



# Java Desktop System Configuration Manager 릴리스 1.1 설치 안내서

---

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

부품 번호: 819-0961-10  
2005년 2월

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 모든 권리는 저작권자의 소유입니다.

이 제품 또는 문서는 저작권에 의해 보호되고 사용권에 따라 사용, 복사, 배포 및 디컴파일은 제한됩니다. 이 제품이나 문서의 어떤 부분도 Sun 및 그 라이선스 허여자의 사전 서면 승인 없이 어떤 형태로든 어떤 수단을 통해서든 복제해서는 안 됩니다. 글꼴 기술을 포함한 타사 소프트웨어에 대한 저작권 및 사용권은 Sun 공급업체에 있습니다.

제품 중에는 캘리포니아 대학으로부터 사용권을 받은 Berkeley BSD 시스템에서 파생된 부분이 포함되어 있을 수 있습니다. UNIX는 미국 및 다른 국가에서 X/Open Company, Ltd.를 통해 독점적으로 사용권이 부여되는 등록 상표입니다.

Sun, Sun Microsystems, Sun 로고, docs.sun.com, AnswerBook, AnswerBook2, 및 Solaris는 미국 및 다른 국가에서 Sun Microsystems, Inc.의 상표 또는 등록 상표입니다. 모든 SPARC 상표는 사용 허가를 받았으며 미국 및 다른 국가에서 SPARC International, Inc.의 상표 또는 등록 상표입니다. SPARC 상표를 사용하는 제품은 Sun Microsystems, Inc.가 개발한 구조를 기반으로 하고 있습니다.

Sun Microsystems, Inc.는 사용자 및 사용 허가자를 위해 OPEN LOOK 및 Sun™ GUI(그래픽 사용자 인터페이스)를 개발했습니다. Sun은 컴퓨터 업계를 위한 시각적 그래픽 사용자 인터페이스의 개념을 연구 개발한 Xerox사의 선구적인 노력을 높이 평가하고 있습니다. Sun은 Xerox 및 Xerox 그래픽 사용자 인터페이스(GUI)의 비독점적 라이선스를 가지며, 이 라이선스는 OPEN LOOK GUI를 구현하고 Sun의 서면 라이선스 계약을 준수하는 Sun 정식 사용자에게 적용됩니다.

미국 정부의 권리 - 상용 소프트웨어정부 기관 사용자는 Sun Microsystems, Inc.의 표준 사용권 조항과 FAR 및 그 부록의 해당 규정을 준수해야 합니다.

본 설명서는 "있는 그대로" 제공되며 상업성, 특정 목적에 대한 적합성, 비침해성에 대한 모든 암시적 보증을 포함하여 모든 명시적 또는 묵시적 조건과 표현 및 보증에 대해 책임을 지지 않습니다. 이러한 보증 부인은 법적으로 허용된 범위 내에서만 적용됩니다.



050105@10536



# 목차

---

<b>1</b>	<b>소개</b>	<b>9</b>
	소개	9
<b>2</b>	<b>LDAP 서버</b>	<b>11</b>
	개념	11
	설치	12
	배포 도구	12
	스키마 확장	12
	조직 매핑	12
	사용자 프로필 매핑	14
	배포	15
	추가 고려 사항	16
<b>3</b>	<b>Sun Web Console</b>	<b>17</b>
	시스템 요구 사항	17
	클라이언트	17
	서버	18
	Sun Web Console 설치	18
	▼ Sun Web Console 설치 방법	18
	콘솔 실행	19
	Sun Web Console 제거	20
	Sun Web Console 포트 정보	20
<b>4</b>	<b>Java Desktop System Configuration Manager, 릴리스 1.1</b>	<b>21</b>
	Configuration Manager 설치	21

▼ Configuration Manager 설치 방법	21
Configuration Manager 실행	22
▼ Configuration Manager 시작 방법	22
Configuration Manager 제거	23
<b>5 Linux 및 Solaris에서 데스크탑 구성 요소 설치</b>	<b>25</b>
Configuration Agent	25
부트스트랩 정보	26
포트 설정	28
변경 감지 간격	28
작동 설정	29
에이전트 설정 적용	31
데이터 액세스/사용자 인증	31
GConf 어댑터	32
Java Preferences 어댑터	32
Mozilla 어댑터	33
StarSuite 어댑터	33
<b>6 Microsoft Windows에서 데스크탑 구성 요소 설치</b>	<b>35</b>
Configuration Agent 설치	35
시작	36
사용권 계약	36
설치 유형	37
설치 준비	44
설치 요약	45
Configuration Agent 설정 변경	46
Configuration Agent 제거	46
<b>A Sun Web Console 패키지</b>	<b>47</b>
알려진 문제	47
보안	47
설치 스크립트 사용	47
Sun Web Console 패키지	48
Solaris 패키지	48
Linux RPM	48

<b>B</b>	<b>Configuration Manager 패키지</b>	<b>51</b>
	Configuration Manager 패키지	51
	Solaris 패키지	51
	Linux RPM	51
<b>C</b>	<b>OpenLDAP 및 Active Directory를 Configuration Manager와 함께 사용</b>	<b>53</b>
	OpenLDAP 서버를 Configuration Manager와 함께 사용	53
	Active Directory 서버를 Configuration Manager와 함께 사용	54



## 그림

---

그림 5-1	Configuration Agent, 구성 저장소	26
그림 5-2	Configuration Agent, 인증 체계	27
그림 5-3	Configuration Agent, 포트 설정	28
그림 5-4	Configuration Agent, 데이터 디렉토리	29
그림 5-5	Configuration Agent, 요청 처리 및 로깅	29
그림 5-6	Configuration Agent, 요약 페이지	31
그림 6-1	Configuration Agent, 시작 페이지	36
그림 6-2	Configuration Agent, 소프트웨어 사용권 계약	36
그림 6-3	Configuration Agent, 설치 유형 페이지	37
그림 6-4	Configuration Agent, 구성 저장소	38
그림 6-5	Configuration Agent, 인증 체계	39
그림 6-6	Configuration Agent, 포트 설정	41
그림 6-7	Configuration Agent, 데이터 디렉토리	42
그림 6-8	Configuration Agent, 요청 처리 및 로깅	42
그림 6-9	Configuration Agent, 요약 페이지	44
그림 6-10	Configuration Agent, 설치 준비 페이지	44
그림 6-11	Configuration Agent, 설치 요약 페이지	45





## 소개

---

Java™ Desktop System Configuration Manager, 릴리스 1.1에 대해 간단히 소개합니다.

---

## 소개

Java™ Desktop System Configuration Manager, 릴리스 1.1은 Java™ Desktop System을 실행하는 데스크탑 호스트를 중앙 집중식으로 구성하기 위한 제품입니다. 이 제품을 사용하면 조직 구조의 다양한 요소에 설정을 지정할 수 있으므로 관리자가 사용자 또는 호스트 그룹을 편리하게 관리할 수 있습니다. 제품의 주요 구성 요소는 다음과 같습니다.

- 관리할 사용자 및 호스트의 조직 구조가 포함되어 있으며 구성 데이터가 보관되는 LDAP 서버
- 관리자가 구성 데이터를 정의하여 조직 구조의 각 요소에 지정할 수 있도록 도와주는 웹 기반 관리 도구
- 클라이언트 호스트에 설치되어 현재 로그인한 사용자 대신 구성 데이터를 검색하여 Java Desktop System을 구성하는 다양한 응용 프로그램에 제공하는 데스크탑 구성 요소

관리 도구는 Sun Web Console 내에서 실행되는 웹 기반 응용 프로그램입니다. 관리자는 이 도구를 사용하여 LDAP 서버의 조직 구조를 탐색하고 각 요소에 정책을 지정할 수 있습니다. 정책은 관리 도구에서 처리할 설정을 정의하는 정책 템플릿에 따라 표시 및 편집됩니다.

데스크탑 구성 요소는 사용자 대신 LDAP 서버로부터 구성 데이터를 검색하여 다수의 구성 시스템 어댑터에 제공하는 Java™ Desktop System Configuration Agent를 중심으로 구성됩니다. 구성 시스템 어댑터는 정책 설정을 사용하여 로컬 구성(응용 프로그램에서 제공하는 기본 설정과 사용자 설정)을 보완합니다. 현재 지원되는 구성 시스템에는 Gnome 데스크탑이나 Evolution과 같은 Gnome 응용 프로그램의 구성을 처리하는 GConf, Mozilla™ Preferences, StarRegistry (StarSuite 구성 시스템) 등이 있습니다.



# LDAP 서버

---

이 장에서는 Configuration Manager와 함께 사용하는 LDAP 서버 설정 및 배포에 대한 정보를 제공합니다.

## 개념

Java Desktop System Configuration Manager 프레임워크 내에서, 구성 데이터는 엔티티와 연결되어 있습니다. 여기서 엔티티는 LDAP 저장소의 항목이며 회사의 조직 구조 요소에 해당합니다.

인식되는 엔티티는 다음과 같습니다.

- 조직: 일반적으로 전체 계층 구조의 조직(사업부, 그룹, 팀) 또는 지역(대륙, 국가, 사이트) 단위를 나타냅니다.
- 사용자: 전체 계층 구조의 리프 노드를 나타내며 이름에서 알 수 있듯이 대체로 사용자가 됩니다.
- 도메인: 네트워크 조직의 논리적 구조 단위를 나타냅니다.
- 호스트: 역시 전체 계층 구조의 리프 노드를 나타내지만 네트워크상의 개별 컴퓨터를 가리킵니다.
- 역할: 일반적으로 기능(관리자, 사이트 관리) 면에서 구별되며 특정 사용자 집합에 적용되는 등록 정보를 나타냅니다.

조직 및 사용자 엔티티는 사용자 트리를 정의하는 데 사용되고 도메인 및 호스트 엔티티는 호스트 트리를 정의합니다. 두 트리는 독립적이지만 프레임워크에서 유사한 방식으로 처리됩니다.

조직 및 도메인 엔티티와 다른 항목 간의 관계는 저장소에서 항목의 물리적 위치에 의해 정의됩니다. 즉, 트리에서 두 엔티티 아래에 위치하는 항목은 모두 조직과 도메인 엔티티에 포함될 수 있습니다. 역할과 사용자 또는 호스트 간의 관계는 사용자 및 호스트 항목의 속성에 의해 정의됩니다.

엔티티에 연결된 구성 데이터는 프레임워크에서 관리하는 특수 항목에 저장됩니다. 특수 항목은 각 항목에 연결된 서비스 이름과 서비스 컨테이너로 식별됩니다.

---

## 설치

기존 LDAP 서버를 Configuration Manager와 함께 사용하려면 다음 작업을 수행해야 합니다.

- Configuration Manager에서 구성 데이터를 저장하는 데 사용하는 사용자 정의 개체 클래스와 속성을 지원하도록 서버 스키마를 확장합니다.
- 저장소에 있는 항목과 Configuration Manager에서 지원하는 엔티티의 매핑 정보를 사용자 정의하여 서버에 저장합니다.

## 배포 도구

기존 LDAP 서버를 Configuration Manager와 함께 사용하려면 설치 CD에 있는 다음 배포 도구가 필요합니다.

- 88apoc-registry.ldif: 구성 데이터를 저장하는 데 필요한 개체 클래스와 속성을 소개하는 스키마 파일
- OrganizationMapping: LDAP 항목과 Configuration Manager 엔티티 간의 매핑을 설명하는 기본 등록 정보 파일
- UserProfileMapping: LDAP 사용자 항목 속성과 Configuration Manager 사용자 프로필 속성 간의 매핑을 설명하는 기본 등록 정보 파일
- createServiceTree: LDAP 저장소에 매핑 파일을 저장하는 스크립트
- deployApoc: LDAP 서버의 스키마를 확장하고 LDAP 저장소에 매핑 파일을 저장하는 스크립트

## 스키마 확장

구성 데이터는 해당 항목에 연결된 항목 트리에 저장됩니다. 각 트리에서 사용되는 개체 클래스와 속성을 LDAP 서버에 저장하려면 먼저 개체 클래스와 클래스를 LDAP 서버 스키마에 추가해야 합니다. 예를 들어, 제공된 스키마 확장 파일은 LDIF 형식을 사용하여 개체와 클래스를 Sun Java™ System Directory Server에 추가합니다. 다른 LDAP 서버에 개체와 클래스를 추가하려면 해당 서버에서 인식하는 형식을 사용해야 합니다.

## 조직 매핑

LDAP 항목과 Configuration Manager 엔티티 간의 매핑을 정의하려면 Organization 매핑 파일을 편집해야 합니다. LDAP 저장소의 레이아웃에 맞는 값을 각 키에 입력해야 합니다.

사용자 엔티티는 모든 엔티티가 사용하는 개체 클래스와 전체 저장소에서 고유한 값을 갖는 속성으로 식별됩니다. 관리 응용 프로그램에서의 사용자 표시에 영향을 주는 표시 이름 형식을 제공할 수 있으며, 조직 내의 사용자 항목이 컨테이너 항목을 사용할 경우 컨테이너 항목을 선택적으로 정의할 수도 있습니다. 키 이름과 해당 기본값은 다음과 같습니다.

```
# Object class that all user entries use
User/ObjectClass=inetorgperson
# Attribute whose value in user entries is unique within the repository
User/UniqueIdAttribute=uid
# Optional container in organization entries of the user entries,
# remove line if not used
User/Container=ou=People
# Display name format within the management application
User/DisplayNameFormat=sn, givenname
```

역할 엔티티는 사용할 수 있는 개체 클래스 목록과 해당 이름 지정 속성으로 식별됩니다. 각 목록의 항목은 <item1>, <item2>, ..., <itemN> 형식으로 정렬되어야 합니다. 즉, 동일한 수의 항목이 목록에 있어야 하고 n번째 개체 클래스에는 n번째 이름 지정 속성을 사용해야 합니다. 두 키는 역할과 사용자 간의 관계는 물론 역할과 호스트 간의 관계도 정의합니다. *VirtualMemberAttribute* 키는 사용자 또는 호스트 항목에서 쿼리할 수 있는 값을 가진 속성을 지정해야 하며 항목이 속해 있는 역할의 전체 DN을 포함해야 합니다. *MemberAttribute* 키는 검색 필터에 사용할 사용자 또는 호스트 항목의 속성을 지정해야 합니다. 사용자 또는 호스트가 속해 있는 역할의 전체 DN을 포함해야 합니다. *VirtualMemberAttribute* 키에는 서비스 클래스 가상 속성을 사용할 수 있지만 *MemberAttribute* 키는 필터에 사용할 수 있는 물리적 속성이어야 합니다. 키 이름과 해당 기본값은 다음과 같습니다.

```
# List of object classes for roles
Role/ObjectClass=nsRoleDefinition
# Aligned list of corresponding naming attributes
Role/NamingAttribute=cn
# Physical attribute (usable in a filter) containing the DNS
# of the roles of a user/host
Role/MemberAttribute=nsRoleDN
# Attribute whose query on a user or host return the DNS of the
# roles it belongs to
Role/VirtualMemberAttribute=nsRole
```

조직 엔티티는 정렬된 개체 클래스 목록과 해당 이름 지정 속성 목록을 사용하여 역할과 유사한 방식으로 식별됩니다. 키 이름과 해당 기본값은 다음과 같습니다.

```
# List of object classes for organizations
Organization/ObjectClass=organization
# Aligned list of corresponding naming attributes
Organization/NamingAttribute=o
```

도메인 엔티티는 조직 엔티티와 유사한 방식으로 식별됩니다. 키 이름과 해당 기본값은 다음과 같습니다.

```
# List of object classes for domains
Domain/ObjectClass=ipNetwork
# Aligned list of corresponding naming attributes
```

```
Domain/NamingAttribute=cn
```

호스트 엔티티는 사용자 엔티티와 유사한 방식으로 식별됩니다. 키 이름과 해당 기본값은 다음과 같습니다.

```
# Object class that all host entries use
Host/ObjectClass=ipHost
# Attribute whose value in host entries is unique within the repository
Host/UniqueIdAttribute=cn
# Optional container in domain entries of the host entries,
# remove line if not used
Host/Container=ou=Hosts
```

## 사용자 프로필 매핑

LDAP 사용자 항목 속성과 Configuration Manager 사용자 항목 속성 간의 매핑을 정의하려면 사용자 프로필 매핑 파일을 편집해야 합니다. 각 키는 Configuration Manager의 사용자 속성을 나타내며 조직 매핑으로 식별되는 사용자 항목 속성의 이름 값으로 지정할 수 있습니다. User/DisplayNameFormat 설정에 사용되는 속성은 *User Profile* 매핑에서 지정해야 합니다. 키 이름과 해당 기본값은 다음과 같습니다.

```
# inetOrgPerson.givenName
org.openoffice.UserProfile/Data/givenname = givenname
# person.sn
org.openoffice.UserProfile/Data/sn = sn
# inetOrgPerson.initials
org.openoffice.UserProfile/Data/initials = initials
# organizationalPerson.street
org.openoffice.UserProfile/Data/street = street,postalAddress,streetAddress
# organizationalPerson.l (city)
org.openoffice.UserProfile/Data/l = l
# organizationalPerson.st (state)
org.openoffice.UserProfile/Data/st = st
# organizationalPerson.postalCode
org.openoffice.UserProfile/Data/postalcode = postalcode
# country.c (country)
org.openoffice.UserProfile/Data/c =
# organizationalPerson.o (company)
org.openoffice.UserProfile/Data/o = o,organizationName
# deprecated -- no LDAP corollary
org.openoffice.UserProfile/Data/position =
# organizationalPerson.title
org.openoffice.UserProfile/Data/title = title
# inetOrgPerson.homePhone
org.openoffice.UserProfile/Data/homephone = homephone
# organizationalPerson.telephoneNumber
org.openoffice.UserProfile/Data/telephonenumber = telephonenumber
# organizationalPerson.facsimileTelephoneNumber
org.openoffice.UserProfile/Data/facsimiletelephonenumber =
facsimiletelephonenumber,officeFax
# inetOrgPerson.mail
org.openoffice.UserProfile/Data/mail = mail
```

## 배포

LDAP 저장소의 상태를 반영하여 매핑 파일의 사용자 정의가 끝나면 파일을 배포할 수 있습니다. 필요한 개체 클래스와 속성이 LDAP 서버 스키마에 이미 포함되어 있으면 createServiceTree 스크립트를 직접 실행할 수 있지만, 그렇지 않은 경우에는 deployApoc 스크립트를 실행해야 합니다.

deployApoc 스크립트는 Sun Java™ System Directory Server에서 사용하도록 작성되었으며 제공된 스키마 확장 파일을 올바른 디렉토리에 복사하고 LDAP 서버를 순환시킨 다음 createServiceTree 스크립트를 호출합니다. 스키마 저장소에 있는 파일을 복사하고 서버를 다시 시작할 수 있는 권한을 가진 사용자가 다음 명령을 실행하여 이 스크립트를 호출해야 합니다.

```
./deployApoc <Directory Server Directory>
```

<Directory Server Directory> 매개 변수는 디렉토리 서버 설치의 slapd-*<server name>* 하위 디렉토리 경로여야 합니다. 설치 시 기본 디렉토리를 사용하고 서버 이름을 *myserver.mydomain*으로 지정한 경우 이 디렉토리는 */var/Sun/mps/slapd-myserver.mydomain*이 됩니다.

직접 호출했던 deployApoc 스크립트에서 호출했던 관계 없이 createServiceTree 스크립트는 사용자에게 LDAP 서버 위치(호스트 이름, 포트 번호 및 기본 DN)와 관리 권한이 있는 사용자의 정의(전체 DN, 암호)를 표시합니다. 그런 다음 이 스크립트는 LDAP 서버에 부트스트랩 서비스 트리를 만들어 매핑 파일을 저장합니다. 누구든지 다음 명령을 실행하여 스크립트를 실행할 수 있습니다.

```
./createServiceTree
```

이 스크립트는 사용자에게 다음 정보를 확인합니다.

- **호스트 이름**(기본값: **localhost**): LDAP 서버의 호스트 이름
- **포트 번호**(기본값: **389**): LDAP 서버의 포트 번호
- **기본 DN**: LDAP 저장소의 기본 DN
- **사용자 DN**(기본값: **cn=Directory Manager**): 기본 DN에 새 항목을 만들 수 있는 권한을 가진 사용자의 전체 DN
- **암호**: 해당 사용자 암호

다음 DN을 가진 항목이 만들어집니다.

```
ou=ApocRegistry,ou=default,ou=OrganizationConfig,ou=1.0,  
ou=ApocService,ou=services, <baseDN>
```

이 항목에는 두 매핑 파일의 내용이 채워집니다.

이미 설명한 것처럼 deployApoc 스크립트는 설치 디렉토리, 레이아웃 및 스키마 확장 절차가 Sun Java System Directory Server와 거의 일치하는 LDAP 서버를 가정하여 작업을 수행합니다. 다른 디렉토리를 사용하려면 createServiceTree 스크립트를 실행하기 전에 스키마를 수동으로 확장해야 합니다. OpenLDAP 및 ActiveDirectory 사용과 관련된 자세한 내용은 **부록 C**를 참조하십시오.

이렇게 만들어진 트리는 엔티티에 연결된 구성 데이터를 보관하는 트리와 일치하며 Sun Java System Identity Server에서 서비스 관리에 사용되는 트리 구조와 연결됩니다.

---

## 추가 고려 사항

Configuration Manager 프레임워크에서는 데스크탑에서 지정한 사용자 또는 호스트 식별자에 연결된 전체 DN을 식별하기 위해 읽기 및 검색 권한이 있는 LDAP 서버 연결을 구성할 수 있어야 합니다. 따라서 익명 연결을 허용하도록 저장소를 구성하거나 읽기 및 검색 액세스 권한이 있는 특수 사용자를 만들어야 합니다.

관리 응용 프로그램은 엔티티로 매핑되는 항목 아래에 해당 엔티티의 구성 데이터를 보관할 서비스 트리를 만듭니다. 이 때문에 관리 목적으로 사용되는 사용자 항목에는 관리하는 항목에 하위 항목을 만들 수 있는 권한이 있어야 합니다.

프레임워크에서 데스크탑 클라이언트 사용자는 Anonymous 및 GSSAPI의 두 가지 방법으로 인증될 수 있습니다. Anonymous 방법을 사용할 경우 데스크탑 클라이언트가 LDAP 서버에서 데이터를 검색할 때 자격 증명을 제공하지 않으므로 전체 저장소에서 읽기 및 검색에 대한 익명 액세스를 허용해야 합니다. GSSAPI 방법(인증에 커버로스 사용)을 사용하려면, *Sun Java™ System Directory Server 5 2004Q2 관리 안내서*의 "인증 및 암호화 관리" 장에 설명된 대로 LDAP 서버를 구성해야 합니다.



# Sun™ Web Console

---

Sun Web Console은 Sun Microsystems의 공통 웹 기반 관리 솔루션으로 설계되었습니다. 사용자가 이 콘솔을 통해 액세스하는 시스템 관리 응용 프로그램은 모두 동일한 사용자 인터페이스를 제공합니다.

다양한 이유로 해서 콘솔은 웹 모델을 기반으로 설계되었지만 주된 이유는 시스템 관리자가 웹 브라우저를 사용하여 시스템 관리 응용 프로그램에 액세스할 수 있도록 하기 위한 것입니다.

Sun Web Console에서 제공하는 기능은 다음과 같습니다.

- 공통 인증 및 권한 부여
- 공통 로깅
- 동일한 HTTPS 기반 포트를 통해 모든 시스템 관리 응용 프로그램에 액세스할 수 있는 단일 진입점
- 공통된 모양 및 색감

콘솔의 가장 큰 이점은 관리자가 한 번만 로그인하면 콘솔 내에서 모든 응용 프로그램을 사용할 수 있다는 것입니다.

---

## 시스템 요구 사항

Sun Web Console은 다양한 클라이언트 및 서버 운영 체제와 여러 브라우저를 지원합니다.

### 클라이언트

- Solaris™ 8 이상에서 실행되는 Netscape™ 6.2x 및 7.x
- Windows 98, 98 SE, ME, 2000 및 XP에서 실행되는 Netscape 4.7x, 6.2x 및 7.x

- Windows 98, 98 SE, ME, 2000 및 XP에서 실행되는 Internet Explorer 5.x 및 6.x
- Linux 및 Solaris에서 실행되는 Mozilla

## 서버

- Solaris 8 이상
- Red Hat 8 이상, Red Hat Enterprise Linux 2.1
- SuSE Linux 2.1 이상
- J2SE™ Version 1.4.1\_03 이상  
서버에서 J2SE 1.4.1 이전 버전이 검색되면 설치 프로그램에서 Java Desktop System Management Tools CD에 있는 J2SE 버전을 사용하여 설치를 업그레이드하라는 메시지를 표시합니다.
- Tomcat: 4.0.3 이상  
Tomcat은 Java Desktop System Management Tools CD에 포함되어 있습니다.

---

## Sun Web Console 설치

Sun Web Console을 설치하기 전에 먼저 본 설명서의 부록 A에서 패키지 요약과 알려진 문제점을 읽어 보십시오.

Java Desktop System Management Tools CD에서는 Solaris SPARC (버전 8 이상) 및 Linux 운영 체제용 Sun Web Console의 설치 바이너리를 사용할 수 있습니다.

### ▼ Sun Web Console 설치 방법

- 단계
1. Java Desktop System Management Tools CD에서, 콘솔을 설치하려는 운영 체제가 있는 Sun Web Console 디렉토리로 변경합니다("swc").
  2. `./setup`을 입력합니다.  
기본적으로 Sun Web Console은 설치 로그 파일을 만들지 않습니다. "logfile"이라는 설치 로그를 만들려면 `./setup | tee logfile`을 입력합니다.

---

주 - setup을 실행하면 웹 콘솔 설치 및 구성 작업이 대부분 자동으로 수행됩니다. Sun Web Console용 설치 응용 프로그램에 대한 자세한 내용은 부록 A를 참조하십시오.

---

3. Sun Web Console을 현지화하려면 각 언어를 위한 두 개의 추가 패키지를 설치해야 합니다. 아래 표에서 해당 언어를 위한 패키지 이름을 확인하고 다음 중 하나를 수행합니다.

- Solaris를 사용하는 경우 `pkgadd -d path/pkgname.pkg pkgname`을 입력합니다. 여기서 *pkgname*은 추가할 언어 패키지의 이름입니다.
- Linux를 사용하는 경우 `rpm -i path/pkgname<...>.rpm`을 입력합니다. 여기서 *pkgname*은 추가할 패키지의 이름입니다.

패키지 이름	설명
SUNWcmcon, SUNWcmctg	중국어 간체 Sun™ Web Console 2.0
SUNWdmcon, SUNWdmctg	독일어 Sun™ Web Console 2.0
SUNWemcon, SUNWemctg	스페인어 Sun™ Web Console
SUNWfmcon, SUNWfmctg	프랑스어 Sun™ Web Console 2.0
SUNWhmcon, SUNWhmctg	중국어 번체 Sun™ Web Console 2.0
SUNWimcon, SUNWimctg	이탈리아어 Sun™ Web Console 2.0
SUNWjmcon, SUNWjmctg	일본어 Sun™ Web Console 2.0
SUNWkmcon, SUNWkmctg	한국어 Sun™ Web Console 2.0
SUNWsmcon, SUNWsmctg	스웨덴어 Sun™ Web Console 2.0

## 콘솔 실행

일반적으로 Sun Web Console 서버를 중지했다가 다시 시작하기만 하면 새 응용 프로그램을 등록할 수 있습니다.



**주의** - 처음 Sun Web Console을 시작하는 경우 먼저 Configuration Manager 설치가 완료되었는지 확인합니다. Sun Web Console은 콘솔에 최소한 한 개의 응용 프로그램을 배포하기 전에는 제대로 실행되지 **않습니다**.

- Sun Web Console을 시작하려면 `smcwebserver start`를 입력합니다.
- Sun Web Console을 중지하려면 `smcwebserver stop`을 입력합니다.
- Sun Web Console에 액세스하려면 브라우저에 다음 URL을 입력합니다.  
`https://<hostname>.<domainname>:6789`

Sun Web Console은 기본적으로 Unix 기반 인증과 역할 기반 액세스 제어(RBAC)를 지원합니다. 그러나 LDAP 인증과 같은 다른 인증 체계를 구성할 수도 있습니다.

---

주 - 기본 세션 시간 제한은 15분입니다. `smreg` 명령을 사용하여 시간 제한을 구성할 수 있습니다. 예를 들어, 시간 제한을 5분으로 설정하려면 `smreg add -p -c session.timeout.value=5`를 입력합니다.

---

Sun Web Console 명령에 대한 자세한 내용은 `smcwebserver` 및 `smreg` 설명서 페이지를 참조하십시오.

---

## Sun Web Console 제거

Sun Web Console을 제거하려면 `/usr/lib/webconsole/setup -u`를 실행합니다.

---

주 - `/usr/lib/webconsole` 디렉토리나 관련 하위 디렉토리에서는 이 명령을 실행하지 마십시오. 그렇지 않으면 `pkgrm`이 실패합니다.

---

---

## Sun Web Console 포트 정보

Configuration Manager는 다음 Sun Web Console 포트를 사용합니다.

- 8005: 서비스 종료
- 6789: https 액세스

이 두 포트는 `/etc/opt/webconsole/server.xml`에서 변경할 수 있습니다. 포트를 변경한 다음 `/usr/sbin/smcwebserver restart`를 사용하여 Sun Web Console을 다시 시작합니다.

# Java™ Desktop System Configuration Manager, 릴리스 1.1

---

Configuration Manager는 Sun Web Console에서 실행되는 관리 도구를 제공합니다. 관리자는 웹 기반 사용자 인터페이스를 통해 조직 계층 구조를 이동하면서 데스크탑 응용 프로그램에 대한 정책을 정의할 수 있습니다. 계층 구조에서 조직, 역할, 사용자, 도메인, 호스트 등 각 항목에 대해 정책을 정의할 수 있습니다. Configuration Manager는 다양한 구성 템플릿을 사용하여 Gnome, Mozilla, StarSuite, Evolution 등의 데스크탑 응용 프로그램별 설정을 표시합니다.

---

## Configuration Manager 설치

Configuration Manager를 설치하려면 Sun Web Console이 이미 설치되어 작동하고 있어야 합니다.

### ▼ Configuration Manager 설치 방법

- 단계
1. Java Desktop System Management Tools CD에서 해당 Configuration Manager 디렉토리("apoc")로 변경합니다.
  2. `./setup`을 입력합니다.
  3. LDAP 서버의 호스트 이름을 입력합니다.  
기본 이름은 `localhost`입니다.
  4. LDAP 서버의 포트 번호를 입력합니다(기본값: 389).
  5. LDAP 저장소의 기본 DN을 입력합니다.
  6. 사용자 엔티티를 식별하는 데 사용되는 개체 클래스 이름을 입력합니다. 기본 개체 클래스는 `inetorgperson`입니다.

자세한 내용은 LDAP 서버 장의 12 페이지 “조직 매핑” 절을 참조하십시오.

7. 전체 LDAP 저장소에서 고유한 속성 이름을 입력합니다. 기본 속성은 uid입니다.  
자세한 내용은 LDAP 서버 장에서 12 페이지 “조직 매핑” 절을 참조하십시오.
8. LDAP 서버에서 쿼리를 수행해야 하는 사용자의 전체 DN을 입력합니다.  
읽기 및 검색 액세스 권한을 가진 임의의 전체 DN을 사용합니다. 익명 액세스를 허용하려면 이 필드를 비워 두십시오.
9. LDAP 액세스 권한이 지정된 사용자 암호를 입력합니다.  
LDAP 서버에 대해 익명 액세스를 설정한 경우에는 이 단계를 무시합니다.  
LDAP를 통해 사용자를 인증할 수 있도록 설치 도중 다른 로그인 모듈이 Sun Web Console에 추가됩니다.  
설치가 끝나면 Sun Web Console이 자동으로 다시 시작되어 Configuration Manager에 액세스할 수 있습니다.

---

주 - 언제든지 /usr/share/webconsole/apoc/configure 스크립트를 사용하여 이전 Configuration Manager 설정을 수정할 수 있습니다. 예를 들어, Configuration Manager를 다시 설치하지 않아도 이 스크립트를 사용하여 다른 LDAP 서버로 변경할 수 있습니다.

---

## Configuration Manager 실행

### ▼ Configuration Manager 시작 방법

- 단계
1. 브라우저에 다음 URL을 입력합니다. `https://<hostname>.<domainname>:6789`
  2. 프롬프트에 기존 LDAP 사용자의 사용자 이름(uid)과 암호를 입력합니다.  
Sun Web Console이 열립니다.
  3. 콘솔 창에서 Sun Java Desktop System Configuration Manager, Release 1.1을 클릭합니다.

#### 자세한 정보 Configuration Manager에 직접 액세스

Sun Web Console 시작 페이지를 건너뛰고 바로 Configuration Manager로 이동하려면 브라우저에 다음 URL을 입력합니다.

`https://<hostname>.<domainname>:6789/apoc`

---

## Configuration Manager 제거

Configuration Manager를 Sun Web Console에서 제거하려면 Java Desktop System Management Tools CD에서 해당 Configuration Manager 디렉토리로 변경한 다음 `./setup -u`를 실행합니다.

---

주 - Configuration Manager를 제거하면 LDAP 로그인 모듈이 Sun Web Console에서 제거됩니다.

---





## 5장

---

# Linux 및 Solaris™에서 데스크탑 구성 요소 설치

---

이 장에서는 Solaris 및 Linux 운영 체제별로 해당되는 정보를 제공합니다.

Configuration Manager에서 구성 데이터에 액세스하려면 데스크탑 클라이언트에 Java™ Desktop System Configuration Agent가 설치되어 있어야 합니다. Configuration Agent는 원격 구성 데이터 저장소 및 어댑터와 통신하고 데이터를 특정 구성 시스템에 통합합니다. 현재 지원되는 구성 시스템은 GConf, Java Preferences, Mozilla Preferences 및 StarSuite Registry입니다.

---

## Configuration Agent

Configuration Agent는 다음 표에 열거된 여러 패키지 중 하나입니다.

Solaris 패키지 이름	Linux RPM 이름	설명
SUNWapbas	apoc-base	Configuration Shared 라이브러리
SUNWapmsc	apoc-misc	Configuration Agent 기타 파일
SUNWapoc	apoc	Configuration Agent
SUNWapdc	apoc-config	Configuration Agent 마법사

이 패키지를 설치할 때 이 API에 필요한 파일이 설치됩니다. 패키지는 수동으로 설치하거나 Java Desktop System 설치를 통해 설치할 수 있습니다. 설치한 후에는 시스템에서 Configuration Agent를 구성하여 활성화해야 합니다.

원격 구성 데이터에 액세스하려면 Configuration Agent에 LDAP 서버의 호스트 이름, 포트 등 최소한의 부트스트랩 정보를 제공해야 합니다. 이 정보는 `policymgr.properties`, `apocd.properties` 및 `os.properties` 같은 등록 정보 파일로 유지 관리됩니다. 이 파일은 `/etc/apoc` 디렉토리에 로컬로 저장되어 있습니다. 이러한 등록 파일을 수동으로 편집하거나 Configuration Agent의 구성 마법사를 사용할 수 있습니다.

구성 마법사는 Configuration Agent의 필요한 설정을 안내하는 그래픽 사용자 인터페이스를 제공합니다. 각 마법사 페이지에서는 해당 도움말 화면을 사용할 수 있습니다. `/usr/bin/apoc-config` 스크립트를 사용하여 슈퍼 유저(`root`)로 마법사를 시작할 수 있습니다. 기본 설정/ 시스템 도구/네트워크 설정, 또는 Nautilus 파일 관리자의 시스템-설정:///네트워크 설정에서도 해당 데스크탑 메뉴 항목을 사용할 수 있습니다.

---

주 - 그래픽 인터페이스를 시작하지 않고 마법사를 시작할 수도 있습니다. 예를 들어, 콘솔 모드로 마법사를 시작하려면 `/usr/bin/apoc-config -nodisplay`를 실행합니다.

---

## 부트스트랩 정보

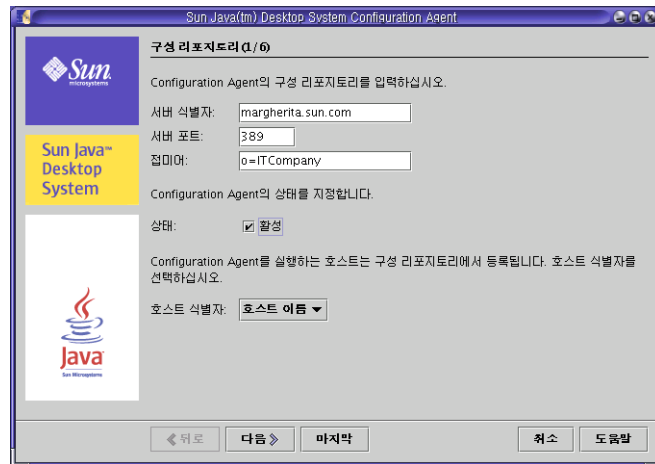


그림 5-1 Configuration Agent, 구성 저장소

---

주 - 해당되는 경우 연결된 등록 정보 파일 키가 괄호 안에 표시되어 있습니다.

---

- 서버 식별자(Server): LDAP 서버의 호스트 이름
- 서버 포트(Port): LDAP 서버의 포트 번호

- **접미어(BaseDn):** LDAP 저장소의 기본 DN
- **상태:** Configuration Agent의 상태. 이 확인란을 사용하여 Configuration Agent를 활성화하거나 비활성화할 수 있습니다. 구성 저장소를 사용하려면 Configuration Agent를 활성화해야 합니다. 활성화하면 inetd에 자동으로 등록됩니다.

---

주 - Configuration Agent를 수동으로 활성화 또는 비활성화하려면 **root**로 로그인하여 각각 /usr/lib/apocd enable 또는 /usr/lib/apocd disable 명령을 입력합니다.

---

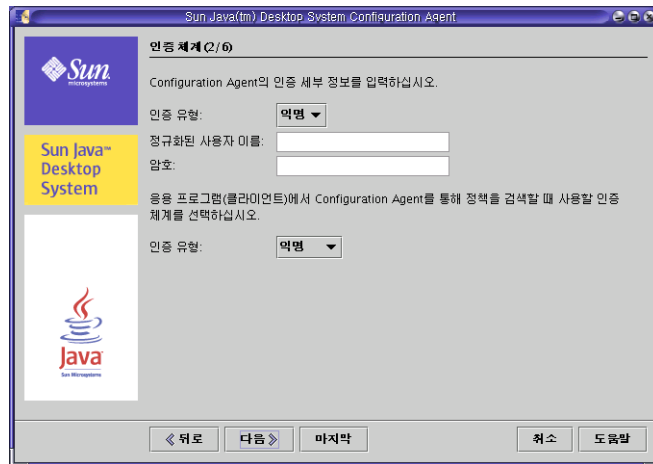


그림 5-2 Configuration Agent, 인증 체계

- **호스트 식별자(HostIdentifier):** "HostName" 또는 "IPAddress" 중 하나일 수 있습니다. 호스트 식별자는 호스트를 식별하는 데 사용되는 LDAP 속성의 내용과 일치하도록 설정해야 합니다. 이 속성은 매핑 파일에서 Host/UniqueIdAttribute로 정의됩니다.
- Configuration Agent의 **인증 유형:** "익명" 또는 "단순" 중 하나일 수 있습니다. "익명"을 선택하면 **정규화된 사용자 이름** 및 **암호** 필드는 자동으로 사용할 수 없게 됩니다.
- **정규화된 사용자 이름(AuthDn):** 저장소에 대한 읽기 및 검색 액세스 권한을 가진 사용자의 전체 DN
- **암호>Password):** 등록된 LDAP 사용자 암호

---

주 - 디렉토리에서 익명 액세스가 활성화되면 **정규화된 사용자 이름** 및 **암호** 설정을 비워둘 수 있습니다.

---

- 응용 프로그램의 **인증 유형(AuthType):** LDAP 서버가 사용자를 인증하는 방법에 따라, 이 옵션은 "익명" 또는 "GSSAPI"가 될 수 있습니다.

---

주 - 자세한 내용은 31 페이지 “데이터 액세스/사용자 인증”을 참고하십시오.

---

## 포트 설정

Configuration Agent는 다음 두 포트를 사용합니다.

- **에이전트 포트(DaemonPort)**: 에이전트가 클라이언트 응용 프로그램과 통신하는 데 사용합니다(기본값: **38900**).
- **관리 포트 (DaemonAdminPort)**: 에이전트 제어 프로그램인 apocd가 에이전트와 통신할 때 사용합니다(기본값: **38901**).

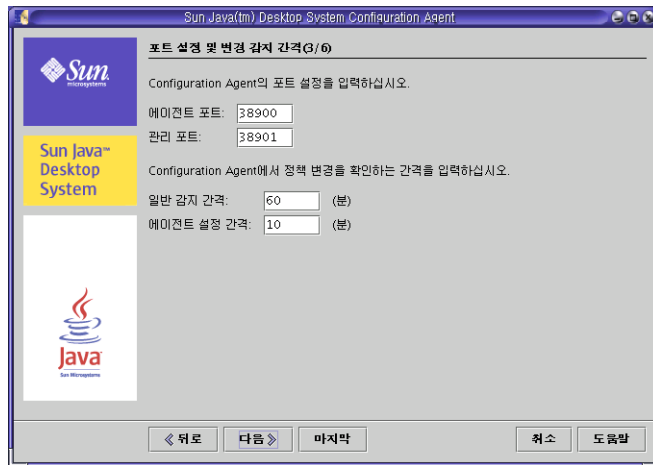


그림 5-3 Configuration Agent, 포트 설정

## 변경 감지 간격

Configuration Agent는 다음 두 가지 간격을 사용하여 구성 데이터의 변경 사항을 정기적으로 확인합니다.

- **일반 감지 간격(ChangeDetectionInterval)**: 데스크탑 응용 프로그램(클라이언트)의 구성 데이터에 대한 변경 감지 주기 간격(분)

---

주 - -1로 지정하면 변경 감지가 수행되지 않습니다.

---

- **에이전트 설정 간격(DaemonChangeDetectionInterval)**: 에이전트 특정 구성 설정에 대한 변경 감지 주기 간격(분)

---

주 - -1로 지정하면 변경 감지가 수행되지 않습니다.

---

일반 감지 간격을 사용하여 원하는 대로 원격 구성 데이터 변경 사항을 클라이언트측 응용 프로그램으로 전파할 수 있습니다. 이 설정에 지정되는 값은 원격 변경 사항이 클라이언트 응용 프로그램에서 적용될 때까지 최대 지연 시간(분)입니다.

작은 값을 지정하면 Configuration Agent와 LDAP 서버 작업이 증가하므로 설정 값을 변경할 때는 주의해야 합니다. 예를 들어, 초기 배포 단계에서 클라이언트 응용 프로그램에 대한 원격 구성의 영향을 쉽게 테스트하기 위해 이 값을 1분으로 설정했다가 테스트가 끝나면 다시 이 설정을 원래 값으로 되돌릴 수 있습니다.

## 작동 설정

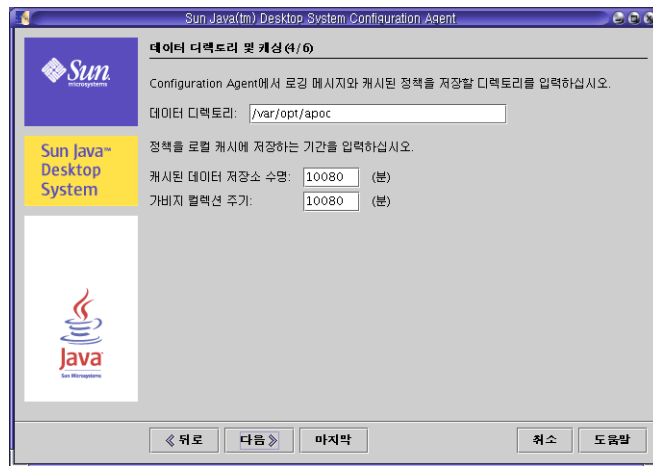


그림 5-4 Configuration Agent, 데이터 디렉토리

다음 설정을 구성할 수 있습니다.

- **데이터 디렉토리(DataDir):** 런타임 데이터를 저장하는 데 사용되는 디렉토리. 기본값은 `/var/opt/apoc`입니다.
- **캐시된 데이터 저장소 수명(TimeToLive):** 오프라인 상태가 아닌 구성 데이터가 로컬 데이터베이스에 보관되는 시간(분)

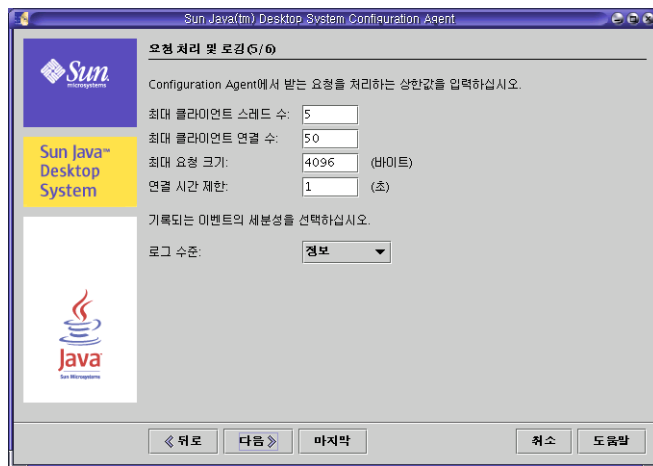


그림 5-5 Configuration Agent, 요청 처리 및 로깅

- **가비지 컬렉션 주기(GarbageCollectionInterval):** 로컬 구성 데이터베이스의 가비지 컬렉션 주기 간격(분)
- **최대 클라이언트 스레드(MaxClientThreads):** 동시에 처리될 수 있는 최대 클라이언트 요청 수
- **최대 클라이언트 연결(MaxClientConnections):** 최대 클라이언트 연결 수
- **최대 요청 크기(MaxRequestSize):** 최대 클라이언트 요청 크기
- **연결 시간 제한(ConnectTimeout):** LDAP 서버가 연결 요청에 응답할 수 있는 간격. 기본값은 1초입니다.
- **로그 수준(LogLevel):** 에이전트 로그 파일에 기록되는 세부 정보 수준. 로깅 수준은 Java Logger 수준과 같아야 합니다. 로깅 수준을 가장 높은 것부터 순서대로 나열하면
  - SEVERE
  - WARNING
  - INFO
  - CONFIG
  - FINE
  - FINER
  - FINEST

---

주 - 데이터 디렉토리 설정과 연결 시간 제한 설정을 제외한 대부분의 작동 설정은 LDAP 서버에 저장된 해당 정책을 통해 중앙 집중식으로 유지 관리할 수도 있습니다. 이 기능을 사용하려면 마법사를 사용하여 해당 설정을 적용하지 마십시오. 대신 Configuration Manager 내의 Configuration Agent 정책을 사용하여 작동 설정을 중앙 집중식으로 지정하십시오.

---

## 에이전트 설정 적용

"데이터 디렉토리" 및 "연결 시간 제한" 설정 외에 Configuration Manager를 사용하여 LDAP 서버에 저장된 작동 설정은 다음 에이전트 구성 변경 감지 주기에 자동으로 적용됩니다(DaemonChangeDetectionInterval 참조).

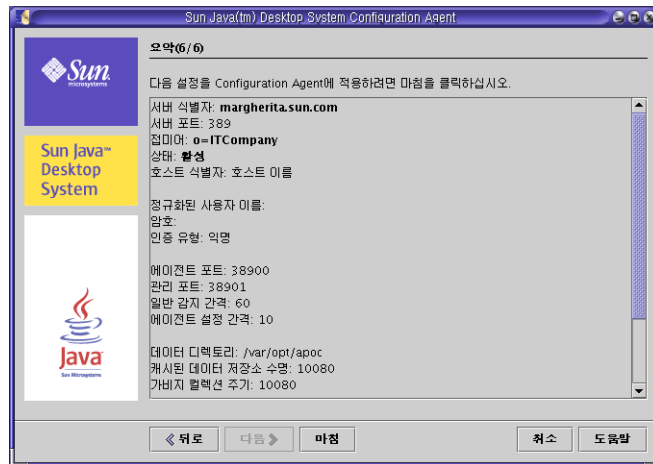


그림 5-6 Configuration Agent, 요약 페이지

로컬에서 변경된 다른 모든 설정의 경우에는 Configuration Agent를 다시 로드하거나 다시 시작해야 합니다. 구성 마법사를 사용하는 경우 다시 로드하거나 다시 시작하는 작업은 자동으로 수행됩니다.

---

주 – Configuration Agent를 수동으로 다시 시작하려면 관련된 클라이언트 응용 프로그램이 실행되고 있지 않은지 확인하고 루트로 로그인하여 `/usr/lib/apoc/apocd restart` 명령을 입력합니다.

---

## 데이터 액세스/사용자 인증

Configuration Agent는 데스크탑 사용자의 로그인 ID에 따라 LDAP 서버로부터 정보를 검색합니다. 조직 매핑 파일의 User/UniqueIdAttribute 설정은 로그인 ID를 LDAP 서버의 사용자 엔티티로 매핑합니다. Configuration Agent는 호스트 이름이나 IP 주소와 같은 호스트 관련 정보도 검색합니다. 이 정보는 조직 매핑 파일의 Host/UniqueIdAttribute 설정을 통해 LDAP 서버의 호스트 엔티티로 매핑됩니다.

LDAP 서버에 액세스하는 데는 익명과 GSSAPI의 두 가지 방법이 있습니다. 익명 액세스를 사용하는 경우 데스크탑에서 별도의 작업을 수행할 필요가 없습니다. GSSAPI 방법을 사용하려면 데스크탑에서 커버로스 자격 증명을 획득해야 합니다. 커버로스 자격 증명 획득과 사용자 로그인을 통합하려면 Java Desktop System 호스트에서 pam\_krb5 모듈을 설치 및 구성해야 합니다.

gdm을 사용하여 커버로스와 사용자 로그인을 통합할 수도 있습니다. 예를 들어 다음 /etc/pam.d/gdm 파일을 사용합니다.

```
#%PAM-1.0
auth required pam_unix2.so nullok #set_secure
auth optional pam_krb5.so use_first_pass missing_keytab_ok ccache=SAFE putenv_direct
account required pam_unix2.so
password required pam_unix2.so #strict=false
session required pam_unix2.so # trace or none
session required pam_devperm.so
session optional pam_console.so
```

이런 방식으로 커버로스와 사용자 로그인을 통합하는 경우에는 화면 보호기의 커버로스 지원을 활성화해야 합니다. 예를 들어 다음 /etc/pam.d/xscreensaver 파일을 사용합니다.

```
auth required pamkrb5.so use_first_pass missing_keytab_ok
ccache=SAFE putenv_direct
```

---

## GConf 어댑터

GConf 어댑터는 Solaris용 SUNWapoc-adapter-gconf 패키지와 Linux용 apoc-adapter-gconf RPM의 일부입니다. 해당 패키지나 RPM에서 어댑터를 설치할 때 /etc/gconf/2/path에 있는 GConf 데이터 소스 경로가 Configuration Manager 소스를 포함하도록 업데이트됩니다. 어댑터는 다음 두 가지 데이터 원본을 제공합니다.

- "apoc:readonly": 보호되지 않은 정책 설정에 대한 액세스를 제공합니다. 이 데이터 원본은 사용자 설정 뒤, 로컬 기본값 앞에 삽입합니다.
- "apoc:readonly:mandatory@": 보호된 정책 설정에 대한 액세스를 제공합니다. 이 데이터 원본은 로컬 필수 설정 뒤, 사용자 설정 앞에 삽입합니다.

---

## Java Preferences 어댑터

Java Preferences 어댑터는 Solaris용 SUNWapcj 패키지와 Linux용 apoc-adapter-java RPM의 일부입니다. 해당 패키지나 RPM에서 어댑터를 설치할 때 Solaris의 경우에는 /opt/SUNWapcj 디렉토리에 Linux의 경우에는 /opt/apocjava에 필요한 파일이 추가됩니다.



---

## Mozilla 어댑터

Mozilla 어댑터는 Solaris용 SUNWmozapoc-adapter 패키지와 Linux용 mozilla-apoc-integration RPM의 일부입니다. 해당 패키지나 RPM에서 어댑터를 설치할 때 필요한 파일이 기존 Mozilla 설치에 추가되고 자동으로 등록됩니다.

---

## StarSuite 어댑터

StarSuite 어댑터는 표준 StarSuite 설치에 포함되어 있으며 특별한 수정 없이 정책 구성 데이터에 액세스할 수 있도록 지원합니다.



# Microsoft Windows에서 데스크탑 구성 요소 설치

---

중앙 구성 기능에 필요한 모든 구성 요소가 제공되고 Java Desktop System의 일부로 설치됩니다. 보통 다른 시스템에서 중앙 구성 관리를 사용할 수 있습니다. 그러나 이런 경우에는 반드시 필수 설치 단계를 수행해야 합니다. Microsoft Windows 시스템에서 간편하게 설치할 수 있도록 하기 위해 GUI 기반 설치 프로그램이 제공되어 설치 단계를 안내합니다.

---

## Configuration Agent 설치

주 - Configuration Agent와 마찬가지로, 설치 프로그램을 사용하려면 Java™ Runtime Environment (JRE) 1.4 이상이 필요합니다. JRE는 설치 단계에 속하지 않습니다. 설치된 Java 버전이 확실치 않은 경우에는 `java -version`을 실행하십시오. 이 명령을 사용하면 올바른 Java 버전이 사용 가능하고 사용자 시스템에 제대로 구성되어 있는지 알 수 있습니다.

설치 프로그램은 배치 파일인 `setup.bat`에 의해 시작됩니다. 이 파일은 Java Desktop System management tools CD에 있습니다. 충분한 관리 권한이 있는 사용자로 스크립트를 실행해야 합니다(**local administrator**). 그렇지 않으면 새 시스템 서비스인 Configuration Agent의 설치와 시작에 실패합니다.

다음을 입력하여 설치 프로그램을 시작합니다.

```
setup[-help]
      [-saveState filename][-no]
      [-noconsole][-state filename]
      [-nodisplay]
```

설치 프로그램에서 인식하는 옵션은 다음과 같습니다.

- help            도움말 텍스트를 표시합니다.
- saveState      대화식 설치 세션에서 입력한 내용을 지정된 파일에 기록합니다. 기본 파일 이름은 statefile.out입니다.
- state           지정된 상태 파일을 자동 설치의 입력 내용으로 사용합니다. 이 옵션은 -noconsole과 함께 사용합니다.
- noconsole      사용자 인터페이스 없이 설치 프로그램을 시작합니다. -state와 함께 사용하여 자동 설치를 시작합니다.
- nodisplay      GUI를 실행하지 않고 명령줄 모드로 설치 프로그램을 시작합니다.
- no             소프트웨어를 설치하지 않고 설치 프로그램을 실행합니다. 이 옵션은 자동 설치를 준비하기 위해 -saveState와 함께 사용하면 유용합니다.

## 시작

Configuration Agent 설치 마법사가 호출된 후에 시작 페이지가 나타납니다. 시작 페이지는 설치 과정과 관련된 정보를 제공합니다.

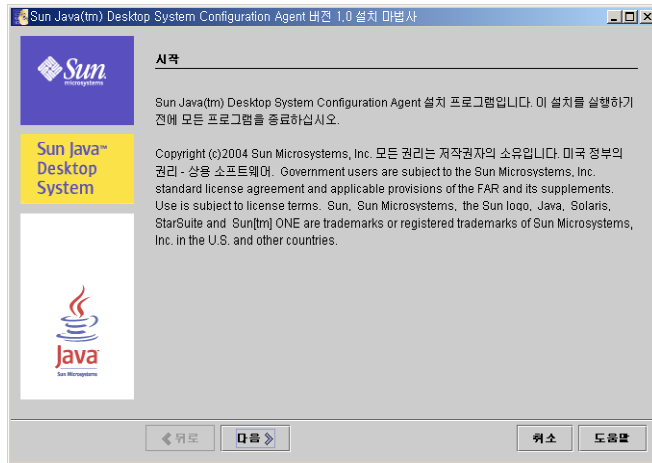


그림 6-1 Configuration Agent, 시작 페이지

다음 버튼을 클릭하여 마법사의 다음 페이지로 진행합니다.

## 사용권 계약

소프트웨어 사용권 계약 페이지에는 Configuration Agent에 대한 라이선스 정보가 있습니다.



그림 6-2 Configuration Agent, 소프트웨어 사용권 계약

사용권 계약을 읽은 다음 예(라이선스 수락) 버튼을 눌러 계약 내용에 동의하고 설치 과정을 계속합니다.

## 설치 유형

설치 프로그램에서는 세 가지 유형의 설치 방법을 제공합니다. 설치 유형에 따라 설치 과정 동안 정의할 수 있는 구성 설정의 수가 달라집니다.

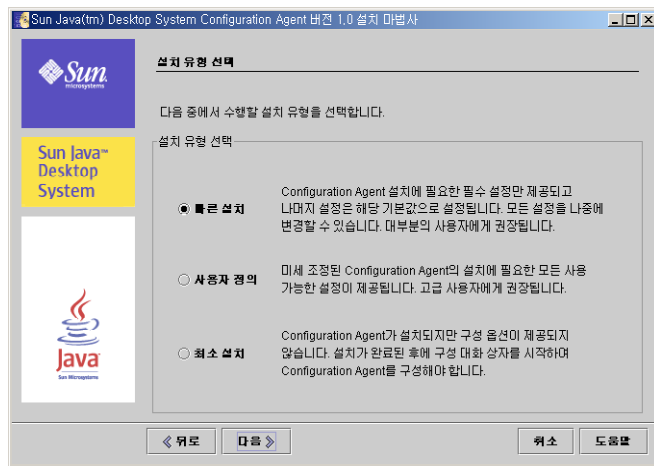


그림 6-3 Configuration Agent, 설치 유형 페이지

- **빠른 설치:** 대부분의 사용자에게 권장되는 설치 유형입니다. 설치 프로그램은 Configuration Agent의 작업 설치에 필요한 필수적인 설정만 표시합니다. 나머지 설정은 적절한 기본값으로 설정됩니다.
- **사용자 정의:** Configuration Agent에 사용할 수 있는 모든 설정이 표시되어 에이전트 동작을 좀 더 세밀하게 조정할 수 있습니다. 이 유형은 고급 사용자만 사용하는 것이 좋습니다.
- **최소 설치:** Configuration Agent가 설치되지만 사용자는 어떤 구성도 입력하거나 설정하지 않습니다. 설치가 완료된 후 구성 마법사를 시작하여 Configuration Agent를 구성해야 합니다.

원하는 설치 유형을 선택한 후 다음을 클릭합니다. 선택한 설치 유형에 따라 여러 구성 설정 페이지가 표시됩니다. 다음 절에서는 이런 설정에 대해서 설명합니다.

---

**주 - 빠른 설치 또는 최소 설치** 옵션을 선택한 경우에는 다음에서 설명하는 구성 페이지 중 일부가 표시되지 않습니다.

---



---

**주 - Configuration Agent**를 시작하려면 LDAP 서버의 호스트 이름과 포트 같은 일부 부트스트랩 정보가 필요합니다. 또한 로깅 수준이나 데이터 디렉토리 같은 여러 고급 설정을 지정할 수도 있습니다. 이 모든 설정은 설치 과정 동안 지정할 수 있습니다. 이 정보는 `policymgr.properties`, `apocd.properties` 및 `os.properties` 같은 등록 정보 파일로 유지 관리됩니다. 이 파일은 `Program Files\Sun\Apoc\lib` 디렉토리에 로컬로 저장됩니다. 이러한 등록 파일을 수동으로 편집하거나 Configuration Agent의 구성 마법사를 사용할 수 있습니다.

---

## Configuration Agent 마법사 및 설정 정보

이 절에서는 Configuration Agent에 사용할 수 있는 모든 구성 설정에 대해 자세히 설명합니다.

구성 마법사는 Configuration Agent의 필요한 설정을 안내하는 그래픽 사용자 인터페이스를 제공합니다. 각 마법사 페이지에서는 해당 도움말 화면을 사용할 수 있습니다.

### 부트스트랩 정보

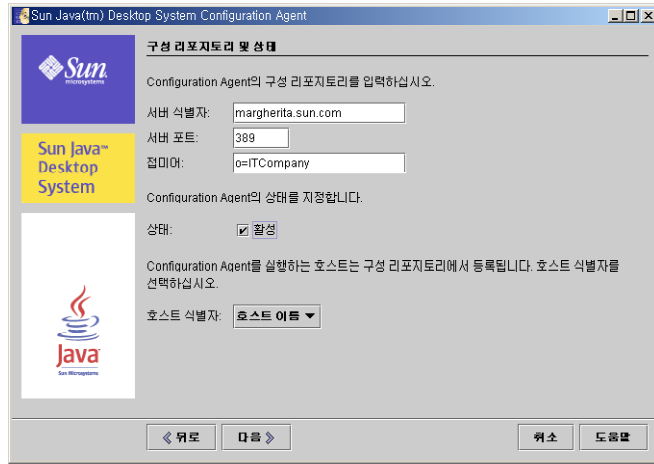


그림 6-4 Configuration Agent, 구성 저장소

---

주 - 해당되는 경우 연결된 등록 정보 파일 키가 괄호 안에 표시되어 있습니다.

---

- 서버 식별자(Server): LDAP 서버의 호스트 이름
- 서버 포트(Port): LDAP 서버의 포트 번호
- 접미어(BaseDn): LDAP 저장소의 기본 DN
- 상태: Configuration Agent의 상태. 이 확인란을 사용하여 Configuration Agent를 활성화하거나 비활성화할 수 있습니다. 구성 저장소를 사용하려면 Configuration Agent를 활성화해야 합니다. 에이전트를 활성화하면 서비스 제어 관리자에 필요한 등록이 자동으로 포함됩니다.

---

주 - Configuration Agent를 수동으로 활성화하거나 비활성화하려면 Administrator로 로그인한 다음 Program Files\Sun\Apoc\bin\apocd enable 또는 Program Files\Sun\Apoc\bin\apocd disable 명령을 입력하십시오.

---

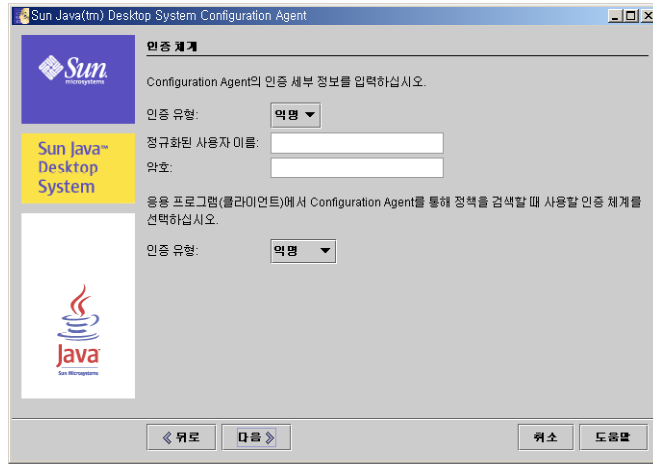


그림 6-5 Configuration Agent, 인증 체계

- **호스트 식별자(HostIdentifier):** "HostName" 또는 "IPAddress" 중 하나일 수 있습니다. 호스트 식별자는 호스트를 식별하는 데 사용되는 LDAP 속성의 내용과 일치하도록 설정해야 합니다. 이 속성은 매핑 파일에서 Host/UniqueIdAttribute로 정의됩니다.
- **Configuration Agent의 인증 유형:** "익명" 또는 "단순" 중 하나일 수 있습니다. "익명"을 선택하면 정규화된 사용자 이름 및 암호 필드는 자동으로 사용할 수 없게 됩니다.
- **정규화된 사용자 이름(AuthDn):** 저장소에 대한 읽기 및 검색 액세스 권한을 가진 사용자의 전체 DN

---

주 - 디렉토리에서 익명 액세스를 허용하려면 이 설정을 비워둘 수 있습니다.

---

- **암호>Password):** 등록된 LDAP 사용자 암호

---

주 - 디렉토리에서 익명 액세스를 허용하려면 이 설정을 비워둘 수 있습니다.

---

- **응용 프로그램의 인증 유형(AuthType):** LDAP 서버가 사용자를 인증하는 방법에 따라, 이 옵션은 "익명" 또는 "GSSAPI"가 될 수 있습니다.

## 포트 설정

Configuration Agent는 다음 두 포트를 사용합니다.

- **에이전트 포트(DaemonPort):** 에이전트가 클라이언트 응용 프로그램과 통신하는 데 사용합니다(기본값: 38900).



- **관리 포트(DaemonAdminPort):** 에이전트 제어 프로그램인 apocd.exe가 에이전트와 통신할 때 사용합니다(기본값: **38901**).

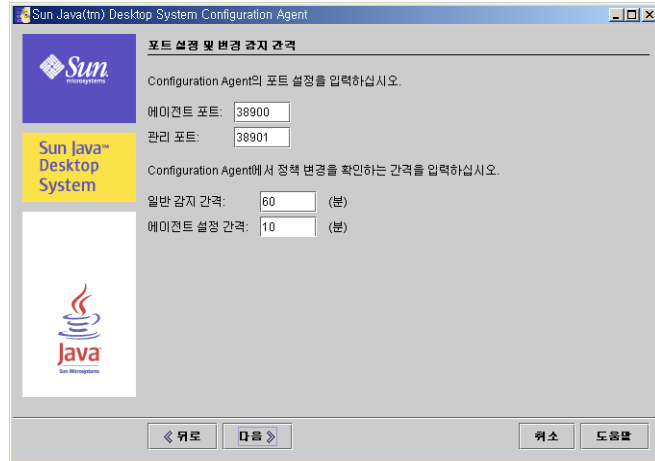


그림 6-6 Configuration Agent, 포트 설정

## 변경 감지 간격

Configuration Agent는 다음 두 가지 간격을 사용하여 구성 데이터의 변경 사항을 정기적으로 확인합니다.

- **일반 감지 간격(ChangeDetectionInterval):** 데스크탑 응용 프로그램(클라이언트)의 구성 데이터에 대한 변경 감지 주기 간격(분)

---

주 - -1로 지정하면 변경 감지가 수행되지 않습니다.

---

- **에이전트 설정 간격(DaemonChangeDetectionInterval):** 에이전트 특정 구성 설정에 대한 변경 감지 주기 간격(분)

---

주 - -1로 지정하면 변경 감지가 수행되지 않습니다.

---

일반 감지 간격을 사용하여 원하는 대로 원격 구성 데이터 변경 사항을 클라이언트측 응용 프로그램으로 전파할 수 있습니다. 이 설정에 지정되는 값은 원격 변경 사항이 클라이언트 응용 프로그램에서 적용될 때까지 최대 지연 시간(분)입니다.

작은 값을 지정하면 Configuration Agent와 LDAP 서버 작업이 증가하므로 설정 값을 변경할 때는 주의해야 합니다. 예를 들어, 초기 배포 단계에서 클라이언트 응용 프로그램에 대한 원격 구성의 영향을 쉽게 테스트하기 위해 이 값을 1분으로 설정했다가 테스트가 끝나면 다시 이 설정을 원래 값으로 되돌릴 수 있습니다.

## 작동 설정

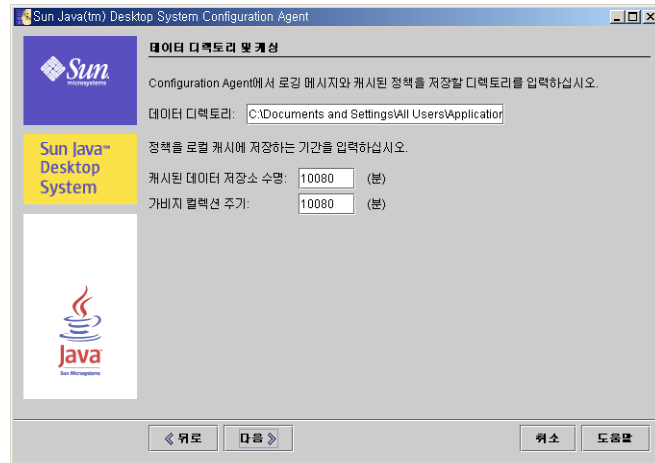


그림 6-7 Configuration Agent, 데이터 디렉토리

다음 설정을 구성할 수 있습니다.

- **데이터 디렉토리(DataDir):** 런타임 데이터를 저장하는 데 사용되는 디렉토리
- **캐시된 데이터 저장소 수명(TimeToLive):** 오프라인 상태가 아닌 구성 데이터가 로컬 데이터베이스에 보관되는 시간(분)

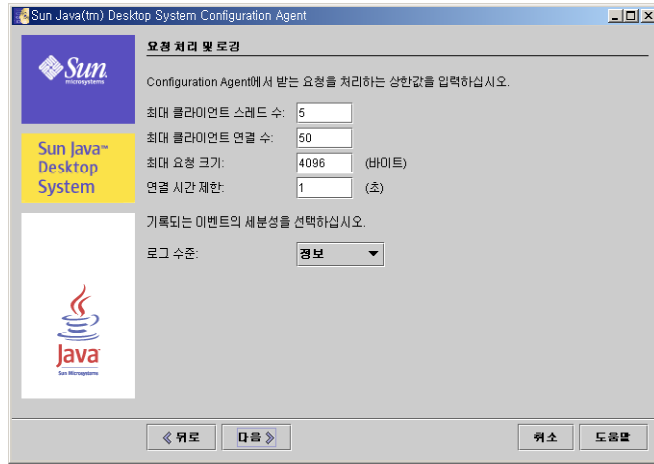


그림 6-8 Configuration Agent, 요청 처리 및 로깅

- **가비지 컬렉션 주기(GarbageCollectionInterval):** 로컬 구성 데이터베이스의 가비지 컬렉션 주기 간격(분)
- **최대 클라이언트 스레드(MaxClientThreads):** 동시에 처리될 수 있는 최대 클라이언트 요청 수
- **최대 클라이언트 연결 수(MaxClientConnections):** 최대 클라이언트 연결 수
- **최대 요청 크기(MaxRequestSize):** 최대 클라이언트 요청 크기
- **연결 시간 제한(ConnectTimeout):** LDAP 서버가 연결 요청에 응답할 수 있는 간격. 기본값은 1초입니다.
- **로그 수준(LogLevel):** 에이전트 로그 파일에 기록되는 세부 정보 수준. 로깅 수준은 Java Logger 수준과 같아야 합니다. 로깅 수준을 가장 높은 것부터 순서대로 나열하면
  - OFF
  - SEVERE
  - WARNING
  - INFO
  - CONFIG
  - FINE
  - FINER
  - FINEST
  - ALL

주 - 데이터 디렉토리 설정과 연결 시간 제한 설정을 제외한 대부분의 작동 설정은 LDAP 서버에 저장된 해당 정책을 통해 중앙 집중식으로 유지 관리할 수도 있습니다. 이 기능을 사용하려면 마법사를 사용하여 해당 설정을 적용하지 마십시오. 대신 Configuration Manager 내의 Configuration Agent 정책을 사용하여 작동 설정을 중앙 집중식으로 지정하십시오.

## 에이전트 설정 요약

"데이터 디렉토리" 및 "연결 시간 제한" 설정 외에 Configuration Manager를 사용하여 LDAP 서버에 저장된 작동 설정은 다음 에이전트 구성 변경 감지 주기에 자동으로 적용됩니다(DaemonChangeDetectionInterval 참조).

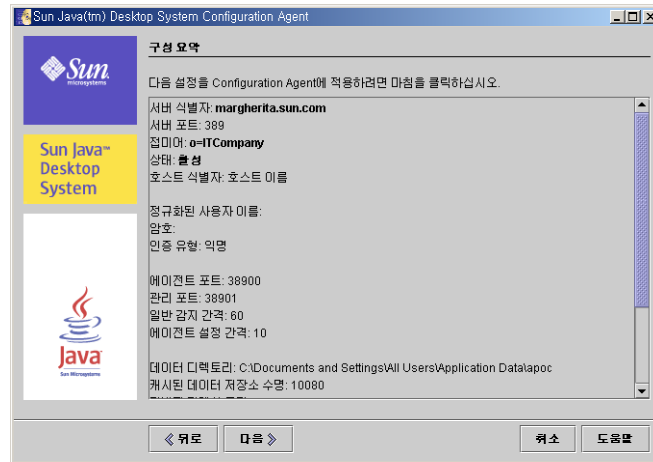


그림 6-9 Configuration Agent, 요약 페이지

로컬에서 변경된 다른 모든 설정의 경우에는 Configuration Agent를 다시 로드하거나 다시 시작해야 합니다. 구성 마법사를 사용하는 경우 다시 로드하거나 다시 시작하는 작업은 자동으로 수행됩니다.

주 - Configuration Agent를 수동으로 다시 시작하려면 실행 중인 관련 클라이언트 응용 프로그램이 없는지 확인하고 Administrator로 로그인한 다음 Program Files\Sun\Apoc\bin\apocd restart 명령을 입력하십시오. 또는, 관리 도구의 제어판에서 "서비스 컨트롤러"를 시작할 수 있습니다. 사용 가능한 서비스 목록에서 Configuration Agent를 선택한 다음 재시작을 클릭하십시오.

## 설치 준비

설치 준비 페이지는 설치할 Configuration Agent 항목을 보여 줍니다.

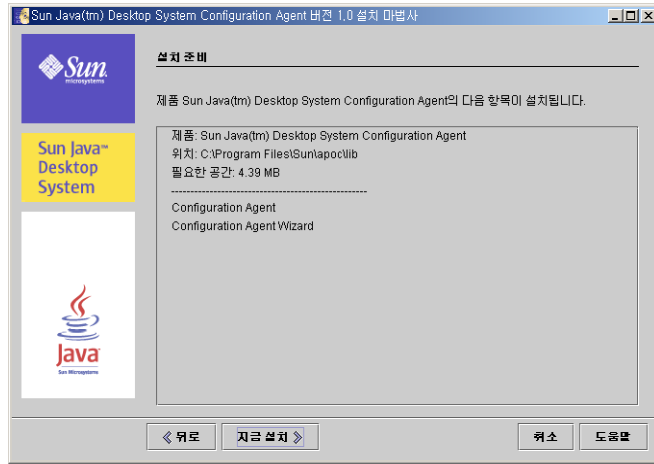


그림 6-10 Configuration Agent, 설치 준비 페이지

이 시점에서는 시스템에 변경된 사항이 없습니다. 지금 설치 버튼을 누르면 실질적인 설치가 시작됩니다.

## 설치 요약

마법사의 설치 요약 페이지는 설치된 항목의 상태를 보여줍니다. 항목에 대한 자세한 내용을 보려면 해당 항목 옆에 있는 자세히 버튼을 클릭하십시오.

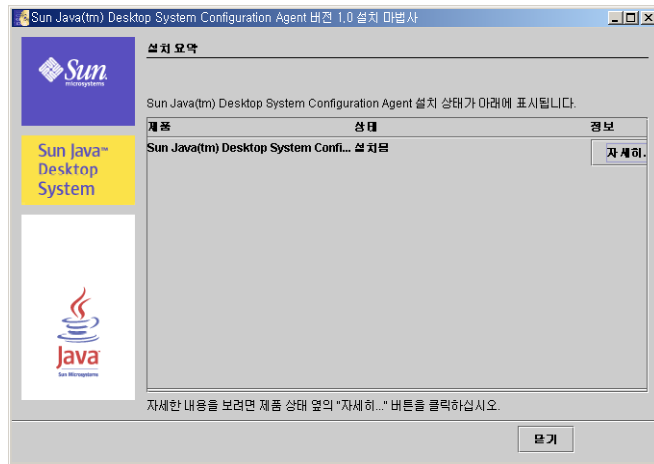


그림 6-11 Configuration Agent, 설치 요약 페이지

설치 프로그램에서는 Configuration Agent 바이너리를 Sun\apoc 아래의 기본 프로그램 파일 디렉토리(보통 C:\Program Files)에 저장합니다. 또한 시작-모든 프로그램-Sun 메뉴에 새 Java Desktop System 폴더를 자동으로 만듭니다. 이 폴더에는 두 개의 바로 가기가 있어서 Configuration Agent를 더 자세하게 구성하거나 제거할 수 있습니다.

---

## Configuration Agent 설정 변경

앞서 언급했듯이 설치 후에도 언제든지 Configuration Agent 설정을 수정할 수 있습니다. Program Files\Sun\Apoc\bin\apoc-config.bat 스크립트를 사용하여 Administrator로 마법사를 시작할 수 있습니다. 또는, 시작 메뉴에서 모든 프로그램-Sun-Java Desktop System 아래에 있는 **Configuration Agent** 항목을 선택하여 Configuration Agent 마법사를 시작할 수도 있습니다.

---

주 - 그래픽 인터페이스를 시작하지 않고 마법사를 시작할 수도 있습니다. 예를 들어 Program Files\Sun\Apoc\bin\apoc-config -nodisplay를 실행하여 콘솔 메뉴에서 마법사를 시작합니다.

---

---

## Configuration Agent 제거

Configuration Agent를 제거하려면 시작 메뉴에서 모든 프로그램-Sun-Java Desktop System-**Configuration Agent**-Configuration Agent 제거를 선택합니다.

## 부록 A

---

# Sun Web Console 패키지

---

## 알려진 문제

### 보안

사용자가 인식하지 못하는 동안 특정 사용자 작업에 의해 세션이 활성 상태로 방치될 수 있습니다. 예를 들어, 사용자가 브라우저 창을 닫아도 Sun Web Console에서 자동으로 로그아웃되지 않습니다. 응용 프로그램 창을 닫기 전에 Sun Web Console을 사용하여 명시적으로 세션에서 로그아웃해야 합니다.

## 설치 스크립트 사용

개요: `setup [-h] | [-n] | [-d <var>, <arch>[, client1, client2, ...]] [-u [-f]]`

-h = 사용법을 인쇄합니다.

-n = 설치 후 서버를 시작하지 않습니다.

-u = Sun Web Console을 제거합니다.

-f = 설치 응용 프로그램과 함께 설치된 Tomcat 및 Java 1.4 패키지를 제거합니다. 이 매개 변수는 -u 매개 변수와 함께 사용해야 합니다.

사용 가능한 설치 매개 변수에 대한 자세한 내용을 보려면 `setup -h`를 실행합니다.

Sun Web Console을 제거하려면 `/usr/lib/webconsole/setup -u`를 실행합니다.

---

주 - /usr/lib/webconsole 디렉토리나 관련 하위 디렉토리에서는 이 명령을 실행하지 마십시오. 그렇지 않으면 pkgrm이 실패합니다.

---

---

## Sun Web Console 패키지

### Solaris 패키지

패키지 이름	설명
SUNWmctag	Sun Web Console UI 태그 라이브러리
SUNWmcon	Sun Web Console
SUNWmcos	Sun Web Console을 위한 공통 Solaris 서비스
SUNWmcosx	Sun Web Console을 위한 Solaris 릴리스별 서비스
SUNWmconr	Sun Web Console 루트
SUNWjato	Sun One Application Framework 런타임
SUNWtcatu	Tomcat

### Linux RPM

패키지 이름	설명
SUNWmctag	Sun Web Console UI 태그 라이브러리
SUNWmcon	Sun Web Console
SUNWmcos	Sun Web Console을 위한 공통 Linux 서비스
SUNWmcosx	Sun Web Console을 위한 Linux 릴리스별 서비스
SUNWmconr	Sun Web Console 루트



패키지 이름	설명
SUNWjato	Sun One Application Framework 런타임
tomcat4	Tomcat



## 부록 B

---

# Configuration Manager 패키지

---

## Configuration Manager 패키지

### Solaris 패키지

패키지 이름	설명
SUNWapm	Configuration Manager
SUNWapmca	Configuration Agent 템플릿
SUNWapmev	Evolution 템플릿
SUNWapg26	Gnome 2.6 템플릿
SUNWapmmo	Mozilla 템플릿
SUNWapmso	StarSuite 템플릿

### Linux RPM

패키지 이름	설명
apoc-manager	Configuration Manager
apoc-agent-templates	Configuration Agent 템플릿

패키지 이름	설명
apoc-evolution-templates	Evolution 템플리트
26-templates	Gnome 2.6 템플리트
apoc-mozilla-templates	Mozilla 템플리트
apoc-staroffice-templates	StarSuite 템플리트

# OpenLDAP 및 Active Directory를 Configuration Manager와 함께 사용

---

## OpenLDAP 서버를 Configuration Manager와 함께 사용

OpenLDAP 서버를 Configuration Manager 데이터를 위한 저장소로 사용하려면, 구성 데이터를 저장하는 데 사용되는 개체 클래스와 속성을 지원하도록 서버의 스키마를 확장해야 합니다. `apoc.schema`라는 사용자 정의 스키마 파일이 Java Desktop System Management Tools CD에 제공된 Configuration Manager 배포 도구의 `openldap` 하위 디렉토리에 있습니다.

OpenLDAP 구성 디렉토리(`/etc/openldap`)의 `schema` 하위 디렉토리에 이 파일을 복사하고 그 디렉토리에 있는 `slapd.conf` 파일에 포함시켜 OpenLDAP 스키마에 추가해야 합니다. 이렇게 하려면 그 파일에 있는 스키마 순서의 끝에 `include /etc/openldap/schema/apoc.schema`라는 행을 삽입하면 됩니다. OpenLDAP 서버의 스키마 확장에 대한 자세한 내용은 서버의 설명서를 참조하십시오.

OpenLDAP 데이터베이스가 구성 데이터를 저장하도록 준비하려면 Configuration Manager와 함께 제공된 배포 도구를 사용해야 합니다. 이전 설치 단계에서 이미 확장된 스키마의 경우 `createServiceTree` 스크립트만 실행하면 됩니다. 이 스크립트는 누구든지 배포 도구 디렉토리에서 다음 명령을 사용하여 시작해야 합니다.

```
./createServiceTree
```

스크립트는 이 문서의 배포 도구 절에서 설명한 대로 사용자에게 OpenLDAP 데이터베이스에 대한 정보를 표시합니다. OpenLDAP에 포함된 일반적인 개체 클래스 및 속성을 사용하는 기본 매핑 파일은 배포 도구의 `openldap` 하위 디렉토리에 있습니다. 이 파일은 `OrganisationalMapping`이라고 하며 `createServiceTree`를 실행하기 전에 주 배포 도구 디렉토리에 있는 동일한 이름의 파일을 덮어써서 배포할 수 있습니다.

---

주 - Configuration Manager Agent에서는 데이터가 필요한 사용자의 DN을 제공하되 암호는 제공하지 않음으로써 익명으로 OpenLDAP 서버에 연결합니다. 일부 OpenLDAP 서버 릴리스에서는 이러한 익명 인증 모드를 사용하지 않도록 기본적으로 설정되어 있습니다. 이 경우에는 OpenLDAP 구성 디렉토리(/etc/openldap)에 있는 slapd.conf 파일에 정의된 공통 서버 매개 변수에 allow bind anon\_cred 행을 추가하여 사용할 수 있도록 설정해야 합니다. 해당 매핑에 대한 자세한 내용은 서버의 설명서를 참조하십시오.

---

## Active Directory 서버를 Configuration Manager와 함께 사용

Active Directory 서버를 Configuration Manager 데이터를 위한 저장소로 사용하려면, 구성 데이터를 저장하는 데 사용되는 개체 클래스와 속성을 지원하도록 서버의 스키마를 확장해야 합니다. apoc-ad.ldf라는 스키마 확장 파일이 Management Tools CD에 제공된 Configuration Manager 배포 도구의 ad 하위 디렉토리에 있습니다. 자세한 내용은 배포 도구 절을 참조하십시오.

다음 단계를 따라 apoc-ad.ldf 파일을 Active Directory 스키마로 가져와야 합니다.

1. 스키마 확장을 활성화합니다. 해당 작업을 수행하는 방법에 대한 자세한 내용은 Active Directory 설명서를 참조하십시오.
2. 명령 프롬프트에서 다음을 실행합니다. `ldifde -i -c "DC=Sun,DC=COM" <BaseDN> -f apoc-ad-registry.ldf.`

---

주 - <BaseDN>을 Active Directory 기본 DN으로 바꿉니다.

---

Active Directory 서버가 구성 데이터를 저장하도록 준비하려면 배포 도구를 사용해야 합니다. 이전 설치 단계에서 이미 확장된 스키마의 경우 createServiceTree 스크립트만 실행하면 됩니다. 이 스크립트는 누구든지 배포 도구 디렉토리에서 다음 명령을 사용하여 시작해야 합니다. ./createServiceTree. 스크립트는 사용자에게 Active Directory 데이터베이스에 대한 정보를 표시합니다. Active Directory에 포함된 일반적인 개체 클래스 및 속성을 사용하는 기본 매핑 파일은 배포 도구 디렉토리의 ad 하위 디렉토리에 있습니다. 파일의 이름은 OrganisationalMapping이며 createServiceTree를 시작하기 전에 이 파일을 배포 도구 디렉토리에 있는 같은 이름의 파일에 복사하여 배포할 수 있습니다.

이 시점부터 Active Directory 서버를 Configuration Manager와 함께 사용할 수 있습니다. Configuration Manager를 설치할 때 트리에 대한 읽기 권한이 있는 사용자의 전체 DN과 암호를 제공하십시오. 이 사용자는 다른 목적으로는 Active Directory를 사용할

수 없는 사용자일 수 있습니다. 해당 사용자를 설정하는 방법에 대한 자세한 내용은 Active Directory 설명서를 참조하십시오. 또한 Configuration Manager를 실행하는 시스템에서 Active Directory의 도메인 이름을 알고 있어야 합니다. 이렇게 하려면 Active Directory 서버의 IP 주소를 도메인 이름과 함께 그 시스템의 /etc/hosts 파일로 매핑하는 행을 추가하면 됩니다.

Java Desktop System 호스트에서 구성 데이터를 검색하려면 해당 호스트에서도 Active Directory의 도메인 이름을 알고 있어야 합니다. Java Desktop System 사용자 인증은 익명 및 GSSAPI 사용의 두 가지 방법으로 수행할 수 있습니다.

- 익명 연결을 사용하여 인증하려면 Active Directory 서버가 모든 사람에게 읽기 권한을 부여하도록 구성되어야 합니다. 해당 작업을 수행하는 방법에 대한 자세한 내용은 Active Directory 설명서를 참조하십시오.
- GSSAPI를 사용하여 인증하려면 커beros 매개 변수를 지정하는 /etc/krb5.conf 파일을 수정하여 Active Directory 영역을 정의하고 Active Directory 서버를 KDC (Key Distribution Center)로 가리키도록 해야 합니다. 또한 기본 암호화 유형으로 Active Directory가 지원하는 DES 유형, 즉 des-cbc-crc 및 des-cbc-md5를 지정해야 합니다. 해당 작업을 수행하는 방법에 대한 자세한 내용은 커beros 설명서를 참조하십시오. 구성 데이터에 액세스하기 전에 Java Desktop System에 로그인한 사용자에 대한 유효한 자격 증명을 획득해야 합니다. 이 작업은 kinit 명령을 실행하고 Active Directory에 정의된 사용자 암호를 제공하여 수동으로 수행할 수 있습니다. 다른 스키마는 로그인 시 자동으로 이러한 자격 증명을 생성할 수 있습니다. 자세한 내용은 Java Desktop System 설명서를 참조하십시오.

