Sun Java System Portal Server Secure Remote Access 7.2 관리 설명서

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

부품 번호: 820–4822
2008년 5월

080618@20490

목차

그림

표

코드 예

머리말

이 설명서는 Sun Java™ System Portal Server Secure Remote Access 7.2 서버 관리 방법을 설명합니다.

Sun Java System Portal Server Secure Remote Access(SRA) 서버는 또한 원격 사용자가 인터넷을 통해 안전하게 사용자 조직의 네트워크 및 자체 서비스에 액세스할 수 있도록 해주는 SRA(Secure Remote Access) 지원을 제공합니다. 그 외에도 SRA는 조직에 안전한 인터넷 포털을 제공하여 직원, 비즈니스 파트너 또는 일반 대중이 컨텐트, 응용 프로그램 및 데이터에 액세스할 수 있게 해줍니다.

이 머리말은 다음 절로 구성됩니다.

본 설명서의 독자

Sun Java System Portal Server Secure Remote Access 7.2 관리 설명서는 Secure Remote Access 서버를 구성 및 관리하는 사용자를 대상으로 합니다.

Sun Java System Portal Server Secure Remote Access 7.2 관리 설명서는 UNIX 시스템 및 TCP/IP 네트워크 관리 경험이 있는 네트워크 관리자 또는 시스템 관리자를 대상으로 하고 있습니다. Secure Remote Access 서버의 여러 구성 요소를 설치하는 데는 필수 시스템에 대한 루트 액세스 권한이 필요하지 않습니다. 사용자 및 서비스의 구성과 같이 기타 작업을 수행하는 경우에는 관리 권한이 필요합니다.

본 설명서를 읽기 전에

Portal Secure Remote Access 서버 관리자는 다음 기술에 대한 지식을 보유하고 있어야 합니다.

- Sun Java System Portal Server
- Sun Java System Directory Server
- Sun Java System Access Manager
- 사용 중인 웹 컨테이너
  - Sun Java System Application Server 8.2
  - Sun Java System Web Server 7.0
- 사용 중인 운영 체제
- 기본적인 UNIX® 관리 절차
- LDAP(Lightweight Directory Access Protocol)
- 원격 포틀릿용 웹 서비스(WSRP)

Rewriter 규칙을 작성하려면 다음 사항에 대한 지식도 필요합니다.

- HTML(Hypertext Markup Language) 및 HTML 태그에 대한 이해
- JavaScript™에 대한 상당한 지식
- XML(Extensible Markup Language)에 대한 기본적인 지식

본 설명서의 구성

본 설명서는 다음으로 구성됩니다.

- 제1부
  - 1 장에서는 Sun Java System Portal Server와 Portal Server Secure Remote Access 사이의 관계를 설명합니다.
  - 2 장에서는 게이트웨이 관련 개념 및 게이트웨이 관리 작업을 설명합니다.
  - 3 장에서는 사용자가 웹 페이지를 구문 분석하지 않고도 게이트웨이를 통해 인트라넷 웹 페이지에 액세스할 수 있게 해주는 Proxylet을 설명합니다.
  - 4 장에서는 Proxylet과 Rewriter를 사용하여 게이트웨를 통해 인트라넷 웹 페이지에 액세스할 수 있는 방법을 설명합니다.
  - 5 장에서는 NetFile을 사용하여 원격 파일 시스템과 디렉토리에 액세스하고 관리하는 방법을 설명합니다.
  - 6 장에서는 사용자가 Netlet을 사용하여 인터넷과 같은 비보안 네트워크를 통해 공통 TCP/IP를 안전하게 실행하는 방법을 설명합니다.
- 제2부
  - 7 장에서는 Portal Server 관리 콘솔에 대한 액세스 관리 방법을 설명합니다.

- 8 장에서는 Portal Server 관리 콘솔에서 게이트웨이 속성을 구성하는 방법을 설명합니다.
- 9 장에서는 Rewriter 탭의 게이트웨이 서비스를 사용하여 여러 작업을 수행하는 방법을 설명합니다.
- 10 장에서는 인증서를 관리하고 인증 기관에서 직접 서명한 인증서를 설치하는 방법을 설명합니다.
- 11 장에서는 Portal Server 관리 콘솔에서 Netlet 속성을 구성하는 방법을 설명합니다.
- 12 장에서는 PDC에서 Netlet을 사용할 수 있도록 클라이언트 브라우저의 Java 플러그인을 구성하는 방법을 설명합니다.
- 13 장에서는 Portal Server 관리 콘솔에서 Proxylet을 구성하는 방법을 설명합니다.
- 14 장에서는 Portal Server 관리 콘솔을 사용하여 NetFile 옵션, 권한 및 기본 설정을 설정하는 방법을 설명합니다.
- 15 장에서는 Portal Server Secure Remote Access Server의 다양한 가속기를 구성하는 방법을 설명합니다.

- 제3부
  - 16 장에서는 게이트웨이 프로필과 게이트웨이 인스턴스를 만드는 방법을 설명합니다.
  - 17 장에서는 네트워크 Identity를 관리하는 다양한 시나리오를 설명합니다.
- 부록 A에서는 각 Portal Server Secure Remote Access 구성 요소에 대해 Portal Server 관리 콘솔을 통해 Sun Java System Portal Server Secure Remote Access용으로 구성할 수 있는 속성을 설명합니다.
- 부록 B에는 디버깅 및 기타 유형의 정보가 포함되어 있습니다.
- 부록 C는 인증서 관리 중에 지정해야 하는 2자리 국가 코드 목록입니다.

관련 설명서

- **Sun Java System Portal Server 7.2 Deployment Planning Guide**
- **Sun Java System Portal Server 7.2 Technical Overview**
- **Sun Java System Portal Server 7.2 관리 설명서**
- **Sun Java System Portal Server 7.2 Command-Line Reference**
- **Sun Java System Portal Server 7.2 릴리스 노트**
- **Sun Java System Portal Server 7.1 Community Sample Guide**
- **Sun Java System Portal Server 7.2 Technical Reference**
- **Sun Java System Portal Server 7.2 Developer's Guide**

Portal Server 개념 및 구성 요소에 대한 소개는 **Sun Java System Portal Server 7.2 Technical Overview**에 나와 있습니다.

기타 서버 설명서

기타 서버 설명서는 다음을 참조하십시오.

- Directory Server 설명서: http://docs.sun.com/coll/1224.1 및 http://docs.sun.com/coll/1586.1
- Access Manager 설명서: http://docs.sun.com/coll/1292.2 및 http://docs.sun.com/coll/1399.2
- Web Server 설명서: http://docs.sun.com/coll/1308.3 및 http://docs.sun.com/coll/1410.2
- Application Server 설명서: http://docs.sun.com/coll/1310.3 및 http://docs.sun.com/coll/1401.2
- Web Proxy Server 설명서: http://docs.sun.com/coll/1311.4 및 http://docs.sun.com/coll/1581.2

타사 웹 사이트

본 설명서에서는 타사 URL을 참조하여 관련 정보를 추가로 제공합니다.

주 – Sun은 본 설명서에서 언급된 타사 웹 사이트의 가용성 여부에 대해 책임을 지지 않습니다. 또한 해당 사이트나 리소스를 통해 제공되는 내용, 광고, 제품 및 기타 자료에 대해 어떠한 보증도 하지 않으며 그에 대한 책임도 지지 않습니다. 따라서 타사 웹 사이트의 내용, 제품 또는 리소스의 사용으로 인해 발생한 실제 또는 주장된 손상이나 피해에 대해서도 책임을 지지 않습니다.

문서, 지원 및 교육

Sun 웹 사이트에서는 다음 추가 자원에 대한 정보를 제공합니다.

- 문서 (http://www.sun.com/documentation/)
- 지원 (http://www.sun.com/support/)
- 교육 (http://www.sun.com/training/)

표기 규칙

이 설명서에 사용된 표기 규칙은 다음 표와 같습니다.

표 P–1 표기 규칙

글자 모양	의미	예
`AaBbCc123`	명령, 파일, 디렉토리의 이름 및 컴퓨터 화면 출력	`.login` 파일을 편집하십시오. 모든 파일을 나열하려면 `ls -a`를 사용하십시오. `machine_name% you have mail.`
`AaBbCc123`	사용자가 입력하는 내용으로, 컴퓨터 화면에 출력되는 내용과 대비됩니다.	`machine_name%` **`su`** `Password:`
aabbcc123	자리 표시자: 실제 이름 또는 값으로 바뀝니다.	파일을 제거하는 명령은 `rm` *filename*입니다.
AaBbCc123	책 제목, 새로운 용어 및 강조 표시할 용어	**사용 설명서**의 6장을 읽으십시오. **캐시**는 로컬로 저장되는 복사본입니다. 파일을 **저장하지 마십시오**. **참고:**일부 강조된 항목이 온라인에서 굵게 표시됩니다.

명령에서 쉘 프롬프트의 예

다음 표에는 C 쉘, Bourne 쉘, Korn 쉘의 기본 UNIX 시스템 프롬프트와 수퍼유저 프롬프트가 나와 있습니다.

표 P–2 쉘 프롬프트

쉘	프롬프트
C 쉘	`machine_name%`
C 쉘 수퍼유저	`machine_name#`
Bourne 쉘 및 Korn 쉘	`$`
Bourne 쉘 및 Korn 쉘 수퍼유저	`#`

1

Secure Remote Access Server 구성 요소

◆ ◆ ◆ 1 장

1 Portal Server Secure Remote Access Server 소개

이 장에서는 Sun Java™ System Portal Server Secure Remote Access에 대한 설명과 더불어 Sun Java System Portal Server와 Sun Java System Portal Server Secure Remote Access 구성 요소 간의 관계를 설명합니다.

이 장에서는 다음 주제를 다룹니다.

Secure Remote Access 소개

Secure Remote Access를 사용하면 원격 사용자가 인터넷을 통해 조직의 네트워크와 서비스에 안전하게 액세스할 수 있습니다. 그 외에도 조직에 안전한 인터넷 포털을 갖추어 직원, 비즈니스 파트너 또는 일반 대중이 컨텐트, 응용 프로그램 및 데이터에 액세스할 수 있게 해줍니다.

Secure Remote Access를 사용하면 어떤 원격 장치에서도 브라우저를 통해 포털 컨텐트와 서비스에 원격으로 안전하게 액세스할 수 있습니다. Secure Remote Access는 안전한 액세스 솔루션으로서 Java™ 기술이 지원되는 브라우저가 있는 어떤 장치에서도 사용자가 액세스할 수 있어 클라이언트 소프트웨어의 필요성을 없애줍니다. Portal Server와의 통합으로 사용자는 액세스 권한을 가지고 있는 컨텐트와 서비스에 암호화된 방법으로 안전하게 액세스할 수 있게 되었습니다.

Secure Remote Access 소프트웨어는 안전한 원격 액세스 포털을 구축하고자 하는 기업에 이상적입니다. 이러한 포털은 보안, 안전 및 인트라넷 리소스의 기밀 유지에 중점을 둡니다. Secure Remote Access 아키텍처는 이러한 포털 유형에 적합합니다. Secure Remote Access 소프트웨어를 사용하면 이러한 자원을 인터넷에 노출시키지 않고 사용자가 인터넷을 통해 인트라넷 자원에 안전하게 액세스할 수 있습니다.

Portal Server는 다음 절에서 설명하는 대로 열린 모드와 보안 모드에서 작동됩니다.

열린 모드

열린 모드에서는 Portal Server가 Secure Remote Access 없이 설치됩니다. 이 모드에서 HTTPS 통신이 가능하지만 안전한 원격 액세스는 불가능합니다. 따라서 사용자는 원격 파일 시스템과 응용 프로그램에 안전하게 액세스할 수 없습니다.

열린 포털과 보안 포털 사이의 주된 차이점은 열린 포털로 제공되는 서비스가 일반적으로 안전한 인트라넷 내부가 아닌 완충 지대(DMZ)에 위치한다는 것입니다. DMZ는 공용 인터넷과 사설 인트라넷 사이의 작은 보호 네트워크로서 일반적으로 양쪽에서 방화벽으로 경계를 이룹니다.

공용 정보를 배포하고 무료 응용 프로그램에 액세스하는 등의 중요한 정보가 포털에 들어 있지 않아서, 이 모드를 사용하면 많은 수의 사용자가 보낸 액세스 요청에 대해 보안 모드를 사용할 때보다 빠르게 응답할 수 있습니다.

열린 모드에서 Portal Server는 방화벽 뒤의 단일 서버에 설치됩니다. 여러 클라이언트가 단일 방화벽을 통해 인터넷에서 Portal Server에 액세스합니다.

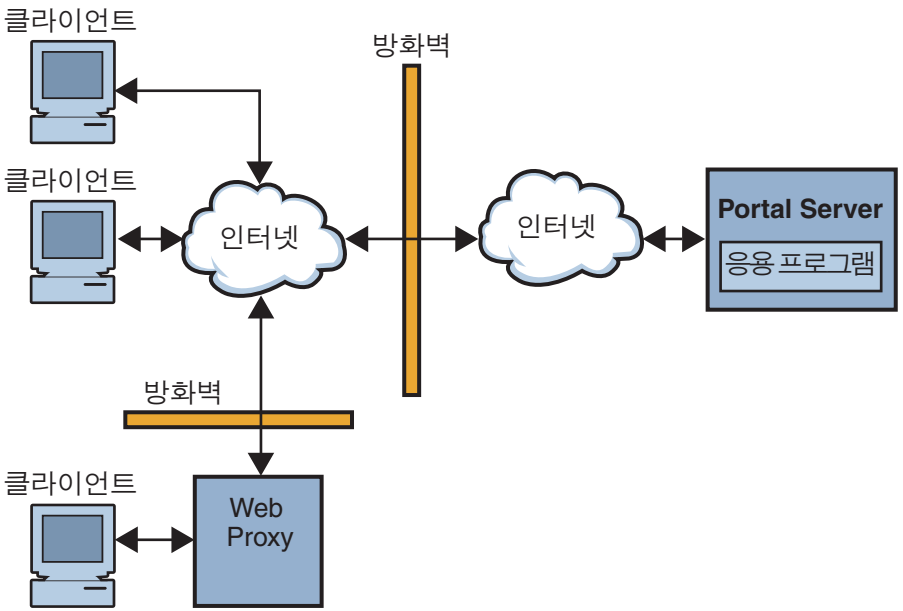


그림 1–1 Secure Remote Access가 있는 열린 모드의 Portal Server

보안 모드

보안 모드에서 사용자는 필요한 인트라넷 파일 시스템과 응용 프로그램에 안전하게 원격으로 액세스할 수 있습니다.

게이트웨이는 완충 지대(DMZ)에 상주합니다. 게이트웨이는 모든 인트라넷 URL과 응용 프로그램에 단일한 보안 액세스 포인트를 제공하여 방화벽에서 열린 포트의 수를 줄입니다. 세션, 인증 및 표준 포털 데스크탑과 같은 기타 모든 Portal Server 서비스는 DMZ 뒤의 안전한 인트라넷에 상주합니다. 클라이언트 브라우저와 게이트웨이 간의 통신은 SSL(Secure Sockets Layer) 상에서 HTTP를 사용하여 암호화됩니다. 게이트웨이와 서버 및 인트라넷 리소스 간의 통신에는 HTTP 또는 HTTPS를 이용할 수 있습니다.

보안 모드에서 SSL은 인터넷을 통한 클라이언트와 게이트웨이 간 연결을 암호화하는 데 사용됩니다. SSL은 게이트웨이와 서버 사이의 연결을 암호화할 때에도 사용됩니다. 인트라넷과 인터넷 사이에 게이트웨이가 있어 클라이언트와 Portal Server 간의 보안 경로가 확장됩니다.

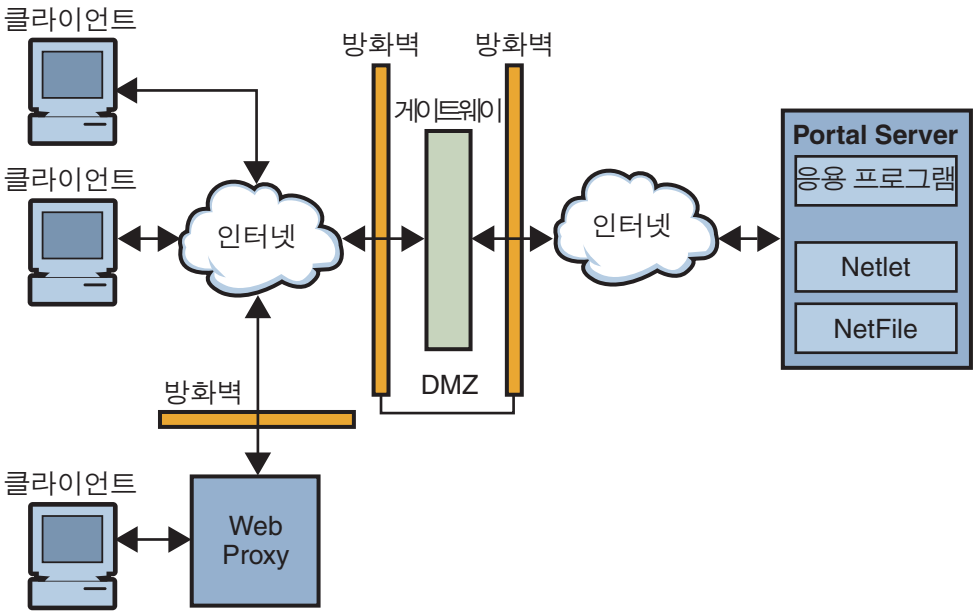


그림 1–2 Secure Remote Access가 있는 보안 모드의 Portal Server

사이트 확장을 위해 부가적인 서버와 게이트웨이를 추가할 수 있습니다. 이 경우 비즈니스 요구 사항에 따라 다양한 방법으로 Secure Remote Access 소프트웨어를 구성할 수 있습니다. 비즈니스 요구 사항에 따른 구성 방법에 대한 자세한 내용은 **Sun Java System Portal Server 7.2 Deployment Planning Guide**를 참조하십시오.

Secure Remote Access 서비스

Secure Remote Access 소프트웨어에는 5가지 주요 구성 요소가 있습니다.

- **게이트웨이**

  SRA 게이트웨이는 인터넷을 통해 들어오는 원격 사용자 세션과 회사 인트라넷 사이에서 인터페이스와 보안 장벽을 제공합니다. 게이트웨이는 원격 사용자에 대한 단일 인터페이스를 통해 내부 웹 서버와 응용프로그램 서버에서 안전하게 컨텐트를 제공합니다.

  웹 서버는 HTML, JavaScript 및 XML과 같은 웹 기반 자원을 사용하여 클라이언트와 게이트웨이 사이에서 통신합니다. Rewriter는 웹 컨텐트를 이용할 수 있도록 만드는 데 사용되는 게이트웨이 구성 요소입니다.

  응용 프로그램 서버는 Telnet 및 FTP와 같은 이진 프로토콜을 사용하여 클라이언트와 게이트웨이 사이에서 통신합니다. 게이트웨이에 상주하는 Netlet이 이 목적으로 사용됩니다. 자세한 내용은 2 장을 참조하십시오.

- **Rewriter**

  Rewriter는 최종 사용자가 인트라넷을 둘러보고 이 페이지의 링크와 기타 URL 참조가 제대로 작동하도록 합니다. Rewriter는 웹 브라우저의 위치 필드에 게이트웨이 URL을 붙여 컨텐트 요청을 게이트웨이를 통해 리디렉션합니다. 자세한 내용은 4 장을 참조하십시오.

- **Netfile**

  NetFile은 사용자가 원격 파일 시스템과 디렉토리에 원격으로 액세스하여 작업할 수 있도록 해주는 파일 관리자 응용 프로그램입니다. NetFile에는 Java 기반의 사용자 인터페이스가 포함되어 있습니다. 자세한 내용은 5 장을 참조하십시오.

- **Netlet**

  Netlet은 원격 데스크탑에서 일반 또는 회사별 응용 프로그램을 안전하게 실행하도록 지원합니다. 해당 사이트에 Netlet을 구현하면 사용자가 Telnet 및 SMTP와 같은 일반적 TCP/IP 서비스와 pcANYWHERE 또는 Lotus Notes같은 HTTP 기반 응용 프로그램을 안전하게 실행할 수 있습니다. 자세한 내용은 6 장을 참조하십시오.

- **Proxylet**

  Proxylet은 클라이언트 컴퓨터에서 실행되는 동적 프록시 서버입니다. Proxylet은 클라이언트 시스템에 있는 브라우저의 프록시 설정을 읽고 로컬 프록시 서버 또는 Proxylet을 가리키도록 수정하여 URL을 게이트웨이로 리디렉션합니다.

Secure Remote Access 속성 구성

다음 서비스를 사용하여 Portal Server 관리 콘솔에서 Secure Remote Access 속성을 구성할 수 있습니다.

- 액세스 제어

  이 서비스를 통해 특정 URL에 대한 액세스를 허용 또는 제한하고 단일 사인온 기능을 관리할 수 있습니다. 자세한 내용은 7 장을 참조하십시오.

- 게이트웨이

  프로필(게이트웨이 인스턴스). 이 서비스를 사용하면 구성 요소 활성화, 쿠키 관리, 프록시 관리, 보안 설정, 성능 조정, Rewriter 매핑 관리 등의 모든 게이트웨이 관련 속성을 구성할 수 있습니다. 자세한 내용은 8 장을 참조하십시오.

- NetFile

  이 서비스를 사용하여 공통 호스트, MIME 유형 및 여러 호스트 유형에 대한 액세스 등 모든 NetFile 관련 속성을 구성할 수 있습니다. 자세한 내용은 14 장을 참조하십시오.

- Netlet

  이 서비스를 사용하여 Netlet 규칙, 필요한 규칙에 대한 액세스, 조직 및 호스트 그리고 기본 알고리즘과 같은 모든 Netlet 관련 속성을 구성할 수 있습니다. 자세한 내용은 11 장을 참조하십시오.

- Rewriter

  이 서비스를 사용하면 모든 Rewriter 규칙 집합을 다운로드, 업로드 및 삭제할 수 있습니다.

- Proxylet

  이 서비스를 사용하면 Proxylet 애플릿 바인딩 IP 주소 및 포트 번호와 같은 Proxylet 관련 속성을 구성할 수 있습니다. 자세한 내용은 13 장을 참조하십시오.

주의 – 게이트웨이는 게이트웨이가 실행되는 동안 이루어지는 속성 변경에 대해 알림을 받지 않습니다. 업데이트된 프로필 속성(게이트웨이 또는 기타 서비스에 속함)을 적용하려면 게이트웨이를 다시 시작합니다. 자세한 내용은 180 페이지 “명령줄 옵션을 사용한 게이트웨이 속성 구성”을 참조하십시오.

충돌 해결 설정

▼ 충돌 해결 수준을 설정하려면

1 **Sun Java System Portal Server 7.2 관리 설명서의 “관리 콘솔에 로그인하려면”에 나와 있는 대로 수행합니다.**

2 **[Secure Remote Access] 탭을 선택하고[Netlet], [Netfile] 또는 [Proxylet] 중에서 필요한 서비스 탭을 누릅니다.**

3 **[DN 선택] 드롭다운 메뉴에서 [조직] 또는 [역할]을 선택합니다.**

4 **[COS 우선 순위] 드롭다운 상자에서 필요한 충돌 해결 수준을 선택합니다.**

5 **[저장]을 눌러 완료합니다.**

지원되는 응용 프로그램

SRA는 다음 응용 프로그램을 지원합니다.

- Sun Java System Calendar Server Release 5.1.1 이상
- Sun Java System Messenger Express 6 2005Q1 - Sun Java System Messaging Server 5.2 이상
- Sun Java System Communications Express 6 2005Q1

시작하기 전에

▼ 포털에서 SRA를 활성화하려면

1 `PortalServer_base/psadmin switch-sra-status -u amadmin -f <passwordfile> on` **명령을 사용하여 SRA 상태를 전환합니다.**

2 `PortalServer_base/psadmin provision-sra -u amadmin -f <passwordfile> -p <portal-id> --gateway-profile <profile-name> --enable` **명령을 사용하여 SRA 상태를 지정합니다.**

◆ ◆ ◆ 2 장

2 게이트웨이 작업

이 장에서는 게이트웨이 관련 개념을 설명합니다. 게이트웨이 관리에 대한 자세한 내용은 16 장을 참조하십시오게이트웨이 구성에 대한 자세한 내용은 8 장을 참조하십시오.

이 장에서는 다음 주제를 다룹니다.

게이트웨이 소개

게이트웨이는 인터넷을 통해 들어오는 원격 사용자 세션과 회사 인트라넷 사이에서 인터페이스와 보안 장벽을 제공합니다. 게이트웨이는 원격 사용자에 대한 단일 인터페이스를 통해 내부 웹 서버와 응용프로그램 서버에서 안전하게 컨텐트를 제공합니다.

각 게이트웨이 인스턴스에 대해 다음 작업을 완료해야 합니다.

기타 게이트웨이 관련 항목은 다음과 같습니다.

- 33 페이지 “게이트웨이 다시 시작”
- 33 페이지 “게이트웨이 워치독 구성”
- 33 페이지 “가상 호스트 지정”
- 34 페이지 “Access Manage에 접속할 프록시 지정”

게이트웨이 프로필 만들기

게이트웨이 프로필은 게이트웨이가 수신하는 포트, SSL 옵션 및 프록시 옵션과 같은 게이트웨이 구성과 관련된 모든 정보를 포함합니다. 게이트웨이를 설치하는 경우 기본값을 선택하면 "default"라는 기본 게이트웨이 프로필이 만들어집니다. 기본 프로필에 해당하는 구성 파일은 다음 위치에 있습니다.
`/etc/opt/SUNWportal/platform.conf.default`.

여기서 `/etc/opt/SUNWportal`은 모든 `platform.conf.*` 파일의 기본 위치입니다. `platform.conf` 파일에 대한 자세한 내용은 34 페이지 “platform.conf 파일 이해”를 참조하십시오.

프로필과 관련하여 다음과 같은 작업을 수행할 수 있습니다.

- 여러 프로필을 만들어 각 프로필에 대한 속성을 정의한 다음 이 프로필을 필요에 따라 서로 다른 게이트웨이에 할당할 수 있습니다.
- 서로 다른 컴퓨터에 있는 게이트웨이 설치에 단일 프로필을 할당할 수 있습니다.
- 같은 컴퓨터에서 실행되는 단일 게이트웨이 인스턴스에 서로 다른 프로필을 할당할 수 있습니다.

주의 – 같은 컴퓨터에서 실행되는 게이트웨이의 서로 다른 인스턴스에 같은 프로필을 할당하지 마십시오. 이렇게 설정하면 포트 번호가 같게 되므로 충돌이 발생합니다.

같은 게이트웨이에 만들어진 서로 다른 프로필에서 같은 포트 번호를 지정하지 마십시오. 동일한 게이트웨이의 포트 번호가 같은 다중 인스턴스를 실행하면 충돌이 발생합니다.

게이트웨이의 여러 인스턴스 만들기

여러 게이트웨이 인스턴스를 만들려면 **Sun Java System Portal Server 7.2 Installation and Configuration Guide**의 4 장, “Installing and Configuring a Gateway With Portal Server”을 참조하십시오.

다중 홈 게이트웨이 인스턴스 만들기

다중 홈 게이트웨이 인스턴스는 하나의 Portal Server에 여러 개의 게이트웨이가 있는 것입니다. 이러한 인스턴스를 만들려면 `platform.conf` 파일을 다음과 같이 수정합니다.

```
gatewaybindipaddress = 0.0.0.0
```

같은 LDAP를 사용하여 게이트웨이 인스턴스 만들기

같은 LDAP를 사용하는 게이트웨이 인스턴스 여러 개를 만드는 경우에는 첫 게이트웨이를 만든 후에 그 뒤의 모든 게이트웨이에서 다음을 수행합니다.

`/etc/opt/SUNWam/config/`에서 `AMConfig-`*instance-name*`.properties`의 다음 영역을 처음 설치한 게이트웨이 인스턴스와 일치하도록 수정합니다.

242 페이지 “같은 LDAP를 사용하여 게이트웨이 인스턴스를 만들려면”을 참조하십시오.

게이트웨이 다시 시작

일반적으로 게이트웨이를 다시 시작할 필요가 없습니다. 다음 이벤트가 발생한 경우에만 다시 시작합니다.

- 새 프로필을 만들고 이를 게이트웨이에 할당해야 하는 경우
- 기존 프로필의 일부 속성을 수정하고 변경 사항을 적용해야 하는 경우
- 메모리 부족(OutOfMemory)과 같은 오류 때문에 게이트웨이가 충돌하는 경우
- 게이트웨이가 응답을 중지하고 요청에 대한 서비스를 제공하지 않는 경우

게이트웨이 워치독 구성

워치독이 게이트웨이의 상태를 모니터링하게 될 시간 간격을 설정할 수 있습니다. 워치독을 시작 또는 중지하려면 `./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off` 명령을 실행합니다. 시간 간격은 기본적으로 60초로 설정됩니다. 이 값을 변경하려면 crontab 유틸리티에서 다음 행을 편집합니다.

```
0-59 * * * * gateway-install-root/SUNWportal/bin/
/var/opt/SUNWportal/.gw. 5 > /dev/null 2>&1
```

`crontab` 항목을 구성하려면 `crontab` man 페이지를 참조하십시오.

가상 호스트 지정

가상 호스트는 같은 시스템 IP와 호스트 이름을 가리키는 추가 호스트 이름입니다. 예를 들어 호스트 이름 `abc`가 호스트 IP 주소 192.155.205.133을 가리키는 경우, 같은 IP 주소를 가리키는 다른 호스트 이름 `cde`를 추가할 수 있습니다.

Access Manage에 접속할 프록시 지정

게이트웨이에서 프록시 호스트를 사용하여 Portal Server에 배포되는 SRA 코어(RemoteConfigServlet)에 접속하도록 지정할 수 있습니다. 이 프록시는 게이트웨이가 Portal Server와 Access Manager에 접속하기 위해 사용됩니다. 245 페이지 "프록시를 지정하려면"을 참조하십시오.

platform.conf 파일 이해

platform.conf 파일은 기본적으로 다음 위치에 있습니다. /etc/opt/SUNWportal.

platform.conf 파일에는 게이트웨이에 필요한 상세 정보가 들어 있습니다. 이 절에는 예제 platform.conf 파일이 나와 있으며 모든 항목에 대해 설명합니다.

모든 컴퓨터별 상세 정보를 구성 파일에 포함시키면 공통 프로필을 여러 컴퓨터에서 실행되는 게이트웨이에서 공유할 수 있다는 장점이 있습니다.

다음은 platform.conf 파일의 샘플입니다.

```
Tue May 30 11:51:23 IST 2006
debug.com.sun.portal.rewriter.original.level=INFO
gateway.favicon=
gateway.bindipaddress=10.12.154.236
debug.com.sun.portal.sra.rproxy.toFromServer.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/Gateway.toFromServer.%u.%g.log
gateway.port=443
rewriterproxy.jvm.flags=-ms64m -mx128m
portal.server.instance=default
debug.com.sun.portal.handler.java.util.logging.FileHandler.filter=
gateway.jdk.dir=/usr/jdk/entsys-j2se
gateway.ignoreURIList=/MSOffice/cltreq.asp,/_vti_bin/owssvr.dll
debug.com.sun.portal.rewriter.rest.level=INFO
gateway.trust_all_server_certs=true
debug.com.sun.portal.handler.java.util.logging.FileHandler.append=true
gateway.cdm.cacheCleanupTime=300000
gateway.httpurl=
debug.com.sun.portal.handler.java.util.logging.FileHandler.count=1
gateway.jvm.classpath=
debug.com.sun.portal.setserverlogs=false
gateway.protocol=https
debug.com.sun.portal.sra.rproxy.toFromServer=java.util.logging.FileHandler
rewriterproxy.jvm.classpath=
gateway.enable.customurl=false
debug.com.sun.portal.sra.rproxy.toFromBrowser=java.util.logging.FileHandler
debug.com.sun.portal.handler.java.util.logging.FileHandler.formatter=com.sun.portal.
log.common.PortalLogFormatter
```

```
debug.com.sun.portal.sra.rproxy.toFromBrowser.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/Gateway.toFromBrowser.%u.%g.log
debug.com.sun.portal.level=INFO
debug.com.sun.portal.rewriter.unaffected.separatefile=true
gateway.enable.accelerator=false
debug.com.sun.portal.rewriter.original.separatefile=true
gateway.virtualhost=nicp236.india.sun.com 10.12.154.236
debug.com.sun.portal.stacktrace=true
gateway.host=nicp236.india.sun.com
debug.com.sun.portal.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/%logger.%sraComponentType.%u.%g.log
gateway.certdir=/etc/opt/SUNWportal/cert/default
gateway.sockretries=3
gateway.allow.client.caching=true
debug.com.sun.portal.rewriter.unaffected.level=INFO
debug.com.sun.portal.rewriter.uriinfo.separatefile=true
log.config.check.period=2000
debug.com.sun.portal.rewriter.rewritten.level=INFO
gateway.userProfile.cacheSize=1024
debug.com.sun.portal.rewriter.rulesetinfo.level=INFO
netletproxy.jvm.classpath=
gateway.userProfile.cacheSleepTime=60000
debug.com.sun.portal.rewriter.uriinfo.level=INFO
debug.com.sun.portal.rewriter.rest.separatefile=true
gateway.notification.url=notification
debug.com.sun.portal.rewriter.rulesetinfo.separatefile=true
gateway.logdelimiter=&&
gateway.ignoreServerList=false
gateway.jvm.flags=-ms64m -mx128m
debug.com.sun.portal.handler.java.util.logging.FileHandler.limit=5000000
gateway.dsame.agent=http\://sunone216.india.sun.com\:8080/portal/RemoteConfigServlet
gateway.httpsurl=
gateway.retries=6
gateway.userProfile.cacheCleanupTime=300000
gateway.logging.password=X03MO1qnZdYdgyfeuILPmQ\=\= UX9x0jIua3hx1YOVRG/TLg\=\=
netletproxy.jvm.flags=-ms64m -mx128m
debug.com.sun.portal.rewriter.rewritten.separatefile=true
gateway.user=noaccess
gateway.external.ip=10.12.154.236
debug.com.sun.portal.handler=java.util.logging.FileHandler
gateway.cdm.cacheSleepTime=60000
rewriterproxy.accept.from.gateways=
rewriterproxy.checkacl=false
```

다음 표에는 `platform.conf` 파일의 모든 필드와 해당 설명이 정리되어 있습니다.

표 2–1 파일 등록 정보

항목	기본값	설명
gateway.user	noaccess	게이트웨이가 이 사용자로 실행됩니다. 게이트웨이는 루트로 시작되어야 하며 초기화 후에는 이 사용자가 되는 루트 권한을 상실합니다.
gateway.jdk.dir		게이트웨이에서 사용하는 JDK 디렉토리의 위치입니다.
gateway.dsame.agent		이 프로필을 얻을 수 있도록 시작하는 중에 게이트웨이에서 접속하는 Access Manager의 URL입니다.
portal.server.protocol portal.server.host portal.server.port		기본 Portal Server 설치에서 사용하는 프로토콜, 호스트 및 포트입니다.
gateway.protocolgateway.hostgateway.port		게이트웨이 프로토콜, 호스트 및 포트입니다. 이 값은 설치 시 지정한 모드 및 포트와 동일합니다. 이 값은 알림 URL을 구성하는 데 사용됩니다.
gateway.trust_all_server_certs	true	게이트웨이에서 모든 서버 인증서를 신뢰해야 하는지 아니면 게이트웨이 인증서 데이터베이스에 있는 서버 인증서만 신뢰해야 하는지를 나타냅니다.
gateway.trust_all_server_cert_domains	false	게이트웨이와 서버 사이에 SSL 통신이 수행될 때 서버 인증서가 게이트웨이에 제공됩니다. 기본적으로 게이트웨이는 서버 호스트 이름이 서버 인증서 CN과 같은지 확인합니다. 이 속성 값이 true로 설정되어 있으면 게이트웨이에서는 수신하는 서버 인증서에 대해 도메인 확인을 사용하지 않습니다.
gateway.virtualhost		게이트웨이 컴퓨터에 구성된 호스트 이름이 여러 개 있을 경우 이 필드에서 이름을 다르게 지정하여 공급자 주소를 구분할 수 있습니다.

표 2–1 파일 등록 정보 *(계속)*

항목	기본값	설명

`gateway.virtualhost.defaultOrg=org`		사용자가 로그인할 기본 조직을 지정합니다.
		예를 들어, 가상 호스트 필드 항목이 다음과 같다고 가정해 보겠습니다.
		`gateway.virtualhost=test.com employee.test.com`
		`Managers.test.com`
		기본 조직 항목이 다음과 같음:
		`test.com.defaultOrg = o=root,dc=test,dc=com`
		`employee.test.com.defaultOrg = o=employee,dc=test,dc=com`
		`Manager.test.com.defaultOrg = o=Manager,dc=test,dc=com`
		사용자는 `https://manager.test.com`을 통해 `https://test.com/o=Manager,dc=test,dc=com` 대신 관리자 조직에 로그인할 수 있습니다.
		주 – virtualhost 및 defaultOrg는 `platform.conf` 파일에서는 대소문자가 구별되지만 URL에 사용할 때에는 구별되지 않습니다.

`gateway.notification.url`		게이트웨이 호스트, 프로토콜 및 포트 조합은 알림 URL을 구성하는 데 사용됩니다. 이 조합은 Access Manager의 세션 알림을 수신하는 데 사용됩니다. 알림 URL은 다른 조직 이름과 같지 않도록 합니다. 알림 URL은 조직 이름과 일치하므로 해당 조직에 연결을 시도하는 사용자에게는 로그인 페이지 대신 공백 페이지가 나타납니다.
`gateway.retries`		시작하는 중에 게이트웨이에서 Portal Server에 접속하려고 시도하는 횟수를 말합니다.

표 2–1 파일 등록 정보 *(계속)*

항목	기본값	설명
gateway.debug	error	게이트웨이의 디버그 수준을 설정합니다. 디버그 로그 파일은 *debug-directory*/files에 있습니다. 디버그 파일 위치는 gateway.debug.dir 항목에 지정되어 있습니다. 디버깅 수준은 다음과 같습니다. ■ 오류 - 디버그 파일에 심각한 오류만 기록됩니다. 일반적으로 이러한 오류가 발생하면 게이트웨이는 기능이 정지합니다. ■ 경고 - 경고 메시지가 기록됩니다. ■ 메시지 - 모든 디버그 메시지가 기록됩니다. ■ 날짜 - 모든 디버그 메시지가 콘솔에 표시됩니다. 디버그 파일은 다음과 같습니다. srapGateway.*gateway-profile-name* - 게이트웨이 디버그 메시지가 들어 있습니다. Gateway_to_from_server.*gateway-profile-name* - 메시지 모드에서는 이 파일에 게이트웨이와 내부 서버 사이의 모든 요청 및 응답 헤더가 들어 있습니다. 이 파일을 생성하려면 /var/opt/SUNWportal/debug 디렉토리에서 쓰기 권한을 변경합니다. Gateway_to_from_browser.*gateway-profile-name* - 메시지 모드에서는 이 파일에 게이트웨이와 클라이언트 브라우저 사이의 모든 요청 및 응답 헤더가 들어 있습니다. 이 파일을 생성하려면 /var/opt/SUNWportal/debug 디렉토리에서 쓰기 권한을 변경합니다.
gateway.debug.dir		모든 디버그 파일이 생성되는 디렉토리입니다. 이 디렉토리에는 gateway.user에서 언급한 사용자가 파일에 쓸 수 있도록 충분한 권한을 가지고 있어야 합니다.
gateway.logdelimiter		현재 사용되지 않음.
gateway.external.ip		다중 홈 게이트웨이 컴퓨터인 경우(IP 주소가 여러 개) 여기서 외부 IP 주소를 지정해야 합니다. 이 IP는 Netlet에서 FTP를 실행하는 데 사용됩니다.
gateway.certdir		인증서 데이터베이스의 위치를 지정합니다.
gateway.allow.client.caching	true	클라이언트 캐싱을 허용하거나 금지합니다. 허용되는 경우 클라이언트 브라우저는 동적 페이지와 이미지를 캐싱하여 성능을 향상시킵니다(네트워크 트래픽 감소를 통해). 금지된 경우 아무 것도 캐싱되지 않으며 보안은 강화되지만 네트워크 부하가 증가하여 성능이 떨어집니다.

표 2–1 파일 등록 정보 *(계속)*

항목	기본값	설명
gateway.userProfile.cacheSize		게이트웨이에서 캐싱되는 사용자 프로필 항목 수입니다. 항목 수가 이 값을 초과하면 캐시를 정리하는 재시도가 자주 이루어집니다.
gateway.userProfile.cacheSleepTime		초 단위로 캐시 정리를 위한 절전 시간을 설정합니다.
gateway.userProfile.cacheCleanupTime		이 시간이 지나면 프로필 항목을 삭제할 수 있는 최대 시간(초).
gateway.bindipaddress		다중 홈 컴퓨터에서 게이트웨이가 serversocket을 바인딩하는 IP 주소입니다. 모든 인터페이스를 청취하도록 게이트웨이를 구성하려면 gateway.bindipaddress=0.0.0.0이 되도록 IP 주소를 변경합니다.
gateway.sockretries	3	현재 사용되지 않음.
gateway.enable.accelerator	false	true로 설정된 경우 외부 가속기 지원이 허용됩니다.
gateway.enable.customurl	false	true로 설정된 경우 관리자는 게이트웨이에서 페이지를 다시 쓸 사용자 정의 URL을 지정할 수 있습니다.
gateway.httpurl		게이트웨이에서 페이지를 다시 쓸 사용자 정의 URL에 대한 HTTP 역 프록시 URL. Proxylet이 사용되는 경우 이 항목을 사용합니다.
gateway.httpsurl		게이트웨이에서 페이지를 다시 쓸 사용자 정의 URL에 대한 HTTPS 역 프록시 URL. Proxylet이 사용되는 경우 이 항목을 사용하지 마십시오.
gateway.favicon		게이트웨이에서 favicon.icon 파일 요청을 재지정하는 URL Internet Explorer 및 Netscape 7.0 이상에 있는 "favorite icon"에 사용됩니다. 이 필드가 비어 있으면 게이트웨이는 '404 찾을 수 없습니다'라는 메시지를 브라우저로 반환합니다.
gateway.logging.password		게이트웨이에서 응용 프로그램 세션을 만드는 데 사용하는 사용자 amService-srapGateway의 LDAP 비밀 번호 암호화되었거나 일반 텍스트일 수 있습니다.
http.proxyHost		이 프록시 호스트는 Portal Server에 접속할 때 사용됩니다.
http.proxyPort		Portal Server에 접속할 때 사용되는 호스트의 포트입니다.
http.proxySet		이 등록 정보는 프록시 호스트가 필요한 경우에 true로 설정됩니다. 이 등록 정보가 false로 설정되면 http.proxyHost 및 http.proxyPort가 무시됩니다.

표 2–1 파일 등록 정보 *(계속)*

항목	기본값	설명
`portal.server.`*instance*		이 등록 정보의 값은 해당 `/etc/opt/SUNWam/config/AMConfig-`*instance-name*`.properties` 파일입니다. 이 값이 기본값이면 `AMConfig.properties`를 가리킵니다.
`gateway.cdm.cacheSleepTime`	60000	캐시 클라이언트 검색 모듈의 응답을 Access Manager에서 게이트웨이로 보내는 경우의 시간 제한 값입니다.
`gateway.cdm.cacheCleanupTime`	300000	캐시 클라이언트 검색 모듈의 응답을 Access Manager에서 게이트웨이로 보내는 경우의 시간 제한 값입니다.
`netletproxy.port`	10555	Netlet 프록시 데몬은 이 포트에서 요청을 수신합니다.
`rewriterproxy.port`	10555	Rewriter 프록시 데몬은 이 포트에서 요청을 수신합니다.
`gateway.ignoreServerList`	false	true로 설정하면 `AMConfig.properties` 파일에 지정된 값을 사용하여 Access Manager 서버 URL이 구성됩니다. Access Manager 서버가 로드 조정기 뒤에 있는 경우 이 등록 정보를 설정합니다.
`rewriterproxy.accept.from.gateways`		해당 IP 주소로부터 들어오는 요청을 수락하도록 Rewriter 프록시를 구성할 수 있는 IP 주소의 목록입니다. 이 등록 정보는 HTTP 및 HTTPS 모드 둘 다에서 작동합니다. 보안을 강화하기 위한 것이며, 이 설정에서 들어오는 요청만 수락하고 기타 모든 요청은 처리하지 않습니다. 이 값은 쉼표로 구분한 IP 주소일 수 있습니다. 기본값은 공백이며 레거시 모드로 간주됩니다. 즉,Rewriter 프록시로 들어오는 모든 요청이 수락됩니다.
`rewriterproxy.checkacl=`	false	이 등록 정보를 사용하면 Rewriter 프록시에서 게이트웨이와 마찬가지로 ACL 값을 확인하도록 할 수 있습니다. 레거시 모드 값은 "false"입니다. true로 설정할 경우 Rewriter 프록시가 특정 DN에서 게이트웨이 액세스 서비스에 지정된 값에 대해 URL을 확인하고 해당 목록 설정에 따라 요청을 허용/거부합니다. 이 값은 HTTP 및 HTTPS 모드 둘 다에서 유용합니다.

웹 프록시 사용

타사 웹 프록시를 사용하여 HTTP 자원에 연결하도록 게이트웨이를 구성할 수 있습니다. 웹 프록시는 클라이언트와 인터넷 사이에 상주합니다.

웹 프록시 구성

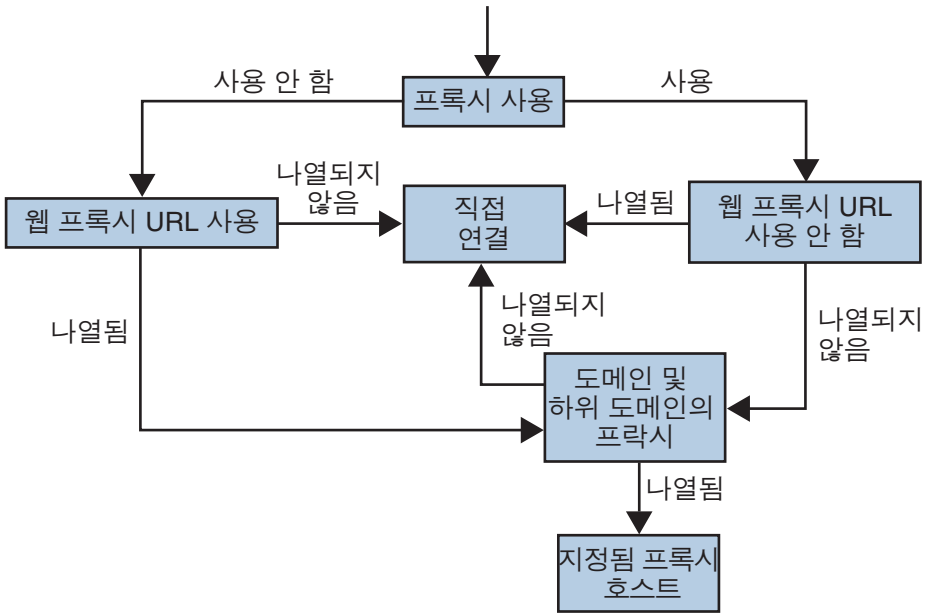
여러 도메인 및 부속 도메인에 서로 다른 프록시가 사용될 수 있습니다. 이 항목은 특정 도메인에서 특정 부속 도메인에 연결할 때 어떤 프록시를 사용할지 게이트웨이에 알려 줍니다. 게이트웨이에 지정된 프록시 구성은 다음과 같이 작동합니다.

- 게이트웨이 서비스의 [도메인 및 부속 도메인의 프록시] 필드에 필요한 프록시와 함께 도메인 및 부속 도메인 목록을 만듭니다.
- 프록시 사용 옵션을 사용하는 경우,
  - [도메인 및 부속 도메인의 프록시] 필드에 지정된 프록시가 지정된 호스트에 사용됩니다.
  - 도메인 및 부속 도메인의 프록시 목록에 지정된 도메인 및 부속 도메인에서 특정 URL에 직접 연결하려면 [웹 프록시 URL 사용 안 함] 필드에서 해당 URL을 지정합니다.

  프록시 사용 옵션이 사용 불가능 상태인 경우,
  - 프록시가 도메인 및 부속 도메인의 프록시 필드에 지정된 도메인과 부속 도메인에서 특정 URL에 사용되도록 하려면 [웹 프록시 URL 사용] 목록에서 해당 URL을 지정합니다.

    프록시 사용 옵션이 사용 불가능 상태이더라도 [웹 프록시 사용]에 나열된 URL에 프록시를 사용하여 연결할 수 있습니다. 이 URL의 프록시는 [도메인 및 부속 도메인의 프록시] 목록에서 가져온 것입니다.

    다음 그림에서는 게이트웨이 서비스의 프록시 구성에 기반하여 웹 프록시 정보가 어떻게 결정되는지 보여줍니다.



40 페이지 “웹 프록시 구성”에서 프록시 사용이 활성화되어 있고, 요청된 URL이 [웹 프록시 URL 사용 안함] 목록에 나열되는 경우 게이트웨이가 대상 호스트에 직접 연결됩니다.

프록시 사용이 활성화되어 있고, 요청된 URL이 [웹 프록시 URL 사용 안함] 목록에 나열되지 않은 경우 게이트웨이는 지정된 프록시를 통해 대상 호스트에 연결됩니다. 프록시가 지정되어 있는 경우에는 [도메인 및 부속 도메인의 프록시] 목록에서 찾으면 됩니다.

프록시 사용이 비활성화되어 있고, 요청된 URL이 [웹 프록시 URL 사용] 목록에 나열되면 게이트웨이는 [도메인 및 부속 도메인의 프록시] 목록에 있는 프록시 정보를 사용하여 대상 호스트에 연결됩니다.

프록시 사용이 비활성화되어 있고, 요청된 URL이 [웹 프록시 URL 사용] 목록에 나열되지 않으면 게이트웨이가 대상 호스트에 직접 연결됩니다.

위에 설명된 조건 중 어느 것에도 해당하지 않아서 직접 연결이 불가능하면 연결할 수 없다는 게이트웨이 오류 메시지를 표시합니다.

주 – 표준 포털 데스크탑의 책갈피 채널을 통해 URL에 액세스하는 중에 위에 설명된 조건 중 어느 것도 충족되지 않으면 게이트웨이는 브라우저로 리디렉션합니다. 그러면 브라우저는 자체 프록시 설정을 통해 URL에 액세스합니다.

구문

```
domainname [web_proxy1:port1]|subdomain1 [web_proxy2:port2]|
```

예

```
sesta.com wp1:8080|red wp2:8080|yellow|* wp3:8080
```

*는 모든 항목과 일치하는 와일드카드입니다.

여기서,

`sesta.com`은 도메인 이름이고 `wp1`은 포트 8080에 연결할 프록시입니다.

`red`는 하위 도메인이고 `wp2`는 포트 8080에 연결할 프록시입니다.

`yellow`는 하위 도메인입니다. 프록시가 지정되어 있지 않고 포트 8080에 도메인에 지정된 프록시 즉, `wp1`이 사용됩니다.

*는 모든 다른 하위 도메인에서 포트 8080에 `wp3`을 사용해야 함을 나타냅니다.

주 – 포트를 지정하지 않은 경우 기본적으로 포트 8080이 사용됩니다.

웹 프록시 정보처리

클라이언트가 특정 URL에 액세스하려고 하면 해당 URL의 호스트 이름이 [도메인 및 하위 도메인의 프록시] 목록의 항목과 일치됩니다. 요청된 호스트 이름의 가장 긴 접미어에 일치하는 항목이 선택됩니다. 예를 들어, 요청된 호스트 이름이 `host1.sesta.com`이라고 가정해 보겠습니다. 다음 검색 작업은 일치하는 항목을 찾을 때까지 수행됩니다.

- [도메인 및 하위 도메인의 프록시]에 `host1.sesta.com`이 있는지 검색합니다. 일치하는 항목이 있으면 이 항목에 지정된 프록시를 통해 그 호스트에 연결됩니다.
- 그렇지 않으면 목록에 `*.sesta.com`이 있는지 검색합니다. 항목을 찾으면 해당 프록시가 사용됩니다.
- 그렇지 않으면 목록에 `sesta.com`이 있는지 검색합니다. 항목을 찾으면 해당 프록시가 사용됩니다.
- 그렇지 않으면 목록에 `*.com`이 있는지 검색합니다. 항목을 찾으면 해당 프록시가 사용됩니다.
- 그렇지 않으면 목록에 `com`이 있는지 검색합니다. 항목을 찾으면 해당 프록시가 사용됩니다.
- 그렇지 않으면 목록에서 *에 대한 검색을 수행합니다. 항목을 찾으면 해당 프록시가 사용됩니다.
- 일치하는 항목이 없으면 직접 연결을 시도합니다.

[도메인 및 부속 도메인의 프록시] 목록에서 다음 항목을 고려합니다.

```
com p1| host1 p2 | host2 | * p3
sesta.com p4 | host5 p5 | * p6
florizon.com | host6
abc.sesta.com p8 | host7 p7 | host8 p8 | * p9
host6.florizon.com p10
host9.sesta.com p11
siroe.com | host12 p12 | host13 p13 | host14 | * p14
siroe.com | host15 p15 | host16 | * p16
* p17
```

게이트웨이에서는 내부적으로 이러한 항목을 다음 표에 나와 있는 것처럼 테이블로 매핑합니다.

표 2–2 [도메인 및 부속 도메인의 프록시] 목록에서 항목 매핑

횟수	도메인 및 부속 도메인의 프록시 목록의 항목	프록시	설명
1	com	p1	목록에 지정된 대로
2	host1.com	p2	목록에 지정된 대로
3	host2.com	p1	host2에 대해 프록시가 지정되지 않았으므로 도메인의 프록시가 사용됩니다.
4	*.com	p3	목록에 지정된 대로
5	sesta.com	p4	목록에 지정된 대로
6	host5.sesta.com	p5	목록에 지정된 대로
7	*.sesta.com	p6	목록에 지정된 대로
8	florizon.com	직접	자세한 내용은 항목 14에 대한 설명 참조
9	host6.florizon.com	–	자세한 내용은 항목 14에 대한 설명 참조
10	abc.sesta.com	p8	목록에 지정된 대로
11	host7.abc.sesta.com	p7	목록에 지정된 대로
12	host8.abc.sesta.com	p8	목록에 지정된 대로
13	*.abc.sesta.com	p9	목록에 지정된 대로 `abc.sesta.com` 도메인에서 `host7`과 `host8`을 제외한 모든 호스트에는 `p9`가 프록시로 사용됩니다.
14	host6.florizon.com	p10	이 항목은 항목 9와 동일합니다. 그러나 항목 9는 직접 연결을 나타내지만, 이 항목은 프록시 p10을 사용해야 함을 나타냅니다. 이 경우와 같이 2개 항목이 있는 경우에는 프록시 정보가 있는 항목이 유효한 항목으로 간주됩니다. 다른 항목은 무시됩니다.
15	host9.sesta.com	p11	목록에 지정된 대로
16	siroe.com	직접	`siroe.com`에 대해 프록시가 지정되지 않았으므로 직접 연결을 시도합니다.
17	host12.siroe.com	p12	목록에 지정된 대로
18	host13.siroe.com	p13	목록에 지정된 대로
19	host14.siroe.com	직접	`host14`에 대해 프록시가 지정되지 않았으므로 직접 연결을 시도합니다.
20	*.siroe.com	p14	항목 23에 대한 설명 참조
21	host15.siroe.com	p15	목록에 지정된 대로

표 2–2 [도메인 및 부속 도메인의 프록시] 목록에서 항목 매핑 *(계속)*

횟수	도메인 및 부속 도메인의 프록시 목록의 항목	프록시	설명
22	host16.siroe.com	직접	host16 또는 siroe.com에 대해 프록시가 지정되지 않았으므로 직접 연결을 시도합니다.
23	*.siroe.com	p16	이 항목은 20번 항목과 비슷하지만 지정된 프록시가 다릅니다. 이런 경우 게이트웨이의 정확한 동작은 알 수 없습니다. 두 프록시 중 하나가 사용됩니다.
24	*	p17	요청된 URL과 일치하는 다른 항목이 없으면 p17이 프록시로 사용됩니다.

참고 – [도메인 및 부속 도메인의 프록시] 목록에서 프록시 항목을 | 기호로 분리하는 대신 목록에서 개별 항목을 별도의 행으로 지정할 수 있습니다. 예를 들어, 다음과 같은 항목 대신에

```
sesta.com p1 | red p2 | * p3
```

이 정보를 다음과 같이 지정할 수 있습니다.

```
sesta.com p1
red.sesta.com p2
*.sesta.com p3
```

이 목록 형식을 사용하면 반복되는 항목이나 기타 모호한 부분을 쉽게 파악할 수 있습니다.

도메인 및 부속 도메인의 프록시 목록에 기반하여 다시 쓰기

[도메인 및 부속 도메인의 프록시] 목록의 항목도 Rewriter에서 사용됩니다. Rewriter는 도메인이 [도메인 및 부속 도메인의 프록시] 목록에 나열된 도메인과 일치하는 모든 URL을 다시 씁니다.

주의 – [도메인 및 하위 도메인의 프록시] 목록의 * 항목은 다시 쓰기에 고려되지 않습니다. 예를 들어 24번 항목은 고려 대상이 되지 않습니다.

Rewriter에 대한 자세한 정보는 4 장을 참조하십시오.

기본 도메인 및 부속 도메인

URL의 대상 호스트가 정규 호스트 이름이 아닐 경우, 정규 이름에 도달하도록 기본 도메인 및 부속 도메인을 사용합니다.

관리 콘솔의 [기본 도메인] 필드 항목이 다음과 같다고 가정해 보겠습니다.

```
red.sesta.com
```

주 – [도메인 및 부속 도메인의 프록시] 목록에 해당하는 항목이 있어야 합니다.

위의 예에서는 `sesta.com`이 기본 도메인이고 기본 하위 도메인은 `red`입니다.

요청된 URL이 `host1`인 경우, 이 항목은 기본 도메인 및 하위 도메인을 통해 `host1.red.sesta.com`으로 결정됩니다. 그런 다음 [도메인 및 하위 도메인의 프록시] 목록에 `host1.red.sesta.com`이 있는지 확인합니다.

자동 프록시 구성 사용

[도메인 및 부속 도메인의 프록시] 목록에 있는 정보를 무시하려면 자동 프록시 구성(PAC) 기능을 활성화합니다.

자동 프록시 구성(PAC) 파일을 사용하는 경우

- Portal Server, Gateway, Netlet 및 Proxylet은 *Rhino* 소프트웨어를 사용하여 PAC 파일을 구문 분석합니다. `SUNWrhino` 패키지는 Java Enterprise System Accessory CD에서 설치할 수 있습니다.

  이 패키지에는 `/usr/share/lib` 디렉토리에 반드시 있어야 하는 `js.jar` 파일이 포함되어 있습니다. 이 디렉토리를 게이트웨이 및 Portal Server 시스템의 `webserver/appserver` 클래스 경로에 추가합니다. 그렇지 않으면 Portal Server, Gateway, Netlet 및 Proxylet에서 PAC 파일을 구문 분석할 수 없습니다.
- `js.jar`은 게이트웨이 컴퓨터의 `$JRE_HOME/lib/ext` 디렉토리에 있어야 합니다. 그렇지 않으면 게이트웨이에서 PAC 파일의 구문을 분석할 수 없습니다.
- 게이트웨이는 부팅 시에 게이트웨이 프로필 [자동 프록시 구성 파일] 위치 필드에 지정된 위치로부터 PAC 파일을 불러옵니다.
- 게이트웨이는 URLConnection API를 사용하여 이 위치에 도달합니다. 프록시가 게이트웨이에 도달하도록 구성해야 하는 경우에는 프록시를 다음과 같이 구성해야 합니다.

  1. 명령줄에서 다음 파일을 편집합니다.

     `/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*
  2. 다음 항목을 추가합니다.

     `http.proxyHost=` *web-proxy-hostname*

     `http.proxyPort=` *web-proxy-port*

     `http.proxySet=true`
  3. 게이트웨이를 다시 시작하여 지정된 프록시를 사용합니다.

     `./psadmin start-sra-instance –u amadmin – f` <*password file*> `–N` <*profile name*>`–t` <*gateway*>

- PAC 파일 초기화가 실패하면 게이트웨이는 [도메인 및 부속 도메인의 프록시] 목록에 있는 정보를 사용합니다.
- PAC 파일로부터 "" (빈 문자열) 이나 "null"이 반환되면 게이트웨이에서는 호스트가 인트라넷에 속하지 않는 것으로 가정합니다. 이는 호스트가 [도메인 및 부속 도메인의 프록시] 목록에 있지 경우와 비슷합니다.

  게이트웨이에서 호스트에 직접 연결되도록 하려면 "DIRECT"를 반환합니다. 47 페이지 “DIRECT 또는 NULL이 반환되는 예제”를 참조하십시오.
- 여러 프록시가 지정되어 있으면 게이트웨이는 첫 번째 반환된 프록시만 사용합니다. 호스트에 지정된 여러 프록시에서 페일오버나 로드 균형 조정을 시도하지 않습니다.
- 게이트웨이는 SOCKS 프록시를 무시하고 직접 연결을 시도하면서 호스트가 인트라넷의 일부라 가정합니다.
- 인트라넷의 일부가 아닌 호스트에 도달하는 데 프록시를 사용하도록 지정하려면 프록시 유형 STARPROXY를 사용합니다. 이 프록시 유형은 PAC 파일 형식의 확장이며 게이트웨이 프로필에 있는 [도메인 및 하위 도메인의 프록시] 부분의 `* proxyHost:port` 항목과 유사합니다. 48 페이지 “STARPROXY가 반환되는 예제”를 참조하십시오.

예제 PAC 파일 사용

다음 예제는 [도메인 및 부속 도메인의 프록시] 목록과 해당하는 PAC 파일에 나열된 URL을 보여줍니다.

DIRECT 또는 NULL이 반환되는 예제

도메인 및 하위 도메인에 사용되는 프록시:

```
*intranet1.com proxy.intranet.com:8080

intranet2.com proxy.intranet1.com:8080
```

해당하는 PAC 파일:

```
// Start of the PAC File
function FindProxyForURL(url, host) {
         if (dnsDomainIs(host, ".intranet1.com")) {
             return "DIRECT";
         }
          if (dnsDomainIs(host, ".intranet2.com")) {
              return "PROXY proxy.intranet1.com:8080";
          }
          return "NULL";
}
//End of the PAC File
```

STARPROXY가 반환되는 예제

도메인 및 하위 도메인에 사용되는 프록시:

```
intranet1.com

intranet2.com.proxy.intranet1.com:8080

internetproxy.intranet1.com:80
```

해당하는 PAC 파일:

```
// Start of the PAC File
function FindProxyForURL(url, host) {
          if (dnsDomainIs(host, ".intranet1.com")) {
              return "DIRECT";
          }
          if (dnsDomainIs(host, ".intranet2.com")) {
              return "PROXY proxy.intranet1.com:8080;" +
                  "PROXY proxy1.intranet1.com:8080";
          }
          return "STARPROXY internetproxy.intranet1.com:80";
}
//End of the PAC File
```

이 경우 요청이 `.intranet2.com` 도메인에 있는 호스트에 대한 것이면 게이트웨이는 `proxy.intranet1.com:8080`에 접속합니다. `proxy.intranet1.com:8080`이 다운되면 요청이 실패합니다. 게이트웨이는 페일오버하지 않고 `proxy1.intranet1.com:8080`에 접속합니다.

PAC 파일 위치 지정

PAC 파일의 위치를 지정하는 형식은 다음과 같이 해당 위치에 따라 다릅니다.

- PAC 파일이 웹 서버에 상주하는 경우 다음과 같이 PAC URL을 입력합니다.

  `http://`hostname`/`*pacfile_name*`.pac`

- PAC 파일이 로컬 파일(예: `c:\\pacfile\\sample.pac`)인 경우 java 1.4.1_x에 대해 다음과 같이 PAC URL을 입력합니다.

  `file://c:/pacfile/sample.pac`

- PAC 파일이 로컬 파일(예: `c:\\pacfile\\sample.pac`)인 경우 java 1.4.2_x에 대해 다음과 같이 PAC URL을 입력합니다.

  `file:///c:/pacfile/sample.pac`

별도 세션에서 서비스 추가

Portal Server 서비스를 별도 세션에서 추가할 경우

- Portal Server가 Portal Server 관리 콘솔의 [게이트웨이] > [핵심] 아래에 나열됩니다.
- Portal Server URL은 [게이트웨이] > [보안] 아래의 비인증 URL에 나열됩니다.

Netlet 프록시 사용

Netlet 패킷은 게이트웨이에서 비밀번호가 해독되어 대상 서버로 보내집니다. 그러나 게이트웨이는 비무장 지대(DMZ)와 인트라넷 사이의 방화벽을 통해 모든 Netlet 대상 호스트에 액세스해야 합니다. 따라서 이렇게 설정하면 방화벽에서 많은 포트를 열어야 합니다. Netlet 프록시는 방화벽에서 열린 포트의 수를 줄이는 데 사용할 수 있습니다.

Netlet 프록시는 클라이언트로부터 게이트웨이를 거쳐 인트라넷에 상주하는 Netlet 프록시에 이르기까지 보안 터널을 확장하여 게이트웨이와 인트라넷 사이의 보안을 강화합니다. 프록시가 있으면 Netlet 패킷은 프록시에서 해독된 후 대상으로 보내집니다.

Netlet 프록시를 사용하면 다음과 같은 장점이 있습니다.

- 보안 계층을 추가할 수 있습니다.
- 상당한 규모의 배치 환경에서 내부 방화벽을 통해 추가 IP 주소와 게이트웨이의 포트 사용을 최대한 줄일 수 있습니다.
- 게이트웨이와 Portal Server 간 개방 포트 수를 1로 제한할 수 있습니다. 이 포트 수는 설치 시 구성 가능합니다.
- 클라이언트와 게이트웨이 사이의 보안 채널을 49 페이지 “Netlet 프록시 사용”의 "Netlet 프록시가 구성되어 있는 경우" 부분에 나와 있듯이 Portal Server까지 확장할 수 있습니다. Netlet 프록시는 데이터 암호화를 통해 보안을 강화한다는 이점이 있지만 시스템 자원을 더 많이 사용할 수 있습니다. Netlet 프록시 설치에 대한 자세한 내용은 Sun Java System 설치 설명서를 참조하십시오.

다음과 같은 작업을 수행할 수 있습니다.

- Portal Server 노드나 별도 노드에 Netlet 프록시를 설치할 수 있습니다.
- 다중 Netlet 프록시를 설치하고 관리 콘솔을 사용하여 단일 게이트웨이에 구성할 수 있습니다. 로드 균형 조절에 유용합니다.
- 단일 시스템에 Netlet 프록시의 다중 인스턴스를 구성할 수 있습니다.
- 게이트웨이의 다중 인스턴스를 Netlet 프록시의 단일 설치에 지정할 수 있습니다.
- 웹 프록시를 통해 Netlet을 통과할 수 있습니다.

Netlet 프록시가 설치된 경우와 설치되지 않은 경우, 게이트웨이와 Portal Server를 구현하는 3가지 구현 샘플이 나와 있습니다. 구성 요소에는 클라이언트, 방화벽 2개, 두 방화벽 사이에 상주하는 게이트웨이, Portal Server 및 Netlet 대상 서버가 포함됩니다.

첫 번째 시나리오는 Netlet 프록시가 설치되지 않은 경우의 게이트웨이와 Portal Server를 보여줍니다. 데이터 암호화는 클라이언트에서 게이트웨이까지만 적용됩니다. 각 Netlet 연결 요청을 위해 두 번째 방화벽에서 포트가 1개 개방되어 있습니다.

두 번째 시나리오는 Netlet 프록시가 Portal Server에 설치된 경우의 게이트웨이와 Portal Server를 보여줍니다. 데이터 암호화는 클라이언트에서 Portal Server까지 전체적으로 적용됩니다. 모든 Netlet 연결이 Netlet 프록시를 통해 라우팅되기 때문에 두 번째 방화벽에서 Netlet 요청에 사용되는 포트는 하나만 열려 있으면 됩니다.

세 번째 시나리오는 Netlet 프록시가 별도 노드에 설치된 경우의 게이트웨이와 Portal Server를 보여줍니다. Netlet 프록시를 별도 노드에 설치하면 Portal Server 노드의 로드가 줄어듭니다. 여기서는 두 번째 방화벽에서 2개의 포트만 개방되어 있으면 됩니다. 한 포트는 Portal Server에 대한 요청을 처리하고 다른 포트는 Netlet 프록시 서버에 대한 Netlet 요청을 라우팅합니다.

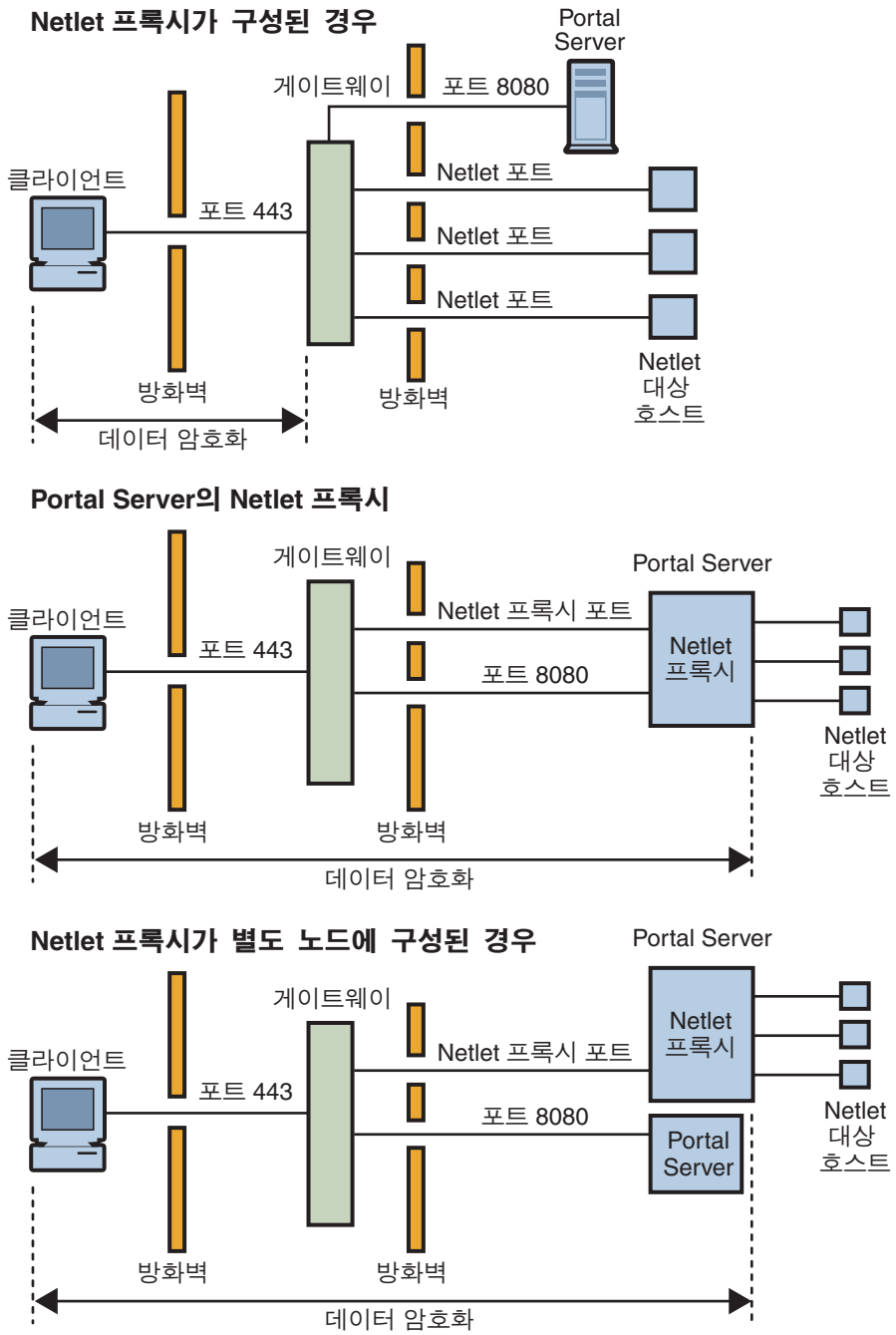


그림 2–1 Netlet 프록시 구현

Netlet 프록시 활성화

Portal Server 관리 콘솔을 사용하여 게이트웨이 서비스를 통해 Netlet 프록시를 사용할 수 있습니다.

Netlet 프록시 다시 시작

프록시가 예기치 않게 중지될 때마다 다시 시작하도록 Netlet 프록시를 구성할 수 있습니다. Netlet 프록시를 모니터링하고 다운된 경우 다시 시작하도록 워치독 프로세스를 예약할 수 있습니다.

Netlet 프록시를 수동으로 다시 시작할 수도 있습니다. 자세한 단계는 246 페이지 “Netlet 프록시를 다시 시작하려면”을 참조하십시오.

Netlet 프록시 워치독을 구성하려면

워치독이 Netlet 프록시의 상태를 모니터하는 시간 간격을 구성할 수 있습니다. 시간 간격은 기본적으로 60초로 설정됩니다. 이 간격을 변경하려면 다음 행을 `crontab` 파일에 추가합니다.

0-59 * * * * *netlet-install-dir*`/bin/checkgw /var/opt/SUNWportal/.gw 5 > /dev/null 2>&1`

주 – 워치독을 시작 또는 중지하려면 `./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off` 명령을 실행합니다.

Rewriter 프록시 사용

Rewriter 프록시는 인트라넷에 설치됩니다. 게이트웨이는 컨텐트를 직접 검색하는 대신, 컨텐트를 가져와 게이트웨이로 반환하는 Rewriter 프록시로 모든 요청을 전달합니다.

Rewriter 프록시를 사용하면 다음과 같은 장점이 있습니다.

- 게이트웨이와 서버 사이에 방화벽이 있는 경우 방화벽에서는 포트를 2개만 열면 됩니다. 하나는 게이트웨이와 Rewriter 프록시 사이의 포트이고 다른 하나는 게이트웨이와 Portal Server 사이의 포트입니다.
- 대상 서버가 HTTP 프로토콜만 지원하고 HTTPS는 지원하지 않아도 게이트웨이와 인트라넷 사이의 HTTP 트래픽이 안정적으로 됩니다.

Rewriter 프록시를 지정하지 않으면 사용자가 인트라넷 컴퓨터에 액세스하려고 할 때 게이트웨이 구성 요소에서 인트라넷 컴퓨터에 직접 연결합니다.

Rewriter 프록시를 로드 조정기로 사용하는 경우에는 Rewriter의 `platform.conf.instance_name`이 로드 조정기 URL을 가리키는지 확인해야 합니다. 또한 [Portal Servers] 목록에서 로드 밸런서 호스트를 지정하십시오.

각 게이트웨이 인스턴스에 대해 여러 Rewriter 프록시 인스턴스가 있는 경우(포털 노드에서는 필요하지 않음) `platform.conf` 파일에서 Rewriter 프록시의 단일 포트 항목이 아니라 *host-name*:*port* 형식으로 각 Rewriter 프록시의 세부 사항을 입력합니다.

Rewriter 프록시의 인스턴스 만들기

`rwpmultiinstance` 스크립트를 사용하여 Portal Server 노드에 Rewriter 프록시의 새 인스턴스를 만듭니다. 게이트웨이 프로필을 만든 후에 이 스크립트를 실행하십시오.

247 페이지 "Rewriter 프록시 인스턴스를 만들려면"을 참조하십시오.

Rewriter 프록시 활성화

Access Manager 관리 콘솔의 SRA 구성에서 게이트웨이 서비스를 통해 Rewriter 프록시를 활성화합니다.

Rewriter 프록시 다시 시작

프록시가 예기치 않게 중지될 때마다 다시 시작하도록 Rewriter 프록시를 구성할 수 있습니다. 이런 문제가 발생하는지 모니터링하고 문제가 발생하면 다시 시작하도록 워치독 프로세스를 예약할 수 있습니다.

Rewriter 프록시를 수동으로 다시 시작할 수도 있습니다.

247 페이지 "Rewriter 프록시를 다시 시작하려면"을 참조하십시오.

Rewriter 프록시 워치독 구성

워치독이 Rewriter 프록시 상태를 모니터링하는 시간 간격을 구성할 수 있습니다. 시간 간격은 기본적으로 60초로 설정됩니다. 시간 간격을 변경하려면 `crontab` 파일에 다음 행을 추가하십시오.

0-59 * * * * *rewriter-proxy-install-root* `/bin/checkgw /var/opt/SUNWportal/.gw 5 > /dev/null 2>&1`

주 – 워치독을 시작 또는 중지하려면 `./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off` 명령을 실행합니다.

게이트웨이에서 역방향 프록시 사용

프록시 서버는 인트라넷에 인터넷 컨텐트를 서비스하고 역 프록시는 인터넷에 인트라넷 컨텐트를 서비스합니다. 로드 균형 조절 및 캐싱을 수행하도록 역방향 프록시의 배포를 구성할 수 있습니다.

이 배포에서 게이트웨이 전방에 타사의 역 프록시가 사용된다면 게이트웨이의 URL 대신 역 프록시의 URL로 응답을 다시 써야 합니다. 이를 위해 다음 구성이 필요합니다.

248 페이지 “역 프록시를 활성화하려면”을 참조하십시오.

클라이언트 정보 가져오기

게이트웨이가 클라이언트 요청을 내부 서버로 전달하면 HTTP 헤더가 HTTP 요청에 추가됩니다. 이 헤더를 사용하여 추가 클라이언트 정보를 가져오고 게이트웨이가 있는지 감지할 수 있습니다.

HTTP 요청 헤더를 보려면 `platform.conf` 파일에서 해당 항목을 `gateway.error=message`로 설정합니다. 그런 다음 서블릿 API에서 `request.getHeader()`를 사용하십시오. 다음 표에는 HTTP 헤더에 있는 정보가 나열되어 있습니다.

표 2–3 HTTP 헤더의 정보

헤더	구문	설명
PS-GW-PDC	`X-PS-GW- PDC: true/false`	게이트웨이에서 PDC의 사용 가능 여부를 나타냅니다.
PS-Netlet	`X-PS-Netlet:enabled=true/false`	게이트웨이에서 Netlet의 사용 가능 여부를 나타냅니다. Netlet이 활성화된 경우 암호화 옵션이 채워져서 게이트웨이가 HTTPS(`encryption=ssl`) 또는 HTTP 모드(`encryption=plain`) 중 어느 쪽에서 실행 중인지 보여줍니다. 예: ■ `PS-Netlet: enabled=false` Netlet이 사용 불가능 상태입니다. ■ `PS-Netlet: enabled=true; encryption=ssl` 게이트웨이가 SSL 모드에서 실행되며 Netlet이 활성화되었습니다. Netlet이 활성화되어 있지 않으면 `encryption=ssl` 또는 `encryption=plain`이 채워지지 않습니다.

표 2–3 HTTP 헤더의 정보 *(계속)*

헤더	구문	설명
PS-GW-URL	`X-PS-GW-URL: http(s)://gatewayURL(:port)`	클라이언트가 연결된 URL을 나타냅니다. 포트가 표준이 아닌 경우, 예를 들어 게이트웨이가 HTTP/HTTPS 모드이고 포트는 80/443이 아닌 경우 `:port`도 채워집니다.
PS-GW-Rewriting-URL	`X-PS-GW-URL: http(s)://gatewayURL(:port)/[SessionInfo]`	게이트웨이가 모든 페이지를 다시 쓰는 URL을 나타냅니다. 1. 브라우저에서 쿠키를 지원하는 경우 이 헤더 값은 PS-GW-URL 헤더와 같습니다. 2. 브라우저가 쿠키를 지원하지 않고 ■ [사용자 세션 쿠키가 전달될 사용자 세션] 필드에 대상 호스트가 있으면 값은 게이트웨이가 페이지를 쓰는 실제 URL이 됩니다(암호화된 세션 아이디 정보 포함). ■ 또는 [사용자 세션 쿠키가 전달될 사용자 세션] 필드에 대상 호스트가 없으면 `SessionInfo` 문자열은 `$SessionID`가 됩니다. **주 –** 응답의 일부로 사용자의 Access Manager sessionId가 변경되면(인증 페이지에서 오는 응답과 같이) 페이지는 이전에 헤더에 표시된 값이 아닌 그 값으로 다시 쓰여집니다. 예: ■ 브라우저에서 쿠키를 지원하는 경우 PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/ ■ 브라우저에서 쿠키를 지원하지 않지만 [사용자 세션 쿠키가 전달될 사용자 세션] 필드에 endserver가 있는 경우 PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/SessIDValCustomEncodedValue/ ■ 브라우저에서 쿠키를 지원하지 않고 [사용자 세션 쿠키가 전달될 사용자 세션] 필드에 endserver가 없는 경우 PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/$SessionID
PS-GW-CLientIP	`X-PS-GW-CLientIP:` *IP*	게이트웨이가 `recievedSocket.getInetAddress().getHostAddress()`로부터 가져온 IP를 나타냅니다. 게이트웨이에 직접 연결된 경우 이 값은 클라이언트의 IP를 제공합니다.

인증 체이닝 사용

인증 체이닝은 인증의 일반 메커니즘보다 높은 수준으로 보안을 강화합니다. 사용자가 2개 이상 인증 메커니즘에 대해 인증 받도록 설정할 수 있습니다.

여기에 설명된 절차는 게이트웨이에서 개인 디지털 인증서(PDC) 인증과 함께 인증 체이닝을 사용하는 경우에만 적용됩니다. 게이트웨이에 PDC 인증이 없는 인증 체이닝에 대한 자세한 내용은 **Access Manager 관리 설명서**를 참조하십시오.

예를 들어, PDC 및 Radius 인증 모듈을 체이닝한 경우에는 사용자가 표준 포털 데스크탑에 액세스하려면 이 3개 모듈에 대한 인증을 모두 거쳐야 합니다.

자세한 단계는 248 페이지 “기존 PDC 인스턴스에 인증 모듈을 추가하려면 ”을 참조하십시오.

주 – 활성화된 경우 PDC는 사용자에게 항상 가정 먼저 제시되는 인증 모듈입니다.

와일드카드 인증 사용

와일드카드 인증에서는 정규 DNS 호스트 이름에 와일드카드 문자가 있는 단일 인증을 수락합니다.

인증서가 있으면 동일한 도메인 내의 여러 호스트가 보호됩니다. 예를 들어, `*.domain.com`에 대한 인증을 `abc.domain.com` 및 `abc1.domain.com`에 사용할 수 있습니다. 이 인증은 `domain.com` 도메인에 있는 모든 호스트에 유효합니다.

브라우저 캐싱 사용 불가능

게이트웨이 구성 요소는 웹 브라우저를 사용하여 어느 위치에서나 백엔드 기업 데이터에 안전하게 액세스하므로 클라이언트가 정보를 로컬로 캐싱할 필요가 없습니다.

특정 게이트웨이의 `platform.conf` 파일에 있는 속성을 수정하여 게이트웨이를 통해 리디렉션된 페이지의 캐싱을 비활성화할 수 있습니다.

이 옵션을 비활성화하면 게이트웨이 성능에 영향을 줄 수 있습니다. 표준 포털 데스크탑을 새로 고칠 때마다 게이트웨이는 브라우저에서 이전에 캐싱한 이미지와 같이 페이지에서 참조되는 모든 항목을 검색해야 합니다. 그러나 이 기능을 사용하면 원격으로 액세스한 보안 컨텐트를 캐싱한 자취가 클라이언트 사이트에 남지 않습니다. 인터넷 카페 또는 기업 IT 제어를 받지 않는 유사한 원격 장소로부터 기업 네트워크에 액세스하는 경우 이 기능은 비중이 있을 수 있습니다.

249 페이지 “브라우저 캐싱을 비활성화하려면”을 참조하십시오.

게이트웨이 서비스 사용자 인터페이스 사용자 정의

이 절에서는 편집할 수 있는 여러 게이트웨이 등록 정보 파일에 대해 설명합니다.

srapGateway.properties 파일 수정

다음과 같은 목적으로 이 파일을 편집할 수 있습니다.

- 게이트웨이 실행 중에 나타날 수 있는 오류 메시지를 사용자 정의할 때
  - HTML-CharSets=ISO-8859-1은 이 파일을 만드는 데 사용되는 문자 집합을 지정합니다.
  - 중괄호 안의 숫자(예: {0})는 값이 런타임으로 표시된다는 것을 뜻합니다. 필요에 따라 이 숫자와 연관된 레이블을 변경하거나 레이블을 재배열할 수 있습니다. 레이블 숫자와 오류는 연관되어 있기 때문에 레이블에 해당하는 표시될 메시지가 있어야 합니다.

  로그 정보를 사용자 정의할 때.

  기본적으로 `srapGateway.properties` 파일은 *portal-server-install-root* `/SUNWportal/locale` 디렉토리에 있습니다. 게이트웨이 컴퓨터에 표시되는 모든 메시지는 해당 메시지의 언어와 상관 없이 이 파일에 있습니다.

  클라이언트 표준 포털 데스크탑에 나타나는 메시지의 언어를 변경하려면 이 파일을 각 로켈 디렉토리로 복사합니다(예: *portal-server-install-root* `/SUNWportal/locale_en_US`).

srapgwadminmsg.properties 파일 수정

다음과 같은 이유로 이 파일을 편집할 수 있습니다.

- 관리 콘솔의 게이트웨이 서비스에 대한 버튼에 나타나는 레이블을 사용자 정의할 때
- 게이트웨이를 구성하는 중에 나타나는 상태 메시지 및 오류 메시지를 사용자 정의할 때

LDAP 디렉토리 공유

Portal Server 및 Access Manager 서버의 두 인스턴스가 같은 LDAP 디렉토리를 공유하는 경우 모든 후속 Portal Server, Access Manager 및 게이트웨이 인스턴스에서 같은 LDAP 디렉토리를 공유합니다. 249 페이지 “LDAP 디렉토리를 공유하려면”을 참조하십시오.

◆ ◆ ◆ 3 장

3 Proxylet 작업

이 장에서는 사용자가 웹 페이지를 구문 분석하지 않고도 게이트웨이를 통해 인트라넷 웹 페이지에 액세스할 수 있게 해주는 Proxylet에 대해 설명합니다.

Proxylet 작업

Proxylet 개요

Proxylet은 클라이언트 시스템에서 자체를 프록시 서버로 설정하는 Java 애플릿입니다. Proxylet은 클라이언트 시스템에 있는 프록시 자동 구성(PAC) 파일의 프록시 설정을 읽고 프록시 설정이 로컬 프록시 서버 또는 Proxylet을 가리키도록 수정합니다.

Proxylet은 게이트웨이에서 전송 모드를 상속합니다. 게이트웨이가 SSL에서 실행되도록 구성되어 있으면 Proxylet은 클라이언트 시스템과 게이트웨이 또는 대상 서버 사이에 보안 채널을 구성합니다. 암호화를 위해, 클라이언트 JVM이 1.4 이상이거나 필요한 jar 파일이 클라이언트 시스템에 상주하는 경우에는 Proxylet에서 JSSE API를 사용합니다. 그렇지 않으면 KSSL API를 사용합니다. 해독 작업은 클라이언트 컴퓨터에서 수행됩니다.

게이트웨이로 리디렉션되는 URL의 도메인 및 부속 도메인은 게이트웨이 프로필에서 지정됩니다. URL이 게이트웨이가 처리하는 도메인에 속하지 않으면 요청은 인터넷으로 지정됩니다. 게이트웨이 프로필에 특정 URL 도메인이 나열되어 있으면 Proxylet은 게이트웨이를 가리키도록 클라이언트 프록시 설정을 재설정합니다.

Proxylet은 PDC(Personal Digital Certificate)가 게이트웨이에서 활성화된 경우 클라이언트용 인증을 지원합니다. PDC 사용 여부를 확인하려면 54 페이지 “클라이언트 정보 가져오기”를 참조하십시오.

Proxylet은 클라이언트 IP 주소 또는 프록시 호스트 이름과 포트를 지정한 Portal Server 관리 콘솔에서 활성화됩니다. Proxylet을 활성화하면 클라이언트 시스템에서 다음 정보가 확인됩니다.

- 적절한 브라우저 권한
- 브라우저가 IE 6.0 sp2, IE 7 및 Firefox 2.0인지 여부
- 시스템 또는 장치에서 서버 응용 프로그램을 실행할 수 있는지 여부

모든 요구 사항이 만족되면 애플릿이 클라이언트 시스템으로 다운로드되어 실행됩니다. 클라이언트에 JRE 1.4.2 이상이 설치되어 있지 않은 경우 인터넷에 연결되어 있고 관리자 권한이 있으면 Proxylet과 함께 JRE가 자동으로 다운로드됩니다.

Proxylet이 사용되는 경우 Proxylet은 PAC(Proxy Auto Configuration) 또는 프록시 구성 목록에서 프록시 설정을 검색합니다.

주 – Proxylet 애플릿을 사용하는 경우 브라우저의 팝업 차단 기능을 비활성화해야 합니다.

HTTPS 지원

Proxylet은 HTTPS를 지원하며 다음 결과가 나타납니다.

- 해독 기능은 클라이언트 서버에서 수행됩니다.
- 대상 서버는 SSL 모드에서 실행하는 경우 액세스할 수 있습니다.
- 클라이언트 인증서는 대상 서버에 직접 제공됩니다.
- 기본 인증 단일 사인온(SSO)은 게이트웨이에서 지원되지 않습니다. (게이트웨이에서 HTTP 헤더에 SSO 정보를 삽입할 수 없습니다.)
- URL 기반 액세스 제어는 지원되지 않으며 호스트 기반 액세스 제어만 지원됩니다.
- 게이트웨이 앞의 외부 가속기와 외부 역 프록시는 현재 지원되지 않습니다.

주 – 이 지원은 Portal Server가 HTTPS를 사용하는 경우 Proxylet을 지원하기 위한 것이 아닙니다.

Proxylet 사용의 이점

Rewriter와 달리 Proxylet은 설치 후에 조금만 변경하거나 변경하지 않아도 됩니다. Microsoft Exchange Server 등의 타사 소프트웨어와 쉽게 통합할 수 있습니다. 또한 Proxylet에서 웹 컨텐트에 대한 작업을 수행하지 않으므로 게이트웨이의 성능이 향상됩니다. Proxylet에서 컨텐트를 수정하거나 데이터를 변경하지 않기 때문에 사용자는 `tar` 및 `gzip` 파일과 같은 컨텐트 유형을 다운로드할 수 있습니다.

Proxylet 구성

Proxylet 사용 및 구성에 대한 자세한 내용은 13 장을 참조하십시오.

주 – 사용자가 Proxylet을 실행할 수 있는 JVM(Java Virtual Machine)을 가지고 있지 않으면 브라우저에서 Sun 웹 사이트에 연결하여 JRE(Java Runtime Environment)를 다운로드합니다. 사용자의 브라우저 설정에 정확한 값이 포함되어 있지 않거나 사용자가 인터넷에 액세스하지 않고 직접 프록시 설정을 사용하는 경우 Proxylet을 다운로드할 수 없습니다.

◆ ◆ ◆ 4 장

4 Rewriter 작업

Secure Remote Access의 Rewriter 구성 요소를 사용하면 사용자가 인트라넷 웹 페이지를 분석하여 게이트웨이를 통해 해당 웹 페이지에 액세스할 수 있습니다.

이 장에서는 다음 주제를 다룹니다.

Rewriter 소개

SRA(Secure Remote Access)의 Rewriter 구성 요소를 통해 최종 사용자가 게이트웨이를 가리키도록 웹 페이지의 URI(Uniform Resource Identifier)를 수정하여 인트라넷을 찾아볼 수 있습니다. URI는 등록된 이름 공간에서 이름을 캡슐화하고 여기에 이름 공간으로 레이블을 설정하는 방법을 정의합니다. URI의 가장 일반적인 유형으로는 URL(Uniform Resource Locator)이 있습니다. Rewriter는 HTTP 또는 HTTPS만 지원합니다. 이러한 지원은 프로토콜에서 대소문자를 구분하지 않고 이루어집니다. Rewriter는 상대 URL의 일부인 경우에만 백슬래시를 지원합니다.

예 4–1 URL 다시 쓰기

`http://abc.sesta.com\\index.html`이 다시 작성됩니다.

다시 작성되지 않는 URL: `http:\\\\abc.sesta.com`. *http:/abc.com*

문자 집합 암호화

HTTP 표준에 따르면 HTTP 헤더 또는 HTML 메타 태그에서 웹 페이지에 사용할 문자 집합을 지정해야 합니다. 하지만 이 정보를 사용할 수 없는 경우도 있습니다. 데이터의 암호화를 설정하고 만든 사람의 의도에 맞게 데이터를 표시하려면 문자 집합을 알아야 합니다.

문자 집합을 검색하려면 `SUNWjchdt` 패키지를 Java Enterprise System Accessory CD에서 설치하십시오. 이 제품이 설치되면 Rewriter에서 필요한 경우 이를 검색하여 사용합니다.

주 – 이 제품을 사용하면 성능이 저하될 수 있기 때문에 필요한 경우에만 설치해야 합니다. 설치, 구성 및 사용에 관한 자세한 내용은 `jcharset_readme.txt`를 참조하십시오.

Rewriter 사용 시나리오

사용자가 게이트웨이를 통해 인트라넷 웹 페이지에 액세스하려고 할 때 Rewriter가 웹 페이지를 사용할 수 있도록 해줍니다. Rewriter는 URLScraper 및 게이트웨이에서 사용됩니다.

URLScraper

URL Scraper 공급자는 구성된 URI에서 컨텐트를 가져오며, 이러한 URI를 브라우저로 보내기 전에 모든 상대 URI를 절대 URI로 확장합니다.

예를 들어 다음과 같이 사이트에 액세스하는 경우

```
<a href="../mypage.html">
```

Rewriter는 이것을 다음으로 변환합니다.

```
<a href="http://yahoo.com/mypage.html">
```

여기서 `http://yahoo.com/test/`는 페이지의 기본 URL입니다.

URLScraper 공급자에 대한 자세한 내용은 **Sun Java System Portal Server 관리 설명서**를 참조하십시오.

게이트웨이

게이트웨이는 인터넷 포털에서 컨텐트를 가져옵니다. 게이트웨이에서 컨텐트를 브라우저로 보내기 전에 게이트웨이 URI를 기존 URI 앞에 추가하므로 브라우저의 후속 URI 요청이 해당 게이트웨이에 도달할 수 있습니다.

예를 들어, 다음과 같은 인터넷 컴퓨터의 HTML 페이지에 액세스하려고 하는 사용자에 대해

```
<a href="http://mymachine.intranet.com/mypage.html>"
```

Rewriter는 이 URL에 다음과 같이 게이트웨이에 대한 참조를 갖는 URL 이름을 접두어로 붙입니다.

```
<a href="https://gateway.company.com/http://mymachine.intranet.com/ mypage.html>"
```

사용자가 이 앵커와 관련된 링크를 누르면 브라우저가 게이트웨이에 접속합니다. 게이트웨이는 `mymachine.intranet.com`에서 `mypage.html`의 컨텐트를 가져옵니다.

게이트웨이는 가져온 웹 페이지에서 다시 작성할 요소를 결정하기 위해 몇 가지 규칙을 사용합니다.

규칙 집합 작성

규칙 집합 정의에 대한 자세한 내용은 **Portal Server 관리 설명서**를 참조하십시오. 새 규칙 집합을 만든 후 필수 규칙을 정의해야 합니다.

이 부분에서는 다음 주제를 다룹니다.

공용 인터페이스(규칙 집합 DTD)

규칙 집합 DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
The following constraints are not represented in DTD, but taken care of programmatically
    1. In a Rule, All Mandatory attributes cannot be "*".
    2. Only one instance of the below elements is allowed, but in any order.
    1)HTMLRules
    2)JSRules
    3)XMLRules
    3. ID should always be in lower case.
-->
```

```
<!ENTITY % eURL 'URL'>
<!ENTITY % eEXPRESSION 'EXPRESSION'>
<!ENTITY % eDHTML 'DHTML'>
<!ENTITY % eDJS 'DJS'>
<!ENTITY % eSYSTEM 'SYSTEM'>

<!ENTITY % ruleSetElements '(HTMLRules | JSRules | XMLRules)?'>
<!ENTITY % htmlElements '(Form | Applet | Attribute)*'>
<!ENTITY % jsElements '(Variable | Function)*'>
<!ENTITY % xmlElements '(Attribute | TagText)*'>

<!ELEMENT RuleSet (%ruleSetElements;,%ruleSetElements;,%ruleSetElements;)>
<!ATTLIST RuleSet
    id ID #REQUIRED
    extends CDATA "none"
>

<!-- Rules for identifying rules in HTML content -->
<!ELEMENT HTMLRules (%htmlElements;)>
<!ELEMENT Form EMPTY>
<!ATTLIST Form
    name CDATA #REQUIRED
    field CDATA #REQUIRED
    valuePatterns CDATA ""
    source CDATA "*"
>

<!ELEMENT Applet EMPTY>
<!ATTLIST Applet
    code CDATA #REQUIRED
    param CDATA "*"
    valuePatterns CDATA ""
    source CDATA "*"
>

<!-- Rules for identifying rules in JS content -->
<!ELEMENT JSRules (%jsElements;)>
<!ELEMENT Variable EMPTY>
<!ATTLIST Variable
    name CDATA #REQUIRED
    type (%eURL; | %eEXPRESSION; | %eDHTML; | %eDJS; | %eSYSTEM;) "EXPRESSION"
    source CDATA "*"
>

<!ELEMENT Function EMPTY>
<!ATTLIST Function
    name CDATA #REQUIRED
    paramPatterns CDATA #REQUIRED
```

```
    type (%eURL; | %eEXPRESSION; | %eDHTML; | %eDJS;) "EXPRESSION"
    source CDATA "*"
>

<!-- Rules for identifying rules in XML content -->
<!ELEMENT XMLRules (%xmlElements;)>
<!ELEMENT TagText EMPTY>
<!ATTLIST TagText
    tag CDATA #REQUIRED
    attributePatterns CDATA ""
    source CDATA "*"
>

<!ELEMENT Attribute EMPTY>
<!ATTLIST Attribute
    name CDATA #REQUIRED
    tag CDATA "*"
    valuePatterns CDATA ""
    type (%eURL; | %eDHTML; | %eDJS; ) "URL"
    source CDATA "*"
>
```

주 – *를 규칙 값의 일부로 사용할 수 있지만 필수 속성 값에는 *만 사용할 수 없습니다. 이러한 규칙은 무시되지만 RuleSetInfo 로그 파일에 메시지가 기록됩니다. 이 로그 파일에 대한 자세한 내용은 95 페이지 “디버깅 파일 이름”을 참조하십시오.

예제 XML DTD

이 절에는 예제 규칙 집합이 들어 있습니다. 140페이지의 "사례 연구"를 통해 Rewriter에서 이러한 규칙을 해석하는 방식을 살펴볼 수 있습니다.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
Rules for integrating a mail client with the gateway.
-->
<!DOCTYPE RuleSet SYSTEM "jar://rewriter.jar/resources/RuleSet.dtd">
<RuleSet type="GROUPED" id="owa">
<HTMLRules>
<Attribute name="action" />
<Attribute name="background" />
<Attribute name="codebase" />
<Attribute name="href" />
<Attribute name="src" />
<Attribute name="lowsrc" />
<Attribute name="imagePath" />
<Attribute name="viewClass" />
<Attribute name="emptyURL" />
```

```
<Attribute name="draftsURL" />
<Attribute name="folderURL" />
<Attribute name="prevMonthImage" />
<Attribute name="nextMonthImage" />
<Attribute name="style" />
<Attribute name="content" tag="meta" />
</HTMLRules>
<JSRules>
<!-- Rules for Rewriting JavaScript variables in URLs -->
<Variable name="URL"> _fr.location </Variable>
<Variable name="URL"> g_szUserBase </Variable>
<Variable name="URL"> g_szPublicFolderUrl </Variable>
<Variable name="URL"> g_szExWebDir </Variable>
<Variable name="URL"> g_szViewClassURL </Variable>
<Variable name="URL"> g_szVirtualRoot </Variable>
<Variable name="URL"> g_szBaseURL </Variable>
<Variable name="URL"> g_szURL </Variable>
<Function name="EXPRESSION" name="NavigateTo" paramPatterns="y"/>
</JSRules>
<XMLRules>
<Attribute name="xmlns"/>
<Attribute name="href" tag="a"/>
<TagText tag="baseroot" />
<TagText tag="prop2" />
<TagText tag="prop1" />
<TagText tag="img" />
<TagText tag="xsl:attribute"
attributePatterns="name=src" />
</XMLRules>
</RuleSet>
```

규칙 작성을 위한 절차

규칙을 작성하는 일반적인 절차는 다음과 같습니다.

- 컨텐트를 다시 작성해야 하는 HTML 페이지가 있는 디렉토리를 확인합니다.
- 이 디렉토리에서 다시 작성해야 하는 페이지를 확인합니다.
- 각 페이지에서 다시 작성해야 하는 URL을 확인합니다. "http" 및 "/"를 검색하여 대부분의 URL이 간단하게 확인할 수 있습니다.
- URL의 컨텐트 유형 확인: HTML, JavaScript 또는 XML.
- Access Manager 관리 콘솔의 [Portal Server 구성] 아래에 있는 [Rewriter 서비스]에서 필요한 규칙 집합을 편집하여 이러한 각 URL을 다시 쓰기 위해 필요한 규칙을 작성합니다.
- 모든 규칙을 해당 도메인에 대한 하나의 규칙 집합으로 결합시킵니다.

규칙 집합 관련 지침

규칙 집합을 작성하는 경우 다음 사항을 주의하십시오.

- 특정 호스트의 우선 순위는 가장 긴 URI 대응부터 결정됩니다. 예를 들어, 다음 규칙 집합에서

  ```
  mail1.central.abc.com|iplanet_mail_ruleset
  *.sfbay.abc.com|sfbay_ruleset
  *.abc.com|generic_ruleset
  ```

  `sfbay_ruleset`이 가장 긴 대응이기 때문에 사용됩니다.
- 규칙 집합의 규칙들은 규칙이 특정 구문과 일치할 때까지 페이지의 각 구문에 차례로 적용됩니다.

  규칙을 작성할 때 규칙의 순서에 주의하십시오. 규칙은 규칙 집합에 있는 순서대로 페이지의 구문에 적용됩니다. 특정한 규칙과 "*"를 포함한 일반적 규칙이 있는 경우, 특정한 규칙을 먼저 정의한 다음 일반적 규칙을 적용하십시오. 그렇게 하지 않으면 특정한 규칙을 발견하기 전에 모든 구문에 일반 규칙이 적용됩니다.
- 모든 규칙은 `<RuleSet>` `</RuleSet>` 태그 내에 넣어야 합니다.
- HTML 컨텐트를 다시 써야 하는 모든 규칙은 규칙 집합의 `<HTMLRules>` `</HTMLRules>` 부분에 포함시키십시오.
- JavaScript 컨텐트를 다시 써야 하는 모든 규칙은 규칙 집합의 `<JSRules>` `</JSRules>` 부분에 포함시키십시오.
- XML 컨텐트를 다시 써야 하는 모든 규칙은 규칙 집합의 `<XMLRules>` `</XMLRules>` 부분에 포함시키십시오.
- 인트라넷 페이지에서, 다시 써야 하는 URL을 확인하고 규칙 집합의 해당 부분(HTML, JSRules 또는 XMLRules)에 필요한 규칙을 포함시키십시오.
- 규칙 집합을 필요한 도메인에 할당합니다.
- 게이트웨이를 다시 시작하여 변경 사항을 적용합니다.

  gateway-install-root`/SUNWportal/bin/gateway -n` *gateway-profile-name* `start`

규칙 집합의 루트 요소 정의

규칙 집합의 루트 요소에는 두 가지 속성이 있습니다.

- `RuleSetName`. 예를 들어 `default_ruleset`입니다. 이 이름은 규칙 집합에서 URI 매핑에 참조합니다.
- `Extends`. 이 속성은 규칙 집합의 상속 기능을 참조합니다. 값은 규칙 집합을 유도할 기준 집합을 가리킵니다.

  값 `none`을 사용하면 이 새로운 독립 규칙 집합이 다른 규칙 집합에 의존하지 않는 다는 것을 나타내며 *RuleSetName*을 지정하면 규칙 집합이 다른 규칙 집합에 의존한다는 것을 나타냅니다.

재귀적 기능 사용

Rewriter에서는 재귀적 기능을 사용하여 대응되는 문자열 패턴의 끝까지에서 같은 패턴을 검색합니다.

예를 들어 Rewriter에서 다음 문자열을 구문 분석하는 경우

```
<a href="src=abc.jpg,src=bcd.jpg,src=xyz.jpg>
```

규칙

```
<Attribute name="href" valuePatterns="*src=**"/>
```

은 처음 나타나는 패턴만을 다시 작성하며 그 결과는 다음과 같습니다.

```
<a href="src=http://jane.sun.com/abc.jpg>
```

재귀적 옵션을 사용하는 경우

```
<Attribute name="href" valuePatterns="REC:*src=**"/>;
```

Rewriter는 대응되는 문자열 패턴에서 끝까지 같은 패턴을 찾기 때문에 다음과 같은 출력을 얻게 됩니다.

```
<a
href="src=http://jane.sun.com/abc.jpg,src=http://jane.sun.com/bcd.jpg,src=http://jane.sun
.com/xyz.jpg>
```

언어 기반 규칙 정의

규칙은 다음 언어를 바탕으로 합니다.

- HTML
- JavaScript
- XML

HTML 컨텐트에 대한 규칙

웹 페이지의 HTML 컨텐트는 속성, 폼 및 애플릿으로 더욱 세분할 수 있습니다. 이에 따라 HTML 컨텐트에 대한 규칙은 다음과 같이 분류됩니다.

- 71 페이지 “HTML 컨텐트에 대한 속성 규칙”
- 73 페이지 “HTML 컨텐트에 대한 폼 규칙”
- 74 페이지 “HTML 컨텐트에 대한 애플릿 규칙”

HTML 컨텐트에 대한 속성 규칙

이 규칙은 값을 다시 써야 하는 대상 태그의 속성을 확인합니다. 속성 값은 단순한 URL, JavaScript 또는 DHTML 컨텐트일 수 있습니다. 예:

- "img" 태그의 `src` 속성은 이미지 위치를 가리킵니다(단순 URL).
- href 속성의 `onClick` 속성은 링크를 누를 때 처리됩니다(DJS).

이 절에서는 다음을 설명합니다.

- 71 페이지 "속성 규칙 구문"
- 71 페이지 "속성 규칙 예제"
- 72 페이지 "DJS 속성 예제"

속성 규칙 구문

```
<Attribute name="attributeName" [tag="*" valuePatterns="" source=  *   type=  URL|DHTML|DJS  ]/>
```

여기서,

`attributeName`은 속성의 이름입니다(필수).

`tag`는 속성이 속하는 태그입니다(옵션, 기본값 `*`, 모든 태그를 의미).

`valuePatterns`에 대해서는 75 페이지 "규칙에 패턴 매칭 사용"을 참조하십시오.

`source`는 이 속성이 정의되는 페이지의 URI를 지정합니다(옵션, 기본값 `*`, 모든 페이지를 의미).

`type`은 값의 유형을 지정합니다(옵션). 다음이 가능합니다.

`URL` - 단순 URL(기본값)

`DHMTL` - DHTML 컨텐트. 이런 종류의 컨텐트는 표준 HTML 컨텐트에서 나타나며 Microsoft의 HTC 형식 파일에서 사용됩니다.

`DJS` - JavaScript 컨텐트. onClick 및 onMouseover와 같은 모든 HTML 이벤트 처리기는 HTML 속성과 연계된 JavaScript를 가지고 있습니다.

속성 규칙 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://mymachine.intranet.com/mypage.html
```

페이지 컨텐트

```
<a href="http://mymachine.intranet.com/mypage.html">
```

규칙

```
<Attribute name="href"/>
or
<Attribute name="href" tag="a"/>
```

결과

```
<a href=gateway-URL/http://mymachine.intranet.com/myhome.html>
```

설명

다시 작성될 URL은 이미 절대 URL이므로 URL의 접두어로는 게이트웨이 URL만 사용됩니다.

DJS 속성 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.sesta.com/focus.html
```

페이지 컨텐트

```
<Form>

<input TYPE=TEXT SIZE=20 value=focus
onClick="Check(\q/focus.html\q,\qfocus\q);return;">

</Form>
```

규칙

```
<Attribute name=  onClick   type=  DJS  />
<Function type="URL" name="Check" paramPatterns="y,"/>
```

결과

```
<Form>

<INPUT TYPE=TEXT SIZE=20 value=focus onClick="Check(\q
gateway-URL
/http://abc.sesta.com/focus.html\q,\qfocus\q);return;">

</Form>
```

설명

지정된 페이지 컨텐트를 다시 쓰기 위해 두 가지 규칙이 필요합니다. 첫 번째 규칙은 `onClick` JavaScript 토큰을 확인합니다. 두 번째 규칙은 다시 작성되어야 하는 `check` 함수의 매개 변수를 확인합니다. 이 경우에는 `paramPatterns`가 첫 번째 매개 변수 자리에 `y` 값을 갖기 때문에 첫 번째 매개 변수만 다시 작성됩니다.

게이트웨이 URL과 JavaScript 토큰이 나타나는 페이지의 기본 URL이 필요한 매개 변수 앞에 덧붙여집니다.

HTML 컨텐트에 대한 폼 규칙

사용자가 찾아보는 HTML 페이지에는 폼이 있을 수 있습니다. 일부 폼 요소는 URL을 값으로 취할 수 있습니다.

이 절은 다음으로 세분됩니다.

폼 규칙 구문

```
<Form name="form1" field="visit" [valuePatterns="" source="*"]/>
```

여기서

`name`은 폼의 이름입니다(필수).

`field`는 값을 다시 작성해야 하는 폼의 필드입니다(필수).

`valuePatterns`에 대해서는 75 페이지 "규칙에 패턴 매칭 사용"을 참조하십시오.

`source`는 이 폼 정의가 있는 html 페이지의 URL입니다(옵션, 기본값 `*`, 모든 페이지를 의미).

폼 규칙 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://test.siroe.com/testcases/html/form.html
```

페이지 컨텐트

페이지 URI가 `form.html`이고 서버의 루트 디렉토리에 있다고 가정합니다.

```
<form name=form1  method=POST action=
"http://test.siroe.com/testcases/html/form.html">
<input type=hidden name=abc1 value="0|1234|/test.html">
</form>
```

`form1`의 일부인 `abc1`이라는 이름의 숨겨진 필드 값에 존재하는`/text.html`을 다시 쓰기 위해 다음 규칙이 필요합니다.

규칙

```
<Form source="*/form.html" name="form1"
field="abc1" valuePatterns="0|1234|"/>
<Attribute name="action"/>
```

결과

```
<FORM name=  form1
method=  POST   action="gateway-URL/
http://test.siroe.com/testcases/html/form.html">
<input type=hidden name=abc1
value="0|1234|gateway-URL/
http://test.siroe.com/test.html">
</FORM>
```

설명

action 태그는 정의된 특정 HTML 속성 규칙을 사용하여 다시 작성됩니다.

입력 태그 속성 값의 value는 결과에 나와 있는 것처럼 다시 작성됩니다. 지정된 valuePatterns를 찾고 일치하는 valuePatterns 이후의 모든 컨텐트는 페이지의 기본 URL과 게이트웨이 URL을 앞에 덧붙여 다시 작성됩니다. 75 페이지 “규칙에 패턴 매칭 사용”을 참조하십시오.

HTML 컨텐트에 대한 애플릿 규칙

단일 웹 페이지에 많은 애플릿이 있을 수 있고 각 애플릿에는 많은 매개 변수가 있을 수 있습니다. Rewriter는 규칙에 지정된 값을 애플릿의 HTML 정의에 대응시키고 애플릿 매개 변수 정의의 일부로 존재하는 URL 값을 수정합니다. 이러한 교체는 사용자가 특정 웹 페이지를 찾아볼 때가 아니라 서버에서 이루어집니다. 이 규칙은 HTML 컨텐트의 개체 태그 및 애플릿 모두에서 매개 변수를 찾아 다시 작성합니다.

이 절은 다음으로 세분됩니다.

- 74 페이지 “애플릿 규칙 구문”
- 75 페이지 “애플릿 규칙 예제”

애플릿 규칙 구문

```
<Applet code="ApplicationClassName/ObjectID
" param="parametername" [valuePatterns="" source="*"] />
```

여기서

code는 애플릿 또는 개체 클래스의 이름입니다(필수).

param은 값을 다시 작성해야 하는 매개 변수의 이름입니다(필수).

valuePatterns에 대해서는 75 페이지 “규칙에 패턴 매칭 사용”을 참조하십시오.

source는 애플릿 정의가 있는 페이지의 URL입니다(옵션, 기본값 *, 모든 페이지를 의미).

애플릿 규칙 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.siroe.com/casestudy/test/HTML/applet/rule1.html
```

페이지 컨텐트

```
<applet codebase=  appletcode   code=
RewriteURLinApplet.class   archive=  /test.jar  >
<param name=Test1 value="/index.html">
</applet>
```

규칙

```
<Applet source="*/rule1.html" code=
"RewriteURLin*.class" param="Test*"/>
```

결과

```
<APPLET codebase=  gateway-URL
/http://abc.siroe.com/casestudy/test/HTML/
applet/appletcode   code=  RewriteURLinApplet.class
 archive=  /test.jar  ><param name=  Test1   value="
gateway-URL/http:
//abc.siroe.com/index.html">
</APPLET>
```

설명

<Attribute name="codebase"/>가 default_gateway_ruleset에서 정의된 규칙이므로 codebase 속성은 다시 작성됩니다.

이름이 Test로 시작되는 모든 매개 변수는 다시 작성됩니다. 애플릿 코드가 표시되는 페이지의 기본 URL과 게이트웨이 URL이 값 param 태그의 value 속성에 접두어로 사용됩니다.

규칙에 패턴 매칭 사용

valuePatterns 필드를 사용하여 패턴 매칭을 수행하고 다시 써야 하는 구문의 특정 부분을 확인할 수 있습니다.

규칙의 일부로 valuePatterns를 지정했다면 매칭된 패턴 이후의 모든 컨텐트는 다시 작성됩니다.

아래 예제 폼 규칙을 생각해 보겠습니다.

```
<Form source="*/source.html
" name="form1" field="visit
" [valuePatterns="0|1234|"]/>
```

여기서

source는 폼이 표시되는 html 페이지의 URL입니다.

name은 폼의 이름입니다.

field는 값을 다시 써야 하는 폼의 필드입니다.

valuePatterns는 다시 써야 하는 문자열 부분을 나타냅니다. valuePatterns 뒤에 나타나는 모든 컨텐트는 다시 작성됩니다(옵션, 기본값 ""는 전체 값을 다시 써야 함을 나타냄).

valuePatterns에 특수 문자 지정

특수 문자는 백슬래시로 이스케이프하면 지정할 수 있습니다. 예:

```
<Form source="*/source.html " name="form1" field=" visit" [valuePatterns="0|1234|
\\;original text|changed text  ]/>
```

valuePatterns에서 와일드카드 사용

와일드카드 별표(*) 문자를 사용하여 다시 쓰기 위한 패턴 매칭을 수행할 수 있습니다.

valuePatterns 필드에 단순히 *만 지정할 수는 없습니다. *는 모든 텍스트와 일치함을 나타내기 때문에 valuePattern 뒤에 텍스트가 없습니다. 따라서 Rewriter가 다시 작성할 텍스트가 없습니다. *를 *abc와 같이 다른 문자열과 연결하여 사용해야 합니다. 이 경우에 *abc 이후의 모든 컨텐트가 다시 작성됩니다.

주 – 별표(*)는 규칙의 어떤 필드에서나 와일드카드로 사용할 수 있습니다. 그러나 규칙의 모든 필드에 *를 포함시킬 수는 없습니다. 모든 필드에 *가 있으면 규칙이 무시됩니다. 오류 메시지는 표시되지 않습니다.

원본 명령문에서 여러 필드를 구별하기 위해 표시되는 구분 문자(세미콜론 또는 쉼표)와 함께 * 또는 **를 사용할 수 있습니다. 하나의 별표(*)는 다시 작성되지 않을 모든 필드와 매칭되며 두 개의 별표(**)는 다시 작성해야 하는 모든 필드와 매칭됩니다.

76 페이지 “valuePatterns에서 와일드카드 사용”에는 * 와일드카드 사용에 대한 몇 가지 예제가 나와 있습니다.

표 4–1 * 와일드카드 사용의 실례

URL	valuePatterns	설명
`url1, url2, url3, url4`	`valuePatterns = "**, *, **, *"`	**가 다시 작성해야 하는 부분을 나타내기 때문에 `url1` 및 `url3`이 다시 작성됩니다.
`XYZABChttp://host1.sesta.com/dir1.html`	`valuePatterns = "*ABC"`	`http://host1.sesta.com/dir1.html` 부분만 다시 작성됩니다. `*ABC` 이후의 모든 부분을 다시 작성해야 합니다.
`"0|dir1|dir2|dir3|dir4|test|url1`	`valuePatterns = "*|*|**|*|**|*|"`	`dir2, dir4` 및 `url1`이 다시 작성됩니다. 다시 작성해야 하는 마지막 필드는 **를 사용하여 나타내지 않아도 됩니다.

JavaScript 컨텐트에 대한 규칙

JavaScript에는 다양한 위치에 URL이 있을 수 있습니다. Rewriter는 JavaScript를 직접 구문 분석하여 URL 부분을 확인할 수 없습니다. 특별한 규칙 집합을 작성하여 JavaScript 처리기가 URL을 확인하여 변환하도록 합니다.

URL 유형을 가진 JavaScript 요소는 다음으로 분류됩니다.

- 77 페이지 “변수”
- 84 페이지 “함수의 인수”

변수

변수의 일반 구문은 다음과 같습니다.

<Variable name="variableName" [type="URL|EXPRESSION|DHTML|DJS|SYSTEM" source="*"]>

JavaScript 변수는 가지고 있는 값의 유형에 따라 5가지 범주로 더욱 세분할 수 있습니다.

- 77 페이지 “URL 변수”
- 79 페이지 “EXPRESSION 변수”
- 80 페이지 “DHTML(Dynamic HTML) 변수”
- 81 페이지 “DJS(Dynamic JavaScript) 변수”
- 82 페이지 “SYSTEM 변수”

URL 변수

변수 값은 URL로 취급할 수 있는 단순한 문자열입니다.

이 절은 다음으로 세분됩니다.

- 78 페이지 “URL 변수 구문”

■ 78 페이지 “URL 변수 예제”

URL 변수 구문

```
<Variable name="variableName" type="URL" [source="*"]>
```

여기서

variableName은 변수의 이름입니다. variableName의 값은 다시 작성됩니다(필수).

type은 URL 변수입니다(필수, 이 값은 URL이어야 함).

source는 이 JavaScript 변수가 발견되는 페이지의 URI입니다(옵션, 기본값 *는 모든 페이지를 의미).

URL 변수 예제

기본 URL이 다음과 같다고 가정합니다.

```
http://abc.siroe.com/tmp/page.html
```

페이지 컨텐트

```
<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc1="/tmp/tmp.jpg";
var imgsrc2="http://srap.sesta.com/tmp/tmp.jpg";
var imgsrc3=imgsrc2;
//-->
</SCRIPT>
```

규칙

```
<Variable name=  imgsrc*   type="URL"/>
```

결과

```
<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc="gateway-URL/http://abc.siroe.com/tmp/tmp.jpg";
var imgsrc="gateway-URL/http://srap.sesta.com/tmp/tmp.jpg";
var imgsrc3=imgsrc2;
//-->
</SCRIPT>
```

설명

유형이 URL이고 이름이 imgsrc로 시작되는 모든 변수가 다시 작성됩니다. 결과의 첫 라인에서 게이트웨이 URL과 변수가 표시되는 페이지의 기본 URL이 앞에 덧붙습니다. 두 번째 라인에는 이미 절대 경로가 있기 때문에 게이트웨이 URL만 덧붙습니다. 세 번째 var imagsrc2는 그 값이 문자열이 아니라 다른 JavaScript 값이기 때문에 다시 작성되지 않습니다.

EXPRESSION 변수

Expression 변수에는 오른쪽에 표현식이 있습니다. 이 표현식의 결과는 URL입니다. Rewriter는 서버에서 이러한 표현식을 평가할 수 없기 때문에 HTML 페이지에 JavaScript 함수(psSRAPRewriter_convert_expression)를 추가합니다. 이 함수는 표현식을 하나의 매개 변수로 받아들여 클라이언트 브라우저에서 필요한 URL로 평가합니다.

명령문에 단순 URL이 있는지 EXPRESSION URL이 있는지 잘 모르는 경우에는 두 시나리오를 모두 처리할 수 있는 EXPRESSION 규칙을 사용하십시오.

이 절은 다음으로 세분됩니다.

- 79 페이지 “EXPRESSION 변수 구문”
- 79 페이지 “EXPRESSION 변수 예제”

EXPRESSION 변수 구문

```
<Variable name="variableName" [type="EXPRESSION" source="*"]/>
```

여기서

variableName은 그 값이 표현식인 JavaScript 변수의 이름입니다(필수).

type은 JavaScript 변수의 유형입니다(옵션, 기본값은 EXPRESSION).

source는 페이지의 URI입니다(옵션, 기본값 *, 모든 소스를 의미)

EXPRESSION 변수 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.siroe.com/dir1/dir2/page.html
```

페이지 컨텐트

```
<script LANGUAGE="Javascript">
<!--
//Expression variables
var expvar= getURIPreFix() + "../../images/graphics"+".gif";
document.write("<A HREF="+expvar+">Link to XYZ content</A><P>")
var expvar="../../images/graphics"+".gif";
//-->
</SCRIPT>
```

규칙

```
<Variable name=  expvar   type="EXPRESSION"/>
or
<Variable name=  expvar  />
```

결과

```
var expvar=psSRAPRewriter_convert_expression(getURIPreFix()
 + "../../images/graphics"+".gif");document.write("<a href="+expvar+">>
Link to XYZ content</A><P>")var expvar=  gateway-URL/http://abc.siroe.com/images/graphics"+".gif";
```

설명

첫 번째 줄에서 함수 psSRAPRewriter_convert_expression이 expvar 표현식 변수의 오른쪽에 덧붙여집니다. 이 함수는 표현식을 처리하고 런타임에 컨텐트를 다시 작성합니다. 세 번째 라인에서 값이 단순 URL로 다시 작성됩니다.

DHTML(Dynamic HTML) 변수

이것은 HTML 컨텐트를 포함한 JavaScript 변수입니다.

이 절은 다음으로 세분됩니다.

- 80 페이지 “DHTML 구문”
- 80 페이지 “DHTML 예제”

DHTML 구문

```
<Variable name="variableName" type="DHTML" [source="*"]/>
```

여기서

variableName은 DHTML 컨텐트가 있는 JavaScript 변수의 이름입니다(필수).

type은 변수의 유형입니다(필수, 이 값은 DHTML이어야 함).

source는 페이지의 URL입니다(옵션, 기본값은 *, 모든 페이지를 의미).

DHTML 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.sesta.com/graphics/set1/
graphics/jsscript/JSVAR/page.html
```

페이지 컨텐트

```
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=../../images/test.html>"
var dhtmlVar="<a href=/images/test.html>"
var dhtmlVar="<a href=images/test.html>"
//-->
</SCRIPT>
```

규칙

```
<Variable name=  dhtmlVar   type="DHTML"/>
<Attribute name="href"/>
or
<Attribute name="href" tag="a"/>
```

결과

```
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=gateway-URL
/http://abc.sesta.com/graphics/
set1/graphics/images/test.html>"
var dhtmlVar="<a href=gateway-URL/
http
://abc.sesta.com/images/test.html>"
var dhtmlVar="<a href=gateway-URL/
http://abc.sesta.com/graphics/set1/
graphics/jscript/JSVAR/images/test.html>"
//--></SCRIPT>
```

설명

JavaScript 구문 분석기는 `dhtmlVar`의 값을 HTML 컨텐트로 읽고 이 컨텐트를 HTML 구문 분석기로 보냅니다. HTML 구문 분석기가 href 속성 규칙이 대응된 HTML 규칙을 적용하기 때문에 URL이 다시 작성됩니다.

DJS(Dynamic JavaScript) 변수

이것은 JavaScript 컨텐트를 포함한 JavaScript 변수입니다.

이 절은 다음으로 세분됩니다.

DJS 구문

```
<Variable name="variableName" type="DJS" [source="*"]/>
```

여기서

`variable`은 그 값이 javascript인 JavaScript 변수입니다.

DJS 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.sesta.com/dir1/dir2/dir3/jscript/dir4/page.html
```

페이지 컨텐트

```
//DJS Var
var dJSVar="var dJSimgsrc=\q/tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=\q../tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=
\qhttp://abc.sesta.com/tmp/tmp.jpg\q;"
```

규칙

```
<Variable name="DJS">dJSVar/>
<Variable name="URL">dJSimgsrc/>
```

결과

```
//DJS Var - need 2 rules
var dJSVar="var dJSimgsrc=\qgateway-URL
/http://abc.sesta.com/tmp/tmp.jpg\q;"var dJSVar="var dJSimgsrc=\q
gateway-URL/http
://abc.sesta.com/dir1/dir2/dir3/jscript/tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=\qgateway-URL/
http://abc.sesta.com/tmp/tmp.jpg\q;"
```

설명

여기에 두 가지 규칙이 필요합니다. 첫 번째 규칙은 동적 JavaScript 변수 `dJSVar`을 찾습니다. 이 변수의 값은 다시 URL 유형의 JavaScript입니다. 이 JavaScript 변수의 값을 다시 쓰기 위해 두 번째 규칙이 적용됩니다.

SYSTEM 변수

사용자가 사용하지 못하도록 선언되어 있어 지원이 제한되는 변수입니다. 이 변수들은 JavaScript 표준의 일부로 사용할 수 있습니다. 예: `window.location.pathname`

이 절은 다음으로 세분됩니다.

SYSTEM 변수 구문

```
<Variable name="variableName" type="SYSTEM" [source="*"]/>
```

여기서

variableName은 JavaScript 시스템 변수입니다. 필수이며 값은document.URL, document.domain, location, doument.location, location.pathname, location.href, location.protocol, location.hostname, location.host 및 location.port 등의 패턴과 일치하는 변수일 수 있습니다. 이러한 값은 generic_ruleset에 있습니다. 이러한 시스템 변수 규칙을 수정하지 마십시오.

type은 시스템 유형 값을 지정합니다(필수이며 값은 DJS임).

source는 이 페이지의 URI입니다(옵션, 기본값 *는 모든 페이지를 의미).

SYSTEM 변수 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.siroe.com/dir1/page.html
```

페이지 컨텐트

```
<script LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(window.location.pathname);
//-->
</SCRIPT>
```

규칙

```
<Variable name=  window.location.pathname   type="SYSTEM"/>
```

결과

```
</SCRIPT>
<SCRIPT LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(psSRAPRewriter_convert_pathname(window.location.pathname));
//-->
</SCRIPT>
```

설명

Rewriter가 규칙에 대응되는 시스템 변수를 찾으면 psSRAPRewriter_convert_system 함수가 접두어로 사용됩니다. 이 함수는 런타임 때 시스템 변수를 처리하고 그에 따라 얻어지는 URL을 다시 씁니다.

함수의 인수

그 값을 다시 작성해야 하는 함수 매개 변수는 4가지 범주로 분류됩니다.

- 84 페이지 "URL 매개 변수"
- 86 페이지 "EXPRESSION 매개 변수"
- 87 페이지 "DHTML 매개 변수"
- 89 페이지 "DJS 매개 변수"

일반 구문

```
<Function name="functionName " paramPatterns="y,y,"
[type="URL|EXPRESSION|DHTML|DJS" source=  *  ]/>
```

여기서

name은 JavaScript 함수의 이름입니다(필수).

paramPatterns는 다시 작성해야 하는 매개 변수를 지정합니다(필수).

y. y의 위치는 다시 작성해야 하는 매개 변수를 나타냅니다. 예를 들어 구문에서 첫 번째 매개 변수는 다시 써야 하지만 두 번째 매개 변수는 다시 쓰지 않아야 합니다.

type은 이 매개 변수에 필요한 값의 종류를 지정합니다(옵션, 기본값은 EXPRESSION 유형).

source는 페이지의 소스 URI입니다(옵션, 기본값은 *, 모든 페이지를 의미).

URL 매개 변수

함수는 이 매개 변수를 문자열로 취하며 이 문자열은 URL로 취급할 수 있습니다.

이 절은 다음으로 세분됩니다.

- 84 페이지 "URL 매개 변수 구문"
- 85 페이지 "URL 매개 변수 예제"

URL 매개 변수 구문

```
<Function name="functionName" paramPatterns="y,," type="URL" [source=  *  ]/>
```

여기서

name은 URL 유형의 매개 변수를 갖는 함수 이름입니다(필수).

paramPatterns는 다시 작성해야 하는 매개 변수를 지정합니다(필수).

y. y의 위치는 다시 작성해야 하는 매개 변수를 나타냅니다. 예를 들어 구문에서 첫 번째 매개 변수는 다시 써야 하지만 두 번째 매개 변수는 다시 쓰지 않아야 합니다.

type은 함수의 유형입니다(필수, 이 값은 URL이어야 함).

source는 이 함수 호출을 갖는 페이지의 URL입니다(옵션, 기본값은 *, 모든 URL을 의미).

URL 매개 변수 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.sesta.com/test/rewriter/test1/jscript/test2/page.html
```

페이지 컨텐트

```
<script language="JavaScript">
<!--
function test(one,two,three){
alert(one + "##" + two + "##" +three);
}
test("/test.html","../test.html","123");
window.open("/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
```

규칙

```
<Function name="URL" name="test" paramPatterns="y,y,"/>
<Function name="URL" name="window.open" paramPatterns="y,,,"/>
```

결과

```
<SCRIPT language="JavaScript">
<!--
function test(one,two,three) {
alert(one + "##" + two + "##" +three);
}
test("gateway-URL/http://abc.sesta.com/test.html","
gateway-URL/http://abc.sesta.com/test/rewriter/
test1/jscript/test.html","123");window.open("gateway-URL/
http://abc.sesta.com/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
```

설명

첫 번째 규칙은 test 이름의 함수에 있는 처음 두 매개 변수를 다시 작성해야 한다고 지정합니다. 따라서 test 함수의 처음 두 매개 변수는 다시 작성됩니다. 두 번째 규칙은window.open 함수의 처음 매개 변수를 다시 작성해야 한다고 지정합니다. window.open 함수 내의 URL에는 이 함수 매개 변수를 포함한 페이지의 기본 URL과 게이트웨이 URL이 앞에 덧붙여집니다.

EXPRESSION 매개 변수

이 매개 변수는 표현식 값을 취하며 평가 결과는 URL이 됩니다.

이 절은 다음으로 세분됩니다.

- 86 페이지 “EXPRESSION 매개 변수 구문”
- 86 페이지 “EXPRESSION 매개 변수 예제”

EXPRESSION 매개 변수 구문

```
<Function name="functionName" paramPatterns="y" [type="EXPRESSION" source=  *  ]/>
```

여기서

name은 함수의 이름입니다(필수).

paramPatterns는 다시 작성해야 하는 매개 변수를 지정합니다(필수).

y. y의 위치는 다시 작성해야 하는 매개 변수를 나타냅니다. 위의 구문에서 첫 번째 매개 변수만 다시 작성됩니다.

type은 값 EXPRESSION을 지정합니다(옵션).

source는 이 함수가 호출되는 페이지의 URI입니다.

EXPRESSION 매개 변수 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.sesta.com/dir1/dir2/page.html
```

페이지 컨텐트

```
<script language="JavaScript">
<!--
function jstest2(){
return ".html";
}
function jstest1(one){
return one;
}
```

```
var dir="/images/test"
var test1=jstest1(dir+"/test"+jstest2());
document.write("<a HREF="+test1+">TEST</a>");
alert(test1);
//-->
</SCRIPT>
```

규칙

```
<Function type="EXPRESSION" name="jstest1" paramPatterns="y"/>
or
<Function name="jstest1" paramPatterns="y"/>
```

결과

```
<script language="JavaScript">
<!--
function jstest2(){
return ".html";
}
function jstest1(one){
return one;
}
var dir="/images/test"
var test1=jstest1(psSRAPRewriter_convert_expression(dir+"/test"+jstest2()));
document.write("<a HREF="+test1+">TEST</a>");
alert(test1);
//-->
</SCRIPT>
```

설명

이 규칙은 `jstest1` 함수의 첫 번째 매개 변수를 EXPRESSION 함수 매개 변수로 취급하여 다시 써야 한다는 것을 지정합니다. 예제 페이지 컨텐트에서 첫 번째 매개 변수는 런타임 때만 평가되는 표현식입니다. Rewriter는 이 표현식에 `psSRAPRewriter_convert_expression` 함수로 접두어를 붙입니다. 이 표현식이 평가되고 `psSRAPRewriter_convert_expression` 함수가 런타임 시 결과를 다시 씁니다.

주 – 위의 예제에서 JavaScript 변수 규칙의 일부로 `test1` 변수가 있을 필요는 없습니다. `jstest1`에 대한 함수 규칙이 다시 쓰기를 처리합니다.

DHTML 매개 변수

그 값이 HTML인 함수 매개 변수입니다.

HTML 페이지를 동적으로 생성하는 `document.write()` 같은 원시 JavaScript 메소드가 이 범주에 속합니다.

이 절은 다음으로 세분됩니다.

DHTML 매개 변수 구문

```
<Function name="functionName" paramPatterns="y" type="DHTML" [source=  *  ]/>
```

여기서

name은 함수의 이름입니다.

paramPatterns는 다시 작성해야 하는 매개 변수를 지정합니다(필수).

y. y의 위치는 다시 작성해야 하는 매개 변수를 나타냅니다. 위의 구문에서 첫 번째 매개 변수만 다시 작성됩니다.

DHTML 매개 변수 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://xyz.siroe.com/test/rewriter/test1/jscript/JSFUNC/page.html
```

페이지 컨텐트

```
<script>
<!--
document.write(\q<a href="/index.html">write</a><BR>\q)
document.writeln(\q<a href="index.html">writeln</a><BR>\q)
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
//-->
</SCRIPT>
```

규칙

```
<Function name="DHTML" name="document.write" paramPatterns="y"/>
<Function name="DHTML" name="document.writeln" paramPatterns="y"/>
<Attribute name="href"/>
```

결과

```
<SCRIPT>
<!--
document.write(\q<a href="gateway-URL/
http://xyz.siroe.com/index.html">write</a><BR>\q)
document.writeln(\q<a href="gateway-URL/
```

```
http://xyz.siroe.com/test/rewriter/test1/
jscript/JSFUNC/index.html">writeln</a><BR>\q)
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
//-->
</SCRIPT>
```

설명

첫 번째 규칙은 `document.write` 함수의 첫 번째 매개 변수를 다시 작성해야 한다고 지정합니다. 두 번째 규칙은 `document.writeln` 함수의 첫 번째 매개 변수를 다시 작성해야 한다고 지정합니다. 세 번째 규칙은 `href` 이름의 모든 속성을 다시 작성해야 한다고 지정하는 단순 HTML 규칙입니다. 이 예에서, DHTML 매개 변수 규칙이 함수에서 다시 작성해야 하는 매개 변수를 확인합니다. 그런 다음 HTML 속성 규칙이 적용되어 실제로 확인된 매개 변수가 다시 작성됩니다.

DJS 매개 변수

그 값이 JavaScript인 함수 매개 변수입니다.

이 절은 다음으로 세분됩니다.

- 89 페이지 "DJS 매개 변수 구문"
- 89 페이지 "DJS 매개 변수 예제"

DJS 매개 변수 구문

```
<Function name="functionName" paramPatterns="y" type="DJS" [source=  *  ]/>
```

여기서

`name`은 하나의 매개 변수가 DJS인 함수의 이름입니다(필수).

paramPatterns는 위 함수에서 어떤 매개 변수가 DJS인지를 지정합니다(필수).

y. y의 위치는 다시 작성해야 하는 매개 변수를 나타냅니다. 위의 구문에서 첫 번째 매개 변수만 다시 작성됩니다.

`type`은 DJS입니다(필수).

`source`는 페이지의 URI입니다(옵션, 기본값은 *, 모든 URI를 의미).

DJS 매개 변수 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.sesta.com/page.html
```

페이지 컨텐트

```
<script>
menu.addItem(new NavBarMenuItem("All Available Information","JavaScript:top.location=\qhttp://abc.sesta.com\q"));
</script>
```

규칙

```
<Function name="DJS" name="NavBarMenuItem" paramPatterns=",y"/>
<Variable name="URL">top.location</Variable>
```

결과

```
<script>
menu.addItem(new NavBarMenuItem("All Available Information",
"JavaScript:top.location=\qgateway-URL/
http://abc.sesta.com\q"));
</script>
```

설명

첫 번째 규칙은 JavaScript를 포함한 `NavBarMenuItem` 함수의 두 번째 매개 변수를 다시 작성해야 한다고 지정합니다. JavaScript 내에서 `top.location` 변수도 다시 작성해야 합니다. 이 변수는 두 번째 규칙을 사용하여 다시 작성됩니다.

XML 컨텐트에 대한 규칙

웹 페이지에 XML 컨텐트가 있을 수 있으며 여기에는 다시 URL이 있을 수 있습니다. 다시 작성해야 하는 XML 컨텐트는 두 범주로 구분됩니다.

- 90 페이지 “태그 텍스트”(태그의 PCDATA 또는 CDATA와 같음)
- 91 페이지 “속성”

태그 텍스트

이 규칙은 태그 요소의 PCDATA 또는 CDATA를 다시 작성하기 위한 것입니다.

이 절은 다음으로 세분됩니다.

- 90 페이지 “태그 텍스트 구문”
- 91 페이지 “태그 텍스트 예제”

태그 텍스트 구문

```
<TagText tag="tagName"
[attributePatterns=  attribute_patterns_for_ this_tag" source=  *  ]/>
```

여기서

`tagName`은 태그의 이름입니다.

attributePatterns는 이 태그에 대한 속성 및 그 값의 패턴입니다(옵션, 이 태그에 속성이 전혀 없다는 것을 의미).

source는 이 xml 파일의 URI입니다(옵션, 기본값은 *, 모든 xml 페이지를 의미).

태그 텍스트 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.sesta.com/test/rewriter/test1/xml/page.html
```

페이지 컨텐트

```
<xml>
<Attribute name="src">test.html</attribute>
<attribute>abc.html</attribute>
</xml>
```

규칙

```
<TagText tag="attribute" attributePatterns="name=src"/>
```

결과

```
<xml>
<Attribute name="src">gateway-URL/
http://abc.sesta.com/test/rewriter/test1/
xml/test.html</attribute><attribute>abc.html</attribute>
</xml>
```

설명

페이지 컨텐트의 첫 번째 행에는 92 페이지 “속성 예제”가 있습니다. 페이지 컨텐트의 두 번째 행에 name이라는 속성이 없는데 name 속성의 값이 src여야 하기 때문에 다시 작성하는 작업은 수행되지 않습니다. 이것도 다시 작성하려면 <TagText tag="attribute"/>가 있어야 합니다.

속성

XML 속성의 규칙은 HTML에 대한 속성 규칙과 유사합니다. XML의 속성 규칙이 대소문자를 구분하는 반면 HTML 속성은 그렇지 않다는 차이점이 있습니다. 이는 근본적으로 HTML에는 없지만 XML에는 있는 대소문자 구분 특성 때문입니다.

Rewriter는 속성 이름을 바탕으로 속성 값을 변환합니다.

이 절은 다음으로 세분됩니다.

- 92 페이지 “속성 구문”

속성 구문

```
<Attribute name="attributeName " [tag="*" type=  URL  valuePatterns="*"
source=  *  ]/>
```

여기서

attributeName은 속성의 이름입니다(필수).

tag는 이 속성이 있는 태그의 이름입니다(옵션, 기본값은 *, 모든 태그를 의미).

valuePatterns에 대해서는 75 페이지 “규칙에 패턴 매칭 사용”을 참조하십시오.

source는 이 XML 페이지의 URI입니다(옵션, 기본값은 *, 모든 XML 페이지를 의미).

속성 예제

페이지의 기본 URL이 다음과 같다고 가정합니다.

```
http://abc.sesta.com/test/rewriter/test1/xml/page.html
```

페이지 컨텐트

```
<xml>
<baseroot href="/root.html"/>
<img href="image.html"/>
<string href="1234|substring.html"/>
<check href="1234|string.html"/>
</xml>
```

규칙

```
<Attribute name="href"tag="check" valuePatterns="1234|"/>
```

결과

```
<xml>
<baseroot href="/root.html"/><img href="image.html"/>
<string href="1234|substring.html"/><check href="1234|
gateway-URL
/http://abc.sesta.com/test/rewriter/test1/xml/string.html"/></xml>
```

설명

위의 예에서 네 번째 라인만 규칙에 지정된 모든 조건을 만족하기 때문에 이 라인만 다시 작성됩니다. 75 페이지 “규칙에 패턴 매칭 사용”을 참조하십시오.

CSS(Cascading Style Sheet)에 대한 규칙

HTML 페이지에서 CSS(CSS2 포함)가 변환됩니다. URL이 CSS의 가져오기 구문과 `url()` 함수에만 있기 때문에 이 변환에 대해 정의된 규칙은 없습니다.

WML에 대한 규칙

WML은 HTML과 유사하므로 WML 컨텐트에 HTML 규칙이 적용됩니다.WML 컨텐트에 일반 규칙 집합을 사용하십시오. 70 페이지 “HTML 컨텐트에 대한 규칙”을 참조하십시오.

재귀적 기능 사용

Rewriter에서는 재귀적 기능을 사용하여 대응되는 문자열 패턴의 끝까지에서 같은 패턴을 검색합니다.

예를 들어 Rewriter에서 다음 문자열을 구문 분석하는 경우

```
<a href="src=abc.jpg,src=bcd.jpg,src=xyz.jpg>
```

규칙

```
<Attribute name="href" valuePatterns="*src=**"/>
```

은 처음 나타나는 패턴만을 다시 작성하며 그 결과는 다음과 같습니다.

```
<a href="src=http://jane.sun.com/abc.jpg>
```

하지만 재귀적 옵션을 사용하면

```
<Attribute name="href" valuePatterns="REC:*src=**"/>;
```

Rewriter는 대응되는 문자열 패턴에서 끝까지 같은 패턴을 찾기 때문에 다음과 같은 출력을 얻게 됩니다.

```
<a href="src=http://jane.sun.com/abc.jpg,src=
http://jane.sun.com/bcd.jpg,src=http://jane.sun.com/xyz.jpg>
```

디버그 로그를 사용한 문제 해결

Rewriter 문제를 해결하려면 디버깅 로그를 사용해야 합니다.

디버깅 메시지는 다음과 같이 분류됩니다.

- 오류– Rewriter가 복구할 수 없는 오류입니다.
- 경고– Rewriter의 기능에 심각한 영향을 미치지 않는 경고입니다. Rewriter는 이 유형의 오류를 복구할 수 있지만 약간의 오작동이 생길 수도 있습니다. 경고 메시지 일부는 정보 제공을 위한 것입니다. 예를 들어 경고 메시지로 "이미지 컨텐트 다시 쓰지 않음"이 기록될 수 있습니다. 이 메시지는 Rewriter에서 이미지를 다시 쓰게 하지 않으므로 문제가 없습니다.
- 메시지– Rewriter가 제공하는 가장 높은 수준의 정보입니다.

Rewriter 디버깅 수준 설정

▼ Rewriter 디버깅 수준을 설정하려면

1 게이트웨이 컴퓨터에 루트로 로그인하여 다음 파일을 편집합니다.

gateway-install-root/SUNWam/config/AMConfig-*instance-name*.properties

2 디버깅 수준을 설정합니다.

com.iplanet.services.debug.level=

디버깅 수준은 다음과 같습니다.

오류 - 디버그 파일에 심각한 오류만 기록됩니다. 이런 오류가 발생하면 보통 Rewriter가 중지됩니다.

경고 - 경고 메시지가 기록됩니다.

메시지 - 모든 디버그 메시지가 기록됩니다.

off - 디버그 메시지가 기록되지 않습니다.

3 AMConfig-*instance-name* .properties 파일의 다음 등록 정보에 디버그 파일의 디렉토리를 지정합니다.

com.iplanet.services.debug.directory=/var/opt/SUNWam/debug

여기서 /var/opt/SUNWam/debug는 기본 디버그 디렉토리입니다.

4 터미널 창에서 게이트웨이를 다시 시작합니다.

./psadmin start-sra-instance –u amadmin – f <*password file*> –N <*profile name*>– t <*gateway*>

디버깅 파일 이름

디버그 수준이 메시지로 설정된 경우 디버그에서 파일 집합이 생성됩니다. 95 페이지 “디버깅 파일 이름”에는 Rewriter 파일과 해당 파일에 포함된 정보가 나와 있습니다.

표 4–2 Rewriter 디버깅 파일

파일 이름	정보
RuleSetInfo	다시 쓰기에 사용된 모든 규칙 집합이 이 파일에 기록됩니다.
Original Pages	페이지 URI, resolveURI(페이지 URI와 다른 경우), 컨텐트 MIME, 페이지에 적용된 규칙 집합, 구문 분석기 MIMIE 및 원본 컨텐트가 들어 있습니다. 구문 분석과 관련된 특정 오류/경고/메시지도 이 파일에 들어 있습니다. 메시지 모드에서는 전체 컨텐트가 기록됩니다. 경고 및 오류 모드에서는 다시 쓰는 동안 발생한 예외만 기록됩니다.
Rewritten Pages	페이지 URI, resolveURI(페이지 URI와 다른 경우), 컨텐트 MIME, 페이지에 적용된 규칙 집합, 구문 분석기 MIMIE 및 재작성된 컨텐트가 들어 있습니다. 이 파일은 디버그 모드가 메시지로 설정되었을 때 채워집니다.
Unaffected Pages	수정되지 않은 페이지 목록을 포함합니다.
URIInfo Pages	발견되어 변환된 URL이 들어 있습니다. 컨텐트가 원본 데이터와 동일하게 유지되는 모든 페이지의 세부 사항이 이 파일에 기록됩니다. 세부적으로 기록되는 내용: 페이지 URI, MIME 및 인코딩 데이터, 재작성에 사용된 rulesetID 그리고 구문 분석기 MIME

위의 파일 이외에도 Rewriter는 위 파일에서 포착하지 않은 디버그 메시지의 파일을 생성합니다. 이 파일 이름은 두 부분으로 구성됩니다. 첫 번째 부분은 `pwRewriter` 또는 `psSRARewriter`이고 두 번째 부분은 `portal` 또는 *gateway-profile-name*을 사용하는 확장 부분입니다.

디버깅 파일은 포털 또는 게이트웨이에 표시됩니다. 이 파일은 `AMConfig-`*instance-name*`.properties` 파일에 표시된 디렉토리에 있습니다.

Rewriter 구성 요소는 다음 파일 집합을 생성하여 디버깅을 지원합니다.

*prefix*_RuleSetInfo.*extension*

*prefix*_OrginalPages.*extension*

*prefix*_RewrittenPages.*extension*

*prefix*_UnaffectedPages.*extension*

*prefix*_URIInfo.*extension*

여기서

*prefix*는 URLScraper 사용 로그의 경우 `psRewriter`이고 게이트웨이 사용 로그의 경우 `psSRAPRewriter`입니다.

*extension*은 URLScraper 사용의 경우 `portal`이고 게이트웨이 사용의 경우 `gateway-profile-name`입니다.

예를 들어, 게이트웨이에서 Rewriter가 페이지를 변환하는데 사용되고 기본 게이트웨이 프로필이 사용되는 경우 디버깅이 다음 파일을 만듭니다.

```
psSRAPRewriter_RuleSetInfo.default

psSRAPRewriter_OriginalPages.default

psSRAPRewriter_RewrittenPages.default

psSRAPRewriter_UnaffectedPages.default

psSRAPRewriter_URIInfo.default

psSRAPRewriter.default
```

작업 예제

이 절에서 다루는 내용은 다음과 같습니다.

- 다시 작성해야 하는 컨텐트를 가진 단순 HTML 페이지
- 컨텐트를 다시 작성할 때 필요한 규칙
- 해당 재작성된 HTML 페이지

이러한 예제 페이지는 *portal-server-URL*`/rewriter` 디렉토리에 있습니다. 규칙을 적용하기 전에 페이지를 둘러본 다음 게이트웨이를 통해 재작성된 결과 파일을 검토하여 규칙의 작용에 대해 알아봅니다. 규칙이 이미 `default_gateway_ruleset`의 일부인 경우도 있습니다. 어떤 예제에서는 `default_gateway_ruleset`에 규칙을 포함시켜야 할 수 있습니다. 해당 위치에서 이에 대해 언급합니다.

주 – 굵게 표시된 부분은 다시 작성된 내용입니다.

다음 예제를 이용할 수 있습니다.

HTML

- 98 페이지 “HTML 속성 예제”
- 102 페이지 “HTML 폼 예제”
- 104 페이지 “HTML 애플릿 예제”

JavaScript

- 변수
  - 105 페이지 “JavaScript URL 변수 예제”
  - 105 페이지 “JavaScript 컨텐트 예제”
  - 110 페이지 “JavaScript DHTML 변수 예제”
  - 112 페이지 “JavaScript DJS 변수 예제”
  - 114 페이지 “JavaScript SYSTEM 변수 예제”

함수

- 115 페이지 “JavaScript URL 함수 예제”
- 117 페이지 “JavaScript EXPRESSION 함수 예제”
- 118 페이지 “JavaScript DHTML 함수 예제”
- 120 페이지 “JavaScript DJS 함수 예제”

XML

- XML 속성 예제

HTML 컨텐트 예제

HTML 속성 예제

▼ HTML 속성 예제를 사용하려면

1 이 예제는 다음에서 액세스할 수 있습니다.

portal-server-URL /rewriter/HTML/attrib/attribute.html

2 게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 `abc.sesta.com` 및 `host1.siroe.com`이 정의되어 있어야 합니다.

정의되어 있지 않으면 직접 연결이 가정되고 게이트웨이 URL이 앞에 덧붙지 않습니다.

이 예제에서 지정한 규칙은 이미 정의되어 있기 때문에 `default_gateway_ruleset`에 추가할 필요가 없습니다.

재작성 전의 HTML

```
<html>
Rewriting starts
<head>
<title>TEST PAGE () </title>
</head>
ID-htmlattr.1
<br><br>
1.a href <a href="http://abc.sesta.com/images/logo.gif">http://..</a>
<br><br>
2. href <a href="https://host1.siroe.com">https://..</a>
<br><br>
3. href <a href="../images/logo.gif">../images/</a>
<br><br>
4. href <a href="images/logo.gif">images/..</a> <br><br>
5. href <a href="../../images/logo.gif">../../images/</a> <br><br>
Rewriting ends
</html>
```

규칙

```
<Attribute name="href"/>
```

재작성 후의 HTML

```
<html>
Rewriting starts
<head>
<title>TEST PAGE () </title>
</head>
ID-htmlattr.1
<br><br>
1. a href <a href="gateway-URL/http://abc.sesta.com/images/logo.gif">http://..</a> <br>
```

<Attrib name="href"/> 규칙이 default_gateway_ruleset에 이미 정의되었기 때문에 이 URL은 다시 작성됩니다. URL이 이미 절대 경로이므로 게이트웨이 URL만 접두어로 사용됩니다. 게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 abc.sesta.com이 정의되어 있어야 합니다. 그렇게 하지 않으면 직접 연결이 가정되기 때문에 게이트웨이 URL이 접두어로 사용되지 않습니다.

```
2. href <a href="gateway-URL/https://host1.siroe.com">https://..</a>
```

// 다시 한번, 게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 host1.siroe.com이 정의되어 있어야 합니다. 그렇게 하지 않으면 직접 연결이 가정되기 때문에 게이트웨이 URL이 접두어로 사용되지 않습니다.

```
<br><br>
3. href <a href="gateway-URL/portal-server-URL/rewriter/HTML/images/logo.gif">../images/</a>
```

// 상대 경로가 지정되었기 때문에 필요한 하위 디렉토리와 함께 게이트웨이 URL과 Portal Server URL이 접두어로 사용됩니다. 제공된 예제 구조에서 HTML 디렉토리 아래에 images라는 디렉토리가 지정되지 않았기 때문에 이 링크가 작동하지 않습니다.

```
<br><br>

4 href <a href="gateway-URL/portal-server-URL/rewriter/HTML/attrib/images/
logo.gif">images/..</a> <br><br>
```

// 상대 경로가 지정되었기 때문에 필요한 하위 디렉토리와 함께 게이트웨이 URL과 Portal Server URL이 접두어로 사용됩니다.

```
5. href <a href="gateway-URL/portal-server-URL/rewriter/images/logo.gif">
../../images/</a> <br><br>
```

// 상대 경로가 지정되었기 때문에 필요한 하위 디렉토리와 함께 게이트웨이 URL과 Portal Server URL이 접두어로 사용됩니다. 제공된 예제 구조에서 Rewriter 디렉토리 아래에 images라는 디렉토리가 지정되지 않았기 때문에 이 링크가 작동하지 않습니다.

```
Rewriting ends</html>
```

HTML 동적 JavaScript 토큰 예제

이 절에서는 HTML JavaScript 토큰 예제 사용에 대해 설명합니다.

▼ HTML JavaScript 토큰 예제를 사용하려면

1. **이 예제는 다음에서 액세스할 수 있습니다.**
   portal-server-URL /rewriter/HTML/jstokens/JStokens.html

2. **이 예제에 지정된 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의 `default_gateway_ruleset`에 추가하십시오.**

3. **Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 `default_gateway_ruleset`을 편집합니다.**

4. **터미널 창에서 게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML

```
<html>
<head>
Rewriting starts
<script language="javascript">
function Check(test,ind){
if (ind == \qblur\q)
{alert("testing onBlur")}
if (ind == \qfocus\q)
{alert("testing onFocus")}
}
</SCRIPT>
</head>
<body>
<form>
<input TYPE=TEXT SIZE=20 value=blur onAbort="Check
(\q/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=blur onBlur="Check
(\q/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onFocus="Check
(\q/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onChange="Check
(\q/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onClick="Check
(\q/focus.html\q,\qblur\q);return;">
<br><br>
```

```
</form>
</body>
Rewriting ends
</html>
```

규칙

```
<Attribute name=  onClick   type=  DJS  />
<Function type="URL" name="Check" paramPatterns="y"/>
```

주 – <Function name="URL" name="Check" paramPatterns="y"/>는 JavaScript 함수 규칙이며 JavaScript 함수 예제에 자세하게 설명되어 있습니다.

재작성 후의 HTML

```
<html>
<head>
Rewriting starts
<script language="javascript">
function Check(test,ind){
if (ind == \qblur\q)
{alert("testing onBlur")}
if (ind == \qfocus\q)
{alert("testing onFocus")}
}
</SCRIPT>
</head>
<body>
<form>
<input TYPE=TEXT SIZE=20 value=blur onAbort="Check
(\qgateway URL/portal-server-URL/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=blur onBlur="Check
(\qgateway URL/portal-server-URL/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onFocus="Check
(\qgateway URL/portal-server-URL/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onChange="Check
(\qgateway URL/portal-server-URL/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onClick="Check
(\qgateway URL/portal-server-URL/focus.html\q,\qblur\q);return;">
```

// 이 샘플에서는 모든 명령문이 다시 작성됩니다. 각 경우에 게이트웨이 및 Portal Server URL이 접두어로 사용됩니다. 이것은 onAbort, onBlur, onFocus, onChange 및 onClick에 대한 규칙이 default_gateway_ruleset 파일에 정의되었기 때문입니다. Rewriter는 JavaScript 토큰을 검색한 다음 추가 처리를 할 수 있도록 JavaScript 함수 규칙으로 전달합니다. 샘플의 두 번째 규칙은 Rewriter에 다시 작성할 매개 변수를 알려줍니다.

```
</body>
<br>

Rewriting ends

</html>
```

HTML 폼 예제

▼ 폼 예제를 사용하려면

1. **다음 위치에서 예제에 액세스합니다.**
   portal-server-URL/rewriter/HTML/forms/formrule.html
2. **게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 `abc.sesta.com`이 정의되어 있어야 합니다.**
   정의되어 있지 않으면 직접 연결이 가정되고 게이트웨이 URL이 앞에 덧붙지 않습니다.
3. **이 예제에 지정된 규칙을 "HTML 속성 재작성을 위한 규칙" 부분의 `default_gateway_ruleset`에 추가하십시오.**
4. **Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 `default_gateway_ruleset`을 편집합니다.**
5. **터미널 창에서 게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
</head>
<body>
RW_START
<p>
<form name="form1" method="Post" action=
"http://abc.sesta.com/casestudy/html/form.html">
<input type="hidden" name="name1" value="0|1234|/test.html">
<input type="hidden" name="name3" value="../../html/test.html">
<form name="form2" method="Post" action="
http://abc.sesta.com/testcases/html/form.html"><br>
<input type="hidden" name="name1" value="0|1234|
../../html/test.html"></form>
```

```
RW_END </p>
</body>
</html>
```

규칙

```
<Form source="*" name="form1" field="name1" valuePatterns="0|1234|"/>
```

재작성 후의 HTML 페이지

```
<HTML>
<HEAD>
RW_START
</HEAD>
<BODY>
<P>
<FORM name=form1  method=POST action="gateway-URL/http://abc.sesta.com/casestudy/html/form.html">
```

<Attribute name="action"/>이 default_gateway_ruleset에서 HTML 규칙의 일부로 정의되었기 때문에 이 URL은 다시 작성됩니다. URL이 이미 절대 경로이므로 접두어로 게이트웨이 URL만 사용하면 됩니다. 게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 abc.sesta.com이 정의되어 있어야 합니다. 그렇게 하지 않으면 직접 연결이 가정되기 때문에 게이트웨이 URL이 접두어로 사용되지 않습니다.

```
<input type=hidden name=name1 value=
"0|1234|gateway URL/portal-server-URL/test.html">
```

// 여기서 폼 이름은 form1이고 필드 이름은 name1입니다. 이것은 규칙에서 지정된 폼 이름 및 필드 이름과 일치합니다. 규칙에 valuePatterns가 0|1234|로 나와 있으며 이 구문의 value와 일치합니다. 따라서 valuePattern 이후에 나오는 URL은 다시 작성됩니다. Portal Server URL과 게이트웨이 URL이 앞에 덧붙습니다. valuePatterns에 대한 자세한 내용은 “ 75 페이지 “규칙에 패턴 매칭 사용”을 참조하십시오.

```
<input type=hidden name=name3 value="../../html/test.html">
```

// name이 규칙에서 지정된 field 이름에 대응되지 않기 때문에 이 URL은 다시 작성되지 않습니다.

```
</FORM>
<FORM name=form2 method=POST action=
"gateway-URL/http://abc.sesta.com/casestudy/html/form.html"><BR>
```

// <Attribute name="action"/>이 기본 규칙 집합에서 HTML 규칙의 일부로 정의되었기 때문에 이 URL은 다시 작성됩니다. URL이 이미 절대 경로이므로 접두어로 게이트웨이 URL만 사용하면 됩니다.

```
<input type=hidden name=name1 value="0|1234|../../html/test.html">
```

// 폼 이름이 규칙에서 지정한 이름과 일치하지 않기 때문에 이 URL은 다시 작성되지 않습니다.

```
</FORM>
</BODY>
RW_END
</HTML>
```

HTML 애플릿 예제

▼ 애플릿 예제를 사용하려면

1. **애플릿 클래스 파일을 얻습니다.** `RewriteURLinApplet.class` **파일은 다음 위치에 있습니다.**

   portal-server-URL/rewriter/HTML/applet/appletcode

   애플릿 코드가 있는 페이지의 기본 URL은 다음과 같습니다.

   portal-server-URL/rewriter/HTML/applet/rule1.html

2. **이 예제에 지정된 규칙을 "HTML 속성 재작성을 위한 규칙" 부분의** `default_gateway_ruleset`**에 추가하십시오.**

3. **Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서** `default_gateway_ruleset`**을 편집합니다.**

4. **게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML

```
<html>
Rewriting starts
<br>
<applet codebase=appletcode code=RewriteURLinApplet.class archive=/test>
<param name=Test1 value="/index.html">
<param name=Test2 value="../index.html">
<param name=Test3 value="../../index.html">
</applet>
Rewriting ends
</html>
```

규칙

```
<Applet source="*/rule1.html" code="RewriteURLinApplet.class" param="Test*" />
```

재작성 후의 HTML

```
<HTML>
Rewriting starts
<BR>
<APPLET codebase=gateway-URL/portal-server-URL
/rewriter/HTML/applet/appletcode=RewriteURLinApplet.class archive=/test>
```

// <Attribute name="codebase"/> 규칙이 default_gateway_ruleset의 일부로 이미 지정되었기 때문에 이 URL은 다시 작성됩니다. 게이트웨이 및 Portal Server URL에는 appletcode 디렉토리까지의 경로가 접두어로 붙습니다.

```
<param name=Test1 value=
"gateway-URL/portal-server-URL/index.html">
```

// 페이지의 기본 URL이 rule1.html이고 매개 변수 이름이 규칙에 지정된 매개 변수 Test* 에 대응되기 때문에 URL은 다시 작성됩니다. index.html이 루트 수준에 있도록 지정되었기 때문에 게이트웨이 및 Portal Server URL이 직접 접두어로 사용됩니다.

```
<param name=Test2 value="gateway-URL
/portal-server-URL/rewriter/HTML/index.html">
```

// 페이지의 기본 URL이 rule1.html이고 매개 변수 이름이 규칙에 지정된 매개 변수 Test* 에 대응되기 때문에 URL은 다시 작성됩니다. 필요에 따라 경로가 앞에 덧붙습니다.

```
<param name=Test3 value="gateway-URL
/portal-server-URL/rewriter/index.html">
```

// 페이지의 기본 URL이 rule1.html이고 매개 변수 이름이 규칙에 지정된 매개 변수 Test* 에 대응되기 때문에 URL은 다시 작성됩니다. 필요에 따라 경로가 앞에 덧붙습니다.

```
</APPLET>
Rewriting ends
</HTML>
```

JavaScript 컨텐트 예제

JavaScript URL 변수 예제

▼ JavaScript URL 변수 예제를 사용하려면

1 이 예제는 다음에서 액세스할 수 있습니다.

portal-server-URL /rewriter/JavaScript/variables/url/js_urls.html

2 **게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 `abc.sesta.com`이 정의되어 있어야 합니다.**
정의되어 있지 않으면 직접 연결이 가정되고 게이트웨이 URL이 앞에 덧붙지 않습니다.

3 **이 예제에 지정된 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의 `default_gateway_ruleset`에 추가하십시오.**

4 **Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 `default_gateway_ruleset`을 편집합니다.**

5 **규칙을 추가한 경우 게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
Rewriting starts
<head>
<title>JavaScript Variable test page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc="/tmp/tmp.jpg";
var imgsrc="./tmp/tmp.jpg";
var imgsrc="../tmp/tmp.jpg";
var imgsrc="../../tmp/tmp.jpg";
var imgsrc="http://abc.sesta.com/tmp/tmp.jpg";
var imgsrc="../../../tmp/tmp.jpg";
var imgsrc="tmp/tmp.jpg";
//-->
</SCRIPT>
<br>
Testing JavaScript variables!
<br>
<img src="images/logo.gif">
<br>
Image
</body>
<br>
Rewriting ends
</html>
```

규칙

```
<Variable name=  imgsrc   type="URL"/>
```

재작성 후의 HTML 페이지

```
<html>
Rewriting starts
<head>
<title>JavaScript Variable test page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc="gateway-URL/portal-server-URL/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/url/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL
/rewriter/JavaScript/tmp/tmp.jpg";
var imgsrc="gateway-URL/http://abc.sesta.com/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL/rewriter/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/url/tmp/tmp.jpg";
```

// 위의 모든 URL은 규칙에 지정된 대로 URL 유형이며 이름이 `imgsrc`인 JavaScript 변수입니다. 따라서 게이트웨이 및 Portal Server URL이 앞에 덧붙습니다. Portal Server URL에 이어지는 경로는 필요에 따라 덧붙입니다.

```
//-->
</SCRIPT>
<br>
Testing JavaScript variables!
<br>
<img src="gateway URL/portal-server-URL/rewriter
/JavaScript/variables/url/images/logo.gif">
```

// `<Attribute name="src"/>`라는 규칙이 `default_gateway_ruleset`에 정의되었기 때문에 이 행은 다시 작성됩니다.

```
<br>
Image
</body>
<br>
Rewriting ends
</html>
```

JavaScript EXPRESSION 변수 예제

▼ JavaScript Expression 변수 예제를 사용하려면

1. 이 예제는 다음에서 액세스할 수 있습니다.

   portal-server-URL /rewriter/JavaScript/variables/expr/expr.html

2. 이 예제에 지정된 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의 `default_gateway_ruleset`에 추가하십시오(아직 없는 경우).

3. Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 `default_gateway_ruleset`을 편집합니다.

4. 규칙을 추가한 경우 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
<head>
<title>JavaScript EXPRESSION Variables Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//Expression variables
var expvar1="images";
var expvar2="/logo.gif";
var expvar = expvar1 + expvar2;
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
var expvar="/images/logo"+".gif";
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
//-->
</SCRIPT>
Testing JavaScript EXPRESSION variables
</body>
</html>
```

규칙

```
<Variable type=  EXPRESSION   name="expvar"/>
```

재작성 후의 HTML 페이지

```
<html>
<head>
<title>JavaScript EXPRESSION Variables Test Page</title>
</head>
<body>
<SCRIPT>
// Rewriter appends the wrapper function
psSRAPRewriter_convert_expression here
</SCRIPT>
<script LANGUAGE="Javascript">
<!--
//Expression variables
var expvar1="images";
var expvar2="/logo.gif";
var expvar =psSRAPRewriter_convert_expression( expvar1 + expvar2);
```

// Rewriter는 이 명령문의 오른쪽을 JavaScript EXPRESSION 변수인 것으로 인식합니다. Rewriter는 서버 쪽에서 이 표현식의 값을 결정할 수 없습니다. 따라서, psSRAPRewriter_convert_expression 함수가 표현식 앞에 덧붙습니다. 이 표현식은 클라이언트 쪽에서 평가되어 필요에 따라 다시 작성됩니다.

```
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
```

// 이전 구문에서 다시 작성된 expvar 값이 이 표현식의 값에 도달하기 위해 사용됩니다. 결과가 유효한 URL이기 때문에(그래픽이 샘플의 이 위치에) 링크가 제대로 작동합니다.

```
var expvar="gateway URL/portal-server-URL/images/logo"+".gif";
```

// Rewriter는 expvar의 오른쪽을 문자열 표현식으로 인식합니다. 이것은 서버 쪽에서 결정할 수 있기 때문에 직접 다시 작성됩니다.

```
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
```

// 이전 구문에서 다시 작성된 expvar 값이 이 표현식의 값에 도달하기 위해 사용됩니다. 결과가 유효한 URL이 아니기 때문에(샘플의 이 위치에 그래픽이 없음) 링크가 제대로 작동하지 않습니다.

```
//-->
</SCRIPT>
Testing JavaScript EXPRESSION variables
</body>
</html>
```

JavaScript DHTML 변수 예제

▼ JavaScript DHTML 변수 예제를 사용하려면

1. 이 예제는 다음에서 액세스할 수 있습니다.

   portal-server-URL /rewriter/JavaScript/variables/dhtml/dhtml.html

2. 게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 `abc.sesta.com`이 정의되어 있어야 합니다. 정의되어 있지 않으면 직접 연결이 가정되고 게이트웨이 URL이 앞에 덧붙지 않습니다.

3. 이 예제에 지정된 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의 `default_gateway_ruleset`에 추가하십시오(아직 없는 경우). Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 `default_gateway_ruleset`을 편집합니다.

4. 규칙을 추가한 경우 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
<head>
<title>JavaScript DHTML Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=../../images/test.html>"
var dhtmlVar="<a href=/../images/test.html>"
var dhtmlVar="<a href=/images/test.html>"
var dhtmlVar="<a href=images/test.html>"
var dhtmlVar="<a href=http://abc.sesta.com/images/test.html>"
var dhtmlVar="<img src=http://abc.sesta.com/images/test.html>"
//-->
</SCRIPT>
<br><br>
Testing DHTML Variables
<br><br>
<img src="images/logo.gif">IMAGE
</body>
</html>
```

규칙

```
<Variable name="DHTML">dhtmlVar</Variable>
```

재작성 후의 HTML 페이지

```
<html>
<head>
<title>JavaScript DHTML Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=gateway-URL/portal-server-URL
/rewriter/JavaScript/images/test.html>"
```

// JavaScript DHTML 규칙이 dhtmlVar 오른쪽을 동적 HTML 컨텐트로 인식합니다. 따라서 default_gateway_ruleset 파일의 HTML 규칙이 적용됩니다. 동적 HTML에는 href 속성이 들어 있습니다. default_gateway_ruleset은 <Attribute name="href"/>의 규칙을 정의합니다. 따라서 href 속성의 값이 다시 작성됩니다. 하지만 URL은 절대 경로가 아닙니다. 따라서 상대 URL은 페이지의 기본 URL과 필요한 하위 디렉토리로 대체됩니다. 여기에 다시 접두어로 게이트웨이 URL이 사용되어 최종적으로 다시 작성된 결과가 유도됩니다.

```
var dhtmlVar="<a href=gateway-URL
/portal-server-URL/../images/test.html>"
```

// 페이지의 기본 URL이 추가되고 게이트웨이 URL이 앞에 덧붙여지지만 결과적인 URL은 올바로 작동하지 않습니다. 초기 URL /../images/test.html이 부정확하기 때문입니다.

```
var dhtmlVar="<a href=gateway-URL
/portal-server-URL/images/test.html>"
```

// 여기서 다시 JavaScript DHTML 규칙은 오른쪽을 동적 HTML 컨텐트로 인식하고 이를 HTML 규칙으로 전달합니다. default_gateway_ruleset의 HTML 규칙 <Attribute name="href"/>가 적용되며 명령문이 다음과 같이 다시 작성됩니다. Portal Server URL과 게이트웨이 URL이 접두어로 사용됩니다.

```
var dhtmlVar="<a href=gateway URL/portal-server-URL/
rewriter/JavaScript/variables/dhtml/images/test.html>"
var dhtmlVar="<a href=gateway URL/http://abc.sesta.com/images/test.html>"
var dhtmlVar="<img src=gateway-URL/
http://abc.sesta.com/images/test.html>"
```

// JavaScript DHTML 규칙은 오른쪽의 동적 HTML 컨텐트를 확인하고 문구를 HTML 규칙으로 전달합니다. default_gateway_ruleset의 <Attribute name="src"/> 규칙이 적용됩니다. URL이 절대 경로이기 때문에 접두어로 게이트웨이 URL만 사용하면 됩니다. 이 URL을 다시 작성하려면 [도메인 및 하위 도메인의 프록시] 목록에 abc.sesta.com이 정의되어 있어야 합니다.

```
//-->
</SCRIPT>
<br><br>
Testing DHTML Variables
<br><br>
<img src="gateway-URL/portal-server-URL/
rewriter/JavaScript/variables/dhtml/images/logo.gif">
```

// <Attribute name="src"/>라는 규칙이 default_gateway_ruleset에 정의되었기 때문에 이 행은 다시 작성됩니다.

```
<br><br>
Image
</body>
</html>
```

JavaScript DJS 변수 예제

▼ JavaScript DJS 변수 예제를 사용하려면

1. **이 예제는 다음에서 액세스할 수 있습니다.**

   portal-server-URL /rewriter/JavaScript/variables/djs/djs.html

2. **게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에** abc.sesta.com **이 정의되어 있어야 합니다. 정의되어 있지 않으면 직접 연결이 가정되고 게이트웨이 URL이 앞에 덧붙지 않습니다.**

3. **이 예제에 지정된 두 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의** default_gateway_ruleset**에 추가하십시오(아직 없는 경우). Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서** default_gateway_ruleset**을 편집합니다.**

4. **게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
<head>
<title>Dynamic JavaScript Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
var dJSVar="var dJSimgsrc=\q/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\q../../../tmp/tmp/jpg\q;"
```

```
var dJSVar="var dJSimgsrc=\qhttp://abc.sesta.com/tmp/tmp/jpg\q;"
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
<br>
<img src="images/logo.gif">
<br>
Image
</body>
</html>
```

규칙

```
<Variable name=  dJSVar   type="DJS"/>
<Variable name="dJSimgsrc   type=URL"/>
```

재작성 후의 HTML 페이지

```
<html>
<head>
<title>Dynamic JavaScript Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
var dJSVar="var dJSimgsrc=\qgateway-URL
/portal-server-URL/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\qgateway-URL
/portal-server-URL/rewriter/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\qgateway-URL
/http://abc.sesta.com/tmp/tmp/jpg\q;"
```

// 위의 모든 구문은 게이트웨이 및 Portal Server URL로 다시 작성됩니다. 필요한 경로가 적합하게 앞에 덧붙여집니다. 첫 번째 규칙은 `dJSVar`의 오른쪽을 동적 JavaScript 변수로 인식합니다. 그런 다음 `dJSimgsrc`의 오른쪽을 URL 유형의 JavaScript 변수로 인식하는 두 번째 규칙으로 전달됩니다. 규칙에 따라 다시 작성됩니다.

```
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
<br>
<img src="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/djs/images/logo.gif">
```

// `<Attribute name="src"/>`라는 규칙이 `default_gateway_ruleset`에 정의되었기 때문에 이 행은 다시 작성됩니다.

```
<br>
Image
</body>
</html>
```

JavaScript SYSTEM 변수 예제

▼ JavaScript System 변수 예제를 사용하려면

1. **이 예제는 다음에서 액세스할 수 있습니다.**

   portal-server-URL /rewriter/JavaScript/variables/system/system.html

2. **이 예제에 지정된 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의 `default_gateway_ruleset`에 추가하십시오(아직 없는 경우).**

3. **Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 `default_gateway_ruleset`을 편집합니다.**

4. **게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
<head>
<title>JavaScript SYSTEM Variables Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(window.location.pathname);
//document.write
("<A HREF="+window.location.pathname+">SYSTEM</A><P>")
//-->
</SCRIPT>
Testing JavaScript SYSTEM Variables
<br>
This page displays the path where
the current page is located when loaded.
</body>
</html>
```

규칙

```
<Variable name=  window.location.pathname   type="SYSTEM"/>
```

재작성 후의 HTML

```
<html>
<head>
<title>JavaScript SYSTEM Variables Test Page</title>
</head>
<body>
<SCRIPT>
convertsystem function definition...
</SCRIPT>
<script LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(psSRAPRewriter_convert_system
(window.location, window.location.pathname,  window.location  ));
```

// Rewriter는 window.location.pathname을 JavaScript SYSTEM 변수로 인식합니다. 이 변수의 값은 서버 쪽에서 결정할 수 없습니다. 따라서 Rewriter는 변수 앞에 psSRAPRewriter_convert_pathname 함수를 덧붙입니다. 이 래퍼 함수는 클라이언트 쪽에서 변수의 값을 결정하고 필요에 따라 다시 작성합니다.

```
//-->
</SCRIPT>
Testing JavaScript SYSTEM Variables
<br>
This page displays the path where
the current page is located when loaded.
</body>
</html>
```

JavaScript URL 함수 예제

▼ JavaScript URL 함수 예제를 사용하려면

1. **이 예제는 다음에서 액세스할 수 있습니다.**

   portal-server-URL /rewriter/JavaScript/functions/url/url.html

2. **이 예제에 지정된 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의 default_gateway_ruleset에 추가하십시오(아직 없는 경우). Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 default_gateway_ruleset을 편집합니다.**

3. **게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
<body>
JavaScript URL Function Test Page
<br>
<script language="JavaScript">
<!--
function test(one,two,three)
{
alert(one + "##" + two + "##" +three);
}
test("/test.html","../test.html","123");
window.open("/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
</body>
</html>
```

규칙

```
<Function type="URL" name="test" paramPatterns="y,y"/>
<Function type="URL" name="window.open" paramPatterns="y"/>
```

재작성 후의 HTML 페이지

```
<html>
<body>
JavaScript URL Function Test Page
<br>
<script language="JavaScript">
<!--
function test(one,two,three)
{
alert(one + "##" + two + "##" +three);
}
test("/test.html","../test.html","123");
window.open("gateway-URL/portal-server-URL
/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
</body>
</html>
```

JavaScript EXPRESSION 함수 예제

▼ JavaScript Expressions 함수 예제를 사용하려면

1. 이 예제는 다음에서 액세스할 수 있습니다.

   `<portal-install-location>/SUNWportal/samples/rewriter`

2. 이 예제에 지정된 규칙을 `JavaScript` 소스 재작성을 위한 규칙 부분의 `default_gateway_ruleset`에 추가하십시오(아직 없는 경우).

3. Portal Server 관리 콘솔을 사용하여 Rewriter 서비스의 `default_gateway_ruleset`을 편집합니다.

4. 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
<body>
JavaScript EXPRESSION Function Test Page
<br><br><br>
<script language="JavaScript">
<!--
function jstest2()
{
return ".html";
}
function jstest1(one)
{
return one;
}
var dir="/images/test"
var test1=jstest1(dir+"/test"+jstest2());
document.write("<a HREF="+test1+">Test</a>");
alert(test1);
//-->
</SCRIPT>
</body>
</html>
```

규칙

```
<Function type="EXPRESSION" name="jstest1" paramPatterns="y"/>
```

재작성 후의 HTML 페이지

```
<html>
<body>
JavaScript EXPRESSION Function Test Page
<br><br><br>
<script>
<!--
// various functions including psSRAPRewriter_
convert_expression appear here.//-->
</SCRIPT>
<script language="JavaScript">
<!--
function jstest2()
{
return ".html";
}
function jstest1(one)
{
return one;
}
var dir="/images/test"
var test1=jstest1(psSRAPRewriter_convert_
expression(dir+"/test"+jstest2()));
```

// 규칙이 EXPRESSION 유형인 `jstest1` 함수의 첫 번째 매개 변수를 다시 써야 한다고 규정합니다. 이 표현식의 값은 `/test/images/test.html`입니다. 이 앞에 게이트웨이 및 Portal Server URL이 덧붙습니다.

```
document.write("<a HREF="+test1+">Test</a>");
alert(test1);
//-->
</SCRIPT>
</body>
</html>
```

JavaScript DHTML 함수 예제

▼ JavaScript DHTML 함수 예제를 사용하려면

1. **이 예제는 다음에서 액세스할 수 있습니다.**

   portal-server-URL /rewriter/JavaScript/functions/dhtml/dhtml.html

2. **이 예제에 지정된 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의 `default_gateway_ruleset`에 추가하십시오(아직 없는 경우).**

3 **Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 default_gateway_ruleset을 편집합니다.**

4 **게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
<head>
Testing JavaScript DHTML Functions
<br>
<br>
<script>
<!--
document.write(\q<a href="/index.html">write</a><BR>\q)
document.writeln(\q<a href="index.html">writeln</a><BR>\q)
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
//-->
</SCRIPT>
</head>
<body BGCOLOR=white>
<br><br>
Testing document.write and document.writeln
</body>
</html>
```

규칙

```
<Function type="DHTML" name=" document.write" paramPatterns="y"/>
<Function type="DHTML" name=" document.writeln" paramPatterns="y"/>
```

재작성 후의 HTML 페이지

```
<html>
<head>
Testing JavaScript DHTML Functions
<br>
<br>
<script>
<!--
document.write(\q<a href="gateway-URL
/portal-server-URL/index.html">write</a><BR>\q)
```

// 첫 번째 규칙은 DHTML JavaScript 함수 document.write의 첫 번째 매개 변수를 다시 작성해야 한다고 지정합니다. Rewriter는 첫 번째 매개 변수를 단순 HTML 명령문으로

파악합니다. default_gateway_ruleset의 HTML 규칙 부분에는 구문을 다시 써야 한다고 지시하는<Attribute name="href" /> 규칙이 있습니다.

```
document.writeln(\q<a href="gateway-URL
/portal-server-URL/rewriter/JavaScript/functions/dhtml/index.html">writeln</a><BR>\q)
```

// 두 번째 규칙은 DHTML JavaScript 함수 document.writeln의 첫 번째 매개 변수를 다시 작성해야 한다고 지정합니다. Rewriter는 첫 번째 매개 변수를 단순 HTML 명령문으로 파악합니다. default_gateway_ruleset의 HTML 규칙 부분에는 구문을 다시 써야 한다고 지시하는<Attribute name="href" /> 규칙이 있습니다.

```
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
```

// DHTML 규칙이 함수 document.write 및 document.writeln을 식별하지만 위의 구문이 다시 작성되지 않습니다. 이 경우의 첫 번째 매개 변수가 단순 HTML이 아니기 때문입니다. 이것은 어떤 문자열도 될 수 있으며 Rewriter는 이를 다시 작성하는 방법을 알지 못합니다.

```
//-->
</SCRIPT>
</head>
<body BGCOLOR=white>
<br><br>
Testing document.write and document.writeln
</body>
</html>
```

JavaScript DJS 함수 예제

▼ JavaScript DJS 함수 예제를 사용하려면

1. **이 예제는 다음에서 액세스할 수 있습니다.**

   portal-server-URL /rewriter/JavaScript/functions/djs/djs.html

2. **게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 abc.sesta.com이 정의되어 있어야 합니다.**

   정의되어 있지 않으면 직접 연결이 가정되고 게이트웨이 URL이 앞에 덧붙지 않습니다.

3. **이 예제에 지정된 규칙을 "JavaScript 소스 재작성을 위한 규칙" 부분의 default_gateway_ruleset에 추가하십시오(아직 없는 경우). Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 default_gateway_ruleset을 편집합니다.**

4. **게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 HTML 페이지

```
<html>
Test for JavaScript DJS Functions
<br>
<script>
menu.addItem(new NavBarMenuItem("All Available
Information","JavaScript:top.location=\qhttp://abc.sesta.com\q"));
//menu.addItem(new NavBarMenuItem("All Available Information","http://abc.sesta.com"));
</script>
</html>
```

규칙

```
<Function type="DJS" name="NavBarMenuItem" paramPatterns=",y"/>
<Variable type="URL" name=  top.location  />
```

재작성 후의 HTML 페이지

```
<html>
Testing JavaScript DJS Functions
<br>
<script>
menu.addItem(new NavBarMenuItem
("All Available Information","javaScript:top.location=
\qgateway-URL/http://abc.sesta.com\q"));
```

// abc.sesta.com은 게이트웨이 서비스에서 [도메인 및 하위 도메인의 프록시] 목록에 있는 항목입니다. 따라서 Rewriter는 이 URL을 다시 작성해야 합니다. 그러나 이 값은 절대 URL이기 때문에 접두어로 Portal Server URL을 사용할 필요가 없습니다. DJS 규칙은 DJS 함수 NavBarMenuItem의 두 번째 매개 변수를 다시 작성해야 한다고 지정합니다. 하지만 두 번째 매개 변수 역시 JavaScript 변수입니다. 이 변수의 값을 다시 쓰기 위해 두 번째 규칙이 필요합니다. 두 번째 규칙은 JavaScript 변수 top.location의 값을 다시 써야 한다고 지정합니다. 모든 조건이 충족되면 URL이 다시 작성됩니다.

```
//menu.addItem(new NavBarMenuItem("All Available Information","http://abc.sesta.com"));
```

// DJS 규칙에서 함수 NavBarMenuItem의 두 번째 매개 변수를 다시 써야 한다고 지정하고 있지만 이 구문에서는 다시 쓰지 않습니다. Rewriter가 두 번째 매개 변수를 단순 HTML로 인식하지 않기 때문입니다.

```
</script>
</html>
```

XML 속성 예제

▼ XML 속성 예제를 사용하려면

1 이 예제는 다음에서 액세스할 수 있습니다.
portal-server-URL /rewriter/XML/attrib.html

2 이 예제에 지정된 규칙을 "XML 소스 재작성을 위한 규칙" 부분의 `default_gateway_ruleset`에 추가하십시오(아직 없는 경우).

3 Portal Server 관리 콘솔에 있는 Portal Server 구성의 Rewriter 서비스에서 `default_gateway_ruleset`을 편집합니다.

4 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

재작성 전의 XML

```
<html>
RW_START
<body>
<xml>
<baseroot href="/root.html"/>
</xml>
<xml>
<img href="image.html"/>
</xml>
<xml>
<string href="1234|substring.html"/>
</xml>
<xml>
<check href="1234|string.html"/>
</xml>
</body>
RW_END
</html>
```

규칙

```
<Attribute name="href" tag="check" valuePatterns="1234|"/>
```

재작성 후의 HTML

```
<html>
Rewriting starts
<br>
```

```
<br>
<body>
<xml><baseroot href="/root.html"/></xml>
<xml><img href="image.html"/></xml>
<xml><string href="1234|substring.html"/></xml>
<xml><check href="1234|gateway-URL/portal-server-URL
/rewriter/XML/string.html"/></xml>
```

// 이 명령문은 규칙에 지정된 조건에 대응되기 때문에 다시 작성됩니다. Attribute name은 href이고 tag는 check이고 valuePatterns는 1234이며, valuePatterns 이후의 문자열이 다시 작성됩니다. valuePatterns에 대한 자세한 내용은 75 페이지 "규칙에 패턴 매칭 사용"을 참조하십시오.

```
</body>
Rewriting ends
</html>
```

사례 연구

여기에는 예제 메일 클라이언트에 대한 소스 HTML 페이지가 포함되어 있습니다. 이 사례 연구에서는 가능한 모든 경우와 규칙을 다루지 않습니다. 실제 인트라넷 페이지의 규칙을 쉽게 구성하도록 돕기 위한 예제 규칙 집합입니다.

가정

이 사례 연구에서는 다음을 가정합니다.

- 메일 클라이언트의 기본 URL이 abc.siroe.com이라고 가정합니다.
- 게이트웨이 URL이 gateway.sesta.com이라고 가정합니다.
- 게이트웨이 서비스의 [도메인 및 부속 도메인의 프록시] 목록에 관련 항목이 있습니다.

예제 페이지 1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0053)http://abc.siroe.com/mailclient/destin/?Cmd=navbar -->
<HTML XMLNS:WM><HEAD>
<META http-equiv=Content-Type content="text/html; CHARSET=utf-8">
<META http-equiv=Pragma content=no-cache>
<META http-equiv=Expires content=0><!--Copyright (c) 2000 Microsoft Corporation.
All rights reserved.--><!--CURRENT FILE== "IE5" "WIN32" navbar -->
<STYLE>WM\\:DROPMENU {
BEHAVIOR: url(http://abc.siroe.com/mailweb/controls/dropmenu.htc)
}
```

```
</STYLE>
<LINK href="destin_files/navbar.css" type=text/css rel=stylesheet>
<SCRIPT language=javascript>
var g_szUserBase= "http://abc.siroe.com/mailclient/destin"+"/";
var g_szFolder= ".";
var g_szVirtualRoot= "http://abc.siroe.com/mailweb";
var g_szImagePath= g_szVirtualRoot + "/img/";
</SCRIPT>
<SCRIPT src="/destin_files/navbar.js"></SCRIPT>
<META content="MSHTML 6.00.2600.0" name=GENERATOR></HEAD>
<BODY oncontextmenu=return(event.ctrlKey); onselectstart=return(false);
id=outbar_mainbody style="BACKGROUND-COLOR: appworkspace" leftMargin=0
topMargin=0 scroll=no>
<TABLE class=nbTableMain id=nbTableMain style="HEIGHT: 100%" cellSpacing=0
cols=1 cellPadding=0 rows="2">
<TBODY>
<TR>
<TD class=treeBrand>
<DIV class=treeOFLOW><IMG
style="PADDING-RIGHT: 0px; PADDING-LEFT: 0px; PADDING-BOTTOM: 0px; PADDING-TOP: 0px"
src="/destin_files/logo-ie5.gif" border=0></DIV></TD></TR>
<TR height="100%">
<TD>
<TABLE class=nbTable cellSpacing=0 cols=1 cellPadding=0 rows="4">
<TBODY>
<TR>
<TD class=nbFlybar id=show_navbar onkeydown=flybar_keydown()
onclick=ToggleTab(this.id) tabIndex=0 noWrap>
<DIV class=treeOFLOW>Shortcuts</DIV></TD></TR>
<TR style="HEIGHT: 100%">
<TD id=idOutbarpane style="TEXT-ALIGN: center" vAlign=top><A
id=inbox
href="http://abc.siroe.com/mailclient/destin/Inbox/?Cmd=contents&amp;Page=1"
target=viewer alt="Go to inbox"><IMG class=nbImage alt="Go to inbox"
src="destin_files/navbar-inbox.gif"></A>
<DIV class=nbLabel>Inbox</DIV><BR><A id=calendar
href="http://abc.siroe.com/mailclient/destin/Calendar/?Cmd=contents"
target=viewer alt="Go to calendar"><IMG class=nbImage
alt="Go to calendar" src="destin_files/navbar-calendar.gif"></A>
<DIV class=nbLabel>Calendar</DIV><BR><A id=contacts
            href="http://abc.siroe.com/mailclient/destin/Contacts/?Cmd=contents"
target=viewer alt="Go to contacts"><IMG class=nbImage
alt="Go to contacts" src="destin_files/navbar-contacts.gif"></A>
<DIV class=nbLabel>Contacts</DIV><BR><A id=options
href="http://abc.siroe.com/mailclient/destin/?Cmd=options"
target=viewer alt="Go to options"><IMG class=nbImage
alt="Go to options" src="destin_files/navbar-options.gif"></A>
<DIV class=nbLabel>Options</DIV></TD></TR>
```

```
<TR style="HEIGHT: 1.5em">
<TD class=nbFlybar id=show_folders onkeydown=flybar_keydown()
onclick=ToggleTab(this.id) tabIndex=0 noWrap>
<DIV class=treeOFLOW>Folders</DIV></TD></TR>
<TR>
<TD class=nbTreeProgress id=treeProgress style="DISPLAY: none"
vAlign=top noWrap><SPAN id=idLoading
style="OVERFLOW: hidden">Loading...</SPAN>
</TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
</BODY></HTML>
```

설명

125 페이지 “설명”에서는 예제 규칙 집합과 사례 연구 사이의 매핑을 보여줍니다.

표 4–3 예제 규칙 집합과 사례 연구 사이의 매핑

페이지 컨텐트	적용 규칙	Rewriter 결과	설명
`var g_szVirtualRoot= "http://abc.siroe.com/ mailweb";`	<Variable name="URL"> g_szVirtualRoot </Variable>	`var g_szVirtualRoot= "http://gateway.sesta.com /http://abc.siroe.com/ mailweb";`	`g_szVirtualRoot`는 그 값이 단순 URL인 변수입니다. 이 규칙은 URL 유형의 변수 `g_szVirtualRoot`를 검색하라고 Rewriter에 지시합니다. 웹 페이지에 이런 변수가 있으면 Rewriter가 이를 절대 URL로 변환하고 접두어로 게이트웨이 URL을 사용합니다.
`src="/destin_files/ logo-ie5.gif"`	<Attribute name="src" />	`src="http://gateway.sesta. com/ http://abc.siroe.com/ destin_files/logo-ie5.gif`	src는 속성 이름이며 태그나 valuePattern이 따라붙지 않습니다. 이 규칙은 이름이 `src`인 모든 속성을 검색하고 그 속성 값을 다시 작성하도록 Rewriter에 지시합니다.
`href="http://abc.siroe. com /mailclient/destin/Inbox/ ?Cmd=contents& Page=1"`	`<Attribute name="href"/>`	`href="http://gateway.sesta. com/ http://abc.siroe.com /mailclient/destin/ Inbox/?Cmd=contents& Page=1"`	href는 속성 이름이며 태그나 valuePattern이 따라붙지 않습니다. 이 규칙은 이름이 `href`인 모든 속성을 검색하고 그 속성 값을 다시 작성하도록 Rewriter에 지시합니다.

주 – 규칙 집합을 적용하는 우선 순위는 `hostname-subdomain-domain`입니다.

예를 들어, 도메인 기반 규칙 집합 목록에 다음 항목이 있다고 가정합니다.

```
sesta.com|ruleset1
eng.sesta.com|ruleset2
host1.eng.sesta.com|ruleset3
```

`ruleset3`은 `host1`의 모든 페이지에 적용됩니다.

`ruleset2`는 `host1`에서 가져온 페이지를 제외하고 `eng` 하위 도메인의 모든 페이지에 적용됩니다.

`ruleset1`은 `eng` 하위 도메인과 `host1`에서 가져온 페이지를 제외하고 `sesta.com` 도메인의 모든 페이지에 적용됩니다.

1. [저장]을 눌러 완료합니다.
2. 터미널 창에서 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

Outlook Web Access의 규칙 집합

Secure Remote Access 서버는 Sun Java System Web Server 및 IBM 응용 프로그램 서버에 설치된 MS Exchange 2000 SP3과 Outlook Web Access(OWA)의 MS Exchange 2003을 지원합니다.

▼ OWA 규칙 집합을 구성하려면

1 Portal Server 관리 콘솔에 관리자로 로그인합니다.

2 [Secure Remote Access] 탭을 선택하고 속성을 설정할 게이트웨이 프로필을 선택합니다.

3 [규칙 집합에 URI 매핑] 필드에서 Exchange 2000이 설치된 서버 이름, 그리고 이어 Exchange 2000 서비스 팩 4 OWA 규칙 집합의 이름을 입력합니다.

예:

```
exchange.domain.com|exchange_2000sp3_owa_ruleset.
```

공용 폴더 사용

Exchange에서 NTLM 인증을 사용하도록 공용 폴더를 구성합니다. HTTP 기본 인증을 사용하려면 이러한 구성이 필요합니다.

이를 수행하려면 Exchange 서버에서 [제어판]-->[관리 도구]를 선택한 다음 [인터넷 정보 서비스(IIS)]를 엽니다.

[기본 웹 사이트] 아래에 [공용]이라는 공용 폴더용 탭이 있습니다. 등록 정보를 마우스 오른쪽 버튼으로 눌러 선택합니다. [디렉터리 보안] 탭을 누릅니다. [익명 액세스 및 인증] 제어판에서 [편집...]을 선택합니다. 다른 옵션은 모두 선택을 취소하고 [기본 인증]만 선택합니다.

6.x 규칙 집합을 3.0과 매핑

다음 표에는 Secure Remote Access 서버 Rewriter 규칙과 이전 릴리스의 Portal Server 제품의 매핑이 정리되어 있습니다.

표 4–4 SP3의 규칙 매핑

Rewriter 6.0 DTD 요소	Rewriter 3.0 목록 상자 이름
HTML 컨텐트에 대한 규칙	
속성 - URL	Rewrite HTML 속성
속성 - DJS	JavaScript를 포함한 Rewrite HTML 속성
폼	Rewrite 폼 입력 태그 목록
애플릿	Rewrite 애플릿/개체 매개 변수 값 목록
JavaScript 컨텐트에 대한 규칙	
변수 - URL	URL의 Rewrite JavaScript 변수
변수 - EXPRESSION	Rewrite JavaScript 변수 함수
변수 - DHTML	HTML의 Rewrite JavaScript 변수
변수 - DJS	JavaScript의 Rewrite JavaScript 변수
변수 - SYSTEM	Rewrite JavaScript 시스템 변수
함수 - URL	Rewrite JavaScript 함수 매개 변수
함수 - EXPRESSION	Rewrite JavaScript 함수 매개 변수 함수
함수 - DHTML	HTML의 Rewrite JavaScript 함수 매개 변수
함수 - DJS	JavaScript의 Rewrite JavaScript 함수 매개 변수
XML 컨텐트에 대한 규칙	
속성 - URL	XML 문서의 Rewrite 속성
TagText	XML 문서의 Rewrite 텍스트 데이터
CSS 컨텐트에 대한 규칙	

표 4–4 SP3의 규칙 매핑 *(계속)*

Rewriter 6.0 DTD 요소	Rewriter 3.0 목록 상자 이름
규칙이 필요 없습니다. 기본적으로 모든 URL이 변환됩니다.	
WML 컨텐트에 대한 규칙	
정의된 규칙이 없습니다. WML은 HTML로 취급되며 HTML 규칙이 적용됩니다.	
WMLScript 컨텐트에 대한 규칙	
WML 스크립트에 대한 지원 없음	

◆ ◆ ◆ 5 장

5 NetFile 작업

이 장에서는 NetFile 및 관련 작업에 대해 설명합니다. NetFile을 구성하려면 14 장을 참조하십시오.

NetFile 소개

NetFile은 사용자가 원격 파일 시스템과 디렉토리에 원격으로 액세스하여 작업할 수 있도록 해주는 파일 관리자 응용 프로그램입니다.

Secure Remote Access의 NetFile 구성 요소는 Java2 애플릿으로 사용할 수 있습니다. Java2 애플릿에는 뛰어난 인터페이스가 있으며 액세스가 더 편합니다.

NetFile에는 다음과 같은 주요 기능이 있습니다.

- 공유 또는 폴더를 추가하거나 제거하는 기능
- 파일 업로드 및 다운로드
- 파일 및 폴더 검색
- GZIP 및 ZIP을 통한 파일 압축
- NetFile 환경의 메일 기능
- 현재 NetFile 세션 정보 저장
- 파일 끌어서 놓기

지원되는 파일 액세스 프로토콜

NetFile에서는 FTP, NFS 및 jCIFS(Microsoft Windows) 프로토콜을 사용하여 원격 시스템에 액세스할 수 있습니다. NetFile에는 다음과 같은 파일 액세스 프로토콜 기능도 있습니다.

- 사용자가 AUTODETECT를 지정하여 시스템을 추가하면 NetFile은 다음 시퀀스를 통해 사용할 프로토콜을 자동으로 감지합니다.
  - 포트 21에서 FTP 서버용 호스트를 확인합니다. FTP 응답에 문자열 "NetWare"가 들어 있으면 NETWARE 호스트로 간주됩니다.
  - 포트 2049에서 NFS 서버용 호스트를 확인합니다.
  - 포트 139에서 Microsoft Windows용 호스트를 확인합니다.
  - 위의 모든 조치가 실패할 경우 호스트 유형을 결정할 수 없다는 메시지가 표시됩니다.

    감지되는 첫 번째 파일 시스템 유형이 요청된 호스트에 연결하는 데 사용됩니다. 호스트 감지 순서는 Portal Server 관리 콘솔(PSConsole)에서 변경할 수 있습니다.

주 – 서버가 비 표준 포트에서 실행 중이면 연결이 실패합니다.

- NetFile을 사용하면 사용자가 원하는 파일 서버와 프로토콜을 선택할 수 있습니다.

  각 프로토콜에 지원되는 플랫폼이 아래에 나열되어 있습니다.

표 5–1 파일 시스템 및 지원되는 프로토콜

파일 시스템/프로토콜	플랫폼
FTP	Novell Netware의 Novell FTP 5.1 Server Win NT 4.0의 MS FTP Server 4.0 Win NT 2000의 MS FTP Server 5.0 Solaris FTP Server WU_FTP 2.6.1 ProFTPD 1.2.8 vsFTPd 1.2.0
NFS	Solaris 2.6 이상
jCIFS	Windows 95/98/NT/2000/ME/XP

주 – NetFile을 사용하여 파일을 ProFTPD 서버에 업로드하려면 ProFTPD 서버가 실행되는 호스트의 `proftpd.conf` 파일에서 "AllowStoreRestart"를 "on"으로 설정해야 합니다.

Novell Netware는 FTP서버를 통해서만 지원되며 원시 액세스를 통해서는 지원되지 않습니다.

Microsoft Windows(SMB/CIFS) 파일 시스템에 액세스하려면 jCIFS를 Portal Server에 설치해야 합니다. jCIFS는 CIFS/SMB 네트워킹 프로토콜을 구현하는 Open Source 클라이언트 라이브러리입니다.

▼ NetFile 정책을 만들려면

1. **Portal 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭과 [NetFile] 탭을 차례로 선택합니다.**
3. **[DN 선택] 드롭다운 상자에서 조직/역할/사용자를 선택합니다.**
4. **호스트 및 서비스에 대한 액세스/거부 권한을 설정합니다.**
5. **[저장]을 누릅니다.**
6. **게이트웨이를 다시 시작합니다.**

◆ ◆ ◆ 6 장

6 Netlet 작업

이 장에서는 Netlet을 사용하여 사용자의 원격 데스크탑과 인트라넷에서 응용 프로그램을 실행하는 서버 사이에서 응용 프로그램을 안전하게 실행하는 방법을 설명합니다. Netlet을 구성하려면 11 장을 참조하십시오.

이번 장은 다음 절로 구성됩니다.

Netlet 소개

Sun Java System Portal Server 소프트웨어 사용자는 원격 데스크탑에서 가장 많이 사용하는 응용 프로그램이나 회사별 응용 프로그램을 안전한 방식으로 실행할 수 있습니다. 플랫폼에 Netlet을 설치하면 이 응용 프로그램에 대한 액세스를 보호할 수 있습니다.

Netlet을 사용하면 인터넷과 같은 비보안 네트워크에서 공통 TCP/IP 서비스를 안전하게 실행할 수 있습니다. TCP/IP 응용 프로그램(예: 텔넷 및 SMTP), HTTP 응용 프로그램 및 모든 고정 포트 응용 프로그램을 실행할 수 있습니다.

TCP/IP 기반 응용 프로그램 또는 고정 포트를 사용하는 응용 프로그램은 Netlet을 통해 실행할 수 있습니다.

주 – 동적 포트는 FTP를 사용할 때에만 지원됩니다. Microsoft Exchange를 사용하려면 Outlook Web Access(OWA)를 사용합니다.

Netlet을 사용하는 경우에는 브라우저에서 팝업 차단 옵션을 비활성화하도록 사용자에게 알려야 합니다.

Netlet 구성 요소

Netlet에서 사용하는 다양한 구성 요소는 134 페이지 “Netlet 구성 요소”에 나와 있습니다.

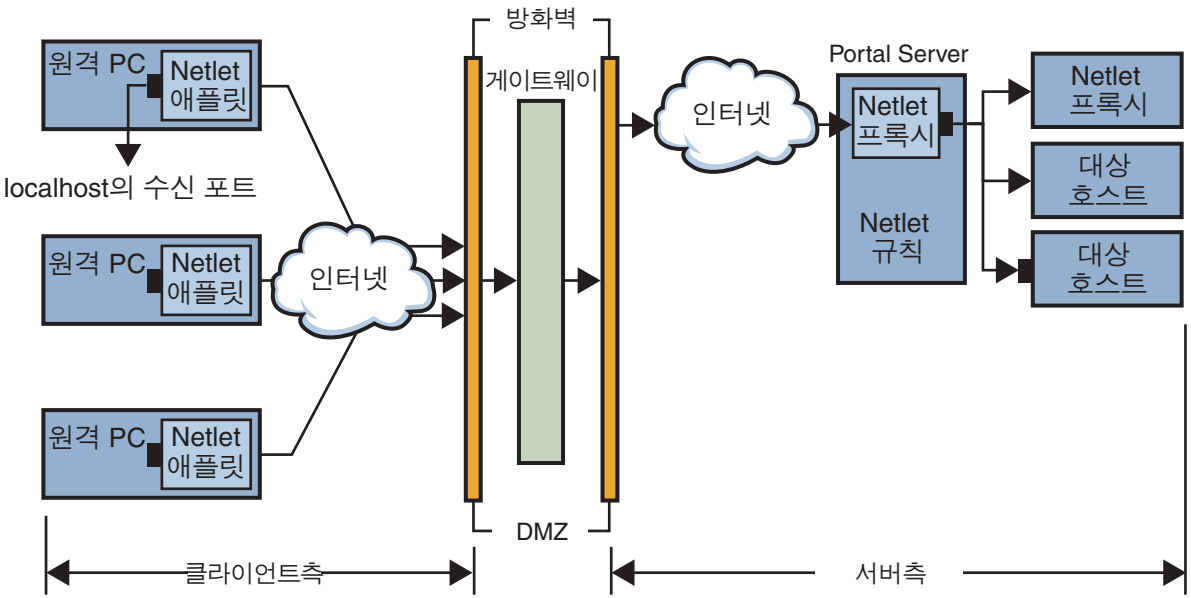


그림 6–1 Netlet 구성 요소

localhost의 수신 포트

Netlet 애플릿이 수신하는 클라이언트 컴퓨터에 있는 포트입니다. 클라이언트 컴퓨터는 localhost입니다.

Netlet 애플릿

Netlet 애플릿은 원격 클라이언트 컴퓨터와 Telnet, Grphon 또는 Citrix와 같은 인트라넷 응용 프로그램 사이에 암호화된 TCP/IP 터널을 설정하는 역할을 합니다. 애플릿은 패킷을 암호화하여 이를 게이트웨이로 보낸 다음, 게이트웨이로부터 받은 응답 패킷의 암호를 해독하여 로컬 응용 프로그램으로 보냅니다.

정적 규칙의 경우 Netlet 애플릿은 사용자가 포털에 로그인하면 자동으로 다운로드됩니다. 동적 규칙의 경우, 사용자가 동적 규칙에 해당하는 링크를 누르면 애플릿이 다운로드됩니다. 정적 및 동적 규칙에 대한 자세한 내용은 140 페이지 “규칙의 유형”을 참조하십시오.

Sun Ray 환경에서 Netlet을 실행하려면 153 페이지 “Sun Ray 환경에서 Netlet 실행”을 참조하십시오.

Netlet 규칙

Netlet 규칙은 클라이언트 시스템에서 실행해야 하는 응용 프로그램을 해당 대상 호스트로 매핑합니다. 이는 Netlet 규칙에 정의된 포트로 패킷이 전송되는 경우에만 Netlet이 작동한다는 것을 의미합니다. 이 옵션을 설정하면 보안이 강화됩니다.

관리자로서 Netlet이 제 기능을 발휘하도록 하려면 특정 규칙을 구성해야 합니다. 이 규칙은 사용할 암호, 불러올 URL, 다운로드할 애플릿, 대상 포트 및 대상 호스트와 같은 다양한 상세 정보를 지정합니다. 클라이언트 시스템에 있는 사용자가 Netlet을 통해 요청하면 이 규칙에 의해 연결 설정 방식이 결정됩니다. 자세한 내용은 137 페이지 “Netlet 규칙 정의”를 참조하십시오.

Netlet 공급자

Netlet의 UI 구성 요소입니다. 공급자는 사용자가 Portal Server 데스크탑에서 필요한 응용 프로그램을 구성할 수 있도록 해줍니다. 공급자에서 링크를 만든 후 사용자가 그 링크를 누르면 필요한 응용 프로그램 실행됩니다. 사용자는 Netlet 공급자의 데스크탑에서 동적 규칙의 대상 호스트를 지정할 수도 있습니다. 137 페이지 “Netlet 규칙 정의”를 참조하십시오.

Netlet 프록시(선택 사항)

게이트웨이는 원격 클라이언트 컴퓨터와 게이트웨이 사이에 보안 터널을 보장합니다. Netlet 프록시는 선택 사항이며 설치 중에 이 프록시의 설치는 선택하지 않아도 됩니다. Netlet 프록시에 대한 자세한 내용은 49 페이지 “Netlet 프록시 사용”을 참조하십시오.

Netlet 사용 시나리오

Netlet을 사용하는 경우 다음 이벤트 시퀀스가 사용됩니다.

1. 원격 사용자가 Portal Server 데스크탑에 로그인합니다.
2. 사용자, 역할 또는 조직에 정적 Netlet 규칙이 정의되어 있으면 Netlet 애플릿이 원격 클라이언트에게 자동으로 다운로드됩니다.

   사용자, 역할 또는 조직에 동적 규칙이 정의되어 있으면 사용자는 Netlet 공급자에서 필요한 응용 프로그램을 구성해야 합니다. 사용자가 Netlet 공급자에서 응용 프로그램 링크를 누르면 Netlet 애플릿이 다운로드됩니다. 정적 및 동적 규칙에 대한 자세한 내용은 137 페이지 “Netlet 규칙 정의”를 참조하십시오.

3. Netlet이 Netlet 규칙에 정의된 로컬 포트에서 수신합니다.
4. Netlet이 Netlet 규칙에 지정된 포트에서 원격 클라이언트와 호스트 사이의 채널을 설정합니다.

Netlet 작업

Netlet이 여러 조직에 있는 다양한 사용자의 필요에 따라 작동할 수 있으려면 다음을 수행해야 합니다.

1. 사용자 요구사항에 따라 정적 규칙이나 동적 규칙 중 어느 것을 만들어야 할지 결정합니다. 140 페이지 "규칙의 유형"을 참조하십시오.
2. Portal Server 관리 콘솔에서 Netlet 서비스의 옵션을 구성합니다. Netlet 구성에 대한 자세한 내용은 11 장을 참조하십시오.
3. 규칙이 조직, 역할 또는 사용자 중 어디에 기반할지 결정하고 각 수준에서 필요에 따라 수정합니다. 조직, 역할, 사용자에 대한 자세한 내용은 Portal Server 관리 설명서를 참조하십시오.

주 – `srapNetletServlet.properties` 파일에서 프레임 집합 매개 변수의 값을 현지화하지 마십시오.

원격 호스트에서 애플릿 다운로드

원격 시스템에서 가져와야 하는 내장된 애플릿이 포함된 URL에서 페이지가 반환되는 경우가 있습니다. 하지만 Java 보안에서는 애플릿을 다운로드한 호스트가 아닌 호스트와 애플릿이 통신하는 것을 허용하지 않습니다. 애플릿이 로컬 네트워크 포트를 통해 게이트웨이와 통신할 수 있도록 허용하려면 Access Manager 관리 콘솔에서 [애플릿 다운로드] 필드를 선택하고 다음 구문을 지정해야 합니다.

local-port:*server-host*:*server-port*

여기서

*local-port*는 Netlet이 애플릿에서 오는 트래픽을 수신하는 로컬 포트입니다.

*server-host*는 애플릿을 다운로드하는 위치입니다.

*server-port*는 애플릿 다운로드에 사용되는 포트입니다.

Netlet 규칙 정의

Netlet 구성은 Portal Server 관리 콘솔에서 Secure Remote Access 구성 탭을 사용하여 구성되는 Netlet 규칙에 의해 정의됩니다. Netlet 규칙은 조직, 역할 또는 사용자용으로 구성할 수 있습니다. Netlet 규칙이 역할이나 사용자용인 경우 조직을 선택한 다음 원하는 역할이나 사용자를 선택합니다.

주의 – Netlet 규칙은 멀티바이트 항목을 지원하지 않습니다. Netlet 규칙의 모든 필드에 멀티바이트 문자를 지정하지 마십시오.

Netlet 규칙에는 64000 보다 큰 포트 번호가 포함되면 안됩니다.

137 페이지 “Netlet 규칙 정의”에는 Netlet 규칙의 필드가 나와 있습니다.

표 6–1 Netlet 규칙의 필드

매개 변수	설명	값
규칙 이름	이 Netlet 규칙의 이름을 지정합니다. 각 규칙마다 고유한 이름을 지정해야 합니다. 특정 규칙에 대한 사용자 액세스를 정의할 때 유용합니다.	
암호화 암호	암호화 암호를 정의하거나 사용자 선택 가능한 암호 목록을 지정합니다.	선택한 암호는 Netlet 공급자에 목록으로 나타납니다. 사용자는 선택된 목록에서 필요한 암호를 선택할 수 있습니다. 기본값 - Netlet 관리 콘솔에 지정된 기본 VM 원시 암호와 기본 Java 플러그인 암호가 사용됩니다.
원격 응용 프로그램 URL	사용자가 Netlet 공급자에서 관련 링크를 누를 때 브라우저에서 여는 URL을 지정합니다. 브라우저는 응용 프로그램 창을 열고 이 규칙의 뒤 부분에 지정된 로컬 포트 `localhost`에 연결합니다. 관련 URL을 지정해야 합니다.	Netlet 규칙에서 불러온 응용 프로그램에 대한 URL. 예: `telnet://localhost :30000` . 응용 프로그램이 애플릿을 사용하여 응용 프로그램을 불러오는 경우 URL을 지정합니다. `null`– 응용프로그램이 URL에 의해 시작되지 않거나 데스크탑에서 제어되지 않는 경우 사용자가 설정한 값. 일반적으로 웹 기반이 아닌 응용 프로그램에는 true입니다.

표 6–1 Netlet 규칙의 필드 *(계속)*

매개 변수	설명	값
애플릿 다운로드 사용	이 규칙에 대한 애플릿 다운로드가 필요한지 여부를 나타냅니다.	■ **클라이언트 포트**는 클라이언트의 대상 포트를 나타냅니다. 이 포트는 기본 루프백 포트와 달라야 합니다. 각 규칙에 고유한 **로컬 포트**를 지정합니다. ■ **서버 호스트**는 애플릿을 다운로드할 서버의 이름입니다. ■ **서버 포트**는 애플릿 다운로드에 사용되는 서버의 포트를 나타냅니다. 애플릿을 다운로드해야 할 경우 서버가 지정되어 있지 않으면 애플릿은 Portal Server 호스트로부터 다운로드됩니다.
세션 확장 사용	Netlet이 활성 상태일 때 Portal Server 세션의 유휴 시간 초과를 제어합니다.	Netlet만 활성 상태이고 포털 응용 프로그램의 나머지가 유휴 상태일 때 포털 세션을 유지하려면 이 확인란을 선택합니다. 기본적으로 이 옵션은 선택되지 않습니다.

표 6–1 Netlet 규칙의 필드 (계속)

매개 변수	설명	값
로컬 포트를 대상 서버 포트에 매핑	로컬 포트	Netlet이 수신하는 클라이언트의 포트 *local-port* 값은 고유해야 합니다. 2개 이상 규칙에서 특정 포트 번호를 지정할 수 없습니다. 다중 연결을 위한 다중 호스트를 지정하는 경우에는 다중 로컬 포트를 지정합니다. 자세한 구문은 145 페이지 "다중 호스트 연결이 있는 정적 규칙"을 참조하십시오. FTP 규칙의 경우 로컬 포트 값이 30021이어야 합니다.
	대상 호스트	Netlet이 수신하는 클라이언트의 포트 Netlet 연결의 수신자. *host* - Netlet 연결을 수신하는 호스트의 이름. 이 값은 정적 규칙에 사용됩니다. `siroe`와 같은 간단한 호스트 이름이나 `siroe.mycompany.com`과 같은 정규 DNS 스타일 호스트 이름을 사용합니다. 다중 호스트를 지정하는 이유는 다음과 같습니다. *local-port* 값은 고유해야 합니다. 2개 이상 규칙에서 특정 포트 번호를 지정할 수 없습니다. 다중 연결을 위한 다중 호스트를 지정하는 경우에는 다중 로컬 포트를 지정합니다. 자세한 구문은 145 페이지 "다중 호스트 연결이 있는 정적 규칙"을 참조하십시오. FTP 규칙의 경우 로컬 포트 값이 30021이어야 합니다. 지정된 각 호스트와 연결을 설정하는 경우. 지정된 각 호스트에 해당하는 클라이언트 및 대상 포트를 지정해야 합니다. 자세한 구문은 145 페이지 "다중 호스트 연결이 있는 정적 규칙"을 참조하십시오. 지정된 호스트 목록에서 임의의 사용 가능한 호스트에 연결을 시도하는 경우. 자세한 구문은 145 페이지 "다중 호스트 연결이 있는 정적 규칙"을 참조하십시오. `TARGET` - 구문에서 `TARGET`을 지정하는 규칙은 동적 규칙입니다. `TARGET`은 최종 사용자가 데스크탑의 Netlet 공급자에서 필요한 대상 호스트(하나 또는 여러 개)를 지정할 수 있음을 나타냅니다. 단일 규칙에 정적 호스트와 TARGET을 조합할 수는 없습니다.

표 6–1 Netlet 규칙의 필드 *(계속)*

매개 변수	설명	값
대상 포트	대상 호스트의 포트 호스트 및 대상 호스트 뿐 아니라 대상 포트도 지정해야 합니다. 다중 대상 호스트가 있는 경우에는 다중 대상 포트를 지정할 수 있습니다. `port1+port2+port3-port4+port5`와 같은 형식으로 다중 포트를 지정합니다. 포트 번호 사이의 플러스(+) 기호는 한 대상 호스트에 대체 포트가 있음을 나타냅니다. 포트 번호 사이의 마이너스(-) 기호는 여러 대상 호스트의 포트 번호를 구분하는 기호입니다. 여기서 Netlet은 `port1`, `port2` 및 `port3`을 순서대로 사용하여 지정된 첫 번째 대상 호스트에 연결을 시도합니다. 연결이 실패하면 Netlet은 `port4`와 `port5`를 순서대로 사용하여 두 번째 호스트에 연결을 시도합니다. 정적 규칙에만 다중 포트를 구성할 수 있습니다.	

게이트웨이에서 Portal Server의 세션 알림을 수신하려면 다음을 추가합니다.

```
com.iplanet.am.jassproxy.trustAllServerCerts=true
```

아래의 등록 정보 파일에 추가합니다.

Portal Server의 `/etc/opt/SUNWam/config/AMConfig.`*instance-name* `.properties`

규칙의 유형

대상 호스트가 어떻게 규칙에 지정되어 있는지에 따라 Netlet 규칙에는 2가지 유형이 있습니다.

정적 규칙

정적 규칙은 대상 호스트를 규칙의 일부로 지정합니다. 정적 규칙을 만드는 경우 사용자는 필요한 대상 호스트를 지정하지 못합니다. 다음 예제에서 `sesta`는 대상 호스트입니다.

규칙 이름	암호화 암호	URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
ftpstatic	SSL_RSA_WITH_RC 4_128_MD5	null	false	true	■ 로컬 포트: 30021 ■ 대상 호스트: sesta ■ 대상 포트: 21

다중 대상 호스트와 포트는 정적 규칙에만 구성할 수 있습니다. 예제는 145 페이지 "다중 호스트 연결이 있는 정적 규칙"을 참조하십시오.

동적 규칙

동적 규칙에서는 대상 호스트가 규칙의 일부로 지정되지 않으며사용자가 Netlet 공급자에서 필요한 대상 호스트를 지정할 수 있습니다. 다음 예제에서 TARGET은 대상 호스트의 자리 표시자를 말합니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
ftpdynamic	SSL_RSA_WIT H_RC4_128_MD5	null	확인란 선택	확인란 선택	■ 로컬 포트: 30021 ■ 대상 호스트: TARGET ■ 대상 포트: 21

암호화 암호

암호화 암호를 기준으로 Netlet 규칙은 다음과 같이 세분화될 수 있습니다.

- **사용자 구성 가능 암호 규칙** - 이 규칙에서 사용자가 선택할 수 있는 암호 목록을 지정할 수 있습니다. 암호 옵션이 Netlet 공급자에 목록으로 나타납니다. 사용자는 목록에서 필요한 암호를 선택할 수 있습니다. 다음 예제에서 사용자는 여러 암호 중에서 선택할 수 있습니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
텔넷	SSL_RSA_WITH_RC4 _128_SHA	null	확인란 선택	확인란 선택	■ 로컬 포트: 30000 ■ 대상 호스트: TARGET ■ 대상 포트: 23
	SSL_RSA_WITH_RC4 _128_MD5				

주 – Portal Server 호스트에 사용 가능하도록 설정된 다양한 비밀번호가 있을 수도 있으나 사용자는 Netlet 규칙의 일부로 구성된 목록에서만 선택할 수 있습니다.

Netlet에서 지원되는 암호 목록은 142 페이지 "지원되는 암호"를 참조하십시오.

- **관리자 구성 암호 규칙** - 이 규칙에서는 Netlet 규칙의 일부로 암호를 구성합니다. 사용자는 필요한 암호를 선택할 수 없습니다. 다음 예제에서는 암호가 SSL_RSA_WITH_RC4_ 128_MD5로 구성되어 있습니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
텔넷	SSL_RSA_WITH_RC4_ 128_MD5	null	확인란 선택	확인란 선택	■ 로컬 포트: 30000 ■ 대상 호스트: TARGET ■ 대상 포트: 23

Netlet에서 지원되는 암호 목록은 142 페이지 "지원되는 암호"를 참조하십시오.

지원되는 암호

142 페이지 "지원되는 암호"에는 Netlet에서 지원되는 암호가 나와 있습니다.

표 6–2 지원되는 암호 목록

암호
원시 VM 암호
KSSL_SSL3_RSA_WITH_3DES_EDE_CBC_SHA
KSSL_SSL3_RSA_WITH_RC4_128_MD5
KSSL_SSL3_RSA_WITH_RC4_128_SHA
KSSL_SSL3_RSA_EXPORT_WITH_RC4_40_MD5
KSSL_SSL3_RSA_WITH_DES_CBC_SHA
Java 플러그인 암호
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_RC4_ 128_MD5
SSL_RSA_WITH_RC4_128_SHA

표 6–2 지원되는 암호 목록 *(계속)*

암호
SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_RSA_WITH_DES_CBC_SHA
SSL_RSA_WITH_NULL_MD5
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA

이전 버전과의 호환성

Portal Server의 이전 버전에서는 Netlet 규칙의 일부로 암호를 지원하지 않습니다. 암호가 없는 기존 규칙과의 이전 호환성을 위해 규칙에서는 기본 암호를 사용합니다. 암호가 없는 기존 규칙은 다음과 같습니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
텔넷		telnet://localhost:30000	확인란 선택 안 함	확인란 선택	■ 로컬 포트: 30000 ■ 대상 호스트: TARGET ■ 대상 포트: 23

아래와 같이 해석됩니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
텔넷	기본 암호	telnet://localhost:30000	확인란 선택 안 함	확인란 선택	■ 로컬 포트: 30000 ■ 대상 호스트: TARGET ■ 대상 포트: 23

이는 암호화 암호 필드가 기본값으로 선택된 관리자 구성 규칙과 비슷합니다.

주 – Netlet 규칙에는 64000보다 큰 포트 번호를 포함할 수 없습니다.

Netlet 규칙 예제

이 절에서는 Netlet 구문의 원리를 설명하기 위해 몇 가지 Netlet 규칙의 예제가 나와 있습니다.

■ 144 페이지 “기본 정적 규칙”

기본 정적 규칙

이 규칙은 클라이언트에서 컴퓨터 sesta로 가는 Telnet 연결을 지원합니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드	세션 확장	로컬 포트를 대상 서버 포트에 매핑
myrule	SSL_RSA_WITH_RC4_128_MD5	null	확인란 선택 안 함	true	■ 로컬 포트: 1111 ■ 대상 호스트: sesta ■ 대상 포트: 23

여기서

myrule은 규칙의 이름입니다.

SSL_RSA_WITH_RC4_ 128_MD5는 사용할 암호를 나타냅니다.

null은 이 응용 프로그램이 URL에 의해 호출되거나 데스크탑을 통해 실행되지 않음을 나타냅니다.

false는 클라이언트가 이 응용 프로그램을 실행하기 위해 애플릿을 다운로드하지 않음을 나타냅니다.

true는 Netlet 연결이 활성 상태인 경우 Portal Server에서 시간 초과가 발생하면 안 됨을 나타냅니다.

1111은 Netlet에서 대상 호스트로부터 오는 연결 요청을 수신하는 클라이언트의 포트입니다.

sesta는 Telnet 연결의 수신자 호스트 이름입니다.

23은 연결에 사용되는 대상 호스트의 포트 번호이며 이 경우에는 Telnet에 잘 알려진 포트입니다.

데스크탑 Netlet 공급자는 링크를 표시하지 않지만 Netlet은 자동으로 시작되어 지정된 포트(1111)에서 수신합니다. 사용자에게 클라이언트 소프트웨어를 시작하라고 지시하십시오. 이 경우에는 포트 1111의 localhost에 연결하는 Telnet 세션을 시작합니다.

예를 들어, Telnet 세션을 시작하려면 클라이언트는 단말기의 UNIX 명령줄에 다음을 입력해야 합니다.

```
telnet localhost 1111
```

다중 호스트 연결이 있는 정적 규칙

이 규칙은 클라이언트에서 2대의 컴퓨터 `sesta` 및 `siroe`로 가는 Telnet 연결을 지원합니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
myrule	`SSL_RSA_WITH_RC4_128_MD5`	null	확인란 선택	확인란 선택	■ 로컬 포트: 1111–1234 ■ 대상 호스트: sesta-siroe ■ 대상 포트: 23

여기서

23은 연결에 사용할 대상 호스트의 포트 번호로, Telnet용으로 예약된 포트입니다.

1111은 Netlet에서 첫 번째 대상 호스트 `sesta`로부터 오는 연결 요청을 수신하는 클라이언트의 포트입니다.

1234는 Netlet에서 두 번째 대상 호스트 `siroe`로부터 오는 연결 요청을 수신하는 클라이언트의 포트입니다.

이 규칙의 처음 6개 필드는 144 페이지 “기본 정적 규칙”에서와 동일합니다. 차이는 두 번째 대상 호스트를 식별하는 3개의 추가 필드가 있다는 점입니다.

규칙에 대상을 추가할 때에는 각각의 새로운 대상 호스트에 `local port`, `destination host` 및 `destination port` 필드를 추가해야 합니다.

주 – 각 대상 호스트에 대한 연결을 설명하는 3개의 필드 집합은 여러 개가 있을 수 있습니다. 원격 클라이언트가 UNIX 기반인 경우 번호가 낮은 포트는 제한되고 수신기를 시작할 수 있으려면 루트여야 하므로 2048 보다 작은 수신 포트 번호는 사용하면 안됩니다.

이 규칙은 앞의 규칙과 동일하게 적용됩니다. Netlet 공급자는 어떠한 링크도 표시하지 않지만 Netlet은 자동으로 시작되어 지정된 2개의 포트(1111과 1234)에서 수신합니다. 사용자는 클라이언트 소프트웨어를 시작해야 합니다. 이 경우 포트 1111의 `localhost`나 포트 1234의 `localhost`에 연결하는 Telnet 세션을 시작하여 두 번째 예제의 호스트에 연결해야 합니다.

다중 호스트 선택이 가능한 정적 규칙

다중 대체 호스트를 지정하려면 이 규칙을 사용합니다. 첫 번째 호스트에 대한 규칙의 연결이 실패하면 Netlet은 지정된 두 번째 호스트에 연결을 시도합니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
gojoe	SSL_RSA_WITH_RC4_128_MD5	/gojoe.html	■ 클라이언트 포트: 8000 ■ 서버 호스트: gojoeserver ■ 서버 포트: 8080	확인란 선택	■ 로컬 포트: 10491 ■ 대상 호스트: siroe+sesta ■ 대상 포트: 35+26+491-35+491

여기서

10491은 Netlet에서 대상 호스트로부터 오는 연결 요청을 수신하는 클라이언트의 포트입니다.

Netlet은 사용 가능한 포트에 따라 **포트** 35, **포트** 26 및 **포트** 491에서 같은 순서로 siroe에 연결을 시도합니다.

siroe에 연결할 수 없으면 Netlet은 포트 35 및 491에서 같은 순서로 sesta에 연결을 시도합니다.

호스트 사이의 더하기(+) 기호는 대체 호스트를 나타냅니다.

포트 번호 사이의 플러스(+) 기호는 한 대상 호스트에 대체 포트가 있음을 나타냅니다.

포트 번호 사이의 마이너스(-) 기호는 여러 대상 호스트의 포트 번호를 구분하는 기호입니다.

주 – 체인에서 제공되는 호스트 연결은 순서대로 연결 시도됩니다. 예를 들어, 규칙이 siroe+ sesta이면 siroe에 대한 연결을 먼저 시도합니다. 연결이 실패하면 sesta에 대한 연결을 시도합니다. 규칙에 먼저 나열된 호스트를 활성 네트워크에서 물리적으로 사용할 수 없는 경우 규칙에서 사용 불가능한 호스트 수가 증가함에 따라 다음 사용 가능 호스트에 연결하는 데 소요되는 시간이 증가합니다.

URL을 불러오기 위한 동적 규칙

이 규칙을 사용하면 사용자가 필요한 대상 호스트를 구성하여 Netlet을 통해 다양한 호스트에 Telnet 연결을 수행할 수 있습니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장 사용	로컬 포트를 대상 서버 포트에 매핑
myrule	SSL_RSA_WITH_RC4_128_MD5	telnet://localhost:30000	확인란 선택 안 함	확인란 선택	■ 로컬 포트: 30000 ■ 대상 호스트: TARGET ■ 대상 포트: 23

여기서

myrule은 규칙의 이름입니다.

SSL_RSA_WITH_RC4_ 128_MD5는 사용할 암호를 나타냅니다.

telnet://localhost :30000은 규칙에서 불러오는 URL입니다.

false는 애플릿이 다운로드되지 않음을 나타냅니다.

세션 확장(true)은 Netlet 연결이 활성 상태이면 Portal Server에서 시간 초과가 발생하면 안 됨을 나타냅니다.

30000은 Netlet에서 이 규칙에 대한 연결 요청을 수신하는 클라이언트의 포트입니다.

TARGET은 사용자가 Netlet 공급자를 사용하여 대상 호스트를 구성해야 함을 나타냅니다.

23은 Netlet에서 열린 대상 호스트의 포트 번호이며, 이 경우에는 Telnet에 잘 알려진 포트입니다.

▼ 규칙을 추가한 후 Netlet을 실행하려면

이 규칙을 추가한 후 Netlet이 예상대로 실행되도록 하려면 사용자는 몇 가지 단계를 완료해야 합니다. 사용자는 클라이언트 쪽에 다음을 수행해야 합니다.

1. **Portal Server 데스크탑의 Netlet 공급자 섹션에서 [편집]을 누릅니다.**
   새 Netlet 규칙이 [새 대상 추가] 섹션의 [규칙 이름]에 나열됩니다.
2. **규칙 이름을 선택하고 대상 호스트의 이름을 입력합니다.**
3. **변경 사항을 저장합니다.**
   새 링크가 Netlet 공급자 섹션에 표시되 있는 상태로 사용자는 데스크탑으로 돌아갑니다.
4. **새 링크를 누릅니다.**
   Netlet 규칙에 주어진 URL로 이동하는 새 브라우저가 시작됩니다.

주 – 이 단계를 반복하여 같은 규칙에 대상 호스트를 2개 이상 추가할 수 있습니다. 선택된 마지막 링크만 활성화됩니다.

애플릿을 다운로드하기 위한 동적 규칙

이 규칙은 클라이언트에서 동적 할당된 호스트로의 연결을 정의합니다. 규칙은 애플릿이 있는 서버에서 클라이언트로 GO-Joe 애플릿을 다운로드하게 됩니다.

규칙 이름	암호화 암호	원격 응용 프로그램 URL	애플릿 다운로드 사용	세션 확장	로컬 포트를 대상 서버 포트에 매핑
gojoe	SSL_RSA_WITH_RC4_128_MD5	/gojoe.html	■ 클라이언트 포트: 8000 ■ 서버 호스트: gojoeserver ■ 서버 포트: 8080	확인란 선택	■ 로컬 포트: 3399 ■ 대상 호스트: TARGET ■ 대상 포트:58

여기서

gojoe는 규칙의 이름입니다.

SSL_RSA_WITH_RC4_ 128_MD5는 사용할 암호를 나타냅니다.

예를 들어 /gojoe.html은 애플릿이 있는 HTML 페이지의 경로이고 이 경로는 포털이 배포된 웹 컨테이너의 문서 루트에 상대적인 값이어야 합니다.

8000:server:8080은 포트 8000은 애플릿을 수신하는 클라이언트의 대상 포트임을 나타내며 gojoeserve는 애플릿을 제공하는 서버의 이름이고, 8080은 애플릿을 다운로드할 서버의 포트입니다.

세션 확장(true)은 Netlet 연결이 활성 상태이면 Portal Server에서 시간 초과가 발생하면 안 됨을 나타냅니다.

3399는 Netlet에서 이 유형의 연결 요청을 수신하는 클라이언트의 포트입니다.

TARGET은 사용자가 Netlet 공급자를 사용하여 대상 호스트를 구성해야 함을 나타냅니다.

58은 Netlet에서 열린 대상 호스트의 포트 번호이며, 이 경우에는 GoJoe용 포트입니다. 포트 58은 대상 호스트가 자체 트래픽과 관련하여 수신하는 포트입니다. Netlet은 새 애플릿에서 이 포트로 정보를 전달합니다.

예제 Netlet 규칙

149 페이지 “예제 Netlet 규칙”에는 일부 공통 응용 프로그램에 대한 예제 Netlet 규칙이 나와 있습니다.

이 표에는 Netlet 규칙의규칙 이름, URL, 애플릿 다운로드, 로컬 포트, 대상 호스트, 대상 포트 필드에 해당하는 7개 열이 있습니다. 마지막 열에는 규칙에 대한 설명이 나와 있습니다.

주 – 149 페이지 “예제 Netlet 규칙”에는 Netlet 규칙의 암호 및 세션 확장 필드는 나와 있지 않습니다. 제시된 예제에 대해 이 필드 값을 "SSL_RSA_WITH_RC4_ 128_MD5" 및 "true"라고 가정하십시오.

표 6–3 예제 Netlet 규칙

규칙 이름	원격 응용 프로그램 URL	애플릿 다운로드 사용	로컬 포트를 대상 서버 포트에 매핑	설명
IMAP	null	확인란 선택 안 함	■ 로컬 포트: 10143 ■ 대상 호스트: imapserver ■ 대상 포트: 143	클라이언트 쪽의 Netlet **로컬 포트**는 서버 쪽의 **대상 포트**와 같지 않아도 됩니다. 표준 IMAP 및 SMTP 포트 이외의 포트를 사용할 경우 클라이언트를 표준 포트가 아닌 포트에 연결하도록 구성해야 합니다. Solaris 클라이언트 사용자는 루트로 실행하고 있지 않으면 1024 보다 작은 포트 번호에 연결할 수 없습니다.
SMTP	null	확인란 선택 안 함	■ 로컬 포트: 10025 ■ 대상 호스트: smtpserver ■ 대상 포트: 25	
Lotus 웹 클라이언트	null	확인란 선택 안 함	■ 로컬 포트: 80 ■ 대상 호스트: lotus-server ■ 대상 포트: 80	이 규칙은 Netlet에 포트 80에서 클라이언트에서 수신하도록 지시하고 포트 80의 서버인 lotus-server에 연결합니다. Lotus 웹 클라이언트의 요구 사항은 클라이언트 수신 포트가 서버 포트와 일치해야 한다는 것입니다.

표 6–3 예제 Netlet 규칙 *(계속)*

규칙 이름	원격 응용 프로그램 URL	애플릿 다운로드 사용	로컬 포트를 대상 서버 포트에 매핑	설명
Lotus Notes 비 웹 클라이언트	null	확인란 선택 안 함	■ 로컬 포트: 1352 ■ 대상 호스트: lotus-domino ■ 대상 포트: 1352	이 규칙을 사용하여 Lotus Notes 클라이언트는 Netlet을 통해 Lotus Domino 서버에 연결할 수 있습니다. 클라이언트에서 서버로 연결을 시도할 때에는 서버 이름으로 `localhost`를 지정하면 안됩니다. Lotus Domino 서버의 실제 서버 이름을 지정해야 합니다. 서버 이름은 서버의 시스템 이름과 같아야 합니다. 클라이언트는 Netlet을 사용할 때 이름을 `127.0.0.1`로 설정해야 합니다. 이름을 이렇게 지정하려면 다음 2가지 방법을 사용합니다. ■ 클라이언트 호스트 테이블에서 서버 이름이 `127.0.0.1`이 되도록 설정합니다. ■ `127.0.0.1`을 가리키는 서버 이름의 DNS 항목을 내보냅니다. 서버 이름은 설치 시 Domino 서버를 구성하는 데 사용한 서버 이름과 같아야 합니다.

표 6–3 예제 Netlet 규칙 *(계속)*

규칙 이름	원격 응용 프로그램 URL	애플릿 다운로드 사용	로컬 포트를 대상 서버 포트에 매핑	설명
Microsoft Outlook 및 Exchange Server Windows NT, 2000 및 XP에서 작동하지 않습니다. Windows NT, 2000 및 XP용 Rewriter를 통해 Outlook Web Access를 사용합니다.	null	확인란 선택 안 함	■ 로컬 포트: 135 ■ 대상 호스트: exchange ■ 대상 포트: 135	이 규칙은 Netlet에 클라이언트의 포트 135에서 수신하여 포트 135에서 서버 `exchange`에 연결하라고 지시합니다. Outlook 클라이언트는 이 포트를 통해 Exchange 서버에 최초로 연결을 시도하고 서버와 통신하는 데 사용할 후속 포트를 결정합니다. 클라이언트 컴퓨터: ■ 사용자는 Outlook 클라이언트에 구성된 Exchange 서버의 호스트 이름을 `localhost`로 변경해야 합니다. 이 옵션의 위치는 Outlook 버전에 따라 다릅니다. ■ 사용자는 호스트 파일을 사용하여 Exchange 서버의 호스트 이름(단일 및 정규)을 IP 주소 `127.0.0.1`로 매핑해야 합니다. ■ Windows 95나 98에서 이 파일은 `\\Windows\\Hosts`에 있습니다. ■ Windows NT4에서 이 파일은 `\\WinNT\\System32\\drivers\\etc\\Hosts`에 있습니다. 항목은 다음과 같습니다. `127.0.0.1 exchange exchange.company.com` Exchange 서버는 자체 이름을 Outlook 클라이언트로 다시 보냅니다. 이러한 매핑을 통해 Outlook 클라이언트는 Netlet 클라이언트를 통해 서버에 다시 연결할 수 있게 됩니다.

표 6–3 예제 Netlet 규칙 *(계속)*

규칙 이름	원격 응용 프로그램 URL	애플릿 다운로드 사용	로컬 포트를 대상 서버 포트에 매핑	설명
FTP	null	확인란 선택 안 함	■ 로컬 포트: 30021 ■ **대상 호스트**:*your-ftp_server.your-domain* ■ 대상 포트: 21	제어된 최종 사용자 계정과 함께 단일 FTP Server에 FTP 서비스를 제공할 수 있습니다. 그러면 최종 사용자 시스템에서 단일 위치로 보안 원격 FTP 전송이 이루어집니다. 아이디가 없으면 FTP URL은 익명의 FTP 연결로 해석됩니다. 포트 30021은 Netlet FTP 규칙에 사용할 로컬 포트로 정의**해야 합니다**. 동적 FTP는 Netlet 연결을 통해 지원됩니다.
Netscape 4.7 Mail Client	null	확인란 선택 안 함	■ 로컬 포트: 30143, 30025. ■ 대상 호스트: TARGET ■ 대상 포트: 10143	Netscape 클라이언트에서 사용자는 다음을 지정해야 합니다. IMAP 또는 수신 메일용 `localhost:30143` SMTP 또는 송신 메일용 `localhost:30025`
Graphon	third_party/xsession_start.html	확인란 선택	■ 로컬 포트: 10491 ■ 대상 호스트: TARGET ■ 대상 포트: 491	Netlet을 통해 Graphon에 액세스하는 데 사용하는 규칙입니다. `xsession_start.html`은 Graphon에 번들로 제공됩니다.
Citrix	third_party/citrix_start.html	확인란 선택	■ 로컬 포트: 1494 ■ 대상 호스트: TARGET ■ 대상 포트: 1494	Netlet을 통해 Citrix에 액세스하는 데 사용하는 규칙입니다. `citrix_start.html`은 Citrix에 번들로 제공됩니다.
RemoteControl	third_party/pca_start.html	확인란 선택	■ 로컬 포트: 5631 5632 ■ 대상 호스트: TARGET TARGET ■ 대상 포트: 5631 5632	Netlet을 통해 Remote Control에 액세스하는 데 사용하는 규칙입니다. `pca_start.html`은 Remote Control에 번들로 제공됩니다.

Netlet 로깅 정보

Netlet 애플릿 또는 jws의 클라이언트쪽 로그는 해당 클라이언트의 Java 콘솔에 표시됩니다.

Netlet의 서버쪽 로그는 `/var/opt/SUNWportal/portals/<portal_ID>/logs/<INSTANCE_ID>` 디렉토리의 `portal.0.0.log` 파일에 저장됩니다.

Sun Ray 환경에서 Netlet 실행

Sun Ray 환경에 있는 클라이언트 컴퓨터에 애플릿을 다운로드해야 하는 응용 프로그램을 실행하려면 HTML 파일을 변경해야 합니다. 다음은 필요한 수정 사항을 보여주는 예제 파일입니다.

새로운 HTML 파일

```
<!-- @(#)citrix_start.html 2.1
98/08/17 Copyright (c) 1998 i-Planet, Inc., All rights reserved.-->
<html>
<script language="JavaScript">
var KEY_VALUES;  // KEY_VALUES[\qkey\q] = \qvalue\q;
function retrieveKeyValues() {
      KEY_VALUES = new Object();
      var queryString  = \q\q + this.location;
      queryString = unescape(queryString);
      queryString = queryString.substring((queryString.indexOf(\q?\q)) + 1);
      if (queryString.length < 1) {
          return false; }
      var keypairs = new Object();
      var numKP = 0;
      while (queryString.indexOf(\q&\q) > -1) {
        keypairs[numKP] = queryString.substring(0,queryString.indexOf(\q&\q));
        queryString = queryString.substring((queryString.indexOf(\q&\q)) + 1);
        numKP++;
      }
      // Store what\qs left in the query string as the final keypairs[] data.
      keypairs[numKP++] = queryString;
      var keyName;
      var keyValue;
      for (var i=0; i < numKP; ++i) {
        keyName = keypairs[i].substring(0,keypairs[i].indexOf(\q=\q));
        keyValue = keypairs[i].substring((keypairs[i].indexOf(\q=\q)) + 1);
        while (keyValue.indexOf(\q+\q) > -1) {
```

```
            keyValue = keyValue.substring(0,keyValue.indexOf(\q+\q)) + \q \q
             + keyValue.substring(keyValue.indexOf(\q+\q) + 1);

          }
          keyValue = unescape(keyValue);
            // Unescape non-alphanumerics
          KEY_VALUES[keyName] = keyValue;
        }
}
function getClientPort(serverPort) {
    var keyName = "clientPort[\q" + serverPort +"\q]";
    return KEY_VALUES[keyName];
}
function generateContent() {
    retrieveKeyValues();
    var newContent =
        "<html>\\n"
          + "<head></head>\\n"
          + "<body>\\n"
          + "<applet code=\\"com.citrix.JICA.class\\" archive=\\
               "JICAEngN.jar\\" width=800 height=600>\\n"
          + "<param name=\\"cabbase\\" value=\\"JICAEngM.cab\\">\\n"
          + "<param name=\\"address\\" value=\\"localhost\\">\\n"
          + "<param name=ICAPortNumber value="
          + getClientPort(\q1494\q)
          + ">\\n"
          + "</applet>\\n"
          + "</body>\\n"
          + "</html>\\n";
    document.write(newContent);
}
</script>
<body onLoad="generateContent();">
</body>
</html>
```

지원되지 않는 HTML 파일

```
<html>
<body>
<applet code="com.citrix.JICA.class" archive=
   "JICAEngN.jar" width=800 height=600>
<param name="cabbase" value="JICAEngM.cab">
<param name="address" value="localhost">
<param name=ICAPortNumber value=1494>
</applet>
</body></html>
```

2

Secure Remote Access Server 구성

대부분의 속성은 Portal Server 관리 콘솔의 Secure Remote Access 탭에서 사용 가능한 옵션을 통해 설정할 수 있습니다. 생성되는 새로운 조직 또는 사용자는 기본적으로 이 값을 상속합니다.

조직, 역할 및 사용자 수준에서 Secure Remote Access에 관련된 속성을 구성할 수 있습니다. 단, 다음의 예외가 있습니다.

- 충돌 해결 수준은 사용자 수준에서 설정할 수 없습니다. 29 페이지 "충돌 해결 설정"을 참조하십시오.
- MIME 유형 구성 파일 위치 속성은 조직 수준에서만 설정할 수 있습니다.

조직 수준에서 설정한 값은 그 아래의 모든 역할과 사용자에게 상속됩니다. 사용자 수준에서 설정된 값은 조직 또는 규칙 수준에서 설정된 값보다 우선합니다.

서비스 구성 수준에서 속성 값을 변경할 수 있습니다. 이러한 새로운 값은 새 조직이 추가될 때에만 반영됩니다.

이 절은 다음 장으로 구성되어 있습니다.

◆ ◆ ◆ 7 장

7 Secure Remote Access Server 액세스 제어 구성

이 장에서는 Sun Java System Portal Server 관리 콘솔에서 사용자의 액세스를 허용하거나 거부하는 방법에 대해 설명합니다.

액세스 제어 구성

이 필드를 사용하여 최종 사용자가 게이트웨이를 통해 액세스할 수 없는 URL의 목록을 지정할 수 있습니다. 게이트웨이에서는 [허용된 URL] 목록을 확인하기 전에 [거부된 URL] 목록을 확인합니다.

게이트웨이를 통해 최종 사용자가 액세스할 수 있는 모든 URL를 지정할 수 있습니다. 기본적으로 이 목록에는 와일드카드 항목(*)이 있으므로 모든 URL에 액세스할 수 있다는 것을 의미합니다. 모든 URL에 대한 액세스를 허용하고, 특정 URL에 대한 액세스만 제한하려면 제한되는 URL을 [거부된 URL] 목록에 추가합니다. 같은 방식으로 특정 URL의 액세스만 허용하려면 [거부된 URL] 필드를 비워두고 [허용된 URL] 필드에서 필요한 URL을 지정합니다.

SRA 소프트웨어의 액세스 제어 서비스에서는 다양한 호스트에 대한 단일 사인온 기능을 제어할 수 있습니다. 단일 사인온 기능을 사용하려면 게이트웨이 서비스에서 [HTTP 기본 인증 사용] 옵션을 활성화해야 합니다.

액세스 제어 서비스에서는 특정 호스트에 대한 단일 사인온을 비활성화할 수 있습니다. 이것은 각 세션마다 단일 사인온이 활성화되어 있지 않는 한, HTPP 기본 인증이 필요한 호스트에 연결할 때마다 최종 사용자가 인증을 받아야 한다는 것을 의미합니다.

특정 호스트에 단일 사인온을 비활성화한 경우 사용자는 단일 포털 서버 세션 내에서만 해당 호스트에 연결할 수 있습니다. 예를 들어, `abc.sesta.com`에 단일 사인온을 사용 불가능으로 설정했다고 가정해 보겠습니다. 사용자가 이 사이트에 처음 연결할 때에는 인증이 필요합니다. 사용자는 다른 페이지를 찾아본 후 이 페이지로 돌아올 수 있으며 이 페이지가 같은 포털 서버 세션에 있으면 인증이 필요치 않습니다.

▼ 액세스 제어를 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택합니다.**
3. **[액세스 제어] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
COS 우선 순위	속성 값의 상속 여부를 결정하는 데 사용하는 값을 지정합니다. 이 속성에 대한 자세한 내용은 Sun Java System Directory Server 관리 설명서를 참조하십시오.
세션별 단일 사인온	세션에서 단일 사인온을 사용하려면 [사용]을 선택합니다.
단일 사인온을 사용하지 않는 호스트	`abc.siroe.com`과 같은 형식으로 호스트 이름을 입력합니다.
허용된 인증 수준	허용되는 인증 수준을 입력합니다. 모든 수준을 허용하려면 별표를 사용합니다. 기본값은 별표(*)입니다.
URL에 대한 액세스 허용/거부	URL 필드에 게이트웨이를 통한 액세스를 허용 또는 거부할 URL을 입력합니다. URL 입력 형식: `http://abc.siroe.com`. [작업] 드롭다운 목록에서 [허용] 또는 [거부] 옵션을 적절하게 누릅니다. `http://*.siroe.com`과 같은 정규식을 사용할 수도 있습니다. 이 경우 사용자는 `siroe.com` 도메인에 있는 모든 호스트에 액세스가 거부됩니다. 게이트웨이는 허용된 URL 목록을 확인하기 전에 먼저 액세스가 거부된 URL을 확인합니다. **주 –** [허용된 URL] 필드에는 기본적으로 *가 있으며, 따라서 게이트웨이를 통해 모든 URL에 액세스할 수 있습니다.

주 – SRA를 설치할 때 모든 사용자가 기본적으로 액세스 제어 서비스를 사용할 수 있는 것은 아닙니다. 이 서비스는 설치 시 기본적으로 만들어지는 `amadmin` 사용자에게만 활성화되어 있습니다. 다른 사용자는 이 서비스 없이 게이트웨이를 통해 데스크탑에 액세스할 수 없습니다. `amadmin`으로 로그인하여 이 서비스를 모든 사용자에게 할당합니다.

5. **[저장]을 눌러 완료합니다.**

◆ ◆ ◆ 8 장

8 Secure Remote Access Gateway 구성

이 장에서는 Sun Java System Portal Server 관리 콘솔에서 게이트웨이 속성을 구성하는 방법을 설명합니다.

이번 장은 다음 절로 구성됩니다.

시작하기 전에

- 게이트웨이 프로필을 만들려면 32 페이지 "게이트웨이 프로필 만들기"를 참조하십시오.

프로필 핵심 옵션 구성

이 절에서는 다음 작업을 설명합니다.

시작 모드 구성

설치 중에 HTTPS 모드에서 게이트웨이를 실행하도록 선택한 경우 설치 후에 게이트웨이가 HTTPS 모드에서 실행됩니다. HTTPS 모드에서 게이트웨이는

브라우저로부터 SSL 연결을 허용하고 SSL이 아닌 연결은 거부합니다. 그러나, 게이트웨이를 HTTP 모드에서 실행되도록 설정할 수도 있습니다. 그러면 SSL 세션 관리와 SSL 트래픽 암호화 및 해독에 관련된 오버헤드가 생기지 않기 때문에 게이트웨이 성능을 높일 수 있습니다.

▼ 시작 모드를 구성하려면

1 **Portal Server 관리 콘솔에 관리자로 로그인합니다.**

2 **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 해당 속성을 수정합니다.**

3 **[핵심] 탭을 선택합니다.**

4 **다음 속성을 수정합니다.**

HTTP 연결	게이트웨이가 비SSL 연결을 수락하도록 하려면 [HTTP 연결] 확인란을 선택합니다.
HTTP 포트	HTTP 포트 번호를 입력합니다. 기본값은 80입니다.
HTTPS 연결	게이트웨이가 SSL 연결을 수락하도록 하려면 [HTTPS 연결] 확인란을 선택합니다. 기본적으로 이 옵션이 선택됩니다.
HTTPS 포트	HTTPS 포트 번호를 입력합니다. 기본값은 443입니다.

주 – 다음 속성은 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin set-attribute”를 사용하여 수정할 수 있습니다

/space/PS/portal/bin/`psadmin set-attribute -u amadmin -f` */space/PS/portal/bin/ps_password -p portal1* `-m gateway --gateway-profile` *profileID* `-a` *sunPortalGatewayDomainsAndRulesets -A $entry*

- sunPortalGatewayDefaultDomainAndSubdomains=Default Domains
- sunPortalGatewayLoggingEnabled=Enable Logging
- sunPortalGatewayEProxyPerSessionLogging=Enable per Session Logging
- sunPortalGatewayEProxyDetailedPerSessionLogging=Enable Detailed per Session Logging
- sunPortalGatewayNetletLoggingEnabled=Enable Netlet Logging
- sunPortalGatewayEnableMIMEGuessing=Enable MIME Guessing
- sunPortalGatewayParserToURIMap=Parser to URI Mappings
- sunPortalGatewayEnableObfuscation=Enable Masking
- sunPortalGatewayObfuscationSecretKey=Seed String for Masking
- sunPortalGatewayNotToObscureURIList=URIs not to Mask
- sunPortalGatewayUseConsistentProtocolForGateway=Make
- 게이트웨이 프로토콜을 원본 URI \n 프로토콜과 같게 표시 sunPortalGatewayEnableCookieManager=Store External Server Cookies
- sunPortalGatewayMarkCookiesSecure=Mark Cookies as secure

5 터미널 창에서 게이트웨이를 다시 시작합니다.

`./psadmin start-sra-instance -u amadmin -f` *passwordfile* `-N` *profilename* `-t gateway`

핵심 구성 요소 구성

Netlet을 사용하여 사용자는 인터넷과 같은 불안전한 네트워크에서 공통 TCP/IP 서비스를 안전하게 실행할 수 있습니다. TCP/IP 응용 프로그램(예: 텔넷 및 SMTP), HTTP 응용 프로그램 및 모든 고정 포트 응용 프로그램을 실행할 수 있습니다. Netlet의 사용이 설정되면 게이트웨이에서 들어오는 트래픽이 Netlet 트래픽인지 또는 Portal Server 트래픽인지를 결정해야 합니다. Netlet을 사용 불가능으로 설정하면 게이트웨이가 들어오는 모든 트래픽이 HTTP이거나 HTTPS 트래픽이라고 가정하기 때문에 이러한 오버헤드가 줄어듭니다. Portal Server에서 응용 프로그램을 전혀 사용하지 않을 Netlet 사용을 해제합니다.

▼ 구성 요소를 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**
3. **[핵심] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
Netlet	Netlet 서비스를 시작하려면 [사용] 확인란을 선택합니다. 기본적으로 이 옵션이 선택됩니다.
Proxylet	Proxylet 서비스를 시작하려면 [사용] 확인란을 선택합니다. 기본적으로 이 옵션이 선택됩니다.

5. **다음 명령 옵션을 사용하여 터미널 창에서 게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

기본 옵션 구성

쿠키 관리 속성 정보

많은 웹 사이트에서는 사용자 세션을 추적하고 관리하기 위해 쿠키를 사용합니다. 게이트웨이가 HTTP 헤더에 쿠키를 설정하는 요청을 웹 사이트로 보낼 때 게이트웨이는 이러한 쿠키를 다음과 같이 폐기시키거나 통과시킵니다.

- 게이트웨이 서비스에서 [쿠키 관리 사용] 속성이 선택되지 않으면 쿠키가 다시 작성되지 않습니다. 따라서 브라우저로부터의 쿠키는 인트라넷에 도달하지 못합니다(반대의 경우도 마찬가지).
- [쿠키 관리 사용] 속성이 선택되면 게이트웨이가 쿠키를 다시 작성합니다. 게이트웨이는 브라우저로부터의 쿠키가 대상 인트라넷 호스트에 도달하도록 합니다(반대의 경우도 마찬가지).

이 설정은 포털 서버가 포털 서버 사용자 세션을 관리하기 위해 사용하는 쿠키에는 적용되지 않습니다. [사용자 세션 쿠키가 전달될 URL] 옵션의 구성으로 제어합니다.

이 설정은 사용자가 액세스할 수 있는 모든 웹 사이트에 적용됩니다(즉, 어떤 사이트의 쿠키는 폐기시키고 어떤 사이트의 쿠키는 유지할 수 없습니다).

주 – 쿠키 없는 게이트웨이에서라도 [쿠키 도메인] 목록에서 URL을 제거하지 마십시오. 쿠키 도메인 목록에 대한 자세한 내용은 **Access Manager 관리 설명서**를 참조하십시오.

HTTP 기본 인증 속성 정보

HTTP 기본 인증은 게이트웨이 서비스에서 설정할 수 있습니다.

HTTP 기본 인증을 통해 방문자가 사이트를 보기 전에 아이디와 비밀 번호를 입력하도록 요구함으로써 웹 사이트를 보호할 수 있습니다(HTTP 응답 코드 401, WWW 인증서: BASIC). Portal Server는 사용자가 기본 보호 웹 사이트를 다시 방문할 때 자격 증명을 다시 입력할 필요가 없도록 사용자 이름과 비밀 번호를 저장할 수 있습니다. 이러한 자격 증명 정보는 디렉토리 서버의 사용자 프로필에 저장됩니다.

이 설정은 사용자가 BASIC으로 보호된 사이트를 방문할 수 있는지 여부가 아니라 사용자가 입력한 자격 증명 정보를 사용자 프로필에 저장할지 여부만을 결정합니다.

이 설정은 사용자가 액세스할 수 있는 모든 웹 사이트에 적용됩니다(즉, HTTP 기본 인증 캐싱을 어떤 사이트에 사용하고 다른 사이트에 사용하지 않을 수는 없습니다).

주 – 기본 인증 대신 Windows NT 시도/응답(HTTP 응답 코드 401, WWW 인증: NTLM)으로 보호되는 Microsoft\qs Internet Information Server(IIS)에서 서비스를 제공하는 URL에 대한 탐색은 지원되지 않습니다.

관리 콘솔에서 액세스 제어 서비스를 사용하여 단일 사인온을 사용할 수도 있습니다.

Portal Server 속성 정보

서비스 요청에 대한 게이트웨이에 여러 Portal Server를 구성할 수 있습니다. 게이트웨이를 설치하는 동안 게이트웨이가 함께 작동해야 하는 Portal Server를 지정했을 것입니다. 이 Portal Server는 기본적으로 Portal Server 필드에 나열됩니다. `http://`*portal-server-name:port number* 형식으로 목록에 Portal Server를 더 추가할 수 있습니다. 게이트웨이는 요청을 처리하기 위해 연속해서 나열된 각 Portal Server에 접속을 시도합니다.

사용자 세션 쿠키가 전달될 URL 속성 정보

포털 서버는 사용자 세션을 추적하기 위해 쿠키를 사용합니다. 이 쿠키는 게이트웨이가 서버에 HTTP 요청을 보낼 때(예를 들어, 사용자의 데스크탑 페이지를 생성하기 위해 데스크탑 서블릿이 호출될 때) 서버로 전달됩니다. 서버의 응용 프로그램은 쿠키를 사용하여 사용자를 검증하고 신원을 확인합니다.

[사용자 세션 쿠키가 전달될 URL] 목록에 서버 외의 시스템 URL을 지정하지 않은 경우에는 Portal Server의 쿠키가 이러한 시스템에 대한 HTTP 요청으로 전달되지 않습니다. 따라서 이 목록에 URL을 추가하면 서블릿과 CGI가 Portal Server의 쿠키를 받아 API를 통해 사용자를 식별할 수 있습니다.

URL은 뒤에 오는 암시적 와일드카드를 사용하여 매칭됩니다. 예를 들어, 목록의 기본 입력이 다음과 같습니다.

```
http://server:8080
```

이 입력은 쿠키가 `http://server:8080`으로 시작되는 모든 URL로 전달되도록 합니다.

추가:

```
http://newmachine.eng.siroe.com/subdir
```

쿠키가 이 정확한 문자열로 시작하는 모든 URL로 전달되도록 합니다.

이 예에서, 문자열 "`http://newmachine.eng/subdir`"은 전달 목록의 정확한 문자열로 시작하지 않기 때문에 쿠키는 이 문자열로 시작되는 URL로 전달되지 않습니다. 쿠키가 이러한 변형된 컴퓨터 이름으로 시작되는 URL로 전달되도록 하려면 전달 목록에 추가 항목을 추가해야 합니다.

마찬가지로, 쿠키는 적합한 항목이 추가되지 않는다면 "`https://newmachine.eng.siroe.com/subdir`"로 시작되는 URL로 전달되지 않습니다.

URL 속성에서 세션 얻기 정보

[URL에서 세션 얻기] 옵션을 선택하면 쿠키 지원 여부에 상관 없이 세션 정보가 URL의 일부로 인코딩됩니다. 즉, 게이트웨이는 클라이언트의 브라우저에서 보내는 세션 쿠키를 사용하지 않고 검증을 위해 URL에 있는 세션 정보를 사용합니다.

▼ 기본 옵션을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**
3. **[핵심] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
쿠키 관리	쿠키 관리를 사용하려면 [사용] 확인란을 선택합니다. 기본적으로 이 옵션이 선택됩니다.

속성 이름	설명
HTTP 기본 인증	[HTTP 기본 인증 사용] 확인란을 선택하여 HTTP 기본 인증의 사용을 설정합니다.
Portal Server	필드에 `http://portal-server-name:port-number` 형식으로 Portal Server를 입력하고 [추가]를 누릅니다. 이 단계를 반복하여 더 많은 Portal Server를 Portal Server 목록에 추가합니다.
사용자 세션 쿠키가 전달될 URL	사용자 세션 쿠키가 전달될 URL을 입력하고 [추가]를 누릅니다. 이 단계를 반복하여 [사용자 세션이 전달될 URL] 목록에 더 많은 URL을 추가합니다.
게이트웨이 최소 인증 수준	인증 수준을 입력합니다. 기본적으로 모든 수준에서 인증을 허용하는 별표(*)가 추가됩니다.
URL에서 세션 가져오기	URL에서 세션 정보를 가져오려면 [예]를 선택합니다. 기본적으로 [아니요] 옵션이 선택됩니다.

배포 옵션 구성

프록시 설정 구성

▼ 프록시 설정을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**
3. **[배포] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
프록시 사용	[프록시 사용] 확인란을 선택하여 웹 프록시의 사용을 활성화합니다.

속성 이름	설명	
웹 프록시 URL	[웹 프록시 URL 사용] 편집 상자에 필요한 URL을 `http://host name.subdomain.com` 형식으로 입력하고 [추가]를 누릅니다. URL이 [웹 프록시 URL 사용] 목록에 추가됩니다.	프록시 사용 옵션이 사용 불가능으로 설정되어 있어도 게이트웨이가 [도메인 및 부속 도메인 프록시] 목록에 나열된 웹 프록시를 통해서만 특정 URL에 접속해야 한다는 것을 지정할 수 있습니다. [웹 프록시 URL 사용] 필드에서 이러한 URL을 지정해야 합니다. 이 값이 프록시 사용에 미치는 영향에 대한 자세한 내용은 34 페이지 "Access Manage에 접속할 프록시 지정"을 참조하십시오.
도메인 및 하위 도메인의 프록시	[도메인 및 부속 도메인의 프록시] 목록 상자에 항목이 추가됩니다. 프록시 정보를 입력하기 위한 형식은 다음과 같습니다. `domainname proxy1:port1|subdomain1 proxy2:port2|subdomain2 proxy3:port3|* proxy4:port4` *는 * 이후에 정의된 프록시를 특별히 언급된 경우를 제외하고 모든 도메인과 부속 도메인에 사용해야 한다는 것을 나타냅니다. 프록시의 포트를 지정하지 않으면 포트 8080이 기본적으로 사용됩니다.	프록시 정보가 다양한 호스트에 적용되는 방법에 대한 자세한 내용은 34 페이지 "Access Manage에 접속할 프록시 지정"을 참조하십시오.
프록시 비밀번호 목록	[프록시 비밀번호 목록] 필드에 각 프록시 서버에 대한 정보를 입력하고 [추가]를 누릅니다. 프록시 정보를 입력하기 위한 형식은 다음과 같습니다. `proxyserver|username|password` `proxyserver`는 도메인 및 하위 도메인의 프록시 목록에 정의된 프록시 서버에 해당합니다.	프록시 서버가 일부 또는 모든 사이트에 액세스하는 데 인증이 필요한 경우 게이트웨이가 지정된 프록시 서버를 인증하는 데 필요한 사용자 이름 및 비밀 번호를 지정해야 합니다.
자동 프록시 구성 지원	[자동 프록시 구성 지원] 확인란을 선택하여 PAC 지원을 활성화합니다.	[자동 프록시 구성 사용] 옵션을 선택하면 [도메인 및 부속 도메인의 프록시] 필드에 제공된 정보가 무시됩니다. 게이트웨이에서는 인트라넷 구성에 프록시 자동 구성(PAC) 파일만 사용합니다. PAC 파일에 대한 내용은 46 페이지 "자동 프록시 구성 사용"을 참조하십시오.
자동 프록시 구성 파일 위치	[위치] 필드에 PAC 파일 이름과 위치를 입력합니다.	

Rewriter 프록시 및 Netlet 프록시 구성

NetLet 프록시 정보

Netlet 프록시는 인트라넷에 상주하는 Netlet 프록시로 가는 게이트웨이를 통해 클라이언트로부터의 보안 터널을 확장하여 게이트웨이와 인트라넷 사이에서 Netlet 트래픽의 보안을 강화합니다.Netlet 프록시가 활성화되면 Netlet 패킷은 Netlet 프록시에서 해독되어 대상 서버로 전달됩니다. 이를 통해 방화벽에서 열어야 하는 포트 수가 줄어듭니다.

Rewriter 프록시 정보

Rewriter 프록시는 게이트웨이와 인트라넷 간에 보안 HTTP 트래픽을 사용하도록 해줍니다. Rewriter 프록시를 지정하지 않으면 사용자가 인트라넷의 시스템에 액세스하려고 할 때 게이트웨이 구성 요소에서 인트라넷에 직접 연결합니다. Rewriter 프록시는 설치 후에 자동으로 실행되지 않습니다. 아래 설명에 따라 Rewriter 프록시를 활성화해야 합니다.

▼ Rewriter 프록시 및 Netlet 프록시를 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**

   주 – Rewriter 프록시와 게이트웨이가 같은 게이트웨이 프로필을 사용해야 합니다.

3. **[배포] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
Rewriter 프록시	Rewriter 프록시 서비스를 사용하려면 [Rewriter 프록시] 확인란을 선택합니다.

속성 이름	설명
Rewriter 프록시 목록	a. [Rewriter 프록시] 편집 상자에 호스트와 포트를 `hostname:port` 형식으로 입력합니다. **참고 –** 필요한 포트가 있고 아직 사용되지 않았는지 확인하려면 명령줄에서 다음을 입력합니다. **`netstat -a | grep`** *port-number* **`| wc -l`** *port-number*는 필수 포트입니다. b. [추가]를 누릅니다.
Netlet 프록시	[Netlet 프록시 사용] 확인란을 선택하여 Netlet 프록시 서비스를 활성화합니다.
Netlet 프록시 호스트	a. [Netlet 프록시 호스트] 필드에 Netlet 프록시 호스트와 포트를 `hostname:port` 형식으로 입력합니다. **참고 –** 필요한 포트가 있고 아직 사용되지 않았는지 확인하려면 명령줄에서 다음을 입력합니다. **`netstat -a | grep`** *port-number* **`| wc -l`** *port-number*는 필수 포트입니다. b. [추가]를 누릅니다.
웹 프록시를 통한 Netlet 터널링	[웹 프록시를 통한 Netlet 터널링 사용] 확인란을 선택하여 터널링을 활성화합니다.

5 서버에서 *portal-server-install-root*`/SUNWportal/bin/certadmin`을 실행하여 Rewriter 프록시의 인증서를 만듭니다.

Rewriter 프록시 설치 중에 인증서를 만들지 않은 경우에만 이 단계가 필요합니다.

6 Rewriter 프록시가 설치된 컴퓨터에 루트로 로그인하여 Rewriter 프록시를 시작합니다.

rewriter-proxy-install-root`/SUNWportal/bin/rwproxyd -n` *gateway-profile-name* `start`

7 게이트웨이가 설치된 컴퓨터에 루트로 로그인하여 게이트웨이를 시작합니다.

`./psadmin start-sra-instance -u amadmin -f` *passwordfile* `-N` *profilename* `-t gateway`

보안 옵션 구성

PDC 및 인증되지 않은 URL 구성

▼ PDC 및 인증되지 않은 URL을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**
3. **[보안] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
인증서 사용 가능 게이트웨이 호스트	a. 게이트웨이 이름을 [인증서 사용 가능 게이트웨이 호스트]에 추가합니다. 게이트웨이를 `host1.sesta.com` 형식으로 추가합니다. b. [추가]를 누릅니다.
비인증 URL	일부 URL에 인증이 필요 없음을 지정할 수 있습니다. 일반적으로 이미지가 있는 디렉토리가 이에 해당합니다. [인증되지 않은 URL] 필드에 필요한 폴더 경로를 `folder/subfolder` 형식으로 입력합니다. 정규화되지 않은 URL(예를 들어, /images)은 포털 URL로 취급됩니다. 포털이 아닌 URL을 추가하려면 해당 URL을 정규화하고 [추가]를 눌러 이 항목을 [인증되지 않은 URL] 목록에 추가합니다.
인증된 SSL 도메인	[인증된 SSL 도메인] 필드에 도메인 이름을 입력하고 [추가]를 누릅니다.

TLS 및 SSL 옵션 구성

▼ TLS 및 SSL 옵션을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**

3 [보안] 탭을 선택합니다.

4 다음 속성을 수정합니다.

속성 이름	설명
40비트 암호화	40비트(취약) SSL(Secure Sockets Layer) 연결을 허용하려는 경우에 이 옵션을 선택하십시오. 이 옵션을 선택하지 않으면 128비트 연결만 지원됩니다. 이 옵션을 선택하지 않으면 사용자의 브라우저가 필요한 연결 유형을 지원하도록 구성되어 있어야 합니다. **주 –** Netscape Navigator 4.7x를 사용하는 경우에는 다음을 수행합니다. a. [커뮤니케이터] 메뉴의 [도구]에서 [보안 정보]를 선택합니다. b. 왼쪽 창에서 [네비게이터] 링크를 누릅니다. c. [고급 보안(SSL) 구성]에서 [SSL v2 구성] 또는 [SSL v3 구성]을 누릅니다. d. 이제 필요한 암호가 사용됩니다.
Null 암호	[Null 암호화 사용] 확인란을 선택하여 Null 암호를 활성화합니다.
SSL 암호 선택	SRA(Secure Remote Access)는 여러 가지 표준 암호를 지원합니다. 사전 구성된 모든 암호의 지원을 선택하거나 필요한 암호만 개별적으로 선택할 수 있습니다. 각 게이트웨이 인스턴스에 특정 SSL 암호를 선택할 수 있습니다. 선택한 암호가 클라이언트 사이트에 있으면 SSL 핸드쉐이크가 성공적으로 이루어집니다.
SSL 버전 2.0	버전 2.0을 사용하려면 [SSL 버전 2.0 사용] 확인란을 선택합니다. 이 옵션은 기본적으로 활성화됩니다. SSL 버전 2.0을 활성화하거나 비활성화할 수 있습니다. SSL 2.0을 비활성화하면 이전 SSL 2.0만 지원하는 브라우저는 SRA(Secure Remote Access)에 대해 인증할 수 없습니다. 그러면 보안 수준이 높아집니다.
SSL2 암호	[SSL 암호화 선택 사용] 확인란 옵션을 선택합니다. SSL 암호 목록에서 필요한 암호를 선택할 수 있습니다.
SSL 버전 3.0	SSL 버전 3.0을 활성화하거나 비활성화할 수 있습니다. SSL 3.0을 비활성화하면 SSL 3.0만 지원하는 브라우저는 SRA 소프트웨어에 대해 인증할 수 없습니다. 그러면 보안 수준이 높아집니다. [SSL 버전 3.0 사용] 확인란을 선택하여 버전 3.0을 활성화합니다.
SSL3 암호	[SSL 암호화 선택 사용] 확인란 옵션을 선택합니다. SSL3 암호 목록에서 필요한 암호를 선택할 수 있습니다.
TLS 암호	[SSL 암호화 선택 사용] 확인란 옵션을 선택합니다. TLS 암호 목록에서 필요한 암호를 선택할 수 있습니다.

성능 옵션 구성

시간 초과 및 재시도 구성

▼ 시간 초과 및 재시도를 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**
3. **[성능] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
서버 재시도 간격(초)	Portal Server, Rewriter 프록시 또는 Netlet 프록시를 사용할 수 없게 되는 경우(충돌이 있거나 중단된 경우 등) 시작 요청 사이의 시간 간격(초)을 지정합니다.
게이트웨이 시간 초과(초)	게이트웨이와 브라우저의 연결이 시간 초과되기까지 소요되는 시간(초)을 지정합니다. [게이트웨이 시간 초과] 필드에 필요한 시간을 초 단위로 지정합니다.
캐시된 소켓 시간 초과(초)	게이트웨이와 Portal Server와의 연결이 시간 초과되기까지 소요되는 시간(초)을 지정합니다.

HTTP 옵션 구성

▼ HTTP 옵션을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**
3. **[성능] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
최대 스레드 풀 크기	원하는 스레드 수를 지정합니다. 게이트웨이 스레드 풀에서 사전에 생성할 수 있는 최대 스레드 수를 지정할 수 있습니다.
HTTP 지속 연결	[HTTP 지속 연결 사용] 확인란을 선택하여 HTTP 연결을 설정합니다. 게이트웨이에서 HTTP 지속 연결을 사용하여 웹 페이지의 모든 개체(이미지 및 스타일 시트 등)에 대해 소켓이 열리는 것을 방지할 수 있습니다.
지속 연결 당 최대 요청 수	최대 요청 수를 입력합니다.
지속적 소켓 연결 시간 초과(초)	원하는 시간 초과 값(초)을 입력합니다.
반환 시간을 위한 계정의 유예 시간 초과(초)	원하는 유예 시간 초과 값(초)을 입력합니다. 이 시간은 클라이언트(브라우저)와 게이트웨이 사이에서 네트워크 트래픽의 왕복 시간입니다. ■ 브라우저가 요청을 보낸 후 이 요청이 게이트웨이에 도달하는 시간 ■ 응답을 보내는 게이트웨이와 이를 실제로 수신하는 브라우저 사이의 시간 이 시간은 네트워크 상태 및 클라이언트의 연결 속도와 같은 요소에 의해 결정됩니다.
최대 연결 대기 길이	게이트웨이가 허용해야 하는 최대 동시 연결 수를 지정합니다. 원하는 연결 수를 지정합니다.

SRA(Secure Remote Access) 성능 모니터링

모니터링 기능을 사용하면 관리자가 SRA(Secure Remote Access)에 있는 다양한 구성 요소의 성능을 평가할 수 있습니다.

▼ SRA(Secure Remote Access) 성능을 모니터링하려면

1. **Portal Server 관리 콘솔에 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 하위 메뉴에서 [모니터링]을 누릅니다.**
3. **모니터링 페이지의 드롭다운 메뉴에서 프록시 인스턴스를 선택합니다.**
4. **MBeans 테이블에서 성능 값을 확인할 속성을 선택합니다.**

Rewriter 옵션 구성

기본 옵션 구성

▼ 기본 옵션을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**
3. **[Rewriter] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
모든 URI 다시 쓰기	[모든 URI 다시 쓰기 사용] 확인란을 선택하여 게이트웨이가 모든 URL을 다시 쓸 수 있게 합니다. 게이트웨이 서비스에서 [모든 URL 다시 쓰기 사용] 옵션을 설정하면 [도메인 및 부속 도메인의 프록시] 목록에 있는 항목을 확인하지 않고 Rewriter가 모든 URL을 다시 씁니다. [도메인 및 하위 도메인의 프록시] 목록에 있는 항목은 무시됩니다.
다시 쓰지 않는 URI	편집 상자에 URI를 추가합니다. **주** – #*를 이 목록에 추가하면 규칙 집합에 href 규칙이 포함되어 있더라도 URI 다시 쓰기를 허용합니다.

규칙 집합에 URI 매핑 구성

규칙 집합은 Portal Server 관리 콘솔의 Portal Server 구성 아래 Rewriter 서비스에서 만들어집니다. 자세한 내용은 **Portal Server 관리 설명서**를 참조하십시오.

규칙 집합을 만든 후 [규칙 집합에 URI 매핑] 필드를 사용하여 도메인과 규칙 집합을 연관시킵니다. 기본적으로 다음 두 항목이 [규칙 집합에 URI 매핑] 필드에 추가됩니다.

- *://*.Sun.COM/portal/*|default_gateway_ruleset
  여기서 sun.com은 포털의 설치 도메인이고 /portal은 포털 설치 컨텍스트입니다.
- *|generic_ruleset

이것은 기본 도메인의 모든 페이지에 대해 기본 게이트웨이 규칙 집합이 적용된다는 것을 의미합니다. 기타 모든 페이지에는 일반 규칙 집합이 적용됩니다. 기본 게이트웨이 규칙 집합과 일반 규칙 집합은 사전 구성된 규칙 집합입니다.

주 – 데스크탑에 나타나는 모든 컨텐트에는 컨텐트가 어디서 가져왔는지 상관 없이 기본 도메인의 규칙 집합이 사용됩니다.

예를 들어, 데스크탑이 URL `yahoo.com`에서 컨텐트를 스크랩하도록 구성되었다고 가정합시다. Portal Server는 `sesta.com`에 있습니다. `sesta.com`에 대한 규칙 집합이 불려온 컨텐트에 적용됩니다.

주 – 규칙 집합을 지정하는 도메인은 [도메인 및 하위 도메인의 프록시] 목록에 있어야 합니다.

▼ 규칙 집합에 URI 매핑을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 해당 속성을 수정합니다.**
3. **[Rewriter] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명

URI	필요한 도메인이나 호스트 이름 그리고 규칙 집합을 [규칙 집합에 URI 매핑] 필드에 입력하고 [추가]를 누릅니다.
	항목이 [규칙 집합에 URI 매핑] 필드에 추가됩니다.
	도메인이나 호스트 이름 그리고 규칙 집합을 지정하는 형식은 다음과 같습니다.
	`domain-name|ruleset-name`
	예:
	`eng.sesta.com|default`
	주 – 규칙 집합을 적용하는 우선 순위는 `hostname-subdomain-domain`**입니다.**
	도메인 기반 규칙 집합 목록 항목의 예는 다음과 같습니다.
	`sesta.com|ruleset1`
	`eng.sesta.com|ruleset2`
	`host1.eng.sesta.com|ruleset3`
	■ `ruleset3`은 `host1`의 모든 페이지에 적용됩니다.
	■ `ruleset2`는 `host1`에서 가져온 페이지를 제외하고 `eng` 하위 도메인의 모든 페이지에 적용됩니다.
	■ `ruleset1`은 `eng` 하위 도메인과 `host1`에서 가져온 페이지를 제외하고 `sesta.com` 도메인의 모든 페이지에 적용됩니다.

구문 분석기를 MIME 유형으로 구성

Rewriter에는 컨텐트 유형(HTML, JAVASCRIPT, CSS 및 XML)에 따라 웹 페이지의 구문을 분석하기 위한 4가지 구문 분석기가 있습니다. 기본적으로 공통 MIME 유형이 이러한 구문 분석기와 연결되어 있습니다. 게이트웨이 서비스의 [구문 분석기를 MIME 유형에 매핑] 필드에서 새로운 MIME 유형을 이러한 구문 분석기와 연관시킬 수 있습니다. 그러면 Rewriter의 기능이 다른 MIME 유형까지 확장됩니다.

여러 항목인 경우는 세미콜론(";")이나 쉼표(",")를 사용하여 구분합니다.)

예:

HTML=text/html;text/htm;text/x-component;text/wml; text/vnl/wap.wml

이것은 이러한 MIME을 가진 모든 컨텐트가 HTML Rewriter로 보내지고 URL을 다시 쓰도록 HTML 규칙이 적용된다는 의미입니다.

참고 – MIME 매핑에서 불필요한 구문 분석기를 제거하면 작동 속도를 높일 수 있습니다. 예를 들어, 특정 인트라넷의 컨텐트에 JavaScript가 없다는 것이 확실하면 MIME 매핑 목록에서 JAVASCRIPT 항목을 제거할 수 있습니다.

▼ 구문 분석기를 MIME 유형으로 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 프로필 이름을 눌러 선택하여 해당 속성을 수정합니다.**
3. **[Rewriter] 탭을 선택합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
구문 분석기	a. [구문 분석기를 MIME 유형에 매핑] 필드의 편집 상자에 필요한 MIME 유형을 추가합니다. 여러 항목을 구분할 때에는 세미콜론이나 콤마를 사용합니다. 항목을 `HTML=text/html;text/htm` 형식으로 지정합니다. b. [추가]를 눌러 목록에 필요한 항목을 추가합니다.

개인 디지털 인증서 인증 구성

PDC는 인증 기관(CA)에서 발행하며 CA 개인 키로 서명됩니다. CA에서는 인증서를 발행하기 전에 요청 주체의 신원을 검증합니다. 따라서 PDC가 있다는 것은 인증 메커니즘이 강력하다는 것을 나타냅니다.

PDC에는 소유자의 공용 키, 소유자 이름, 만료 날짜, 디지털 인증서를 발행한 인증 기관의 이름, 일련 번호와 함께 기타 정보도 포함될 수 있습니다.

사용자는 PDC와 스마트 카드 및 Java 카드와 같은 암호화된 장치를 Portal Server에서 인증을 얻는 데 사용할 수 있습니다. 인코딩된 장치는 카드에 저장된 PDC를 전자적 형태로 보관합니다. 사용자가 이러한 메커니즘 중 하나를 사용하여 로그인하면 로그인 화면과 인증 화면이 나타나지 않습니다.

PDC 인증 프로세스에는 다음 단계가 있습니다.

1. 브라우저에서 사용자가 예를 들어 `https://my.sesta.com`과 같이 연결 요청을 입력합니다.

   이 요청에 대한 응답은 `my.sesta.com`에 대한 게이트웨이가 인증서를 허용하도록 구성되었는지 여부에 달려있습니다.

   주 – 게이트웨이가 인증서를 허용하도록 구성된 경우 인증서가 있는 로그인만 허용되며 다른 종류의 로그인은 허용되지 않습니다.

   게이트웨이는 인증서가 알려진 인증 기관에서 발행된 것인지, 만료되지 않았는지 그리고 위조되지 않았는지 점검합니다. 인증서가 유효하면 게이트웨이는 사용자를 인증 프로세스의 다음 단계로 진행시킵니다.
2. 게이트웨이는 인증서를 서버의 PDC 인증 모듈로 전달합니다.

▼ PDC 및 코드화된 장치를 구성하려면

1 Portal Server 시스템의 `/etc/opt/SUNWam/config/AMConfig.properties` 파일에 다음 행을 추가합니다. `com.iplanet.authentication.modules.cert.gwAuthEnable=yes` **.**

2 PDC를 사용할 게이트웨이의 인증서 데이터베이스로 필요한 인증서를 가져옵니다. 인증서를 구성하려면 179 페이지 "게이트웨이 시스템에서 루트 CA 인증서를 가져오려면"을 참조하십시오.

3 Access Manager 관리 콘솔에 관리자로 로그인하여 다음을 수행합니다.

a. [Identity 관리] 탭을 선택한 다음 [조직]을 선택합니다.

b. [보기] 드롭다운 메뉴에서 조직에 대한 서비스를 누릅니다.

c. [추가]를 눌러 인증서를 등록합니다.

4 Access Manager 관리 콘솔에서 다음을 수행합니다.

a. 원하는 조직을 선택하고 [인증서] 옆의 화살표를 누릅니다.

b. [인증된 원격 호스트] 목록 상자에서 아무 항목도 선택하지 않고 [제거]를 누릅니다.

c. 텍스트 필드에 내용을 입력하고 [추가]를 누릅니다.

d. [저장]을 누릅니다.

5 **Access Manager 관리 콘솔에서 다음을 수행합니다.**

a. **원하는 조직을 선택한 다음 [보기] 드롭다운 메뉴에서 [서비스]를 선택합니다.**
서비스 목록이 표시됩니다.

b. **[인증 구성] 핵심 서비스 옆의 화살표를 누르고 [새로 만들기]를 누릅니다.**
새 서비스 인스턴스 페이지가 표시됩니다.

c. **서비스 인스턴스 이름으로 `gatewaypdc`를 입력합니다.**

d. **[제출]을 누릅니다.**
gatewaypdc 서비스 인스턴스 목록이 표시됩니다.

e. **gatewaypdc를 눌러 서비스를 편집합니다.**
gatewaypdc 등록 정보 페이지가 표시됩니다.

f. **[인증 구성] 옆의 [편집] 링크를 누르고 [추가]를 누릅니다.**
모듈 추가 페이지가 표시됩니다.

g. **[모듈 이름] 필드에서 [인증서]를 선택하고 [적용 기준]에 [필수]를 선택한 다음 [확인]을 누릅니다.**

h. **[확인]을 눌러 완료합니다.**

6 **Access Manager 관리 콘솔에서 다음을 수행합니다.**

a. **[핵심] 옆의 화살표를 누릅니다.**

b. **[조직 인증 모듈] 목록 상자에서 gatewaypdc를 선택합니다.**

c. **[사용자 프로필에서 동적으로 생성] 드롭다운 메뉴를 선택합니다.**

d. **[저장]을 눌러 완료합니다.**

7 **Portal Server 관리 콘솔에 관리자로 로그인하여 다음을 수행합니다.**

a. **[Secure Remote Access] 탭을 선택하고 적절한 게이트웨이 프로필을 선택합니다.**

b. **[보안] 탭을 선택합니다.**

c. **[인증서 사용 가능 게이트웨이 호스트] 목록 상자에서 게이트웨이 이름을 추가합니다.**

d. **[저장]을 누릅니다.**

8 **터미널 창에서 게이트웨이 프로필을 다시 시작합니다.**

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t
gateway
```

9 **CA에서 발행된 클라이언트 인증서를 PDC 사용 가능 게이트웨이에 액세스하는 브라우저에 설치합니다.**

10 **클라이언트 인증서를 JVM 키 저장소에 설치합니다. JVM 제어판은 Windows 시스템에서 [시작] > [설정] > [제어판] > [Java]를 선택하여 액세스할 수 있습니다.**

다음을 애플릿 런타임 매개 변수에 추가합니다.

- `Djavax.net.ssl.keyStore=`**키 저장소 경로**
- `Djavax.net.ssl.keyStorePassword=`**비밀번호**
- `Djavax.net.ssl.keyStoreType=`**유형**

11 **게이트프로필과 조직에 액세스합니다.**

https://gateway:instance-port/YourOrganization

인증서 이름으로 아이디와 비밀번호 프롬프트 없이 로그인되어 있어야 합니다.

▼ 게이트웨이 시스템에서 루트 CA 인증서를 가져오려면

1 **게이트웨이 시스템에서 루트 CA 인증서를 가져옵니다.**

a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`
Certadmin 메뉴가 표시됩니다.

b. **옵션 3을 선택하고 인증서 경로를 입력합니다.**

자세한 내용은 10 장을 참조하십시오.

2 **CA에 제출하기 위한 인증서 서명 요청을 생성합니다.**

a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`
Certadmin 메뉴가 표시됩니다.

b. **옵션 2를 선택하고 적절한 정보를 입력합니다.**

c. **파일을 저장합니다.**

3 **인증서 서명 요청을 CA에 제출하고 승인을 받습니다. CA 서명 후 인증서 응답을 저장합니다.**

4 **CA에서 승인을 받은 후 서버 인증서를 가져옵니다.**

   a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`
   Certadmin 메뉴가 표시됩니다.

   b. **옵션 4를 선택합니다.**

   c. **서버 인증서가 포함된 파일의 위치를 지정합니다.**

5 **Portal Server 시스템에 루트 CA 인증서를 가져옵니다.**

명령줄 옵션을 사용한 게이트웨이 속성 구성

이 절에서는 다음 작업을 위해 터미널 창에서 게이트웨이 속성을 구성하는 명령줄 옵션을 제공합니다.

▼ 외부 서버 쿠키 저장소를 관리하려면

외부 서버 쿠키 저장 옵션을 사용하면 게이트웨이가 게이트웨이를 통해 액세스할 수 있는 타사 응용 프로그램이나 서버에 대한 쿠키를 저장하고 관리합니다. 응용 프로그램이나 서버가 쿠키 없는 장치를 서비스할 수 없거나 상태 관리(레거시 관련 이유로)를 위해 쿠키에 의존하는 경우에도 게이트웨이는 쿠키 없는 장치를 서비스하고 있다는 사실로부터 응용 프로그램이나 서버를 투명하게 숨깁니다.

쿠키 없는 장치와 클라이언트 검색에 대한 자세한 내용은 *Access Manager Customization and API Guide*를 참조하십시오.

- **다음 명령을 입력하고 Enter를 눌러 외부 서버 쿠키 저장소를 관리합니다.**

  - **사용하려면**

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a CookieManagement true

  - **사용하지 않으려면**

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a CookieManagement false

  - **속성 값을 가져오려면**

    PS_INSTALL_DIR /bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a CookieManagement

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

▼ 쿠키를 안전하다고 표시하려면

쿠키가 안전한 것으로 표시되면 브라우저가 이 쿠키를 추가 보안을 통해 취급합니다. 보안의 구현은 브라우저에 따라 다릅니다. 이 작업을 위해 [쿠키 관리 사용] 속성을 사용해야 합니다.

- **다음 명령을 입력하고 Enter를 눌러 쿠키를 안전한 것으로 표시합니다.**

  - **사용하려면**

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MarkCookiesSecure true

  - **사용하지 않으려면**

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MarkCookiesSecure false

  - **속성 값을 가져오려면**

    PS_INSTALL_DIR /bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MarkCookiesSecure

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

▼ 사용하지 않을 프록시의 URL 목록을 만들려면

게이트웨이는 [웹 프록시 URL 사용 안함] 목록에 나열된 URL에 직접 연결을 시도합니다. 웹 프록시는 이러한 URL에 연결하는데 사용되지 않습니다.

- **다음 명령을 입력하고 Enter를 눌러 사용하지 않을 프록시의 URL을 관리합니다.**

주 – URL이 두 개 이상이면 각 URL을 공백으로 구분합니다.

- **사용하지 않을 URL을 지정하려면**

  PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DontUseWebProxyURL -A` *"LIST_OF_URLS"*

- **기존 URL 목록에 추가하려면**

  PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DontUseWebProxyURL -A` *"LIST_OF_URLS"*

- **기존 URL 목록에서 제거하려면**

  PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DontUseWebProxyURL -E` *"LIST_OF_URLS"*

- **기존 URL 목록을 가져오려면**

  PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DontUseWebProxyURL`

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

▼ URI와 규칙 집합의 매핑을 관리하려면

SRA(Secure Remote Access)는 Microsoft Exchange 2000 SP3 설치 및 Outlook Web Access(OWA)의 MS Exchange 2003을 지원합니다.

1 URI를 기존 목록에 추가하려면

PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -A "`*URI*`|`*RULE_SET_NAME* *URI*`|`*RULE_SET_NAME*`"`

2 URI를 기존 목록에서 제거하려면

PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -E "`*URI*`|`*RULE_SET_NAME* *URI*`|`*RULE_SET_NAME*`"`

3 기존 목록을 가져오려면

PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DomainsAndRulesets`

4 다음 명령을 입력하고 Enter를 눌러 Outlook Web Access에 대한 규칙 집합을 관리합니다.

- **규칙 집합을 추가하려면**

  PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -A "`*EXCHANGE2000_SERVER_NAME*`exchange_2000sp3_owa_ruleset"`

- **규칙 집합을 제거하려면**

  PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -E "`*EXCHANGE2000_SERVER_NAME* `exchange_2000sp3_owa_ruleset"`

- **규칙 집합과 URI의 매핑 목록을 설정하려면**

  PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DomainsAndRulesets "`*URI*`|`*RULE_SET_NAME* *URI*`|`*RULE_SET_NAME* `"`

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

▼ 기본 도메인을 지정하려면

기본 도메인은 URL에 도메인과 부속 도메인 없이 호스트 이름만 있을 때 유용합니다. 이 경우에 게이트웨이는 호스트 이름이 기본 도메인 목록에 있다고 가정하고 그에 따라 진행합니다.

예를 들어, URL의 호스트 이름이 `host1`이고 기본 도메인과 하위 도메인이 `red.sesta.com`으로 지정된 경우, 호스트 이름은 `host1.red.sesta.com`으로 확인됩니다.

- **다음 명령을 입력하고 Enter를 눌러 기본 도메인을 지정합니다.**

  - **기본 도메인을 설정하려면**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DefaultDomainsAndSubdomains "`*DOMAIN_NAME*`"`

  - **기본 도메인을 가져오려면**

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DefaultDomainsAndSubdomains`

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

▼ MIME 추측을 관리하려면

Rewriter는 페이지의 MIME 유형에 따라 구문 분석기를 선택합니다. WebLogic 및 Oracle같은 일부 웹 서버는 MIME 유형을 보내지 않습니다. 이 문제를 해결하려면 [구문 분석기와 URI의 매핑] 목록 상자에 데이터를 추가하여 MIME 추측 기능을 활성화합니다.

- **다음 명령을 입력하고 Enter를 눌러 MIME 추측을 관리합니다.**

  - **MIME 추측을 사용하려면**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableMIMEGuessing true`

  - **MIME 추측을 사용하지 않으려면**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableMIMEGuessing false`

- **값을 가져오려면**

  PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a EnableMIMEGuessing

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 "psadmin set-attribute" 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 "psadmin get-attribute"

▼ 구문 분석할 URI 매핑 목록을 만들려면

[MIME 추측] 확인란이 선택된 상태에서 서버가 MIME 유형을 보내지 않으면 구문 이 상자를 사용하여 분석기를 URI에 매핑합니다.

각 URI는 세미콜론으로 구분합니다.

예: HTML=*.html; *.htm;*Servlet이것은 html, htm 또는 Servlet 확장을 가진 모든 페이지에 대한 컨텐트를 다시 쓰기 위해 HTML Rewriter가 사용된다는 것을 의미합니다.

- **다음 명령을 입력하고 Enter를 눌러 구문 분석할 URI 매핑 목록을 만듭니다.**

  - **구문 분석할 URI 매핑 목록을 설정하려면**

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap

  - **기존 목록에 추가하려면**

    PS_INSTALL_DIR /bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap -A *LIST*

  - **기존 목록에서 제거하려면**

    PS_INSTALL_DIR /bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap -E *LIST*

  - **기존 목록을 가져오려면**

    PS_INSTALL_DIR /bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME*-a MIMEMap

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 "psadmin set-attribute"

▼ 마스크를 관리하려면

마스크를 사용하면 Rewriter에서 페이지의 인트라넷 URL이 보이지 않도록 URI를 다시 쓸 수 있습니다.

- **다음 명령을 입력하고 Enter를 눌러 마스크를 관리합니다.**

  - **마스크를 활성화하려면**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableObfuscation true`

  - **마스크를 비활성화하려면**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableObfuscation false`

  - **값을 가져오려면**

    PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableObfuscation`

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

▼ 마스크 씨드 문자열을 지정하려면

씨드 문자열은 URI의 마스크에 사용됩니다. 마스크 알고리즘을 통해 문자열을 생성합니다.

주 – 이 씨드 문자열이 변경되거나 게이트웨이가 다시 시작되면 마스크된 URI를 책갈피에 추가하지 못할 수 있습니다.

- **다음 명령을 입력하고 Enter를 눌러 마스크 씨드 문자열을 지정합니다.**

  - **마스크 씨드 문자열을 설정하려면**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a ObfuscationSecretKey` *SECRET_KEY*

  - **값을 가져오려면**

    PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a ObfuscationSecretKey`

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

▼ 마스크하지 않을 URI 목록을 만들려면

일부 응용 프로그램(애플릿 등)은 인터넷 URI가 필요하기 때문에 마스크할 수 없습니다. 이러한 응용 프로그램을 지정하려면 URI를 목록 상자에 추가합니다.

예를 들어, `*/Applet/Param*`을 목록 상자에 추가한 경우 규칙 집합 규칙에서 컨텐트 URI `http://abc.com/Applet/Param1.html`이 일치하면 URL이 마스크되지 않습니다.

주 – URI가 두 개 이상이면 각 URI를 공백으로 구분합니다.

- **다음 명령을 입력하고 Enter를 눌러 마스크하지 않을 URI 목록을 만듭니다.**

  - **마스크하지 않을 URI 목록을 설정하려면**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a NotToObscureURIList` *LIST_OF_URI*

  - **기존 목록에 추가하려면**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a NotToObscureURIList -A` *LIST_OF_URI*

  - **기존 목록에서 제거하려면**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a NotToObscureURIList -E` *LIST_OF_URI*

  - **기존 값을 가져오려면**

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a NotToObscureURIList`

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

▼ 게이트웨이 프로토콜을 원본 URI 프로토콜과 같게 하려면

게이트웨이가 HTTP와 HTTPS 모드 모두에서 실행되는 경우 Rewriter가 일관된 프로토콜을 사용하여 HTML 컨텐트의 참조 리소스에 액세스하게 할 수 있습니다.

예를 들어 원본 URL이 `http://intranet.com/Public.html`이라면 http 게이트웨이가 추가됩니다. 원본 URL이 `https://intranet.com/Public.html`이라면 https 게이트웨이가 추가됩니다.

주 – 이는 Javascript로 생성된 동적 URI가 아닌 정적 URI에만 적용됩니다.

- **다음 명령을 입력하고 Enter를 눌러 게이트웨이 프로토콜을 원본 URI 프로토콜과 같게 합니다.**

  - **사용하려면**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway true`

  - **사용하지 않으려면**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway false`

  - **값을 가져오려면**

    PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway`

자세한 정보 **관련 항목**

Sun Java System Portal Server 7.2 Command-Line Reference의 “psadmin set-attribute” 및 **Sun Java System Portal Server 7.2 Command-Line Reference**의 “psadmin get-attribute”

◆ ◆ ◆ 9 장

9 게이트웨이 서비스에서 Rewriter 구성

요약 내용이 여기에 나옵니다.

이번 장은 다음 절로 구성됩니다.

Rewriter 규칙에 대한 자세한 내용은 70 페이지 “언어 기반 규칙 정의”를 참조하십시오.

Rewriter 문제에 대한 자세한 내용은 94 페이지 “디버그 로그를 사용한 문제 해결”을 참조하십시오.

Rewriter 예제는 96 페이지 “작업 예제”를 참조하십시오.

규칙 집합과 URI의 매핑 목록 만들기

규칙 집합을 만든 후 [규칙 집합에 URI 매핑] 필드를 사용하여 도메인을 규칙 집합과 연관시킵니다. 기본적으로 다음 두 항목이 [규칙 집합에 URI 매핑] 필드에 추가됩니다.

- `*://*.Sun.COM/portal/*|default_gateway_ruleset`

  여기서 `sun.com`은 포털의 설치 도메인이고 /portal은 포털 설치 컨텍스트입니다.
- `*|generic_ruleset`

즉, 도메인 `sun.com`을 가진 포털 디렉토리의 모든 페이지에 `default_gateway_ruleset`이 적용됩니다. 기타 모든 페이지에는 일반 규칙 집합이 적용됩니다.
`default_gateway_ruleset` 및 `generic_ruleset`은 사전 구성된 규칙 집합입니다.

주 – 표준 포털 데스크탑에 나타나는 모든 컨텐트에 대해 컨텐트를 가져온 위치에 관계 없이 `default_gateway_ruleset`의 규칙 집합이 사용됩니다.

예를 들어, 표준 포털 데스크탑이 URL `yahoo.com`의 컨텐트를 사용하도록 구성되었다고 가정합니다. Portal Server는 `sesta.com`에 있습니다. `sesta.com`에 대한 규칙 집합이 불려온 컨텐트에 적용됩니다.

주 – 규칙 집합을 지정하는 도메인은 [도메인 및 하위 도메인의 프록시] 목록에 있어야 합니다.

구문 내에서 와일드카드 사용

규칙 집합에서 별표를 사용하여 정규 URI 또는 부분적 URI를 매핑할 수 있습니다.

예를 들어, 다음과 같이 `java_index_page_ruleset`을 `index.html` 페이지에 적용할 수 있습니다.

```
www.sun.com/java/index.html/java_index_page_ruleset
```

또는 다음과 같이 Java 디렉토리의 모든 페이지를 `java_directory_ruleset`에 적용할 수 있습니다.

```
www.sun.com/java/* /java_directory_ruleset
```

게이트웨이 서비스에서 Rewriter 구성

[Rewriter] 탭 아래에서 게이트웨이 서비스를 사용하여 기본 및 고급의 두 범주 내에서 다음 작업을 수행할 수 있습니다.

기본 작업

- 190 페이지 “게이트웨이가 모든 URL을 다시 쓰도록 하려면”
- 191 페이지 “다시 쓰지 않을 URI를 지정하려면”
- 191 페이지 “URI를 규칙 집합에 매핑하려면”
- 192 페이지 “MIME 매핑을 지정하려면”
- 193 페이지 “기본 도메인을 지정하려면”

▼ 게이트웨이가 모든 URL을 다시 쓰도록 하려면

게이트웨이 서비스에서 [모든 URL 다시 쓰기 사용] 옵션을 설정하면 [도메인 및 부속 도메인의 프록시] 목록에 있는 항목을 확인하지 않고 Rewriter가 모든 URL을 다시 씁니다. [도메인 및 하위 도메인의 프록시] 목록에 있는 항목은 무시됩니다.

1. Portal Server 관리 콘솔에 관리자로 로그인합니다.
2. [Secure Remote Access] 탭을 선택하고 속성을 수정할 게이트웨이 프로필을 선택합니다.
3. [Rewriter] 탭을 선택합니다.
4. [기본 옵션]에서 [모든 URI 다시 쓰기 사용] 확인란을 선택하여 게이트웨이가 모든 URL을 다시 쓸 수 있게 합니다.
5. [저장]을 눌러 완료합니다.
6. 터미널 창에서 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ 다시 쓰지 않을 URI를 지정하려면

1. Portal Server 관리 콘솔에 관리자로 로그인합니다.
2. [Secure Remote Access] 탭을 선택하고 속성을 설정할 게이트웨이 프로필을 선택합니다.
3. [Rewriter] 탭을 선택합니다.
4. [기본 옵션]에서 [추가] 텍스트 필드에 URI를 입력한 다음 [추가]를 누릅니다.
   URI 값이 [다시 쓰지 않을 URI] 상자에 표시됩니다.

   주 – #*를 이 목록에 추가하면 규칙 집합에 href 규칙이 포함되어 있더라도 URI 다시 쓰기를 허용합니다.

5. [저장]을 눌러 완료합니다.
6. 터미널 창에서 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ URI를 규칙 집합에 매핑하려면

1. Portal Server 관리 콘솔에 관리자로 로그인합니다.
2. [Secure Remote Access] 탭을 선택하고 속성을 설정할 게이트웨이 프로필을 선택합니다.
3. [Rewriter] 탭을 선택합니다.

4. **[Rewriter 옵션]에서 [URI를 규칙 집합에 매핑]을 선택하고 [행 추가]를 누릅니다.**

5. **필수 도메인 또는 호스트 이름을 [URI] 필드에 입력하고 [규칙 집합] 필드에 해당 도메인에 대해 적절한 규칙 집합을 입력합니다.**

   항목이 [규칙 집합에 URI 매핑] 목록에 추가됩니다. 도메인이나 호스트 이름 그리고 규칙 집합을 지정하는 형식은 다음과 같습니다.

   ```
   domain name|ruleset name
   ```

   예:

   ```
   eng.sesta.com|default
   ```

6. **[저장]을 눌러 완료합니다.**

7. **터미널 창에서 게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ MIME 매핑을 지정하려면

Rewriter에는HTML, JAVASCRIPT, CSS 및 XML 컨텐트 유형에 따라 웹 페이지를 구문 분석할 수 있는 네 개의 구문 분석기가 있습니다. 기본적으로 공통 MIME 유형이 이러한 구문 분석기와 연결되어 있습니다. 게이트웨이 서비스의 [구문 분석기를 MIME 유형에 매핑] 필드에서 새로운 MIME 유형을 이러한 구문 분석기와 연관시킬 수 있습니다. 그러면 Rewriter의 기능이 다른 MIME 유형까지 확장됩니다.

여러 항목인 경우는 세미콜론(";")이나 쉼표(",")를 사용하여 구분합니다.예:

```
HTML=text/html;text/htm;text/x-component;text/wml; text/vnl/wap.wml
```

이것은 이러한 MIME을 가진 모든 컨텐트가 HTML Rewriter로 보내지고 URL을 다시 쓰도록 HTML 규칙이 적용된다는 의미입니다.

참고 – MIME 매핑에서 불필요한 구문 분석기를 제거하면 작동 속도를 높일 수 있습니다. 예를 들어, 특정 인트라넷의 컨텐트에 JavaScript가 없다는 것이 확실하면 MIME 매핑 목록에서 JAVASCRIPT 항목을 제거할 수 있습니다.

---

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**

2. **[Secure Remote Access] 탭을 선택하고 속성을 설정할 게이트웨이 프로필을 선택합니다.**

3. **[Rewriter] 탭을 선택합니다.**

4. **[Rewriter 옵션]에서 [구문 분석기를 MIME 유형에 매핑]을 선택합니다.**

   항목을 `HTML=text/html;text/htm` 형식으로 지정합니다.

5 **[행 추가]를 눌러 항목을 목록에 추가합니다. [MIME 유형] 필드에 매핑할 구문 분석기 값과 해당 MIME 값을 입력합니다.**

6 **[저장]을 눌러 완료합니다.**

7 **터미널 창에서 게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ 기본 도메인을 지정하려면

기본 도메인 및 부속 도메인은 URL에 도메인과 부속 도메인 없이 호스트 이름만 있을 때 유용합니다. 이 경우에 게이트웨이는 호스트 이름이 기본 도메인 및 부속 도메인에 있다고 가정하고 그에 따라 진행합니다.

예를 들어, URL의 호스트 이름이 `host1`이고 기본 도메인과 하위 도메인이 `red.sesta.com`으로 지정된 경우, 호스트 이름은 `host1.red.sesta.com`으로 확인됩니다.

1 **Portal Server 관리 콘솔에 관리자로 로그인합니다.**

2 **[Secure Remote Access] 탭을 선택하고 속성을 설정할 게이트웨이 프로필을 선택합니다.**

3 **[배포] 탭을 선택합니다.**

4 **[도메인 및 하위 도메인의 프록시] 필드에 프록시 없이 필수 도메인 이름을 입력합니다.**

5 **[저장]을 눌러 완료합니다.**

6 **터미널 창에서 게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

◆ ◆ ◆ 10장

10 인증서 작업

이 장에서는 인증서 관리를 설명하고 직접 서명한 인증서와 인증 기관에서 받은 인증서를 설치하는 방법을 알아봅니다.

이번 장은 다음으로 구성되어 있습니다.

SSL 인증서 소개

Sun Java System Portal Server Secure Remote Access 소프트웨어는 원격 사용자를 위해 인증서 기반 인증을 제공합니다. SRA는 SSL(Secure Socket Layer)을 사용하여 보안 통신을 가능하게 합니다. SSL 프로토콜은 두 컴퓨터 간 보안 통신을 가능하게 해줍니다.

SSL 인증서에서는 공개 키와 개인 키 쌍을 사용하여 암호화 및 비밀번호 해독 기능을 제공합니다.

인증서 유형은 2가지입니다.

- 직접 서명한 인증서(또는 루트 CA 인증서라고도 함)

- 인증 기관(CA)에서 발급한 인증서

기본적으로 게이트웨이를 설치할 때에는 직접 서명한 인증서가 생성 및 설치됩니다.

설치후 언제라도 인증서를 설치, 습득 또는 교체할 수 있습니다.

또한 개인 디지털 인증서(PDC)를 통해 클라이언트 인증을 지원합니다. PDC는 SSL 클라이언트 인증을 통해 사용자를 인증하는 메커니즘입니다. SSL 클라이언트 인증을 사용하면 SSL 핸드셰이크가 게이트웨이에서 종료됩니다. 게이트웨이는 사용자의 PDC를 추출하여 인증된 서버로 전달합니다. 그러면 이 서버는 PDC를 사용하여 사용자를 인증합니다. 인증 체이닝을 사용하여 PDC를 구성하려면 56 페이지 "인증 체이닝 사용"을 참조하십시오.

SRA에는 SSL 인증서를 관리하는 데 사용할 수 있는 `certadmin`이라는 도구가 있습니다. 자세한 내용은 201 페이지 "certadmin 스크립트"를 참조하십시오.

주 – 인증서 팝업 창은 SSL 응용 프로그램에서 공통적으로 나타납니다. 사용자에게 경고를 승인하고 계속 진행하도록 알려주십시오.

인증서 파일

인증서 관련 파일은 `/etc/opt/SUNWportal/cert/` *gateway-profile-name*에 있습니다. 이 디렉토리에는 기본적으로 파일이 5개 들어 있습니다.

196 페이지 "인증서 파일"에는 이러한 파일과 해당 설명이 나와 있습니다.

표 10–1 인증서 파일

파일 이름	형식	설명
`cert8.db`, `key3.db`, `secmod.db`	이진	인증서, 키 및 암호화 모듈을 위한 데이터가 들어 있습니다. `certadmin` 스크립트로 조작할 수 있습니다. 필요에 따라 이 파일은 Portal Server 호스트와 게이트웨이 구성 요소 또는 게이트웨이 사이에서 공유될 수 있습니다.
`.jsspass`	숨은 텍스트 파일	SRA 키 데이터베이스를 위한 암호화된 비밀번호가 들어 있습니다.

표 10–1 인증서 파일 *(계속)*

파일 이름	형식	설명
`.nickname`	숨은 텍스트 파일	*token-name:certificate-name* 형식으로 게이트웨이에서 사용해야 하는 토큰과 인증서 이름을 저장합니다. 기본 토큰(기본 내부 소프트웨어 암호화 모듈에 있는 토큰)을 사용하는 경우 토큰 이름을 생략하십시오. 대부분의 경우 `.nickname` 파일은 인증서 이름만 저장합니다. 관리자로서 이 파일의 인증서 이름을 수정할 수 있습니다. 지정한 인증서를 이제 게이트웨이에서 사용합니다.

인증서 트러스트 속성

인증서의 트러스트 속성은 다음과 같은 정보를 나타냅니다.

- 인증서(클라이언트 또는 서버 인증서의 경우)를 인증된 기관에서 발행했는지 여부.
- 인증서(루트 인증서의 경우)가 서버 또는 클라이언트 인증서의 발급자로 인증될 수 있는지 여부.

각 인증서에 사용할 수 있는 트러스트 범주는 "SSL, 전자 메일, 개체 서명" 순서로 표시됩니다. 첫 번째 범주만 게이트웨이에 유용합니다. 각 범주 위치에서 트러스트 속성 코드가 사용되지 않을 수도 있고 많이 사용되기도 합니다.

범주에 대한 속성 코드는 쉼표로 분리되며 전체 속성 집합은 따옴표로 묶입니다. 예를 들어, 게이트웨이 설치 시 생성 및 설치된 직접 서명한 인증서는 'u,u,u'로 표시되는데 이는 루트 CA 인증서와는 반대로 서버 인증서(사용자 인증서)임을 의미합니다.

197 페이지 “인증서 트러스트 속성”에는 사용 가능한 속성 값과 각 값의 의미가 나와 있습니다.

표 10–2 인증서 트러스트 속성

속성	설명
p	유효한 피어
P	인증된 피어(p 내포)
c	유효한 CA
T	클라이언트 인증서를 발급할 수 있도록 인증된 CA(c 내포)
C	서버 인증서를 발급할 수 있도록 인증된 CA(SSL 전용)(c 내포)
u	인증서를 인증이나 서명에 사용할 수 있음

표 10–2 인증서 트러스트 속성 (계속)

속성	설명
w	경고 전송(해당 컨텍스트에서 인증서가 사용될 경우 다른 속성과 함께 사용하여 경고 포함)

CA 트러스트 속성

잘 알려진 공인 CA는 대부분 인증서 데이터베이스에 들어 있습니다. 공인 CA의 트러스트 속성을 수정하는 방법에 대한 자세한 내용은 207 페이지 “인증서의 트러스트 속성 수정”을 참조하십시오.

198 페이지 “CA 트러스트 속성”에는 가장 일반적인 인증 기관과 트러스트 속성이 나와 있습니다.

표 10–3 공인 인증 기관

인증 기관 이름	트러스트 속성
Verisign/RSA Secure Server CA	CPp,CPp,CPp
VeriSign Class 4 Primary CA	CPp,CPp,CPp
GTE CyberTrust Root CA	CPp,CPp,CPp
GTE CyberTrust Global Root	CPp,CPp,CPp
GTE CyberTrust Root 5	CPp,CPp,CPp
GTE CyberTrust Japan Root CA	CPp,CPp,CPp
GTE CyberTrust Japan Secure Server CA	CPp,CPp,CPp
Thawte Personal Basic CA	CPp,CPp,CPp
Thawte Personal Premium CA	CPp,CPp,CPp
Thawte Personal Freemail CA	CPp,CPp,CPp
Thawte Server CA	CPp,CPp,CPp
Thawte Premium Server CA	CPp,CPp,CPp
American Express CA	CPp,CPp,CPp
American Express Global CA	CPp,CPp,CPp
Equifax Premium CA	CPp,CPp,CPp
Equifax Secure CA	CPp,CPp,CPp
BelSign Object Publishing CA	CPp,CPp,CPp

표 10–3 공인 인증 기관 *(계속)*

BelSign Secure Server CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 0 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 1 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 2 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 3 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 4 CA	CPp,CPp,CPp
ABAecom (sub., Am. Bankers Assn.) Root CA	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 1	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 3	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 2	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 4	CPp,CPp,CPp
Deutsche Telekom AG Root CA	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority	CPp,CPp,CPp
Verisign Class 2 Public Primary Certification Authority	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 2 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 4 Public Primary Certification Authority - G2	CPp,CPp,CPp
GlobalSign Root CA	CPp,CPp,CPp
GlobalSign Partners CA	CPp,CPp,CPp
GlobalSign Primary Class 1 CA	CPp,CPp,CPp
GlobalSign Primary Class 2 CA	CPp,CPp,CPp
GlobalSign Primary Class 3 CA	CPp,CPp,CPp
ValiCert Class 1 VA	CPp,CPp,CPp
ValiCert Class 2 VA	CPp,CPp,CPp
ValiCert Class 3 VA	CPp,CPp,CPp
Thawte Universal CA Root	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority - G3	CPp,CPp,CPp

표 10–3 공인 인증 기관 (계속)

Verisign Class 2 Public Primary Certification Authority - G3	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority - G3	CPp,CPp,CPp
Verisign Class 4 Public Primary Certification Authority - G3	CPp,CPp,CPp
Entrust.net Secure Server CA	CPp,CPp,CPp
Entrust.net Secure Personal CA	CPp,CPp,CPp
Entrust.net Premium 2048 Secure Server CA	CPp,CPp,CPp
ValiCert OCSP Responder	CPp,CPp,CPp
Baltimore CyberTrust Code Signing Root	CPp,CPp,CPp
Baltimore CyberTrust Root	CPp,CPp,CPp
Baltimore CyberTrust Mobile Commerce Root	CPp,CPp,CPp
Equifax Secure Global eBusiness CA	CPp,CPp,CPp
Equifax Secure eBusiness CA 1	CPp,CPp,CPp
Equifax Secure eBusiness CA 2	CPp,CPp,CPp
Visa International Global Root 1	CPp,CPp,CPp
Visa International Global Root 2	CPp,CPp,CPp
Visa International Global Root 3	CPp,CPp,CPp
Visa International Global Root 4	CPp,CPp,CPp
Visa International Global Root 5	CPp,CPp,CPp
beTRUSTed Root CA	CPp,CPp,CPp
Xcert Root CA	CPp,CPp,CPp
Xcert Root CA 1024	CPp,CPp,CPp
Xcert Root CA v1	CPp,CPp,CPp
Xcert Root CA v1 1024	CPp,CPp,CPp
Xcert EZ	CPp,CPp,CPp
CertEngine CA	CPp,CPp,CPp
BankEngine CA	CPp,CPp,CPp
FortEngine CA	CPp,CPp,CPp
MailEngine CA	CPp,CPp,CPp
TraderEngine CA	CPp,CPp,CPp

표 10–3 공인 인증 기관 (계속)

USPS Root	CPp,CPp,CPp
USPS Production 1	CPp,CPp,CPp
AddTrust Non-Validated Services Root	CPp,CPp,CPp
AddTrust External Root	CPp,CPp,CPp
AddTrust Public Services Root	CPp,CPp,CPp
AddTrust Qualified Certificates Root	CPp,CPp,CPp
Verisign Class 1 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Class 2 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Class 3 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Secure Server OCSP Responder	CPp,CPp,CPp
Verisign Time Stamping Authority CA	CPp,CPp,CPp
Thawte Time Stamping CA	CPp,CPp,CPp
E-Certify CA	CPp,CPp,CPp
E-Certify RA	CPp,CPp,CPp
Entrust.net Global Secure Server CA	CPp,CPp,CPp
Entrust.net Global Secure Personal CA	CPp,CPp,CPp

certadmin 스크립트

다음과 같은 인증서 관리 작업에 `certadmin` 스크립트를 사용할 수 있습니다.

직접 서명한 인증서 생성

각 서버와 게이트웨이 사이의 SSL 통신을 위해서는 인증서를 생성해야 합니다.

▼ 설치 후 직접 서명한 인증서를 생성하려면

1 인증서를 생성하려는 게이트웨이 컴퓨터에서 루트로 certadmin 스크립트를 실행합니다.

portal-server-install-root/SUNWportal/bin/certadmin -n *gateway-profile-name*

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
10) 종료
선택: [10]
1
```

2 인증서 관리 메뉴의 옵션 1을 선택합니다.

인증서 관리 스크립트에서 기존 데이터베이스 파일을 유지할 것인지 묻습니다.

3 조직별 정보, 토큰 이름 및 인증서 이름을 입력합니다.

주 – 와일드카드 인증서에는 호스트의 정규 DNS 이름에 *를 지정합니다. 예를 들어, 호스트의 정규 DNS 이름이 abc.sesta.com이면 *.sesta.com으로 지정합니다. 이제 생성된 인증서는 sesta.com 도메인에 있는 모든 호스트 이름에 유효합니다.

```
이 호스트의 정규화된 DNS 이름은 무엇입니까? [host_name.domain_name]
조직의 종류는 무엇입니까(예: 회사)? []
조직 구성 단위는 무엇입니까(예: 부서)? []
구/군/시의 이름은 무엇입니까? []
시/도의 이름은 무엇입니까(약어는 사용 못함)? []
이 조직 구성 단위에 대한 2자로 된 국가 번호는 무엇입니까? []
토큰 이름은 기본 내부 (소프트웨어) 암호화 모듈을 사용하지 않는 경우,
예를 들어 암호화 카드를 사용하려는 경우 등에만 필요합니다(토큰 이름은
다음 명령을 사용하여 나열할 수 있음:
modutil -dbdir /etc/opt/SUNWportal/cert/gateway-profile-name -list);
그렇지 않으면 아래의 Return을 누르십시오.
토큰 이름을 입력하십시오. []
이 인증서에 대해 원하는 이름을 입력하십시오?
이 인증서의 유효 기간(개월)을 입력하십시오. [6]
직접 서명한 인증서가 생성되고 프롬프트로 돌아갑니다.
```

토큰 이름(기본적으로 비어 있음)과 인증서 이름은 /etc/opt/SUNWportal/cert/*gateway-profile-name*의 nickname 파일에 저장됩니다.

4 인증서가 적용되도록 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

CSR(인증서 서명 요청) 생성

CA가 발급하는 인증서를 주문하기 전에 CA에서 요구하는 정보가 들어 있는 인증서 서명 요청을 만들어야 합니다.

▼ CSR을 생성하려면

1 루트로서 certadmin 스크립트를 실행합니다.

portal-server-install-root/SUNWportal/bin/certadmin -n *gateway-profile-name*

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
10) 종료
선택: [10]
2
```

2 인증서 관리 메뉴의 옵션 2를 선택합니다.

스크립트에서 조직별 정보, 토큰 이름 및 웹 마스터의 전자 메일과 전화 번호를 입력하라는 메시지를 표시합니다.

호스트의 정규 DNS 이름을 반드시 지정해야 합니다.

```
이 호스트의 정규화된 DNS 이름은 무엇입니까? [snape.sesta.com]
조직의 종류는 무엇입니까(예: 회사)? []
조직 구성 단위는 무엇입니까(예: 부서)? []
구/군/시의 이름은 무엇입니까? []
시/도의 이름은 무엇입니까(약어는 사용 못함)? []
이 조직 구성 단위에 대한 2자로 된 국가 번호는 무엇입니까? []
토큰 이름은 기본 내부 (소프트웨어) 암호화 모듈을 사용하지 않는 경우,
예를 들어 암호화 카드를 사용하려는 경우 등에만 필요합니다(토큰 이름은
다음 명령을 사용하여 나열할 수 있음:
```

```
modutil -dbdir /etc/opt/SUNWportal/cert -list);
그렇지 않으면 아래의 Return을 누르십시오.
토큰 이름을 입력하십시오. []
이제 인증서를 생성할 컴퓨터의
웹 마스터 연락 정보를
입력하십시오.
이 서버의 관리자 또는 웹 마스터의 전자 메일 주소는 무엇입니까 [] ?
이 서버의 관리자 또는 웹 마스터의 전화 번호는 무엇입니까 [] ?
```

3 필요한 정보를 모두 입력하십시오.

주 – 웹 마스터의 전자 메일과 전화 번호를 공백으로 남겨두지 마십시오. 이 정보는 유효한 CSR을 받는 데 필요합니다.

CSR이 생성되어 *portal-server-install-root*/SUNWportal/bin/csr.hostname.datetimestamp 파일에 저장됩니다. CSR은 화면에도 인쇄됩니다. CA가 발급하는 인증서를 주문할 때 CSR을 직접 복사한 후 붙여넣을 수 있습니다.

루트 CA 인증서 추가

클라이언트 사이트에서 게이트웨이 인증서 데이터베이스에 알려지지 않은 CA에서 서명한 인증서를 제시하면 SSL 핸드셰이크가 실패합니다.

이를 방지하려면 루트 CA 인증서를 인증서 데이터베이스에 추가해야 합니다. 그러면 게이트웨이에서 CA를 인식할 수 있게 됩니다.

CA의 웹 사이트를 찾아서 해당 CA의 루트 인증서를 얻으십시오. certadmin 스크립트를 사용할 때 파일 이름과 루트 CA 인증서의 경로를 지정합니다.

▼ 루트 CA 인증서를 추가하려면

1 루트로서 certadmin 스크립트를 실행합니다.

portal-server-install-root/SUNWportal/bin/certadmin -n *gateway-profile-name*

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
```

```
10) 종료
선택: [10]
3
```

2 **인증서 관리 메뉴의 옵션 3을 선택합니다.**

3 **루트 인증서가 들어 있는 파일 이름을 입력한 다음 인증서 이름을 입력합니다.**
그러면 루트 CA 인증서가 인증서 데이터베이스에 추가됩니다.

인증 기관에서 발급한 SSL 인증서 설치

게이트웨이를 설치하는 동안 기본적으로 직접 서명한 인증서가 만들어져 설치됩니다. 설치 후 언제라도 공식 인증 기관(CA) 서비스를 제공하는 공급업체나 기업 CA에 의해 서명된 SSL 인증서를 설치할 수 있습니다.

이 작업은 다음과 같은 3단계로 이루어집니다.

- 203 페이지 “CSR(인증서 서명 요청) 생성”
- 205 페이지 “CA에서 발급하는 인증서 주문”
- 205 페이지 “CA에서 받은 인증서 설치”

CA에서 발급하는 인증서 주문

인증서 서명 요청(CSR)을 만들었으면 CSR을 사용하여 CA가 발급하는 인증서를 주문해야 합니다.

▼ CA가 발급하는 인증서를 주문하려면

1 **인증 기관의 웹 사이트로 가서 인증서를 주문합니다.**

2 **CA의 요청에 따라 CSR을 제공합니다. CA의 요청에 따라 기타 정보도 제공합니다.**
그러면 CA가 발급하는 인증서를 받게 됩니다. 인증서를 파일에 저장합니다. 파일에 인증서와 함께 "BEGIN CERTIFICATE" 및 "END CERTIFICATE" 라인을 포함시킵니다.

다음 예제에서는 실제 인증서 데이터를 생략하였습니다.

```
-----BEGIN CERTIFICATE-----
The certificate contents...
----END CERTIFICATE-----
```

CA에서 받은 인증서 설치

`certadmin` 스크립트를 사용하여 CA에서 받은 인증서를 `/etc/opt/SUNWportal/cert/`*gateway-profile-name*의 로컬 데이터베이스 파일에 설치합니다.

▼ CA에서 받은 인증서를 설치하려면

1 루트로서 certadmin 스크립트를 실행합니다.

portal-server-install-root/SUNWportal/bin/certadmin -n *gateway-profile-name*

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
10) 종료
선택: [10]
4
```

2 인증서 관리 메뉴의 옵션 4를 선택합니다.

스크립트에서 인증서 파일 이름, 인증서 이름 및 토큰 이름을 입력하라고 요청합니다.

```
인증서를 포함하는 파일의 이름(경로 포함)은 무엇입니까?
이 인증서의 CSR을 만들 때 사용한 토큰 이름을 입력하십시오. []
```

3 필요한 정보를 모두 입력하십시오.

인증서가 /etc/opt/SUNWportal/cert/*gateway-profile-name*에 설치되고 화면 메시지가 나타납니다.

4 인증서가 적용되도록 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

인증서 삭제

인증서 관리 스크립트를 사용하면 인증서를 삭제할 수 있습니다.

▼ 인증서를 삭제하려면

1 루트로서 certadmin 스크립트를 실행합니다.

portal-server-install-root/SUNWportal/bin/certadmin -n

여기서 *gateway-profile-name*은 게이트웨이 인스턴스의 이름입니다.

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
10) 종료
선택: [10]
5
```

2 인증서 관리 메뉴의 옵션 5를 선택합니다.

3 삭제할 인증서의 이름을 입력하십시오.

인증서의 트러스트 속성 수정

인증서의 트러스트 속성을 수정해야 하는 한 경우는 게이트웨이에서 클라이언트 인증이 사용될 때입니다. 클라이언트 인증의 한 예는 PDC(Personal Digital Certificate)입니다. PDC를 발급하는 CA는 게이트웨이에 의해 인증되어야 하며 CA 인증서에는 SSL용으로 "T"라고 표시되어 있어야 합니다.

게이트웨이 구성 요소가 HTTPS 사이트와 통신하도록 설정된 경우 HTTPS 사이트 서버 인증서의 CA는 게이트웨이에서 인증되어야 하며 CA 인증서에는 SSL용의 "C" 표시가 있어야 합니다.

▼ 인증서의 트러스트 속성을 수정하려면

1 루트로서 certadmin 스크립트를 실행합니다.

gateway-install-root/SUNWportal/bin/certadmin -n *gateway-profile-name*

여기서 *gateway-profile-name*은 게이트웨이 인스턴스의 이름입니다.

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
```

```
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
10) 종료
선택: [10]
6
```

2 **인증서 관리 메뉴의 옵션 6을 선택합니다.**

3 **인증서의 이름을 입력합니다. 예를 들어, Thawte Personal Freemail CA와 같이 입력하면 됩니다.**

```
이 인증서의 이름을 입력하시겠습니까?
Thawte Personal Freemail CA
```

4 **인증서의 트러스트 속성을 입력합니다.**

```
인증서에 포함할 트러스트 속성을 입력하십시오. [CT,CT,CT]
```

인증서 트러스트 속성이 변경됩니다.

루트 CA 인증서 나열

인증서 관리 스크립트를 사용하면 모든 루트 CA 인증서를 볼 수 있습니다.

▼ 루트 CA 목록을 보려면

1 **루트로서 `certadmin` 스크립트를 실행합니다.**

portal-server-install-root`/SUNWportal/bin/certadmin -n` *gateway-profile-name*

여기서 *gateway-profile-name*은 게이트웨이 인스턴스의 이름입니다.

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
10) 종료
선택: [10]
7
```

2 인증서 관리 메뉴의 옵션 7을 선택합니다.
모든 루트 CA 인증서가 표시됩니다.

모든 인증서 나열

인증서 관리 스크립트를 사용하면 모든 인증서에 해당하는 트러스트 속성을 볼 수 있습니다.

▼ 모든 인증서를 나열하려면

1 루트로서 certadmin 스크립트를 실행합니다.

portal-server-install-root
/SUNWportal/bin/certadmin -n
gateway-profile-name

여기서 *gateway-profile-name*은 게이트웨이 인스턴스의 이름입니다.

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
10) 종료
선택: [10]
8
```

2 인증서 관리 메뉴의 옵션 8을 선택합니다.
모든 CA 인증서가 표시됩니다.

인증서 인쇄

인증서 관리 스크립트를 사용하면 인증서를 인쇄할 수 있습니다.

▼ 인증서를 인쇄하려면

1 루트로서 certadmin 스크립트를 실행합니다.

portal-server-install-root/SUNWportal/bin/certadmin -n
gateway-profile-name

여기서 *gateway-profile-name*은 게이트웨이 인스턴스의 이름입니다.

인증서 관리 메뉴가 표시됩니다.

```
1) 직접 서명한 인증서 생성
2) CSR(인증서 서명 요청) 생성
3) 루트 CA 인증서 추가
4) CA(인증 기관)에서 인증서 설치
5) 인증서 삭제
6) 인증서의 트러스트 속성 수정(예: PDC)
7) 루트 CA 인증서 표시
8) 모든 인증서 표시
9) 인증서 내용 인쇄
10) 종료
선택: [10]
9
```

2 인증서 관리 메뉴의 옵션 9를 선택합니다.

3 인증서의 이름을 입력합니다.

◆ ◆ ◆ 11장

11 Netlet 구성

이 장에서는 Sun Java System Portal Server 관리 콘솔에서 Netlet 속성을 구성하는 방법을 설명합니다. 조직 수준에서 구성할 수 있는 모든 속성은 사용자 수준에서도 구성할 수 있습니다. 조직, 역할 및 사용자 수준 속성에 대한 자세한 내용은 **Access Manager 관리 설명서**를 참조하십시오.

이번 장은 다음 절로 구성됩니다.

Netlet 속성 구성

다음 작업을 수행하여 Netlet을 구성할 수 있습니다.

▼ 기본 속성을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭과 [Netlet] 탭을 차례로 선택합니다.**
3. **[DN 선택] 목록에서 사용자 또는 조직의 DN을 선택하거나 DN을 추가합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
COS 우선 순위	속성 값의 상속 여부를 결정하는 데 사용하는 값을 지정합니다. 이 속성에 대한 자세한 내용은 **Sun Java System Directory Server 관리 설명서**를 참조하십시오.
다음을 사용하여 Netlet 시작	Java Webstart 또는 Applet 옵션 중에서 모드를 선택하여 Netlet 서비스를 시작합니다.
기본 루프백 포트	Netlet을 통해 애플릿을 다운로드할 때 로컬 시스템에서 사용할 포트를 지정합니다. Netlet 규칙에서 값이 대체되지 않는다면 기본값 58000이 사용됩니다. 필수 포트 번호를 입력합니다.
연결 유지 간격(초)	클라이언트에서 웹 프록시를 통해 게이트웨이에 연결하는 경우 유휴 Netlet 연결은 프록시 시간 초과로 인해 해제됩니다. 이를 방지하기 위해 프록시 시간 초과 값보다 작은 값을 입력합니다.

5 [저장]을 눌러 완료합니다.

▼ 고급 속성 구성

1 Portal Server 관리 콘솔에 관리자로 로그인합니다.

2 [Secure Remote Access] 탭과 [Netlet] 탭을 차례로 선택합니다.

3 [DN 선택] 목록에서 사용자 또는 조직의 DN을 선택하거나 DN을 추가합니다.

4 다음 속성을 수정합니다.

속성 이름	설명
포털 로그아웃할 때 Netlet 종료	사용자가 포털 서버에서 로그아웃할 때 모든 연결이 종료되도록 하려는 경우 [예]를 선택합니다. 이 옵션을 설정하면 보안이 강화됩니다. 기본적으로 이 옵션이 선택됩니다. 사용자가 포털 서버 데스크탑을 로그아웃한 후에도 Netlet 연결이 계속 유지되도록 하려면 [아니요]를 선택합니다. **주 –** [아니요] 옵션을 선택하면 사용자가 Portal Server에서 로그아웃 후 새 Netlet 연결을 만들 수 없습니다. 기존 연결만 보존됩니다.
연결 재인증	Netlet을 통해 애플릿을 다운로드할 때 로컬 시스템에서 사용할 포트를 지정하려면 [예]를 선택합니다. Netlet 규칙에서 값이 대체되지 않는다면 기본값 58000이 사용됩니다. 기본적으로 [아니요] 옵션이 선택됩니다.

속성 이름	설명
연결에 대해 경고 팝업 표시	사용자가 Netlet을 사용하여 응용 프로그램을 실행하고 있을 때 다른 사용자가 수신 포트를 통해 Netlet에 연결하려고 하는 경우 사용자의 데스크탑에 경고 팝업 대화 상자를 표시하려면 [예]를 선택합니다. 기본적으로 [예] 옵션이 선택됩니다.
포트 경고 대화 상자에 확인란 표시	관리 콘솔에서 활성화되어 있는 경우 Netlet에서 로컬 시스템에서 사용할 수 있는 포트를 통해 대상 호스트에 연결할 때 사용자 데스크탑에 경고 대화 상자를 표시하려면 [예]를 선택합니다. 기본적으로 [예] 옵션이 선택됩니다.
Netlet 규칙	전역 수준에서 Netlet 규칙을 만듭니다. 이러한 규칙은 생성되는 새로운 조직에 상속됩니다. Netlet 규칙 만들기, 수정 및 삭제에 대한 자세한 내용은 214 페이지 “Netlet 규칙을 만들거나 수정하거나 삭제하려면”을 참조하십시오.
기본 원시 VM 암호	드롭다운 상자에서 Netlet 규칙의 기본 암호를 선택합니다. 규칙의 일부로 암호를 포함하지 않은 기존 규칙을 사용할 때 유용합니다. 자세한 내용은 143 페이지 “이전 버전과의 호환성” 절을 참조하십시오.
기본 Java 플러그 인 암호화	드롭다운 상자에서 기본 Java 플러그 인 암호를 선택합니다. 지원되는 암호 목록은 142 페이지 “지원되는 암호”를 참조하십시오.
허용/거부된 호스트	호스트 주소 확인란을 선택하고 사용자 또는 조직 유형에 따라 액세스를 허용할 호스트를 선택한 다음 드롭다운 상자에서 [허용] 또는 [거부] 옵션을 선택합니다. 새 호스트를 추가하려면 a. [행 추가]를 누릅니다. b. 정규화된 호스트 주소를 입력합니다. 예: `abc.abc.sesta.com`과 같이 입력합니다. **주 –** 기존 호스트를 삭제하려면 [호스트] 목록에서 호스트를 선택하고 [삭제]를 누릅니다. 특정 조직, 역할 또는 사용자에 대해 특정 호스트에 대한 액세스를 정의하거나 거부할 수 있습니다. 예를 들어, 사용자가 Telnet으로 연결할 수 있는 5개의 호스트로 허용 목록을 설정할 수 있습니다. 조직 내에서 특정 호스트에 대한 액세스를 거부할 수 있습니다. 각 규칙에 고유한 **로컬 포트**를 지정합니다. **주 –** 이 필드에서 별표(*)는 지정 도메인의 모든 호스트에 액세스할 수 있다는 것을 나타냅니다. 예를 들어, `*.sesta.com`을 지정하면 사용자가 `sesta.com` 도메인 내의 모든 Netlet 대상을 실행시킬 수 있습니다. `xxx.xxx.xxx.*`와 같이 IP 주소를 와일드 카드로 지정할 수도 있습니다.
Netlet 규칙 액세스/거부	Nelet 규칙을 선택하고 드롭다운 상자에서 [허용] 또는 [거부] 옵션을 선택합니다. 특정 조직, 역할 또는 사용자의 특정 Netlet 규칙에 대한 액세스를 정의할 수 있습니다. 특정 조직, 역할 또는 사용자의 특정 Netlet 규칙에 대한 액세스를 거부할 수 있습니다. **주 –** 이 필드에서 별표(*)는 선택된 조직에 정의된 모든 Netlet 규칙을 사용할 수 있다는 것을 나타냅니다.

5 **[저장]을 눌러 완료합니다.**

▼ Netlet 규칙을 만들거나 수정하거나 삭제하려면

조직, 역할 또는 사용자 수준에서 새 규칙을 만들거나 기존 규칙을 수정할 수도 있습니다. 이러한 규칙은 생성되는 새로운 조직에 상속됩니다.

1 **Portal Server 관리 콘솔에 관리자로 로그인합니다.**

2 **[Secure Remote Access] 탭과 [Netlet] 탭을 차례로 선택합니다.**

3 **[DN 선택] 목록에서 사용자 또는 조직의 DN을 선택하거나 DN을 추가합니다.**

4 **[고급] > [Netlet 규칙]에서 [새 규칙]을 누릅니다.**

- **규칙을 삭제하려면 원하는 규칙을 선택하고 [삭제]를 누릅니다.**
- **규칙을 수정하려면 해당 규칙 이름을 누릅니다.**
  다음 단계에서 설명하는 대로 Netlet 페이지에서 매개 변수를 수정합니다.

5 **[규칙 이름] 필드에 규칙 이름을 입력합니다.**

6 **사용 가능한 암호 목록에서 [기타]를 선택하고 [암호화 암호] 목록에서 하나 이상의 암호화 암호를 선택하거나 [기본값]을 선택하여 기본 암호화 암호를 그대로 사용합니다.**
규칙의 일부로 암호를 포함하지 않은 기존 규칙을 사용할 때 유용합니다. 자세한 내용은 "역방향 호환성" 절을 참조하십시오. 암호에 대한 자세한 내용은 "기본 암호화 암호 지정"을 참조하십시오.

7 **[원격 응용 프로그램 URL] 필드에 호출할 응용 프로그램의 URL을 입력합니다.**

8 **애플릿을 다운로드해야 하는 경우에는 [클라이언트 포트] 확인란을 선택합니다. 클라이언트 포트 번호, 서버 호스트 주소 및 서버 포트 번호를 [클라이언트 포트], [서버 호스트] 및 [서버 포트] 필드에 입력합니다. 각 규칙에 고유한 로컬 포트를 지정합니다.**
기본적으로 [애플릿 다운로드 사용] 확인란은 비활성화되어 있습니다. Portal Server 호스트가 아닌 호스트에서 애플릿을 다운로드해야 하는 경우에만 애플릿 세부 사항을 지정합니다. 자세한 내용은 136 페이지 "원격 호스트에서 애플릿 다운로드"를 참조하십시오.

9 **[세션 확장 사용] 확인란을 선택하여 이 규칙에 해당하는 Netlet 세션이 실행되는 동안에 Portal Server 세션 시간이 연장되도록 합니다.**

10 **[로컬 포트를 대상 서버 포트에 매핑]에서 다음을 수행합니다.**

a. **[로컬 포트] 필드에 Netlet이 수신할 로컬 포트를 입력합니다.**
FTP 규칙의 경우 로컬 포트 값이 30021이어야 합니다.

b. **[대상 호스트] 필드에 항목을 입력합니다.**
정적 규칙의 경우, Netlet 연결에 대한 대상 컴퓨터의 호스트 이름을 입력합니다. 동적 규칙의 경우, "TARGET"을 입력합니다.

c. **[대상 포트] 필드에 대상 호스트의 포트를 입력합니다.**

11 **[저장]을 눌러 완료합니다.**
규칙 이름이 Netlet 홈 페이지에 표시됩니다.

Netlet용 프록시 구성

사용자 수준에서 다음과 같은 속성을 구성할 수 있습니다.

- 브라우저 프록시 유형
- 브라우저 프록시 호스트
- 브라우저 프록시 포트
- 브라우저 프록시 대체 목록

관리 콘솔에서 이 값을 지정하지 않은 경우 Netlet에서 브라우저 프록시 설정을 확인할 수 없으면 사용자가 처음으로 Netlet을 통해 연결할 때 이 정보를 묻습니다. 이 정보는 저장되었다가 나중에 사용자 연결에 사용됩니다.

다음 시나리오에서는 Netlet이 브라우저 프록시 설정을 확인하지 못합니다.

- 사용자가 Java 플러그인(버전 1.4.0보다 낮음)이 설치된 Internet Explorer 4.x, 5.x 또는 6.x를 사용하고, Java 플러그인 제어 패널의 [프록시] 탭에서 “브라우저 설정 사용” 옵션을 선택하고, Internet Explorer의 LAN 설정 대화 상자에 있는 “자동 구성 스크립트 사용” 필드에서 애드온 제품이나 INS 파일을 지정한 경우.
- 사용자가 Java 플러그인(버전 1.3.1_01 이상)이 설치된 Netscape 6.2를 사용하고 Java 플러그인 제어 패널의 [프록시] 탭에서 “브라우저 설정 사용” 옵션을 선택한 경우.

두 경우 모두, Netlet이 브라우저 설정을 확인할 수 없기 때문에 사용자에게 다음 정보를 제공하도록 요청합니다.

- 브라우저 프록시 유형

  이 속성은 DIRECT 또는 MANUAL 값을 가질 수 있습니다. 사용자가 드롭다운 목록에서 DIRECT를 선택하면 Netlet이 게이트웨이 호스트에 직접 연결합니다.
- 브라우저 프록시 호스트

  Netlet의 연결을 위해 필요한 프록시 호스트를 지정합니다.

- 브라우저 프록시 포트

  Netlet의 연결을 위해 필요한 프록시 호스트의 포트를 지정합니다.
- 브라우저 프록시 대체 목록(쉼표로 구분)

  Netlet이 프록시를 통해 연결하지 않도록 할 호스트를 지정합니다. 이 목록에는 쉼표로 구분된 여러 호스트 이름이 있을 수 있습니다.

◆ ◆ ◆ 12장

12 개인 도메인 인증서로 Netlet 구성

이 장에서는 Netlet을 PDC와 함께 사용할 수 있도록 클라이언트 브라우저의 Java 플러그인을 구성하는 방법을 설명합니다.

주 – JSSE가 있는 Virtual Machine (VM)만 PDC가 있는 Netlet을 지원합니다.

PDC를 위한 Netlet 구성

소개 텍스트 위치

▼ PDC를 위한 Netlet을 구성하려면

1. **Portal Server 시스템의 `/ect/opt/SUNWam/config/AMConfig.properties` 파일에서 아무 위치에나 `com.iplanet.authentication.modules.cert.gwAuthEnable=yes`를 추가합니다.**
2. **PDC를 사용할 게이트웨이의 인증서 데이터베이스로 필요한 인증서를 가져옵니다.**
3. **게이트웨이 시스템에서 루트 CA 인증서를 가져옵니다.**
4. **CA 인증서를 게이트웨이 프로필에 추가합니다.**

참고 – 자체 게이트웨이 프로필을 만들어 PDC를 테스트합니다.

다음 단계에 따라 인증서를 게이트웨이 프로필에 추가합니다.

a. **게이트웨이 설치 디렉토리`/SUNWportal/bin/certadmin -n gateway profile name`**
   Certadmin 메뉴가 표시됩니다.

b. 옵션 3을 선택합니다.

c. 인증서 경로를 입력합니다.
인증서가 추가되었다는 메시지가 표시됩니다.

5 CA에 제출하기 위한 인증서 서명 요청을 생성합니다.
다음 단계를 수행하여 인증서 서명 요청을 제출합니다.

a. 게이트웨이 설치 디렉토리/SUNWportal/bin/certadmin -n gateway profile name
Certadmin 메뉴가 표시됩니다.

b. 옵션 2를 선택합니다.

c. 질문에 대한 적절한 답을 입력합니다.

d. 요청을 파일에 저장합니다.

6 인증서 서명 요청을 CA에 제출하고 승인을 받습니다.

참고 – CA 서명 후 인증서 서명 응답을 저장합니다.

---

7 CA가 승인한 서버 인증서를 가져옵니다.
다음 단계를 수행하여 서버 인증서를 가져옵니다.

a. 게이트웨이 설치 디렉토리/SUNWportal/bin/certadmin -n gateway profile name
Certadmin 메뉴가 표시됩니다.

b. 옵션 4를 선택합니다.

c. 서버 인증서가 포함된 파일의 위치를 입력합니다.

8 루트 CA 인증서를 Portal Server 시스템으로 가져옵니다.

- **Application Server의 경우 다음 명령을 사용하여** `root-ca`**를 추가합니다.**
  `./certutil -A -n rootca -t "TCu,TCu,TCuw" -d /var/opt/SUNWappserver/domains/domain1/config -a -i` *path to root-ca*

◆ ◆ ◆ 13장

13 Proxylet 구성

이 장에서는 Sun Java System Portal Server 관리 콘솔에서 Proxylet을 구성하는 방법에 대해 설명합니다.

이번 장은 다음 절로 구성됩니다.

Proxylet 속성 구성

배포 옵션에서 [Proxylet 애플릿 자동으로 다운로드]를 선택하면 사용자가 로그인할 때 자동으로 Proxylet이 시작되도록 구성할 수 있습니다. [Proxylet 자동으로 다운로드] 확인란을 선택하지 않은 경우, 사용자는 표준 포털 데스크탑의 Proxylet 채널에 있는 'Proxylet 시작' 링크를 눌러 필요할 때마다 Proxylet을 가져올 수 있습니다.

▼ Proxylet 속성을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭과 [Proxylet] 탭을 차례로 선택합니다.**
3. **[DN 선택] 목록 상자에서 적절한 DN을 선택하거나 특정 사용자 또는 조직의 기존 DN을 추가합니다.**
4. **Proxylet 페이지에서 다음을 수행합니다.**

속성 이름	설명
COS 우선 순위	옵션 목록에서 Proxylet 트래픽에 대한 서비스 클래스를 선택합니다.
Proxylet 애플릿을 자동으로 다운로드	Proxylet 애플릿을 클라이언트 시스템에 자동으로 다운로드하려면 [예]를 누릅니다. Proxylet 애플릿을 다운로드하기 위한 기본 요구 사항은 다음과 같습니다. 클라이언트 컴퓨터에서 서버 응용 프로그램 실행 가능 클라이언트 시스템의 Java 버전 1.4 이상 브라우저: IE 6.0 sp2 또는 Firefox 2.0 정확한 브라우저 권한
Proxylet을 통한 포털 새로 고침	Proxylet 실행 후 포털 데스크탑을 새로 고친 경우 트래픽이 Proxylet을 통해 전달되도록 하려면 [예]를 누릅니다. [Proxylet 시작 후 포털 새로 고침] 및 [Proxylet 애플릿 자동으로 다운로드]를 모두 사용할 경우 "App Urls"가 작동하지 않습니다.
시작 모드	Java Web Start 또는 애플릿을 선택합니다.
기본 Proxylet 애플릿 바인드 IP	Proxylet이 웹 브라우저에서 보낸 요청을 바인딩하고 수신할 IP 주소를 입력합니다.
기본 Proxylet 애플릿 포트	Proxylet이 브라우저의 요청을 수신할 포트 번호를 입력합니다.
자동 프록시 구성 파일 위치	PAC(프록시 자동 구성) 파일 또는 프록시 구성 목록에서 프록시 설정이 포함되어 있는 구성 파일의 위치를 입력합니다.

5 [Proxylet 규칙] 옵션에서 다음을 수행합니다.

a. Proxylet 서비스를 통해 실행할 응용 프로그램의 규칙을 지정합니다.

b. [추가]를 누릅니다.

c. [도메인] 필드에 `www.google.com`과 같은 도메인 이름을 입력합니다.

d. Proxylet에서 처리할 도메인의 호스트와 해당 포트 번호를 입력합니다. 이렇게 하면 Proxylet이 HTTP 요청을 처리하며 이 요청이 게이트웨이를 통해 전달되지 않습니다.

6 [저장]을 눌러 완료합니다.

응용 프로그램을 포털 데스크탑으로 구성

HTTP, FTP 등의 요청은 Proxylet 서비스를 통해 전달됩니다. 관리자는 Proxylet 규칙을 이용하여 프로토콜, 호스트 또는 포트를 기반으로 도메인에 대한 매핑을 지정할 수 있습니다. Proxylet 규칙을 사용하면 프록시 자동 구성(PAC) 파일의 도메인 및 프록시 설정을 지정할 수 있습니다. 예를 들어, 모든 FTP 트래픽이 Netlet을 통해 경로 지정되고 모든 HTTP 트래픽이 Proxylet을 통해 경로 지정되도록 규칙을 만들 수 있습니다. Proxylet 서비스를 통해 렌더링해야 하는 사전 정의된 응용 프로그램을 구성할 수 있습니다. 이 작업은 사용자 또는 조직 기본 설정에 따라 수행됩니다. Proxylet에서 처리할 응용 프로그램을 추가하면 사용자 데스크탑을 관리하기 쉬워지며 성능도 향상됩니다.

▼ 응용 프로그램을 포털 데스크탑으로 구성하려면

시작하기 전에

- [Proxylet] 옵션이 활성화되어 있는지 확인합니다. Proxylet 활성화에 대한 자세한 내용은 게이트웨이 프로필 장을 참조하십시오.

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[포털] 탭을 선택하고 수정할 포털 인스턴스를 선택합니다.**
   데스크탑 페이지가 표시됩니다.
3. **[DN 선택] 목록 상자에서 적절한 DN을 선택하거나 특정 사용자 또는 조직의 기존 DN을 추가합니다.**
4. **[컨테이너 및 채널 관리] 링크를 누릅니다.**
   컨테이너 및 채널 관리 페이지가 표시됩니다.
5. **왼쪽 창에서 Proxylet을 선택합니다.**
6. **오른쪽 창에서 `Appurls` 링크를 선택합니다.**
7. **등록 정보 마법사에서 응용 프로그램 이름과 값을 입력합니다. 필요에 따라 응용 프로그램 등록 정보를 수정합니다. 예를 들어, 적절한 응용 프로그램 이름과 http://www.example.com을 입력합니다.**
8. **[닫기]를 눌러 끝마칩니다.**
   이제 사용자 또는 조직 수준에서 포털 데스크탑의 응용 프로그램 링크를 볼 수 있습니다.

Java Web Start 또는 애플릿 모드에서 Proxylet 실행

포털 데스크탑에서 Java Web Start 또는 애플릿 모드를 통해 Proxylet을 시작할 수 있습니다.

▼ Java Web Start 또는 애플릿 모드에서 Proxylet을 실행하려면

1. **포털 데스크탑에 Proxylet 사용자로 로그온합니다.**
2. **맨 처음 페이지에서 Proxylet 채널로 이동한 후 [편집] 아이콘을 누릅니다.**
3. **[시작 모드] 목록 상자에서 Java Web Start 또는 애플릿 옵션을 선택합니다.**
4. **[완료]를 누릅니다.**

   Proxylet을 호출하려면 Proxylet 채널에서 응용 프로그램을 선택합니다. 이렇게 하면 Java Web Start 또는 애플릿 모드에서 응용 프로그램이 시작됩니다.

   - [자동으로 다운로드]를 선택한 경우 Proxylet 채널에서 응용 프로그램을 누릅니다.
   - 사용자 기본 설정에 따라 Java Web Start 또는 애플릿 모드의 선택을 바탕으로 Proxylet 콘솔이 표시됩니다. 모든 인증서를 수락하고 응용 프로그램 작업을 계속합니다.

◆ ◆ ◆ 14장

14 NetFile 구성

이 장에서는 Sun Java System Portal Server 관리 콘솔에서 NetFile을 구성하는 방법을 설명합니다.

이번 장은 다음 절로 구성됩니다.

NetFile 구성 작업

이 절에서는 다음 작업을 다룹니다.

▼ 기본 옵션을 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭과 [Netfile] 탭을 차례로 선택합니다.**
3. **[DN 선택] 목록에서 사용자 또는 조직의 DN을 선택하거나 DN을 추가합니다.**
4. **다음 속성을 수정합니다.**

속성 이름	설명
COS 우선 순위	속성 값의 상속 여부를 결정하는 데 사용하는 값을 지정합니다. 이 속성에 대한 자세한 내용은 **Sun Java System Directory Server 관리 설명서**를 참조하십시오.
도메인/호스트 기본 설정	허용된 호스트에 접속하기 위해 NetFile에 필요한 기본 도메인을 입력합니다. 이 기본 도메인 값은 사용자가 NetFile을 사용하여 호스트를 추가하면서 정규 호스트 이름을 지정하지 않은 경우에만 적용할 수 있습니다. **주** – [기본 도메인] 필드가 비어있지 않고 유효한 도메인 이름이 들어 있는지 확인합니다.
기본 WINS/DNS 서버	NetFile이 Microsoft Windows 호스트 액세스에 사용하는 WINS/DNS 서버 호스트 주소를 입력합니다. **주** – 사용자가 컴퓨터를 추가하면서 다른 값을 지정하여 이 값을 무시할 수 있습니다.
호스트 검색 순서	위쪽 및 아래쪽 버튼을 사용하여 호스트 검색 순서를 지정합니다.
공통 호스트	호스트 이름 또는 정규화된 이름을 입력하고 [추가]를 누릅니다. 제공한 호스트 이름이 사용자가 구성한 호스트 이름과 일치하면 두 정보 집합이 병합되고 사용자 지정 값이 우선적으로 적용됩니다. 모든 원격 NetFile 사용자가 NetFile을 통해 이용할 수 있는 호스트 목록을 구성합니다.

주 – 예를 들어, `sesta`, `siroe`, `florizon` 및 `abc`의 4가지 공통 호스트를 구성했다고 가정합니다. 사용자가 3개의 호스트를 구성했는데 이 중 2개가 `sesta`와 `siroe`입니다. 충돌 상황에서는 사용자가 지정한 값이 관리자가 지정한 값을 덮어씁니다. `florizon` 및 `abc`도 사용자의 NetFile에 나열되며 사용자가 해당 호스트에서 다양한 작업을 수행할 수 있습니다. `florizon`을 거부된 호스트 목록에 포함시킨 경우는 `florizon`이 사용자의 NetFile에 나열은 되지만 `florizon`에서 작업을 수행할 수 없습니다.

호스트 유형—공통 호스트 목록에 나열된 호스트를 이미 추가했다면 사용자 설정이 우선합니다. 유형이 충돌하면 해당 사용자에 대해 관리자가 추가한 공유가 추가되지 않습니다. 사용자와 관리자가 같은 공유를 추가하면 공유가 추가되지만 사용자가 설정한 비밀번호가 우선합니다.

5 [저장]을 눌러 완료합니다.

▼ 액세스 권한을 구성하려면

1 **Portal Server 관리 콘솔에 관리자로 로그인합니다.**

2 **[Secure Remote Access] 탭과 [Netfile] 탭을 차례로 선택합니다.**

3 **[DN 선택] 목록에서 사용자 또는 조직의 DN을 선택하거나 DN을 추가합니다.**

4 **[액세스 권한]을 누르고 다음 속성을 수정합니다.**

속성 이름	설명
Windows 호스트에 액세스	사용자가 Windows 호스트에 액세스할 수 있도록 하려면 [허용] 확인란을 선택합니다. 기본적으로 [허용] 확인란이 선택되어 있습니다.
FTP 호스트에 액세스	사용자가 FTP 호스트에 액세스할 수 있도록 하려면 [허용] 확인란을 선택합니다.
NFS 호스트에 액세스	사용자가 NFS 호스트에 액세스할 수 있도록 하려면 [허용] 확인란을 선택합니다.
Netware 호스트에 액세스	사용자가 Netware 호스트에 액세스할 수 있도록 하려면 [허용] 확인란을 선택합니다.

5 **[저장]을 눌러 완료합니다.**

▼ 호스트 기본 설정을 구성하려면

1 **Portal Server 관리 콘솔에 관리자로 로그인합니다.**

2 **[Secure Remote Access] 탭과 [Netfile] 탭을 차례로 선택합니다.**

3 **[DN 선택] 목록에서 사용자 또는 조직의 DN을 선택하거나 DN을 추가합니다.**

4 **[호스트 허용/거부] 목록에 * 항목이 있기 때문에 사용자는 기본적으로 NetFile을 통해 모든 호스트에 액세스할 수 있습니다. 이를 변경하려면 * 항목을 제거하고 사용자가 NetFile을 통해 액세스해야 하는 호스트만 이 목록에서 지정합니다. 또는 입력된 *를 그대로 두고 [거부된 호스트] 목록에서 액세스를 거부할 호스트를 지정할 수 있습니다. 이 경우 [거부된 호스트] 목록에 지정된 호스트를 제외한 모든 호스트에 액세스가 허용됩니다.**

주 – 사용자가 NetFile 창에서 이미 추가한 호스트의 액세스를 거부하는 경우에도 거부된 호스트는 사용자의 NetFile 창에 계속 표시됩니다. 그러나 사용자는 이 호스트에서 어떤 작업도 수행할 수 없습니다. NetFile Java2에서 응용 프로그램에 나타나는 경우 거부된 호스트에는 빨간색 십자 모양이 표시되어 액세스할 수 없음을 나타냅니다. [허용된 호스트] 및 [거부된 호스트] 목록이 모두 비어 있으면 어떤 호스트에도 액세스가 허용되지 않습니다.

5 [저장]을 눌러 완료합니다.

▼ 작업 기본 설정을 구성하려면

1 Portal Server 관리 콘솔에 관리자로 로그인합니다.

2 [Secure Remote Access] 탭과 [Netfile] 탭을 차례로 선택합니다.

3 [DN 선택] 목록에서 사용자 또는 조직의 DN을 선택하거나 DN을 추가합니다.

4 다음 속성을 수정합니다.

속성 이름	설명
기본 압축 유형	드롭다운 상자에서 기본 파일 압축 형식으로 ZIP 또는 GZ를 선택합니다.
기본 압축 수준	드롭다운 상자에서 기본 압축 수준을 선택합니다. 기본값은 6입니다.
임시 디렉토리 위치	임시 파일의 위치를 입력합니다. 지정된 임시 디렉토리는 서버에 없는 경우에 만들어집니다. 파일 메일링과 같은 일부 파일 작업에서는 임시 디렉토리가 반드시 필요합니다. 기본 임시 디렉토리는 `/tmp`입니다. 임시 파일은 필요한 작업이 수행된 다음에 삭제됩니다. **주 –** 웹 서버 실행에 사용하고 있는 아이디(`nobody` 또는 `noaccess` 등)에 지정 디렉토리에 대한 `rwx` 권한이 있는지 확인하십시오. 이 아이디에 필요한 임시 디렉토리의 전체 경로에 대한 `rx` 권한이 있는지도 확인하십시오. **참고 –** NetFile에 별도 임시 디렉토리를 만들어야 하는 경우가 있습니다. Portal Server의 모든 모듈에 공통된 임시 디렉토리를 지정하면 디스크 공간이 금방 부족해질 수 있습니다. 파일 메일링과 같은 NetFile의 일부 작업은 임시 디렉토리에 공간이 없으면 작동하지 않습니다.

속성 이름	설명
파일 업로드 한계(MB)	이 필드에 업로드할 수 있는 최대 파일 크기를 입력합니다. 기본값은 5MB입니다. 업로드하는 파일의 크기가 여기에 지정된 값을 초과하면 오류 메시지가 표시되고 파일이 업로드되지 않습니다. 잘못된 값을 입력하면 NetFile이 해당 값을 기본값으로 재설정합니다. 사용자마다 다른 파일 업로드 제한 크기를 지정할 수 있습니다.
디렉토리 검색 제한	한번의 검색으로 검색할 수 있는 최대 디렉토리 수를 입력합니다. 이 제한은 많은 사용자가 동시에 로그인하여 네트워크 체증을 유발하고 액세스 속도를 저하시키는 것을 방지합니다. 기본값은 100입니다. 사용자에게 *A*라는 디렉토리가 있다고 가정합니다. 이 *A* 디렉토리에는 100개의 하위 디렉토리가 있습니다. 최대 검색 디렉토리를 100개로 지정하면 검색 과정이 디렉토리 A에서 끝납니다. 디렉토리 *A*가 100개 한계에 도달했으므로 사용자 시스템의 다른 디렉토리에서는 검색이 수행되지 않습니다. 검색 제한에 도달할 때까지 누적된 검색 결과는 검색이 제한을 초과했음을 알리는 오류 메시지와 함께 사용자에게 표시됩니다. 검색을 계속하려면 사용자가 다음 디렉토리에서 수동으로 검색을 다시 시작해야 합니다. 검색 작업은 하위 디렉토리 우선 방식으로 수행됩니다. 즉, 검색 작업은 사용자가 선택한 디렉토리의 모든 하위 디렉토리를 거친 후에 다음 디렉토리로 이동합니다.

5 [저장]을 눌러 완료합니다.

▼ 작업 권한을 구성하려면

원격 호스트에서 다음 작업을 할 수 있는 권한을 허용 또는 거부할 수 있습니다.

1 Portal Server 관리 콘솔에 관리자로 로그인합니다.

2 [Secure Remote Access] 탭과 [Netfile] 탭을 차례로 선택합니다.

3 [DN 선택] 목록에서 사용자 또는 조직의 DN을 선택하거나 DN을 추가합니다.

4 다음 속성을 수정합니다.

속성 이름	설명
파일 이름 바꾸기	사용자가 파일 이름을 바꿀 수 있게 하려면 [허용] 확인란을 선택합니다. 이 옵션은 기본적으로 선택됩니다.
파일/폴더 삭제	사용자가 파일 및 디렉토리를 삭제할 수 있게 하려면 [허용] 확인란을 선택합니다. 이 옵션은 기본적으로 선택됩니다.
파일 업로드	사용자가 파일을 업로드할 수 있게 하려면 [허용] 확인란을 선택합니다. 이 옵션은 기본적으로 선택됩니다.
파일/폴더 다운로드	사용자가 파일 또는 디렉토리를 다운로드할 수 있게 하려면 [허용] 확인란을 선택합니다. 이 옵션은 기본적으로 선택됩니다.
파일 검색	사용자가 파일 검색 작업을 수행할 수 있게 하려면 [허용] 확인란을 선택합니다. 이 옵션은 기본적으로 선택됩니다.
파일 메일	사용자가 메일에 액세스할 수 있게 하려면 [허용] 확인란을 선택합니다. 이 옵션은 기본적으로 선택됩니다.
파일 압축	사용자가 압축 유형을 선택할 수 있게 하려면 [허용] 확인란을 선택합니다. 이 옵션은 기본적으로 선택됩니다.
사용자 아이디 변경	사용자가 자신의 사용자 아이디를 변경할 수 있게 하려면 [허용] 확인란을 선택합니다. 사용자는 다양한 아이디를 사용하여 NetFile을 사용하는 호스트에 연결할 수 있습니다. 대규모 조직에서 사용자는 여러 개의 사용자 아이디를 가질 수 있습니다. 이때 사용자가 단일 사용자 아이디를 사용하도록 제한해야 할 수 있는데,이런 경우 [사용자 아이디 변경 허용] 옵션을 비활성화할 수 있습니다. 그러면 특정 조직의 모든 사용자가 해당 사용자 아이디를 변경할 수 없고 하나의 아이디(데스크탑 로그인 아이디)로만 NetFile을 사용하여 호스트에 연결할 수 있습니다. 또 다른 경우에, 사용자가 여러 컴퓨터에서 서로 다른 로그인 아이디를 가질 수 있으며 이 때에는 사용자가 필요에 따라 아이디를 변경하도록 허용해야 할 수 있습니다.
Microsoft Windows 도메인 변경	사용자가 기본 Microsoft Windows 도메인 호스트를 변경할 수 있게 하려면 [허용] 확인란을 선택합니다. 이 옵션은 기본적으로 선택됩니다. 사용자가 도메인 이름을 지정하면 이 도메인의 아이디와 비밀번호도 지정해야 합니다. 호스트의 아이디와 비밀번호를 사용해야 하는 경우 사용자가 [사용자 도메인 이름] 필드에서 도메인을 제거해야 합니다.

주 – 위 옵션 중에 선택 취소한 항목이 있는 경우 사용자가 Portal Server 데스크탑에 다시 로그인한 후에 변경 사항이 적용됩니다.

5 [저장]을 눌러 완료합니다.

◆ ◆ ◆ 15장

15 SSL(Secure Socket Layer) 가속기 구성

이 장에서는 다양한 Sun Java System Portal Server Secure Remote Access용 가속기에 대해 설명합니다.

이번 장은 다음 절로 구성됩니다.

가속기 소개

외부 가속기는 서버 CPU의 SSL(Secure Socket Layer) 기능을 분담함으로써 CPU가 다른 작업을 수행하도록 하여 SSL 트랜잭션의 처리 속도를 높이는 전용 하드웨어 코프로세서입니다.

Sun Crypto Accelerator 1000

Sun™ Crypto Accelerator 1000(Sun CA1000) 보드는 암호화 코프로세서로 작동하여 공용 키와 대칭 암호화를 가속화하는 짧은 형태의 PCI 보드입니다. 이 제품에는 외부 인터페이스가 없습니다. 이 보드는 내부 PCI 버스 인터페이스를 통해 호스트와 통신합니다. 이 보드는 eCommerce 응용 프로그램에서 보안 프로토콜을 위한 다양한 계산 집약적 암호화 알고리즘을 가속화하기 위한 목적으로 사용됩니다.

RSA [7] 및 Triple-DES (3DES) [8]와 같은 다수의 핵심 암호화 기능을 응용 프로그램에서 Sun CA1000으로 분담시켜 병렬로 수행할 수 있습니다. 그러면 CPU가 자유롭게 다른 작업을 수행할 수 있어 SSL 트랜잭션의 처리 속도가 증가합니다.

자세한 수행 단계는 232 페이지 “Crypto Accelerator 1000을 구성하려면”을 참조하십시오.

Crypto Accelerator 1000 사용

Portal Server Secure Remote Access가 설치되어 있고 게이트웨이 서버 인증서(직접 서명 또는 CA에서 발행)가 설치되었는지 확인합니다. 자세한 내용은 10 장을 참조하십시오.

232 페이지 “Crypto Accelerator 1000 사용”은 SSL 가속기를 설치하기 전에 필요한 정보를 추적하는 일을 돕는 점검 목록이며 Crypto Accelerator 1000 매개 변수와 값을 나열합니다.

표 15–1 Crypto Accelerator 1000 설치 점검 목록

매개 변수	값
SRA 설치 기본 디렉토리	/opt
SRA 인증서 데이터베이스 경로	/etc/opt/SUNWportal/cert/default
SRA 서버 인증서 별명	server-cert
영역	sra-keystore
영역 사용자	crypta

▼ Crypto Accelerator 1000을 구성하려면

1. **사용 설명서의 지침에 따라 하드웨어를 설치합니다. 다음을 참조하십시오.**

   `http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2450-11.pdf`

2. **CD에서 다음 패키지를 설치합니다.**

   `SUNWcrypm, SUNWcrypu, SUNWcrysu, SUNWdcar, SUNWcrypr, SUNWcrysl, SUNWdcamn, SUNWdcav`

3. **다음 패치를 설치합니다. (`http://sunsolve.sun.com`에서 얻을 수 있습니다.)**

   110383-01, 108528-05, 112438-01

4. **`pk12util` 및 `modutil` 도구가 있는지 확인하십시오.**

   이 도구는 `/usr/sfw/bin`에 설치되어 있습니다. `/usf/sfw/bin` 디렉토리에서 해당 도구를 사용할 수 없는 경우 Sun Java System 배포 매체에서 수동으로 SUNWtlsu를 추가해야 합니다.

   `Solaris_[sparc/x86]/Product/shared_components/`

5. **슬롯 파일을 만듭니다.**

   `vi /etc/opt/SUNWconn/crypto/slots`

   그리고 파일의 처음이자 유일한 라인으로 'crypta@sra'를 넣습니다.

6 영역을 만들고 설정합니다.

a. 루트로 로그인합니다.

b. 다음 명령을 입력합니다.

```
cd /opt/SUNWconn/bin/secadm
```

```
secadm> create realm=sra
```

영역 sra가 성공적으로 만들어졌습니다.

7 사용자를 만듭니다.

a. 다음 명령을 입력하고 응답합니다.

secadm> `set realm=sra`

secadm{srap}> `su`

secadm{root@sra}> `create user=crypta`

초기 비밀번호:

비밀번호 확인:

사용자 crypta가 성공적으로 만들어졌습니다.

8 만든 사용자로 로그인합니다.

secadm{root@sra}> `login user=crypta`

비밀번호:

secadm{crypta@sra}> `show key`

이 사용자에게 키가 없습니다.

9 Sun Crypto 모듈을 로드합니다.

환경 변수 `LD_LIBRARY_PATH`는 `/usr/lib/mps/secv1/`을 가리켜야 합니다.

다음을 입력합니다.

```
modutil -dbdir /etc/opt/SUNWportal/cert/default -add "Sun Crypto Module"
-libfile /opt/SUNWconn/crypto/lib/libpkcs11.so
```

다음 명령을 사용하여 이 모듈이 로드되었는지 확인합니다.

```
modutil -list -dbdir /etc/opt/SUNWportal/cert /default
```

10 게이트웨이 인증서와 키를 "Sun Crypto Module"로 내보냅니다.

환경 변수 `LD_LIBRARY_PATH`는 `/usr/lib/mps/secv1/`을 가리켜야 합니다.

다음을 입력합니다.

```
pk12util -o servercert.p12 -d /etc/opt/SUNWportal/cert/default -n server-cert
pk12util -i servercert.p12 -d /etc/opt/SUNWportal/cert/default -h "crypta@sra"
```

이제 show key 명령을 실행합니다.

```
secadm{crypta@sra}> show key
```

이 사용자에게 2개의 키가 나타나야 합니다.

11 /etc/opt/SUNWportal/cert/default/.nickname 파일에서 별명을 변경합니다.

```
vi /etc/opt/SUNWportal/cert/default/.nickname
```

`server-cert`를 `crypta@sra:server-cert`로 교체합니다.

12 가속화용 암호를 활성화합니다.

SUN CA1000은 RSA 기능을 가속화하지만 DES와 3DES 암호에 대한 가속만 지원합니다.

13 가속기를 사용할 수 있도록 `/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*을 수정합니다.

```
gateway.enable.accelerator=true
```

14 터미널 창에서 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

주 – 게이트웨이는 게이트웨이 프로필에서 https 포트로 언급된 포트의 일반 ServerSocket(비 SSL)에 바인딩합니다.

들어오는 클라이언트 트래픽에 대해 SSL 암호화 또는 복호화가 수행되지 않습니다. 가속기에서 이 작업을 수행합니다.

PDC는 이 모드에서 작동하지 않습니다.

Sun Crypto Accelerator 4000

Sun™ Crypto Accelerator 4000 보드는 Sun 서버에서 IPsec 및 SSL(대칭 및 비대칭 모두)에 대한 암호화 하드웨어 가속을 지원하는 기가비트 이더넷 기반 네트워크 인터페이스 카드입니다.

암호화되지 않은 네트워크 트래픽을 위한 표준 기가비트 이더넷 네트워크 카드로 작동하는 외에 이 보드에는 암호화 IPsec 트래픽에 높은 처리 속도를 지원할 암호 하드웨어가 포함되어 있습니다.

Crypto Accelerator 4000 보드는 하드웨어와 소프트웨어 모두에서 암호화 알고리즘을 가속화합니다. 암호 DES 및 3DES에 대한 대량 암호화도 지원합니다.

자세한 수행 단계는 235 페이지 “Configure Crypto Accelerator 4000을 구성하려면”을 참조하십시오.

Crypto Accelerator 4000 사용

SRA가 설치되어 있고 게이트웨이 서버 인증서(직접 서명 또는 CA에서 발행)가 설치되었는지 확인합니다. 다음 점검 목록으로 SSL 가속기를 설치하기 전에 필요한 정보를 쉽게 확인할 수 있습니다.

232 페이지 “Crypto Accelerator 1000 사용”에는 Crypto Accelerator 4000 매개 변수 및 값이 나와 있습니다.

표 15–2 Crypto Accelerator 4000 설치 점검 목록

매개 변수	값
Portal Server Secure Remote Access 설치 기본 디렉토리	/opt
SRA 인스턴스	default
SRA 인증서 데이터베이스 경로	/etc/opt/SUNWportal/cert/default
SRA 서버 인증서 별명	server-cert
CA4000 키 저장소	srap
CA4000 키 저장소 사용자	crypta

▼ Configure Crypto Accelerator 4000을 구성하려면

1 **사용 설명서의 지침에 따라 하드웨어와 소프트웨어 패키지를 설치합니다. 다음을 참조하십시오.**

http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2450-11.pdf

2 **다음 패치를 설치합니다. (http://sunsolve.sun.com에서 얻을 수 있습니다.)** 114795

3 **`certutil`, `pk12util` 및 `modutil` 도구가 있는지 확인하십시오.**

이 도구는 `/usr/sfw/bin` 아래 설치되어 있습니다.

`/usf/sfw/bin` 디렉토리에서 도구를 사용할 수 없는 경우에는

Sun Java System 배포 매체에서 수동으로 SUNWtlsu 패키지를 추가해야 합니다.

`Solaris_[sparc/x86]/Product/shared_components/`

4 **보드를 초기화합니다.**

`/opt/SUNWconn/bin/vcadm` 도구를 실행하여 암호화 보드를 초기화하고 다음 값을 설정합니다.

초기 보안 관리 이름: `sec_officer`

키 저장소 이름: `sra-keystore`

FIPS 140-2 모드에서 실행: `No`

5 사용자를 만듭니다.

vcaadm{vca0@localhost, sec_officer}> `create user`

새 사용자 이름: `crypta`

새 사용자 비밀번호 입력:

비밀번호 확인:

사용자 crypta가 성공적으로 만들어졌습니다.

6 키 저장소에 토큰을 매핑합니다.

```
vi /opt/SUNWconn/cryptov2/tokens
```

그리고 파일에 `sra-keystore`를 추가합니다.

7 대량 암호화의 사용을 설정합니다.

```
touch /opt/SUNWconn/cryptov2/sslreg
```

8 Sun Crypto 모듈을 로드합니다.

환경 변수 `LD_LIBRARY_PATH`는 `/usr/lib/mps/secv1/`을 가리켜야 합니다.

다음을 입력합니다.

```
modutil -dbdir /etc/opt/SUNWportal/cert/default -add "Sun Crypto Module"
-libfile /opt/SUNWconn/cryptov2/lib/libvpkcs11.so
```

다음 명령을 사용하여 이 모듈이 로드되었는지 확인할 수 있습니다.

```
modutil -list -dbdir /etc/opt/SUNWportal/cert/default
```

9 게이트웨이 인증서와 키를 "Sun Crypto Module"로 내보냅니다.

환경 변수 `LD_LIBRARY_PATH`는 `/usr/lib/mps/secv1/`을 가리켜야 합니다.

```
pk12util -o servercert.p12 -d /etc/opt/SUNWportal/cert/default -n server-cert
```

```
pk12util -i servercert.p12 -d /etc/opt/SUNWportal/cert/default -h "sra-keystore"
```

다음 명령을 사용하여 키가 내보내졌는지 확인할 수 있습니다.

```
certutil -K -h "sra-keystore" -d /etc/opt/SUNWportal/cert/default
```

10 `/etc/opt/SUNWportal/cert/default/.nickname` 파일에서 별명을 변경합니다.

```
vi /etc/opt/SUNWportal/cert/default/.nickname
```

server-cert를 sra-keystore:server-cert로 교체합니다.

11 가속화용 암호를 활성화합니다.

12 터미널 창에서 게이트웨이를 다시 시작합니다.

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

키 저장소 비밀번호를 입력하라는 게이트웨이 프롬프트가 표시됩니다.

"sra-keystore":crypta:crytpa-password에 대한 비밀번호 또는 PIN을 입력합니다.

주 – 게이트웨이는 게이트웨이 프로필에서 https 포트로 언급된 포트의 일반 ServerSocket(비 SSL)에 바인딩합니다.

들어오는 클라이언트 트래픽에 대해 SSL 암호화 또는 복호화가 수행되지 않습니다. 가속기에서 이 작업을 수행합니다.

PDC는 이 모드에서 작동하지 않습니다.

외부 SSL 장치 및 프록시 가속기

열린 모드에서 외부 SSL 장치를 Portal Server Secure Remote Access(SRA) 전방에서 실행할 수 있습니다. 이 장치는 클라이언트와 SRA 사이에 SSL 링크를 제공합니다.

다음 작업을 수행할 수 있습니다.

- 237 페이지 "외부 SSL 장치 가속기를 사용하려면"
- 238 페이지 "외부 SSL 장치 가속기를 구성하려면"

▼ 외부 SSL 장치 가속기를 사용하려면

1 SRA가 설치되어 있고 게이트웨이가 열린 모드(HTTP 모드)에서 실행되는지 확인합니다.

2 HTTP 연결을 사용합니다.

다음 표에는 외부 SSL 장치와 프록시 가속기 매개 변수 및 값이 정리되어 있습니다.

매개 변수	값
SRA 인스턴스	default
게이트웨이 모드	http

매개 변수	값
게이트웨이 포트	880
외부 장치/프록시 포트	443

▼ 외부 SSL 장치 가속기를 구성하려면

1. **사용 설명서의 지침에 따라 하드웨어와 소프트웨어 패키지를 설치합니다.**
2. **해당하는 경우 필요한 패치를 설치합니다.**
3. **HTTP를 사용하도록 게이트웨이 인스턴스를 구성합니다.**
4. **`platform.conf` 파일에 다음 값을 입력합니다.**

   ```
   gateway.enable.customurl=true

   gateway.enable.accelerator=true

   gateway.httpurl=https:// external-device-URL:port-number
   ```

5. **두 가지 방법으로 게이트웨이 알림을 구성할 수 있습니다.**
   - Access Manager가 포트 880에서 게이트웨이 컴퓨터와 접속할 수 있는 경우(HTTP로 세션 알림) `platform.conf` 파일에 값을 입력합니다.

     ```
     vi /etc/opt/SUNWportal/platform.conf.default

     gateway.protocol=http

     gateway.port=880
     ```

     - Access Manager가 포트 443에서 외부 장치/프록시와 접속할 수 있는 경우(HTTPS 세션 알림) `platform.conf` 파일에 값을 입력합니다.

       ```
       vi /etc/opt/SUNWportal/platform.conf.default

       gateway.host=External Device/Proxy Host Name

       gateway.protocol=https

       gateway.port=443
       ```

6. **SSL 장치/프록시가 작동하고 있으며 게이트웨이 포트로 트래픽을 넘기도록 구성되어 있는지 확인합니다.**
7. **터미널 창에서 게이트웨이를 다시 시작합니다.**

   ```
   ./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
   ```

3

Secure Remote Access Server 관리

Secure Remote Access Server에는 두 가지 관리 인터페이스가 있습니다.

- Portal Server 관리 콘솔
- **Sun Java System Portal Server 7.2 Command-Line Reference**의 1 장, “psadmin Utility”에 나와 있는 명령줄 유틸리티

대부분의 관리 작업은 로컬 또는 원격으로 웹 브라우저를 사용하여 액세스할 수 있는 웹 기반 Portal Server 관리 콘솔을 통해 수행됩니다. 자세한 내용은 **Sun Java System Portal Server 7.2 관리 설명서**의 “Portal Server 관리 콘솔 사용”을 참조하십시오.

그러나 파일 수정과 같은 작업은 UNIX 명령줄 인터페이스를 통해 관리해야 합니다.

◆ ◆ ◆ 16장

16 게이트웨이 관리

여기에 요약 내용이 나옵니다.

게이트웨이 관리 작업

이 절에서는 다음과 같은 Portal Server 게이트웨이 관리 작업을 설명합니다.

▼ 게이트웨이 프로필을 만들려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 누르고 [새 프로필]을 누릅니다.**
   새 프로필 페이지가 표시됩니다.
3. **새 게이트웨이 프로필의 이름을 입력합니다.**
4. **드롭다운 목록에서 새 프로필을 만들 때 사용할 프로필을 선택합니다.**
   기본적으로 만들어지는 새 프로필은 모두 사전 제공된 기본 프로필을 기준으로 합니다. 사용자 정의 프로필을 만든 경우 드롭다운 목록에서 해당 프로필을 선택할 수 있습니다. 새 프로필은 선택한 프로필의 모든 속성을 상속합니다.

   기존 프로필을 복사하여 새 프로필을 만드는 경우 포트도 동일하게 복사됩니다. 새 프로필의 포트를 기존 프로필과 충돌하지 않도록 변경하십시오.
5. **[확인]을 누릅니다.**
   새 프로필이 만들어지고 프로필 페이지에 나열됩니다.

주의 – 사용 중인 기존 포트와 충돌하지 않도록 인스턴스의 포트를 변경했는지 확인하십시오.

6 **인스턴스를 만들어야 하는 시스템에 텔넷으로 연결합니다. 기본 게이트웨이 인스턴스가 시작되어 이 시스템에서 실행됩니다.**

7 **지금 구성 모드에서 AM-SDK를 설치합니다.**

8 **지금 구성 모드에서 UI 설치 프로그램을 사용하여 게이트웨이를 설치하거나 나중에 구성 모드를 선택합니다.**

9 **/opt/SUNWportal/template/sra/GWConfig.properties.template 파일을 임시 위치에 복사합니다. 예:** /tmp

10 **필요에 따라 값을 수정합니다.**

주 – 이 값은 새 프로필에 대한 게이트웨이 인스턴스의 포트 번호와 일치해야 합니다.

11 **작업이 완료되면 다음 명령을 실행합니다.**

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file
location>.template -t gateway
```

12 **변경 사항을 적용하려면 이 게이트웨이 프로필 이름의 게이트웨이를 다시 시작합니다.**

./psadmin start-sra-instance –u amadmin – f *<password file>* –N *<profile name>*– t *<gateway>*

게이트웨이 시작 및 중지에 대한 자세한 내용은 243 페이지 "게이트웨이 인스턴스를 시작하려면"을 참조하십시오. 게이트웨이를 구성하려면 8 장을 참조하십시오.

▼ 같은 LDAP를 사용하여 게이트웨이 인스턴스를 만들려면

1 **암호의 암호화와 해독에 사용되는 키를 첫 게이트웨이와 같은 문자열로 대체합니다.**

```
am.encryption.pwd= string_key_specified_in gateway-install
```

2 **응용 프로그램 인증 모듈의 공유 비밀에 해당하는 키를 대체합니다.**

```
com.iplanet.am.service.secret= string_key_specified_in gateway-install
```

3 **/etc/opt/SUNWam/config/ums에서 serverconfig.xml의 다음 영역을 처음 설치한 Portal Server 인스턴스와 다른 값으로 수정합니다.**

```
<DirDN> cn=puser,ou=DSAME Users,dc=sun,dc=net</DirDN>

<DirPassword> string_key_specified_in gateway-install</DirPassword>

<DirDN>cn=dsameuser,ou=DSAME Users,dc=sun,dc=net</DirDN>

<DirPassword>string_key_specified_in gateway-install </DirPassword>
```

4 **Access Manager 서비스를 다시 시작합니다.**

▼ 게이트웨이 인스턴스를 시작하려면

기본적으로 게이트웨이는 사용자 noaccess로 시작됩니다.

1 **게이트웨이를 설치하고 필요한 프로필을 만든 후 다음 명령을 실행하여 게이트웨이를 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>
```

default는 설치 중에 만들어지는 기본 게이트웨이 프로필입니다. 나중에 고유한 프로필을 만들고 새 프로필로 게이트웨이를 다시 시작할 수 있습니다. 32 페이지 "게이트웨이 프로필 만들기"를 참조하십시오.

주 – 게이트웨이의 다른 인스턴스를 시작하려면 <*profile name*>을 적절한 프로필 이름으로 대체합니다.

서버(게이트웨이 인스턴스가 구성되어 있는 시스템)를 다시 시작하면 모든 게이트웨이 인스턴스가 다시 시작됩니다.

/etc/opt/SUNWportal 디렉토리에 백업된 프로필이 없는지 확인하십시오.

---

2 **다음 명령을 실행하여 지정 포트에서 게이트웨이가 실행되고 있는지 확인합니다.**

```
netstat -an | grep port-number
```

기본 게이트웨이 포트는 443입니다.

▼ 게이트웨이를 중지하려면

1 **게이트웨이를 중지하려면 다음 명령을 사용합니다.**

```
./psadmin stop-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>
```

주 – 게이트웨이의 다른 인스턴스를 시작하려면 <*profile name*>을 적절한 프로필 이름으로 대체합니다.

2 **다음 명령을 실행하여 게이트웨이 프로세스가 아직 실행되고 있는지 확인합니다.**

```
/usr/bin/ps -ef | grep entsys
```

▼ 관리 콘솔을 사용하여 게이트웨이를 시작 및 중지하려면

1 **Sun Java System Portal Server 7.2 관리 설명서의 "관리 콘솔에 로그인하려면"에 나와 있는 대로 수행합니다.**

2 **[Secure Remote Access] 탭을 선택합니다.**

3 **[인스턴스 관리] 하위 메뉴를 누릅니다.**

4 **[SRA 프록시 인스턴스]에서 인스턴스를 선택합니다.**

- **인스턴스를 시작하려면 [시작]을 누릅니다.**
- **인스턴스를 중지하려면 [중지]를 누릅니다.**

▼ 다른 프로필로 게이트웨이를 다시 시작하려면

- **게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>
```

▼ 게이트웨이를 다시 시작하려면

- **단말기 창에서 루트로 연결하고 다음 작업 중 하나를 수행합니다.**
  - 워치독 프로세스를 시작합니다.

```
./psadmin sra-watchdog -u uid -f password-filename -t instance-type on
```

`[--adminuser | -u] uid`	관리자의 고유 이름(DN) 또는 사용자 아이디를 지정합니다.
`[-passwordfile | -f] password-filename`	비밀 번호 파일에 관리자 비밀 번호를 지정합니다.
`[--type | -t] instance-type`	Secure Remote Access 인스턴스의 유형을 지정합니다. gateway, nlproxy또는 rwproxy 중 하나를 입력합니다.

워치독 명령에 대한 자세한 내용은 **Sun Java System Portal Server Command Line Reference Guide**를 참조하십시오.

그러면 crontab 유틸리티에 항목이 만들어지고 워치독 프로세스가 활성 상태가 됩니다. 워치독은 특정 컴퓨터 및 게이트웨이 포트에서 실행 중인 모든 게이트웨이 인스턴스를 모니터링하여 다운된 경우 게이트웨이를 다시 시작합니다.

▼ 가상 호스트를 지정하려면

1 **루트로 로그인하여 필요한 게이트웨이 인스턴스의 `platform.conf` 파일을 편집합니다.**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **다음 항목을 추가합니다.**

`gateway.virtualhost=`*fully-qualified-gateway-host gateway-ip-address fully-qualified-reverse-proxyhost*

`gateway.enable.customurl=true`(이 값은 기본적으로 false로 설정되어 있습니다.)

3 **게이트웨이를 다시 시작합니다.**

`./psadmin start-sra-instance –u amadmin – f` <*password file*> `–N` <*profile name*>`– t` <*gateway*>

값이 지정되어 있지 않으면 게이트웨이에서는 기본적으로 일반적인 작동을 합니다.

▼ 프록시를 지정하려면

1 **명령줄에서 다음 파일을 편집합니다.**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **다음 항목을 추가합니다.**

`http.proxyHost=`*proxy-host*
`http.proxyPort=`*proxy-port*
`http.proxySet=true`

3 **서버에 제출된 요청에 지정된 프록시를 사용할 수 있도록 게이트웨이를 다시 시작합니다.**

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t <gateway>`

▼ Netlet 프록시 인스턴스를 만들려면

1 **인스턴스를 만들어야 하는 시스템에 텔넷으로 연결합니다. 기본 게이트웨이 인스턴스가 시작되어 이 시스템에서 실행됩니다.**

2 **`/opt/SUNWportal/template/sra/NLPConfig.properties.template` 파일을 임시 위치에 복사합니다. 예: `/tmp`**

3 **필요에 따라 파일에서 새 프로필에 대한 값을 수정합니다.**

4 **작업이 완료되면 다음 명령을 실행합니다.**

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file location>.template -t nlproxy
```

5 **필요한 게이트웨이 프로필 이름으로 Netlet 프록시의 새 인스턴스를 시작하여 변경 사항이 적용되었는지 확인합니다.**

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t nlproxy`

▼ Netlet 프록시를 다시 시작하려면

- **단말기 창에서 루트로 연결하고 다음 작업 중 하나를 수행합니다.**

  - **워치독 프로세스를 시작합니다.**

    `psadmin sra-watchdog -u uid -f password-filename -t instance-type on`

    instance-type 대신 `nlproxy`를 입력합니다. 이 명령에 대한 자세한 내용은 **Sun Java System Portal Server Command Line Reference Guide**를 참조하십시오.

    그러면 crontab 유틸리티에 항목이 만들어지고 워치독 프로세스가 활성 상태가 됩니다. 워치독은 Netlet 프록시 포트를 모니터링하여 프록시가 다운되면 표시합니다.

  - **Netlet 프록시를 수동으로 시작합니다.**

    ```
    psadmin start-sra-instance -u uid -f password-filename -N sra-instance-name -t instance-type
    ```

instance-type 대신 nlproxy를 입력합니다. 이 이름은 필요한 Netlet 프록시 인스턴스에 해당하는 프로필 이름입니다. 이 명령에 대한 자세한 내용은 **Sun Java System Portal Server Command Line Reference Guide**를 참조하십시오.

▼ Rewriter 프록시 인스턴스를 만들려면

1. **인스턴스를 만들어야 하는 시스템에 텔넷으로 연결합니다. 기본 게이트웨이 인스턴스가 시작되어 이 시스템에서 실행됩니다.**
2. **/opt/SUNWportal/template/sra/GWConfig.properties.template 파일을 임시 위치에 복사합니다. 예:** /tmp
3. **필요에 따라 파일에서 새 프로필에 대한 값을 수정합니다.**
4. **작업이 완료되면 다음 명령을 실행합니다.**

   ```
   ./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file location>.template -t rwproxy
   ```

5. **필요한 게이트웨이 프로필 이름으로 Rewriter 프록시의 새 인스턴스를 시작하여 변경 사항이 적용되었는지 확인합니다.**

   ```
   ./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t rwproxy
   ```

▼ Rewriter 프록시를 다시 시작하려면

- **단말기 창에서 루트로 연결하고 다음 작업 중 하나를 수행합니다.**
  - **워치독 프로세스를 시작합니다.**

    ```
    psadmin sra-watchdog -u uid -f password-filename -t instance-type on
    ```

    instance-type 대신 rwproxy를 입력합니다. 이 명령에 대한 자세한 내용은 **Sun Java System Portal Server Command Line Reference Guide**를 참조하십시오.

    그러면 crontab 유틸리티에 항목이 만들어지고 워치독 프로세스가 활성 상태가 됩니다. 워치독은 Rewriter 프록시 포트를 모니터링하여 프록시가 다운되면 해당 프록시를 시작합니다.

  - **Rewriter 프록시를 수동으로 시작합니다.**

    ```
    start-sra-instance -u uid -f password-filename -N sra-instance-name -t instance-type
    ```

instance-type 대신 rwproxy를 입력합니다. 이 이름은 필요한 Rewriter 프록시 인스턴스에 해당하는 프로필 이름입니다. 이 명령에 대한 자세한 내용은 **Sun Java System Portal Server Command Line Reference Guide**를 참조하십시오.

▼ 역 프록시를 활성화하려면

1 **루트로 로그인하여 필요한 게이트웨이 인스턴스의** platform.conf **파일을 편집합니다.**

/etc/opt/SUNWportal/platform.conf. *gateway-profile-name*

2 **다음 항목을 추가합니다.**

gateway.virtualhost= *fully-qualified-gateway-host gateway-ip-address fully-qualified-reverse-proxyhost*

gateway.enable.customurl=true(이 값은 기본적으로 false로 설정되어 있습니다.)

gateway.httpurl= *http reverse-proxy-URL*

gateway.httpsurl=*https reverse-proxy-URL*

gateway.httpurl은 게이트웨이 프로필에서 HTTP 포트로 나열된 포트에서 수신된 요청에 대한 응답을 다시 쓰는 데 사용됩니다.

gateway.httpsurl은 게이트웨이 프로필에서 HTTPS 포트로 나열된 포트에서 수신된 요청에 대한 응답을 다시 쓰는 데 사용됩니다.

3 **게이트웨이를 다시 시작합니다.**

./psadmin start-sra-instance –u amadmin – f <*password file*> –N <*profile name*> – t <*gateway*>

값이 지정되어 있지 않으면 게이트웨이에서는 기본적으로 일반적인 작동을 합니다.

▼ 기존 PDC 인스턴스에 인증 모듈을 추가하려면

1 **Access Manager 관리 콘솔에 관리자로 로그인합니다.**

2 **필요한 조직을 선택합니다.**

3 **[보기] 드롭다운 상자에서 [서비스]를 선택합니다.**
서비스가 표시됩니다.

4 **[인증 구성]을 누릅니다.**
서비스 인스턴스 목록이 표시됩니다.

5 **Gatewaypdc를 누릅니다.**
Gatewaypdc 등록 정보 페이지가 표시됩니다.

6 **[편집]을 누릅니다.**
모듈 추가 페이지가 표시됩니다.

7 **[모듈 이름]을 선택하고 [플래그]를 [필요]로 설정합니다.**

8 **[확인]을 누릅니다.**

9 **모듈을 하나 이상 추가한 다음 [저장]을 누릅니다.**

10 gatewaypdc **등록 정보 페이지에서 [저장]을 누릅니다.**

11 **변경 사항이 적용되도록 게이트웨이를 다시 시작합니다.**
gateway-install-location/SUNWportal/bin/psadmin start-sra-instance –u amadmin – f <*password file*> –N <*profile name*>– t <*gateway*>

▼ 브라우저 캐싱을 비활성화하려면

1 **루트로 로그인하여 필요한 게이트웨이 인스턴스의** platform.conf **파일을 편집합니다.**
/etc/opt/SUNWportal/platform.conf.*gateway-profile-name*

2 **다음 행을 편집합니다.**
gateway.allow.client.caching=true
이 값은 기본적으로 true로 설정되어 있습니다. 값을 false로 변경하여 클라이언트 쪽에서 브라우저 캐싱을 비활성화합니다.

3 **게이트웨이를 다시 시작합니다.**
./psadmin start-sra-instance –u amadmin – f <*password file*> –N <*profile name*> – t <*gateway*>

▼ LDAP 디렉토리를 공유하려면

1 **첫 번째로 설치된 Portal Server 및 Access Manager 서버의 인스턴스와 동기화되도록** AMConfig.properties**의 다음 영역을 수정합니다.**
# 비밀번호 암호화 및 암호 해독에 사용하는 키.
am.encryption.pwd=t/vnY9Uqjf12NbFywKuAaaHibwlDFNLO <== REPLACE THIS STRING WITH THE ONE FROM FIRST PORTAL INSTALL

/* The following key is the shared secret for application auth module */
com.iplanet.am.service.secret=AQICxIPLNc0WWQRVlYZN0PnKgyvq3gTU8JA9 <== 이 문자열을 첫 번째 포털을 설치했을 때의 문자열로 바꾸십시오.

2 /etc/opt/SUNWam/config/ums에서 serverconfig.xml의 다음 영역을 처음 설치한 Portal Server 및 Access Manager 서버와 동기화되도록 수정합니다.

```
<DirDN>
    cn=puser,ou=DSAME Users,dc=sun,dc=net
</DirDN>
    <DirPassword>
         AQICxIPLNc0WWQT22gQnGgnCp9rUf+FuaqpY
         <==  REPLACE THIS STRING WITH THE ONE FROM FIRST PORTAL INSTALL
  </DirPassword>

<DirDN>
   cn=dsameuser,ou=DSAME Users,dc=sun,dc=net
</DirDN>
     <DirPassword>
          AQICxIPLNc0WWQT22gQnGgnCp9rUf+FuaqpY
          <==  REPLACE THIS STRING WITH THE ONE FROM FIRST PORTAL INSTALL
     </DirPassword>
```

3 Access Manager 서비스를 다시 시작합니다.

◆ ◆ ◆ 17장

17 연합 관리 시나리오

이 장은 다음과 같은 항목으로 구성됩니다.

연합 관리 사용

연합 관리를 사용하면 사용자가 하나의 네트워크 아이디를 가질 수 있도록 로컬 아이디를 집계할 수 있습니다. 연합 관리에서는 네트워크 아이디를 사용하여 사용자가 한 서비스 공급자의 사이트에 로그인할 경우 아이디를 재인증 받지 않고도 다른 서비스 공급자의 사이트에 액세스할 수 있도록 해 줍니다. 이를 단일 사인온이라 합니다.

연합 관리는 Portal Server에서 개방 모드 및 보안 모드로 구성할 수 있습니다. Portal Server 관리 설명서에서는 개방 모드로 연합 관리를 구성하는 방법에 대해 설명합니다. 연합 관리를 Portal Server Secure Remote Access 서버를 사용하여 보안 모드에서 구성하려면 개방 모드에서 올바로 작동하는지 확인해야 합니다. 사용자가 같은 브라우저에서 개방 모드와 보안 모드 모두에 대해 연합 관리를 사용할 수 있도록 하려면 쿠키를 지우고 브라우저로부터 캐싱해야 합니다.

연합 관리에 대한 자세한 내용은 **Access Manager Federation 관리 설명서**를 참조하십시오.

연합 관리 시나리오

사용자가 최초 서비스 공급자에게 인증을 받습니다. 서비스 공급자는 웹 기반 서비스를 제공하는 상업적 조직이거나 비영리 조직을 말합니다. 이렇게 넓은 범주에는 인터넷 포털, 대리점, 운송 공급자, 금융 기관, 엔터테인먼트 회사, 도서관, 대학 및 정부 기관이 모두 포함될 수 있습니다.

서비스 공급자는 쿠키를 사용하여 클라이언트 브라우저에 사용자의 세션 정보를 저장합니다. 쿠키에도 사용자의 아이디 공급자가 포함될 수 있습니다.

아이디 공급자는 인증 서비스를 전문적으로 제공하는 서비스 공급자를 말합니다. 인증을 위한 관리 서비스로서아이디 정보의 유지 및 관리도 수행합니다. 아이디 공급자에 의해 허가된 인증은 제휴 관계에 있는 모든 서비스 공급자에게 유효합니다.

사용자가 아이디 공급자와 제휴되지 않은 서비스에 액세스하려고 하면 아이디 공급자는 쿠키를 비제휴 서비스 공급자에게 전달합니다. 그런 다음 이 서비스 공급자가 쿠키에 명명된 아이디 공급자에게 액세스할 수 있습니다.

그러나 쿠키는 여러 DAN 도메인에서 읽을 수 없기 때문에서비스 공급자를 올바른 아이디 공급자에게 리디렉션하여 사용자에게 단일 사인온이 가능하도록 공용 도메인 쿠키 서비스를 사용합니다.

연합 관리 리소스 구성

연합 자원, 서비스 공급자, 아이디 공급자 및 공용 도메인 쿠키 서비스(CDCS)는 상주해 있는 위치를 기준으로 게이트웨이 프로필에 구성됩니다. 이 절에서는 3가지 시나리오를 구성하는 방법에 대해 설명합니다.

▼ 연합 관리 자원을 구성하려면

1. **모든 자원이 기업 인트라넷 안에 있는 경우**
2. **일부 자원이 기업 인트라넷에 있지 않거나 아이디 공급자가 인터넷에 상주하는 경우**
3. **일부 자원이 기업 인트라넷에 있지 않거나 서비스 공급자는 인터넷에 상주하는 타사이고 아이디 공급자는 게이트웨이에서 보호되는 경우**

구성 1

이 구성에서는 서비스 공급자, 아이디 공급자 및 공용 도메인 쿠키 서비스가 같은 기업 인트라넷에 배치되고 아이디 공급자는 인터넷 DNS(Domain Name Server)에 게시되지 않습니다. CDCS는 선택 사항입니다.

이 구성에서는 게이트웨이가 Portal Server가 되는 서비스 공급자를 지정합니다. 이 구성은 Portal Server의 다중 인스턴스에 유효합니다.

▼ 게이트웨이를 서비스 공급자(Portal Server)로 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 적절한 게이트웨이 프로필을 선택하여 해당 속성을 수정합니다.**
   게이트웨이 프로필 편집 페이지가 표시됩니다.
3. **[핵심] 탭을 선택합니다.**
4. **[쿠키 관리 사용] 확인란을 선택하고 쿠키 관리의 사용을 활성화합니다.**
5. **[보안] 탭을 선택합니다.**
6. **[Portal Server] 필드에 [인증되지 않은 URL] 목록에 나열된 `/amserver` 또는 `/portal/dt` 등의 상대 URL을 사용할 Portal Server 이름을 입력합니다. 예:**

   `http://` *idp-host:port*`/amserver/js`

   `http://` *idp-host:port*`/amserver/UI/Login`

   `http://`*idp-host:port* `/amserver/css`

   `http://`*idp-host:port* `/amserver/SingleSignOnService`

   `http://`*idp-host:port*`/amserver/UI/blank`

   `http://`*idp-host:port* `/amserver/postLogin`

   `http://` *idp-host:port*`/amserver/login_images`
7. **[Portal Server] 필드에 Portal Server 이름을 입력합니다. 예를 들어, `/amserver`와 같이 입력합니다.**
8. **[저장]을 누릅니다.**
9. **[보안] 탭을 선택합니다.**
10. **[인증되지 않은 URL] 목록에 연합 자원을 추가합니다. 예:**

    `/amserver/config/federation`

    `/amserver/IntersiteTransferService`

    `/amserver/AssertionConsumerservice`

```
/amserver/fed_images

/amserver/preLogin

/portal/dt
```

11 **[추가]를 누릅니다.**

12 **[저장]을 누릅니다.**

13 **웹 프록시에서 [인증되지 않은 URL] 목록에 나열된 URL에 연결해야 하는 경우 [배포] 탭을 선택합니다.**

14 **[도메인 및 하위 도메인의 프록시] 필드에 필요한 웹 프록시를 입력합니다.**

15 **[추가]를 누릅니다.**

16 **[저장]을 누릅니다.**

17 **터미널 창에서 게이트웨이를 다시 시작합니다.**

`./psadmin start-sra-instance –u amadmin – f` <*password file*> `–N` <*profile name*>`– t` <*gateway*>

구성 2

이 구성에서는 아이디 공급자, 아이디 공급자 및 공용 도메인 쿠키 공급자(CDCP)가 같은 기업 인트라넷에 배치되지 않았거나 아이디 공급자가 인터넷에 상주하는 타사 공급자입니다.

이 구성에서는 게이트웨이가 Portal Server가 되는 서비스 공급자를 지정합니다. 이 구성은 Portal Server의 다중 인스턴스에 유효합니다.

▼ 게이트웨이를 서비스 공급자(Portal Server)로 구성하려면

1 **Portal Server 관리 콘솔에 관리자로 로그인합니다.**

2 **[Secure Remote Access] 탭을 선택하고 적절한 게이트웨이 프로필을 선택하여 해당 속성을 수정합니다.**

3 **[핵심] 탭을 선택합니다.**

4 **[쿠키 관리 사용] 확인란을 선택하고 쿠키 관리의 사용을 활성화합니다.**

5 **[Portal Server] 필드에 [인증되지 않은 URL] 목록에 나열된 /amserver 또는 /portal/dt 등의 상대 URL을 사용할 서비스 공급자의 Portal Server 이름을 입력합니다.**

`http://`*idp-host:port*`/amserver/js`

`http://`*idp-host:port* `/amserver/UI/Login`

`http://`*idp-host:port* `/amserver/css`

`http://` *idp-host:port*`/amserver/SingleSignOnService`

`http://`*idp-host:port* `/amserver/UI/blank`

`http://`*idp-host:port* `/amserver/postLogin`

`http://` *idp-host:port*`/amserver/login_images`

6 **[저장]을 누릅니다.**

7 **[보안] 탭을 누릅니다.**

8 **[인증되지 않은 URL] 목록에 연합 자원을 추가합니다. 예:**

`/amserver/config/federation`

`/amserver/IntersiteTransferService`

`/amserver/AssertionConsumerservice`

`/amserver/fed_images`

`/amserver/preLogin`

`/portal/dt`

9 **[추가]를 누릅니다.**

10 **[저장]을 누릅니다.**

11 **웹 프록시에서 [인증되지 않은 URL] 목록에 나열된 URL에 연결해야 하는 경우 [배포] 탭을 선택합니다.**

12 **[도메인 및 하위 도메인의 프록시] 필드에 웹 프록시 정보를 입력합니다.**

13 **[추가]를 누릅니다.**

14 **[저장]을 누릅니다.**

15 **터미널 창에서 게이트웨이를 다시 시작합니다.**

`./psadmin start-sra-instance –u amadmin – f <`*password file*`> –N <`*profile name*`>– t <`*gateway*`>`

구성 3

이 구성에서는 아이디 공급자, 아이디 공급자 및 공용 도메인 쿠키 공급자(CDCP)가 같은 기업 인트라넷에 배치되지 않았거나 서비스 공급자가 인터넷에 상주하는 타사이고 아이디 공급자는 게이트웨이에 의해 보호됩니다.

이 구성에서는 게이트웨이가 Portal Server가 되는 아이디 공급자를 지정합니다.

이 구성은 Portal Server의 다중 인스턴스에 유효합니다. 이 구성은 인터넷에서는 구현되는 경우가 거의 없지만 어떤 기업 네트워크에는 인트라넷에 이러한 구성이 있을 수 있습니다. 즉, 아이디 공급자는 방화벽으로 보호되는 서브넷에 있고 서비스 공급자는 기업 네트워크 내에서 직접 액세스 가능한 경우를 말합니다.

▼ 게이트웨이를 Identity 공급자(Portal Server)로 구성하려면

1. **Portal Server 관리 콘솔에 관리자로 로그인합니다.**
2. **[Secure Remote Access] 탭을 선택하고 적절한 게이트웨이 프로필을 선택하여 해당 속성을 수정합니다.**
3. **[핵심] 탭을 선택합니다.**
4. **[쿠키 관리 사용] 확인란을 선택하고 쿠키 관리의 사용을 활성화합니다.**
5. **[Portal Server] 필드에 [인증되지 않은 URL] 목록에 나열된 `/amserver` 또는 `/portal/dt` 등의 상대 URL을 사용할 서비스 공급자의 Portal Server 이름을 입력합니다.**

   `http://`*idp-host:port*`/amserver/js`

   `http://`*idp-host:port* `/amserver/UI/Login`

   `http://`*idp-host:port* `/amserver/css`

   `http://` *idp-host:port*`/amserver/SingleSignOnService`

   `http://`*idp-host:port* `/amserver/UI/blank`

   `http://`*idp-host:port* `/amserver/postLogin`

   `http://` *idp-host:port*`/amserver/login_images`
6. **[저장]을 누릅니다.**
7. **[보안] 탭을 선택합니다.**
8. **[인증되지 않은 URL] 목록에 연합 자원을 추가합니다. 예:**

   `/amserver/config/federation`

   `/amserver/IntersiteTransferService`

```
/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt
```

9 **[추가]를 누릅니다.**

10 **[저장]을 누릅니다.**

11 **웹 프록시에서 [인증되지 않은 URL] 목록에 나열된 URL에 연결해야 하는 경우 [배포] 탭을 선택합니다.**

12 **[도메인 및 하위 도메인의 프록시] 필드에 웹 프록시 정보를 입력합니다.**

13 **[추가]를 누릅니다.**

14 **[저장]을 누릅니다.**

15 **터미널 창에서 게이트웨이를 다시 시작합니다.**

```
./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t
<gateway>
```

◆ ◆ ◆ 부 록 A

구성 속성

이 부록에서는 각 Portal Server Secure Remote Access 구성 요소에 대해 Portal Server 관리 콘솔을 통해 Sun Java System Portal Server Secure Remote Access용으로 구성할 수 있는 속성을 설명합니다.

액세스 제어 서비스

259 페이지 “액세스 제어 서비스”에는 액세스 제어 서비스 속성이 정리되어 있습니다.

표 A–1 액세스 제어 서비스 속성

속성	기본값	설명
거부된 URL		최종 사용자가 게이트웨이를 통해 액세스할 수 없는 URL 목록
허용된 URL	*	최종 사용자가 게이트웨이를 통해 액세스할 수 없는 URL 목록
단일 사인온을 사용하지 않는 호스트		호스트 목록에 대해 단일 사인온 사용을 해제합니다.
세션별 단일 사인온 활성화		세션에 대해 단일 사인온의 사용을 설정합니다.

표 A–1 액세스 제어 서비스 속성 *(계속)*

속성	기본값	설명
허용된 인증 수준	*	인증의 신뢰 정도를 나타냅니다. 모든 인증 수준을 허용하려면 별표(*)를 사용합니다. 인증 수준에 대한 자세한 내용은 **Access Manager 관리 설명서**를 참조하십시오.

게이트웨이 서비스

게이트웨이 서비스를 누르면 오른쪽 표시 영역에 새 프로필을 만들기 위한 버튼과 만든 게이트웨이 프로필 목록이 표시됩니다.

[새로 만들기]를 누르면 다음 표시 영역에 새 게이트웨이 프로필 이름을 입력하라는 메시지가 표시됩니다. 기본 템플리트 또는 이전에 만든 게이트웨이 프로필을 템플리트로 사용할 수 있습니다.

나열된 게이트웨이 프로필 이름 중 하나를 누르면 탭 목록이 제시됩니다. 다음과 같습니다.

- 260 페이지 “핵심”
- 262 페이지 “프록시”
- 263 페이지 “보안”
- 265 페이지 “Rewriter”

핵심

260 페이지 “핵심”에는 게이트웨이 서비스 핵심 속성이 나와 있습니다.

표 A–2 게이트웨이 서비스 핵심 속성

속성	기본값	설명
HTTPS 연결 사용		HTTPS 연결의 사용을 설정합니다.
HTTPS 포트	443	HTTPS 포트를 지정합니다.
HTTP 연결 사용	*	HTTP 연결의 사용을 설정합니다.
HTTP 포트	80	HTTP 포트를 지정합니다.
Rewriter 프록시 사용	*	게이트웨이와 인트라넷 사이에서 보안 HTTP 트래픽의 사용을 설정합니다. Rewriter 프록시와 게이트웨이는 같은 게이트웨이 프로필을 사용합니다.

표 A–2 게이트웨이 서비스 핵심 속성 *(계속)*

속성	기본값	설명
Rewriter 프록시 목록		Rewriter 프록시 목록. Rewriter 프록시의 인스턴스가 여러 개인 경우에는 *host-name:port* 형식으로 각각에 대한 세부 사항을 입력합니다.
Netlet 사용	선택	TCP/IP(Telnet 및 SMTP 등), HTTP 응용 프로그램 및 고정 포트 응용 프로그램에 보안을 사용합니다.
Proxylet 사용	선택	클라이언트 컴퓨터에서 Proxylet의 다운로드를 허용합니다.
Netlet 프록시 사용		클라이언트로부터 게이트웨이를 거쳐 인트라넷에 상주하는 Netlet 프록시까지 보안 터널을 확장함으로써 게이트웨이와 인트라넷 사이의 Netlet 트래픽 보안을 강화합니다. 포털 서버에서 응용 프로그램을 사용하지 않으려는 경우 해제합니다.
Netlet 프록시 호스트		Netlet 프록시 호스트를 다음 형식으로 나열합니다. hostname:port
쿠키 관리 사용		사용자가 액세스할 수 있는 모든 웹 사이트의 사용자 세션을 추적하고 관리합니다. 포털 서버가 포털 서버 사용자 세션을 관리하기 위해 사용하는 쿠키에는 적용되지 않습니다.
HTTP 지속 연결 사용	선택	게이트웨이에서 HTTP 지속 연결을 사용하여 웹 페이지의 모든 개체(이미지 및 스타일 시트 등)에 대해 소켓이 열리는 것을 방지합니다.
지속 연결당 최대 요청 수	10	지속 연결당 요청 수를 지정합니다.
지속적 소켓 연결 시간 초과	50	소켓이 닫힐 때까지 경과해야 하는 시간을 지정합니다.
반환 시간을 위한 계정의 유예 시간 초과	20	브라우저가 요청을 보낸 후 요청이 게이트웨이에 도달하기 위한 유예 시간과 응답을 보내는 게이트웨이와 실제로 이를 받는 브라우저 사이의 시간을 지정합니다.
사용자 세션 쿠키가 전달될 URL		서블릿 및 CGI가 Portal Server의 쿠키를 수신하고 API를 사용하여 사용자를 확인하도록 합니다.
최대 연결 대기 길이	50	게이트웨이가 허용할 수 있는 최대 동시 연결을 지정합니다.

표 A–2 게이트웨이 서비스 핵심 속성 (계속)

속성	기본값	설명
게이트웨이 시간 초과(초)	120	게이트웨이와 브라우저의 연결이 시간 초과되기까지 소요되는 시간(초)을 지정합니다.
최대 스레드 풀 크기	200	게이트웨이 스레드 풀에서 사전에 생성할 수 있는 최대 스레드 수를 지정합니다.
캐시된 소켓 시간 초과	200	게이트웨이가 Portal Server와의 연결에서 시간 초과할 시간을 초 단위로 지정합니다.
Portal Server		`http://` *portal server name:port -number* 형식으로 Portal Server를 지정합니다. 게이트웨이는 요청에 대한 서비스를 제공하기 위해 라운드 로빈 방식으로 나열된 각 Portal Server에 연결을 시도합니다.
서버 재시도 간격(초)	120	Portal Server, Rewriter 프록시 또는 Netlet 프록시를 사용할 수 없게 되는 경우(충돌이 있거나 중단된 경우 등) 시작 요청 사이의 시간 간격을 지정합니다.
외부 서버 쿠키 저장		게이트웨이에서 게이트웨이를 통해 액세스하는 타사 응용프로그램이나 서버에 대한 쿠키를 저장하고 관리할 수 있습니다.
URL에서 세션 정보 얻기		쿠키가 지원되는지 여부에 상관 없이 세션 정보를 URL의 일부로 코드화합니다. 게이트웨이는 클라이언트의 브라우저에서 보내는 세션 쿠키를 사용하지 않고 검증을 위해 URL에 있는 이 세션 정보를 사용합니다.

프록시

262 페이지 “프록시”에는 게이트웨이 서비스 프록시 속성이 나와 있습니다.

표 A–3 게이트웨이 서비스 프록시 속성

속성	기본값	설명
프록시 사용		웹 프록시의 사용을 설정합니다.
웹 프록시 URL 사용		게이트웨이가 [도메인 및 하위 도메인의 프록시] 목록에 표시된 웹 프록시를 통해서만 연결해야 하는 URL을 나열하며 여기에는 [프록시 사용] 옵션이 해제된 경우도 해당됩니다.

표 A–3 게이트웨이 서비스 프록시 속성 (계속)

속성	기본값	설명
웹 프록시 URL 사용 안 함		게이트웨이가 직접 연결할 수 있는 URL을 나열합니다.
도메인 및 하위 도메인의 프록시	iportal.com sun.com	특정 도메인의 특정 부속 도메인에 접속하기 위해 사용할 프록시를 지정합니다.
프록시 비밀번호 목록		프록시 서버가 일부 또는 모든 사이트에 액세스하는 데 인증이 필요한 경우 게이트웨이가 지정된 프록시 서버를 인증하는 데 필요한 서버 이름, 사용자 이름 및 비밀 번호를 지정합니다.
자동 프록시 구성 지원 사용		[도메인 및 하위 도메인의 프록시] 필드에 제공된 정보가 무시되도록 지정합니다.
자동 프록시 구성 파일 위치		PAC 지원에 사용할 파일 위치를 지정합니다.
웹 프록시를 통한 Netlet 터널링 사용		클라이언트로부터의 보안 터널을 게이트웨이를 통해 인트라넷에 있는 웹 프록시로 연장합니다.

보안

263 페이지 “보안”에는 게이트웨이 서비스 보안 속성이 나와 있습니다.

표 A–4 게이트웨이 서비스 보안 속성

속성	기본값	설명
HTTP 기본 인증 사용	선택	BASIC으로 보호된 웹 사이트를 다시 방문할 때 자격 증명 정보를 다시 입력하지 않도록 아이디와 비밀번호를 저장할 수 있습니다.

표 A–4 게이트웨이 서비스 보안 속성 (계속)

속성	기본값	설명
비인증 URL	/portal/desktop/images /amserver/login_images /portal/desktop/css /amserver/jss /amconsole/console/css /portal/searchadmin/console/js /amconsole/console/js /amserver/css	이미지가 있는 디렉토리와 같이 인증이 필요 없는 URL을 지정합니다.
인증서 사용 가능 게이트웨이 호스트		인증서 사용 가능 게이트웨이 호스트를 나열합니다.
40비트 암호화 허용		40비트(취약) SSL(Secure Sockets Layer) 연결을 허용합니다. 이 옵션을 선택하지 않으면 128비트 연결만 지원됩니다.
SSL 버전 2.0 사용	선택	SSL 버전 2.0의 사용을 설정합니다. SSL 2.0을 비활성화하면 이전 SSL 2.0만 지원하는 브라우저에서 SRA를 인증할 수 없습니다.그러면 보안 수준이 높아집니다.
SSL 암호 선택 사용		SSL 암호 선택의 사용을 설정합니다. 사전 구성된 모든 암호의 지원을 선택하거나 필요한 암호만 개별적으로 선택할 수 있습니다. 각 게이트웨이 인스턴스에 특정 SSL 암호를 선택할 수 있습니다.
SSL2 암호		선택할 수 있는 SSL 버전 2 암호를 나열합니다.
SSL3 암호		선택할 수 있는 SSL 버전 3 암호를 나열합니다.
TLS 암호		TLS 암호를 나열합니다.
SSL 버전 3.0 사용	선택	SSL 버전 3.0을 사용합니다. SSL 3.0을 비활성화하면 SSL 3.0만 지원하는 브라우저에서 SRA를 인증할 수 없습니다. 그러면 보안 수준이 높아집니다.
Null 암호화 사용		Null 암호를 활성화합니다.
인증된 SSL 도메인		인증된 SSL 도메인을 나열합니다.
쿠키를 안전하다고 표시		쿠키를 안전하다고 표시합니다. [쿠키 관리 사용] 옵션을 선택해야 합니다.

Rewriter

Rewriter 탭에는 두 개의 하위 부분이 있습니다.

기본

265 페이지 “기본”에는 게이트웨이 서비스 Rewriter 기본 속성이 나와 있습니다.

표 A–5 게이트웨이 서비스 Rewriter 속성 - 기본

속성	기본값	설명
모든 URI 다시 쓰기 사용		도메인 및 부속 도메인의 프록시 목록에 있는 항목을 점검하지 않고 모든 URI가 다시 작성되도록 지정합니다.

규칙 집합에 URI 매핑	`*://*.iportal.com*/portal/*|default_gateway_ruleset`
	`*/portal/NetFileOpenFileServlet*`
	`|null_ruleset`
	`*|generic_ruleset`
	`REPLACE_WITH_IPLANET_MAIL_SERVER_NAME|iplanet_mail_ruleset`
	`REPLACE_WITH_EXCHANGE_SERVER_NAMEexchange_2000sp3_owa_ruleset`
	`*://*.iportal.com*/amconsole/*|default_gateway_ruleset`
	`REPLACE_WITH_INOTES_SERVER_NAME|inotes_ruleset`
	`http*://*/portal/NetFileController*|null_ruleset`
		[규칙 집합에 URI 매핑] 목록을 사용하여 도메인을 규칙 집합에 연결합니다. 규칙 집합은 Access Manager 관리 콘솔의 [Portal Server 구성]에서 만들어집니다.

구문 분석기를 MIME 유형에 매핑	`JAVASCRIPT=application/x-java`
	`XML=text/xml`
	`HTML=text/html;text/htm;text/x-component;text/wml; text/vnd.wap.wml`
	`CSS=text/css`
		HTML, JAVASCRIPT, CSS 또는 XML에 새로운 MIME 유형을 연결합니다. 여러 항목인 경우는 세미콜론이나 쉼표를 사용하여 구분합니다.

표 A–5 게이트웨이 서비스 Rewriter 속성 - 기본 *(계속)*

속성	기본값	설명
다시 쓰지 않는 URI		다시 쓰지 않을 URI를 나열합니다. 참고: #*를 이 목록에 추가하면 규칙 집합에 href 규칙이 포함되어 있더라도 URI 다시 쓰기를 허용합니다.
기본 도메인		기본 도메인 및 부속 도메인에 대한 호스트 이름을 확인합니다. 이 정보는 설치 중에 구성됩니다.

고급

266 페이지 “고급”에는 게이트웨이 서비스 Rewriter 고급 속성이 나와 있습니다.

표 A–6 게이트웨이 서비스 Rewriter 속성 고급

속성	기본값	설명
MIME 추측 사용		MIME이 전송되지 않을 때 MIME 추측의 사용을 설정합니다. 구분 분석기와 URI의 매핑 목록 상자에 데이터를 추가해야 합니다.
구문 분석기와 URI의 매핑		구문 분석기를 URI에 매핑합니다. 각 URI는 세미콜론으로 구분합니다. 예: HTML=*.html; *.htm;*Servlet 이 것은 html, htm 또는 Servlet 확장을 가진 모든 페이지에 대한 컨텐트를 다시 쓰기 위해 Rewriter가 사용된다는 것을 의미합니다.
마스크 사용		Rewriter가 페이지의 인트라넷 URL이 보이지 않도록 URI를 다시 쓸 수 있습니다.
마스킹용 씨드 문자열		URI를 마스크하는 데 사용되는 씨드 문자열을 지정합니다. 마스크 알고리즘을 통해 이 임의 문자열을 생성합니다.

표 A–6 게이트웨이 서비스 Rewriter 속성 고급 (계속)

속성	기본값	설명
마스크하지 않을 URI		마스크하지 않을 인터넷 URI를 지정합니다. 응용 프로그램(애플릿 등)에 인터넷 URI가 필요한 경우에 사용됩니다. 예를 들어 다음을 목록 상자에 추가하면 */Applet/Param* 컨텐트 URI http://abc.com/Applet/Param1.html이 규칙 집합의 규칙에서 매칭되는 경우 URL이 마스크되지 않습니다.
게이트웨이 프로토콜을 원본 URI 프로토콜과 같게 만들기		HTML 컨텐트의 참조 리소스에 액세스할 때 Rewriter가 일관된 프로토콜을 사용하도록 합니다. 이는 Javascript로 생성된 동적 URI가 아닌 정적 URI에만 적용됩니다.

NetFile 서비스

NetFile 서비스를 누르면 오른쪽 창에 탭이 표시됩니다. 다음과 같습니다.

- 267 페이지 “호스트”
- 268 페이지 “권한”
- 269 페이지 “보기”
- 269 페이지 “작업”
- 271 페이지 “일반”

호스트

호스트 탭에는 두 개의 하위 부분이 있습니다.

- 267 페이지 “구성”
- 268 페이지 “액세스”

구성

267 페이지 “구성”에는 NetFile 호스트 구성 속성이 나와 있습니다.

표 A–7 NetFile 서비스 호스트 구성 속성

속성	기본값	설명

표 A–7 NetFile 서비스 호스트 구성 속성 (계속)

OS 문자 집합	유니코드(UTF-8)	호스트와의 통신을 위한 기본 인코딩으로 사용할 문자 집합을 지정합니다.
호스트 검색 순서	WIN, NETWARE, FTP, NFS	호스트 검색 순서를 지정합니다.
공통 호스트		NetFile을 통해 모든 원격 NetFile 사용자가 사용할 수 있게 만들 호스트를 지정합니다.
기본 도메인		허용된 호스트에 접속하기 위해 NetFile이 사용해야 하는 기본 도메인을 지정합니다.
기본 Microsoft Windows 도메인/워크그룹		사용자가 Windows 호스트에 액세스할 때 선택하는 기본 Windows 도메인 또는 워크그룹을 지정합니다.
기본 WINS/DNS 서버		NetFile이 Windows 호스트에 액세스할 때 사용하는 WINS/DNS 서버를 지정합니다.

액세스

268 페이지 “액세스”에는 NetFile 서비스 호스트 액세스 속성이 나와 있습니다.

표 A–8 NetFile 서비스 호스트 액세스 속성

속성	기본값	설명
Windows 호스트에 액세스 허용	선택	Microsoft Windows 호스트에 액세스를 허용합니다.
FTP 호스트에 액세스 허용	선택	FTP 호스트에 액세스를 허용합니다.
NFS 호스트에 액세스 허용	선택	NFS 호스트에 액세스를 허용합니다.
Netware 호스트에 액세스 허용	선택	Netware 호스트에 액세스를 허용합니다.
허용된 호스트	*	사용자가 NetFile을 통해 액세스할 수 있는 호스트를 지정합니다.
거부된 호스트		사용자가 NetFile을 통해 액세스할 수 없는 호스트를 지정합니다.

권한

사용자가 NetFile을 사용하기 시작한 후에 이러한 옵션을 비활성화하면 사용자가 NetFile을 로그아웃하고 다시 로그인하는 경우에만 변경 사항이 적용됩니다.

268 페이지 “권한”에는 NetFile 서비스 권한 속성이 나와 있습니다.

표 A–9 NetFile 서비스 권한 속성

속성	기본값	설명
파일 이름 변경 허용	선택	사용자가 파일의 이름을 바꿀 수 있습니다.
파일/폴더 삭제 허용	선택	사용자가 파일 및 폴더를 삭제할 수 있습니다.
파일 업로드 허용	선택	사용자가 파일을 업로드할 수 있습니다.
파일/폴더 다운로드 허용	선택	사용자가 파일 및 폴더를 다운로드할 수 있습니다.
파일 검색 허용	선택	사용자가 검색할 수 있습니다.
파일 메일 허용	선택	파일 메일을 허용합니다.
파일 압축 허용	선택	파일 압축을 허용합니다.
사용자 아이디 변경 허용	선택	사용자가 다른 아이디를 사용할 수 있습니다.
Windows 도메인 변경 허용	선택	사용자가 Microsoft Windows 도메인을 변경할 수 있습니다.

보기

269 페이지 “보기”에는 NetFile 서비스 보기 속성이 나와 있습니다.

표 A–10 NetFile 서비스 보기 속성

속성	기본값	설명
창 크기	700|400	사용자 데스크탑에서 픽셀 단위로 NetFile 창의 크기를 지정합니다. 잘못된 값을 입력하면 NetFile이 기본값을 사용합니다.
창 위치	100|50	사용자 데스크탑에서 NetFile 창이 표시되는 위치를 지정합니다. 잘못된 값을 입력하면 NetFile이 기본값을 사용합니다.

작업

작업 탭에는 다음 하위 부분이 있습니다.

- 270 페이지 “트래픽”
- 270 페이지 “검색”
- 270 페이지 “압축”

트래픽

270 페이지 "트래픽"에는 NetFile 서비스 작업 트래픽 속성이 나와 있습니다.

표 A–11 NetFile 서비스 작업 - 트래픽 속성

속성	기본값	설명
임시 디렉토리 위치	/tmp	다양한 NetFile 파일 작업을 위한 임시 디렉토리를 지정합니다. 웹 서버 실행에 사용하고 있는 아이디(nobody 또는 noaccess 등)에 지정 디렉토리에 대한 rwx 권한이 있는지 확인하십시오. 이 아이디에 필요한 임시 디렉토리의 전체 경로에 대한 rx 권한이 있는지도 확인하십시오. NetFile에 별도 임시 디렉토리를 만들어야 하는 경우가 있습니다. Portal Server의 모든 모듈에 공통된 임시 디렉토리를 지정하면 디스크 공간이 금방 부족해질 수 있습니다. 임시 디렉토리에 공간이 없으면 NetFile이 작동하지 않습니다.
파일 업로드 한계(MB)	5	업로드할 수 있는 최대 파일 크기를 지정합니다. 잘못된 값을 입력하면 NetFile이 잘못된 값을 기본값으로 재설정합니다. 정수 값만 입력해야 합니다. 사용자마다 다른 파일 업로드 제한 크기를 지정할 수 있습니다.

검색

270 페이지 "검색"에는 NetFile 서비스 작업 검색 속성이 나와 있습니다.

표 A–12 NetFile 서비스 작업 - 검색 속성

속성	기본값	설명
디렉토리 검색 제한	100	한번의 검색으로 검색되는 최대 디렉토리 수를 지정합니다.

압축

270 페이지 "압축"에는 NetFile 서비스 작업 압축 속성이 나와 있습니다.

표 A–13 NetFile 서비스 작업 - 압축 속성

속성	기본값	설명
기본 압축 유형	Zip	Zip 또는 Gzip 압축 유형을 지정합니다.
기본 압축 수준	6	1과 9 사이에서 압축 수준을 지정합니다.

일반

271 페이지 “일반”에는 Netfile 서비스 일반 속성이 나와 있습니다.

표 A–14 NetFile 서비스 일반 속성

속성	기본값	설명
MIME 유형 구성 파일 위치 지정	/opt/S1PS62/SUNWportal/samples /config/netfile	클라이언트 브라우저로 보낼 응답 컨텐트 유형을 지정합니다.

Netlet 서비스

271 페이지 “Netlet 서비스”에는 Netlet 서비스 속성이 나와 있습니다.

표 A–15 Netlet 서비스 속성

속성	기본값	설명
Netlet 규칙		규칙의 추가 또는 삭제를 선택합니다.
규칙을 추가하면 다음 9가지 속성이 필요합니다.		
--규칙 이름		규칙에 대한 고유 이름을 지정합니다.
--암호화 비밀번호		필요한 암호를 지정합니다.
--URL		호출될 응용 프로그램에 대한 URL을 지정합니다.
--애플릿 다운로드		애플릿을 다운로드해야 하는지를 지정합니다. 애플릿이 사용되는 경우, 관련 편집 상자의 구문은 다음과 같습니다. local-port:server-host:server-port

표 A–15 Netlet 서비스 속성 *(계속)*

속성	기본값	설명
--세션 확장		이 규칙에 해당하는 Netlet 세션이 실행되는 동안에 Portal Server 세션 시간이 연장되도록 합니다.
--로컬 포트를 대상 서버 포트에 매핑		로컬 포트, 대상 호스트 및 대상 포트를 지정합니다. 이 값을 입력한 후(이 표의 다음 3개 행에) [추가]를 눌러 목록에 나타나도록 합니다.
--로컬 포트		Netlet이 수신할 로컬 포트를 지정합니다. FTP 규칙의 경우 로컬 포트 값이 30021이어야 합니다.
--대상 호스트		정적 규칙에는 Netlet 연결 대상 컴퓨터의 호스트 이름이 포함됩니다. 동적 규칙에는 단어 "TARGET"이 포함됩니다.
--대상 포트		대상 호스트의 포트를 지정합니다.
기본 원시 VM 암호		Netlet 규칙의 기본 암호를 지정합니다. 규칙의 일부로 암호를 포함하지 않은 기존 규칙을 사용할 때 유용합니다.
기본 Java 플러그인 암호		Netlet 규칙의 기본 암호를 지정합니다. 규칙의 일부로 암호를 포함하지 않은 기존 규칙을 사용할 때 유용합니다.
기본 루프백 포트	58000	Netlet을 통해 애플릿을 다운로드할 때 클라이언트에서 사용할 포트를 지정합니다. Netlet 규칙에서 기본값을 무시할 수 있습니다.
연결 재인증		Netlet 연결을 구성해야 할 때마다 사용자가 Netlet 비밀번호를 입력하도록 합니다.
연결에 대해 경고 팝업 표시	선택	사용자가 Netlet에서 응용 프로그램 실행할 때, 그리고 침입자가 수신 포트를 통해 데스크탑에 액세스하려고 할 때 메시지가 표시됩니다.
포트 경고 대화 상자에 확인란 표시	선택	Netlet에서 사용자의 표준 포털 데스크탑에 있는 대상 호스트에 연결하려 할 때 경고 대화 상자 팝업을 억제하는 옵션을 사용자에게 제공합니다.
연결 유지 시간(분)	0	클라이언트에서 웹 프록시를 통해 게이트웨이에 연결하는 경우 유휴 Netlet 연결은 프록시 시간 초과로 인해 해제됩니다. 이를 방지하려면 이 매개 변수에 프록시 시간 초과 값보다 작은 값을 지정합니다.

표 A–15 Netlet 서비스 속성 (계속)

속성	기본값	설명
포털 로그아웃할 때 Netlet 종료	선택	사용자가 Portal Server를 로그아웃할 때 모든 연결이 종료되도록 합니다.
Netlet 규칙에 액세스	*	특정 조직, 역할 또는 사용자의 특정 Netlet 규칙에 대한 액세스를 정의합니다.
Netlet 규칙 거부		특정 조직, 역할 또는 사용자의 특정 Netlet 규칙에 대한 액세스를 정의합니다.
허용된 호스트	*	특정 조직, 역할 또는 사용자의 특정 호스트에 대한 액세스를 정의합니다.
거부된 호스트		조직 내에서 특정 호스트에 대한 액세스를 거부합니다.

Proxylet 서비스

273 페이지 “Proxylet 서비스”에는 Proxylet 서비스 속성이 나와 있습니다.

표 A–16 Proxylet 서비스 속성

속성	기본값	설명
Proxylet 애플릿을 자동으로 다운로드		이 확인란을 선택하면, 사용자가 로그인할 때 클라이언트 컴퓨터에 Proxylet이 다운로드됩니다.
기본 Proxylet 애플릿 바인드 IP	127.0.0.1	Proxylet 애플릿이 있는 IP 주소
기본 Proxylet 애플릿 포트	58081	Proxylet이 청취하는 포트입니다.

◆ ◆ ◆ 부 록 B

로그 파일

다음 로그 파일은 기본적으로 /var/opt/SUNWportal/debug 디렉토리에 있으며 디버그 및 기타 유형의 정보가 포함되어 있습니다.

로그 파일 정보

표 B–1 정보 및 디버그 파일

파일 이름	내용
다음 로그 파일은 /etc/opt/SUNWam/debug/ 기본 디렉토리의 AMConfig-*instance-name*.properties 파일에 있는 디버그 매개 변수로 제어됩니다. Linux 경로 이름은 "Solaris와 Linux 경로 이름 비교"를 참조하십시오.	
amconsole	Netfile, Netlet 및 게이트웨이 관리 파일
srapNetFile	NetFile 정보 파일
srapNetlet	Netlet 정보 파일
srapProxylet	Proxylet 정보 파일
다음 로그 파일은 /etc/opt/SUNWportal 기본 디렉토리에 있는 platform.conf.gateway-profile-name 파일의 gateway.debug 디버그 매개 변수로 제어됩니다.	
srapGateway.gateway-profile-name	게이트웨이 정보

표 B–1 정보 및 디버그 파일 (계속)

파일 이름	내용
Gateway_to_from_server.gateway-profile-name	
Gateway_to_from_browser.gateway-profile-name	
srapNetletProxy.gateway-profile-name	
srapRewriterProxy.gateway-profile-name	
rwproxy.log.rewriter-proxy-instance-name	Rewriter 프록시의 시작 및 중지 시간
nlproxy.log.netlet-proxy-instance-name	Netlet 프록시의 시작 및 중지 시간
gateway.log.gateway.instance.name	게이트웨이의 시작 및 중지 시간
다음 Rewriter 파일은 `/etc/opt/SUNWam/debug/` 기본 디렉토리의 `AMConfig-`*instance-name*`.properties` 파일에 있는 디버그 매개 변수로 제어됩니다. 자세한 내용은 94 페이지 "디버그 로그를 사용한 문제 해결"을 참조하십시오.	
RuleSetInfo	다시 쓰기에 사용된 모든 규칙 집합이 이 파일에 기록됩니다.
Original Pages	페이지 URI, 확인된 URI(확인된 URI가 페이지 URI와 다른 경우), 컨텐트 MIME, 페이지에 적용된 규칙 집합, 구문 분석기 MIMIE 및 원본 컨텐트가 들어 있습니다. 구문 분석과 관련된 특정 오류/경고/메시지도 이 파일에 들어 있습니다. 메시지 모드에서는 전체 컨텐트가 기록되고 경고와 오류 모드에서는 재작성 중에 발생한 예외만 기록됩니다.
Rewritten Pages	페이지 URI, 확인된 URI(확인된 URI가 페이지 URI와 다른 경우), 컨텐트 MIME, 페이지에 적용된 규칙 집합, 구문 분석기 MIMIE 및 다시 작성된 컨텐트가 들어 있습니다. 이 파일은 디버그 모드가 메시지로 설정되었을 때 채워집니다.
Unaffected Pages	수정되지 않은 페이지 목록을 포함합니다.

표 B–1 정보 및 디버그 파일 *(계속)*

파일 이름	내용
URIInfo Pages	이 파일에는 발견되어 변환된 URL이 들어 있습니다. 컨텐트가 원본 데이터와 동일하게 유지되는 모든 페이지의 세부 사항이 이 파일에 기록됩니다. 세부적으로 기록되는 내용: 페이지 URI, MIME 및 인코딩 데이터, 재작성에 사용된 rulesetID 그리고 구문 분석기 MIME

◆ ◆ ◆ 부 록 C

국가 코드

다음 표에는인증서 관리 중에 지정해야 하는 2자로 된 국가 코드의 목록이 정리되어 있습니다.

국가 코드 목록

표 C–1 2자로 된 국가 코드

ad	안도라
ae	아랍에미리트
af	아프가니스탄
ag	앤티가 바부다
ai	앙귈라
al	알바니아
am	아르메니아
an	네덜란드령 안틸레스
ao	앙골라
aq	남극
ar	아르헨티나
arpa	구식 아르파네트
as	미국령 사모아
at	오스트리아

표 C–1 2자로 된 국가 코드 *(계속)*

au	오스트레일리아
aw	아루바
az	아제르바이젠
ba	보스니아 헤르체고비나
bb	바베이도스
bd	방글라데시
be	벨기에
bf	부르키나파소
bg	불가리아
bh	바레인
bi	부룬디
bj	베냉
bm	버뮤다
bn	브루나이
bo	볼리비아
br	브라질
bs	바하마
bt	부탄
bv	부베이 섬
bw	보츠와나
by	벨로루시
bz	벨리즈
ca	캐나다
cc	코코스 군도
cf	중앙 아프리카
cd	콩고 민주 공화국
cg	콩고
ch	스위스
ci	코트디부와르

표 C–1 2자로 된 국가 코드 *(계속)*

ck	쿠크 군도
cl	칠레
cm	카메룬
cn	중국
co	콜롬비아
com	상용
cr	코스타리카
cs	구 체코슬로바키아
cu	쿠바
cv	카보베르데
cx	크리스마스 섬
cy	사이프러스
cz	체코
de	독일
dj	지부티
dk	덴마크
dm	도미니카
do	도미니카 공화국
dz	알제리
ec	에쿠아도르
edu	교육적
ee	에스토니아
eg	이집트
eh	서사하라
er	에리트레아
es	스페인
et	에디오피아
fi	핀란드
fj	피지

표 C–1 2자로 된 국가 코드 *(계속)*

fk	포클랜드
fm	마이크로네시아
fo	페로 군도
fr	프랑스
fx	프랑스(유럽 영토)
ga	가봉
gb	영국
gd	그레나다
ge	그루지야
gf	프랑스령 가이아나
gh	가나
gi	지브랄타
gl	그린랜드
gm	감비아
gn	기니
gov	미 정부
gp	과달루프(프랑스령)
gq	적도 기니
gr	그리스
gs	그루지야및샌드위치 제도
gt	과테말라
gu	괌(미국령)
gw	기니비사우
gy	가이아나
hk	홍콩
hm	허드 섬 및 맥도널드 군도
hn	온두라스
hr	크로아티아
ht	아이티

표 C–1 2자로 된 국가 코드 *(계속)*

hu	헝가리
id	인도네시아
ie	아일랜드
il	이스라엘
in	인도
int	국제
io	영인도 제도
iq	이라크
ir	이란
일 때	아이슬란드
it	이탈리아
jm	자메이카
jo	요르단
jp	일본
ke	케냐
kg	키르기스스탄
kh	캄보디아
ki	키리바시
km	코모로
kn	세인트 크리스토퍼 네비스
kp	북한
kr	대한민국
kw	쿠웨이트
ky	카이만 군도
kz	카자흐스탄
la	라오스
lb	레바논
lc	세인트 루시아
li	리히텐슈타인

표 C–1 2자로 된 국가 코드 *(계속)*

lk	스리랑카
lr	라이베리아
ls	레소토
lt	리투아니아
lu	룩셈부르크
lv	라트비아
ly	리비아
ma	모로코
mc	모나코
md	몰다비아
mg	마다가스카르
mh	마셜 군도
mil	미군
mk	마케도니아
ml	말리
mm	미얀마
mn	몽골
mo	마카오
mp	북마리아나 군도
mq	말티니크(프랑스령)
mr	모리타니
ms	몬트세라트
mt	몰타
mu	모리셔스
mv	몰디브
mw	말라위
mx	멕시코
my	말레이시아
mz	모잠비크

표 C–1 2자로 된 국가 코드 *(계속)*

na	나미비아
nato	NATO(이 항목은 1996년 삭제되었음 -hq.nato.int 참조)
nc	뉴 칼레도니아(프랑스령)
ne	니제르
net	네트워크
nf	노퍽 섬
ng	나이지리아
ni	니카라과
nl	네덜란드
no	노르웨이
np	네팔
nr	나우루
nt	중립 지역
nu	니우에
nz	뉴질랜드
om	오멘
org	비영리 조직(sic)
pa	파나마
pe	페루
pf	폴리네시아(프랑스령)
pg	파푸아뉴기니
ph	필리핀
pk	파키스탄
pl	폴란드
pm	세인트 피에르 미켈론
pn	핏케언 군도
pr	푸에르토리코
pt	포르투갈
pw	팔라우

표 C–1 2자로 된 국가 코드 (계속)

py	파라과이
qa	카타르
re	리유니언(프랑스령)
ro	루마니아
ru	러시아
rw	르완다
sa	사우디아라비아
sb	솔로몬 군도
sc	세이셸
sd	수단
se	스웨덴
sg	싱가포르
sh	세인트 헬레나
si	슬로베니아
sj	스발바르드 얀마이엔 군도
sk	슬로바키아
sl	시에라리온
sm	산마리노
sn	세네갈
so	소말리아
sr	수리남
st	상투메 프린시페
su	구 USSR
sv	엘살바도르
sy	시리아
sz	스와질랜드
tc	터크스 카이코스 군도
td	차드
tf	프랑스 남부 지방

표 C–1 2자로 된 국가 코드 *(계속)*

tg	토고
th	태국
tj	타지키스탄
tk	토켈라우
tm	투르크메니스탄
tn	튀니지
to	통가
tp	동티모르
tr	터키
tt	트리니다드 토바고
tv	투발루
tw	대만
tz	탄자니아
ua	우크라이나
ug	우간다
uk	영국
um	미국령 군도
us	미국
uy	우루과이
uz	우즈베키스탄
va	바티칸 시국
vc	세인트 빈센트 그레나딘스
ve	베네수엘라
vg	버진 군도(영국령)
vi	버진 군도(미국령)
vn	베트남
vu	바누아투
wf	월리스 푸투나
ws	사모아

표 C–1 2자로 된 국가 코드 *(계속)*

ye	예멘
yt	마요트
yu	유고슬라비아
za	남아프리카
zm	잠비아
zr	자이르
zw	짐바브웨

색인

O

P

R

호

활