไฟล์ ATE ดีฟอลต์” ในหน้า 690 ลำดับของคีย์ CAPTURE_KEY จะไม่ทำงานขณะที่เทอร์มินัลทำการถ่ายโอนไฟล์ และมันจะใช้ได้เฉพาะเมื่อมีการเชื่อมต่อแล้ว ถ้าคุณกดลำดับของคีย์ CAPTURE_KEY ก่อนที่การเชื่อมต่อจะเริ่ม คำสั่งถัดไปที่ถูกใส่จะไม่สำเร็จ และข้อความข้อผิดพลาดจะถูกแสดง

ไอเท็ม	คำอธิบาย
PREVIOUS_KEY	กลับไปยังหน้าจอที่ถูกแสดงก่อนหน้านี้ PREVIOUS_KEY ถูกใช้เพื่อหยุดการถ่ายโอนไฟล์ ลำดับดีฟอลต์ของคีย์สำหรับ PREVIOUS_KEY คือ Ctrl-R PREVIOUS_KEY สามารถถูกใช้จากเมนูหลักของ ATE
MAINMENU_KEY	แสดงเมนูหลักของ Connected เพื่อที่คุณสามารถใช้คำสั่งย่อย ATE ลำดับของคีย์แบบดีฟอลต์สำหรับ MAINMENU_KEY คือ Ctrl-V ใช้คีย์การควบคุมนี้เพื่อแสดงเมนูหลักของ Connected หลังจากการทำการเชื่อมต่อไปยังระบบแบบรีโมตแล้ว ถ้าคุณกดลำดับของคีย์ MAINMENU_KEY ก่อนที่การเชื่อมต่อจะเริ่ม คำสั่งถัดไปที่ถูกใส่จะไม่สำเร็จ และข้อความข้อผิดพลาดจะถูกแสดง โดยการปรับแต่งไฟล์ ATE ดีฟอลต์ คุณสามารถเปลี่ยนการตั้งค่าคีย์ควบคุมแบบถาวร และชื่อของไฟล์ดักจับ โปรดดู “การแก้ไขไฟล์ ATE ดีฟอลต์” ในหน้า 690

การปรับแต่ง ATE

ATE จะสร้างไฟล์ `ate.def` ดีฟอลต์ในไดเร็กทอรีปัจจุบัน เมื่อผู้ใช้รัน ATE เป็นครั้งแรก แก้ไขไฟล์ `ate.def` เพื่อปรับแต่งลักษณะต่างๆของ ATE

ตัวอย่างเช่น ผู้ใช้สามารถเปลี่ยนชื่อของไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ ชนิดของโปรโคตอลการถ่ายโอนที่ใช้เพื่อส่งและรับไฟล์จากระบบแบบรีโมต และอัตรา baud ที่ ATE ต้องการให้โมเด็มใช้ อ้างถึง“การแก้ไขไฟล์ ATE ดีฟอลต์” ในหน้า 690 สำหรับข้อมูลเพิ่มเติมเกี่ยวกับไฟล์ `ate.def`

ผู้ใช้สามารถเปลี่ยนลักษณะโดยเฉพาะของ ATE เป็นการชั่วคราวโดยใช้คำสั่งย่อย **modify** และ **alter** คำสั่งย่อยเหล่านี้สามารถเปลี่ยนค่าดีฟอลต์ของ ATE ทั้งหมด ยกเว้นลำดับของคีย์ควบคุม (ซึ่งสามารถถูกเปลี่ยนโดยการแก้ไขไฟล์ดีฟอลต์) และชื่อของไดเร็กทอรีการหมุนโทรศัพท์ (ซึ่งสามารถถูกเปลี่ยนโดยใช้คำสั่งย่อย **directory** หรือแก้ไขดีฟอลต์ไฟล์) การเปลี่ยนแปลงที่ทำโดยใช้คำสั่งย่อย **modify**, **alter** หรือ **directory** จะมีผลเฉพาะกับเซสชันนั้นของ ATE ครั้งต่อไปที่ผู้ใช้รัน ATE ค่าที่ตั้งที่ถูกใช้จะเป็นค่าที่ถูกกำหนดในไฟล์ดีฟอลต์

เมื่อใช้โมเด็มกับ ATE ผู้ใช้สามารถสร้างไดเร็กทอรีการหมุนโทรศัพท์ได้มากถึง 20 หมายเลข คำสั่งย่อย **directory**จะแสดงหมายเลขโทรศัพท์ในรูปแบบเมนูและให้ผู้ใช้เลือกระบบที่ต้องการเรียก โปรดอ้างถึง “การตั้งค่าไดเร็กทอรีการหมุนโทรศัพท์ของ ATE” ในหน้า 685 สำหรับข้อมูลเพิ่มเติม

โดยการใช้ไดเร็กทอรีการหมุนโทรศัพท์ ผู้ใช้สามารถหลีกเลี่ยงการหาหมายเลขโทรศัพท์เมื่อต้องการเรียกไปยังระบบนั้นๆ ผู้ใช้ยังสามารถระบุคุณลักษณะการส่งข้อมูลนั้นๆในไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ นี่จะมีประโยชน์ถ้าบางการเชื่อมต่อใช้คุณลักษณะที่ต่างจาก ATE แบบดีฟอลต์

คุณสามารถสร้างไดเร็กทอรีการหมุนโทรศัพท์ส่วนตัว และผู้ดูแลระบบสามารถสร้างไดเร็กทอรีการหมุนโทรศัพท์ของทั้งระบบ ระบุไดเร็กทอรีการหมุนโทรศัพท์ที่ต้องการใช้ในไฟล์ ATE ดีฟอลต์ โปรดดู “การตั้งค่าไดเร็กทอรีการหมุนโทรศัพท์ของ ATE” ในหน้า 685 สำหรับข้อมูลเพิ่มเติม

ไฟล์คอนฟิกูเรชัน ate.def:

ไฟล์ `ate.def` จะตั้งค่าดีฟอลต์สำหรับใช้ในการเชื่อมต่อแบบอะซิงโครนัสและการถ่ายโอนไฟล์

ไฟล์นี้ถูกสร้างในไดเร็กทอรีปัจจุบันระหว่างการรัน ATE เป็นครั้งแรก ไฟล์ `ate.def` ประกอบด้วยค่าดีฟอลต์ในโปรแกรม ATE ที่ใช้สำหรับต่อไปนี้ :

- คุณลักษณะการส่งข้อมูล
- คุณลักษณะระบบแบบโลคัล
- ไฟล์ไดเร็กทอรีการหมุน
- คีย์ควบคุม

ครั้งแรก โปรแกรม ATE จะรันจากไดเร็กทอรีโดยเฉพาะ มันจะสร้างไฟล์ `ate.def` ในไดเร็กทอรีนั้น

```
LENGTH          8
STOP            1
PARITY          0
RATE            1200
DEVICE          tty0
INITIAL         ATDT
FINAL
WAIT            0
ATTEMPTS        0
TRANSFER        p
CHARACTER       0
NAME            kapture
LINEFEEDS       0
ECHO            0
VT100           0
WRITE           0
XON/XOFF        1
DIRECTORY       /usr/lib/dir
CAPTURE_KEY     002
MAINMENU_KEY    026
PREVIOUS_KEY    022
```

แก้ไขไฟล์ `ate.def` ด้วยเท็กซ์เอดิเตอร์ใดๆเพื่อแก้ไขค่าของคุณลักษณะเหล่านี้แบบถาวร การเปลี่ยนค่าของคุณลักษณะเหล่านี้แบบชั่วคราวโดยคำสั่งย่อย ATE **alter** และ **modify** ที่สามารถเข้าถึงได้จาก ATE Main Menu

พิมพ์ชื่อของพารามิเตอร์ในตัวอักษรพิมพ์ใหญ่ในไฟล์ `ate.def` สะกดชื่อพารามิเตอร์ให้ตรงกับที่มันปรากฏในไฟล์ดีฟอลต์ดั้งเดิม ระบุเพียงหนึ่งพารามิเตอร์ต่อบรรทัด การกำหนดค่าสำหรับพารามิเตอร์ไม่ถูกต้องจะทำให้ ATE คืนข้อความของระบบ อย่างไรก็ตาม โปรแกรมยังจะรันต่อโดยใช้ค่าดีฟอลต์ เหล่านี้เป็นพารามิเตอร์ของไฟล์ `ate.def` :

LENGTH

ระบุจำนวนบิตในอักขระข้อมูล ความยาวนี้ต้อง เท่ากับความยาวที่กำหนดโดยระบบรีโมต

อ็อพชัน:
7 หรือ 8
ดีฟอลต์: 8

STOP ระบุจำนวนของ stop บิตที่ต่อท้ายตัวอักษรเพื่อส่งสัญญาณว่าเป็นตัวอักษรสิ้นสุดระหว่างการส่งข้อมูล จำนวนนี้ต้องตรง กับจำนวนบิตหยุดที่ระบบรีโมตใช้

อ็อพชัน:
1 หรือ 2
ดีฟอลต์: 1

PARITY

ตรวจสอบว่าอักขระถูกส่งไปยังหรือจากระบบรีโมต สำเร็จหรือไม่ ต้องตรงกับพาริตีของระบบรีโมต

ตัวอย่างเช่น หากผู้ใช้เลือกพาริตีคู่ เมื่อจำนวนของบิต 1 ใน อักขระเป็นเลขคี่ พาริตีบิตถูกเปิดทำงานเพื่อทำให้จำนวนบิต 1 เป็นคู่

อ็อพชัน: 0 (ไม่มี), 1 (จำนวนคี่) หรือ 2 (จำนวนคู่)
ดีฟอลต์ : 0

RATE ระบุอัตรา baud หรือจำนวนของบิตที่ส่งต่อวินาที (bps) ความเร็ว ต้องตรงกับความเร็วของโมเด็มและของระบบรีโมต

อ็อพชัน : 50,75,110,134,150,300,600,1200,1800,2400,4800,9600,19200
ดีฟอลต์: 1200

DEVICE

ระบุชื่อของพอร์ตอะซิงโครนัสที่ใช้เชื่อมต่อไปยัง ระบบรีโมต

อ็อพชัน: ชื่อพอร์ตที่สร้างแบบโลคัล
ดีฟอลต์: tty0

INITIAL

ระบุส่วนนำหน้าการหมุนโทรศัพท์ที่ต้องขึ้นต้นหมายเลขโทรศัพท์เมื่อผู้ใช้ หมุนโทรศัพท์อัตโนมัติด้วยโมเด็ม สำหรับคำสั่งการหมุนโทรศัพท์ที่เหมาะสม ศึกษาจากเอกสารของโมเด็ม

อ็อพชัน: ATDT, ATDP หรืออื่นๆ จะขึ้นอยู่กับชนิดของโมเด็ม
ดีฟอลต์: ATDT

FINAL ระบุส่วนลงท้ายการหมุนโทรศัพท์ สตริงที่ต้องต่อท้ายหมายเลขโทรศัพท์เมื่อ หมุนโทรศัพท์อัตโนมัติด้วยโมเด็ม สำหรับคำสั่งการหมุนโทรศัพท์ที่เหมาะสม ศึกษาจากเอกสารของโมเด็ม

อ็อพชัน: Blank (ไม่มี) หรือส่วนต่อท้ายโมเด็มที่ใช้ได้
ดีฟอลต์ : ไม่มีค่าดีฟอลต์

WAIT ระบุเวลาที่รอระหว่างการพยายามหมุนโทรศัพท์ใหม่ ช่วงเวลารอจะไม่เริ่มต้นจนกว่า จำนวนครั้งของความพยายามเชื่อมต่อครบ หรือจนกระทั่งถูกอินเตอร์รัปต์ ถ้าพารามิเตอร์ ATTEMPTS ถูกตั้งเป็น 0 จะไม่มีการพยายามหมุนโทรศัพท์ใหม่

อ็อพชัน: 0 (ไม่มี) หรือจำนวนเต็มบวกที่ระบุจำนวนวินาทีที่จะรอ
ดีฟอลต์: 0

ATTEMPTS

ระบุจำนวนครั้งสูงสุดที่โปรแกรม ATE โทรซ้ำ เพื่อทำการเชื่อมต่อ ถ้าพารามิเตอร์ ATTEMPTS ถูกตั้งเป็น 0 จะไม่มีการพยายามหมุนโทรศัพท์ใหม่

อ็อพชัน: 0 (ไม่มี) หรือจำนวนเต็มบวกที่ระบุจำนวนของความพยายาม
ดีฟอลต์: 0

TRANSFER

กำหนดรูปแบบของอะซิงโครนัสโปรโตคอลที่ถ่ายโอนไฟล์ระหว่าง การเชื่อมโยง

p (pacing)

File transfer protocol ควบคุมอัตราการส่งข้อมูลโดยการรอ สำหรับตัวอักขระที่ระบุ หรือเป็นจำนวนวินาทีค่าหนึ่งระหว่าง การส่งข้อมูลบรรทัด สิ่งนี้จะช่วยป้องกันข้อมูลสูญหายเมื่อบล็อกการส่งข้อมูล มีขนาดใหญ่เกินไป หรือส่งเร็วเกินไปที่ระบบจะสามารถนำไปประมวลผลได้

x (xmodem)

File transfer protocol 8 บิตเพื่อตรวจหาข้อผิดพลาดการส่งข้อมูล และส่งข้อมูลใหม่

อ็อพชัน: p (
pacing) หรือ x (xmodem)
ดีฟอลต์: p

CHARACTER

ระบุชนิดของการ pacing โปรโตคอลที่ใช้ สัญญาณส่งหนึ่งบรรทัด เลือกหนึ่งตัวอักษร

เมื่อคำสั่งย่อย **send** พบอักขระ line-feed ขณะส่งข้อมูล คำสั่งย่อยจะรอรับ อักขระการกำหนดก่อนส่งบรรทัดถัดไป

เมื่อคำสั่งย่อย **receive** พร้อมรับข้อมูล จะส่งอักขระการกำหนด จากนั้นรอ 30 วินาทีเพื่อรับข้อมูล คำสั่งย่อย **receive** ส่ง อักขระการกำหนดอีกครั้งเมื่อใดก็ตามที่พบอักขระปัดแคร่ใน ข้อมูล คำสั่งย่อย **receive** สิ้นสุดเมื่อไม่ได้รับ ข้อมูลเป็นเวลา 30 วินาที

อ็อพชัน : ตัวอักขระใดๆ
ดีฟอลต์: 0

Interval

จำนวนวินาทีที่ระบบรอระหว่างการส่งแต่ละบรรทัด ค่าของตัวแปร Interval ต้องเป็นเลขจำนวนเต็ม ค่าดีฟอลต์คือ 0 ระบุว่าการกำหนดการหน่วงเวลา 0 วินาที

ดีฟอลต์: 0

NAME ชื่อไฟล์สำหรับข้อมูลที่เข้ามา (ไฟล์ที่ดักจับ)

อ็อพชัน: ชื่อไฟล์ที่ใช้ได้ยาวน้อยกว่า 40 ตัวอักษร
ดีฟอลต์: kapture

LINEFEEDS

เพิ่มอักขระ ป้อนบรรทัดหลังอักขระปัดแคร่ทุกครั้งในกระแสข้อมูลขาเข้า

อ็อพชัน: 1 (on) หรือ 0 (off)
ดีฟอลต์: 0

ECHO แสดงอินพุตที่ถูกพิมพ์ของผู้ใช้ สำหรับรีโมตคอมพิวเตอร์ซึ่งสนับสนุน การ echo อักขระแต่ละตัวที่ส่งไปจะส่งคืนและแสดงบนหน้าจอ เมื่อพารามิเตอร์ ECHO ถูกเปิดใช้ แต่ละอักขระจะถูกแสดงสองครั้ง ครั้งแรก เมื่อถูกใส่เข้าไป และอีกครั้งเมื่อส่งคืนผ่านการเชื่อมต่อ เมื่อพารามิเตอร์ ECHO ถูกปิด แต่ละอักขระจะถูกแสดงเมื่อมันถูกส่งคืนผ่านการเชื่อมต่อ

อ็อพชัน: 1 (on) หรือ 0 (off)
ดีฟอลต์: 0

VT100

โลคัลคอนโซลจะ อีมูเลตเทอร์มินัล DEC VT100 ดังนั้นสามารถใช้โค้ด DEC VT100 กับ ระบบแบบรีโมต เมื่อแฟล็ก VT100 ถูกปิดใช้โลคัลคอนโซลจะทำหน้าที่ เหมือนเวิร์กสเตชัน

อ็อพชัน: 1 (on) หรือ 0 (off)
ดีฟอลต์: 0

WRITE

ดักจับข้อมูลที่เข้ามาและเราต์มันไปยังไฟล์ที่ถูกระบุในพารามิเตอร์ NAME เช่นเดียวกับเพื่อแสดง การรวมกันของอักขระปัดแคร่และ line-feed จะถูกแปลงเป็นอักขระขึ้นบรรทัดใหม่ก่อนเขียนลงในไฟล์การดักจับ ในไฟล์ที่มีอยู่แล้ว ข้อมูลจะถูกต่อท้ายไฟล์

CAPTURE_KEY (โดยทั่วไปเป็นลำดับของคีย์ Ctrl-B) สามารถถูกใช้เพื่อเปิดปิดโหมดการดักจับระหว่างการเชื่อมต่อ

อ็อพชัน: 1 (on) หรือ 0 (off)
ดีฟอลต์: 0

XON/XOFF

ควบคุมการส่งข้อมูลที่พอร์ตดังต่อไปนี้ :

- เมื่อได้รับสัญญาณ XOFF การส่งจะหยุด
- เมื่อได้รับสัญญาณ XON การส่งจะทำงานต่อ
- สัญญาณ XOFF จะถูกส่งเมื่อบัฟเฟอร์รับข้อมูลเกือบเต็มแล้ว
- ส่งสัญญาณ XON เมื่อบัฟเฟอร์ ว่างแล้ว

อ็อพชัน: 1 (On) หรือ 0 (Off)
ดีฟอลต์: 1

DIRECTORY

ชื่อของไฟล์ที่ประกอบด้วยไดเร็กทอรีการหมุนโทรศัพท์ของผู้ใช้

ดีฟอลต์: ไฟล์ /usr/lib/dir

CAPTURE_KEY

กำหนดลำดับของคีย์ที่เปิดปิดโหมดการดักจับ เมื่อกด CAPTURE_KEY (โดยทั่วไปเป็นลำดับของคีย์ Ctrl-B) จะเริ่มหรือหยุดการดักจับ (การบันทึก) ข้อมูลที่ถูกแสดงบนหน้าจอระหว่างการเชื่อมต่อที่แอ็คทีฟ

อ็อพชัน: ตัวอักขระการควบคุมแบบ ASCII ใดๆ
ดีฟอลต์: ASCII octal 002 (STX)

MAINMENU_KEY

กำหนดลำดับของคีย์ควบคุมที่คืน Connected Main Menu เพื่อที่ผู้ใช้สามารถใช้คำสั่งระหว่างการเชื่อมต่อที่แอ็คทีฟ MAINMENU_KEY (โดยทั่วไปเป็นลำดับของคีย์ Ctrl-V) จะทำงานเฉพาะจากสถานะที่เชื่อมต่ออยู่

อ็อพชัน: ตัวอักขระการควบคุมแบบ ASCII ใดๆ
ดีฟอลต์: ASCII octal 026 (SYN)

PREVIOUS_KEY

กำหนดลำดับของคีย์ที่แสดงหน้าจอก่อนหน้านี้ทุกเวลาระหว่างโปรแกรม หน้าจอจะถูกแสดงต่างกันออกไปโดยขึ้นอยู่กับหน้าจอที่ใช้เมื่อผู้ใช้กด PREVIOUS_KEY (โดยทั่วไปเป็นลำดับของคีย์ Ctrl-R)

อ็อพชัน: ตัวอักขระการควบคุมแบบ ASCII ใดๆ
ดีฟอลต์: ASCII octal 022 (DC2) ตัวอักขระควบคุมแบบ ASCII ที่ถูกแม็พกับสัญญาณอินเตอร์รัปต์

การตั้งค่าไดเร็กทอรีการหมุนโทรศัพท์ของ ATE

ไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ของ ATE จะลิสต์หมายเลขโทรศัพท์ที่โปรแกรม ATE ใช้เพื่อเริ่มต้นการเชื่อมต่อแบบรีโมตด้วยโมเด็ม

ในการตั้งไดเร็กทอรีการหมุนโทรศัพท์ของ ATE สิ่งที่ต้องการก่อนต่อไปนี้ต้องพร้อม :

- โปรแกรม Asynchronous Terminal Emulation (ATE) ต้องถูกตั้งค่าบนระบบ
- ในการตั้งไดเร็กทอรีการหมุนโทรศัพท์ของทั้งระบบ ผู้ใช้ต้องมีสิทธิที่จะเขียนไฟล์ /usr/lib/dir

ผู้ใช้ตั้งชื่อไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ด้วยชื่อไฟล์ที่ถูกต้อง และวางมันในไดเร็กทอรีที่มีสิทธิอ่านและเขียน แก้ไขไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ด้วยเท็กซ์เอดิเตอร์แบบ ASCII ข้อมูลดีฟอลต์สำหรับไดเร็กทอรีที่ใช้หมุนโทรศัพท์สำหรับโปรแกรม ATE จะอยู่ในไฟล์ /usr/lib/dir ดังแสดงต่อไปนี้ :

หมายเหตุ: ในเนื้อหาต่อไปนี้ บาง entry ของ ATE ถูกแยกเป็นคนละบรรทัดเพื่อให้สามารถอ่านได้ ในไฟล์ไดเร็กทอรีการหมุนโทรศัพท์จริง ส่วนประกอบทั้งหมดจะอยู่บนบรรทัดเดียว

```
#   COMPONENT_NAME: BOS dir
#
#   FUNCTIONS:
#
#   ORIGINS: 27
#
#
#   (C) COPYRIGHT International Business Machines Corp. 1985, 1989
#   Licensed Materials - Property of IBM
#
#   US Government Users Restricted Rights - Use, duplication or
#   disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
#   dir - sample dialing directory
#
#
# Micom   9,555-9400 1200 7 1 2 0 0
# R2O     9,555-9491 1200 7 1 2 0 0
# QT      9,555-8455 1200 7 1 2 0 0
# Dallas1 9,555-7051 1200 8 1 0 0 0
```

ผู้ใช้สามารถเข้าถึงข้อมูลของไดเร็กทอรีการหมุนโทรศัพท์จากภายใน ATE โดยใช้คำสั่งย่อย **directory** ที่มีใน **UNCONNECTED MAIN MENU** หน้าจอจะแสดงข้อมูลไดเร็กทอรีดังที่มันจะแสดงจากภายในโปรแกรม ATE

ผู้ใช้สามารถมีได้มากกว่าหนึ่งไดเร็กทอรีการหมุนโทรศัพท์ เพื่อเปลี่ยนไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ที่โปรแกรม ATE ใช้ ผู้ใช้ต้องแก้ไขไฟล์ ate.def ในไดเร็กทอรีปัจจุบัน

> **หมายเหตุ:** ไฟล์ไดเร็กทอรีการหมุนโทรศัพท์สามารถมีได้มากถึง 20 บรรทัด (หนึ่ง entry ต่อบรรทัด) ATE จะไม่สนใจบรรทัดที่ตามมา

ไฟล์ไดเร็กทอรีการหมุนโทรศัพท์จะเหมือนกับหน้าในสมุดโทรศัพท์ที่ประกอบด้วย entry สำหรับระบบรีโมตที่จะถูกเรียกไปโดยโปรแกรม ATE รูปแบบของ entry ของไดเร็กทอรีการหมุนโทรศัพท์ คือ :

```
Name Phone Rate Length StopBit Parity Echo Linefeed
```

ฟิลด์ต้องถูกแยกโดยอย่างน้อยหนึ่งช่องว่าง ช่องว่างที่เพิ่มเติมสามารถถูกใช้เพื่อทำให้แต่ละ entry อ่านง่ายขึ้น ฟิลด์ คือ :

Name จะระบุหมายเลขโทรศัพท์ ชื่อสามารถเป็นการรวมกันของอักขระ 20 ตัว หรือน้อยกว่า ใช้ _ (ขีดล่าง) แทนที่ช่องว่างระหว่างคำในชื่อ ตัวอย่างเช่น data_bank

Phone หมายเลขโทรศัพท์ที่จะโทร ตัวเลขสามารถยาวถึง 40 อักขระ ศึกษาเอกสารของโมเด็มสำหรับลิสต์ของตัวเลขและอักขระที่สามารถรับได้ ตัวอย่างเช่น ถ้า 9 ต้องถูกหมุนเพื่อต่อออกภายนอก ให้ใส่ 9, (เลข 9 และคอมมา) ก่อนหมายเลขโทรศัพท์ ดังนี้ : 9,1112222

แม้ว่าหมายเลขโทรศัพท์สามารถยาวถึง 40 ตัวอักษร คำสั่งย่อย directory จะแสดงเพียง 26 ตัวแรก

`Rate` อัตราการส่งข้อมูลหรือ อัตรา baud ในหน่วยบิตต่อวินาที (bps) กำหนดจำนวนของตัวอักขระที่ถูกส่งต่อวินาที เลือกอัตรา baud ที่สอดคล้องกับสายการสื่อสารที่ใช้ ต่อไปนี้เป็นอัตราที่สามารถใช้ได้ :

50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 9600 หรือ 19200

สำหรับอัตรา baud ที่ไม่ใช่ POSIX การตั้งค่าอัตราที่ 50 ทำให้ ATE ใช้อัตรา baud ที่ถูกตั้งค่าผ่าน SMIT สำหรับอุปกรณ์นั้น

`Length` จำนวนบิตที่ประกอบเป็นตัวอักขระ entry สำหรับฟิลด์ Length สามารถเป็น 7 หรือ 8

`StopBit`
stop บิตแสดงจุดสิ้นสุดของอักขระ entry สำหรับฟิลด์ StopBit สามารถเป็น 1 หรือ 2

`Parity` ตรวจสอบว่าอักขระถูกส่งไปยังหรือจากระบบรีโมต สำเร็จหรือไม่ entry สำหรับฟิลด์ Parity สามารถเป็น 0 (ไม่มี), 1 (คี่) หรือ 2 (คู่)

`Echo` กำหนดว่าอักขระที่ถูกพิมพ์จะถูกแสดงแบบโลคัล entry สำหรับฟิลด์ Echo สามารถเป็น 0 (ปิด) หรือ 1 (เปิด)

ป้อนบรรทัด
เพิ่มอักขระ line-feed ที่ท้ายแต่ละบรรทัด ของข้อมูลขาเข้าจากระบบรีโมต อักขระ line-feed ทำหน้าที่เดียวกับอักขระ carriage-return และขึ้นบรรทัดใหม่ entry สำหรับฟิลด์ Linefeed สามารถเป็น 0 (ปิด) หรือ 1 (เปิด)

หมายเหตุ: การแก้ไขหรือการแม็พใหม่อาจจำเป็นถ้าคีย์การควบคุมชนกับแอ็พพลิเคชัน ตัวอย่างเช่น ถ้าคีย์การควบคุมที่ถูกแม็พสำหรับโปรแกรม ATE ชนกับในเท็กซ์เอดิเตอร์ ให้แม็พคีย์ควบคุมของ ATE ใหม่

หมายเหตุ: อักขระการควบคุม ASCII อาจเป็นรูปแบบ เลขฐานแปด เลขฐานสิบ หรือเลขฐานสิบหก ดังต่อไปนี้ :

เลขฐานแปด
000 ถึง 037 ต้องนำหน้าด้วยศูนย์

เลขฐานสิบ
0 ถึง 31

เลขฐานสิบหก
0x00 ถึง 0x1F ต้องนำหน้าด้วย 0x x อาจเป็นตัวพิมพ์ใหญ่ หรือตัวพิมพ์เล็ก

สร้างไฟล์ `ate.def` ที่กำหนดคุณลักษณะเหล่านั้นเพื่อเปลี่ยนคุณลักษณะของ ATE อีมูเลชัน ตัวอย่างเช่น เพื่อเปลี่ยน RATE เป็น 300 bps DEVICE เป็น tty3 โหมด TRANSFER เป็น x (โปรโตคอล xmodem) และ DIRECTORY เป็น my.dir สร้าง `ate.def` พร้อมกับ entry ต่อไปนี้ ในไดเร็กทอรีที่รันโปรแกรม ATE :

```
RATE        300
DEVICE      tty3
TRANSFER    x
DIRECTORY   my.dir
```

โปรแกรมจะใช้ค่าที่ถูกระบุจากเวลาที่โปรแกรม ATE สตาร์ทจากไดเร็กทอรีนั้น

1. สร้างไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ :
   a. เปลี่ยนเป็นไดเร็กทอรีที่ไฟล์ไดเร็กทอรีการหมุนโทรศัพท์อยู่
   b. คัดลอกไฟล์ `/usr/lib/dir` เพื่อใช้เป็นเท็มเพลต เปลี่ยนชื่อไฟล์เป็นชื่อไฟล์ใดๆที่ถูกต้อง
   c. สร้าง entry ของหมายเลขโทรศัพท์โดยใช้รูปแบบที่ให้ในรูปแบบของไฟล์ไดเร็กทอรีการหมุนโทรศัพท์

d. บันทึกไฟล์

หมายเหตุ: ถ้าไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ใหม่จะต้องเป็นไฟล์ดีฟอลต์ของระบบ บันทึกไฟล์ด้วยชื่อ /usr/lib/dir

2. ถ้าชื่อไฟล์ของไดเร็กทอรีการหมุนโทรศัพท์ไม่ใช่ชื่อดีฟอลต์ (/usr/lib/dir) แก้ไขไฟล์ ate.def ในไดเร็กทอรีที่โปรแกรม ATE รัน เปลี่ยนพารามิเตอร์ *DIRECTORY* ในไฟล์ ate.def เป็นไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ใหม่ ดูที่ "การแก้ไขไฟล์ ATE ดีฟอลต์" ในหน้า 690
3. สตาร์ท ATE และดูไดเร็กทอรีการหมุนโทรศัพท์ด้วยคำสั่งย่อย **directory**

การโทรออกด้วย ATE

ใช้โพรซีเดอร์นี้เพื่อโทรออกจากระบบโดยใช้ ATE และไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ /usr/lib/dir ที่ถูกปรับแต่ง

ตรวจสอบว่าสิ่งที่ต้องการก่อนต่อไปนี้และเงื่อนไขมีพร้อมก่อนที่จะพยายามโทรออก

- ATE ถูกติดตั้งบนระบบ
- เชื่อมต่อโมเด็ม ตั้งค่า และพร้อมใช้งาน
- ผู้ใช้เป็นสมาชิกของกลุ่ม UUCP (ดูที่ "การตั้งค่า ATE" ในหน้า 677 สำหรับข้อมูลเพิ่มเติม)
- ไฟล์ไดเร็กทอรีการหมุนโทรศัพท์ /usr/lib/dirถูกปรับแต่งด้วยข้อมูลที่ถูกต้อง
- ไดเร็กทอรีการทำงานต้นกำเนิดของผู้ใช้ (pwd) ประกอบด้วยไฟล์ ate.def ที่ถูกอัพเดตอย่างถูกต้อง
- พอร์ต /dev/tty ต้องมีฟิลด์ ENABLE login ใน SMIT ตั้งเป็น disable, share หรือ delay

1. ป้อน:

   ```
   ate
   ```

2. ที่เมนูหลัก พิมพ์ d และกด Enter
3. พิมพ์ชื่อไฟล์ของไดเร็กทอรีที่คุณต้องการแสดงและกด Enter เพื่อใช้ไดเร็กทอรีปัจจุบัน แค่กด Enter
4. ใส่หมายเลขของ entry ของไดเร็กทอรีที่เหมาะสมภายใต้คอลัมน์ # เพื่อหมุนหมายเลขที่สอดคล้องกับหมายเลขโทรศัพท์

การถ่ายโอนไฟล์โดยใช้ ATE

ใช้โพรซีเดอร์ต่อไปนี้เพื่อถ่ายโอนไฟล์จากโลคัลโฮสต์ไปยังระบบรีโมต

ตรวจสอบว่าสิ่งที่ต้องการก่อนต่อไปนี้และเงื่อนไขมีพร้อมก่อนที่จะถ่ายโอนไฟล์โดยใช้ ATE:

- ต้องมีการสร้างการเชื่อมต่อโดยใช้โปรแกรม **ATE**
- โปรโตคอล Xmodem file transfer ต้องมีอยู่บนทั้งระบบแบบโลคัลและรีโมต บนระบบปฏิบัติการ Xmodem จะอยู่ในไดเร็กทอรี /usr/bin

1. รันคำสั่ง **xmodem** ต่อไปนี้บนระบบแบบรีโมตหลังจากล็อกอิน :

   ```
   xmodem -r newfile
   ```

   โดยที่ **r** เป็นแฟล็ก Xmodem เพื่อรับและ *newfile* เป็นชื่อของไฟล์ที่จะได้รับ ชื่อนี้ไม่จำเป็นต้องเหมือนกับไฟล์ที่จะถูกส่ง
2. กด Enter
3. ข้อความต่อไปนี้จะถูกแสดง :

   ```
   ate: 0828-005 The system is ready to receive file newfile. Use Ctrl-X to stop xmodem.
   ```

   ถ้าข้อความไม่ถูกแสดง ระบบอาจไม่ได้ติดตั้งโปรแกรม **xmodem** หรืออยู่ในคำสั่ง PATH ของมัน

4. กด Ctrl-V เพื่อกลับไปยัง ATE CONNECTED MAIN MENU
5. กดคีย์ S เพื่อส่งไฟล์
6. ข้อความต่อไปนี้จะถูกแสดง :

   ```
   Type the name of the file you wish to send and press Enter. To use the last
   file name (), just press Enter.
   ```

7. ใส่ชื่อและพาธแบบเต็มของไฟล์ที่จะถูกถ่ายโอน
8. กด Enter
9. ATE จะแสดงข้อความต่อไปนี้และเริ่มต้นถ่ายโอนไฟล์:

   ```
   ate: 0828-024 The program is ready to send file newfile. You will receive another
   message when the file transfer is complete.
   ate: 0828-025 The system is sending block 1.
   ate: 0828-025 The system is sending block 2.
   ate: 0828-015 The file transfer is complete.
   ate: 0828-040 Press Enter
   ```

10. กด Enter เมื่อการถ่ายโอนเสร็จสิ้น

การรับไฟล์โดยใช้ ATE

ใช้โพรซีเดอร์นี้เพื่อรับไฟล์ที่ถูกถ่ายโอนจากโฮสต์แบบรีโมต

ตรวจสอบว่าสิ่งที่ต้องการก่อนต่อไปนี้และเงื่อนไขมีพร้อมก่อนที่จะรับไฟล์โดยใช้ ATE:

- ต้องมีการสร้างการเชื่อมต่อโดยใช้โปรแกรม ATE
- โปรโตคอล Xmodem file transfer ต้องมีอยู่บนทั้งระบบแบบโลคัลและรีโมต บนระบบปฏิบัติการ Xmodem จะอยู่ในไดเร็กทอรี `/usr/bin`

1. รันคำสั่ง **xmodem** ต่อไปนี้บนระบบแบบรีโมตหลังจากล็อกอิน :

   ```
   xmodem -s newfile
   ```

   โดยที่ **s** เป็นคำสั่ง **xmodem** เพื่อส่งและ *newfile* เป็นชื่อและพาธแบบเต็มของไฟล์ที่จะถูกส่ง
2. กด Enter
3. ข้อความต่อไปนี้จะถูกแสดง :

   ```
   ate: 0828-005 The system is ready to send file newfile. Use ctrl-X to stop xmodem.
   ```

   ถ้าข้อความไม่ถูกแสดง ระบบอาจไม่ได้ติดตั้งโปรแกรม **xmodem** หรืออยู่ในคำสั่ง PATH ของมัน
4. กด Ctrl-V เพื่อกลับไปยัง ATE CONNECTED MAIN MENU
5. กดคีย์ R เพื่อรับไฟล์
6. ข้อความต่อไปนี้จะถูกแสดง :

   ```
   พิมพ์ชื่อของไฟล์ที่คุณต้องการเก็บข้อมูลที่ได้รับและกด
   Enter.  เพื่อใช้ชื่อไฟล์ล่าสุด () แค่กด Enter
   ```

7. ใส่ชื่อและพาธแบบเต็มของไฟล์ที่จะถูกถ่ายโอน
8. กด Enter
9. ATE จะแสดงข้อความต่อไปนี้และเริ่มต้นถ่ายโอนไฟล์:

```
ate: 0828-020 The program is ready to receive file newfile. You will
receive another message when the file transfer is complete.
ate: 0828-028 The system is receiving block 1.
ate: 0828-028 The system is receiving block 2.
ate: 0828-040 Press Enter.
```

10. กด Enter เมื่อการถ่ายโอนเสร็จสิ้น

การแก้ไขไฟล์ ATE ดีฟอลต์

เพื่อแก้ไขไฟล์ ATE ดีฟอลต์โปรแกรม ATE ต้องถูกตั้งค่าบนระบบ

เมื่อต้องการเปลี่ยนค่าที่ตั้งในไฟล์ `ate.def` :

1. เปิดไฟล์ `ate.def` ด้วยเท็กซ์เอดิเตอร์แบบ ASCII
2. ใส่ค่าใหม่สำหรับพารามิเตอร์ที่จะถูกเปลี่ยน ค่าอื่นสามารถถูกลบหรือปล่อยไว้ ระบบจะใช้ค่าดีฟอลต์ของมันสำหรับพารามิเตอร์ที่ถูกลบ
3. บันทึกไฟล์ `ate.def` ที่ถูกแก้ไข

การแก้ไขที่ทำกับไฟล์ `ate.def` จะมีผลในครั้งต่อไปที่ ATE ถูกรันจากไดเร็กทอรีที่ประกอบด้วยไฟล์ `ate.def` ที่ถูกปรับแต่ง

คุณสามารถเก็บคัดลอกของไฟล์ `ate.def` ในไดเร็กทอรีใดๆที่คุณมีสิทธิอ่านและเขียน ตัวอย่างเช่น ถ้าคุณต้องการรันโปรแกรม ATE โดยมีค่าดีฟอลต์ที่ต่างกันในเวลาต่างกัน ให้เก็บหลายคัดลอกของไฟล์ `ate.def` โดยมีค่าตั้งที่เหมาะสม ในไดเร็กทอรีย่อยที่ต่างกันของไดเร็กทอรี `$HOME` อย่างไรก็ตาม หลายคัดลอกของไฟล์ `ate.def` จะใช้หน่วยเก็บข้อมูลซองระบบทางเลือกอื่น เปลี่ยนค่าตั้งส่วนใหญ่เป็นการชั่วคราวด้วยคำสั่งย่อย ATE **alter** และ **modify** ใช้ entry ของไดเร็กทอรีการหมุนโทรศัพท์เพื่อเปลี่ยนค่าที่ตั้งสำหรับแต่ละการเชื่อมต่อของโมเด็ม โปรดดู “การตั้งค่าไดเร็กทอรีการหมุนโทรศัพท์ของ ATE” ในหน้า 685

การแก้ไขปัญหา ATE

เมื่อพบปัญหาทั่วไปเกี่ยวกับ ATE ต่อไปนี้ ให้พิจารณาโซลูชันเหล่านี้

ปัญหา: เมื่อคำสั่งการถ่ายโอนหรือรับไฟล์ **xmodem** ดูเหมือนจะแฮงค์ Ctrl-X จะแก้ปัญหา

โซลูชัน:

ตรวจสอบเมนู Alter เพื่อตรวจสอบว่าโปรโตคอล xmodem (หรือวิธี Transfer) ถูกใช้

ปัญหา: เมื่อถ่ายโอนหรือรับไฟล์ไฟล์จะถูกเลื่อนข้ามหน้าจอและข้อความถูกแสดงระบุว่าการถ่ายโอนหรือรับเรียบร้อยแล้ว เมื่อจริงๆแล้วมันยังไม่เรียบร้อย

โซลูชัน:

ตรวจสอบเมนู Alter เพื่อตรวจสอบว่าโปรโคตอล **xmodem** protocol (หรือวิธี Transfer) ถูกใช้

ปัญหา: เมื่อสตาร์ท ATE ผู้ใช้ได้รับข้อผิดพลาดต่อไปนี้ :

```
ate: 0828-008 The system tried to open port /dev/tty0 but failed. If the port name is not correct,
change it using the Alter menu. Or, take the action indicated by the system message shown below.

Connect: The file access permissions do not allow the specified action.
ate: 0828-040 Press Enter.
```

โซลูชัน:

บรรทัด Connect: ในข้อความแสดงข้อผิดพลาดจะทำให้ปัญหาแคบลง ตรวจสอบว่าผู้ใช้ที่จะรัน ATE เป็นสมาชิกของกลุ่ม UUCP เพื่อตรวจสอบสิ่งนี้ ผู้ใช้สามารถใส่ *id* บนบรรทัดรับคำสั่ง `uucp` ควรถูกแสดงในลิสต์ของเอาต์พุต

ปัญหา: เมื่อพยายามเชื่อมต่อกับ ATE ได้รับข้อผิดพลาดต่อไปนี้

```
ate: 0828-008 The system tried to open port /dev/tty0 but failed. If the port name is not
correct, change it using the Alter menu. Or, take the action indicated by the system
message shown below.

Connect: A file or directory in the path name does not exist.
ate: 0828-040 Press Enter.
```

โซลูชัน:

เลือก tty ที่ไม่ถูกต้องหรือไม่พร้อมใช้งานสำหรับใช้โดย ATE ตรวจสอบหน้าจอ Alter ใน ATE

ปัญหา: ถ่ายโอนไฟล์ถูกต้อง แต่ขนาดของไฟล์ใหญ่กว่าไฟล์ต้นฉบับ

โซลูชัน:

โปรโตคอล xmodem ขยายไฟล์ขณะถ่ายโอน เพื่อหลีกเลี่ยงปัญหานี้ ให้ใช้คำสั่ง **tar** เพื่อบีบอัดไฟล์แล้วถ่ายโอนมัน นี่ยังช่วยเอาชนะข้อจำกัดอื่นๆของ xmodem ที่สามารถส่งได้ทีละไฟล์ ผู้ใช้สามารถ **tar** หลายไฟล์เข้าด้วยกันใน tar อิมเมจเดียวและถ่ายโอนมันโดยใช้ xmodem

คำสั่ง ATE และคำสั่งย่อย

นี่คือลิสต์ของคำสั่ง ATE และคำสั่งย่อยพร้อมกับคำอธิบายแบบสั้นเกี่ยวกับสิ่งที่มันทำ

โปรดดู “รูปแบบของไฟล์ ATE”สำหรับข้อมูลอ้างอิงเพิ่มเติม

ไอเท็ม	คำอธิบาย
ate	สตาร์ทโปรแกรม ATE สำหรับคำจำกัดความของคำสั่งย่อยของมัน ที่ตามมา อ้างถึงคำสั่ง **ate** :
break	ใส่กิจกรรมปัจจุบันบนระบบแบบรีโมต
connect	เชื่อมต่อไปยังรีโมตคอมพิวเตอร์
directory	แสดงไดเร็กทอรีการหมุน ATE และให้คุณเลือก entry จากไดเร็กทอรีเพื่อเชื่อมต่อกับระบบแบบรีโมต
วิธีใช้	จัดเตรียมความช่วยเหลือสำหรับคำสั่งย่อย ATE
perform	ให้คุณใช้คำสั่งเวิร์กสเตชันของระบบปฏิบัติการขณะใช้ ATE
quit	ออกจากโปรแกรม ATE
receive	รับไฟล์จากระบบรีโมต
send	ส่งไฟล์ไปยังระบบไฟล์แบบรีโมต
terminate	ยุติการเชื่อมต่อ ATE ไปยังระบบรีโมต

นอกจากนี้ คำสั่ง **xmodem** มีประโยชน์สำหรับการถ่ายโอนไฟล์กับโปรโตคอล xmodem ซึ่งจะตรวจจับข้อผิดพลาดของการส่งข้อมูลระหว่างการส่งแบบอะซิงโครนัส

รูปแบบของไฟล์ ATE

รูปแบบของไฟล์ Asynchronous Terminal Emulation (ATE) จะรวม `ate.def`และรูปแบบของไดเร็กทอรีการหมุนโทรศัพท์

ไอเท็ม	คำอธิบาย
ate.def	ตั้งค่าดีฟอลต์สำหรับการเชื่อมต่อ
ไดเร็กทอรีการหมุนโทรศัพท์	กำหนดหมายเลขโทรศัพท์และตั้งค่าสำหรับการเชื่อมต่อโมเด็มที่ระบุ

ดูที่ “คำสั่ง ATE และคำสั่งย่อย” ในหน้า 691 สำหรับข้อมูลอ้างอิงเพิ่มเติม

ยูทิลิตี Dynamic screen

ยูทิลิตี dynamic screen หรือ คำสั่ง **dscreen**แบบฟิสิคัลเดียวสามารถเชื่อมต่อกับเทอร์มินัลเสมือนหลายเซสชัน (หน้าจอ) ในเวลาเดียวกันให้เทอร์มินัล

จุดประสงค์หลักเพื่อใช้กับเทอร์มินัลที่มีสองเพจของหน่วยความจำของหน้าจอหรือมากกว่า (ตัวอย่างเช่น IBM 3151 โมเดล 310 หรือหน้าจอ 410 พร้อม Cartridge สำหรับขยาย) ด้วยเทอร์มินัลเหล่านั้น การสลับระหว่างหน้าจอเสมือนยังสลับระหว่างเพจของหน้าจอเทอร์มินัลแบบฟิสิคัลทำให้แต่ละอิมเมจของหน้าจอเสมือนถูกบันทึกหรือเรียกคืน บนเทอร์มินัลที่ไม่มีหลายเพจของหน่วยความจำหน้าจอ คำสั่ง **dscreen** ยังสามารถถูกใช้เพื่อสลับระหว่างเซสชันของหน้าจอเสมือนแม้ว่าการแสดงของหน้าจอจะไม่ถูกเก็บไว้

หมายเหตุ: สำหรับการสนับสนุนแบบเต็มที่ของยูทิลิตี **dscreen** เทอร์มินัลต้องสามารถสลับเพจของหน้าจอภายในบนคำสั่งและต้องจำตำแหน่งของเคอร์เซอร์สำหรับแต่ละเพจ ขณะที่ยูทิลิตี **dscreen** จะทำงานบนทั้งเทอร์มินัลแบบฉลาดและแบบดัมม์ อิทเมจของหน้าจอจะไม่ถูกบันทึกระหว่างการเปลี่ยนแปลงของหน้าจอบนดัมม์เทอร์มินัล

ไฟล์ข้อมูลคอนฟิกูเรชันเทอร์มินัล

ยูทิลิตีไฟล์ข้อมูลคอนฟิกูเรชันเทอร์มินัล **dscreen** (หรือไฟล์ `dsinfo`) ถูกใช้เพื่อกำหนดชุดของคีย์ที่แตกต่างกันที่จะถูกใช้กับยูทิลิตี **dscreen**

ตัวอย่างเช่น นี่อาจทำได้เมื่อคีย์ยูทิลิตี **dscreen** ที่ถูกกำหนดไว้ดั้งเดิมชนกับแอ็พพลิเคชันซอฟต์แวร์ที่ใช้บนระบบ

ไฟล์ `dsinfo` ชนิดของเทอร์มินัลสันนิษฐานว่ามีเพจเดียวของหน่วยความจำของหน้าจอ ดังนั้น ถ้าเทอร์มินัลสนับสนุนเพจของหน่วยความจำของหน้าจอเพิ่มเติม ไฟล์ `dsinfo` ต้องถูกปรับแต่งเพื่อใช้ลำดับสำหรับการควบคุมเพจของหน่วยความจำที่เหมาะสม ดูที่เอกสารอ้างอิงเทอร์มินัลที่เหมาะสมสำหรับลำดับของการควบคุมที่ระบุ

ไฟล์ `dsinfo` ดีฟอลต์คือ `/usr/lbin/tty/dsinfo` ใช้แฟล็ก **–i** เพื่อระบุไฟล์ `dsinfo` อื่น ส่วนที่เหลือของส่วนนี้จะอ้างถึงไฟล์ดีฟอลต์นี้ อย่างไรก็ตาม ข้อมูลเดียวกันใช้ได้กับไฟล์ `dsinfo` ที่ถูกปรับแต่งที่คุณสร้าง

สำหรับข้อมูลเพิ่มเติมที่เกี่ยวข้องกับไฟล์ `dsinfo` ดูที่ “การกำหนดหน้าจอแบบไดนามิก” ในหน้า 694

การกำหนดแอ็คชันของคีย์ dscreen

เมื่อคำสั่ง **dscreen** ถูกประมวลผล มันจะสตาร์ทหน้าจอเสมือน บางคีย์บนคีย์บอร์ดของเทอร์มินัลจะไม่ถูกผ่านไปยังหน้าจอเสมือน ยูทิลิตี **dscreen** จะดักคีย์เหล่านี้และแอ็คชันนั้นแทนเมื่อมันถูกกด

แอ็คชันจะรวมถึง :

ไอเท็ม	คำอธิบาย
เลือก (ดูที่ “คีย์ dscreen Select”)	เลือกหน้าจอที่ระบุ
บล็อก (ดูที่ “คีย์ dscreen Block”)	บล็อกอินพุตและเอาต์พุต
ใหม่ (ดูที่ “คีย์ dcreen New”)	สตาร์ทเซสชันของหน้าจอใหม่
สิ้นสุด (ดูที่ “คีย์ dscreen End และ Quit” ในหน้า 694)	สิ้นสุดยูทิลิตี **dscreen**
ออก (ดูที่ “คีย์ dscreen End และ Quit” ในหน้า 694)	ออจจากยูทิลิตี **dscreen**
ก่อนหน้านี้ (ดูที่ “คีย์ dscreen Previous” ในหน้า 694)	สลับไปหน้าจอก่อนหน้านี้
ลิสต์ (ดูที่ “คีย์ dscreen List” ในหน้า 694)	ลิสต์คีย์ **dscreen** ที่ถูกกำหนดและแอ็คชันของมัน

ฟังก์ชันของแต่ละคีย์ขึ้นอยู่กับเทอร์มินัลและคำอธิบายของเทอร์มินัลในไฟล์ `/usr/lbin/tty/dsinfo`

คีย์ dscreen Select:

เมื่อหน้าจอเสมือนถูกสร้าง มันจะถูกกำหนดคีย์ที่เลือก

การกดคีย์ select จะทำให้เกิดแอ็คชันต่อไปนี้ :

- สวิตช์จากเทอร์มินัลแบบฟิสิคัลเป็นหน้าวิดีโอที่เชื่อมโยงกับหน้าจอเสมือนนั้นๆ
- อินพุตและเอาต์พุตจะเหมาะสมกับระหว่างเทอร์มินัลแบบฟิสิคัลและหน้าจอเสมือน

หลังจากคีย์การเลือกที่ถูกกำหนดในไฟล์ `dsinfo` มีหน้าจอเสมือนที่กำหนดให้มัน จะไม่มีหน้าจอที่ถูกสร้างขึ้นอีก แต่ละเซสชันของหน้าจอจะสิ้นสุดเมื่อกระบวนการของเซลล์ดั้งเดิมจบลง นี่จะเป็นการทำให้คีย์การเลือกที่เชื่อมโยงว่างลงสำหรับใช้กับหน้าขอเสมือนอื่น ยูทิลิตี **dscreen** จะสิ้นสุดเมื่อไม่มีหน้าจอที่แอ็คทีฟ

คีย์ dscreen Block:

คีย์ Block ถูกใช้เพื่อหยุดเอาต์พุตเหมือนกับคีย์ Ctrl-S เมื่อใช้ IXON โฟลว์คอนโทรล

วัตถุประสงค์ของคีย์เหล่านี้คือยอมให้การตั้งค่าเทอร์มินัลเซสชันบนเครื่องคอมพิวเตอร์สองเครื่องโดยใช้เทอร์มินัลที่มีสองซีเรียลพอร์ต

คีย์ dcreen New:

การกดคีย์ new screen จะสร้างหน้าจอแบบโลจิคัลและกำหนดมันกับหนึ่งในคีย์ที่เลือก

แต่ละหน้าจอใหม่ต้องการ :

- คีย์ที่เลือกที่ถูกกำหนดในไฟล์ dsinfo
- อุปกรณ์ **dscreen** pseudo-terminal
- มีหน่วยความจำเพียงพอสำหรับโครงสร้างต่างๆที่ถูกใช้ในการติดตามหน้าจอ
- กระบวนการที่รัน shell

ถ้าสิ่งเหล่านี้ไม่พร้อม การดำเนินการหน้าจอใหม่จะล้มเหลวพร้อมกับข้อความที่ระบุสาเหตุของความล้มเหลว

คีย์ dscreen End และ Quit:

เมื่อคีย์ End และ Quit ถูกกด ชุดของแอ็คชันจะเกิดขึ้น

การกดคีย์ End ทำให้สิ่งต่อไปนี้เกิกขึ้น:

- ส่งสัญญาณ **SIGHUP** ไปยังเซสชันหน้าจอทั้งหมด
- Clean Up
- ออกจากระบบ ด้วยสถานะ 0

การกดคีย์ Quit จะกระทำแอ็คชันเดียวกันแต่จะออกจากระบบด้วยสถานะ 1

คีย์ dscreen Previous:

การกดคีย์ previous จะสลับเทอร์มินัลไปที่หน้าจอที่ถูกแสดงล่าสุด

หมายเหตุ:

1. ห้ามสลับหน้าจอเมื่อหน้าจอปัจจุบันกำลังถูกเขียน ลำดับของ escape อาจถูกตัดและทำให้เทอร์มินัลอยู่ในสถานะที่ไม่รู้จัก
2. บางเทอร์มินัลสามารถบันทึกตำแหน่งขอเคอร์เซอร์สำหรับแต่ละหน้าจอ แต่ไม่สามารถบันทึกสถานอื่น เช่นโหมด การกลับวิดีโอ และอื่นๆ ถ้าเป็นกรณีนี้ คุณควรหลีกเลี่ยงโหมดเหล่านี้ขณะสลับหน้าจอ

คีย์ dscreen List:

การกดคีย์ list จะแสดงลิสต์ของคีย์และแอ็คชันของมันบนหน้าจอของเทอร์มินัล

เฉพาะคีย์ที่ถูกรู้จักโดยยูทิลีตี **dscreen** ที่จะถูกแสดง เมื่อหน้าจอใหม่ถูกสร้างโดยใช้ยูทิลีตี **dscreen** ข้อความ `Press KEY for help` โดยที่ `KEY` เป็นชื่อของคีย์จะถูกแสดงบนเทอร์มินัล โปรดสังเกตว่าข้อความจะถูกแสดง *เฉพาะ* มีการกำหนดคีย์ list เท่านั้น

การกำหนดหน้าจอแบบไดนามิก

entry ของคำอธิบายเทอร์มินัลในไฟล์ `/usr/lbin/tty/dsinfo` จะมีจำนวนของคีย์การเลือกหน้าจอเหมือนกับที่เทอร์มินัลมีหน้าจอแบบฟิสิคัล ถ้ามีการกำหนดคีย์การเลือกหน้าจอจำนวนมากกว่าจำนวนของหน้าจอแบบฟิสิคัล ยูทิลีตี **dscreen** จะกำหนดหน้าจอแบบฟิสิคัลให้กับหน้าจอเสมือนแบบไดนามิก

เมื่อหน้าจอเสมือนถูกเลือกที่ไม่มีหน้าที่เชื่องโยงกับของหน่วยความจำของหน้าจอ ยูทิลีตี **dscreen** จะกำหนดหน้าจะแบบฟิสิคัลที่ถูกใช้ล่าสุดให้กับหน้าจอเสมือน ขึ้นอยู่กับข้อกำหนดที่ถูกเก็บในไฟล์คำอธิบาย `/usr/lbin/tty/dsinfo` การระบุว่าหน้าจอแบบฟิสิคัลถูกเชื่อมกับหน้าจอเสมือนที่แตกต่างกันอาจสังเกตได้ ตัวอย่างเช่น หน้าจอถูกลบ

ไฟล์ dsinfo

ไฟล์ `dsinfo` เป็นฐานข้อมูลของคำอธิบายของเทอร์มินัลที่ถูกมใช้โดยยูทิลิตีหลายหน้าจอ **dscreen**

ไฟล์ประกอบด้วยข้อมูลต่อไปนี้ :

- คีย์ยูทิลิตี **dscreen** และฟังก์ชันที่มันทำ
- จำนวนของเพจของหน่วยความจำหน้าจอสำหรับเทอร์มินัล
- ลำดับของโค้ดที่ส่งและรับเพื่อใช้คุณลักษณะข้างบน

entry ของชนิดของเทอร์มินัลในไฟล์ dsinfo ดีฟอลต์ประกอบด้วยค่าเทอร์มินัล 3151 ASCII ต่อไปนี้ :

```
# The Cartridge for Expansion (pn: 64F9314) needed for this entry
ibm3151|3151|IBM 3151,
dsks=\E!a^M|Shift-F1|,           # Selects first screen
dsks=\E!b^M|Shift-F2|,           # Selects second screen
dsks=\E!c^M|Shift-F3|,           # Selects third screen
dsks=\E!d^M|Shift-F4|,           # Selects fourth screen
dskc=\E!e^M|Shift-F5|,           # Creates a new screen
dske=\E!f^M|Shift-F6|\E pA\EH\EJ, # Go to screen 1 and end
dskl=\E!g^M|Shift-F7|,           # Lists function keys (help)
dskp=\E!h^M|Shift-F8|,           # Go to previous screen
dskq=\E!i^M|Shift-F9|\E pA\EH\EJ, # Go to screen 1 and quit
dsp=\E pA|\EH\EJ,                # Terminal sequence for screen 1
dsp=\E pB|\EH\EJ,                # Terminal sequence for screen 2
dsp=\E pC|\EH\EJ,                # Terminal sequence for screen 3
dsp=\E pD|\EH\EJ,                # Terminal sequence for screen 4
dst=10,                          # Allow 1 second timeout buffer
```

รูปแบบของ entry สำหรับ dsinfo:

Entry ในไฟล์ dsinfo ประกอบด้วยฟิลด์ที่คั่นด้วยคอมมา

ฟิลด์แรกเป็นลิสต์ของชื่ออื่นสำหรับเทอร์มินัล แต่ละชื่อถูกคั่นด้วยอักขระไพพ์ (|) ข้อความใดๆที่อยู่หน้าอักขระปาวด์ (#) จะถือว่าเป็นหมายเหตุและถูกข้ามโดย **dscreen** ฟิลด์ที่เหลือเป็นสตริงที่อธิบายความสามารถของเทอร์มินัลกับยูทิลิตี **dscreen** ภายในสตริงเหล่านี้โค้ด escape ต่อไปนี้จะถูกรู้จัก :

ตารางที่ 110. ฟิลด์ของไฟล์ dsinfo

Escape Sequence	รายละเอียด
\E,\e	อักขระ escape
\n,\l	อักขระบรรทัดใหม่ (หรือ linefeed)
\r	ปัดแคร่
\t	อักขระแท็บ
\b	อักขระถอยกลับ
\f	อักขระ formfeed
\s	อักขระช่องว่าง
nnn	อักขระพร้อมกับค่า octal *nnn*
^*x*	Ctrl-X สำหรับค่า *x* ที่เหมาะสม

อักขระอื่นใดๆที่อยู่หน้าแบ็กสแลชจะหมายถึงอักขระมันเอง สตริงที่ถูกใส่เป็น *type=string* โดยที่ *type* เป็นชนิดของสตริงดังที่ลิสต์ด้านล่าง และ *string* เป็นค่าของสตริง

มีความจำเป็นที่ entry ของฟิลด์ในไฟล์ dsinfo ต้องถูกคั่นด้วยคอมมา ถ้าคอมมาถูกตัดจากส่วนท้ายของ entry ของไฟล์ dsinfo ไฟล์จะไม่สามารถถูกอ่านได้โดยยูทิลิตี **dscreen** และข้อผิดพลาดจะถูกแสดงบนหน้าจอ

ชนิดของสตริง disinfo:

ชนิดของสตริง disinfo สามารถถูกอ้างถึงที่นี่

ชนิดของสตริงเป็นดังต่อไปนี้ :

ไอเท็ม **คำอธิบาย**

dskx ชนิดของสตริงที่เริ่มต้นด้วย dsk จะอธิบายถึงคีย์ ชนิดต้องยาวสี่ตัวอักษร และตัวอักษรที่สี่ *x* จะระบุแอ็คชันที่ถูกกระทำเมื่อได้รับคีย์ ชนิดของคีย์คือ :

ชนิก	การดำเนินการ
dsks	สลับหน้าจอ
dskb	บล็อกอินพุตและเอาต์พุต
dske	สิ้นสุด **dscreen**
dskq	ออกจาก **dscreen** (สถานะของการออก=1)
dskc	สร้างหน้าจอใหม่
dskp	สลับไปยังหน้าจอก่อนหน้านี้
dskl	ลิสต์คีย์และแอ็คชัน

ชนิดของคีย์อื่น (ซึ่งคือ ชนิดของสตริง dsk*x* ที่ไม่ได้ลงท้ายด้วย in s, b, e, q, p, or l) จะทำให้ไม่มีแอ็คชัน **dscreen** ภายใน แต่จะแสดงในลิสต์ของคีย์ และจะถูกรู้จักและปิดใช้งาน ชนิดของ dskn (n คือ No Operation) ควรถูกใช้เมื่อไม่ต้องการแอ็คชัน **dscreen** ภายใน

สตริงของค่าสำหรับแต่ละคีย์จะมี 3 สตริงย่อย ซึ่งจะถูกแยกด้วยอักขระไพพ์ (|)
หมายเหตุ: ใช้ \ | เพื่อรวมอักขระ | ในหนึ่งในสตริงย่อย

สตริงย่อยแรกเป็นลำดับของอักขระที่เทอร์มินัลส่งเมื่อคีย์ถูกกด สตริงย่อยที่สอง เป็นเลเบลสำหรับคีย์ที่ถูกพิมพ์เมื่อลิสต์ของคีย์ถูกแสดง สตริงย่อยที่สามเป็นลำดับของอักขระที่ **dscreen** ส่งไปยังเทอร์มินัลเมื่อคีย์นี้ถูกกดก่อนที่จะทำแอ็คชันที่คีย์นี้ร้องขอ

dsp ชนิดของสตริงของ dsp จะอธิบายหน้าจอแบบฟิสิคัลในเทอร์มินัล หนึ่งสตริงของ dsp คุณมีอยู่สำหรับแต่ละหน้าจอแบบฟิสิคัลในเทอร์มินัล สตริงของค่าสำหรับแต่ละหน้าจอแบบฟิสิคัลจะมีสองสตริงย่อย ซึ่งจะถูกแยกด้วยอักขระไพพ์ (|)

สตริงย่อยแรกเป็นลำดับของอักขระที่ส่งไปยังเทอร์มินัลเพื่อแสดงและส่งเอาต์พุตไปยังหน้าแบบฟิสิคัลบนเทอร์มินัล

สตริงย่อยที่สองถูกส่งไปยังเทอร์มินัลเมื่อหน้าถูกใช้กับอะไรที่ใหม่ สตริงย่อยที่สองนี้มันถูกตั้งเพื่อลบลำดับของหน้าจอ มันจะถูกส่งภายใต้ 2 เงื่อนไขต่อไปนี้ :

1. เมื่อเซสชันใหม่ของเทอร์มินัลเสมือนถูกสร้าง
2. เมื่อมีเทอร์มินัลเสมือนมากกว่าหน้าจอแบบฟิสิคัล ถ้าเทอร์มินัลเสมือนที่ถูกเลือกต้องการ **dscreen** เพื่อใช้หนึ่งในหน้าจอฟิสิคัลใหม่ มันจะส่งลำดับนี้ไปยังหน้าจอที่ระบุว่าเนื้อหาของหน้าจอไม่ตรงกับเอาต์พุตของเทอร์มินัลเสมือนที่ถูกเชื่อมต่อ

หมายเหตุ: การรันเทอร์มินัลเสมือนมากกว่าหน้าจอแบบฟิสิคัลอาจทำให้เกิดความสับสนและไม่แนะนำ มันสามารถหลีกเลี่ยงได้โดยการไม่กำหนดคีย์การเลือกหน้าจอ (dsks=) มากกว่าหน้าจอแบบฟิสิคัล (dsp=) ใน entry ของ dsinfo

dst A สตริงพร้อมกับชนิดของ dst จะปรับการหมดเวลาอินพุตของ **dscreen** ค่าของสตริงจะเป็นเลขจำนวนเต็มสิบ ค่า timeout เป็นสิบเท่าของวินาที และมีค่าสูงสุดเป็น 255 (ดีฟอลต์=**1** [หรือ 0.1 วินาที])

เมื่อ **dscreen** รู้จักส่วนที่นำหน้าของลำดับของคีย์อินพุต แต่ไม่มีอักขระทั้งหมดของลำดับ มันจะรออักขระเพิ่มเติมที่จะถูกส่งจนกว่ามันจะสามารถรู้จัก ถ้ามีการหมดเวลาก่อนที่จะได้รับอักขระเพิ่มเติม อักขระจะถูกส่งไปยังหน้าจอเสมือน และ **dscreen** จะไม่พิจารณาอักขระเหล่านี้เป็นส่วนหนึ่งของลำดับของคีย์อินพุต

มันอาจจำเป็นต้องเพิ่มค่านี้ ถ้าคีย์ **dscreen** หนึ่งคีย์หรือมากกว่าถูกทริกเกอร์บนจำนวนที่แท้จริงของการกดคีย์(ซึ่งคือการกำหนด Ctrl-Z 1, Ctrl-Z 2, Ctrl-Z 3 เป็นต้น, สำหรับการเลือกหน้าจอ และ Ctrl-Z N สำหรับหน้าจอใหม่ และอื่นๆ)

ตัวอย่างของ dysinfo:

ตัวอย่างต่อไปนี้ dysinfo สำหรับ Wyse-60 พร้อมกับเซสชัน 3 หน้าจอ

```
wy60|wyse60|wyse model 60,
dsks=^A`^M|Shift-F1|,
dsks=^Aa^M|Shift-F2|,
dsks=^Ab^M|Shift-F3|,
dskc=\200|Ctrl-F1|,
dske=\201|Ctrl-F2|\Ew0\E+,
dskl=\202|Ctrl-F3|,
dsp=\Ew0|\E+,
dsp=\Ew1|\E+,
dsp=\Ew2|\E+,
```

พร้อมกับ entry นี้ :

- Shift-F1 ถึง Shift-F3 ถูกใช้สำหรับการเลือกหน้าจอ 1 ถึง 3
- Ctrl-F1 สร้างหน้าจอใหม่
- Ctrl-F2 ส่ง: Esc w 0 Esc + ไปที่หน้าจอ (การสลับไปที่หน้าต่าง 0 และลบหน้าจอ) และ จากนั้นสิ้นสุด **dscreen**
- Ctrl-F3 ลิสต์คีย์และฟังก์ชันของมัน

แต่ละครั้งที่หน้าจอแลลฟิสิคัลถูกใช้สำหรับหน้าจอใหม่ ลำดับของ Esc + จะถูกส่งไปยังเทอร์มินัล ซึ่งจะลบหน้าจอ

ตัวอย่างต่อไปนี้สำหรับ Wyse-60 พร้อมกับเซสชัน 3 หน้าจอ แต่หนึ่งในหน้าจออยู่บนคอมพิวเตอร์เครื่องที่สองที่สื่อสารผ่านซีเรียลพอร์ตที่สองบนเทอร์มินัล :

```
wy60-1|wyse60-1|wyse model 60 - first serial port
dsks=^A`^M|Shift-F1|,
dsks=^Aa^M|Shift-F2|,
dsks=^Ab^M|Shift-F3|\Ed#^Ab\r^T\Ee9,
dskc=\200|Ctrl-F1|,
dske=\201|Ctrl-F2|\Ed#\201^T\Ew0\E+,
dskl=\202|Ctrl-F3|,
dsp=\Ew0|\E+,dsp=\Ew1|\E+,
wy60-2|wyse60-2|wyse model 60 - second serial port
dsks=^A`^M|Shift-F1|\Ed#^A`\r^T\Ee8,
dsks=^Aa^M|Shift-F2|\Ed#^Aa\r^T\Ee8,
dsks=^Ab^M|Shift-F3|,
dskc=\200|Ctrl-F1|,
dske=\201|Ctrl-F2|\Ed#\201^T\Ew0\E+,
dskl=\202|Ctrl-F3|,
dsp=\Ew2|\E+,
```

dscreen ต้องถูกรันบนทั้งสองคอมพิวเตอร์ ด้วยชนิดของเทอร์มินิล wy60-1 บนคอมพิวเตอร์เครื่องแรกและชนิดของเทอร์มินัล wy60-2 บนคอมพิวเตอร์เครื่องที่สอง (โดยใช้อ็อพชัน **-t** กับ **dscreen**) entry wy60-1 จะถูกตรวจสอบเป็นลำดับแรก

entry ของคีย์สองตัวแรกไม่ถูกเปลี่ยนจาก entry wy60 ดั้งเดิม อย่างไรก็ตาม คีย์ที่สาม เป็นชนิด dskb ซึ่งหมายความว่าบล็อกทั้งอินพุตและเอาต์พุต เมื่อคีย์นี้ถูกกด ลำดับ:

```
Esc d # Ctrl-A b CR Ctrl-T Esc e 9
```

จะถูกส่งไปยังเทอร์มินัล หลังจากเอาต์พุตนี้ถูกบล็อกและ **dscreen** ยังคงสแกนอินพุตสำหรับลำดับของคีย์และจะทิ้งอินพุตอื่นทั้งหมด

ลำดับ Esc d # จะทำให้เทอร์มินัลอยู่ในโหมดการพิมพ์แบบ transparent ซึ่งจะ echo อักขระทั้งหมดจนถึง Ctrl-T ออกไปยังซีเรียลพอร์ตอื่น

อักขระ Ctrl-A b CR จะถูกส่งออกไปยังซีเรียลพอร์ตอื่น จะบอกกระบวนการ **dscreen** บนคอมพิวเตอร์อื่นว่ามันควรเปิดใช้หน้าต่างที่เชื่อมโยงกับคีย์ Shift-F3

ลำดับของคีย์ Ctrl-T จะออกจากโหมดการพิมพ์แบบ transparent ลำดับของคีย์ Esc 9 จะทำให้เทอร์มินัลสลับไปยังซีเรียลพอร์ต AUX อื่นสำหรับการสื่อสารข้อมูล

ในตอนนี้ คอมพิวเตอร์จะรับช่วง ส่ง Esc w 2 เพื่อสลับไปยังหน้าจอฟิสิคัลที่สาม และจากนั้นเรียกคืนการสื่อสารปกติ

entry wy60-2 จะเป็นไปตามแบบแผนทั่วไปเดียวกันสำหรับคีย์ Shift-F1 และ Shift-F2:

- สลับไปยังโหมดการพิมพ์แบบ transparent
- ส่งฟังก์ชันคีย์สตริงไปยังคอมพิวเตอร์อื่น
- ปิดการพิมพ์แบบ transparent
- สลับไปยังซีเรียลพอร์ตอื่น

ส่งคีย์การสิ้นสุด Ctrl-F2 ทำงานเหมือนกันสำหรับคอมพิวเตอร์ทั้งสอง มันจะส่งลำดับของคีย์สิ้นสุดไปยังคอมพิวเตอร์อื่นผ่านกลไกการพิมพ์แบบ transparent สลับเทอร์มินัลไปยังหน้าต่าง 0 ลบหน้าจอ จากนั้นออกจากระบบ

สภาวะแวดล้อม generic data link control

Generic data link control (GDLC) เป็นคำจำกัดความของอินเตอร์เฟสทั่วไป ที่จัดเตรียมให้ชุดคำสั่งทั่วไปให้กับผู้ใช้แอ็พพลิเคชันและเคอร์เนลเพื่อควบคุมตัวจัดการอุปกรณ์ data link control (DLC) ภายในระบบปฏิบัติการ

สำหรับการระบุปัญหา ดูที่ การระบุปัญหาของ GDLC ใน *หลักการเขียนโปรแกรมการสื่อสาร*

Generic data link control (GDLC) เป็นคำจำกัดความของอินเตอร์เฟสทั่วไป ที่จัดเตรียมให้ชุดคำสั่งทั่วไปให้กับผู้ใช้แอ็พพลิเคชันและเคอร์เนลเพื่อควบคุมตัวจัดการอุปกรณ์ DLC ภายในระบบปฏิบัติการ

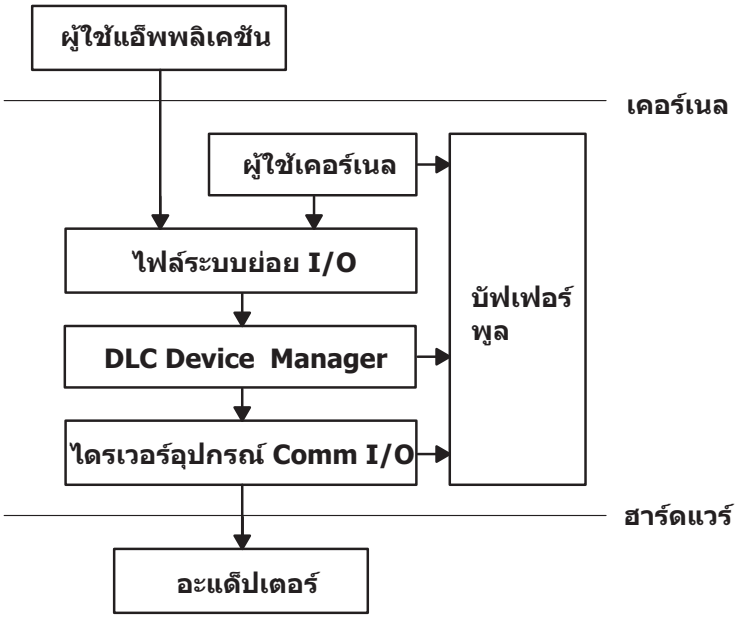
ข้อกำหนดที่ระบุสำหรับ GDLC อินเตอร์เฟสสำหรับคำจำกัดความ entry point ฟังก์ชั่นที่ถูกจัดเตรียม และโครงสร้างข้อมูลของตัวจัดการอุปกรณ์ DLC ทั้งหมด DLCs ที่เป็นไปตาม GDLC อินเตอร์เฟสรวมถึง:

- 8023 (IEEE 802.3 สำหรับ Ethernet)
- ETHER (Ethernet มาตรฐาน)
- SDLC (Synchronous Data Link Control)
- TOKEN (Token-Ring)
- FDDI (Fiber Distributed Data Interface)

ตัวจัดการอุปกรณ์ DLC ทำงานเป็นโปรโตคอลและฟังก์ชันระดับที่สูงกว่าขอบเขตของไดรเวอร์อุปกรณ์ของเคอร์เนล อย่างไรก็ตาม ตัวจัดการที่อยู่ภายในเคอร์เนลสำหรับประสิทธิภาพสูงสุดและใช้ไดรเวอร์อุปกรณ์ของเคอร์เนลสำหรับการร้องขอ I/O ของมันไปยังอะแด็ปเตอร์ ผู้ใช้ DLC จะอยู่เหนือหรือภายในเคอร์เนล

Synchronous data link control (SDLC) และ IEEE 802.2 Data Link Control เป็นตัวอย่างของตัวจัดการอุปกรณ์ DLC ตัวจัดการอุปกรณ์ DLC แต่ละตัวจะทำงานกับไดรเวอร์ของอุปกรณ์ที่ระบุ หรือชุดของไดรเวอร์ของอุปกรณ์ ตัวอย่างเช่น SDLC ทำงานกับไดรเวอร์ของอุปกรณ์แบบหลายโปรโตคอลสำหรับผลิตภัณฑ์ของระบบและอะแด็ปเตอร์ที่เกี่ยวข้องกับมัน

โครงสร้างพื้นฐานของสภาวะแวดล้อม DLC จะแสดงในรูป "สภาวะแวดล้อมตัวจัดการอุปกรณ์ DLC" ผู้ใช้ภายในเคอร์เนลมีสิทธิเข้าถึงบัฟเฟอร์ของหน่วยความจำการสื่อสาร (mbufs) และเรียกใช้การเพิ่ม entry point ผ่านเซอร์วิสของเคอร์เนล **fp** ผู้ใช้ที่อยู่เหนือเคอร์เนลเข้าถึงไดรเวอร์ของอุปกรณ์แบบ interface-to-kernel มาตรฐาน และระบบไฟล์จะเรียกใช้ **dd** entry points การถ่ายโอนข้อมูลต้องการการย้ายข้อมูลระหว่างผู้ใช้และพื้นที่ของเคอร์เนล



รูปที่ 44. สภาวะแวดล้อมตัวจัดการอุปกรณ์ DLC

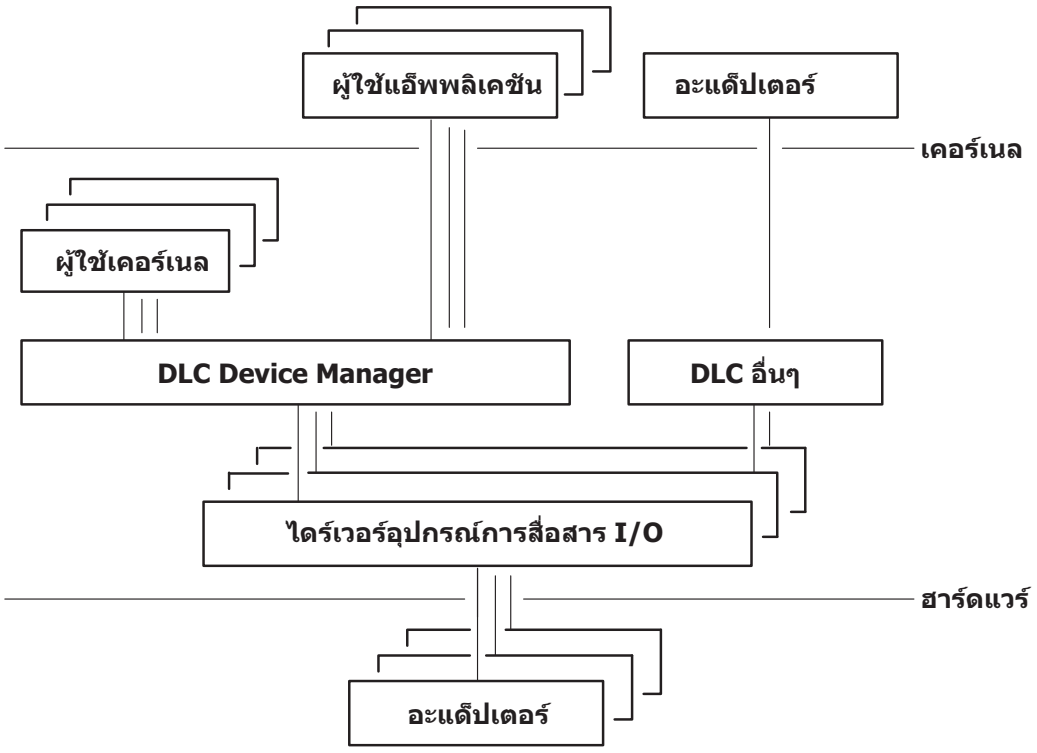
นี่จะแสดงลิงก์ระหว่างผู้ใช้แอ็พพลิเคชันและอะแด็ปเตอร์ (ระดับฮาร์ดแวร์) พื้นที่ที่อยู่ระหว่างกลางคือผู้ใช้เคอร์เนล File I/O Subsystem ตัวจัดการอุปกรณ์ DLC ไดรเวอร์อุปกรณ์ Comm I/O และบัฟเฟอร์พูล entity "ที่อยู่ระหว่างกลาง" เหล่านี้จะอยู่ที่ระดับของเคอร์เนล

ส่วนประกอบของสภาวะแวดล้อมตัวจัดการอุปกรณ์ DLC คือ :

ไอเท็ม	คำอธิบาย
ผู้ใช้แอ็พพลิเคชัน	ที่อยู่เหนือเคอร์เนลเป็นแอ็พพลิเคชันหรือวิธีการเข้าถึง
ผู้ใช้เคอร์เนล	ที่อยู่ภายในเคอร์เนลเป็นกระบวนการของเคอร์เนลหรือตัวจัดการอุปกรณ์
File I/O Subsystem	แปลงรูทีนย่อย file-descriptor และ file-pointer เป็นการเข้าถึงตัวชี้ไฟล์ของตารางสวิตช์
บัฟเฟอร์พูล	จัดเตรียมเซอร์วิสของบัฟเฟอร์ข้อมูลสำหรับระบบย่อยของการสื่อสาร
ไดรเวอร์อุปกรณ์ Comm I/O	ควบคุมฮาร์ดแวร์อะแด็ปเตอร์ I/O และการลงทะเบียน direct memory access (DMA) และส่งแพ็กเกตที่ได้รับไปยังหลาย DLC
อะแด็ปเตอร์	เชื่อมต่อกับสื่อของการสื่อสาร

ตัวจัดการอุปกรณ์จะถูกเขียนโดยสอดคล้องกับข้อกำหนดของ GDLC ที่รันบนคอนฟิกูเรชันฮาร์ดแวร์ของระบบคอนฟิกูเรชันปฏิบัติการที่ประกอบด้วยไดรเวอร์ของอุปกรณ์การสื่อสาและอะแด็ปเตอร์เป้าหมายของมัน ตัวจัดการอุปกรณ์แต่ละตัวจะสนับสนุนหลายผู้ใช้ด้านบนรวมถึงหลายไดรเวอร์อุปกรณ์และอะแด็ปเตอร์ด้านล่าง โดยทั่วไป ผู้ใช้จะทำงานพร้อมกันบนอะแด็ปเตอร์เดียว หรือแต่ละผู้ใช้ทำงานบนหลายอะแด็ปเตอร์ ตัวจัดการอุปกรณ์ DLC จะแตกต่างกันโดยขึ้นอยู่กับเงื่อนไขของโปรโตคอลของมัน

รูปที่ 45แสดงคอนฟิกูเรชันแบบหลายผู้ใช้:



รูปที่ 45. คอนฟิกูเรชันแบบหลายผู้ใช้และหลายอะแด็ปเตอร์

นี่จะแสดงอีกมุมมองของระดับของเคอร์เนลระหว่างผู้ใช้แอ็พพลิเคชันและอะแด็ปเตอร์ มันจะแสดงหลาย entity ที่แทนหลายผู้ใช้

เงื่อนไขของ GDLC

GDLC อินเตอร์เฟสต้องมีเงื่อนไขต่อไปนี้

- มีความยืดหยุ่นและสามารถเข้าถึงได้กับทั้งแอ็พพลิเคชันและผู้ใช้เคอร์นัล

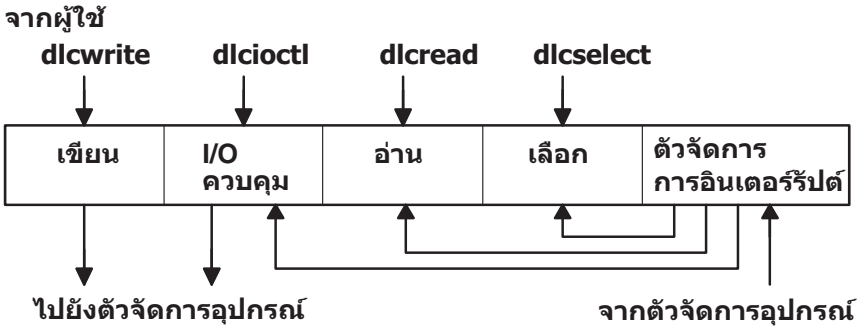
- มีความสามารถกำหรับหลายผู้ใช้และหลายอะแด็ปเตอร์ ยอมให้โปรโตคอลใช้ประโยชน์ของหลายเซสชันและพอร์ต
- สนับสนุนทั้งเซอร์วิสแบบ connection-oriented และ connectionless เมื่อเป็นไปได้
- ยอมให้มีการถ่ายโอนข้อมูลแบบ transparent สำหรับความต้องการพิเศษที่นอกเหนือจากขอบเขตของตัวจัดการอุปกรณ์ DLC ที่ใช้

อินเตอร์เฟส GDLC

แต่ละตัวจัดการอุปกรณ์ DLC เป็น entry /dev มาตรฐานที่ทำงานในเคอร์เนล เป็นตัวจัดการอุปกรณ์แบบ multiplex สำหรับโปรโตคอลที่ระบุ

สำหรับอะแด็ปเตอร์ที่ไม่ถูกใช้โดย DLC แต่ละรูทีนย่อย **open** กับตัวจัดการอุปกรณ์ DLC จะสร้างกระบวนการเคอร์เนล รูทีนย่อย **open** ยังถูกสร้างไปยังตัวจัดการอุปกรณ์อะแด็ปเตอร์เป้าหมาย ถ้าต้องการ สร้างรูทีนย่อย **open** เพิ่มเติมสำหรับหลาย DLC อะแด็ปเตอร์พอร์ตของโปรโตคอลเดียวกัน รูทีนย่อย **open** ใดๆที่มีเป้าหมายไปยังพอร์ตเดียวกันจะไม่สร้างกระบวนการเคอร์เนลเพิ่มเติม แต่จะลิงก์รูทีนย่อย **open** กับกระบวนการที่มีอยู่ จะมีเพียงหนึ่งกระบวนการเคอร์เนลสำหรับแต่ละพอร์ตที่ถูกใช้

โครงสร้างภายในของตัวจัดการอุปกรณ์ DLC จะมีโครงสร้างพื้นฐานเดียวกับตัวจัดการอุปกรณ์เคอร์เนล ยกเว้นกระบวนการเคอร์เนลจะแทนที่ตัวจัดการอินเตอร์รัปต์ในเหตุการณอะซิงโครนัส ฟังก์ชันการอ่าน เขียน ควบคุม I/O และเลือกบล็อก ดังแสดงในรูป "ตัวจัดการอุปกรณ์เคอร์เนลมาตรฐาน"



รูปที่ 46. ตัวจัดการอุปกรณ์เคอร์เนลมาตรฐาน

นี่จะแสดงโครงสร้างภายในของตัวจัดการอุปกรณ์ DLC โครงสร้างนี้ประกอบด้วยการเขียน ควบคุม I/O การอ่าน การเลือก และตัวจัดการอินเตอร์รัปต์ ตัวจัดการอุปกรณ์จะรับข้อมูลจากผู้ใช้ที่มันถูกผ่านไปยังพื้นที่ต่างๆก่อนที่จะถูกผ่านไปยังตัวจัดการอุปกรณ์

GDLC data link controls

คุณสามารถติดตั้ง DLCs แยกต่างหากหรือในกลุ่ม ตัวจัดการอุปกรณ์ DLC จะถูกเพิ่มเข้ากับเคอร์นัลโดยอัตโนมัติและตั้งสถานะเป็น "Available" สำหรับแต่ละชนิดของ DLC ที่ถูกติดตั้ง

การติดตั้งสามารถถูกตรวจสอบโดยใช้คำสั่ง **lslpp** ดังต่อไปนี้ :

```
lslpp -h dlctype
```

โดยที่ *dlctype* เป็นหนึ่งในต่อไปนี้ :

ไอเท็ม	คำอธิบาย
bos.dlc.8023	IEEE Ethernet (802.3) Data Link Control
bos.dlc.ether	Standard Ethernet Data Link Control
bos.dlc.fddi	FDDI Data Link Control
bos.dlc.sdlc	SDLC Data Link Control
bos.dlc.token	Token-Ring Data Link Control

ข้อมูลเกี่ยวกับ DLC ที่ติดตั้งสามารถแสดงผ่าน System Management Interface Tool (SMIT) หรือบรรทัดรับคำสั่ง บนระบบหรือพอร์ตการสื่อสารที่ใช้งานหนัก มันอาจจำเป็นที่จะเปลี่ยนแอ็ตทริบิวต์ DLC เพื่อปรับแต่งประสิทธิภาพของ DLC อย่างละเอียด ถ้าประสิทธิภาพการรับช้าลง ล็อกข้อผิดพลาดของระบบจะระบุว่าคิวของ ring ระหว่าง DLC และตัวจัดการอุปกรณ์เต็ม เพิ่มความลึกของคิวของ DLC สำหรับข้อมูลที่เข้ามา ท้ายสุด แนะนำให้ลบ DLC ที่ถูกติดตั้งจากเคอร์นัลเมื่อไม่ต้องการใช้เป็นเวลานาน การลบนี้จะไม่ลบ DLC จากระบบ แต่เป็นการทำให้รีซอร์สของเคอร์นัลว่างสำหรับงานอื่นจนกว่า DLC จะถูกต้องการอีกครั้ง วิธีการสำหรับงานเหล่านี้จะอยู่ใน “การจัดการ DLC ไดรเวอร์อุปกรณ์” ในหน้า 705

การทำงานของ GDLC อินเตอร์เฟส ioctl entry point

generic data link control (GDLC) อินเตอร์เฟสสนับสนุนการทำงานของรูทีนย่อย **ioctl** เหล่านี้

ไอเท็ม	คำอธิบาย
DLC_ENABLE_SAP	เปิดใช้งาน service access point (SAP)
DLC_DISABLE_SAP	ปิดใช้งาน SAP
DLC_START_LS	สตาร์ท link station บน SAP นั้นๆเป็นผู้เรียกหรือผู้ฟัง
DLC_HALT_LS	หยุด link station.
DLC_TRACE	ติดตามกิจกรรมของ link station สำหรับกิจกรรมแบบสั้นหรือยาว
DLC_CONTACT	ติดต่อรีโมตสเตชันสำหรับโลคัล link station นั้น
DLC_TEST	ทดสอบลิงก์ไปยังรีโมตสำหรับโลคัล link station นั้นๆ
DLC_ALTER	สลับพารามิเตอร์การตั้งค่าของ link station
DLC_QUERY_SAP	เคียวรีสถิติของ SAP นั้นๆ
DLC_QUERY_LS	เคียวรีสถิติของลิงก์สเตชันนั้นๆ
DLC_ENTER_LBUSY	เข้าสู่โหมด local-busy บน link station นั้นๆ
DLC_EXIT_LBUSY	ออกจากโหมด local-busy บน link station นั้นๆ
DLC_ENTER_SHOLD	เข้าสู่โหมด short-hold บน link station นั้นๆ
DLC_EXIT_SHOLD	ออกจากโหมด short-hold บน link station นั้นๆ
DLC_GET_EXCEP	ส่งการแจ้งเตือนข้อยกเว้นอะซิงโครนัสไปยังผู้ใช้แอ็พพลิเคชัน **หมายเหตุ:** การทำงานของรูทีนย่อย **ioctl** นี้ไม่ได้ถูกใช้โดยผู้ใช้เคอร์เนล เนื่องจากเงื่อนไขของข้อยกเว้นจะถูกผ่านไปยังผู้ใช้เคอร์เนลผ่านตัวจัดการข้อยกเว้น
DLC_ADD_GRP	เพิ่มกลุ่มหรือแอดเดรสที่ได้รับ multicast เข้ากับพอร์ต
DLC_DEL_GRP	ลบกลุ่มหรือแอดเดรสที่ได้รับ multicast จากพอร์ต
DLC_ADD_FUNC_ADDR	เพิ่มกลุ่มหรือแอดเดรสการทำงานที่ได้รับ multicast เข้ากับพอร์ต
DLC_DEL_FUNC_ADDR	ลบกลุ่มหรือแอดเดรสการทำงานที่ได้รับ multicast จากพอร์ต
IOCINFO	ส่งโครงสร้างที่อธิบายถึงตัวจัดการอุปกรณ์ GDLC ดูที่รูปแบบของไฟล์ `/usr/include/sys/devinfo.h` สำหรับข้อมูลเพิ่มเติม

GDLC service access point

service access point (SAP) จะระบุเซอร์วิสของผู้ใช้ที่ส่งและรับคลาสของข้อมูลที่ระบุ

ทำให้คลาสของข้อมูลที่ต่างกันถูกส่งแยกกันไปยังตัวจัดการเซอร์วิสที่เกี่ยวข้องของมัน DLC เหล่านั้นที่สนับสนุน หลาย SAP แบบใช้พร้อมกันมีแอดเดรสที่รู้จักว่าเป็น Destination SAP และ Source SAP ถูกฝังอยู่ในส่วนหัวของแพ็กเก็ตของมัน DLC เฉพาะที่สามารถสนับสนุน SAP เดียวไม่ต้องการหรือไม่ใช้ SAP แอดเดรส แต่ยังใช้แนวคิดของการเปิดใช้งาน SAP เดียว โดยทั่วไปจะมี SAP เดียวที่ถูกเปิดใช้งานสำหรับแต่ละผู้ใช้ DLC บนแต่ละพอร์ต

ค่า SAP แอดเดรสส่วนใหญ่ถูกกำหนดโดย IEEE standardized network-management entities หรือค่าที่ผู้ใช้กำหนดดังที่ถูกระบุใน *Token-Ring Network Architecture Reference* SAP แอดเดรสทั่วไปบางส่วนคือ :

ไอเท็ม	คำอธิบาย
Null SAP (0x00)	ให้ความสามารถบางอย่างในการตอบสนองต่อโหนดแบบรีโมต แม้ว่าเมื่อไม่ได้เปิดใช้งาน SAP SAP นี้สนับสนุนเฉพาะเซอร์วิสแบบ connectionless และตอบสนองเฉพาะกับ exchange identification (XID) และ TEST Link Protocol Data Units (LPDUs)
SNA Path Control (0x04)	แสดงแต่ละ SAP แอดเดรสแบบดีฟอลต์ที่ถูกใช้โดยโหนด Systems Network Architecture (SNA)
PC Network NETBIOS (0xF0)	ถูกใช้สำหรับการสื่อสาร DLC ทั้งหมดที่ถูกใช้โดย Network Basic Input/Output System (NetBIOS) อีมูเลชัน
Discovery SAP (0xFC)	ถูกใช้โดยเซอร์วิสการค้นหาชื่อของ local area network (LAN)
Global SAP (0xFF)	ระบุ SAP ที่แอ็คทีฟทั้งหมด

GDLC link station

link station (LS) จะระบุการเชื่อมต่อระหว่างสองโหนดสำหรับคู่ของ SAP นั้นๆ

การเชื่อมต่อนี้สามารถดำเนินการแบบ connectionless service (datagram) หรือ connection-oriented service (การถ่ายโอนข้อมูลแบบตามลำดับแบบเต็มที่พร้อมกับการแก้ไขข้อผิดพลาด) โดยทั่วไป จัมีเพียง LS เดียวที่ถูกสตาร์ทสำหรับแต่ละการเชื่อมต่อแบบรีโมต

โหมด GDLC Local-Busy

เมื่อ LS ดำเนินการในโหมด connection-oriented มันต้องหยุดการส่งแพ็กเกจข้อมูลของเครื่องแบบรีโมต เนื่องจากสาเหตุเช่น รีซอร์สไม่เพียงพอ จากนั้นการแจ้งเตือนสามารถถูกส่งไปยังเครื่องรีโมตเพื่อทำให้เครื่องโลคัลเข้าสู่โหมด local-busy

เมื่อทรัพยสกรพร้อมใช้งานอีกครั้ง เครื่องโลคัลจะแจ้งเครื่องรีโมตว่ามันไม่ยุ่งแล้วและข้อมูลจะไหลได้อีกครั้ง แฑาะข้อมูลที่ถูกจัดลำดับที่จะถูกหยุดโดยโหมด local-busy แพ็กเกจชนิดอื่นจะไม่ได้รับผลนี้

โหมด GDLC Short-Hold

คุณสามารถใช้การทำงานในโหมด short-hold เมื่อดำเนินการบนบางเน็ตเวิร์กข้อมูล

โหมด Short-hold มีประโยชน์กับเน็ตเวิร์กข้อมูลที่มีคุณลักษณะต่อไปนี้ :

- มีเวลา call-setup ที่สั้น
- โครงสร้างแบบอัตราภาษีที่ระบุค่าใช้จ่ายที่น้อยกว่าสำหรับการเริ่มการเรียกเปรียบเทียบกับการคิดจากเวลาที่เชื่อมต่อ

ระหว่างโหมด short-hold การเชื่อมต่อระหว่าง 2 สเตชันจะถูกคงไว้เฉพาะเมื่อมีข้อมูลสำหรับถ่ายโอนระหว่าง 2 สเตชัน เมื่อไม่มีข้อมูลที่จะส่ง การเชื่อมต่อจะถูกลบหลังจากช่วงเวลา timeout ที่ระบุ และจะเริ่มใหม่เมื่อมีข้อมูลที่จะถ่ายโอน

การทดสอบและการติดตาม GDLC ลิงก์

เพื่อทดสอบการเชื่อมต่อระหว่าง 2 สเตชัน โดยให้ LS ส่งแพ็กเก็ตทดสอบจากโลคัลสเตชัน แพ็กเก็ตนี้จะถูกส่งกลับมาจากรีโมตสเตชันถ้าการเชื่อมต่อทำงานอย่างถูกต้อง

บาง data links ถูกจำกัดในการสนับสนุนฟังก์ชันนี้ของมันเนื่องจากเงื่อนไขของโปรโตคอล ตัวอย่างเช่น SDLC จะสร้างแพ็ตเก็ตการทดสอบเฉพาะจากโฮสต์หรือสเตชันลำดับแรกเท่านั้น อย่างไรก็ตาม โปรโตคอลอื่นส่วนใหญ่ยอมให้แพ็ตเก็ตการทดสอบถูกเริ่มจากสเตชันใดก็ได้

เพื่อติดตามลิงก์ สายข้อมูล และเหตุการณ์พิเศษ (เช่นการเปิดใช้สเตชัน เทอรีมินัล และการหมดเวลา) รับแชนแนลการติดตามทั่วไปและให้ LS เขียนล็อกการติดตามของมันไปยังเครื่องมือการติดตามทั่วไปสำหรับแต่ละ LS ฟังก์ชันนี้ช่วยระบุสาเหตุของปัญหาการสื่อสารของการเชื่อมต่อนั้นๆ ทั้ง entry การติดตามแบบสั้นและยาวที่ได้รับการสนับสนุน

สถิติ GDLC

ทั้งสถิติ SAP และ LS สามารถถูกเคียวรีโดยผู้ใช้ GDLC

สถิติสำหรับ SAP ประกอบด้วยสถานะปัจจุบันของ SAP และข้อมูลเกี่ยวกับตัวจัดการอุปกรณ์ สถิติ LS ประกอบด้วยสถานะปัจจุบันของสเตชันและความน่าเชื่อถือ ความพร้อมใช้งาน ตัวนับความสามารถให้บริการที่มอนิเตอร์กิจกรรมของสเตชันจากเวลาที่มันถูกสตาร์ท

เซอร์วิสพิเศษของเคอร์เนลของ GDLC

Generic data link control (GDLC) มีเซอร์วิสพิเศษสำหรับผู้ใช้เคอร์เนล

อย่างไรก็ตาม สภาวะแวดล้อมที่เชื่อถือได้ ต้องมีอยู่ภายในเคอร์เนล แทนที่ตัวจัดการอุปกรณ์ DLC จะคัดลอกข้อมูลเหตุการณ์อะซิงโครนัสไปยังพื้นที่ของผู้ใช้ ผู้ใช้เคอร์เนลต้องระบุตัวชี้ฟังก์ชันไปยังรูทีนพิเศษที่เรียกว่าตัวจัดการฟังก์ชัน ตัวจัดการฟังก์ชันถูกเรียกใช้โดย DLC เมื่อเวลาการประมวลผล นี่ทำให้ได้ประสิทธิสูงสุดระหว่างผู้ใช้เคอร์เนลและเลเยอร์ของ DLC แต่ละผู้ใช้เคอร์เนลต้องจำกัดจำนวนของตัวจัดการฟังก์ชันเพื่อให้ความยาวของพาธสั้นที่สุด และใช้การสื่อสาร memory buffer (mbuf) scheme

ตัวจัดการฟังก์ชันต้องไม่ถูกเรียกใช้โดย entry อื่นของ DLC โดยตรง เนื่องจากการเรียกใช้โดยตรงจะถูกทำภายใต้การล็อกทำให้เกิด fatal sleep ข้อยกเว้นเดียวสำหรับกฎนี้คือผู้ใช้เคอร์เนลอาจเรียกใช้ **dlcwritex** entry point ระหว่างเซอร์วิสของมันของ 4 ฟังก์ชัน receive data ใดๆ การเรียกใช้ **dlcwritex** entry point จะทำให้การตอบสนองในทันทีถูกสร้างโดยไม่สลับงานในทันที ต้องการตรรกะแบบพิเศษภายในตัวจัดการอุปกรณ์ DLC เพื่อตรวจสอบการระบุกระบวนการของผู้ใช้ที่กำลังเรียกใช้การเขียน ถ้ามันเป็นกระบวนการ DLC process และความสามารถในการเข้าคิวภายในของ DLC การเขียนจะถูกส่งกลับพร้อมกับคืนโค้ดที่ไม่ดี (คืนค่า **EAGAIN**) แทนที่จะทำให้กระบวนการที่เรียก (DLC) เป็น sleep ดังนั้นมันจะขึ้นอยู่กับรูทีนของผู้ใช้ที่เรียกที่จะให้การแจ้งเตือนพิเศษกับ DLC จากฟังก์ชัน receive data ของมันเพื่อให้แน่ใจว่าบัฟเฟอร์ที่รับถูกลองใหม่ภายหลัง

ตัวจัดการฟังก์ชันที่ผู้ใช้กำหนดคือ :

ไอเท็ม	คำอธิบาย
Datagram Data Received Routine	ถูกเรียกใช้แต่ละครั้งที่แพ็กเก็ต datagram ถูกได้รับสำหรับผู้ใช้เคอร์เนล
Exception Condition Routine	ถูกเรียกใช้แต่ละครั้งที่เหตุการณ์อะซิงโครนัสเกิดขึ้นที่ต้องแจ้งผู้ใช้เคอร์เนล เช่น SAP Closed หรือ Station Contacted
I-Frame Data Received Routine	ถูกเรียกใช้แต่ละครั้งที่ลำดับของแพ็ตเก็ตข้อมูลปกติถูกได้รับสำหรับผู้ใช้เคอร์เนล
Network Data Received Routine	ถูกเรียกใช้แต่ละครั้งที่ข้อมูลที่ระบุสำหรับเน็ตเวิร์กถูกได้รับสำหรับผู้ใช้เคอร์เนล
XID Data Received Routine	ถูกเรียกใช้แต่ละครั้งที่แพ็กเก็ต exchange identification (XID) ถูกได้รับสำหรับผู้ใช้เคอร์เนล

dlcread และ **dlcselect** entry point สำหรับ DLC ไม่ถูกเรียกโดยผู้ใช้เคอร์เนลเนื่องจาก entry การทำงานแบบอะซิงโครนัสถูกเรียกโดยตรงจากตัวจัดการอุปกรณ์ DLC โดยทั่วไป การเข้าคิวของเหตุการณ์เหล่านี้ต้องเกิดขึ้นในตัวจัดการฟังก์ชันของผู้ใช้ อย่างไรก็ตาม ผู้ใช้เคอร์เนลไม่สามารถจัดการแพ็กเก็ตที่ได้รับนั้นๆ ตัวจัดการอุปกรณ์ DLC อาจเก็บบัฟเฟอร์ที่ได้รับครั้งล่าสุดและเข้าสู่หนึ่งในสองโหมด user-busy :

User-Terminated Busy Mode (I-frame เท่านั้น)

ถ้าผู้ใช้เคอร์เนลไม่สามารถจัดการ I-frame ที่ได้รับ (เนื่องจากปัญหาเช่น queue blockage) จะได้รับโค้ด DLC_FUNC_BUSY กลับมา และ DLC จะเก็บตัวชี้บัฟเฟอร์และเข้าสู่โมด local-busy เพื่อหยุดการส่งข้อมูลของรีโมตสเตชัน ผู้ใช้เคอร์เนลต้องเรียกใช้ฟังก์ชัน Exit Local Busy เพื่อรีเซ็ตโหมด local-busy และเริ่มการรับ I-frames อีกครั้ง เฉพาะลำดับของ I-frames ที่ปกติเท่านั้นที่สามารถถูกหยุด XID datagram และข้อมูลเน็ตเวิร์กจะไม่ได้รับผลโดยโหมด local-busy

Timer-Terminated Busy Mode (ทุกชนิดของ frame)

ถ้าผู้ใช้เคอร์เนลไม่สามารถจัดการกับแพ็กเก็ตที่ได้รับ และต้องการให้ DLC เก็บบัฟเฟอร์ที่ได้รับเป็นระยะเวลาสั้นๆ และจากนั้นเรียกใช้ฟังก์ชั้น user receive อีกครั้ง โค้ด DLC_FUNC_RETRY จะถูกส่งกลับไปยัง DLC ถ้าแพ็กเก็ตที่ได้รับเป็นลำดับของ I-frame สเตชันจะเข้าสู่ใหม่ local-busy ในเวลานั้น ในทุกกรณี การจับเวลาจะเริ่มต้น เมื่อหมดเวลา entry ของฟังก์ชัน receive data จะถูกเรียกอีกครั้ง

การจัดการ DLC ไดรเวอร์อุปกรณ์

DLC ต้องถูกเพิ่มเข้ากับระบบก่อนที่จะใช้

DLC ที่ถูกติดตั้งแต่ละตัวจะถูกเพิ่มโดยอัตโนมัติหลังจากการติดตั้งและเมื่อแต่ละระบบถูกรีสตาร์ท (ดูที่ "GDLC data link controls" ในหน้า 701) ถ้า DLC ถูกลบโดยไม่มีการรีสตาร์ท มันสามารถถูกเพิ่มใหม่ได้

ตารางที่ 111. งานสำหรับการจัดการ DLC ไดรเวอร์อุปกรณ์

งาน	วิธีลัด SMIT	คำสั่งหรือไฟล์
เพิ่ม DLC ที่ถูกติดตั้ง	เลือกหนึ่ง (โดยชื่อของไดรเวอร์อุปกรณ์): `smit cmddlc_sdlc` `smit cmddlc_token` `smit cmddlc_qllc` `smit cmddlc_ether`[1] `smit cmddlc_fddi` จากนั้นเลือก **Add**	**mkdev**[2]
เปลี่ยนแอ็ตทริบิวต์ DLC [3,4]	เลือกหนึ่ง (โดยชื่อไดรเวอร์อุปกรณ์): `smit cmddlc_sdlc_ls` `smit cmddlc_token_ls` `smit cmddlc_qllc_ls` `smit cmddlc_ether_ls`[1] `smit cmddlc_fddi_ls`	**chdev**[2]
สตาร์ทการมอนิเตอร์การติดตาม DLC Local Area Network [5]	`smit trace`	**trace -j** *nnn* โดยที่ค่า *nnn* เป็น hook ID ที่จะถูกติดตาม
หยุดการทำงานการมอนิเตอร์การติดตาม DLC Local Area Network	`smit trcstop`	**trcstop**[2]
สร้างรายงานการมอนิเตอร์การติดตาม Generate DLC Local Area Network	`smit trcrpt`	**trcrpt -d** *nnn* where the value *nnn* is the hook ID to be reported
แสดงข้อมูล DLC ปัจจุบัน [3]	เลือกหนึ่ง (โดยชื่อไดรเวอร์อุปกรณ์): `smit cmddlc_sdlc_ls` `smit cmddlc_token_ls` `smit cmddlc_qllc_ls` `smit cmddlc_ether_ls`[1] `smit cmddlc_fddi_ls`	**lsdev**[2] หรือ **lsattr**[2]

ตารางที่ 111. งานสำหรับการจัดการ DLC ไดรเวอร์อุปกรณ์ (ต่อ)

งาน	วิธีลัด SMIT	คำสั่งหรือไฟล์
ลบ DLC[3,6]	เลือกหนึ่ง (โดยชื่อไดรเวอร์อุปกรณ์): `smit cmddlc_sdlc_rm` `smit cmddlc_token_rm` `smit cmddlc_qllc_rm` `smit cmddlc_ether_rm`[1] `smit cmddlc_fddi_rm`	**rmdev**[2]

หมายเหตุ:

1. SMIT fast path สำหรับตัวจัดการอุปกรณ์ Ethernet จะรวมทั้งตัวจัดการ Standard Ethernet และ IEEE 802.3 Ethernet
2. รายละเอียดเกี่ยวกับอ็อพชันของบรรทัดรับคำสั่งถูกจัดเตรียมในคำอธิบายของคำสั่งสำหรับ **mkdev, chdev, trace, trcstop, trcrpt, lsdev, lsattr** หรือ **rmdev** ใน *ข้อมูลอ้างอิงคำสั่ง วอลุ่ม 4*
3. DLC ต้องถูกติดตั้งและถูกเพิ่มก่อนที่คุณจะสามารถลิสต์ แสดง เปลี่ยน หรือลบแอ็ตทริบิวต์ของมัน (ดูที่ "GDLC data link controls" ในหน้า 701) การเปลี่ยนแอ็ตทริบิวต์จะสำเร็จถ้าไม่มีแอ็คทีฟที่เปิดกับ DLC เป้าหมาย ก่อนที่จะทำแอ็คชันการเปลี่ยน ผู้ใช้อาจต้องหยุดเซอร์วิส เช่น SNA, OSI หรือ NetBIOS จากการใช้ DLC
4. การเปลี่ยนขนาดของคิวการรับข้อมูลจะมีผลกับรีซอร์สของระบบโดยตรง ทำการเปลี่ยนแปลงเฉพาะถ้า DLC มีปัญหาเกี่ยวกับคิวการรับข้อมูล เช่น ประสิทธิภาพช้าลง หรือ overflow ระหว่าง DLC และตัวจัดการอุปกรณ์ของมัน
5. ทำการเปิดใช้งานการมอนิเตอร์การติดตามอย่างระมัดระวังเนื่องจากมันมีผลโดยตรงกับประสิทธิภาพของ DLCs และสิ่งที่เกี่ยวข้องกับมัน
6. การลบ DLC จะสามารถทำได้ถ้าไม่มีแอ็คทีฟที่เปิดกับ DLC เป้าหมาย ก่อนที่จะทำแอ็คชันการลบ ผู้ใช้อาจต้องหยุดเซอร์วิส เช่น SNA, OSI หรือ NetBIOS จากการใช้ DLC

การอ้างอิงการสื่อสารและเน็ตเวิร์กอะแด็ปเตอร์

สถานการณ์จำลองการของคอนฟิกูเรชันที่แตกต่างกันสำหรับทั้ง PCI อะแด็ปเตอร์และอะซิงโครนัสอะแด็ปเตอร์สามารถถูกอ้างอิงที่นี่

อะแด็ปเตอร์ PCI

ข้อมูลการติดตั้งและคอนฟิกูเรชันสำหรับอะแด็ปเตอร์ PCI ที่แนะนำไว้ที่นี้

หัวข้อที่กล่าวถึงนี้ได้รับการสนับสนุนและมีข้อมูลคอนฟิกูเรชันสำหรับอะแด็ปเตอร์ PCI Wide Area Network (WAN) ("ไดรเวอร์อุปกรณ์เน็ตเวิร์ก HDLC มัลติโปรโตคอลแบบ 2 พอร์ต" และ "อะแด็ปเตอร์ ARTIC960Hx PCI" ในหน้า 707)

ไดรเวอร์อุปกรณ์เน็ตเวิร์ก HDLC มัลติโปรโตคอลแบบ 2 พอร์ต

ไดรเวอร์อุปกรณ์อะแด็ปเตอร์มัลติโปรโตคอลแบบ 2 พอร์ตในการเชื่อมต่อข้อมูลระดับสูง (HDLC) คือคอมโพเนนต์ของการสื่อสารระบบย่อย I/O ไดรเวอร์อุปกรณ์นี้ จัดเตรียมสนับสนุนสำหรับการดำเนินการ HDLC ผ่านอะแด็ปเตอร์มัลติโปรโตคอลแบบ 2 พอร์ต ที่มีความเร็วมากสุด 1.544Mbps

อ็อพชันต่อไปนี้จัดเตรียมการเข้าถึงไดรเวอร์อุปกรณ์เน็ตเวิร์ก HDLC แบบมัลติโปรโตคอลแบบ 2 พอร์ต:

- Systems Network Architecture (SNA)
- เวอร์ชัน synchronous data link control (SDLC) ของ GDLC Programming Interface

- แอ็พพลิเคชันที่เขียนโดยผู้ใช้ทำงานร่วมกันได้กับ SDLC MPQP-API (Multiprotocol Quad Port-Application Programming Interface)

หมายเหตุ: อ็อพชันข้างต้นจำเป็นต้องใช้ไฟล์พิเศษ `mpc`*n* ซึ่งอนุญาตให้เข้าถึงไดรเวอร์อุปกรณ์ HDLC ของอะแด็ปเตอร์มัลติโปรโตคอลแบบ 2 พอร์ต ผ่านระบบย่อยอีมูเลชันไดรเวอร์อุปกรณ์ SDLC COMIO ระบบย่อยนี้ ต้องถูกติดตั้งและตั้งค่าสำหรับอุปกรณ์เน็ตเวิร์ก HDLC แต่ละตัว

- แอ็พพลิเคชันถูกเขียนโดยผู้ใช้ทำงานร่วมกันกับ HDLC Common Data Link Interface (CDLI) API

ไดรเวอร์อุปกรณ์อะแด็ปเตอร์มัลติโปรโตคอลแบบ 2 พอร์ตอนุญาตให้ใช้ภาวะการเชื่อมต่อกับ ระบบรีโมตโฮสต์โดยใช้อะแด็ปเตอร์แบบมัลติโปรโตคอลแบบ 2 พอร์ต ไม่ว่าจะทางตรง ผ่านสายเช่าหรือผ่านวงจรที่สับเปลี่ยน ไดรเวอร์อุปกรณ์สามารถจัดเตรียมเกตเวย์ ระหว่างสภาพแวดล้อมกลุ่มงานและสิ่งอำนวยความสะดวกในการประมวลผลข้อมูลรีโมต

คอนฟิกูเรชันอะแด็ปเตอร์แบบมัลติโปรโตคอลแบบ 2 พอร์ต

ใช้คำอธิบายหล่านี้เพื่อตั้งค่าอะแด็ปเตอร์มัลติโปรโตคอลแบบ 2 พอร์ต

ตารางที่ 112. ภารกิจสำหรับการตั้งค่าอะแด็ปเตอร์แบบมัลติโปรโตคอลแบบ 2 พอร์ต

ภารกิจ	วิธีลัด SMIT
เพิ่มไดรเวอร์อุปกรณ์ให้กับอะแด็ปเตอร์	`smit mkhdlcdpmpdd`
ตั้งค่าไดรเวอร์อุปกรณ์บนอะแด็ปเตอร์อีกครั้ง	`smit chhdlcdpmpdd`
ลบไดรเวอร์อุปกรณ์บนอะแด็ปเตอร์	`smit rmhdlcdpmpdd`
ทำให้ไดรเวอร์อุปกรณ์ที่นิยามไว้พร้อมใช้งาน	`smit cfghdlcdpmpdd`
เพิ่มอีมูเลเตอร์ SDLC COMIO บนอะแด็ปเตอร์	`smit mksdlcsciedd`
ตั้งค่าอีมูเลเตอร์ SDLC COMIO บนอะแด็ปเตอร์	`smit chsdlcsciedd`
ลบอีมูเลเตอร์ SDLC COMIO บนอะแด็ปเตอร์	`smit rmsdlcsciedd`
ทำให้อีมูเลเตอร์ SDLC COMIO ที่นิยามไว้พร้อมใช้งาน	`smit cfgsdlcsciedd`

อะแด็ปเตอร์ ARTIC960Hx PCI

อะแด็ปเตอร์ ARTIC960Hx PCI ไดรเวอร์อุปกรณ์อีมูเลเตอร์ MPQP COMIO คือส่วนประกอบของระบบย่อยการสื่อสาร I/O ไดรเวอร์อุปกรณ์นี้จัดเตรียมส่วนสนับสนุน สำหรับอะแด็ปเตอร์ ARTIC960Hx PCI ที่ความเร็วสูงสุดของ 2M bps

โมเด็มที่ใช้ต้องจัดเตรียมนาฬิกา เนื่องจากนาฬิกาภายนอก ได้รับการสนับสนุนเท่านั้น

อ็อพชันต่อไปนี้จัดเตรียมการเข้าถึงอะแด็ปเตอร์ ARTIC960Hx PCI ไดรเวอร์อุปกรณ์ MPQP COMIO:

- Systems Network Architecture (SNA)
- generic data link control (GDLC) Programming Interface
- แอ็พพลิเคชันที่เขียนโดยผู้ใช้ทำงานร่วมกันได้กับ MPQP-API (Multiprotocol Quad Port-Application Programming Interface) เช่น แอ็พพลิเคชัน SDLC และ BiSync

อ็อพชันเหล่านี้ต้องการใช้ไฟล์พิเศษ mpq*x* ซึ่งอนุญาตให้เข้าถึงอะแด็ปเตอร์ ARTIC960Hx PCI ผ่านไดรเวอร์อุปกรณ์ MPQP COMIO อีมูเลชัน ไดรเวอร์อุปกรณ์นี้ต้องถูกติดตั้งและตั้งค่าสำหรับพอร์ตแต่ละพอร์ตบนอะแด็ปเตอร์ ARTIC960Hx PCI ไฟล์พิเศษ mpq*x* ตั้งอยู่ในไดเร็กทอรี /dev

หมายเหตุ: *x* ใน mpq*x* ระบุอินสแตนซ์ของไดรเวอร์อุปกรณ์ เช่น mpq0

ไดรเวอร์อุปกรณ์อีมูเลชัน MPQP COMIO อนุญาตให้ใช้ภาวะเชื่อมต่อกับระบบโฮสต์แบบรีโมต โดยใช้อะแด็ปเตอร์ ARTIC960Hx PCIผ่านสายเช่าโดยตรง ไดรเวอร์อุปกรณ์สามารถจัดเตรียมเกตเวย์ ระหว่างสภาพแวดล้อมกลุ่มงานและสิ่งอำนวยความสะดวกในการประมวลผลข้อมูลรีโมต

คอนฟิกูเรชันไดรเวอร์อีมูเลชัน MPQP COMIO ผ่านอะแด็ปเตอร์ ARTIC960Hx PCI

ใช้คำอธิบายเหล่านี้เพื่อตั้งค่าไดรเวอร์อีมูเลชัน MPQP COMIO ผ่านอะแด็ปเตอร์ ARTIC960Hx PCI

ตารางที่ 113. ภารกิจสำหรับการตั้งค่าไดรเวอร์อีมูเลชัน MPQP COMIO

ภารกิจ	วิธีลัด SMIT
เพิ่มไดรเวอร์อุปกรณ์	smit mktsdd
ตั้งค่าไดรเวอร์อีมูเลชัน MPQP COMIO อีกครั้ง	smit chtsdd
ลบไดรเวอร์อุปกรณ์	smit rmtsdd
ตั้งค่าไดรเวอร์อุปกรณ์ที่นิยามไว้	smit cfgtsdd
เพิ่มพอร์ต	smit mktsdports
ตั้งค่าพอร์ตอีมูเลชัน MPQP COMIO อีกครั้ง	smit chtsdports
ลบพอร์ต	smit rmtsdports
ตั้งค่าพอร์ตที่นิยามไว้	smit cfgtsdports
ติดตามไดรเวอร์อีมูเลชัน MPQP COMIO	smit trace_link

อะซิงโครนัสอะแด็ปเตอร์

อะซิงโครนัสอะแด็ปเตอร์แบบ 8 พอร์ตและ 16 พอร์ตมาตรฐาน ที่แสดงในตารางนี้

ตารางต่อไปนี้จะสรุปผลิตภัณฑ์เหล่านี้ :

ตารางที่ 114. อะซิงโครนัสอะแด็ปเตอร์

การเชื่อมต่อแบบอะซิงโครนัส	ชนิดของบัส	โค้ดคุณลักษณะ หรือ ชนิดของเครื่อง (โมเดล)	อัตราข้อมูลสูงสุดต่อพอร์ต (KBits/วินาที)	คุณลักษณะที่ไม่มีการโต้ตอบ
8-พอร์ต EIA 232	Micro Channel	2930	76.8	มาตรฐานที่แพรหลาย
8-พอร์ต EIA 422A	Micro Channel	2940	76.8	ระยะที่มากกว่า
8-พอร์ต MIL-STD 188	Micro Channel	2950	สามารถเลือกได้โดยขึ้นอยู่กับความเร็วของสัญญาณนาฬิกาของตัวสร้างอัตรา baud ของ UART	MIL-STD 188-114 สำหรับอินเตอร์เฟสโวลท์เตจดิจิตอลแบบ unbalanced

ตารางที่ 114. อะซิงโครนัสอะแด็ปเตอร์ (ต่อ)

การเชื่อมต่อแบบอะซิงโครนัส	ชนิดของบัส	โค้ดคุณลักษณะ หรือ ชนิดของเครื่อง (โมเดล)	อัตราข้อมูลสูงสุดต่อพอร์ต (KBits/วินาที)	คุณลักษณะที่ไม่มีการโต้ตอบ
8-พอร์ต EIA 232	ISA	2931	115.2	มีประสิทธิภาพมากกว่า
8-พอร์ต EIA 232	ISA	2932	115.2	มีประสิทธิภาพมากกว่า
8-พอร์ต EIA 422	PCI	2943	230	มีประสิทธิภาพมากกว่า
16-พอร์ต EIA 232	Micro Channel	2955	76.8	โฟกัสที่การเชื่อมต่อแบบโลคัล
16 พอร์ต EIA 422A	Micro Channel	2957	76.8	ระยะที่มากกว่า
-	ISA	2933	-	-
-	PCI	2944	-	-

ตารางต่อไปนี้แสดงคุณลักษณะของผลิตภัณฑ์แบบละเอียด

ตารางที่ 115. คุณลักษณะผลิตภัณฑ์ที่เชื่อมต่อแบบอะซิงโครนัส

	ซีเรียลพอร์ตแบบดั้งเดิม	8-พอร์ต		16-พอร์ต	128-พอร์ตพร้อมกับ RAN	
		MC	ISA		MC	ISA
จำนวนของพอร์ตแบบอะซิงโครนัสต่ออะแด็ปเตอร์	n/a	8	8	16	128	128
จำนวนของอะแด็ปเตอร์สูงสุด	n/a	8	7	8	7	7
จำนวนของพอร์ตแบบอะซิงโครนัสสูงสุด	2 หรือ 3	64	56	128	896	896
จำนวนของพอร์ตแบบอะซิงโครนัสต่อ RAN	n/a	n/a	n/a	n/a	16	16
จำนวนของ RAN สูงสุด	n/a	n/a	n/a	n/a	56	56
ความเร็วสูงสุด (KBits/วินาที)	สามารถเลือกได้โดยขึ้นอยู่กับความเร็วของสัญญาณนาฬิกาของตัวสร้างอัตรา baud ของ UART	76.8	115.2	76.8	230	230
วิธีการเชื่อมต่อ	มาตรฐาน	โดยตรง	โดยตรง	โดยตรง	โหนด	โหนด
การสนับสนุนอินเตอร์เฟสอะซิงโครนัสแบบไฟฟ้า	EIA 232	EIA 232 EIA 422A[4] MIL-STD[4] 188-114[4]	EIA 232 EIA 422A	EIA 232 EIA 422A	EIA 232 EIA 422	EIA 232 EIA 422
มาตรฐานตัวเชื่อมต่อ	DB25M/ MODU	DB25M	DB25M	DB25M	RJ-45[2]	RJ-45[2]

ตารางที่ 115. คุณลักษณะผลิตภัณฑ์ที่เชื่อมต่อแบบอะซิงโครนัส (ต่อ)

	ซีเรียลพอร์ตแบบดั้งเดิม	8-พอร์ต		16-พอร์ต	128-พอร์ตพร้อมกับ RAN	
อ็อพชันของสายเคเบิล DB25	n/a	n/a	n/a	n/a	RJ-45-DB25	RJ-45-DB25
อ็อพชันแบบยึด Rack	n/a	n/a	n/a	n/a	yes	yes
แหล่งจ่ายไฟ	n/a	n/a	n/a	n/a	external	external
สัญญาณที่ได้รับการสนับสนุน (EIA 232)	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS[3]- DTR -DCD -	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI

หมายเหตุ:

1. ซ็อกเก็ตยอมรับปลั๊ก 8p RJ-45, 6p RJ-11 หรือ 4p RJ-11 พร้อมกับลดสัญญาณที่สนับสนุน
2. RTS จะมีค่าสูง (+12V) ในกล่องตัวเชื่อมต่อ fanout ของสายเคเบิล16-พอร์ตอินเตอร์เฟส EIA 232 (FC 2996)
3. Micro Channel เท่านั้น

การเสนอของแต่ละผลิตภัณฑ์ถูกทำให้เป็นคุณลักษณะโดยการแสดงสถานการณ์จำลองสำหรับความแข็งแกร่งของมัน ต่อไปนี้เป็นข้อแนะนำสำหรับแต่ละตัว

หลักการที่เกี่ยวข้อง:

"อุปกรณ์เทอร์มินัล TTY" ในหน้า 624
อุปกรณ์เทอร์มินัล tty เป็นอุปกรณ์อักขระที่ดำเนินการอินพุตเอาต์พุตบนพื้นฐานของอักขระต่ออักขระ

ข้อมูลที่เกี่ยวข้อง:

คู่มือการติดตั้งและการใช้อะแด็ปเตอร์ EIA-232 PCI แบบอะซิงโครนัส 2 พอร์ต

8-พอร์ต Micro Channel

คุณลักษณะของ 8-พอร์ต Micro Channel อะแด็ปเตอร์รวมถึง Micro Channel บัสสล็อตสำหรับ I/O แบบอะซิงโครนัส

คุณลักษณะอื่นรวมถึงต่อไปนี้:

- น้อยกว่า 8 พอร์ตโดยมีการขยายน้อย หรือไม่มีเลย
- เทอร์มินัลแบบโลคัลทั้งหมดอยู่ในระยย 61 เมตร (200 ฟุต) จากระบบ
- ต้องการรีโมตเทอร์มินัล (สนับสนุนผ่าน OEM มัลติเพล็กเซอร์/โมเด็ม)
- ต้องการแบนด์วิดธ์ของอุปกรณ์ต่ำหรือปานกลาง (มากถึง 76.8 Kbps)

8-พอร์ต ISA bus EIA 232 หรือ EIA 232/EIA 422

คุณลักษณะของ 8-พอร์ต ISA bus EIA 232 หรือ EIA 232/EIA 422 อะแด็ปเตอร์ จะรวม ISA สล็อต

คุณลักษณะอื่นรวมถึงต่อไปนี้:

- ต้องการน้อยกว่า 8 พอร์ตโดยมีการขยายน้อยหรือไม่มี
- ต้องการพอร์ต EIA 232 ทั้งหมด EIA 422 ทั้งหมด หรือผสมกันระหว่างพอร์ต EIA 232 และ EIA 422

- ลดโหลดการอินเตอร์รัปต์ตัวอักษร และการประมวลผลเทอร์มินัล I/O จาก CPU หลัก
- ความเร็วอะซิงโครนัสถึง 115.2 Kbps
- ประสิทธิภาพสูงสุดสำหรับโมเด็มความเร็วสูง (28.8 Kbps) พร้อมกับการบีบอัดข้อมูล

16-พอร์ต Micro Channel

คุณลักษณะของ 16-พอร์ต Micro Channel อะแด็ปเตอร์จะรวม Micro Channel บัสสล็อตที่พร้อมใช้งานสำหรับอะซิงโครนัส I/O

คุณลักษณะอื่นรวมถึงต่อไปนี้:
- ตอนนี้มี 8 พอร์ต น้อยกว่า 16 พอร์ตโดยมีการขยายน้อยหรือไม่มี
- เทอร์มินัลแบบโลคัลทั้งหมดอยู่ในระยย 61 เมตร (200 ฟุต) จากระบบ
- ต้องการรีโมตเทอร์มินัล (สนับสนุนผ่าน OEM มัลติเพล็กเซอร์/โมเด็ม)
- อุปกรณ์ไม่ต้องการสัญญาณ EIA 232 ทั้งหมด
- อุปกรณ์ต้องการแบนด์วิดธ์น้อยถึงปานกลาง (มากถึง 38.4 Kbps สำหรับอุปกรณ์แบบอะซิงโครนัส)

128-พอร์ต อะแด็ปเตอร์ (Micro Channel, ISA)

คุณลักษณะของ 128-พอร์ต Micro Channel หรือ ISA อะแด็ปเตอร์จะรวม 16 พอร์ตที่ขยายได้มากถึง 128 พอร์ตโดยไม่มีสล็อตเพิ่มเติม

คุณลักษณะอื่นรวมถึงต่อไปนี้:
- Micro Channel, ISA หรือ PCI บัสสล็อต ที่มีให้สำหรับอะซิงโครนัส I/O (สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ PCI ดูที่ “การพิจารณาเลือกผลิตภัณฑ์” ในหน้า 613)
- ระยะของเทอร์มินัลส่วนใหญ่จะอยู่ที่ 300 เมตร (1000 ฟุต) จากระบบ ที่อัตราของข้อมูลสูงสุดสำหรับ Micro Channel และ ISA อะแด็ปเตอร์
- มีวางแผนสำหรับเทอร์มินัล: ใกล้ๆหรือในสถานที่ ระยะไกลหรือในสถานที่ และรีโมต
- ต้องการปริมาณงานของอะซิงโครนัสสูงโดยต้องการตัวประมวลผลต่ำ
- ต้องการความสามารถเชื่อมเครื่องพิมพ์กับเทอร์มินัล
- ต้องการเชื่อมต่อสถานที่แบบรีโมตผ่าน fiber-optic หรืออะซิงโครนัสโมเด็ม

การลิสต์ Micro Channel 128-พอร์ต อะซิงโครนัสอะแด็ปเตอร์ที่ถูกกำหนดโดยใช้ SMIT

ใช้โพรซีเดอร์นี้เพื่อลิสต์ 128-พอร์ต อะซิงโครนัสอะแด็ปเตอร์ โดยไม่สนใจว่ามันจะพร้อมใช้งานหรือไม่

1. ใช้ `smit lsd128psync` fast path ระบบจะสแกนสำหรับข้อมูลและแสดงมัน
2. ออกจาก SMIT อินเตอร์เฟส

8-พอร์ต อะซิงโครนัส ISA/PCI อะแด็ปเตอร์

8-พอร์ต อะซิงโครนัส ISA อะแด็ปเตอร์เป็นคุณลักษณะการสื่อสารแบบซีเรียลแบบ มัลติแชนแนลและฉลาด ที่พร้อมใช้งานสำหรับคอมพิวเตอร์ POWER ที่อิงตัวประมวลผล

ISA อะแด็ปเตอร์ประกอบด้วย 128K ของ Random Access Memory (RAM) แบบ dual-ported high-speed ที่ถูกใช้สำหรับโปรแกรมโค้ดและการบัฟเฟอร์ข้อมูล พอร์ตแบบอะซิงโครนัสถูกรันโดยตัวประมวลผล 32-บิต 16 MHz IDT 3041 ที่สนับสนุนความเร็วของปริมาณงานถึง 115 Kbps

ตัวประมวลผล 3041 และ RAM แบบ dual-ported จะช่วยลดโหลดจำนวนมากของการประมวลผลตัวอักษรจากระบบ บล็อกขนาดใหญ่ของข้อมูลถูกถ่ายโอนไปยังอะแด็ปเตอร์โดยตรง และจากนั้นส่งออกไปบนซีเรียลพอร์ตทีละหนึ่งตัวอักษร

RAM แบบ dual-ported สามารถถูกเข้าถึงเพื่อการดำเนินการอ่านและเขียนโดยทั้งอะแด็ปเตอร์และคอมพิวเตอร์ คอมพิวเตอร์จะเห็น RAM แบบ dual ported เป็นหน่วยความจำของมันเองและเข้าถึงมันโดยใช้คำสั่งที่อ้างอิงหน่วยความจำความเร็วสูงเดียวกับที่มันใช้สำหรับหน่วยความจำภายใน

8-พอร์ต EIA 232 ISA อะแด็ปเตอร์สนับสนุนอุปกรณ์ EIA 232 เท่านั้น อะแด็ปเตอร์นี้ต้องการให้ติดตั้งแพ็กเกจของอุปกรณ์ `devices.isa.cxia` บนระบบ

8-พอร์ต EIA 232/422 ISA อะแด็ปเตอร์สนับสนุนอุปกรณ์ EIA 232 หรือ EIA 422 อุปกรณ์ทั้งสองชนิดอาจถูกตั้งค่าในการรวมใดๆแบบต่อพอร์ต อะแด็ปเตอร์นี้ต้องการให้ติดตั้งแพ็กเกจของอุปกรณ์ `devices.isa.pc8s` บนระบบ

แพ็กเกจด้านบนต้องการแพ็กเกจ `devices.common.IBM.cx`

การติดตั้ง 8-พอร์ตอะแด็ปเตอร์:

ISA อะแด็ปเตอร์ไม่สามารถถูกตรวจพบโดยอัตโนมัติโดยระบบปฏิบัติการ และต้องถูกติดตั้งแบบแมนวล

1. เมื่อต้องการตั้งค่า IBM 8-พอร์ต อะซิงโครนัส EIA 232/EIA 422 ISA อะแด็ปเตอร์ใช้ `smit mkdev_isa` fast path เพื่อเข้าถึง **หน้าจอ Add an ISA Adapter**
2. เลือก **pcxr** (สำหรับ 8-พอร์ต EIA 232 อะแด็ปเตอร์) หรือ **pc8s** (สำหรับ 8-พอร์ต EIA 232/EIA 422 อะแด็ปเตอร์) และกด Enter
3. เลือกบัสที่เหมาะสมและกด Enter
4. ในฟิลด์ Bus I/O Address ตั้งแอดเดรสเป็นแอดเดรสของอะแด็ปเตอร์ (ตั้งโดย DIP สวิตช์บนอะแด็ปเตอร์) สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ DIP สวิตช์ อ้างถึง *คู่มือการติดตั้งอะแด็ปเตอร์ Asynchronous ISA 8 พอร์ต* คอนฟิกูเรชันของอะแด็ปเตอร์ที่เหลือจะถูกทำโดยอัตโนมัติเมื่อระบบแสดง `saX Available`
5. เมื่อทำเสร็จ เลือก **Do**

stty-cxma เป็นยูทิลิตีโปรแกรมที่ตั้งและแสดงอ็อพชันของเทอร์มินัลสำหรับ Micro Channel 128-พอร์ต และ ISA 8- และ 128-พอร์ต อะแด็ปเตอร์ และอยู่ในไดเร็กทอรี `/usr/lbin/tty` รูปแบบคือ:

```
stty-cxma [-a] [option(s)] [ttyname]
```

โดยไม่มีอ็อพชัน **stty-cxma** จะแสดงการตั้งค่าไดรเวอร์พิเศษทั้งหมด สัญญาณโมเด็ม และพารามิเตอร์มาตรฐานทั้งหมดที่ถูกแสดงโดย **stty(1)** สำหรับอุปกรณ์ tty ที่ถูกอ้างถึงโดยอินพุตมาตรฐาน อ็อพชันของคำสั่งถูกจัดเตรียมเพื่อเปลี่ยนการตั้งค่าโฟลว์คอนโทรล ตั้งอ็อพชันการพิมพ์แบบ transparent บังคับสายควบคุมโมเด็ม และแสดงการตั้งค่า tty ทั้งหมด อ็อพชันที่ไม่รู้จักใดๆจะถูกผ่านไปยัง **stty(1)** สำหรับการแปล อ็อพชันจะเหมือนกับที่ถูกใช้สำหรับ PCI อะแด็ปเตอร์ สำหรับข้อมูลเพิ่มเติม ดูที่ “อ็อพชันของเทอร์มินัล stty-cxma” ในหน้า 656

พอร์ต I/O มาตรฐาน

หน่วยของระบบส่วนใหญ่มีสองซีเรียลพอร์ตแบบอะซิงโครนัสแบบรวม (มาตรฐาน) EIA 232

โหมด M20/M2A มีคุณลักษณะซีเรียลพอร์ตอะซิงโครนัสแบบรวมเดียวที่สามารถถูกแปลงเพื่อสนับสนุนอุปกรณ์ที่มี 2 ซีเรียลโดยใช้สายเคเบิล fanout ที่เป็นอ็อพชัน อุปกรณ์ EIA 232 อะซิงโครนัสแบบซีเรียลสามารถเชื่อมต่อโดยตรงกับซีเรียลพอร์ตแบบมาตรฐานโดยใช้สายซีเรียลแบบมาตรฐานโดยตัวเชื่อมต่อ 9-pin หรือ 25-pin D-shell

> **หมายเหตุ:** สำหรับแพล็ตพอร์ม Itanium-based อุปกรณ์ EIA 232 อะซิงโครนัสแบบซีเรียลสามารถเชื่อมต่อโดยตรงกับซีเรียลพอร์ตแบบมาตรฐานโดยใช้สายซีเรียลแบบมาตรฐานโดยตัวเชื่อมต่อ 9-pin D-shell

เครื่องที่มีความสามารถหลายการประมวลผลมี 3 ซีเรียลพอร์ต

การตั้งค่าอุปกรณ์เทอร์มินัลอะซิงโครนัส EIA 232:

โพรซีเดอร์นี้ให้คุณสามารถกำหนดและตั้งค่าอุปกรณ์ tty ที่เชื่อมต่อกับซีเรียลพอร์ตแบบมาตรฐาน 8-พอร์ต หรือ 16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์

1. ใช้ `smit mktty` fast path เพื่อเข้าถึงเมนู **Add a TTY**
2. เลือก **Add a TTY**
3. เลือก **tty rs232 Asynchronous Terminal**
4. ทำการเลือกจากอะแด็ปเตอร์ I/O, 8-พอร์ต หรือ 16-มาตรฐานที่ถูกแสดงบนหน้าจอ ถ้าไม่มีอะแด็ปเตอร์ถูกแสดง หรือถ้ามันถูกแสดงในสถานะที่ถูกกำหนดแล้ว ตรวจสอบค่าคอนฟิกูเรชัน สายเคเบิล และตั้งค่าอีกครั้ง
5. ในฟิลด์ไดอะล็อกที่ถูกแสดง คุณสามารถเพิ่ม หรือเปลี่ยนแปลงแอ็ตทริบิวต์ของ tty
6. เมื่อทำเสร็จ เลือก **Do**

การตั้งค่าอุปกรณ์เครื่องพิมพ์/พล็อตเตอร์ EIA 232 อะซิงโครนัส:

โพรซีเดอร์นี้ให้คุณสามารถกำหนดและตั้งค่าอุปกรณ์เครื่องพิมพ์/พลอตเตอร์ที่เชื่อมต่อกับซีเรียลพอร์ตแบบมาตรฐาน 8-พอร์ต อะซิงโครนัสอะแด็ปเตอร์ หรือ 16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์

1. เพื่อสร้างอุปกรณ์เครื่องพิมพ์/พล็อตเตอร์บนอะซิงโครนัสอะแด็ปเตอร์ ใช้ `smit pdp` fast path เพื่อเข้าถึงเมนู **Printer/Plotter Devices**
2. เลือก **Add a Printer/Plotter**
3. ทำการเลือกจากลิสต์ของเครื่องพิมพ์ และ พลอตเตอร์ที่ถูกแสดงบนหน้าจอ และกด Enter สำหรับตัวอย่างนี้ ได้ทำการเลือกดังต่อไปนี้:

   ```
   osp Other serial printer
   ```

4. เลือกอ็อพชัน **rs232**
5. ทำการเลือกจากตัวควบคุม 8-พอร์ตที่มีอยู่บนหน้าจอ ถ้าไม่มีตัวควบคุมถูกแสดง หรือถ้ามันถูกแสดงในสถานะที่ถูกกำหนดแล้ว ตรวจสอบค่าคอนฟิกูเรชัน สายเคเบิล และตั้งค่าอีกครั้ง
6. ในฟิลด์ไดอะล็อกที่ถูกแสดง คุณสามารถเพิ่ม หรือเปลี่ยนแปลงแอ็ตทริบิวต์ของอุปกรณ์เครื่องพิมพ์/พล็อตเตอร์
7. เมื่อทำเสร็จ เลือก **Do**

อะแด็ปเตอร์อะซิงโครนัส 8-พอร์ต Micro Channel

ตระกูลของอะแด็ปเตอร์อะซิงโครนัสจะขึ้นอยู่กับการออกแบบการทำงานทั่วไป อย่างไรก็ตาม แต่ละคุณสมบัติของอะแด็ปเตอร์ จะถูกกำหนดโดยอินเตอร์เฟสของอุปกรณ์ที่ได้รับการสนับสนุน

หมายเหตุ: ส่วนต่อไปนี้ใช้ไม่ได้กับแพล็ตฟอร์ม Itanium-based

ตระกูลประกอบด้วย 3 อะแด็ปเตอร์ :

- 8-พอร์ต อะซิงโครนัส อะแด็ปเตอร์ - EIA 232
- 8-พอร์ต อะซิงโครนัส อะแด็ปเตอร์ - MIL-STD-188
- 8-พอร์ต อะซิงโครนัส อะแด็ปเตอร์ - EIA 422A

ตระกูลของ 8-พอร์ต อะแด็ปเตอร์ จะขึ้นอยู่กับชิป dual universal asynchronous receiver and transmitter (DUART) ซึ่งจัดเตรียมแชนแนลการสื่อสารแบบซีเรียล 2 แชนแนล

ส่วนต่อไปนี้ประกอบด้วยข้อมูลรายละเอียดเกี่ยวกับ 8-พอร์ต อะแด็ปเตอร์

8-พอร์ต อะซิงโครนัส อะแด็ปเตอร์ - EIA 232:

EIA 232 เป็น 8-พอร์ต อะซิงโครนัส อะแด็ปเตอร์ ที่ให้การสนับสนุนสำหรับการเชื่อมต่ออุปกรณ์ซีเรียลอะซิงโครนัส EIA 232D มากถึง 8 อุปกรณ์ (เช่น โมเด็ม เทอร์มินัล พล็อตเตอร์ และเครื่องพิมพ์) เข้ากับหน่วยของระบบ

ระบบต้องขึ้นอยู่กับ Micro Channel บัส หรือ ISA บัส และสนับสนุนอะแด็ปเตอร์ 8-พอร์ตมากถึง 8 อะแด็ปเตอร์

อะแด็ปเตอร์นี้สามารถตั้งโปรแกรมได้แบบเต็มที่และสนับสนุนการสื่อสารแบบอะซิงโครนัสเท่านั้น มันยังสามารถเพิ่มและลบ start และ stop บิต และสนับสนุน พารีตีแบบคู่ คี่ หรือไม่มีพาริตีบนข้อมลแบบซีเรียล ตัวกำเนิดอัตรา baud ที่สามารถโปรแกรมได้ยอมให้การทำงานจาก 50 ถึง 38,400 bps สำหรับ Micro Channel บัส และ 50 ถึง 115,200 bps สำหรับ ISA บัส อะแด็ปเตอร์สนับสนุนอักขระ 5-, 6-, 7- หรือ 8-บิต โดยมี 1, 1.5, or 2 stop บิต ระบบอินเตอร์รัปต์ที่มีลำดับความสำคัญจะควบคุมการส่ง รับ ข้อผิดพลาด สถานะของสาย และอินเตอร์รัปต์ชุดของข้อมูล

การติดตั้ง 8-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

8-พอร์ตอะซิงโครนัสอะแด็ปเตอร์จะพอดีกับสล็อต Micro Channel เดียวในระบบ ใช้ขั้นตอนเหล่านี้เพื่อติดตั้งอะแด็ปเตอร์

1. ตรวจสอบว่าผู้ใช้ล็อกออฟจากระบบและรันคำสั่งต่อไปนี้ :
   ```
   shutdown -F
   ```
2. เมื่อคำสั่ง **shutdown** ทำเสร็จแล้ว ปิดสวิตช์ไปที่ตำแหน่ง off
3. เปิดตัวครอบของระบบ และใส่ 8-พอร์ต อะซิงโครนัสอะแด็ปเตอร์เข้าไปที่สล็อต Micro Channel ที่ว่าง
4. ติดตัวเชื่อมต่อ 78-pin D-shell จากสายเคเบิลอินเตอร์เฟส 8-พอร์ต เข้ากับ 8-พอร์ตอะแด็ปเตอร์
5. ใส่ฝาครอบกลับเข้ากับระบบ
6. เปิดสวิตช์ของระบบไปที่ตำแหน่ง on ระบบจะรู้จักและตั้งค่า 8-พอร์ต อะแด็ปเตอร์ ระหว่างกระบวนการบูต
7. หลังจากการบูตเรียบร้อยแล้ว ล็อกอินโดยใช้ ID ของผู้ใช้ root สำหรับกระบวนการบูต
   ```
   lsdev -Cc adapter | pg
   ```

เฉพาะอะแด็ปเตอร์เหล่านั้นที่อยู่ในสถานะพร้อมใช้ที่พร้อมใช้งานโดยระบบ

ถ้าอะแด็ปเตอร์ที่ถูกติดตั้งใหม่ *ไม่* พร้อมใช้งาน ดังนั้นตรวจสอบ :

- อะแด็ปเตอร์ถูกติดตั้งอย่างถูกต้องเข้ากับสล็อต Micro Channel
- สายเคเบิลที่จำเป็นถูกเชื่อมต่อและติดแน่นอยู่กับที่ของมัน
- รันคำสั่ง **errpt -a | pg** และตรวจสอบรายงานข้อผิดพลาดของระบบสำหรับปัญหาที่เกี่ยวข้องกับอะแด็ปเตอร์
- รันคำสั่ง : **cfgmgr -v | pg** คำสั่งนี้จะพยายามตั้งค่าอะแด็ปเตอร์ใหม่โดยไม่ต้องรีบูต สังเกตเอาต์พุตของข้อผิดพลาดที่ถูกแสดง

ถ้าการรัน **cfgmgr** ล้มเหลว จำเป็นต้องรีบูต

ข้อมูลฮาร์ดแวร์ของ 8-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

อินเตอร์เฟสของระบบแสดง 3-บิตแอดเดรส และ 8-บิตข้อมูล เช่นเดียวกับสายควบคุมกับชิป DUART ข้อมูลจากอินเตอร์เฟสของระบบจัดลำดับเพื่อส่งไปยังอุปกรณ์ภายนอก ข้อมูลแบบเรียงลำดับอาจจะรวมพาริตีบิตที่ขอบเขตของไบต์ ในทางตรงข้าม ข้อมูลจากอุปกรณ์ภายนอกจะถูกยกเลิกการเรียงลำดับสำหรับส่งไปยังอินเตอร์เฟสของระบบ ข้อมูลนี้ยังอาจรวมพาริตีบิต ซึ่งสามารถถูกเลือกที่จะตรวจสอบได้ โดยที่เป็นอ็อพชัน แชนแนลสามารถทำงานในโหมด first-in-first-out (FIFO)

ในโหมด FIFO ข้อมูลมากถึง 16 ไบต์สามารถถูกเก็บบัฟเฟอร์ทั้งในตัวส่งและตัวรับ ซีเรียลอินเตอร์เฟสใช้โปรโตคอลแบบ start-stop สำหรับทั้งการส่งและรับข้อมูล ซึ่งคือ แต่ละไบต์ (รวมพาริตีบิต) ถูกจัดเป็นเฟรมโดยใช้หนึ่งหรือมากกว่าของ start และ stop บิต ซึ่งยอมให้การซิงโครไนซ์บนพื้นฐานแบบทีละตัวอักษร (ไบต์)

ชิป DUART ใช้ออสซิลเลเตอร์ 12.288 MHz เพื่อสร้างสัญญาณเวลาภายในเพื่อขับวงจรตัวส่งและตัวรับ แชนแนลสนับสนุนการทำงานแบบ full duplex ชิป DUART สี่ตัวถูกนำมาใช้บนแต่ละ 8-พอร์ตอะแด็ปเตอร์

13 การลงทะเบียน system-accessible มีให้ใช้งาน คุณลักษณะที่สามารถตั้งโปรแกรมได้บนแต่ละแชนแนล รวมถึง:

- ความยาวตัวอักษร: 5, 6, 7 หรือ 8 บิต
- การสร้างพาริตี/การตรวจจับ: คู่, คี่ หรือไม่มี
- จำนวนของ stop บิต: 1, 1.5 หรือ 2
- เปิดใช้งาน/ปิดใช้งาน อินเตอร์รัปต์ ข้อมูลที่ถูกรับมีให้ใช้งาน
- Transmitter holding register ว่างเปล่า
- สถานะของสาย
- ข้อผิดพลาด Overrun
- ข้อผิดพลาดพาริตี
- ข้อผิดพลาด Framing
- Break

ตารางต่อไปนี้เป็นสรุปของคุณสมบัติของพอร์ต (อินเตอร์เฟสของอุปกรณ์) สำหรับอะแด็ปเตอร์

ตารางที่ 116. คุณลักษณะของ 8-พอร์ต อะซิงโครนัสอะแด็ปเตอร์พอร์ต

พารามิเตอร์	EIA 232	MIL-STD 188	EIA 422A
โทโปโลยี	จุดต่อจุด	จุดต่อจุด	จุดต่อจุด
อัตราของข้อมูลสูงสุด	138.4Kbps (MC)/115.2 (ISA)	138.4Kbps	138.4Kbps
สื่อการส่งข้อมูล	Multiconductor	Multiconductor	Multiconductor
จำนวนสายของสายเคเบิล	9 รวมถึงสัญญาณกราวด์	9 รวมถึงสัญญาณกราวด์	5 รวมถึงสัญญาณกราวด์
ความยาวสูงสุดของสายเคเบิล	61 เมตร (200 ฟุต)	130 เมตร ที่ 38.4Kbps	1200 เมตร ที่ < 90Kbps
ตัวเชื่อมต่อของอุปกรณ์	25-pin D	25-pin D	25-pin D
อินเตอร์เฟสทางไฟฟ้า	Unbalanced	Unbalanced	Balanced
การเข้ารหัสบิต	Digital bi-level	Digital bi-level	Digital bi-level

ตรรกะการเลือกอินเตอร์รัปต์จะตั้งลำดับความสำคัญสำหรับอะแด็ปเตอร์ตาม scheme ต่อไปนี้

```
อะแด็ปเตอร์     ลำดับความสำคัญ
    1        สูงสุด
    2           |
    3           |
    4           |
    5           |
    6           |
    7           |
    8        ต่ำสุด
```

ลำดับความสำคัญของแชนแนลการสื่อสาร:

แชนแนล DUART กับอินเตอร์รัปท์ที่ค้างอยู่จะถูกให้บริการโดยขึ้นอยู่กับ scheme ลำดับความสำคัญที่คงที่

ลำดับความสำคัญสูงสุดถูกกำหนดให้กับพอร์ต 0 ลำดับความสำคัญถัดไปคือพอร์ต 1 และอย่างต่อไป ลำดับความสำคัญต่ำสุดคือพอร์ต 7

คำอธิบายตรรกะการอินเตอร์รัปต์ของ 8-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

ตรรกะการอินเตอร์รัปต์ถูกแบ่งออกเป็นตรรกะการสร้างอินเตอร์รัปต์ และตรรกะการเลือกอินเตอร์รัปต์

ส่วนของตรรกะทั้งสองถูกนำไปใช้บนทุก 8-พอร์ตอะแด็ปเตอร์ ตรรกะการสร้างอินเตอร์รัปต์จัดเตรียมอินเตอร์เฟสกับระบบ ตรรกะนี้จะสร้างคำร้องขออินเตอร์รัปต์ของระบบและประกอบด้วยวงจรการแบ่งใช้อินเตอร์รัปต์

ฟังก์ชันของการเลือกอินเตอร์รัปต์ใช้เพื่อระบุ 8-พอร์ต อะแด็ปเตอร์โดยการอินเตอร์รัปต์ที่มีลำดับความสำคัญสูงสุดที่ค้างอยู่ จากนั้นตรรกะจะใส่ข้อมูลการอินเตอร์รัปต์ของพอร์ตที่มีลำดับความสำคัญสูงสุดในการลงทะเบียนการเลือกอินเตอร์รัปต์ นี่ทำได้ในการดำเนินการอ่านเพียงครั้งเดียว

ตรรกะการเลือกอินเตอร์รัปต์จะเป็นหนึ่งเดียวกับ 8-พอร์ตอะแด็ปเตอร์ และไม่ควรสับสนกับตรรกกะการเลือก Micro Channel

ตรรกะการสร้างอินเตอร์รัปต์ 8-พอร์ต:

อะแด็ปเตอร์อะซิงค์ใช้สายคำร้องขออินเตอร์รัปต์ 8 ระบบ

อะแด็ปเตอร์ใช้สายคำร้องขออินเตอร์รัปต์ 8 ระบบต่อไปนี้ :

- IRQ 3
- IRQ 5
- IRQ 9
- IRQ 10
- IRQ 11
- IRQ 12
- IRQ 14
- IRQ 15

จะมีเพียงหนึ่งสายคำร้องขอเท่านั้นที่จะแอ็คทีฟระหว่างการทำงานตามปกติ อะแด็ปเตอร์ 8-พอร์ตทั้งหมดในระบบเดียวควรใช้ระดับของอินเตอร์รัปต์เดียวกันเพื่อประสิทธิภาพของระบบที่ดีที่สุด สายที่แอ็ตทีฟถูกเลือกโดยการเขียนไปยังการลงทะเบียน POS ที่เหมาะสมระหว่างรอบของการตั้งค่า อะแด็ปเตอร์สนับสนุนการแบ่งใช้อินเตอร์รัปต์ และใช้คอนฟิกูเรชันของตัวเชื่อมต่อแบบเปิด ในการกำหนดนี้สายอินเตอร์รัปต์จะถูกดึงให้สูงโดย pull-up resistor ของระบบ อะแด็ปเตอร์จะดึงสายให้ต่อเพื่อระบุคำร้องขออินเตอร์รัปต์ที่แอ็คทีฟ

ตรรกะการเลือกอินเตอร์รัปต์ 8-พอร์ต:

ตรรกะการเลือแอินเตอร์รัปต์จะตรวจสอบพาริตีสำหรับเซอร์วิสของซอฟต์แวร์ เมื่ออะแด็ปเตอร์ 8-พอร์ต หรือ 16-พอร์ต 2 อะแด็ปเตอร์หรือมากกว่าสร้างอินเตอร์รัปต์

อะแด็ปเตอร์ 8-พอร์ตมากถึง 8 อะแด็ปเตอร์สามารถอยู่ร่วมกันและทำงานพร้อมกันในระบบเดียว ตรรกะนี้จัดเตรียมอะแด็ปเตอร์และการระบุพอร์ตให้กับระบบ เช่นเดียวกับชนิดของอินเตอร์รัปต์ในการอ่านเพียงครั้งเดียว หลังจากคำร้องของอินเตอร์รัปต์ถูกตรวจพบ ระบบจะอ่านการลงทะเบียนการอินเตอร์รัปต์แบบ 16-บิต ซึ่งอยู่ที่ I/O แอดเดรส 0130

สัญญาณอินเตอร์เฟส MIL-STD 188 ของ 8-พอร์ต อะซิงโครนัสอะแด็ปเตอร์:

สัญญาณของอินเตอร์เฟสเหล่านี้ถูกนำไปใช้บนแต่ละพอร์ตของอะแด็ปเตอร์

Signal	**คำนิยาม**
Tx Data	ส่งข้อมูล
RTS	Request To Send
CTS	Clear To Send
DSR	Data Set Ready
Rx Data	รับข้อมูล
DCD	Data Carrier Detect
DTR	Data Terminal Ready
RI	Ring Indicator
Sig Gnd	สัญญาณกราวด์

ระดับโวลท์เตจของสัญญาณของ 8-พอร์ต MIL-STD 188:

ระดับโวลท์เตจสำหรับ MIL-STD 188 อะแด็ปเตอร์สามารถถูกอธิบายผ่านขั้วของ mark และ space แบบธรรมดา หรือขั้วของ mark และ space แบบกลับ

ระดับโวลท์เตจสำหรับ MIL-STD 188 อะแด็ปเตอร์ถูกอธิบายในส่วนต่อไปนี้ :

- ขั้วของ Mark และ Space แบบธรรมดา
- ขั้วของ Mark และ Space แบบกลับกัน

สัญญาณจะอยู่ในสถานะ mark เมื่อโวลท์เตจบนวงจรการแลกเปลี่ยน (ถูกวัดที่จุดเชื่อมต่อ) น้อยกว่า -4 V dc เมื่อเทียบกับสัญญาณกราวด์ สัญญาณอยู่ในสถานะ space เมื่อโวลท์เตจมากกว่า +4 V dc เมื่อเทียบกับสัญญาณกราวด์ ขอบเขตระหว่าง +4 V dc และ -4 V dc ถูกกำหนดเป็นขอบเขตของการเปลี่ยนแปลงและเป็นระดับที่ใช้ไม่ได้ โวลท์เตจที่น้อยกว่า -6 V dc หรือมากกว่า +6 V dc เป็นระดับที่ใช้ไม่ได้

ระหว่างการส่งข้อมูล สถานะ mark จะแทนไบนารี 1 และสถานะ space จะแทนไบนารี 0

สำหรับวงจรควบคุมอินเตอร์เฟส ฟังก์ชันจะเป็น "on" เมื่อโวลท์เตจมากกว่า +4 V dc เมื่อเทียบกับสัญญาณกราวด์ และจะเป็น "off" เมื่อโวลท์เตจน้อยกว่า -4 V dc เมื่อเทียบกับสัญญาณกราวด์ ระดับของสัญญาณ MIL-STD 188 จะถูกแสดงในตารางต่อไปนี้:

ตารางที่ 117. ระดับของสัญญาณ MIL-STD 188

โวลท์เตจการแลกเปลี่ยน	สถานะไบนารี	เงื่อนไขของสัญญาณ	ฟังก์ชันการควบคุมอินเตอร์เฟส
+ โวลท์เตจ	0	Space	On
- โวลท์เตจ	1	Mark	ปิด

มาตรฐานทางทหารของ MIL-STD 188 ต้องการให้อะแด็ปเตอร์มีความสามารถที่จะเลือกการกลับขั้วของสถานะ mark และ space ของสายการส่งและรับ ความสามารถนี้จะถูกจัดเตรียมบนแต่ละพอร์ตอย่างอิสระ

บิต 2 ของ DUART modem control register (Out 2) ถูกใช้เพื่อวัตถุประสงค์นี้ เมื่อบิต 3 ถูกตั้งเป็นค่า 1 ขั้วสำหรับสถานะ mark และ space จะถูกตั้งเป็นสถานะปกติ เมื่อบิต 3 ถูกตั้งเป็นค่า 0 ขั้วสำหรับสถานะ mark และ space จะถูกกลับ

สัญญาณจะอยู่ใน *สถานะ space* เมื่อโวลท์เตจน้อยกว่า -4 V dc เมื่อเทียบกับสัญญาณกราวด์ สัญญาณอยู่ในสถานะ mark เมื่อโวลท์เตจมากกว่า +4 V dc เมื่อเทียบกับสัญญาณกราวด์

ขอบเขตระหว่าง +4 V dc และ -4 V dc ถูกกำหนดเป็น *ขอบเขตการเปลี่ยนแปลง* และเป็นระดับที่ใช้ไม่ได้ โวลท์เตจที่น้อยกว่า -6 V dc หรือมากกว่า +6 V dc เป็นระดับที่ใช้ไม่ได้

คุณลักษณะทางไฟฟ้าของอะแด็ปเตอร์พอร์ต MIL-STD 188 ของ 8-พอร์ต อะซิงโครนัส เป็นไปตามส่วนเหล่านี้ของ MIL-STD 188-114 ที่พูดถึงอินเตอร์เฟสแบบ unbalanced voltage มตาราฐานนี้เมื่อ วันที่ 24 มีนาคม 1976

อะแด็ปเตอร์พอร์ตตรงกับข้อกำหนดของการทำงานสำหรับการทำงานของอะซิงโครนัส (โปรโตคอลแบบ start-stop) ดังอธิบายในมาตรฐาน EIA 232C เมื่อ ตุลาคม 1969 และในมาตรฐาน EIA 232D เมื่อมกราคม 1987

สัญญาณอินเตอร์เฟส EIA 422A ของ 8-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

สัญญาณอินเตอร์เฟส EIA 422A ต่อไปนี้ถูกนำมาใช้บนแต่ละพอร์ตของอะแด็ปเตอร์

Signal	คำนิยาม
TxA	ส่งข้อมูล
TxB	ส่งข้อมูล
RxA	รับข้อมูล
RxB	รับข้อมูล
Sig Gnd	สัญญาณกราวด์

ระดับโวลท์เตจของสัญญาณของ EIA 422A 8-พอร์ต:

line driver จะสร้างโวลท์เตจที่แตกต่างในช่วง 2 ถึง 6 โวลท์ (วันที่จุดเชื่อมต่อของตัวสร้าง) ความสูงของโวลท์เตจที่แตกต่างที่ตัวรับต้องอยู่ในช่วง 200 มิลลิโวลท์ถึง 6 โวลท์ (จัดที่จุดเชื่อมต่อของโหลด)

การวัดทำที่เทอร์มินัล A (ขั้วบวก) เมื่อเทียบกับเทอร์มินัล B (ขั้วลบ) ตารางต่อไปนี้อธิบายสถานะของสัญญาณเมื่อเทียบกับระดับของโวลท์เตจ:

ตารางที่ 118. สถานะของสัญญาณ EIA 422A 8-พอร์ต

โวลท์เตจการแลกเปลี่ยน	สถานะไบนารี	เงื่อนไขของสัญญาณ
+ โวลท์เตจ	0	Space
- โวลท์เตจ	1	Mark

อะแด็ปเตอร์ EIA 422A 8-พอร์ต อะซิงโครนัสสนับสนุนการเดินสายภายในยาวถึง 1200 เมตร (4000 ฟุต) สายที่ความยาวดังกล่าวจะไวต่อการกระชากของโวลท์เตจเนื่องจากโวลท์เตจที่ถูกเหนี่ยวนำ เช่น ฟ้าผ่า วงจรการป้องกันไฟกระชากลำดับที่สองถูกนำมาใช้บนอะแด็ปเตอร์ EIA 422A เพื่อป้องกันมันจากปัญหาไฟกระชากนี้ วงจรการป้องกันไฟกระชากถูกนำมาใช้บนสายของข้อมูลของอะแด็ปเตอร์อินเตอร์เฟส

วงจรแบบ Fail-safe ถูกเพิ่มเข้ากับขาอินพุตของตัวรับ EIA 422A แต่ละตัวเพื่อป้องกันเงื่อนไขการผิดพลาดเมื่อตัวรับไม่ได้ต่ออยู่กับไดรเวอร์ (เคเบิลเปิด) วงจร fail-safe ตั้งตัวรับเป็นสถานะ mark (ไบนารี 1) เมื่อตัวรับไม่ได้เชื่อมต่อกับไดรเวอร์

คุณลักษณะทางไฟฟ้าของพอร์ตของอะแด็ปเตอร์ EIA 422A 8-พอร์ต อะซิงโครนัส เป็นไปตามมาตรฐาน EIA 422A เมื่อธันวาคม 1978

สัญญาณอินเตอร์เฟส EIA 232 ของ 16-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

สัญญาณของอินเตอร์เฟสเหล่านี้ถูกนำไปใช้บนแต่ละพอร์ตของ 8-พอร์ต อะซิงโครนัส อะแด็ปเตอร์

สัญญาณอินเตอร์เฟสต่อไปนี้ถูกใช้บนแต่ละพอร์ตของอะแด็ปเตอร์

Signal	คำนิยาม
TxD	ส่งข้อมูล
RTS	Request To Send
CTS	Clear To Send
DSR	Data Set Ready
RxD	รับข้อมูล
DCD	Data Carrier Detect
DTR	Data Terminal Ready
RI	Ring Indicator
Sig Gnd	สัญญาณกราวด์

ระดับโวลท์เตจของสัญญาณของ EIA 232 8-พอร์ต:

สัญญาณอยู่ในสถานะ mark เมื่อโวลท์เตจบนวงจรการแลกเปลี่ยน (ถูกวัดที่จุดการเชื่อมต่อ) น้อยกว่า -3 V dc เมื่อเทียบกับสัญญาณกราวด์ สัญญาณอยู่ในสถานะ space เมื่อโวลท์เตจมากกว่า +3 V dc เมื่อเทียบกับสัญญาณกราวด์ ขอบเขตระหว่าง +3 V dc และ -3 V dc ถูกกำหนดเป็น *ขอบเขตการเปลี่ยนแปลง* และเป็นระดับที่ใช้ไม่ได้ โวลท์เตจที่น้อยกว่า -15 V dc หรือมากกว่า +15 V dc เป็นระดับที่ใช้ไม่ได้

ระหว่างการส่งข้อมูล สถานะ mark จะแทนสถานะไบนารี 1 และสถานะ space จะแทนสถานะไบนารี 0

สำหรับวงจรควบคุมอินเตอร์เฟส ฟังก์ชันจะถูกเปิดเมื่อโวลท์เตจมากกว่า +3 V dc เมื่อเทียบกับสัญญาณกราวด์ และปิดเมื่อโวลท์เตจน้อยกว่า -3 V dc เมื่อเทียบกับสัญญาณกราวด์ ดูที่ตารางต่อไปนี้สำหรับระดับของสัญญาณของ EIA 232

ตารางที่ 119. ระดับสัญญาณของ EIA 232

โวลท์เตจการแลกเปลี่ยน	สถานะไบนารี	เงื่อนไขของสัญญาณ	ฟังก์ชันการควบคุมอินเตอร์เฟส
+ โวลท์เตจ	0	Space	On
- โวลท์เตจ	1	Mark	ปิด

คุณลักษณะทางไฟฟ้าของพอร์ตของอะแด็ปเตอร์ EIA 232 8-พอร์ต อะซิงโครนัส เป็นไปตามมาตรฐาน EIA 232C เมื่อ ตุลาคม 1969 และมาตรฐาน EIA 232D เมื่อ มกราคม 1987

อะแด็ปเตอร์พอร์ตตรงกับข้อกำหนดของการทำงานสำหรับการทำงานของอะซิงโครนัส (โปรโตคอลแบบ start-stop) ดังอธิบายในมาตรฐาน EIA 232C เมื่อ ตุลาคม 1969 และในมาตรฐาน EIA 232D เมื่อมกราคม 1987

ตรรกะการควบคุม 8-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

ส่วนของตรรกะการควบคุมแบบ PAL-based จะทำงานกับกิจกรรมของฟังก์ชันของอะแด็ปเตอร์หลักทั้งหมด

มันถูกให้สัญญาณนาฬิกาด้วยตัวสร้างคลื่น square-ware 40 MHz มันจะรบกวนกับ Micro Channel และฟังก์ชันของมันรวมกับการถอดรหัสแอดเดรส การตรวจสอบแอดเดรสพาริตี การตอบสนองกับสัญญาณควบคุม I/O ที่เหมาะสม และขับสายคำร้องขออินเตอร์รัปต์ (IRQ) (หนึ่งใน 8 สาย IRQ)

ตรรกะการควบคุมเชื่อมต่อกับบล็อกของตรรกะอะแด็ปเตอร์อื่น และความสามารถนี้จัดเตรียมสายควบคุมกับแชนแนลการสื่อสาร (DUART) และตรรกะการเลือกอินเตอร์รัปต์ ตรรกะการควบคุมยังรบกวนตรรกะของไดรเวอร์บัสข้อมูล และจัดจัดเตรียมการควบคุมสำหรับทิศทางการไหลของข้อมูลและสำหรับการเลือกไบต์ข้อมูล ซึ่งจะถูกใส่บนโลคัลบัส มันจะควบคุมตัวสร้างข้อมูลพาริตี ตัวตรวจสอบพาริตี และแล็ตช์

16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์

ตระกูลของอะแด็ปเตอร์จะขึ้นอยู่กับการออกแบบการทำงานทั่วไป อย่างไรก็ตาม แต่ละคุณสมบัติของอะแด็ปเตอร์ จะถูกกำหนดโดยอินเตอร์เฟสของอุปกรณ์ที่ได้รับการสนับสนุน ตระกูลประกอบด้วย 2 อะแด็ปเตอร์ 16-พอร์ต EIA 422A อะซิงโครนัส อะแด็ปเตอร์ และ 16-พอร์ต EIA 232 อะซิงโครนัส อะแด็ปเตอร์

หมายเหตุ: ส่วนต่อไปนี้ใช้ไม่ได้กับแพล็ตฟอร์ม Itanium-based

ตระกูลของ 16-พอร์ต อะแด็ปเตอร์ จะขึ้นอยู่กับชิป dual universal asynchronous receiver and transmitter (DUART) ซึ่งจัดเตรียมแชนแนลการสื่อสารแบบซีเรียล 2 แชนแนล ข้อมูลเพิ่มเติมเกี่ยวกับชิป DUART และการทำงานของมันสามารถดูได้ในข้อมูลฮาร์ดแวร์ของ 16-พอร์ต อะซิงโครนัส อะแด็ปเตอร์

16-พอร์ต อะซิงโครนัส อะแด็ปเตอร์ - EIA 422A:

16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์ - EIA 232 จัดเตรียมการสนับสนุนสำหรับการเชื่อมต่ออุปกรณ์ EIA 232 อะซิงโครนัสแบบซีเรียล มากถึง 16 อุปกรณ์ (เครื่องพิมพ์และเทอร์มินัล) กับหน่วยของระบบ

อะแด็ปเตอร์มากถึง 8 อะแด็ปเตอร์ (การรวมกันในตระกูล) สามารถถูกใช้ในหน่วยของระบบเดียว

อะแด็ปเตอร์นี้สามารถตั้งโปรแกรมได้แบบเต็มที่และสนับสนุนการสื่อสารแบบอะซิงโครนัสเท่านั้น มันจะเพิ่มและลบ start และ stop บิต อะแด็ปเตอร์สนับสนุนพาริตีแบบ คู่ คี่ หรือไม่มีพาริตีบนข้อมูลแบบตามลำดับ ตัวสร้าง baud-rate ที่สามารถโปรแกรมได้จะยอมรับการทำงานจาก 50 ถึง 38400 bps อะแด็ปเตอร์สนับสนุนตัวอักษร 5-, 6-, 7- หรือ 8-บิต ที่มี 1, 1.5 หรือ 2 stop บิต ระบบการอินเตอร์รัปต์พาริตีจะควบคุมการส่ง การรับ ข้อผิดพลาด สถานะของสาย และการอินเตอร์รัปต์ชุดข้อมูล ตังเชื่อมต่อ 16 ตัวสำหรับการเชื่อมต่ออุปกรณ์จะถูกจัดเตรียมในการประกอบสายเคเบิล EIA 422A 16-พอร์ต

อะแด็ปเตอร์ EIA 422A 16-พอร์ตมีคุณลักษณะต่อไปนี้ :

- การ์ด Micro Channel form factor มาตรฐาน
- อัตราของข้อมูลมากถึง 38.4K bps ต่อพอร์ต
- การบัฟเฟอร์ 16 ไบต์บนการส่งและการรับ
- ตัวเชื่อมต่อ 78-ขา เอาต์พุตเดียว (สายเคเบิลอินเตอร์เฟสหลายพอร์ตจะเชื่อมต่อกับตัวเชื่อมต่อนี้)
- วงจรป้องกันไฟกระชาก
- สนับสนุนการเดินสายยาวถึง 1200 เมตร (4000 ฟุต)
- สนับสนุนสัญญาณอินเตอร์เฟส TxD และ RxD
- อินเตอร์เฟส Micro Channel slave แบบ 8-บิต/16-บิต

การติดตั้ง 16-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

16-พอร์ตอะซิงโครนัสอะแด็ปเตอร์จะพอดีกับสล็อต Micro Channel เดียวในเซิร์ฟเวอร์ เพื่อติดตั้งอะแด็ปเตอร์ ใช้ขั้นตอนเหล่านี้

1. ตรวจสอบว่าผู้ใช้ล็อกออฟจากระบบและรันคำสั่งต่อไปนี้ :

   ```
   shutdown -F
   ```

2. เมื่อคำสั่ง **shutdown** ทำเสร็จแล้ว ปิดสวิตช์ไปที่ตำแหน่ง "off"
3. เปิดตัวครอบของเซิร์ฟเวอร์ และใส่ 16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์เข้าไปที่สล็อต Micro Channel ที่ว่าง

4. ติดตัวเชื่อมต่อ 78-pin D-shell จากสาย 16-พอร์ต อินเตอร์เฟส เข้ากับ 16-พอร์ต อะแด็ปเตอร์
5. ใส่ฝาครอบกลับเข้ากับระบบ
6. เปิดสวิตช์ของระบบไปที่ตำแหน่ง On ระบบจะรู้จักและตั้งค่า 16-พอร์ต อะแด็ปเตอร์ ระหว่างกระบวนการบูต

หลังจากการบูตเรียบร้อยแล้ว ล็อกอินโดยใช้ ID ของผู้ใช้ root และใช้คำสั่งต่อไปนี้เพื่อตรวจสอบความพร้อมใช้งานของอะแด็ปเตอร์ :

```
lsdev -Cc adapter | pg
```

เฉพาะอะแด็ปเตอร์ที่อยู่ในสถานะพร้อมใช้งานที่พร้อมสำหรับใช้โดยระบบ

ถ้าอะแด็ปเตอร์ที่ถูกติดตั้งใหม่ไม่ พร้อมใช้งาน ดังนั้นให้ตรวจสอบต่อไปนี้ :

1. อะแด็ปเตอร์ถูกติดตั้งอย่างถูกต้องเข้ากับสล็อต Micro Channel
2. สายเคเบิลที่จำเป็นถูกเชื่อมต่อและติดแน่นอยู่กับที่ของมัน
3. รันคำสั่ง **errpt -a | pg** และตรวจสอบรายงานข้อผิดพลาดของระบบสำหรับปัญหาที่เกี่ยวข้องกับอะแด็ปเตอร์
4. รันคำสั่ง : **cfgmgr -v | pg** คำสั่งนี้จะพยายามตั้งค่าอะแด็ปเตอร์ใหม่โดยไม่ต้องรีบูต ดูเอาต์พุตของข้อผิดพลาดที่ถูกแสดง
5. ถ้าการรัน **cfgmgr** ล้มเหลว จำเป็นต้องรีบูต

ข้อมูลฮาร์ดแวร์ของ 16-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

อินเตอร์เฟสของระบบแสดง 3-บิตแอดเดรส และ 8-บิตข้อมูล เช่นเดียวกับสายควบคุมกับชิป ข้อมูลจากอินเตอร์เฟสของระบบจัดลำดับเพื่อส่งไปยังอุปกรณ์ภายนอก ข้อมูลแบบเรียงลำดับอาจจะรวมพาริตีบิตที่ขอบเขตของไบต์ ในทางตรงข้าม ข้อมูลจากอุปกรณ์ภายนอกจะถูกยกเลิกการเรียงลำดับสำหรับส่งไปยังอินเตอร์เฟสของระบบ ข้อมูลนี้ยังอาจรวมพาริตีบิต ซึ่งสามารถถูกเลือกที่จะตรวจสอบได้โดยที่เป็นอ็อพชัน แชนแนลสามารถทำงานในโหมด first-in-first-out (FIFO)

ในโหมด FIFO ข้อมูลมากถึง 16 ไบต์สามารถถูกเก็บบัฟเฟอร์ทั้งในตัวส่งและตัวรับ ซีเรียลอินเตอร์เฟสใช้โปรโตคอลแบบ start-stop สำหรับทั้งการส่งและรับข้อมูล ซึ่งคือ แต่ละไบต์ (รวมพาริตีบิต) ถูกจัดเป็นเฟรมโดยใช้ start และ stop บิต ซึ่งยอมให้การซิงโครไนซ์บนพื้นฐานแบบทีละตัวอักษร (ไบต์)

ชิป DUART ใช้ออสซิลเลเตอร์ 12.288 MHz เพื่อสร้างสัญญาณเวลาภายในเพื่อขับวงจรตัวส่งและตัวรับ แชนแนลสนับสนุนการทำงานแบบ full duplex ชิป DUART แปดตัวถูกนำมาใช้บนแต่ละ 16-พอร์ตอะแด็ปเตอร์

13 การลงทะเบียน system-accessible มีให้ใช้งาน คุณลักษณะที่สามารถตั้งโปรแกรมได้บนแต่ละแชนแนล รวมถึง:

- ความยาวตัวอักษร : 5,6, 7 หรือ 8 บิต
- การสร้างพาริตี/การตรวจจับ: คู่, คี่ หรือไม่มี
- จำนวนของ stop บิต: 1, 1.5 หรือ 2
- เปิดใช้งาน/ปิดใช้งาน อินเตอร์รัปต์ ข้อมูลที่ถูกรับมีให้ใช้งาน
- Transmitter holding register ว่างเปล่า
- สถานะของสาย
- ข้อผิดพลาด Overrun
- ข้อผิดพลาดพาริตี
- ข้อผิดพลาด Framing

- Break

ตารางต่อไปนี้เป็นสรุปของคุณลักษณะของพอร์ต (อินเตอร์เฟสของอุปกรณ์) สำหรับอะแด็ปเตอร์

ตารางที่ 120. คุณลักษณะของ 16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์พอร์ต

พารามิเตอร์	**EIA 232**	**EIA 422A**
โทโปโลยี	จุดต่อจุด	จุดต่อจุด
อัตราของข้อมูลสูงสุด (มาตรฐาน)	20Kbps	2Mbps
อัตราของข้อมูลสูงสุด (กว้าง)	38.4Kbps	38.4Kbps
สื่อการส่งข้อมูล	Multiconductor	Multiconductor
จำนวนสายของสายเคเบิล	5 รวมถึงสัญญาณกราวด์	5 รวมถึงสัญญาณกราวด์
ความยาวสูงสุดของสายเคเบิล	61 เมตร (200 ฟุต)	1200 เมตร < 90Kbps
ตัวเชื่อมต่อของอุปกรณ์	25-pin D	25-pin D
อินเตอร์เฟสทางไฟฟ้า	Unbalanced	Balanced
การเข้ารหัสบิต	Digital bi-level	Digital bi-level

ลำดับความสำคัญของบอร์ดอะแด็ปเตอร์ของ 16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์:

ตรรกะการเลือกอินเตอร์รัปต์จะตั้งลำดับความสำคัญสำหรับอะแด็ปเตอร์ตาม scheme ที่ระบุ

```
อะแด็ปเตอร์     ลำดับความสำคัญ
    0        สูงสุด
    1           |
    2           |
    3           |
    4           |
    5           |
    6           |
    7           |
    8           |
    9           |
    10          |
    11          |
    12          |
    13          |
    14          |
    15       ต่ำสุด
```

แชนแนล DUART พร้อมกับอินเตอร์รัปต์ที่ค้างอยู่จะได้รับบริการตาม scheme ของลำดับความสำคัญที่คงที่ ลำดับความสำคัญสูงสุดถูกกำหนดให้กับพอร์ต 0 ลำดับความสำคัญถัดไปคือพอร์ต 1 และอย่างต่อไป ลำดับความสำคัญต่ำสุดคือพอร์ต 15

ตรรกะอินเตอร์รัปต์ของ 16-พอร์ต อะซิงโครนัสอะแด็ปเตอร์:

สำหรับ 16-พอร์ต ตรรกะของอะซิงโครนัสอะแด็ปเตอร์ อินเตอร์รัปต์ถูกแบ่งเป็น ตรรกะการสร้างอินเตอร์รัปต์และ ตรรกะการเลือกอินเตอร์รัปต์

ส่วนของตรรกะทั้งสองถูกนำไปใช้บนทุก 16-พอร์ตอะแด็ปเตอร์ ตรรกะการสร้างอินเตอร์รัปต์จัดเตรียมอินเตอร์เฟสกับระบบ ตรรกะนี้จะสร้างคำร้องขออินเตอร์รัปต์ของระบบและประกอบด้วยวงจรการแบ่งใช้อินเตอร์รัปต์

ฟังก์ชันของการเลือกอินเตอร์รัปต์ใช้เพื่อระบุ 16-พอร์ต อะแด็ปเตอร์โดยการอินเตอร์รัปต์ที่มีลำดับความสำคัญสูงสุดที่ค้างอยู่ จากนั้นตรรกะจะใส่ข้อมูลการอินเตอร์รัปต์ของพอร์ตที่มีลำดับความสำคัญสูงสุดในการลงทะเบียนการเลือกอินเตอร์รัปต์ นี่ทำได้ในการดำเนินการอ่านเพียงครั้งเดียว

ตรรกะการเลือกอินเตอร์รัปต์จะเป็นหนึ่งเดียวกับ 16-พอร์ตอะแด็ปเตอร์ และไม่ควรสับสนกับตรรกกะการเลือก Micro Channel

ตรรกะการสร้างอินเตอร์รัปต์ 16-พอร์ต:

อะแด็ปเตอร์อะซิงโครนัส 16-พอร์ตใช้สายคำร้องขออินเตอร์รัปต์ 8 ระบบ

อะแด็ปเตอร์ใช้สายคำร้องขออินเตอร์รัปต์ (IRQ) 8 ระบบต่อไปนี้ :

- IRQ 3
- IRQ 5
- IRQ 9
- IRQ 10
- IRQ 11
- IRQ 12
- IRQ 14
- IRQ 15

จะมีเพียงหนึ่งสายคำร้องขอเท่านั้นที่จะแอ็คทีฟระหว่างการทำงานตามปกติ อะแด็ปเตอร์ 16-พอร์ตทั้งหมดในระบบเดียวควรใช้ระดับของอินเตอร์รัปต์เดียวกันเพื่อประสิทธิภาพของระบบที่ดีที่สุด สายที่แอ็ตทีฟถูกเลือกโดยการเขียนไปยังการลงทะเบียน POS ที่เหมาะสมระหว่างรอบของการตั้งค่า อะแด็ปเตอร์สนับสนุนการแบ่งใช้ และใช้คอนฟิกูเรชันของตัวเชื่อมต่อแบบเปิดดังที่ถูกกำหนดในสถาปัตยกรรม Micro Channel ในการกำหนดนี้สายอินเตอร์รัปต์จะถูกดึงให้สูงโดย pull-up resistor ของระบบ อะแด็ปเตอร์จะดึงสายให้ต่อเพื่อระบุคำร้องขออินเตอร์รัปต์ที่แอ็คทีฟ

ตรรกะการเลือกอินเตอร์รัปต์ 16-พอร์ต:

ตรรกะการเลือแอินเตอร์รัปต์จะตรวจสอบพาริตีสำหรับเซอร์วิสของซอฟต์แวร์ เมื่ออะแด็ปเตอร์ 8-พอร์ต หรือ 16-พอร์ต 2 อะแด็ปเตอร์หรือมากกว่าสร้างอินเตอร์รัปต์

อะแด็ปเตอร์ 8-พอร์ต หรือ 16-พอร์ต มากถึง 8 อะแด็ปเตอร์สามารถอยู่ร่วมกันและทำงานพร้อมกันในระบบเดียว ตรรกะนี้จัดเตรียมอะแด็ปเตอร์และการระบุพอร์ตให้กับระบบ เช่นเดียวกับชนิดของอินเตอร์รัปต์ในการอ่านเพียวครั้งเดียว หลังจากคำร้องของอินเตอร์รัปต์ถูกตรวจพบ ระบบจะอ่านการลงทะเบียนการเลือกอินเตอร์รัปต์แบบ 16-บิต ซึ่งอยู่ที่ I/O แอดเดรส 0130

สัญญาณอินเตอร์เฟส EIA 232 ของ 16-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

สัญญาณอินเตอร์เฟสต่อไปนี้ถูกใช้บนแต่ละพอร์ตของ 16-พอร์ตอะซิงโครนัสอะแด็ปเตอร์

Signal	คำนิยาม
TxD	ส่งข้อมูล
DCD	Data carrier detect
DTR	Data terminal ready
RxD	รับข้อมูล
Sig Gnd	สัญญาณกราวด์

ระดับโวลท์เตจของสัญญาณของ EIA 232 16-พอร์ต:

สัญญาณอยู่ในสถานะ mark เมื่อโวลท์เตจบนวงจรการแลกเปลี่ยน (ถูกวัดที่จุดการเชื่อมต่อ) น้อยกว่า -3 V dc เมื่อเทียบกับสัญญาณกราวด์ สัญญาณอยู่ในสถานะ space เมื่อโวลท์เตจมากกว่า +3 V dc เมื่อเทียบกับสัญญาณกราวด์ ขอบเขตระหว่าง +3 V dc และ -3 V dc ถูกกำหนดเป็นขอบเขตของการเปลี่ยนแปลงและเป็นระดับที่ใช้ไม่ได้ โวลท์เตจที่น้อยกว่า -15 V dc หรือมากกว่า +15 V dc เป็นระดับที่ใช้ไม่ได้

ระหว่างการส่งข้อมูล สถานะ mark จะแทนสถานะไบนารี 1 และสถานะ space จะแทนสถานะไบนารี 0

สำหรับวงจรควบคุมอินเตอร์เฟส ฟังก์ชันจะถูกเปิดเมื่อโวลท์เตจมากกว่า +3 V dc เมื่อเทียบกับสัญญาณกราวด์ และปิดเมื่อโวลท์เตจน้อยกว่า -3 V dc เมื่อเทียบกับสัญญาณกราวด์ ดูที่ตารางต่อไปนี้สำหรับระดับของสัญญาณของ EIA 232

ตารางที่ 121. ระดับสัญญาณของ EIA 232

โวลท์เตจการแลกเปลี่ยน	สถานะไบนารี	เงื่อนไขของสัญญาณ	ฟังก์ชันการควบคุมอินเตอร์เฟส
+ โวลท์เตจ	0	Space	On
- โวลท์เตจ	1	Mark	ปิด

คุณลักษณะทางไฟฟ้าของพอร์ตของอะแด็ปเตอร์ EIA 232 16-พอร์ต อะซิงโครนัส เป็นไปตามมาตรฐาน EIA 232C เมื่อตุลาคม 1969 และมาตรฐาน EIA 232D เมื่อ มกราคม 1987

อะแด็ปเตอร์พอร์ตตรงกับข้อกำหนดของการทำงานสำหรับการทำงานของอะซิงโครนัส (โปรโตคอลแบบ start-stop) ดังอธิบายในมาตรฐาน EIA 232C เมื่อ ตุลาคม 1969 และในมาตรฐาน EIA 232D เมื่อมกราคม 1987

สัญญาณอินเตอร์เฟส EIA 422A ของ 16-พอร์ตอะซิงโครนัสอะแด็ปเตอร์:

สัญญาณอินเตอร์เฟส EIA 422A เหล่านี้ถูกนำมาใช้บนแต่ละพอร์ตของอะแด็ปเตอร์อะซิงโครนัส 16-พอร์ต

Signal	คำนิยาม
TxA	ส่งข้อมูล
TxB	ส่งข้อมูล
RxA	รับข้อมูล
RxB	รับข้อมูล
Sig Gnd	สัญญาณกราวด์

ระดับโวลท์เตจของสัญญาณของ EIA 422A 16-พอร์ต:

line driver จะสร้างโวลท์เตจที่แตกต่างในช่วง 2 ถึง 6 โวลท์ (วัดที่จุดเชื่อมต่อของตัวสร้าง) ความสูงของโวลท์เตจที่แตกต่างที่ตัวรับต้องอยู่ในช่วง 200 มิลลิโวลท์ถึง 6 โวลท์ (จัดที่จุดเชื่อมต่อของโหลด)

การวัดทำที่เทอร์มินัล A (ขั้วบวก) เมื่อเทียบกับเทอร์มินัล B (ขั้วลบ) ตารางต่อไปนี้อธิบายสถานะของสัญญาณเมื่อเทียบกับระดับของโวลท์เตจ:

ตารางที่ 122. สถานะของสัญญาณ EIA 422A 16-พอร์ต

โวลท์เตจการแลกเปลี่ยน	สถานะไบนารี	เงื่อนไขของสัญญาณ
+ โวลท์เตจ	0	Space
- โวลท์เตจ	1	Mark

อะแด็ปเตอร์ EIA 422A 16-พอร์ต อะซิงโครนัสสนับสนุนการเดินสายภายในยาวถึง 1200 เมตร (4000 ฟุต) สายที่ความยาวดังกล่าวจะไวต่อการกระชากของโวลท์เตจเนื่องจากโวลท์เตจที่ถูกเหนี่ยวนำ เช่น ฟ้าผ่า วงจรการป้องกันไฟกระชากลำดับที่สองถูกนำมาใช้บนอะแด็ปเตอร์ EIA 422A เพื่อป้องกันมันจากปัญหาไฟกระชากนี้ วงจรการป้องกันไฟกระชากถูกนำมาใช้บนสายของข้อมูลของอะแด็ปเตอร์อินเตอร์เฟส

วงจรแบบ Fail-safe ถูกเพิ่มเข้ากับขาอินพุตของตัวรับ EIA 422A แต่ละตัวเพื่อป้องกันเงื่อนไขการผิดพลาดเมื่อตัวรับไม่ได้ต่ออยู่กับไดรเวอร์ (เคเบิลเปิด) วงจร fail-safe ตั้งตัวรับเป็นสถานะ mark (ไบนารี 1) เมื่อตัวรับไม่ได้เชื่อมต่อกับไดรเวอร์

คุณลักษณะทางไฟฟ้าของพอร์ตของอะแด็ปเตอร์ EIA 422A 16-พอร์ต อะซิงโครนัส เป็นไปตามมาตรฐาน EIA 422A เมื่อธันวาคม 1978

ตารางการแปลง ASCII เลขฐานสิบ เลขฐานแปด และไบนารี

ข้อมูลที่เป็นประโยชน์สำหรับการแปลงค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี สามารถถูกอ้างถึงในตารางนี้

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
null	0	0	0	0
เริ่มต้นของส่วนหัว	1	1	1	1
เริ่มต้นข้อความ	2	2	2	10
สิ้นสุดของข้อความ	3	3	3	11
สิ้นสุดการส่งข้อมูล	4	4	4	100
enquire	5	5	5	101
acknowledge	6	6	6	110
bell	7	7	7	111
แบ็คสเปซ	8	8	10	1000
แท็บแนวนอน	9	9	11	1001
linefeed	10	A	12	1010
แท็บแนวตั้ง	11	B	13	1011
form feed	12	C	14	1100
ปัดแคร่	13	D	15	1101

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี (ต่อ)

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
shift out	14	E	16	1110
shift in	15	F	17	1111
data link escape	16	10	20	10000
ตัวควบคุมอุปกรณ์ 1/Xon	17	11	21	10001
ตัวควบคุมอุปกรณ์ 2	18	12	22	10010
ตัวควบคุมอุปกรณ์ 3/Xoff	19	13	23	10011
ตัวควบคุมอุปกรณ์ 4	20	14	24	10100
negative acknowledge	21	15	25	10101
synchronous idle	22	16	26	10110
สิ้นสุดของการส่งบล็อก	23	17	27	10111
ยกเลิก	24	18	30	11000
สิ้นสุดส่วนที่อยู่ตรงกลาง	25	19	31	11001
สิ้นสุดไฟล์/ แทนที่	26	1A	32	11010
escape	27	1B	33	11011
ตัวแยกไฟล์	28	1C	34	11100
ตัวแยกกลุ่ม	29	1D	35	11101
ตัวแยกเร็กคอร์ด	30	1E	36	11110
ตัวแยกหน่วย	31	1F	37	11111
พื้นที่ว่าง	32	20	40	100000
!	33	21	41	100001
″	34	22	42	100010
#	35	23	43	100011
$	36	24	44	100100
%	37	25	45	100101
&	38	26	46	100110
’	39	27	47	100111
(	40	28	50	101000
)	41	29	51	101001
*	42	2A	52	101010
+	43	2B	53	101011
,	44	2C	54	101100

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี (ต่อ)

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
-	45	2D	55	101101
.	46	2E	56	101110
/	47	2F	57	101111
0	48	30	60	110000
1	49	31	61	110001
2	50	32	62	110010
3	51	33	63	110011
4	52	34	64	110100
5	53	35	65	110101
6	54	36	66	110110
7	55	37	67	110111
8	56	38	70	111000
9	57	39	71	111001
:	58	3A	72	111010
;	59	3B	73	111011
<	60	3C	74	111100
=	61	3D	75	111101
>	62	3E	76	111110
?	63	3F	77	111111
@	64	40	100	1000000
A	65	41	101	1000001
B	66	42	102	1000010
C	67	43	103	1000011
D	68	44	104	1000100
E	69	45	105	1000101
F	70	46	106	1000110
G	71	47	107	1000111
H	72	48	110	1001000
I	73	49	111	1001001
J	74	4A	112	1001010
K	75	4B	113	1001011

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี (ต่อ)

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
L	76	4C	114	1001100
M	77	4D	115	1001101
N	78	4E	116	1001110
O	79	4F	117	1001111
P	80	50	120	1010000
Q	81	51	121	1010001
R	82	52	122	1010010
S	83	53	123	1010011
T	84	54	124	1010100
U	85	55	125	1010101
V	86	56	126	1010110
W	87	57	127	1010111
X	88	58	130	1011000
Y	89	59	131	1011001
Z	90	5A	132	1011010
[	91	5B	133	1011011
\	92	5C	134	1011100
]	93	5D	135	1011101
^	94	5E	136	1011110
_	95	5F	137	1011111
`	96	60	140	1100000
a	97	61	141	1100001
b	98	62	142	1100010
c	99	63	143	1100011
d	100	64	144	1100100
e	101	65	145	1100101
f	102	66	146	1100110
g	103	67	147	1100111
h	104	68	150	1101000
i	105	69	151	1101001
j	106	6A	152	1101010

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี (ต่อ)

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
k	107	6B	153	1101011
l	108	6C	154	1101100
m	109	6D	155	1101101
n	110	6E	156	1101110
o	111	6F	157	1101111
p	112	70	160	1110000
q	113	71	161	1110001
r	114	72	162	1110010
s	115	73	163	1110011
t	116	74	164	1110100
u	117	75	165	1110101
v	118	76	166	1110110
w	119	77	167	1110111
x	120	78	170	1111000
y	121	79	171	1111001
z	122	7A	172	1111010
{	123	7B	173	1111011
|	124	7C	174	1111100
}	125	7D	175	1111101
~	126	7E	176	1111110
DEL	127	7F	177	1111111
	128	80	200	10000000
	129	81	201	10000001
	130	82	202	10000010
	131	83	203	10000011
	132	84	204	10000100
	133	85	205	10000101
	134	86	206	10000110
	135	87	207	10000111
	136	88	210	10001000
	137	89	211	10001001

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี (ต่อ)

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
	138	8A	212	10001010
	139	8B	213	10001011
	140	8C	214	10001100
	141	8D	215	10001101
	142	8E	216	10001110
	143	8F	217	10001111
	144	90	220	10010000
	145	91	221	10010001
	146	92	222	10010010
	147	93	223	10010011
	148	94	224	10010100
	149	95	225	10010101
	150	96	226	10010110
	151	97	227	10010111
	152	98	230	10011000
	153	99	231	10011001
	154	9A	232	10011010
	155	9B	233	10011011
	156	9C	234	10011100
	157	9D	235	10011101
	158	9E	236	10011110
	159	9F	237	10011111
	160	A0	240	10100000
	161	A1	241	10100001
	162	A2	242	10100010
	163	A3	243	10100011
	164	A4	244	10100100
	165	A5	245	10100101
	166	A6	246	10100110
	167	A7	247	10100111
	168	A8	250	10101000

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี (ต่อ)

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
	169	A9	251	10101001
	170	AA	252	10101010
	171	AB	253	10101011
	172	AC	254	10101100
	173	AD	255	10101101
	174	AE	256	10101110
	175	AF	257	10101111
	176	B0	260	10110000
	177	B1	261	10110001
	178	B2	262	10110010
	179	B3	263	10110011
	180	B4	264	10110100
	181	B5	265	10110101
	182	B6	266	10110110
	183	B7	267	10110111
	184	B8	270	10111000
	185	B9	271	10111001
	186	BA	272	10111010
	187	BB	273	10111011
	188	BC	274	10111100
	189	BD	275	10111101
	190	BE	276	10111110
	191	BF	277	10111111
	192	C0	300	11000000
	193	C1	301	11000001
	194	C2	302	11000010
	195	C3	303	11000011
	196	C4	304	11000100
	197	C5	305	11000101
	198	C6	306	11000110
	199	C7	307	11000111

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี (ต่อ)

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
	200	C8	310	11001000
	201	C9	311	11001001
	202	CA	312	11001010
	203	CB	313	11001011
	204	CC	314	11001100
	205	CD	315	11001101
	206	CE	316	11001110
	207	CF	317	11001111
	208	D0	320	11010000
	209	D1	321	11010001
	210	D2	322	11010010
	211	D3	323	11010011
	212	D4	324	11010100
	213	D5	325	11010101
	214	D6	326	11010110
	215	D7	327	11010111
	216	D8	330	11011000
	217	D9	331	11011001
	218	DA	332	11011010
	219	DB	333	11011011
	220	DC	334	11011100
	221	DD	335	11011101
	222	DE	336	11011110
	223	DF	337	11011111
	224	E0	340	11100000
	225	E1	341	11100001
	226	E2	342	11100010
	227	E3	343	11100011
	228	E4	344	11100100
	229	E5	345	11100101
	230	E6	346	11100110

ตารางที่ 123. การแปลงระหว่างค่า ASCII เลขฐานสิบ เลขฐานแปด และไบนารี (ต่อ)

ASCII	เลขฐานสิบ	เลขฐานสิบหก	เลขฐานแปด	ไบนารี
	231	E7	347	11100111
	232	E8	350	11101000
	233	E9	351	11101001
	234	EA	352	11101010
	235	EB	353	11101011
	236	EC	354	11101100
	237	ED	355	11101101
	238	EE	356	11101110
	239	EF	357	11101111
	240	F0	360	11110000
	241	F1	361	11110001
	242	F2	362	11110010
	243	F3	363	11110011
	244	F4	364	11110100
	245	F5	365	11110101
	246	F6	366	11110110
	247	F7	367	11110111
	248	F8	370	11111000
	249	F9	371	11111001
	250	FA	372	11111010
	251	FB	373	11111011
	252	FC	374	11111100
	253	FD	375	11111101
	254	FE	376	11111110
	255	FF	377	11111111

uDAPL (ระดับผู้ใช้ Direct Access Programming Library)

uDAPL (ผู้ใช้ Direct Access Programming Library) คือกรอบงานการเข้าถึงโดยตรงที่ต้องถูกรันตามการส่งผ่านซึ่งสนับสนุนการเข้าถึงข้อมูลโดยตรง เช่น InfiniBand , RNIC เป็นต้น

DAT Collaborative ระบุ **uDAPL** API http://www.datcollaborative.org

โค้ดเบส uDAPL จาก Open Fabrics จะต่อกับ AIX และสนับสนุนผ่าน GX++ HCA และการ์ด 4X DDR Expansion (CFFh) อะแด็ปเตอร์ InfiniBand

uDAPL 1.2 เวอร์ชันได้รับการสนับสนุนบน AIX 6.1 ที่มี 6100-06 และอื่นๆ อิมเมจการติดตั้ง **uDAPL** ถูกจัดส่งมาบนแพ็กส่วนขยายในรูปของ *udapl.rte* อิมเมจนี้ จัดส่งไฟล์ส่วนหัว DAT ซึ่งวางอยู่ภายใต้ **/usr/include/dat** อิมเมจการติดตั้งยังจัดส่งไลบรารีสองชุด คือ *libdat.a* และ *libdapl.a*

แอ็พพลิเคชันสอดแทรกไฟล์ส่วนหัว DAT และลิงก์กับไลบรารี DAT (*libdat.a* ใน **/usr/include/dat**) เลเยอร์ DAT กำหนดไลบรารีที่ระบุเฉพาะการส่งผ่านอย่างเหมาะสม

AIX **uDAPL** Provider ลงทะเบียนตนเองด้วยรีจิสเตอร์ DAT โดยใช้รายการ *dat.conf* ไฟล์ */etc/dat.conf* ถูกจัดส่งด้วยรายการดีฟอลต์ และไฟล์มีรายละเอียดตามรูปแบบของรายการ

สำหรับวัตถุประสงค์ในการดีบัก ไลบรารี **uDAPL** สนับสนุนการติดตามของระบบ AIX **การติดตามระบบ uDAPL** ฮุค ids สอดแทรก 5C3 (สำหรับเหตุการณ์ DAPL), 5C4 (สำหรับเหตุการณ์ข้อผิดพลาด DAPL), 5C7 (สำหรับเหตุการณ์ DAT) และ 5C8 (สำหรับเหตุการณ์ข้อผิดพลาด DAT) ระดับการติดตามเริ่มต้น สามารถแก้ไขได้โดยใช้ตัวแปรสภาวะแวดล้อม *DAT_TRACE_LEVEL* และ *DAPL_TRACE_LEVEL* ซึ่งสามารถใช้ค่าตัวเลขที่มีช่วงจาก 0 ถึง 10 จำนวนของเหตุการณ์และจำนวนของข้อมูลการติดตามเพิ่มขึ้นด้วยระดับ ซึ่งเป็นระดับของการติดตามคีย์

```
TRC_LVL_ERROR  = 1,
TRC_LVL_NORMAL = 3,
TRC_LVL_DETAIL = 7
```

คุณลักษณะความสามารถในการให้บริการ AIX มาตรฐาน เช่น บันทึกข้อผิดพลาด AIX อาจมีประโยชน์สำหรับการกำหนดปัญหา และคุณลักษณะความสามารถในการให้บริการ ของเลเยอร์การส่งผ่าน เช่น คำสั่ง ibstat และการติดตามคอมโพเนนต์ InfiniBand และยังมีประโยชน์สำหรับปัญหาในการวินิจฉัย

DAT APIs ส่งคืนโค้ดส่งคืนมาตรฐานที่สามารถถอดรหัส ด้วยวิธีใช้ไฟล์ */usr/include/dat/dat_error.h* คำอธิบายโดยละเอียดของโค้ดส่งคืน ถูกวางอยู่ในข้อกำหนดคุณสมบัติ **uDAPL** จาก DAT Collaborative

“อินเตอร์เน็ตโปรโตคอลบน InfiniBand (IPoIB)” ในหน้า 418

uDAPL APIs ที่สนับสนุนใน AIX

หากไม่มี **uDAPL** APIs ที่ระบุไว้โดย DAT Collaborative มี APIs ไม่กี่ตัวที่ไม่สนับสนุนใน AIX

ต่อไปนี้คือ APIs ที่การนำ **uDAPL** ไปใช้งานในอุตสาหกรรมทั่วไป ไม่ได้รับการสนับสนุน และจะไม่สนับสนุนบน AIX

ไอเท็ม	คำอธิบาย
dat_cr_handoff	//ใน DAT 1.2
dat_ep_create_with_srq	//ใน DAT 1.2
dat_ep_recv_query	//ใน DAT 1.2
dat_ep_set_watermark	//ใน DAT 1.2
dat_srq_create	//ใน DAT 1.2
dat_srq_post_recv	//ใน DAT 1.2
dat_srq_resize	//ใน DAT 1.2
dat_srq_set_lw	//ใน DAT 1.2
dat_srq_free	//ใน DAT 1.2
dat_srq_query	//ใน DAT 1.2

APIs เพิ่มเติมที่ AIX ไม่สนับสนุนคือ

- dat_lmr_sync_rdma_read
- dat_lmr_sync_rdma_write
- dat_registry_add_provider
- dat_registry_add_provider

สำหรับ APIs ที่ไม่สนับสนุนทั้งหมด AIX เป็นไปตามกลไกที่ระบุไว้ซึ่งกล่าวถึงในข้อกำหนดคุณสมบัติของ DAT เพื่อระบุ การขาดแคลนของการสนับสนุน เหล่านี้ประกอบด้วยค่าแอ็ตทริบิวต์ (เช่น max_srq ที่มีค่าศูนย์) และโค้ดส่งคืนที่ระบุไว้ (เช่น DAT_MODEL_NOT_SUPPORTED) ความสอดคล้องกันกับการนำไปใช้งานในอุตสาหกรรมและข้อกำหนดคุณสมบัติ DAT DAT_NOT_IMPLEMENTED อาจยังส่งคืนฟังก์ชัน ที่ไม่ได้รับการสนับสนุน

ส่วนสนับสนุน RMR ที่เกี่ยวข้องกับ APIs เช่น *dat_rmr_create* ,*dat_rmr_bind* ,*dat_rmr_free* และ *dat_rmr_query* ขึ้นอยู่กับความสามารถของ HCA และความสำเร็จหรือความล้มเหลว ที่ถูกกำหนดไว้โดยกรอบงาน IB ในปัจจุบัน GX++ HCA และการ์ด 4X DDR Expansion (CFFh) อะแด็ปเตอร์ InfiniBand ไม่สนับสนุน การดำเนินการ RMR เหล่านี้

“uDAPL (ระดับผู้ใช้ Direct Access Programming Library)” ในหน้า 734

“แอ็ตทริบิวต์ที่ระบุเฉพาะผู้ขายสำหรับ uDAPL”

“อินเตอร์เน็ตโปรโตคอลบน InfiniBand (IPoIB)” ในหน้า 418

แอ็ตทริบิวต์ที่ระบุเฉพาะผู้ขายสำหรับ uDAPL

มีแอ็ตทริบิวต์ที่ระบุเฉพาะผู้ขายที่สนับสนุนใน AIX ชื่อแอ็ตทริบิวต์ คือ **delayed_ack_supported , vendor_extension, vendor_ext_version, debug_query** และ **debug_modify**

delayed_ack_supported

ผู้ให้บริการ AIX สำหรับการส่งผ่าน InfiniBand (IB) ประกอบด้วยแอ็ตทริบิวต์อะแด็ปเตอร์อินเตอร์เฟส (IA®) เฉพาะผู้จำหน่ายซึ่งมีชื่อว่า **delayed_ack_supported** ค่าของแอ็ตทริบิวต์นี้ คือ **true** หรือ **false** อย่างใดอย่างหนึ่ง เมื่อเป็น **true** จุดปลายที่เชื่อมโยงกับ IA นี้มีแอ็ตทริบิวต์เฉพาะผู้ให้บริการซึ่งสามารถปรับเปลี่ยนได้ ที่ชื่อ **delayed_ack** เมื่อแอ็ตทริบิวต์ **delayed_ack_supported** มีค่า **false** แอ็ตทริบิวต์ที่ระบุเฉพาะผู้ให้บริการ **delayed_ack** ของจุดปลาย ไม่สามารถแก้ไขได้ ค่าดีฟอลต์ของแอ็ตทริบิวต์ **delayed_ack** ของจุดปลายคือ **false** การตั้งค่าเป็น **true** (via dat_ep_modify) เปิดใช้งานคุณลักษณะพิเศษ ack ที่หน่วงเวลาของ IB Host Channel Adapter (HCA) สำหรับคู่ของคิว IB เฉพาะซึ่งเชื่อมโยงกับจุดปลาย คุณลักษณะของฮาร์ดแวร์นี้ไม่ได้ใช้โดย HCAs ทั้งหมด และไม่พร้อมใช้งานสำหรับ IAs ทั้งหมด การเปิดใช้งานคุณลักษณะนี้อาจทำให้ HCA หน่วงเวลาการส่งการตอบรับจนกว่าการดำเนินการถ่ายโอนข้อมูล สามารถมองเห็นได้ในหน่วยความจำระบบของเซิร์ฟเวอร์ ซึ่งมีความหมายที่สำคัญ มากกว่าที่จะถูกจัดเตรียมไว้ในข้อกำหนดคุณสมบัติของ IB ด้วยต้นทุน ของการเพิ่มแฝงที่เล็กมาก

vendor_extension, vendor_ext_version, debug_query และ debug_modify

สำหรับวัตถุประสงค์ในการดีบัก ไลบรารี **uDAPL** สนับสนุนการติดตามของระบบ AIX ระดับ การติดตามเริ่มต้นสามารถแก้ไขได้โดยใช้ตัวแปรสภาวะแวดล้อม *DAT_TRACE_LEVEL* และ *DAPL_TRACE_LEVEL* ในการเปลี่ยนระดับของการติดตามเหล่านี้แบบไดนามิกผ่าน API เราจัดเตรียมส่วนสนับสนุน ระดับของการติดตามบน AIX ไว้ในการตรวจสอบว่าไลบรารีมีส่วน

สนับสนุนระดับการติดตามแบบไดนามิก แอ็พพลิเคชัน สามารถเคียวรีแอ็ตทริบิวต์ IA เฉพาะผู้จำหน่าย นั่นคือ **vendor_extension** สำหรับการส่งกลับจากเคียวรี การมีอยู่ของแอ็ตทริบิวต์ **vendor_extension** จะบ่งชี้ถึงส่วนสนับสนุนระดับของการติดตามแบบไดนามิก ค่าของแอ็ตทริบิวต์ จะมีค่า **true** แต่ไม่คำนึงถึงว่า การมีอยู่ของแอ็ตทริบิวต์ จะบ่งชี้ถึงส่วนสนับสนุน เมื่อแอ็ตทริบิวต์ **vendor_extension** มีอยู่ แอ็พพลิเคชันสามารถรับตัวชี้ฟังก์ชันไปยัง **dat_trclvl_query( )** และ **dat_trclvl_modify( )** โดย เคียวรีแอ็ตทริบิวต์ IA เฉพาะผู้จำหน่าย นั่นคือ **debug_query** และ **debug_modify** ค่าของแอ็ตทริบิวต์เหล่านี้จะมีตัวชี้ไปยัง ฟังก์ชันที่สอดคล้องกัน เพื่อทำให้อินเตอร์เฟส **vendor_extension** นี้สามารถขยายเพิ่มได้ สำหรับการใช้งานในอนาคต เรามีแอ็ตทริบิวต์ IA เฉพาะผู้จำหน่ายตัวอื่น นั่นคือ **vendor_ext_version** เนื่องจากเราจะสนับสนุนเฉพาะเวอร์ชันที่ถูกต้องเพียงหนึ่งเวอร์ชันเท่านั้น ค่าของแอ็ตทริบิวต์นี้ต้องเป็น **1.0** หากแอ็ตทริบิวต์ **vendor_extension** ไม่มีอยู่ แอ็พพลิเคชันจะไม่สามารถแก้ไขระดับของการติดตามแบบไดนามิกได้

ตัวอย่างของวิธีการจัดการกับแอ็ตทริบิวต์เหล่านี้จะสอดแทรกอยู่ในโค้ดตัวอย่าง **uDAPL** ที่ติดตั้งไว้พร้อมกับการนำ AIX ไปใช้งาน

“uDAPL (ระดับผู้ใช้ Direct Access Programming Library)” ในหน้า 734

“อินเตอร์เน็ตโปรโตคอลบน InfiniBand (IPoIB)” ในหน้า 418

การสนับสนุนอะแด็ปเตอร์ PCIe2 10 GbE RoCE

อะแด็ปเตอร์ PCIe2 10GbE RDMA Over Converged Ethernet (RoCE) ได้รับการสนับสนุนครั้งแรกในระบบปฏิบัติการ AIX เป็นอุปกรณ์ที่มีความสามารถ remote direct memory access (RDMA) เท่านั้น ซอฟต์แวร์ที่สนับสนุนเป็น ซอฟต์แวร์ของ IBM ตามสแต็ก AIX InfiniBand การสนับสนุนนี้เรียกว่า AIX RoCE AIX 7 ที่มี 7100-02 หรือสูงกว่า สนับสนุนอะแด็ปเตอร์ในสองโหมด ซึ่งเป็น AIX RoCE และการสนับสนุนอีเทอร์เน็ต 10G ยังเรียกใช้ การ์ดอินเตอร์เฟสเครือข่ายด้วย (AIX NIC) ตอนนี้ AIX 7 ที่มี 7100-03 ใหม่ สนับสนุน RDMA พร้อมกับโหมด NIC และ OpenFabrics Enterprise Distribution (OFED) Host bus adapter (HBA) ซึ่งไม่มีอยู่ในเวอร์ชันก่อนหน้านี้ของระบบปฏิบัติการ AIX จะจัดการเลือกโหมดที่ เปิดใช้งาน

ตารางต่อไปนี้แสดงวิวัฒนาการของ ซอฟต์แวร์อะแด็ปเตอร์ PCIe2 10GbE:

ระดับ AIX	MODE 1	MODE 2
ก่อน AIX 7 ที่มี 7100-02	AIX RoCE	NA
AIX 7 ที่มี 7100-02	AIX RoCE	AIX NIC
AIX 7 ที่มี 7100-03	AIX RoCE	AIX NIC + OFED RoCE

เมื่อต้องการดาวน์โหลดไดรเวอร์อุปกรณ์ล่าสุดสำหรับอะแด็ปเตอร์นี้ ให้ทำ ขั้นตอนต่อไปนี้:

1. ไปยังเว็บไซต์ IBM (www.ibm.com)
2. คลิก **การสนับสนุนและดาวน์โหลด**
3. ดาวน์โหลดเฟิร์มแวร์ล่าสุดไปยังตำแหน่งโฮสต์ AIX (`/etc/microcode`)
4. รันเครื่องมือ **diag** เพื่ออัพเดตเฟิร์มแวร์โดยเลือก โพรซีเดอร์อย่างใดอย่างหนึ่งต่อไปนี้:
   - โพรซีเดอร์พาธสั้น
     a. ป้อนคำสั่งต่อไปนี้:

     ```
     *diag -d entX -T download
     ```

     หมายเหตุ: แทนที่ **entX** ด้วย **roceX** ถ้า คุณกำลังใช้สแต็ก RoCE จากเวอร์ชันก่อนหน้านี้

b. เลือกไมโครโค้ดที่บันทึกไว้ในไดเร็กทอรี `/etc/microcode`

- โพรซีเดอร์พาธยาว
  a. ป้อนคำสั่งต่อไปนี้:

     ```
     *diag
     ```

  b. คลิก: **การเลือกภารกิจ** > **ภารกิจไมโครโค้ด** > **ดาวน์โหลดไมโครโค้ด**.
  c. เลือก **entX** หรือ **roceX**
  d. เลือกไมโครโค้ดที่บันทึกไว้ในไดเร็กทอรี `/etc/microcode`

โดยดีฟอลต์ อะแด็ปเตอร์มีการกำหนดคอนฟิกเพื่อสนับสนุนโหมด AIX RoCE ทำขั้นตอนในส่วน “AIX NIC + OFED RDMA”เพื่อเปลี่ยนเป็นโหมดอื่น

AIX NIC + OFED RDMA

ตั้งแต่ AIX 7 ที่มี 7100-02 สามารถกำหนดคอนฟิกอะแด็ปเตอร์ PCIe2 10 GbE RoCE เพื่อรันในคอนฟิกูเรชัน AIX NIC ตั้งแต่ AIX 7 ที่มี 7100-03 มีการเพิ่มการทำงาน OFED RDMA ลงในคอนฟิกูเรชัน AIX NIC ด้วย ถ้าคุณไม่มีแอ็พพลิเคชันที่มุ่งเน้นเครือข่าย ซึ่งได้รับประโยชน์จาก RDMA คุณสามารถใช้อะแด็ปเตอร์เป็น Network Interface Card (NIC) เท่านั้น

เมื่อต้องการใช้อะแด็ปเตอร์ PCIe2 10 GbE RoCE ในคอนฟิกูเรชัน AIX NIC + OFED RoCE หรือคอนฟิกูเรชัน AIX RoCE ชุดไฟล์ต่อไปนี้เป็นสิ่งจำเป็นและมีอยู่ในแผ่นซีดีของระบบปฏิบัติการพื้นฐาน AIX 7 ที่มี 7100-03

devices.ethernet.mlx
ไดรเวอร์อุปกรณ์หลัก Converged Ethernet Adapter (mlxentdd) เพื่อสนับสนุน คอนฟิกูเรชัน AIX NIC + OFED RoCE

devices.pciex.b315506b3157265
การสนับสนุนแพ็กเกจสำหรับ NGP ITE Converge Ethernet Adapter ASIC2

devices.pciex.b3155067b3157365
การสนับสนุนแพ็กเกจสำหรับ NGP ITE Converge Ethernet Adapter ASIC1

devices.pciex.b315506714101604
แพ็กเกจสำหรับ Mellanox 2 Ports 10 GbE Converge Ethernet Adapter ที่มีเครื่องรับส่ง small form factor pluggable (SFP+)

devices.pciex.b315506714106104
แพ็กเกจสำหรับ Mellanox 2 Ports 10 GbE Converge Ethernet Adapter ที่สนับสนุนเครื่องรับส่ง SFP+ ใดๆ

devices.common.IBM.ib
ไดรเวอร์อุปกรณ์ ICM ที่จำเป็นในการใช้คอนฟิกูเรชัน AIX RoCE

devices.pciex.b3154a63
ไดรเวอร์อุปกรณ์ Mellanox 10 GbE Converge Ethernet Adapter ที่ จำเป็นในการใช้คอนฟิกูเรชัน AIX RoCE

ofed.core
ชุดไฟล์ OFED Core Runtime Environment ที่ต้องใช้เฉพาะถ้า ต้องการ OFED RDMA

หลังจากอัพเดตชุดไฟล์ AIX RoCE ที่มีอยู่ด้วยชุดไฟล์ใหม่แล้ว ทั้งอุปกรณ์ roce และ ent อาจจะมีการกำหนดคอนฟิกแล้ว ถ้าทั้งสองอุปกรณ์มีการกำหนดคอนฟิกเมื่อคุณรันคำสั่ง **lsdev** บนอะแด็ปเตอร์ ให้ทำขั้นตอนต่อไปนี้:

1. ลบอินสแตนซ์ของ *roceX* ที่เกี่ยวข้อง กับ PCIe2 10 GbE RoCE Adapter โดยป้อนคำสั่งต่อไปนี้:

   ```
   # rmdev -dl roce0[, roce1][, roce2,...]
   ```

2. ลบอินสแตนซ์ของ *entX* ที่เกี่ยวข้อง กับ PCIe2 10 GbE RoCE Adapter โดยป้อนคำสั่งต่อไปนี้:

   ```
   # rmdev -dl ent1[,ent2][, ent3...]
   ```

3. ถ้ามีหนึ่งหรือหลาย converged host bus adapters (hbaX) ที่เกี่ยวข้องกับ PCIe2 10 GbE RoCE Adapter ให้ลบออกโดยป้อนคำสั่งต่อไปนี้:

   ```
   # rmdev -dl hba0[, hba1][,hba2...]
   ```

4. รันโปรแกรมจัดการคอนฟิกูเรชันเพื่อรวมการเปลี่ยนแปลงโดยป้อน คำสั่งต่อไปนี้:

   ```
   # cfgmgr
   ```

ทำขั้นตอนต่อไปนี้เพื่อสลับไปยังคอนฟิกูเรชัน AIX NIC + OFED RoCE จากคอนฟิกูเรชัน AIX RoCE:

1. หยุดแอ็พพลิเคชัน RDMA ทั้งหมดที่รันอยู่บน PCIe2 10 GbE RoCE Adapter
2. ลบหรือกำหนดอินสแตนซ์ของ *roceX* อีกครั้งโดยป้อน คำสั่งอย่างใดอย่างหนึ่งต่อไปนี้:
   - `# rmdev -d -l roce0`
   - `# rmdev -l roce0`

   คำสั่ง `rmdev -l roce0` จะรักษาคำนิยาม ของคอนฟิกูเรชัน `roce0` เพื่อให้คุณสามารถใช้ใน ครั้งถัดไปเพื่อสร้างอินสแตนซ์
3. เปลี่ยนแอ็ตทริบิวต์ของค่าติดตั้ง hba stack_type จาก `aix_ib` (AIX RoCE) เป็น `ofed` (AIX NIC + OFED RoCE) โดยการป้อน คำสั่งต่อไปนี้:

   ```
   # chdev -l hba0 -a stack_type=ofed
   ```

4. รันเครื่องมือโปรแกรมจัดการคอนฟิกูเรชันเพื่อให้โฮสต์บัสอะแด็ปเตอร์ สามารถกำหนดคอนฟิก PCIe2 10 GbE RoCE Adapter เป็นอะแด็ปเตอร์ NIC โดยป้อน คำสั่งต่อไปนี้:

   ```
   # cfgmgr
   ```

5. ตรวจสอบว่าตอนนี้ อะแด็ปเตอร์รันอยู่ในคอนฟิกูเรชัน NIC โดย ป้อนคำสั่งต่อไปนี้:

   ```
   # lsdev -C -c adapter
   ```

   ตัวอย่าง ต่อไปนี้แสดงผลลัพธ์เมื่อคุณรันคำสั่ง **lsdev** บนอะแด็ปเตอร์ เมื่อมีการกำหนดคอนฟิกไว้ในโหมด AIX NIC + OFED RoCE:

```
ent1 Available 00-00-01 PCIe2 10GbE RoCE Converged Network Adapter
ent2 Avaliable 00-00-02 PCIe2 10GbE RoCE Converged Network Adapter
hba0 Available 00-00 PCIe2 10GbE RoCE Converged Host Bus Adapter (b315506714101604)
```

*รูปที่ 47. ตัวอย่างเอาต์พุตของคำสั่ง **lsdev** บนอะแด็ปเตอร์ที่มีคอนฟิกูเรชัน AIX NIC + OFED RoCE*

เนื่องจาก AIX 7 ที่มี 7100-03, AIX ยังสนับสนุน OFED RDMA ในโหมด AIX NIC ด้วย ดังนั้น ถ้าต้องเปิดใช้งาน OFED RDMA คุณต้องทำขั้นตอนเพิ่มเติมสองขั้นตอนต่อไปนี้:

1. ติดตั้งแพ็กเกจ ofed.core
2. ตั้งค่าโหมด RDMA ในอุปกรณ์ ent1, ent2 โดยป้อนคำสั่ง ต่อไปนี้:

```
# chdev -l ent1 -a rdma=desired
# chdev -l ent2 -a rdma=desired
```

โหมด RDMA มีการตั้งค่า ก่อนหน้าการกำหนดคอนฟิกอินเตอร์เฟส en1 หรือ en2

3. คุณสามารถปิดใช้งานโหมด RDMA โดยป้อนคำสั่งต่อไปนี้:

```
# chdev -l ent1 -a rdma=disabled
# chdev -l ent2 -a rdma=disabled
```

AIX RoCE

อะแด็ปเตอร์ PCIe2 10 GbE RoCE มีการกำหนดคอนฟิกไว้ล่วงหน้า เพื่อให้ทำงานในโหมด AIX RoCE เครือข่ายที่ใช้ RDMA ให้ประสิทธิภาพการทำงานที่ดีกว่าอะแด็ปเตอร์ซึ่งใช้เป็น NIC สำหรับแอ็พพลิเคชันที่มุ่งเน้นเครือข่าย โหมดนี้มัก มีประโยชน์สำหรับหน่วยเก็บข้อมูลเครือข่ายหรือการคำนวณประสิทธิภาพสูง

คอนฟิกูเรชัน AIX RoCE ต้องใช้ไลบรารีหรืออินเตอร์เฟสเช่นดังต่อไปนี้:

- Direct Access Programming Library (uDAPL) ซึ่งใช้โดย ระบบฐานข้อมูล DB2®
- Message Passing Interface (MPI) ซึ่งใช้โดยการคำนวณประสิทธิภาพสูง (HPC)

รูปที่ 48 ในหน้า 741 แสดงเอาต์พุตเมื่ออะแด็ปเตอร์ กำลังรันอยู่ในโหมด AIX RoCE

อะแด็ปเตอร์ PCIe2 10 GbE RoCE แสดงอินสแตนซ์ของอะแด็ปเตอร์เพียงรายการเดียวเมื่อ อยู่ในโหมด AIX RoCE แต่สามารถมีพอร์ตมากถึงสองพอร์ต ใช้คำสั่ง **ibstat** เพื่อให้ทราบจำนวนพอร์ตที่กำหนดคอนฟิกไว้โดยทำขั้นตอน ต่อไปนี้:

1. กำหนดว่าส่วนขยายเคอร์เนล `icm` ถูกกำหนดคอนฟิกไว้โดยป้อนคำสั่งต่อไปนี้:

```
# lsdev -C | grep icm
```

2. ถ้าไม่ได้กำหนดคอนฟิกเคอร์เนล icm ให้กำหนดคอนฟิก เคอร์เนลโดยป้อนคำสั่งต่อไปนี้:

```
# mkdev -c management -s infiniband -t icm
```

3. รันคำสั่ง **ibstat** โดยป้อนคำสั่ง ต่อไปนี้:

```
# ibstat roce0
```

ในขณะที่อะแด็ปเตอร์ PCIe2 10 GbE RoCE มีการกำหนดคอนฟิก เป็นครั้งแรกเพื่อใช้โหมด AIX RoCE คุณอาจต้องสลับกลับจากคอนฟิกูเรชัน AIX NIC + OFED RoCE เมื่อต้องการสลับ จากคอนฟิกูเรชัน AIX NIC + OFED RoCE ไปยังคอนฟิกูเรชัน AIX RoCE ให้ทำขั้นตอนต่อไปนี้:

1. ตรวจสอบว่าอะแด็ปเตอร์อยู่ในโหมด AIX NIC + OFED RoCE โดยป้อน คำสั่งต่อไปนี้:

```
# lsdev -C -c adapter
```

เอาต์พุต ของคำสั่ง **lsdev** คล้ายกับตัวอย่าง ใน รูปที่ 47 ในหน้า 739

2. หยุดทราฟฟิก TCP/IP และถอดอินเตอร์เฟส IP โดยป้อน คำสั่งต่อไปนี้:

```
# ifconfig en1 down detach; ifconfig en2 down detach
```

3. ลบหรือวางอินสแตนซ์ NIC ในสถานะที่นิยามไว้โดยป้อนหนึ่งในคำสั่งต่อไปนี้ :
   - `# rmdev -d -l ent1; rmdev -d -l ent2`
   - `# rmdev -l ent1; rmdev -l ent2`

คำสั่ง `rmdev -l ent1; rmdev -l ent2` จะรักษา คำนิยามของอุปกรณ์อีเทอร์เน็ตเพื่อให้คุณสามารถใช้ใน ครั้งถัดไปที่คุณสร้างอินสแตนซ์

4. เปลี่ยนแอ็ตทริบิวต์ของ hba stack_type จาก `ofed` (AIX NIC + OFED RoCE) เป็น `aix_ib` (AIX RoCE) โดยป้อนคำสั่งต่อไปนี้:

   ```
   # chdev -l hba0 -a stack_type=aix_ib
   ```

5. รันเครื่องมือ configuration manager เพื่อให้โฮสต์บัสอะแด็ปเตอร์ สามารถกำหนดคอนฟิกอะแด็ปเตอร์ PCIe2 10 GbE RoCE เป็นอะแด็ปเตอร์ AIX RoCE โดยป้อนคำสั่งต่อไปนี้:

   ```
   # cfgmgr
   ```

6. ตรวจสอบว่าตอนนี้ อะแด็ปเตอร์รันอยู่ในคอนฟิกูเรชัน AIX RoCE โดยป้อนคำสั่ง ต่อไปนี้:

   ```
   # lsdev -C -c adapter
   ```

   ตัวอย่าง ต่อไปนี้แสดงผลลัพธ์เมื่อคุณรันคำสั่ง **lsdev** สำหรับอะแด็ปเตอร์ และอะแด็ปเตอร์มีการกำหนดคอนฟิกไว้ในโหมด AIX RoCE

```
roce0 Avaliable 00-00-00 PCIe2 10GbE RoCE Converged Network Adapter
hba0 Available  00-00-00 PCIe2 10GbE RoCE Converged Host Bus Adapter (b315506714101604)
```

*รูปที่ 48. ตัวอย่าง เอาต์พุตของคำสั่ง **lsdev** สำหรับอะแด็ปเตอร์เมื่อ ใช้คอนฟิกูเรชัน AIX RoCE*

การสนับสนุนอะแด็ปเตอร์ PCIe3 40 GbE RoCE

อะแด็ปเตอร์ PCIe3 40 GbE RDMA Over Converged Ethernet (RoCE) สนับสนุน remote direct memory access (RDMA) ที่มี OpenFabrics Enterprise Distribution (OFED) ในโหมด NIC ปกติ RDMA ได้รับการสนับสนุนและ เปิดใช้งานโดยดีฟอลต์ถ้าติดตั้งซอฟต์แวร์ OpenFabrics ไว้

เมื่อต้องการดาวน์โหลดไดรเวอร์อุปกรณ์ล่าสุดสำหรับอะแด็ปเตอร์นี้ ให้ทำ ขั้นตอนต่อไปนี้:

1. ไปยังเว็บไซต์ IBM (www.ibm.com)
2. คลิก **การสนับสนุนและดาวน์โหลด**
3. ดาวน์โหลดเฟิร์มแวร์ล่าสุดไปยังตำแหน่งโฮสต์ AIX (`/etc/microcode`)
4. รันเครื่องมือ **diag** เพื่ออัพเดตเฟิร์มแวร์โดยเลือก โพรซีเดอร์อย่างใดอย่างหนึ่งต่อไปนี้:
   - โพรซีเดอร์พาธสั้น
     a. ป้อนคำสั่งต่อไปนี้:

        ```
        *diag -d entX -T download
        ```

        หมายเหตุ: ถ้า อุปกรณ์อีเทอร์เน็ตเป็นสมาชิกของโฮสต์บัสอะแด็ปเตอร์เดียวกัน (ตัวอย่างเช่น hba0, hba1 และต่อไป) ให้ดาวน์โหลดเฟิร์มแวร์ไปยังอุปกรณ์ **ent** อย่างใดอย่างหนึ่ง

     b. เลือกไมโครโค้ดที่บันทึกไว้ในไดเร็กทอรี `/etc/microcode`
   - โพรซีเดอร์พาธยาว
     a. ป้อนคำสั่งต่อไปนี้:

```
*diag
```

b. คลิก **การเลือกภารกิจ** > **ภารกิจไมโครโค้ด** > **ดาวน์โหลดไมโครโค้ด**.

c. เลือก **entX**

d. เลือกไมโครโค้ดที่บันทึกไว้ในไดเร็กทอรี `/etc/microcode`

เมื่อต้องการใช้อะแด็ปเตอร์ PCIe3 40 GbE RoCE และ AIX NIC + OFED RoCE ต้องใช้ชุดไฟล์ต่อไปนี้ ชุดไฟล์เหล่านี้มีอยู่ในแผ่นซีดีของระบบปฏิบัติการพื้นฐาน AIX 7 ที่มี 7100-03

devices.ethernet.mlx	ไดรเวอร์อุปกรณ์หลัก Converged Ethernet Adapter (mlxentdd) เพื่อสนับสนุนคอนฟิกูเรชัน AIX NIC + OFED RoCE
devices.pciex.b31503101410b504	การจัดทำแพ็กเกจสำหรับ Mellanox 2 Ports 40 Gb Converged Ethernet Adapter ที่ใช้พอร์ต passive copper Quad Small Form-factor Pluggable (QSFP)
ofed.core	ชุดไฟล์ OFED Core Runtime Environment ที่ ต้องใช้เฉพาะถ้าต้องการการทำงาน OFED RDMA

เมื่อต้องการปิดใช้งานการทำงาน RDMA ให้ป้อนคำสั่งต่อไปนี้:

```
chdev  -l  <Ethernet_device> rdma=disabled
```

ตัวอย่าง:

```
# chdev -l ent1 -a rdma=disabled
# chdev -l ent2 -a rdma=disabled
```

เมื่อต้องการเปิดใช้งานการทำงาน RDMA ให้ป้อนคำสั่งต่อไปนี้:

```
chdev -l <Ethernet_device> rdma=desired
```

คำประกาศ

ข้อมูลนี้จัดทำขึ้นสำหรับผลิตภัณฑ์และเซอร์วิสที่นำเสนอในสหรัฐฯ

IBM อาจไม่นำเสนอผลิตภัณฑ์ เซอร์วิส หรือคุณลักษณะที่อธิบายในเอกสารนี้ในประเทศอื่น โปรดปรึกษาตัวแทน IBM ในท้องถิ่นของคุณสำหรับข้อมูลเกี่ยวกับผลิตภัณฑ์ และเซอร์วิส ที่มีอยู่ในพื้นที่ของคุณในปัจจุบัน การอ้างอิงใดๆ ถึงผลิตภัณฑ์ โปรแกรม หรือเซอร์วิสของ IBM ไม่ได้มีวัตถุประสงค์ที่จะระบุหรือตีความว่า สามารถใช้ได้เฉพาะผลิตภัณฑ์ โปรแกรม หรือเซอร์วิสของ IBM เพียงอย่างเดียว เท่านั้น ผลิตภัณฑ์ โปรแกรม หรือเซอร์วิสใดๆ ที่สามารถทำงานได้เท่าเทียมกัน และไม่ละเมิดสิทธิทรัพย์สินทางปัญญาของ IBM อาจนำมาใช้ แทนได้ อย่างไรก็ตาม ถือเป็นความรับผิดชอบของผู้ใช้ที่จะประเมินและตรวจสอบการดำเนินการของ ผลิตภัณฑ์ โปรแกรม หรือเซอร์วิสใดๆ ที่ไม่ใช่ของ IBM

IBM อาจมีสิทธิบัตร หรืออยู่ระหว่างดำเนินการขอ สิทธิบัตรที่ครอบคลุมถึงหัวข้อซึ่งอธิบายในเอกสารนี้ การนำเสนอเอกสารนี้ไม่ได้เป็นการให้ไลเซนส์ใดๆ ในสิทธิบัตรเหล่านี้แก่คุณ คุณสามารถส่งการสอบถามเกี่ยวกับไลเซนส์ เป็นลายลักษณ์อักษรไปยัง:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

หากมีคำถามเกี่ยวกับข้อมูลชุดอักขระไบต์คู่ (DBCS) โปรดติดต่อแผนกทรัพย์สินทางปัญญาของ IBM ในประเทศของคุณ หรือส่งคำถาม เป็นลายลักษณ์อักษร ไปยัง:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

ย่อหน้าต่อไปนี้ไม่ได้ใช้กับสหราชอาณาจักร หรือประเทศอื่นใดที่ ข้อกำหนดดังกล่าวไม่สอดคล้องกับกฎหมายท้องถิ่น: INTERNATIONAL BUSINESS MACHINES CORPORATION นำเสนอสิ่งพิมพ์นี้ "ตามสภาพ" โดยไม่มีการรับประกันใดๆ โดยชัดแจ้งหรือ โดยนัย ซึ่งรวมถึงแต่ไม่จำกัดเฉพาะการรับประกันโดยนัยถึงการไม่ละเมิด การขายได้ หรือความเหมาะสมสำหรับวัตถุประสงค์เฉพาะ เนื่องจากบางรัฐไม่อนุญาตให้ปฏิเสธการรับประกันโดยชัดแจ้งหรือ โดยนัยในธุรกรรมบางอย่าง ดังนั้น ข้อความสั่งนี้จึงอาจไม่ใช้กับคุณ

ข้อมูลนี้อาจมีความไม่ถูกต้องด้านเทคนิคหรือข้อผิดพลาดจากการพิมพ์ มีการเปลี่ยนแปลง ข้อมูลในเอกสารนี้เป็นระยะ และการเปลี่ยนแปลงเหล่านี้จะรวมอยู่ในเอดิชันใหม่ของ สิ่งพิมพ์ IBM อาจปรับปรุง และ/หรือเปลี่ยนแปลงในผลิตภัณฑ์ และ/หรือโปรแกรมที่อธิบายในสิ่งพิมพ์นี้ได้ตลอดเวลา โดยไม่ต้องแจ้งให้ทราบ

การอ้างอิงใดๆ ในข้อมูลนี้ถึงเว็บไซต์ไม่ใช่ของ IBM มีการจัดเตรียมเพื่อความสะดวกเท่านั้น และ ไม่ได้เป็นการรับรองเว็บไซต์เหล่านั้นในลักษณะใดๆ เอกสารประกอบที่เว็บไซต์เหล่านั้นไม่ได้เป็น ส่วนหนึ่งของเอกสารประกอบสำหรับผลิตภัณฑ์ IBM นี้ และการใช้เว็บไซต์เหล่านั้นถือเป็นความเสี่ยงของคุณเอง

IBM อาจใช้ หรือแจกจ่าย ข้อมูลใดๆ ที่คุณให้ในวิธีที่ IBM เชื่อว่าเหมาะสมโดยไม่ก่อให้เกิดข้อผูกมัดใดๆ กับ คุณ

ผู้รับไลเซนส์ของโปรแกรมนี้ที่ต้องการข้อมูลเกี่ยวกับโปรแกรมสำหรับวัตถุประสงค์ในการเปิดใช้งาน: (i) การแลกเปลี่ยนข้อมูลระหว่างโปรแกรมที่สร้างขึ้นอย่างอิสระกับโปรแกรมอื่น (รวมถึง โปรแกรมนี้) และ (ii) การใช้ข้อมูลซึ่งแลกเปลี่ยนร่วมกัน ควร ติดต่อ:

IBM Corporation
Dept. LRAS/Bldg. 903
11501 Burnet Road
Austin, TX 78758-3400
USA

ข้อมูลดังกล่าวอาจพร้อมใช้งาน ภายใต้ข้อตกลงและเงื่อนไขที่เหมาะสม รวมถึง การชำระค่าธรรมเนียมในบางกรณี

โปรแกรมที่มีไลเซนส์ซึ่งอธิบายในเอกสารนี้ และเอกสารประกอบที่มีไลเซนส์ทั้งหมดสำหรับโปรแกรม นั้น มีการจัดเตรียมโดย IBM ภายใต้ข้อตกลงของข้อตกลงกับลูกค้าของ IBM, ข้อตกลงไลเซนส์โปรแกรมระหว่างประเทศของ IBM หรือข้อตกลงที่เท่าเทียมกันใดๆ ระหว่างเรา

ข้อมูลประสิทธิภาพใดๆ ที่มีในเอกสารนี้ถูกกำหนดในสภาวะแวดล้อมที่ควบคุม ด้วยเหตุนี้ ผลลัพธ์ที่ได้ในสภาวะแวดล้อมการปฏิบัติการอื่นจึงอาจแตกต่างไปอย่างมาก การวัดบางอย่างอาจ ดำเนินการบนระบบที่อยู่ระหว่างการพัฒนา และไม่มีการรับประกันว่าการวัดเหล่านี้จะ เหมือนกันบนระบบที่พร้อมใช้งานโดยทั่วไป ยิ่งไปกว่านั้น การวัดบางอย่างอาจมีการประเมินโดยวิธีการ ประมาณค่านอกช่วง ผลลัพธ์จริงอาจแตกต่างไป ผู้ใช้เอกสารนี้จึงควรตรวจสอบ ข้อมูลที่สามารถใช้ได้สำหรับสภาวะแวดล้อมของตน

ข้อมูลเกี่ยวกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM ได้รับมาจากซัพพลายเออร์ของผลิตภัณฑ์เหล่านั้น ประกาศที่เผยแพร่ หรือแหล่งข้อมูลที่เปิดเผยต่อสาธารณะ IBM ไม่ได้ทดสอบผลิตภัณฑ์ดังกล่าว และไม่สามารถยืนยันความถูกต้องของ ประสิทธิภาพ ความเข้ากันได้ หรือการเรียกร้องอื่นใดที่เกี่ยวข้องกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM คำถามเกี่ยวกับ ความสามารถของผลิตภัณฑ์ที่ไม่ใช่ของ IBM ควรส่งไปยังซัพพลายเออร์ของผลิตภัณฑ์เหล่านั้น

ข้อความทั้งหมดเกี่ยวกับทิศทางหรือเจตนาในอนาคตของ IBM อาจมีการเปลี่ยนแปลง หรือเพิกถอนได้โดยไม่ต้องแจ้งให้ทราบ และแสดงถึงเป้าหมายและวัตถุประสงค์เท่านั้น

ราคาของ IBM ทั้งหมดที่แสดงเป็นราคาขายปลีกที่แนะนำของ IBM ซึ่งเป็นราคาปัจจุบัน และอาจเปลี่ยนแปลงได้โดยไม่ต้องแจ้งให้ทราบ ราคาของผู้แทนจำหน่ายอาจแตกต่างไป

ข้อมูลนี้ใช้สำหรับวัตถุประสงค์ของการวางแผนเท่านั้น ข้อมูลในเอกสารนี้อาจมีการเปลี่ยนแปลง ก่อนผลิตภัณฑ์ที่อธิบายจะวางจำหน่าย

ข้อมูลนี้มีตัวอย่างของข้อมูลและรายงานที่ใช้ในการดำเนินการทางธุรกิจรายวัน เพื่อ สาธิตข้อมูลให้สมบูรณ์ที่สุดเท่าที่จะเป็นไปได้ ตัวอย่างจึงมีชื่อของแต่ละบุคคล บริษัท ยี่ห้อ และผลิตภัณฑ์ ชื่อทั้งหมดเหล่านี้เป็นชื่อสมมติ และการคล้ายคลึงในชื่อและที่อยู่ซึ่งหน่วยธุรกิจจริงใช้เป็นความบังเอิญโดยสิ้นเชิง

ไลเซนส์ลิขสิทธิ์:

ข้อมูลนี้มีตัวอย่างแอ็พพลิเคชันโปรแกรมในภาษาต้นฉบับ ซึ่งแสดงถึง เทคนิคด้านโปรแกรมในหลากหลายแพล็ตฟอร์ม คุณอาจคัดลอก ปรับเปลี่ยน และแจกจ่าย โปรแกรมตัวอย่างเหล่านี้ในรูปแบบใดๆ โดยไม่ต้องชำระเงินให้แก่ IBM สำหรับวัตถุประสงค์ในการพัฒนา การใช้ การตลาด หรือการแจกจ่ายโปรแกรมแอ็พพลิเคชัน ที่สอดคล้องกับอินเตอร์เฟสการเขียนโปรแกรมแอ็พพลิเคชันสำหรับแพล็ตฟอร์มปฏิบัติการ ซึ่งเขียน โปรแกรมตัวอย่าง ตัวอย่างเหล่านี้ยังไม่ได้ผ่านการทดสอบในทุกสภาพ ดังนั้น IBM จึงไม่สามารถรับประกัน หรือบอกเป็นนัยถึง ความน่าเชื่อถือ ความสามารถบริการได้ หรือฟังก์ชันของโปรแกรมเหล่านี้ โปรแกรมตัวอย่างมีการนำเสนอ "ตาม สภาพ" โดยไม่มีการรับประกันประเภทใดๆ IBM ไม่รับผิดชอบ ต่อความเสียหายใดๆ ที่เกิดขึ้นเนื่องจากการใช้โปรแกรมตัวอย่างของคุณ

แต่ละสำเนา หรือส่วนใดๆ ของโปรแกรมตัวอย่างเหล่านี้ หรืองานที่สืบเนื่องใดๆ ต้องมี คำประกาศลิขสิทธิ์ดังนี้:

ส่วนของโค้ดนี้ได้มาจากโปรแกรมตัวอย่างของ IBM Corp.

สิ่งที่ต้องพิจารณาเกี่ยวกับนโยบายความเป็นส่วนตัว

ผลิตภัณฑ์ซอฟต์แวร์ของ IBM รวมถึงโซลูชันบริการระบบซอฟแวร์ ("ข้อเสนอซอฟต์แวร์") อาจใช้คุกกี้หรือเทคโนโลยีอื่นเพื่อรวบรวมข้อมูลการใช้งานผลิตภัณฑ์ เพื่อช่วยในการปรับปรุงประสบการณ์การใช้งานของผู้ใช้ขั้นปลาย เพื่อปรับแต่งการโต้ตอบกับ ผู้ใช้ขั้นปลาย หรือเพื่อวัตถุประสงค์อื่นๆ ในหลายๆ กรณี จะไม่มีการรวบรวม ข้อมูลอัตลักษณ์ส่วนบุคคลโดย ข้อเสนอซอฟต์แวร์ ซึ่งข้อเสนอซอฟต์แวร์บางอย่าง สามารถช่วยให้คุณรวบรวมข้อมูลอัตลักษณ์ ส่วนบุคคลได้ ถ้าข้อเสนอซอฟต์แวร์นี้ใช้คุกกี้ เพื่อรวบรวมข้อมูลอัตลักษณ์, ระบุข้อมูล เกี่ยวกับการใช้คุกกี้ของข้อเสนอนี้ถูกกำหนดไว้ด้านล่าง

ข้อเสนอซอฟต์แวร์นี้ไม่ใช้คุกกี้ หรือเทคโนโลยีอื่นเพื่อรวบรวมข้อมูลอัตลักษณ์ส่วนบุคคล

ถ้าคอนฟิกูเรชันถูกปรับใช้สำหรับ ข้อเสนอกนี้จัดเตรียมให้คุณในฐานะลูกค้าสามารถรวบรวม ข้อมูลอัตลักษณ์ส่วนบุคคลจากผู้ใช้ขั้นปลายผ่านทางคุกกี้ และเทคโนโลยีอื่น คุณควรปรึกษากับที่ปรึกษาด้านกฎหมายเกี่ยวกับ ที่ใช้บังคับในการรวบรวมข้อมูล รวมถึงข้อกำหนดต่างๆ เพื่อการแจ้งเตือนและการยินยอม

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการใช้ เทคโนโลยีต่างๆ รวมถึงคุกกี้ สำหรับวัตถุประสงค์เหล่านี้ โปรดดู นโยบายความเป็นส่วนตัวของ IBM ที่ http://www.ibm.com/privacy และ คำชี้แจงสิทธิส่วนบุคคลออนไลน์ของ IBM ที่ส่วน http://www.ibm.com/privacy/details "Cookies, Web Beacons and Other Technologies" และ "IBM Software Products and Software-as-a-Service Privacy Statement" ที่ http://www.ibm.com/software/info/product-privacy

---

เครื่องหมายการค้า

IBM, ตราสัญลักษณ์ IBM , และ ibm.com เป็นเครื่องหมายการค้าหรือเครื่องหมายการค้าที่จดทะเบียนของ International Business Machines Corp. ซึ่งจดทะเบียนในหลายเขตอำนาจศาลทั่วโลก ชื่อผลิตภัณฑ์และการบริการอื่นอาจเป็นเครื่องหมายการค้าของ IBM หรือบริษัทอื่น รายการปัจจุบันของเครื่องหมายการค้า IBM มีอยู่บนเว็บไซต์ที่ ข้อมูลลิขสิทธิ์และเครื่องหมายการค้า ที่ www.ibm.com/legal/copytrade.shtml

INFINIBAND, InfiniBand Trade Association, และ ลักษณะแบบ INFINIBAND คือเครื่องหมายการค้าและ/หรือลักษณะเซอร์วิสของ INFINIBAND Trade Association

ดัชนี

อักขระพิเศษ

ตัวเลข

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

ก

ข

ค

จ

ช

ซ

ด

ต

ท

น

บ

ป

ผ

พ

ฟ

ภ

ม

ย

ร

ล

ว

ส

ห

อ

ฮ

พิมพ์ในสหรัฐอเมริกา