

IBM

@server

iSeries

Geïntegreerd bestandssysteem

Versie 5 Release 3





@server

iSeries

Geïntegreerd bestandssysteem

Versie 5 Release 3

Opmerking

Voordat u deze informatie en het product gebruikt, leest u eerst de informatie in "Kennisgevingen", op pagina 127.

Zesde uitgave (augustus 2005)

| Deze uitgave is van toepassing op versie 5, release 3, modificatie 0 van IBM Operating System/400 (product-
| nummer 5722-SS1) en op alle volgende releases en modificaties totdat in nieuwe uitgaven anders wordt aangege-
| ven. Deze versie kan niet op alle RISC-computers (Reduced Instruction Set Computer) worden uitgevoerd en niet
| op CICS-modellen.

© Copyright IBM Corp. 1999, 2005.

Inhoudsopgave

Geïntegreerd bestandssysteem.	1	Journaaloverzicht	88
Informatie over uitsluiting van garanties voor code	1	Journaal starten	92
Nieuw in V5R3	2	Journaal wijzigen	93
Dit onderwerp afdrukken	3	Journaal beëindigen	93
Overzicht van het geïntegreerd bestandssysteem	3	Ondersteuning voor programmeren	93
Wat is het geïntegreerd bestandssysteem?.	3	Gegevens kopiëren tussen stroombestanden en databasebestanden	94
Waarom gebruikmaken van het geïntegreerd bestandssysteem?.	4	Gegevens kopiëren tussen stroombestanden en opslagbestanden.	97
Concepten	5	Handelingen uitvoeren met behulp van API's	98
Directory	6	Socketondersteuning	107
Koppeling	11	Naamgeving en internationale ondersteuning	107
Padnaam	15	Gegevensconversie	108
Stroombestand	15	Voorbeeld: C-functies Geïntegreerd bestandssysteem	108
Naamcontinuïteit	17	Werken met bestanden en mappen met behulp van iSeries Navigator	114
Uitgebreide kenmerken	17	Een bestand inchecken	114
Ondersteuning van scans	18	Een bestand uitchecken	114
Werken met bestandssystemen	23	Een map maken	114
Vergelijking bestandssystemen	24	Een map verwijderen.	115
Bestandssysteem "hoofddirectory" (/)	28	Bestanden en mappen naar een andere bestandssysteem verplaatsen	115
Bestandssysteem open systemen (QOpenSys)	30	Instellingsmachtigingen	116
Door de gebruiker gedefinieerd bestandssysteem (UDFS)	32	Bestandstekstconversie instellen	116
Bibliotheekbestandssysteem (QSYS.LIB)	37	Een bestand of map naar een ander systeem verzenden	117
Onafhankelijke ASP QSYS.LIB	40	Opties wijzigen voor een pakketdefinitie	117
Bestandssysteem voorzieningen documentenbibliotheek (QDLS)	43	Een datum en tijd plannen om uw bestand of map te verzenden	118
Bestandssysteem optische media (QOPT)	45	Een bestandsshare maken	118
Bestandssysteem NetWare (QNetWare)	47	Een bestandsshare wijzigen.	118
Bestandssysteem iSeries NetClient (QNTC)	50	Hiermee stelt u in of objecten moeten worden gescand	118
OS/400Bestandssysteem bestandserver (QFileSvr.400).	53	Aanvullende informatie met betrekking tot geïntegreerde bestandssystemen	119
Netwerkbestandssysteem (NFS)	57	Transportonafhankelijke RPC	119
Toegang tot het geïntegreerd bestandssysteem.	59	Verwijzingen	124
Toegang met behulp van menu's en schermen.	60	Ervaringsrapporten	125
Toegang met behulp van CL-opdrachten.	61	Bijlage. Kennisgevingen.	127
Toegang met behulp van API's	80	Informatie over programmeerinterfaces.	129
Toegang met behulp van iSeries Navigator	80	Merken	129
Toegang met behulp van iSeries NetServer	81	Bepalingen en voorwaarden voor het downloaden en afdrukken van publicaties	129
Toegang met behulp van bestandsoverdrachtsprogramma's	82		
Toegang met behulp van een PC	83		
Directory's converteren van *TYPE1 naar *TYPE2.	84		
Overzicht conversie van *TYPE1 naar *TYPE2	84		
Tips: Conversies	84		
Objecten opnemen in een journaal.	88		

Geïntegreerd bestandssysteem

Het geïntegreerd bestandssysteem is een onderdeel van OS/400 dat stroominvoer/uitvoer en opslag-beheer ondersteunt zoals met een PC- of UNIX-besturingssystemen, terwijl u over een geïntegreerde structuur beschikt voor alle gegevens die op de server zijn opgeslagen.

Raadpleeg de volgende onderwerpen voor meer informatie over het geïntegreerd bestandssysteem op OS/400.

Nieuw in V5R3

Hier leest u hoe dit onderwerp is gewijzigd sinds de vorige release.

Afdrukbare PDF

Hier leest u hoe u een PDF-versie van dit onderwerp kunt afbeelden en afdrukken.

Overzicht van het geïntegreerd bestandssysteem

Hier worden de grondbeginselen van het geïntegreerd bestandssysteem uitgelegd.

Concepten

Hier vindt u informatie over de verschillende concepten van het geïntegreerd bestandssysteem.

Werken met bestandssystemen

Hier leest u hoe de verschillende bestandssystemen in OS/400 worden beheerd.

Toegang tot het geïntegreerd bestandssysteem

Hier komt u te weten op welke manieren u toegang kunt krijgen tot een geïntegreerd bestands-systeem. De manieren zijn: via menu's en schermen, via CL-opdrachten, via API's, via iSeries NetServer, via iSeries Navigator en via bestandsoverdrachtprogramma's.

Directory's converteren van *TYPE1 naar *TYPE2

Hier komt u te weten hoe het systeem automatisch *TYPE1-directory's converteert naar *TYPE2.

Objecten opnemen in een journaal

Hier leest u hoe u journaals voor IFS-objecten kunt starten, beëindigen en wijzigen.

Ondersteuning voor programmeren

Hier leest u hoe u gegevens kunt kopiëren, API's kunt gebruiken en andere ondersteunende functies kunt uitvoeren.

Werken met bestanden en mappen met behulp van iSeries Navigator

Hier leest u hoe u taken met bestanden en mappen kunt uitvoeren, zoals het in- en uitchecken van een bestand en het maken en wissen van een map.

Aanvullende informatie met betrekking tot geïntegreerde bestandssystemen

Informatie vinden in het iSeries Informatiecentrum en op internet over meer aspecten van het geïntegreerd bestandssysteem.

Opmerking: Lees de "Informatie over uitsluiting van garanties voor code" voor belangrijke juridische informatie.

Informatie over uitsluiting van garanties voor code

Dit document bevat programmeervoorbeelden.

IBM verleent een niet-exclusieve copyrightlicentie voor gebruik van alle codevoorbeelden waarmee u vergelijkbare functies kunt maken, aangepast aan uw specifieke wensen.

Alle voorbeeldcode wordt door IBM aangeboden ter illustratie. Deze voorbeelden zijn niet uitgebreid onder alle omstandigheden getest. IBM kan daarom de betrouwbaarheid, de bruikbaarheid of de functies van deze programma's niet garanderen.

De hierin opgenomen programma's worden op "AS IS" basis aan u verstrekt. Er wordt uitdrukkelijk geen garantie gegeven betreffende verhandelbaarheid of geschiktheid voor een bepaald doel, noch dat er geen inbreuk wordt gemaakt op rechten van derden.

Nieuw in V5R3

| De volgende gegevens zijn in deze release toegevoegd of bijgewerkt:

API's in het geïntegreerd bestandssysteem

| Het geïntegreerd bestandssysteem heeft meer functionaliteit voor en ondersteuning van de volgende API's:

- | • Bestand verwijderen (fclear())
- | • Bestand verwijderen (voor groot bestand geschikt) (fclear64())
- | • Handtekening scan wijzigen (QP0LCHSG)
- | • Geraadpleegde objecten ophalen (QP0LRRO)
- | • Handtekening scan ophalen (QP0LRTSG)

CL-opdrachten voor het geïntegreerd bestandssysteem

| In V5R3 heeft het geïntegreerd bestandssysteem meer functionaliteit voor en ondersteuning van de volgende CL-opdrachten:

- | • De opdracht CHGJRNOBJ (In journaal vastgelegd object wijzigen)
- | • De opdracht PRDIRINF (Directory-informatie afdrukken)
- | • De opdracht RTVDIRINF (Directory-informatie ophalen)

| Raadpleeg "Werken met uitvoer van de opdrachten RTVDIRINF en PRDIRINF" op pagina 66 voor informatie over de uitvoer van de opdrachten PRDIRINF en RTVDIRINF.

Twee stuurprogramma's

| Het bestandssysteem hoofddirectory (/) in de directory /dev/xti bevat nu twee stuurprogramma's, udp en tcp. In het volgende onderwerp worden stuurprogramma's besproken:

- | • "UDP- en TCP-apparatuur in het bestandssysteem hoofddirectory (/)" op pagina 30

Kerberos-configuratie voor het bestandssysteem QNTC

| In vorige releases gebruikte QNTC een wachtwoord om zichzelf te verifiëren bij servers op afstand. Ter vereenvoudiging van het wachtwoordbeheer gebruikt QNTC voortaan een Kerberos-serverticket voor de verificatie bij servers op afstand. "Kerberos inschakelen voor gebruik met het bestandssysteem QNTC" op pagina 52 bevat meer informatie over hoe u Kerberos moet configureren voor gebruik met het bestandssysteem QNTC.

Automatische conversie naar *TYPE2-directory's

| Kort na het installeren van V5R3 begint het systeem bestandssystemen die nog *TYPE1-directory's ondersteunen automatisch te converteren naar *TYPE2-directory's. Raadpleeg "Directory's converteren van *TYPE1 naar *TYPE2" op pagina 84 voor meer informatie.

| **Ondersteuning van scans**

| Als optie beschikt iSeries nu over de mogelijkheid om scans uit te voeren op objecten in het geïntegreerd bestandssysteem. De scans kunnen plaatsvinden tijdens het openen en afsluiten van de objecten. Objecten kunnen om uiteenlopende redenen gescand worden, bijvoorbeeld op virussen. U kunt er ook voor kiezen de scans uit te voeren op de momenten dat het u uitkomt. In het volgende onderwerp wordt uitgebreider aandacht besteed aan de ondersteuning voor scans en wat er nodig is om dit te activeren:

- | • "Ondersteuning van scans" op pagina 18

Dit onderwerp afdrukken

Zie Geïntegreerd bestandssysteem  als u de PDF-versie van dit document wilt bekijken of downloaden (ongeveer 1,455 kB).

PDF-bestanden opslaan

U kunt een PDF-bestand als volgt op uw werkstation opslaan om het te lezen of af te drukken:

1. Klik met de rechtermuisknop op het PDF-bestand in de browser (klik met de rechtermuisknop op de bovenstaande koppeling).
2. Klik op **Doel opslaan als...** als u met Internet Explorer werkt. Klik op **Koppeling opstaan als...** als u met Netscape Communicator werkt.
3. Ga naar de directory waarin u het PDF-bestand wilt opslaan.
4. Klik op **Opslaan**.

Adobe Acrobat Reader downloaden

U hebt Adobe Acrobat Reader nodig om deze PDF-bestanden te bekijken of af te drukken. U kunt dit programma downloaden van de Adobe-website (www.adobe.com/products/acrobat/readstep.html). 

Overzicht van het geïntegreerd bestandssysteem

De volgende onderwerpen geven een beschrijving van het geïntegreerd bestandssysteem op uw iSeries-server en geven aan hoe u het op uw server kunt gebruiken.

- "Wat is het geïntegreerd bestandssysteem?"
- "Waarom gebruikmaken van het geïntegreerd bestandssysteem?" op pagina 4

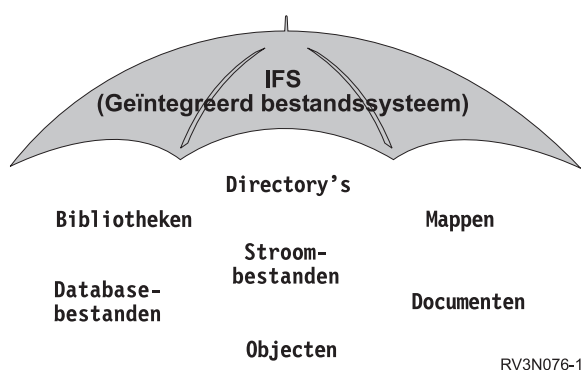
Wat is het geïntegreerd bestandssysteem?

Het **geïntegreerd bestandssysteem** is een onderdeel van OS/400 dat stroominvoer/uitvoer en opslagbeheer ondersteunt zoals met een PC- of UNIX-besturingssystemen, terwijl u over een geïntegreerde structuur beschikt voor alle gegevens die op de server zijn opgeslagen.

Het geïntegreerd bestandssysteem bestaat uit elf bestandssystemen, elk met een eigen verzameling logische structuren en regels voor het interactief werken met opslag. Raadpleeg Werken met bestandssystemen voor meer informatie.

De belangrijkste functies van het geïntegreerd bestandssysteem zijn:

- Ondersteuning van de opslag van informatie in stroombestanden, die lange, doorlopende gegevensreeksen kunnen bevatten. De reeksen kunnen bijvoorbeeld bestaan uit de tekst van een document of de beeldpunten van een afbeelding. De ondersteuning van stroombestanden is ontworpen voor efficiënt gebruik met client/server-toepassingen.
- Een hiërarchische directorystructuur waarmee objecten kunnen worden gerangschikt als fruit aan de takken van een boom. Toegang tot het object wordt verkregen door het pad door de directory's op te geven.
- Een gemeenschappelijke interface waarmee gebruikers en toepassingen niet alleen toegang kunnen krijgen tot stroombestanden, maar ook tot databasebestanden, documenten en andere objecten op uw server.
- Een gemeenschappelijk beeld van stroombestanden die lokaal op uw server staan, op een Integrated xSeries Server for iSeries of op een Windows NT-server op afstand. Stroombestanden kunnen ook op afstand worden opgeslagen op een LAN, een Novell NetWare-server, een andere iSeries-server of een netwerkserver voor bestandssystemen.



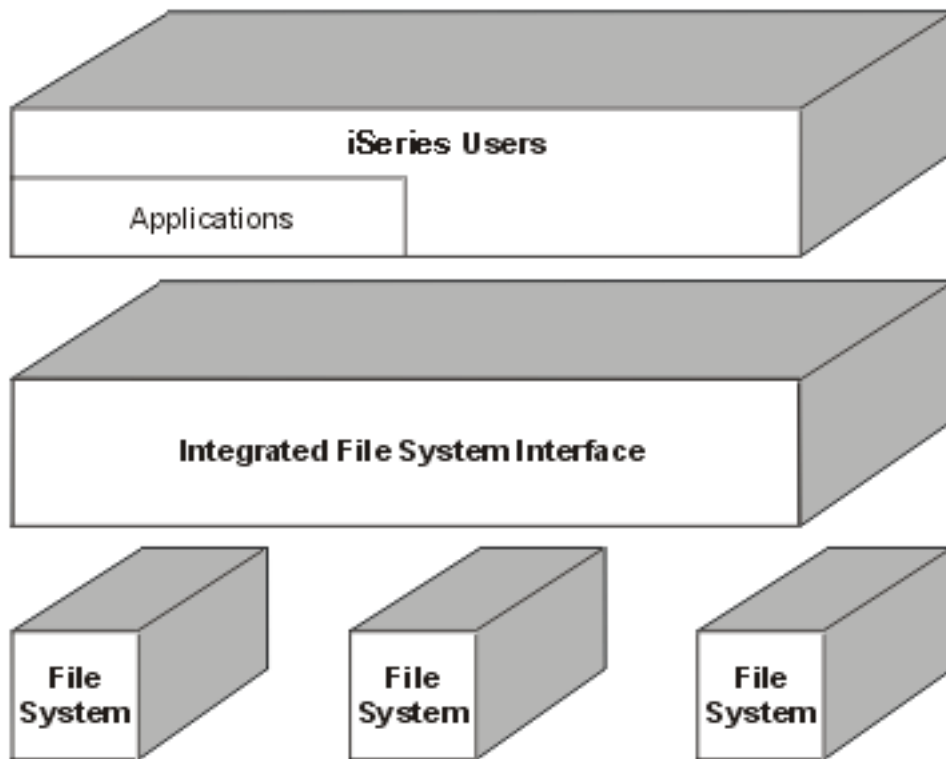
Figuur 1. Een overkoepelende structuur voor alle informatie op een iSeries-server

Waarom gebruikmaken van het geïntegreerd bestandssysteem?

Het geïntegreerd bestandssysteem vergroot de reeds aanzienlijke mogelijkheden voor gegevensbeheer van OS/400 zodat nieuwe vormen van informatieverwerking, zoals client/server, open systemen en multimedia beter kunnen worden ondersteund.

U kunt het geïntegreerd bestandssysteem gebruiken voor het volgende:

- Snelle toegang tot OS/400-gegevens, met name voor toepassingen zoals Client Access die de OS/400-bestandserver gebruiken.
- Efficiëntere verwerking van bepaalde soorten stroomgegevens, zoals afbeeldingen, geluid en bewegende beelden.
- Als een basissysteem voor bestanden en -directory's ter ondersteuning van op UNIX gebaseerde open systemen, zoals Portable Operating System Interface for Computer Environments (POSIX) en XPG. Deze bestandsstructuur en directorystructuur fungeren tevens een vertrouwde omgeving voor gebruikers van PC-besturingssystemen zoals DOS (Disk Operating System) en Windows.
- Bestandsondersteuning met unieke mogelijkheden, zoals op records gerichte databasebestanden, op Unix gebaseerde stroombestanden en het aanleveren van bestanden. Deze mogelijkheden worden verwerkt als aparte bestandssystemen, maar worden niettemin beheerd via een gemeenschappelijke interface.



Figuur 2. Een gemeenschappelijke interface voor gescheiden bestandssystemen

- PC-gebruikers beter laten profiteren van de mogelijkheden van hun grafische gebruikersinterface. Windows-gebruikers kunnen bijvoorbeeld de grafische hulpprogramma's van Windows gebruiken voor serverstroombestanden en andere objecten van iSeries op dezelfde manier waarop ze deze gebruiken voor programma's op hun eigen PC.
- Samenhang in objectnamen en bijbehorende objectinformatie in alle talen. Dit zorgt er bijvoorbeeld voor dat afzonderlijke tekens niet veranderen als u van de codetabel van een taal overschakelt naar de codetabel van een andere taal.

Concepten

In dit deel worden de concepten van het geïntegreerd bestandssysteem behandeld. Raadpleeg een of meer van de onderwerpen uit de onderstaande lijst om te weten te komen hoe het geïntegreerd bestandssysteem op uw server functioneert.

- "Directory" op pagina 6
- "Koppeling" op pagina 11
- "Padnaam" op pagina 15
- "Stroombestand" op pagina 15
- "Naamcontinuïteit" op pagina 17
- "Uitgebreide kenmerken" op pagina 17
- "Ondersteuning van scans" op pagina 18

Directory

Een **directory** is een bijzonder object dat wordt gebruikt om een object te zoeken op een naam die u opgeeft. Elke directory bevat een lijst met gekoppelde objecten. Die lijst kan ook andere directory's bevatten.

Het geïntegreerd bestandssysteem kent een hiërarchische directorystructuur waarmee u toegang kunt krijgen tot alle objecten op uw server. U kunt zich deze directorystructuur voorstellen als een omgekeerde boom: de stam steekt omhoog en de takken zitten onder. De takken stellen de directory's in de directoryhiërarchie voor. Deze directorytakken hebben zijtakken: de subdirectory's. Aan de diverse directory- en subdirectorytakken zitten objecten vast, zoals bestanden. Om een object te vinden, moet u een pad opgeven langs de directory's naar de subdirectory waar het object aan vastzit. Van objecten die aan een bepaalde directory vastzitten, zegt men wel dat ze zich "in" die directory bevinden.

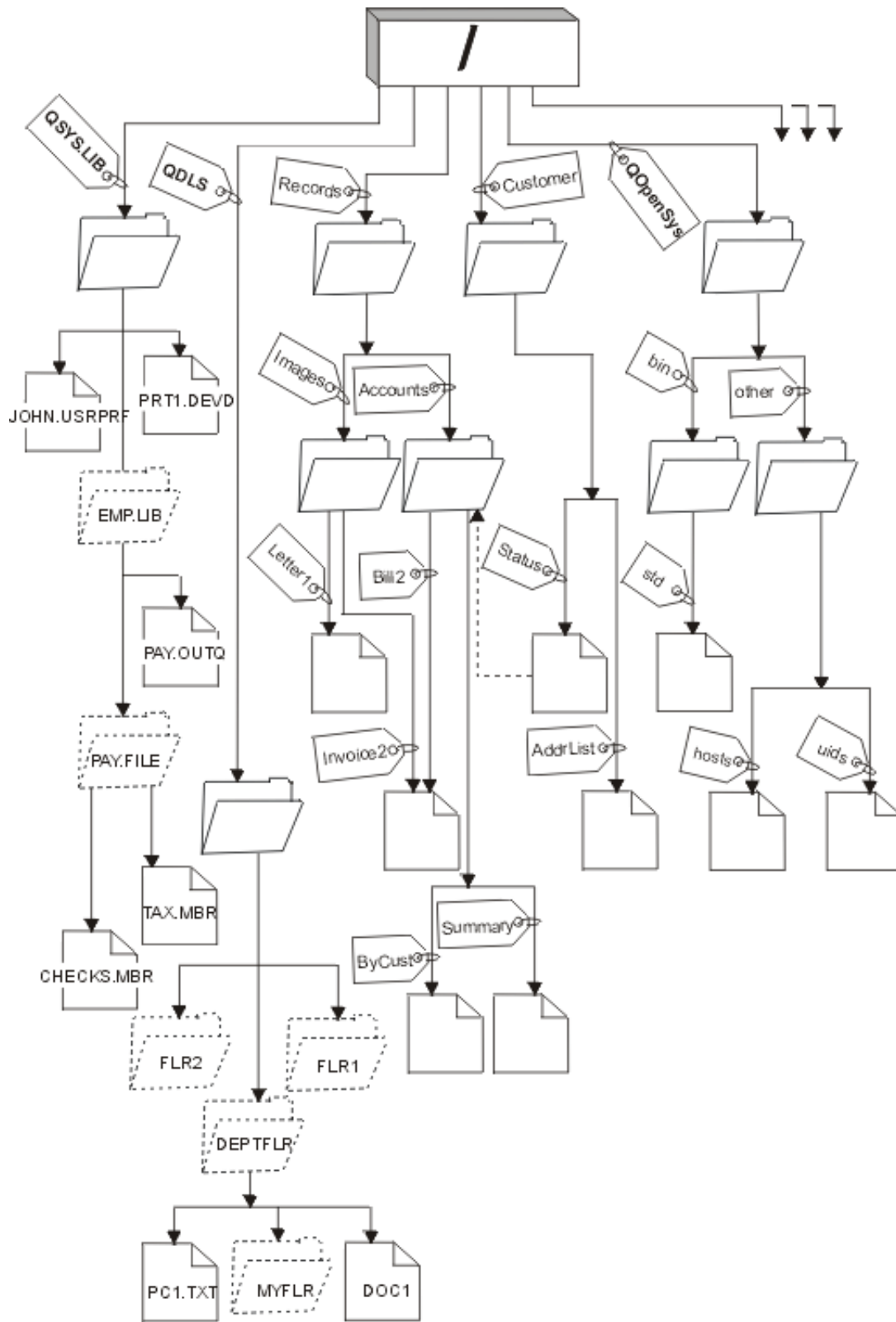
Een bepaalde directorytak, samen met alle zijtakken (subdirectory's) en alle objecten die aan de takken zitten, wordt een **substructuur** genoemd. Ieder bestandssysteem is een belangrijke substructuur in de directorystructuur van het geïntegreerd bestandssysteem. In de substructuren van de bestandssystemen van QSYS.LIB en onafhankelijke ASP QSYS.LIB wordt een bibliotheek op dezelfde manier beheerd als een subdirectory. Objecten in een bibliotheek worden beheerd als objecten in een subdirectory. Omdat databasebestanden objecten kunnen bevatten (databasebestandssecties), worden ze beheerd als subdirectory's en niet als objecten. In het bestandssysteem documentenbibliotheekservices (QDLS-substructuur) worden mappen beheerd als subdirectory's en documenten in mappen als objecten in een subdirectory.

Vanwege de verschillen tussen de bestandssystemen werken handelingen die u in een bepaalde substructuur van de directoryhiërarchie kunt uitvoeren soms niet in een andere substructuur.

De directory-omgeving van het geïntegreerd bestandssysteem komt overeen met die van het DOS-bestandssysteem. Daarnaast beschikt de directory-omgeving over UNIX-functies, zoals de mogelijkheid om een bestand slechts één keer te hoeven opslaan en het via koppelingen langs allerlei paden te openen.

Raadpleeg de volgende onderwerpen over directory's in het geïntegreerd bestandssysteem.

- "Actieve directory" op pagina 8
- "Homedirectory" op pagina 8
- "Beschikbare directory's" op pagina 8
- "*TYPE2-directory's" op pagina 10



Figuur 3. Bestandssystemen en objecten zijn takken aan de directoryboom van het geïntegreerd bestandssysteem

Actieve directory

De **actieve directory** is de eerste directory waarin het besturingssysteem zoekt naar programma's en bestanden en waarin het de tijdelijke bestanden en uitvoer opslaat. Als u opdracht geeft tot bewerking van een object, zoals een bestand, zoekt het systeem dat object eerst in de actieve directory, tenzij u een ander pad opgeeft. De actieve directory lijkt qua concept op de actuele bibliotheek. Hij wordt ook wel de **actieve werkdirectory** of **werkdirectory** genoemd.

Homedirectory

Wanneer u zich aanmeldt op het systeem, wordt de **homedirectory** als de actieve directory gebruikt. De naam van de homedirectory wordt opgegeven in het gebruikersprofiel. Als uw taak wordt gestart, zoekt het systeem in uw gebruikersprofiel naar de naam van uw homedirectory. Als er op het systeem geen directory met die naam bestaat, wordt de "hoofd"directory (/) de homedirectory.

De systeembeheerder die het gebruikersprofiel voor een gebruiker maakt, maakt standaard ook een homedirectory voor de gebruiker. Het is aan te raden om onder de algemene homedirectory voor elke gebruiker een eigen homedirectory te definiëren. De /homedirectory is een subdirectory onder de "hoofd"directory (/). Het systeem verwacht standaard dat de naam van de homedirectory van een gebruiker gelijk is aan de naam van het gebruikersprofiel.

Zo wijst bijvoorbeeld de opdracht `CRTUSRPRF USRPRF(John) HOMEDIR(*USRPRF)` de homedirectory van John toe aan /home/JOHN. Als de directory /home/JOHN niet bestaat, wordt de hoofddirectory (/) Johns homedirectory.

U kunt elke keer wanneer u zich aanmeldt een andere directory dan de homedirectory opgeven als de actieve directory. Dit doet u met de CL-opdracht `CHGCURDIR` (Actieve directory wijzigen), de API `chdir()` of de API `fchdir()`.

De homedirectory die tijdens het starten wordt gekozen, blijft de standaardhomedirectory van een thread. Dit gebeurt ongeacht of uw actieve gebruikersprofiel voor de thread is gewijzigd na het starten. Met de API `QWTCHGJB` (Taak wijzigen) kunt u de homedirectory die voor een thread wordt gebruikt, zo wijzigen dat de homedirectory van het actieve gebruikersprofiel van die thread de homedirectory wordt (of de hoofddirectory (/) als de homedirectory niet bestaat). Secundaire threads nemen altijd de homedirectory over van de thread die deze heeft gemaakt. De actieve directory van het proces verandert niet als u de homedirectory van de thread wijzigt met de opdracht `QWTCHGJB`. De actieve directory functioneert op het procesniveau terwijl de homedirectory op threadniveau functioneert. Het wijzigen van de actieve werkdirectory in een thread heeft een wijziging van het hele proces tot gevolg. Het wijzigen van de homedirectory van een thread heeft geen wijziging van de actieve werkdirectory tot gevolg.

Raadpleeg voor meer gegevens over de API `QWTCHGJB` het onderwerp API's.

Beschikbare directory's

Het geïntegreerd bestandssysteem maakt de volgende directory's als het systeem opnieuw wordt gestart, tenzij deze al bestaan:

/tmp De directory /tmp wordt door toepassingen gebruikt om tijdelijke bestanden op te slaan. Deze directory is een subdirectory van de "hoofd"directory (/), het pad luidt dus /tmp.

Een bestand dat door een toepassing in de directory /tmp is gezet, blijft daar staan totdat het door de toepassing of de gebruiker wordt verwijderd. Het systeem verwijdert bestanden niet automatisch uit /tmp. Ook voert het geen andere bijzondere verwerkingen op bestanden in /tmp uit.

U kunt de directory /tmp en de bestanden daarin beheren met behulp van de gebruikersschermen en -opdrachten die het geïntegreerd bestandssysteem ondersteunen. U kunt bijvoorbeeld met het scherm Werken met objectkoppelingen of de opdracht `WRKLNK` de directory /tmp en de bestanden in die directory kopiëren, wissen en hernoemen. Alle gebruikers krijgen de machtiging *ALL voor die directory, wat inhoudt dat zij de meeste geldige acties mogen uitvoeren op die directory.

Een toepassing kan de API's die het geïntegreerd bestandssysteem ondersteunen, gebruiken om /tmp en de bestanden daarin te beheren (zie "Toegang met behulp van API's" op pagina 80). De toepassing kan bijvoorbeeld een bestand in /tmp wissen met de API unlink().

Als /tmp wordt verwijderd, wordt deze directory automatisch opnieuw gemaakt wanneer het systeem weer wordt gestart.

De directory /tmp wordt nu geleverd met het kenmerk "restrict rename and unlink" ingeschakeld. Dit kenmerk kan worden gewijzigd met de opdracht CHGATR (Kenmerk wijzigen). De gebruiker mag dit kenmerk uitschakelen, maar het wordt automatisch weer ingeschakeld als het systeem opnieuw wordt gestart. Als het kenmerk is ingeschakeld, kent /tmp de volgende beperkingen:

- Hernoemen en ontkoppelen is beperkt tot objecten in deze directory. Objecten kunnen aan deze directory worden gekoppeld, maar worden hernoemd of ervan worden losgekoppeld, tenzij er sprake is van een of meer van de onderstaande feiten:
 1. De gebruiker is de eigenaar van het object
 2. De gebruiker is de eigenaar van de directory
 3. De gebruiker heeft de speciale machtiging *ALLOBJ

/home Systeembeheerders gebruiken de directory /home om een aparte directory voor elke gebruiker in op te slaan. De systeembeheerder stelt de homedirectory die bij het gebruikersprofiel hoort meestal zo in dat dit de gebruikersdirectory is in /home, bijvoorbeeld /home/john. Zie "Homedirectory" op pagina 8 voor meer informatie.

/etc De directory /etc bevat bestanden die betrekking hebben op het beheer en de configuratie en andere systeembestanden.

/usr De directory /usr heeft subdirectory's die informatie bevatten die door het systeem wordt gebruikt. Bestanden in /usr worden doorgaans niet vaak gewijzigd.

/usr/bin

De directory /usr/bin bevat de standaardhulpprogramma's.

/QIBM

De directory /QIBM is de systeemdirectory en wordt bij het systeem geleverd.

/QIBM/ProdData

De directory /QIBM/ProdData is een systeemdirectory voor gegevens van gelicentieerde programma's.

/QIBM/UserData

De directory /QIBM/UserData is een systeemdirectory voor gebruikersgegevens van gelicentieerde programma's, zoals configuratiebestanden.

/QOpenSys/QIBM

De directory /QOpenSys/QIBM is een systeemdirectory voor het QOpenSys-bestandssysteem.

/QOpenSys/QIBM/ProdData

De directory /QOpenSys/QIBM/ProdData is een systeemdirectory voor het QOpenSys-bestandssysteem en wordt gebruikt voor de gegevens van gelicentieerde programma's.

/QOpenSys/QIBM/UserData

De directory /QOpenSys/QIBM/UserData is een systeemdirectory voor het QOpenSys-bestandssysteem en wordt gebruikt voor gebruikersgegevens van gelicentieerde programma's, zoals configuratiebestanden.

/asp_naam/QIBM

De directory /asp_naam/QIBM is de systeemdirectory voor alle onafhankelijke ASP's op uw systeem, waarbij asp_naam de naam van de onafhankelijke ASP is.

/asp_naam/QIBM/UserData

De directory /asp_naam/QIBM/UserData is een systeemdirectory voor gebruikersgegevens van

gelicentieerde programma's, zoals de configuratiebestanden voor alle onafhankelijke ASP's op uw systeem, waarbij asp_name de naam van de onafhankelijke ASP is.

| /dev De directory /dev bevat diverse systeembestanden en -directory's.

| /dev/xti

| De directory /dev/xti bevat de UDP- en TCP-stuurprogramma's. Zie "UDP- en TCP-apparatuur
| in het bestandssysteem hoofddirectory (/)" op pagina 30 voor meer informatie.

***TYPE2-directory's**

De hoofddirectory (/), QOpenSys en door de gebruiker gedefinieerde bestandssystemen (UDFS) in het geïntegreerd bestandssysteem ondersteunen de *TYPE2-directory-indeling. De *TYPE2-directory-indeling is een verbetering van de oorspronkelijke *TYPE1-directory-indeling. *TYPE2-directory's hebben een andere interne structuur en implementatie dan *TYPE1-directory's.

De voordelen van de *TYPE2-directory's zijn:

- Betere prestaties
- Grotere betrouwbaarheid
- Meer functionaliteit
- Minder hulpgeheugen (in veel gevallen).

Dankzij *TYPE2-directory's zijn de prestaties van bestandssystemen verbeterd ten opzichte van *TYPE1-directory's, met name bij het maken en wissen van directory's.

*TYPE2-directory's zijn betrouwbaarder dan *TYPE1-directory's. Na een systeemcrash worden *TYPE2-directory's volledig hersteld, tenzij er sprake was van een fout in het hulpgeheugen. *TYPE1-directory's kunnen soms alleen volledig worden hersteld met de opdracht RCLSTG (Geheugenruimte opnieuw beschikbaar maken).

*TYPE2-directory's hebben de volgende extra functies:

1. *TYPE2-directory's ondersteunen het wijzigen van bestandsnamen met alleen hoofdletters in namen met kleine letters, bijvoorbeeld van A naar a.
- | 2. Een object in een *TYPE2-directory kan maximaal 1 miljoen koppelingen hebben, vergeleken met
| 32.767 in een *TYPE1-directory. Dit betekent dat u 1 miljoen vaste koppelingen kunt hebben naar een
| stroombestand en dat een *TYPE2-directory 999.998 subdirectory's kan bevatten.
3. Met behulp van iSeries Navigator wordt de itemlijst automatisch gesorteerd in binaire volgorde als u een directory van het *TYPE2 opent.
- | 4. Sommige nieuwe functies, zoals ondersteuning voor scans in geïntegreerde bestandssystemen, zijn
| alleen beschikbaar voor objecten in *TYPE2-directory's.

*TYPE2-directory's met minder dan 350 objecten gebruiken standaard minder hulpgeheugenruimte dan *TYPE1-directory's met hetzelfde aantal objecten. *TYPE2-directory's met meer dan 350 objecten zijn gemiddeld 10 procent groter dan *TYPE1-directory's.

U kunt op verschillende manieren de beschikking krijgen over *TYPE2-directory's op uw systeem:

- | • Nieuwe iSeries-servers met reeds geïnstalleerde OS/400 V5R2 of hoger zijn voorzien van *TYPE2-
| directory's. Conversie is niet nodig voor de hoofddirectory (/), QOpenSys en UDFS's in ASP's 1-32.
- | • Met een schone installatie van OS/400 V5R2 of hoger op een iSeries-server worden *TYPE2-directory's
| aangemaakt. Conversie is niet nodig voor de hoofddirectory (/), QOpenSys en UDFS's in ASP's 1-32.
- | • Als u de bestandssystemen hebt geconverteerd met het conversiehulpprogramma V5R1 of V5R2.
- | • Als de UDFS's in een onafhankelijke ASP nog niet zijn geconverteerd naar *TYPE2, gebeurt dat de eer-
| ste keer dat de onafhankelijke ASP online wordt gezet naar een systeem waarop OS/400 V5R2 of hoger
| is geïnstalleerd.

- Alle andere ondersteunde bestandssystemen die nog *TYPE1-directory's gebruiken, behalve UDFS's op onafhankelijke ASP's, worden automatisch door het systeem geconverteerd. Deze conversie begint met een na de installatie van V5R3 en is doorgaans geen grote belasting voor uw systeem. Zie "Directory's converteren van *TYPE1 naar *TYPE2" op pagina 84 voor extra informatie.

Om te bepalen welke directory-indeling de bestandssystemen op uw server hebben, gebruikt u de opdracht CVTDIR (Directory converteren):

```
CVTDIR OPTION(*CHECK)
```

Opmerking: *TYPE2-directory's worden ondersteund door OS/400 V5R1 of hoger, maar er bestaat verschil ten opzichte van de normale ondersteuning van *TYPE2-directory's. Raadpleeg voor meer informatie *TYPE2-directory's gebruiken in OS/400 V5R1 of V5R2.

Raadpleeg voor meer informatie over *TYPE2-directory's "Directory's converteren van *TYPE1 naar *TYPE2" op pagina 84.

- ***TYPE2-directory's gebruiken in OS/400 V5R1 of V5R2:** De hoofddirectory (/), QOpenSys en door de gebruiker gedefinieerde bestandssystemen (UDFS) in het geïntegreerd bestandssysteem ondersteunen de *TYPE2-directory-indeling in OS/400 V5R1, V5R2 en hoger. De *TYPE2-directory-indeling is een verbetering van de oorspronkelijke *TYPE1-directory-indeling. *TYPE2-directory's hebben een andere interne structuur dan *TYPE1-directory's en zijn betrouwbaarder en functioneler.


- Als u versie V5R1 of V5R2 hebt geïnstalleerd, kunt u de directory's met behulp van het juiste conversie-hulpprogramma naar *TYPE2 converteren. Kort na het installeren van V5R3 begint het systeem bestandssystemen die nog *TYPE1-directory's ondersteunen automatisch te converteren naar *TYPE2-directory's. Om deze automatische conversie te voorkomen, kunt u converteren naar *TYPE2-directory's voordat u V5R3 installeert. Meer informatie vindt u in "Directory's converteren van *TYPE1 naar *TYPE2" op pagina 84.

- Ondersteuning voor *TYPE2-directory's in V5R2 is beschikbaar in het iSeries Informatiecentrum via de opdracht CVTDIR (Directory converteren).

Ondersteuning voor *TYPE2-directory's in V5R1 is beschikbaar via fixes. Het conversie-hulpprogramma is enigszins anders dan voor versie V5R2. Voor de volledige documentatie over *TYPE2-directory's in V5R1 raadpleegt u APAR II13161. De APAR is toegankelijk op de volgende manieren:

1. Download de APAR naar uw iSeries-server en bekijk het rapport. Gebruik de volgende opdrachten:

```
SNDPTFORD PTFID((II13161))  
DSPPTFCVR LICPGM(INFOAS4) SELECT(II13161)
```

2. Ga naar <http://www.ibm.com/eserver/iseries/support/supporthome.nsf/document/1000045>  en bekijk de APAR online. Zie **Probleemoplossing** —> **Technische databases** —> **APAR's (Authorized Program Analysis Reports)** —> **V5R1 APAR's** —> **APAR nummer II13161.**

Koppeling

Een **koppeling** is een verbinding, voorzien van een naam, tussen een directory en een object. Een gebruiker of een programma kan aangeven waar de server een bepaald object kan vinden door een koppeling, inclusief naam, naar het object op te geven. Een koppeling kan worden gebruikt als een pad of als een gedeelte van een padnaam.

Voor gebruikers van op directory's gebaseerde bestandssystemen is het handig om een object, zoals een bestand, te zien als iets dat een naam heeft om voor de server herkenbaar te zijn. In feite maakt het pad het object herkenbaar. Soms kunt u een object openen door alleen de "naam" op te geven. Dat kan alleen omdat het systeem zo ontworpen is dat het onder bepaalde omstandigheden het directorygedeelte van

het pad kan "raden". Het idee van een koppeling maakt gebruik van het feit dat de koppeling zich in het pad bevindt dat het object identificeert. De naam wordt gegeven aan de koppeling in plaats van aan het object.

Als u eenmaal gewend bent aan het feit dat de koppeling de naam heeft en niet het object dienen zich ongekende mogelijkheden aan. U kunt verschillende koppelingen maken naar hetzelfde object. Twee gebruikers kunnen bijvoorbeeld gemeenschappelijk gebruik maken van een bestand door elk vanuit hun eigen gebruikersdirectory een koppeling te maken naar het bestand (zie "Homedirectory" op pagina 8). Bepaalde typen koppelingen kunnen functioneren met andere bestandssystemen en bestaan zonder een object.

Er zijn twee typen koppelingen: "Vaste koppeling" en "Symbolische koppeling" op pagina 13. Bij het kiezen van padnamen in programma's kunt u kiezen tussen vaste en symbolische koppelingen. Beide typen hebben voor- en nadelen. Hieronder ziet u een overzicht van de voor- en nadelen onder verschillende omstandigheden:

Tabel 1. Vergelijking van vaste en symbolische koppelingen

Item	Vaste koppeling	Symbolische koppeling
Naamomzetting	Sneller. Een vaste koppeling bevat een directe verwijzing naar het object.	Langzamer. Een symbolische koppeling bevat een pad naar het object. Dit pad moet eerst worden omgezet voordat het object wordt gevonden.
Aanwezigheid object	Vereist. Om een vaste koppeling naar een object te kunnen maken, moet het object bestaan.	Optioneel. Een symbolische koppeling naar object is mogelijk ook al bestaat dat object niet.
Objecten wissen	Beperkt. Alle vaste koppelingen naar een object moeten worden ontkoppeld (verwijderd) voordat het object kan worden gewist.	Onbeperkt. Een object kan worden gewist, ook al bestaan er koppelingen die ernaar verwijzen.
Statische objecten (kenmerken blijven ongewijzigd)	Sneller. Naamomzetting heeft voor een statisch object de hoogste functionele prioriteit. Omzetting van een naam gebeurt sneller bij gebruik van een vaste koppeling.	Langzamer. Omzetting van een naam gaat langzamer bij gebruik van een symbolische koppeling.
Bereik	Beperkt. Vaste koppelingen functioneren alleen in hun eigen bestandssysteem.	Onbeperkt. Symbolische koppelingen kunnen ook met andere bestandssystemen functioneren.

Raadpleeg de volgende onderwerpen voor meer informatie over koppelingen:

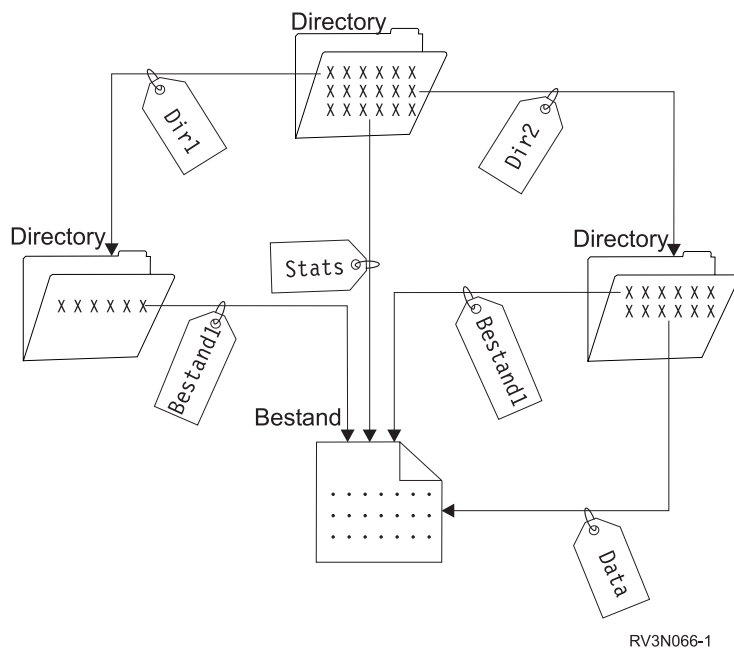
- Vaste koppeling
- Symbolische koppeling

Vaste koppeling

Een **vaste koppeling** wordt soms gewoon koppeling genoemd en kan alleen bestaan als er sprake is van een koppeling naar een bestaand object. Als in een directory een object wordt gemaakt, door bijvoorbeeld een bestand naar een directory te kopiëren, wordt de eerste vaste koppeling gemaakt tussen de directory en het object. Gebruikers en toepassingen kunnen andere vaste koppelingen toevoegen. Elke vaste koppeling wordt in de directory aangegeven met een apart directorygegeven. Koppelingen in één directory kunnen niet dezelfde naam hebben; koppelingen in verschillende directory's wel.

Indien het bestandssysteem dit ondersteunt, kunnen er verschillende vaste koppelingen naar een object bestaan, hetzij vanuit dezelfde directory, hetzij vanuit verschillende directory's. De enige uitzondering is

als het object een directory is. Er kan maar één vaste koppeling bestaan van een directory naar een andere directory.



Figuur 4. Iedere vaste koppeling wordt gedefinieerd door een directorygegeven.

Vaste koppelingen kunnen worden verwijderd zonder het voortbestaan van een object in gevaar te brengen, mits ten minste één vaste koppeling naar het object gehandhaafd blijft. Wanneer de laatste vaste koppeling wordt verwijderd, wordt het object van de server verwijderd, tenzij het object geopend is een toepassing. De toepassing(en) waarin het object geopend is, kan (kunnen) het object blijven gebruiken tot dat het object in de betreffende toepassing wordt afgesloten. Als het object wordt afgesloten in de laatste toepassing waarin het wordt gebruikt, wordt het object van de server verwijderd. Nadat de laatste vaste koppeling is verwijderd, kan een object niet meer worden geopend.

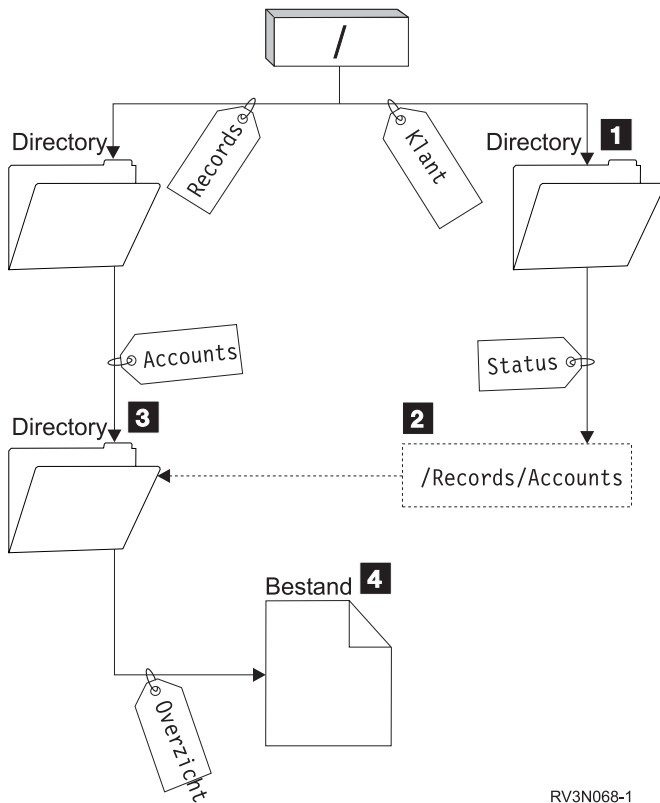
Het concept vaste koppeling kan ook worden toegepast in de bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB en in de documentenbibliotheekservices (QDLS), zij het met een beperking. Een bibliotheek heeft één vaste koppeling naar elk object in de bibliotheek. Een map heeft één vaste koppeling naar elk document in die map. Meerdere vaste koppelingen naar *hetzelfde object* zijn echter niet toegestaan in QSYS.LIB, onafhankelijke ASP QSYS.LIB en QDLS.

Een vaste koppeling kan niet in meerdere bestandssystemen worden gebruikt. Een directory in het bestandssysteem QOpenSys bijvoorbeeld kan geen vaste koppeling hebben met een object in het bestandssysteem QSYS.LIB of onafhankelijke ASP QSYS.LIB, of naar een document in het bestandssysteem QDLS.

Symbolische koppeling

Een symbolische koppeling, ook wel een zachte koppeling genaamd, is een padnaam die in een bestand is opgesloten. Als het systeem een symbolische koppeling tegenkomt, volgt het de padnaam die de symbolische koppeling opgeeft en gaat het door op een eventueel pad dat volgt op de symbolische koppeling. Als de padnaam begint met een / gaat het systeem terug naar de "hoofddirectory" en volgt het vanaf dat punt het pad. Als de padnaam niet met een / begint, gaat het systeem terug naar de direct voorafgaande directory en volgt het de padnaam in de symbolische koppeling vanaf die directory.

Hieronder volgt een voorbeeld van hoe een symbolische koppeling kan worden gebruikt:



RV3N068-1

Figuur 5. Een voorbeeld van het gebruik van een symbolische koppeling

U selecteert een menuoptie om de stand van zaken (status) van het klantenbestand af te beelden. Het programma dat het menu afbeeldt, gebruikt het volgende pad:

`/Klant/Status/Overzicht`

Het systeem volgt de koppeling *Klant* die naar een directory **1** leidt en volgt dan de koppeling *Status*. De koppeling *Status* is een symbolische koppeling die een pad **2** bevat. Omdat het pad met een / begint, gaat het systeem terug naar de "hoofddirectory" en volgt het achtereenvolgens de koppelingen *Records* en *Accounts*. Dit pad leidt naar een andere directory **3**. Nu voltooit het systeem het pad dat door het programma in de padnaam werd gegeven. Het volgt de koppeling *Overzicht* die naar een bestand leidt **4** met de gegevens die u nodig hebt.

In tegenstelling tot een vaste koppeling is een symbolische koppeling een object (objecttype *SYMLNK). Hij kan bestaan zonder naar een bestaand object te verwijzen. U kunt een symbolische koppeling bijvoorbeeld gebruiken om een pad op te geven naar een bestand dat nog moet worden toegevoegd of later wordt vervangen.

Ook kan een symbolische koppeling buiten het eigen bestandssystemen functioneren. Als u in een bepaald bestandssysteem werkt, kunt u bijvoorbeeld een symbolische koppeling maken naar een bestand in een ander bestandssysteem. Hoewel de bestandssystemen QSYS.LIB, onafhankelijke ASP QSYS.LIB en QDLS het maken en opslaan van symbolische koppelingen niet ondersteunen, kunt u een symbolische koppeling maken in het bestandssysteem hoofddirectory (/) of QOpenSys waarmee u het volgende kunt doen:

- Een databasebestandssectie openen in de bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB.
- Een document openen in het bestandssysteem QDLS.

Zie ook Tabel 1 op pagina 12.

Padnaam

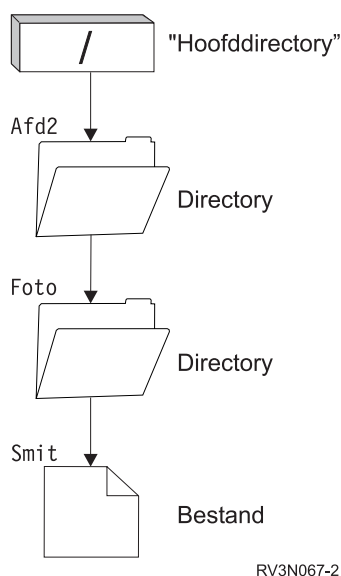
Een **padnaam** (op sommige systemen **pad** geheten) vertelt de server waar een object zich bevindt. De padnaam wordt opgegeven als een reeks directorynamen gevolgd door de naam van het object. De verschillende directory's en de objectnaam worden van elkaar gescheiden met een schuine streep naar rechts (/), bijvoorbeeld:

```
directory1/directory2/bestand
```

Bij geïntegreerd bestandssysteem-opdrachten kunt u voor het gemak de schuine streep naar rechts vervangen door een schuine streep naar links (\).

U kunt een padnaam op twee manieren aangeven:

- Een **absolute padnaam** begint op het hoogste niveau, de "hoofddirectory", aangegeven met de schuine streep naar rechts (/). Neem bijvoorbeeld het volgende pad, van de hoofddirectory (/) naar het bestand Smith.



Figuur 6. De onderdelen van een padnaam

De absolute padnaam naar het bestand Smith luidt als volgt:

```
/Dept2/Foto/Smith
```

De absolute padnaam heet ook wel **volledige padnaam**.

- Als de padnaam niet met / begint, gaat het systeem er vanuit dat het pad begint in uw actieve directory. Dit type padnaam heet een **relatieve padnaam** (of pad vanaf de actieve directory). Als bijvoorbeeld uw actieve directory Dept2 is en deze een subdirectory heeft genaamd Foto, met een bestand getiteld Smith, dan ziet de relatieve padnaam er als volgt uit:

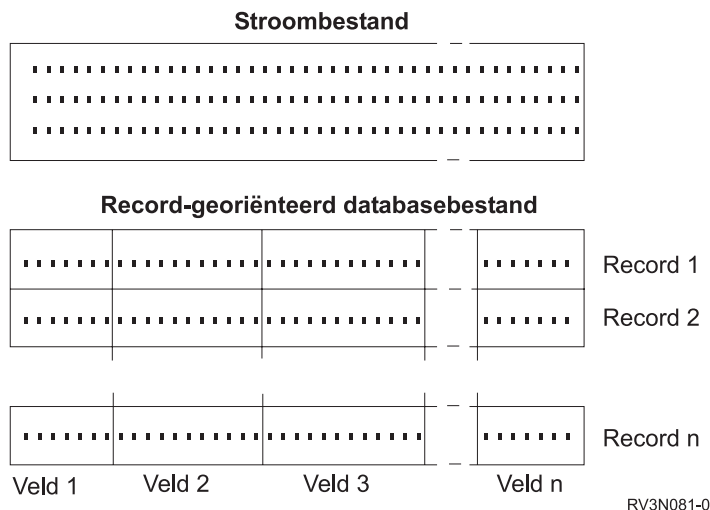
```
Foto/Smith
```

Wat opvalt, is dat het pad niet de naam van de actieve directory bevat. Het eerste gedeelte van de naam is de directory of het object in het *eerste niveau onder* de actieve directory.

Stroombestand

Een **stroombestand** is een willekeurig toegankelijke opeenvolging van bytes waarop geen door het systeem opgelegde structuur van toepassing is. Het geïntegreerd bestandssysteem ondersteunt het opslaan van en werken met informatie in de vorm van stroombestanden. Documenten die de mappen op uw server worden opgeslagen zijn stroombestanden. Andere voorbeelden van stroombestanden zijn PC-bestanden en de bestanden op UNIX-systemen. Een geïntegreerd bestandssysteem stroombestand is een systeemobject en wel van het type *STMF.

Om stroombestanden beter te kunnen begrijpen, is het handig om ze te vergelijken met iSeries-databasebestanden. Een databasebestand is gericht op records. Het heeft vooraf gedefinieerde onderverdelingen die bestaan uit een of meer velden met specifieke kenmerken, zoals lengte en gegevenstype.



Figuur 7. Vergelijking tussen een stroombestand en een recordsbestand

Stroombestanden en recordsbestanden hebben niet dezelfde structuur. Dit verschil heeft invloed op de manier waarop de bestanden worden gebruikt. De structuur van een bestand beïnvloedt de manier waarop een toepassing communiceert met het bestand en de plaats in de toepassing waar elk bestandstype het best kan worden gebruikt. Een recordsbestand is bijvoorbeeld geschikt voor het opslaan van klantgegevens, zoals naam, adres en saldo. Een recordsbestand biedt de mogelijkheid om deze vooraf gedefinieerde velden afzonderlijk te openen en te bewerken met de uitgebreide programmeervoorzieningen van uw server. Een stroombestand daarentegen is beter geschikt voor de opslag van informatie zoals de foto van een klant, die bestaat uit een aangesloten reeks bits die de kleurschakeringen voorstellen. Stroombestanden zijn met name geschikt voor het opslaan van gegevensreeksen, zoals tekst, afbeeldingen, geluid en bewegende beelden.

Een bestand heeft als bestandsindeling *TYPE1-stroombestand of *TYPE2-stroombestand. De bestandsindeling is afhankelijk van de softwareversie van OS/400 waarmee het bestand is gemaakt of, indien het bestand gemaakt is op een door de gebruiker gedefinieerd bestandssysteem, van de waarde die voor dat bestandssysteem is opgegeven.

Raadpleeg voor meer informatie over stroombestanden in het geïntegreerd bestandssysteem:

- "Gegevens kopiëren tussen stroombestanden en databasebestanden" op pagina 94.
- "*TYPE1-stroombestanden".
- "*TYPE2-stroombestanden".

*TYPE1-stroombestanden

Een *TYPE1-stroombestand heeft dezelfde indeling als stroombestanden die zijn gemaakt met softwareversies van OS/400 die ouder zijn dan V4R4. De minimumgrootte bedraagt 4096 bytes. *TYPE1-stroombestanden hebben een maximale objectgrootte van 256 gigabytes.

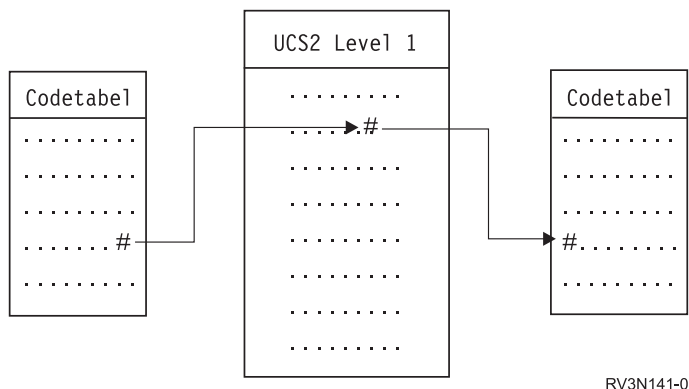
*TYPE2-stroombestanden

*TYPE2-stroombestanden werden geïntroduceerd in versie V4R4 van OS/400 en beschikken over geavanceerde bestandstoegang. In V5R3 hebben *TYPE2-stroombestanden een maximale objectgrootte van ongeveer 1TB in bestandssystemen "hoofddirectory" (/) en QOpenSys en door de gebruiker gedefinieerde bestandssystemen. *TYPE-stroombestanden van voor V5R3 hebben een maximale grootte van ongeveer

- | 256 gigabytes. Ze zijn ook geschikt voor geheugenstoeuwijzingsschema's en het opgeven van een kenmerk voor de optimalisering van de hoofdgeheugentoewijzing. Alle bestanden die zijn gemaakt met V4R4 en hoger zijn *TYPE2-stroombestanden, tenzij ze zijn gemaakt in een door de gebruiker gedefinieerd bestandssysteem waarin *TYPE1 is opgegeven als bestandsindeling.
- | **Opmerking:** Bestanden die groter zijn dan 256 gigabytes kunnen niet worden opgeslagen in of teruggezet naar systemen ouder dan V5R3.

Naamcontinuïteit

Wanneer u de bestandssystemen "hoofddirectory" (/), QOpenSys of door de gebruiker gedefinieerde bestandssystemen gebruikt, kunt u gebruik maken van een systeemfunctie die ervoor zorgt dat tekens in objectnamen niet veranderen. Dit gaat ook op als u deze bestandssystemen gebruikt op iSeries-servers en aangesloten apparatuur die gebruik maken van verschillende codeerschema's (codetabellen). Uw server slaat de tekens van de namen op in een 16-bitsformaat dat UCS2 Level 1 heet (ook wel **Unicode** genoemd) voor *TYPE1-directory's en UTF-16 voor *TYPE2-directory's. Raadpleeg *TYPE2-directory's voor meer informatie over directory-indelingen. UCS2 Level 1 en UTF-16 zijn subsets van de ISO 10646-standaard. Wanneer de naam wordt gebruikt, vertaalt het systeem de opgeslagen vorm van de tekens in de juiste tekens voor de gebruikte codetabel. De namen van uitgebreide kenmerken die bij een object horen, worden op dezelfde manier geconverteerd.



Figuur 8. Tekens gelijk houden ongeacht het codeerschema

Dankzij deze functie kunt u vanaf apparatuur die een andere codetabel gebruikt makkelijker samenwerken met een server. Een PC-gebruiker kan bijvoorbeeld een bestand op een iSeries-server openen met dezelfde bestandsnaam, ook al maakt de PC gebruik van een andere codetabel dan uw server. De conversie van de ene codetabel naar de andere wordt automatisch door uw server uitgevoerd. Vanzelfsprekend moet het apparaat gebruik maken van een codetabel die de tekens bevat die in de naam worden gebruikt.

Uitgebreide kenmerken

Een uitgebreid kenmerk is informatie die bij een object hoort en aanvullende informatie verstrekt over dat object. Een uitgebreid kenmerk bestaat uit een naam, die wordt gebruikt om ernaar te verwijzen, en een waarde. De waarde kan bestaan uit tekst, binaire gegevens of andere soorten gegevens.

Een uitgebreid kenmerk voor een object bestaat net zo lang als het object bestaat.

Er zijn verschillende soorten uitgebreide kenmerken en ze kunnen uiteenlopende informatie bevatten. Het is handig om met name de volgende drie uitgebreide kenmerken te kennen:

.SUBJECT

Een korte beschrijving van de inhoud of de functie van het object.

.TYPE Het gegevenstype van het object. Het gegevenstype kan tekst zijn, binaire gegevens, sourcecode van een programma, een gecompileerd programma of andere informatie.

.CODEPAGE

De codetabel die voor het object moet worden gebruikt. De codetabel die voor het object wordt gebruikt, wordt ook voor het uitgebreide kenmerk bij het object gebruikt.

Een punt (.) als eerste teken van de naam betekent dat het uitgebreide kenmerk een uitgebreide standaardkenmerk van het systeem is en gereserveerd is voor gebruik binnen het systeem.

Er zijn diverse objecten in diverse bestandssystemen die uitgebreide kenmerken kunnen hebben. De bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB ondersteunen drie vooraf gedefinieerde uitgebreide kenmerken: .SUBJECT, .TYPE en .CODETABEL. In het bestandssysteem documentenbibliotheek-services (QDLS) kunnen mappen en documenten elk type uitgebreid kenmerk hebben. Sommige mappen en documenten hebben een uitgebreid kenmerk, andere niet. In de bestandssystemen "hoofddirectory" (/), QOpenSys en door de gebruiker gedefinieerde bestandssystemen kunnen alle directory's, stroombestanden en symbolische koppelingen alle soorten uitgebreide kenmerken hebben. Er zijn er echter die helemaal geen uitgebreide kenmerken hebben.

De opdracht WRKLNK (Werken met objectkoppelingen) kan worden gebruikt om het uitgebreide kenmerk .SUBJECT voor een object af te beelden. Er bestaat geen andere ondersteuning in het geïntegreerd bestandssysteem waarmee toepassingen en gebruikers toegang kunnen krijgen tot uitgebreide kenmerken en deze kunnen wijzigen. De enige uitzonderingen hierop zijn de CL-opdrachten DSPUDFS (UDFS afbeelden) en DSPMFSINF (Informatie over aangekoppeld bestandssysteem afbeelden), waarmee uitgebreide kenmerken aan gebruikers kunnen worden getoond.

Uitgebreide kenmerken die bij sommige objecten in het QDLS horen, kunnen echter worden gewijzigd via interfaces die door het hiërarchisch bestandssysteem worden aangeboden. Zie "Bestandssysteem voorzieningen documentenbibliotheek (QDLS)" op pagina 43 en "Bestandssysteem optische media (QOPT)" op pagina 45 voor meer informatie over deze bestandssystemen.

Als een client-PC is aangesloten op een iSeries-server via OS/2 of Windows, kunnen de programmeerinterfaces van de respectievelijke besturingssystemen (zoals DosQueryFileInfo en DosSetFileInfo) worden gebruikt om een query uit te voeren op uitgebreide kenmerken voor elk bestandsobject en deze in te stellen. OS/2-gebruikers kunnen ook de uitgebreide kenmerken van een object op het bureaublad wijzigen in het venster Instellingen. Selecteer daartoe Instellingen in het voorgrondmenu van het object.

Bij het definiëren van uitgebreide kenmerken gelden de volgende richtlijnen voor de naamgeving:

- De naam van een uitgebreid kenmerk kan maximaal 255 tekens lang zijn.
- Gebruik geen punt (.) als eerste teken van de naam. Een uitgebreid kenmerk dat met een punt begint, wordt gezien als een uitgebreid standaardkenmerk van het systeem.
- Om naamconflicten zo veel mogelijk te vermijden, dient u voor uitgebreide kenmerken een consistente naamgevingsstructuur te hanteren. De volgende vorm wordt aanbevolen:

BedrijfsnaamProductnaam.Kenmerk_Naam

Ondersteuning van scans

iSeries beschikt nu over de mogelijkheid om scans uit te voeren op objecten in het geïntegreerd bestandssysteem. Deze nieuwe functie geeft iSeries-gebruikers meer mogelijkheden: u kunt scans uitvoeren op diverse items; de gebruiker bepaalt wanneer de scan plaatsvindt; de gebruiker beslist welke acties op basis van de scan moeten worden ondernomen. Raadpleeg voor meer informatie over het uitvoeren van scans de volgende onderwerpen:

- "Verwante systeemwaarden" op pagina 20
- "Zoekvoorvallen" op pagina 21
- "Zoekvoorbeelden" op pagina 19

| De twee nieuwe exit-points die verband houden met deze functie zijn:

- | • QIBM_QP0L_SCAN_OPEN - Exitprogramma geïntegreerd bestandssysteem scannen bij poging tot openen

| Voor dit exit-point wordt het exitprogramma Geïntegreerd bestandssysteem scannen bij poging tot openen opgeroepen om een scan uit te voeren wanneer een object in het geïntegreerd bestandssysteem onder bepaalde voorwaarden wordt geopend.

- | • QIBM_QP0L_SCAN_CLOSE - Exitprogramma geïntegreerd bestandssysteem scannen bij poging tot afsluiten

| Voor dit exit-point wordt het exitprogramma Geïntegreerd bestandssysteem scannen bij poging tot afsluiten opgeroepen om een scan uit te voeren wanneer een object in het geïntegreerd bestandssysteem onder bepaalde voorwaarden wordt afgesloten.

| Voor meer informatie over al dan niet uitvoeren van scans op objecten raadpleegt u "Hiermee stelt u in of objecten moeten worden gescand" op pagina 118.

| **Opmerking:** Alleen objecten in bestandssystemen die volledig zijn geconverteerd naar *TYPE2-directory's worden gescand.

| **Zoekvoorbeelden**

| Hieronder vindt u een aantal voorbeelden van het soort scans dat een exitprogramma kan uitvoeren:

- | • Virussen

| Een exitprogramma kan op virussen scannen. Als er in een bestand een virus wordt ontdekt, kan het anti-virusprogramma hierop reageren door het probleem op te lossen of te proberen het virus te isoleren. Omdat de iSeries in dat geval zelf niet met het virus besmet raakt, is het resultaat dat er tussen servers minder virussen worden verzonden.

- | • Oproep om te zien wanneer een bestand is geopend

| U kunt ook scannen om te zien wanneer een bestand is geopend. Met deze scan kunt u de datum en tijd achterhalen waarop bepaalde bestanden zijn geopend. Dit is nuttige informatie als u bijvoorbeeld het gedrag van bepaalde gebruikers wilt opsporen.

| De scan kan op twee momenten plaatsvinden. Dit hangt af van de instellingen van de systeemwaarden en de samenstelling van de scanomgeving. In de onderstaande lijst worden de twee momenten beschreven.

| 1. Runtime-scan

| Een runtime-scan is een scan op een bestand of op bestanden tijdens de normale, alledaagse activiteiten. Dit waarborgt de integriteit van uw bestanden telkens al ze worden geopend. Door tijdens uw normale activiteiten te scannen, zorgt u ervoor dat uw bestanden actueel zijn in relatie tot datgene waarop u scant.

| **Voorbeeld van een runtime-scan op virussen**

| Op een woensdag wilt u vanaf uw PC een bestand openen op het geïntegreerd bestandssysteem. Wanneer het bestand wordt geopend vanaf uw PC wordt er een scan uitgevoerd omdat een exitprogramma voor openen is geregistreerd en de systeemwaarde QSCANFS is ingesteld om bestanden in de bestandssystemen "hoofddirectory" (/), QOpenSys en UDFS te scannen. De scan vindt een virus, waarna het anti-virusprogramma het probleem verhelpt. Omdat het exitprogramma het bestand heeft gerepareerd, is het niet langer geïnfecteerd en kan de PC niet besmet raken via het openen van het bestand. Het virus kan zich dus niet verspreiden.

| Stel dat u geen runtime-scan op virussen uitvoert. U opent dus vanaf uw PC het geïnfecteerde bestand, waardoor het virus zich verspreidt naar uw PC. Door een runtime-scan uit te voeren, voorkomt u dus dat het virus zich verspreidt naar uw PC.

| Het grootste nadeel van deze methode is dat scans resourcetijd in beslag nemen. Als u een bestand wilt openen, moet u wachten totdat de scan is voltooid voordat het kunt gebruiken. Het systeem probeert ervoor te zorgen dat het scannen alleen wordt gedaan als het nodig is en niet elke keer dat u het bestand opent. Zie QIBM_QP0L_SCAN_OPEN en QIBM_QP0L_SCAN_CLOSE voor meer informatie.

2. Meervoudige of handmatig geactiveerde scans
- Deze optie gebruikt u als u een scan op veel items tegelijk wilt uitvoeren. In dit geval programmeert u de scan in het weekend wanneer de server een tijdje niets te doen heeft. Dit zou erg weinig belasting opleveren wanneer u bestanden opent tijdens de gewone dagelijkse activiteiten. Aangezien de scan offline wordt uitgevoerd, kan het de algemene belasting door runtime-scans verminderen op bestanden die niet wijzigen na de meervoudige scan. Opnieuw scannen wanneer ze worden geopend, is namelijk dan niet nodig.

Verwante systeemwaarden

Er zijn twee nieuwe systeemwaarden die bij de nieuwe ondersteuning voor scans horen. Met behulp van deze twee systeemwaarden kunt u de scanomgeving voor uw server instellen. Hieronder vindt u de namen van de twee systeemwaarden en hun respectievelijke beschrijvingen. Deze systeemwaarden en hun instellingsmogelijkheden zijn beschreven voor iSeries Navigator. De vergelijkbare op tekens gebaseerde interfacewaarden worden tussen haakjes afgebeeld achter de iSeries Navigator-namen. Met de systeemwaarde QSCANFSCCTL krijgt u, als de besturingsoptie 'Scan alleen bij toegang via bestandservers' van iSeries Navigator is geselecteerd, in principe dezelfde resultaten als met de op tekens gebaseerde besturingsoptie *FSVRONLY.

De namen en beschrijvingen zijn:

1. Geregistreerd exitprogramma gebruiken om scan uit voeren op de bestandssystemen "hoofddirectory" (/) en QOpenSys en door de gebruiker gedefinieerde bestandssystemen (QSCANFS)
Met deze systeemwaarde kunt u opgeven of bestandssystemen moeten worden gescand. Alleen objecten in de bestandssystemen "hoofddirectory" (/) en QOpenSys en door de gebruiker gedefinieerde bestandssystemen worden gescand als het systeem volledig is geconverteerd. Met deze waarde geeft u op of objecten moeten worden gescand door exitprogramma's die zijn geregistreerd bij een van de exit points van het geïntegreerd bestandssysteem die betrekking hebben op scans.
De standaardwaarde is dat objecten worden gescand als er een exitprogramma is geregistreerd.
2. Besturing scan (QSCANFSCCTL)
U kunt voor deze systeemwaarde de standaardbesturingsopties gebruiken of u kunt eigen opties opgeven. Hieronder vindt u korte beschrijvingen van de verschillende besturingsopties op basis van de systeemwaarden van iSeries Navigator.
 - Scan alleen bij toegang via bestandservers - (*FSVRONLY opgegeven)
Een scan vindt alleen plaats als u toegang zoekt tot de iSeries vanaf een bestandsserver. Als deze optie niet is geselecteerd, worden alle pogingen tot toegang gescand.
 - Opdracht mislukt als exitprogramma mislukt - (*ERRFAIL opgegeven)
Als er een fout optreedt als het exitprogramma wordt opgeroepen, mislukt de opdracht of bewerking die de oproep naar het exitprogramma genereerde. Als deze optie niet is geselecteerd, slaat het systeem het mislukte exitprogramma over en wordt het object behandeld alsof het niet is gescand.
 - Schrijftoegang upgraden - (*NOWRTUPG is niet opgegeven)
De toegangsupgrade vindt plaats voor de scandescriptor die is doorgegeven naar het exitprogramma waarin schrijftoegang is opgenomen. Als de optie *NOWRTUPG niet is geselecteerd, probeert het systeem de schrijftoegang niet te upgraden.
Als *NOWRTUPG is opgegeven probeert het systeem de toegangsupgrade voor de scandescriptor die is doorgegeven naar het exitprogramma waarin schrijftoegang is opgenomen **niet** uit te voeren. Als de optie *NOWRTUPG is geselecteerd, probeert het systeem de schrijftoegang te upgraden.
 - Gebruik het kenmerk 'alleen als een object is gewijzigd' om het scannen te besturen - (*USEOCOATR niet opgegeven)
Het kenmerk 'alleen bij objectwijziging' (een object alleen scannen als het is gewijzigd) wordt gebruikt. Als deze optie niet is geselecteerd, wordt dit kenmerk niet gebruikt en wordt het object gescand nadat het is gewijzigd en wanneer de scansoftware een update meldt.
 - Afsluitopdracht mislukt als scan mislukt tijdens afsluiten - (*NOFAILCLO is niet opgegeven)

Als een scan op een object mislukt tijdens het afsluiten, mislukt de afsluitopdracht. Als deze optie niet is geselecteerd, mislukt de afsluitopdracht niet. Indien niet geselecteerd, vervangt deze waarde de specificatie van de waarde 'Opdracht mislukt als exitprogramma mislukt'.

Als *NOFAILCLO is opgegeven, laat het systeem de afsluitopdracht **niet** mislukken met een indicatie dat de scan is mislukt, ook al is de scan op een object mislukt als onderdeel van de afsluitbewerking.

- Zoeken bij volgende toegang als object is hersteld - (*NOPOSTRST is niet opgegeven)

Objecten worden gescand nadat ze zijn hersteld. Als het kenmerk 'het object wordt niet gescand' is opgegeven, wordt het object eenmaal gescand nadat het is hersteld. Als het kenmerk 'alleen bij objectwijziging' is opgegeven, wordt het object gescand nadat het is hersteld.

Als *NOPOSTRST is opgegeven terwijl de objecten worden hersteld, worden ze niet gescand alleen omdat ze zijn hersteld. Als het objectkenmerk 'het object wordt niet gescand' is, wordt het helemaal niet gescand. Als het objectkenmerk 'alleen bij objectwijziging' is, wordt het alleen gescand als het is gewijzigd nadat het is hersteld.

Raadpleeg voor meer informatie over deze systeemwaarden QSCANFS of QSCANFCTL.

Zoekvoorvallen

Scans kunnen om diverse redenen voorkomen. De onderstaande lijst geeft aan wanneer en waarom een scan kan plaatsvinden.

- "Objectwijziging"
- "Handtekeningwijziging"
- "Afwijkend CCSID" op pagina 22
- "Tijdens opslagbewerking" op pagina 23
- "Objectintegriteit controleren" op pagina 23

Om de actuele scanstatus en het actuele kenmerk van een object te zien, gebruikt u de opdracht WRKLNK (Werken met objectkoppelingen), de API Kenmerken ophalen (Qp0lGetAttr()) API of de pagina Eigenschappen van iSeries Navigator.

Objectwijziging: Een scan vindt doorgaans plaats als een object wordt geopend nadat het is gewijzigd. Gewoonlijk zijn het de objectgegevens die worden gewijzigd. Voorbeelden van wijzigingen aan een object zijn: direct of via een geheugentoe wijzigingsschema naar een object schrijven, een object afkappen en een object leegmaken. Als het CCSID-kenmerk van het object wijzigt, heeft dit bij de volgende toegangspoging ook een scan tot gevolg.

Handtekeningwijziging: Een scan vindt plaats bij het openen van een object als de algemene handtekening verschilt van de objecthandtekening. De algemene handtekeningen of handtekeningen van onafhankelijke-ASP-groepen vertegenwoordigen het softwareniveau van de scanexitprogramma's. Raadpleeg QIBM_QP0L_SCAN_OPEN voor meer informatie. De objecthandtekening geeft de algemene handtekening weer of de handtekening van de onafhankelijke ASP op het moment dat het object voor het laatst werd gescand. Wanneer een object zich niet in een onafhankelijke-ASP-groep bevindt, wordt de objecthandtekening vergeleken met de algemene scanhandtekening. Wanneer een object zich in een onafhankelijke-ASP-groep bevindt, wordt de objecthandtekening vergeleken met de scanhandtekening van de gerelateerde onafhankelijke-ASP-groep.

Opmerking: In het volgende voorbeeld worden de scansleutel en de scansleutelhandtekening gebruikt. Een scansleutel is een methode om een set scansoftware te identificeren. Een voorbeeld hiervan is voor een bepaald bedrijf. Dankzij de scansleutelhandtekening kan de set scansoftware het ondersteuningsniveau aangeven dat wordt geleverd. Een voorbeeld hiervan is een verzameling virusdefinities. Zie QIBM_QP0L_SCAN_OPEN voor meer informatie.

Hier is een voorbeeld van het plaatsvinden van een scan terwijl het object zich niet in een onafhankelijke-ASP-groep bevindt:

| 1. Een exitprogramma is geregistreerd bij het exit point QIBM_QP0L_SCAN_OPEN. Een scansleutel en een scansleutelhandtekening zijn als volgt opgegeven:

| Zoeksleutel: XXXXXX
| Zoeksleutelhandtekening: 0000000000

| De algemene scanhandtekening is 0000 en is niet bijgewerkt.

| 2. Vervolgens wordt een exitprogramma geregistreerd bij het exit point QIBM_QP0L_SCAN_CLOSE. Een scansleutel en een scansleutelhandtekening zijn als volgt opgegeven:

| Zoeksleutel: XXXXXX
| Zoeksleutelhandtekening: 1111111111

| De algemene scanhandtekening wordt bijgewerkt tot 0001.

| 3. Vervolgens wordt een bestand geopend met een actuele objecthandtekening 0000. De aanwezigheid van exitprogramma's plus het verschil tussen de algemene scanhandtekeningen (0000 resp. 0001) start een scan. Als de scan met succes is afgerond, wordt de bestandshandtekening bijgewerkt tot 0001.

| 4. Als het bestand door een andere gebruiker wordt geopend, wordt het niet opnieuw gescand om de object- en algemene handtekening overeenkomen.

| Hieronder volgt een voorbeeld waarbij het exitprogramma opnieuw een scan wil veroorzaken:

| 1. Aan het systeem is ondersteuning voor scannen naar nieuwe virussen toegevoegd. De API QP0LCHSG (scanhandtekening wijzigen) wordt geopend om de scansleutelhandtekening van de scansleutel bij te werken. Een scansleutel en een scansleutelhandtekening zijn als volgt opgegeven:

| Zoeksleutel: XXXXXX
| Zoeksleutelhandtekening: 2222222222

| De algemene scansleutelhandtekening wordt dan bijgewerkt tot 0002.

| 2. Als het eerder gescande bestand nu wordt geopend, veroorzaakt het verschil tussen de handtekeningen een nieuwe scan.

| Het voorbeeld wordt vervolgd en laat zien wat er gebeurt als een object zich in een onafhankelijke ASP bevindt:

| 1. Een onafhankelijke ASP wordt voor het eerst online gezet en er wordt een bestand in de onafhankelijke ASP geopend. Wanneer het eerste bestand wordt geopend, wordt de scansleutel lijst van de onafhankelijke ASP vergeleken met de scansleutel lijst van het systeem. De twee verschillen omdat er geen scansleutel lijst van de onafhankelijke ASP bestaat. In dat geval krijgt de scansleutel lijst van de onafhankelijke ASP de algemene scansleutel lijst. De scansleutel lijst van de onafhankelijke ASP heeft dan een scansleutel met de waarde XXXXXX en een scansleutelhandtekening met de waarde 2222222222. Hierdoor wordt de scanhandtekening van de onafhankelijke ASP gewijzigd in 0001. Wanneer het bestand in de onafhankelijke ASP dat op dit moment de objecthandtekening 0000 heeft, wordt geopend, wordt deze vergeleken met de scanhandtekening van de onafhankelijke ASP met de waarde 0001. Vanwege het verschil wordt het bestand gescand. Na een geslaagde scan wordt de bestands-handtekening bijgewerkt tot 0001.

| **Opmerking:** Een wijziging van een handtekening genereert een scan tenzij voor het object het kenmerk 'alleen bij objectwijziging' en de systeemwaarde *USEOCOATR zijn opgegeven.

| **Afwijkend CCSID:** Als een object wordt geopend met een ander CCSID dan waarin het eerder werd gescand, wordt een scan gegenereerd.

| Een voorbeeld hiervan is wanneer een gegevensbestand dat is opgeslagen in CCSID 819 wordt geopend in CCSID 1200 en gescand. Zolang de gegevens in het bestand niet worden gewijzigd, wordt geen scan gegenereerd als het bestand in CCSID 1200 wordt geopend. Wordt het bestand echter in een ander CCSID

| geopend, bijvoorbeeld 37, dan wordt een scan gegenereerd voor CCSID 37. Lukt die scan, dan wordt
| voor volgende toegangspogingen met CCSID 1200 en 37 geen nieuwe scan gegenereerd.

| Om de hoeveelheid gegevens op het systeem te beperken, worden slechts twee CCSID's en één binaire
| aanwijzing bewaard. Als u een object regelmatig met veel verschillende CCSID's opent, leidt dat tot een
| groot aantal extra scans.

| **Tijdens opslagbewerking:** Dit is nog een voorbeeld van wanneer een scan kan plaatsvinden. Er kan om
| een scan worden gevraagd wanneer een object wordt opgeslagen. De opdracht SAV bevat daartoe nu een
| SCAN-parameter waarmee u kunt opgeven of de bestanden moeten worden gescand als ze worden opge-
| slagen. U kunt ook opgeven dat een object niet wordt opgeslagen als eerder een scan is mislukt of als de
| scan mislukt tijdens het opslaan. Hiermee voorkomt u dat bestanden die niet konden worden gescand op
| een opslagmedium worden opgeslagen en mogelijk verplaatst naar andere systemen.

| **Opmerking:** Dit betekent niet dat een object wordt aangemerkt als gescand wanneer het wordt hersteld.
| Wanneer een object wordt hersteld, wordt de hele statushistorie van de scans gewist.

| **Objectintegriteit controleren:** Ten slotte kan opdracht worden gegeven tot een scan als voor de parame-
| ter SCANFS van de opdracht CHKOBJITG (Objectintegriteit controleren) de waarde *YES is opgegeven.
| Dit is een ideale optie als u wilt vaststellen of een bestand in orde is zonder het te openen. Als SCANFS
| (*Status) is opgegeven, wordt van alle objecten waarvoor een eerdere scan is mislukt een logboek
| gemaakt met de mislukte scans.

Werken met bestandssystemen

Een **bestandssysteem** biedt u toegang tot bepaalde opslagsegmenten die als logische eenheden zijn gerangschikt. Deze logische eenheden op uw server zijn bestanden, directory's, bibliotheken en objecten.

Elk bestandssysteem heeft een aantal logische structuren en regels voor het interactief werken met opslag. Deze structuren en regels kunnen per bestandssysteem verschillen. Vanuit het oogpunt van structuren en regels kunt u de ondersteuning van OS/400 voor het openen van databasebestanden en diverse andere objecten via bibliotheken beschouwen als een bestandssysteem. Ook de ondersteuning van OS/400 voor het openen van documenten (die eigenlijk stroombestanden zijn) via de mappenstructuur kunt u zien als een apart bestandssysteem.

Het geïntegreerd bestandssysteem beschouwt de ondersteuning van bibliotheken en de ondersteuning van mappen als aparte bestandssystemen. Andere soorten ondersteuning van bestandsbeheer met andere mogelijkheden worden eveneens als aparte bestandssystemen gezien.

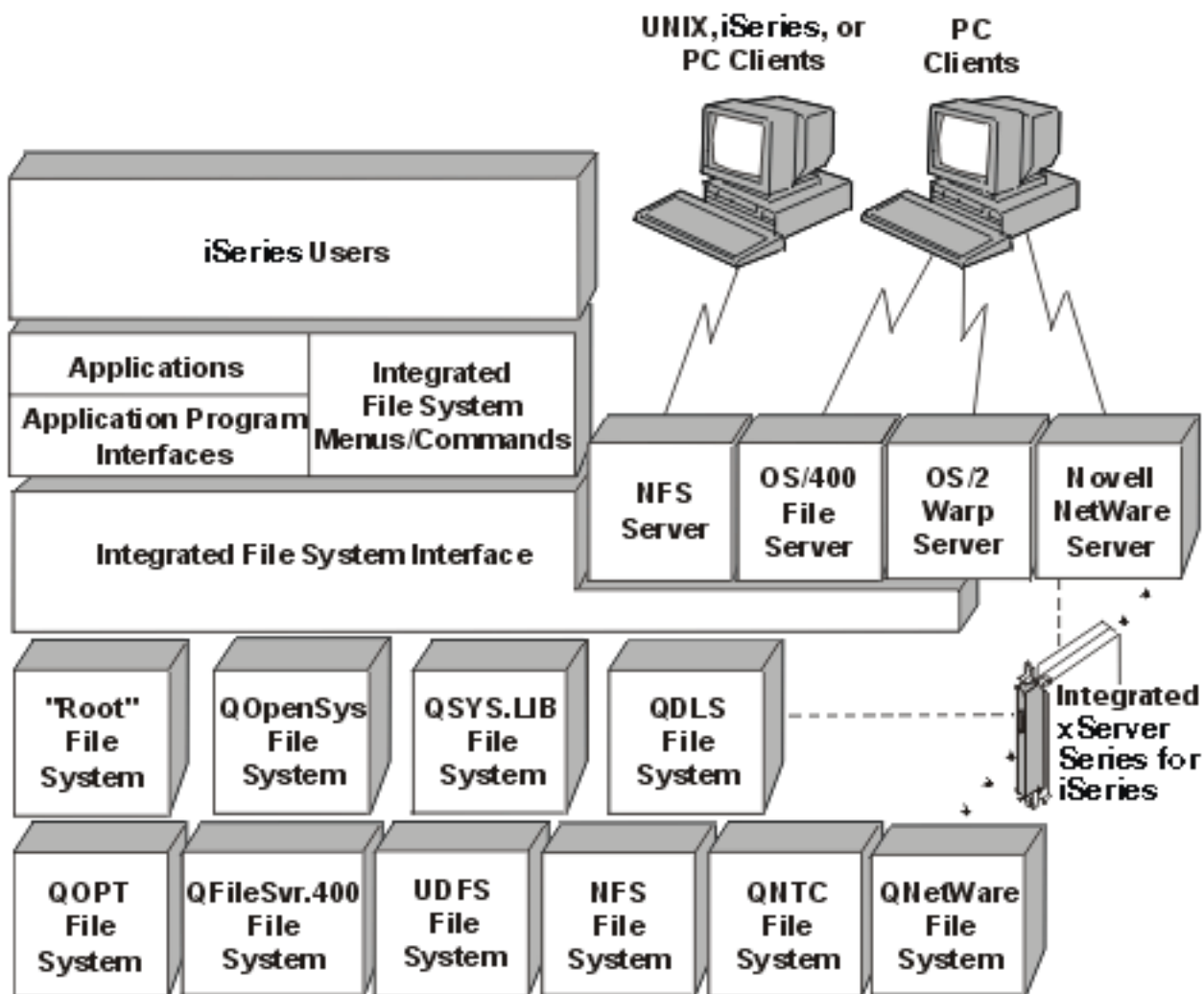
Een vergelijking van de functies en beperkingen van elk bestandssysteem vindt u in "Vergelijking bestandssystemen" op pagina 24.

De bestandssystemen in het geïntegreerd bestandssysteem zijn:

- De "hoofddirectory" (/)
- Bestandssysteem open systemen (QOpenSys)
- Door de gebruiker gedefinieerd bestandssysteem (UDFS)
- Bibliotheekbestandssysteem (QSYS.LIB)
- Onafhankelijke ASP QSYS.LIB
- Bestandssysteem documentenbibliotheek (QDLS)
- Bestandssysteem optische media (QOPT)
- Bestandssysteem NetWare (QNetWare)
- Bestandssysteem iSeries NetClient (QNTC)
- OS/400 Bestandssysteem bestandserver (QFileSvr.400)



- Netwerkbestandssysteem (NFS)

Via een gemeenschappelijke interface kunt u interactief werken met alle bestandssystemen. Deze interface is geoptimaliseerd voor de invoer en uitvoer van stroomgegevens (in tegenstelling tot in- en uitvoer van records via de gegevensbeheerinterfaces). In deze gemeenschappelijke interface kunt u interactief werken met de bestandssystemen door middel van de beschikbare opdrachten, menu's en schermen en API's.



Figuur 9. Bestandssystemen, bestandsservers en het IFS-interface

Raadpleeg voor meer informatie de volgende onderwerpen en publicaties:

- Ondersteuning voor optisch media 
- OS/400 Network File System Support 

Vergelijking bestandssystemen

In Tabel 2 op pagina 25 en Tabel 3 op pagina 26 worden de functies en beperkingen van elk bestandssysteem samengevat.

Tabel 2. Overzicht bestandssystemen (deel 1 van 2)

Mogelijkheden	"hoofd-directory" (/)	QOpenSys	QSYS.LIB ¹⁶	QDLS	QNTC
Standaardonderdeel van OS/400	Ja	Ja	Ja	Ja	Ja
Bestandstype	Stroom	Stroom	Record ¹²	Stroom	Stroom
Max. bestandsgrootte	T2 = 1 TB; T1=256 GB	T2 = 1 TB; T1=256 GB	Groottes database-bestand	4 GB	Wisselend ¹⁷
Geïntegreerd met OfficeVision (bestand kan bijvoorbeeld per mail worden verzonden)	Nee	Nee	Nee	Ja	Nee
Toegankelijk via OS/400-bestandserver	Ja	Ja	Ja	Ja	Ja
Directe toegang via I/O-processor bestandserver ¹	Nee	Nee	Nee	Nee	Ja
Relatieve snelheid voor openen/sluiten	Gemiddeld ²	Gemiddeld ²	Laag ²	Laag ²	Gemiddeld ²
Hoofdlettergevoelige zoekopdracht naar naam	Nee	Ja	Nee ⁴	Nee ⁵	Nee
Maximumlengte van elk onderdeel in pad	255 tek	255 tek	10,6 tek ⁶	8,3 tek ⁷	255 tek
Maximumlengte van pad ⁸	16 MB	16 MB	55-66 tek ⁴	82 tek	255 tek
Maximumlengte van de uitgebreide kenmerken van een object	2 GB	2 GB	Wisselend ⁹	32 kB	0 ¹⁸
Maximumniveaus van directoryhiërarchie binnen bestandssysteem	Geen maximum ¹⁰	Geen maximum ¹⁰	3	32	127
Maximumaantal koppelingen per object ¹¹	Wisselend ¹⁵	Wisselend ¹⁵	1	1	1
Ondersteunt symbolische koppelingen	Ja	Ja	Nee	Nee	Nee
Object/bestand kan eigenaar hebben	Ja	Ja	Ja	Ja	Nee
Ondersteunt geïntegreerd bestandssysteem-opdrachten	Ja	Ja	Ja	Ja	Ja
Ondersteunt geïntegreerd bestandssysteem-API's	Ja	Ja	Ja	Ja	Ja
Ondersteunt API's voor hiërarchische bestandssystemen	Nee	Nee	Nee	Ja	Nee
Threadsafe ¹³	Ja	Ja	Ja	Nee	Ja
Ondersteunt objecten opnemen in een journaal	Ja	Ja	Ja ¹⁴	Nee	Nee

Tabel 2. Overzicht bestandssystemen (deel 1 van 2) (vervolg)

Mogelijkheden	"hoofd- directory" (/)	QOpenSys	QSYS.LIB ¹⁶	QDLS	QNTC
Opmerkingen:					
1. De I/O-processor bestandserver is hardware die wordt gebruikt door de LAN-server.					
2. Indien toegang verkregen via de OS/400-bestandserver.					
3. Indien toegang verkregen via een LAN-server-client-PC. Toegang via iSeries-API's is betrekkelijk langzaam.					
4. De maximale padlengte van het QSYS.LIB-bestandssysteem bedraagt 55 tekens. Zie "Bibliotheekbestands- systeem (QSYS.LIB)" op pagina 37 voor meer informatie. De maximale padlengte van het bestandssysteem onafhankelijke ASP QSYS.LIB bedraagt 66 tekens. Zie "Onafhankelijke ASP QSYS.LIB" op pagina 40 voor meer informatie.					
5. Zie "Bestandssysteem voorzieningen documentenbibliotheek (QDLS)" op pagina 43 voor meer informatie.					
6. Maximaal 10 tekens voor de objectnaam en maximaal 6 tekens voor het type object. Zie "Bibliotheekbestands- systeem (QSYS.LIB)" op pagina 37 voor meer informatie.					
7. Maximaal 8 tekens voor de naam en 1-3 tekens voor de bestandsextensie (indien gebruikt). Zie "Bestands- systeem voorzieningen documentenbibliotheek (QDLS)" op pagina 43 voor meer informatie.					
8. Uitgaande van een absolute padnaam die begint met /, gevolgd door de bestandssysteemnaam (bijvoorbeeld /QDLS...).					
9. De bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB ondersteunen drie vooraf gedefinieerde uitge- breide kenmerken: .SUBJECT, .CODEPAGE, en .TYPE. De maximumlengte wordt bepaald door de gecombi- neerde lengte van deze drie uitgebreide kenmerken.					
10. In de praktijk zijn de directoryniveaus beperkt door de grenzen van de programma- en systeemruimte.					
11. Met uitzondering van een directory, die kan slechts één koppeling naar een andere directory hebben.					
12. De gebruikersruimten in de bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB ondersteunen in- en uitvoer van stroombestanden.					
13. API's van het IFS zijn threadsafe als de bewerking wordt uitgevoerd op een object dat zich in een threadsafe bestandssysteem bevindt. Als deze API's worden toegepast op objecten in bestandssystemen die niet threadsafe zijn terwijl bij een taak meerdere threads worden uitgevoerd, mislukt de API.					
14. De bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB ondersteunen journaals voor andere object- typen dan de bestandssystemen "hoofddirectory" (/), UDFS en QOpenSys. Raadpleeg het onderwerp Journaal- beheer in het iSeries Informatiecentrum voor meer informatie over journaals voor objecten in de bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB.					
15. *TYPE2-directory's kunnen maximaal 1 miljoen koppelingen per object hebben en maximaal 999.998 subdirectory's. *TYPE1-directory's kunnen maximaal 32.767 koppelingen per object hebben. Raadpleeg voor meer informatie *TYPE2-directory's.					
16. De gegevens in deze tabel hebben betrekking op de bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB.					
17. Is afhankelijk van het systeem dat wordt geopend					
18. QNTC ondersteunt geen uitgebreide kenmerken					
Afkortingen					
tek = tekens					
T1 = *TYPE1 *STMF					
T2 = *TYPE2 *STMF					
B = bytes kB = kilobytes MB = megabytes GB = gigabytes TB = terabytes					

Tabel 3. Overzicht bestandssystemen (deel 2 van 2)

Mogelijkheden	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Standaardonderdeel van OS/400	Ja	Ja	Ja	Ja	Nee
Bestandstype	Stroom	Stroom	Stroom	Stroom	Stroom

Tabel 3. Overzicht bestandssystemen (deel 2 van 2) (vervolg)

Mogelijkheden	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Max. bestandsgrootte	4 GB	4 GB	T2 = 1 TB; T1=256 GB	Wisselend ¹⁶	
Geïntegreerd met OfficeVision (bestand kan bijvoorbeeld per mail worden verzonden)	Nee	Nee	Nee	Nee	Nee
Toegankelijk via OS/400-bestandsserver	Ja	Ja	Ja	Ja	Ja
Directe toegang via de Integrated PC Server ¹	Nee	Nee	Nee	Nee	Ja
Relatieve snelheid voor openen/sluiten	Laag	Laag ²	Gemiddeld ²	Gemiddeld ²	Hoog ¹¹
Hoofdlettergevoelige zoekopdracht naar naam	Nee	Nee ²	Ja ¹²	Wisselend ²	Nee
Maximumlengte van elk onderdeel in pad	Wisselend ⁴	Wisselend ²	255 tek	Wisselend ²	255 tekens ¹³
Maximumlengte van padnaam	294 tekens	Geen maximum ²	16 MB	Geen maximum ²	255 tek
Maximumlengte van de uitgebreide kenmerken van een object	8 MB	0 ⁶	2 GB ¹⁰	0 ⁶	64 kB
Maximumniveaus van directoryhiërarchie binnen bestandssysteem	Geen maximum ⁷	Geen maximum ²	Geen maximum ⁷	Geen maximum ²	100
Maximumaantal koppelingen per object ⁷	1	1	Wisselend ¹⁵	Wisselend ²	1
Ondersteunt symbolische koppelingen	Nee	Nee	Ja	Ja ²	Nee
Object/bestand kan eigenaar hebben	Nee	Nee ⁹	Ja	Ja ²	Ja
Ondersteunt geïntegreerd bestandssysteem-opdrachten	Ja	Ja	Ja	Ja	Ja
Ondersteunt geïntegreerd bestandssysteem-API's	Ja	Ja	Ja	Ja	Ja
Ondersteunt API's voor hiërarchische bestandssystemen	Ja	Nee	Nee	Nee ²	Nee
Threadsafe ¹⁴	Ja	Ja	Ja	Ja	Nee
Ondersteunt objecten opnemen in een journaal	Nee	Nee	Ja	Nee	Nee

Tabel 3. Overzicht bestandssystemen (deel 2 van 2) (vervolg)

Mogelijkheden	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Opmerkingen:					
1. De I/O-processor bestandserver is hardware die wordt gebruikt door de LAN-server.					
2. Hangt af van het bestandssysteem op afstand waartoe Aan wordt gezocht.					
3. Indien toegang via de OS/400-bestandserver					
4. Zie "Bestandssysteem optische media (QOPT)" op pagina 45 voor meer informatie.					
5. Uitgaande van een absolute padnaam die begint met /, gevolgd door de bestandssysteemnaam.					
6. Het bestandssysteem QFileSvr.400 zendt geen uitgebreide kenmerken terug, ook al ondersteunt het bestandssysteem dat wordt geopend uitgebreide kenmerken.					
7. In de praktijk zijn de directoryniveaus beperkt door de grenzen van de programma- en systeemruimte.					
8. Met uitzondering van een directory, die kan slechts één koppeling naar een andere directory hebben.					
9. Het bestandssysteem dat wordt geopend, ondersteunt eventueel objecteigenaren.					
10. De maximumlengte van uitgebreide kenmerken van het UDFS kan niet meer dan 40 bytes zijn.					
11. Indien toegang via een Novell NetWare client-PC. Toegang via iSeries-API's is betrekkelijk langzaam.					
12. Hoofdlettergevoeligheid kan worden opgegeven wanneer een UDFS wordt gemaakt. Als bij het maken van een UDFS de parameter *MIXED wordt gebruikt, kunnen hoofdlettergevoelige scans worden uitgevoerd.					
13. NDS-objecten kunnen uit maximaal 255 tekens bestaan. Bestanden en directory's zijn beperkt tot de DOS 8.3-indeling.					
14. API's van het IFS zijn threadsafe als ze worden geopend in een multithread proces. Het bestandssysteem staat niet toe dat bestandssystemen die niet threadsafe zijn, worden geopend.					
15. *TYPE2-directory's kunnen maximaal 1 miljoen koppelingen per object hebben. *TYPE1-directory's kunnen maximaal 32.767 koppelingen per object hebben. Raadpleeg voor meer informatie *TYPE2-directory's.					
16. Is afhankelijk van het systeem dat wordt geopend					
Afkortingen					
tek = tekens					
T1 = *TYPE1 *STMF					
T2 = *TYPE2 *STMF					
B = bytes kB = kilobytes MB = megabytes GB = gigabytes TB = terabytes					

Bestandssysteem "hoofddirectory" (/)

Het bestandssysteem hoofddirectory (/) maakt optimaal gebruik van de stroombestandondersteuning en hiërarchische directorystructuur van het geïntegreerd bestandssysteem. Het bestandssysteem hoofddirectory (/) heeft de kenmerken van de DOS- en OS/2-bestandssystemen.

Daarnaast geldt:

- Het is geoptimaliseerd voor in- en uitvoer van stroombestanden.
- Het ondersteunt meervoudige vaste koppelingen en symbolische koppelingen.
- Het ondersteunt lokale sockets.
- Het ondersteunt threadsafe API's.
- Het ondersteunt *FIFO-objecten.
- Het ondersteunt niet alleen de *CHRSF-objecten /dev/null en /dev/zero, maar ook andere *CHRSF-objecten.
- Het ondersteunt het in een journaal opnemen van objectwijzigingen.
- Het ondersteunt het scannen van objecten met behulp van de exit points van het geïntegreerd bestandssysteem die betrekking hebben op scannen.

Het bestandssysteem hoofddirectory (/) ondersteunt de tekengeoriënteerde speciale bestanden (*CHRSF) /dev/null en /dev/zero. Tekengeoriënteerde speciale bestanden horen bij een apparaat of een resource van een computersysteem. Ze hebben paden die voorkomen in directory's en hebben dezelfde toegangsbescherming als gewone bestanden. De tekengeoriënteerde speciale bestanden /dev/null en /dev/zero zijn altijd leeg en gegevens die naar /dev/null en /dev/zero worden geschreven, worden gewist. De bestanden /dev/null en /dev/zero hebben *CHRSF als objecttype en kunnen worden gebruikt als gewone bestanden, met dat verschil dat er nooit gegevens in het bestand /dev/null worden ingelezen en dat het bestand /dev/zero altijd wordt teruggezonden met alle gegevens gewist (louter nullen).

Raadpleeg voor meer informatie over het bestandssysteem hoofddirectory (/) Het bestandssysteem "hoofddirectory" (/) gebruiken.

Het bestandssysteem "hoofddirectory" (/) gebruiken

Het bestandssysteem "hoofddirectory" (/) kan worden geopend via de IFS-interface, met behulp van de OS/400-bestandserver of van opdrachten, gebruikersschermen of API's van het geïntegreerd bestandssysteem.

- "Hoofdlettergevoeligheid in het bestandssysteem hoofddirectory (/)"
- "Padnamen in het bestandssysteem hoofddirectory (/)"
- "Koppelingen in het bestandssysteem hoofddirectory (/)"
- "Opdrachten van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem hoofddirectory (/)" op pagina 30
- "API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem hoofddirectory (/)" op pagina 30
- "Objectwijzigingen in de hoofddirectory (/) opnemen in een journaal" op pagina 30
- "UDP- en TCP-apparatuur in het bestandssysteem hoofddirectory (/)" op pagina 30

Hoofdlettergevoeligheid in het bestandssysteem hoofddirectory (/): Het bestandssysteem handhaaft de hoofd- en kleine letters die zijn opgegeven in de objectnamen, maar maakt geen onderscheid tussen hoofd- en kleine letters als de server naar namen zoekt.

Padnamen in het bestandssysteem hoofddirectory (/):

- Padnamen hebben de volgende vorm:

```
Directory/Directory . . . /Object
```

- Elk onderdeel van een padnaam kan bestaan uit maximaal 255 tekens. Dat is veel langer dan in de bestandssystemen QSYS.LIB en QDLS. De volledige padnaam kan extreem lang zijn, tot maximaal 16MB.
- Er geldt geen maximale diepte van de directoryhiërarchie, behalve de grenzen van het programma en de server.
- Wanneer de namen worden opgeslagen, worden de tekens erin geconverteerd naar UCS2 Level 1-indeling (voor *TYPE1 directory's) en UTF-16 (voor *TYPE2-directory's) (zie "Naamcontinuïteit" op pagina 17). Raadpleeg *TYPE2-directory's voor meer informatie over directory-indelingen.

Koppelingen in het bestandssysteem hoofddirectory (/): Meerdere vaste koppelingen naar hetzelfde object toegestaan in het bestandssysteem "hoofddirectory" (/). Symbolische koppelingen worden volledig ondersteund. Een symbolische koppeling kan worden gebruikt om vanuit het bestandssysteem van de "hoofddirectory" (/) een koppeling te maken een object in een ander bestandssysteem, zoals QSYS.LIB, onafhankelijke ASP QSYS.LIB of QDLS.

Raadpleeg voor een beschrijving van koppelingen "Koppeling" op pagina 11.

Opgaven van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem hoofddirectory (/): Alle opgaven die worden afgebeeld in "Toegang met behulp van CL-opgaven" op pagina 61 en alle schermen die worden beschreven in "Toegang met behulp van menu's en schermen" op pagina 60 werken in het bestandssysteem "hoofddirectory" (/). Het is echter soms niet veilig om deze opgaven te gebruiken in een multithread-proces.

| **API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem hoofddirectory (/):** Alle API's die worden afgebeeld in "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in het bestandssysteem hoofddirectory (/). Zie API's voor meer informatie over API's, zoals threadbeveiliging.

| **Objectwijzigingen in de hoofddirectory (/) opnemen in een journaal:** Objecten in het bestandssysteem hoofddirectory (/) kunnen worden opgenomen in een journaal. De belangrijkste functie van journaal-beheer is dat u aan de hand daarvan wijzigingen die aan een object zijn gemaakt sinds het voor het laatst werd opgeslagen, kunt herstellen. Raadpleeg voor meer informatie over het bijhouden van objectwijzigingen in het bestandssysteem hoofddirectory (/) "Objecten opnemen in een journaal" op pagina 88.

| **UDP- en TCP-apparatuur in het bestandssysteem hoofddirectory (/):** Het bestandssysteem hoofddirectory (/) in de directory /dev/xti bevat nu twee stuurprogramma's, udp en tcp. Beide stuurprogramma's zijn tekengeoriënteerde speciale bestanden (*CHRSE) die gemaakt worden tijdens de eerste opstartprocedure. De stuurprogramma's udp en tcp worden gebruikt om verbinding te maken met de transportproviders UDP en TCP. Beide stuurprogramma's zijn gebruikersapparatuur en krijgen een nieuw primair apparaatnummer. Ze hebben ook gekloonde openingen, hetgeen betekent dat iedere opening een unieke instance van het apparaat krijgt. Het gebruik van deze apparatuur wordt alleen ondersteund in PASE (Portable Application Solutions Environment). In Tabel 4 hieronder ziet u de objecten die worden gemaakt en hun eigenschappen.

| *Tabel 4. Stuurprogrammaobjecten en hun eigenschappen*

Padnaam	Type	Primair	Secundair	Eigenaar	Eigenaar gegevens- machtigingen	Groep	Groeps- gegevens- machtigingen	Vrij toe- gankelijke gegevens- machtigingen
/dev/xti	*DIR	N.v.t.	N.v.t.	QSYS	*RWX	Geen	*RX	*RX
/dev/xti/tcp	*CHRSE	Kloon	TCP	QSYS	*RW	Geen	*RW	*RW
/dev/xti/udp	*CHRSE	Kloon	UDP	QSYS	*RW	Geen	*RW	*RW

| Bestandssysteem open systemen (QOpenSys)

Het bestandssysteem QOpenSys is compatibel met op UNIX gebaseerde open systeemstandaards als POSIX en XPG. Net als het bestandssysteem hoofddirectory (/) maakt dit bestandssysteem gebruik van de ondersteuning van stroombestanden en directory's die het geïntegreerd bestandssysteem biedt.

Daarnaast geldt:

- Het wordt geopend via een hiërarchische directorystructuur die lijkt op UNIX-systemen.
- Het is geoptimaliseerd voor in- en uitvoer van stroombestanden.
- Het ondersteunt meervoudige vaste koppelingen en symbolische koppelingen.
- Het ondersteunt hoofdlettergevoelige namen.
- Het ondersteunt lokale sockets.
- Het ondersteunt threadsafe API's.
- Het ondersteunt *FIFO-objecten.
- Het ondersteunt het in een journaal opnemen van objectwijzigingen.

- Het ondersteunt het scannen van objecten met behulp van de exit points van het geïntegreerd bestandssysteem die betrekking hebben op scannen.

Het bestandssysteem QOpenSys heeft dezelfde kenmerken als het bestandssysteem “hoofddirectory” (/), alleen is het hoofdlettergevoelig om ondersteuning mogelijk te maken van op UNIX gebaseerde open systeemstandaards.

Zie QOpenSys gebruiken voor meer informatie over QOpenSys .

QOpenSys gebruiken

QOpenSys kan worden geopend via de IFS-interface met behulp van de OS/400-bestandserver of de opdrachten, gebruikersschermen en API's van het geïntegreerd bestandssysteem.

- “Hoofdlettergevoeligheid in het bestandssysteem QOpenSys”
- “Padnamen in het bestandssysteem QOpenSys”
- “Koppelingen in het bestandssysteem QOpenSys”
- “Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QOpenSys”
- “API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QOpenSys” op pagina 32
- “Objectwijzigingen in het bestandssysteem QOpenSys in een journaal opnemen” op pagina 32

Hoofdlettergevoeligheid in het bestandssysteem QOpenSys: In tegenstelling tot het bestandssysteem “hoofddirectory” (/) maakt het bestandssysteem QOpenSys onderscheid tussen hoofd- en kleine letters bij het zoeken naar objectnamen. Een reeks hoofdletters komt bijvoorbeeld niet overeen met dezelfde reeks letters waarvan een teken een kleine letter is.

Dankzij deze hoofdlettergevoeligheid kunt u werken met duplicaatnamen, mits de tekens waaruit de namen bestaan qua hoofd- en kleine letters niet helemaal identiek zijn. U kunt bijvoorbeeld in één directory van QOpenSys een object hebben met de naam Loonlijst, een object genaamd Loonlijst en een object genaamd L00NLIJST.

Padnamen in het bestandssysteem QOpenSys:

- Padnamen hebben de volgende vorm:
Directory/Directory/ . . . /Object
- Elk onderdeel van een padnaam kan bestaan uit maximaal 255 tekens. De volledige padnaam kan daardoor maximaal 16MB lang zijn.
- Er geldt geen maximale diepte van de directoryhiërarchie, behalve de grenzen van het programma en de server.
- Wanneer de namen worden opgeslagen, worden de tekens erin geconverteerd naar UCS2 Level 1-indeling (voor *TYPE1 directory's) en UTF-16 (voor *TYPE2-directory's) (zie “Naamcontinuïteit” op pagina 17). Raadpleeg *TYPE2-directory's voor meer informatie over directory-indelingen.

Koppelingen in het bestandssysteem QOpenSys: Meerdere vaste koppelingen naar hetzelfde object toegestaan in het bestandssysteem QOpenSys. Symbolische koppelingen worden volledig ondersteund. Een symbolische koppeling kan worden gebruikt om een koppeling te maken van het bestandssysteem QOpenSys naar een object in een ander bestandssysteem.

Raadpleeg “Koppeling” op pagina 11 voor een beschrijving van koppelingen.

Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QOpenSys: Alle opdrachten die worden afgebeeld in “Toegang met behulp van CL-opdrachten” op pagina 61 en alle schermen die worden beschreven in “Toegang met behulp van menu's en schermen” op pagina 60 werken in het bestandssysteem QOpenSys. Het is echter soms niet veilig om deze opdrachten te gebruiken in een multithread-proces.

| **API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QOpenSys:** Alle API's
| die worden afgebeeld in "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in het
| bestandssysteem QOpenSys. Zie API's voor meer informatie over API's, zoals threadbeveiliging.

Objectwijzigingen in het bestandssysteem QOpenSys in een journaal opnemen: Objecten in het
bestandssysteem QOpenSys kunnen in een journaal worden opgenomen. De belangrijkste functie van
journaalbeheer is dat u aan de hand daarvan wijzigingen die aan een object zijn gemaakt sinds het voor het
laatst werd opgeslagen, kunt herstellen. Raadpleeg voor meer informatie over het bijhouden van object-
wijzigingen in het bestandssysteem QOpenSys "Objecten opnemen in een journaal" op pagina 88.

Door de gebruiker gedefinieerd bestandssysteem (UDFS)

Een door de gebruiker gedefinieerd bestandssysteem (UDFS) bevindt zich in een door u te kiezen ASP of
onafhankelijke ASP. U maakt en beheert deze bestandssystemen.

Daarnaast hebben ze de volgende kenmerken:

- Ze bieden een hiërarchische directorystructuur die lijkt op PC-besturingssystemen zoals DOS en OS/2.
- Ze zijn geoptimaliseerd voor de in- en uitvoer van stroombestanden.
- Ze ondersteunen meervoudige vaste koppelingen en symbolische koppelingen.
- Ze ondersteunen lokale sockets.
- Ze ondersteunen threadsafe API's.
- Ze ondersteunen *FIFO-objecten.
- Ze ondersteunen het in een journaal vastleggen van wijzigingen aan objecten.
- | • Ze ondersteunen het scannen van objecten met behulp van de exit points van het geïntegreerd
| bestandssysteem die betrekking hebben op scannen.

| Voor meer informatie over het maken van UDFS-en raadpleegt u CRTUDFS. U kunt meerdere UDFS-en
| maken door elk een unieke naam te geven. U kunt tijdens het maken andere kenmerken opgeven voor
| een UDFS, zoals:

- Een ASP-nummer of onafhankelijke-ASP-naam, waar de objecten die zich in het UDFS bevinden, wor-
den opgeslagen.
- De hoofdlettergevoelige kenmerken van de objectnamen die zich in een UDFS bevinden.
De hoofdlettergevoeligheid van een UDFS bepaalt of kleine letters en hoofdletters gelijk worden behan-
deld bij het zoeken naar een objectnaam in een UDFS.
- | • Het scankenmerk voor gemaakte objecten dat bepaalt welk scankenmerk objecten krijgen die in een
| UDFS zijn gemaakt. Raadpleeg voor informatie over het instellen van dit kenmerk "Hiermee stelt u in
| of objecten moeten worden gescand" op pagina 118.
- | • Het kenmerk hernoemen en ontkoppelen beperken.
- | • De auditwaarde voor een UDFS.
- | • De twee stroombestandsindelingen, *TYPE1 en *TYPE2. Raadpleeg voor extra informatie over deze
| twee stroombestandsindelingen "Stroombestand" op pagina 15.

Raadpleeg de volgende onderwerpen voor meer informatie over door de gebruiker gedefinieerde
bestandssystemen (UDFS-en):

- UDFS-concepten
- UDFS gebruiken via de IFS-interface

UDFS-concepten


| Net als in de bestandssystemen "hoofddirectory" (/) en QOpenSys kunt u in een UDFS directory's,
| stroombestanden, symbolische koppelingen, lokale sockets en *FIFO-objecten maken.

Een UDFS wordt voorgesteld door een enkel blokgeoriënteerd speciaal bestandsobject. Als u een UDFS maakt, maakt u tegelijk een blokgeoriënteerd speciaal bestand. Het blokgeoriënteerde speciale bestand is alleen toegankelijk voor de gebruiker via de generieke opdrachten en API's van het geïntegreerd bestandssysteem en via de QFileSvr.400-interface.

Een UDFS bestaat slechts in twee werkstanden: **aangekoppeld** en **ontkoppeld**. Als u een UDFS aankoppelt, zijn de objecten erin toegankelijk. Als u een UDFS ontkoppelt, zijn de objecten erin niet toegankelijk.

Om toegang te hebben tot de objecten in een UDFS moet u het UDFS aankoppelen op een directory, bijvoorbeeld /home/JON. Als u een UDFS aankoppelt op een directory, wordt de oorspronkelijke inhoud van die directory ontoegankelijk, inclusief objecten en subdirectory's. Als u een UDFS aankoppelt, wordt de inhoud van het UDFS toegankelijk via het pad van de directory waarop u het UDFS aankoppelt. De directory /home/JON bijvoorbeeld bevat een bestand genaamd /home/JON/payroll. Een UDFS bevat drie directory's, mail, action en outgoing. Na het aankoppelen van het UDFS op de directory /home/JON is het bestand /home/JON/payroll ontoegankelijk en zijn de drie UDFS-directory's toegankelijk als /home/JON/mail, /home/JON/action en /home/JON/outgoing. Nadat u het UDFS hebt losgekoppeld, is het bestand /home/JON/payroll weer toegankelijk en zijn de drie directory's in het UDFS ontoegankelijk.

Opmerking: Op een UDFS op een onafhankelijke ASP kan niet worden aangekoppeld.

Om meer te weten te komen over het aankoppelen van bestandssystemen raadpleegt u OS/400 Ondersteuning netwerkbestandssysteem .

UDFS gebruiken via de IFS-interface

Een UDFS kan worden geopend via de interface van het geïntegreerd bestandssysteem met behulp van de opdrachten, gebruikersschermen en API's van de OS/400-bestandserver en van het geïntegreerd bestandssysteem. Bij gebruik van de interface van het geïntegreerd bestandssysteem moet u met de volgende aspecten en beperkingen rekening houden:

- "Hoofdlettergevoeligheid in een geïntegreerd bestandssysteem UDFS"
- "Padnamen in een geïntegreerd bestandssysteem UDFS" op pagina 34
- "Koppelingen in een geïntegreerd bestandssysteem UDFS" op pagina 34
- "Opdrachten van het geïntegreerd bestandssysteem gebruiken in een UDFS" op pagina 34
- "API's van het geïntegreerd bestandssysteem gebruiken in een UDFS" op pagina 35
- "Grafische gebruikersinterface voor een UDFS" op pagina 35
- "Een geïntegreerd bestandssysteem UDFS maken" op pagina 35
- "Een geïntegreerd bestandssysteem UDFS wissen" op pagina 35
- "Een geïntegreerd bestandssysteem UDFS afbeelden" op pagina 36
- "Een geïntegreerd bestandssysteem UDFS aankoppelen" op pagina 36
- "Een geïntegreerd bestandssysteem UDFS ontkoppelen" op pagina 36
- "Een geïntegreerd bestandssysteem UDFS opslaan en terugzetten" op pagina 36
- "Objectwijzigingen in een UDFS in een journaal opnemen" op pagina 36
- "UDFS en onafhankelijke ASP" op pagina 36

Hoofdlettergevoeligheid in een geïntegreerd bestandssysteem UDFS: U kunt opgeven of objectnamen in het UDFS hoofdlettergevoelig zijn of niet.

Als u kiest voor hoofdlettergevoeligheid, wordt er bij het zoeken naar objectnamen onderscheid gemaakt tussen hoofd- en kleine letters. Een naam met alleen hoofdletters bijvoorbeeld komt niet overeen met dezelfde naam waarin een of meer kleine letters voorkomen. /home/MURPH/ en /home/murph/ worden dus beschouwd als twee verschillende directory's. Om een hoofdlettergevoelige UDFS te maken, geeft u bij parameter CASE van de opdracht CRTUDFS de waarde *MIXED op.

Wanneer u niet-hoofdlettergevoelig kiest, maakt de server geen onderscheid tussen hoofd- en kleine letters bij het zoeken naar namen. Daarom beschouwt de server /home/CAYCE en /HOME/cayce als één directory, niet als twee. Om een niet-hoofdlettergevoelige UDFS te maken, geeft u bij parameter CASE van de opdracht CRTUDFS de waarde *MONO op.

In beide gevallen slaat het bestandssysteem de vorm op waarin de gebruiker de namen opgeeft. De optie hoofdlettergevoelig heeft alleen betrekking op de manier waarop de gebruiker naar namen zoekt op de server.

Padnamen in een geïntegreerd bestandssysteem UDFS: Een UDFS wordt voorgesteld door een blokgeoriënteerd speciaal bestandsobject wanneer het hele UDFS en alle objecten erin moeten worden bewerkt. Als uw UDFS zich op het systeem of een primaire gebruikers-ASP bevindt, moet de naam van een blokgeoriënteerd speciaal bestand de volgende vorm hebben:

```
/dev/QASPXX/udfs_naam.udfs
```

, waarbij XX het ASP-nummer is waar u het UDFS opslaat en udfs_naam de unieke naam van het UDFS in die ASP. Vergeet niet dat de UDFS-naam moet eindigen met de extensie .udfs.

Als uw UDFS zich op een onafhankelijke ASP bevindt, moeten blokgeoriënteerde speciale bestanden de volgende indeling hebben:

```
/dev/asp_naam/udfs_naam.udfs
```

waarbij asp_naam de naam is van de onafhankelijke ASP waar u het UDFS opslaat en udfs_naam de unieke naam van het UDFS in die onafhankelijke ASP. Vergeet niet dat de UDFS-naam moet eindigen met de extensie .udfs.

Padnamen van objecten in een UDFS zijn gerelateerd aan de directory waaraan het wordt gekoppeld. Als u bijvoorbeeld het UDFS /dev/qasp01/wysocki.udfs aan /home/dennis koppelt, beginnen de paden van alle objecten in het UDFS met /home/dennis.

Extra regels voor padnamen:

- Elk onderdeel van een padnaam kan bestaan uit maximaal 255 tekens. De volledige padnaam kan daardoor maximaal 16MB lang zijn.
- Er geldt geen maximale diepte van de directoryhiërarchie, behalve de grenzen van het programma en de server.
- Wanneer de namen worden opgeslagen, worden de tekens erin geconverteerd naar UCS2 Level 1-indeling (voor *TYPE1 directory's) en UTF-16 (voor *TYPE2-directory's) (zie "Naamcontinuïteit" op pagina 17). Raadpleeg *TYPE2-directory's voor meer informatie over directory-indelingen.

| **Koppelingen in een geïntegreerd bestandssysteem UDFS:** In een UDFS zijn meerdere vaste koppelingen naar hetzelfde object mogelijk en worden symbolische koppelingen volledig ondersteund. Met een symbolische koppeling kan een koppeling worden gemaakt van een UDFS naar een object in een ander bestandssysteem.

Raadpleeg "Koppeling" op pagina 11 voor een beschrijving van koppelingen.

Opdrachten van het geïntegreerd bestandssysteem gebruiken in een UDFS: Alle opdrachten die worden afgebeeld in "Toegang met behulp van CL-opdrachten" op pagina 61 en alle schermen die worden beschreven in "Toegang met behulp van menu's en schermen" op pagina 60 werken in een UDFS. Er bestaan enkele CL-opdrachten die specifiek zijn voor het UDFS en andere aangekoppelde bestandssystemen in het algemeen. In de onderstaande tabel worden ze beschreven.

Tabel 5. CL-opdrachten voor het UDFS

Opdracht	Beschrijving
ADDMFS	Aangekoppeld bestandssysteem toevoegen. Hiermee plaatst u geëxporteerde serverbestandssystemen op afstand boven lokale clientdirectory's.
CRTUDFS	UDFS maken. Hiermee maakt u een UDFS.
DLTUDFS	UDFS wissen. Hiermee wist u een UDFS.
DSPMFSINF	Informatie over aangekoppeld bestandssysteem afbeelden. Hiermee beeldt u informatie af over een aangekoppeld bestandssysteem.
DSPUDFS	UDFS afbeelden. Hiermee beeldt u informatie af over een UDFS.
MOUNT	Een bestandssysteem aankoppelen. Hiermee plaatst u geëxporteerde serverbestandssystemen op afstand boven lokale clientdirectory's. Deze opdracht is een alias van de opdracht ADDMFS.
RMVMFS	Aangekoppeld bestandssysteem wissen. Hiermee verwijdert u geëxporteerde serverbestandssystemen op afstand van de naamruimte van de lokale client.
UNMOUNT	Een bestandssysteem afkoppelen. Hiermee verwijdert u geëxporteerde serverbestandssystemen op afstand van de naamruimte van de lokale client. Deze opdracht is een alias van de opdracht RMVMFS.

Opmerking: Voordat de opdrachten van het geïntegreerd bestandssysteem functioneren in een UDFS moet u dit UDFS aankoppelen.

| **API's van het geïntegreerd bestandssysteem gebruiken in een UDFS:** Alle API's die worden afgebeeld in "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in een UDFS.

Opmerking: Voordat de API's geïntegreerd bestandssysteem functioneren in een UDFS moet u dit UDFS aankoppelen.

Grafische gebruikersinterface voor een UDFS: iSeries Navigator is een grafische gebruikersinterface op uw PC die makkelijk en handig toegang biedt tot UDFS-en. Met deze interface kunt u van een Windows-client een UDFS maken, wissen, afbeelden, aankoppelen en ontkoppelen.

U kunt handelingen uitvoeren op een UDFS via iSeries Navigator. Basistaken zijn:

- "Een geïntegreerd bestandssysteem UDFS maken".
- "Een geïntegreerd bestandssysteem UDFS aankoppelen" op pagina 36.
- "Een geïntegreerd bestandssysteem UDFS ontkoppelen" op pagina 36.

Een geïntegreerd bestandssysteem UDFS maken: Met de opdracht CRTUDFS (Door de gebruiker gedefinieerd bestandssysteem maken) maakt u een bestandssysteem dat zichtbaar kan worden gemaakt met de naamruimte, API's en CL-opdrachten van het geïntegreerd bestandssysteem. Met de opdracht ADDMFS of MOUNT koppelt u het UDFS aan "bovenop" de reeds bestaande lokale directory. U kunt een UDFS maken in een ASP of onafhankelijke ASP naar keuze.

U kunt voor een UDFS tevens de volgende elementen opgeven:

- Hoofdlettergevoeligheid
- | • Of objecten die zijn gemaakt in het UDFS moeten worden gescand
- | • De auditwaarde voor objecten die in het UDFS zijn gemaakt
- | • De waarde van de kenmerken beperkt, hernoemen en ontkoppelen

Een geïntegreerd bestandssysteem UDFS wissen: Met de opdracht DLTUDFS (Door de gebruiker gedefinieerd bestandssysteem wissen) kunt u een bestaande, ontkoppelde UDFS wissen, inclusief alle objecten

die zich daarin bevinden. De opdracht mislukt als het UDFS nog aangekoppeld is. Als u een UDFS wist, worden alle objecten in dat UDFS ook gewist. Als u niet de juiste machtiging hebt om alle objecten in een UDFS te wissen, wordt geen van de objecten gewist.

Een geïntegreerd bestandssysteem UDFS afbeelden: Met de opdracht DSPUDFS (Door de gebruiker gedefinieerd bestandssysteem afbeelden) beeldt u de kenmerken van een bestaand bestandssysteem af, of dit nu aangekoppeld is of niet. Met de opdracht DSPMFSINF (Aangekoppeld bestandssysteem afbeelden) beeldt u ook informatie over een aangekoppelde UDFS af, maar ook over andere aangekoppelde bestandssystemen.

Een geïntegreerd bestandssysteem UDFS aankoppelen: Met de opdrachten ADDMFS (Aangekoppeld bestandssysteem toevoegen) en MOUNT maakt u objecten in een bestandssysteem toegankelijk voor de naamruimte van het geïntegreerd bestandssysteem. U koppelt een UDFS aan door bij parameter TYPE van de opdracht ADDMFS *UDFS op te geven.

Opmerking: Op een UDFS op een onafhankelijke ASP kan niet worden aangekoppeld.

Een geïntegreerd bestandssysteem UDFS ontkoppelen: Met de opdracht ontkoppelen maakt u de inhoud van een UDFS ontoegankelijk voor de interfaces van het geïntegreerd bestandssysteem. De objecten in een UDFS zijn niet meer afzonderlijk toegankelijk zodra het UDFS is ontkoppeld. Met de opdrachten RMVMFS (Aangekoppeld bestandssysteem verwijderen) en UNMOUNT maakt u objecten in een bestandssysteem ontoegankelijk voor de naamruimte van het geïntegreerd bestandssysteem. Als een object in het bestandssysteem in gebruik is als u deze opdracht geeft (er is bijvoorbeeld een bestand geopend) ontvangt u een foutmelding. Het UDFS blijft aangekoppeld. Als u aangekoppeld hebt over een deel van het UDFS, dan kan dit UDFS pas worden ontkoppeld als het weer is vrijgemaakt.

U koppelt het UDFS /dev/qasp02/jenn.udfs aan op /home/judy in de naamruimte van het geïntegreerd bestandssysteem. Als u vervolgens een ander bestandssysteem, bijvoorbeeld /pubs op /home/judy aankoppelt, wordt de inhoud van jenn.udfs ontoegankelijk. Bovendien kunt u jenn.udfs niet ontkoppelen totdat u het tweede bestandssysteem hebt ontkoppeld van /home/judy.

Opmerking: Op een UDFS op een onafhankelijke ASP kan niet worden aangekoppeld.

Een geïntegreerd bestandssysteem UDFS opslaan en terugzetten: U kunt alle UDFS-objecten en hun bijbehorende machtigingen opslaan en terugzetten. Met de opdracht SAV (Opslaan) kunt u objecten opslaan in een UDFS en met de opdracht RST (Terugzetten) kunt UDFS-objecten weer terugzetten. Beide opdracht werken ongeacht of het UDFS is aangekoppeld of ontkoppeld. Om de UDFS-kenmerken echter goed te kunnen opslaan, en niet alleen de objecten in een UDFS, moet het UDFS ontkoppeld zijn.

Objectwijzigingen in een UDFS in een journaal opnemen: Objecten in door de gebruiker gedefinieerde bestandssystemen kunnen worden opgenomen in een journaal. De belangrijkste functie van journaalbeheer is dat u aan de hand daarvan wijzigingen die aan een object zijn gemaakt sinds het voor het laatst werd opgeslagen, kunt herstellen. Raadpleeg voor meer informatie over het bijhouden van objectwijzigingen in een UDFS "Objecten opnemen in een journaal" op pagina 88.

UDFS en onafhankelijke ASP: Als u een onafhankelijke ASP online zet, vinden er in het bestandssysteem hoofddirectory (/) verschillende wijzigingen plaats:

- Voor de onafhankelijke ASP wordt in de directory /dev directory een directory gemaakt. De naam van de directory komt overeen met de naam van de apparatuurbeschrijving die bij de ASP hoort. Als deze directory al bestaat voordat de online-opdracht wordt gegeven en niet leeg is, wordt het online zetten wel uitgevoerd maar kunt u op de ASP niet met UDFS-en werken. Als dit gebeurt, zet dan de onafhankelijke ASP offline, hernoem de directory of verwijder de inhoud ervan en geen de opdracht online zetten opnieuw.
- In de directory /dev/asp_naam vindt u de blokgeoriënteerde speciale bestanden die horen bij de UDFS-en die zich op de onafhankelijke ASP bevinden. Er bestaat altijd een standaard systeem-UDFS. Het pad naar het blokgeoriënteerde speciale bestand is: /dev/asp_naam/QDEFAULT.UDFS

- | • Het standaard-UDFS wordt aangekoppeld op de directory /asp_naam. De directory /asp_naam hoeft niet te bestaan voordat de online-opdracht wordt gegeven. Als hij echter wel bestaat, dan moet hij leeg zijn. Als de directory namelijk niet leeg is, wordt de ASP wel online gezet, maar wordt het UDFS niet aangekoppeld. Als dit gebeurt, hernoem dan de directory of verwijder de inhoud ervan. Geef vervolgens de opdracht offline zette en daarna online zetten, of koppel het standaard-UDFS aan met de opdracht MOUNT.
- | • Als de onafhankelijke ASP een primaire of secundaire ASP is en het standaard-UDFS is met succes aangekoppeld, wordt een extra bestandssysteem aangekoppeld. Het bestandssysteem onafhankelijke ASP QSYS.LIB wordt aangekoppeld op /asp_naam/QSYS.LIB. Zie "Onafhankelijke ASP QSYS.LIB" op pagina 40 voor meer informatie.

| **Opmerking:** Dit bestandssysteem kan niet onafhankelijk van het standaard-UDFS worden aan- of ontkoppeld. Het wordt altijd automatisch aan- of ontkoppeld.

| **Bibliotheekbestandssysteem (QSYS.LIB)**

Het bestandssysteem QSYS.LIB ondersteunt de iSeries-server-bibliotheekstructuur. Dit bestandssysteem biedt u toegang tot databasebestanden alle andere iSeries-server-objecttypen die bibliotheekondersteuning beheert het systeem en de basisgebruikers-ASP's.

Daarnaast geldt:

- Ondersteuning van alle gebruikersinterfaces en programmeerinterfaces die werken op iSeries-server-bibliotheken en de objecten in die bibliotheken
- Ondersteuning van alle programmeertalen en voorzieningen die werken met databasebestanden
- Uitgebreide beheerondersteuning voor iSeries-serverobjecten
- Ondersteuning van stroom-invoer/uitvoerbewerkingen op secties van fysieke bestanden, gebruikersruimten en opslagbestanden

Voor versie 3 of van OS/400 zou het bestandssysteem QSYS.LIB waarschijnlijk *het* bestandssysteem van de iSeries-server zijn genoemd. Programmeurs die talen gebruikten als RPG en COBOL en toepassingen als DDS om toepassingen te ontwikkelen, maakten gebruik van het bestandssysteem QSYS.LIB. Systeemoperators die gebruik maakten van opdrachten, menu's en schermen om uitvoerwachtrijen te manipuleren gebruikten het bestandssysteem QSYS.LIB, evenals systeembeheerders die gebruikersprofielen maakten en wijzigden.

Al deze voorzieningen en toepassingen gebaseerd op deze voorzieningen functioneren nog net zo als voor de introductie van het geïntegreerd bestandssysteem. Deze voorzieningen kunnen QSYS.LIB echter niet openen via de IFS-interface.

Raadpleeg voor meer informatie over QSYS.LIB QSYS.LIB gebruiken via de geïntegreerd bestandssysteem-interface.

QSYS.LIB gebruiken via de geïntegreerd bestandssysteem-interface

Het bestandssysteem QSYS.LIB kan worden geopend via de IFS-interface met behulp van de OS/400-bestandserver of de opdrachten, gebruikersschermen en API's van het geïntegreerd bestandssysteem. Bij gebruik van de interface van het geïntegreerd bestandssysteem moet u met de volgende aspecten en beperkingen rekening houden.

- "Lijst van machtigingen QPWFSESERVER in het bestandssysteem QSYS.LIB" op pagina 38
- "Beperkingen t.a.v. bestandsbeheer in het bestandssysteem QSYS.LIB" op pagina 38
- "Ondersteuning voor gebruikersruimten in het bestandssysteem QSYS.LIB" op pagina 38
- "Ondersteuning voor opslagbestanden in het bestandssysteem QSYS.LIB" op pagina 38
- "Hoofdlettergevoeligheid in het bestandssysteem QSYS.LIB" op pagina 38
- "Padnamen in het bestandssysteem QSYS.LIB" op pagina 39
- "Koppelingen in het bestandssysteem QSYS.LIB" op pagina 39

- “Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QSYS.LIB” op pagina 39
- “API’s van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QSYS.LIB” op pagina 39

Lijst van machtigingen QPWFSEVER in het bestandssysteem QSYS.LIB: De QPWFSEVER is een lijst van machtigingen (objecttype *AUTL) die extra toegangsvereisten geeft voor alle objecten in het bestandssysteem QSYS.LIB die worden geopend vanaf een client op afstand. De machtigingen die in deze lijst van machtigingen zijn opgegeven, zijn van toepassing op alle objecten in het bestandssysteem QSYS.LIB.

De standaardmachtiging voor dit object is de machtiging PUBLIC *USE. De beheerder kan met de opdrachten EDTAUTL (Lijst van machtigingen bewerken) of WRKAUTL (Werken met lijst van machtigingen) de waarde van de machtiging wijzigen. De beheerder kan de machtiging PUBLIC *EXCLUDE toewijzen aan de lijst van machtigingen zodat het grote publiek geen QSYS.LIB-objecten kan openen vanaf clients op afstand.

Beperkingen t.a.v. bestandsbeheer in het bestandssysteem QSYS.LIB:

- Logische bestanden worden niet ondersteund.
- Fysieke bestanden die worden ondersteund voor toegang in de werkstand Tekst zijn door het programma beschreven fysieke bestanden met een enkel veld en fysieke bronbestanden met een enkel tekstveld. Tot de fysieke bestanden die worden ondersteund voor toegang in de binaire werkstand behoren, naast de bestanden die worden ondersteund voor toegang in de werkstand Tekst, extern beschreven fysieke bestanden.
- Byte-range-vergrendeling wordt niet ondersteund. (Raadpleeg voor informatie over byte-range-vergrendeling het onderwerp fcntl() in het iSeries Informatiecentrum.)
- Als een taak een databasebestandssectie geopend heeft, krijgt telkens slechts één taak schrijftoegang voor die bestandssectie. Andere aanvragen krijgen alleen leestoegang.

Ondersteuning voor gebruikersruimten in het bestandssysteem QSYS.LIB: QSYS.LIB ondersteunt stroominvoer- en -uitvoerbewerkingen naar objecten in de gebruikersruimte. Een programma kan bijvoorbeeld stroomgegevens naar een gebruikersruimte schrijven en gegevens van een gebruikersruimte lezen. De maximumgrootte van een gebruikersruimte is 16 776 704 bytes.

Vergeet niet dat gebruikersruimten niet zijn gelabeld met een CCSID. Daarom is het teruggezonden CCSID het standaard-CCSID van de taak.

Ondersteuning voor opslagbestanden in het bestandssysteem QSYS.LIB: Het bestandssysteem QSYS.LIB ondersteunt stroominvoer/uitvoerbewerkingen voor het opslaan van bestandsobjecten. Een bestaand opslagbestand heeft bijvoorbeeld gegevens die door een ander bestand gelezen of gekopieerd worden totdat het nodig is de gegevens in een ander bestaand en leeg opslagbestandsobject te plaatsen. Als een opslagbestand geopend is voor schrijfbewerkingen zijn geen andere openinstances van het bestand toegestaan. Een opslagbestand staat **wel** meerdere openinstances voor leesbewerkingen toe, mits de taak meer dan één openinstance van het bestand heeft voor leesbewerkingen. Een opslagbestand kan niet worden geopend voor lee/schrijftoegang. Stroominvoer- en/of uitvoerbewerkingen om bestandsgegevens op te slaan zijn niet toegestaan als er in een taak meerdere threads worden uitgevoerd.

Stroominvoer- en/of uitvoerbewerkingen naar een opslagbestand worden niet ondersteund als een opslagbestand of zijn directory wordt geëxporteerd via de netwerkbestandssysteemserver. Ze mogen echter worden geopend vanaf een PC-client en via het bestandssysteem QFileSvr.400.

Hoofdlettergevoeligheid in het bestandssysteem QSYS.LIB: Doorgaans maakt het bestandssysteem QSYS.LIB geen onderscheid tussen hoofd- en kleine letters in objectnamen. Een zoekbewerking naar objectnamen levert hetzelfde resultaat op, of de naam nu bestaat uit hoofd- of kleine letters.

Als een naam echter tussen aanhalingstekens staat, wordt de kast van elk teken bewaard. Als u dus zoekt met een tekenreeks tussen aanhalingstekens, geeft u een hoofdlettergevoelige zoekopdracht.

Padnamen in het bestandssysteem QSYS.LIB:

- Elke component van de padnaam moet de objectnaam bevatten gevolgd door het objecttype van het object. Bijvoorbeeld:

/QSYS.LIB/QGPL.LIB/PRT1.OUTQ

/QSYS.LIB/EMP.LIB/PAY.FILE/TAX.MBR

Objectnaam en -type zijn van elkaar gescheiden door een punt (.). Objecten in een bibliotheek kunnen dezelfde naam hebben als het om verschillende objecttypen gaat. Het objecttype moet dus worden opgegeven om het object te identificeren.

- De objectnaam in elke component mag maximaal 10 tekens lang zijn, het objecttype maximaal 6 tekens.
- De directoryhiërarchie in QSYS.LIB kan al naar gelang het object twee tot drie niveaus hebben (twee of drie componenten in de padnaam). Als het object een databasebestand is, kan de hiërarchie drie niveaus hebben (bibliotheek, bestand, sectie). Anders zijn slechts twee niveaus mogelijk (bibliotheek, object). De combinatie van de lengte van elke componentnaam en het aantal directoryniveaus bepaalt de maximumlengte van de padnaam.

Als de eerste twee niveaus hoofddirectory (/) en QSYS.LIB zijn, kan de directoryhiërarchie van QSYS.LIB vijf niveaus hebben.

- Wanneer de namen worden opgeslagen, worden de tekens geconverteerd naar CCSID 37. Namen met aanhalingstekens echter worden opgeslagen met het CCSID van de taak.

Raadpleeg voor meer informatie over CCSID het onderwerp Lokaliseren in het iSeries Informatiecentrum.

Koppelingen in het bestandssysteem QSYS.LIB: Symbolische koppelingen kunnen in het bestandssysteem QSYS.LIB niet worden gemaakt of opgeslagen.

De relatie tussen een bibliotheek en objecten in een bibliotheek is gelijk aan één vaste koppeling tussen de bibliotheek en elk object in de bibliotheek. Het geïntegreerd bestandssysteem behandelt de relatie bibliotheek/object als een koppeling. Daarom is het mogelijk een koppeling te maken tussen een bestandssysteem dat symbolische koppelingen ondersteunt en een object in het bestandssysteem QSYS.LIB.

Raadpleeg "Koppeling" op pagina 11 voor een beschrijving van koppelingen.

Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem

QSYS.LIB: De opdrachten die worden afgebeeld in het onderwerp "Toegang met behulp van CL-opdrachten" op pagina 61 werken in het bestandssysteem QSYS.LIB, met uitzondering van de volgende:

- De opdracht ADDLNK kan alleen worden gebruikt om een symbolische koppeling te maken *naar* een object in QSYS.LIB.
- Bestandsbewerkingen kunnen alleen worden uitgevoerd op door het programma beschreven fysieke bestanden en fysieke bronbestanden.
- De opdrachten STRJRN en ENDJRN kunnen niet worden gebruikt op fysieke databasebestanden.

Dezelfde beperkingen zijn van toepassing op de gebruikersschermen beschreven in "Toegang met behulp van menu's en schermen" op pagina 60.

API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QSYS.LIB: De API's die worden afgebeeld in het onderwerp "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in het bestandssysteem QSYS.LIB, met uitzondering van de volgende:

- Bestandsbewerkingen kunnen alleen worden uitgevoerd op door het programma beschreven fysieke bestanden en fysieke bronbestanden.

- De functie symlink() kan alleen worden gebruikt om naar een object te koppelen in QSYS.LIB vanaf een ander bestand dat symbolische koppelingen ondersteunt.
- De API's QjoStartJournal() en QjoEndJournal() kunnen niet worden gebruikt op fysieke database-bestanden.

Onafhankelijke ASP QSYS.LIB

Het bestandssysteem onafhankelijke ASP QSYS.LIB ondersteunt de iSeries-server-bibliotheekstructuur in ASP's die u maakt en definieert. Dit bestandssysteem biedt u toegang tot databasebestanden alle andere iSeries-server-objecttypen die bibliotheekondersteuning beheert in de onafhankelijke ASP's.

Daarnaast geldt:

- Ondersteuning van alle gebruikersinterfaces en programmeerinterfaces die werken op iSeries-server-bibliotheeken en de objecten in die bibliotheeken in onafhankelijke ASP's
- Ondersteuning van alle programmeertalen en voorzieningen die werken met databasebestanden
- Uitgebreide beheerondersteuning voor iSeries-server-objecten
- Ondersteuning van stroom-invoer/uitvoerbewerkingen op secties van fysieke bestanden, gebruikersruimten en opslagbestanden

Raadpleeg voor meer informatie over het bestandssysteem onafhankelijke ASP QSYS.LIB het onderwerp Onafhankelijke ASP's gebruiken QSYS.LIB via de geïntegreerd bestandssysteem-interface.

Onafhankelijke ASP QSYS.LIB gebruiken via de geïntegreerd bestandssysteem-interface

Het bestandssysteem onafhankelijke ASP QSYS.LIB kan worden geopend via de IFS-interface met behulp van de OS/400-bestandserver of de opdrachten, gebruikersschermen en API's van het geïntegreerd bestandssysteem. Bij gebruik van de interface van het geïntegreerd bestandssysteem moet u met de volgende aspecten en beperkingen rekening houden.

- "Lijst van machtigingen QPWFSERVER in het bestandssysteem onafhankelijke ASP QSYS.LIB"
- "Beperkingen t.a.v. bestandsbeheer in het bestandssysteem onafhankelijke ASP QSYS.LIB" op pagina 41
- "Ondersteuning voor gebruikersruimten in het bestandssysteem onafhankelijke ASP QSYS.LIB" op pagina 41
- "Ondersteuning voor opslagbestanden in het bestandssysteem onafhankelijke ASP QSYS.LIB" op pagina 41
- "Hoofdlettergevoeligheid in het bestandssysteem onafhankelijke ASP QSYS.LIB" op pagina 41
- "Padnamen in het bestandssysteem onafhankelijke ASP QSYS.LIB" op pagina 41
- "Koppelingen in het bestandssysteem onafhankelijke ASP QSYS.LIB" op pagina 42
- "Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem onafhankelijke ASP QSYS.LIB" op pagina 42
- "API's van de onafhankelijke ASP geïntegreerd bestandssysteem gebruiken in het bestandssysteem QSYS.LIB" op pagina 42

Lijst van machtigingen QPWFSERVER in het bestandssysteem onafhankelijke ASP QSYS.LIB: De QPWFSERVER is een lijst van machtigingen (objecttype *AUTL) die extra toegangsvereisten geeft voor alle objecten in het bestandssysteem onafhankelijke ASP QSYS.LIB die worden geopend vanaf een client op afstand. De machtigingen die in deze lijst van machtigingen zijn opgegeven, zijn van toepassing op alle objecten in het bestandssysteem onafhankelijke ASP QSYS.LIB.

De standaardmachtiging voor dit object is de machtiging PUBLIC *USE. De beheerder kan met de opdrachten EDTAUTL (Lijst van machtigingen bewerken) of WRKAUTL (Werken met lijst van machtigingen) de waarde van de machtiging wijzigen. De beheerder kan de machtiging PUBLIC *EXCLUDE toe-wijzen aan de lijst van machtigingen zodat het grote publiek geen onafhankelijke ASP QSYS.LIB-objecten kan openen vanaf clients op afstand.

Beperkingen t.a.v. bestandsbeheer in het bestandssysteem onafhankelijke ASP QSYS.LIB:

- Logische bestanden worden niet ondersteund.
- Fysieke bestanden die worden ondersteund voor toegang in de werkstand Tekst zijn door het programma beschreven fysieke bestanden met een enkel veld en fysieke bronbestanden met een enkel tekstveld. Tot de fysieke bestanden die worden ondersteund voor toegang in de binaire werkstand behoren, naast de bestanden die worden ondersteund voor toegang in de werkstand Tekst, extern beschreven fysieke bestanden.
- Byte-range-vergrendeling wordt niet ondersteund. (Raadpleeg voor informatie over byte-range-vergrendeling het onderwerp fcntl() in het iSeries Informatiecentrum.)
- Als een taak een databasebestandssectie geopend heeft, krijgt telkens slechts één taak schrijftoegang voor die bestandssectie. Andere aanvragen krijgen alleen leestoegang.

Ondersteuning voor gebruikersruimten in het bestandssysteem onafhankelijke ASP QSYS.LIB: Onafhankelijke ASP QSYS.LIB ondersteunt stroominvoer- en -uitvoerbewerkingen naar objecten in de gebruikersruimte. Een programma kan bijvoorbeeld stroomgegevens naar een gebruikersruimte schrijven en gegevens van een gebruikersruimte lezen. De maximumgrootte van een gebruikersruimte is 16 776 704 bytes.

Vergeet niet dat gebruikersruimten niet zijn gelabeld met een CCSID. Daarom is het teruggezonden CCSID het standaard-CCSID van de taak.

Ondersteuning voor opslagbestanden in het bestandssysteem onafhankelijke ASP QSYS.LIB: De onafhankelijke ASP QSYS.LIB ondersteunt stroominvoer/uitvoerbewerkingen voor het opslaan van bestandsobjecten. Een bestaand opslagbestand heeft bijvoorbeeld gegevens die door een ander bestand gelezen of gekopieerd worden totdat het nodig is de gegevens in een ander bestaand en leeg opslagbestandsobject te plaatsen. Als een opslagbestand geopend is voor schrijfbewerkingen zijn geen andere openinstances van het bestand toegestaan. Een opslagbestand staat **wel** meerdere openinstances voor leesbewerkingen toe, mits de taak meer dan één openinstance van het bestand heeft voor leesbewerkingen. Een opslagbestand kan niet worden geopend voor lee/schrijftoegang. Stroominvoer- en/of uitvoerbewerkingen om bestandsgegevens op te slaan zijn niet toegestaan als er in een taak meerdere threads worden uitgevoerd.

Stroominvoer- en/of uitvoerbewerkingen naar een opslagbestand worden niet ondersteund als een opslagbestand of zijn directory wordt geëxporteerd via de netwerkbestandssysteemserver. Ze mogen echter worden geopend vanaf een PC-client en via het bestandssysteem QFileSvr.400.

Hoofdlettergevoeligheid in het bestandssysteem onafhankelijke ASP QSYS.LIB: Doorgaans maakt het bestandssysteem onafhankelijke ASP QSYS.LIB geen onderscheid tussen hoofd- en kleine letters in objectnamen. Een zoekbewerking naar objectnamen levert hetzelfde resultaat op, of de naam nu bestaat uit hoofd- of kleine letters.

Als een naam echter tussen aanhalingstekens staat, wordt de kast van elk teken bewaard. Als u dus zoekt met een tekenreeks tussen aanhalingstekens, geeft u een hoofdlettergevoelige zoekopdracht.

Padnamen in het bestandssysteem onafhankelijke ASP QSYS.LIB:

- Elke component van de padnaam moet de objectnaam bevatten gevolgd door het objecttype van het object. Bijvoorbeeld:

```
/asp_naam/QSYS.LIB/QGPL.LIB/PRT1.OUTQ
```

```
/asp_naam/QSYS.LIB/EMP.LIB/PAY.FILE/TAX.MBR
```

waarbij asp_naam de naam van de onafhankelijke ASP is. Objectnaam en -type zijn van elkaar gescheiden door een punt (.). Objecten in een bibliotheek kunnen dezelfde naam hebben als het om verschillende objecttypen gaat. Het objecttype moet dus worden opgegeven om het object te identificeren.

- De objectnaam in elke component mag maximaal 10 tekens lang zijn, het objecttype maximaal 6 tekens.

- De directoryhiërarchie in de onafhankelijke ASP QSYS.LIB kan al naar gelang het object twee tot drie niveaus hebben (twee of drie componenten in de padnaam). Als het object een databasebestand is, kan de hiërarchie drie niveaus hebben (bibliotheek, bestand, sectie). Anders zijn slechts twee niveaus mogelijk (bibliotheek, object). De combinatie van de lengte van elke componentnaam en het aantal directoryniveaus bepaalt de maximumlengte van de padnaam.

Als /, asp_naam en QSYS.LIB bij de eerste drie niveaus horen, kan de directoryhiërarchie van de onafhankelijke ASP QSYS.LIB zes niveaus bevatten.

- Wanneer de namen worden opgeslagen, worden de tekens geconverteerd naar CCSID 37. Namen met aanhalingstekens echter worden opgeslagen met het CCSID van de taak.

Raadpleeg voor meer informatie over CCSID het onderwerp Lokaliseren in het iSeries Informatiecentrum.

Koppelingen in het bestandssysteem onafhankelijke ASP QSYS.LIB: Symbolische koppelingen kunnen in het bestandssysteem onafhankelijke ASP QSYS.LIB niet worden gemaakt of opgeslagen.

De relatie tussen een bibliotheek en objecten in een bibliotheek is gelijk aan één vaste koppeling tussen de bibliotheek en elk object in de bibliotheek. Het geïntegreerd bestandssysteem behandelt de relatie bibliotheek/object als een koppeling. Daarom is het mogelijk een koppeling te maken tussen een bestandssysteem dat symbolische koppelingen ondersteunt en een object in het bestandssysteem onafhankelijke ASP QSYS.LIB.

Raadpleeg “Koppeling” op pagina 11 voor een beschrijving van koppelingen.

Oprachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem onafhankelijke ASP QSYS.LIB: De opdrachten die worden afgebeeld in het onderwerp “Toegang met behulp van CL-opdrachten” op pagina 61 werken in het bestandssysteem onafhankelijke ASP QSYS.LIB, met uitzondering van:

- De opdracht ADDLNK kan alleen worden gebruikt om een symbolische koppeling te maken *naar* een object in de onafhankelijke ASP QSYS.LIB.
- Bestandsbewerkingen kunnen alleen worden uitgevoerd op door het programma beschreven fysieke bestanden en fysieke bronbestanden.
- De opdrachten STRJRN en ENDJRN kunnen niet worden gebruikt op fysieke databasebestanden.
- U kunt geen bibliotheken in het bestandssysteem onafhankelijke ASP QSYS.LIB verplaatsen naar ASP's met behulp van de opdracht MOV. U kunt echter wel bibliotheken in de onafhankelijke ASP QSYS.LIB verplaatsen naar de systeem-ASP of andere onafhankelijke ASP's.
- Als u SAV of RST gebruikt om bibliotheekobjecten op een onafhankelijke ASP op te slaan of terug te zetten, moet die onafhankelijke ASP bij de taak horen die de SAV of RST uitvoeren of de onafhankelijke ASP moet worden opgegeven bij de parameter ASPDEV. De naamgeving /asp_naam/QSYS.LIB/object.type wordt niet ondersteund bij SAV en RST.

Dezelfde beperkingen zijn van toepassing op de gebruikersschermen beschreven in “Toegang met behulp van menu's en schermen” op pagina 60.

API's van de onafhankelijke ASP geïntegreerd bestandssysteem gebruiken in het bestandssysteem QSYS.LIB: De API's die worden afgebeeld in het onderwerp “Handelingen uitvoeren met behulp van API's” op pagina 98 werken in het bestandssysteem onafhankelijke ASP QSYS.LIB, met uitzondering van de volgende:

- Bestandsbewerkingen kunnen alleen worden uitgevoerd op door het programma beschreven fysieke bestanden en fysieke bronbestanden.
- De functie symlink() kan alleen worden gebruikt om *naar* een object te koppelen in de onafhankelijke ASP QSYS.LIB vanaf een ander bestand dat symbolische koppelingen ondersteunt.
- De API's QjoStartJournal() en QjoEndJournal() kunnen niet worden gebruikt op fysieke databasebestanden.

Bestandssysteem voorzieningen documentenbibliotheek (QDLS)

Het bestandssysteem QDLS ondersteunt de mappenstructuur. Het biedt toegang tot documenten en mappen.

Daarnaast geldt:

- Ondersteuning van mappen en objecten in de documentenbibliotheek van iSeries-server.
- Ondersteunt gegevens opgeslagen in stroombestanden

Raadpleeg voor meer informatie over QDLS het onderwerp QDLS gebruiken via de interface van het geïntegreerd bestandssysteem.

QDLS gebruiken via de interface van het geïntegreerd bestandssysteem

Het bestandssysteem QDLS kan worden geopend via de IFS-interface met behulp van de OS/400-bestandserver of de opdrachten, gebruikersschermen en API's van het geïntegreerd bestandssysteem. Bij gebruik van de interface van het geïntegreerd bestandssysteem moet u met de volgende aspecten en beperkingen rekening houden.

Raadpleeg voor meer informatie over het QDLS de volgende onderwerpen.

- "Geïntegreerd bestandssysteem en hiërarchisch bestandssysteem in het QDLS-bestandssysteem"
- "Gebruikersregistratie in het bestandssysteem QDLS"
- "Hoofdlettergevoeligheid in het bestandssysteem QDLS"
- "Padnamen in het bestandssysteem QDLS" op pagina 44
- "Koppelingen in het bestandssysteem QDLS" op pagina 44
- "Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QDLS" op pagina 44
- "API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QDLS" op pagina 44

Geïntegreerd bestandssysteem en hiërarchisch bestandssysteem in het QDLS-bestandssysteem:

Bewerkingen op objecten in het bestandssysteem QDLS kunnen niet alleen worden uitgevoerd met behulp van CL-opdrachten voor objecten in de documentenbibliotheek, maar ook via de IFS-interface of via API's die beschikbaar zijn via een hiërarchisch bestandssysteem. In tegenstelling tot het geïntegreerd bestandssysteem, dat gebaseerd is op het ILE-programmamodel (Integrated Language Environment), is het hiërarchisch bestandssysteem gebaseerd op het oorspronkelijke iSeries-serverprogrammamodel.

Met de API's van het hiërarchisch bestandssysteem kunt u een aantal extra bewerkingen uitvoeren die het geïntegreerd bestandssysteem niet ondersteunt. Met name kunt u API's van het hiërarchisch bestandssysteem gebruiken om uitgebreide directorykenmerken (of *directorygegevenskenmerken*) te openen en te wijzigen. Vergeet niet dat de naamgevingsregels voor het gebruik van API's van het hiërarchisch bestandssysteem anders zijn dan voor API's via de IFS-interface.

Raadpleeg voor meer informatie over het hiërarchisch bestandssysteem het onderwerp over API's van het hiërarchisch bestandssysteem in het iSeries Informatiecentrum.

Gebruikersregistratie in het bestandssysteem QDLS: Om te kunnen werken met objecten in QDLS moet u zijn geregistreerd in het adressenbestand van het systeem.

Hoofdlettergevoeligheid in het bestandssysteem QDLS: QDLS converteert de kleine letters **a** tot en met **z** in hoofdletters als deze worden gebruikt in objectnamen. Daarom is een zoekbewerking naar objectnamen met alleen die tekens niet hoofdlettergevoelig.

Alle andere tekens in QDLS zijn hoofdlettergevoelig.

Raadpleeg voor meer informatie het onderwerp Map- en documentnaam in het iSeries Informatiecentrum.

Padnamen in het bestandssysteem QDLS:

- Elk onderdeel van de padnaam kan uit slechts één naam bestaan, zoals:

/QDLS/FLR1/DOC1

of een naam plus een extensie (zoals in DOS), zoals:

/QDLS/FLR1/DOC1.TXT

- De naam in elke component mag maximaal 8 tekens lang zijn en de extensie, indien gebruikt, 3 tekens. De maximumlengte van de padnaam is 82 tekens, uitgaande van een absolute padnaam die begint met /QDLS.
- De directoryhiërarchie in QDLS kan 32 niveaus hebben. Als / en QDLS tot de eerste twee niveaus behoren, kan de directoryhiërarchie 34 niveaus hebben.
- De tekens in namen worden geconverteerd naar de codetabel van de taak als de namen worden opgeslagen, tenzij gegevensgebied Q0DEC500 is gemaakt in de bibliotheek QUSRSYS. Als dit gegevensgebied bestaat, worden de tekens in namen geconverteerd naar codetabel 500 wanneer de namen worden opgeslagen. Deze functie zorgt voor compatibiliteit met de werking van het bestandssysteem QDLS in vorige releases. Een naam kan worden geweigerd als deze niet kan worden geconverteerd naar de juiste codetabel.

Raadpleeg voor meer informatie over codetabellen het onderwerp Lokaliseren in het iSeries Informatiecentrum.

Koppelingen in het bestandssysteem QDLS: Symbolische koppelingen kunnen in het bestandssysteem QDLS niet worden gemaakt of opgeslagen.

Het geïntegreerd bestandssysteem beschouwt de relatie tussen een map en objecten in documentenbibliotheek als identiek aan aan één koppeling tussen de map en elk object in de map. Daarom is het mogelijk een koppeling te maken naar een object in het bestandssysteem QDLS vanaf een bestandssysteem dat symbolische koppelingen ondersteunt.

Raadpleeg "Koppeling" op pagina 11 voor een beschrijving van koppelingen.

Opgdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem

QDLS: De opdrachten die worden afgebeeld in het onderwerp "Toegang met behulp van CL-opdrachten" op pagina 61 werken in het bestandssysteem QDLS, met uitzondering van de volgende:

- De opdracht ADDLNK kan alleen worden gebruikt om een koppeling *naar* een object in QDLS te maken vanaf een ander bestand dat symbolische koppelingen ondersteunt.
- De opdrachten CHKIN en CHKOUT worden ondersteund voor bestanden maar niet voor directory's.
- De opdrachten APYJRNCHG, CHGJRNOBJ, ENDJRN, SNDJRNE en STRJRN worden niet ondersteund.

Dezelfde beperkingen zijn van toepassing op de gebruikersschermen beschreven in "Toegang met behulp van menu's en schermen" op pagina 60.

API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QDLS: De API's die worden afgebeeld in het onderwerp "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in het bestandssysteem QDLS, met uitzondering van de volgende:

- De functie symlink() kan alleen worden gebruikt om naar een object te koppelen in QDLS vanaf een ander bestand dat symbolische koppelingen ondersteunt.
- De volgende functies worden niet ondersteund:

givedescriptor()

ioctl()

link()

QjoEndJournal()

QjoRetrieveJournalInformation()

QJorjidi()
QJOSJRNE()
QjoStartJournal()
Qp0lGetPathFromFileID()
readlink()
takedescriptor()

Bestandssysteem optische media (QOPT)

Het bestandssysteem QOPT biedt toegang tot stroomgegevens die zijn opgeslagen op optische media.

Daarnaast geldt:

- Het biedt een hiërarchische directorystructuur die lijkt op PC-besturingssystemen zoals DOS en OS/2.
- Het is geoptimaliseerd voor in- en uitvoer van stroombestanden.
- Ondersteunt gegevens opgeslagen in stroombestanden

Raadpleeg voor meer informatie over QOPT het onderwerp QOPT gebruiken via het geïntegreerd bestandssysteem.

QOPT gebruiken via het geïntegreerd bestandssysteem

Het bestandssysteem QOPT kan worden geopend via het geïntegreerd bestandssysteem met behulp van de opdrachten, gebruikersschermen en API's van de PC-server of het geïntegreerd bestandssysteem. Bij gebruik van de interface van het geïntegreerd bestandssysteem moet u met de volgende aspecten en beperkingen rekening houden.

Raadpleeg voor meer informatie over het QOPT de volgende onderwerpen.

- “Geïntegreerd bestandssysteem en hiërarchisch bestandssysteem in het QOPT-bestandssysteem”
- “Hoofdlettergevoeligheid in het bestandssysteem QOPT” op pagina 46
- “Padnamen in het bestandssysteem QOPT” op pagina 46
- “Koppelingen in het bestandssysteem QOPT” op pagina 46
- “Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QOPT” op pagina 46
- “API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QOPT” op pagina 47

Raadpleeg voor meer informatie de publicatie Ondersteuning voor optische media .

Geïntegreerd bestandssysteem en hiërarchisch bestandssysteem in het QOPT-bestandssysteem:

Bewerkingen op objecten in het bestandssysteem QOPT kunnen worden uitgevoerd via de IFS-interface of via API's die beschikbaar zijn via een hiërarchisch bestandssysteem. In tegenstelling tot het geïntegreerd bestandssysteem, dat gebaseerd is op het ILE-programmamodel (Integrated Language Environment), is het hiërarchisch bestandssysteem gebaseerd op het oorspronkelijke iSeries-serverprogramma-model.


Met de API's van het hiërarchisch bestandssysteem kunt u een aantal extra bewerkingen uitvoeren die het geïntegreerd bestandssysteem niet ondersteunt. Met name kunt u API's van het hiërarchisch bestandssysteem gebruiken om uitgebreide directorykenmerken (of *directorygegevenskenmerken*) te openen en te wijzigen en om te werken met vastgehouden optische-mediabestanden. Vergeet niet dat de naamgevingsregels voor het gebruik van API's van het hiërarchisch bestandssysteem anders zijn dan voor API's via de IFS-interface.

Raadpleeg voor meer informatie over API's van het hiërarchisch bestandssysteem het onderwerp over API's van het hiërarchisch bestandssysteem in het iSeries Informatiecentrum of in de publicatie Ondersteuning voor optische media  .

Hoofdlettergevoeligheid in het bestandssysteem QOPT: Afhankelijk van het optische medium worden hoofd- en kleine letters al dan niet gehandhaafd bij het maken van bestanden en directory's in QOPT. Zoekbewerkingen van bestanden en directory's zijn echter niet hoofdlettergevoelig, ongeacht de indeling van het optische medium.

Padnamen in het bestandssysteem QOPT:

- De padnaam moet met een schuine streep naar rechts (/) beginnen. Het pad bestaat uit de naam van het bestandssysteem, de naam van het volume, de namen van de directory en subdirectory en de bestandsnaam. Bijvoorbeeld:
`/QOPT/VOLUMENAAM/DIRECTORYNAAM/SUBDIRECTORYNAAM/BESTANDSNAAM`
- De naam van het bestandssysteem, QOPT, is vereist.
- De naamlength van het volume en het pad variëren al naar de gelang de indeling van het optische medium.
- U kunt /QOPT opgeven in de padnaam of een of meer directory's of subdirectory's. In een directory- of bestandsnaam kan elk teken staan behalve X'00' t/m X'3F' en X'FF'. Afhankelijk van de indeling van het optische medium kunnen er nog andere beperkingen gelden.
- De bestandsnaam is het laatste onderdeel van de padnaam. De lengte van de bestandsnaam is beperkt door de lengte van de directorynaam in het pad.

Raadpleeg voor meer informatie over de regels voor padnamen in het bestandssysteem QOPT het onderwerp "Padnaamregels" in de publicatieOndersteuning voor optische media  .

Koppelingen in het bestandssysteem QOPT: Het bestandssysteem QOPT ondersteunt slechts één koppeling naar een object. Symbolische koppelingen kunnen niet worden gemaakt of opgeslagen in QOPT. Bestanden in QOPT kunnen worden geopend met een symbolische koppeling vanuit het bestandssysteem "koppeling" (/) of QOpenSys.

Raadpleeg "Koppeling" op pagina 11 voor een beschrijving van koppelingen.

Oprachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QOPT: De meeste opdrachten die worden afgebeeld in het onderwerp "Toegang met behulp van CL-opdrachten" op pagina 61 werken in het bestandssysteem QOPT. Er zijn echter in het bestandssysteem QOPT een paar uitzonderingen. Bedenk dat het onveilig kan zijn om deze CL-opdrachten te gebruiken in een multithread-proces. Afhankelijk van de indeling van het optische medium kunnen er nog andere beperkingen gelden. Dezelfde beperkingen zijn van toepassing op de gebruikersschermen beschreven in "Toegang met behulp van menu's en schermen" op pagina 60.

De onderstaande opdrachten van het geïntegreerd bestandssysteem worden niet ondersteund in het bestandssysteem QOPT:

- ADDLNK
- APYJRNCHG
- CHGJRNOBJ
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRNE
- STRJRN

- WRKOBJOWN
- WRKOBJPGP

API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QOPT: Alle API's die worden afgebeeld in het onderwerp "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in het bestandssysteem "hoofddirectory" (/) op een threadsafe manier, behalve:

- QjoEndJournal()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()

Bestandssysteem NetWare (QNetWare)

Het bestandssysteem QNetWare biedt toegang tot gegevens op een lokale Integrated xSeries Server for iSeries of op afstand die draaien met Novell NetWare 4.10 of 4.11, of tot zelfstandige PC-servers die draaien op Novell NetWare 3.12, 4.10, 4.11 of 5.0.

Daarnaast geldt:

- Geeft toegang tot NDS-objecten
- Ondersteunt gegevens opgeslagen in stroombestanden
- Biedt dynamisch aankoppelen van NFS in de lokale naamruimte

Opmerking: Het bestandssysteem QNetWare is alleen beschikbaar als NetWare Enhanced Integration for iSeries 400, BOSS optie 25, op het systeem is geïnstalleerd. Na de volgende opstartprocedure verschijnen de directory /QNetWare en zijn subdirectory's als onderdeel van de directorystructuur van het geïntegreerd bestandssysteem.

Raadpleeg de volgende onderwerpen voor meer informatie over het bestandssysteem QNetWare:

- NetWare-bestandssystemen aankoppelen
- QNetWare-directorystructuur
- QNetWare gebruiken via de IFS-interface

NetWare-bestandssystemen aankoppelen

NetWare-bestandssystemen die zich op Novell NetWare-servers bevinden, kunnen worden aangekoppeld op de bestandssystemen "hoofddirectory" (/), QOpenSys en andere bestandssystemen zodat ze makkelijker kunnen worden geopend en beter presteren dan onder de directory /QNetWare. Het aankoppelen van NetWare-bestandssystemen kan ook worden gebruikt om te profiteren van de mogelijkheden van de opdracht ADDMFS (Gekoppeld bestandssysteem toevoegen), zoals het aankoppelen van een lezen/schrijven-bestandssysteem als een alleen lezen-bestandssysteem.

NetWare-bestandssystemen kunnen worden aangekoppeld met een NDS-pad of door een NetWare-pad op te geven met de indeling SERVER/VOLUME:directory/directory. Om bijvoorbeeld de directory doorgang aan te koppelen, die zich bevindt op volume Nest op de server Dreyfuss, gebruikt u deze syntaxis:

```
DREYFUSS/NEST:doorgang
```

Deze padsyntaxis lijkt erg op de MAP-opdrachtsyntaxis van NetWare. NDS-paden kunnen worden gebruikt om een pad op te geven naar een NetWare-volume, maar kunnen niet zelfstandig worden aangekoppeld.

QNetWare-directorystructuur

De /QNetWare-directorystructuur staat voor meerdere specifieke bestandssystemen:

- De structuur beeldt Novell NetWare-servers en -volumes af in de volgende vorm:

/QNetWare/SERVER.SVR/VOLUME

De extensie .SVR wordt gebruikt om een Novell NetWare-server aan te geven.

- Wanneer een volume onder een server wordt geopend via de menu's, opdrachten of API's van het geïntegreerd bestandssysteem, wordt de hoofddirectory van het NetWare-volume automatisch aangekoppeld op de directory VOLUME onder /QNetWare.
- QNetWare geeft NDS-boomstructuren als volgt weer:

/QNetWare/CORP_TREE.TRE/USA.C/ORG.0/ORG_UNIT.OU/SVR1_VOL.CN

De extensie .TRE staat voor NDS-boomstructuren, .C staat voor landen, .0 staat voor organisaties, .OU staat voor organisatie-eenheden en .CN wordt gebruikt voor gemeenschappelijke namen. Als een Novell NetWare-volume wordt geopend via het NDS-pad via een volume-object of een alias van een volume-object, wordt zijn hoofddirectory ook automatisch aangekoppeld op het NDS-object.

QNetWare gebruiken via de IFS-interface

Het QNetWare-bestandssysteem kan worden geopend met de IFS-interface via de opdrachten, gebruikersschermen en API's van de OS/400-bestandserver of het geïntegreerd bestandssysteem. Houd rekening met de volgende aspecten, beperkingen en dependency's.

Raadpleeg voor meer informatie over het bestandssysteem QNetWare:

- "Machtigingen en eigendom in het bestandssysteem QNetWare"
- "Audit in het bestandssysteem QNetWare"
- "Bestanden en directory's in het bestandssysteem QNetWare"
- "NDS-objecten in het bestandssysteem QNetWare"
- "Koppelingen in het bestandssysteem QNetWare" op pagina 49
- "Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QNetWare" op pagina 49
- "API's van het geïntegreerd bestandssysteem gebruiken in het QNetWare-bestandssysteem" op pagina 49

Machtigingen en eigendom in het bestandssysteem QNetWare: Bestanden en directory's in QNetWare worden opgeslagen en beheerd door Novell NetWare-servers. Bij het gebruik van opdrachten en API's om de machtigingen van eigenaars of gebruikers op te halen of in te stellen, wijst QNetWare de NetWare-gebruikers toe aan iSeries-servergebruikers op basis van de gebruikersnaam. Als de NetWare-naam de tien tekens overschrijdt of een corresponderende iSeries-server niet bestaat, wordt de machtiging niet toegewezen. Eigenaars die niet kunnen worden toegewezen, worden automatisch toegewezen naar het gebruikersprofiel QDFTOWN. De machtigingen van gebruikers kunt u afbeelden en wijzigen met de opdrachten WRKAUT en CHGAUT. Als machtigingen worden overgebracht van en naar de server worden ze toegewezen aan iSeries-servermachtigingen.

Audit in het bestandssysteem QNetWare: Hoewel Novell NetWare de audit van bestanden en directory's ondersteunt, kan het bestandssysteem QNetWare de auditwaarden van deze objecten niet wijzigen. Daarom wordt de opdracht CHGAUD niet ondersteund.

Bestanden en directory's in het bestandssysteem QNetWare: Het bestandssysteem QNetWare onthoudt niet de hoofd- en kleine letters waarin bestanden en directory's worden opgegeven bij een opdracht of API. Bij verzending naar de NetWare-server worden alle namen in hoofdletters gezet. Novell NetWare ondersteunt ook de naamruimte van meerdere platforms, zoals DOS, OS/2, Apple Macintosh en NFS. Het bestandssysteem QNetWare ondersteunt alleen de DOS-naamruimte. Aangezien de DOS-naamruimte vereist is op alle Novell NetWare-volumes verschijnen alle bestanden en directory's in het bestandssysteem QNetWare.

NDS-objecten in het bestandssysteem QNetWare: Het bestandssysteem QNetWare ondersteunt het afbeelden van NDS-namen in hoofd- en kleine letters.

Koppelingen in het bestandssysteem QNetWare: Het bestandssysteem QNetWare ondersteunt slechts één koppeling naar een object. Symbolische koppelingen kunnen niet worden gemaakt of opgeslagen in QNetWare. Symbolische koppelingen kunnen echter wel worden gemaakt in de directory's "hoofd-directory" (/) en QOpenSys die naar een QNetWare-bestand of -directory verwijzen.

Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QNetWare: De opdrachten die worden afgebeeld in het onderwerp "Toegang met behulp van CL-opdrachten" op pagina 61 werken in het bestandssysteem QNetWare, met uitzondering van de volgende:

- ADDLINK
- APYJRNCHG
- CHGAUD
- CHGJRNOBJ
- CHGPGP
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

Behalve de genoemde opdrachten kunnen ook de volgende opdrachten niet worden gebruikt op NDS-objecten, servers en volumes:

- CHGOWN
- CPYFRMSTMF
- CPYTOSTMF
- CRTDIR

API's van het geïntegreerd bestandssysteem gebruiken in het QNetWare-bestandssysteem: De API's die worden afgebeeld in het onderwerp "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in het bestandssysteem QNetWare, met uitzondering van de volgende:

- givedescriptor()
- link()
- QjoEndJournal()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()
- readlink()
- symlink()
- takedescriptor()

Behalve de genoemde API's kunnen ook de volgende API's niet worden gebruikt op NDS-objecten, servers en volumes:

- chmod()
- chown()
- create()
- fchmod()

fchown()
fcntl()
ftruncate()
lseek()
mkdir()
read()
readv()
unmask()
write()
writev()

Bestandssysteem iSeries NetClient (QNTC)

Het bestandssysteem QNTC biedt toegang tot gegevens en objecten die zijn opgeslagen op een lokale Integrated xSeries Server for iSeries of op afstand die op Windows NT 4.0 Server of hoger draaien of op een zelfstandige server. Hierdoor kunnen toepassingen op de iSeries-server dezelfde gegevens gebruiken als Windows-clients.

- | Het bestandssysteem QNTC maakt deel uit van het basisbesturingssysteem OS/400. Om /QNTC te kunnen openen, is het niet nodig om iSeries 400 Integration with Windows NT Server, option 29 van het besturingssysteem te installeren.

Raadpleeg voor meer informatie over QNTC het onderwerp QNTC gebruiken via de IFS-interface.

QNTC gebruiken via de IFS-interface

- | Door de opdrachten, gebruikersschermen en API's van iSeries NetServer, iSeries Navigator en het geïntegreerd bestandssysteem te gebruiken, kunt het QNTC-bestandssysteem openen via de IFS-interface. Houd rekening met de volgende aspecten en beperkingen.

De volgende onderwerpen geven meer informatie over het QNTC-bestandssysteem.

- "Machtigingen en eigendom in het bestandssysteem QNTC"
- "Hoofdlettergevoeligheid in het bestandssysteem QNTC"
- "Padnamen in het bestandssysteem QNTC" op pagina 51
- "Koppelingen in het bestandssysteem QNTC" op pagina 51
- "Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QNTC" op pagina 51
- "De opdracht MKDIR gebruiken in het bestandssysteem QNTC" op pagina 52
- "API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QNTC" op pagina 52
- | • "Kerberos inschakelen voor gebruik met het bestandssysteem QNTC" op pagina 52

Machtigingen en eigendom in het bestandssysteem QNTC: Het bestandssysteem QNTC ondersteunt het concept van eigendom van een bestand of een directory niet. Een opdracht of API om het eigendom van bestanden die in een QNTC zijn opgeslagen te wijzigen, mislukt. Een systeemgebruikersprofiel genaamd QDFTOWN is eigenaar van alle bestanden en alle directory's van QNTC.

De machtiging voor NT-serverbestanden en -directory's wordt beheerd vanaf de Windows NT-server. QNTC ondersteunt de opdrachten WRKAUT en CHGAUT niet.

Hoofdlettergevoeligheid in het bestandssysteem QNTC: Het bestandssysteem handhaaft de hoofd- en kleine letters die zijn opgegeven in de objectnamen, maar maakt geen onderscheid tussen de hoofd- en kleine letters in de namen. Een zoekbewerking naar objectnamen levert hetzelfde resultaat op, of de naam nu bestaat uit hoofd- of kleine letters.

Padnamen in het bestandssysteem QNTC:

- De padnaam moet met een schuine streep naar rechts beginnen en kan maximaal 255 tekens bevatten.
- Padnamen zijn hoofdlettergevoelig.
- Het pad bestaat uit de naam van het bestandssysteem, de Windows NT-servernaam, de sharenaam, de namen van de directory's en subdirectory's en de objectnaam. Padnamen hebben de volgende vorm:
 /QNTC/Servernaam/Sharenaam/Directory/ . . . /Object
 (QNTC is een verplicht onderdeel van de padnaam.)
- De servernaam kan maximaal 15 tekens lang zijn. Hij moet deel uitmaken van het pad.
- De sharenaam kan maximaal 12 tekens lang zijn.
- Elk onderdeel van de padnaam achter de sharenaam kan bestaan uit maximaal 255 tekens.
- In QNTC zijn meestal 130 hiërarchieniveaus beschikbaar. Als alle onderdelen van de padnaam zijn opgenomen als hiërarchieniveaus, kan de directoryhiërarchie 132 niveaus diep zijn.
- Namen worden opgeslagen in het Unicode-CCSID.
- Elke functionele Windows NT-server in het lokale subnet, verschijnt automatisch als een directory onder /QNTC. Gebruik de opdracht MKDIR (Directory maken, zie Tabel 7 op pagina 61) of de API mkdir() (zie "Toegang met behulp van API's" op pagina 80) om op Windows gebaseerde servers buiten het lokale subnet toe te voegen.

Koppelingen in het bestandssysteem QNTC: Het bestandssysteem QNTC ondersteunt slechts één koppeling naar een object. U kunt in QNTC geen symbolische koppelingen maken of opslaan. Om gegevens te openen in QNTC kunt u een symbolische koppeling gebruiken in het bestandssysteem "hoofd-directory" (/) of QOpenSys.

Raadpleeg "Koppeling" op pagina 11 voor een beschrijving van koppelingen.

Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem

QNTC: De opdrachten die worden afgebeeld in het onderwerp "Toegang met behulp van CL-opdrachten" op pagina 61 werken in het bestandssysteem QNTC, met uitzondering van de volgende:

- ADDLNK
- APYJRNCHG
- | CHGJRNOBJ
- CHGOWN
- CHGAUT
- CHGPGP
- CHKIN
- CHKOUT
- DSPAUT
- ENDJRN
- | RST (leverbaar bij Integrated xSeries Servers)
- | SAV (leverbaar bij Integrated xSeries Servers)
- SNDJRNE
- STRJRN
- WRKAUT
- WRKOBJOWN
- WRKOBJPGP

Dezelfde beperkingen zijn van toepassing op de gebruikersschermen die worden beschreven in "Toegang met behulp van menu's en schermen" op pagina 60.

De opdracht MKDIR gebruiken in het bestandssysteem QNTC: Met de opdracht MKDIR (Directory maken) voegt u een serverdirectory toe aan de directory /QNTC. Een QNTC-directory wordt automatisch gemaakt voor alle functionele Windows-servers in het iSeries NetServer-domein en lokale subnet. De op Windows gebaseerde servers buiten het lokale subnet of iSeries NetServer-domein moeten worden toegevoegd met de opdracht MKDIR of de API mkdir(). Bijvoorbeeld:

```
MKDIR '/QNTC/NTSRV1'
```

voegt de NTSRV1-server toe in de directorystructuur van het QNTC-bestandssysteem om toegang tot bestanden en directory's op die server mogelijk te maken.

U kunt een nieuwe server ook toevoegen aan de directorystructuur met behulp van het TCP/IP-adres. Bijvoorbeeld:

```
MKDIR '/QNTC/9.130.67.24'
```

voegt de server toe in de directorystructuur van het QNTC-bestandssysteem.

Opmerking: Door iSeries NetServer te configureren voor WINS is het mogelijk om automatisch directory's voor servers buiten uit subnet te maken.

Opmerking: Directory's die met de API mkdir() of de CL-opdracht MKDIR toevoegt aan de directorystructuur, blijven niet zichtbaar na een herstart. Na elke herstart moet de opdracht MKDIR of de API mkdir() opnieuw worden uitgevoerd.

API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem QNTC: De API's die worden afgebeeld in het onderwerp "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in het bestandssysteem QNTC, met uitzondering van de volgende:

- De functies chmod(), fchmod(), utime() en umask() hebben geen invloed op de objecten in QNTC, maar er treedt geen fout op als u ze toch probeert te gebruiken.
- Het bestandssysteem QNTC ondersteunt het volgende functies niet:

```
chown()
fchown()
givedescriptor()
link()
QjoEndJournal()
QjoRetrieveJournalInformation()
QJORJIDI()
QJOSJRNE()
QjoStartJournal()
Qp0lGetPathFromFileID()
readlink()
symlink()
takedescriptor()
```

| **Kerberos inschakelen voor gebruik met het bestandssysteem QNTC:** Met QNTC kunt u met de iSeries
| toegang krijgen tot Windows-servers die het Kerberos V5-verificatieprotocol ondersteunen. In plaats van
| te verifiëren met elke server door middel van een wachtwoord van het LAN-beheertype, kunt u met de
| iSeries toegang krijgen tot ondersteunde CIFS-servers met een enkele aanmeldbewerking.

| Voordat u Kerberos inschakelt voor gebruik met het bestandssysteem QNTC moet u het volgende instel-
| len:

- | • Netwerkverificatieservice
- | • EIM (Enterprise Identity Mapping)

- | Als de bovenstaande items zijn geconfigureerd, kunt u een gebruiker in staat stellen Kerberos te gebruiken met het bestandssysteem QNTC. Om een gebruiker gebruik te kunnen laten maken van de Kerberos-ondersteuning, moet u de volgende stappen uitvoeren.
- | • Van het iSeries-gebruikersprofiel van de gebruiker moet de lokale wachtwoordbeheerparameter LCLPWDMGT zijn ingesteld op *NO. Door *NO op te geven heeft de gebruiker geen wachtwoord voor het systeem en kan hij zich niet aanmelden op een 5250-sessie. De enige manier om toegang te krijgen tot het systeem is via voor Kerberos geschikte toepassingen als iSeries Navigator.
- | Als de gebruikers *YES opgeeft, wordt het wachtwoord beheerd door het systeem en wordt de gebruiker gemachtigd zonder Kerberos.
- | • U moet beschikken over een Kerberos-ticket en een iSeries Navigator-verbinding.
- | • Het Kerberos-ticket dat u voor de iSeries gebruikt, moet kunnen worden doorgezonden. U maakt een ticket als volgt doorzendbaar:
 - | – Open het hulpprogramma 'Gebruikers en computers actieve directory' op het sleutel-distributiecentrum voor uw Kerberos-realm
 - | – Selecteer gebruikers
 - | – Selecteer de naam die overeenkomt met de service-eigenaar
 - | – Selecteer Eigenschappen
 - | – Selecteer het tabblad Account
 - | – Kruis bij de Account-opties 'Account is betrouwbaar voor machtiging'

OS/400 Bestandssysteem bestandserver (QFileSvr.400)

Het bestandssysteem OS/400-bestandserver biedt transparante toegang tot andere bestandssystemen die op iSeries-servers op afstand bevinden. Het wordt geopend via een hiërarchische directorystructuur.

U kunt het bestandssysteem QFileSvr.400 beschouwen als een client die namens gebruikers opslagopdrachten uitvoert. QFileSvr.400 werkt samen met de OS/400-bestandserver op het doelsysteem om de feitelijke bestandsbewerking uit te voeren.

Raadpleeg voor meer informatie over QFileSvr.400 QFileSvr.400 gebruiken via de IFS-interface.

QFileSvr.400 gebruiken via de IFS-interface

Het bestandssysteem QFileSvr.400 kan worden geopend via de IFS-interface met behulp van de OS/400-bestandserver of de opdrachten, gebruikersschermen en API's van het geïntegreerd bestandssysteem. Bij gebruik van de interface van het geïntegreerd bestandssysteem moet u met de volgende aspecten en beperkingen rekening houden.

Opmerking: De kenmerken van het bestandssysteem QFileSvr.400 worden bepaald door de kenmerken van het bestandssysteem dat wordt geopend op de doelserver.

De volgende onderwerpen geven meer informatie over het bestandssysteem QFileSvr.400

- "Hoofdlettergevoeligheid in het bestandssysteem OS/400-bestandserver" op pagina 54
- "Padnamen in het bestandssysteem OS/400-bestandserver" op pagina 54
- "Communicatie in het bestandssysteem OS/400-bestandserver" op pagina 54
- "Beveiliging en objectmachtiging in het bestandssysteem OS/400-bestandserver" op pagina 55
- "Koppelingen in het bestandssysteem OS/400-bestandserver" op pagina 56
- "Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem OS/400-bestandserver" op pagina 56
- "API's van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem OS/400-bestandserver" op pagina 56

Hoofdlettergevoeligheid in het bestandssysteem OS/400-bestandsserver: Als directory van het eerste niveau die feitelijk de "hoofddirectory" (/) van het doelsysteem voorstelt, handhaaft het bestandssysteem QFileSvr.400 de hoofd- en kleine letters van de opgegeven objectnamen. Als QFileSvr.400 naar namen zoekt, wordt echter geen onderscheid gemaakt tussen hoofd- en kleine letters.

Voor alle andere directory's is de hoofdlettergevoeligheid afhankelijk van het specifieke bestandssysteem dat wordt geopend. QFileSvr.400 handhaaft de hoofd- en kleine letters die zijn opgegeven in de objectnamen als een opslagopdracht wordt verzonden naar de OS/400-bestandsserver.

Padnamen in het bestandssysteem OS/400-bestandsserver:

- Padnamen hebben de volgende vorm:

/QFileSvr.400/NaamLocatieOpAfstand/Directory/Directory . . . /Object

De directory van het eerste niveau (d.w.z. NaamLocatieOpAfstand in het voorbeeld hierboven) vertegenwoordigt het volgende:

- De naam van de doelserver die wordt gebruikt om een communicatieverbinding tot stand te brengen. De naam van de doelserver kan zijn:
 - Een TCP/IP-hostnaam, bijvoorbeeld beowulf.newyork.corp.com
 - Een SNA LU 6.2-naam, bijvoorbeeld appn.newyork
- De "hoofddirectory" (/) van de doelserver

Hierdoor worden, als een directory van het eerste niveau wordt gemaakt met behulp van een geïntegreerd bestandssysteem-interface, opgegeven kenmerken genegeerd.

Opmerking: Directory's van het eerste niveau blijven niet vast staan gedurende verschillende opstartprocedures. Directory's van het eerste niveau moeten dus na iedere opstartprocedure opnieuw worden gemaakt.

- Elk onderdeel van een padnaam kan bestaan uit maximaal 255 tekens. De volledige padnaam kan daardoor maximaal 16MB lang zijn.

Opmerking: Het bestandssysteem waarvan het object deel uitmaakt kan de componentlengte en padnaamlengte beperken tot minder dan het maximum dat is toegestaan in de QFileSvr.400.

- Er geldt geen maximale diepte van de directoryhiërarchie, behalve de grenzen van het programma en het systeem en eventuele maximumwaarden die gelden voor het bestandssysteem waartoe toegang wordt gezocht.
- De tekens in namen worden geconverteerd naar UCS2 Level 1 wanneer namen worden opgeslagen (zie "Naamcontinuïteit" op pagina 17).

Communicatie in het bestandssysteem OS/400-bestandsserver:

- TCP-verbindingen met de bestandsserver op een doelserver kunnen alleen tot stand worden gebracht als het QSERVER-subsysteem op de doelserver actief is.
- Er wordt alleen geprobeerd een SNA LU 6.2-verbinding te maken als er een lokaal bestuurd sessie bestaat die niet in gebruik is (bijvoorbeeld een sessie die specifiek tot stand is gebracht voor gebruik door de LU 6.2-verbinding. Bij het tot stand brengen van een LU 6.2-verbinding gebruikt het bestandssysteem QFileSvr.400 de werkstand BLANK. Op het doelsysteem wordt een taak genaamd QPWFSERV aangeboden aan het QSERVER-subsysteem. Het gebruikersprofiel van deze taak wordt gedefinieerd door het communicatie-item voor de werkstand BLANK. Raadpleeg voor meer informatie over LU 6.2-

communicatie de publicatie APPC Programming .

- Bestandsserveraanvragen die gebruik maken van TCP als communicatieprotocol worden uitgevoerd binnen de context van de taak die de aanvraag opgeeft. Bestandsserveraanvragen die gebruik maken van SNA als communicatieprotocol worden uitgevoerd door de systeemtaak OS/400.

- Als er nog geen verbinding tot stand is gebracht met de doelserver, neemt het bestandssysteem QFileSvr.400 aan dat de directory van het eerste niveau een TCP/IP-hostnaam voorstelt. Het bestandssysteem QFileSvr.400 doorloopt de volgende stappen om een verbinding met de doelserver tot stand te brengen:

1. De naam van de locatie op afstand omzetten naar een IP-adres.
2. Verbinding maken met de servertoewijzer van de hostserver op de welbekende poort 449 met behulp van het omgezette IP-adres. Vervolgens een query verzenden naar de servertoewijzer voor de servicenaam "as-file." Ten gevolge van de query gebeurt er een van de volgende dingen:
 - Als "as-file" zich in de servicetabel op de doelserver bevindt, zendt de servertoewijzer de poort terug waarop de OS/400 bestandsserverdaemon luistert.
 - Als de servertoewijzer niet actief is op de doelserver wordt het standaardpoortnummer voor "as-file" (8473) gebruikt.

Het bestandssysteem QFileSvr.400 probeert vervolgens een TCP-verbinding tot stand te brengen met de OS/400 bestandsserverdaemon op de doelserver. Wanneer verbinding tot stand is gebracht, wisselt QFileSvr.400 aanvragen en antwoorden uit met de bestandsserver. In het subsysteem QSERVER nemen de QPWFSEVSO-aanvragen de besturing van de verbinding over. Elke voorstarttaak draait onder zijn eigen gebruikersprofiel.

3. Als de lokatiennaam op afstand niet wordt omgezet naar een IP-adres wordt aangenomen dat de directory van het eerste niveau een SNA LU 6.2-naam is. Daarom wordt geprobeerd een APPC-verbinding tot stand te brengen met de OS/400-bestandsserver.
- Het bestandssysteem QFileSvr.400 controleert periodiek (iedere twee uur) of er niet-gebruikte verbindingen zijn (bijvoorbeeld geopende bestanden die bij de verbinding horen) en of die verbindingen inactief waren gedurende een periode van twee uur. Als een dergelijke verbinding wordt aangetroffen, wordt deze beëindigd.
 - Het bestandssysteem QFileSvr.400 kan geen lussen ontdekken. De volgende padnaam is een voorbeeld van een lus:

```
/QFileSvr.400/Remote2/QFileSvr.400/Remote1/QFileSvr.400/Remote2/...
```

waarbij Remote1 het lokale systeem is. Wanneer u een padnaam met een lus opgeeft, geeft het bestandssysteem QFileSvr.400 na korte tijd een foutbericht. De fout geeft aan dat er een timeout is opgetreden.

Het bestandssysteem QFileSvr.400 gebruikt een bestaande vrije sessie bij de communicatie over SNA. Om een verbinding tot stand te brengen met het communicatiesysteem op afstand is het noodzakelijk de werkstand te starten en een sessie voor QFileSvr.400 tot stand te brengen.

Beveiliging en objectmachtiging in het bestandssysteem OS/400-bestandsserver: Als voor beide systemen Kerberos is geconfigureerd en de gebruiker gemachtigd is voor Kerberos, dan kan Kerberos worden gebruikt om een bestandssysteem te verifiëren dat deel uitmaakt van de iSeries-server. Als de Kerberos-verificatie mislukt, kunnen het gebruikers-ID en -wachtwoord worden gebruikt om toegang te bevestigen.

Opmerking: Als het tickettoewijzende ticket of het serverticket vervalft nadat de doelserver uw toegang heeft bevestigd, vervalft het betreffende ticket pas als de verbinding met de doelserver is beëindigd. Raadpleeg voor meer informatie over Kerberos het onderwerp Netwerkverificatieservice in het iSeries Informatiecentrum

- Om een bestandssysteem te openen dat deel uitmaakt van een doel-iSeries-server moet u een gebruikers-ID en -wachtwoord hebben op de doelserver die overeenkomen met het ID en wachtwoord op de lokale server, als Kerberos niet wordt gebruikt voor de verificatie.

Opmerking: Als uw wachtwoord op de lokale of doelserver wordt gewijzigd nadat de doelserver uw toegang heeft bevestigd, dat wordt de wijziging pas uitgevoerd als de verbinding met de doelserver is beëindigd. Er is echter geen vertraging als uw gebruikersprofiel op de lokale server wordt gewist en een ander gebruikersprofiel wordt gemaakt met hetzelfde gebruikers-ID. In dat geval bevestigt het bestandssysteem QFileSvr.400 dat uw toegang hebt tot de doelserver.

- Objectmachtiging is gebaseerd op het gebruikersprofiel dat deel uitmaakt van de doelserver. Dat wil zeggen dat u alleen een object in het bestandssysteem op de doelserver kunt openen als uw gebruikersprofiel op de doelserver de juiste machtiging voor het object heeft.

Koppelingen in het bestandssysteem OS/400-bestandsserver: Het bestandssysteem QFileSvr.400 ondersteunt slechts één koppeling naar een object. Symbolische koppelingen kunnen niet worden gemaakt of opgeslagen in QFileSvr.400. Bestanden in QFileSvr.400 kunnen echter worden geopend met behulp van een symbolische koppeling vanaf de “hoofddirectory” (/), QOpenSys of door de gebruiker gedefinieerde bestandssystemen.

Raadpleeg “Koppeling” op pagina 11 voor een beschrijving van koppelingen.

Opdrachten en schermen van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem OS/400-bestandsserver: De opdrachten die worden afgebeeld in het onderwerp “Toegang met behulp van CL-opdrachten” op pagina 61 werken in het bestandssysteem QFileSvr.400, met uitzondering van de volgende:

```

ADDLNK
APYJRNCHG
CHGAUT
| CHGJRNOBJ
CHGOWN
DSPAUT
ENDJRN
RST
SAV
SNDJRNE
STRJRN
WRKOBJOWN
WRKOBJPGP

```

Dezelfde beperkingen zijn van toepassing op de gebruikersschermen beschreven in “Toegang met behulp van menu’s en schermen” op pagina 60.

API’s van het geïntegreerd bestandssysteem gebruiken in het bestandssysteem OS/400-bestandsserver: De API’s die worden afgebeeld in het onderwerp “Handelingen uitvoeren met behulp van API’s” op pagina 98 werken in het bestandssysteem QFileSvr.400, met uitzondering van de volgende:

```

chown()
fchown()
| fclear()
| fclear64()
givedescriptor()
link()
QjoEndJournal()
QjoRetrieveJournalInformation()
QJORJIDI()
QJOSJRNE
QjoStartJournal
Qp0lGetPathFromFileID()
symlink()

```

takedescriptor()

Netwerkbestandssysteem (NFS)

Het NFS biedt de gebruiker toegang tot gegevens en objecten die zijn opgeslagen op een NFS-server op afstand. Een NFS-server kan een netwerkbestandssysteem exporteren dat vervolgens door een NFS-client kan worden aangekoppeld.

Bovendien bezit elk bestandssysteem dat lokaal wordt aangekoppeld via het netwerkbestandssysteem de functies, kenmerken, beperkingen en dependency's van de directory of het bestandssysteem op de server op afstand waarvandaan het werd aangekoppeld. Bewerkingen aan een aangekoppeld bestandssystemen worden niet lokaal uitgevoerd. De opdrachten gaan via de verbinding naar de server en moeten voldoen aan de vereisten en beperkingen van het type bestandssysteem op de server.

Raadpleeg voor meer informatie over NFS NFS-bestandssystemen gebruiken via de IFS-interface.

NFS-bestandssystemen gebruiken via de IFS-interface

Het NFS (netwerkbestandssysteem) is toegankelijk via de IFS-interface en heeft de volgende kenmerken en beperkingen.

De volgende onderwerpen geven meer informatie over het NFS-bestandssysteem.

- "Kenmerken van het netwerkbestandssysteem"
- "Verschillende servers en clients op het netwerkbestandssysteem"
- "Koppelingen in het netwerkbestandssysteem" op pagina 58
- "Opdrachten van het geïntegreerd bestandssysteem gebruiken in het netwerkbestandssysteem" op pagina 58
- "API's van het geïntegreerd bestandssysteem gebruiken in het netwerkbestandssysteem" op pagina 59

Kenmerken van het netwerkbestandssysteem: De kenmerken van een bestandssysteem dat via het NFS wordt aangekoppeld zijn afhankelijk van het type bestandssysteem dat van de server werd aangekoppeld. U moet niet vergeten dat opdrachten die worden gegeven in een directory of bestandssysteem dat lokaal lijkt te zijn, in werkelijkheid via de NFS-verbinding op de server worden uitgevoerd.

Deze verhouding tussen client en server kan verwarrend zijn. U hebt bijvoorbeeld het bestandssysteem QDLS vanaf de server aangekoppeld boven een tak van de "hoofddirectory" (/) van uw client. Hoewel het aangekoppelde bestandssysteem een uitbreiding van de lokale directory lijkt te zijn, functioneert het in feite als het bestandssysteem QDLS.

Het is belangrijk bij het verwerken van opdrachten lokaal en via de serververbinding u bewust te zijn van de positie van bestandssystemen die via het NFS zijn aangekoppeld. Dat een opdracht op lokaal niveau goed wordt uitgevoerd, wil nog niet zeggen dat deze in een van de server aangekoppelde directory ook goed werkt. Elke directory die op de client wordt aangekoppeld, heeft de eigenschappen en kenmerken van het serverbestandssysteem.

Verschillende servers en clients op het netwerkbestandssysteem: Er zijn drie hoofdopties voor client/server-verbindingen die de functies en kenmerken van het netwerkbestandssysteem beïnvloeden:

1. De gebruiker koppelt een bestandssysteem aan op een client vanaf de iSeries-server.
2. De gebruiker koppelt een bestandssysteem aan op een client vanaf een UNIX-server.
3. De gebruiker koppelt een bestandssysteem aan op een client vanaf een andere server dan een iSeries- of UNIX-server.

In het eerste geval gedraagt het aangekoppelde bestandssysteem zich net zo op de client als op de iSeries-server. De kenmerken van zowel het netwerkbestandssysteem als het bestandssysteem dat wordt bediend, moeten echter in aanmerking worden genomen. Als u bijvoorbeeld op een client het bestandssysteem QDLS vanaf de server aankoppelt, heeft het de kenmerken en beperkingen van het bestands-

systeem QDLS. Padnaamcomponenten hebben in het bestandssysteem QDLS bijvoorbeeld een beperking van 8 tekens plus een extensie van drie tekens. Het aangekoppelde bestandssysteem heeft daarnaast echter ook de kenmerken en beperkingen van het NFS. U kunt bijvoorbeeld niet de opdracht CHGAUD gebruiken om de controlewaarde van een NFS-object te wijzigen.

In het tweede geval moet u in de gaten houden dat elk bestandssysteem dat wordt aangekoppeld van een UNIX-server zich vrijwel net zo zal gedragen als het bestandssysteem QOpenSys van de iSeries-server. U komt meer te weten over het QOpenSys-bestandssysteem in het onderwerp "Bestandssysteem open systemen (QOpenSys)" op pagina 30.


In het derde geval moet u zich verdiepen in de documentatie van het bestandssysteem dat bij het besturingssysteem van de server hoort.

Koppelingen in het netwerkbestandssysteem: Over het algemeen zijn meerdere vaste koppelingen naar hetzelfde object toegestaan in het netwerkbestandssysteem. Symbolische koppelingen worden volledig ondersteund. Een symbolische koppeling kan worden gebruikt om een koppeling te maken van een netwerkbestandssysteem naar een object in een ander bestandssysteem. De mogelijkheden voor meerdere vaste koppelingen en symbolische koppelingen zijn volledig afhankelijk van het bestandssysteem dat via NFS wordt aangekoppeld.

Raadpleeg "Koppeling" op pagina 11 voor een beschrijving van koppelingen.

Opdrachten van het geïntegreerd bestandssysteem gebruiken in het netwerkbestandssysteem: Alle opdrachten die worden afgebeeld in het onderwerp "Toegang met behulp van CL-opdrachten" op pagina 61 en de schermen die worden beschreven in het onderwerp "Toegang met behulp van menu's en schermen" op pagina 60 werken in het netwerkbestandssysteem, behalve de volgende:

- APYJRNCHG
- CHGJRNOBJ
- CHGAUD
- CHGATR
- CHGAUT
- CHGOWN
- CHGPGP
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRNE
- STRJRN

Er bestaan enkele CL-opdrachten die specifiek zijn voor het netwerkbestandssysteem en andere aangekoppelde bestandssystemen. Het is echter soms niet veilig om deze opdrachten te gebruiken in een multithread-proces. In de onderstaande tabel worden deze opdrachten behandeld. Voor een volledige beschrijving van de opdrachten en schermen die specifiek bij het netwerkbestandssysteem horen, raadpleegt u OS/400 NFS-ondersteuning .

Tabel 6. CL-opdrachten voor het netwerkbestandssysteem

Opdracht	Beschrijving
ADDMFS	Aangekoppeld bestandssysteem toevoegen. Hiermee plaatst u geëxporteerde serverbestandssystemen op afstand boven lokale clientdirectory's.
CHGNFSEXP	Netwerkbestandssysteem wijzigen. De functie Exporteren voegt directorystructuren toe aan of verwijdert ze uit de exporttabel van netwerkbestandssystemen die worden geëxporteerd naar NFS-clients.


Tabel 6. CL-opdrachten voor het netwerkbestandssysteem (vervolg)

Opdracht	Beschrijving
DSPMFSINF	Informatie over aangekoppeld bestandssysteem afbeelden. Hiermee beeldt u informatie af over een aangekoppeld bestandssysteem.
ENDNFSSVR	Bestandssysteem van netwerkserver beëindigen. Hiermee beëindigt u één of meer NFS-daemons op de server.
EXPORTFS	Een bestandssysteem exporteren. Hiermee voegt u directorystructuren toe aan of verwijdert u ze uit de exporttabel van netwerkbestandssystemen die worden geëxporteerd naar NFS-clients.
MOUNT	Een bestandssysteem aankoppelen. Hiermee plaatst u geëxporteerde serverbestandssystemen op afstand boven lokale clientdirectory's. Deze opdracht is een alias van de opdracht ADDMFS.
RLSIFSLCK	Vergrendelingen van het geïntegreerd bestandssysteem vrijgeven. Hiermee geeft u alle byte-range-vergrendelingen vrij van een NFS-client of een object.
RMVMFS	Aangekoppeld bestandssysteem wissen. Hiermee verwijdert u geëxporteerde serverbestandssystemen op afstand van de naamruimte van de lokale client.
STRNFSSVR	Server netwerkbestandssysteem starten. Hiermee start u één of alle NFS-daemons op de server.
UNMOUNT	Een bestandssysteem afkoppelen. Hiermee verwijdert u geëxporteerde serverbestandssystemen op afstand van de naamruimte van de lokale client. Deze opdracht is een alias van de opdracht RMVMFS.

Opmerking: Een netwerkbestandssysteem moet zijn aangekoppeld voordat u er opdrachten op kunt uitvoeren.

API's van het geïntegreerd bestandssysteem gebruiken in het netwerkbestandssysteem: Alle API's die worden afgebeeld in het onderwerp "Handelingen uitvoeren met behulp van API's" op pagina 98 werken in het netwerkbestandssysteem, behalve de volgende:

- QjoEndJournal()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()

Voor een volledige beschrijving van CL-opdrachten die specifiek verband houden met het netwerkbestandssysteem raadpleegt u OS/400 NFS-ondersteuning  .

Opmerking: Een netwerkbestandssysteem moet zijn aangekoppeld voordat u er opdrachten op kunt uitvoeren.

Toegang tot het geïntegreerd bestandssysteem

Alle gebruikersinterfaces, zoals menu's, opdrachten en schermen, die u gebruikt om te kunnen werken met de bibliotheken, objecten, databasebestanden, mappen en documenten op uw systeem, werken nog net zo als voor de introductie van het geïntegreerd bestandssysteem. De interfaces kunnen echter niet worden gebruikt om te werken met de stroombestanden, directory's en andere objecten die door het geïntegreerd bestandssysteem worden ondersteund.

Er bestaat een aparte set interfaces voor het geïntegreerd bestandssysteem. Deze interfaces kunnen worden gebruikt voor objecten in elk bestandssysteem dat u kunt openen via de geïntegreerd bestandssysteem-directory's.

U kunt interactief omgaan met de directory's en objecten van het geïntegreerd bestandssysteem vanaf uw server met behulp van menu's en schermen of door CL-opdrachten te gebruiken. Bovendien met behulp van API's gebruik maken van de stroombestanden, directory's en andere ondersteuning van het geïntegreerd bestandssysteem.

U kunt ook interactief werken met het geïntegreerd bestandssysteem via iSeries Navigator. Dit is een grafische gebruikersinterface die wordt gebruikt voor het beheer van uw server vanaf het bureaublad van Windows.

U kunt op diverse manieren interactief werken met het geïntegreerd bestandssysteem:

Toegang met behulp van menu's en schermen

U kunt met bestanden en andere objecten in het geïntegreerd bestandssysteem werken met behulp van een serie menu's en schermen die op uw server aanwezig zijn.

Toegang met behulp van CL-opdrachten

CL-opdrachten werken met bestanden en andere objecten in alle bestandssystemen die toegankelijk zijn via het geïntegreerd bestandssysteem.

Toegang met behulp van API's

De API's waarmee u kunt werken met directory's en stroombestanden van het geïntegreerd bestandssysteem hebben de vorm van C-taalfuncties.

Toegang met behulp van de iSeries Navigator

iSeries Navigator is de grafische gebruikersinterface voor het beheer van uw servers vanaf het bureaublad van Windows.

Toegang met behulp van iSeries NetServer

Met behulp van iSeries NetServer, kunnen Windows-gebruikers toegang krijgen tot gedeelde paden en uitvoerwachtrijen van OS/400.

Toegang met behulp van bestandsoverdrachtsprogramma's

Met bestandsoverdrachtprogramma's kunt u bestanden verplaatsen vanaf diverse verschillende bestandssystemen. Met een bestandsoverdrachtprogramma kunt u ook mappen en documenten verplaatsen die in het QDLS-bestandssysteem worden gevonden.

Toegang met behulp van een PC

Met een PC die aangesloten is op het iSeries-systeem kunt u communiceren met mappen en objecten die in het geïntegreerd bestandssysteem zijn opgeslagen. De mappen en objecten verschijnen alsof ze op uw PC zijn opgeslagen.

Toegang met behulp van menu's en schermen

U kunt met bestanden en andere objecten in het geïntegreerd bestandssysteem werken met behulp van een serie menu's en schermen die op uw server aanwezig zijn. U beeldt menu's van het geïntegreerd bestandssysteem als volgt af:

1. Meld u aan bij de server.
2. Druk op **Enter** om verder te gaan.
3. Kies **Bestandssystemen** in het iSeries-hoofdmenu.
4. Kies **IFS (Integrated File System)**.

Vanaf hier kunt u in het geïntegreerd bestandssysteem werken met directoryopdrachten, objectopdrachten en beveiligingsopdrachten. Als u weet welke CL-opdracht u wilt gebruiken, kunt u die ook op de opdrachtregel onder aan het scherm opgeven en op **Enter** drukken. Op die manier slaat u het optiemenu over.

Daarnaast kunt u toegang krijgen tot het geïntegreerd bestandssysteem vanuit elk menu op uw server door de volgende stappen uit te voeren:

1. Typ GO DATA op de opdrachtregel om het menu Bestanden, bibliotheken en folders af te beelden.
2. Kies Geïntegreerd bestandssysteem.

Typ op een opdrachtregel GO CMDNFS om een menu van NFS-opdrachten af te beelden. Typ op een opdrachtregel GO CMDUDFS om een menu met door de gebruiker gedefinieerde opdrachten voor het netwerkbestandssysteem af te beelden.

In de menu's van het geïntegreerd bestandssysteem kunt u schermen openen waarin de volgende handelingen kunt uitvoeren:

- Een directory maken, converteren en verwijderen
- De naam van de actieve directory afbeelden en wijzigen
- Objectkoppelingen toevoegen, afbeelden, wijzigen en verwijderen
- Objecten kopiëren, verplaatsen en hernoemen
- Objecten in- en uitchecken
- Objecten opslaan (reservekopie) en herstellen
- Objecteigenaren en gebruikersmachtigingen afbeelden en wijzigen
- Objectkenmerken afbeelden en wijzigen
- Gegevens kopiëren van en naar stroombestanden en databasebestandssecties
- De status van door de gebruiker gedefinieerde bestandssystemen maken, wissen en afbeelden
- Bestandssystemen exporteren vanaf een server
- Bestandssystemen aankoppelen en ontkoppelen op een client

Sommige bestandssystemen ondersteunen niet al deze bewerkingen. Raadpleeg voor de specifieke beperkingen van bestandssystemen het onderwerp Werken met bestandssystemen.

Raadpleeg het volgende onderwerp voor meer informatie over de menu's en schermen van het geïntegreerd bestandssysteem:

- Regels voor paden van CL-opdrachten en -schermen
- Toegang met behulp van CL-opdrachten

Toegang met behulp van CL-opdrachten

Alle bewerkingen die u met de menu's en schermen van het geïntegreerd bestandssysteem kunt uitvoeren (zie "Toegang met behulp van menu's en schermen" op pagina 60), kunt u ook uitvoeren door CL-opdrachten op te geven. Deze opdrachten werken met bestanden en andere objecten in een bestandssysteem die toegankelijk zijn via het IFS-interface.

Tabel 1 geeft een samenvatting van de opdrachten van het geïntegreerd bestandssysteem. Voor meer informatie over CL-opdrachten, het netwerkbestandssysteem en aangekoppelde bestandssystemen in het algemeen, raadpleegt u "Door de gebruiker gedefinieerd bestandssysteem (UDFS)" op pagina 32 en "Netwerkbestandssysteem (NFS)" op pagina 57. Voor het gemak van OS/2- en DOS-gebruikers bestaat voor opdrachten die dezelfde bewerking uitvoeren als een OS/2- of DOS-opdracht, een alias (een alternatieve opdracht naam).

Tabel 7. Opdrachten voor het geïntegreerd bestandssysteem

Opdracht	Beschrijving	Alias
ADDLNK	Koppeling toevoegen. Hiermee voegt u een koppeling tussen een directory en een object toe.	

Tabel 7. Opdrachten voor het geïntegreerd bestandssysteem (vervolg)

Opdracht	Beschrijving	Alias
ADDMFS	Aangekoppeld bestandssysteem toevoegen. Hiermee plaatst u geëxporteerde serverbestandssystemen op afstand boven lokale clientdirectory's.	MOUNT
APYJRNCHG ²	In journaal opgenomen wijzigingen toepassen. Hiermee past u aan de hand van journaalgegevens wijzigingen toe die zijn ingevoerd nadat een in het journaal opgenomen object werd opgeslagen, of past u wijzigingen tot een bepaald punt toe.	
CHGATR	Kenmerk wijzigen. Hiermee wijzigt u een kenmerk voor een enkel object, een groep objecten of een directorystructuur, waarbij van de directory en van de inhoud van de directory en alle subdirectory's het kenmerk wordt gewijzigd.	
CHGAUD	Auditwaarde wijzigen. Hiermee schakelt u voor een object het uitvoeren van een audit in of uit.	
CHGAUT	Machtiging wijzigen. Hiermee geeft u een gebruiker of een groep gebruikers een bepaalde machtiging voor een object.	
CHGCURDIR	Actieve directory wijzigen. Hiermee maakt u van een directory de actieve directory.	CD, CHDIR
CHGJRNOBJ ²	In het journaal vastgelegde objecten wijzigen. Hiermee wijzigt u de kenmerken voor het vastleggen van een object of lijst met objecten in een journaal zonder dat u het vastleggen daarvan hoeft te beëindigen en opnieuw te starten.	
CHGNFSEXP	Exporteren van netwerkbestandssysteem wijzigen. Hiermee voegt u directorystructuren toe of verwijdert u ze uit de exporttabel die naar NFS-clients wordt geëxporteerd.	EXPORTFS
CHGOWN	Eigenaar wijzigen. Hiermee draagt u het eigendom van een object over naar de andere gebruiker.	
CHGPGP	Primaire groep wijzigen. Hiermee draagt u de primaire groep over aan een andere gebruiker.	
CHKIN	Inchecken. Hiermee checkt u een object in dat eerder werd uitgecheckt.	
CHKOUT	Uitchecken. Hiermee checkt u een object uit, waardoor andere gebruikers het niet kunnen wijzigen.	
CPY	Kopiëren. Hiermee kopieert u één object of een groep objecten.	COPY
CPYFRMSTMF	Kopiëren van stroombestand. Hiermee kopieert u gegevens van een stroombestand naar een databasebestandssectie.	
CPYTOSTMF	Kopiëren naar stroombestand. Hiermee kopieert u gegevens van een databasebestandssectie naar een stroombestand.	
CRTDIR	Directory maken. Hiermee voegt u een nieuwe directory toe aan het systeem.	MD, MKDIR
CRTUDFS	UDFS maken. Hiermee maakt u een door de gebruiker gedefinieerd bestandssysteem (UDFS).	
CVTDIR	Directory converteren. Hiermee krijgt u informatie over het converteren van directory's van een geïntegreerd bestandssysteem van *TYPE1-formaat naar *TYPE2-formaat.	
CVTRPCSRC	RPC-bron converteren. Hiermee genereert u C-code vanuit een invoerbestand dat is geschreven in de taal Remote Procedure Call (RPC).	RPCGEN
DLTUDFS	UDFS wissen. Hiermee wist u een door de gebruiker gedefinieerd bestand.	

Tabel 7. Opdrachten voor het geïntegreerd bestandssysteem (vervolg)

Opdracht	Beschrijving	Alias
DSPAUT	Machtiging afbeelden. Hiermee beeldt u af welke gebruikers voor een bepaald object zijn gemachtigd en welke machtiging(en) ze voor dat object hebben.	
DSPCURDIR	Actieve directory afbeelden. Hiermee beeldt u de naam van de actieve directory af.	
DSPLNK	Objectkoppelingen afbeelden. Hiermee beeldt u een lijst met objecten in een directory af en kunt u informatie over die objecten afbeelden.	
DSPF	Stroombestand afbeelden. Hiermee beeldt u een stroombestand of een databasebestand af.	
DSPMFSINF	Informatie over aangekoppeld bestandssysteem afbeelden. Hiermee beeldt u informatie af over een aangekoppeld bestandssysteem.	STATFS
DSPUDFS	UDFS afbeelden. Hiermee beeldt u een door de gebruiker gedefinieerd bestandssysteem (UDFS) af.	
EDTF	Stroombestand bewerken. Hiermee bewerkt u een stroombestand of een databasebestand af.	
ENDJRN ²	Journal beëindigen. Hiermee beëindigt u het in een journal vastleggen van wijzigingen aan een object of een lijst met objecten.	
ENDNFSSVR	Bestandssysteem van netwerkserver beëindigen. Hiermee beëindigt u één of alle NFS-daemons op de server of de client.	
ENDRPCBIND	RPC-binderdaemon beëindigen. Hiermee beëindigt u de procedureaanroep op afstand (RPF) RPCBind daemon.	
MOV	Verplaatsen. Hiermee verplaatst u een object naar een andere directory.	MOVE
PRDIRINF	Directory-informatie afdrukken. Hiermee drukt u directory-informatie af van objecten in het geïntegreerd bestandssysteem die is verzameld met de opdracht RTVDIRINF (Directory-informatie ophalen).	
RLSIFSLCK	Vergrendelingen van het geïntegreerd bestandssysteem vrijgeven. Hiermee geeft u alle byte-range-vergrendelingen vrij van een NFS-client of een object.	
RMVDIR	Directory wissen. Hiermee wist u een directory van het systeem.	RD, RMDIR
RMVLNK	Koppeling verwijderen. Hiermee verwijdert u de koppeling naar een object.	DEL, ERASE
RMVMFS	Aangekoppeld bestandssysteem wissen. Hiermee verwijdert u geëxporteerde serverbestandssystemen op afstand van de lokale clientdirectory's.	UNMOUNT
RNM	Hernoemen. Hiermee wijzigt u de naam van een object in een directory	REN
RPCBIND	RPC-Binderdaemon starten. Hiermee start u de procedureaanroep op afstand (RPF) RPCBind Daemon.	
RST	Herstellen. Hiermee kopieert u een object of een groep objecten van een backupapparaat naar het systeem.	
RTVCURDIR	Actieve directory ophalen. Hiermee haalt u de naam van de actieve directory op en plaats u deze in een opgegeven variabele (gebruikt in CL-programma's).	

Tabel 7. Opdrachten voor het geïntegreerd bestandssysteem (vervolg)

Opdracht	Beschrijving	Alias
RTVDIRINF	Directory-informatie ophalen. Hiermee kunt u kenmerken verzamelen van objecten in het geïntegreerd bestandssysteem.	
SAV	Opslaan. Hiermee kopieert u een object of een groep objecten van het systeem naar een backupapparaat.	
SNDJRNE ²	Journalitem verzenden. Hiermee voegt u journalitems van de gebruiker, eventueel horend bij een in het journal opgenomen object, toe aan een journalontvanger.	
STRJRN ²	Journal starten. Hiermee start u het opnemen van wijzigingen (aan een object of lijst met objecten) in een bepaald journal.	
STRNFSSVR	Server netwerkbestandssysteem starten. Hiermee start u één of alle NFS-deemons op de server of de client.	
WRKAUT	Werken met machtigingen. Hiermee beeldt u een lijst van gebruikers en hun machtigingen af en krijgt u mogelijkheden om een gebruiker toe te voegen, gebruikersmachtigingen te wijzigen of een gebruiker te wissen.	
WRKLNK	Werken met objectkoppelingen. Hiermee beeldt u een lijst met objecten in een directory af en krijgt u mogelijkheden om acties uit te voeren op de objecten.	
WRKOBJOWN ¹	Werken met objecten per eigenaar. Hiermee beeldt u een lijst met objecten af die eigendom zijn van een gebruikersprofiel en krijgt u mogelijkheden om acties uit te voeren op de objecten.	
WRKOBJPGP ¹	Werken met objecten per primaire groep. Hiermee beeldt u een lijst met objecten af die worden bestuurd door een primaire groep en krijgt u mogelijkheden om acties uit te voeren op de objecten.	

Opmerking:

1. Met de opdrachten WRKOBJOWN en WRKOBJPGP kunt u alle objecttypen afbeelden, maar ze werken mogelijk niet volledig in alle bestandssystemen.
2. Zie Journalbeheer in het iSeries-Informatiecentrum voor meer informatie.

Raadpleeg de volgende onderwerpen voor meer informatie over CL-opdrachten voor geïntegreerde bestandssystemen en de beperkingen van deze opdrachten bij gebruik in bepaalde bestandssystemen:

- Werken met bestandssystemen
- Regels voor paden van CL-opdrachten en -schermen
- Werken met uitvoer van de opdrachten RTVDIRINF en PRTDIRINF
- Het onderwerp CL in het iSeries Informatiecentrum

Regels voor paden van CL-opdrachten en -schermen

Als u wilt werken met een object via een opdracht of scherm in het geïntegreerd bestandssysteem geeft u het object aan met zijn pad. Hieronder volgt een samenvatting van de regels waaraan u moet denken bij het opgeven van een pad. De term **object** verwijst bij deze regels naar een directory, bestand, koppeling of ander object.

- In een directory moet elke objectnaam uniek zijn.
- Het pad dat wordt opgegeven in een CL-opdracht van een geïntegreerd bestandssysteem moet vertegenwoordigd zijn in het CCSID dat op dat moment voor die taak actief is. Als het CCSID van de taak 65535 is, moet het pad vertegenwoordigd zijn in het standaard-CCSID van de taak. Omdat tekstreeksen doorgaans zijn gecodeerd in CCSID 37, moet u hardgecodeerde paden converteren naar het CCSID van de taak voordat u het pad opgeeft in de opdracht.

- Paden die worden opgegeven op een opdrachtregel, moeten tussen enkele aanhalingstekens staan ('). Deze tekens zijn niet verplicht als het pad wordt opgegeven in een scherm. Als het pad echter tekenreeksen tussen aanhalingstekens bevat, moeten ook de omsluitende tekens ' ' worden gebruikt.
- Paden worden van links naar rechts opgegeven, te beginnen met de directory van het hoogste niveau en ten slotte de naam van het object dat moet worden bewerkt. Alle componentnamen in het pad worden van elkaar gescheiden door een schuine streep naar rechts (/) of een schuine streep naar links (\), bijvoorbeeld:

```
'Dir1/Dir2/Dir3/GebrBestand'
```

of

```
'Dir1\Dir2\Dir3\GebrBestand'
```

- De tekens / en \ kunnen niet in de afzonderlijke componenten van het pad worden gebruikt, omdat ze als scheidingstekens worden gebruikt. Ook het cijfer 0 kan niet worden gebruikt. Kleine letters worden door de opdrachten niet gewijzigd in hoofdletters. Soms wordt de naam wel gewijzigd in hoofdletters, afhankelijk of het bestandssysteem met het object hoofdlettergevoelig is en of het object wordt gemaakt of gezocht.
- De lengte van een objectnaam is afhankelijk van het bestandssysteem waarin het object zich bevindt en van de maximumlengte die een opdrachtreeks mag hebben. De opdrachten kunnen objectnamen verwerken van maximaal 255 tekens en paden van maximaal 5000 tekens.
Raadpleeg Werken met bestandssystemen om te zien welke beperkingen elk bestandssysteem heeft ten aanzien van padnamen.
- Een / of \ aan het begin van een pad geeft aan dat het pad begint bij de bovenste directory, de "hoofddirectory" (/), bijvoorbeeld:

```
'/Dir1/Dir2/Dir3/GebrBestand'
```

- Als het pad niet met een / of \ begint, wordt aangenomen dat het pad begint in de actieve directory van de gebruiker die de opdracht opgeeft, bijvoorbeeld:

```
'MijnDir/MijnBestand'
```

, waarbij MyDir een subdirectory is van de actieve directory van de gebruiker.

- Een tilde (~) gevolgd door een schuine streep naar rechts of naar links aan het begin van een pad geeft aan dat het pad begint bij de homedirectory van de gebruiker die de opdracht opgeeft, bijvoorbeeld:

```
'~/GebrDir/GebrObj'
```

- Een tilde (~) gevolgd door een gebruikersnaam en vervolgens door een schuine streep naar rechts of naar links aan het begin van een pad geeft aan dat het pad begint bij de homedirectory van de gebruiker die wordt aangegeven met de gebruikersnaam, bijvoorbeeld:

```
'~user-name/GebrDir/GebrObj'
```

- Bij sommige opdrachten kan in de laatste component van een pad een sterretje (*) of een vraagteken (?) worden gebruikt om te zoeken naar padnaamtypes. Met het sterretje (*) laat u het systeem zoeken naar namen die een willekeurig aantal tekens op de plaats van het sterretje hebben staan. Met het vraagteken (?) laat u het systeem zoeken naar namen die één teken op de plaats van het vraagteken hebben staan. In het volgende voorbeeld zoekt het systeem naar objectnamen die beginnen met een *d* en eindigen met *txt*:

```
'/Dir1/Dir2/Dir3/d*txt'
```

In het volgende voorbeeld zoekt het systeem naar objectnamen die beginnen met een *d*, gevolgd door één willekeurig teken en eindigen met *txt*:

```
'/Dir1/Dir2/Dir3/d?txt'
```

- Om verwisseling met de speciale waarden van de iSeries-server te voorkomen, mogen paden niet met één sterretje (*) beginnen. Om een zoekreeks op te geven aan het begin van een pad gebruikt u twee sterretjes (**), bijvoorbeeld:

```
'**.file'
```

Opmerking: Dit slaat alleen op paden vanaf de gebruikersdirectory waar geen andere tekens voor het sterretje (*) staan.

- Wanneer u werkt met objecten in het QSYS.LIB-bestandssysteem moeten de componentnamen de vorm *naam.objecttype* hebben, bijvoorbeeld:

```
'/QSYS.LIB/PAY.LIB/TAX.BESTAND'
```

Zie "Bibliotheekbestandssysteem (QSYS.LIB)" op pagina 37 voor meer informatie.

- Wanneer u werkt met objecten in het bestandssysteem onafhankelijke ASP QSYS.LIB moeten de componentnamen de vorm *naam.objecttype* hebben, bijvoorbeeld:

```
'/asp_name/QSYS.LIB/PAYDAVE.LIB/PAY.FILE'
```

Zie "Onafhankelijke ASP QSYS.LIB" op pagina 40 voor meer informatie.

- Als een van de onderstaande tekens in een componentnaam wordt gebruikt, moet het pad worden omsloten door extra enkele (') of dubbele aanhalingstekens ("):
 - Sterretje (*)
 - Vraagteken (?)
 - Enkel aanhalingsteken (')
 - Dubbel aanhalingsteken (")
 - Tilde (~), indien deze wordt gebruikt als het eerste teken van de eerste component van het pad. Als een tilde op een andere plaats wordt gebruikt, wordt deze gezien als een gewoon teken.

Bijvoorbeeld:

```
"/Dir1/Dir/A*Smith"
```

of

```
'''/Dir1/Dir/A*Smith'''
```

Deze werkwijze wordt afgeraden omdat er een misverstand kan ontstaan over de betekenis van een teken in een opdrachtreeks en het waarschijnlijker wordt dat de opdrachtreeks verkeerd wordt opgegeven.

- Gebruik geen dubbele punt (:) in paden. De dubbele punt heeft in het systeem namelijk een bijzondere betekenis.
- De ondersteuning voor de verwerking van opdrachten en bijbehorende gebruikersschermen herkent codepunten lager dan hexadecimaal 40 niet als tekens die kunnen worden gebruikt in opdrachtreeksen en schermen. Als u deze codepunten wilt gebruiken, moet u ze invoeren in hexadecimale code, bijvoorbeeld:

```
crtmdir dir(x'02')
```

Daarom wordt het gebruik in paden van codepunten lager dan hexadecimaal 40 afgeraden. Deze beperking geldt alleen voor opdrachten en bijbehorende schermen, niet voor API's (zie "Toegang met behulp van API's" op pagina 80). Daarnaast is in padnamen de hexadecimale waarde 0 ook niet toegestaan.

Raadpleeg voor beperkingen ten aanzien van het gebruik van een bepaalde opdracht de Help bij opdrachten of het onderwerp Control language (CL) in het iSeries Informatiecentrum.

| **Werken met uitvoer van de opdrachten RTVDIRINF en PRTDIRINF**

| De opdracht RTVDIRINF wordt gebruikt om kenmerken te verzamelen van objecten in het geïntegreerd
| bestandssysteem. De verzamelde informatie wordt opgeslagen in databasebestanden (tabellen). Deze krij-
| gen een naam op basis van de prefix van het informatiebestand die is opgegeven met de parameter INF-
| FILEPFX. De tabellen worden gemaakt in de bibliotheek die is opgegeven met de parameter INFLIB. Voor
| meer informatie over bijvoorbeeld de naamgeving van databasebestanden en tabellen raadpleegt u
| RTVDIRINF.

| Drie tabellen worden gemaakt naar aanleiding van de opdracht RTVDIRINF. In een tabel worden de
| objectkenmerken opgeslagen, in een andere de directory's, terwijl de derde tabel wordt gebruikt om te
| bepalen welke bestanden gebruikt zijn om objectkenmerken op te slaan.

| “De gegevens van RTVDIRINF openen” op pagina 79 bevat informatie over hoe u toegang kunt krijgen
 | tot de gegevens in deze drie tabellen. Raadpleeg voor meer informatie over het gebruik van de gegevens
 | “Met de gegevens van RTVDIRINF” op pagina 80.

| Tabel 8 geeft een beschrijving van de velden die beschikbaar zijn voor de tabellen waarin de object-
 | kenmerken zijn opgeslagen. Als *GEN is opgegeven bij de parameter INFFILEPFX worden de database-
 | bestanden gemaakt met een unieke prefix die is gegenereerd door deze opdracht. De prefix begint met
 | QAEZD, gevolgd door vier cijfers. De bestanden die worden gemaakt om de verzamelde informatie op te
 | slaan, krijgen een naam met deze prefix, gevolgd door de letter 'D' (voor het bestand met de directory-in-
 | formatie) of de letter 'O' (voor het bestand dat de informatie over de objecten in de directory's bevat). De
 | eerste keer bijvoorbeeld dat de opdracht wordt uitgevoerd terwijl *GEN is opgegeven, worden de bestan-
 | den QAEZD0001D en QAEZD0001O gemaakt in de bibliotheek die is opgegeven met de parameter
 | INFLIB. De gebruiker kan een bestandsprefix opgeven voor gebruik bij de naamgeving van deze data-
 | base. Deze prefix mag maximaal uit negen tekens bestaan.

| *Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan)*

Veldnaam	Type veld	Veldbeschrijving
QEZDIRIDX	INTEGER	De index van de bovenliggende directory.
QEZOBJNAM ¹	VARGRAPHIC (1024)	De objectnaam. ²
QEZOBJLEN	INTEGER	Aantal bytes dat het objectnaamveld bevat.
QEZNMCCSID	INTEGER	Het CCSID dat een objectnaam vertegenwoordigt.
QEZREGION	GRAPHIC (2)	Een ID van twee tekens dat het land van de objectnaam vertegenwoordigt. (Heeft invloed op acties die doorgaans worden gedefinieerd door de locatie van de actie, zoals sorteervolgorde.)
QEZLANGID	GRAPHIC (3)	Een ID van drie tekens voor de taal van de objectnaam.
QEZMODE	INTEGER	De toegangswerkstand en het type van het bestand. Raadpleeg voor meer informatie over werkstanden de API Open bestand open().
QEZOBJTYPE ¹	GRAPHIC (10)	Het type object.
QEZCCSID	INTEGER	Het CCSID van de gegevens en de uitgebreide kenmerken van het object.
QEZALCSIZE ¹	BIGINT	Het aantal bytes dat is toegewezen aan dit object.
QEZDTASIZE	BIGINT	De grootte van de gegevens in dit object in bytes. Deze grootte is exclusief objectheaders en de uitgebreide kenmerken die bij het object horen.
QEZEAS	BIGINT	Het aantal uitgebreide kenmerken dat bij dit object hoort.
QEZCEAS	BIGINT	Het aantal essentiële uitgebreide kenmerken dat bij dit object hoort.
QEZEXTATRS	BIGINT	Het totaal aantal bytes voor de gegevens van alle uitgebreide kenmerken.
QEZCRTTIM	TIMESTAMP	De datum en tijd waarop het object is gemaakt.
QEZACCTIM	TIMESTAMP	De datum en tijd waarop de objectgegevens voor het laatst werden geopend.
QEZCHGTIMA ¹	TIMESTAMP	De datum en tijd waarop de objectkenmerken voor het laatst werden gewijzigd.
QEZCHGTIMD	TIMESTAMP	De datum en tijd waarop de objectgegevens voor het laatst werden gewijzigd.
QEZSTGFREE ¹	SMALLINT	Of de objectgegevens offline gezet zijn, waardoor de online opslagruimte ervan wordt vrijgemaakt. Geldige waarden zijn: 0 - De objectgegevens zijn niet offline. 1 - De objectgegevens zijn offline.

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZCHKOUT ¹	SMALLINT	Een indicator om aan te geven of een object is uitgecheckt. Geldige waarden zijn: 0 - Het object is niet uitgecheckt. 1 - Het object is uitgecheckt.
QEZCHKOWN	GRAPHIC (10)	De gebruiker die het object heeft uitgecheckt. Dit veld is leeg als het object niet is uitgecheckt.
QEZCHKTIM	TIMESTAMP	De datum en tijd waarop het object is uitgecheckt. Als het object niet is uitgecheckt, bevat dit veld geen gegevens.
QEZLOCAL	SMALLINT	Of een object lokaal is opgeslagen of op een systeem op afstand. De beslissing of een object lokaal is of op afstand hangt af van de respectievelijke bestandssysteemregels. Objecten in bestandssystemen die geen indicator voor lokaal of op afstand hebben, worden behandeld als waren ze op afstand. Geldige waarden zijn: 1 - De objectgegevens zijn lokaal opgeslagen. 2 - De objectgegevens zijn op een systeem op afstand opgeslagen.
QEZOWN ¹	GRAPHIC (10)	De naam van het gebruikersprofiel dat de eigenaar is van het object of van de volgende bijzondere waarde: *NOUSRPRF - Deze bijzondere waarde wordt gebruikt door het netwerkbestandssysteem om aan te geven dat er op de lokale iSeries-server geen gebruikersprofiel bestaat met een gebruikers-ID dat overeenkomt met het gebruikers-ID van het object op afstand.
QEZUID	INTEGER	Elke gebruiker op het systeem moet een uniek gebruikers-ID (UID) hebben.
QEZOWNPGP	GRAPHIC (10)	De naam van gebruikersprofiel dat de primaire groep van een object of volgende bijzondere waarden is: *NONE - Het object heeft geen primaire groep *NOUSRPRF - Deze bijzondere waarde wordt gebruikt door het netwerkbestandssysteem om aan te geven dat er op de lokale server geen gebruikersprofiel bestaat met een groeps-ID dat overeenkomt met het groeps-ID van het object op afstand.
QEZGID	INTEGER	Groepsprofielen worden geïdentificeerd met een uniek groeps-ID (GID).
QEZAUTLST	GRAPHIC (10)	De naam van de lijst van machtigingen die wordt gebruikt voor de beveiliging van het benoemde object. De waarde *NONE geeft aan dat geen lijst van machtigingen is gebruikt om de machtiging voor het object vast te stellen.
QEZASP	SMALLINT	De hulpgeheugenpool (ASP) waarin het object is opgeslagen.
QEZJRNSTS ¹	SMALLINT	De actuele journaalstatus van het object. Dit veld heeft een van de volgende waarden: 0 (NOT_JOURNALED) - Op dit moment wordt dit object niet opgenomen in een journaal. 1 (JOURNALED) - Op dit moment wordt dit object opgenomen in een journaal.

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZJSUBTRE	SMALLINT	<p>Wanneer deze vlag wordt teruggezonden, is dit object een directory met een substructuursemantiek voor journaals in het geïntegreerd bestandssysteem.</p> <p>0 - Dit object wordt niet opgenomen in een journaal met substructuursemantiek.</p> <p>1 - Dit object wordt opgenomen in een journaal met substructuursemantiek. Nieuwe objecten die in de substructuur van deze directory worden gemaakt, nemen de journaalkenmerken en -opties van deze directory over.</p>
QEZJOPTENT	SMALLINT	<p>Wanneer journaals actief zijn, worden items die worden beschouwd als optioneel in een journaal opgenomen. De lijst met optionele journaalitems verschilt voor elk type object.</p> <p>0 - Object is niet opgenomen in journaal met optionele items.</p> <p>1 - Object is opgenomen in journaal met optionele items.</p>
QEZJAFTERI	SMALLINT	<p>Wanneer journaals actief zijn, wordt de afbeelding van een object na een wijziging opgenomen in het journaal.</p> <p>0 - Object is niet in journaal opgenomen met beelden na wijziging.</p> <p>1 - Object is in journaal opgenomen met beelden na wijziging.</p>
QEZJBEFORI	SMALLINT	<p>Wanneer journaals actief zijn, wordt de afbeelding van een object voor een wijziging opgenomen in het journaal.</p> <p>0 - Object is niet opgenomen in journaal met beelden voor wijziging.</p> <p>1 - Object is opgenomen in journaal met beelden voor wijziging.</p>
QEZJRNID	GRAPHIC (10)	<p>Dit veld geeft aan een object dat wordt opgenomen in een journaal een ID dat kan worden gebruikt in verschillende opdrachten en API's die betrekking hebben op journaals.</p>
QEZJRNNAM	GRAPHIC (10)	<p>Als de waarde van de journaalstatus JOURNALED is, bevat dit veld de naam van het journaal dat op dit moment wordt gebruikt. Als de waarde van de journaalstatus NOT_JOURNALED is, bevat dit veld de naam van het journaal dat het laatst werd gebruikt voor dit object. Alle bytes in dit veld worden gezet op binair nul als dit object nog nooit in een journaal is opgenomen.</p>
QEZJRNLIB	GRAPHIC (10)	<p>Als de waarde van de journaalstatus JOURNALED is, bevat dit veld de naam van de bibliotheek die het journaal bevat dat op dit moment wordt gebruikt. Als de waarde van de journaalstatus NOT_JOURNALED is, bevat dit veld de naam van de bibliotheek die het journaal bevat dat het laatst werd gebruikt voor dit object. Alle bytes in dit veld worden gezet op binair nul als dit object nog nooit in een journaal is opgenomen.</p>
QEZJRNSTR	TIMESTAMP	<p>Het aantal seconden sinds de Epoch, die overeenkomt met de meest recente datum en tijd waarop een journaal voor het object is gestart. Dit veld wordt op binair nul gezet als dit object nog nooit in een journaal is opgenomen.</p>

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZAUDT	GRAPHIC (10)	De auditwaarde die bij het object hoort. Geldige waarden zijn: *NONE - Er vindt geen audit plaats voor dit object wanneer het wordt gelezen of gewijzigd, ongeacht de gebruiker die het object opent. *USRPRF - Voer een audit uit op dit object alleen als dat voor de actuele gebruiker wordt gedaan. De actuele gebruiker wordt onderzocht om te kijken of er op dit object een audit moet worden uitgevoerd. Het gebruikersprofiel kan opgeven of er alleen een audit op dit object moet worden uitgevoerd bij een wijzigingspoging of ook bij een leesopgiving. *CHANGE - Op iedere wijzigingspoging voor dit object door alle gebruikers op het systeem een audit uitvoeren. *ALL - Op iedere toegangspoging voor dit object door alle gebruikers op het systeem een audit uitvoeren. Alle toegangspogingen worden gedefinieerd als een lees- of een wijzigingspoging.
QEZBLKSIZ	INTEGER	De blok grootte van een object.
QEZNLNK	INTEGER	Het aantal vaste koppelingen naar een object.
QEZFILEID ¹	GRAPHIC (16)	Het bestands-ID van een object. Een ID dat bij het object hoort. Een bestands-ID kan met Qp0lGetPathFromFileID() worden gebruikt om de padnaam van een object op te halen.
QEZFILEIDS	INTEGER	Het 4-bytes bestands-ID van het bestand. Dit getal is het unieke ID van het object binnen een bestandssysteem. De getal kan het object niet in het hele systeem identificeren.
QEZGENID	BIGINT	Het generatie-ID dat bij het bestands-ID hoort.
QEZFSID	BIGINT	Het bestands-ID waarbij het object hoort. Dit getal is het unieke ID van het bestandssysteem waarbij het object hoort.
QEZRDEV	BIGINT	Als het object een speciaal apparaatbestand voorstelt: het echte apparaat dat het voorstelt.
QEZDOM	GRAPHIC (10)	Het domein van het object. Geldige waarden zijn: *SYSTEM - Object bestaat in systeemdomein. *USER - Object bestaat in gebruikersdomein.
QEZCRTAUD	GRAPHIC (10)	De auditwaarde die bij een object hoort dat in deze directory is gemaakt. Geldige waarden zijn: *NONE - Er vindt geen audit plaats voor dit object wanneer het wordt gelezen of gewijzigd, ongeacht de gebruiker die het object opent. *USRPRF - Voer een audit uit op dit object alleen als dat voor de actuele gebruiker wordt gedaan. De actuele gebruiker wordt onderzocht om te kijken of er op dit object een audit moet worden uitgevoerd. Het gebruikersprofiel kan opgeven of er alleen een audit op dit object moet worden uitgevoerd bij een wijzigingspoging of ook bij een leesopgiving. *CHANGE - Op iedere wijzigingspoging voor dit object door alle gebruikers op het systeem een audit uitvoeren. *ALL - Op iedere toegangspoging voor dit object door alle gebruikers op het systeem een audit uitvoeren. Alle toegangspogingen worden gedefinieerd als een lees- of een wijzigingspoging.

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZSCN	GRAPHIC (1)	<p>Of objecten moeten worden gescand door exitprogramma's die zijn geregistreerd bij een van de exit points van het geïntegreerd bestandssysteem die betrekking hebben op scans. Voor meer informatie raadpleegt u "Ondersteuning van scans" op pagina 18.</p> <p>Geldige waarden zijn:</p> <p>'x40' (SCANNING_NO) - Het object wordt niet gescand volgens de regels die zijn beschreven in de exitprogramma's die betrekking hebben op scannen. Opmerking: Als de waarde *NOPOSTRST van QSCANFCTL (Bestandssysteembesturing scannen) niet is opgegeven wanneer een object met dit kenmerk wordt hersteld, wordt het object na het herstellen minstens één keer gescand.</p> <p>'x80' (SCANNING_YES) - Het object wordt gescand volgens de regels die zijn beschreven in de exitprogramma's die betrekking hebben op scannen voor het geval het object is gewijzigd of de scansoftware is bijgewerkt sinds de laatste keer dat het object werd gescand.</p> <p>'x20' (SCANNING_CHGONLY) - Het object wordt alleen gescand volgens de regels die zijn beschreven in de exitprogramma's die betrekking hebben op scannen, als het object is gewijzigd sinds de laatste keer dat het object werd gescand. Het wordt niet gescand als de scansoftware wordt bijgewerkt. Dit kenmerk wordt alleen actief als voor de systeemwaarde QSCANFCTL (Bestandssysteembesturing scannen) *USEOCOATR is opgegeven. Anders wordt het behandeld alsof het kenmerk SCANNING_YES is. Opmerking: Als de waarde *NOPOSTRST van QSCANFCTL (Bestandssysteembesturing scannen) niet is opgegeven wanneer een object met dit kenmerk wordt hersteld, wordt het object na het herstellen minstens één keer gescand.</p>

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZINHSCN	GRAPHIC (1)	<p>Of objecten die zijn gemaakt in een directory moeten worden gescand door exitprogramma's die zijn geregistreerd bij een van de exit points van het geïntegreerd bestandssysteem die betrekking hebben op scans. Voor meer informatie raadpleegt u "Ondersteuning van scans" op pagina 18.</p> <p>Geldige waarden zijn:</p> <p>x'40' - Nadat een object in de directory is gemaakt, wordt het niet gescand volgens de regels die zijn beschreven in de exitprogramma's die betrekking hebben op scannen. Opmerking: Als de waarde *NOPOSTRST van QSCANFCTL (Bestandssysteembesturing scannen) niet is opgegeven wanneer een object met dit kenmerk wordt hersteld, wordt het object na het herstellen minstens één keer gescand.</p> <p>x'80' - Nadat een object in de directory is gemaakt, wordt het gescand volgens de regels die zijn beschreven in de exitprogramma's die betrekking hebben op scannen voor het geval het object is gewijzigd of de scansoftware is bijgewerkt sinds de laatste keer dat het object werd gescand.</p> <p>x'20' - Nadat een object in de directory is gemaakt, wordt het alleen gescand volgens de regels die zijn beschreven in de exitprogramma's die betrekking hebben op scannen voor het geval het object is gewijzigd sinds de laatste keer dat het object werd gescand. Het wordt niet gescand als de systeemwaarde QSCANFCTL (Bestandssysteembesturing scannen) *USEOCOATR is opgegeven. Anders wordt het behandeld alsof het kenmerk SCANNING_YES is. Opmerking: Als de waarde *NOPOSTRST van QSCANFCTL (Bestandssysteembesturing scannen) niet is opgegeven wanneer een object met dit kenmerk wordt hersteld, wordt het object na het herstellen minstens één keer gescand.</p>

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZSTATUS	GRAPHIC (1)	<p>De scanstatus die bij dit object hoort. Dit veld heeft een van de volgende waarden:</p> <p>x'00' (SCAN_REQUIRED) - Voor het object is een scan vereist, omdat het nog niet gescand is door de scanexitprogramma's of omdat de objectgegevens of het CCSID sinds de laatste scan is gewijzigd. Voorbeelden van wijzigingen aan objectgegevens of het CCSID zijn: schrijven naar het object (rechtstreeks of via geheugentoe wijzing), afkappen van het object, wissen van het object en wijzigen van het CCSID-kenmerk van het object.</p> <p>x'01' (SCAN_SUCCESS) - Het object is gescand door een scanexitprogramma en het is bij de laatste scanopdracht goed door de scan gekomen.</p> <p>x'02' (SCAN_FAILURE) - Het object is gescand door een scanexitprogramma en het is bij de laatste scanopdracht niet goed door de scan gekomen en de bewerking is niet voltooid. Wanneer een object als mislukt is gemarkeerd, wordt het pas weer gescand als de scanhandtekening van het object anders is dan de algemene sleutelhandtekening of de sleutelhandtekening van de onafhankelijk-ASP-groep, afhankelijk van het desbetreffende object. Latere verzoeken om met het object te werken zullen daarom mislukken en resulteren in een scanfoutmelding. Verzoeken die zullen mislukken zijn bijvoorbeeld openen van het object, wijzigen van het object-CCSID, kopiëren van het object.</p> <p>x'05' (SCAN_PENDING_CVN) - Het object bevindt zich niet in een *TYPE2-directory's en wordt derhalve niet gescand totdat de directory is geconverteerd.</p> <p>x'06' (SCAN_NOT_REQUIRED) - Het object hoeft te worden gescand omdat het is gemarkeerd om niet te worden gescand.</p>
QEZSSIGDF	GRAPHIC (1)	<p>De scanhandtekeningen geven een indicatie van het niveau van de scansoftwareondersteuning.</p> <p>Wanneer een object zich in een onafhankelijke-ASP-groep bevindt, wordt de scanhandtekening van het object vergeleken met de bijbehorende scanhandtekening van de onafhankelijke-ASP-groep. Wanneer een object zich niet in een onafhankelijke-ASP-groep bevindt, wordt de scanhandtekening van het object vergeleken met de waarde van de algemene scanhandtekening. Dit veld heeft een van de volgende waarden:</p> <p>x'00' - De vergeleken handtekeningen verschillen niet.</p> <p>x'01' - De vergeleken handtekeningen verschillen.</p>
QEZSBINARY	GRAPHIC (1)	<p>Geeft aan of het object, toen het eerder werd gescand, in de binaire werkstand werd gescand. Dit veld heeft een van de volgende waarden:</p> <p>x'00' - Het object werd niet in de binaire werkstand gescand.</p> <p>x'01' - Het object werd in de binaire werkstand gescand. Als de scanstatus van het object SCAN_SUCCESS is, werd het object met succes in de binaire werkstand gescand. Als de scanstatus van het object SCAN_FAILURE is, werd het object niet in de binaire werkstand gescand.</p>
QEZSCCSID1	INTEGER	<p>Geeft aan of het object, toen het eerder werd gescand, in het afgebeelde CCSID werd gescand. Als de scanstatus van het object SCAN_SUCCESS is, werd het object met succes in dit CCSID gescand. Als de scanstatus van het object SCAN_FAILURE is, werd het object niet in dit CCSID gescand. De waarde 0 houdt in dat dit veld niet van toepassing is.</p>

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZSCCSID2	INTEGER	Geeft aan of het object, toen het eerder werd gescand, in het afgebeelde CCSID werd gescand. Als de scanstatus van het object SCAN_SUCCESS is, werd het object met succes in dit CCSID gescand. Als de scanstatus van het object SCAN_FAILURE is, dan is de waarde van dit veld 0. De waarde 0 houdt in dat dit veld niet van toepassing is.
QEZUDATE	TIMESTAMP	Het aantal seconden sinds de Epoch die overeenkomt met de datum waarop het object voor het laatst werd gebruikt. De waarde van dit veld is 0 als het object is gemaakt. Als voor het OS/400-type of het bestandssysteem waartoe een object behoort geen gebruiksgegevens worden beheerd, is dit veld 0.
QEZUDCOUNT	INTEGER	Het aantal dagen dat een object is gebruikt. Gebruik heeft al naar het gelang het bestandssysteem en de afzonderlijke objecttypen die binnen een bestandssysteem worden ondersteund een andere betekenis. Gebruik kan het openen of sluiten van een bestand betekenen, maar het kan ook slaan op het toevoegen van koppelingen of hernoemen, herstellen en uitchecken van een object. Dit getal wordt groter per dag dat een object wordt gebruikt. U kunt het weer op nul zetten met de API Qp0lSetAttr().
QEZURESET	INTEGER	Het aantal seconden sinds de Epoch die overeenkomt met de datum waarop het aantal gebruikte dagen het laatst op nul (0) teruggezet is. Deze datum is ingesteld op de huidige datum als de API Qp0lSetAttr() wordt aangeroepen om het aantal gebruikte dagen op nul te zetten.
QEZPRMLNK	SMALLINT	Als een object verschillende namen heeft, wordt dit veld alleen ingesteld voor de eerste naam die wordt gevonden.
QEZALWCKPW	SMALLINT	Of een stroombestand (*STMF) met leesprogramma's en schrijfprogramma's kan worden gedeeld tijdens de controlepuntverwerking van de opslagbewerking met actieve objecten. Geldige waarden zijn: 0 - Het object kan alleen met leesprogramma's worden gedeeld. 1 - Het object kan met leesprogramma's en schrijfprogramma's worden gedeeld.
QEZSIG ¹	SMALLINT	Of een object een OS/400 digitale handtekening heeft. Geldige waarden zijn: 0 - Het object heeft geen OS/400 digitale handtekening. 1 - Het object heeft een OS/400 digitale handtekening.
QEZSYSSIG	SMALLINT	Of het object van een handtekening werd voorzien door een bron die het systeem betrouwbaar vindt. Geldige waarden zijn: 0 - Geen van de handtekeningen is afkomstig van een bron die het systeem betrouwbaar vindt. 1 - Het object werd van een handtekening voorzien door een bron die het systeem betrouwbaar vindt. Als het object meer dan één handtekening heeft, is er minstens één afkomstig van een bron die het systeem betrouwbaar vindt.
QEZMLTSIG	SMALLINT	Of een object meer dan één OS/400 digitale handtekening heeft. Geldige waarden zijn: 0 - Het object heeft slechts één digitale handtekening. 1 - Het object heeft meer dan één digitale handtekening. Als het veld QEZSYSSIG de waarde 1 heeft, is minstens één handtekening afkomstig van een bron die het systeem betrouwbaar vindt.

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZDSTGOPT	SMALLINT	<p>Deze optie dient u te gebruiken om vast te stellen hoe voor een opgegeven object hulpgeheugen wordt toegewezen door het systeem. Deze optie kan alleen worden opgegeven voor stroombestanden in de bestandssystemen hoofddirectory (/), QOpenSys en door de gebruiker gedefinieerde bestandssystemen. Deze optie is niet beschikbaar voor *TYPE1-bytestroombestanden. Geldige waarden zijn:</p> <p>0 - Het hulpgeheugen wordt normaal toegewezen. Dat wil zeggen dat wanneer extra hulpgeheugen nodig is, dit wordt toegewezen in vaste gebieden met een logische grootte zodat aan de huidige en toekomstige behoeften wordt voldaan, terwijl het aantal schijf-I/O-bewerkingen tot een minimum wordt beperkt.</p> <p>1 - Het hulpgeheugen wordt toegewezen om de geheugenruimte die door het object wordt gebruikt tot een minimum te beperken. Dat wil zeggen dat wanneer extra hulpgeheugen nodig is, dit wordt toegewezen in kleine vaste gebieden zodat aan de huidige behoeften aan geheugenruimte wordt voldaan. Door een object te openen dat is samengesteld uit veel kleine vaste gebieden, kan het aantal schijf-I/O-bewerkingen voor dat object toenemen.</p> <p>2 - Het systeem bepaalt de optimale hulpgeugentoeewijzing voor het object dynamisch en brengt daarbij een evenwicht aan tussen het gebruikte geheugen en de schijf-I/O-bewerkingen. Als een bestand bijvoorbeeld kleine vaste gebieden heeft en toch vaak wordt gelezen en bewaard, wordt het hulpgeheugen in de toekomst toegewezen in grotere vaste gebieden, om het aantal schijf-I/O-bewerkingen tot een minimum te beperken. Als een bestand daarentegen vaak wordt afgekapt, wordt het hulpgeheugen in de toekomst toegewezen in kleinere vaste gebieden om de geheugenruimte die wordt gebruikt tot een minimum te beperken. Daarnaast wordt informatie behouden over de groottes van stroombestanden voor dit systeem en zijn activiteiten. Deze bestandsgrootte-informatie wordt ook gebruikt om te helpen bepalen wat de optimale hulpgeugentoeewijzing is voor dit object in verhouding tot de grootte van andere objecten.</p>

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZMSTGOPT	SMALLINT	<p>Deze optie dient u te gebruiken om vast te stellen hoe voor het opgegeven object hoofdgeheugen wordt toegewezen en gebruikt door het systeem. Deze optie kan alleen worden opgegeven voor stroombestanden in de bestandssystemen hoofddirectory (/), QOpenSys en door de gebruiker gedefinieerde bestandssystemen. Geldige waarden zijn:</p> <p>0 - Het hoofdgeheugen wordt normaal toegewezen. Dat wil zeggen dat zoveel mogelijk hoofdgeheugen wordt toegewezen en gebruikt. Dit beperkt het aantal schijf-I/O-bewerkingen tot een minimum aangezien de gegevens worden gecached in het hoofdgeheugen.</p> <p>1 - Het hoofdgeheugen wordt toegewezen om de geheugenruimte die door het object wordt gebruikt tot een minimum te beperken. Dat wil zeggen dat zo min mogelijk hoofdgeheugen wordt toegewezen en gebruikt. Dit beperkt het gebruik van het hoofdgeheugen tot een minimum, terwijl het aantal schijf-I/O-bewerkingen toeneemt aangezien minder gegevens worden gecached in het hoofdgeheugen.</p> <p>2 - Het systeem bepaalt de optimale hoofdgeheugentoeewijzing voor het object dynamisch, afhankelijk van de andere systeemactiviteiten en het gebruik van het hoofdgeheugen. Dat wil zeggen dat als er weinig vraag naar hoofdgeheugen is, zoveel mogelijk geheugen wordt toegewezen en gebruikt om het aantal schijf-I/O-bewerkingen tot een minimum te beperken. En als er veel vraag is naar hoofdgeheugen wordt minder hoofdgeheugen toegewezen en gebruikt om conflicten rond het hoofdgeheugen tot een minimum te beperken. Deze optie sorteert alleen effect als de pool-optie van de pool *CALC is. Als de pagingoptie van de pool *FIXED is, is het gedrag gelijk aan STG_NORMAL. Wanneer het object wordt geopend via een bestandserver heeft deze optie geen invloed. In plaats daarvan is het gedrag gelijk aan STG_NORMAL.</p>
QEZDIRTYP2	SMALLINT	<p>De indeling van het opgegeven directory-object. Geldige waarden zijn:</p> <p>0 - De directory-indeling is *TYPE1.</p> <p>1 - De directory-indeling is *TYPE2.</p> <p>Voor meer informatie raadpleegt u “*TYPE2-directory’s” op pagina 10.</p>
QEZFILTYP2 ¹	SMALLINT	<p>De indeling van het stroombestand (*STMF). Geldige waarden zijn:</p> <p>0 - De indeling van het stroombestand is *TYPE1.</p> <p>1 - De indeling van het stroombestand is *TYPE2.</p> <p>Voor meer informatie raadpleegt u “Stroombestand” op pagina 15.</p>
QEZUDFTYP2	SMALLINT	<p>De standaardbestandsindeling van stroombestanden (*STMF) gemaakt in het door de gebruiker gedefinieerde bestandssysteem. Geldige waarden zijn:</p> <p>0 - De indeling van het stroombestand is *TYPE1.</p> <p>1 - De indeling van het stroombestand is *TYPE2.</p> <p>Voor meer informatie raadpleegt u “Stroombestand” op pagina 15.</p>

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZNONSAV	SMALLINT	Of het object kan worden opgeslagen of niet. Geldige waarden zijn: 0 - Object wordt opgeslagen. 1 - Object wordt niet opgeslagen. Daarnaast wordt, als dit object een directory is, geen van de objecten in de substructuur van de directory opgeslagen, tenzij expliciet is opgegeven dat ze moesten worden opgeslagen. De substructuur bevat alle subdirectory's en de objecten in die subdirectory's.
QEZCLSTRSP	SMALLINT	Het object is het geheugen dat is toegewezen aan xSeries-servers om te gebruiken als virtuele schijven voor de xSeries-servers. Gezien vanuit het perspectief van de iSeries-server verschijnen virtuele schijven als bestandsbestanden binnen het geïntegreerd bestandssysteem. 0 - Object is geen virtuele schijfgeheugen. 1 - Object is virtuele schijfgeheugen.
QEZCASE	SMALLINT	Geeft de hoofdlettergevoeligheid aan van het bestandssysteem dat dit object bevat. 0 - Bestandssysteem is niet hoofdlettergevoelig. 1 - Bestandssysteem is hoofdlettergevoelig.
QEZOFLOW	SMALLINT	Geeft aan of het object de hulpgeheugenpool (ASP) waarin het zich bevindt heeft doen overlopen. Geldige waarden zijn: 0 - ASP is niet overgelopen. 1 - ASP is overgelopen.
QEZPCREAD	SMALLINT	Of naar het object kan worden geschreven of het object kan worden gewist, of zijn uitgebreide kenmerken kunnen worden gewijzigd of gewist, of zijn grootte kan worden gewijzigd. Geldige waarden zijn: 0 - Het object kan worden gewijzigd. 1 - Het object kan niet worden gewijzigd.
QEZPCHID ¹	SMALLINT	Of het object kan worden afgebeeld met een gewone directorylijst. 0 - Het object is niet verborgen. 1 - Het object is verborgen.
QEZPCSYS	SMALLINT	Of het object een systeembestand is en uitgesloten is van standaard zoekacties in de directorystructuur. 0 - Het object is geen systeembestand. 1 - Het object is een systeembestand.
QEZPCARC	SMALLINT	Of het object is gewijzigd sinds het voor het laatst werd onderzocht. 0 - Het object is niet gewijzigd. 1 - Het object is gewijzigd.

Tabel 8. QAEZDxxxxO (Kenmerken van object opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZSYSARC	SMALLINT	Of het object is gewijzigd en moet worden opgeslagen. Dit wordt ingeschakeld als de wijzigingstijd van een object wordt bijgewerkt en uitgeschakeld wanneer het object is opgeslagen. 0 - Het object is niet gewijzigd en hoeft niet te worden opgeslagen. 1 - Het object is gewijzigd en moet worden opgeslagen.
QEZJRCVNAM	GRAPHIC (10)	De oudste journaalontvanger moest journaalwijzigen toepassen. Als het veld Gegevens toepassen is ingesteld op PARTIAL_TRANSACTION, bevat de journaalontvanger het begin van de gedeeltelijke bewerking. Anders bevat de journaalontvanger het begin van de opslagbewerking.
QEZJRCVLIB	GRAPHIC (10)	De naam van de bibliotheek met de journaalontvanger die nodig is om de journaalwijzigingen te kunnen toepassen.
QEZJRCVASP	GRAPHIC (10)	De naam van de ASP met de journaalontvanger die nodig is om de journaalwijzigingen te kunnen toepassen. De geldige waarden zijn: *SYSBAS - De journaalontvanger maakt deel uit van het systeem of de gebruikers-ASP. ASP-apparaat - De naam van het ASP-apparaat dat de journaalontvanger bevat.
QEZJTRNI	GRAPHIC (1)	Dit veld bevat informatie over de actuele staat van het object in verhouding tot de grenzen van commitment control. De geldige waarden zijn: x'00' (NONE) - Er zijn geen partiële bewerkingen. x'01' (PARTIAL_TRANSACTION) - Het is object is hersteld met een partiële bewerking. Dit object kan niet worden gebruikt totdat de opdracht APYJRNCHG (Journaalwijzigingen toepassen) of RMVJRNCHG (Journaalwijzigingen verwijderen) is gebruikt om de partiële bewerkingen te voltoeien of terug te draaien. x'02' (ROLLBACK_ENDED) - Voor het object is een gegevensherstelbewerking beëindigd met de optie "Ongedaan maken beëindigen" in het scherm WRKCMDFN (Werken met commitment control-definities). Het is het beste om het object te herstellen, aangezien het niet kan worden gebruikt. Als laatste mogelijkheid kan de opdracht CHGJRNOBJ (In journaal vastgelegd object wijzigen) worden gebruikt om te zorgen dat het object weer kan worden gebruikt. Als u dit doet, kan het object echter onsamenhangend worden.
<p>Opmerking:</p> <ol style="list-style-type: none"> Dit veld is toegevoegd aan de subset met velden van de opdracht PRDIRINF. In dit veld wordt alleen de objectnaam opgeslagen. De rest van de padnaam wordt opgeslagen in Tabel 9, in het veld QEZDIRNAM1 als de directorynaam kleiner is dan 1 kB of in QEZDIRNAM2 als de directorynamen groter zijn dan 1 kB. 		

Tabel 9 is een voorbeeld van een tabel waarin de directory's zijn afgebeeld die door de opdracht RTVDIRINF zijn verwerkt.

Tabel 9. QAEZDxxxD (Kenmerken van directory opslaan)

Veldnaam	Type veld	Veldbeschrijving
QEZDIRIDX	INTEGER	ID van pad (heeft alleen betrekking op directory's).
QEZDIRNAM1 ¹	VARGRAPHIC (1024)	Het bovenliggende directorypad. Alleen gebruikt als de padlengte minder dan 1 kB bedraagt.

Tabel 9. QAEZDxxxD (Kenmerken van directory opslaan) (vervolg)

Veldnaam	Type veld	Veldbeschrijving
QEZDIRNAM2 ¹	DBCLOB (16 M)	Het bovenliggende directorypad. Allen gebruikt als de padlengte meer dan 1 kB bedraagt. Kan paden tot 16 MB opslaan.
QEZDRCCSID	INTEGER	Het directory-CCSID.
QEZDREGION	GRAPHIC (2)	Het padregio-ID.
QEZLANGID	GRAPHIC (3)	Het padtaal-ID.
QEZDIRLEN ¹	INTEGER	Lengte van de padnaam.
QEZDIRFID	GRAPHIC (16)	Het bestands-ID van de directory. Een ID dat bij het object hoort. Een bestands-ID kan met Qp0lGetPathFromFileID() worden gebruikt om de padnaam van een object op te halen.
QEZDFID	INTEGER	Het bestands-ID van de directory.
QEZDIRFSID	BIGINT	Het bestandssysteem-ID van de directory.
QEZDIRGID	BIGINT	Het generatie-ID.
Opmerking:		
1. Dit veld is toegevoegd aan de subset met velden van de opdracht PRTDIRINF.		

Tabel 10 is een voorbeeld van een tabel die is gemaakt met met de opdracht RTVDIRINF. Deze tabel wordt door de opdracht PRTDIRINF gebruikt om te weten welke databasebestanden zijn gebruikt voor de opslag van gegevens die door verschillende instances van de opdracht RTVDIRINF zijn opgehaald.

Tabel 10. QUSRSYS/QAEZDBFILE (Opslagbestanden gemaakt)

Veldnaam	Type veld	Veldbeschrijving
QEZDIRSRC	VARGRAPHIC (5000)	Pad dat opgegeven is in de parameter DIR (RTVDIRINF).
QEZPRCCSID	INTEGER	Het pad-CCSID.
QEZPREGION	GRAPHIC (2)	Het regio-ID van het pad.
QEZPLANGID	GRAPHIC (3)	Het taal-ID van het pad.
QEZOBJFILE ¹	VARGRAPHIC (20)	De naam van het bestand dat wordt gegenereerd om de objectkenmerken in op te slaan.
QEZDIRFILE ¹	VARGRAPHIC (20)	De naam van het bestand dat wordt gegenereerd om de directory-indexen in op te slaan.
QEZLIB ¹	VARGRAPHIC (20)	Bibliotheek waar beide gegenereerde bestanden zich bevinden.
QEZSTRTIME	TIMESTAMP	Datum/tijd waarop RTVDIRINF werd aangeboden.
QEZENDTIME	TIMESTAMP	Datum/tijd waarop RTVDIRINF werd voltooid.
Opmerking:		
1. Dit veld is toegevoegd aan de subset met velden van de opdracht PRTDIRINF.		

De gegevens van RTVDIRINF openen: Er bestaan verschillende opties om toegang te krijgen tot de gegevens in de tabellen. Hieronder zijn de manieren afgebeeld waarop u toegang kunt krijgen tot de gegevens die zijn gemaakt met de opdracht RTVDIRINF:

- Met de opdracht PRTDIRINF

Deze opdracht gebruikt u om directory-informatie af te drukken over objecten en directory-informatie in het geïntegreerd bestandssysteem. De gegevens die worden afgedrukt, zijn al opgeslagen in het databasebestand dat is opgegeven door de gebruiker bij de opdracht RTVDIRINF.

- Met elke opdracht of elk programma van IBM dat query's kan uitvoeren op een DB2-tabel op de iSeries.

| Enkele van de meer gebruikelijke hulpmiddelen zijn de opdracht STRSQL (interactieve SQL-sessie starten) en iSeries Navigator.

| Als u bijvoorbeeld objecten in een bepaald pad wilt selecteren (voorheen verzameld met de opdracht RTVDIRINF) die een toewijzingsgrootte hebben van meer dan 10 kB, kunt u de volgende query uitvoeren:

```
| SELECT QEZOBJNAM, QEZALCSIZE FROM bibliotheek_naam/QAEZDxxxxO WHERE  
| QEZALCSIZE > 10240
```

| • U kunt met elke geldige DB-methode uw eigen programma's maken en de databasetabellen openen. Voorbeelden van deze methodes zijn Embedded SQL en SQL CLIs.

| **Met de gegevens van RTVDIRINF:** Hieronder treft u voorbeelden aan die laten zien waarom de gegevens belangrijk zijn of hoe u de gegevens die zijn geproduceerd vanuit elk van de drie tabellen kunt gebruiken.

| • Voor Tabel 8 op pagina 67 kunt u query's uitvoeren om rapporten of statistieken te maken op basis van een van de velden in deze tabel. PRDIRINF bevat geen rapporten die zijn gebaseerd op alle velden. In plaats daarvan wordt een subset gebruikt.

| • De gegevens in Tabel 9 op pagina 78 bevatten alle directory's in het pad dat is opgegeven bij de parameter DIR van de opdracht RTVDIRINF. Als u bepaalde kenmerken van de padnaam te weten wilt komen, bijvoorbeeld het CCSID, het taal-ID of de lengte, dan zijn deze gegevens nuttig. Daarnaast heeft elke directory die in deze tabel wordt opgeslagen een unieke waarde of index die deze waarde aangeeft. In Tabel 8 op pagina 67 treft u hetzelfde veld aan, QEZDIRIDX, dat u vertelt welk object bij welke directory hoort. Om te weten te komen welk object bij welke directory hoort, kunt u een query uitvoeren met joins. Met de volgende query-instructie bijvoorbeeld selecteert u de namen van alle objecten die in de directory "/MYDIR" bestaan:

```
| SELECT QEZOBJNAM FROM bibliotheek_naam/QAEZxxxx0, bibliotheek_naam/QAEZxxxxD WHERE QEZDIRNAM1 = "/MYDIR" AND bibliotheek_
```

| • Tabel 10 op pagina 79 wordt door de opdracht PRDIRINF voornamelijk gebruikt om bepaalde gegevens over RTVDIRINF-uitvoeringen te verkrijgen. Voorbeelden hiervan zijn: de namen van de gemaakte tabellen, de bibliotheek waar de tabellen deel van uitmaken en de begin- en eindtijd van de verwerking. U kunt deze tabel ook gebruiken om te weten te komen wanneer een opdracht RTVDIRINF is gegeven en welke tabellen moeten worden doorzocht om er een query op uit te kunnen voeren.

Toegang met behulp van API's

U kunt toegang krijgen tot het geïntegreerd bestandssysteem door middel van API's. Raadpleeg voor informatie over het gebruik van API's met het geïntegreerd bestandssysteem Perform operations using APIs.

Toegang met behulp van iSeries Navigator

iSeries Navigator is de grafische gebruikersinterface voor het beheer van uw systemen vanaf het bureaublad van Windows. Met iSeries Navigator wordt het gebruik en beheer van uw systeem makkelijker en productiever. U kunt bijvoorbeeld een gebruikersprofiel naar een ander systeem kopiëren door het van de ene iSeries-server naar de andere iSeries-server te slepen. Er zijn wizards die u helpen met het instellen van de beveiliging en de TCP/IP-services en toepassingen.

Met iSeries Navigator kunt u veel taken uitvoeren. Om u op weg te helpen, is hieronder een aantal gemeenschappelijke bestandssysteemtaken afgebeeld:

Werken met bestanden en mappen

- "Een map maken" op pagina 114
- "Een map verwijderen" op pagina 115
- "Een bestand inchecken" op pagina 114
- "Een bestand uitchecken" op pagina 114

- “Instellingsmachtigingen” op pagina 116
- “Bestandstekstconversie instellen” op pagina 116
- “Een bestand of map naar een ander systeem verzenden” op pagina 117
- “Opties wijzigen voor een pakketdefinitie” op pagina 117
- “Een datum en tijd plannen om uw bestand of map te verzenden” op pagina 118
- “Hiermee stelt u in of objecten moeten worden gescand” op pagina 118

Werken met bestandsshare

- “Een bestandsshare maken” op pagina 118
- “Een bestandsshare wijzigen” op pagina 118

Werken met door de gebruiker gedefinieerde bestandssystemen

- “Een geïntegreerd bestandssysteem UDFS maken” op pagina 35
- “Een geïntegreerd bestandssysteem UDFS aankoppelen” op pagina 36
- “Een geïntegreerd bestandssysteem UDFS ontkoppelen” op pagina 36

Objecten opnemen in een journaal

- “Journaal starten” op pagina 92
- “Journaal beëindigen” op pagina 93

Toegang met behulp van iSeries NetServer

iSeries Support for Windows Network Neighborhood (iSeries NetServer) is een functie van IBM Operating System/400 (OS/400) waarmee Windows-clients toegang kunnen krijgen tot gedeelde paden en uitvoerwachtrijen van OS/400. Dankzij iSeries NetServer kunnen PC's die op Windows-software draaien probleemloos toegang krijgen tot gegevens en printers die worden beheerd door uw iSeries. PC-clients op een netwerk gebruiken de bestands- en printersharefuncties van hun eigen besturingssysteem. Dat betekent dat u geen extra software op uw PC hoeft te installeren om iSeries NetServer te kunnen gebruiken.

Ook LINUX-clients die de Samba clientsoftware hebben geïnstalleerd, kunnen probleemloos toegang krijgen tot gegevens en printers via iSeries NetServer. Gedeelde iSeries NetServer-directory's kunnen op Linux-clients worden aangekoppeld als Samba-bestandssystemen (smbfs) op een manier die te vergelijken is met het aankoppelen van NFS-bestandssystemen die zijn geëxporteerd vanaf de iSeries. Raadpleeg voor meer informatie het onderwerp iSeries NetServer in het iSeries Informatiecentrum.

Een iSeries NetServer-bestandsshare is een pad dat iSeries NetServer deelt met PC-clients op het iSeries-netwerk. Een bestandsshare kan bestaan uit elke geïntegreerd bestandssysteem-directory op de iSeries. Voordat u met gemeenschappelijk bestandsgebruik kunt werken via iSeries NetServer, moet u een iSeries NetServer-bestandsshare maken en zo nodig een iSeries NetServer-bestandsshare wijzigen met behulp van iSeries Navigator.

U krijgt als volgt toegang tot bestandsshare van het geïntegreerd bestandssysteem met behulp van iSeries NetServer:

1. Klik met de rechtermuisknop op **Start** en selecteer **Verkennen** om Windows Verkenner te openen op uw Windows-PC.
2. Ga naar het menu **Extra** en kies de menuoptie **Netwerkverbinding maken**.
3. Selecteer een letter van een vrij station voor de bestandsshare, zoals het I:\-station.
4. Geef de naam op van een iSeries NetServer-bestandsshare. U kunt bijvoorbeeld de volgende syntaxis opgeven: `\\QSYSTEM1\Sharenaam`

Opmerking: QSYSTEM1 is de systeemnaam van iSeries NetServer en Sharenaam is de naam van de bestandsshare die u wilt gebruiken.

5. Klik op OK.

Opmerking: Als u verbinding maakt met iSeries NetServer kan de servernaam verschillend van de naam die wordt gebruikt door iSeries Access Family. De naam van iSeries NetServer kan bijvoorbeeld QAS400X zijn en het pad om met bestanden te werken \\QAS400X\QDLS\MYFOLDER.FLR\MYFILE.DOC. De naam van iSeries Access Family kan echter AS400X zijn en het pad om met bestanden te werken \\AS400X\QDLS\MYFOLDER.FLR\MYFILE.DOC.

U kunt kiezen welke directory's u wilt delen met het netwerk via iSeries NetServer. Deze directory's verschijnen als het eerste niveau onder de servernaam. Als u bijvoorbeeld de directory /home/fred met de naam fredsdir deelt, kan een gebruiker die directory openen vanaf de PC met de naam \\QAS400X\FREDSDIR of vanaf een Linux-client met de naam //qas400x/fredsdir.

Het bestandssysteem hoofddirectory (/) is veel beter geschikt voor bestandsbediening vanaf een PC dan de andere iSeries-bestandssystemen. Het kan handig zijn om bestanden te verplaatsen naar het bestandssysteem hoofddirectory (/). Raadpleeg voor meer informatie "Bestanden en mappen naar een andere bestandssysteem verplaatsen" op pagina 115.

Raadpleeg voor meer informatie over iSeries NetServer en bestandsshare de volgende onderwerpen onder **Netwerken** in het iSeries Informatiecentrum:

- iSeries NetServer
- iSeries NetServer-bestandsshare
- Toegang krijgen tot iSeries NetServer-bestandsshare met een Windows PC-client

Toegang met behulp van bestandsoverdrachtsprogramma's

Met de FTP-client kunt u bestanden verplaatsen die zich op de iSeries-server bevinden, met inbegrip van de bestanden in de bestandssystemen hoofddirectory (/), QOpenSys, QSYS.LIB, onafhankelijke ASP QSYS.LIB, QOPT en QFileSvr.400. U kunt hiermee ook mappen en documenten in het bestandssysteem QDLS (documentenbibliotheekservices) verplaatsen. De FTP-client kan interactief worden uitgevoerd in de onbewaakte werkstand Batchverwerking, waarbij client-subopdrachten worden gelezen van een bestand en de antwoorden op de subopdrachten worden geschreven naar een bestand. Andere functies om bestanden op uw server te bewerken zijn ook aanwezig.

U kunt met FTP-ondersteuning bestanden van en naar de volgende bestandssystemen verplaatsen:

- Bestandssysteem "hoofddirectory" (/)
- Bestandssysteem Open systemen (QOpenSys)
- Bestandssysteem Bibliotheek (QSYS.LIB)
- Bestandssysteem Onafhankelijke ASP QSYS.LIB
- Bestandssysteem Documentenbibliotheekservices (QDLS)
- Bestandssysteem Optische media (QOPT)
- Netwerkbestandssysteem (NFS)
- bestandssysteem NetWare (QNetWare)
- Bestandssysteem iSeries NetClient (QNTC)

Houd echter rekening met de volgende beperkingen:

- In het geïntegreerd bestandssysteem beperkt FTP-ondersteuning zich tot het verplaatsen van alleen bestandsgegevens. U kunt geen kenmerkgegevens verplaatsen met FTP.
- In de bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB beperkt de FTP-ondersteuning zich tot secties van fysieke bestanden, secties van fysieke bronbestanden en opslagbestanden. U kunt met

FTP geen andere objecttypen verplaatsen, zoals programma's (*PGM). U kunt echter wel andere objecttypen in een opslagbestand opslaan, het opslagbestand verplaatsen en vervolgens de objecten herstellen.

Raadpleeg voor informatie over FTP de volgende onderwerpen onder **Netwerken** in het iSeries Informatiecentrum:

- FTP
- Bestanden verplaatsen met FTP

Toegang met behulp van een PC

Als uw PC is aangesloten op een iSeries-server kunt u interactief werken met de directory's en objecten op het geïntegreerd bestandssysteem alsof ze zich op uw PC bevonden. U kunt objecten tussen directory's kopiëren door ze in Windows Explorer te verslepen en neer te zetten. Als u dat wilt, kunt u een object van uw server naar de PC kopiëren door het op het serverstation te selecteren en naar het PC-station te verslepen.

Objecten die tussen iSeries-server en PC worden gekopieerd met de Windows-interface worden automatisch van en naar EBCDIC en ASCII geconverteerd. EBCDIC staat voor extended binary-coded decimal interchange code en ASCII voor American National Standard Code for Information Interchange. iSeries Access Family kan zo worden ingesteld dat het deze conversie automatisch uitvoert. U kunt zelfs opgeven dat de conversie wordt uitgevoerd op bestanden met een bepaalde extensie. Vanaf OS/400 V4R4 kan iSeries NetServer ook zo worden ingesteld dat het de conversie van bestanden uitvoert.

Al naar gelang het type object kunt u ermee werken met PC-interfaces en soms PC-toepassingen. Een stroombestand bijvoorbeeld dat tekst bevat, kan worden bewerkt met een PC-editor.

Als u verbinding met een iSeries-server hebt gemaakt met uw PC, stelt het geïntegreerd bestandssysteem de directory's en objecten op de server beschikbaar voor de PC. PC's kunnen werken met bestanden in het geïntegreerd bestandssysteem met behulp van de gemeenschappelijk bestandsgebruikersclients die in het besturingssysteem Windows zijn ingebouwd, een FTP-client of iSeries Navigator (onderdeel van iSeries Access Family). Uw PC gebruikt gemeenschappelijk bestandsgebruikersclients van Windows om toegang te krijgen tot iSeries NetServer, dat draait op uw iSeries-server.

Bestanden verplaatsen met FTP

Met de FTP-client kunt u bestanden verplaatsen die zich op uw iSeries-server bevinden, met inbegrip van de bestanden in de bestandssystemen hoofddirectory (/), QSYS.LIB, onafhankelijke ASP QSYS.LIB, QOpenSys, QOPT en QFileSvr.400. U kunt hiermee ook mappen en documenten in het bestandssysteem QDLS (documentenbibliotheekservices) verplaatsen.

Werken met bestanden via iSeries Navigator

iSeries Access Family bevat iSeries Navigator die verbinding maakt met een iSeries-server en het geïntegreerd bestandssysteem beschikbaar stelt voor de PC. iSeries Navigator is de grafische gebruikersinterface voor beheer en besturing van uw iSeries-server vanaf een Windows-bureaublad.

Werken met bestanden via iSeries NetServer

iSeries NetServer is onderdeel van OS/400. Het zorgt ervoor dat de bestands- en printersharing die is ingebouwd in Windows-clients samenwerkt met uw server.

Opmerking: De nieuwste versie van iSeries Access Family is voor de toegang tot het geïntegreerd bestandssysteem geheel afhankelijk van NetServer. NetServer-ondersteuning is alleen beschikbaar voor TCP/IP-verbindingen met een iSeries-server die draaien op OS/400 V4R2 of hoger.

Directory's converteren van *TYPE1 naar *TYPE2

De hoofddirectory (/), QOpenSys en door de gebruiker gedefinieerde bestandssystemen (UDFS) in het geïntegreerd bestandssysteem ondersteunen *TYPE2-directory's vanaf OS/400 V5R1. De *TYPE2-directory-indeling is een verbetering van de oorspronkelijke *TYPE1-directory-indeling. *TYPE2-directory's hebben een andere interne structuur dan *TYPE1-directory's en zijn betrouwbaarder en functioneler.

Kort na het installeren van V5R3 begint het systeem bestandssystemen die nog *TYPE1-directory's ondersteunen automatisch te converteren naar *TYPE2-directory's. Deze conversie hoort geen grote belasting voor uw systeem te zijn.

- "Overzicht conversie van *TYPE1 naar *TYPE2"
- "Tips: Conversies"

Overzicht conversie van *TYPE1 naar *TYPE2

De hoofddirectory (/), QOpenSys en door de gebruiker gedefinieerde bestandssystemen (UDFS) in het geïntegreerd bestandssysteem ondersteunen *TYPE2-directory's vanaf OS/400 V5R1. De *TYPE2-directory-indeling is een verbetering van de oorspronkelijke *TYPE1-directory-indeling. *TYPE2-directory's hebben een andere interne structuur dan *TYPE1-directory's en zijn betrouwbaarder en functioneler. Naast verbeterde functionaliteit en grotere betrouwbaarheid, zijn sommige nieuwe functies, zoals ondersteuning voor scans in geïntegreerde bestandssystemen, alleen beschikbaar voor objecten in *TYPE2-directory's. Zie "Ondersteuning van scans" op pagina 18 voor meer informatie.

Kort na het installeren van V5R3 begint het systeem bestandssystemen die nog *TYPE1-directory's ondersteunen automatisch te converteren naar *TYPE2-directory's. Deze conversie hoort geen grote belasting voor uw systeem te zijn omdat hij wordt uitgevoerd op de achtergrond met een lage prioriteit.

Als het systeem opnieuw wordt gestart (bewust of door een fout) terwijl de conversie nog niet is voltooid, wordt deze voortgezet nadat de herstart is voltooid. De conversie wordt voortgezet na iedere herstart totdat alle daarvoor in aanmerking komende bestandssystemen zijn geconverteerd.

Als u wilt weten of de conversie is voltooid, raadpleegt u "Conversiestatus vaststellen" op pagina 85.

De bestandssystemen die in aanmerking komen voor deze automatische conversie zijn de hoofddirectory (/), het QOpenSys en de door de gebruiker gedefinieerde bestandssystemen voor de ASP's 1 t/m 32. Zie "Tips: Onafhankelijke ASP" op pagina 87 voor informatie over door de gebruiker gedefinieerde bestandssystemen op onafhankelijke ASP's.

Opmerking: U kunt de automatische conversie naar *TYPE2-directory's voorkomen door de bestandssystemen te converteren voordat u V5R3 installeert. Zie "*TYPE2-directory's gebruiken in OS/400 V5R1 of V5R2" op pagina 11.

Tips: Conversies

Kort na het installeren van V5R3 begint het systeem bestandssystemen die nog *TYPE1-directory's ondersteunen automatisch te converteren naar *TYPE2-directory's. Deze conversie vindt plaats in een secundaire thread van de systeemtaak QFILESYS1.

U moet tijdens de conversie met verschillende dingen rekening houden:

- "Conversiestatus vaststellen" op pagina 85
- "Gebruikersprofielen maken" op pagina 85
- "Hernoemde objecten" op pagina 85
- "Overwegingen ten aanzien van het gebruikersprofiel" op pagina 86
- "Vereisten voor hulpegeugen" op pagina 86

- “Tips: Symbolische koppeling” op pagina 87
- “Tips: Onafhankelijke ASP” op pagina 87
- “Tips: Opslaan en herstellen” op pagina 87
- | • “RCLSTG (Geheugenruimte beschikbaar maken)” op pagina 87
- | • “Scannen op het geïntegreerd bestandssysteem” op pagina 87

| **Conversiestatus vaststellen**

| Kort na het installeren van V5R3 begint het systeem bestandssystemen die nog *TYPE1-directory’s ondersteunen automatisch te converteren naar *TYPE2-directory’s. Deze conversie vindt plaats in een secundaire thread van de systeemtaak QFILESYS1.

| Om de status van het conversieproces vast te stellen, kunt u de opdracht CVTDIR (Directory converteren) als volgt gebruiken:

| CVTDIR OPTION(*CHECK)

| Door de opdracht CVTDIR te geven, wordt de huidige directory-indeling van de hoofddirectory (/), QOpenSys en door de gebruiker gedefinieerde bestandssystemen afgebeeld en wordt aangegeven of deze bestandssystemen op dit moment worden geconverteerd. Daarnaast wordt de actuele prioriteit van de conversiefunctie afgebeeld, het bestandssysteem dat op dat moment wordt geconverteerd door het systeem en het aantal koppelingen en het percentage directory’s dat is verwerkt voor dat bestandssysteem. De prioriteit die het systeem standaard toekent aan de conversiefunctie is zeer laag (99) zodat deze het systeem nauwelijks belast. U kunt de prioriteit van de conversiefunctie echter wijzigen met de opdracht CVTDIR, parameter OPTION, waarde *CHGPTY. Zie CVTDIR voor extra informatie over deze parameterspecificatie.

| Aangezien de conversie wordt uitgevoerd door de taak QFILESYS1, kunt u het taaklogboek van QFILESYS1 afbeelden voor berichten over eventuele fouten met de conversie. Daarnaast worden er verschillende voortgangsberichten verzonden over de conversie van de bestandssystemen. Deze berichten bevatten informatie als: welk bestandssysteem wordt geconverteerd, het aantal koppelingen en het aantal directory’s dat in dat bestandssysteem is verwerkt, etc. Alle foutberichten en veel voortgangsberichten worden ook verstuurd naar de berichtenwachtrij QSYSOPR. Om in toekomst deze berichten te kunnen raadplegen, is het verstandig de QHST- logboeken en de QFILESYS1-taaklogboeken te bewaren omdat die de betreffende berichten bevatten. Als de bestandssystemen volledig zijn geconverteerd en het geïntegreerd bestandssysteem naar bevrediging functioneert, kunt u deze historische gegevens wissen.

| **Gebruikersprofielen maken**

| De conversiefunctie maakt een gebruikersprofiel dat wordt gebruikt zolang de conversie wordt uitgevoerd. Dit gebruikersprofiel heet QP0FCWA. Het wordt door de conversiefunctie gebruikt om eigenaar te kunnen zijn van geconverteerde directory’s in het bestandssysteem als de oorspronkelijke eigenaar niet in staat is eigenaar te zijn van zijn eigen directory’s.

| Het profiel wordt gewist als de conversie is voltooid, indien mogelijk. Als het eigendom van een directory aan dit gebruikersprofiel wordt gegeven, wordt bericht CPIA08B verzonden naar taaklogboek QFILESYS1 en berichtenwachtrij.

| **Hernoemde objecten**

| Namen van koppelingen in *TYPE2-directory’s moeten geldige UTF-16-namen zijn. Dit in tegenstelling tot *TYPE1-directory’s, die UCS2 Level 1-namen hebben. Om deze reden kunnen tijdens een directoryconversie ongeldige of duplicaatnamen worden aangetroffen. Als een ongeldige of duplicaatnaam wordt aangetroffen, wordt deze gewijzigd in een unieke, geldige UTF-16-naam. Vervolgens wordt bericht CPIA08B verzonden naar taaklogboek QFILESYS1 en berichtenwachtrij met daarin de oude en de nieuwe naam. Samengestelde tekens of ongeldige surrogaattekenparen in een naam kunnen de reden zijn dat een object wordt hernoemd.

Raadpleeg voor meer informatie over tekens “Samengestelde tekens” of “Surrogaattekens”.

Ga voor meer informatie over UTF-16 naar de Unicode-website: <http://www.unicode.org> ).

Samengestelde tekens: Sommige tekens bestaan uit meer dan één Unicode-teken. Er zijn bijvoorbeeld tekens met een accent of een trema. Deze tekens moeten worden gewijzigd (genormaliseerd) in een gemeenschappelijke indeling voordat ze worden opgeslagen in de directory, zodat alle objecten een unieke naam hebben. Een gecombineerd teken normaliseren is een proces waarbij het teken naar een bekende en voorspelbare indeling wordt vertaald. De indeling die voor *TYPE2-directory's is gekozen, is de canonië samengestelde vorm. Als er in een *TYPE1-directory twee objecten bestaan met dezelfde samengestelde tekens, worden ze genormaliseerd naar dezelfde naam. Dit veroorzaakt een conflict, ook als het ene object opgemaakte samengestelde tekens bevat en het andere niet-opgemaakte samengestelde tekens. Daarom wordt de naam van een van de twee gewijzigd voordat het wordt gekoppeld in de *TYPE2-directory.

Surrogaattekens: Sommige tekens beschikken niet over een geldige Unicode-variant. De tekens hebben bijzondere waarden. Ze bestaan uit twee Unicode-tekens in twee verschillende reeksen. Het eerste Unicode-teken bevindt zich in een reeks, bijvoorbeeld 0xD800-0xD8FF) en het tweede Unicode-teken bevindt zich in de tweede reeks, bijvoorbeeld 0xDC00-0xDCFF. Dit wordt een surrogaatpaar genoemd. Als een van de Unicode-tekens ontbreekt of als ze in de verkeerde volgorde staan (een gedeelte van een teken), is de naam ongeldig. Dit type namen was toegestaan in *TYPE1-directory's. Dit is niet langer het geval in *TYPE2-directory's. Om door te kunnen gaan met de conversie wordt, indien een van deze ongeldige namen wordt gevonden, de naam gewijzigd voordat het object in de *TYPE2-directory wordt gekoppeld.

Overwegingen ten aanzien van het gebruikersprofiel

Bij het uitvoeren van de conversie wordt alles in het werk gesteld om te zorgen het gebruikersprofiel dat eigenaar is van een *TYPE1-directory ook de eigenaar wordt van de geconverteerde *TYPE2-directory. Omdat de *TYPE1- en *TYPE2-directory's tijdelijk naast elkaar bestaan, heeft dit invloed op de hoeveelheid geheugen die eigendom is van het gebruikersprofiel en het aantal items in het gebruikersprofiel.

De volgende onderwerpen geven meer informatie over gebruikersprofielen.

- “Maximale geheugenruimte voor een gebruikersprofiel wijzigen”
- “Eigenaar van een directory wijzigen”

| **Maximale geheugenruimte voor een gebruikersprofiel wijzigen:** Tijdens het converteren van de
| directory's bestaat een aantal directory's tijdelijk in beide indelingen en zijn ze eigendom van hetzelfde
| gebruikersprofiel. Als tijdens de conversie de geheugenlimiet voor een gebruikersprofiel wordt bereikt,
| wordt de geheugenruimte voor dat gebruikersprofiel vergroot. Bericht CPIA08C wordt verzonden naar
| taaklogboek QFILESYS1 en berichtenwachtrij QSYSOPR.

| **Eigenaar van een directory wijzigen:** Als het gebruikersprofiel dat eigenaar is van een *TYPE1-directory
| niet in staat is eigenaar te worden van de *TYPE2-directory die is gemaakt, wordt als eigenaar van die
| *TYPE2-directory het alternatieve gebruikersprofiel aangewezen, zoals wordt beschreven in het onder-
| werp Gebruikersprofielen maken. Bericht CPIA08B verzonden naar taaklogboek QFILESYS1 en berichten-
| wachtrij QSYSOPR, en de conversie wordt voortgezet.

Vereisten voor hulpgeheugen

| U moet bij het converteren van directory's in een bestandssysteem naar *TYPE2 rekening houden met de
| vereisten voor hulpgeheugen. Er zijn verschillende aspecten ten aanzien van de hulpgeheugenvereisten:

- De uiteindelijke omvang na de conversie naar *TYPE2
- Extra geheugenruimte tijdens de conversie

Veelal is de uiteindelijke omvang van de *TYPE2-directory geringer dan van de *TYPE1-directory. *TYPE2-directory's met minder dan 350 objecten gebruiken standaard minder hulpgeheugenruimte dan *TYPE1-directory's met hetzelfde aantal objecten. *TYPE2-directory's met meer dan 350 objecten zijn gemiddeld 10 procent groter dan *TYPE1-directory's.

| Tijdens de conversie is extra geheugenruimte nodig. De conversiefunctie vereist namelijk dat beide versies - *TYPE1 en *TYPE2 - van de directory's gelijktijdig bestaan.

| **Opmerking:** Het is misschien een idee om voordat u V5R3 installeert de optie *ESTIMATE van de opdracht CVTDIR in V5R2 uit te voeren, omdat dit een schatting oplevert van de minimale hoeveelheid hulpgeheugen die u tijdens de conversie nodig hebt.

Tips: Symbolische koppeling

Symbolische koppelingen zijn objecten in het geïntegreerd bestandssysteem die een pad bevatten naar een ander object. Er zijn momenten tijdens de conversie waarop een objectnaam kan worden gewijzigd. Als een van de elementen van het pad in een symbolische koppeling wordt gewijzigd tijdens de conversie, verwijst de inhoud van de symbolische koppeling niet meer naar het object. Zie Hernoemde objecten voor meer informatie over het hernoemen van objecten.

Tips: Onafhankelijke ASP

| Als de door de gebruiker gedefinieerde bestandssystemen in een onafhankelijke ASP nog niet zijn geconverteerd naar de *TYPE2-directory-indeling, gebeurt dat alsnog bij de eerste keer dat de onafhankelijke ASP online wordt gezet met een systeem dat waarop OS/400 V5R2 of later is geïnstalleerd. In verband met de planning heeft OS/400 V5R1 een functie waarmee u kunt schatten hoelang de conversie gaat duren. Voordat u de onafhankelijke ASP online zet met de server met V5R2 of later, moet u de volgende API uitvoeren op uw systeem met V5R1 wanneer de onafhankelijke ASP (genaamd ASP_NAME) online staat en actief is:

```
CALL QP0FCVT2 (*ESTIMATE ASP_NAME *TYPE2)
```

Tips: Opslaan en herstellen

*TYPE1-directory's kunnen worden opgeslagen en hersteld in een bestandssysteem dat is geconverteerd naar *TYPE2. Ook kunnen *TYPE-directory's worden opgeslagen en hersteld in een bestandssysteem met *TYPE1-indeling, mits geen van de *TYPE1-limieten is overschreden toen de directory nog de *TYPE2-indeling had.

RCLSTG (Geheugenruimte beschikbaar maken)

| Tijdens de conversie van de bestandssystemen hoofddirectory (/), QOpenSys en gebruikers-ASP UDFS kan de opdracht RCLSTG niet worden uitgevoerd op directory's in het geïntegreerd bestandssysteem, met inbegrip van die in onafhankelijke ASP's. De parameterwaarden OMIT(*DIR) kan echter worden gebruikt om directory's van het geïntegreerd bestandssysteem achterwege te laten en niet-directorygerelateerde objecten terug te halen. Om te bepalen of de bestandssystemen volledig geconverteerd zijn zodat u RCLSTG kunt uitvoeren inclusief de directory's op het geïntegreerd bestandssysteem, raadpleegt u "Conversiestatus vaststellen" op pagina 85.

Scannen op het geïntegreerd bestandssysteem

| Op objecten in de bestandssystemen hoofddirectory (/), QOpenSys en gebruikers-ASP UDFS worden geen scans uitgevoerd met behulp van de exit points van het geïntegreerd bestandssysteem die betrekking hebben op het scannen, voordat de bestandssystemen volledig zijn geconverteerd naar de *TYPE2-directory-indeling. De kenmerken die betrekking hebben op het scannen kunnen worden ingesteld voor objecten in *TYPE1- en *TYPE2-directory's om op te geven of de objecten moeten worden afgezocht of niet, ook al is het systeem nog niet volledig geconverteerd. Voor meer informatie over de ondersteuning van scans in het geïntegreerd bestandssysteem, raadpleegt u "Ondersteuning van scans" op pagina 18.

| Terwijl het systeem objecten converteert van *TYPE1-directory-indeling naar *TYPE2-directory-indeling, wordt de systeemwaarde voor de besturing van scans, 'Scannen bij volgende toegang nadat object is her-

l steld', in overweging genomen, alsof het geconverteerde object werd hersteld. Als de waarde voor 'Scannen bij volgende toegang nadat object is hersteld' wordt opgegeven terwijl de conversie aan de gang is, wordt een object dat zich in een *TYPE1-directory bevond en het kenmerk 'het object wordt niet gescand' had gekregen, toch ten minste één keer gescand nadat het systeem volledig is geconverteerd. Voor meer informatie over systeemwaarden die betrekking hebben op scannen, raadpleegt u "Verwante systeemwaarden" op pagina 20.

l Om te bepalen wanneer de bestandssystemen volledig geconverteerd zijn zodat de scans voor het geïntegreerd bestandssysteem kunnen beginnen in dat bestandssysteem, raadpleegt u "Conversiestatus vaststellen" op pagina 85.

Objecten opnemen in een journaal

l De belangrijkste functie van een journaal is dat u aan de hand daarvan wijzingen die aan een object zijn gemaakt sinds het voor het laatst werd opgeslagen, kunt herstellen. Daarnaast is het kopiëren van objectwijzigingen naar een ander systeem voor grotere beschikbaarheid of spreiding van de werkbelasting een belangrijke functie van een journaal.

U vindt hier een kort overzicht van journaalbeheer, alsmede enige overwegingen voor het in een journaal opnemen van geïntegreerd bestandssysteem-objecten en een beschrijving van journaalondersteuning voor geïntegreerd bestandssysteem-objecten.

Raadpleeg de volgende onderwerpen voor meer informatie over objecten opnemen in journalen.

- "Journaaloverzicht"
- "Journaal starten" op pagina 92
- l • "Journaal wijzigen" op pagina 93
- "Journaal beëindigen" op pagina 93

Journaaloverzicht

De volgende onderwerpen schetsen in het kort welke ondersteuning er is voor het opnemen van geïntegreerd bestandssysteem-objecten in een journaal:

- "Journaalbeheer"
- "Objecten die u in een journaal zou moeten opnemen" op pagina 89
- "In een journaal opgenomen geïntegreerd bestandssysteem-objecten" op pagina 89
- "Journaalbewerkingen" op pagina 90
- "Bijzondere overwegingen voor journaalitems" op pagina 91
- "Overwegingen voor meerdere vaste koppelingen en het opnemen in journaals" op pagina 92

Voor uitgebreide informatie over het opnemen in journalen van geïntegreerd bestandssysteem-objecten raadpleegt u het onderwerp Journaalbeheer in het iSeries Informatiecentrum.

Journaalbeheer

De belangrijkste functie van journaalbeheer is dat u aan de hand daarvan wijzingen die aan een object zijn gemaakt sinds het voor het laatst werd opgeslagen, kunt herstellen. U kunt journaalbeheer ook voor het volgende gebruiken:

- Een audittrail van activiteiten met betrekking tot objecten op het systeem
- Het vastleggen van activiteiten met betrekking tot een object die niet in een journaal kunnen worden opgenomen
- Een snellere herstelprocedure bij het herstellen vanaf media met opslagbewerking met actieve objecten
- l • Ondersteuning bij het kopiëren van objectwijzigingen naar een ander systeem voor grotere beschikbaarheid of spreiding van de werkbelasting
- l • Ondersteuning bij het testen van toepassingen

Een kunt een journaal gebruiken om te definiëren welke objecten u wilt beveiligen met journaalbeheer. Zie "Objecten die u in een journaal zou moeten opnemen" voor meer overwegingen ten aanzien van het opnemen in een journaal van objecten. In het geïntegreerd bestandssysteem kunt u stroombestanden, directory's en symbolische koppelingen opnemen in journaals. Alleen objecten in de bestandssystemen hoofddirectory (/), QOpenSys en UDFS worden ondersteund.

Objecten die u in een journaal zou moeten opnemen

Stel uzelf de volgende vragen als u twijfelt of u een geïntegreerd bestandssysteem-object in een journaal moet opnemen:

- Hoe vaak wordt een object gewijzigd? Een object dat tussen opslagbewerkingen vaak wordt gewijzigd is een goede kandidaat om opgenomen te worden in een journaal.
- Hoe moeilijk is het om de wijzigingen aan het object te herstellen? Worden er veel wijzigingen aan het object gemaakt zonder dat dit wordt weggeschreven? Een object dat bijvoorbeeld wordt gebruikt voor invoer van telefonische bestellingen is moeilijker te reconstrueren dan een object dat wordt gebruikt voor schriftelijke bestellingen.
- Hoe belangrijk zijn de gegevens in het object? Als de laatste versie van het object die is opgeslagen, moet worden teruggezet, welke gevolgen heeft de vertraging die de reconstructie van de wijzigingen met zich meebrengt dan voor de bedrijfsvoering?
- Welk verband is er tussen het object en andere objecten op de server? Hoewel een bepaald object misschien maar zelden wordt gewijzigd, kunnen de gegevens van dat object van cruciaal belang zijn voor een ander, dynamischer object op de server. Veel objecten zijn bijvoorbeeld afhankelijk van een klantenstambestand. Als u bestellingen moet reconstrueren, moeten in het klantenstambestand de nieuwe klanten staan en de wijzigingen van de kredietlimieten die sinds de laatste opslagbewerking hebben plaatsgevonden.

In een journaal opgenomen geïntegreerd bestandssysteem-objecten

Sommige geïntegreerd bestandssysteem-objecttypen kunnen in een journaal worden opgenomen met behulp van de journaalondersteuning van OS/400. De ondersteunde objecttypen zijn stroombestanden, directory's en symbolische koppelingen. De hoofddirectory (/), QOpenSys en UDFS zijn de enige bestandssystemen die journaals voor deze objecttypen ondersteunen. IFS-objecten kunnen in een journaal worden opgenomen met de traditionele systeeminterface (CL-opdrachten en API's) of met iSeries Navigator. Via iSeries Navigator kunt u Journaal starten en Journaal beëindigen. Ook kunt u statusgegevens van journaals afbeelden.

Opmerking: Stroombestanden die naar het geheugen wijzen en stroombestanden die worden gebruikt door de Integrated xSeries Server for iSeries (IXS) voor virtuele geheugenruimte op een schijf kunnen niet in een journaal worden opgenomen. Directory's die blokgeoriënteerde speciale bestandsobjecten bevatten kunnen niet in een journaal worden opgenomen. Voorbeelden hiervan zijn: /dev/QASP01, /dev/QASP22 en /dev/IASPNAM.

Onderstaande lijst geeft een overzicht van de ondersteuning voor journaals in het geïntegreerd bestandssysteem:

- Om journaalbewerkingen uit te voeren op de ondersteunde objecttypen kunt u gewone opdrachten of API's gebruiken. Deze interfaces accepteren over het algemeen identificatie in de vorm van een padnaam of een bestands-ID of beide.
- Sommige journaalopdrachten, zoals Journaal starten, Journaal beëindigen, Journaal wijzigen en Journaalwijzigingen toepassen, kunnen worden uitgevoerd op complete substructuren van geïntegreerd bestandssysteem-objecten. Optioneel kunt u de Include- en Exclude-lijsten gebruiken waarbij joker tekens voor objectnamen mogelijk zijn. U kunt bijvoorbeeld met de opdracht Journaal starten opgeven dat u wilt starten met alle objecten in de structuur "/MijnBedrijf" die overeenkomen met het model "*.data", maar met uitsluiting van objecten die overeenkomen met "A*.data" en "B*.data".
- Tot de journaalondersteuning voor directory's behoren directorybewerkingen als koppelingen toevoegen, koppelingen verwijderen, objecten maken, objecten hernoemen en objecten verplaatsen binnen de directory.

Directory's in een journaal ondersteunen een kenmerk dat zo kan worden ingesteld dat nieuwe objecten in de substructuur de actuele journaalstatus van de directory overnemen. Als dit kenmerk voor een in een journaal opgenomen directory wordt ingeschakeld, wordt voor alle stroombestanden, directory's en symbolische koppelingen die zijn gemaakt in of gekoppeld naar de directory (door een vaste koppeling toe te voegen of door het object te hernoemen of te verplaatsen) het opnemen in een journaal automatisch gestart.

Opmerking: Overwegingen met betrekking tot het overnemen van het journaal kenmerk:

- Als u een object hernoemt en laat staan in de directory waarin het zich op dat moment bevindt, wordt het opnemen in een journaal voor het object niet gestart, ook al is die directory ingesteld om de huidige journaalstatus over te nemen.
- Als een directory wordt verplaatst naar een directory die ingesteld is om het journaal kenmerk over te nemen, wordt alleen voor de verplaatste directory het journaal gestart, indien van toepassing. Dit geldt niet voor de objecten in de verplaatste directory.
- Als een object wordt hersteld naar een directory die ingesteld is om het journaal kenmerk over te nemen, wordt het object niet opgenomen in het journaal als dat object ooit in een journaal is opgenomen.
- Als u de opdracht APYJRNCHG (Journaalwijzigingen toepassen) gebruikt, wordt het actuele journaal kenmerk niet gebruikt voor alle directory's. In plaats daarvan wordt voor alle objecten die worden gemaakt ten gevolge van het toepassen, het opnemen in een journaal al dan niet gestart op basis van de gebeurtenissen tijdens de runtime-activiteit die wordt toegepast.

- Objectnamen en volledige padnamen worden vastgelegd in verschillende journaalitems van geïntegreerd bestandssysteem-objecten. Objectnamen en padnamen zijn geschikt voor taalondersteuning.
- Als het systeem abnormaal wordt beëindigd, zorgt het systeem tijdens de opstartprocedure voor herstel van geïntegreerd bestandssysteem-objecten in een journaal.
- De maximale schrijfcapaciteit die door de verschillende schrijfprogrammainterfaces wordt ondersteund, is 2 gigabytes-1. Als RCVSIZOPT (*MAXOPT2 of *MAXOPT3) is opgegeven, is de maximale journaalitemgrootte 4.000.000.000 bytes. Is dat niet het geval, dan is de maximale journaalitemgrootte 15.761.440 bytes. Als u uw stroombestand opneemt in een journaal en een van de schrijfbewerkingen is groter dan 15.761.440 bytes, kunt u beter *MAXOPT2 of *MAXOPT3 gebruiken om fouten te voorkomen.

Voor uitgebreide informatie over het opnemen in journalen van geïntegreerd bestandssysteem-objecten raadpleegt u het onderwerp Journaalbeheer.

Voor meer informatie over de indeling van de verschillende journaalitems is er een Include-bestand in C-taal, p0ljrn1.h, dat wordt meegeleverd in sectie QSYSINC/H (QP0LJRN1). Het bevat informatie over de specifieke gegevensinhoud en -indelingen van journaalitems in het geïntegreerd bestandssysteem.

Voor een complete lijst met alle journaalitems die zijn geregistreerd voor geïntegreerd bestandssysteem-objecten en voor definities van de hierboven vermelde indelingen raadpleegt u Zoek journaalitem-informatie in het onderwerp Journaalbeheer.

Journaalbewerkingen

De volgende bewerkingen worden alleen in een journaal opgenomen als het type object of koppeling dat de bewerking gebruikt, geschikt is om opgenomen te worden in een journaal:

- Object maken.
- Koppeling toevoegen aan een bestaand object.
- Koppeling ontkoppelen.
- Koppeling hernoemen.
- Bestands-ID hernoemen
- Koppeling naar of uit een directory verplaatsen

De volgende journaalbewerkingen zijn specifiek voor een stroombestand:

- Gegevens schrijven of verwijderen
- Bestand afkappen/uitbreiden
- Gegevens gedwongen opslaan
- Opslaan met vrijgemaakte geheugenruimte

De volgende journaalbewerkingen zijn van toepassing op alle objecttypen die in een journaal worden opgenomen:

- Kenmerkwijzigingen (met inbegrip van beveiligingswijzigingen, zoals machtigingen en eigendom)
- Openen
- Sluiten
- Journaal starten
- Opdracht CHGJRNOBJ (In journaal opgenomen object wijzigen)
- Journaal beëindigen
- Opdracht APYJRNCHG (Journaalwijzigingen toepassen) starten
- Opdracht APYJRNCHG (Journaalwijzigingen toepassen) beëindigen
- Opslaan
- Herstellen

Raadpleeg voor meer informatie over het in een journaal opnemen van geïntegreerd bestandssysteem-objecten het onderwerp Journaalbeheer. Voor een complete lijst met alle journaalitems die zijn geregistreerd voor geïntegreerd bestandssysteem-objecten raadpleegt u Zoek journaaliteminformatie in het onderwerp Journaalbeheer.

Bijzondere overwegingen voor journaalitems

Veel bewerkingen in het geïntegreerd bestandssysteem die worden bijgehouden in een journaal, maken intern gebruik van commitment control van de verschillende functies die tijdens de bewerkingen worden uitgevoerd om een enkele transactie te maken. Deze in een journaal opgenomen bewerkingen kunt u pas als voltooid beschouwen als de commitment-controlcyclus een journaalitem Commit heeft (journaalcode C, type CM). In een journaal opgenomen bewerkingen met een journaalitem voor ongedaan maken (journaalcode C, type RB) in de commitment-controlcyclus, zijn mislukte bewerkingen en de journaalitems daarin mogen niet worden gereproduceerd of gedupliceerd.

In een journaal opgenomen geïntegreerd bestandssysteem-items (journaalcode B) die op deze manier commitment control gebruiken, zijn:

- AA — Auditwaarde wijzigen
- B0 — Maken beginnen
- B1 — Samenvatting maken
- B2 — Koppeling toevoegen
- B3 — Hernoemen/verplaatsen
- B4 — Ontkoppelen (bovenliggende directory)
- B5 — Ontkoppelen (koppeling)
- FA — Kenmerkwijziging
- JT — Journaal starten (alleen als een journaal is gestart door een bewerking in een directory die ingesteld is om het journaal kenmerk over te nemen)
- OA — Machtigingswijziging
- OG — Wijziging van object primaire groep
- OO — Wijziging van objecteigenaar

Diverse journaalitems van het geïntegreerd bestandssysteem hebben een bepaald gegevensveld dat aan geeft of een item een samenvattingsitem is. bewerkingen die samenvattingsitems verzenden, verzenden twee items van hetzelfde type naar het journaal. Het eerste item bevat een subset met de itemspecifieke gegevens. Het tweede item bevat de volledige itemspecifieke gegevens en geeft aan dat het een samenvattingsitem is. Programma's die het object repliceren of de bewerking opnieuw uitvoeren, zijn meestal alleen geïnteresseerd in de samenvattingsitems.

Om een item in een in een journaal opgenomen directory te maken wordt het journaalitem B1 (samenvatting maken) beschouwd als het samenvattingsitem.

Sommige journaalbewerkingen moeten een journaalitem verzenden dat een omgekeerde relatie tot de bewerking heeft. Een commitment-controlcyclus met bijvoorbeeld een B4-journaalitem (ontkoppelen) kan ook een B2-journaalitem bevatten (koppeling toevoegen). Dit scenario komt alleen voor bij bewerkingen die het resultaat zijn van het journaalitem voor ongedaan maken (C — RB) .

Dit scenario kan zich om twee redenen voordoen:

1. De bewerking stond op het punt te mislukken en het item was intern nodig voor het opschonen van fouten.
2. De bewerking werd afgebroken door een systeemstoring en tijdens de daaropvolgende opstart-procedure werd het herstel dat het item moest verzenden, uitgevoerd om de onderbroken bewerking terug te draaien.

| **Overwegingen voor meerdere vaste koppelingen en het opnemen in journaals**

| Raadpleeg voor informatie over vaste koppelingen (wat het is en hoe ze werken) "Vaste koppeling" op pagina 12.

| Als er meerdere vaste koppelingen zijn naar een IFS-object dat in een journaal is opgenomen, moeten alle koppelingen tegelijk worden opgeslagen en hersteld om te zorgen dat de koppeling en de bijbehorende journaalgegevens bewaard blijven.

| Als bij sommige van de journaalopdrachten namen zijn opgegeven en als de namen in werkelijkheid meerdere vaste koppelingen zijn, wordt het object slechts 'één keer' bewerkt. De andere vaste koppelingen worden in feite genegeerd.

| Aangezien meerdere vaste koppelingen verwijzen naar hetzelfde object en het journaalitem alleen het bestands-ID heeft dat hetzelfde is voor het object, beelden journaalinterfaces die de padnaam afbeelden, zoals DSPJRN (Journaal afbeelden) slechts één koppelingsnaam voor het object af. Dit zou echter geen probleem moeten zijn aangezien men een object met elke willekeurige naam kan bewerken, met hetzelfde resultaat.

Journaal starten

De belangrijkste functie van een journaal is dat u aan de hand daarvan wijzingen die aan een object zijn gemaakt sinds het voor het laatst werd opgeslagen, kunt herstellen.

| U begint als volgt met het opnemen van een object in een journaal via iSeries Navigator:

1. Klik in **iSeries Navigator** op uw systeem.
2. Klik op **Bestandssystemen**.
3. Klik met de rechtermuisknop op het object dat u wilt opnemen in een journaal en selecteer **Journaal...**
4. Nadat u de gewenste journaalopties hebt geselecteerd, klikt u op **Starten**.

| Om een object op te nemen in een journaal via de op tekens gebaseerde interface, gebruikt u de opdracht STRJRN (Journaal starten) of de API QjoStartJournal.

Voor uitgebreide informatie over journaals en geïntegreerd bestandssysteem-objecten raadpleegt u Journaalbeheer in het iSeries Informatiecentrum.

Journaal wijzigen

De belangrijkste functie van een journaal is dat u aan de hand daarvan wijzingen die aan een object zijn gemaakt sinds het voor het laatst werd opgeslagen, kunt herstellen. Zie Journaal starten voor meer informatie over hoe u kunt beginnen met het opnemen van een object in een journaal. Als een journaal eenmaal is gestart, wilt u misschien de journaalkenmerken voor een object wijzigen zonder het journaal te beëindigen en vervolgens opnieuw te starten. Met de opdracht CHGJRNOBJ (In journaal opgenomen object wijzigen) kunt u in een journaal opgenomen objecten wijzigen.

Journaal beëindigen

De belangrijkste functie van een journaal is dat u aan de hand daarvan wijzingen die aan een object zijn gemaakt sinds het voor het laatst werd opgeslagen, kunt herstellen. Zie Journaal starten voor meer informatie over hoe u kunt beginnen met het opnemen van een object in een journaal. Als een journaal eenmaal is gestart, wilt u misschien voor dat object het journaal weer beëindigen.

U beëindigt een journaal voor een object via iSeries Navigator als volgt:

1. Klik in **iSeries Navigator** op uw systeem.
2. Klik op **Bestandssystemen**.
3. Klik met de rechtermuisknop op het object dat u niet langer wilt opnemen in een journaal en selecteer **Journaal...**
4. Klik op **Beëindigen**.

Om het opnemen van een object in een journaal te beëindigen via de op tekens gebaseerde interface, gebruikt u de opdracht ENDJRN (Journaal beëindigen) of de API QjoEndJournal.

Voor uitgebreide informatie over journaals en geïntegreerd bestandssysteem-objecten raadpleegt u Journaalbeheer in het iSeries Informatiecentrum.

Ondersteuning voor programmeren

De toevoeging van geïntegreerd bestandssysteem aan de iSeries-server heeft geen invloed op de bestaande servertoepassingen van de iSeries. De programmeertalen, hulpprogramma's en systeemondersteuning zoals DDS (Data Description Specifications) werken nog net zo als voor het toevoegen van het geïntegreerd bestandssysteem.

Om echter profijt te hebben van de stroombestanden, directory's en andere ondersteuning van het geïntegreerd bestandssysteem moet u een set API's gebruiken om toegang te krijgen tot de functies van het geïntegreerd bestandssysteem.

Daarnaast kunt u dankzij het toevoegen van het geïntegreerd bestandssysteem gegevens kopiëren tussen fysieke databasebestanden en stroombestanden. U deze kopieën uitvoeren met behulp van CL-opdrachten, de gegevensoverdrachtfunctie van iSeries Access Family of met API's.

In de volgende onderwerpen wordt uitgelegd hoe u kopieerfuncties kunt gebruiken met geïntegreerd bestandssysteem-stroombestanden en hoe u toegang krijgt tot geïntegreerd bestandssysteem-functies met API's:

- Gegevens kopiëren tussen stroombestanden en databasebestanden
- Gegevens kopiëren tussen stroombestanden en opslagbestanden
- Handelingen uitvoeren met behulp van API's
- Socketondersteuning
- Naamgeving en internationale ondersteuning

- Gegevensconversie
- Voorbeeld: C-functies Geïntegreerd bestandssysteem

Gegevens kopiëren tussen stroombestanden en databasebestanden

Als u bekend bent met het werken met databasebestanden met behulp van recordsfuncties, zoals DDS (Data Description Specifications), kan het zijn dat het werken met stroombestanden in sommige opzichten voor u geheel anders is. De verschillen zijn het gevolg van de verschillen in structuur (of misschien het ontbreken van structuur) tussen stroombestanden en databasebestanden. Om gegevens in een stroombestand te openen, geeft u een byte-offset en een lengte op. Om gegevens in een databasebestand te openen, geeft u gewoonlijk de velden op die moeten worden gebruikt en het aantal records dat moet worden verwerkt.

Omdat u de indeling en kenmerken van een recordsbestand vooraf opgeeft, kent het besturingssysteem het bestand. Daardoor kunt u voorkomen dat u bewerkingen uitvoert die niet stroken met de indeling en kenmerken van het bestand. Bij stroombestanden heeft het besturingssysteem weinig of geen weet van de indeling van het bestand. De toepassing moet weten hoe het bestand er uitziet en hoe er op de juiste manier mee moet worden gewerkt. Stroombestanden maken een buitengewoon flexibele programmeeromgeving mogelijk, maar de prijs daarvoor is dat u weinig of geen hulp krijgt van het besturingssysteem. Voor sommige programmeeromstandigheden zijn stroombestanden beter geschikt, voor andere recordsbestanden.

U kunt gegevens tussen stroombestanden en databasebestanden op het geïntegreerd bestandssysteem op verschillende manier kopiëren:

- Gegevens kopiëren met CL-opdrachten
- Gegevens kopiëren met API's
- Gegevens kopiëren met behulp van de functie gegevensoverdracht

Gegevens kopiëren met CL-opdrachten

Er bestaan twee opdrachtcombinaties waarmee u gegevens kunt kopiëren tussen stroombestanden en databasebestandssecties:

- CPYTOSTMF en CPYFRMSTMF
- CPYTOIMPF en CPYFRMIMPF

Opgdrachten CPYTOSTMF en CPYFRMSTMF

Met de opdrachten CPYFRMSTMF (Van stroombestand kopiëren) en CPYTOSTMF (Naar stroombestand kopiëren) kunt u gegevens kopiëren tussen stroombestanden en databasebestandssecties. U kunt een stroombestand maken van een databasebestandssectie met de opdracht CPYTOSTMF. U kunt ook een databasebestandssectie maken van een stroombestand met de opdracht CPYFRMSTMF. Als het bestand of de sectie dat/die het doel is van de kopie niet bestaat, wordt deze/dit gemaakt.

Er zijn echter enkele beperkingen. Het databasebestand moet een door het programma beschreven fysiek bestand zijn dat uit slechts één veld bestaat of een fysiek bronbestand dat uit slechts één veld bestaat. De opdrachten bieden u diversie opties om de gegevens die worden gekopieerd te converteren en opnieuw te formatteren.

De opdrachten CPYTOSTMF en CPYFRMSTMF kunnen ook worden gebruikt om gegevens te kopiëren tussen een stroombestand en een opslagbestand.

De opdrachten CPYTOIMPF en CPYFRMIMPF

Met de opdrachten CPYTOIMPF (Kopiëren naar importbestand) en CPYFRMIMPF (Kopiëren uit importbestand) kunt u gegevens kopiëren tussen stroombestanden en databasebestandssecties. Met de opdrachten CPYTOSTMF en CPYFRMSTMF kunt u geen gegevens verplaatsen van complexe, extern beschreven

(DDS-beschreven) databasebestanden. Het woord *importbestand* verwijst naar het stroombestandstype. De term slaat doorgaans op een bestand dat is gemaakt met als doel gegevens te kopiëren tussen heterogene databases.

Als u van een stroom- of importbestand kopieert, kunt u met de opdracht CPYFRMIMPF een velddefinitiebestand (FDF) opgeven dat de gegevens in het stroombestand beschrijft. U kunt ook opgeven dat het stroombestand begrensd is en welke tekens worden gebruikt om de reeks, het veld en de recordgrenzen te markeren. Ook opties om bijzondere gegevenstypen als datum en tijd te converteren zijn ook beschikbaar.

Bij deze opdrachten is gegevensconversie mogelijk als het doelstroombestand of de doeldatabasesectie reeds bestaat. Als het bestand niet bestaat, kunt u de gegevens converteren door de volgende twee stappen uit te voeren:

1. Kopieer met de opdrachten CPYTOIMPF en CPYFRMIMPF de gegevens tussen het extern beschreven bestand en het fysieke bronbestand.
2. Kopieer met de opdrachten CPYTOSTMF en CPYFRMSTMF (die volledige gegevensconversie bieden ongeacht of het doelbestand bestaat) om te kopiëren tussen het fysieke bronbestand en het stroombestand.

Bijvoorbeeld:

```
CPYTOIMPF FROMFILE(DB2FILE) TOFILE(EXPFILE) DTAFMT(*DLM)
          FLDDLML(';') RCDDLML('X'07') STRDLM('"'') DATFMT(*USA) TIMFMT(*USA)
```

Met de parameter DTAFMT geeft u op dat het invoerstroombestand wordt begrensd. De andere keuzemogelijkheid is DTAFMT(*FIXED), waarvoor u een velddefinitiebestand moet opgeven. Met de parameters FLDDLML, RCDDLML en STRDLM geeft u de tekens op die als begrenzers fungeren of als scheidingstekens voor velden, records en reeksen.

De parameters DATFMT en TIMFMT geven de indeling aan voor de datum- en tijdgegevens die naar het importbestand worden gekopieerd.

De opdrachten zijn handig omdat ze in een programma kunnen worden geplaatst en helemaal op uw server kunnen worden uitgevoerd. De interfaces zijn echter ingewikkeld.

Raadpleeg voor meer informatie de opdracht-Help of het onderwerp Control language (CL) in het iSeries Informatiecentrum.

Gegevens kopiëren met API's

Als u databasebestandssecties wilt kopiëren naar een stroombestand in een toepassing kunt u met de geïntegreerd bestandssysteem-functies open(), read() en write() een sectie openen, gegevens eruit lezen en gegevens er naar toe of naar een ander bestand schrijven. Raadpleeg het onderwerp API's in het Geïntegreerd bestandssysteem in het iSeries Informatiecentrum voor meer informatie.

Gegevens kopiëren met behulp van de functie gegevensoverdracht

De toepassingen voor gegevensoverbrenging van de iSeries Access Family hebben het voordeel van een makkelijk te volgen grafische interface en van automatische gegevensconversie van cijfers en tekens. Voor gegevensoverdracht is echter de installatie van het product iSeries Access Family vereist, alsmede het gebruik van zowel PC- als iSeries-server-resources en de communicatie tussen de twee.

Als u de iSeries Access Family op uw PC en server hebt geïnstalleerd, kunt u de gegevensoverdracht-toepassingen gebruiken voor de gegevensoverdracht tussen stroombestanden en databasebestanden. U kunt ook gegevens overbrengen naar een nieuw databasebestand dat is gebaseerd op een bestaand databasebestand, naar een extern beschreven databasebestand en naar een databasebestand met een nieuwe bestandsdefinitie.

De volgende taken helpen u bij het kopiëren en overbrengen gegevens met behulp van de gegevens-overdrachttoepassingen:

- Gegevens overbrengen van een databasebestand naar een stroombestand
- Gegevens overbrengen van een stroombestand naar een databasebestand
- Gegevens overbrengen naar een databasebestand met een nieuwe bestandsdefinitie
- Een bestand met bestandsbeschrijving maken

Gegevens overbrengen van een databasebestand naar een stroombestand: U brengt als volgt gegevens over van een databasebestand naar een stroombestand op uw server:

1. Maak verbinding met de server.
2. Wijs een netwerkstation toe aan het juiste pad in het iSeries-bestandssysteem.
3. Kies **Gegevensoverdracht vanaf iSeries** in het venster iSeries Access for Windows.
4. Selecteer de server waarvandaan u wilt overbrengen.
5. Selecteer de bestandsnamen met behulp van de iSeries-databasebibliotheek en -bestandsnaam waarvandaan u wilt kopiëren, evenals het netwerkstation waar het resulterende stroombestand moet worden opgeslagen. U kunt ook via **PC - Bestand - Details** een bestandsindeling voor het stroombestand kiezen. Gegevensoverdracht ondersteunt algemene PC-bestandstypen, zoals ASCII, BIFF3, CSV, DIF, tekst met tabscheidingen en WK4.
6. Klik op **Gegevens overbrengen vanaf iSeries** om de bestandsoverdracht uit te voeren.

U kunt deze gegevensoverdracht met de gegevensoverdrachttoepassingen ook in een batchtaak uitvoeren. Volg bovenstaande stappen, maar selecteer de menuoptie **Bestand** om de overdrachtsopdracht op te slaan. De toepassing Gegevensoverdracht naar iSeries-server maakt een .DTT- of een .TFR-bestand. De toepassing Gegevensoverdracht vanaf iSeries maakt een .DTF- of .TTO-bestand. In de iSeries Access Family-directory kunnen vanaf een opdrachtregel twee programma's als batch worden uitgevoerd:

- RTOPCB accepteert een .DTF- of .TTO-bestand als parameter
- RFROMPCB accepteert een .DTT- of .TFR-bestand als parameter

U kunt beide opdracht met een plannertoepassing zo instellen dat ze gepland worden uitgevoerd. U kunt bijvoorbeeld met de System Agent Tool (onderdeel van Microsoft Plus Pack) instellen welk programma moet worden uitgevoerd (bijvoorbeeld RTOPCB MYFILE.TTO) en wanneer.

Gegevens overbrengen van een stroombestand naar een databasebestand: U brengt als volgt gegevens van een stroombestand over naar een databasebestand op uw server:

1. Maak verbinding met de server.
2. Wijs een netwerkstation toe aan het juiste pad in het iSeries-bestandssysteem.
3. Kies **Gegevensoverdracht naar iSeries-server** in het venster iSeries Access for Windows.
4. Selecteer het PC-bestand dat u wilt overbrengen. Voor de naam van het PC-bestand kunt u **Bladeren** kiezen voor het netwerkstation dat u hebt toegewezen en een stroombestand kiezen. U kunt ook een stroombestand gebruiken dat zich op uw PC zelf bevindt.
5. Selecteer de server waarop het extern beschreven databasebestand moet staan.
6. Klik op **Gegevens overbrengen naar iSeries-server** om de bestandsoverdracht uit te voeren.

Opmerking: Als u gegevens verplaatst naar een bestaande databasebestandsdefinitie op de server, vereist de toepassing Gegevensoverdracht naar iSeries-server dat u een bijbehorend bestand met indelingsbeschrijving (FDF) gebruikt. Een FDF-bestand beschrijft de indeling van een stroombestand en wordt gemaakt door de toepassing Gegevensoverdracht vanaf iSeries wanneer gegevens worden overgebracht van een databasebestand naar een stroombestand. Om de overdracht van gegevens van een stroombestand naar een databasebestand te voltooien, klikt u op **Gegevens overbrengen naar iSeries**. Als er geen bestaand .FDF-bestand beschikbaar is, kunt u snel een .FDF-bestand maken.

U kunt deze gegevensoverdracht met de gegevensoverdrachttoepassingen ook in een batchtaak uitvoeren. Volg bovenstaande stappen, maar selecteer de menuoptie **Bestand** om de overdrachtsopdracht op te

slaan. De toepassing Gegevensoverdracht naar iSeries-server maakt een .DTT- of een .TFR-bestand. De toepassing Gegevensoverdracht vanaf iSeries maakt een .DTF- of .TTO-bestand. In de IBM e(logos)server iSeries Access Family-directory kunnen vanaf een opdrachtregel twee programma's als batch worden uitgevoerd:

- RTOPCB accepteert een .DTF- of .TTO-bestand als parameter
- RFROMPCB accepteert een .DTT- of .TFR-bestand als parameter

U kunt beide opdracht met een plannertoepassing zo instellen dat ze gepland worden uitgevoerd. U kunt bijvoorbeeld met de System Agent Tool (onderdeel van Microsoft Plus Pack) instellen welk programma moet worden uitgevoerd (bijvoorbeeld RTOPCB MYFILE.TTO) en wanneer.

Gegevens overbrengen naar een databasebestand met een nieuwe bestandsdefinitie: U brengt als volgt gegevens over naar een databasebestand met een nieuwe bestandsdefinitie:

1. Maak verbinding met de server.
2. Wijs een netwerkstation toe aan het juiste pad in het iSeries-bestandssysteem.
3. Kies **Gegevensoverdracht naar iSeries-server** in het venster iSeries Access for Windows.
4. Open het menu **Hulpprogramma's** in de toepassing Gegevensoverdracht naar iSeries-server.
5. Kies **iSeries-databasebestand maken**.

Er verschijnt een wizard waarmee u van een bestaand PC-bestand een nieuw iSeries-databasebestand kunt maken. U moet de naam opgeven van het PC-bestand waarop het iSeries-bestand dat u maakt is gebaseerd, de naam van het iSeries-bestand dat u gaat maken en nog andere vereiste gegevens. Dit hulpprogramma ontleedt een gegeven stroombestand om nummer, type en afmeting te bepalen van de velden die benodigd zijn in het resulterende databasebestand. Het hulpprogramma kan vervolgens de databasebestandsdefinitie op uw server maken.

Een bestand met bestandsbeschrijving maken: Als u gegevens verplaatst naar een bestaande databasebestandsdefinitie op de server, vereist de toepassing Gegevensoverdracht naar iSeries-server dat u een bijbehorend FDF-bestand (Format Description File) gebruikt. Een FDF-bestand beschrijft de indeling van een stroombestand en wordt gemaakt door de toepassing Gegevensoverdracht vanaf iSeries wanneer gegevens worden overgebracht van een databasebestand naar een stroombestand.

U maakt als volgt een .FDF-bestand:

1. Maak een extern beschreven databasebestand met een indeling die overeenkomt met uw bronstroombestand (aantal velden, soort gegevens).
2. Maak in het databasebestand één tijdelijk gegevensrecord.
3. Maak met behulp van de functie Gegevensoverdracht vanaf iSeries een stroombestand en het bijbehorende .FDF-bestand van dit databasebestand.
4. Nu kunt u de functie Gegevensoverdracht naariSeries-server gebruiken. Geef het .FDF-bestand op met het bronstroombestand dat u wilt overbrengen.

Gegevens kopiëren tussen stroombestanden en opslagbestanden

Er wordt een opslagbestand gebruikt met opslag- en herstelopdrachten om gegevens te handhaven die anders naar band of diskette zouden worden geschreven. Het bestand kan ook worden gebruikt als databasebestand om records te lezen en te schrijven die opslag-/herstelinformatie bevatten. Een opslagbestand kan ook worden gebruikt om objecten te verzenden naar een andere gebruiker op het SNADS-netwerk.

U kunt met de opdracht CPY een opslagbestand kopiëren van en naar een stroombestand. Als u een stroombestand echter wilt terugkopiëren naar een opslagbestandsobject, moeten de gegevens geldige opslagbestandsgegevens zijn (ze moeten afstammen van een opslagbestand en zijn gekopieerd naar een stroombestand).

Door een PC-client te gebruiken, kunt u het opslagbestand ook openen en de gegevens naar uw PC-geheugen of het LAN kopiëren. Vergeet niet dat de gegevens in opslagbestanden niet kunnen worden geopend via het NFS (netwerkbestandssysteem).

Handelingen uitvoeren met behulp van API's

Veel API's waarmee u bewerkingen uitvoert op objecten van het geïntegreerd bestandssysteem hebben de vorm van functies in de C-programmeertaal. U kunt kiezen uit twee sets met functies. Beide kunt u gebruiken in programma's die zijn gemaakt met Integrated Language Environment (ILE) C/400:

- C-functies van het geïntegreerd bestandssysteem die OS/400 bezit.
- C-functies die worden aangeboden door het gelicentieerd programma C/400.

Voor informatie over exitprogramma's die het geïntegreerd bestandssysteem ondersteunt, selecteert u Tabel 12 op pagina 103.

De geïntegreerd bestandssysteem-functies werken alleen via de stroominvoer/uitvoerondersteuning van het geïntegreerd bestandssysteem. De volgende API's worden ondersteund:

Tabel 11. Geïntegreerd bestandssysteem-API's

Functie	Beschrijving
access()	Toegankelijkheid bestand vaststellen
accessx()	Toegankelijkheid bestand vaststellen voor een gebruikers-klasse
chdir()	Actieve directory wijzigen
chmod()	Bestandsmachtigingen wijzigen
chown()	Eigenaar en groep wijzigen voor bestand
close()	Bestandsdescriptor sluiten
closedir()	Directory sluiten
creat()	Nieuw bestand maken of bestaand bestand herschrijven
creat64()	Nieuw bestand maken of bestaand bestand herschrijven (geschikt voor grote bestanden)
DosSetFileLocks()	Bytebereik vergrendelen en ontgrendelen
DosSetFileLocks64()	Bytebereik van een bestand vergrendelen en ontgrendelen (geschikt voor grote bestanden)
DosSetRelMaxFH()	Maximumaantal bestandsdescriptors wijzigen
dup()	Open open bestandsdescriptor dupliceren
dup2()	Open open bestandsdescriptor dupliceren naar een andere descriptor
faccessx()	Toegankelijkheid van bestand vaststellen voor een gebruikers-klasse op basis van descriptor
fchdir()	Actieve directory wijzigen op basis van descriptor
fchmod()	Bestandsmachtigingen wijzigen op basis van descriptor
fchown()	Eigenaar en groep van bestand wijzigen op basis van descriptor
fclear()	Bestand verwijderen
fclear64()	Bestand verwijderen (geschikt voor grote bestanden)
fcntl()	Bestandsbesturing uitvoeren
fpathconf()	Te configureren padvariabelen ophalen op basis van descriptor

Tabel 11. Geïntegreerd bestandssysteem-API's (vervolg)

Functie	Beschrijving
fstat()	Bestandsgegevens ophalen op basis van descriptor
fstat64()	Bestandsgegevens ophalen op basis van descriptor (geschikt voor grote bestanden)
fstatvfs()	Gegevens ophalen op basis van descriptor
fstatvfs64()	Gegevens ophalen op basis van descriptor (geschikt voor grote bestanden)
fsync()	Wijzigingen aan bestand synchroniseren
ftruncate()	Bestand afkappen
ftruncate64()	Bestand afkappen (geschikt voor grote bestanden)
getcwd()	Pad van actieve directory ophalen
getegid()	Effectief groeps-ID ophalen
geteuid()	Effectief gebruikers-ID ophalen
getgid()	Echt groeps-ID ophalen
getgrgid()	Groepsgegevens ophalen met behulp van groeps-ID
getgrnam()	Groepsgegevens ophalen met behulp van groepsnaam
getgroups()	Groeps-ID ophalen
getwponam()	Gebruikersgegevens voor gebruikersnaam ophalen
getpwuid()	Gebruikersgegevens voor gebruikers-ID ophalen
getuid()	Echt gebruikers-ID ophalen
givedescriptor()	Bestandstoegang verlenen aan andere taak
ioctl()	I/O-besturing van bestand uitvoeren
link()	Koppeling naar bestand maken
lseek()	Vershoven uitvoer lezen/schrijven van bestand instellen
lseek64()	Offset lezen/schrijven van bestand instellen (geschikt voor grote bestanden)
lstat()	Bestands- of koppelingsgegevens ophalen
lstat64()	Bestands- of koppelingsgegevens ophalen (geschikt voor grote bestanden)
mmap()	Een overzicht van geheugenadressen maken
mmap64()	Een overzicht van geheugenadressen maken (geschikt voor grote bestanden)
mprotect()	Beveiliging van een overzicht van geheugenadressen wijzigen
msync()	Een overzicht van geheugenadressen synchroniseren
munmap()	Een overzicht van geheugenadressen verwijderen
mkdir()	Directory maken
mkfifo()	Bijzonder bestand FIFO maken
open()	Bestand openen
open64()	Bestand openen (geschikt voor grote bestanden)
opendir()	Directory openen
pathconf()	Te configureren padvariabelen ophalen
pipe()	Interproceskanaal met sockets maken

Tabel 11. Geïntegreerd bestandssysteem-API's (vervolg)

Funcctie	Beschrijving
pread()	Met verschoven uitvoer lezen van descriptor
pread64()	Met verschoven uitvoer lezen van descriptor (geschikt voor grote bestanden)
pwrite()	Met verschoven uitvoer schrijven naar descriptor
pwrite64()	Met verschoven uitvoer schrijven naar descriptor (geschikt voor grote bestanden)
QjoEndJournal()	Journal beëindigen
QjoRetrieveJournal Information()	Journalgegevens ophalen
QJORJIDI()	Journal-ID-gegevens ophalen
QJOSJRNE()	Journalitem verzenden
QjoStartJournal()	Journal starten
QlgAccess()	Toegankelijkheid bestand vaststellen (met behulp van voor taalondersteuning geschikt pad)
QlgAccessx()	Toegankelijkheid bestand vaststellen voor een gebruikersklasse (met behulp van voor taalondersteuning geschikt pad)
QlgChdir()	Actieve directory wijzigen (met behulp van voor taalondersteuning geschikt pad)
QlgChmod()	Bestandsmachtigingen wijzigen (met behulp van voor taalondersteuning geschikt pad)
QlgChown()	Eigenaar en groep van bestand wijzigen (met behulp van voor taalondersteuning geschikt pad)
QlgCreat()	Nieuw bestand maken of bestaand bestand herschrijven (met behulp van voor taalondersteuning geschikt pad)
QlgCreat64()	Nieuw bestand maken of bestaand bestand herschrijven (geschikt voor grote bestanden en met behulp van voor taalondersteuning geschikt pad)
QlgCvtPathToQSYSObjName()	Pad van geïntegreerd bestandssysteem omzetten naar QSYS-objectnaam (met behulp van voor taalondersteuning geschikt pad)
QlgGetAttr()	Systeemkenmerken ophalen voor een object (met behulp van voor taalondersteuning geschikt pad)
QlgGetcwd()	Pad voor actieve directory ophalen (met behulp van voor taalondersteuning geschikt pad)
QlgGetPathFromFileID()	Pad van een object ophalen vanuit zijn bestands-ID (met behulp van voor taalondersteuning geschikt pad)
QlgGetpwnam()	Gebruikersgegevens ophalen voor gebruikersnaam (met behulp van voor taalondersteuning geschikt pad)
QlgGetpwnam_r()	Gebruikersgegevens ophalen voor gebruikersnaam (met behulp van voor taalondersteuning geschikt pad)
QlgGetpwuid()	Gebruikersgegevens ophalen voor gebruikers-ID (met behulp van voor taalondersteuning geschikt pad)
QlgGetpwuid_r()	Gebruikersgegevens ophalen voor gebruikers-ID (met behulp van voor taalondersteuning geschikt pad)
QlgLchown()	Eigenaar en groep van symbolische koppeling wijzigen (met behulp van voor taalondersteuning geschikt pad)

Tabel 11. Geïntegreerd bestandssysteem-API's (vervolg)

Funcctie	Beschrijving
QlgLink()	Koppeling naar bestand maken (met behulp van voor taalondersteuning geschikt pad)
QlgLstat()	Bestands- of koppelingsgegevens ophalen (met behulp van voor taalondersteuning geschikt pad)
QlgLstat64()	Bestands- of koppelingsgegevens ophalen (geschikt voor grote bestanden en met behulp van voor taalondersteuning geschikt pad)
QlgMkdir()	Directory maken (met behulp van voor taalondersteuning geschikt pad)
QlgMkfifo()	Speciaal FIFO-bestand maken (met behulp van voor taalondersteuning geschikt pad)
QlgOpen()	Bestand openen (met behulp van voor taalondersteuning geschikt pad)
QlgOpen64()	Bestand openen (geschikt voor grote bestanden en met behulp van voor taalondersteuning geschikt pad)
QlgOpendir()	Directory openen (met behulp van voor taalondersteuning geschikt pad)
QlgPathconf()	Configureerbare padvariabelen ophalen (met behulp van voor taalondersteuning geschikt pad)
QlgProcessSubtree()	Directory's en objecten verwerken in een directorystructuur (met behulp van voor taalondersteuning geschikt pad)
QlgReaddir()	Directorygegevens lezen (met behulp van voor taalondersteuning geschikt pad)
QlgReaddir_r()	Directorygegevens lezen (threadsafe en met behulp van voor taalondersteuning geschikt pad)
QlgReadlink()	Waarde van symbolische koppeling (met behulp van voor taalondersteuning geschikt pad)
QlgRenameKeep()	Bestand of directory hernoemen, behoud <i>nieuw</i> als dit/deze bestaat (met behulp van voor taalondersteuning geschikt pad)
QlgRenameUnlink()	Bestand of directory hernoemen, ontkoppel <i>nieuw</i> als dit/deze bestaat (met behulp van voor taalondersteuning geschikt pad)
QlgRmdir()	Directory verwijderen (met behulp van voor taalondersteuning geschikt pad)
QlgSaveStgFree()	Objectgegevens opslaan en het geheugen ervan vrijmaken (met behulp van voor taalondersteuning geschikt pad)
QlgSetAttr()	Systeemkenmerken instellen voor een object (met behulp van voor taalondersteuning geschikt pad)
QlgStat()	Bestandsgegevens ophalen (met behulp van voor taalondersteuning geschikt pad)
QlgStat64()	Bestandsgegevens ophalen (geschikt voor grote bestanden en met behulp van voor taalondersteuning geschikt pad)
QlgStatvfs()	Bestandssysteemgegevens ophalen (met behulp van voor taalondersteuning geschikt pad)
QlgStatvfs64()	Bestandssysteemgegevens ophalen (geschikt voor grote bestanden en met behulp van voor taalondersteuning geschikt pad)

Tabel 11. Geïntegreerd bestandssysteem-API's (vervolg)

Funcctie	Beschrijving
QlgSymlink()	Symbolische koppeling maken (met behulp van voor taalondersteuning geschikt pad)
QlgUnlink()	Bestand ontkoppelen (met behulp van voor taalondersteuning geschikt pad)
QlgUtime()	Bestandstoegang en wijzigingstijden instellen (met behulp van voor taalondersteuning geschikt pad)
QP0FPTOS()	Diverse bestandssysteemfuncties uitvoeren
QP0LCHSG()	Scanhandtekening wijzigen
Qp0lCvtPathToSYSObjName()	Pad van geïntegreerd bestandssysteem omzetten naar QSYS-objectnaam
QP0LFLOP()	Diverse objectbewerkingen uitvoeren
Qp0lGetAttr()	Systeemkenmerken ophalen voor een object
Qp0lGetPathFromFileID()	Pad van een object ophalen vanuit zijn bestands-ID
Qp0lOpen()	Bestand openen met voor taalondersteuning geschikt pad
Qp0lProcessSubtree()	Directory's of objecten verwerken in een directorystructuur
Qp0lRenameKeep()	Bestand of directory hernoemen, behoud <i>nieuw</i> als dit/deze bestaat
Qp0lRenameUnlink()	Bestand of directory hernoemen, ontkoppel <i>nieuw</i> als dit/deze bestaat
QP0LROR()	Objectverwijzingen ophalen
QP0LRRO()	Objecten ophalen waarnaar is verwezen
QP0LRTSG()	Scanhandtekening ophalen
Qp0lSaveStgFree()	Objectgegevens opslaan en het geheugen ervan vrijmaken
Qp0lSetAttr()	Systeemkenmerken instellen voor een object
Qp0lUnlink()	Bestand ontkoppelen met voor taalondersteuning geschikt pad
qsyssetgid()	Effectief groeps-ID instellen
qsyssetuid()	Effectief gebruikers-ID instellen
qsyssetgid()	Groeps-ID instellen
qsyssetregid()	Echt en effectief groeps-ID instellen
qsyssetreuid()	Echt en effectief gebruikers-ID instellen
qsyssetuid()	Gebruikers-ID instellen
QZNFRTVE()	NFS-exportgegevens ophalen
read()	Lezen van bestand
readdir()	Directorygegevens lezen
readdir_r()	Directorygegevens lezen (threadsafe)
readlink()	Waarde van symbolische koppeling lezen
readv()	Lezen van bestand (vector)
rename()	Bestand of directory hernoemen Kan worden gedefinieerd met de semantiek van Qp0lRenameKeep() of Qp0lRenameUnlink().
rewinddir()	Directorystroom opnieuw instellen
rmdir()	Directory verwijderen

Tabel 11. Geïntegreerd bestandssysteem-API's (vervolg)

Funcctie	Beschrijving
select()	I/O-status van meerdere bestandsdescriptors controleren
stat()	Bestandsgegevens ophalen
stat64()	Bestandsgegevens ophalen (geschikt voor grote bestanden)
statvfs()	Bestandssysteem informatie ophalen
statvfs64()	Bestandssysteemgegevens ophalen (geschikt voor grote bestanden)
symlink()	Symbolische koppeling maken
sysconf()	Systeemconfiguratievariabelen ophalen
takedescriptor()	Bestandstoegang nemen van een andere taak
umask()	Machtigingsmasker instellen voor taak
unlink()	Koppeling verwijderen naar bestand
utime()	Bestandstoegang en wijzigingstijden instellen
write()	Naar bestand schrijven
writenv()	Naar bestand schrijven (vector)

Opmerking: Sommige van de functies worden ook gebruikt voor OS/400 -sockets. Om te zien welke beperkingen deze functies hebben bij gebruik in bepaalde bestandssystemen, raadpleegt u het onderwerp Werken met bestandssystemen. Raadpleeg "Voorbeeld: C-functies Geïntegreerd bestandssysteem" op pagina 108 voor een voorbeeldprogramma met gebruik van C-functies van het geïntegreerd bestandssysteem.

Tabel 12. Exitprogramma's van het geïntegreerd bestandssysteem

Funcctie	Beschrijving
API scan van geïntegreerd bestandssysteem bij sluiten	Wordt aangeroepen tijdens het sluiten, zoals met de API close(). Dit exitprogramma moet door de gebruiker worden aangeboden.
API scan van geïntegreerd bestandssysteem bij openen	Wordt aangeroepen tijdens het openen, zoals met de API open(). Dit exitprogramma moet door de gebruiker worden aangeboden.
Pad verwerken	Wordt aangeroepen door de API Qp0lProcessSubtree() voor ieder object in de zoekbewerking van de API die aan de selectiecriteria voldoen. Dit exitprogramma moet door de gebruiker worden aangeboden.
Vrij geheugen opslaan	Wordt aangeroepen door de API Qp0lSaveStgFree() om een iSeries-objecttype *STMF op te slaan

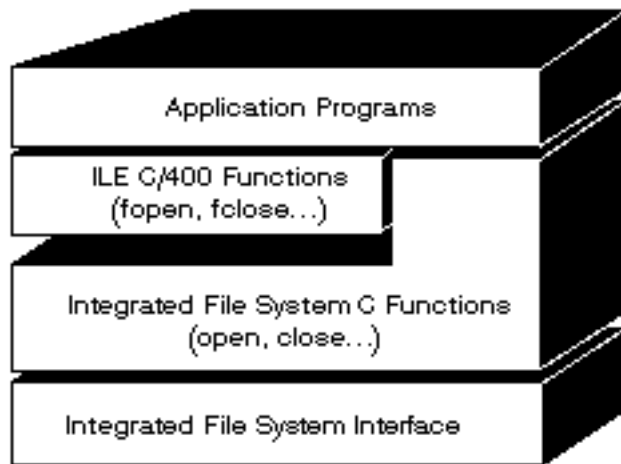
Raadpleeg de volgende onderwerpen over API's in het geïntegreerd bestandssysteem.

- ILE C/400-functies
- Ondersteuning van grote bestanden voor API's
- Padregels voor API's
- Bestandsdescriptor
- Beveiliging
- Gegevens kopiëren met API's
- Het onderwerp API's in het iSeries Informatiecentrum

ILE C/400-functies

ILE C/400 biedt de standaard C-functies die gedefinieerd zijn door ANSI. Deze functies zijn actief via de I/O-ondersteuning van het gegevensbeheer of de stroom-I/O-ondersteuning van het geïntegreerd bestandssysteem, afhankelijk van wat u hebt opgegeven bij het maken van het C-programma. Het compileerprogramma gebruikt de gegevensbeheer-I/O tenzij u iets anders opgeeft.

Als het compileerprogramma de stroom-I/O van het geïntegreerd bestandssysteem moet gebruiken, moet u *IFSIO opgeven voor de parameter SYSIFCOPT (systeeminterfaceoptie) in de opdracht CRTCMOD (ILE C/400-module maken) of in de opdracht CRTBNDC (gebonden C-programma maken). Als u *IFSIO opgeeft, worden de I/O-functies van het geïntegreerd bestandssysteem gebonden in plaats van de I/O-functies van het gegevensbeheer. Het komt erop neer dat de ILE C/400 C-functies de geïntegreerd bestandssysteem-functies gebruiken om I/O uit te voeren.



Figuur 10. ILE C/400-functies gebruiken de stroom-I/O-functies van het geïntegreerd bestandssysteem

Raadpleeg voor meer informatie over het gebruik van ILE C/400-functies met stroom-I/O van het geïntegreerd bestandssysteem de uitgave WebSphere Development Studio: ILE C/C++ Programmers Guide



Informatie over de afzonderlijke ILE C/400 functies vindt u in de uitgave WebSphere Development Studio: C/C++ Language Reference.

Ondersteuning van grote bestanden

- | De API's van het geïntegreerd bestandssysteem zijn uitgebreid zodat toepassingen ook zeer grote bestanden kunnen opslaan en bewerken. Met het geïntegreerd bestandssysteem zijn stroombestandgrootten mogelijk van ongeveer één terabyte in de hoofddirectory (/), QOpenSys en door de gebruiker gedefiniëerde bestandssystemen.

Het geïntegreerd bestandssysteem biedt een aantal 64-bit API's van het type UNIX en zorgt voor een eenvoudige toewijzing van bestaande 32-bit API's naar 64-bit API's die grote bestandsgrootten en -offsets kunnen openen met behulp van 8-byte integercriteria. Informatie over de afzonderlijke 64-bit API's vindt u in het onderwerp over Geïntegreerd bestandssysteem API's in het iSeries Informatiecentrum.

Onderstaande mogelijkheden worden aangeboden zodat toepassingen grote bestanden kunnen ondersteunen:

1. Als het macrolabel `_LARGE_FILE_API` gedefinieerd is tijdens het compileren, hebben toepassingen toegang tot API's en gegevensstructuren die geschikt zijn voor 64-bit. Voorbeeld: een toepassing die API `stat64()` en structuur `stat64` moet gebruiken, moet tijdens het compileren `_LARGE_FILE_API` definiëren.
2. Als het macrolabel `_LARGE_FILES` door de toepassingen tijdens het compileren is gedefinieerd, worden bestaande API's en gegevensstructuren toegewezen aan de corresponderende 64-bit versies. Voorbeeld: als een toepassing tijdens het compileren `_LARGE_FILES` definieert, wordt een aanroep aan de API `stat()` toegewezen aan de API `stat64()` en de structuur `stat()` aan de structuur `stat64()`.

De toepassingen die ondersteuning voor grote bestanden moeten gebruiken, kunnen tijdens het compileren `_LARGE_FILE_API` definiëren en rechtstreeks naar de 64-bit API's coderen of ze kunnen tijdens het compileren `_LARGE_FILES` definiëren. Alle relevante API's en gegevensstructuren worden vervolgens automatisch toegewezen 64-bit versie.

Voor toepassingen die niet de ondersteuning voor grote bestanden moeten gebruiken, heeft dit geen gevolgen. Zij kunnen zonder wijzigingen API's van het geïntegreerd bestandssysteem blijven gebruiken.

Padregels voor API's

Als u een object met een API van het geïntegreerd bestandssysteem of een ILE C/400-API wilt sturen, geeft u het object aan met zijn pad. Hieronder volgt een samenvatting van de regels waaraan u moet denken bij het opgeven van paden in de API's. De term **object** verwijst bij deze regels naar een directory, bestand, koppeling of ander object.

- Padnamen worden opgegeven in hiërarchische volgorde, te beginnen met het hoogste niveau van de directoryhiërarchie. De namen van de afzonderlijke componenten in het pad worden gescheiden door een schuine streep naar rechts (/); bijvoorbeeld:

```
Dir1/Dir2/Dir3/UsrFile
```

De schuine streep naar links (\) wordt niet als scheidingsteken herkend. Deze wordt beschouwd als een gewoon teken in een naam.

- In een directory moet elke objectnaam uniek zijn.
- De maximumlengte van elke component van de padnaam en de maximumlengte van de padnaamreeks kan per bestandssysteem verschillen. Zie "Vergelijking bestandssystemen" op pagina 24 voor de beperkingen van elk bestandssysteem.
- Een / aan het begin van een pad geeft aan dat het pad begint bij de "hoofddirectory" (/), bijvoorbeeld:

```
/Dir1/Dir2/Dir3/UsrFile
```

- Als het pad niet met een / begint, wordt aangenomen dat het pad begint in de actieve directory, bijvoorbeeld:

```
MijnDir/MijnBestand
```

, waarbij `MijnDir` een subdirectory is van de actieve directory.

- Om verwisseling met de speciale waarden van de iSeries-server te voorkomen, mogen paden niet met één sterretje (*) beginnen. Om een padnaam op te geven die begint met een willekeurig aantal tekens gebruikt u twee sterretjes (**), bijvoorbeeld:

```
'**.file'
```

Dit slaat alleen op paden vanaf de gebruikersdirectory waar geen andere tekens voor het sterretje (*) staan.

- Wanneer u werkt met objecten in het QSYS.LIB-bestandssysteem moeten de componentnamen de vorm *naam.objecttype* hebben, bijvoorbeeld:

```
/QSYS.LIB/PAYROLL.LIB/PAY.FILE
```

Zie "Bibliotheekbestandssysteem (QSYS.LIB)" op pagina 37 voor meer informatie.

- Wanneer u werkt met objecten in het bestandssysteem onafhankelijke ASP QSYS.LIB moeten de componentnamen de vorm *naam.objecttype* hebben, bijvoorbeeld:

```
 '/asp_name/QSYS.LIB/PAYDAVE.LIB/PAY.FILE
```

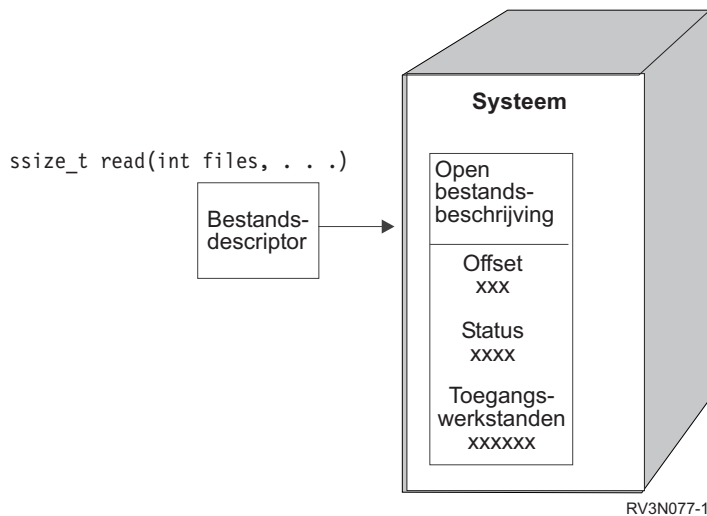
Zie "Onafhankelijke ASP QSYS.LIB" op pagina 40 voor meer informatie.

- Gebruik geen dubbele punt (:) in paden. De dubbele punt heeft voor de server namelijk een bijzondere betekenis.
- In tegenstelling tot padnamen in geïntegreerd bestandssysteem hebben opdrachten (zie “Regels voor paden van CL-opdrachten en -schermen” op pagina 64), een sterretje (*), een vraagteken (?), een enkel aanhalingsteken ('), een dubbel aanhalingsteken (") en een tilde (~) geen speciale betekenis. Ze worden verwerkt als elk ander teken in een naam. De enige API's die hierop een uitzondering vormen, zijn QjoEndJournal en QjoStartJournal.
- Bij gebruik van de Qlg-API's (met behulp van voor taalondersteuning geschikte padnamen) is een nullteken niet toegestaan als teken in de padnaam, tenzij een nullteken is opgegeven als scheidingsteken.

Bestandsdescriptor

Bij gebruik van ILE C/400-stroom-I/O-functies zoals gedefinieerd door ANSI. Bij gebruik van C-functies van het geïntegreerd bestandssysteem identificeert u het bestand door een **bestandsdescriptor** op te geven. Een bestandsdescriptor is een positief geheel getal dat uniek moet zijn voor elke taak. De taak gebruikt een bestandsdescriptor om een open bestand te identificeren bij het uitvoeren van bewerkingen van dat bestand. De bestandsdescriptor wordt weergegeven door *files* in C-functies die werken op het geïntegreerd bestandssysteem en door de variabele *descriptor* in C-functies die werken op sockets.

Elke bestandsdescriptor verwijst naar een **open bestandsbeschrijving** die gegevens bevat zoals een bestandsoffset, bestandsstatus en toegangswerkstanden voor het bestand. Naar één open bestandsbeschrijving kan worden verwezen door meer dan één bestandsdescriptor, maar een bestandsdescriptor kan slechts verwijzen naar één open bestandsbeschrijving.



Figuur 11. Bestandsdescriptor en open bestandsbeschrijving

Als een ILE C/400stroom-I/O-functie wordt gebruikt met het geïntegreerd bestandssysteem converteert de C/400-runtime-ondersteuning de bestandspointer naar een bestandsdescriptor.

Bij gebruik van de bestandssystemen “hoofddirectory” (/), QOpenSys of een door de gebruiker gedefiniëerd bestandssysteem, kunt u de toegang tot een open bestandsbeschrijving doorgeven van de ene naar de andere taak. Hierdoor kan de taak toegang krijgen tot het bestand. U geeft de bestandsdescriptor door van de ene naar de andere taak met behulp van de functie `givedescriptor()` of `takedescriptor()`. Raadpleeg voor een beschrijving van deze functies de onderwerpen Sockets programmeren of Sockets APIs in het iSeries Informatiecentrum.

Beveiliging

Bij gebruik van de geïntegreerd bestandssysteem-API's kunt u de toegang tot objecten beperken, net als bij het gebruik van gegevensbeheerinterfaces. Bedenk wel dat tijdelijke machtigingen niet worden ondersteund. Een geïntegreerd bestandssysteem-API gebruikt de machtiging van het gebruikersprofiel waaronder de taak wordt uitgevoerd.

Elk bestandssysteem kan zijn eigen speciale machtigingsvereisten hebben. NFS-servertaken vormen de enige uitzondering op deze regel. Netwerkbestandssysteemopdrachten worden uitgevoerd onder het profiel van de gebruiker wiens gebruikers-ID door de NFS-server is ontvangen ten tijde van de opdracht.

Machtigingen op uw server zijn het equivalent van **permisies** op UNIX-systemen. De soorten machtigingen zijn lezen en schrijven voor een bestand of directory en uitvoeren voor een bestand en zoeken voor een directory. De machtigingen worden aangegeven door een reeks machtigingsbits die de "wijze van toegang" voorstellen van het bestand of de directory. U kunt de machtigingsbits wijzigen met de functies `chmod()` of `fchmod()` voor "werkstandwijziging". U kunt ook de functie `umask()` gebruiken om te bepalen welke bestandsmachtigingsbits worden ingesteld wanneer een taak een bestand maakt.

Uitgebreide informatie over gegevensbeveiliging en machtigingen vindt u in de uitgave Security —

Reference .

Socketondersteuning

Als uw toepassing de bestandssystemen "hoofddirectory" (/), QOpenSys of door een de gebruiker gedefinieerde bestandssysteem gebruikt, kunt u gebruik maken van de **lokale socketondersteuning** van het geïntegreerd bestandssysteem. Met een lokaal socketobject (objecttype *SOCKET) kunt u twee taken uitvoeren op hetzelfde systeem om een communicatieverbinding met elkaar tot stand te brengen.

Een van de taken brengt een verbindingspunt tot stand door middel van de C-functie `bind()` om een lokaal socketobject te maken. De andere taak geeft de naam van het lokale socketobject op bij de functie `connect()`, `sendto()` of `sendmsg()`. Deze functies en de concepten van sockets in het algemeen worden beschreven in het onderwerp Sockets programmeren in het iSeries Informatiecentrum.

Nadat de verbinding tot stand is gebracht, kunnen de taken gegevens naar elkaar zenden en ontvangen met behulp van geïntegreerd bestandssysteem-functies zoals `write()` en `read()`. Geen van de gegevens die worden verzonden gaat werkelijk door het socketobject. Het socketobject is slechts een ontmoetingspunt waar de twee taken elkaar kunnen vinden.

Wanneer de twee taken klaar zijn met communiceren, sluit elke taak de socketverbinding met de functie `close()`. Het lokale socketobject blijft aanwezig in het systeem totdat het wordt verwijderd met de functie `unlink()` of de opdracht RMVLNK (Koppeling verwijderen).

Een lokaal socketobject kan niet worden opgeslagen.

Naamgeving en internationale ondersteuning

De ondersteuning voor de bestandssystemen "hoofddirectory" (/) en QOpenSys zorgt ervoor dat de tekens in objectnamen constant blijven bij alle codeerschema's die worden gebruikt voor de verschillende talen en apparaten. Wanneer een objectnaam wordt doorgegeven aan het systeem wordt elk teken van de naam geconverteerd naar een 16-bits vorm waarin alle tekens standaard een gecodeerde gedaante hebben (zie "Naamcontinuïteit" op pagina 17). Wanneer de naam wordt gebruikt, wordt hij geconverteerd naar de code die voor de codetabel wordt gebruikt.

Als de codetabel waarnaar de naam wordt geconverteerd een teken dat in de naam wordt gebruikt niet bevat, wordt de naam als ongeldig afgewezen.

Omdat tekens constant blijven over alle codetabellen, kunt u beter geen bewerking uitvoeren in de veronderstelling dat een bepaald teken wel zal veranderen in een ander teken wanneer een bepaalde codetabel wordt gebruikt. U kunt er bijvoorbeeld niet van uitgaan dat een 'hekje' zal veranderen in een pondteken, ook al hebben ze in verschillende codetabellen dezelfde code.

Onthoud dat de namen van de uitgebreide kenmerken van een object op dezelfde manier worden geconverteerd als de naam van het object. Dezelfde overwegingen zijn dus hier ook van toepassing.

Raadpleeg voor meer informatie over codetabellen het onderwerp Globalization in het iSeries Informatiecentrum.

Gegevensconversie

Wanneer u bestanden opent via het geïntegreerd bestandssysteem worden gegevens in de bestanden al dan niet geconverteerd, afhankelijk van de openwerkstand die werd opgegeven toen het bestand werd geopend.

Een open bestand kan een van de twee volgende werkstanden hebben:

Binair De gegevens van het bestand worden gelezen en zonder conversie naar het bestand geschreven. De toepassing is verantwoordelijk voor de afhandeling van de gegevens.

Tekst De gegevens van het bestand worden gelezen en naar het bestand geschreven in de veronderstelling dat het in tekstvorm is. Als de gegevens uit het bestand worden gelezen, worden deze geconverteerd van het CCSID van het bestand naar het CCSID van de ontvangende toepassing of taak of het ontvangende systeem. Als gegevens naar het bestand worden geschreven, worden deze geconverteerd van het CCSID van de toepassing, de taak of het systeem naar het CCSID van het bestand. Bij echte stroombestanden worden eventuele regelindelingstekens (zoals terugloop, tab en bestandseinde) eenvoudig van het ene naar het andere CCSID geconverteerd.

Bij het lezen van recordbestanden die worden gebruikt als stroombestand, worden de tekens voor het einde van een regel (terugloop en regelopschuiving) toegevoegd aan het einde van de gegevens in elk record. Bij het schrijven naar recordbestanden:

- worden tekens voor het einde van een regel verwijderd;
- worden tabs vervangen door een corresponderend aantal spaties tot aan de volgende tab;
- worden lijnen opgevuld met spaties (voor een sectie van een fysiek bronbestand) of nullen (voor een sectie van een fysiek gegevensbestand).

Bij een openopdracht kan een van de volgende zaken worden opgegeven:

Binair, gedwongen

De gegevens worden binair verwerkt ongeacht de feitelijke inhoud van het bestand. De toepassing is verantwoordelijk voor de afhandeling van de gegevens.

Tekst, gedwongen

De gegevens worden beschouwd als tekst. De gegevens worden geconverteerd van het CCSID van het bestand naar het CCSID van de toepassing.

Een standaardwaarde van *Binair, gedwongen* wordt gebruikt voor de functie geïntegreerd bestandssysteem open().

Voorbeeld: C-functies Geïntegreerd bestandssysteem

Dit eenvoudige programma in C-taal laat het gebruik van een aantal functies van het geïntegreerd bestandssysteem zien. Het programma voert de volgende bewerkingen uit:

- 1** Gebruikt de functie **getuid()** om het echte gebruikers-ID (uid) vast te stellen.
- 2** Gebruikt de functie **getcwd()** om de actieve directory vast te stellen.

- 3** Gebruikt de functie **open()** om een bestand te maken. Het geeft de eigenaar (de gebruiker die het bestand heeft gemaakt) lees-, schrijf- en uitvoerrechten voor het bestand.
- 4** Gebruikt de functie **write()** om een bytereeks naar het bestand te schrijven. De bestandsdescriptor die bij het openen is verstrekt (**3**), kenmerkt het bestand.
- 5** Gebruikt de functie **close()** om het bestand te sluiten.
- 6** Gebruikt de functie **mkdir()** om in de actieve directory een nieuwe subdirectory te maken. De eigenaar krijgt lees-, schrijf- en uitvoerrechten voor de subdirectory.
- 7** Gebruikt de functie **chdir()** om van de nieuwe subdirectory te veranderen naar de actieve directory.
- 8** Gebruikt de functie **link()** om een koppeling te maken naar het bestand dat daarvoor is gemaakt (**3**).
- 9** Gebruikt de functie **open()** om het bestand te openen voor alleen lezen. De koppeling die gemaakt is in (**8**) geeft toegang tot het bestand.
- 10** Gebruikt de functie **read()** om een bytereeks uit het bestand te lezen. De bestandsdescriptor die bij het openen is verstrekt (**9**), kenmerkt het bestand.
- 11** Gebruikt de functie **close()** om het bestand te sluiten.
- 12** Gebruikt de functie **unlink()** om de koppeling naar het bestand te verwijderen.
- 13** Gebruikt de functie **chdir()** om de actieve directory terug te wijzigen in de bovenliggende directory waarin de nieuwe subdirectory is gemaakt.
- 14** Gebruikt de functie **rmdir()** om de subdirectory die daarvoor is gemaakt (**6**) te verwijderen.
- 15** Gebruikt de functie **unlink()** om het bestand dat daarvoor is gemaakt (**3**) te verwijderen.

Opmerking: Het voorbeeldprogramma wordt correct uitgevoerd op systemen waarop het CCSID van de taak waarin het wordt uitgevoerd 37 is. In de API's van het geïntegreerd bestandssysteem moeten objectnamen en paden gecodeerd zijn in het CCSID van de taak. Het C-compileerprogramma slaat tekenconstanten echter op in CCSID 37. Voor een volledige compatibiliteit dient u tekenconstanten, zoals objectnamen en paden, te converteren voordat API's worden doorgegeven naar het CCSID van de taak.

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>

#define BUFFER_SIZE      2048
#define NEW_DIRECTORY    "testdir"
#define TEST_FILE        "test.file"
#define TEST_DATA        "Hallo iedereen!"
#define USER_ID          "gebruikers_id_"
#define PARENT_DIRECTORY ".."

char InitialFile[BUFFER_SIZE];
char LinkName[BUFFER_SIZE];
char InitialDirectory[BUFFER_SIZE] = ".";
char Buffer[32];
int FilDes = -1;
int BytesRead;
int BytesWritten;
uid_t UserID;
```

```

void CleanUpOnError(int level)
{
    printf("Fout opgetreden, opschonen gestart.\n");
    switch ( level )
    {
        case 1:
            printf("Actieve kan niet worden opgehaald.\n");
            break;
        case 2:
            printf("Bestand %s kan niet worden gemaakt.\n",TEST_FILE);
            break;
        case 3:
            printf("Naar bestand %s kan niet worden geschreven.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 4:
            printf("Bestand %s kan niet worden gesloten.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 5:
            printf("Directory %s kan niet worden gemaakt.\n",NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 6:
            printf("Naar directory %s kan niet worden gewisseld.\n",NEW_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 7:
            printf("Koppeling %s naar %s kan niet worden gemaakt.\n",LinkName,InitialFile);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 8:
            printf("Koppeling %s kan niet worden geopend.\n",LinkName);
            unlink(LinkName);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 9:
            printf("Koppeling %s kan niet worden gelezen.\n",LinkName);
            close(FilDes);
            unlink(LinkName);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 10:
            printf("Koppeling %s kan niet worden gesloten.\n",LinkName);
            close(FilDes);
            unlink(LinkName);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 11:
            printf("Koppeling %s kan niet worden ontkoppeld.\n",LinkName);
            unlink(LinkName);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 12:

```

```

        printf("Naar directory %s kan niet worden gewisseld.\n",PARENT_DIRECTORY);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 13:
        printf("Directory %s kan niet worden verwijderd.\n",NEW_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 14:
        printf("Bestand %s kan niet worden ontkoppeld.\n",TEST_FILE);
        unlink(TEST_FILE);
        break;
    default:
        break;
}
printf("Programma foutief beëindigd.\n"
       "Alle testbestanden en -directory's zijn eventueel niet verwijderd.\n");
}

int main ()
{
    1
    /* Het echte gebruikers-ID ophalen en afdrucken met de functie getuid(). */
    UserID = getuid();
    printf("Het echte gebruikers-ID is %u. \n",UserID);

    2
    /* De actieve directory ophalen en opslaan in InitialDirectory. */
    if ( NULL == getcwd(InitialDirectory,BUFFER_SIZE) )
    {
        perror("getcwd Error");
        CleanupOnError(1);
        return 0;
    }
    printf("De actieve directory is %s.\n",InitialDirectory);

    3
    /* Het bestand TEST_FILE voor schrijven maken, als het nog niet bestaat.
       De eigenaar krijgt lees-, schrijf- en uitvoerrechten. */
    FilDes = open(TEST_FILE, O_WRONLY | O_CREAT | O_EXCL, S_IRWXU);
    if ( -1 == FilDes )
    {
        perror("open Error");
        CleanupOnError(2);
        return 0;
    }
    printf("%s gemaakt in directory %s.\n",TEST_FILE,InitialDirectory);

    4
    /* TEST_DATA schrijven naar TEST_FILE via FilDes */
    BytesWritten = write(FilDes,TEST_DATA,strlen(TEST_DATA));
    if ( -1 == BytesWritten )
    {
        perror("write Error");
        CleanupOnError(3);
        return 0;
    }
    printf("%s geschreven naar bestand %s.\n",TEST_DATA,TEST_FILE);

    5
    /* TEST_FILE sluiten via FilDes */
    if ( -1 == close(FilDes) )
    {
        perror("close Error");
        CleanupOnError(4);
    }
}

```

```

return 0;
}
FilDes = -1;
printf("Bestand %s gesloten.\n",TEST_FILE);

```

6

```

/* Nieuwe directory in de actieve directory maken en
de eigenaar lees-, schrijf- en uitvoerrechten verlenen */
if ( -1 == mkdir(NEW_DIRECTORY, S_IRWXU) )
{
    perror("mkdir Error");
    CleanupOnError(5);
    return 0;
}
printf("Directory %s gemaakt in directory %s.\n",NEW_DIRECTORY,InitialDirectory);

```

7

```

/* De actieve directory wijzigen in de
directory NEW_DIRECTORY die zojuist is gemaakt. */
if ( -1 == chdir(NEW_DIRECTORY) )
{
    perror("chdir Error");
    CleanupOnError(6);
    return 0;
}
printf("Gewisseld naar directory %s/%s.\n",InitialDirectory,NEW_DIRECTORY);

```

```

/* PARENT_DIRECTORY kopiëren naar InitialFile en
"/" en TEST_FILE toevoegen aan InitialFile. */
strcpy(InitialFile,PARENT_DIRECTORY);
strcat(InitialFile,"/");
strcat(InitialFile,TEST_FILE);

```

```

/* USER_ID kopiëren naar LinkName en vervolgens het
gebruikers-ID als tekenreeks toevoegen aan LinkName. */
strcpy(LinkName, USER_ID);
sprintf(Buffer, "%d\0", (int)UserID);
strcat(LinkName, Buffer);

```

8

```

/* Een koppeling maken naar de naam van InitialFile met de LinkName. */
if ( -1 == link(InitialFile,LinkName) )
{
    perror("link Error");
    CleanupOnError(7);
    return 0;
}
printf("Koppeling %s naar %s gemaakt.\n",LinkName,InitialFile);

```

9

```

/* Het bestand LinkName openen voor alleen lezen. */
if ( -1 == (FilDes = open(LinkName,O_RDONLY)) )
{
    perror("open Error");
    CleanupOnError(8);
    return 0;
}
printf("%s geopend voor lezen.\n",LinkName);

```

10

```

/* Gelezen uit het bestand LinkName via FilDes naar Buffer. */
BytesRead = read(FilDes,Buffer,sizeof(Buffer));
if ( -1 == BytesRead )
{
    perror("read Error");
    CleanupOnError(9);
    return 0;
}

```

```

    }
    printf("Read %s from %s.\n",Buffer,LinkName);
    if ( BytesRead != BytesWritten )
    {
        printf("WAARSCHUWING: het aantal gelezen bytes is "\
              "niet gelijk aan het aantal geschreven bytes.\n");
    }

```

11

```

/* Het bestand LinkName sluiten via FilDes. */
if ( -1 == close(FilDes) )
{
    perror("close Error");
    CleanUpOnError(10);
    return 0;
}
FilDes = -1;
printf("Closed %s.\n",LinkName);

```

12

```

/* De koppeling LinkName naar InitialFile ont koppelen. */
if ( -1 == unlink(LinkName) )
{
    perror("unlink Error");
    CleanUpOnError(11);
    return 0;
}
printf("%s is ontkoppeld.\n",LinkName);

```

13

```

/* De actieve directory weer wijzigen
in de begindirectory. */
if ( -1 == chdir(PARENT_DIRECTORY) )
{
    perror("chdir Error");
    CleanUpOnError(12);
    return 0;
}
printf("Directory wordt gewijzigd in %s.\n",InitialDirectory);

```

14

```

/* De directory NEW_DIRECTORY verwijderen */
if ( -1 == rmdir(NEW_DIRECTORY) )
{
    perror("rmdir Error");
    CleanUpOnError(13);
    return 0;
}
printf("Directory %s wordt verwijderd.\n",NEW_DIRECTORY);

```

15

```

/* Het bestand TEST_FILE ont koppelen */
if ( -1 == unlink(TEST_FILE) )
{
    perror("unlink Error");
    CleanUpOnError(14);
    return 0;
}
printf("Bestand %s wordt ontkoppeld.\n",TEST_FILE);

printf("Programma goed voltooid.\n");
return 0;
}

```

Opmerking: Lees de “Informatie over uitsluiting van garanties voor code” op pagina 1 voor belangrijke juridische informatie.

Werken met bestanden en mappen met behulp van iSeries Navigator

U kunt met bestanden en mappen de volgende taken uitvoeren:

- “Een bestand inchecken”
- “Een bestand uitchecken”
- “Een map maken”
- “Een map verwijderen” op pagina 115
- “Bestanden en mappen naar een andere bestandssysteem verplaatsen” op pagina 115
- “Instellingsmachtigingen” op pagina 116
- “Bestandstekstconversie instellen” op pagina 116
- “Een bestand of map naar een ander systeem verzenden” op pagina 117
- “Opties wijzigen voor een pakketdefinitie” op pagina 117
- “Een datum en tijd plannen om uw bestand of map te verzenden” op pagina 118
- “Een bestandsshare maken” op pagina 118
- “Een bestandsshare wijzigen” op pagina 118
- “Hiermee stelt u in of objecten moeten worden gescand” op pagina 118

Een bestand inchecken

U checkt een bestand als volgt in:

1. Klik in **iSeries Navigator** met de rechtermuisknop op het bestand dat u wilt inchecken.
2. Selecteer **Eigenschappen**.
3. Selecteer **Bestandseigenschappen** -> **Pagina gebruiken**.
4. Klik op **Inchecken**.

Als u liever sleept, kunt u een UDFS aankoppelen door het naar een map te slepen in het geïntegreerd bestandssysteem op dezelfde server. U kunt het UDFS niet naar /dev, /dev/QASPxx, /dev/asp_name, een ander systeem of het werkblad slepen.

Een bestand uitchecken

U checkt een bestand als volgt uit:

1. Klik in **iSeries Navigator** met de rechtermuisknop op het bestand dat u wilt uitchecken.
2. Selecteer **Eigenschappen**.
3. Selecteer **Bestandseigenschappen** -> **Pagina gebruiken**.
4. Klik op **Uitchecken**.

Een map maken

U maakt als volgt een map:

1. Klik op het systeem dat u wilt gebruiken in **iSeries Navigator**.
2. Klik op **Bestandssystemen**.
3. Klik op **IFS (Integrated File System)**.
4. Klik met de rechtermuisknop op het bestandssysteem waaraan u de nieuwe map wilt toevoegen en selecteer **Nieuwe map**.
5. Typ in het dialoogvenster **Nieuwe map** de nieuwe naam voor het object.
6. Klik op **OK**.

| U moet ook in overweging nemen of er op de objecten die u in deze map maakt, scans moeten kunnen
| worden uitgevoerd. Voor meer informatie over dit onderwerp raadpleegt u “Hiermee stelt u in of objec-
| ten moeten worden gescand” op pagina 118.

- | Als u een map maakt op de iSeries-server moet u beslissen of u de nieuwe map (of het object) wilt
- | beschermen met journaalbeheer. Zie het onderwerp Journaalbeheer voor meer informatie.

Verwante onderwerpen:

- Journaal starten
- Journaal beëindigen

Een map verwijderen

U verwijdert een map als volgt:

1. Klik op het systeem dat u wilt gebruiken in **iSeries Navigator**.
2. Klik op **Bestandssystemen**.
3. Klik op **IFS (Integrated File System)**. Klik net zo lang verder totdat de map of het bestand dat u wilt verwijderen zichtbaar is.
4. Klik met de rechtermuisknop op het bestand of de map en selecteer **Wissen**.

Bestanden en mappen naar een andere bestandssysteem verplaatsen

Elk bestandssysteem heeft zijn eigen unieke kenmerken. Het verplaatsen van een object naar een ander bestandssysteem kan dus betekenen dat u de voordelen verliest van het bestandssysteem waarin het object op dit moment is opgeslagen. Misschien wilt u een object naar een bepaald bestandssysteem verplaatsen om juist te kunnen profiteren van bepaalde kenmerken. Voordat u objecten gaat verplaatsen naar een ander bestandssysteem dient u inzicht te krijgen in de bestandssystemen op het geïntegreerd bestandssysteem en hun kenmerken. Raadpleeg voor meer informatie het onderwerp Werken met bestandssystemen.

Houd ook rekening met het volgende:

- Gebruikt u toepassingen die gebruik maken van de voordelen van het bestandssysteem waarin het object zich bevindt?

Sommige bestandssystemen ondersteunen interfaces die geen deel uitmaken van de ondersteuning van het geïntegreerd bestandssysteem. Toepassingen die deze interfaces gebruiken, kunnen misschien geen toegang meer krijgen tot objecten die zijn verplaatst naar een ander bestandssysteem. De bestandssystemen QDLS en QOPT ondersteunen bijvoorbeeld het hiërarchisch bestandssysteem. API's en opdrachten werken met document- en mapobjecten. U kunt deze interfaces niet gebruiken met objecten die zich in andere bestandssystemen bevinden.

- Welke objectkenmerken zijn voor u van belang?

Niet alle kenmerken worden ondersteund door alle bestandssystemen. Met de bestandssystemen QSYS.LIB en onafhankelijke ASP QSYS.LIB kunt u bijvoorbeeld slechts enkele uitgebreide kenmerken opslaan en herstellen, terwijl u met de bestandssystemen "hoofddirectory" (/) en QOpenSys alle uitgebreide kenmerken kunt opslaan en herstellen. Daarom zijn QSYS.LIB en onafhankelijke ASP QSYS.LIB geen goede kandidaten als u object wilt opslaan met uitgebreide kenmerken.

Goede kandidaten voor verplaatsen zijn de PC-bestanden die worden opgeslagen in QDLS. De meeste PC-toepassingen kunnen doorwerken met PC-bestanden die zijn verplaatst van QDLS naar andere bestandssystemen. De bestandssystemen hoofddirectory (/), QOpenSys, QNetWare en QNTC zijn geschikt om deze PC-bestanden in op te slaan. Omdat ze veel bestandssysteemkenmerken van OS/2 ondersteunen, bieden deze bestandssystemen snellere toegang tot de bestanden. Zie "Vergelijking bestandssystemen" op pagina 24 voor een vergelijking van de verschillende kenmerken van elk bestandssysteem.

Om objecten te verplaatsen naar een ander bestandssysteem voert u de volgende stappen uit:

1. Sla een kopie op van alle objecten die u wilt verplaatsen.

Door reservekopieën te maken, kunt de objecten naar het oorspronkelijke bestandssysteem terugzetten voor het geval het bestandssysteem waar u ze naar toe verplaatst ze niet kan openen.

Opmerking: U kunt objecten niet in het ene bestandssysteem opslaan en ze naar een ander bestandssysteem terugzetten.

2. Maak met behulp van de opdracht CRTDIR (Directory maken) de directory's in het bestandssysteem waar u de objecten naar toe wilt verplaatsen.

Bekijk de kenmerken van de directory waar de objecten zich nu bevinden goed en bepaal of u die kenmerken wilt kopiëren naar de nieuwe directory's. Bijvoorbeeld, de gebruiker die de nieuwe directory maakt, is daar de eigenaar van, en niet de gebruiker die eigenaar is van de oude directory. U wilt misschien het eigendom naar de nieuwe directory overbrengen als u die hebt gemaakt, mits het bestandssysteem het instellen van de eigenaar van een directory ondersteunt.

3. Verplaats de bestanden naar het gekozen bestandssysteem met de opdracht MOV (Verplaatsen).

MOV wordt aanbevolen omdat met deze opdracht het eigendom van de objecten wordt gehandhaafd, mits het bestandssysteem het instellen van het eigendom van objecten ondersteunt. U kunt het eigendom van objecten ook handhaven met de parameter OWNER(*KEEP) van de opdracht CPY (Kopiëren). Vergeet niet dat dit alleen werkt op bestandssystemen die het instellen van het eigendom van objecten ondersteunen. Denk bij het gebruik van MOV of CPY aan het volgende:

- Kenmerken kunnen niet overeenkomen en dientengevolge worden gewist.
- Uitgebreide kenmerken kunnen worden gewist.
- Machtigingen kunnen een lager niveau hebben en worden gewist.

Dit betekent dat u met het oog op mogelijk gewiste kenmerken en machtigingen niet wilt volstaan met de opdracht verplaatsen of kopiëren als u een object wilt terugzenden naar zijn oorspronkelijke bestandssysteem. De veiligste manier om een object terug te zetten, is het te herstellen met behulp van een opgeslagen versie ervan.

Instellingsmachtigingen

Door machtigingen toe te voegen aan een object, controleert u de mate waarin anderen met dat object kunnen werken. Met machtigingen kunt u sommige gebruikers het recht geven bepaalde objecten alleen te bekijken, terwijl andere gebruikers ze ook mogen bewerken.

U stelt machtigingen voor een bestand of map als volgt in:

1. Klik op het systeem dat u wilt gebruiken in het venster **iSeries Navigator**.
2. Klik op **Bestandssystemen**.
3. Klik op **IFS (Integrated File System)**. Klik net zo lang verder totdat u het object ziet waaraan u machtigingen wilt toevoegen.
4. Klik met de rechtermuisknop op het object waaraan u machtigingen wilt toevoegen en selecteer **Machtigingen**.
5. Klik in het dialoogvenster **Machtigingen op Toevoegen**.
6. Selecteer een of meer gebruikers en groepen of geef een gebruikers- of groepsnaam op in het naamveld in het dialoogvenster **Toevoegen**.
7. Klik op **OK**. De gebruikers en/of groepen worden bovenaan in de lijst toegevoegd.
8. Klik op de knop **Details** voor uitgebreidere machtigingen.
9. Breng de machtigingen die u de gebruiker wilt geven aan door de betreffende selectievakjes te selecteren.
10. Klik op **OK**.

Bestandstekstconversie instellen

In kunt in iSeries Navigator automatische bestandstekstconversie instellen. Met automatische bestandstekstconversie kunt u extensies gebruiken voor bestandsgegevensconversie. Het geïntegreerd bestandssysteem kan een gegevensbestand converteren als het wordt verzonden tussen de iSeries en een PC. Als u een gegevensbestand opent vanaf een PC wordt het verwerkt alsof het in ASCII staat.

U stelt bestandstekstconversie als volgt in:

1. Klik op het systeem dat u wilt gebruiken in **iSeries Navigator**.
2. Klik op **Bestandssystemen**.
3. Klik met de rechtermuisknop op **IFS (Integrated File System)** en selecteer **Eigenschappen**.
4. Geef de extensie op die u automatisch wilt converteren in het tekstvak **Extensies voor bestandstekstconversie** en klik op **Toevoegen**.
5. Herhaal stap 4 voor alle extensies die u automatisch wilt converteren.
6. Klik op **OK**.

Een bestand of map naar een ander systeem verzenden

U verzendt een bestand of map als volgt naar een ander systeem:

1. Klik op het systeem dat u wilt gebruiken in **iSeries Navigator**.
2. Klik op **Bestandssystemen**.
3. Klik op **IFS (Integrated File System)**. Klik net zo lang verder totdat de map of het bestand dat u wilt verzenden zichtbaar is.
4. Klik met de rechtermuisknop op het bestand of de map en selecteer **Verzenden**. Het bestand of de map verschijnt in de lijst Geselecteerde bestanden en mappen van het dialoogvenster **Bestanden verzenden vanaf**.
5. Klik op de lijst met beschikbare systemen en groepen.
6. Selecteer een systeem en klik op **Toevoegen** om het systeem toe te voegen aan de lijst **Doelsystemen en -groepen**. Herhaal deze stap voor alle systemen waar u dit bestand of deze map naar toe wilt verzenden.
7. Klik op **OK** om het bestand of de map te verzenden met de standaardpakketdefinities en -planninggegevens.

U kunt ook "Opties wijzigen voor een pakketdefinitie" of "Een datum en tijd plannen om uw bestand of map te verzenden" op pagina 118.

Als u een pakketdefinitie maakt, wordt deze opgeslagen en kan deze op elk moment worden gebruikt om de opgegeven set bestanden en mappen te verzenden naar meerdere eindpuntsystemen en -systeemgroepen. Met een momentopname van uw bestanden kunt u van dezelfde set bestanden meer dan een versie bewaren. Als u een momentopname verzendt, weet u zeker dat de bestanden tijdens de distributie niet worden bijgewerkt, zodat het laatste doelsysteem precies dezelfde objecten ontvangt als het eerste doelsysteem.

Opties wijzigen voor een pakketdefinitie

Met een pakketdefinitie kunt u een reeks OS/400-objecten of bestanden uit het geïntegreerd bestandssysteem groeperen. De pakketdefinitie biedt u ook de mogelijkheid om dezelfde groep bestanden af te beelden als een logische set (of een fysieke set) door een momentopname te maken van de bestanden, zodat deze worden behouden voor distributie op een later tijdstip.

U wijzigt de opties van de pakketdefinities als volgt:

1. Voer de stappen uit van het onderwerp "Een bestand of map naar een ander systeem verzenden".
2. Klik op het tabblad **Opties**. De standaardopties zijn het bijvoegen van submappen bij het maken van een pakket en het vervangen van een bestaand bestand door het bestand dat wordt verzonden.
3. Wijzig deze opties desgewenst.
4. Klik op **Geavanceerd** om de geavanceerde opties voor opslaan en herstellen in te stellen.
5. Klik op **OK** om de geavanceerde opties op te slaan.
6. Klik op **OK** om het bestand te verzenden of op **Planning** om een tijd in te stellen voor het verzenden van het bestand.

Verwante onderwerpen:

- “Een datum en tijd plannen om uw bestand of map te verzenden”.

Een datum en tijd plannen om uw bestand of map te verzenden

Door de plannerfunctie te gebruiken krijgt u de flexibiliteit om uw werk te doen op het moment dat het u uitkomt. U plant als volgt een datum en tijd om uw bestand of map te verzenden:

1. Voer de stappen uit van het onderwerp “Een bestand of map naar een ander systeem verzenden” op pagina 117.
2. Klik op **Planning**.
3. Selecteer het tijdstip waarop u het bestand of de map wilt verzenden.

Een bestandsshare maken

Een bestandsshare is een pad dat iSeries NetServer gemeenschappelijk heeft met PC-clients op het iSeries-netwerk. Een bestandsshare kan bestaan uit elke geïntegreerd bestandssysteem-directory op de iSeries.

U maakt als volgt een bestandsshare:

1. Klik op uw systeem in **iSeries Navigator**.
2. Klik op **Bestandssystemen**.
3. Klik op **IFS (Integrated File System)**.
4. Klik op het bestandssysteem dat de map bevat waarvoor u een bestandssysteem wilt maken.
5. Klik met de rechtermuisknop op de map waarvoor u een share wilt maken en selecteer **Gemeenschappelijk gebruiken**.
6. Selecteer **Nieuwe share**.

Een bestandsshare wijzigen

Een bestandsshare is een pad dat iSeries NetServer gemeenschappelijk heeft met PC-clients op het iSeries-netwerk. Een bestandsshare kan bestaan uit elke geïntegreerd bestandssysteem-directory op de iSeries.

U wijzigt als volgt een bestandsshare:

1. Klik op uw systeem in **iSeries Navigator**.
2. Klik op **Bestandssystemen**.
3. Klik op **IFS (Integrated File System)**.
4. Klik op de map met de share die u wilt wijzigen.
5. Klik met de rechtermuisknop op de map waarvoor de share is gedefinieerd die u wilt selecteren voor **Gemeenschappelijk gebruik**.
6. Selecteer **Nieuwe share**.

| Hiermee stelt u in of objecten moeten worden gescand

| Of objecten moeten worden gescand, stelt u als volgt in:

- | 1. Klik op uw systeem in **iSeries Navigator**.
- | 2. Klik op **Bestandssystemen**.
- | 3. Klik op **IFS (Integrated File System)**.
- | 4. Klik op de gewenste map of het gewenste bestand.
- | 5. Klik met de rechtermuisknop op de map of het bestand en selecteer **Eigenschappen**
- | 6. Selecteer het tabblad **Beveiliging**.
- | 7. Selecteer **Objecten scannen** met de optie die u wilt.

| Kijk hieronder voor meer informatie over de opties. De beschrijving van deze opties gelden voor bestanden. Alleen bestanden kunnen worden gescand. Voor mappen en door de gebruiker gedefinieerde bestandssystemen kunt u opgeven welke scankenmerken bestanden krijgen die in die map of dat UDFS worden gemaakt.

- | • Ja

| Het object wordt gescand volgens de regels die zijn beschreven in de exitprogramma's die betrekking hebben op scannen voor het geval het object is gewijzigd of de scansoftware is bijgewerkt sinds de laatste keer dat het object werd gescand.

- | • Nee

| Het object wordt niet gescand door het exitprogramma dat betrekking heeft op scannen.

| **Opmerking:** Als de optie 'Scannen bij volgende poging tot toegang nadat het object is hersteld' is geselecteerd in de systeemwaarden wanneer een object met dit kenmerk wordt hersteld, wordt het object na het herstellen minstens één keer gescand.

- | • Alleen als het object is gewijzigd

| Het object wordt alleen gescand volgens de regels die zijn beschreven in de exitprogramma's die betrekking hebben op scannen, als het object is gewijzigd sinds de laatste keer dat het object werd gescand. Het wordt niet gescand als de scansoftware wordt bijgewerkt.

| Als de systeemwaarde 'Alleen als het object is gewijzigd' **niet** is opgegeven, wordt dit kenmerk niet gebruikt en wordt het object gescand als het is gewijzigd of als de scansoftware aangeeft dat er een update heeft plaatsgevonden.

| **Opmerking:** Op dit tabblad voor bestanden kunt u ook de scanstatus van een object bepalen.

| **Opmerking:** Als de optie 'Scannen bij volgende poging tot toegang nadat het object is hersteld' is geselecteerd in de systeemwaarden wanneer een object met dit kenmerk wordt hersteld, wordt het object na het herstellen minstens één keer gescand.

Aanvullende informatie met betrekking tot geïntegreerde bestandssystemen

De volgende onderwerpen bevatten aanvullende informatie met betrekking tot het geïntegreerd bestandssysteem:

- "Transportonafhankelijke RPC"
- "Verwijzingen" op pagina 124
- "Ervaringsrapporten" op pagina 125

Transportonafhankelijke RPC

RPC (procedureaanroep op afstand) is ontwikkeld door Sun Microsystems en is in staat op eenvoudige wijze clienttoepassingen van een servermechanisme te scheiden en te distribueren. Het bevat een standaard voor XDR (externe gegevensweergave) waardoor meer dan één type machine toegang kan krijgen tot verzonden gegevens. Transportonafhankelijke RPC (TI-RPC) is de laatste versie van RPC. Het biedt een methode voor het isoleren van het onderliggende protocol dat wordt gebruikt in de netwerklaag, waardoor een naadloze overgang van het ene protocol naar het andere wordt bereikt. De enige protocollen die op dit moment beschikbaar zijn op de iSeries-server zijn TCP en UDP.

Het ontwikkelen van gedistribueerde toepassingen verspreid over een netwerk is bij gebruik van RPC een naadloze taak. De belangrijkste doelen zijn die toepassingen die neigen naar het distribueren van de gebruikersinterface of het ophalen van gegevens.

Raadpleeg voor meer informatie over transportonafhankelijke RPC's een van de volgende onderwerpen:

- "Netwerkselecties" op pagina 120

- “Vertaling naam naar adres”
- “Externe gegevensrepresentatie (XDR)”
- “Verificatie” op pagina 122
- “Transportonafhankelijke RPC (TI-RPC)” op pagina 122

Netwerkselecties

De volgende API's bieden de mogelijkheid om het transport te kiezen waarmee een toepassing wordt uitgevoerd.

Voor deze API's moet op het systeem het bestand *STMF /etc/netconfig bestaan. Als het netconfig-bestand niet in de directory /etc bestaat, moet de gebruiker dit kopiëren uit de directory /QIBM/ProdData/OS400/RPC. Het netconfig-bestand bevindt zich altijd in de directory /QIBM/ProdData/OS400/RPC.

API	Beschrijving
endnetconfig()	Geeft de pointer vrij voor de records in het netconfig-bestand
freenetconfigent()	Geeft de netconfig-structuur die wordt teruggezonden na de oproep vrij voor de functie getnetconfigent()
getnetconfig()	Zendt de pointer terug naar het actuele record in het netconfig-bestand en verhoogt deze pointer naar het volgende record
getnetconfigent()	Zendt de pointer terug naar de netconfig-structuur die overeenkomt met het invoer-netid
setnetconfig()	Initialiseert de recordpointer op het eerste gegeven in het netconfig-bestand. De functie setnetconfig() moet worden gebruikt voordat u de functie getnetconfig() voor de eerste keer gebruikt. De functie setnetconfig() zendt een unieke handle (een pointer naar de records die zijn opgeslagen in het netconfig-bestand) terug voor gebruik door de functie getnetconfig().

Vertaling naam naar adres

Met de onderstaande API's kan een toepassing het adres van een service of een opgegeven host verkrijgen op een van het transport onafhankelijke manier.

API	Beschrijving
netdir_free()	Maakt structuren vrij die zijn toegewezen aan API's voor de conversie van naam naar adres
netdir_getbyaddr()	Definieert adressen in hostnamen en servicenamen
netdir_getbyname()	Wijst de hostnaam en servicenaam die zijn opgegeven in de serviceparameter toe aan een set adressen die consistent zijn met het transport dat is opgegeven in de netconfig-structuur
netdir_options()	Verleent interfaces aan transportspecifieke mogelijkheden zoals het broadcastadres en de gereserveerde poortvoorzieningen van TCP en UDP
netdir_spperror()	Verzendt een statusbericht dat aangeeft waarom een van de API's voor de conversie van naam naar adres is mislukt
taddr2uaddr()	Vertaalt een transportspecifiek (lokaal) adres naar een transportonafhankelijk (universeel) adres
uaddr2taddr()	Vertaalt een transportonafhankelijk (universeel) adres naar een transportspecifiek (lokaal) adres (netbuf-structuur)

Externe gegevensrepresentatie (XDR)

Met de volgende API's kunnen RPC-toepassingen willekeurige gegevensstructuren verwerken, ook als er voor de hosts verschillende bytevolgorden of structuurindelingsafspraken gelden.

API	Beschrijving
xdr_array()	Een filterelement dat vertaalt tussen arrays met variabele lengte en hun corresponderende externe representaties. Deze functie wordt aangeroepen om elk element van de array te coderen of te decoderen
xdr_bool()	Een filterelement dat vertaalt tussen booleaanse operators (C geheel getal) en hun corresponderende externe representaties. Bij het coderen van deze gegevens produceert dit filter de waarden 1 of 0.
xdr_bytes()	Een filterelement dat vertaalt tussen tellerbytearrays en hun externe representaties. De functie bewerkt een subset van generieke arrays waarvan bekend is dat de omvang 1 is en de externe beschrijving van elk element is ingebouwd. De lengte van de bytereeks is expliciet te vinden in de geheel getal zonder plus- of minteken. De bytevolgorde eindigt niet met een nullteken. De externe representatie van de bytes is gelijk aan de interne representatie.
xdr_char()	Een filterelement dat vertaalt tussen C-tekens en hun externe representaties.
xdr_double()	Een filterelement dat vertaalt tussen dubbele precisiegetallen in C en hun externe representaties.
xdr_double_char()	Een filterelement dat vertaalt tussen 2-byte-tekens in C en hun externe representaties.
xdr_enum()	Een filterelement dat vertaalt tussen opsommingen in C (enum) en hun externe representaties
xdr_free()	Maakt het object waarnaar door de ingevoerde pointer wordt verwezen recursief vrij
xdr_float()	Een filterelement dat vertaalt tussen drijvende kommagetallen in C (genormaliseerde enkele drijvende kommagetallen) en hun externe representaties.
xdr_int()	Een filterelement dat vertaalt tussen hele getallen in C en hun externe representaties.
xdr_long()	Een filterelement dat vertaalt tussen grote gehele getallen in C en hun externe representaties.
xdr_netobj()	Een filterelement dat vertaalt tussen ondoorzichtige gegevens met een variabele lengte en hun externe representaties.
xdr_opaque()	Een filterelement dat vertaalt tussen ondoorzichtige gegevens met een vaste lengte en hun externe representaties.
xdr_pointer()	Verleent pointer-chasing binnen structuren en serialiseert nullpointers. Kan recursieve gegevens zoals binaire boomstructuren en gekoppelde lijsten vertegenwoordigen.
xdr_reference()	Een filterelement dat pointer-chasing binnen structuren verleent. Met dit element is serialiseren, deserialiseren en het bevrijden van pointers binnen één structuur waarnaar wordt verwezen door een andere structuur toegestaan. De functie xdr_reference() hecht geen bijzondere betekenis aan een nullpointer tijdens het serialiseren en het doorgeven van het adres van een nullpointer kan een geheugenfout tot gevolg hebben. Daarom moet de programmeur gegevens beschrijven met een tweezijdige discriminantvereniging. De ene kant wordt gebruikt als de pointer geldig is, de andere als de pointer null is.
xdr_short()	Een filterelement dat vertaalt tussen korte gehele getallen in C en hun externe representaties.
xdr_string()	Een filterelement dat vertaalt tussen strings in C en hun corresponderende externe representaties.
xdr_u_char()	Een filterelement dat vertaalt tussen C-tekens zonder plus- of minteken en hun externe representaties.
xdr_u_int()	Een filterelement dat vertaalt tussen gehele getallen in C zonder plus- of minteken en hun externe representaties.

API	Beschrijving
xdr_u_long()	Een filterelement dat vertaalt tussen grote gehele getallen in C zonder plus- of minteken en hun externe representaties.
xdr_u_short()	Een filterelement dat vertaalt tussen korte gehele getallen in C zonder plus- of minteken en hun externe representaties.
xdr_union()	Een filterelement dat vertaalt tussen discriminantverenigingen en hun corresponderende externe representaties.
xdr_vector()	Een filterelement dat vertaalt tussen arrays met vaste lengte en hun corresponderende externe representaties.
xdr_void()	Heeft geen parameters. Deze wordt doorgegeven naar andere RPC-functies die een parameter vereisen, maar verzendt geen gegevens
xdr_wrapstring()	Een element dat de API xdr_string(xdr, sp, maxuint) oproept, waarbij maxuint de maximumwaarde is van een geheel getal zonder plus- of minteken. De functie xdr_wrapstring() is nuttig omdat het RPC-pakket maximaal twee XDR-functies doorgeeft als parameters, terwijl de functie xdr_string() er drie vereist.

Verificatie

De volgende API's verlenen verificatie aan transportonafhankelijke de RPC-toepassingen (TI-RPC).

API	Beschrijving
auth_destroy()	Vernietigt de gegevensstructuur van de verificatie waarnaar wordt verwezen door de parameter auth
authnone_create()	Maakt een standaard RPC-verificatiehandle die nullverificatiegegevens doorgeeft met elke RPC en zendt deze terug.
authsys_create()	Maakt een RPC-verificatiehandle die verificatiegegevens bevat en zendt deze terug

Transportonafhankelijke RPC (TI-RPC)

De volgende API's bieden een gedistribueerde ontwikkelomgeving voor toepassingen doordat ze de toepassing isoleren van specifieke transportfuncties. Dit vergroot de gebruikersvriendelijkheid van de transports.

Raadpleeg voor meer informatie over transportonafhankelijke RPC's een van de volgende onderwerpen:

- "Vereenvoudigde API's voor TI-RPC"
- "API's voor TI-RPC van topniveau" op pagina 123
- "API's voor TI-RPC van middenniveau" op pagina 123
- "API's voor TI-RPC van expertniveau" op pagina 123
- "Andere API's voor TI-RPC" op pagina 123

Vereenvoudigde API's voor TI-RPC:

Met de volgende vereenvoudigde API's kunt u het te gebruiken type transport opgeven. Toepassingen die dit niveau gebruiken, hoeven niet expliciet handles te maken.

API	Beschrijving
rpc_call()	Een procedure op afstand oproepen op het opgegeven systeem
rpc_reg()	Een procedure registreren met RPC-servicepakket

API's voor TI-RPC van topniveau: Met de volgende API's kan de toepassing het type transport opgeven.

API	Beschrijving
clnt_call()	Een procedure op afstand oproepen die bij de client hoort
clnt_control()	Gegevens van een clientobject wijzigen
clnt_create()	Een generieke clienthandle maken clnt_destroy() De RPC-handle van de client vernietigen
clnt_destroy()	De RPC-handle van de client vernietigen
svc_create()	Een server-handle maken
svc_destroy()	Een RPC-servicetransporthandle vernietigen

API's voor TI-RPC van middenniveau:

De volgende API's zijn gelijk aan de API's van topniveau, met het verschil dat de gebruikerstoepassingen de transportspecifieke gegevens selecteren met behulp van netwerkselectie-API's:

API	Beschrijving
clnt_tp_create()	Een client-handle maken
svc_tp_create()	Een server-handle maken

API's voor TI-RPC van expertniveau: Met de volgende API's kan de toepassing het te gebruiken type transport selecteren. Ze bieden tevens een betere besturing van de details van de handles CLIENT en SVCXPRT. Deze API's zijn gelijk aan de API's van middenniveau, maar met extra besturing dankzij het gebruik van API's die namen naar adressen converteren.

Extra besturing dankzij het gebruik van API's die namen naar adressen converteren.

API	Beschrijving
clnt_tli_create()	Een client-handle maken
rpcb_getaddr()	Het algemene adres van een service zoeken
rpcb_set()	Het serveradres met de RPCbind registreren
rpcb_unset()	Gebruikt door de server om hun adressen te deregistreren
svc_reg()	Programma en versie bij verzending koppelen
svc_tli_create()	Een server-handle maken
svc_unreg()	Een koppeling wissen die is ingesteld door svc_reg()






Andere API's voor TI-RPC: Met deze API's kunnen de diverse toepassingen samenwerken met de vereenvoudigde API's en de API's van middenniveau, topniveau en expertniveau.



API	Beschrijving
clnt_freeres()	Gegevens vrijmaken die zijn toegewezen door het RPC- of XDR-systeem
clnt_geterr()	De foutstructuur ophalen van de clienthandle
svc_freeargs()	Gegevens vrijmaken die zijn toegewezen door het RPC- of XDR-systeem
svc_getargs()	Decodeert de argumenten van een RPC-opdracht
svc_getrpccaller()	Het netwerkadres van de initiatiefnemer ophalen
svc_run()	Wacht tot RPC-opdracht gearriveerd is

API	Beschrijving
svc_sendreply()	Verzendt de resultaten van een procedureaanroep naar een client op afstand
svcerr_decode()	Verzendt gegevens naar client om fout te decoderen
svcerr_noproc()	Verzendt gegevens naar client voor procedurenummerfout
svcerr_systemerr()	Verzendt gegevens naar client voor systeemfout

Verwijzingen

Deze bibliografie bevat uitgaven over de iSeries-server met achtergronden of verdere informatie bij de onderwerpen die in dit boek aan de orde komen.

- In het onderwerp CL onder **Programmeren** in het iSeries Informatiecentrum vindt u een beschrijving van de opdrachttaal (CL) van iSeries en de bijbehorende opdrachten. Elke opdrachtbeschrijving bevat een syntaxisdiagram, parameters, standaardwaarden, trefwoorden en een voorbeeld.
- In het onderwerp Globalization in het iSeries Informatiecentrum wordt verder ingegaan op verschillende aspecten van taalondersteuning, zoals tekenset en codetabel. Hier vindt u ook de informatie die u nodig hebt om de mogelijkheden voor taalondersteuning en meertaligheid van de iSeries-server vast te stellen en te plannen en gebruiken.
- Het onderwerp API's onder **Programmeren** van het iSeries Informatiecentrum bevat een beschrijving van elke OS/400-API, met inbegrip van API's van het geïntegreerd bestandssysteem, alsmede exit points en programma's van het geïntegreerd bestandssysteem.
- In het onderwerp Journaalbeheer onder **Systeembeheer** in het iSeries Informatiecentrum vindt u informatie over hoe u SMAPP (door het systeem beheerde beveiliging van toegangspaden), lokale journaals en journaals op afstand kunt instellen en beheren en problemen daarmee kunt oplossen op een iSeries-server.
- In het onderwerp Commitment Control onder **Database** in het iSeries Informatiecentrum wordt uitgelegd hoe u een groep wijzigingen aan resources, zoals databasebestanden of geïntegreerd bestandssysteem-bestanden, kunt definiëren en verwerken als een logische eenheid.
- OS/400 Network File System Support  In dit boek wordt het netwerkbestandssysteem beschreven aan de hand van praktijktoepassingen. U treft er onder meer informatie aan over exporteren, aankoppelen, bestandsvergrendeling en beveiligingsoverwegingen. In dit boek kunt u leren hoe u met gebruik van het netwerkbestandssysteem een veilige netwerknaamruimte kunt creëren en ontwikkelen.
- Ondersteuning voor optische media  Deze uitgave is een gebruikershandleiding en naslagwerk voor IBM Ondersteuning voor optische media op OS/400. Dankzij de informatie in dit boek is de gebruiker beter in staat te begrijpen wat de concepten van een bibliotheekserver op een optisch medium inhouden, hoe hij een bibliotheek op een optisch medium kan plannen, hoe hij een bibliotheekserver op een optisch medium kan beheren en bedienen en hoe hij problemen met een optische server kan oplossen.
- WebSphere Development Studio: ILE C/C++ Programmers Guide  Deze uitgave bevat informatie die u nodig hebt om ILE C/400-programma's op de iSeries-server te ontwerpen, bewerken, compileren, uitvoeren en herstellen.
- WebSphere Development Studio: C/C++ Language Reference  Deze uitgave bevat informatie over de structuur van ILE C/400-programma's en verdere informatie over de bibliotheekfuncties en Include-bestanden (headerbestanden).
- Security — Reference  Dit boek bevat uitgebreide technische informatie over OS/400-beveiliging, waaronder beveiligingssysteemwaarden die gevolgen hebben voor scanbewerkingen van een geïntegreerd bestandssysteem.

- **APPC Programming**  Deze uitgave beschrijft APPC-ondersteuning (Advanced Program-to-Program Communications) voor de iSeries-server. Het boek biedt hulp bij het ontwikkelen van toepassingsprogramma's die APPC gebruiken en bij het definiëren van een communicatie-omgeving voor APPC.
- **Backup en herstelprocedure**  Dit boek bevat algemene informatie over opties voor herstel en beschikbaarheid voor de IBM iSeries-server.

| **Ervaringsrapporten**

| Ervaringsrapporten worden geschreven door IBM-ontwikkelaars. Zij beschrijven daarin hun praktijkervaringen bij het toepassen van scenario's en oplossingen. U kunt ze gebruiken om de ervaringen van IBM-ontwikkelaars bij het uitvoeren van een bepaalde iSeries-oplossing te volgen, compleet met stapsgewijze aanwijzingen en tips.

| Het volgende ervaringsrapport heeft betrekking op bestanden en bestandssystemen.

| Een backup maken van het IFS

Bijlage. Kennisgevingen

Deze informatie is ontwikkeld voor producten en diensten die worden aangeboden in de Verenigde Staten.

IBM levert de producten, diensten en voorzieningen die in deze publicatie worden besproken mogelijk niet in andere landen. Raadpleeg uw lokale IBM-vertegenwoordiger voor informatie over de producten en voorzieningen die in uw regio beschikbaar zijn. Verwijzing in deze publicatie naar producten, programma's of diensten van IBM houdt niet in dat uitsluitend IBM producten, programma's of diensten kunnen worden gebruikt. Functioneel gelijkwaardige producten, programma's of diensten kunnen in plaats daarvan worden gebruikt, mits dergelijke producten, programma's of diensten geen inbreuk maken op intellectuele eigendomsrechten van IBM. Het is echter de verantwoordelijkheid van de gebruiker om niet door IBM geleverde producten, diensten en voorzieningen te controleren.

IBM kan over patenten of patenttoepassingen beschikken die onderwerpen behandelen die in dit document worden beschreven. Aan het feit dat deze publicatie aan u ter beschikking is gesteld, kan geen recht op licentie of ander recht worden ontleend. Informatie over het verkrijgen van een licentie kunt u opvragen, door te schrijven naar:

- | IBM Director of Licensing
- | IBM Corporation
- | 500 Columbus Avenue
- | Thornwood, NY 10594-1785
- | Verenigd Staten

Voor licentie-informatie over DBCS (Dubbelbyte), kunt u contact opnemen met het IBM Intellectual Property Department in uw land of schrijven naar:

- | IBM World Trade Asia Corporation
- | Licensing
- | 2-31 Roppongi 3-chome, Minato-ku
- | Tokio 106, Japan

Deze paragraaf is niet van toepassing op het Verenigd Koninkrijk of elk ander land waar deze voorwaarden strijdig zijn met de lokale wetgeving: INTERNATIONAL BUSINESS MACHINES VERSTREKT DEZE PUBLICATIE "AS IS", ZONDER ENIGE GARANTIE, UITDRUKKELIJK NOCH STILZWIJGEND, MET INBEGRIIP VAN DIE VOOR HET VOORGENOMEN GEBRUIK WAARVOOR HET PROGRAMMA IS BESTEMD OF GESCHIKTHEID VOOR EEN SPECIFIEK DOEL. In sommige landen is het uitsluiten van uitdrukkelijke of stilzwijgende garanties niet toegestaan. Het is daarom mogelijk dat bovenstaande bepaling niet op u van toepassing is.

In deze publicatie kunnen technische onjuistheden en drukfouten staan. Periodiek worden wijzigingen aangebracht aan de informatie in deze publicatie. Deze wijzigingen worden opgenomen in nieuwe uitgaven van deze publicatie. IBM kan op elk moment zonder kennisgeving verbeteringen en/of wijzigingen aanbrengen in de product(en) en/of programma('s) die in deze publicatie zijn beschreven.

Verwijzingen in deze publicatie naar niet-IBM-webpagina's dienen slechts tot het gemak van de gebruiker en betekenen in geen geval dat deze webpagina's door IBM worden aanbevolen. Het materiaal op deze websites vormt geen onderdeel van het materiaal voor dit IBM-product en het gebruik van die websites is voor eigen risico.

- | IBM kan de informatie die u levert op elke manier distribueren die zij toepasselijk acht, zonder enige verplichting jegens u te scheppen.

Licentiehouders die informatie over dit programma willen ontvangen over: (i) het uitwisselen van informatie tussen in eigen beheer gemaakte programma's en andere programma's (waaronder dit programma) en (ii) het gemeenschappelijk gebruik van de uitgewisselde informatie, dienen contact op te nemen met:

- | IBM Corporation
- | Software Interoperability Coordinator, Department 49XA
- | 3605 Highway 52N
- | Rochester, MN 55901
- | Verenigd Staten

Dergelijke informatie kan beschikbaar worden gesteld onder de daarvoor geldende voorwaarden. In bepaalde gevallen dient u hiervoor een vergoeding te betalen.

Het gelicentieerde programma dat in deze informatie wordt beschreven en al het beschikbare gelicentieerde materiaal wordt door IBM geleverd onder de voorwaarden van de IBM Klantenovereenkomst, de IBM Internationale Programmalicentie-overeenkomst (IPLA) of een andere vergelijkbare overeenkomst.

Alle gegevens in dit gedeelte over prestaties zijn vastgesteld in een gecontroleerde omgeving. Resultaten die worden verkregen in andere verwerkingsomgevingen kunnen daarom afwijken. Bepaalde metingen zijn verricht op systemen in de ontwikkelingsfase en er is geen enkele garantie dat deze metingen hetzelfde zullen zijn in algemeen verkrijgbare systemen. Bovendien is een aantal metingen afgeleid. Werkelijke resultaten kunnen variëren. Gebruikers van deze publicatie moeten controleren welke gegevens geschikt zijn voor hun specifieke omgeving.

Informatie over niet door IBM geleverde producten werd verkregen van de leveranciers van de betreffende producten, uit de publicaties van deze leveranciers of uit andere publiek toegankelijke bronnen. IBM heeft deze producten niet getest en staat niet in voor de prestaties van deze producten, de compatibiliteit of enig andere eis die kan worden gesteld aan niet door IBM geleverde producten. Vragen over de prestaties van niet door IBM geleverde producten dienen te worden gesteld aan de leveranciers van deze producten.

Alle uitingen over de toekomstige richting of intentie van IBM kunnen te allen tijde zonder enige kennisgeving worden teruggetrokken en vertegenwoordigen alleen doelen en doelstellingen.

Alle afgebeelde IBM prijzen zijn voorgestelde, actuele prijzen die zonder enige kennisgeving kunnen worden gewijzigd. Prijzen van dealers kunnen variëren.

Deze informatie is alleen bestemd voor planningdoeleinden. Deze informatie kan worden aangepast voordat de beschreven producten beschikbaar komen.

Deze informatie bevat voorbeelden van gegevens en rapporten die tijdens de dagelijkse zakelijke activiteiten worden gebruikt. Om deze zo volledig mogelijk te illustreren, bevatten de voorbeelden de namen van personen, bedrijven, merken en producten. Al deze namen zijn fictief en eventuele overeenkomsten met de namen en adressen van bestaande bedrijven zijn toevallig.

COPYRIGHTLICENTIE:

Deze informatie bevat voorbeeldtoepassingenprogramma's in de brontaal die programmeertechnieken op verschillende besturingsplatforms kunnen toelichten. U kunt deze voorbeeldprogramma's gratis kopiëren, wijzigen en distribueren om toepassingenprogramma's te ontwikkelen, te gebruiken, te verhandelen of te distribueren die in overeenstemming worden gebracht met de API (Application Programming Interface) voor het besturingsplatform waarvoor de voorbeeldprogramma's zijn geschreven. Deze voorbeelden zijn niet uitgebreid getest onder alle omstandigheden. IBM kan daarom de betrouwbaarheid, de bruikbaarheid of de functies van deze programma's niet garanderen. U kunt deze voorbeeldprogramma's gratis kopiëren.

ren, wijzigen en distribueren om toepassingsprogramma's te ontwikkelen, te gebruiken, te verhandelen of te distribueren die in overeenstemming worden gebracht met de API's (Application Programming Interface) van IBM.

Bij elk exemplaar of elk deel van deze voorbeeldprogramma's of daarvan afgeleide programma's moet de auteursrechtvermelding als volgt worden vermeld:

© IBM Corp., 2004. Delen van deze code zijn overgenomen van IBM Corp. Voorbeeldprogramma's. © Copyright IBM Corp. 2004. Alle rechten voorbehouden.

Indien u deze publicatie online bekijkt, worden foto's en illustraties wellicht niet afgebeeld.

Informatie over programmeerinterfaces

- | In deze informatie wordt programmeerinterfaces beschreven die bedoeld zijn om de klant programma's
- | te laten schrijven waarmee hij gebruik kan maken van de functionaliteit van het Geïntegreerd
- | bestandssysteem.

Merken

De volgende namen zijn merken van International Business Machines Corporation in de Verenigde Staten, andere landen of beide:

Application System/400
AS/400
e (logo)
IBM
iSeries
Operating System/400
OS/400
400C/400
DB2
Integrated Language Environment (ILE)
OfficeVision
OS/2
WebSphere xSeries

- | Lotus, Freelance en WordPro zijn handelsmerken van International Business Machines Corporation en
- | Lotus Development Corporation in de Verenigde Staten en/of andere landen.

Microsoft, Windows, Windows NT, en het Windows logo zijn handelsmerken van Microsoft Corporation in de Verenigde Staten, in andere landen, of beide.

UNIX is een merk van The Open Group in de Verenigde Staten en andere landen.

Andere namen van ondernemingen, producten en services kunnen handelsmerken van anderen zijn.

Bepalingen en voorwaarden voor het downloaden en afdrucken van publicaties

Toestemming voor gebruik van de publicaties die u gaat downloaden wordt verleend na acceptatie van de volgende voorwaarden en bepalingen.

Persoonlijk gebruik: U mag deze publicaties verveelvoudigen voor persoonlijk, niet-commercieel gebruik, mits voorzien van alle kennisgevingen. U mag deze publicaties, noch enig deel hiervan, niet verspreiden, afbeelden of hiervan afgeleide teksten produceren zonder de uitdrukkelijke toestemming van IBM.

Commercieel gebruik: U mag deze Publicaties alleen verveelvoudigen, verspreiden of afbeelden binnen uw onderneming en onder voorbehoud van alle eigendomsrechten. Het is niet toegestaan om afgeleid werk te maken op basis van deze publicaties of om deze publicaties of delen daarvan te reproduceren, te distribueren of af te beelden buiten uw bedrijf zonder uitdrukkelijke toestemming van IBM.

Met uitzondering van de uitdrukkelijke toestemming in dit document, worden er geen andere licenties of rechten verleend, uitdrukkelijk of stilzwijgend, voor publicaties of enige andere informatie, gegevens, software of intellectuele eigendomsrechten.

IBM behoudt zich het recht voor de hier verleende toestemming in te trekken, wanneer, naar het eigen oordeel van IBM, het gebruik van de publicaties haar belangen schaadt of als bovenstaande aanwijzingen niet naar behoren worden opgevolgd, zulks vast te stellen door IBM.

Het is alleen toegestaan deze informatie te downloaden, te exporteren of opnieuw te exporteren indien alle van toepassing zijnde wetten en regels, inclusief alle exportwetten en -regels van de Verenigde Staten, volledig worden nageleefd. IBM GEEFT GEEN ENKELE GARANTIE MET BETREKKING TOT DE INHOUD VAN DEZE PUBLICATIES. DE PUBLICATIES WORDEN VERSTREKT "AS-IS" EN ZONDER ENIGE GARANTIE, UITDRUKKELIJK OF STILZWIJGEND, MET INBEGRIJ VAN, MAAR NIET BEPERKT TOT, DIE VOOR VERHANDELBAARHEID EN GESCHIKTHEID VOOR EEN BEPAALD DOEL.

Het copyright van alle materiaal is eigendom van IBM Corporation.

Door een publicatie van deze site te downloaden of af te drukken, geeft u te kennen akkoord te gaan met deze bepalingen en voorwaarden.



Gedrukt in Nederland