

CICS® Transaction Gateway



Linux 게이트웨이 관리

버전 4.0

CICS® Transaction Gateway



Linux 게이트웨이 관리

버전 4.0

주!

이 책과 이 책이 지원하는 제품을 사용하기 전에 143 페이지의 『부록C. 주의사항』의 일반 정보를 반드시 읽으십시오.

초판(2001년 6월)

이 개정판은 Linux용 CICS® Transaction Gateway의 버전 4.0, 프로그램 번호 5724-A75에 적용됩니다. 후속판에 특별히 명시되기 전까지 이 책은 모든 후속판, 릴리스 및 개정판에 적용됩니다.

본 개정판은 SC34-5895를 대체합니다. 페이지 왼쪽에 있는 세로선은 본 개정판에 새로 소개된 자료를 나타냅니다.

© Copyright International Business Machines Corporation 1996, 2001. All rights reserved.

목차

그림	vii	보안 종료.	19
표	ix	CICS Transaction Gateway 및 객체 요청 브 로커	19
변경사항 요약	xi	기타 기능.	19
이 책에 대하여.	xiii	Euro 지원	20
이 책의 사용자.	xiii	액세스 기능	20
이 책에서 사용된 규약 및 용어	xiv	제2장 설치 전의 계획.	21
전제조건 및 관련 정보	xiv	하드웨어 요구사항	21
의견서 보내기	xiv	지원 소프트웨어.	21
IBM에서 책 구하기	xv	GPL 사용권 및 저작권 문제	26
제1장 개요.	1	제3장 설치	27
CICS Transaction Gateway의 제공 내용	2	Linux용 CICS [®] Transaction Gateway 설치	27
Java™ 기술	4	자국어 지원	28
Java 언어	4	원격 시스템에서 X-Windows 사용	28
Java 애플릿	4	CICS Transaction Gateway 설치제거	29
Java 서블릿	5	제4장 구성	31
Java 응용프로그램	6	CICS Transaction Gateway에 대한 프로그래 밍 환경 구성.	31
JavaBeans™	6	Linux에서 CLASSPATH 설정.	32
방화벽	7	구성 도구 사용	32
웹 브라우저 및 네트워크 컴퓨터.	8	구성 도구 인터페이스.	33
웹 서버.	8	CICS Transaction Gateway 설정값 구성	36
CICS Transaction Gateway가 CICS에 액세스 하는 방법	9	클라이언트 설정값 구성	42
CICS Transaction Gateway: 쓰레드 모델	11	서버 설정값 구성	45
외부 액세스 인터페이스(EPI, ECI, SI)	13	트레이스 설정값.	51
EPI(External Presentation Interface)	14	구성 변환 도구	54
ECI(External Call Interface)	15	변환 도구 사용	55
ESI(External Security Interface).	16	구성 파일 편집	56
네트워크 보안	16	GATEWAY 섹션	57
SSL(Secure Sockets Layer)	17	CLIENT 섹션	57
HTTPS	18	SERVER 섹션	58
키 및 인증	18	DRIVER 섹션	58

구성 이름 바꾸기 및 파일 맵핑	58	제7장 CICS Transaction Gateway 단말기	
클라이언트 키보드 맵핑 구성	59	서블릿	95
키보드 맵핑 파일 구분	59	CICS Transaction Gateway 단말기 서블릿이	
키보드 맵핑 파일	60	단말기?	95
화면 색상 사용자 정의	63	단말기 서블릿 설치 및 구성	98
색상 맵핑 구분	64	웹 서버 CLASSPATH 및 PATH 설정 구	
색상 맵핑 파일	65	성	98
국지 CICS Transaction Gateway 지원 사용		단말기 서블릿을 웹 서버 구성에 추가	98
준비	67	서블릿 초기화 매개변수 구성	99
제5장 연결 보안	69	기타 구성 선택항목 고려사항	104
개요	69	단말기 서블릿 로드	104
암호화란?	70	단말기 서블릿 사용	104
디지털 사인 및 디지털 인증	71	CICS로 연결 및 트랜잭션 시작	105
디지털 인증서 확보	72	단말기 서블릿 호출	105
키링	73	다음 내용	107
SSL 및 인증	74	화면 및 필드 표시	107
HTTPS	76	CICS로 다시 화면 전송	109
ctgikey 도구	76	AID 설정	110
클라이언트 워크스테이션으로 iKeyman 분		연결해제	110
산	77	특성 및 매개변수 참조서	111
외부 사인 인증 사용(SSLight)	78	서블릿 구성 특성	111
SSL 서버 구성	78	페이지 맵핑 특성	115
SSL 클라이언트 구성	81	요청 매개변수	115
자가 사인 인증 사용(SSLight)	84	표시가능한 특성	116
SSL 서버 구성	84	OS/390®용 CICS® Transaction Gateway	
SSL 클라이언트 구성	86	웹 인터페이스	117
서버 키링에 대한 액세스 제한	88	제8장 문제점 판별 및 문제점 해결	119
SSL 및 HTTPS용 CICS Transaction		기본 점검	119
Gateway 구성	88	다음 수행 조치	120
클라이언트 키링 지정	89	메시지	120
제6장 CICS Transaction Gateway 조작 91		트레이스	120
게이트웨이 시작	91	프로그램 지원	124
preset 선택항목으로 게이트웨이 시작	91	자주 발생하는 문제점 판별	124
사용자가 지정한 선택항목으로 게이트웨이		설치 문제점	124
시작	92	Java 예외	124
게이트웨이 중지	94	게이트웨이가 Windows NT® 서비스로 실행	
		될 때의 문제점	125

	java.lang.OutOfMemory 예외	126	CICS Universal Clients 책	138
	프로세스가 잠긴 경우	127	CICS 관련 책	139
	코드 페이지 문제점	127	책 파일명	140
	SSL 예외	128	예제 구성 문서	140
	JDK™ AppletViewer 문제점	129	기타 책	141
	부록A. CICS Universal Clients 자료 변환	131	온라인 문서 열람	141
	지원되는 변환	131	PDF 책 열람	142
	부록B. CICS Transaction Gateway 및		부록C. 주의사항	143
	CICS Universal Clients 라이브러리	137	상표	145
	CICS Transaction Gateway 책	137	색인	147

그림

1. CICS Transaction Gateway	2	7. SSL은 서버 인증과 데이터를 교환합니 다.	75
2. TCP/IP 및 SSL용 CICS Transaction Gateway 쓰레드 모델	12	8. URL로 호출된 단말기 서블릿을 사용하 는 CICS Transaction Gateway	97
3. HTTP/HTTPS용 CICS Transaction Gateway 쓰레드 모델	13	9. 샘플 Java 스택 트레이스	122
4. 구성 도구	34	10. 샘플 Java 스택 덤프.	125
5. CICS Universal Clients의 샘플 키보드 맵핑 파일	62	11. Windows NT® 서비스로 실행 중인 계 이트웨이의 트레이스	125
6. 샘플 색상 맵핑 파일	66		

표

1. CICS Transaction Gateway 플랫폼에서 의 쓰레드 한계	13	4. 서블릿 초기화 매개변수	99
2. 지원 제품	22	5. CICS Transaction Gateway 및 CICS Universal Clients 책과 파일명	140
3. 맵핑할 수 있는 CICS Universal Client 키	60		

변경사항 요약

다음은 CICS Transaction Gateway 버전 4.0의 기능상 변경사항입니다.

- TCP62가 지원됩니다.
- 다음에 대한 지원은 없어졌습니다.
 - Cobol
 - PL/I
 - REXX
 - CICSTELD

이 개정판은 SC34-5895를 대체합니다. 페이지 왼쪽에 있는 세로선은 본 개정판에 새로 소개된 자료를 나타냅니다.

이 책에 대하여

이 책은 다음과 같은 장으로 구성됩니다.

- 제1장에서는 CICS Transaction Gateway를 소개하고, 이 제품의 사용에 따른 이점과 제품이 제공하는 기능을 요약합니다.
- 제2장에서는 CICS Transaction Gateway의 하드웨어 및 소프트웨어 요구사항 등의 계획 문제에 대해 설명합니다.
- 제3장에서는 CICS Transaction Gateway 설치 방법을 설명합니다.
- 제4장에서는 CICS Transaction Gateway 구성 방법을 설명합니다.
- 제5장에서는 CICS Transaction Gateway의 SSL 및 HTTPS 프로토콜에 대한 보안 설정 방법을 설명합니다.
- 제6장에서는 CICS Transaction Gateway 시작과 중지 등의 CICS Transaction Gateway 작동 방법을 설명합니다.
- 제7장에서는 CICS Transaction Gateway와 함께 제공되는 단말기 서블릿에 대해 소개합니다.
- 제8장에서는 CICS Transaction Gateway에 대한 문제점 판별을 설명합니다.
- 부록B에서는 CICS Transaction Gateway 및 CICS Universal Clients 라이브러리에서 온라인 정보를 열람하는 방법과 책을 인쇄 및 주문하는 방법에 대해 설명합니다.

이 책의 사용자

이 책은 CICS Transaction Gateway의 계획, 설치, 조정 또는 운영과 관련된 모든 사용자를 대상으로 합니다.

이 책에서는 사용자가 CICS Transaction Gateway가 실행되고 있는 운영 체제에 익숙한 것으로 간주합니다.

인터넷 용어를 이해할 수 있다면 도움이 됩니다.

이 책에서 사용된 규약 및 용어

본 서적에서 *CICS Universal Client*라는 용어는 CICS Transaction Gateway의 클라이언트 구성요소를 나타냅니다.

『System/390[®]용 CICS』는 다음 CICS 서버 제품을 나타냅니다.

- MVS/ESA[™]용 CICS[®]
- OS/390[®]용 CICS[®] Transaction Gateway
- VSE/ESA[™]용 CICS[®] Transaction Gateway
- CICS/VSE[®]

CICS Transaction Gateway는 Windows NT[®] 및 Windows 2000에서 실행됩니다. 본 서적에서 Windows의 특정 버전이 언급되지 않은 한, Windows[®]라고 하면 Windows NT와 Windows 2000을 모두 나타냅니다.

텍스트에서 OS/390이라고 하면 OS/390 및 z/OS 운영 체제를 모두 나타냅니다.

전제조건 및 관련 정보

이 제품에 대해 사용 가능한 책에 대해서는 137 페이지의 『부록B. CICS Transaction Gateway 및 CICS Universal Clients 라이브러리』를 참조하십시오. 이 장에서는 소프트웨어를 보고 인쇄하는 방법, IBM에 인쇄된 사본을 주문하는 방법도 자세히 설명합니다.

의견서 보내기

고객의 의견은 더 정확하고 향상된 고급 정보를 제공하는 데 도움이 되는 중요한 내용입니다. 이 책 또는 다른 CICS 책에 대해 의견이 있으면 다음과 같은 방법으로 보내 주십시오.

- 다음 주소의 웹 사이트를 방문하여,

www.ibm.com/software/ts/cics/

Library로 링크하면 고객 의견서 양식을 볼 수 있습니다.

여기에서 의견을 입력하고 제출할 수 있습니다.

- ibmkspoe@kr.ibm.com 주소로 전자 메일을 보내주십시오.
- 다음 주소로 팩스를 보내주십시오.

IBM 고객만족센터: (02) 3781-5200

- 다음 주소로 우편을 보내주십시오.

135-270

서울 특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터

어느 방법을 이용하시든, 다음 내용을 확인하십시오.

- 책 이름
- 책의 양식 번호
- 가능한 경우, 제품 버전
- 의견이 있는 특정 텍스트 위치. 예를 들면, 페이지 번호나 표 번호

독자가 IBM에 정보를 제공한 경우, 독자는 정보를 적절한 방식으로 사용하거나 배포할 수 있는 비독점적 권한을 IBM에 부여한 것이며 독자는 이에 대해 책임을 지지 않습니다.

IBM에서 책 구하기

다운로드할 수 있는 책에 대해서는 다음 주소의 웹 사이트를 방문하십시오.

www.ibm.com/software/ts/cics/

그런 다음 **Library** 링크를 따르십시오.

다음과 같이 하드카피 책을 주문할 수 있습니다.

- IBM 대표나 해당 지역을 관할하는 IBM 지사를 통해 주문할 수 있습니다.
- 미국에서는 1-800-879-2755로 전화하십시오.
- 다음 주소의 웹 사이트에서도 주문할 수 있습니다.

www.ibm.com/shop/publications/order

제1장 개요

IBM CICS Transaction Gateway는 웹 브라우저 및 네트워크 컴퓨터에서, 일정 범위 구성의 표준 인터넷 프로토콜을 사용하는 CICS Transaction Server 또는 TXSeries™ 서버에서 실행 중인 비즈니스 고유의 응용프로그램으로의 안전하고 쉬운 액세스를 제공합니다.

CICS Transaction Gateway는 웹 서버에 대한 견고하고 조정가능한 소프트웨어 이므로 IBM WebSphere™용 e-business 커넥터 즉, Java™ 서블릿에 대한 런타임 환경으로 구현할 수 있습니다.

CICS Transaction Gateway는 Windows NT®, AIX®, Linux, Solaris, HP-UX, OS/390® 플랫폼용으로 제공됩니다.

2 페이지의 그림1에서는 웹 클라이언트가 CICS 프로그램 및 데이터에 액세스하는 방법을 보여줍니다. CICS Transaction Gateway는 웹 서버 시스템에 설치된 것으로 나타나 있습니다. 이런 배치는 Java 애플릿이 설치된 CICS Transaction Gateway를 사용할 경우에만 필수입니다.

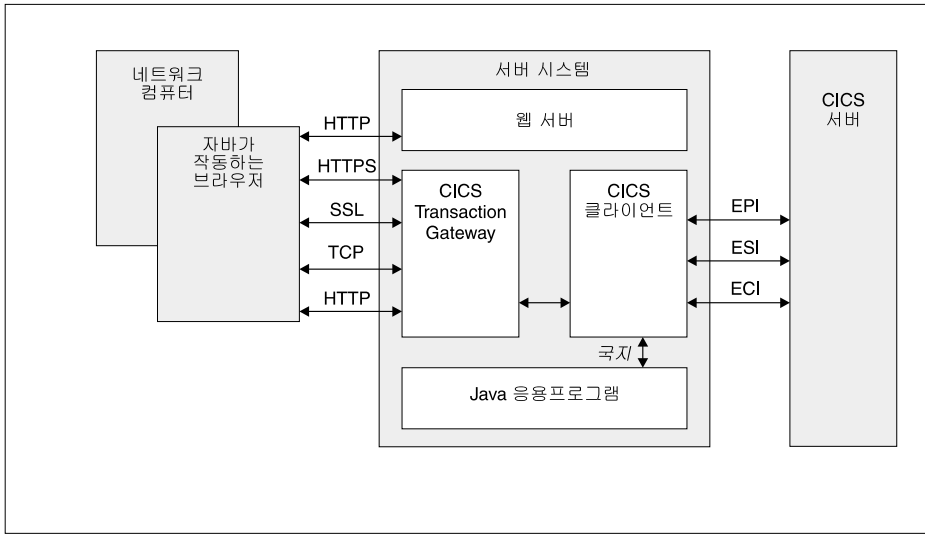


그림 1. CICS Transaction Gateway

CICS Transaction Gateway와의 통신은 다음 프로토콜을 기반으로 합니다.

- TCP/IP 소켓
- HTTP(Hypertext Transfer Protocol)
- SSL(Secure Sockets Layer)
- HTTPS(TTP over SSL)

TCP/IP 소켓과 SSL은 인트라넷 응용프로그램에 대해 효과적인 통신 방법을 제공합니다. 방화벽이 있는 경우 HTTP와 그 보안 대체 HTTPS는 인터넷 응용프로그램에 대해 효과적인 통신 프로토콜입니다(16 페이지의 『네트워크 보안』 참조).

CICS Transaction Gateway의 제공 내용

CICS Transaction Gateway에서 제공하는 내용은 다음과 같습니다.

1. **Java 게이트웨이 응용프로그램** - 이 응용프로그램은 보안을 위해 주로 웹 서버 워크스테이션에 상주합니다. 이 프로그램은 CICS Universal Clients에서 제공하는 ECI(External Call Interface), EPI(External Presentation Interface) 또는 ESI(External Security Interface) 인터페이스를 통해 CICS 서버에서 실행

행 중인 CICS 응용프로그램과 통신합니다. 이 Java 응용프로그램은 이전에도 IBM Java용 CICS Gateway에서 사용할 수 있었습니다.

2. **CICS Universal Client** - 단말기 애플리케이션 기능뿐만 아니라 ECI, EPI 및 ESI 인터페이스도 제공합니다. **ECI** 인터페이스를 사용하면 비 CICS 클라이언트 응용프로그램에서 서버루틴으로 CICS 프로그램을 동기적 또는 비동기적으로 호출할 수 있습니다. **EPI** 인터페이스를 사용하면 비 CICS 클라이언트 응용프로그램이 논리적 3270 단말기 역할을 할 수 있으므로 CICS 3270 응용프로그램을 제어할 수 있습니다. **ESI** 인터페이스를 사용하면 비 CICS 클라이언트 응용프로그램에서 APPC 암호 만기 관리(PEM)가 제공하는 서비스를 호출할 수 있습니다. CICS Universal Client를 사용하면 운영 체제에 따라 TCP/IP, APPC 및 기타 프로토콜을 사용하여 CICS 서버와 통신할 수 있습니다.
3. **CICS Java 클래스 라이브러리** - API를 제공하는 클래스가 있으며, 이런 클래스는 Java 게이트웨이 응용프로그램과 Java 응용프로그램(애플릿 또는 서블릿)간의 통신에 사용됩니다. **JavaGateway** 클래스는 게이트웨이와의 통신을 설정하는 데 사용되며, Java 소켓 프로토콜을 사용합니다. **ECIRequest**, **EPIRequest** 및 **ESIRRequest** 클래스는 각기 게이트웨이로 전달되는 ECI, EPI 및 ESI 호출을 지정하는 데 사용됩니다.
4. 단말기 서블릿 - 임의의 CICS 서버에서 실행 중인 3270 응용프로그램용 애플레이터로 웹 브라우저를 사용할 수 있도록 합니다. 단말기 서블릿은 JSDK(Java Servlet Development Kit) 버전 1.1 이상과 동일한 지원을 제공하는 웹 서버 또는 서블릿 엔진과 함께 사용할 수 있습니다. 이것은 IBM CICS 클라이언트 버전 2에서 CICS 인터넷 게이트웨이가 제공하는 향상된 버전의 기능입니다. 단말기 서블릿은 EPI 프로그래밍 인터페이스 사용에 대한 대안을 제공합니다.
5. **Java EPI Bean** 세트 - 프로그래밍하지 않고 기존의 CICS 3270 응용프로그램용 Java 프론트 엔드를 작성하는 데 사용합니다.

CICS Transaction Gateway는 연결된 웹 브라우저에 대한 여러 통신 링크를 동시에 관리하고, 복수 CICS 서버 시스템에 대한 비동기 대화를 제어할 수 있습니다. 멀티쓰레드화된 CICS Transaction Gateway 구조를 사용하면 단일 게이트웨이에서 동시에 연결되어 있는 여러 사용자를 지원할 수 있습니다.

Java™ 기술

이 절에서는 개발할 수 있는 프로그램 유형 및 보안 관련사항을 포함하여 Java 언어에 대해 설명합니다.

Java 언어

Java 언어는 Java 서블릿, Java 애플릿 및 Java 응용프로그램을 작성하는 데 사용될 수 있습니다.

Java는 해석된 객체 지향 언어이며, 소스와 객체 양식의 플랫폼에 독립적인 프로그램을 작성하는 데 사용할 수 있는 C++와 비슷합니다. 웹 서버뿐만 아니라 웹 브라우저도 확장하는 고유한 작동 특성을 사용하면 인터넷 응용프로그램에서 새롭고 강력한 기능을 사용할 수 있습니다.

플랫폼 독립성을 확보하기 위해 Java 언어는 플랫폼에 종속되는 작동을 허용하지 않으며, 프리프로세서, 운영자 과부하, 복수 상속 및 포인터와 같은 일부 C++ 기능을 제외합니다. 모든 Java 프로그래밍은 클래스 안에서 캡슐화되며, JDK(Java Development Kit)에는 GUI 기능, 입출력 기능 및 네트워크 통신 등 플랫폼 독립성을 확보하는 데 중요한 특수 클래스가 있습니다.

Java 컴파일러는 시스템과는 관계 없는 중간 바이트 코드 형식을 생성합니다. 이 바이트 코드는 Java 해독기로 실행시 처리됩니다. 또한 해독기는 실행시 바이트 코드를 검사하여 시스템 환경에 대한 유효성과 안전성을 검증합니다. Java 해독기가 제공하는 분리로 인해, 이 해독기를 때로 Java 가상 시스템(JVM)라고도 합니다.

Java 애플릿

Java 애플릿은 Java 지원 웹 브라우저 또는 네트워크 컴퓨터에 다운로드하여 실행할 수 있는 작은 응용프로그램입니다. Java 애플릿은 보통 클라이언트 코드가 클라이언트/서버 구조로 수행하는 유형의 조작을 수행합니다. Java 애플릿은 입력 데이터를 편집하고, 화면을 제어하고, 데이터나 데이터베이스를 조작하는 서버로 트랜잭션을 보냅니다.

애플릿은 <applet> HTML 태그를 사용하여 시작합니다. 이 태그는 애플릿에 제어어를 제공하며 애플릿에서 사용할 화면을 지정합니다. Java 지원 서버에서 페이지

를 다운로드하는 중 이 태그를 발견하면 HTML 이미지 태그에서 참조하는 이미지를 다운로드할 때와 같은 방식으로 애플릿 바이트 코드도 다운로드합니다. Java 지원 브라우저에서 애플릿 바이트 코드를 해석하여 실행합니다. 애플릿은 화면 입력을 편집하고, 화면 출력을 생성하고, 다운로드를 실행한 컴퓨터와 다시 통신합니다. 여러 개의 애플릿이 동시에 실행될 수 있습니다.

애플릿의 한 예로, 서버와 계속 통신하며 주식 거래 정보를 받아 화면의 창에 그 내용을 갱신하는 애플릿이 있습니다.

애플릿이 일반적으로 그리 크지 않으므로 애플릿 다운로드가 사용자 종료율을 위한 응답 시간에 대해 중요한 성능상의 저해를 일으켜선 안됩니다. 애플릿이 웹 서버로 반복 작업을 생략하여 브라우저 전반에 걸친 성능을 개선할 수 있습니다. 또한 이미지가 웹 브라우저에 캐시될 때 애플릿이 캐시되어 잦은 애플릿의 다운로드가 최소화됩니다.

Java 서블릿

Java 서블릿은 웹 브라우저로 다운로드되는 Java 애플릿과는 달리 웹 서버 시스템에서 실행되는 소형 Java 응용프로그램입니다.

Java 서블릿은 CGI(Common Gateway Interface) 프로그램의 대응으로 일반화되었습니다. CGI 응용프로그램에 비한 Java 서블릿의 장점은 다몬 프로세스에서 스레드로 호출되므로 더 신속하게 실행될 수 있다는 것입니다. 즉, Java 서블릿은 메모리에서 지속적이며 여러 요청을 실행할 수 있습니다.

다음과 같은 3가지 방법으로 서블릿을 실행할 수 있습니다.

URL(Uniform Resource Locator)로 서블릿 호출

서블릿 URL의 양식은 보통 다음과 같습니다.

`http://machine-name:port/servlet/servlet-name`

이 방법은 사용자가 정보를 입력할 필요가 없을 때 사용됩니다. 요청 매개변수는 URL에서 조회 문자열 양식으로 인코딩될 수 있습니다.

양식으로 서블릿 호출

이 방법은 웹 페이지에서의 HTML 양식 작성과 관련됩니다.

```
<FORM METHOD="GET" ACTION="/servlet/servlet-name">  
attributes  
</FORM>
```

이 방법은 사용자가 정보를 입력해야 하는 경우 사용됩니다.

서버측 include로 서블릿 호출

서버측 include는 웹 페이지가 사용자에게 전송되기 전에 웹 서버에서 처리합니다. HTML 소스에서, 서버측 include는 다음과 같습니다.

```
<SERVLET NAME="TerminalServlet" >  
<PARAM NAME="request" VALUE="send">  
<PARAM NAME="transaction" VALUE="CECI">  
<PARAM NAME="display" VALUE="none">  
</SERVLET>
```

Java 응용프로그램

Java 응용프로그램은 컴퓨터에서 국지로 실행되는 프로그램입니다. 여기에는 애플릿이 제공하는 기능과 함께 플랫폼 종속 기능도 있습니다. Java 응용프로그램은 국지 파일에 액세스하고, 일반 네트워크 연결을 작성하여 승인하고, 시스템 고유의 라이브러리에서 고유 C 또는 C++ 기능을 호출할 수 있습니다.

JavaBeans™

Sun Microsystems에서 개발한 JavaBeans API를 사용하면 Java에서 구성요소 소프트웨어를 작성할 수 있습니다. 구성요소에는 자체 완비된, 재사용 가능한 소프트웨어 단위이며, VisualAge® for Java와 같은 비주얼 응용프로그램 빌더 도구를 사용하여 애플릿 또는 서블릿에 시각적으로 구성해 넣을 수 있습니다. 모든 JDK 버전 1.1 호환 브라우저나 도구는 JavaBeans를 지원합니다.

JavaBeans 구성요소를 Bean이라 합니다. 대부분의 빌더 도구를 사용하여 팔레트나 도구 박스에서 Bean을 유지관리할 수 있습니다. 도구 박스에서 특정 Bean을 선택하여 양식에 놓은 후, 모양과 작동을 수정한 다음 다른 Bean과의 상호작용 방식을 정의할 수 있습니다. 애플릿, 서블릿 또는 새로운 Bean으로 선택한 Bean과 다른 Bean을 구성해 넣을 수 있으며, 이러한 작업은 코딩 없이 수행할 수 있습니다. JavaBeans에 대한 자세한 정보는 Sun 웹 사이트(www.java.sun.com)를 참조하십시오.

CICS Transaction Gateway는 고급 EPI 인터페이스에 기초한 **EPI Java Bean** 을 제공합니다. 이런 Bean을 사용하면 프로그래밍하지 않고도 기존 CICS 3270 응용프로그램에서 데이터에 액세스하는 Java 응용프로그램(애플릿 및 응용프로그램)을 쉽게 작성할 수 있습니다. VisualAge for Java와 같은 Bean 작성기 도구를 사용하면 CICS에 연결하고, 트랜잭션을 실행하며, 3270 화면에서 데이터를 표시하고, 사용자 입력을 다시 CICS 서버로 전송할 수 있는 새로운 Java 프론트엔드를 쉽고 빠르게 작성할 수 있습니다. 자세한 정보는 *CICS Transaction Gateway: Gateway Programming* 서적을 참조하십시오.

방화벽

Java 애플릿 통신 사용에서 현재 디자인 고려사항은 방화벽의 영향입니다. 방화벽은 신뢰성 있는 네트워크와 신뢰성 없는 네트워크 사이에서 허가 없는 전송을 예방하는 소프트웨어 구성의 용어입니다. 방화벽은 외부 침입으로부터 회사 자산을 보호하기 위해 지정되지만, 합법적인 통신도 제한할 수 있습니다. 방화벽은 다음과 같은 두 가지 방식으로 작동할 수 있습니다.

1. 외부 사용자에게 대한 서버의 일반 액세스 기능 - 인바운드 제한사항
2. 방화벽 외부에서 특정 네트워크 기능을 수행할 수 있는 방화벽 내부의 일반 사용자 기능 - 아웃바운드 제한사항

게이트웨이 처리기가 방화벽 외부에 놓여 방화벽을 통해 CICS 서버와 연결될 수 있으므로 CICS Transaction Gateway 구성은 첫번째 영역의 문제점을 예방하는데 적합합니다. 일반 사용자와 관련된 아웃바운드 방화벽은 문제가 될 수 있습니다. 대기업에서는 방화벽을 사용하여 사용할 수 있는 연결 유형과 프로토콜을 제한할 수 있습니다.

인트라넷(인터넷의 국지 구현)에서 Java를 사용할 경우에는 방화벽이 보통 한 요인이 되지 못하므로 매우 잘 작동합니다. 그러나 회사 외부의 일반 사용자를 위해 인터넷 응용프로그램을 디자인할 경우, 일반 사용자 방화벽이 구현 요인이 될지 여부를 결정해야 합니다. 구현 요인이 되면 웹 서버에서 Java 서블릿으로 Java 코드를 실행할 때와 같이 해당 사용자에게 대한 대체 프로세스가 필요할 수 있습니다. 또한 CICS Transaction Gateway에서 지원하는 HTTP 및 HTTPS 프로토콜 사용도 고려해야 합니다. 17 페이지의 『SSL(Secure Sockets Layer)』 및 18 페이지의 『HTTPS』를 참조하십시오.

웹 브라우저 및 네트워크 컴퓨터

CICS Transaction Gateway를 사용하려면 JDK 버전 1.1이 사용 가능한 Java 지원 웹 브라우저가 필요합니다. 예를 들면, Windows 및 AIX의 경우 Netscape Communicator 4.5.1 이상입니다. 자세한 정보는 21 페이지의 『지원 소프트웨어』를 참조하십시오.

웹 브라우저는 HTML(HyperText Markup Language) 페이지 다운로드를 요청하는 HTTP(HyperText Transport Protocol)를 사용하여 웹 서버와 통신합니다. 이런 HTML 페이지에는 Java 애플릿(4 페이지의 『Java 애플릿』 참조) 또는 서블릿에 대한 호출이 포함될 수 있으며, 여러 애플릿들이 동시에 실행될 수 있습니다.

각 브라우저에서 같은 정보를 다르게 표시할 수도 있습니다.

네트워크 컴퓨터는 인터넷 사용자를 위한 저가 컴퓨터이며, 웹 브라우저와 같은 기능을 수행합니다.

웹 서버

웹 서버는 웹 브라우저에서 생성한 정보 요청에 응답하는 소프트웨어 프로그램입니다. 브라우저에서 요청을 받으면 웹 서버는 요청을 처리하여 다음과 같이 취할 조치를 결정합니다.

- 요청된 문서 리턴
- 요청 거부
- 외부 응용프로그램을 사용한 처리를 위해 요청 전달. 예를 들어, 요청은 검색 요청을 수행하는 데이터베이스에 대한 요청 또는 Lotus Domino™와 같은 더 동적인 정보 전달 양식에 대한 요청이 될 수 있습니다.

웹 서버와 외부 응용프로그램 간의 통신은 분명하므로, 웹 서버의 URL만 알면 요청을 지정할 수 있습니다. 또한 모든 웹 서버는 여러 브라우저의 요청을 동시에 처리할 수 있습니다.

제한된 사용자 집합에 대한 액세스를 제한하거나 상품 또는 서비스 구입을 위한 보안을 제공하도록 특별히 구성된 서버도 있습니다.

웹 서버는 거의 모든 플랫폼에 대해 있으며, 여러 제공업체의 제품이 있습니다. CICS Transaction Gateway에서 지원하는 웹 서버에 대해서는 21 페이지의 『지원 소프트웨어』를 참조하십시오.

CICS Transaction Gateway가 CICS에 액세스하는 방법

이 절에서는 CICS Transaction Gateway를 사용하여 CICS 프로그램 및 데이터에 액세스하는 방법에 대해 설명합니다.

Java 애플릿의 경우, 다음과 같이 액세스합니다.

1. 웹 브라우저 또는 네트워크 컴퓨터는 HTTP 프로토콜을 사용하여 웹 서버에서 HTML 페이지를 요청합니다.
2. 웹 서버는 Java 애플릿을 식별하는 태그가 있는 HTML 페이지를 리턴합니다.
3. 브라우저는 웹 서버의 Java 클래스와 관련된 요청을 시작합니다.
4. 웹 서버는 요청된 CICS Transaction Gateway 클래스가 있는 Java 클래스를 리턴합니다.
5. 클래스가 리턴되면 Java 애플릿이 시작됩니다.
6. Java 애플릿은 **JavaGateway** 객체를 작성하여 CICS Transaction Gateway에 연결합니다. 그러면 Java의 소켓 프로토콜을 사용하여 브라우저와 장기간 실행되는 게이트웨이 프로세스 간의 통신이 설정됩니다.
7. Java 애플릿은 각기 ECI, EPI 또는 ESI 호출이 있는 **ECIRequest**, **EPIRequest** 또는 **ESIRRequest** 객체를 작성한 후 **JavaGateway.flow** 메소드를 사용하여 게이트웨이로 해당 객체를 보냅니다.
8. 게이트웨이는 요청을 받아 압축을 해제한 후, CICS Universal Client에 대한 해당 ECI, EPI 또는 ESI 호출을 작성합니다.
9. CICS Universal Client는 ECI, EPI 또는 ESI 호출을 원하는 CICS 서버로 전달합니다.
10. CICS 서버는 필요한 경우 사용자 ID 및 암호에 대한 검증을 포함하여 호출을 처리한 후 CICS 적용업무 프로그램으로 제어 및 사용자 데이터를 전달합니다.

11. CICS 적용업무 프로그램이 처리를 완료하면 제어와 데이터를 CICS로 리턴합니다. CICS에서는 제어 및 데이터를 클라이언트로 리턴하고 클라이언트는 게이트웨이로 리턴합니다.
12. 게이트웨이는 이 결과를 압축하여 웹 브라우저에서 실행 중인 Java 애플릿으로 리턴합니다.

Java 서블릿의 경우, 다음과 같이 액세스합니다.

1. 웹 서버는 서블릿을 로드하여 초기화합니다. 이런 프로세스는 웹 서버가 시작되거나 서블릿에 대한 요청이 처음 이루어질 때 수행될 수 있습니다. 이때 서블릿은 **JavaGateway** 객체를 작성하여 CICS Transaction Gateway에 연결합니다.
2. 서블릿이 해당 HTTP 요청으로 호출되면 웹 서버는 자세한 요청과 함께 **Service** 메소드를 호출합니다.
3. Java 서블릿은 각기 ECI, EPI 또는 ESI 호출이 있는 **ECIRequest**, **EPIRequest** 또는 **ESIRRequest** 객체를 작성한 후 **JavaGateway.flow** 메소드를 사용하여 해당 객체를 게이트웨이로 보냅니다.
4. 게이트웨이는 요청을 받아 압축을 해제한 후, CICS Universal Client에 대한 해당 ECI, EPI 또는 ESI 호출을 작성합니다.
5. CICS Universal Client는 ECI, EPI 또는 ESI 호출을 원하는 CICS 서버로 전달합니다.
6. CICS 서버는 필요한 경우 사용자 ID 및 암호에 대한 검증을 포함하여 호출을 처리한 후 CICS 적용업무 프로그램으로 제어 및 사용자 데이터를 전달합니다.
7. CICS 적용업무 프로그램이 처리를 완료하면 제어와 데이터를 CICS로 리턴합니다. CICS에서는 제어 및 데이터를 클라이언트로 리턴하고 클라이언트는 게이트웨이로 리턴합니다.
8. 게이트웨이는 이 결과를 압축하여 Java 애플릿으로 리턴합니다.
9. 서블릿은 CICS Transaction Gateway에 대한 요청의 결과를 받으면 웹 브라우저에 리턴할 HTTP 응답을 생성합니다.

주: ECI, EPI 및 ESI 호출 모두가 **ECIRequest**, **EPIRequest** 및 **ESIRRequest**에서 지원되는 것은 아닙니다. 자세한 정보는 *CICS Transaction Gateway: Gateway Programming*을 참조하십시오.

CICS Transaction Gateway: 스레드 모델

CICS Transaction Gateway는 네트워크 연결을 처리하고, Java 클라이언트의 요청과 그에 대한 응답을 위해 스레드를 지정하는 멀티스레드 모델을 제공합니다. 스레드 모델에는 다음과 같은 구성요소가 있습니다.

ConnectionManagers

ConnectionManager는 특정 Java 클라이언트(애플릿 또는 응용프로그램)로부터의 모든 연결을 관리합니다. 요청을 받으면 사용 가능한 Worker 스레드 풀에서 Worker 스레드를 할당하여 요청을 실행합니다. 초기 ConnectionManager 자원 풀의 크기는 연결 매니저 스레드의 초기 수 구성 설정값으로 정의됩니다. 연결 매니저 스레드의 최대 수 설정값으로 ConnectionManager 풀의 최대 크기를 지정할 수 있습니다. 자세한 정보는 32 페이지의 『구성 도구 사용』을 참조하십시오. 또한 CICS Transaction Gateway를 시작한 경우에도 이런 한계를 지정할 수도 있습니다. 92 페이지의 『사용자가 지정한 선택항목으로 게이트웨이 시작』을 참조하십시오.

Workers

Worker는 Java 클라이언트의 요청을 실제로 실행하는 객체입니다. 각 Worker 객체에는 자체 스레드가 있으며, 이런 스레드는 수행할 작업이 있을 때 활성화됩니다. Worker 스레드가 완료되면 사용 가능한 Worker 스레드 풀로 다시 돌아갑니다. ConnectionManager와 마찬가지로, Worker 자원 풀의 초기 크기는 **Worker** 스레드의 초기 수 설정값을 사용하여 지정됩니다. **Worker** 스레드의 최대 수 설정값으로 Worker 풀의 최대 크기를 지정할 수 있습니다. 32 페이지의 『구성 도구 사용』을 참조하십시오. 또한 CICS Transaction Gateway를 시작한 경우에도 이런 한계를 지정할 수도 있습니다. 92 페이지의 『사용자가 지정한 선택항목으로 게이트웨이 시작』을 참조하십시오.

스레드 모델에 대해서는 다음 그림에서 설명합니다.

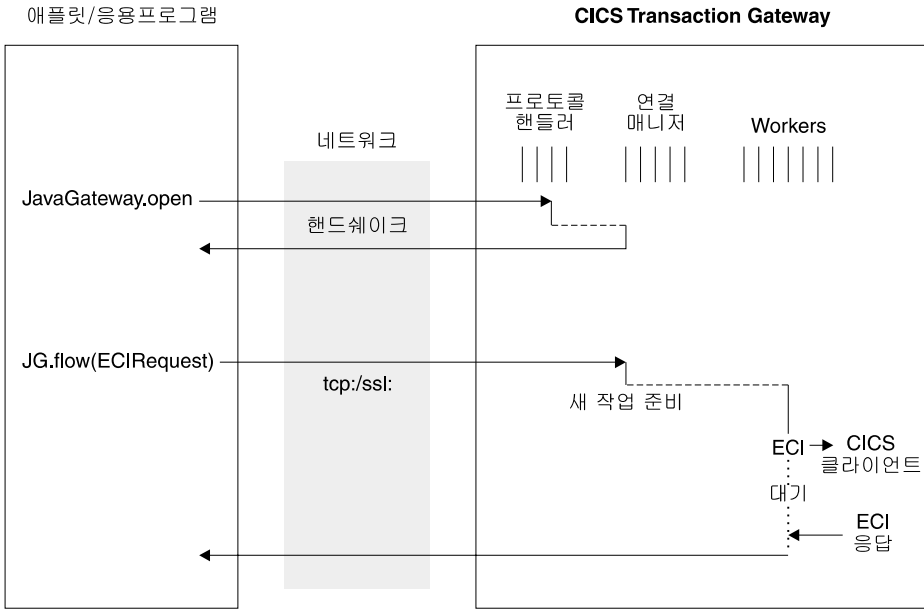


그림 2. TCP/IP 및 SSL용 CICS Transaction Gateway 쓰레드 모델. 이 프로토콜은 지속적 소켓을 가집니다.

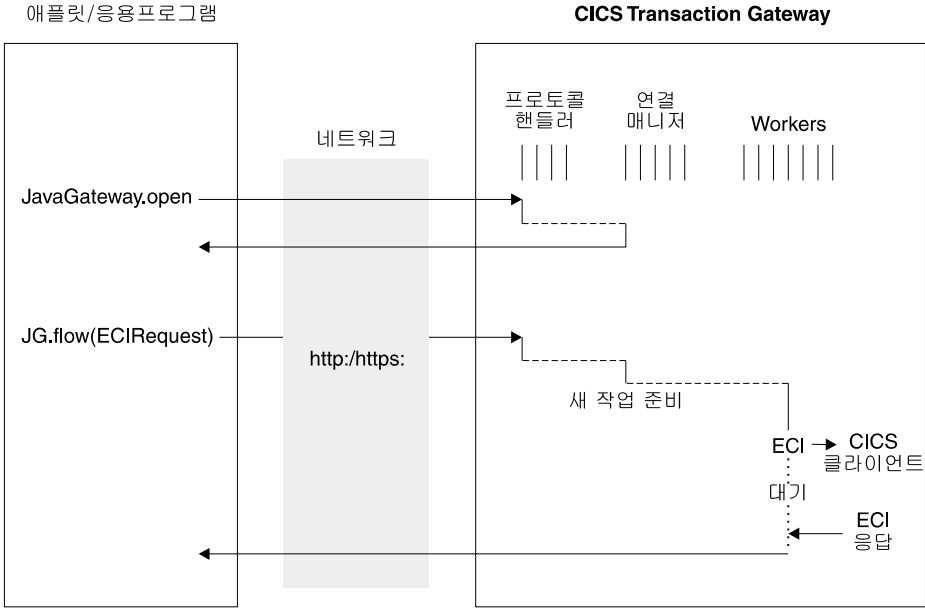


그림 3. HTTP/HTTPS용 CICS Transaction Gateway 쓰레드 모델. 이 프로토콜은 지속적 소켓을 가지지 않습니다.

표1에서는 여러 플랫폼에서 연결 매니저 및 Worker 쓰레드를 설정할 때 고려해야 하는 쓰레드 한계를 보여줍니다.

표1. CICS Transaction Gateway 플랫폼에서의 쓰레드 한계

운영 체제	최대 쓰레드 수: 시스템 범용 한계	프로세스 한계 : 쓰레드 수
Linux	최대 프로세스 수와 같음	1024(자세한 정보는 LinuxThreads 문서 참조)

Java가 쓰레드 메모리를 할당하는 방식에 대한 자세한 정보는 *CICS Transaction Gateway: Gateway Programming* 서적의 『Performance issues』를 참조하십시오.

외부 액세스 인터페이스(EPI, ECI, SI)

외부 액세스 인터페이스를 사용하면 비 CICS 응용프로그램에서 CICS 트랜잭션을 시작하거나 CICS 프로그램을 호출하여 CICS 자원에 액세스하거나 갱신할 수 있습니다. CICS 통신 기능과 함께 사용할 경우, 비 CICS 프로그램이 모든 CICS 시스템의 자원에 액세스하고 갱신할 수 있습니다. 외부 액세스 인터페이스는 CICS

응용프로그램용 GUI(Graphical User Interface) 프론트엔드 개발 및 CICS 시스템과 비 CICS 시스템의 통합 허용과 같은 활동을 지원하기도 합니다.

EPI(EXTERNAL PRESENTATION INTERFACE)를 사용하면 기존의 CICS 시스템 또는 새로운 응용프로그램용 어느 것으로든 GUI를 개발할 수 있습니다. 특히 변경할 필요가 없는 기존의 CICS 트랜잭션에 대해 새로운 GUI 프론트엔드를 개발할 경우 유용합니다. 응용프로그램은 EPI를 사용하여 트랜잭션 클라이언트와 통신하고, 클라이언트 시스템의 프리젠테이션 기능을 활용하여 일반 사용자와 통신할 수 있습니다.

CICS와 비 CICS 시스템과의 통합에는 주로 비 CICS 시스템의 프로그램과 CICS 프로그램 간의 사용자 정의 데이터 전달이 포함되며, ECI(EXTERNAL CALL INTERFACE)가 이 프로세스에 사용될 수 있습니다.

이런 경우, EPI 또는 ECI에 대한 선택 문제가 항상 명확한 것은 아닙니다. 두 인터페이스 모두를 비 CICS 응용프로그램과 CICS 프로그램 사이에서 데이터를 전달하는 데 사용할 수 있기 때문입니다. 그러나 EPI용 3270 데이터 스트림과 ECI용 COMMAREA에서의 응용프로그램 정의 형식의 메커니즘은 다릅니다.

EPI(External Presentation Interface)

EPI를 사용하면 비 CICS 응용프로그램이 연결될 CICS 서버에 3270 단말기로 간주될 수 있습니다. 즉, EPI 응용프로그램과 CICS 단말기는 CICS 서버에서 트랜잭션을 계획할 수 있습니다. 응용프로그램은 여러 서버의 기능을 동시에 사용할 수 있으며, 여러 3270 단말기 역할을 할 수 있습니다. 응용프로그램은 CICS 트랜잭션을 계획할 수 있으며, 이런 트랜잭션에 대해 이것이 핵심적인 기능입니다.

EPI 전반에 걸쳐 액세스를 지원하는 CICS 서버의 경우, 서버에서 실행하는 다른 CICS 트랜잭션은 CICS START 명령을 사용하여 초기의 단말기로 비 CICS 응용프로그램을 사용하는 트랜잭션을 계획할 수 있습니다. 비 CICS 응용프로그램이 EPI를 사용하여 CICS 서버에서 트랜잭션을 시작할 때 3270 데이터 스트림과 이벤트가 서버와 응용프로그램 사이에서 전달됩니다. 응용프로그램은 응용프로그램의 운영 환경에 적당한 방식으로 사용자에게 단말기 I/O의 내용을 제시할 수 있습니다.

다. 트랜잭션은 표준 트랜잭션 라우팅으로 다른 CICS 시스템에 라우트될 수 있습니다. 다른 CICS 시스템의 자원에는 Function shipping으로 액세스할 수 있습니다.

서버 트랜잭션은 3270 입력 및 출력을 사용하는 기존의 트랜잭션이 될 수 있습니다. 단, 몇 가지 제한사항이 있습니다.

ECI(External Call Interface)

ECI를 사용하면 비 CICS 응용프로그램에서 CICS 서버의 CICS 프로그램을 호출할 수 있습니다. 응용프로그램은 동시에 여러 서버에 연결될 수 있으므로, 동시에 여러 프로그램 호출을 미해결 상태로 둘 수 있습니다.

CICS 프로그램은 단말기 I/O를 수행할 수 없지만, 다른 모든 CICS 자원에 액세스하여 갱신할 수 있습니다. ECI를 사용하는 비 CICS 응용프로그램 또는 EXEC CICS LINK를 사용하는 CICS 프로그램에서 같은 CICS 프로그램을 호출할 수 있습니다. 데이터는 CICS의 경우와 비슷한 방법으로 COMMAREA를 사용하여 두 프로그램 사이에서 교환됩니다. COMMAREA 데이터 길이를 지정하여 성능을 최적화할 수 있습니다.

호출을 동기 또는 비동기로 수행할 수 있습니다. 동기 호출은 호출된 프로그램이 완료되면 제어를 리턴하므로, 리턴된 정보는 즉시 사용할 수 있습니다. 비동기 호출은 호출된 프로그램의 완료 여부를 참조하지 않고 제어를 리턴하므로, 응용프로그램이 정보의 사용 가능한 시기를 알려주도록 요청할 수 있습니다.

호출을 확장할 수도 있습니다. 즉, 한 논리적 작업단위(LUW)가 두 개 이상의 연속적인 호출을 담당할 수 있습니다. 각 논리적 작업단위에 대해 한 호출만 사용할 수 있는 경우에도 해당됩니다. 비동기 호출을 사용할 경우, 응용프로그램은 많은 논리적 작업단위를 동시에 관리할 수 있습니다.

호출된 프로그램은 자체 시스템의 자원을 갱신하고, DPL(DISTRIBUTED PROGRAM LINK)을 사용하여 다른 시스템의 CICS 프로그램을 호출하고, Function Shipping, DTP(Distributed Transaction Processing) 또는 OS/390®용 CICS® Transaction Gateway나 VSE/ESA™용 CICS® Transaction Gateway™ 환경에서는 FEPI(front end programming interface)를 사용하여 다른 CICS 시스템의 자원에 액세스할 수 있습니다.

외부 액세스 인터페이스에 대해서는 *CICS® Family: Client/Server Programming* 을 참조하십시오.

ESI(External Security Interface)

ESI를 사용하면 비 CICS 응용프로그램에서 APPC PEM(password expiration management)이 제공하는 서비스를 호출할 수 있습니다.

CICS를 갖춘 APPC PEM은 CICS 서버에 사용자 ID를 사인 온한 후 다음을 수행하여 암호에 대한 요청을 처리하는 APPC 구조의 사인 온 트랜잭션을 지원합니다.

- 사용자 식별 및 사용자의 ID 인증
- 암호가 만료되었는지 감정하는 동안 특정 사용자에게 알림
- 암호가 만료된 경우(또는 그 전에) 사용자가 자신의 암호를 변경할 수 있도록 함
- 현재 암호의 유효기간을 사용자에게 알림
- 특정 사용자 ID로 서버에 액세스하려 하는 허가받지 않은 시도에 대한 정보 제공

APPC PEM을 사용하려면 CICS Universal Client가 APPC를 통해 CICS 서버에 연결되어 있어야 합니다. RACF®(Resource Access Control Facility)와 같은 외부 보안 관리 프로그램(ESM)을 CICS 서버에 사용할 수 있어야 합니다. ESI 호출은 ECI 또는 EPI 응용프로그램 안에 포함될 수 있습니다.

CICS_EciListSystems 및 **CICS_EpiListSystems** 기능으로 리턴된 CICS 서버만 허용될 수 있습니다.

네트워크 보안

CICS Transaction Gateway는 SSL(Secure Sockets Layer)과 HTTPS 프로토콜을 사용하여 보안 통신을 제공합니다. 안전한 통신은 성공적인 인터넷 조작에 매우 중요합니다.

CICS Transaction Gateway에서의 네트워크 보안과 그 설치에 대해서는 69 페이지의 『제5장 연결 보안』에서 자세히 설명합니다

SSL(Secure Sockets Layer)

SSL은 인터넷을 통해 보안 및 개인 정보 보호를 제공하도록 개발된 핸드셰이크 프로토콜입니다. SSL 프로토콜을 사용하면 다음 사항이 보장됩니다.

기밀성 클라이언트와 서버 사이에서 교환되는 데이터는 암호화되므로, 해당 클라이언트(사용자 응용프로그램 또는 애플릿)와 해당 서버(CICS Transaction Gateway)만 데이터를 이해할 수 있습니다.

SSL은 보안 메커니즘으로 공용 키 암호화를 사용하여 서버와 클라이언트 사이에서 보안 키를 분배합니다. 공용 키 암호화는 암호화와 암호 해독에 비대칭 키 쌍을 사용하는 기법입니다. SSL의 경우, 암호(대칭) 키는 클라이언트와 서버 사이에서 공용 키 암호화를 사용하여 전달되며, SSL 연결을 통한 모든 통신을 암호화 및 암호 해독하는 데 사용됩니다. 도청을 시도하는 다른 파트에게는 데이터를 암호 해독하는 데 필요한 보안 키가 없으므로 이 암호화는 다른 파트로부터 데이터를 보호합니다. 따라서 신용카드 번호와 같은 개인 정보가 안전하게 전송됩니다.

완전성 메시지 전송에는 안전한 해싱 알고리즘에 기초한 메시지 완전성 검사가 포함됩니다. 이 알고리즘은 메시지를 보낼 때와 메시지를 받을 때 수행됩니다. 두 해시 값이 일치하지 않으면 수신자는 메시지가 함부로 변경되었을 수 있다는 경고를 받게 됩니다.

책임 책임은 디지털 신호로 보증되므로, 뭔가 잘못 될 경우 책임이 있는 파트를 식별할 수 있습니다.

인증 CICS Transaction Gateway의 SSL 프로토콜 구현에서는 서버 인증을 제공하므로, 클라이언트가 CICS Transaction Gateway와의 연결을 설정할 때 서버의 세부사항을 인증해야 합니다. 클라이언트 인증도 사용할 수 있으며, 이 경우 서버는 클라이언트의 세부사항을 인증하게 됩니다.

인증 메커니즘은 디지털 인증(X.509v3 인증)의 교환에 기초합니다. 이런 디지털 인증에는 시스템 이름과 공용 키 같은 엔티티와 서버의 디지털 사인에 대한 정보가 있습니다. 디지털 인증은 CA(Certificate Authority)에서 발행되며, CA의 개인용 키를 사용하여 암호화됩니다. CA의 공용 키를 사용하여 인증을 암호 해독할 수 있으면 인증 안에 있는 정보가 신뢰할 수 있는 것 즉, 인증이 인증을 요청한 사용자에게 실제로 전달된 것임을 의미합니다.

HTTPS

현재 대부분의 브라우저는 SSL을 사용하여 HTTP 서버에 연결하는 URL 접근 방법인 HTTPS를 지원합니다. HTTPS(HTTP+SSL)는 보안 트랜잭션을 처리하기 위한 HTTP의 일종입니다.

보안 연결은 보통 “https://someAddress”와 같은 URL을 통해 설정됩니다. 기본 HTTPS 포트 번호는 443이며, 이 번호는 IANA(Internet Assigned Numbers Authority)에서 할당한 것입니다.

키 및 인증

SSL 프로토콜은 공용 키 암호화를 사용하는데 공용 키 암호화는 ISO 인증 프레임워크(또는 X.509 프로토콜)와 함께 사용하면 좋습니다. 이 프레임워크는 네트워크에서의 인증에 사용됩니다.

가장 중요한 X.509 파트는 공용 키 인증을 위한 구조입니다. 신뢰성 있는 CA(Certification Authority)에서 각 사용자에게 고유한 이름을 지정하고, 이름과 사용자의 공용 키가 있는 사인된 인증서를 발행합니다.

CICS Transaction Gateway를 사용하면 CA에서 외부 사인 인증서를 받을 수 있거나, CA로 자신을 설정하여 “자가 사인” X.509 인증서를 발행할 수 있습니다. 외부 사인 인증서는 인터넷 사용에 적합한 반면, 자가 사인 인증서는 조직 안에서의 내부 사용에 적합할 수 있습니다.

X.509 디지털 인증은 Java 클래스 파일로 “캡슐화”되어, SSL 및 HTTPS 프로토콜에 사용될 수 있습니다. 이런 Java 클래스 파일을 키링 클래스라 합니다.

CICS Transaction Gateway에는 키링 클래스 관리를 위한 iKeyMan 도구가 들어 있습니다. 이 도구를 사용하면 키링 파일을 작성하고, 인증을 생성하며, 키링에 인증을 저장하고, 다양한 다른 관리 기능을 수행할 수 있습니다. 자세한 정보는 69 페이지의 『제5장 연결 보안』을 참조하십시오.

보안 종료

사용자가 공용 키 암호화와 같은 보안 조작을 정의할 수 있도록 하는 보안 종료가 제공됩니다. 이 기능은 데이터 압축에도 사용할 수 있습니다. 이런 기능에 대해 설명하는 몇몇 예제 소스 파일도 제공되어 있습니다.

보안 종료 메커니즘은 클라이언트 인증이 사용 가능할 때 X.509 클라이언트 인증을 인증하거나 분석하는 데에도 사용할 수 있습니다.

자세한 정보는 *CICS Transaction Gateway: Gateway Programming* 서적을 참조하십시오.

CICS Transaction Gateway 및 객체 요청 브로커

일부 웹 서버에는 OMG(Object Management Group)의 CORBA(Common Object Request Broker) 표준을 준수하는 ORB(Object Request Broker)가 있습니다. 이런 웹 서버에서는 CORBA IIOP(Internet InterORB Protocol)를 사용하여 원격 브라우저에서 호출할 수 있는 서블릿 객체를 개발할 수 있습니다. IIOP는 HTTP를 사용하여 브라우저로 전송됩니다.

몇몇 브라우저에는 ORB가 있으며, ORB를 통해 브라우저는 IIOP 요청을 발행하는 프로그램을 실행할 수 있습니다. 이 기능은 결국 IIOP로 맵핑되는 바인딩 메커니즘인 Java RMI(Remote Method Invocation)를 사용하여 통신하는 Java 애플릿에서 사용합니다. 이 기법으로 애플릿에서 IIOP를 사용하여 서블릿과 통신할 수 있고, 서블릿으로 CICS 트랜잭션을 호출할 수 있습니다.

기타 기능

*Host on-Demand*도 지원되어, Java 지원 웹 브라우저에서 EPI를 통해 CICS 3270 응용프로그램에 액세스할 수 있는 방법을 제공합니다. IBM eNetwork™ Host on-Demand 버전 2.0 이상이 필요합니다.

국지 *CICS Transaction Gateway*에 대한 지원도 제공됩니다. 따라서 Java 프로그램에서 네트워크를 사용하지 않고도 국지로 설치된 CICS Transaction Gateway

와 통신할 수 있습니다. 국지 CICS Transaction Gateway 구성 방법에 대해서는 67 페이지의 『국지 CICS Transaction Gateway 지원 사용 준비』를 참조하십시오.

Euro 지원

CICS Transaction Gateway는 데이터 흐름에서 Euro 문자의 사용을 지원합니다. CICS Transaction Gateway 클래스 CicsCpRequest는 Euro 지원 JDK에 적합하며 CICS Universal Client에서 CICS 서버로 전달하는 코드페이지/CCSID와 동일한 Java 인코드 이름을 리턴합니다

예를 들어, CICS Universal Client에서 CICS 서버로 코드페이지 5348(코드페이지 1252의 Euro 지원 버전)을 전달하면 Java 클라이언트에 리턴된 Java 인코딩은 Cp1252가 됩니다. Java 인코딩 Cp1252는 JDK 자체가 Euro를 지원할 경우 Euro를 지원합니다. Java 프로그램의 Euro 지원에 대한 자세한 정보는 해당 Java 문서를 참조하십시오.

액세스 기능

CICS Transaction Gateway의 액세스 기능은 Windows 운영 체제에서만 테스트되었습니다. 구성 도구를 제외한 제품에 액세스할 수 있습니다. ini 파일을 수동으로 편집하는 방안이 해결책으로 제안되었습니다. 액세스 기능 요구사항을 충족하는 텍스트 편집기에서 그 파일을 읽을 수 있습니다. 향후 릴리스에서는 이 문제가 해결될 것입니다.

제2장 설치 전의 계획

이 장에서는 지원되는 웹 서버 및 브라우저를 포함하여 하드웨어 및 소프트웨어 요구사항에 대해 설명하여 CICS Transaction Gateway 설치 계획에 도움을 줍니다. CICS Transaction Gateway에서 연결할 수 있는 CICS 서버가 나열되어 있습니다.

하드웨어 및 소프트웨어 요구사항에 대한 최신 변경사항은 제품의 **README** 파일을 참조하십시오.

하드웨어 요구사항

CICS Transaction Gateway는 관련 운영 체제 및 기타 필수조건 소프트웨어를 실행할 수 있는 모든 하드웨어에서 실행됩니다.

CICS Transaction Gateway의 하드 디스크 요구사항은(영문으로된 문서만 포함할 경우) **30MB**입니다.

설치 중 사용되는 임시 파일용으로 30MB가 더 필요합니다. 그리고 이중 최소한 12MB는 Windows 드라이브에 있어야 합니다.

지원 소프트웨어

다음과 같은 제품이 지원됩니다. 아래에 지정된 곳에서, 제품에 대한 CICS Transaction Gateway 테스트가 수행되었습니다.

표 2. 지원 제품 운영 체제

CICS Transaction Gateway는 다음 운영 체제에서 실행되고 테스트되었습니다.

- 서비스팩 25가 설치된 AIX V4.3.3(UNIX-UNIX 복사 프로그램 bos.net.uucp 포함)
- HP-UX 11.0
- S/390[®]용 Linux - SuSe 7.0(커널 레벨 2.2.16)
- OS/390 V2.8
- OS/390 V2.10
- Solaris 7(32비트 모드 및 64비트 모드)
- Solaris 8(32비트 모드 및 64비트 모드)
- zSeries 및 S/390용 TurboLinux Server 6
- 서비스팩 6a가 설치된 Windows NT Workstation V4.0 또는 서비스팩 6a가 설치된 Windows NT Server V4.0
주: Windows NT Terminal Server는 지원되지 않습니다.
- 서비스팩 1이 설치된 Windows 2000 Professional 또는 서비스팩 1이 설치된 Windows 2000 Server(Windows 2000 단말기 서비스 기능 구비)
- z/OS V1

웹 브라우저

- HTML/HTTP 기능: HTML V1.0을 지원하는 모든 브라우저는 이 제품과 함께 작동합니다.
- Java[™] 기능: JDK[™] 1.1을 지원하는 모든 웹 브라우저는 이 제품과 함께 작동합니다.

CICS Transaction Gateway는 다음 브라우저에서 테스트되었습니다.

- HotJava[™] 브라우저 V 1.1(Solaris용)
- Microsoft[®] Internet Explorer 5.0
- Netscape Communicator 4.7 및 6.0

주: Netscape Communicator와 함께 제공된 JVM 안에서 실행 중인 애플릿이 CICS Transaction Gateway와 연결되지 못했습니다. 메시지 번호 CCL6652E가 애플릿으로 리턴됩니다. 현재 이 문제점에 대한 해결책은 없습니다.

표 2. 지원 제품 (계속)

Telnet 클라이언트

Telnet을 통해 CICS 클라이언트를 실행할 경우, 특정 Telnet 클라이언트로 인해 화면표시에 문제가 발생할 수 있습니다. 예를 들어, Telnet 세션에서 특정 길이를 초과하는 메시지 행을 자를 수도 있습니다. 이 문제는 주로 사용 중인 Telnet 클라이언트 또는 사용하려는 단말기 유형의 문제입니다. 현재 이 문제점에 대한 해결책은 없습니다.

JDK 레벨

CICS Transaction Gateway는 다음과 같은 JDK 레벨을 지원합니다.

- OS/390: IBM Java SDK 1.3
- AIX: IBM Java SDK 1.3(서비스 릴리스 6 이상)
- HP-UX: Java SDK 1.3
- Linux for S/390: IBM Java SDK 1.2.2
- Solaris: Sun Java SDK 1.3
- Windows: IBM Java SDK 1.3(서비스 릴리스 6 이상)
- OS/390, IBM Java SDK 1.2.2(서비스 릴리스 10 및 10a)를 제외한 모든 운영 체제 또한 지원됩니다.

주: CICS Transaction Gateway는 JDK 1.1 지원 웹 브라우저에서 실행되는 애플릿을 지원합니다.

표 2. 지원 제품 (계속)

CICS 서버

CICS 서버는 실시간 다중 사용자 응용프로그램을 실행하며, 연관된 자원과 데이터를 관리합니다. CICS Transaction Gateway는 다음과 같은 서버에서 테스트되었습니다.

- MVS/ESA™용 CICS® V4.1(SNA 및 TCP62를 통한 통신 설정, 사인 온 지원 단말기의 경우 APAR PQ30167 적용)
- OS/400®용 CICS® V4.4
- OS/2®용 CICS® Transaction Gateway V4.1(CSD 3)
- CICS® OS/390®용 Transaction Server V1.2(사인 온 지원 단말기의 경우 APAR PQ30168 적용)
- CICS® OS/390®용 Transaction Server V1.3(사인 온 지원 단말기의 경우 APAR PQ30168 및 EXCI의 경우 OS/390® APAR PQ38644 적용)
- VSE/ESA™용 CICS® Transaction Gateway 1.1.0(사인 온 지원 단말기의 경우 APAR PQ30170 적용)
- VSE/ESA™용 CICS® Transaction Gateway 1.1.1
- CICS/VSE 2.3(사인 온 지원 단말기의 경우 APAR PQ30169 적용)
- TXSeries V4.2 + PTF 9(HP-UX)
- TXSeries V4.3 + PTF 4(Windows® NT, AIX®, Solaris)
- OS/2® 및 Windows NT용 VisualAge® CICS Enterprise Application Development(응용프로그램 개발 전용) V3.1(CSD 3)

웹 서버

CICS Transaction Gateway는 다음 서버에서 테스트되었습니다.

- Apache HTTP Server 1.3.12(SuSe V7.0)
- Domino™ Go Webserver V4.6.2(Windows, AIX 및 Solaris)
- IBM HTTP Server 1.3(Windows, AIX 및 Solaris)
- Microsoft® Internet Information Server V5.0(Windows)

응용프로그램 서버

CICS Transaction Gateway는 다음 서버에서 테스트되었습니다.

- Windows 및 AIX용 IBM WebSphere™ Application Server V3.5.3 Enterprise Edition(PTF 3)
- Windows 및 AIX용 IBM WebSphere™ Application Server V3.5.3 Advanced Edition(PTF 3)

주: 단말기 서블릿(서블릿 샘플) 또한 위의 응용프로그램 서버에서 테스트되었습니다.

TCP/IP 통신

운영 체제는 TCP/IP를 지원합니다.

표 2. 지원 제품 (계속)

TCP62 통신 CICS Transaction Gateway는 TCP62를 지원합니다.
컴파일러 및 응용프로그램 개발 도구 CICS Transaction Gateway는 다음 컴파일러 및 응용프로그램 개발 도
 구에서 테스트되었습니다.

AIX

- AIX용 IBM C V4.4
- AIX용 IBM C/Set++ V3.6.6
- IBM VisualAge C++ V4.0
- IBM VisualAge C++ V5.0

HP-UX

- HP ANSI C B.11.01.07
- HP aC++ B.11.01.06

S/390용 Linux

- GNU compiler C runtime V2.95.2
- GNU compiler C++ runtime V2.95.2

Solaris

Sun WorkShop C++ V5.0

주: C++ 컴파일러에는 사용 가능한 모든 서비스 패치가 적용되어 있
 어야 합니다.

Windows

- IBM VisualAge for Java V3.5.3
- IBM VisualAge Interspace™ V5.2
- Microsoft Visual C++ V6.0
- Microsoft Visual Basic V6.0
- Microsoft Visual Basic Script(VBScript) V5.0

주: CICS Transaction Gateway는 MTS 또는 COM+의 MTS 구성요소
 가 있는 COM 라이브러리를 지원하지 않습니다.

기타 도구 CICS Transaction Gateway는 다음 도구를 지원합니다.

- Adobe Acrobat
- Linux용 GhostView

GPL 사용권 및 저작권 문제

이 제품은 glibc 패키지의 라이브러리인 libnsl.so, libm.so, libdl.so, ld.so, libc.so 및 libpthread.so를 사용합니다. 이 라이브러리의 여러 가지 저작권 선언문 및 사용권 조건에 대해서는 시스템의 glibc 패키지를 참조하십시오.

또한 이 제품은 ncurses 패키지의 libncurses.so 또한 사용합니다. 이 라이브러리에 적용할 수 있는 저작권 선언문 및 사용권 조건에 대해서는 시스템에 있는 이 패키지를 참조하십시오.

주의사항: libstdc++ 또는 egcs-c++ 패키지 사용시 GNU GPL 사용권 조건을 따라야 합니다. 이 조건에서는 특정 환경에서 소스 코드를 제공하도록 요구할 수 있습니다. IBM은 예를 들어, CICS 클라이언트 및 Gateways C++ 라이브러리의 소스 코드를 제공하지 않습니다.

제3장 설치

이 장에서는 Linux용 CICS[®] Transaction Gateway의 설치 방법에 대해 설명합니다.

Linux용 CICS[®] Transaction Gateway는 실행 가능 스크립트 파일이 들어 있는 압축된 tar 파일로 제공됩니다. 실행 시 스크립트 파일은 라이브러리 파일, 명령, 메시지 및 조정(.INI) 파일을 작성합니다.

Linux용 CICS[®] Transaction Gateway 설치

Linux용 CICS[®] Transaction Gateway를 설치하려면 다음과 같이 수행하십시오.

1. 루트로 로그인합니다.
2. 미디어에서 tar 파일을 찾으십시오. 이 파일의 이름은 ctg-4.0.0.tar.Z입니다.
3. ctg-4.0.0.tar.Z 파일을 하드 디스크에 있는 작업 디렉토리로 복사하십시오.
4. 파일을 압축해제하여 파일을 추출하십시오.

```
uncompress ctg-4.0.0.tar.Z  
tar -xvf ctg-4.0.0.tar.Z
```

그러면 현재 디렉토리에 ctg_install이라는 스크립트 파일이 작성됩니다.

5. 다음과 같이 입력하여 CICS Transaction Gateway를 설치하십시오.

```
ctg_install
```

주: 설치 프로세스는 *uudecode* 기능을 사용합니다. 이 기능은 *UNIX-to-UNIX* 복사 프로그램의 일부입니다. CICS Transaction Gateway를 설치하기 전에 시스템에 이 프로그램이 있는지 확인하십시오.

주: Linux 커널이 대규모 클라이언트 데이터 흐름을 허용하도록 구성하는 방법에 대해서는 *CICS Transaction Gateway: Linux 클라이언트 관리*의 『UNIX[®] 시스템의 내부 클라이언트 통신』을 참조하십시오.

자국어 지원

다음 명령을 입력하여 Linux용 CICS® Transaction Gateway에서 표시하는 사용자 메시지의 언어를 선택할 수 있습니다.

```
mkclimsgs XX Code Set
```

여기서 XX는 2자의 메시지 언어입니다. 사용 가능한 언어 목록을 보려면 매개변수 없이 명령을 입력하십시오.

Linux용 CICS 클라이언트 - 사용자 메시지

언어	로케일	코드 세트
us US 영어	en_US	ISO-8859-1
	EN_US	UTF-8
fr 불어	fr_FR	ISO-8859-1
	FR_FR	UTF-8
de 독어	de_DE	ISO-8859-1
	DE_DE	UTF-8
it 이탈리아어	it_IT	ISO-8859-1
	IT_IT	UTF-8
es 스페인어	es_ES	ISO-8859-1
	ES_ES	UTF-8
tr 터키어	tr_TR	ISO-8859-9
	TR_TR	UTF-8
jp 일어	ja_JP	EUCJP
	JA_JP	UTF-8
kr 한글	ko_KR	EUCKR
	KO_KR	UTF-8
cn 중국어	zh_CN	EUCCN
	ZH_CN	UTF-8

구문: mkclimsgs <Language> <Code Set>

원격 시스템에서 X-Windows 사용

원격 시스템에서 X-Windows를 사용할 경우 예를 들면, 구성 도구 및 iKeyman에 액세스하려면 DISPLAY 환경 변수를 설정하여 응용프로그램이 해당 시스템에서 그 창을 표시하도록 해야 합니다.

디스플레이 시스템(창을 표시할 시스템)에서 다음 명령을 입력하십시오.

```
xhost +appl
```

여기서 *appl*은 응용프로그램을 수행하기 위해 사용되는 시스템의 네트워크 이름입니다.

응용프로그램 시스템에서, 응용프로그램을 수행하기 전에 다음 명령을 입력하십시오.

```
DISPLAY=disp:0
```

다음 명령도 입력하십시오.

```
export DISPLAY
```

여기서 *disp*는 창이 표시될 시스템의 호스트 이름입니다(*disp* 다음에는 콜론과 표시 id(보통 0)가 옵니다). 그러면 응용프로그램 창이 *disp* 시스템에 표시됩니다.

CICS Transaction Gateway 설치제거

CICS Transaction Gateway를 설치제거하려면 다음 명령을 사용하십시오.

```
rpm -e ctg-4.0.0-1
```

제4장 구성

이 장에서는 다음과 같은 내용을 설명합니다.

- 『CICS Transaction Gateway에 대한 프로그래밍 환경 구성』
- 32 페이지의 『구성 도구 사용』
- 54 페이지의 『구성 변환 도구』
- 56 페이지의 『구성 파일 편집』
- 58 페이지의 『구성 이름 바꾸기 및 파일 맵핑』
- 59 페이지의 『클라이언트 키보드 맵핑 구성』
- 63 페이지의 『화면 색상 사용자 정의』
- 67 페이지의 『국지 CICS Transaction Gateway 지원 사용 준비』

CICS Transaction Gateway에 대한 프로그래밍 환경 구성

JVM(Java Virtual Machine)에서는 CLASSPATH 환경 변수를 사용하여 클래스와 클래스가 있는 zip 또는 jar 아카이브를 찾습니다. JVM으로 클래스 파일에 액세스하려면 클래스 파일 또는 아카이브가 있는 디렉토리의 전체 경로를 지정해야 합니다.

CICS Transaction Gateway와 함께 사용하기 위해 Java 응용프로그램을 컴파일하고 실행하려면 다음의 전체 경로를 CLASSPATH 환경 변수에 추가하십시오.

- ctgclient.jar
- ctgserver.jar(국지 게이트웨이를 사용할 경우)
- cfwk.zip(게이트웨이에 대한 SSL 연결을 사용할 경우)

해당 아카이브는 CICS Transaction Gateway가 설치되어 있는 디렉토리의 클래스 서브디렉토리에 있습니다.

Linux에서 CLASSPATH 설정

CLASSPATH 환경 변수를 설정하려면 다음과 같은 셸 명령을 사용하십시오.

```
export CLASSPATH=./opt/classes:/opt/ctg/classes/ctgclient.jar:${CLASSPATH}
```

이 셸 명령은 현재 디렉토리인 /opt/classes 디렉토리 및 JVM에서 액세스할 수 있는 아카이브 /opt/ctg/classes/ctgclient.jar에 클래스를 작성합니다.

홈 디렉토리에 있는 .profile 파일에 셸 명령을 입력하여, 시스템에 로그인할 때 CLASSPATH 환경 변수를 자동으로 설정할 수도 있습니다.

CLASSPATH 환경 변수 값을 표시하려면 다음 셸 명령을 사용하십시오.

```
echo $CLASSPATH
```

구성 도구 사용

구성 도구를 사용하여 CICS Transaction Gateway 및 CICS Universal Clients의 구성 매개변수를 설정할 수 있습니다.

구성 도구를 실행하기 전에 다음 정보를 알아두십시오.

- CICS 서버에 연결하는 데 사용할 프로토콜(TCP/IP, TCP62)
- 서버의 호스트 이름 또는 IP 주소
- CTG를 연결될 서버의 포트 번호
- 게이트웨이에 사용할 프로토콜(TCP, SSL, HTTP 및 HTTPS)

구성 도구를 원격으로 사용하려면 28 페이지의 『원격 시스템에서 X-Windows 사용』에 설명된 명령을 사용하여 표시화면을 반출해야 합니다.

구성 도구를 시작하려면 **ctgcfg** 명령을 입력하십시오.

작업 안내서

구성 도구를 처음 실행할 때 작업 안내서는 시스템에 대한 기본 정보를 정의하도록 도와줍니다. CICS 서버에 대한 세부사항을 입력하고 사용할 프로토콜을 지정할 수 있습니다. 구성 도구를 사용하여 나중에 여기에 입력한 설정을 변경할 수 있습니다. 작업 안내서를 생략하고, 작업 안내서의 열기 화면에서 취소 버튼을 눌러 구성 도구로 직접 이동할 수 있습니다.

작업 안내서의 화면 완료에 대한 도움말은 온라인 도움말이나 『구성 도구 인터페이스』의 관련 절을 참조하십시오.

구성 파일

기본적으로 구성 세부사항은 CTG를 설치한 bin 서브디렉토리의 CTG.INI 파일에 저장됩니다. 구성 도구를 통해서만 이 파일을 편집하는 것이 좋습니다.

CICSCLI 환경 변수를 설정하면 구성 파일의 다른 위치를 지정할 수 있습니다. CICS Transaction Gateway 로그는 CTG.INI의 위치를 표시합니다. CTG.INI가 없으면 CICS Transaction Gateway는 실행되지 않습니다.

bin에 CTG.INI가 없는 경우에도 작업 안내서는 실행됩니다. 구성 파일의 다른 위치를 지정했으며 기존 구성을 편집하려는 경우, 취소를 눌러 작업 안내서를 생략하십시오. 일단 구성 도구가 실행되면 구성 파일을 여십시오.

구성 도구 인터페이스

화면에는 다음과 같은 네 가지 영역이 있습니다. 네 가지 영역은 아래에 별도로 설명되어 있습니다.

1. 메뉴 표시줄(파일, 편집, 도구, 도움말 풀다운 메뉴로 구성)
2. 다음 아이콘이 있는 도구 모음: 새 파일, 열기, 저장, 잘라내기, 복사, 붙여넣기, 새 서버, 트레이스 설정
3. 트리 구조
4. 설정 패널

다섯번째 영역인 상태 표시줄에는 사용 중인 구성 파일과 운영 체제가 표시됩니다. 상태 표시줄은 표시 전용입니다.

네 가지 입력 영역 간에 이동하려면 F10, Escape 및 Tab 키를 사용하십시오.

- 메뉴를 활성화하려면 F10을 누르십시오.
- 선택하지 않고 풀다운 메뉴를 닫으려면 Escape 키를 누르십시오.
- 도구 모음, 트리 구조, 설정 패널 간에 이동하려면 Tab 키를 사용하십시오. Tab 키를 반복하여 누르면 설정 패널의 모든 객체를 탐색한 다음, 모든 아이콘을 탐색하고, 마지막으로 트리 구조로 돌아옵니다.

구성

- Tab 키를 눌렀는데 다음 영역으로 이동하지 못한 경우, Tab 키를 다시 누르기 전에 F10 키를 누른 다음 Escape 키를 누르십시오(메뉴를 활성화하고 닫음).

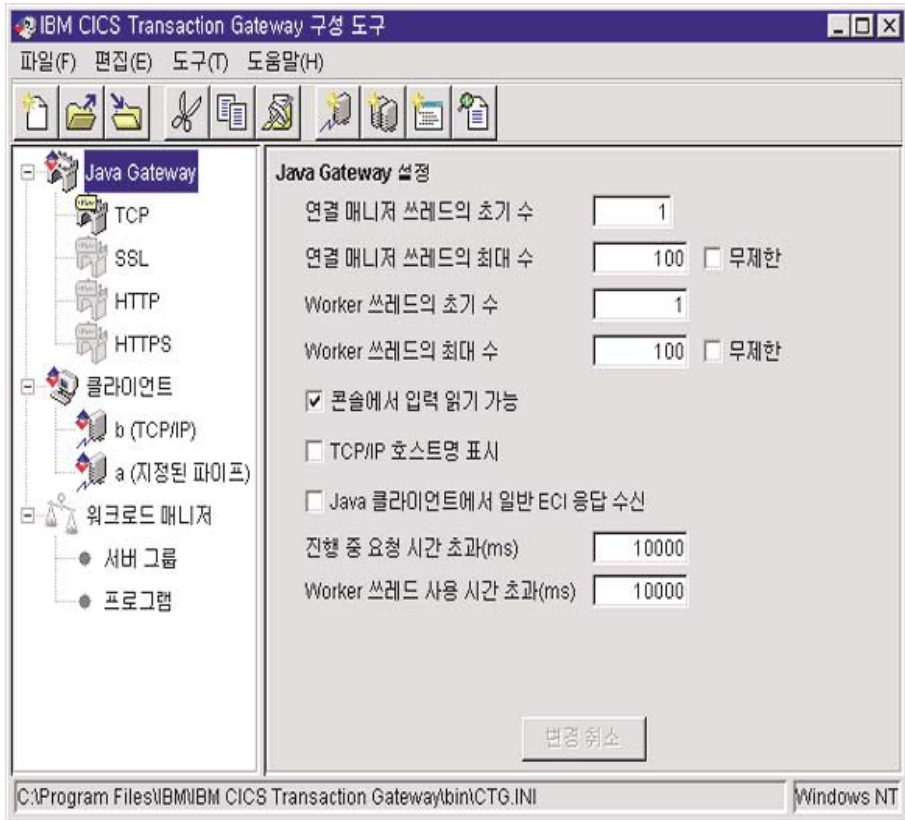


그림 4. 구성 도구

주: 사용자 플랫폼에서는 구성 도구가 그림4와 같이 나타나지 않을 수 있습니다.

트리 구조

트리 구조(그림4 참조)를 이용하면 모든 구성 설정을 탐색할 수 있습니다. 루트 노드 유형은 다음과 같습니다.

Java 게이트웨이

최대 4개의 서브노드까지 들어갈 수 있습니다. 즉, CICS Transaction Gateway에서 각 프로토콜당 하나를 사용할 수 있습니다(TCP, SSL, HTTP 및 HTTPS).

클라이언트 각 서버 정의에 대해 한 서브노드가 있습니다.

액세스 기능: 노드와 서브노드 간에 이동하려면 위쪽 및 아래쪽 화살표키를 사용하십시오. 구성 도구를 열면 모든 노드가 확장됩니다. 노드를 접으려면 해당 노드로 이동한 다음 왼쪽 화살표키를 누르십시오. 노드를 확장하려면 오른쪽 화살표키를 누르십시오.

메뉴 표시줄

메뉴 표시줄에는 파일, 편집, 도구 및 도움말 폴더가 있습니다.

파일 폴더에는 다음 선택항목이 있습니다.

새 구성 새로운 구성을 작성합니다.

열기 기존 구성을 엽니다.

저장 현재 구성을 저장합니다. 새로운 구성이 작성되면 파일 이름은 CTG.INI(bin 서브디렉토리에 있음)입니다.

다른 이름으로 저장

구성 파일의 기본 경로와 이름을 덮어쓰기합니다.

나감 구성 도구를 나갑니다. 구성을 저장할 것인지 여부를 묻습니다.

편집 폴더에는 각기 표준 텍스트 기능을 수행하는 잘라내기, 복사, 붙여넣기 선택항목이 있습니다.

도구 폴더에는 다음 선택항목이 있습니다.

트레이스 설정값

트레이스 설정값 대화상자를 표시합니다.

새 서버 새로운 서버 정의를 작성하기 위해 작업 안내서를 시작합니다.

서버 삭제 트리 구조에서 서버 노드를 삭제합니다.

도움말 폴더에는 다음 선택항목이 있습니다.

문맥 도움말 현재 화면 문맥에 따라 온라인 도움말 정보를 표시합니다. 예를 들면, 특정 구성 설정값.

목차 온라인 도움말 정보의 목차를 표시합니다.

색인 온라인 도움말 정보의 주제 색인을 표시합니다.

제품 정보 구성 도구의 버전 정보 및 기타 정보를 표시합니다.

도구 모음

도구 모음에는 메뉴 표시줄의 기능과 같은 기능이 있지만, 다른 이름으로 저장 또는 나감 선택항목은 제외됩니다. 아이콘에는 풍선 도움말이 있으므로, 커서를 이동할 때 선택항목에 대해 설명하는 텍스트 상자가 나타납니다.

패널 설정

트리 구조에서 노드를 선택하면 관련 설정값 패널이 표시됩니다. 패널의 설정값은 CTG.INI 파일의 매개변수에 해당됩니다.

각 설정값 패널에는 변경사항 취소 단추가 있어 작성한 변경사항을 취소할 수 있습니다.

CICS Transaction Gateway 설정값 구성

게이트웨이 설정값 패널을 표시하려면 트리 구조에서 게이트웨이 노드를 선택하십시오. 설정값은 CTG.INI 파일의 게이트웨이 섹션에 있는 매개변수에 맵핑됩니다.

구성 도구를 사용하면 게이트웨이 명령 행 선택항목을 사용하여 지정할 수 있는 매개변수의 사전설정 값을 제공할 수 있습니다.

게이트웨이를 시작할 때 구성 도구를 사용하여 정의한 매개변수를 연관된 명령 행 선택항목을 통해 지정할 경우, 명령 행 설정값이 우선합니다.

일반 게이트웨이 설정값

일반 게이트웨이 설정값은 다음과 같습니다.

연결 매니저 스레드의 초기 수: 초기 ConnectionManager 스레드 수를 입력하십시오. 기본값은 1입니다.

이 설정은 `ctgstart -initconnect` 명령으로 대체할 수 있습니다.

연결 매니저 스레드의 최대 수: 최대 ConnectionManager 스레드 수를 입력하십시오. 기본값은 100입니다.

무제한 선택란을 선택하면 ConnectionManager 쓰레드 수에는 한계가 적용되지 않습니다.

이 설정은 `ctgstart -maxconnect` 명령으로 대체할 수 있습니다.

쓰레드 한계에 대한 정보는 13 페이지의 표1을 참조하십시오.

Worker 쓰레드의 초기 수: 초기 Worker 쓰레드 수를 입력하십시오. 기본값은 1입니다.

이 설정은 `ctgstart -initworker` 명령으로 대체할 수 있습니다.

Worker 쓰레드의 최대 수: 최대 Worker 쓰레드 수를 입력하십시오. 기본값은 100입니다.

무제한 선택란을 선택할 경우, Worker 쓰레드 수에는 한계가 적용되지 않습니다.

이 설정은 `ctgstart -maxworker` 명령으로 대체할 수 있습니다.

쓰레드 한계에 대한 정보는 13 페이지의 표1을 참조하십시오.

콘솔의 입력 읽기 가능: 콘솔에서 입력을 읽을 수 있도록 하려면 이 선택란을 선택하십시오.

콘솔의 입력 읽기는 기본적으로 사용 가능합니다.

이 설정은 `ctgstart -noinput` 명령으로 대체할 수 있습니다.

TCP/IP 호스트명 표시: 메시지에 TCP/IP 호스트 이름을 표시하려면 이 선택란을 선택하십시오.

이 선택항목을 사용하면 메시지에 TCP/IP 주소가 표시되는 방식을 선택할 수 있습니다. 기본적으로, CICS Transaction Gateway는 TCP/IP 주소를 숫자 양식의 메시지로 표시합니다. 이 선택항목을 사용 가능하게 하면 CICS Transaction Gateway에서 DNS(도메인 이름 시스템)를 사용하여 숫자 TCP/IP 주소를 메시지의 심볼릭 TCP/IP 호스트 이름으로 변환합니다. 그러면 메시지는 더 읽기 쉬워집니다.

주: 이 선택항목을 선택하면 일부 시스템 성능이 크게 저하될 수 있습니다.

Java 클라이언트에서 일반 ECI 응답 수신: Java 클라이언트에서 CICS Transaction Gateway의 일반 ECI 응답을 확보할 수 있도록 하려면 이 선택란을 선택하십시오.

일반 응답은 Call_Type: ECI_GET_REPLY 또는 ECI_GET_REPLY_WAIT를 사용하여 확보된 응답입니다. 특정 응답은 Call_Type:

ECI_GET_SPECIFIC_REPLY 또는 ECI_GET_SPECIFIC_REPLY_WAIT를 사용하여 확보된 응답입니다. 이 설정은 국지 게이트웨이에는 적용되지 않습니다.

진행 중 요청 시간 초과: 밀리초 단위로 값을 입력하십시오.

Java 클라이언트 프로그램이 게이트웨이로부터 연결해제될 경우, 게이트웨이가 그 프로그램을 대신하여 요청을 처리할 수 있습니다. 작업이 계속 진행될 경우, Java 클라이언트 대신 요청을 관리하던 ConnectionManager가 시간 종료될 때까지 진행 중인 요청이 완료되기를 기다립니다. 이 후에도 진행 중인 요청이 있으면 ConnectionManager는 계속해서 그 요청의 처리를 수행한 후 새로운 연결에 사용 가능한 것으로 표시합니다. 기본적으로, 이 시간 종료는 10 000밀리초로 설정되지만, 그 기본값을 대체할 값을 입력할 수 있습니다.

이 값을 0으로 설정하면 ConnectionManager는 진행 중인 요청이 완료되기를 기다리지 않습니다.

Worker 스레드 사용 시간 종료: 밀리초 단위로 값을 입력하십시오.

ConnectionManager에서 요청을 허용하면 Worker 스레드를 할당하여 그 요청을 실행해야 합니다. 그러나 Worker가 시간 종료 시간 안에 사용 가능한 상태가 되지 않으면 그 요청을 거부하는 메시지가 송신되어 요청이 실행되지 않습니다. 기본적으로, 이 시간 종료는 10 000밀리초로 설정되지만 그 기본값을 대체할 값을 입력할 수 있습니다.

이 값을 0으로 설정할 경우, Worker가 즉시 사용 가능하지 않으면 그 요청은 거부됩니다.

TCP 프로토콜 설정값

TCP의 설정값을 표시하려면 **TCP** 서브노드를 선택하십시오.

프로토콜 핸들러 사용 기능: 이 프로토콜을 사용하도록 CICS Transaction Gateway를 구성하려면 이 선택란을 선택하십시오.

포트: 다음과 같이 프로토콜의 TCP/IP 포트 번호를 입력하십시오.

TCP/IP의 기본 포트는 2006입니다.

ctgstart -port 명령을 사용하면 TCP 프로토콜에 대한 이 설정을 대체할 수 있습니다.

핸들러 웨이크업 시간 종료: 밀리초 단위로 값을 입력하십시오.

이 설정은 프로토콜 핸들러가 인바운드 연결을 허용한 이후로 얼마나 자주 개시(wake)되는지를 제어합니다. 핸들러가 개시될 때 게이트웨이가 중지되는지 확인하므로 이 값은 게이트웨이가 제대로 종료되는 데 소요되는 시간에 영향을 미칩니다. 이 값을 0으로 설정하면 새로운 연결이 허용될 때만 핸들러가 개시되므로, 그 시간이 될 때까지 게이트웨이는 제대로 종료되지 않게 됩니다.

연결 시간 초과: 밀리초 단위로 값을 입력하십시오.

새로운 연결이 허용될 때 이 값은 ConnectionManager 스레드가 사용 가능해지는 동안 프로토콜 핸들러가 대기하는 시간을 지정합니다. ConnectionManager 스레드가 이 시간 안에 사용 가능해지지 않으면 연결은 거부됩니다. 이 값을 0으로 설정할 경우, ConnectionManager가 즉시 사용 가능하지 않으면 그 요청은 거부됩니다.

유휴 시간 종료: 밀리초 단위로 값을 입력하십시오.

이 설정은 연결이 대기 상태에 남아 있을 수 있는 시간을 지정합니다. 유휴 시간 종료 기간은 요청이 마지막으로 연결로 흘러 갔을 때부터 계산됩니다. 유휴 시간 종료기가 만기되면, 연결을 대신하는 작업이 진행 중인 경우 그 작업은 연결되어 있을 수 있더라도 연결은 끊어집니다. 이 값을 설정하지 않거나 0으로 설정하면 유휴 연결은 끊어지지 않습니다.

서버에 이 기능에 대한 지원이 설치되어 있는 경우 시간 종료 후에만 정리 처리가 수행됩니다.

Ping 시간 빈도: 밀리초 단위로 값을 입력하십시오.

이 값은 클라이언트가 계속 사용 중인지 확인하기 위해 게이트웨이에서 PING 메시지가 접속된 클라이언트에 송신되는 횟수를 지정합니다. 다음 PING 메시지가 송신될 때까지 PONG 응답이 수신되지 않으면 그 연결은 끊어집니다. 다시 말하지만 연결 대신 작업이 계속 진행 중이면(drop working connections 값 설정에 따라) 연결된 상태로 남아 있을 수 있습니다. 이 값을 설정하지 않거나 0으로 설정하면 PING 메시지가 송신되지 않습니다.

작동 중인 연결 제거: 작업이 해당 연결 대신 여전히 진행 중일 *경우에도* 유틸리티 시간 종료나 PING/PONG 실패로 연결해제가 되도록 지정하려면 이 선택란을 선택하십시오.

SO_LINGER 설정: 이 핸들러에서 사용하는 소켓의 SO_LINGER 설정을 입력하십시오. SO_LINGER 설정은 소켓 닫기 지연값을 초 단위로 지정합니다. 이 값을 입력하지 않거나 0으로 설정하면 이 프로토콜 핸들러에서 사용하는 어떤 소켓에도 SO_LINGER를 사용할 수 없습니다.

SO_LINGER가 사용 가능하며 데이터 전송이 완료되지 않은 경우, 데이터가 전송될 때까지 또는 연결이 시간 종료될 때까지 닫기 호출은 호출 프로그램을 블로킹합니다. SO_LINGER가 사용 불가능화된 경우, 닫기 호출은 호출자를 블로킹하지 않고 리턴되며 TCP/IP는 여전히 데이터 보내기를 시도합니다. 일반적으로 이런 전송은 성공하긴 하지만 보장할 순 없습니다. TCP/IP는 지정된 시간 동안만 보내기 요청을 반복하기 때문입니다.

클라이언트가 보안 클래스를 사용하도록 함: 게이트웨이가 보안 클래스를 사용하는 연결만 허용하도록 하려면 이 선택란을 선택하십시오.

Java 클라이언트 프로그램에서 게이트웨이에 연결할 때 그 연결에서 사용해야 하는 보안 클래스 쌍을 지정할 수 있습니다. 그러나 기본적으로 게이트웨이는 이 보안 클래스 쌍을 지정하지 않는 프로그램으로부터의 연결도 허용합니다.

게이트웨이에서 액세스할 수 있는 xxxServerSecurity 클래스 세트를 제어하여 올바른 보안 클래스를 제어할 수 있습니다.

CICS Transaction Gateway 보안 종료에 대한 자세한 정보는 19 페이지의 『보안 종료』를 참조하십시오.

SSL 프로토콜 설정값

SSL의 설정값을 표시하려면 **SSL** 서브노드를 선택하십시오. SSL에는 다음과 같은 설정값도 있다는 점을 제외하면 설정값은 TCP의 경우와 같습니다.

클라이언트 인증 사용: CICS Transaction Gateway에 대한 클라이언트 인증을 사용 가능화하려면 이 선택란을 선택하십시오. 기본값은 클라이언트 인증 사용 불 가능화입니다.

클라이언트 인증이 사용 가능하면 **ssl:** 또는 **https:** 핸들러에 대해 시도한 연결에서 클라이언트가 소유한 클라이언트 인증(또는 Digital ID)을 제시해야 합니다.

클라이언트에 대해 클라이언트 인증을 확보하는 방법에 대해서는 81 페이지의 『SSL 클라이언트 구성』을 참조하십시오.

키링 클래스명: 서버 키링의 클래스 이름을 입력하십시오.

서버 키링은 클라이언트에 연결하기 위해 해당 서버를 식별하는 데 사용되는 유효한 x.509 인증서로 구성됩니다. 이 키링 클래스는 이 제품과 함께 제공되는 SSL 도구를 사용하여 생성됩니다.

SSL와 연관 키링 클래스에 대해서는 69 페이지의 『제5장 연결 보안』을 참조하십시오.

키링 암호: 생성 프로세스 중 지정한 서버 키링 클래스의 암호를 입력하십시오.

SSL의 기본 포트는 8050입니다.

HTTP 프로토콜 설정값

HTTP의 설정값을 표시하려면 **HTTP** 서브노드를 선택하십시오. 설정값은 TCP의 경우와 같습니다.

HTTP의 기본 포트는 8080입니다.

HTTPS 프로토콜 설정값

HTTPS에 대한 설정값을 표시하려면 **HTTPS** 서브노드를 선택하십시오. HTTPS에는 클라이언트 인증 사용, 키링 클래스명 및 키링 암호 설정도 있다는 점을 제외하면 설정값은 HTTP의 경우와 같습니다. 세부사항은 41 페이지의 『SSL 프로토콜 설정값』의 내용을 참조하십시오.

HTTPS의 기본 포트는 443입니다.

클라이언트 설정값 구성

클라이언트 설정값 패널을 표시하려면 트리 구조에서 클라이언트 노드를 선택하십시오. 설정값은 CTG.INI 파일의 클라이언트 섹션에 있는 매개변수로 맵핑됩니다.

기본 서버

드롭 다운 목록에서 기본 서버를 선택하십시오.

Applid

8자까지 입력하거나 이 필드를 기본값 *로 두십시오.

이 값은 CICS 서버에서 시스템으로 자동 설치될 양식으로 CICS Universal Client 워크스테이션의 applid를 지정합니다. 이름은 CICS 서버 시스템에서 고유해야 합니다. * 값은 고유한 것으로 보증되는 이름을 자동으로 생성합니다.

주: 클라이언트가 여러 CICS 서버에 자동 설치될 경우 또한 applid의 특정 이름을 입력할 경우, 그 이름은 연결되는 모든 서버에 대해 고유해야 합니다. 이름이 고유하지 않으면 또 다른 클라이언트가 이미 같은 이름으로 설치되었을 수 있으므로 서버에 연결하려는 시도가 거부될 수 있습니다. * 이름을 사용할 경우, 클라이언트는 각 서버에서 서로 다른 고유 이름으로 알려질 수 있습니다.

최대 버퍼 크기

4 - 32 범위 안의 KB 수를 입력하십시오. 기본값은 32KB입니다.

이 값은 응용프로그램이나 단말기 데이터가 흐를 전송 버퍼 크기를 지정합니다. 값은 가능한 가장 큰 COMMAREA나 사용될 단말기 입출력 영역(TIOA)에 대해 제공할 수 있을만큼 충분히 커야 합니다. 값에는 일부 프로토콜에 대해 클라이언트에서 요구하는 512바이트의 오버헤드가 없습니다.

클라이언트가 메모리 제한 환경에서 실행 중인 경우에만 이 기본 설정을 바꾸십시오.

단말기 나감

1 - 4자의 문자열을 입력하십시오. 기본값은 EXIT입니다.

언제든 트랜잭션 이름을 입력할 수 있는 곳에서 터미널 에뮬레이터에 입력한 경우 이 문자열은 터미널 에뮬레이터를 종료시킵니다. 문자열에는 어떤 공백 문자도 넣어서 안됩니다.

문자열에서는 대소문자가 구분됩니다. 터미널 에뮬레이터가 CICS 단말기 정의에 대문자 변환을 수반할 경우, 이 문자열을 대문자로 입력해야 합니다.

최대 서버

1 - 256 범위 안의 값을 입력하십시오. 기본값은 10입니다.

이 값은 클라이언트에서 동시에 액세스할 수 있는 최대 서버 수를 지정합니다.

최대 요청 수

1 - 10 000 범위 안의 값을 입력하십시오. 기본값은 256입니다.

이 값은 클라이언트에서 동시에 실행할 수 있는 최대 동시 항목 수를 지정합니다. 여기서 항목은 다음 중 하나입니다.

- 터미널 에뮬레이터
- EPI 단말기
- ECI 작업 단위
- ESI 작업 단위

이 값은 오류 발생시 응용프로그램에서 서버에 과도한 요청을 제출할 수 있는 초과 상황을 발견하는 데 사용됩니다. 다른 운영 체제 한계(예: 메모리 제한 또는 통신 세션 수)가 적용될 경우, 실제 한계는 이 설정보다 작을 수도 있습니다.

인쇄 명령

1 - 256자의 문자열을 입력하십시오.

지정한 문자열은 클라이언트가 실행되는 운영 체제에 대해 고유한 명령입니다. 인쇄 요청이 클라이언트에 수신되면 클라이언트는 인쇄 요청마다 고유한 이름을 사용하여 임시 인쇄 파일을 생성합니다.

매개변수 문자열이 임시 파일 이름에 추가되고, 그 결과로 생기는 명령이 실행됩니다. 이 문자열을 사용하여 인쇄 요청을 파일에 복사하고, 국지 프린터로 지정하고, 문서에 포함하기 위해 형식화하는 등의 작업을 할 수 있습니다.

처리를 완료한 후 임시 인쇄 파일을 삭제하는 것은 인쇄 명령에서 이루어집니다.

인쇄 명령(예: **lpr**)과 그 뒤에 삭제 명령(**rm**)이 있는 셸 스크립트를 사용하십시오.

자세한 정보는 인쇄 파일 설명을 참조하십시오.

인쇄 파일

1 - 256자의 문자열을 입력하십시오.

이 선택항목은 인쇄 명령 설정이 생략될 경우에만 적용할 수 있습니다.

지정한 문자열은 클라이언트에 수신된 인쇄 요청의 출력을 보낼 파일을 식별합니다. 각 인쇄 요청은 현재 파일의 끝에 추가됩니다.

주: 이 설정은 기본값으로만 작동합니다. 단말기 및 인쇄 에뮬레이터는 이 값을 대체할 수 있는 선택항목을 제공합니다.

코드 페이지 ID 대체

국지 코드 페이지 ID를 대체할 CCSID(Coded Character Set Identifier)를 나타내는 값을 입력하십시오.

플랫폼이 Euro를 지원하도록 갱신되고 CICS 서버에서 Euro를 지원할 경우 이 설정을 사용하십시오. 예를 들어, Latin-1 국가의 경우에는 CCSID 값으로 858을 사용하십시오. 이 값은 코드 페이지 850에 Euro 지원이 포함됨을 나타냅니다. 코드 페이지 1252의 경우, CCSID 값 5348을 지정하십시오.

cicsterm은 항상 코드 페이지 ID 대체 설정에 지정된 값에 관계없이 워크스테이션의 국지 코드 페이지를 기초로 문자를 표시합니다.

또한 CCSID를 사용하여 사용되는 코드 페이지 ID를 변경할 경우, 이전에 서버에 저장된 데이터에 클라이언트를 검색할 때 코드 포인트에서 다른 문자를 생성하게 되는 문자가 있으면 그 데이터는 수정될 수 있습니다.

CCSID 및 자료 변환에 대한 자세한 내용은 131 페이지의 『부록A. CICS Universal Clients 자료 변환』을 참조하십시오.

서버 재시도 간격

초 수를 입력하십시오. 이 값은 클라이언트에서 이전에 연결되었던 서버에 다시 연결하기 위해 시도하는 횟수 사이의 간격(초)입니다. 기본 간격은 60초입니다.

클라이언트에서 연결되었던 서버가 더이상 사용 중이 아님을 인식하면 비활성화되었다가 1초 후에 서버에 다시 연결하려 합니다. 그 뒤로 재시도는 서버 재시도 간격에 정의된 간격으로 발생합니다.

로그 파일

문제점 진단에 사용될 로그 파일의 이름을 입력하십시오.

지정하지 않으면 로그 파일 이름의 기본값으로 /var/cicscli 서브디렉토리의 CICSCLI.LOG가 사용됩니다.

서버 설정값 구성

서버 설정값 패널을 표시하려면 트리 구조에서 서버 노드를 선택하십시오. 설정값은 CTG.INI 파일의 서버 섹션에 있는 매개변수로 맵핑됩니다.

서버 이름

1 - 8자 사이의 이름을 입력하십시오. 이 이름은 클라이언트의 국지 서버의 통신-프로토콜에 독립적인 이름입니다.

ECI, EPI, ESI 또는 터미널 에뮬레이터에서 서버에 대한 액세스를 요청할 때 이 이름을 통해 서버를 참조합니다.

설명

서버에 대한 설명을 1 - 60자로 입력하십시오. 설명 입력은 선택사항입니다.

설명은 CICS_EpiListSystems 및 CICS_EciListSystems 기능을 통해 클라이언트에서 실행 중인 응용프로그램에 리턴됩니다(*CICS® Family: Client/Server Programming* 서적 참조).

초기 트랜잭션

1 - 128자 사이의 트랜잭션 ID를 입력하십시오.

이 문자열은 대소문자를 구별하고 터미널 에뮬레이터에서 서버에 연결할 때 실행할 초기 트랜잭션(및 임의 매개변수)을 식별합니다. 아무것도 입력하지 않으면 초기 트랜잭션은 실행되지 않습니다. 처음 네 문자 또는 문자열에서 첫 번째 공백까지의 문자를 트랜잭션으로 취합니다. 나머지 데이터는 호출시 트랜잭션으로 전달됩니다.

모델 단말기 정의

1 - 16자의 문자열을 입력하십시오.

이 문자열은 대소문자를 구별하고 서버의 모델 단말기 정의의 이름을 식별하며 클라이언트에서 자동 설치할 단말기의 특성을 식별합니다. 서버에서 모델을 찾을 수 없거나 아무것도 입력하지 않은 경우 기본 단말기 정의를 사용합니다. 이 기본값은 서버에 따라 달라집니다.

모델 단말기 정의 설정의 해석은 서버에 따라 달라집니다. 예를 들어, AIX용 TXSeries 서버의 경우 이 값은 1 - 16자이며 모델로 사용되는 CICS 단말기 정의 항목의 DevType 항목입니다.

대문자 보안 사용

ECI 응용프로그램 또는 사용자 프롬프트에서 얻은 사용자 ID 또는 암호를 대문자로 변환하려면 이 선택란을 선택하십시오.

이 설정은 기본값으로 사용 불가능합니다.

네트워크 프로토콜

드롭 다운 목록에서 서버 연결에 사용할 프로토콜을 선택하십시오.

- TCP/IP
- TCP62

선택한 프로토콜에 따라 패널에 표시된 프로토콜 설정값이 변경됩니다.

TCP/IP 설정값

호스트명 또는 IP 주소: CICS 서버를 실행 중인 호스트의 문자 또는 숫자로 된 TCP/IP ID를 입력하십시오. 예를 들면, cicssrv2.company.com(HostName) 또는 9.20.4.1(IPAddress)가 있습니다.

호스트 이름은 이름 서버를 통해 또는 etc 하위 디렉토리의 호스트 파일에서 IP 주소로 맵핑됩니다. 그러나 IP 주소가 변하는 경우 호스트 이름을 사용하는 것이 더 낫습니다.

포트: 클라이언트에서 연결할 서버의 포트 번호를 정의하는 0 - 65 535 사이의 숫자 값을 입력하십시오. 기본값은 0입니다.

0 값은 TCP 프로토콜을 사용하여 CICS 서비스의 포트 번호를 찾으려면 /etc 디렉토리에 있는 services 파일을 사용해야 함을 의미합니다.

서비스 파일에 항목이 없는 경우, 값은 1435로 간주됩니다. 이 포트는 TCP/IP 구조에서 CICS Universal Clients에 할당된 포트입니다.

연결 시간 초과: 연결 설정에 소요되는 최대 시간을 초 단위로 지정하여, 0 - 3600 범위 안의 값을 입력하십시오. 기본값인 0은 클라이언트에서 어떤 한계도 설정되지 않음을 의미합니다.

연결 설정에 지정한 시간보다 많은 시간이 소요될 경우 시간 종료 발생입니다. TCP/IP 소켓이 닫히고 리턴 코드인 ECI_ERR_NO_CICS 또는 CICS_EPI_ERR_FAILED가 클라이언트 응용프로그램에 전달됩니다.

서버에 이 기능에 대한 지원이 설치되어 있는 경우 시간 종료 후에만 정리 처리가 수행됩니다.

TCP/IP Keepalive 패킷 송신: TCP/IP가 정기적으로 KeepAlive 패킷을 서버에 송신하여 연결을 확인하도록 하려면 이 선택란을 선택하십시오.

TCP62 설정값

호스트 파일: TCP62 CICS 서버를 구성할 경우 국지 호스트 파일도 갱신해야 합니다. 파일 경로는 다음과 같습니다.

Unix 시스템
/etc/hosts

파일은 IP 주소를 호스트 이름에 맵핑합니다. 각 입력항목을 별도의 행에 두십시오. 입력항목의 양식은 다음과 같습니다.

9.20.101.6 IYCQST34.GBIBMIYA.hursley.ibm.com

입력항목의 각 요소에 대한 설명은 다음과 같습니다.

9.20.101.6

CICS 시스템의 IP 주소. 1열에서 시작해야 하며, 나머지 입력항목과 적어도 한 공백을 두고 분리되어야 합니다.

IYCQST34.GBIBMIYA.

역순으로 표시한 상대 LU 이름. 위의 예제에서는 GBIBMIYA. IYCQST34가 구성 도구에 입력된 상대 LU 이름입니다.

hursley.ibm.com

Anynet 도메인 이름 접미부.

상대 LU 이름: 클라이언트의 SNA 구성에 알려진 서버의 LU 이름을 입력하십시오. 이 이름은 ABC3XYZ4.PQRS1234와 같이 규정된 17문자 이름이어야 합니다.

국지 LU 이름 또는 템플리트: CICS 서버에 연결할 때 사용할 국지 LU 이름의 템플리트 또는 실제 LU 이름을 입력하십시오.

템플리트를 입력할 경우, LU 이름 템플리트의 IP 주소 마스크(선택항목) 필드를 채워야 합니다.

템플리트에는 템플리트 대체 문자(즉, *)가 있으며, LU 이름을 동적으로 생성하는데 사용됩니다. 예를 들어, CTS8BC** 템플리트는 두 개의 별표를 대체하도록 이

름이 동적으로 생성되어야 함을 나타냅니다. 고유한 국지 LU 이름(예: CTS8BC38)을 작성하기 위해 템플리트, IP 주소 마스크 및 워크스테이션 IP 주소를 기초로 하는 알고리즘이 사용됩니다.

국지 LU 이름에 대해 동적으로 이름을 생성할 경우, 서로 다른 국지 LU 이름이 클라이언트 시스템의 IP 주소 기준으로 생성되므로 모든 CICS 클라이언트에서 같은 구성을 사용할 수 있습니다. 따라서 VTAM® 안에서 여러 LU를 정의할 필요가 없습니다.

수 백 가지의 클라이언트가 동적 LU 이름 생성을 사용하여 CICS 서버에 연결할 경우, 국지 LU 이름에 더 많은 템플리트 대체 문자를 사용해야 합니다(예: CTS8****). 사실, OS/390®용 CICS® Transaction Gateway는 LU 이름의 마지막 네 자를 연결 이름으로 사용하므로 이런 문자는 고유해야 합니다.

LU 이름 템플리트의 IP 주소 마스크(선택항목): IP 주소 마스크를 큰 끝 형식(big-endian) 즉, 최상위 바이트(MSB) 순으로 입력하십시오. 이 값은 16진수여야 합니다(예: FFFFFFF80).

이 주소 마스크는 동적 LU 이름 생성에 사용됩니다. 자세한 내용은 국지 LU 이름 또는 템플리트 설명을 참조하십시오.

국지 LU 이름이 템플리트가 아닐 경우 이 매개변수는 무시됩니다. 기본값은 00000000입니다.

모드 이름: 서버에 연결할 때 사용할 모드 이름을 지정하는 1 - 8자의 이름을 입력하십시오. ModeName을 지정하지 않으면 기본 모드 이름인 TCP62를 사용합니다.

최대 논리 SNA 세션 수, 최대 SNA RU 크기 그리고 SNA 페이징 크기 설정을 사용하여 모드 이름에 지정된 모드를 정의할 수 있습니다.

일반 TCP62 설정값

완전한 CP 이름 또는 템플리트: 시작할 노드의 완전한 CP 이름 또는 완전한 CP 이름의 템플리트를 입력하십시오.

템플리트를 입력할 경우, **CP 이름의 IP 주소 마스크(선택항목)** 필드를 채워야 합니다.

템플리트에는 템플리트 대체 문자(즉, *)가 있으며, CP 이름을 동적으로 생성하는데 사용됩니다. 예를 들어, USIBMJKA.CP62***** 템플리트는 네 개의 별표를 대체하도록 이름이 동적으로 생성되어야 함을 나타냅니다. 고유한 CP 이름(예: USIBMJKA.CP62AH38)을 작성하기 위해 템플리트, IP 주소 마스크 및 워크스테이션 IP 주소를 기초로 하는 알고리즘이 사용됩니다.

CP 이름에 대해 동적으로 이름을 생성할 경우, 서로 다른 CP 이름이 클라이언트 시스템의 IP 주소 기준으로 생성되므로 모든 CICS 클라이언트 시스템에서 같은 구성을 사용할 수 있습니다. 따라서 VTAM에 여러 CP를 정의할 필요가 없습니다.

수 백 가지의 클라이언트가 동적 CP 이름 생성을 사용할 경우, 템플리트에 더 많은 템플리트 대체 문자를 사용해야 합니다(예: USIBMJKA.CP*****).

템플리트의 네트 ID 부분(첫번째 부분)에는 임의 템플리트 대체 문자가 없어야 합니다.

CP 이름의 IP 주소 마스크(선택항목): IP 주소 마스크를 큰 끝 형식(big-endian) 즉, 최상위 바이트(MSB) 순으로 입력하십시오. 이 값은 16진수여야 합니다(예: FFFF8080).

이 주소 마스크는 동적 CP 이름 생성에 사용됩니다. 자세한 내용은 완전한 CP 이름 또는 템플리트 설명을 참조하십시오.

완전한 CP 이름 또는 템플리트가 템플리트가 아닐 경우, 이 매개변수는 무시됩니다. 기본값은 00000000입니다.

AnyNet 도메인 이름 접미부: AnyNet® 도메인 이름 접미부를 입력하십시오. 이 매개변수를 상대(상대 LU 이름)와 함께 사용하여 그 LU와 연관되는 호스트의 IP 주소를 판별합니다.

예를 들어, 도메인 이름 접미부가 sna.ibm.com이고 상대 LU 이름은 NETID.LUA 이면 LUA.NETID.sna.ibm.com 호스트 이름에 대해 gethostbyname 호출을 발행하여 LU 이름이 NETID.LUA인 노드의 IP 주소를 얻습니다.

기본값은 sna.ibm.com입니다.

TCP62 포트: 0 - 65535의 포트 번호를 입력하여 클라이언트에서 연결해야 할 서버의 포트 번호를 정의하십시오. 기본값은 0이며, TCP/IP 서비스 MPTN(서비스 파일에 정의되어 있는)의 포트를 사용해야 함을 의미합니다.

원격 노드 비활동 폴 간격: 0 - 65535 사이의 초 수를 입력하십시오. 기본값은 60초입니다. 0 값은 폴링을 해제합니다.

구성 설정은 비활동 상태의 연결을 폴링하기 전의 CICS Universal Client 대기 시간을 제어합니다.

고급 TCP62 설정값

최대 논리 SNA 세션 수: 1 - 255 범위 안의 값을 입력하십시오. 기본값은 8입니다.

이 매개변수는 논리 SNA 세션 수를 정의하기 위해 TCP62에 사용됩니다.

최대 SNA RU 크기: 256 - 4096 범위 안의 값을 입력하십시오. 기본값은 1024입니다.

이 매개변수는 모드 이름 설정에 지정된 모드를 정의하기 위해 TCP62에 사용됩니다.

SNA 페이싱 크기: 0 - 63 범위 안의 값을 입력하십시오. 기본값은 8입니다.

이 매개변수는 모드 이름 설정에 지정된 모드를 정의하기 위해 TCP62에 사용됩니다.

트레이스 설정값

트레이스 설정을 구성하려면 도구 메뉴에서 트레이스 설정 선택항목을 선택하십시오.

트레이스 설정

트레이스가 켜져 있을 때 트레이스할 CICS Transaction Gateway 및 CICS Universal Client 구성요소를 지정하려면 선택란을 선택하십시오.

모든 트레이스	모든 구성요소.
클라이언트 API 레벨 1	클라이언트 API 층(레벨 1).
클라이언트 API 레벨 2	클라이언트 API 층(레벨 1 및 2).
CICSCLI 명령 행	cicscli 명령 인터페이스.
CICSTERM 및 CICSPRINT	cicsterm 및 cicsprnt 에뮬레이터.
CPP 클래스	C++ 클래스 라이브러리.
클라이언트 디몬	CICS Universal Client 디몬.
전송 층	프로세스간 통신.
게이트웨이	CICS Transaction Gateway.
JNI 트레이스	JNI 트레이스를 사용 가능화 하려면 CTG_JNI_TRACE=FILENAME 환경 변수를 설정하십시오 여기서 FILENAME은 트레이스를 기록할 파일입니다.
프로토콜 드라이버 레벨 1	프로토콜 드라이버(예: TCP). 송수신된 데이터를 트레이스하며, 실패에 대한 보충 정보를 제공합니다.
프로토콜 드라이버 레벨 2	프로토콜 드라이버 및 다른 소프트웨어 구성요소와의 상호작용을 통해 내부 흐름을 트레이스합니다. 현재 CCLTCP62 프로토콜 드라이버에서만 이 레벨을 지원합니다.

주: 위의 설정으로 JNI 트레이스를 사용 가능화할 수 없습니다. JNI 트레이스를 사용 가능화하려면 정보를 저장할 파일을 지시하도록 CTG_JNI_TRACE 환경 변수를 설정해야 합니다.

cicscli 명령에서 `-m` 매개변수를 지정하여 트레이스 구성요소를 지정할 수도 있습니다(게이트웨이 구성요소 제외). cicscli를 사용하여 지정한 구성요소 추적은 구성 도구로 지정한 추적을 대체합니다. 구성요소 추적을 cicscli 명령이나 구성 도구로 지정하지 않은 경우, 기본 구성요소 세트가 트레이스됩니다(보통, 프로토콜 드라이버, 클라이언트 디몬 및 클라이언트 API 레벨 1). 선택란 중 하나를 선택할 경우, 해당 선택이 기본 구성요소 세트를 대체합니다.

API 구성요소의 경우, 트레이스할 정보의 레벨을 지정할 수 있습니다. 클라이언트 API 레벨 1 선택란은 기본 API 관련 정보가 트레이스되도록 지정합니다(예를 들어, ECI, EPI 및 ESI 호출 이전이나 이후). 클라이언트 API 레벨 2 선택란은 레벨 1의 항목 외에도 API 트레이스 항목이 더 생성되도록 지정합니다.

cicscli `-d=nnn` 명령은 트레이스할 데이터 영역의 최대 크기를 설정할 때 사용됩니다. `nnn`을 예상되는 데이터 크기보다 작게 설정하면 트레이스 데이터가 잘릴 수 있습니다.

게이트웨이 트레이스 파일

트레이스 메시지를 기록할 트레이스 파일의 경로를 입력하십시오. 그 파일의 추적이 사용 가능해집니다. 어떤 경로도 지정하지 않으면 트레이스는 CICS Transaction Gateway가 설치된 bin 서브디렉토리의 CTG.TRC 파일에 기록됩니다.

ctgstart 명령에 `-tfile` 선택항목을 사용하여 트레이스 파일을 지정할 수도 있습니다.

트레이스 파일은 CICS Transaction Gateway가 시작될 때마다 추가되는 것이 아니라, 덮어쓰기됩니다.

성능: 게이트웨이 트레이스를 설정하면 성능에 커다란 영향을 줍니다. 생성 환경에서는 게이트웨이 트레이스를 사용 가능화하지 않는 것이 좋습니다.

게이트웨이 트레이스 랩 크기

0 - 1 000 000KB 범위의 값을 입력하십시오. 기본값은 0입니다.

이 값은 게이트웨이 트레이스 파일이 확장되는 크기를 KB 단위로 지정합니다. 후속 트레이스 입력항목은 파일 시작 부분에서부터 계속 기록됩니다.

구성

클라이언트 트레이스 파일

트레이스가 사용 가능하면 트레이스 메시지를 기록할 트레이스 파일의 경로 이름을 입력하십시오.

파일 유형 **.BIN**이 항상 생성되므로(또는 트레이스 파일이 래핑될 경우에는 **.WRP**) 파일 이름에 대한 확장자를 입력할 필요가 없습니다.

어떤 경로도 지정하지 않으면 트레이스는 **CICSCLI.BIN** 파일에 기록됩니다. 이 파일은 **/var/cicscli** 서브디렉토리에 있습니다.

성능 저하를 최소화하려면 클라이언트 트레이스 파일이 2진 형식으로 기록됩니다. 이 파일을 읽으려면 **cicsftrc** 명령을 사용하여 ASCII로 변환해야 합니다.

트레이스에 대한 자세한 정보는 운영 체제용 *CICS Transaction Gateway Client Administration* 서적의 문제점 판별 절을 참조하십시오.

최대 클라이언트 랩 크기

0 - 999KB 범위 안의 값을 입력하십시오. 기본값은 0입니다.

이 값은 트레이스 파일이 래핑되기 전에 증가될 수 있는 크기를 지정합니다. 기본 값 0을 지정하면 트레이스 래핑을 사용할 수 없습니다.

구성 변환 도구

구성 변환 도구(ctgconv)를 사용하면 CICS Transaction Gateway, IBM CICS Universal Clients 및 Java용 CICS Gateway 이전 버전의 구성 파일을 CICS Transaction Gateway 버전 4.0 구성 파일의 새로운 형식으로 변환할 수 있습니다.

CICS Transaction Gateway, IBM CICS Universal Clients, Java용 CICS Gateway 3.1 이전 버전에서 작성한 구성 파일의 경우 변환 도구를 사용하십시오. 버전 3.1에서 업그레이드할 경우에는 사용할 필요가 없습니다.

변환 도구는 다음을 변환합니다.

Gateway.properties

CICS Transaction Gateway 버전 3.0 및 Java용 CICS Gateway의 특성 파일.

CICSCLI.INI

CICS Clients 버전 2 및 버전 3.0의 클라이언트 초기화 파일.

변환 도구는 기본적으로 CTG.INI라는 한 출력 파일을 생성합니다. 이 파일의 예제는 56 페이지의 『구성 파일 편집』에 나와 있습니다.

이전 파일은 .BAK 확장자로 이름이 변경되지만, 배너(banner)가 삽입되어 해당 파일이 사용되지 않음을 나타냅니다.

변환 도구 사용

ctgconv의 매개변수는 다음과 같습니다.

```
ctgconv -c=file [-g=file] [-o=file]
```

각 매개변수에 따라 *file*은 다음이 될 수 있습니다.

- 확장자가 있는 파일 이름, 여기서는 /opt/ctg/bin 디렉토리로 가정합니다. 확장자는 .INI여야 하지만, CICS Transaction Gateway 입력 파일의 경우에는 확장자가 .properties입니다.
- 전체 경로 이름
- 디렉토리 이름(끝에 /가 없음), 여기서는 기본 파일 이름

-c=file

변환할 클라이언트 초기화 파일을 지정합니다.

*file*에서 디렉토리를 지정하면 파일 이름을 CICSCLI.INI로 가정합니다.

매개변수를 지정하지 않으면 CICSCLI 환경 변수가 클라이언트 초기화 파일을 찾는 데 사용됩니다. CICSCLI를 설정하지 않은 경우 파일 /opt/ctg/bin/CICSCLI.INI로 가정합니다.

-g=file

변환할 Gateway.properties 파일을 지정합니다.

-g 매개변수를 지정하지 않으면 CTG.INI에서 게이트웨이 섹션은 작성되지 않습니다.

-o=file

변환한 파일의 경로 이름을 지정합니다. 기본값은 /opt/ctg/bin/CTG.INI입니다.

구성

ctgconv 사용에 대한 도움말을 보려면 `ctgconv -?`를 입력하십시오.

변환하는 동안, 마지막 3 글자인 파일 확장자가 `.BAK`가 될 수 있도록 입력 파일이 재명명됩니다. 해당 이름을 가진 파일은 덮어쓰기되고, 변환하기 전의 출력 파일은 같은 방식으로 덮어쓰기됩니다.

이전 구성 파일에서 나머지 매개변수들은 제거되고, 변환된 파일에서 다른 매개변수는 새로운 이름을 가집니다.

구성 파일 편집

구성 도구를 사용하는 것이 좋지만, 구성 파일을 편집하여 구성할 수 있습니다.

구성 파일은 CICS Universal Clients 및 CICS Transaction Gateway 모두에 사용되며, 다음 섹션을 포함합니다.

1. GATEWAY
2. CLIENT
3. SERVER
4. DRIVER

해당 섹션의 형식은 다음과 같습니다.

```
SECTION sectionname [=value]
    property1=value
    property2=value
    ...
    propertyN=value
ENDSECTION
```

CICS Universal Client 및 CICS Transaction Gateway를 다시 시작하여 구성 파일의 변경사항을 추출해야 합니다.

다음 절에서는 각 섹션의 특성에 대한 예제를 설명합니다. 특성에 대해서는 구성 도구에 있는 해당 설정값의 설명을 참조하십시오.

주: 주석을 지정하는 데 # 문자를 사용할 경우, 이 문자는 행의 시작 부분에 있거나 공백 또는 Tab 문자 다음에 와야 합니다. 모드 이름과 같은 일부 올바른 이름이 # 문자로 시작할 수 있기 때문입니다.

GATEWAY 섹션

```
SECTION GATEWAY
# -----
# Properties file for the CICS Transaction Gateway
# -----
      initconnect=1 # Initial number of ConnectionManager threads
      maxconnect=100 # Maximum number of ConnectionManager threads
      initworker=1 # Initial number of Worker threads
      maxworker=100 # Maximum number of Worker threads
      trace=on # Enable extra tracing messages
      notime=on # Disable timing information in messages
      noinput=on # Disable the reading of input from the console
      nonames=on # Do not display TCP/IP hostnames
# -----
# Entries to define the protocols which this CICS Transaction Gateway daemon will
# accept.
#
# Entries for a particular protocol begin with protocol@<protocol-name>.
# followed by which property is being set. The following properties MUST be set
# for each protocol handler to be used :
#
# (1) protocol@<protocol-name>.handler = ....
#     The name of the class which provides the handler for this protocol
#
# (2) protocol@<protocol-name>.parameters = ....
#     The parameters passed to the protocol handler. The possible parameters
#     vary upon the protocol handler.
#
      protocol@tcp.handler=com.ibm.ctg.server.TCPHandler
      Protocol@tcp.parameters=port=2006;sotimeout=1000;
      connecttimeout=2000;idletimeout=6000000;pingfrequency=60000
      protocol@http.handler=com.ibm.ctg.server.HttpHandler
      Protocol@http.parameters=port=8080;sotimeout=1000;connecttimeout=2000;
      idletimeout=120000;requiresecurity
      protocol@ssl.handler=com.ibm.ctg.server.SslHandler
      Protocol@ssl.parameters=port=8050;sotimeout=1000;connecttimeout=2000;
      Idletimeout=600000;pingfrequency=60000;keyring=ServerKeyRing;keyringpw=default;
      clientauth=on;
      Protocol@https.handler=com.ibm.ctg.server.HttpsHandler
      Protocol@https.parameters=port=443;sotimeout=1000;
      connecttimeout=2000;Idletimeout=120000;keyring=ServerKeyRing;keyringpw=default;
      clientauth=off;
# -----
# Advanced settings.
      ecigenericreplies=off
      workertimeout=10000
      closetimeout=10000
ENDSECTION
```

CLIENT 섹션

```
SECTION CLIENT = *
# -----
# Client section - This section defines the local CICS client. There
# should only be one Client section.
#
      MaxServers = 1 # Only allow one server connection
      MaxRequests = 256 # Limit the maximum server interaction
      MaxBufferSize = 32 # Allow for a 32K maximum COMMAREA
#
      CCSID=850
ENDSECTION
```

SERVER 섹션

```
SECTION SERVER = cicstcp1
#-----
# Server section - This section defines a server to which the client may
# connect. There may be several Server sections.
# Description = A TCP/IP Server # Arbitrary description for the server
# Protocol = TCPIP # Matches with a Driver section below
# NetName = cicstcp.ibm.com # The server's TCP/IP name or address
# Port = 0 # Use the default TCP/IP CICS port
ENDSECTION
```

DRIVER 섹션

```
SECTION DRIVER = TCPIP ; Matches the Server's Protocol value
#-----
# Driver section - This section defines a communications protocol DLL
# used to communicate with a server. There may be
# several Driver sections.
# DriverName = CCLIBMIP #Use the IBM TCP/IP communications library
ENDSECTION
```

주: DRIVER 섹션은 구성 도구의 어느 설정값에도 해당되지 않습니다. 구성 도구는 정확한 프로토콜 드라이버를 자동으로 선택합니다.

구성 이름 바꾸기 및 파일 맵핑

CICS Universal Clients의 /opt/ctg/bin 디렉토리에는 기본적으로 다음 파일이 있습니다.

CTGSAMP.INI	샘플 클라이언트 구성 파일. (기본 구성 파일 이름은 CTG.INI입니다.)
CICSKEY.INI	키보드 맵핑 파일. (샘플 파일인 CICSKEYSAMP.INI도 제공됩니다.)
CICSCOL.INI	색상 맵핑 파일. (샘플 파일인 CICSCOLSAMP.INI도 제공됩니다.)

이런 파일의 사용자 정의 버전은 다른 이름으로 작성하는 것이 좋습니다. 서비스 갱신을 설치하면 파일이 덮어쓰기되어 사용자 정의 사항이 유실되기 때문입니다.

다음과 같은 환경 변수를 통해 사용자 정의 파일을 지정하십시오.

파일	환경 변수
클라이언트 구성 파일	CICSCLI

키보드 맵핑 파일	CICSKEY
색상 맵핑 파일	CICSCOL

클라이언트 키보드 맵핑 구성

터미널 에뮬레이터 조작을 위한 키보드 맵핑은 키보드 맵핑 파일에 정의되어 있습니다. 기본적으로 /opt/ctg/bin에 CICSKEYSAMP.INI 파일이 제공됩니다. 그러나 자체 사용자 정의 맵핑 파일을 작성하는 것이 좋습니다.

키보드 맵핑 파일은 다음을 사용하여 지정할 수 있습니다.

- `cicsterm` 명령의 `-k` 선택항목: 특정 단말기로 키보드 맵핑 파일 지정(페이지 『클라이언트 키보드 맵핑 구성』 참조).
- `CICSKEY` 환경 변수. 예를 들어, 다음과 같습니다.

```
export CICSKEY=/var/cicscli/MYKEYS.INI
```

아이콘을 통해 이 환경 변수를 설정할 수 있습니다.

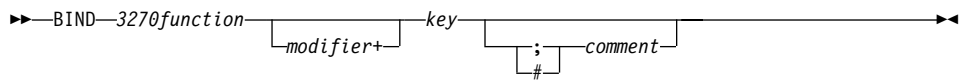
아무것도 지정하지 않으면 /opt/ctg/bin 디렉토리에 있는 CICSKEY.INI 파일 이름이 사용됩니다.

다음에 터미널 에뮬레이터를 시작할 때까지는 변경사항이 적용되지 않지만 언제든지 키보드 맵핑 파일을 변경할 수 있습니다.

키보드 맵핑 파일 구문

이 절에서는 키보드 맵핑 파일 구문에 대해 설명합니다. 기본 지정이 없으므로(영문자 및 숫자 키 제외) 필요한 각 키에 대한 명령문을 제공해야 합니다. 대소문자는 구별되지 않으므로, 키워드와 값을 대문자, 소문자 또는 대소문자 혼용하여 입력할 수 있습니다. 각 바인딩은 별도의 행에 있어야 하며 그 양식은 다음과 같습니다.

키보드 맵핑 파일 구문



클라이언트 키보드 맵핑 구성

예를 들어, 3270 기능 EraseEof를 Ctrl+Delete 키 동시 누르기에 맵핑하려는 경우 바인딩은 다음과 같습니다.

```
bind EraseEof Ctrl+Delete ;erase to end of field
```

키보드 맵핑 파일

맵핑 파일에서 3270function은 다음 중 하나입니다.

backspace	pa1	pf1	pf13
backtab	pa2	pf2	pf14
clear	pa3	pf3	pf15
cursor down		pf4	pf16
cursor left	printscreen	pf5	pf17
cursor right	reset	pf6	pf18
cursor select	tab	pf7	pf19
cursor up		pf8	pf20
delete	ignore	pf9	pf21
enter		pf10	pf22
erase eof		pf11	pf23
erase input		pf12	pf24
home			
insert			
newline			

키보드에서 원하지 않는 제어 키를 무시하기 위해 ignore 값이 제공됩니다. 예기치 않은 glyphs는 생성되지 않습니다.

수정자는 다음 중 하나입니다.

```
Ctrl  
Shift
```

키는 표3의 키 중 하나이지만 몇 가지 수정자+키 조합은 지원되지 않습니다.

표3. 맵핑할 수 있는 CICS Universal Client 키

그룹	키
Escape 키	Escape
기능 키	f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12
숫자 키	0 1 2 3 4 5 6 7 8 9

표 3. 맵핑할 수 있는 CICS Universal Client 키 (계속)

그룹	키
알파벳 키	a b c d e f g h i j k l m n o p q r s t u v w x y z
Tab 키	Tab
이동 키	newline backspace insert home pageup delete end pagedown up left down right
키패드 키	keypad/ keypad* keypad- keypad7 keypad8 keypad9 keypad4 keypad5 keypad6 keypad+ keypad1 keypad2 keypad3 keypad0 keypad. keypadenter

주: Linux용 CICS Universal Client의 경우, Ctrl/Act, Print Screen, Scroll Lock 및 Pause 키는 사용할 수 없습니다. 기본적으로 Esc 키에는 3270 기능 지우기가 지정되어 있습니다.

키 조합

다음과 같은 수정자와 키 조합을 맵핑할 수 있습니다.

수정자 없음	모든 키를 맵핑에 사용할 수 있음
Ctrl 수정자	기능 키, 이동 키, 알파벳 키, Tab 키, 키패드만을 맵핑할 수 있음
Shift 수정자	기능 키, 숫자 키, Tab 키, 알파벳 키만을 맵핑할 수 있음

62 페이지의 그림5에 샘플 키 맵핑 파일이 나와 있습니다.

클라이언트 키보드 맵핑 구성

```
*****  
;* IBM CICS Client - 3270 Emulator Keyboard Binding File *  
*****  
; Format:  
;   bind 3270_key [modifier+]workstation_key  
;  
; Refer to the "IBM CICS Clients Administration" book for full details  
; of available values.  
; Note: There are no default key bindings, ensure all required 3270  
;       keys are mapped. "Enter" and "Clear" are particularly  
;
```

그림 5. CICS Universal Clients의 샘플 키보드 맵핑 파일 (1/3)

```
; Miscellaneous keys  
bind Enter      Newline  
bind Enter      KeypadEnter  
bind Enter      Ctrl+KeypadEnter  
bind Clear      Escape  
bind Reset      Ctrl+R  
bind Newline    Ctrl+N  
bind Tab        Tab  
bind Backtab    Shift+Tab  
bind Backspace  Backspace  
bind Delete     Delete  
bind Delete     Keypad.  
bind Insert     Insert  
bind Insert     Keypad0  
bind EraseEof   Ctrl+E  
bind EraseInput Ctrl+A  
bind Ignore     End  
bind PrintScreen Ctrl+P  
; Cursor movement  
bind CursorUp   Up  
bind CursorUp   Keypad8  
bind CursorDown Down  
bind CursorDown Keypad2  
bind CursorLeft Left  
bind CursorLeft Keypad4  
bind CursorRight Right  
bind CursorRight Keypad6  
bind CursorSelect Ctrl+F3  
bind Home       Home  
bind Home       Keypad7
```

그림 5. CICS Universal Clients의 샘플 키보드 맵핑 파일 (2/3)


```

; PF and PA keys
bind Pf1      F1
bind Pf2      F2
bind Pf3      F3
bind Pf4      F4
bind Pf5      F5
bind Pf6      F6
bind Pf7      F7
bind Pf7      Keypad9
bind Pf7      PageUp
bind Pf8      F8
bind Pf8      Keypad3
bind Pf8      PageDown
bind Pf9      F9
bind Pf10     F10
bind Pf11     F11
bind Pf12     F12
bind Pf13     Shift+F1
bind Pf14     Shift+F2
bind Pf15     Shift+F3
bind Pf16     Shift+F4
bind Pf17     Shift+F5
bind Pf18     Shift+F6
bind Pf19     Shift+F7
bind Pf20     Shift+F8
bind Pf21     Shift+F9
bind Pf22     Shift+F10
bind Pf23     Shift+F11
bind Pf24     Shift+F12
bind Pa1      Ctrl+Insert
bind Pa2      Ctrl+Home
bind Pa3      Ctrl+PageUp

```

그림 5. CICS Universal Clients의 샘플 키보드 맵핑 파일 (3/3)

화면 색상 사용자 정의

화면 색상과 속성에 대해서는 색상 맵핑 파일에 정의되어 있습니다. 사용자가 조정할 수 있는 샘플이 제공되어 있습니다. 제공되는 기본 파일은 /opt/ctg/bin 디렉토리의 CICSCOL.INI입니다. 자체 사용자 정의 맵핑 파일을 작성하는 것이 좋습니다.

색상 맵핑 파일은 다음에 따라 식별할 수 있습니다.

화면 색상 사용자 정의

- 특정 서버로 색상 맵핑 파일을 지정하는 `cicsterm` 명령의 `-c` 선택항목(페이지 63 페이지의 『화면 색상 사용자 정의』 참조).
- `CICSCOL` 환경 변수: 예를 들면, 다음과 같습니다.

```
export CICSCOL=/var/cicscli/MYCOLS.INI
```

아무것도 지정하지 않으면 현재 디렉토리에 있는 `CICSCOL.INI` 파일 이름이 사용됩니다.

색상 맵핑 파일은 3270 화면 속성(예: 감빡임 또는 밀줄)의 정확한 복제가 불가능한 하드웨어 환경에서 대체 표현을 제공하는 데 사용됩니다. 따라서 색상 맵핑 파일은 클라이언트 하드웨어에서 3270 화면 속성이 에뮬레이트되는 방법을 정의합니다.

색상 맵핑 파일은 선택사항입니다. 그러나 대부분의 하드웨어 환경에서 에뮬레이터가 감빡임 또는 밀줄 지원을 요구할 경우 맵핑 파일이 필요합니다.

주:

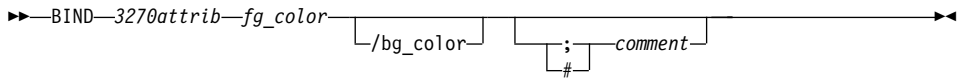
1. 색상 맵핑 파일에 속성에 대한 맵핑이 지정되어 있는 경우, 클라이언트가 실행되고 있는 하드웨어가 실제로 화면 속성을 지원하더라도 이 맵핑이 사용됩니다.
2. 응용프로그램이 3270 필드에 가령, 밀줄이 표시되도록 요청했으나 아무 에뮬레이션 설정이 지정되어 있지 않아서 하드웨어가 밀줄을 표시할 수 없는 경우, 필드는 아무것도 강조하지 않은 상태로 표시됩니다.

다음에 터미널 에뮬레이터를 시작할 때까지는 변경사항이 적용되지 않지만 언제든지 색상 맵핑 파일을 변경할 수 있습니다.

색상 맵핑 구문

색상 맵핑 파일 구문은 다음과 같습니다. 대소문자는 구별되지 않으므로, 키워드와 값의 대문자, 소문자 또는 대소문자 혼용하여 입력할 수 있습니다. 각 바인딩은 별도의 행에 있어야 하며 그 양식은 다음과 같습니다.

색상 맵핑 파일 구문



색상 맵핑 파일

맵핑 파일에서 *3270attrib*는 다음 중 하나입니다.

```

normal_protected      intensified_protected
normal_unprotected   intensified_unprotected
default              blinking_default      underscored_default
blue                 blinking_blue        underscored_blue
green                blinking_green      underscored_green
cyan                 blinking_cyan        underscored_cyan
red                  blinking_red         underscored_red
magenta              blinking_magenta     underscored_magenta
white                blinking_white       underscored_white
yellow               blinking_yellow      underscored_yellow
default_highlight
operator_information_area
    
```

각 *fg_color* 및 *bg_color*(전경 색상 및 배경 색상)는 다음 중 하나입니다.

```

black      light_gray
blue       light_blue
brown      yellow
cyan       light_cyan
green      light_green
magenta    light_magenta
red        light_red
gray       white
    
```

*bg_color*가 생략될 경우, 기본값은 검정색입니다.

66 페이지의 그림6에 샘플 색상 맵핑 파일이 나와 있습니다.

화면 색상 사용자 정의

```
*****
;* IBM CICS Client - 3270 Emulator Color Binding File *
*****
; Format:
; bind 3270_field foreground_color/background_color
;
; Refer to the "IBM CICS Clients Administration" book for full details
; of available values.
; Operation information area at the bottom of the screen.
bind operator_information_area black/green
; Color used as default for a terminal defined as monochrome, or to display
; characters displayed in the default color when character or field
; attributes are used
bind default light_green/black
; Color used for an intense field when either the terminal is defined as
; monochrome, or the screen has been formatted with extended attributes but
; the data to be displayed has default color and highlight.
bind default_highlight white/black
; Colors used for datastream formatted with field attributes when no
; extended character or field attributes are in use
bind normal_unprotected light_green/black
bind intensified_unprotected light_red/black
bind normal_protected light_cyan/black
bind intensified_protected white/black
; Colors used for datastream formatted with extended field attributes
; or datastream formatted with character attributes.
bind red light_red/black
bind green light_green/black
bind blue light_blue/black
bind magenta light_magenta/black
bind cyan light_cyan/black
bind yellow yellow/black
bind white white/black
; The following bindings provide a mapping for 3270 blink and under-
; score attributes, as most workstation displays do not provide blink
; or underscore capabilities. For those that do these bindings could
; be deleted.
bind blinking_default light_green/gray
bind blinking_red light_red/gray
bind blinking_green light_green/gray
bind blinking_blue light_blue/gray
bind blinking_magenta light_magenta/gray
bind blinking_cyan light_cyan/gray
bind blinking_yellow yellow/gray
bind blinking_white white/gray
```

그림 6. 샘플 색상 맵핑 파일 (1/2)

```

bind underscored_default    light_green/light_gray
bind underscored_red        light_red/light_gray
bind underscored_green      light_green/light_gray
bind underscored_blue       light_blue/light_gray
bind underscored_magenta     light_magenta/light_gray
bind underscored_cyan        light_cyan/light_gray
bind underscored_yellow      yellow/light_gray
bind underscored_white       white/light_gray
;* End of file

```

*

그림 6. 샘플 색상 맵핑 파일 (2/2)

국지 CICS Transaction Gateway 지원 사용 준비

국지 CICS Transaction Gateway 지원을 사용하는 응용프로그램을 실행하려면 환경 변수를 제대로 구성해야 합니다. 다음과 같은 사항이 충족되어야 합니다.

1. 정확히 구성된 Java 환경

국지 CICS Transaction Gateway 지원을 사용하는 Java 응용프로그램은 일반적인 Java 프로그램으로 간주될 수 있으므로, 표준 JDK Java 실행 파일과 함께 실행됩니다. 따라서 환경을 Java 2진 파일, 라이브러리 및 클래스를 찾을 수 있도록 제대로 설정해야 합니다. 올바른 환경 설정값에 대해서는 JDK 문서를 참조해야 합니다.

2. CICS Transaction Gateway 2진 파일을 찾을 수 있도록 하는 정확한 설정값

CICS Transaction Gateway .class 파일을 사용할 수 있어야 합니다. CICS Transaction Gateway의 ctgserver.jar 및 ctgclient.jar 아카이브를 포함하도록 CLASSPATH 환경 변수를 설정해야 합니다. 또한 관련 2진 파일도 사용할 수 있어야 합니다.

응용프로그램에서 libCTGJNI.so를 찾을 수 있도록 CICS Transaction Gateway의 bin 서브디렉토리를 포함시켜 LD_LIBRARY_PATH 환경 변수를 설정해야 합니다.

응용프로그램을 실행하려면 JDK java 명령을 적절한 선택항목을 적용하여 사용하십시오.

제5장 연결 보안

이 장에서는 네트워크 프로토콜인 SSL 및 HTTPS를 사용하여 CICS Transaction Gateway를 설정하는 방법에 대해 설명합니다.

- 『개요』에서는 네트워크 연결 보안 개념 및 용어를 설명합니다. 암호화 키, 디지털 인증 및 키링 개념도 소개합니다.
- 74 페이지의 『SSL 및 인증』에서는 SSL 프로토콜 및 인증 유형에 대해서도 설명합니다.
- 76 페이지의 『ctgikey 도구』에서는 CICS Transaction Gateway가 디지털 인증을 관리하기 위해 제공하는 도구인 iKeyMan에 대해 소개합니다.
- 78 페이지의 『외부 사인 인증 사용(SSLight)』에서는 외부 사인 디지털 인증을 얻어 이를 CICS Transaction Gateway가 사용할 수 있도록 키링 파일에 저장하는 방법에 대해 설명합니다.
- 84 페이지의 『자가 사인 인증 사용(SSLight)』에서는 CICS Transaction Gateway가 사용할 자가 사인 디지털 인증을 생성하는 방법에 대해 설명합니다.
- 88 페이지의 『SSL 및 HTTPS용 CICS Transaction Gateway 구성』에서는 보안 프로토콜을 사용하도록 CICS Transaction Gateway를 구성하는 방법에 대해 설명합니다.

CICS Transaction Gateway에서 제공하는 SSL 구현은 순전히 Java로 작성되었으며, **SSLight**라 합니다. 또 다른 구현인 시스템 **SSL**은 OS/390의 CICS Transaction Gateway에만 적용되며, 해당 플랫폼에 있는 SSL 서버에 대해서만 사용할 수 있습니다.

개요

CICS Transaction Gateway는 안전한 통신을 위한 광범위한 지원을 제공하며, 이는 성공적인 인터넷 조작에 매우 중요합니다. 보안 네트워크 프로토콜을 사용하면 클라이언트 애플릿 및 응용프로그램에서 SSL을 사용하여 CICS Transaction Gateway와 안전하게 통신할 수 있습니다. SSL 및 HTTPS 프로토콜에 대해서는

보안 개념에 대한 개요

1 페이지의 『제1장 개요』에서 소개했으며, 이 장에서는 이러한 프로토콜을 사용하도록 CICS Transaction Gateway를 설정하는 좀 더 자세한 내용을 살펴봅니다.

다음은 보안 통신의 특성입니다.

- 기밀성

기밀성은 메시지가 인터넷 또는 인트라넷을 통해 전달될 때 그 내용이 개인적인 것으로 남아 있음을 의미합니다. 기밀성은 메시지 암호화를 통해 보장됩니다.

- 완전성

완전성은 메시지가 전송되는 동안 변경되지 않음을 의미합니다. 임의의 라우터가 테스트를 삽입하거나 삭제할 수 있고, 전달할 때 메시지를 왜곡시킬 수도 있습니다. 완전성 없이는, 전송한 메시지가 수신한 메시지와 일치하는지 보장할 수 없습니다. 완전성은 암호화와 디지털 사인으로 보장됩니다.

- 책임

책임은 전송자와 수신자 모두가 교환이 발생했음에 동의하는 것을 의미합니다. 책임 없이 수신자는 메시지가 도달되지 않았음을 말할 수 있습니다. 책임은 디지털 신호로 보증되므로, 잘못된 점을 발견하면 책임이 있는 파트를 식별할 수 있습니다.

- 인증

인증은 대화하고 있는 사용자를 알고 있고 그 사람을 신뢰할 수 있음을 의미합니다. 인증에는 ID 식별이 필요하므로, 다른 사람의 존재를 확신할 수 있습니다. 디지털 사인이나 디지털 인증을 통해 인증할 수 있습니다.

암호화란?

암호화는 인터넷을 통한 전송에서 기밀성을 보장합니다. 가장 단순한 양식으로 되어 있는 암호화는 나중에 수신자가 메시지를 해독할 때까지 읽을 수 없도록 메시지를 뒤섞는 것입니다. 전송자는 알고리즘 패딩이나 키를 사용하여 메시지를 암호화하고, 수신자는 해독 키를 사용하여 메시지를 해독합니다.

디지털 사인 및 인증에 대해서 뿐만 아니라 암호화에 사용할 수 있는 두가지 종류의 키가 있습니다.

1. 대칭
2. 비대칭

대칭키를 사용하여, 전송자와 수신자는 몇몇 종류의 패턴을 공유하며, 이 패턴은 전송자가 메시지를 암호화하는 데 사용되고, 수신자가 메시지를 해독하는 데 사용됩니다. 대칭 키에 포함된 위험성은 보안 키를 통신하려는 사람과 공유할 때 사용할 안전한 전송 방법을 발견해야 하는 것입니다.

비대칭 키를 사용하여 키 쌍을 작성합니다. 이 키 쌍은 공용 키와 개인용 키로 구성됩니다. 대칭 키와는 달리, 비대칭 키는 서로 다르며 개인용 키는 공용 키보다 더 비밀스런 암호화 패턴을 유지합니다.

전송자는 공용 키를 안전하게 통신하려는 사람에게 전파할 수 있습니다. 전송자는 개인용 키를 가지고 있으며 암호로 이를 보호합니다. 전송자만이 공용 키로 암호화된 수신 메시지를 해독할 수 있습니다. 전송자만이 개인용 키를 갖고 있기 때문입니다.

SSL과 같은 프로토콜은 비대칭(또는 공용 키) 암호작성법과 대칭키 암호작성법 모두를 사용합니다. 공용 키 암호작성법은 TCP/IP 핸드셰이크에 사용됩니다. 핸드셰이크 중, 마스터 키는 클라이언트에서 서버로 전달됩니다. 클라이언트와 서버는 마스터 키를 사용하여 자신의 세션키를 작성합니다. 그런 다음 세션 키는 나머지 세션에 대한 데이터를 암호화하고 해독하는 데 사용됩니다.

디지털 사인 및 디지털 인증

디지털 사인은 책임을 보증하는 수학적으로 계산된 고유한 사인입니다.

디지털 인증을 사용하면 엔티티를 고유하게 식별할 수 있으며, 이는 신뢰성 있는 기관에서 발행한 전자 ID 카드에 필수입니다.

디지털 인증은 두가지 목적으로 제공됩니다. 즉, 자신의 ID를 설정하고, 자신의 공용 키를 사용할 수 있도록 합니다. 디지털 인증은 신뢰된 기관인 인증 기관(CA)(예를 들면, VeriSign Inc, Thawte)에서 발행됩니다. 기간이 제한되어 발행되며, 만기일이 지나면 대체해야 합니다.

디지털 인증은 다음으로 구성됩니다.

- 인증되는 사람의 공용 키.
- 인증되는 사람의 이름과 주소, 고유 이름(DN)이라고도 함.

보안 개념에 대한 개요

- CA의 디지털 사인.
- 발행 날짜.
- 만기일.

DN(Distinguished Name)은 개인 또는 조직의 이름과 주소입니다. 인증 요청의 일부로 고유 이름을 입력합니다. 디지털 사인 인증에는 자신의 고유 이름은 포함되지 않지만, CA 검증을 허용하는 CA의 식별 이름은 포함됩니다.

안전하게 통신하려면 전송시 송신자가 사용 중인 인증서를 발행한 CA를 수신자가 신뢰해야 합니다. 그 결과 송신자가 메시지에 사인할 때마다 수신자는 신뢰 루트 키로 지정된 해당 CA의 사인자 인증서 및 공용 키를 갖고 있어야 합니다. 예제와 같이, 웹 브라우저는 신뢰성 있는 CA에 대한 사인자 인증서의 기본 목록을 갖게 됩니다. 다른 CA로부터의 인증서를 신뢰하려면 해당 CA의 인증서를 받아 신뢰 루트 키로 지정해야 합니다.

공용 키가 있는 디지털 인증서를 다른 사람에게 전송한 경우 수신자가 사용자의 디지털 인증서를 잘못 사용하여 사용자인 것처럼 행세하지 못하도록 하는 것은 무엇입니까? 답은 개인용 키입니다. 디지털 인증만으로는 다른 사람의 신원을 확인할 근거가 되지 못합니다. 디지털 인증만을 사용하면 소유자의 디지털 사인을 검사하는 데 필요한 공용 키를 제공하여 소유자의 신원을 검증할 수 있습니다. 따라서 디지털 인증 소유자는 소유하고 있는 개인용 키를 디지털 인증서에 있는 공용 키로 보호해야 합니다. 그렇지 않고 개인용 키가 도난당하면 다른 사람이 디지털 인증서의 법적 소유자로 행세할 수 있습니다.

디지털 인증서 확보

두가지 방식으로 인증서를 얻을 수 있습니다.

1. CA로부터 인증서 구입
2. 스스로 인증서 발행 즉, 자신의 CA 역할을 함

CA로부터 인증서 구입

인터넷에서 상업적인 사업을 하려는 경우, VeriSign Inc 같은 CA에서 서버 인증서를 구입해야 합니다. 홈 페이지는 <https://www.verisign.com/>입니다.

VeriSign으로 인증서 요청을 제출하면 인증서를 발행하기 전에 사용자의 신분을 검증합니다. 사용자, 사용자의 조직 및 VeriSign을 보호하기 위해 승인 프로세스가 필수이지만, 다소 시간이 오래 걸릴 수 있습니다. VeriSign은 인증서 요청에 디지털식으로 사인하고, 전자 우편을 통해 고유한 인증서를 리턴합니다.

외부 사인 인증서 확보에 대해서는 78 페이지의 『외부 사인 인증 사용(SSLight)』에서 설명합니다.

주: VeriSign 서버 인증서는 다른 시스템에 있는 서버와 공유할 수 없습니다.

인증 자가 발행

직접 CA 역할을 할 경우, 자신의 또는 다른 사람의 인증 요청에 사인할 수 있습니다. 조직 안에서의 인증만 필요한 경우엔 좋지만 외부 인터넷 사업용으로는 적합하지 않습니다. 이 시나리오에서는, 인트라넷 안에서 주의깊게 제어한 핵심 인물의 그룹에만 액세스를 허용할 수 있습니다.

핵심 인물은 Netscape Navigator와 같은 브라우저를 가지고 있으며, 자가 사인 CA 인증서를 받아 신뢰 루트로 지정합니다. 그런 다음에는 통신을 신뢰하고 안전하게 정보를 공유할 수 있습니다.

자가 사인 인증서 생성에 대해서는 84 페이지의 『자가 사인 인증 사용(SSLight)』에서 설명합니다.

키링

디지털 인증서 및 연관된 키를 보관하는 위치는 어디입니까? 대답은 공용 키, 개인용 키, 인증서 및 신뢰 루트 키는 키링 파일에 보관한다는 것입니다.

CICS Transaction Gateway는 Java 클래스를 사용하여 SSL 서버와 SSL 클라이언트에서 인증서와 키 데이터를 보관합니다. CICS Transaction Gateway 디몬(SSL 서버 역할을 하는)에서 서버 키링(예를 들면, **ServerKeyRing.class**)을 사용하는 반면, 모든 SSL 클라이언트는 클라이언트 키링(예를 들면, **ClientKeyRing.class**)을 사용합니다. SSL 및 HTTPS 프로토콜의 경우, 보안 연결을 설정하는 데 이런 Java 클래스에 대한 액세스를 요구합니다. CICS Transaction Gateway 구성시 이 액세스를 설정합니다. 88 페이지의 『SSL 및 HTTPS용 CICS Transaction Gateway 구성』을 참조하십시오.

보안 개념에 대한 개요

이 장에 있는 뒤의 절에서는 다음 방법에 대해 설명합니다.

- 키링 파일 작성
- 디지털 인증서 확보
- 디지털 인증서를 키링 파일로 수신

SSL 및 인증

SSL을 사용하면 클라이언트가 서버의 ID를 인증할 수 있으며, 이를 *서버 인증*이라고 합니다.

SSL 버전 3을 사용하면 서버가 클라이언트를 인증할 수 있으며, 이를 *클라이언트 인증*이라고 합니다. 클라이언트 인증은 서버가 응답하기 전에 클라이언트가 누구인지 확인해야 할 경우에 사용됩니다. SSL 클라이언트 인증이 설정되면 서버는 클라이언트가 SSL 연결을 작성할 때마다 클라이언트의 인증서를 요청합니다. 서버는 문서를 제공하기 전에 클라이언트의 인증서에 있는 DN 정보로 클라이언트 요청에 있는 DN 정보를 검증합니다.

SSL은 보안 핸드셰이크를 사용하여 클라이언트와 서버 간의 TCP/IP 연결을 시작합니다. 핸드셰이크 중, 클라이언트와 서버는 세션에 사용할 보안 키와 암호화에 사용할 알고리즘에 동의합니다. 클라이언트는 서버를 인증합니다. 뿐만 아니라 클라이언트가 SSL 클라이언트 인증으로 보호되는 문서를 요청하면 서버는 클라이언트의 인증을 요청합니다. 핸드셰이크 후, SSL이 사용되어 다음을 포함하여 클라이언트 요청 및 서버 응답에 있는 모든 정보를 암호화하고 해독합니다.

- 클라이언트가 요청하는 URL
- 제출되는 양식 내용
- 사용자 이름 및 암호와 같은 액세스 권한 정보
- 클라이언트와 서버 사이에서 전송되는 모든 데이터

SSL 핸드셰이크는 그림7에 나와 있습니다.

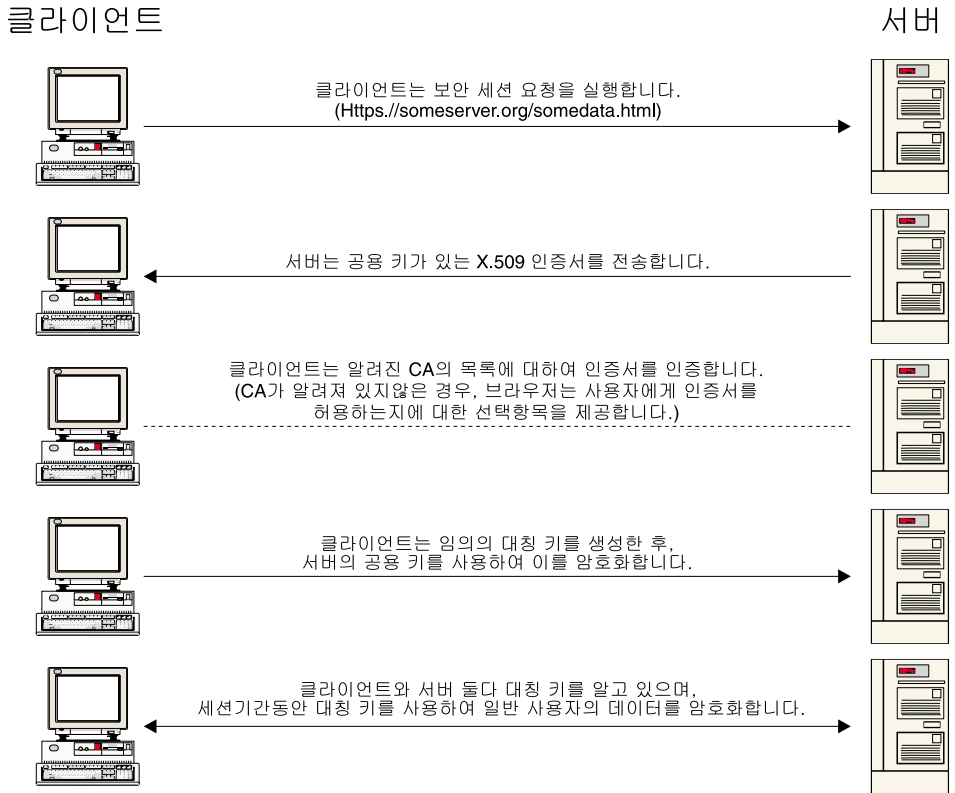


그림 7. SSL은 서버 인증과 데이터를 교환합니다.

HTTPS

HTTPS는 SSL과 HTTP를 결합한 고유한 프로토콜입니다. SSL 보호 문서로 링크하는 HTML 문서에서는 앵커로 `https://`를 지정해야 합니다. 클라이언트 사용자는 `https://`를 지정하여 URL을 열고 SSL 보호 문서를 요청할 수 있습니다.

HTTPS(HTTP + SSL)와 HTTP는 서로 다른 프로토콜이며 보통 서로 다른 포트(각각 443과 8080)를 사용하므로, SSL과 비 SSL 요청을 동시에 수행할 수 있습니다. 그 결과, 보안을 사용하지 않는 모든 사용자에게는 정보를 제공하고, 보안 요청을 수행한 브라우저에게만 특별한 정보를 제공하도록 선택할 수 있습니다. 이 방법은 인터넷의 소매점에서 사용자가 보안 없이 광고를 보고 주문 양식을 채운 다음 보안을 사용하여 신용카드 번호를 전송할 때 사용됩니다.

SSL을 통한 HTTP를 지원하지 않는 브라우저는 보통 HTTPS를 사용하여 URL을 요청할 수 없습니다. 비 SSL 브라우저는 안전하게 제출해야 하는 양식 제출을 허용하지 않습니다.

ctgikey 도구

CICS Transaction Gateway는 디지털 인증을 유지하는 **iKeyman** 도구를 제공합니다. **ctgikey** 명령을 사용하여 정확한 환경을 설정하고(JAVA_HOME 환경 변수 포함) iKeyman을 호출합니다.

iKeyMan을 사용하면 다음을 수행할 수 있습니다.

- CA로부터 디지털 인증서를 요청 및 수신, 78 페이지의 『외부 사인 인증 사용(SSLight)』 참조
- 자가 사인 인증서 생성, 84 페이지의 『자가 사인 인증 사용(SSLight)』 참조
- 키링 파일로 인증 추가
- 키링 암호 변경
- 기본값으로 개인용 키 설정
- 키 삭제
- 키를 파일로 복사하여 키 내보내기
- 내보낸 사본에서 키를 가져와 키링에 추가

클라이언트 워크스테이션으로 iKeyman 분산

KeyRing 관리 작업중 대부분이 CICS Transaction Gateway 시스템에서 수행되더라도, SSL 서버에 연결된 클라이언트에 iKeyman 도구를 제공해야 할 수 있습니다. 클라이언트 시스템에는 최소 레벨의 CICS Transaction Gateway Java 지원인 IBM Java SDK 1.2.2가 설치 있어야 합니다. 또한 iKeyman Java 시작 클래스를 호출하는 스크립트나 명령 파일도 필요합니다.

예제와 같이, 다음은 JDK로 iKeyman을 호출하는 Windows NT 명령 행입니다.

```
java.exe -classpath
"e:\ikeyman\cfwk.zip;e:\ikeyman\gsk4cls.jar;e:\ikeyman
\swingall.jar;" -Dkeyman.javaOnly=true com.ibm.gsk.ikeyman.Ikeyman
```

또는 JRE로 호출하는 명령행입니다.

```
jre.exe -classpath
"e:\ikeyman\cfwk.zip;e:\ikeyman\gsk4cls.jar;e:\ikeyman
\swingall.jar;" -Dkeyman.javaOnly=true com.ibm.gsk.ikeyman.Ikeyman
```

이런 예제에서는 클라이언트 워크스테이션의 ikeyman 디렉토리에 다음 파일이 있는 것으로 가정합니다.

- cfwk.zip
- cfwk.sec
- gsk4cls.jar
- swingall.jar
- ikmininit.properties

이런 파일은 모두 CICS Transaction Gateway가 설치된 classes 서브디렉토리에 있습니다.

주: cfwk.zip이 CLASSPATH 설정값에 나타나는 첫번째 IBM 제공 아카이브가 되어야 합니다.

외부 사인 인증 사용(SSLight)

CICS Transaction Gateway는 SSL 서버 역할을 할 수 있으며, 이 서버에는 SSL 클라이언트를 인증하고 VeriSign Inc와 같은 인증 기관의 외부 사인 인증서를 허용할 수 있는 기능이 있습니다.

이 절에서는 인증 관리 인터페이스 iKeyman을 사용하여 SSL 서버와 SSL 클라이언트를 구성하는 방법에 대해 설명합니다. 이 인터페이스는 순전히 Java로만 작성되므로, 적합한 JVM이 설치되어 있는 클라이언트/서버 워크스테이션으로 분배할 수 있습니다.

SSL 서버 및 클라이언트 구성 작업에는 키링 클래스 작성, 디지털 인증 확보 및 키링 클래스로 인증서 수신 작업이 포함됩니다.

서버 키링 클래스에는 서버 인증서, 해당 개인용 키 및 사인자 인증서가 있습니다. 서버 인증서는 연결 중인 클라이언트에 SSL 서버를 식별하는 데 사용되는 디지털 인증서입니다.

클라이언트 키링 클래스에는 클라이언트 인증이 필요한 경우의 최소 클라이언트 x.509 인증서와 함께 SSL 서버의 사인자 인증서가 있습니다.

SSL 서버 구성

이 절에서는 VeriSign 웹 사이트(www.verisign.com)로부터 테스트 서버 인증서를 확보하는 방법에 대해 설명합니다. VeriSign을 사용하면 14일 동안 시험 서버 인증서를 사용할 수 있습니다. 이 인증서는 데모용 서버 인증서이므로, 신뢰성 있는 VeriSign 인증 기관이 아닌 VeriSign 테스트 CA에서 사인한 것입니다.

모든 SSL 클라이언트는 자신의 클라이언트 키링에 또는 HTTPS 연결인 경우엔 브라우저에 있는 저장소에 VeriSign 테스트 사인자 인증서 루트키를 설치해야 합니다.

이 책의 예제에서는 사용자가 Netscape Communicator Version 4.5를 사용하고 있다고 가정합니다.

완전한 VeriSign 서버 인증서 확보 절차도 테스트 버전의 경우와 같습니다.

서버 키링 작성

첫번째 단계는 서버 키링 클래스를 작성하는 것입니다. 여기에는 결국 서버 인증서(및 연관 개인용 키)와 함께 사인자 인증서가 포함됩니다. 이 저장소는 암호로 보호되며, .class가 iKeyman으로 작성될 때 암호 “강도”가 표시됩니다. 연속된 영숫자를 사용하는 것이 좋습니다. 그러면 암호가 좀 더 강력해져 “어찌해볼 수 없는 사전” 정도의 공격이 됩니다.

인증서를 확보하려면 다음과 같이 수행하십시오.

1. ctgikey를 시작하십시오.
2. 키 데이터베이스 파일 --> 새 파일을 선택하십시오.
3. 키 데이터베이스 유형에서, **SSLight** 키 데이터베이스 클래스를 선택하십시오.
4. 파일 이름에 ServerKeyRing.class를 입력하십시오.
5. 위치에서 ServerKeyRing.class를 저장하는 데 적합한 위치를 입력하십시오.
6. 확인을 선택하십시오.

기본적으로, 생성된 ServerKeyRing.class에는 일정한 수의 일반 사인자 인증서가 있으며, 여기에는 VeriSign 루트 사인자 인증서와 테스트 사인자 인증서가 포함됩니다. 또한 VeriSign Class 1 - 4 Public Certification Authority 사인자 인증서도 있으며, 이를 통해 서버는 VeriSign 클라이언트 인증서로 클라이언트를 인증할 수 있습니다. 이 내용은 81 페이지의 『SSL 클라이언트 구성』에서 더 자세히 설명합니다.

인증 요청 준비

www.verisign.com에서 서버 인증을 얻으려면 SSL 서버가 인증 요청을 작성하여 국지로 저장해야 합니다.

1. iKeyman에서 작성을 선택하십시오.
2. 새 인증서 요청 작성을 선택하십시오.
3. 인증 요청을 작성해야 합니다. 몇몇 필드는 선택적이지만, 최소한 다음 사항은 필수입니다(아래 예제).

Key Label	verisignServerCert
Key Size	1024

Common Name	servermachine.hursley.ibm.com
Organization	IBM UK
Country	GB

4. 인증 요청을 저장할 파일 이름을 지정한 후 확인을 선택하십시오. 그러면 iKeyMan은 공용/개인용 키 쌍을 생성합니다. 이때 처리기 속도에 따라 다소 시간이 걸릴 수 있습니다.

서버 인증 확보

다음 단계는 VeriSign 웹 사이트에 연결하여 테스트 서버 인증을 요청하는 것입니다. 브라우저에서 <http://www.verisign.com> 페이지를 열고 <https://digitalid.verisign.com/server/trial/trialIntro.htm> 페이지로 이동하십시오. 이제 테스트 서버 인증서를 등록할 준비가 되었습니다.

1. 계속을 선택하십시오.
2. 계속을 다시 선택하면 **Step 2 of 5 - Submit CSR**이 나타납니다.
3. 지정된 텍스트 상자에 인증 요청 파일 내용을 붙여넣으십시오. 인증 요청의 내용은 다음과 같아야 합니다.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIB1jCCAT8CAQAwgZUxIDAEgNVBAMTF2Nocm1zdHAuaHVyc2x1eS5pYm0u
Y29tMRswGQYDVQQLEExJDbG11bnRzICsgR2F0ZXdheXMxZDZANBgNVBAoTBk1C
TSBVSzETMBEGA1UEBxMKV01uY2h1c3R1cjEOMAwGA1UECBMSGfudHMxETAP
BgNVBBETCFNPMjEgMkpOMQswCQYDVQQGEWJHJjCBnzANBGRkqhkIG9w0BAQEF
AA0BjQAwgYkCgYEAAkAth9Ar6k6ijNZ3JxdPGH6yikiwYTUA0RZDLZBSpaSEx
4qKN/CrdF1LgfQYbZcN5NGCeC4sC478Nht1tf5dnR3pNWBzEzmWn5mN01H
tqJ3oib0UmdUi+tQc2J9z6iRBKjkCQWjP1Jp0sp5KKsev1ahAETL7LqmqMIq
pJ1zki0CAwEAaAAAMA0GCSqGSIb3DQEBAUA4GBAaHMhpizPs8Q3bi3I6dh
4yw0UNhojt1S1+ffiph3hK981MHJuMztr0UUBL1/SZGNw850JRiWuDjGYUQW
inJ0uNH34IUsnygBmt78+WlXT5nJuayg+UrAc5A02H8QzPRE5Sfaoc81QcvY
p1TggCdMxpYN7I33LrZD13Po0TT8gjsxQ
-----END NEW CERTIFICATE REQUEST-----
```

4. 계속을 선택하십시오.
5. 다음 페이지에서는 인증 요청의 내용을 확인할 수 있습니다. 또한 요청된 개인의 세부사항을 제공해야 합니다. 테스트 버전이 아닌 VeriSign 서버 ID 전체가 요청되면 이런 세부사항은 응용프로그램을 인증하는 데 사용됩니다.
6. 세부사항을 입력하고 VeriSign 계약서를 읽고 난 뒤에는 승인을 선택하십시오.

7. 다음 단계(4 / 5)는 SSL 서버 연결에 사용할 브라우저에 테스트 사인자 인증서 루트를 설치하는 것입니다. Java 애플릿에서 CICS Transaction Gateway 로의 SSL 연결에는 이 단계가 필요하지 않습니다. 클라이언트 애플릿에서 클라이언트 키링 클래스를 요구하는데 기본적으로 이 클래스에는 테스트 사인자 인증서가 포함되어 있기 때문입니다. 그러나 HTTPS 프로토콜은 브라우저의 저장소를 사용하므로, 테스트 사인자 인증서를 설치해야 합니다. 전체 VeriSign 서버 인증에 적용할 경우엔 테스트 사인자 인증서를 설치하지 않아도 됩니다.

VeriSign 테스트 서버 인증은 개인 세부사항에서 지정한 주소로 e-메일을 통해 전달됩니다. 보통 1 - 3 시간 정도 걸릴 수 있습니다.

서버 키링으로 서버 인증 수신

서버 인증을 얻을 때 iKeyman을 사용하여 서버 키링으로 “수신”해야 합니다.

1. 먼저 테스트 편집기를 사용하여 서버 인증 데이터를 빈 텍스트 파일로 복사하고 붙여넣기한 다음, 파일을 verisignServerID.arm에 저장하십시오.
2. iKeyman에서는, 풀다운 메뉴에서 개인 인증서를 선택하십시오. 이 선택항목은 키 데이터베이스 내용 레이블 아래에 있습니다.
3. 수신...을 선택하십시오.
4. 서버 인증 데이터가 있는 텍스트 파일을 찾으십시오. 이 텍스트 파일은 Base64로 인코딩된 ASCII입니다.
5. 확인을 선택하십시오.
6. 키 데이터베이스 파일과 나감을 선택하십시오.

이제 **ServerKeyRing.class**를 CICS Transaction Gateway와 함께 사용할 준비가 되었습니다. 여기에는 VeriSign 서버 인증 및 해당 개인용 키와 함께 기본 사인자 인증서가 있습니다.

SSL 클라이언트 구성

일반(기본)적인 작동시, CICS Transaction Gateway는 SSL 핸드셰이크를 수행할 때 서버 인증만을 사용하며 클라이언트는 제시된 서버 인증을 승인하기만 하면 됩니다. 서버 인증에 대해 작업해야 하는 경우 서버 키링 클래스, HTTPS 경우에는 브라우저 인증 저장소에 서버 인증의 사인자 인증서가 들어갑니다. 예제에서, 사인자 인증서는 VeriSign 테스트 사인자 인증서입니다.

외부 사인 인증 사용

서버 키링 클래스와 마찬가지로, iKeyman을 사용하여 클라이언트 키링 클래스를 사용할 때 여기에는 대부분의 일반적인 사인자 인증서의 기본 선택사항이 포함됩니다.

서버 인증뿐만 아니라, CICS Transaction Gateway에서도 클라이언트 인증을 지원합니다. 이 선택항목을 사용하면 ssl: 또는 https: 핸들러로 연결을 시도할 때에는 클라이언트가 자체 클라이언트 인증(또는 디지털 ID)을 제공해야 합니다.

다음 절에서는 VeriSign 디지털 ID를 확보하여 필수 클라이언트 키링 클래스를 생성하는 방법에 대해 설명합니다.

클라이언트 인증 확보

서버 인증을 확보할 때와는 반대로, iKeyman을 사용하여 인증 요청 양식을 생성할 필요가 없습니다. VeriSign은 Netscape 또는 Internet Explorer 브라우저를 사용하여 클래스 1 디지털 ID를 확보하는 메소드를 제공합니다.

일단 클라이언트 인증을 확보하여 브라우저 저장소에 설치하고 나면, 연관된 개인용 키와 함께 인증 데이터를 #PKCS12 파일이라는 보안 저장소로 내보낼 수 있습니다. 이 파일은 암호로 보호되며 iKeyman 도구를 사용하여 클라이언트 키링 클래스로 가져올 수 있습니다.

1. Netscape Communicator 버전 4.5를 사용하여 www.verisign.com 페이지를 열고 **Individual Certificates**의 링크를 따르십시오.
2. 일단 <https://digitalid.verisign.com/client/class1Netscape.htm>의 HTTPS 페이지에 도달하면 요청된 대로 세부사항을 작성하십시오. 세부사항은 VeriSign에서 클라이언트 응용프로그램을 인증하는 데 사용됩니다.
3. 승인을 선택하십시오.
4. Netscape는 개인용 키를 생성하지 않고, 이 개인용 키를 보호하기 위해 암호를 요청합니다.
5. VeriSign은 응용프로그램에서 사용자가 제출한 주소로 전자 우편을 보내 클래스 1 디지털 ID를 다운로드하여 설치하는 정보를 제공합니다.

설치 프로세스에서, Netscape Communicator를 사용하여 암호로 보호되는 파일(#PKCS12 형식)에 클라이언트 인증을 저장할 수 있습니다. iKeyman 도구는

#PKCS12 형식 파일을 지원하며, 이 도구를 사용하여 클라이언트 키링 클래스로 인증(개인용 키와 함께)을 가져올 수 있습니다.

클라이언트 키링 작성

클라이언트 키링 클래스를 작성하려면 다음과 같이 수행하십시오.

1. ctgikey를 시작하십시오.
2. 키 데이터베이스 파일 --> 새 파일을 선택하십시오.
3. 키 데이터베이스 유형에서, **SSLight** 키 데이터베이스 클래스를 선택하십시오.
4. 파일 이름에 ClientKeyRing.class를 입력하십시오.
5. 위치에서, ClientKeyRing.class를 저장할 적합한 위치를 입력하십시오.
6. 확인을 선택하십시오.

기본 사인자 인증서가 있는 ClientKeyRing.class가 다음과 같이 작성됩니다.

```
VeriSign Class 1 Public Primary Certification Authority
VeriSign Class 2 Public Primary Certification Authority
VeriSign Class 3 Public Primary Certification Authority
RSA Secure Server Certification Authority
Thawte Personal Basic CA
VeriSign Test CA Root Certificate
Thawte Personal Premium CA
Thawte Premium Server CA
Thawte Personal Freemail CA
Thawte Server CA
```

클라이언트 키링으로 클라이언트 인증 가져오기

클라이언트 인증이 있는 #PKCS12 저장소 파일을 가져오려면 다음과 같이 수행하십시오.

1. 키 데이터베이스 내용 레이블 아래에 있는 폴다운 선택자에서 개인 인증서를 선택하십시오.
2. 가져오기...를 선택하십시오.
3. 키 파일 유형을 PKCS12 파일로 설정하십시오.
4. 저장된 #PKCS12 파일을 찾으십시오.
5. 확인을 선택하십시오.
6. 키 데이터베이스 파일과 달기를 선택하십시오.

외부 사인 인증 사용

7. 키 데이터베이스 파일과 나감을 선택하십시오.

이제 **ClientKeyRing.class**를 CICS Transaction Gateway와 함께 사용할 준비가 되었습니다. 여기에는 VeriSign 클라이언트 인증(클래스 1 디지털 ID) 및 해당 개인용 키와 함께 기본 사인자 인증서가 있습니다.

자가 사인 인증 사용(SSLight)

CICS Transaction Gateway는 인증을 “자가 사인”하는 메커니즘을 제공합니다. 자신을 자신의 인증 기관으로 설정하고, 서버(CICS Transaction Gateway) 및 클라이언트(브라우저)측에 대한 X.509 디지털 인증을 생성합니다.

이 절에서는 인증 관리 인터페이스 iKeyman을 사용하여 SSL 서버와 SSL 클라이언트를 구성하는 방법에 대해 설명합니다.

SSL 서버 구성

SSL 서버 구성에는 서버 키링 클래스 작성, 자가 사인 인증 생성 및 이 인증을 키링 클래스로 수신하는 작업이 포함됩니다.

서버 키링 작성

첫번째 단계는 서버 인증 및 키 정보를 보관할 서버 키링 클래스 파일을 작성하는 것입니다.

1. ctgikey를 시작하십시오.
2. 키 데이터베이스 파일 --> 새 파일을 선택하십시오.
3. 키 데이터베이스 유형에서, **SSLight** 키 데이터베이스 클래스를 선택하십시오.
4. ServerKeyRing.class를 파일 이름으로 입력하십시오.
5. 위치에서 ServerKeyRing.class를 저장하는 데 적합한 위치를 입력하십시오.
6. 확인을 선택하십시오.
7. 키링 파일의 암호를 입력하십시오.

서버 인증 생성

자가 사인 서버 인증을 작성하여 개인용 키와 함께 서버 키링 클래스에 저장할 준비가 되었습니다.

1. iKeyman의 풀다운 메뉴에서 개인 인증서를 선택하십시오. 이 선택항목은 키 데이터베이스 내용 레이블 아래에 있습니다.
2. 새 자가 사인...을 선택하십시오.
3. 인증 세부사항을 작성해야 합니다. 몇몇 필드는 선택적이지만, 최소한 다음 사항은 필수입니다(아래 예제).

Key Label	exampleServerCert
Version	X509 V3
Key Size	1024
Common Name	clientmachine.hursley.ibm.com
Organization	IBM UK
Country	GB
Validity Period	365(days)

4. 확인을 선택하십시오. 그러면 iKeyMan은 공용/개인용 키 쌍을 생성합니다. 이 때 처리기 속도에 따라 다소 시간이 걸릴 수 있습니다.
5. iKeyman이 자가 사인 서버 인증을 작성하면 이 인증이 개인 인증 창에 나타납니다. 인증은 이 예제 exampleServerCert에서 생성 프로세스 중 지정된 **Key Label**에 따라 명명됩니다.
6. **exampleServerCert**를 강조표시한 상태에서, 보기/편집을 선택하십시오. **issued to** 및 **issued by** 텍스트상자에 있는 인증 정보는 같으므로, 인증 요청자(발행 대상)와 사인자(발행인)가 같습니다. 이 인증을 표시하는 서버와 SSL 연결을 설정하려면 클라이언트가 exampleServerCert의 사인자를 신뢰해야 합니다. 이를 위해서는 클라이언트 키 저장소에 exampleServerCert를 표시하는 사이트의 사인자 인증서가 있어야 합니다.

서버 사인자 인증서 반출

이제 SSL 서버의 사인자(공용) 인증을 내보냅니다.

1. **exampleServerCert**를 강조표시한 상태에서, 인증서 추출...을 선택하십시오.
2. 내보낸 인증의 데이터 유형을 선택하십시오. 보통 Base64 인코딩된 ASCII 데이터입니다.

자가 사인 인증 사용

3. 내보낸 인증의 이름/위치를 입력하십시오. 예제에서는 `exampleServercert.arm`을 사용합니다.
4. 확인을 선택하십시오.

내보낸 인증은 안전한 장소에 저장해야 하며, 특정 SSL 서버 인증과 핸드셰이크를 해야 하는 클라이언트 키 저장소로 가져와야 합니다.

SSL 클라이언트 구성

CICS Transaction Gateway에서 사용하는 SSL 핸들러를 서버 인증만 지원하도록 구성하면 자가 사인 클라이언트 인증을 생성하지 않아도 됩니다. 이런 경우, 클라이언트 키링 클래스는 서버의 사인자 인증서만 포함해야 하며, 이 인증서는 예제 `exampleServercert.arm`에서 내보낸 인증 파일입니다.

다음 단계에서는 클라이언트 키링 작성 및 서버의 사인자 인증서 가져오기 프로세스에 대해 설명합니다.

클라이언트 키링 작성

1. `ctgikey`를 시작하십시오.
2. 키 데이터베이스 파일 --> 새 파일을 선택하십시오.
3. 키 데이터베이스 유형에서, **SSLight** 키 데이터베이스 클래스를 선택하십시오.
4. `ClientKeyRing.class`를 파일 이름으로 입력하십시오.
5. 위치에서, `ClientKeyRing.class`를 저장할 적합한 위치를 입력하십시오.
6. 확인을 선택하십시오.
7. 키링의 암호를 입력하십시오.
8. 일단 `ClientKeyRing.class`가 생성되면 기존 사인자 목록을 참조해야 합니다.

서버의 사인자 인증서 가져오기

다음 단계는 서버의 사인자 인증서를 가져오는 것입니다.

1. 추가를 선택하십시오.
2. 저장된 서버 Base64 인코드 ASCII 인증 파일(예제에서는 `exampleServercert.arm`)을 찾으십시오.

3. 이 사인자 인증서에 고유한 레이블을 지정하십시오. 예를 들면, My Self-Signed Server Authority입니다.
4. 확인을 선택하십시오. 이 새로운 사인자 인증서가 기본 사인자 목록에 추가됩니다.

생성된 ClientKeyRing.class는 CICS Transaction Gateway의 SSL 프로토콜과 함께 사용할 수 있으며, 이 프로토콜은 서버 인증을 지원하도록 구성됩니다. HTTPS 프로토콜이 애플릿에서의 보안 연결을 설정하는 데 사용되면 애플릿이 수행되고 있는 특정 브라우저는 exampleServercert.arm을 키/인증 저장소로 가져와야 합니다.

Base64 인코드 ASCII 인증 파일 반입에 대해서는 브라우저 온라인 도움말을 참조하십시오.

클라이언트 인증 생성

클라이언트 인증에서는 클라이언트 키링 클래스가 연결된 클라이언트를 식별하는 데 사용되는 자가 사인 인증을 포함해야 합니다.

자가 사인 서버 인증을 생성하는 것과 같은 단계를 수행하여 클라이언트 키링 클래스(예제에서는 ClientKeyRing.class)를 작성(또는 열기)하십시오.

1. iKeyman의 풀다운 메뉴에서 개인 인증서를 선택하십시오. 이 선택항목은 키 데이터베이스 내용 레이블 아래에 있습니다.
2. 새 자가 사인...을 선택하십시오.
3. 인증 세부사항을 작성하십시오.
4. 확인을 선택하십시오. 그러면 iKeyMan은 공용/개인용 키 쌍을 생성합니다. 이 때 처리기 속도에 따라 다소 시간이 걸릴 수 있습니다.
5. SSL 서버와 마찬가지로, 클라이언트는 이 사인자 인증서를 SSL 서버의 키/인증 저장소에 설치해야 합니다. 이렇게 하면 SSL 서버가 클라이언트의 세부사항을 검증할 수 있습니다. exampleClientCert를 강조표시한 상태에서, 인증서 추출...을 선택하십시오.
6. 내보낸 인증의 데이터 유형을 선택하십시오. 보통 Base64 인코드된 ASCII 데이터입니다.
7. 내보낸 인증의 이름/위치를 입력하십시오. 예제에서는 exampleClientcert.arm을 사용합니다.

8. 확인을 선택하십시오.

내보낸 인증은 안전한 장소에 저장해야 하며, 특정 SSL 클라이언트 인증과 핸드셰이크해야 하는 클라이언트 키 저장소로 가져와야 합니다.

서버 키링에 대한 액세스 제한

서버 키링의 내용은 암호화되어 있지만, 다음을 수행하는 것이 좋습니다.

- 정확한 파일 권한을 가지고 있는지 확인
- 적용가능한 경우 CICS Transaction Gateway 시스템에 대한 액세스 제한

실제로 여러 서버 사이에서 인증을 공유하는 것은 좋지 않습니다. 특히 다른 시스템에서 실행되고 있는 경우에는 여러 서버에서 개인용 키를 공유하지 않도록 합니다. 개인용 키를 서로 통신해선 안됩니다.

SSL 및 HTTPS용 CICS Transaction Gateway 구성

CICS Transaction Gateway는 클라이언트와 통신하기 위한 다양한 프로토콜(TCP, HTTP, SSL 및 HTTPS 포함)을 지원합니다. SSL 및 HTTPS 프로토콜을 사용하려면 구성 도구를 사용할 수 있어야 합니다(32 페이지의 『구성 도구 사용』 참조). SSL 및 HTTPS 프로토콜은 CICS Transaction Gateway가 실행될 때 시작됩니다. 연결 보안 프로토콜 중 하나만 사용하거나 모두를 사용하도록 지정할 수 있습니다. 이러한 핸들러가 지정되면 CICS Transaction Gateway가 시작될 때 포트 8050에서 SSL 요청을 청취하고, HTTPS의 경우는 포트 443에서 요청을 청취합니다.

SSL 및 HTTPS 프로토콜 고유의 설정값은 다음과 같습니다.

키링 클래스명 이 설정값은 서버 KeyRing 클래스 파일의 이름을 지정합니다. CLASSPATH 환경 변수는 이 클래스를 찾을 수 있도록 설정해야 합니다.

키링 암호 이 설정값은 암호화된 서버 KeyRing을 해독하는 데 사용되는 암호를 지정합니다.

클라이언트 인증 사용

이 설정값은 클라이언트 인증이 사용되도록 지정합니다. 기본값은 서버 인증이 사용되는 것입니다.

구성 도구를 사용하면 구성 파일에서 필수 항목이 작성됩니다. SSL 및 HTTPS 프로토콜 항목에 대한 전체 예제는 예제 구성 파일 CTGSAMP.INI에 있습니다.

CICS Transaction Gateway에서는 SSL 및 HTTPS 연결을 설정하는 데 사용할 수 있는 두 개의 기본 키링 클래스 파일을 제공합니다. **ClientKeyRing** 및 **ServerKeyRing**는 암호 **default**를 사용하여 암호화되며, 테스트 환경에서 사용하는 경우에만 권장합니다. 따라서 SSL 및 HTTPS 프로토콜을 사용하려면 앞 절에서 설명한 것처럼 자신의 키링을 생성하는 것이 좋습니다.

클라이언트 키링 지정

사용되는 보안 프로토콜은 클라이언트 키링이 필요한지 여부를 결정합니다. HTTPS 프로토콜은 Java 애플릿 안에서 보안 통신용으로 설계되었으며, 여기서 브라우저(클라이언트) 자체에는 CICS Transaction Gateway(서버)와의 보안 연결을 설정하는 데 필요한 기능이 있습니다. 이런 이유로, HTTPS 프로토콜은 서버측 키링 지정만 요구하며 클라이언트측은 브라우저 소프트웨어로 처리됩니다.

SSL 프로토콜은 안전한 방식으로 서버와 클라이언트를 처리하는 CICS Transaction Gateway의 코드가 있는 하위 레벨에서 설계되었습니다. SSL 프로토콜은 서버 및 클라이언트에 대한 키링 클래스 파일을 요구합니다.

클라이언트 키링은 SslJavaGateway.class에서 static 필드를 설정하여 지정됩니다. 이 클래스는 CICS Transaction Gateway 클라이언트측 코드 부분을 구성합니다.

SslJavaGateway.class는 클라이언트 키링에 대해 “getting” 및 “setting” 메소드를 제공합니다.

```
public static void setKeyRing(String strSetKeyRing, String strSetKeyRingPW)
public static String getKeyRing()
```

SSL 프로토콜에서 클라이언트 키링 클래스를 사용하도록 설정하려면 클라이언트 응용프로그램이나 애플릿이 다음 호출에 대해 static 호출을 작성합니다.

```
SslJavaGateway.setKeyRing(CLASSname, PASSword);
```

자카 사인 인증 사용

여기서,

- CLASSname은 클라이언트에 대해 생성된 Java 키링 클래스의 클래스 이름을 의미합니다.
- PASSword은 포함된 X.509 인증을 해독하는 데 사용됩니다.

SslJavaGateway.class는 “getter” 메소드인 **getKeyRing()**을 제공하여 현재 지정된 클라이언트 키링의 CLASSname을 리턴합니다.

SSL/HTTPS 프로토콜을 사용하여 클라이언트 응용프로그램이나 애플릿에서 CICS Transaction Gateway로 연결하는 것은 TCP나 HTTP 프로토콜을 사용하는 것과 다르지 않습니다. 클라이언트 응용프로그램이나 애플릿은 관련 URL을 사용하여 그 요청을 단순히 CICS Transaction Gateway로 “흘러보냅니다”. 예를 들어, SSL의 경우 응용프로그램은 `ssl://transGatewayMachine:8050`을 사용하고, HTTPS의 경우 `https://transGatewayMachine:443`을 사용합니다.

클라이언트측 프로그램의 설계 및 구현에 대해서는 *CICS Transaction Gateway: Gateway Programming* 책과 CICS Transaction Gateway programming interface HTML 페이지를 참조하십시오.

제6장 CICS Transaction Gateway 조작

이 장에서는 다음에 대해 설명합니다.

- 『게이트웨이 시작』
- 94 페이지의 『게이트웨이 중지』

게이트웨이 시작

게이트웨이를 설치한 컴퓨터의 운영 체제 명령 프롬프트에서 게이트웨이를 시작합니다.

먼저, 작업 디렉토리를 CICS Transaction Gateway의 bin 디렉토리로 설정하십시오.

다음 2가지 방식으로 CICS Transaction Gateway를 시작할 수 있습니다.

- 『preset 선택항목으로 게이트웨이 시작』
- 92 페이지의 『사용자가 지정한 선택항목으로 게이트웨이 시작』

preset 선택항목으로 게이트웨이 시작

사전 정의된 선택항목을 사용하여 게이트웨이를 시작하려면 명령 프롬프트에서 ctgstart를 입력하고 Enter 키를 누르십시오.

이런 방식으로 게이트웨이를 시작하면 기본 구성 설정값 또는 구성 도구를 사용하여 구성한 설정값이 사용됩니다(32 페이지의 『구성 도구 사용』 참조).

게이트웨이 콘솔 세션이 시작되며, 다음 메시지가 표시됩니다.

CCL6500I: 게이트웨이가 기본값으로 시작 중입니다.

이 메시지는 다음과 같은 사용 중인 값을 표시하는 행 앞에 옵니다.

CCL6502I: [초기 ConnectionManager = 1, 최대 ConnectionManager = 100,
CCL6502I: 초기 Worker = 1, 최대 Worker = 100, 포트 = 2345]

사용자가 지정한 선택항목으로 게이트웨이 시작

start 명령에 대해 정의할 수 있는 선택항목은 다음과 같습니다.

```
-port=port_number
-initconnect=number
-maxconnect=number
-initworker=number
-maxworker=number
-trace
-noinput
-tfile=pathname
-x
-tfilesize=number
-truncationsize=number
-dnsnames
-dumpoffset=number
-stack
```

여기서,

-port

게이트웨이에서 청취할 TCP/IP 포트 번호를 지정합니다.

-initconnect

ConnectionManager 쓰레드의 초기 번호를 지정합니다.

-maxconnect

ConnectionManager 쓰레드의 최대수를 지정합니다. 이 값을 *-1*로 설정하면 ConnectionManager 쓰레드 수에 한계가 적용되지 않습니다.

-initworker

Worker 쓰레드의 초기 번호를 지정합니다.

-maxworker

Worker 쓰레드의 최대수를 지정합니다. 이 값을 *-1*로 설정하면 ConnectionManager 쓰레드 수에 한계가 적용되지 않습니다.

-trace

표준 트레이스를 사용 가능으로 설정합니다(120 페이지의 『트레이스』 참조). 기본적으로, 트레이스 출력은 데이터 블록의 첫 128바이트(예: COMMAREA 또는 네트워크 플로우)만 표시합니다. CLASSPATH 변수 값 및 코드 페이지와 같은 기타 유용한 정보가 트레이스 출력 맨위에 표시됩니다.

구성 도구를 사용하여 게이트웨이를 구성한 경우, 트레이스 출력의 기본 목적지는 CICS Transaction Gateway가 설치되어 있는 bin 서브디렉토리의 CTG.TRC 파일입니다. 그 외의 경우, 트레이스 출력은 stderr에 기록됩니다.

-noinput

콘솔에서 입력을 읽을 수 없도록 합니다.

-tfile=pathname

트레이스가 사용 가능하면 *pathname*에 지정된 파일에 트레이스 출력이 기록됩니다. 그러면 트레이스 출력의 기본 목적지가 덮어쓰기됩니다(**-trace** 선택항목 참조).

- x** 전체 디버그 트레이스를 사용 가능화합니다(120 페이지의 『트레이스』 참조). 기본적으로, 트레이스 출력은 전체 데이터 블록(예: COMMAREA 또는 네트워크 플로우)을 보여줍니다. 또한 표준 트레이스보다 내부 게이트웨이 처리 정보를 많이 표시합니다. 트레이스 출력 목적지에 대한 정보는 **-trace** 및 **-tfile** 선택항목을 참조하십시오.

디버그 트레이스는 성능을 크게 저하시킵니다.

-tfilesize=number

트레이스 출력 파일의 최대 크기를 KB 단위로 지정합니다.

-truncationsize=number

number 값은 트레이스에 표시된 최대 데이터 블록 크기를 지정합니다. 이 선택항목을 **-trace** 또는 **-x** 선택항목과 함께 사용하여 기본 크기를 대체할 수 있습니다. 올바른 수는 양의 정수입니다. 0 값을 지정하면 트레이스에 데이터 블록이 표시되지 않습니다.

-dnsnames

메시지에 심볼릭 TCP/IP 호스트 이름을 표시할 수 있도록 합니다. 자세한 정보는 37 페이지의 『TCP/IP 호스트명 표시』를 참조하십시오.

-dumpoffset=number

number 값은 데이터 블록 표시가 시작되는 오프셋을 지정합니다. 오프셋이 표시할 총 데이터 길이보다 크면 오프셋 0이 사용됩니다.

-stack

Java 예외 스택 트레이스를 사용 가능화합니다(120 페이지의 『트레이스』 참

조). 게이트웨이 정상 조작 중 예상되는 예외를 포함한 Java 예외가 트레이스됩니다. 다른 트레이스 출력은 작성되지 않습니다. 트레이스 출력 목적지에 대한 정보는 **-trace** 및 **-tfile** 선택항목을 참조하십시오.

시작 기본값을 덮어쓰기하려면 명령 프롬프트에서 **ctgstart**를 입력한 후 필요한 시작 선택항목을 입력한 다음 Enter 키를 누르십시오. 명령 행에서 지정한 선택항목이 구성 도구를 사용하여 지정한 선택항목을 덮어쓰기합니다.

다음 시작 메시지가 표시됩니다.

```
CCL6501I: CICS Transaction Gateway가 사용자 지정 값으로 시작 중입니다.
```

이 메시지 뒤에는 사용 중인 값을 표시하는 행이 옵니다. 예를 들면, 다음과 같습니다.

```
CCL6502I: [ 초기 ConnectionManager = 10, 최대 ConnectionManager = 100,  
CCL6502I: 초기 Worker = 10, 최대 Worker = 100, 포트 = 2345 ]
```

시작 선택항목에 대한 도움말을 보려면 다음과 같이 입력하십시오.

```
ctgstart ?
```

게이트웨이 중지

게이트웨이를 중지시키려면 다음과 같이 수행하십시오.

- **-noinput** 선택항목으로 게이트웨이를 시작하지 않은 경우에는, 게이트웨이 콘솔 세션에 정확한 문자를 입력한 후 Enter 키를 눌러 게이트웨이를 중지할 수 있습니다. 허용할 수 있는 문자는 사용자 국가에 맞도록 국지화될 수 있으며, 허용된 기본 문자는 Q 또는 - 문자입니다.

게이트웨이 콘솔 세션에서 간단히 Enter 키를 눌러 게이트웨이를 중지시킬 문자를 판별할 수 있습니다. 다음과 같은 메시지가 표시됩니다.

```
CCL6508I: CICS Transaction Gateway를 중지하려면 Q 또는 -를 입력하십시오.
```

- **-noinput** 선택항목을 사용한 경우, 다음과 같은 다른 메소드를 사용하여 게이트웨이 프로세스를 중지시켜야 합니다.
 - **kill** 명령을 사용하십시오.

제7장 CICS Transaction Gateway 단말기 서블릿

이 장에서는 단말기 서블릿을 사용하여 웹 브라우저를 사용하는 CICS 3270 응용프로그램에 액세스하는 방법에 대해 설명합니다.

- 『CICS Transaction Gateway 단말기 서블릿이란?』에서는 서블릿이 무엇인지, 특히 CICS Transaction Gateway 단말기 서블릿이 실행하는 내용을 알려줍니다.
- 98 페이지의 『단말기 서블릿 설치 및 구성』에서는 단말기 서블릿을 설치하고 구성하는 방법에 대해 설명합니다.
- 104 페이지의 『단말기 서블릿 사용』에서는 단말기 서블릿과 함께 수행할 수 있는 내용, 웹 페이지에서 이를 호출할 수 있는 방법에 대해 설명합니다.
- 111 페이지의 『특성 및 매개변수 참조서』에서는 단말기 서블릿에 대해 지정할 수 있는 서블릿 특성의 완전한 참조를 제공합니다.
- 117 페이지의 『OS/390®용 CICS® Transaction Gateway 웹 인터페이스』에서는 CICS 웹 인터페이스가 사용하기 위해 생성한 HTML 템플릿을 단말기 서블릿이 사용하는 방법에 대해 설명합니다.

단말기 서블릿을 구성 및 사용하기 전에 이 장 전체를 읽는 것이 좋습니다.

CICS Transaction Gateway 단말기 서블릿이란?

서블릿은 웹 서버에서 실행되는 Java 프로그램이며 Webshere Application Server와 같이 Java가 작동 가능한 웹 서버 또는 서블릿 엔진 안에 있습니다(97 페이지의 그림8 참조). Java 서블릿은 CGI(Common Gateway Interface) 프로그램의 대용으로 일반화되었습니다. 단말기 서블릿은 CGI 프로그램이었던 IBM CICS Clients 버전 2와 함께 제공되는 CICS 인터넷 게이트웨이를 대체합니다.

서블릿은 웹 서버가 시작될 때 자동으로 로드되거나, 처음 서블릿 서비스를 클라이언트가 요청할 때 로드됩니다. CGI 프로그램과는 달리, 서블릿은 영속적입니다. 즉,

CICS Transaction Gateway 단말기 서블릿

한번 로드되면 실행 중인 상태로 유지하며 추가 클라이언트 요청을 기다립니다. 따라서 서블릿이 CGI 프로그램보다 빠르므로, 프로세스를 작성 및 삭제하는 데 시간을 낭비하지 않아도 됩니다.

Java 프로그램인 서블릿은 운영 체제 사이에서 이동할 수 있습니다. 또한 서블릿 인터페이스는 표준이므로 다른 플랫폼에 있는 다른 웹 서버와 함께 사용할 수 있습니다.

단말기 서블릿을 통해 웹 브라우저를 사용하여 CICS 3270 응용프로그램에 액세스할 수 있습니다. CICS 트랜잭션 시작, CICS 서버로부터 전송된 화면 정보 표시 및 다시 CICS로 화면 전송을 수행하기 위해 단말기 서블릿을 호출하는 웹 페이지를 작성합니다.

단말기 서블릿은 다음을 수행할 수 있습니다.

- 단순 터미널 에뮬레이터로 작동.
CICS 화면은 HTML 양식의 일부로 브라우저에 표시됩니다. 양식에서 단추 중 하나를 누르면 양식이 제출되고 데이터는 CICS로 전송됩니다.
- CICS 화면에서 데이터를 HTML 템플릿 파일로 대체.
HTML 템플릿 파일은 브라우저에 CICS 데이터를 표시하는 방법을 결정하는 데 사용됩니다. 단말기 서블릿은 CICS 화면 정보로 HTML 템플릿 파일의 변수를 대체하며, 그 결과는 브라우저로 전송됩니다. 자세한 정보는 108 페이지의 『변수 대체 사용』을 참조하십시오.
- 서버측 `includes`를 사용하여 웹 페이지에 있는 변수 정보를 통합하거나 사용자 간섭없이 터미널 에뮬레이터를 시작합니다.

서버측 `includes`는 HTML 파일(.shtml)에 있는 명령문이며, 웹 페이지가 브라우저로 리턴되기 전에 웹 서버에서 처리합니다. 예를 들어, 서버측 `include`를 사용하여, CICS 화면에 있는 필드값을 표시하기 위해 단말기 서블릿을 호출할 수 있습니다. 자세한 정보는 106 페이지의 『서버측 `include`로 서블릿 호출』을 참조하십시오.

- 웹 페이지로 CICS 화면 맵핑.

단말기 서블릿의 페이지 맵핑 특성을 사용하여, 특정 CICS 화면과 특정 웹 페이지를 연관시킬 수 있습니다. 예를 들어, 단말기가 쉬고 있고 실행되는 트랜잭션이 없을 때 웹 페이지가 표시되도록 지정할 수 있습니다. 자세한 정보는 115 페이지의 『페이지 맵핑 특성』을 참조하십시오.

EPI 프로그래밍처럼, 단말기 서블릿을 사용하면 기존의 CICS 트랜잭션에 대한 새로운 프론트엔드를 작성할 수 있습니다. 그러나 Java 프로그래밍의 강력하고, 유연성있는 플랫폼 독립성을 발전시킬 수도 있습니다.

주: 단말기 서블릿은 3270 데이터 스트림의 DBCS 필드를 지원하지 않습니다.

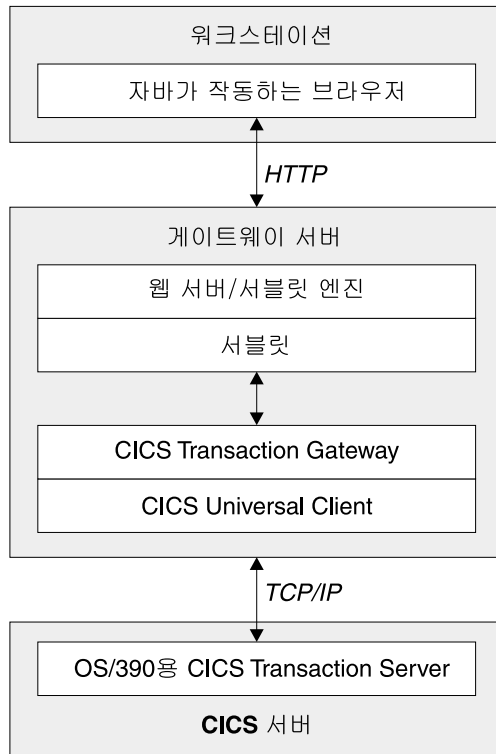


그림 8. URL로 호출된 단말기 서블릿을 사용하는 CICS Transaction Gateway

단말기 서블릿 설치 및 구성

이 절에서는 단말기 서블릿을 웹 서버에 설치하고, 단말기 서블릿의 초기화 매개 변수를 설정하는 방법에 대해 설명합니다.

단말기 서블릿 설치 및 구성 절차는 웹 서버에 따라 다양합니다. 그러나 적어도 다음 작업을 수행해야 합니다.

- 웹 서버의 CLASSPATH 및 PATH 설정 구성
- 단말기 서블릿을 웹 서버 구성에 추가
- 서블릿 초기화 매개변수 구성
- 보안, 로그인 및 웹 서버가 서버측 includes를 처리할 수 있는지 여부 등과 같은 구성 선택항목 고려.

특정 웹 서버에서의 단말기 서블릿 설치 및 구성 예제는 사용 가능한 구성 문서에 있습니다. 140 페이지의 『예제 구성 문서』를 참조하십시오. 해당 문서에서 설명된 원칙은 다른 웹 서버에도 적용되어야 합니다.

사용하는 웹 서버와는 관계없이, 서블릿을 설치 및 구성하는 지시사항에 대해서는 문서를 참조해야 합니다.

웹 서버 CLASSPATH 및 PATH 설정 구성

두 개의 CICS Transaction Gateway jar 파일(**ctgclient.jar** 및 **ctgserver.jar**)이 웹 서버 클래스 경로에 있는지 확인해야 합니다. WebSphere Application Server의 경우 WebSphere 관리 도구의 Java 엔진 섹션에서 이 작업을 수행하십시오.

정확한 클래스 경로 설정값을 작성하지 않으면 서블릿을 갱신할 수 없고 클래스 **com.ibm.ctg.servlet.TerminalServlet**을 로드할 수 없다는 메시지를 받게 됩니다.

또한 CICS Transaction Gateway bin 서브디렉토리가 웹 서버의 경로에 있는지 확인해야 합니다.

단말기 서블릿을 웹 서버 구성에 추가

웹 서버 구성에 단말기 서블릿을 추가하려면 서블릿 이름을 지정한 후 이 서블릿 인스턴스와 단말기 서블릿 클래스를 연관시켜야 합니다. 이 클래스는

com.ibm.ctg.servlet.TerminalServlet입니다. WebSphere Application Server의 경우 WebSphere 관리 도구의 서블릿 구성 섹션에서 이 작업을 수행하십시오.

하나 이상의 단말기 서블릿 인스턴스를 구성할 수 있습니다. 각 인스턴스는 서로 다른 서블릿 이름을 갖지만, 같은 서블릿 클래스를 참조합니다. 실제로, 연결하려는 각 CICS 서버에 대해 다른 단말기 서블릿 인스턴스를 작성해야 합니다. 각 단말기 서블릿의 인스턴스는 다른 초기화 매개변수를 사용할 수 있으므로, 다른 CICS 서버에 대해 구성할 때 유용할 수 있습니다.

서블릿 초기화 매개변수 구성

단말기 서블릿을 웹 서버의 구성에 추가한 후에는 단말기 서블릿에 대한 초기화 매개변수를 설정해야 합니다. WebSphere Application Server의 경우 WebSphere 관리 도구의 서블릿 구성 섹션에서 이 작업을 수행하십시오.

다음 절에서는 단말기 서블릿의 작동 방법으로, 적합한 서블릿 초기화 매개변수를 설정하는 데 도움이 되는 내용에 대해 설명합니다.

이제 단말기 서블릿에서는 사인 온 가능 단말기를 지원합니다. 이런 유형의 단말기를 제공하는 초기화 매개변수를 비롯하여 전체 초기화 매개변수 세트는 111 페이지의 『특성 및 매개변수 참조서』를 참조하십시오.

먼저 다음 몇 가지 매개변수 설정을 고려해야 합니다.

표 4. 서블릿 초기화 매개변수

매개변수 이름	주
servlet@propertyFile	서블릿은 특성 파일에서 구성 정보를 로드할 수 있습니다. /samples/terminalServlet 서브디렉토리에 샘플 servlet.properties 파일이 제공되어 있습니다. 특성 파일을 사용하려면 단말기 서블릿이 로드할 특성 파일의 전체 경로명으로 이 매개변수를 설정하십시오. 특성 파일을 지정하면 다른 모든 서블릿 매개변수에 대해서도 이 파일을 사용할 수 있습니다. 이 경우, 설정해야 할 매개변수는 servlet@propertyFile 입니다.

CICS Transaction Gateway 단말기 서블릿

표 4. 서블릿 초기화 매개변수 (계속)

매개변수 이름	주
servlet@extendedLogging	서블릿은 로그 정보(예를 들면, 오류 메시지)를 표준 서블릿 로그에 기록합니다. 이 특성을 true로 설정하여 좀 더 많은 정보를 로그 파일로 기록하십시오. 처음 서블릿을 구성할 때 그 유용성을 알 수 있지만, 일단 서블릿이 정확히 설정되고 나면 오프로 설정되어야 합니다.
pool@maxTerminals	이 특성을 CICS 서버에 대해 허용된 최대 연결 수로 설정하십시오. CICS 서버는 CICS 서버가 이 통신 갯수의 동시 연결을 처리하기에 충분한 타스크를 가지고 있는지 확인해야 합니다. 또한 CICS 클라이언트 최대 요청 수 구성 설정값이 적어도 이 특성보다 크도록 해야 합니다.
pool@serverName	단말기 서블릿의 각 인스턴스는 한 CICS 서버에만 연결할 수 있습니다. 이것으로 자원 이용 및 보안을 보다 잘 제어할 수 있게 되었습니다. 구성에서 해당 서버 이름 설정값에서 정의한 것처럼, 이 특성을 필요한 CICS 서버 이름으로 설정하십시오. 이 특성을 설정하지 않으면 기본 CICS 서버가 사용됩니다.
pool@gatewayUrl	서블릿이 네트워크를 통해 CICS Transaction Gateway와 연결하려는 경우 이 특성을 설정하십시오.

특성 파일의 이름과는 별도로, 모든 서블릿 특성은 특성 파일에서 로드되거나, 서블릿 초기화 매개변수로 설정될 수 있습니다. 초기화 매개변수는 특성 파일 입력항목을 대체합니다.

서블릿 특성 이름에서는 대소문자가 구분되지 않습니다.

단말기 풀링

단말기 서블릿은 단말기 풀을 사용하며, 서블릿 구성 특성을 사용하여 단말기 풀의 작동을 제어할 수 있습니다. 기본 작동은 사용자가 단말기를 요구할 때마다 새로운 단말기가 최대 **pool@maxTerminals** 한계까지 CICS로 연결되는 것입니다. 사용자가 단말기와의 작업을 완료하면 연결해제됩니다.

pool@minTerminals 특성을 설정하면 서블릿이 시작될 때 이 갯수의 단말기가 CICS로 연결되며, 이 갯수까지 CICS에 연결될 수 있지만 항상 사용할 수 있는 것은 아닙니다. 이를 사용하면 새로운 사용자가 연결을 기다리지 않고도 즉시 단말기를 할당받을 수 있습니다.

pool@reusingTerminals 특성을 설정하면 단말기를 다시 사용할 수 있습니다. 즉, 사용자 세션이 종료되면 단말기를 연결 해제하지 않고 새로운 사용자에게 할당할 수 있습니다. **pool@minTerminals**인 경우, 이 특성은 아무 효과가 없습니다. 단말기에서 실행 중인 모든 트랜잭션은 다시 사용되기 전에 종료되고 화면은 지워집니다. 그러나 사용자가 CICS(예를 들면, CESN을 사용하여)로 사인 온하면 단말기는 다시 사용될 때 계속 사인 온하게 됩니다. 이러한 이유로, 환경에 대해 적합한 지 확인할 수 없으면 이 특성을 설정하지 마십시오.

pool@idleTimeout 특성을 사용하면 사용자에게 할당된 단말기가 지정된 기간 동안 사용되지 않을 경우 해제됩니다. 이 특성의 기본값은 0 즉, 시간 종료 없음입니다.

pool@connectedTerminals, **pool@freeTerminals**, **pool@terminalsInUse** 값으로 표시 요청을 사용하면 특정 단말기 서블릿 인터페이스에서 사용 중인 단말기, 연결된 단말기, 사용 가능 단말기 수를 조회할 수 있습니다.

페이지 맵핑

페이지 맵핑은 단말기 서블릿에게 웹 페이지가 특정 CICS 세션 또는 단말기 상태에 대해 표시할 내용을 알려줍니다.

표시할 페이지에 대한 서블릿의 선택사항은 현재 페이지의 맵핑 설정값, 현재 화면 핸들러 클래스 및 단말기 상태를 근거로 합니다.

먼저 단말기 서블릿은 페이지 맵핑 특성을 찾아 다음과 같이 현재 상태의 세션에 대해 특정 페이지 세트가 있는지 여부를 참조합니다.

세션 상태	페이지 맵핑 특성
오류가 발생했음	page@error
단말기가 사용되고 있지 않음 - 실행 중인 트랜잭션이 없음	page@idle
단말기가 연결해제됨	page@disconnected
현재 세션에 대한 화면 핸들러가 있음	화면 핸들러의 클래스 이름에 대한 페이지 맵핑

특정 페이지가 없으면 페이지 맵핑 특성인 **page@default**로 지정된 페이지가 사용됩니다. 이 특성을 설정하지 않으면 대신 오류 메시지가 표시됩니다.

CICS Transaction Gateway 단말기 서블릿

페이지 맵핑 특성값은 2가지 방식으로 설정될 수 있습니다.

1. URL, 예를 들면, 다음과 같습니다.

```
http://webserver/index.html
```

2. 변수 대체를 위해 로드된 템플릿 파일. 이것은 "file://"로 시작되어야 합니다.

예를 들면, 다음과 같습니다.

```
file:///d:/pages/template.html
```

servlet@templateDir 특성을 설정하면 템플릿 디렉토리를 기준으로 한 상대 템플릿 파일 이름이 지정될 수 있습니다.

페이지 맵핑 특성에 대한 설정값 예제는 다음과 같습니다.

```
page@error=http://server/errors.html
page@idle=http://server/idle.html
page@disconnected=/ctglab/html/servlet/epissam.html
page@default=file:///epissam1.html
```

특정 화면의 경우 페이지 맵핑을 설정하려면 해당 화면을 인식할 수 있는 화면 핸들러를 작성해야 합니다. 그런 다음 단말기 서블릿에서 화면 핸들러 Bean을 로드할 수 있도록 **pool@handlerPath** 특성을 설정해야 합니다. 예를 들어, testmaps 패키지에서 MAP1ScreenHandler 화면 핸들러 Bean을 작성하면 해당 화면 핸들러에 대한 페이지 맵핑 특성은 다음과 같습니다.

```
page@testmaps.MAP1ScreenHandler.
```

클래스 이름에서는 대소문자가 구분됩니다. 표준 페이지 맵핑과 마찬가지로, 표시할 웹 페이지 이름으로 특성을 설정합니다. 예를 들면, 다음과 같습니다.

```
page@testmaps.MAP1ScreenHandler=http://server/test.html
```

페이지 맵핑은 요청 매개변수에서 단말기 서블릿으로 설정할 수도 있습니다. 이런 경우 페이지 맵핑은 해당 요청에만 적용되며, 사용되는 서블릿 인스턴스와 연관된 페이지 맵핑을 덮어쓰기합니다. 해당 요청으로 페이지 맵핑의 현재 설정값을 조회할 수 있습니다(116 페이지의 『표시가능한 특성』 참조).

화면 핸들러 Bean 및 단말기 연결해제

단말기가 연결해제되려면, 실행 중인 트랜잭션에서 나가야 합니다. 이를 수행하기 위한 단말기 서블릿의 기본 작동은 AID PF3 키를 CICS로 전송하는 것입니다. 그

러나 작동하지 않는 트랜잭션을 실행할 경우, 관련 화면의 종료 방법을 알고 있는 하나 이상의 화면 핸들러 Bean을 작성해야 합니다.

CICS Transaction Gateway가 지원하는 BMS 맵 변환 유틸리티(BMSMapConvert)를 사용하여 BMS 맵에서 자동으로 화면 핸들러 Bean을 생성하거나, 자신의 클래스를 작성할 수 있습니다. 화면 핸들러 및 그 작성법에 대해서는 *CICS Transaction Gateway: Gateway Programming* 책을 참조하십시오.

화면 핸들러를 서블릿으로 로드하려면 **pool@handlerPath** 특성을 설정하십시오. 핸들러 경로에는 서브디렉토리를 찾지 못할 경우에도 .jar 파일, .zip 파일 및 디렉토리를 포함시킬 수 있습니다. 예를 들어, 서블릿 특성 파일에서 다음을 추가하여 `pool@handlerPath=d:/handlers/handlers.jar;d:/handlers/test`

.jar 파일 및 디렉토리에서 화면 핸들러 클래스를 로드할 수 있습니다. 이 특성 설정값에 대해서는 예제 특성 파일을 참조하십시오.

또한 **pool@defaultHandler** 특성을 사용하여 기본 화면 핸들러를 지정할 수도 있으며, 이 핸들러는 다른 화면 핸들러에서 화면을 인식하지 못할 경우 화면에서 나가는 데 사용됩니다. 예제 기본 화면 핸들러는 AID PF3을 CICS에 전송하는 **com.ibm.ctg.epi.DefaultScreenHandler**입니다. 이 클래스에 대한 소스는 `samples/java/com/ibm/ctg/epi` 디렉토리에 있습니다. 클래스 경로에서 또는 핸들러 경로에서 로드될 수 있는 화면 핸들러는 기본 화면 핸들러로 사용될 수 있습니다.

서블릿은 실행 중인 트랜잭션을 종료하려고 계속해서 시도하지 않습니다. **pool@exitRetryLimit** 특성은 트랜잭션을 종료하기 위해 시도하는 횟수를 제어합니다. 이 특성의 기본값은 0이지만, 사용자 환경에 대해 충분히 크도록 해야 합니다. 트랜잭션을 종료하기 위해 지정된 횟수만큼 시도한 후에도 단말기에 계속 실행 중인 트랜잭션이 있음을 서블릿에서 발견하면 단말기를 '작동안함' 상태로 간주하여 연결해제하거나 사용자에게 할당하지 않습니다. 작동하지 않는 단말기를 버리는 유일한 방법은 서블릿을 로드해제하고, CICS Transaction Gateway를 중지 및 다시 시작한 다음 다시 서블릿을 로드하는 것입니다.

서블릿 특성과 제공된 화면 핸들러 Bean을 구성했으면 작동하지 않는 단말기가 없어야 합니다.

기타 구성 선택항목 고려사항

고려해야 하는 기타 구성 선택항목은 다음과 같습니다.

- 보안
단말기 서블릿에 대한 액세스를 제어해야 합니다.
- 로깅
단말기 서블릿은 표준 서블릿 로그로 로그 정보를 기록합니다. 로그 온 또는 오프로 설정할 수 있습니다.
- 세션 추적
세션 추적을 사용하도록 단말기 서블릿을 구성해야 합니다.
- 서버측 includes
서버측 includes를 처리하도록 웹 서버를 구성하려 할 수 있습니다.

주: IBM WebSphere 응용프로그램 서버 버전 1.01은 서블릿에 의한 DBCS 데이터의 입력 또는 출력을 완전히 지원하지 않으며, 기타 서블릿 엔진도 같은 제한을 받습니다. 이 상태에서, 단말기 서블릿은 DBCS 문자를 ????(물음표)로 표시합니다.

단말기 서블릿 로드

단말기 서블릿을 구성하고 설정값을 저장한 후에는, 이를 로드해야 합니다. 이를 즉시 로드할 것인지 또는 웹 서버가 시작할 때마다 로드할 것인지 여부를 선택할 수 있습니다.

단말기 서블릿을 제대로 구성하지 않으면 웹 서버에서 단말기 서블릿을 로드하려 할 때 오류 메시지가 표시됩니다. 자세한 정보는 *CICS Transaction Gateway: Gateway Messages* 서적의 Terminal Servlet 절을 참조하십시오.

단말기 서블릿 사용

이 절에서는 단말기 서블릿 사용법에 대해 설명합니다. 단말기 서블릿을 호출할 수 있는 다른 방법, 단말기 서블릿을 사용하는 웹 페이지를 개발하는 다른 방법에 대해 설명합니다.

/samples/terminalervlet 서브디렉토리에 샘플 파일이 있습니다. 샘플은 /samples/Samples.txt 파일에 설명되어 있습니다.

CICS로 연결 및 트랜잭션 시작

단말기 서블릿 인스턴스는 한 CICS 서버에만 연결될 수 있습니다. 액세스를 허용하려는 각 CICS 서버에 대해서는, 서블릿 인스턴스를 작성합니다. 또한 다른 초기화 매개변수를 갖는 서블릿의 인스턴스를 작성할 수도 있습니다.

CICS 서버에 대한 트랜잭션을 시작하려면 적합한 요청으로 단말기 서블릿을 호출해야 합니다.

단말기 서블릿 호출

단말기 서블릿을 호출하는 데에는 3가지 방식이 있습니다.

- URL로.
- HTML 양식으로.
- 서버측 include로.

각각의 경우, 사용하려는 서블릿 인스턴스의 이름을 알고, 수행할 조치를 알려주는 하나 이상의 요청 매개변수를 서블릿에 전달해야 합니다. 매개변수 이름과 값은 모든 경우에 같지만, 매개변수를 지정하는 방식은 다릅니다.

CICS 서버에서 트랜잭션을 시작하기 위해 제공해야 하는 매개변수는 다음과 같습니다.

매개변수 이름	매개변수 값
request	send
transaction	시작하려는 트랜잭션의 트랜잭션 ID. 그러나 ATI 트랜잭션은 단말기 서블릿에 대해 시작될 수 없음을 기억하십시오.

서블릿은 필요한 경우 사용자에게 단말기를 할당합니다.

전체 요청 매개변수 세트는 115 페이지의 『요청 매개변수』에 나열되어 있습니다.

URL로 서블릿 호출

서블릿 인스턴스 TerminalServlet을 호출하려면 다음 URL로 링크하십시오.

```
http://your_webserver/servlet/TerminalServlet
```

여기서 *your_webserver*는 웹 서버의 호스트 이름입니다.

URL에 있는 요청 매개변수를 조회 문자열 양식으로 인코딩할 수 있습니다. 예를 들면, 다음과 같습니다.

```
http://your_webserver/servlet/TerminalServlet?request=send&transaction=CECI
```

사용자가 정보를 입력할 필요가 없을 때 서블릿을 호출하는 이 메소드를 사용해야 합니다.

HTML 양식으로 서블릿 호출

이 메소드에서는 웹 페이지에서 HTML 양식을 작성합니다. 예를 들어, 다음은 TerminalServlet 서블릿을 호출하는 양식입니다.

```
<FORM METHOD="GET" ACTION="/servlet/TerminalServlet">  
(텍스트 입력, 단추 및 프롬프트가 위치할 태그 부분)  
</FORM>
```

METHOD는 GET이나 POST가 될 수 있습니다. 양식의 여러 구성요소에는 이름과 값이 있으며, 이들은 양식이 제출될 때 서블릿으로 전송됩니다. 특정 요청 매개변수를 설정하려는 경우, 양식에서 이를 숨겨진 항목으로 추가하십시오. 예를 들면, `<input type="hidden" name="request" value="send">`입니다. 숨겨진 입력은 항상 서블릿으로 전송됩니다. 일반적으로, 양식 구성요소의 이름은 요청 매개변수의 이름이며, 그 값은 요청 매개변수의 값입니다.

이것은 사용자가 입력한 정보를 서블릿으로 전송하는 유일한 방법입니다.

제공된 샘플 `epissam3.html`에서는 HTML 양식 사용 방법에 대해 설명합니다. 제품 CD의 `Samples.txt` 파일을 참조하십시오.

서버측 include로 서블릿 호출

서버측 include는 웹 페이지가 사용자에게 전송되기 전에 웹 서버에 의해 처리됩니다. 서블릿이 호출되고, 생성한 출력은 서버측 include 태그 대신 웹 페이지에 포함됩니다. 사용자만이 웹 페이지에서 서블릿 출력을 보게 됩니다.

HTML 소스에서, 서버의 서버측 include는 다음과 같습니다.

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="request" VALUE="send">
<PARAM NAME="transaction" VALUE="CECI">
<PARAM NAME="display" VALUE="none">
</SERVLET>
```

서버측 include는 변수 정보를 웹 페이지에 추가하거나, 단말기를 시작하여 사용자를 개입시키지 않고 웹 페이지가 표시될 때마다 일부 조치를 수행하는 데 사용할 수 있습니다.

원하는 만큼의 많은 서버측 include를 웹 페이지에 추가할 수 있지만, 다음을 기억하십시오.

- 한 페이지에 있는 서버측 includes는 병렬로 처리될 수 있습니다. 따라서 일반적으로 소스에서 발생한 순서인 경우라도 수행되는 순서는 확실하지 않을 수도 있습니다.
- 서블릿은 사용자에게 할당된 단말기에 대한 액세스를 허용하는 세션 정보를 저장합니다. 사용자 세션이 서버측 include에 의해 호출되면 같은 페이지에 있는 후속 서버측 includes는 세션 정보에 액세스할 수 없게 됩니다. 일단 페이지가 사용자에게 전송되면 세션이 설정되고, 서버측 include가 여럿인 페이지는 제대로 작동합니다.

제공된 샘플 epissam2.shtml에서는 서버측 include 메소드에 대해 설명합니다. 제품 CD의 Samples.txt 파일을 참조하십시오.

다음 내용

서블릿이 양식에서 호출되거나 URL로의 링크에 의해 호출되면 일부 출력은 브라우저로 다시 전송되어야 합니다. 사용자에게 표시되어야 하는 내용을 결정하기 위해, 서블릿은 사용자가 설정한 페이지 맵핑 특성을 사용합니다. 101 페이지의 『페이지 맵핑』의 내용을 참조하십시오.

화면 및 필드 표시

웹 페이지에 CICS 화면 정보를 포함시키는 데에는 두가지 방법이 있습니다.

- 서버측 includes

단말기 서블릿 사용

- 변수 대체

원하면 두가지 접근 방식을 혼합할 수 있습니다.

서버측 includes 사용

웹 서버는 서버측 includes를 처리하도록 구성되어야 합니다. 서블릿을 작성하는 서버측 includes가 있는 SHTML 페이지를 작성합니다. 서블릿의 출력은 서버측 include 대신 페이지로 삽입됩니다.

예를 들어, 다음과 같이 HTML 양식의 일부로 화면을 표시하거나,

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="display" VALUE="screen">
</SERVLET>
```

화면에 있는 22번째 필드 내용을 다음과 같이 표시합니다.

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="display" VALUE="22">
</SERVLET>
```

변수 대체 사용

서블릿이 로드할 HTML 템플릿 파일을 작성합니다. 서블릿은 파일에 있는 변수를 CICS 화면 정보로 대체하며, 그 결과는 브라우저로 전송됩니다. 페이지 맵핑 특성을 사용하여 서블릿에게 템플릿 파일이 로드할 내용을 알려줍니다(101 페이지의 『페이지 맵핑』 참조).

예를 들어, CESN 사인 온 화면에 대한 템플릿 파일은 다음과 같습니다.

```
<html>
<head>
<title>CICS Signon</title>
</head>
<body>
<h1>CICS Signon</h1>
<form method="POST" action="/servlet/TermServlet">
<input type="hidden" name="request" value="send">
<pre>
  Userid: <input type="text" name="USERID" value="&USERID;" size="8" >
  Password: <input type="text" name="PASSWORD" value="&PASSWORD;" size="8" >
</pre>
<input type="submit" name="DFH_ENTER" value="Sign on">
<input type="submit" name="DFH_PF1" value="Help">
<input type="submit" name="DFH_PF3" value="Exit">
</form>
</body></html>
```

여기서 변수는 &USERID; 및 &PASSWORD;입니다. 숫자로 필드를 식별하여 필드 이름을 사용할 수 있습니다. 이 예제에서는 화면을 인식하고 이름으로 필드를 설정할 수 있는 화면 핸들러가 있어야 합니다.

페이지 맵핑 특성값은 HTML 템플릿 파일 이름 또는 URL이 될 수 있습니다. 페이지 맵핑 특성이 HTML 템플릿을 식별함을 서블릿에서 알고 있으므로, 이를 `file://template_filename` 값으로 설정해야 합니다. 예를 들면,
`file:///d:/html/templates/test.html`.

표시되는 정보

서버측 include로 표시할 수 있는 정보는 변수 대체에 사용될 수 있으며, 반대의 경우도 가능합니다. 템플릿에 사용되는 변수명은 서버측 include에 있는 **display** 매개변수 값과 같습니다.

일반적으로, 다음을 표시할 수 있습니다.

- HTML 양식 일부로서의 화면
- 필드 - 필드에 대한 화면 핸들러 Bean이 있는 경우 번호로 또는 이름으로 식별됨
- 서블릿 구성 설정값
- 연결됨, 사용 중 및 미사용 상태인 단말기 수

표시될 수 있는 특성 세트 모두는 111 페이지의 『특성 및 매개변수 참조서』에 나열되어 있습니다.

CICS로 다시 화면 전송

CICS로 화면을 다시 전송하려면 "send"로 설정된 **request** 매개변수로 서블릿을 호출해야 합니다. **transaction** 매개변수를 제공할 필요는 없습니다. 트랜잭션이 이미 수행되고 있는 경우는 무시됩니다. 화면을 CICS로 다시 전송하기 전에 새로운 정보로 화면을 갱신하려면 필드를 식별하는(이름 또는 번호로) 요청 매개변수와 설정해야 하는 내용을 지정합니다.

예를 들어, 서버측 include는 다음과 같습니다.

단말기 서블릿 사용

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="request" VALUE="send">
<PARAM NAME="USERID" VALUE="sysad">
<PARAM NAME="PASSWORD" VALUE="sysad">
<PARAM NAME="display" VALUE="none">
</SERVLET>
```

USERID 및 PASSWORD 필드를 제공된 값으로 설정한 다음, 화면을 CICS에 전송합니다. HTML 양식에서, 양식 구성요소 `<input type="text" name="USERID" value="" size="8">`을 사용할 수 있습니다. 사용자가 양식을 제출하면 요청 매개 변수 USERID는 사용자가 입력한 값으로 설정됩니다.

AID 설정

기본 AID는 Enter입니다.

HTML 양식에서 AID를 설정하려면 제출 단추를 포함시키십시오. 이것은 다음과 같이 지정됩니다.

```
<input type="submit" name="DFH_PF3" value="Exit" >.
```

사용자가 단추를 누르면 서블릿에 "Exit" 값(이것은 서블릿이 무시함)을 가진 요청 매개변수 "DFH_PF3"이 전달됩니다. AID는 PF3으로 설정됩니다. 서버측 include에서, 다음과 같이 AID를 설정할 수 있습니다.

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="request" VALUE="send">
<PARAM NAME="DFH_PF3" VALUE="ignored">
<PARAM NAME="display" VALUE="none">
</SERVLET>
```

전체 요청 매개변수 세트는 111 페이지의 『특성 및 매개변수 참조서』에 나열되어 있습니다.

연결해제

서블릿이 단말기를 사용자에게 할당하면 다음이 발생할 때 까지 유지됩니다.

- 서블릿이 "disconnect"로 설정된 매개변수 "request"와 함께 호출
- 웹 서버 만기시의 사용자 세션

- **pool@idleTimeout** 특성을 설정한 경우 단말기는 지정된 기간 동안 액세스되지 않으면 자동으로 연결해제 됩니다.
- CICS 클라이언트 중지
- CICS 서버로의 연결이 유실됨 - 또는 서버가 다운됨
- 서블릿이 웹 서버에 의해 언로드됨

더 이상 단말기가 필요하지 않으면 단말기를 명시적으로 해제해야 합니다.

사용자가 서블릿과의 세션을 설정하고 단말기가 할당되면 사용자가 액세스하는 서블릿 인스턴스는 세션을 계속합니다. 사용자가 연결해제하거나 다른 이유로 Euro 세션이 종료될 때 까지 같은 단말기가 사용됩니다. 특정 요청에 사용되는 페이지 맵핑 및 기타 특성은 요청을 처리하는 서블릿 인스턴스에 대해 설정된 것입니다.

요청 매개변수(예를 들면, 페이지 맵핑)로 설정될 수 있는 특성은 서블릿의 설정값을 덮어쓰기하지만, 보통 현재 요청에만 적용되며 다른 사용자 요청 또는 같은 사용자에 의한 후속 요청에는 적용되지 않습니다.

특성 및 매개변수 참조서

이 절에서는 단말기 서블릿 특성에 대해 설명합니다.

서블릿 구성 특성

일반적으로, 이런 특성은 특성 파일에서, 웹 서버 구성시에는 서블릿 초기화 매개변수로 지정될 수 있습니다. 이름에서는 대소문자를 구분하지 않습니다. 서블릿 초기화 매개변수로 제공된 값은 특성 파일에서 로드된 설정값을 덮어쓰기합니다. **servlet@debug** 및 **servlet@extendedLogging** 특성 외에, 서블릿 실행 중 특성 설정값을 변경할 수 없습니다. 그러나 요청 매개변수 **display**를 사용하면 특성의 현재 설정값을 조회할 수 있습니다. 116 페이지의 『표시가능한 특성』의 내용을 참조하십시오.

servlet@coloring

Y로 설정하면 3270 데이터 스트림에 표시된 색상으로 지정된 필드가 표시됩니다.

특성 및 매개변수 참조서

HTML의 제한점 때문에, 비보호 필드에는 색상을 설정할 수 없으며 항상 흰색 배경에 검은색 텍스트로 표시됩니다.

특성을 지정하지 않으면 단말기 서블릿은 흰색 배경에 검은색 텍스트의 화면을 표시합니다.

servlet@debug

true로 설정하면 서블릿의 모든 인스턴스에 대해 트레이스가 온으로 설정됩니다. 서블릿이 실행하고 있는 동안에는 트레이스를 온이나 오프로 설정될 수 있으며, 출력은 표준 출력으로 지정됩니다. 트레이스를 온으로 설정하면 성능에 상당한 영향을 줍니다.

servlet@extendedLogging

true로 설정하면 확장된 로그온이 온으로 설정됩니다. 출력은 서블릿 로그로 보내집니다.

servlet@propertyFile

로드할 특성 파일의 전체 경로명. 이 특성은 보통 서블릿 초기화 매개변수로만 설정될 수 있습니다.

servlet@templateDir

HTML 템플릿 파일이 로드될 수 있는 디렉토리 이름. 이 특성을 설정하면 사용자가 템플릿 디렉토리 또는 그 서브디렉토리에 없는 파일을 볼 수 없게 됩니다. 이 특성을 설정하지 않으면 서블릿은 액세스하는 웹 서버의 파일을 로드할 수 있습니다.

pool@defaultHandler

특정 핸들러를 찾을 수 없을 때 화면을 나가기 위해 사용할 화면 핸들러. 이것을 화면 핸들러 경로에서 로드된 화면 핸들러 중 하나로 설정하면 핸들러 경로가 기본 핸들러 앞에 설정되도록 하십시오.

pool@deviceType

사용할 단말기 모델 정의. 이 특성을 설정하지 않으면 기본값이 사용됩니다.

pool@encoding

사용할 Java 인코딩을 지정할 수 있습니다. 자세한 설명은 *CICS Transaction Gateway: Gateway Programming* 서적의 "부록 A. Java encoding"을 참조하십시오.

pool@exitRetryLimit

단말기 연결해제를 수행하는 동안 서블릿이 실행 중인 트랜잭션을 나가려고 시도하는 횟수. 이것으로 화면을 종료할 수 없는 경우 루핑이 예방됩니다.

pool@gatewayUrl

서블릿이 연결해야 하는 CICS Transaction Gateway의 URL. 기본값은 국지 게이트웨이 즉, local://을 사용하는 것입니다.

pool@handlerPath

화면 핸들러 Bean이 로드되는 경로. 디렉토리, .jar 파일 및 .zip 파일이 포함될 수 있지만 서브디렉토리는 탐색되지 않습니다.

pool@idleTimeout

단말기가 자동으로 연결해제되기 전에 유휴상태로 남아 있을 수 있는 최대 시간(초 단위). 기본값은 0이며, 이는 시간 종료 없음 의미합니다.

pool@installTimeout

장애가 발생하기 전까지 단말기를 설치할 수 있는 최대 시간(초 단위). 유효한 범위는 0 - 3600입니다. 기본값은 0이며, 시간 종료 없음 의미합니다.

pool@maxTerminals

동시에 CICS에 연결될 수 있는 최대 단말기 수. 기본값은 5입니다.

pool@minTerminals

사용되지 않을 때 CICS에 연결된 채로 유지해야 하는 단말기 수. 기본값은 0입니다.

pool@password

작성시 사인 온 불가능 단말기와 연관할 암호를 지정합니다.

pool@readTimeout

시간 종료 전까지 대화식 트랜잭션이 허용되는 최대 시간(초 단위). 유효한 범위는 0 - 3600입니다. 0 값은 시간 종료 없음 의미합니다.

pool@reusingTerminals

true로 설정하면 연결해제된 단말기가 다음 사용자를 위해 연결된 채로 유지됩니다. 기본 작동은 사용자 세션이 종료될 때 단말기를 연결해제하는 것입니다. 또한 이 특성을 적용하려면 **pool@minTerminals**를 설정해야 합니다.

pool@serverName

연결할 CICS 서버 이름. 이 특성을 설정하지 않으면 구성에서 정의된 대로 기본 CICS 서버가 사용됩니다.

pool@signonCapability

확장된 단말기의 사인 온 기능을 지정합니다. 올바른 입력항목은 EPI_SIGNON_CAPABLE 또는 EPI_SIGNON_INCAPABLE입니다.

pool@pool@userid

작성시 사인 온 불가능 단말기와 연관된 사용자 ID를 지정합니다.

screen@cursoring

true로 설정하면 JavaScript는 화면 커서 위치를 설정합니다. 기본 설정값은 false입니다.

screen@NeutralColor

화면 핸들러에서 분명치 않은 3270 색상을 변환할 HTML 색상을 지정합니다.

screen@stripBlanks

true로 설정하면 웹 브라우저는 필드 텍스트를 형식화합니다.

screen@stripEmptyRows

true로 설정하면 화면 핸들러에서 데이터가 없는 화면 행을 제거합니다.

screen@TrimFields

true로 설정하면 화면 핸들러에서 입력 필드의 시작과 끝에서 과도한 공백을 제거합니다.

페이지 맵핑 특성

페이지 맵핑은 웹 페이지를 특정 화면 또는 단말기 상태와 연결시킵니다. 페이지 맵핑은 웹 서버를 구성할 때 특성 파일에서, 서블릿 초기화 매개변수로 지정될 수 있습니다. 그러나 요청 매개변수 **display**를 사용하면 페이지 맵핑의 현재 설정값을 조회할 수 있습니다. 116 페이지의 『표시가능한 특성』의 내용을 참조하십시오.

표준 페이지 맵핑 특성	값
page@disconnected	단말기가 연결해제될 때 표시할 페이지
page@error	오류가 발생할 때 표시할 페이지
page@idle	단말기가 유휴상태일 때 즉, 트랜잭션이 수행되고 있지 않을 때 표시할 페이지
page@default	지정된 다른 페이지가 없는 경우의 페이지
page@screen	이 화면에 대해 표시할 페이지

요청 매개변수

이런 매개변수는 요청할 때에만 설정됩니다. 또한 요청시 설정할 수 있는 다른 특성을 지정할 수도 있습니다. 예를 들면, **servlet@extendedLogging**입니다.

매개변수 이름	매개변수 값
request	"send"로 설정되어 화면을 CICS로 전송하거나, "disconnect"로 설정되어 세션을 종료합니다. 필요한 경우 새로운 단말기 세션이 시작됩니다.
display	표시될 수 있는 특성 이름. 116 페이지의 『표시가능한 특성』 참조.
translateText	"true"로 설정되어 문자 <, > 및 &를 < > 및 &로 각기 변환합니다. 이것은 텍스트를 표시할 때 사용됩니다.
transaction	시작될 트랜잭션의 트랜잭션 ID. 이것은 요청이 "send"이고 단말기 상태가 유휴상태일 때에만 사용됩니다. 모든 트랜잭션이 단말기 서블릿에 대해 시작될 수 있는 것은 아닙니다.
transactionData	트랜잭션에 전달될 추가 매개변수. 이것은 요청이 "send"이고 단말기 상태가 유휴상태이며 트랜잭션이 지정되었을 때에만 사용됩니다.

특성 및 매개변수 참조서

매개변수 이름	매개변수 값
DFH_CURSOR	화면 커서가 위치해야 하는 필드(필드 색인 또는 번호), 또는 화면 핸들러 Bean을 현재 화면에서 사용할 수 있는 경우는 필드 이름. 이 매개변수 이름은 대문자여야 합니다.
DFH_ENTER, DFH_CLEAR, DFH_PA1 - 3, DFH_PF1 - 24	값은 무시됩니다. 이들 매개변수 중 하나를 사용하여 CICS에 전송할 AID 키를 지정하십시오. 이것은 대문자여야 합니다.
필드 색인	필드에서 설정할 텍스트. 필드 색인을 알 경우는 필드 내용을 설정하는 데 사용됩니다.
usingSession	필요한 단말기 세션이 없을 때 false로 설정됩니다. 이 매개변수가 설정되면 이 요청에 사용되는 단말기는 사용된 뒤 버려집니다.
matchOnIdle	기본값인 false로 설정합니다. 보통 서블릿은 단말기 상태가 '클라이언트'일 때 즉, 대화식 또는 의사대화식 트랜잭션이 응답을 기다리고 있을 때에만 현재 화면에 대한 ScreenHandler를 찾으려 합니다. 이 특성을 true로 설정하면 화면 상태와는 관계없이 화면이 변경될 때마다 서블릿은 현재 ScreenHandler를 점검합니다. BMS 맵을 사용하는 트랜잭션은 있지만 대화식 또는 의사대화식이 아닌 경우 이를 수행하려 합니다. ScreenHandlers 수가 많은 경우, 서블릿은 이 특성이 true로 설정될 때 느리게 수행됩니다. 이러한 이유로, 기본값은 false입니다.

뿐만 아니라 화면 핸들러 Bean을 현재 화면에서 사용할 수 있으면 요청시 이를 지정하여 해당 Bean에 대한 특성을 설정할 수 있습니다. BMSMapConvert가 생성한 Bean의 경우, BMS 맵에 정의된 필드명은 필드에서 필수 테스트로 설정될 수 있는 특성입니다.

표시가능한 특성

다음 표에서는 "display" 요청 매개변수에 대해 지정되거나 변수 대체에 사용될 수 있는 값을 보여줍니다. 108 페이지의 『변수 대체 사용』을 참조하십시오.

디스플레이 값	표시내용
none	아무 것도 표시하지 않습니다. 서버측 include 에서 출력을 저지하는 데 사용됩니다.
screen	HTML 양식 일부로서의 화면.
errorMessage	오류가 발생한 경우의 오류 메시지.
errorCause	오류가 발생한 경우 오류에 대한 추가 정보.
DFH_CURSOR	필드 이름을 결정할 수 없는 경우는 커서가 있는 필드 이름. 결정할 수 있는 경우는 필드 색인.
필드 색인	필드 텍스트
화면 핸들러 특성명	결정할 수 없는 경우의 특성값.
위에 나열된 서블릿 구성 특성 중 임의의 것.	현재 서블릿 인스턴스에 대한 특성값.
페이지 맵핑 특성	현재 서블릿 인스턴스에 대한 특성값.
요청 매개변수, "request" 및 "display"의 경우는 제외됨.	현재 요청의 특성값.
pool@connectedTerminals	이 서블릿 인스턴스의 경우, 현재 연결된 단말기 수.
pool@freeTerminals	이 서블릿 인스턴스의 경우, 사용되지 않는 현재 연결 단말기 수.
pool@terminalsInUse	이 서블릿 인스턴스의 경우, 사용자에게 할당된 현재 단말기 수.

OS/390[®]용 CICS[®] Transaction Gateway 웹 인터페이스

OS/390[®]용 CICS[®] Transaction Gateway 버전 1.2 이상에 포함된 CICS 웹 인터페이스는 웹 브라우저에서 CICS 트랜잭션 처리 서비스로의 직접 액세스를 지원하는 트랜잭션 및 프로그램 콜렉션입니다.

CICS 웹 인터페이스가 사용할 HTML 템플리트를 생성한 경우, 단말기 서블릿에서 템플리트를 HTML 템플리트로 사용할 수도 있습니다. CICS 웹 인터페이스 대신 서블릿을 호출하기 위해 몇몇 사항을 변경해야 하지만, 형식은 기본적으로 같습니다.

템플리트 파일을 사용하려면 다음과 같이 수행하십시오.

1. 템플리트 파일을 웹 서버에 있는 디렉토리로 복사하십시오.

특성 및 매개변수 참조서

2. <FORM> 조치를 /servlet/TerminalServlet으로 변경하도록 편집하고, 이름이 "request"이고 값이 "send"인 <FORM> 구성요소를 추가하십시오.
3. HTML 템플릿이 참조하는 BMS 맵에 대해 맵 클래스와 화면 핸들러를 생성하십시오.
4. 생성된 Java 소스 파일을 컴파일합니다.
5. 컴파일된 클래스 파일을 웹 서버의 적합한 디렉토리로 복사하십시오.
6. 새로운 클래스를 로드할 수 있도록 서블릿 특성 **pool@handlerPath**를 설정하십시오.
7. 새로운 화면 핸들러 Bean을 HTML 템플릿 파일과 연관시키도록 페이지 맵핑 특성을 설정하십시오.

제8장 문제점 판별 및 문제점 해결

문제점을 조사하는 동안 문제점을 해결할 충분한 정보를 찾더라도 문제점 판별과 문제점 해결을 혼동해선 안됩니다. 발생할 수 있는 문제점 유형의 예제는 다음과 같습니다.

- 일반 사용자 오류
- 프로그래밍 오류
- 구성 오류

기본 점검

문제점을 조사하기 전에, 확실한 원인이 있는지 점검해야 합니다.

- 시스템이 이전에 제대로 실행되었습니까?
- 시스템 구성 또는 추가된 새로운 기능이나 프로그램에 변경사항이 있습니까?
- CLASSPATH 경로가 제대로 설정되었습니까?

CLASSPATH 설정에 대한 정보는 31 페이지의 『CICS Transaction Gateway 에 대한 프로그래밍 환경 구성』을 참조하십시오.

CICS Transaction Gateway 출력 맨위에 CLASSPATH 변수가 표시됩니다. 트 레이스를 사용 가능화하여 게이트웨이를 시작하는 데 대한 정보는 92 페이지의 『사용자가 지정한 선택항목으로 게이트웨이 시작』을 참조하십시오.

- CICS Universal Client 환경이 제대로 구성되었습니까?

다음 명령을 사용하십시오.

```
cicscli -s=servername
```

여기서 *servername*은 CICS 서버의 이름이며 그 뒤에 다음 명령을 사용하여

```
cicscli -l
```

클라이언트에서 서버에 연결할 수 있는지 여부를 알아보십시오.

- 실패를 설명하는 메시지가 있습니까?
- 실패를 다시 발생시킬 수 있습니까?

다음 수행 조치

문제가 CICS Transaction Gateway에 있는 것으로 생각되면 가능한 많은 정보를 수집하여 지원센터에 문의해야 합니다.

트레이스를 사용 가능으로 설정하지 않은 상태에서 게이트웨이를 시작한 경우 다음을 수행하십시오. 게이트웨이를 중지한 다음, 트레이스 선택항목을 지정하여 다시 시작하고, 문제를 다시 발생시키십시오.

문제가 CICS Universal Client 또는 다른 제품에 있는 것으로 판단되면 다른 제품에서 제공하는 문제점 판별 절차를 수행해야 합니다. CICS Universal Client에서의 문제점 판별에 대해서는 *CICS Transaction Gateway: 클라이언트 관리* 서적을 참조하십시오. CICS 서버의 경우, 해당 서버의 문제점 판별 서적을 참조하십시오.

메시지

CICS Transaction Gateway 메시지의 접미부는 *CCL*이며, 이는 보통 CICS 클라이언트에 사용됩니다.

CICS Transaction Gateway가 생성한 메시지 목록에 대해서는 *CICS Transaction Gateway: Gateway Messages* 책을 참조하십시오.

트레이스

게이트웨이 및 응용프로그램 트레이스에는 다음과 같은 세 가지 기본 레벨이 있습니다.

스택 트레이스 트레이스 입력항목은 Java 예외가 발생할 때만 작성됩니다. 이런 입력항목은 예외 원인을 판별하는데 도움을 줍니다. 성능을 관리하는 것이 중요한 경우에 이 레벨을 사용하십시오. 122 페이지의 그림9를 참조하십시오.

표준 트레이스 Java 예외, 기본 게이트웨이 기능 및 이벤트를 트레이스합니다. 기본적으로, 게이트웨이는 데이터 블

록의 첫 128바이트(예: COMMAREA 또는 네트 워크 플로우)만 표시합니다.

디버그 트레이스

Java 예외, 기본 게이트웨이 기능 및 이벤트를 스택 또는 표준 트레이스의 경우보다 더 자세히 트레이스합니다. 기본적으로, 게이트웨이는 트레이스의 모든 데이터 블록을 표시합니다. 성능이 중요하지 않을 경우 또는 표준 트레이스가 문제점을 해결할 만한 충분한 정보를 제공하지 않을 경우에만 이 기능을 사용하십시오.

게이트웨이 트레이스

CICS Transaction Gateway 프로세스 또는 게이트웨이를 호출하는 사용자 응용 프로그램에서 트레이스를 시작할 수 있습니다.

게이트웨이 트레이스를 사용 가능화하고 제어하려면 **ctgstart** 명령에 선택항목을 사용하십시오. 예를 들어, 표준 트레이스를 사용 가능화하려면 게이트웨이 시작시 다음 명령을 입력하십시오.

```
ctgstart -trace
```

구성 도구를 사용하여 게이트웨이를 구성한 경우, 트레이스 출력의 기본 목적지는 CICS Transaction Gateway가 설치되어 있는 bin 서브디렉토리의 CTG.TRC 파일입니다. 그 외의 경우, 트레이스 출력은 stderr에 기록됩니다. **ctgstart -tfile** 선택항목을 사용하여 트레이스 출력의 기본 목적지를 대체할 수 있습니다. 예를 들어, 다음 명령을 사용하십시오.

```
ctgstart -x -tfile pathname
```

이 명령은 디버그 트레이스를 사용 가능화하여 게이트웨이를 시작하고 *pathname* 에 지정된 파일에 트레이스 출력을 기록합니다.

성능 고려사항: 트레이스, 특히 디버그 트레이스는 성능을 크게 저하시킵니다. 트레이스로 인한 성능 저하를 줄이기 위해 CICS Transaction Gateway 버전 4.0은 다음과 같이 변경되었습니다.

- **-tfile** 선택항목을 사용하여 트레이스 출력 파일을 지정한 경우, 트레이스 출력은 stderr로 기록되지 않습니다.

다음 수행 조치

- 트레이스 출력 내 데이터 블록의 첫 행에만 시간소인이 지정됩니다.
- **-truncationsize** 및 **-dumpoffset** 선택항목을 사용하면 트레이스에 기록된 데이터 블록의 크기를 제한할 수 있습니다.
- **-tfilesize** 선택항목을 사용하면 트레이스 출력 파일의 크기를 제한할 수 있습니다.

트레이스 선택항목을 설정하고 CICS Transaction Gateway를 시작하는 데 대한 자세한 정보는 91 페이지의 『게이트웨이 시작』을 참조하십시오.

```
10:38:12:262 : main: S-C: java.io.IOException
10:38:12:263 : main: S-C: at com.ibm.gskssl.SSLSocketImpl.socketCreate(Native Method)
10:38:12:264 : main: S-C: at com.ibm.gskssl.SSLSocketImpl.create(SSLSocketImpl.java:133)
10:38:12:265 : main: S-C: at com.ibm.gskssl.SSLServerSocket.<init>(SSLServerSocket.java:109)
10:38:12:265 : main: S-C: at com.ibm.gskssl.SSLServerSocket.<init>(SSLServerSocket.java:73)
10:38:12:266 : main: S-C: at com.ibm.ctg.server.GskSslHandler.initialize(GskSslHandler.java:487)
10:38:12:266 : main: S-C: at com.ibm.ctg.server.JGate.main(JGate.java:941)
```

그림 9. 샘플 Java 스택 트레이스

응용프로그램 트레이스

다음과 같은 두 가지 방식으로 응용프로그램의 트레이스를 사용 가능화할 수 있습니다. JVM을 시작할 때 Java 지정문을 사용하거나 CICS Transaction Gateway 트레이스 API에 호출을 추가하십시오. **java** 명령에 **-D** 선택항목을 사용하여 Java 지정문을 지정하십시오. 트레이스 API는 CICS Transaction Gateway의 T 클래스에 있는 여러 static 메소드로 구성됩니다. 자세한 내용은 *CICS Transaction Gateway: Gateway Programming* 서적의 "Java 클라이언트 프로그램 트레이스"를 참조하십시오.

JNI 트레이스

게이트웨이 트레이스 선택항목은 JNI 트레이스를 사용 가능화하지 않습니다. 트레이스를 사용 가능화하려면 출력을 기록할 파일을 지시하도록 CTG_JNI_TRACE 환경 변수를 설정해야 합니다.

WebSphere의 서블릿 트레이스 및 로그

IBM WebSphere Application Server V3.5는 고급 관리 콘솔을 제공하여 응용 프로그램 관리 관련 타스크를 간소화하였습니다.

WebSphere에서 서블릿 트레이스를 사용 가능화하려면 다음과 같이 수행하십시오.

1. 선택한 파일로 표준 출력과 표준 오류를 보내도록 WebSphere를 구성하십시오.

관리 콘솔에서 트레이스를 사용 가능화할 응용프로그램 서버를 선택하십시오. 일반 탭 아래에는 표준 출력 및 표준 오류라는 레이블이 붙은 텍스트 입력 상자가 있습니다. 이 상자에 파일 이름을 입력하여 기본 목적지를 덮어쓸 수 있습니다.

2. 응용프로그램에서 트레이스를 사용 가능화하려면 다음과 같이 수행하십시오.

- CICS Transaction Gateway 단말기 서블릿의 경우, **servlet@debug** 특성을 사용하여 트레이스를 사용 가능화하십시오.

자세한 내용은 111 페이지의 『특성 및 매개변수 참조서』를 참조하십시오.

- 사용자가 작성한 응용프로그램의 경우, Java 지정문을 사용하여 트레이스를 사용 가능화하십시오. 관리 콘솔의 명령 행 인수 상자에 이 내용을 입력할 수 있습니다. Java 지정문에는 **java** 명령 행 호출과 동일한 형식을 사용하십시오. 예를 들어, 디버그 트레이스를 사용 가능화하려면 첫번째 명령 행 인수로 **-Dgateway.T=on**을 입력하십시오.

트레이스용 Java 지정문에 대한 자세한 내용은 *CICS Transaction Gateway: Gateway Programming* 서적의 "Java 클라이언트 프로그램 트레이스"를 참조하십시오.

3. 적용 단추를 눌러 WebSphere 설정을 갱신한 후 서블릿을 실행하십시오.

서블릿을 실행할 때 WebSphere는 표준 출력과 동일한 목적지에 로그 정보를 기록합니다. 이 정보에는 시작 시간, 중지 시간, 오류 메시지가 포함됩니다. CICS Transaction Gateway 단말기 서블릿의 경우, **servlet@extendedLogging** 특성을 사용하여 확장 로그 기록을 사용 가능화하십시오. 확장 로그 기록을 사용 가능화할 경우, 단말기 서블릿은 요청을 수신할 때마다 로그에 기록합니다. 확장 로그 기록은 문제점 진단을 위한 유용한 정보를 제공할 수 있지만, 그렇게 하면 성능이 저하됩니다.

단말기 서블릿이 실행되는 동안 트레이스와 확장 로그 기록을 모두 사용할 수 있습니다. 단말기 서블릿 특성에 대해서는 111 페이지의 『특성 및 매개변수 참조서』를 참조하십시오.

프로그램 지원

IBM 지원 센터에 문의하는 법에 대한 자세한 내용은 *CICS Transaction Gateway: 클라이언트 관리* 서적의 "프로그램 지원"을 참조하십시오.

자주 발생하는 문제점 판별

설치 문제점

설치 프로세스는 `uudecode` 기능을 사용합니다. 이 기능은 *UNIX-to-UNIX* 복사 프로그램의 일부입니다. CICS Transaction Gateway 설치를 시도하는 중 다음 메시지가 표시되는 경우:

```
ERROR: uudecode cannot be found in your PATH,  
please refer to the Gateway Administration book  
Problem Determination section for further information
```

프로그램이 시스템에 없는 것일 수도 있습니다. CICS Transaction Gateway를 설치하기 전에, 시스템에 프로그램이 있으며 PATH가 올바르게 구성되었는지 확인하십시오.

Java 예외

응용프로그램이나 CICS Transaction Gateway에서 예외를 처리할 수 없는 경우, JVM(Java Virtual Machine)은 Java 스택 덤프를 기록합니다. 덤프 출력의 목적지는 JVM 구현 방식에 따라 다릅니다. 자세한 내용은 Java 문서를 참조하십시오.

Java 스택 덤프에 기록되는 정보를 늘리려면 JIT(Just-In-Time) 컴파일러를 사용 불가능화하십시오. 이 정보에는 예외가 발생한 Java 소스 코드 행이 포함되어 있을 수 있습니다. JIT 컴파일러를 사용 불가능화하는 방법은 JVM 구현 방식에 따라 다릅니다. 자세한 내용은 Java 문서를 참조하십시오.

125 페이지의 그림10에서는 JIT 컴파일러가 사용 불가능화된 샘플 Java 스택 덤프를 보여줍니다.

```
Exception in thread "main" java.lang.OutOfMemoryError
  at java.lang.Thread.start(Native Method)
  at com.ibm.ctg.server.ThreadManager.createObject(ThreadManager.java:345)
  at com.ibm.ctg.server.ThreadManager.<init>(ThreadManager.java:131)
  at com.ibm.ctg.server.ManagedResources.<init>(ManagedResources.java:106)
  at com.ibm.ctg.server.JGate.main(JGate.java:895)
```

그림 10. 샘플 Java 스택 덤프

CICS Transaction Gateway에서 예외를 처리할 경우, Java 스택 덤프는 트레이스가 사용 가능한 때에만 기록됩니다. 트레이스 사용 가능 상태에서 문제점을 다시 발생시켜 보십시오. 그러면 예외가 발생하기 전에 어떤 일이 발생하는지 알 수 있습니다. 트레이스에 대한 자세한 정보는 120 페이지의 『트레이스』를 참조하십시오.

게이트웨이가 Windows NT[®] 서비스로 실행될 때의 문제점

게이트웨이가 Windows NT[®] 서비스로 실행될 때 시작되지 않거나 연결될 수 없는 경우, 콘솔 입력이 사용 가능화되어 있기 때문일 수 있습니다. 그림 11과 같은 입력항목은 게이트웨이가 콘솔로부터 읽기를 시도함을 나타냅니다.

```
11:37:58:472 : main: S-C: -> [ServerMessages: getKeys] ()
11:37:58:472 : main: S-C: <- [getKeys] = Q-
11:37:58:482 : main: S-C: java.io.IOException: The handle is invalid
11:37:58:482 : main: S-C: at java.io.FileInputStream.readBytes(Native Method)
11:37:58:482 : main: S-C: at java.io.FileInputStream.read(FileInputStream.java:190)
11:37:58:482 : main: S-C: at java.io.BufferedInputStream.read1(BufferedInputStream.java:229)
11:37:58:482 : main: S-C: at java.io.BufferedInputStream.read(BufferedInputStream.java:286)
11:37:58:482 : main: S-C: at java.io.FilterInputStream.read(FilterInputStream.java:99)
11:37:58:482 : main: S-C: at java.io.InputStreamReader.fill(InputStreamReader.java:180)
11:37:58:482 : main: S-C: at java.io.InputStreamReader.read(InputStreamReader.java:256)
11:37:58:482 : main: S-C: at java.io.BufferedReader.fill(BufferedReader.java:145)
11:37:58:482 : main: S-C: at java.io.BufferedReader.readLine(BufferedReader.java(Compiled Code))
11:37:58:482 : main: S-C: at java.io.BufferedReader.readLine(BufferedReader.java:368)
11:37:58:482 : main: S-C: at com.ibm.ctg.server.JGate.main(JGate.java:940)
11:37:58:482 : main: S-C: CCL6559E: Unexpected exception occurred. [java.io.IOException: The handle is invalid]
```

그림 11. Windows NT[®] 서비스로 실행 중인 게이트웨이의 트레이스

콘솔 입력 읽기를 사용 불가능화하려면 다음과 같이 수행하십시오.

- 구성 도구의 일반 게이트웨이 설정 패널의 ‘콘솔로부터 입력 읽기 사용 가능’ 상자에 체크 표시가 되어 있는지 확인하십시오. 32 페이지의 『구성 도구 사용』을 참조하십시오.
- **ctgservice** 명령에 **-A** 선택항목을 사용하여 **-noinput** 선택항목을 **ctgstart** 명령으로 전달하십시오.

java.lang.OutOfMemory 예외

CICS Transaction Gateway를 오랫동안 실행한 후 이 현상이 발생하면 메모리 유출로 인한 것일 수 있습니다. 게이트웨이가 과부화될 때에만 이런 현상이 발생하면 사용 가능 JVM(Java Virtual Machine) 메모리에 너무 많은 쓰레드가 활성화되었기 때문일 수 있습니다.

메모리 유출

게이트웨이 프로세스에 대해 메모리 사용 모니터를 실행하십시오. 메모리 유출이 있는 경우, 시간이 지남에 따라 게이트웨이에서 사용하는 메모리의 양이 늘어납니다. CICS Transaction Gateway에 연결하는 모든 사용자 응용프로그램에서 JavaGateway() 연결 객체를 사용하고 난 후 연결을 닫았는지 확인하십시오. 그렇게 해도 문제점이 해결되지 않으면 게이트웨이 디버그 트레이스를 실행하고 IBM 지원센터에 문의하십시오. **-tfilesize** 선택항목을 사용하면 트레이스 출력 파일의 크기를 제한할 수 있습니다. 또한 **ctgstart -truncationsize** 선택항목을 사용하면 트레이스에 표시할 데이터 블록의 크기를 0으로 줄일 수 있습니다. 그러면 성능 저하를 줄일 수 있습니다.

쓰레드가 너무 많음

다음과 같이 JVM에서 요구하는 가상 메모리의 양을 추정할 수 있습니다. 다음을 추가하십시오.

- 연결 매니저 쓰레드의 수
- Worker 쓰레드의 수
- Java가 사용하는 백그라운드 쓰레드의 수
- 사용 중인 서로 다른 네트워크 프로토콜의 수

그 다음 JVM이 각 쓰레드에 할당한 Java 스택의 크기에 이 총계를 곱하십시오.

이 계산에서, 활성 백그라운드 쓰레드의 수가 대략 5가 된다고 가정합니다. 더 정확한 값은 Java 문서를 참조하십시오. CICS Transaction Gateway의 구성에서 최대 구성 매니저 쓰레드와 Worker 쓰레드 수를 설정하십시오. 자세한 내용은 36 페이지의 『일반 게이트웨이 설정값』을 참조하십시오.

Java가 메모리를 할당하는 방식은 JVM 구현 방식에 따라 다릅니다. 대부분의 JVM에서는 Java 스택 및 Java 힙 크기 한계를 설정하여 메모리 할당을 제어할 수 있습니다.

쓰레드 한계에 대한 자세한 정보는 13 페이지의 표1을 참조하십시오. Java 메모리 할당에 대한 자세한 내용은 *CICS Transaction Gateway: Gateway Programming* 서적 및 Java 문서를 참조하십시오.

OS/390® 및 UNIX® 시스템에서는 각 프로세스에서 사용 가능한 메모리 양이 제한될 수 있습니다. 자세한 정보는 운영 체제 문서를 참조하십시오.

프로세스가 잠긴 경우

모든 운영 체제

잠금 발생 위치를 판별하려면 응용프로그램, 게이트웨이, 클라이언트에서 트레이스를 사용 가능화하십시오. 게이트웨이 및 응용프로그램 트레이스에 대한 자세한 정보는 120 페이지의 『트레이스』를 참조하십시오. 클라이언트 트레이스에 대한 내용 및 잠금이 클라이언트에서 발생했는지 서버에서 발생했는지 여부를 판별하는 법에 대해서는 *CICS Transaction Gateway: 클라이언트 관리* 서적을 참조하십시오. 잠금이 게이트웨이에서 발생한 경우, IBM 지원센터에 문의하여 게이트웨이 트레이스를 제시하십시오.

AIX

일부 JVM(Java Virtual Machine)에서 Java가 스택 덤프를 기록하고 현재 쓰레드 상태를 보여주도록 할 수 있습니다. 예를 들어, IBM Java SDK 1.3에서 **SIGQUIT (-3)** 신호를 Java 프로세스로 보내 스택 덤프를 stderr에 기록하도록 할 수 있습니다. 그러면 현재 쓰레드의 상태가 표시됩니다. 이 작업은 작업 중인 현업 시스템에서 수행하지 말고 완전히 잠긴 시스템에서만 수행하십시오.

코드 페이지 문제점

게이트웨이 트레이스는 JVM이 실행 중인 코드 페이지를 보여줍니다. 게이트웨이 트레이스는 트레이스 맨위의 시스템 등록 정보 섹션에 표시됩니다. 클라이언트에서 서버에 보낸 코드 페이지를 찾는 데 대한 정보는 *CICS Transaction Gateway: 클라이언트 관리* 서적을 참조하십시오.

SSL 예외

일반

CICS Transaction Gateway에서 스택 트레이스를 사용 가능화하십시오. 그러면 예외가 발생하기 전에 어떤 일이 발생하는지 알 수 있습니다. CLASSPATH 변수와 같은, 구성 정보가 제공됩니다. 그래도 문제점을 진단하기 위한 충분한 정보를 얻지 못할 경우, 표준 트레이스를 실행하고 IBM 지원센터에 문의하십시오.

CICS Transaction Gateway를 시작할 때

게이트웨이를 시작할 때 다음과 같은 메시지가 발행됩니다.

```
CCL6525W: ssl1 프로토콜에 대한 핸들러를 시작할 수 없습니다.  
[java.lang.ClassNotFoundException:  
server KeyRing class name]
```

이 메시지는 게이트웨이에서 KeyRing 클래스를 로드할 수 없음을 나타냅니다. 트레이스 사용 가능 상태에서 게이트웨이를 시작하십시오. 트레이스 출력 맨위에 표시되는 CLASSPATH 변수에 클래스가 있는 디렉토리에 대한 입력항목이 있는지 확인하십시오. 또한 CLASSPATH에 cfwk.zip 파일에 대한 입력항목이 있는지 확인하십시오. 게이트웨이에서 이 파일을 찾지 못하면 KeyRing 파일을 로드할 수 없습니다.

구성 도구의 SSL 프로토콜 설정 패널에 있는 **KeyRing** 클래스 이름 필드에 서버 KeyRing 클래스 이름을 지정하십시오. KeyRing 클래스 이름을 지정할 때는 디렉토리 또는 파일 확장자를 넣지 마십시오. CLASSPATH에는 KeyRing 클래스가 있는 디렉토리에 대한 입력항목이 있어야 합니다.

사용자 응용프로그램에서

사용자 응용프로그램에서 API 호출 SslJavaGateway.setKeyRing("ClientKeyRingName", "thePassword");가 다음 오류로 실패합니다.

```
java.lang.NoClassDefFoundError: com/ibm/sslight/SSLightKeyRing
```

CLASSPATH 변수에 cfwk.zip 파일에 대한 입력항목이 있는지 확인하십시오. 이 파일은 CICS Transaction Gateway와 함께 제공됩니다.

JDK™ AppletViewer 문제점

AppletViewer를 사용하여 국지 파일 시스템에서 CICS Transaction Gateway 예제 애플릿 중 몇몇을 수행하면

CCL6664E *protocol*을 지원하기 위한 관련 클래스를 로드할 수 없습니다.

메시지가 표시됩니다.

이 문제점을 해결하기 위해서는 AppletViewer에서 애플릿을 선택한 다음, 등록 정보를 선택한 후 네트워크 액세스와 클래스 액세스를 모두 무제한으로 설정하십시오.

부록A. CICS Universal Clients 자료 변환

ECI 및 EPI는 클라이언트 시스템에서 실행 중인 비 CICS 응용프로그램이 CICS 기능과 CICS 서버 시스템에서 관리하는 데이터로 액세스할 수 있도록 합니다.

문자 데이터가 클라이언트와 서버 사이에 전달될 때 이 데이터를 변환해야 할 수도 있습니다. 예를 들어, CICS Universal Client 시스템에서는 데이터가 ASCII로 인코드되고 CICS/390 서버 시스템에서는 EBCDIC로 인코드됩니다. 자료 변환은 서버 시스템에서 수행됩니다.

가능 ASCII 및 EBCDIC 인코드 설계에 대해서는 SC09-2190 CRDA(Character Data Representation Architecture Reference and Registry)에서 자세히 설명합니다. 간단히 말해, 각 인코드 설계는 CCSID(Coded Character Set Identifier) 및 CPGID(Code Page Global Identifier)로 식별됩니다. 여기서 CCSID는 그래픽 문자 세트를 정의하며, CPGID는 그래픽 문자를 나타내는 데 사용되는 코드 포인트를 지정합니다.

서버 시스템에서 관리하는 데이터는 각기 다양한 ASCII 인코드 설계를 사용하는 여러 클라이언트 시스템에서 액세스할 수 있습니다. 그러한 액세스를 지원하려면 각 클라이언트 시스템은 데이터가 올바르게 변환될 수 있도록 CCSID '태그'를 제공할 수 있어야 합니다.

지원되는 변환

자료 변환에 사용되는 방법은 서버 플랫폼에 따라 다릅니다. 지원되는 자료 변환 범위 또한 플랫폼에 따라 다릅니다. 다음 표는 *Communicating from CICS on System/390*, SC33-1697 ASCII에서 발췌한 것이며, 지리 그룹 또는 언어 그룹에 EBCDIC CCSID가 지정됩니다.

ASCII와 EBCDIC 사이의 자료 변환이 지원됩니다. 여기서 두 CCSID 모두 동일한 그룹에 속합니다. 그 목적은 다른 CICS 서버에서 동일한 지원을 제공하도록 하는 데 있습니다.

자료 변환

아랍:

	CCSID	CPGID	
ASCII	00864	00864	PC data: Arabic
	01089	01089	ISO 8859-6: Arabic
	01256	01256	MS Windows: Arabic
EBCDIC	00420	00420	Host: Arabic

발트어:

	CCSID	CPGID	
ASCII	00921	00921	PC data: Latvia, Lithuania
	01257	01257	MS Windows: Baltic Rim
EBCDIC	01112	01112	Host: Latvia, Lithuania

키릴:

	CCSID	CPGID	
ASCII	00866	00866	PC data: Cyrillic
	00915	00915	ISO 8859-5: Cyrillic
	01251	01251	MS Windows: Cyrillic
EBCDIC	01025	01025	Host: Cyrillic multilingual

에스토니아:

	CCSID	CPGID	
ASCII	00922	00922	PC data: Estonia
	01257	01257	MS Windows: Baltic Rim
EBCDIC	01122	01122	Host: Estonia

그리스:

	CCSID	CPGID	
ASCII	00869	00869	PC data: Greece
	00813	00813	ISO 8859-7: Grek
	01253	01253	MS Windows: Greek
EBCDIC	00875	00875	Host: Grek

헤브루어:

	CCSID	CPGID	
ASCII	00856	00856	PC data: Hebrew
	00916	00916	ISO 8859-8: Hebrew
	01255	01255	MS Windows: Hebrew
EBCDIC	00424	00424	Host: Hebrew

라틴-1:

	CCSID	CPGID	
ASCII	00437	00437	PC data: USA, many other countries
	00819	00819	ISO 8859-1: Latin-1 countries
	00850	00850	PC data: Latin-1 countries
EBCDIC	01252	01252	MS Windows: Latin-1 countries
	00037	00037	Host: USA, Canada, etc
	00273	00273	Host: Austria, Germany
	00277	00277	Host: Denmark, Norway
	00278	00278	Host: Finland, Sweden
	00280	00280	Host: Italy
	00284	00284	Host: Spain, Latin America (Spanish)
	00285	00285	Host: United Kingdom
	00297	00297	Host: France
	00500	00500	Host: International Latin-1
	00871	00871	Host: Iceland
01047	01047	Host: Latin-1 Open Systems	

Euro 지원 포함 라틴-1

	CCSID	CPGID	
ASCII	00858	00858	PC data: Latin-1 countries
	05348	01252	MS Windows: Latin-1 countries, version 2
EBCDIC	01140	01140	Host: USA, Canada, etc
	01141	01141	Host: Austria, Germany
	01142	01142	Host: Denmark, Norway
	01143	01143	Host: Finland, Sweden
	01144	01144	Host: Italy
	01145	01145	Host: Spain, Latin America (Spanish)
	01146	01146	Host: United Kingdom
	01147	01147	Host: France
	01148	01148	Host: International Latin-1
	01149	01149	Host: Iceland

라틴-2:

	CCSID	CPGID	
ASCII	00852	00852	PC data: Latin-2 multilingual
	00912	00912	ISO 8859-2: Latin-2 multilingual
	01250	01250	MS windows: Latin-2
EBCDIC	00870	00870	Host: Latin-2 multilingual

라틴-5:

	CCSID	CPGID	
ASCII	00857	00857	PC data: Latin-5 (Turkey)
	00920	00920	ISO 8859-9: Latin-5 (Turkey)
	01254	01254	MS Windows: Turkey
EBCDIC	01026	01026	Host: Latin-5 (Turkey)

자료 변환

라틴-9:

	CCSID	CPGID	
ASCII	00923	00923	ISO 8859-15: Latin-9
EBCDIC	00924	00924	Host: Latin-9

일어:

	CCSID	CPGID	
ASCII	00932	00897	PC data: SBCS
		00301	PC data: DBCS
	00942	01041	PC data: extended SBCS
		00301	PC data: DBCS
	00943	00897	PC data: SBCS
00954	00941	PC data: DBCS for Open environment	
	00895	G0: JIS X201 Roman	
	00952	G1: JIS X208-1990	
	00896	G2: JIS X201 Katakana	
	00953	G3: JIS X212	
EBCDIC	00930	00290	Host: Katakana, extended SBCS
		00300	Host: DBCS
	00931	00037	Host: Latin-1, SBCS
		00300	Host: DBCS
00939	01027	Host: Latin-1, extended SBCS	
	00300	Host: DBCS	

한국어:

	CCSID	CPGID	
ASCII	00934	00891	PC data: SBCS
		00926	PC data: DBCS
	00944	01040	PC data: extended SBCS
		00926	PC data: DBCS
	00949	01088	PC data: SBCS, IBM KS code
00970	00951	PC data: DBCS, IBM KS code	
	00367	G0: ASCII	
	00971	G1: KSC X5601-1989	
01363	01126	MS Windows: Korean SBCS	
	01362	MS Windows: Korean DBCS	
EBCDIC	00933	00833	Host: extended SBCS
		00834	Host: DBCS

중국어:

	CCSID	CPGID	
ASCII	00946	01042	PC data: extended SBCS
		00928	PC data: DBCS
	01381	01115	PC data: extended SBCS, IBM GB

		01380	PC data: DBCS, IBMGB
	01383	00367	G0: ASCII
		01382	G1: GB 2312-80 set
	01386	01114	PC data: SBCS, S-Chinese GBK, T-Chinese IBM BIG-5
EBCDIC	00935	01385	PC data: DBCS, S-Chinese GBK
		00836	Host: extended SBCS
		00837	Host: DBCS

대만어:

	CCSID	CPGID	
ASCII	00938	00904	PC data: SBCS
		00927	PC data: DBCS
	00948	01043	PC data: extended SBCS
		00927	PC data: DBCS
	00950	01114	PC data: SBCS, IBM BIG-5
		00947	PC data: DBCS
	00964	00367	G0: ASCII
		00960	G1: CNS 11643 plane 1
		00961	G2: CNS 11643 plane 2
EBCDIC	00937	00037	Host: Latin-1 SBCS
		00835	Host: DBCS

부록B. CICS Transaction Gateway 및 CICS Universal Clients 라이브러리

이 장에서는 CICS Transaction Gateway, CICS Universal Clients 및 관련 책을 나열하고, 이들을 사용할 수 있는 다양한 양식에 대해 논의합니다.

장의 제목은 다음과 같습니다.

- 『CICS Transaction Gateway 책』
- 138 페이지의 『CICS Universal Clients 책』
- 139 페이지의 『CICS 관련 책』
- 140 페이지의 『책 파일명』
- 140 페이지의 『예제 구성 문서』
- 141 페이지의 『기타 책』
- 141 페이지의 『온라인 문서 열람』

CICS Transaction Gateway 책

- *CICS Transaction Gateway: Windows[®] 게이트웨이 관리, SA30-1342*
이 책에서는 Windows[®]용 CICS[®] Transaction Gateway 관리에 대해 설명합니다.
- *CICS Transaction Gateway: AIX[®] 게이트웨이 관리, SA30-1343*
이 책에서는 AIX[®]용 CICS[®] Transaction Gateway 관리에 대해 설명합니다.
- *CICS Transaction Gateway: Solaris Gateway Administration, SC34-5934*
이 책에서는 Solaris용 CICS[®] Transaction Gateway 관리에 대해 설명합니다.
- *CICS Transaction Gateway: Linux 게이트웨이 관리, SA30-1344*
이 책에서는 Linux용 CICS[®] Transaction Gateway 관리에 대해 설명합니다.
- *CICS Transaction Gateway: HP-UX Gateway Administration, SC34-5936*
이 책에서는 HP-UX용 CICS[®] Transaction Gateway 관리에 대해 설명합니다.

CICS Transaction Gateway 및 CICS Universal Clients 라이브러리

- *CICS Transaction Gateway: OS/390[®] Gateway Administration, SC34-5935*
이 책에서는 OS/390[®]용 CICS[®] Transaction Gateway 관리에 대해 설명합니다.
- *CICS Transaction Gateway: Gateway Messages*
이 온라인 책은 CICS Transaction Gateway가 생성할 수 있는 오류 메시지를 나열하고 설명합니다.
이 책은 주문할 수 없습니다.
- *CICS Transaction Gateway: Gateway Programming, SC34-5938*
이 책은 CICS Transaction Gateway 내에서의 Java[™] 프로그램에 대해 소개합니다.
프로그래밍 참조 정보가 있는 HTML 페이지가 추가로 있습니다.

CICS Universal Clients 책

- *CICS Transaction Gateway: Windows[®] 클라이언트 관리, SA30-1339*
이 책에서는 Windows용 CICS Universal Client 관리에 대해 설명합니다.
- *CICS Transaction Gateway: AIX[®] 클라이언트 관리, SA30-1340*
이 책에서는 AIX용 CICS Universal Client 관리에 대해 설명합니다.
- *CICS Transaction Gateway: Solaris Client Administration, SC34-5942*
이 책에서는 Solaris용 CICS Universal Client 관리에 대해 설명합니다.
- *CICS Transaction Gateway: Linux 클라이언트 관리, SA30-1341*
이 책에서는 Linux용 CICS Universal Client 관리에 대해 설명합니다.
- *CICS Transaction Gateway: HP-UX Client Administration, SC34-5943*
이 책에서는 HP-UX용 CICS Universal Client 관리에 대해 설명합니다.
- *CICS Transaction Gateway: Client Messages*
이 온라인 책은 CICS Universal Clients가 생성할 수 있는 오류 및 트레이스 메시지를 나열하고 설명합니다.
이 책은 주문할 수 없습니다.
- *CICS Transaction Gateway: C++ Programming, SC34-5945*

CICS Transaction Gateway 및 CICS Universal Clients 라이브러리

이 책은 ECI와 EPI에 대한 객체 지향 프로그램을 C++ 언어로 작성하는 방법에 대해 설명합니다.

- *CICS Transaction Gateway COM Automation Programming, SC34-5946*
이 책에서는 COM(Component Object Model) 표준에 따라 ECI 및 EPI에 대한 객체 지향 프로그램을 작성하는 방법에 대해 설명합니다.

CICS 관련 책

- *CICS® Family: Client/Server Programming, SC34-5947*
이 책은 CICS 클라이언트/서버 프로그래밍인 ECI(External Call Interface), EPI(External Presentation Interface) 및 ESI(External Security Interface)와 연관된 프로그래밍 인터페이스에 대해 설명합니다. 클라이언트 응용프로그램을 개발하여 CICS 서버 시스템과 통신하려는 응용프로그램 디자이너와 프로그래머를 위한 책입니다.

책 파일명

표5에서는 CICS Transaction Gateway 및 CICS Universal Client의 소프트카피 파일명을 보여줍니다.

표 5. CICS Transaction Gateway 및 CICS Universal Clients 책과 파일명

책 제목	파일명
CICS Transaction Gateway: Client Messages	CCLIAB
CICS Transaction Gateway: AIX [®] 클라이언트 관리	CCLIAD
CICS Transaction Gateway: Windows [®] 클라이언트 관리	CCLIAF
CICS Transaction Gateway: Solaris Client Administration	CCLIAG
CICS Transaction Gateway: Linux 클라이언트 관리	CCLIAR
CICS Transaction Gateway: HP-UX Client Administration	CCLIAT
CICS Transaction Gateway: OS/390 [®] Gateway Administration	CCLIAI
CICS Transaction Gateway: Gateway Messages	CCLIAJ
CICS Transaction Gateway: Gateway Programming	CCLIAK
CICS Transaction Gateway: Windows [®] 게이트웨이 관리	CCLIAL
CICS Transaction Gateway: AIX [®] 게이트웨이 관리	CCLIAN
CICS Transaction Gateway: Solaris Gateway Administration	CCLIAO
CICS Transaction Gateway: Linux 게이트웨이 관리	CCLIAS
CICS Transaction Gateway: HP-UX Gateway Administration	CCLIAU
CICS Transaction Gateway: C++ Programming	CCLIAP
CICS Transaction Gateway COM Automation Programming	CCLIAQ
CICS [®] Family: Client/Server Programming	DFHZAD
주: 이 표의 파일명에는 2자리 접미부가 붙지 않았습니다.	

예제 구성 문서

많은 수의 예제 구성 문서들은 PDF(Portable Document Format) 형식으로 사용될 수 있습니다.

이들 문서들은 예를 들어 여러 프로토콜을 사용하여 CICS 서버와 통신하도록 CICS Universal Client를 구성하는 데 도움이 되는 단계별 안내를 제공합니다. 이들은 CICS Transaction Gateway 및 CICS Universal Client 라이브러리에 있는 정보를 확장하는 상세한 명령문도 제공합니다.

좀 더 많은 예제 구성 문서를 사용할 수 있게 되므로, 웹 사이트에서 이를 다운로드하고, 다음으로 이동한 다음, **Library** 링크를 따르십시오.

www.ibm.com/software/ts/cics/

기타 책

다음 ITSO(International Technical Support Organization) Redbook™에는 클라이언트/서버 구성에 대한 많은 예제가 들어 있습니다.

- *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked, SG24-5277*

이 책은 다음 책을 대신합니다.

- *CICS Clients Unmasked, GG24-2534*

많은 소스로부터 ITSO 레드 북을 구할 수 있습니다. 최신 정보에 대해서는 다음을 참조하십시오.

www.ibm.com/redbooks/

다음에서 CICS 제품에 대한 정보를 찾을 수 있습니다.

www.ibm.com/software/ts/cics/

온라인 문서 열람

온라인 라이브러리에서 CICS Transaction Gateway 및 CICS Universal Client가 제공하는 모든 문서에 액세스할 수 있습니다(영어 전용). 온라인 라이브러리를 사용하려면 Adobe Acrobat Reader 및 적합한 웹 브라우저가 필요합니다(이들을 구성하는 데 필요할 수 있습니다).

온라인 라이브러리를 보려면 **ctgdoc** 스크립트를 실행하십시오. 라이브러리 홈 페이지가 표시됩니다.

온라인 라이브러리를 사용하면 다음으로 링크할 수 있습니다.

- PDF 형식으로 된 CICS Transaction Gateway 및 CICS Universal Clients 책.
- HTML(HyperText Markup Language) 파일에 있는 프로그래밍 참조 문서(CICS Transaction Gateway에 대해서만 제공됨).

온라인 문서 열람

- README 파일.
- PDF 형식으로 된 예제 구성 문서.
- CICS 웹 사이트.

Acrobat Reader 사용에 관한 안내 정보도 제공됩니다.

때때로 갱신된 버전의 책이 제공될 수 있습니다. 다음 주소의 웹 사이트를 확인한 후 **Library** 링크를 따르십시오.

www.ibm.com/software/ts/cics/

주: 일부 서적은 다른 언어로 번역됩니다. 이러한 서적은 온라인 라이브러리에 포함되지 않으며, 위의 웹 사이트에서 독립형 PDF 파일로 사용할 수 있습니다.

PDF 책 열람

PDF 정보는 다음을 위한 강력한 기능을 제공합니다.

- 정보의 선택진행. PDF 문서내에는 기타 PDF 문서 및 웹 페이지에 대한 하이퍼텍스트 링크가 있습니다.
- 특정 정보 탐색.
- 포스트스크립트 프린터에서 PDF 문서 일부 또는 전부 인쇄.

다음 Adobe 웹 사이트에서 Acrobat Reader에 대해 좀 더 많은 내용을 찾을 수 있습니다.

www.adobe.com/acrobat/

부록C. 주의사항

이 책은 미국에서 제공되는 제품 및 서비스를 위해 개발되었습니다. IBM이 그 밖의 국가에서는 이 책에 설명된 제품, 서버스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용 가능한 제품 및 서비스에 대한 정보는 한국 IBM 영업대표에 문의하십시오. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 반드시 IBM 제품, 프로그램 또는 서비스만을 사용하라는 의미는 아닙니다. IBM의 지적 재산권을 침해하지 않는 한, 타사의 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스 운용에 대한 평가 및 확인은 사용자의 책임입니다.

IBM에는 이 책에 설명된 주제를 다룰 독점권 또는 보류 중인 독점 응용프로그램이 있을 수도 있습니다. 이 책을 구입한다고 해서 이러한 독점권에 대한 사용권이 부여되지는 않습니다. 사용권에 대한 의문사항은 다음으로 문의하시기 바랍니다.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2 바이트(DBCS) 정보에 관한 사용권 의문사항은 한국 IBM 고객만족센터 또는 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

다음 단락은 그러한 제안이 지역 법규와 일치하지 않는 영국 또는 그 밖의 국가에는 적용되지 않습니다. IBM은 제삼자의 권리를 침해하지 않는다는 보증, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않

음) 어떠한 종류의 묵시적 또는 명시적 보증없이 “현상태대로” 이 책을 제공합니다. 일부 국가에서는 특정 거래에 있어서 명시적 또는 묵시적 보증의 포기를 허용하지 않으므로 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 책의 정보는 주기적으로 변경되며, 책의 새로운 개정판에 변경사항이 반영됩니다. IBM은 이 책에 설명된 제품(들) 및/또는 프로그램(들)을 언제든지 통지하지 않고 사전 통지 없이 언제라도 이 책에서 설명하는 제품 및 프로그램을 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM 웹 사이트는 편의상 제공된 것으로 이들 웹 사이트를 옹호하기 위한 것은 절대 아닙니다. 이들 웹 사이트에 나오는 자료들은 이 IBM 제품에 대한 자료의 일부가 아니며 이들 웹 사이트의 사용은 사용자 본인의 책임입니다.

IBM은 적당하다고 생각하는 방식으로 사용자가 제공하는 정보를 사용하거나 배포할 수 있으며 사용자는 이에 대해 책임을 지지 않습니다.

(i) 독자적으로 작성된 프로그램과 다른 프로그램(이 프로그램을 포함) 사이의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 정보를 필요로 하는 이 프로그램의 사용권자는 다음으로 문의해야 합니다.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조항 및 조건(예를 들면, 사용료 지불 등)에 따라 사용할 수 있습니다.

IBM은 이 책에 설명된 사용 허가된 프로그램과 이에 대한 모든 사용 허가된 자료를 IBM Customer Agreement, IBM International Programming License Agreement 또는 모든 동등한 쌍방간의 계약 조건하에 제공합니다.

비IBM 제품에 대한 정보는 이들 제품의 공급업체, 해당 업체의 공개자료 또는 기타 범용 자료로부터 입수한 것입니다. IBM은 IBM 제품과 관련한 배상 청구, 호

환성 또는 성능의 정확성 등에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대해서는 이들 제품의 공급업체에 문의하십시오.

상표

다음 용어는 미국과 기타 국가에서 사용되는 IBM Corporation의 상표입니다.

AIX	Anynet
CICS	eNetwork
IBM	MQSeries
MVS	MVS/ESA
OS/390	OS/400
RETAIN	TXSeries
VisualAge	VSE/ESA
VTAM	WebSphere

Lotus Domino, Lotus Notes 및 Domino Go Webserver는 Lotus Development Corporation의 상표입니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 및 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Java 및 Java 관련 등록상표와 로고는 미국 및 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표 또는 등록상표입니다.

UNIX는 미국과 기타 국가에서 X/Open Company Limited를 통해 독점적으로 사용권을 부여받은 등록상표입니다.

기타 회사, 제품 및 서비스 이름은 기타 회사의 상표 또는 서비스표입니다.

색인

[가]

개발 도구 25
게이트웨이 중지 94
게이트웨이 트레이스 121
게이트웨이 트레이스 랩 크기 53
게이트웨이 트레이스 파일 구성 설정 53
계획 21
고유 이름 71
공용 키 암호작성법 71
구문 표현 xiv
구성 31
 구성 도구 32
 구성 파일 32
 웹 서버 31
 프로그래밍 환경 31
 CLASSPATH 31
 Gateway.properties 파일 32
구성 도구 32
구성 변환 도구 54
구성 설정
 게이트웨이 트레이스 랩 크기 53
 게이트웨이 트레이스 파일 53
 국지 LU 이름 또는 템플리트 48
 단말기 나감 43
 대문자 보안 사용 46
 로그 파일 45
 모델 단말기 정의 46
 모드 이름 49
 상대 LU 이름 48
 서버 이름 45
 서버 재시도 간격 45
 설명 45
 연결 매니저 쓰레드의 초기 수 36
 연결 매니저 쓰레드의 최대 수 36
 연결 시간 초과 39, 47

구성 설정 (계속)

완전한 CP 이름 또는 템플리트 49
원격 노드 비활동 폴 간격 51
휴식 시간 종료 39
인쇄 명령 43
인쇄 파일 44
작동 중인 연결 제거 40
진행 중 요청 시간 초과 38
초기 트랜잭션 46
최대 논리 SNA 세션 수 51
최대 버퍼 크기 42
최대 서버 43
최대 요청 수 43
최대 클라이언트 랩 크기 54
최대 SNA RU 크기 51
코드 페이지 ID 대체 44
콘솔의 입력 읽기 기능 37
클라이언트 인증 사용 41
클라이언트 트레이스 파일 54
클라이언트가 보안 클래스를 사용하도록
 록 함 40
키링 암호 41
키링 클래스명 41
트레이스 52
포트 39, 47
프로토콜 핸들러 사용 가능 39
핸들러 웨이크업 시간 종료 39
호스트명 또는 IP 주소 47
Anynet 도메인 이름 접미부 50
Applid 42
CP 이름의 IP 주소 마스크(선택항
 목) 50
Java 클라이언트에서 일반 ECI 응답
 수신 38

구성 설정 (계속)

LU 이름 템플리트의 IP 주소 마스크
 (선택항목) 49
Ping 시간 빈도 40
SNA 페이징 크기 51
SO_LINGER 설정 40
TCP/IP Keepalive 패킷 송신 47
TCP/IP 호스트명 표시 37
Worker 쓰레드 사용 시간 종료 38
Worker 쓰레드의 초기 수 37
Worker 쓰레드의 최대 수 37
구성 파일 32
 이름 바꾸기 58
국지 CICS Transaction Gateway 19,
 67
국지 LU 이름 또는 템플리트 구성 설정
 48

[나]

네트워크 컴퓨터 8

[다]

단말기 나감 구성 설정 43
단축키 20
대문자 보안 사용 구성 설정 46
대칭 키 71
도구

 기타 25

 응용프로그램 개발 25

동적 이름 생성(TCP62) 49

디지털 사인 71

디지털 인증 17, 71

[라]

레이로드 다이어그램 xiv

로그 파일 구성 설정 45

로드 균형화 19

[마]

메모리 요구사항 43

메모리 유출 126

메시지 120

모델 단말기 정의 구성 설정 46

모드 이름 구성 설정 49

문서 137

HTML 141

PDF 142

문제점 판별 119

게이트웨이 트레이스 121

기본 점검 119

메시지 120

응용프로그램 트레이스 122

자주 발생하는 문제점 판별 124

트레이스 120

프로그램 지원 124

CICS Universal Client 문제점 120

JNI 트레이스 122

WebSphere의 서블릿 트레이스 및 로그 기록 122

문제점, 판별 124

[바]

방화벽 7

변수, 환경 58

보안 46

개요 69

고유 이름 71

공용 키 암호작성법 71

권한 16

디지털 사인 71

디지털 인증 17, 71

디지털 인증 유지 보수 76

보안 종료 19

사인자 인증서 72

서버 인증 74

보안 46 (계속)

신뢰 루트 키 72

암호화 17, 70

외부 사인 인증 18, 78

인증 18, 74

인증 기관(CA) 71

인증 포함 84

자가 사인 인증 18

자가 사인 CA 인증 84

클라이언트 워크스테이션으로

iKeyman 분산 77

클라이언트 인증 74

키 71

키링 파일 73

키링s 18, 73

키인증 저장소 87

ctgikey 76

ESI(External Security

Interface) 16

HTTPS 18, 88

SSL 88

SSL 핸드셰이크 75

SSL(Secure Sockets Layer) 17

보안 종료 19

브라우저 22

비대칭 키 71

[사]

사용 가능한 메모리 43

사용자 정의

키보드 59

화면 색상 63

사인자 인증서 72

상대 LU 이름 구성 설정 48

색상 맵핑 파일 63

샘플 텍스트 파일 105

서버

웹 24

응용프로그램 24

CICS 24

서버 이름 구성 설정 45

서버 재시도 간격 구성 설정 45

서버 키링 클래스 파일 73

서블릿 5

설명 구성 설정 45

설치

문제점 124

Linux용 CICS® Transaction

Gateway 27

소프트웨어 책, PDF 142

신뢰 루트 키 72

쓰레드 한계 13

[아]

암호화 17, 70

액세스 기능 20

연결 매니저 쓰레드의 초기 수 구성 설정 36

연결 매니저 쓰레드의 최대 수 구성 설정 36

연결 시간 초과 구성 설정 39, 47

온라인 문서 열람 141

온라인 문서, HTML 141

온라인 책, PDF 142

완전한 CP 이름 또는 템플릿 구성 설정 49

외부 사인 인증 78

운영 체제 22

원격 노드 비활동 폴 간격 구성 설정 51

웹 브라우저 8, 22

웹 서버 24

웹 서버s 8

유휴 시간 종료 구성 설정 39

응용프로그램 개발 도구 25

응용프로그램 서버 24

응용프로그램 트레이스 122

인쇄 명령 구성 설정 43

인쇄 파일 구성 설정 44

인증 기관(CA) 71

[자]

- 자가 사인 인증 84
- 자국어 지원 28
- 자주 발생하는 문제점 진단
 - 코드 페이지 문제점 127
 - 프로세스가 잠긴 경우 127
 - SSL 예외 128
- 자주 발생하는 문제점 판별 124
 - 게이트웨이가 Windows NT[®] 서비스 로 실행될 때 125
 - Java 예외 124
 - java.lang.OutOfMemory 예외 126
- 작동
 - 게이트웨이 시작 91
 - 게이트웨이 중지 94
- 작동 불가능 20
- 작동 중인 연결 제거 구성 설정 40
- 잠금 127
- 정지 127
- 진행 중 요청 시간 초과 구성 설정 38

[차]

- 책 137
 - 온라인 141
 - 인쇄된 142
- CICS Transaction Gateway 및 CICS Universal Clients 라이브러리 137
- PDF 142
- 책, CICS Transaction Gateway 및 CICS Universal Clients 라이브러리 137
- 초기 트랜잭션 구성 설정 46
- 초기화 파일
 - CICSCOL.INI 63
 - CICSKEY.INI 59
- 최대 논리 SNA 세션 수 구성 설정 51
- 최대 버퍼 크기 구성 설정 42
- 최대 서버 구성 설정 43

- 최대 요청 수 구성 설정 43
- 최대 클라이언트 램 크기 구성 설정 54
- 최대 SNA RU 크기 구성 설정 51

[카]

- 컴파일러 25
- 코드 페이지 문제점 127
- 코드 페이지 ID 대체 구성 설정 44
- 콘솔의 입력 읽기 가능 구성 설정 37
- 클라이언트 인증 사용 구성 설정 41
- 클라이언트 키링 클래스 파일 73
- 클라이언트 트레이스 파일 구성 설정 54
- 클라이언트가 보안 클래스를 사용하도록 함 구성 설정 40
- 키링 암호 구성 설정 41
- 키링 클래스명 구성 설정 41
- 키링 파일 73
- 키링s 18, 73
- 키보드 20
- 키보드 맵핑 파일 59

[타]

- 트레이스 120
 - 성능 고려사항 121
- 트레이스 구성 설정 52
- 트레이스, JNI 122

[파]

- 파일 맵핑
 - 이름 바꾸기 58
- 포스트스크립트 책 142
- 포트 구성 설정 39, 47
- 프로그램 지원 124
- 프로세스 잠금 127
- 프로토콜
 - APPC 3
 - HTML 9
 - HTTPS 16

- 프로토콜 (계속)
 - SSL(Secure Sockets Layer) 16
 - TCP/IP 3
- 프로토콜 핸들러 사용 가능 구성 설정 39

[하]

- 하드웨어 요구사항 21
- 하드카피 책 142
- 핸들러 웨이크업 시간 종료 구성 설정 39
- 호스트명 또는 IP 주소 구성 설정 47
- 환경 변수 58
 - CICSCOL 64
 - CICSKEY 59
 - CLASSPATH 31
 - LD_LIBRARY_PATH 67
- 휴대용 문서 형식(PDF) 142

A

- Anynet 도메인 이름 접미부 구성 설정 50
- AppletViewer 129
- Appid 구성 설정 42

B

- beans, Java 6
- BMS 맵 변환 유틸리티 103, 116

C

- CICS Transaction Gateway
 - 구성 31
 - 국지 19, 67
 - 로드 균형화 19
 - 문제점 판별 119
 - 사용자가 지정한 선택항목으로 게이트웨이Gateway 시작 92

CICS Transaction Gateway (계속)
시작 선택항목 92
중지 94
하드웨어 요구사항 21
Host on-Demand 19
preset 선택항목으로 CICS
Transaction Gateway 시작 91
CICS Transaction Gateway 설치제거
29
CICS Transaction Gateway 시작 91
CICS 서버 24
CICSCLI.LOG 45
CICSCOL 환경 변수 64
CICSKEY 환경 변수 59
CLASSPATH 31
CLASSPATH 설정값 32, 77
CORBA(Common Object Request
Broker) 표준 19
CP 이름의 IP 주소 마스크(선택항목) 구
성 설정 50
ctgconv, 변환 도구 55
ctgikey 76

D

DISPLAY 환경 변수 28

E

ESI(External Security Interface) 16
ESI(external security interface) 16
Euro 20
Euro 지원 44

G

Gateway.properties 파일 32

H

Host on-Demand 19
HTML 문서, 열람 141

HTML(HyperText Markup
Language) 141
HTTPS 18
HyperText Markup
Language(HTML) 141

I

IIOP(Internet InterOrb Protocol) 19

J

Java
방화벽 7
서블릿 5
애플릿 4
응용프로그램 6
클래스 3
beans 6
Java 언어 4
Java 개발 킷 23
Java 예외 124, 125
Java 클라이언트에서 일반 ECI 응답 수
신 구성 설정 38
java.lang.OutOfMemory 예외 126
JAVA_HOME 환경 변수 76
JDK 레벨 23
JNI 트레이스 52, 122
JSDK(Java Servlet Development
Kit) 3

L

LD_LIBRARY_PATH 67
LU 이름 템플릿의 IP 주소 마스크(선택
항목) 구성 설정 49

O

ORB (object request broker) 19
ORB(object request broker) 19

P

PDF 책, 열람하기 142
PDF(휴대용 문서 형식) 142
Ping 시간 빈도 구성 설정 40

R

RMI(Remote Method Invocation) 19
RMI(Remote Method Invocation) 19

S

Samples.txt 105, 106, 107
SNA 페이징 크기 구성 설정 51
SO_LINGER 설정 구성 설정 40
SSL 예외 128
SSL 핸드셰이크 75
SSL(Secure Sockets Layer) 17

T

TCP62 25
포트 번호 51
TCP/IP Keepalive 패킷 송신 구성 설정
47
TCP/IP 호스트명 표시 구성 설정 37
TCP/IP(Transmission Control
Protocol/Internet Protocol) 24
Telnet 23

U

uudecode 27, 124

W

WebSphere의 서블릿 트레이스 및 로그
기록 122
Worker 쓰레드 사용 시간 종료 구성 설
정 38
Worker 쓰레드의 초기 수 구성 설정 37

Worker 스레드의 최대 수 구성 설정 37

X

X-Windows 28



SA30-1344-00



Spine information:



**CICS Transaction
Gateway**

Linux 게이트웨이 관리

버전 4.0