

# Oracle® Solaris 11.2 시스템 설치

**ORACLE**

부품 번호: E53732  
2014년 7월

Copyright © 2011, 2014, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 계약서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지 사항이 적용됩니다.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 자회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록상표입니다.

본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제 3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스에 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다.

# 목차

---

이 설명서 사용 .....	15
<b>I Oracle Solaris 11.2 설치 옵션 .....</b>	<b>17</b>
<b>1 설치 옵션 개요 .....</b>	<b>19</b>
설치 옵션 비교 .....	19
추가 설치 옵션 .....	21
설치의 새로운 기능 .....	21
<b>II 설치 매체를 사용하여 설치 .....</b>	<b>23</b>
<b>2 설치 준비 .....</b>	<b>25</b>
라이브 매체 및 텍스트 설치의 시스템 요구 사항 .....	25
다중 운영 체제 설치를 위한 시스템 준비 .....	25
시스템 분할 영역 .....	26
적절한 장치 드라이버가 있는지 확인 .....	30
Oracle Configuration Manager 사용 .....	30
<b>3 라이브 매체 사용 .....</b>	<b>33</b>
GUI 설치 프로그램으로 설치 .....	33
시스템이 콘솔 모드로 부트하는 경우 수행할 작업 .....	39
라이브 매체 설치 후 소프트웨어 추가 .....	40
<b>4 텍스트 설치 프로그램 사용 .....</b>	<b>43</b>
텍스트 설치 프로그램으로 설치 .....	43
텍스트 설치 작업 .....	44
<b>5 매체에서 부트하는 자동 설치 .....</b>	<b>55</b>
AI 매체를 사용한 설치 개요 .....	55
AI 매체를 사용한 설치 .....	55
<b>6 Oracle Solaris 인스턴스 구성 해제 또는 재구성 .....</b>	<b>63</b>
기능 그룹 개요 .....	63
Oracle Solaris 인스턴스 구성 해제 .....	64

시스템 재구성 .....	65
SCI 도구를 사용하여 시스템 구성 프로파일 만들기 .....	69
<b>III 설치 서버를 사용하여 설치 .....</b>	<b>71</b>
<b>7 다중 클라이언트의 자동 설치 .....</b>	<b>73</b>
자동 설치란? .....	73
자동 설치 프로그램의 구성 요소 .....	73
AI 보안 .....	77
AI 및 영역 .....	77
AI 구성 프로세스 개요 .....	77
AI 클라이언트 부트 .....	78
AI 서버 계획 .....	79
자동 설치 프로그램 사용 사례 .....	80
<b>8 AI 서버 설정 .....</b>	<b>89</b>
AI 서버 설정 작업 .....	89
AI 서버 요구 사항 .....	89
설치 서비스 작업 권한 .....	90
AI 서버 구성 .....	91
설치 서비스 작업 .....	96
설치 서비스 만들기 .....	96
클라이언트를 설치 서비스와 연관 .....	102
설치 지침 사용자 정의 .....	104
AI SMF 서비스 관리 .....	106
자동 설치의 보안 수준 향상 .....	106
설치 서비스에 대한 정보 표시 .....	120
설치 서비스 관리 .....	126
AI 매니페스트 관리 .....	129
시스템 구성 프로파일 관리 .....	130
<b>9 설치 사용자 정의 .....</b>	<b>133</b>
클라이언트를 설치 및 구성 지침과 일치 .....	133
AI 매니페스트 선택 .....	134
시스템 구성 프로파일 선택 .....	135
선택 조건 .....	135
<b>10 클라이언트 시스템 프로비전 .....</b>	<b>139</b>
XML AI 매니페스트 파일 사용자 정의 .....	139
클라이언트 설치 타임에 AI 매니페스트 만들기 .....	142
AI 매니페스트 마법사를 사용하여 AI 매니페스트 만들기 .....	158
예제 AI 매니페스트 .....	160

기본 SI 매니페스트 .....	164
<b>11 클라이언트 시스템 구성 .....</b>	<b>165</b>
구성 프로파일 제공 .....	165
시스템 구성 프로파일에 구성 지정 .....	167
시스템 구성 프로파일 템플릿 사용 .....	178
예제 시스템 구성 프로파일 .....	180
<b>12 영역 설치 및 구성 .....</b>	<b>197</b>
SI의 비전역 영역 설치 방법 .....	197
전역 영역 SI 매니페스트에서 비전역 영역 지정 .....	198
비전역 영역 구성 및 설치 데이터 .....	198
<b>13 처음 부트 중 사용자 정의 스크립트 실행 .....</b>	<b>205</b>
처음 부트 시 한번 실행 컨트롤 구현 .....	205
처음 부트 시 실행할 스크립트 만들기 .....	206
SMF 매니페스트 파일 만들기 .....	209
스크립트 및 서비스에 대한 IPS 패키지 만들기 .....	213
SI 클라이언트에 처음 부트 패키지 설치 .....	215
처음 부트 서비스 테스트 .....	216
<b>14 클라이언트 시스템 설치 .....</b>	<b>219</b>
클라이언트가 설치되는 방법 .....	219
SPARC 및 x86 클라이언트 시스템 요구 사항 .....	220
SI 클라이언트 설정 .....	221
클라이언트 설치 .....	222
<b>15 자동 설치 문제 해결 .....</b>	<b>229</b>
클라이언트 설치 실패 .....	229
설치 시작 없이 설치 환경 부트 .....	240
명령줄에서 자동 설치 시작 .....	241
<b>IV 관련 작업 수행 .....</b>	<b>243</b>
<b>A Oracle Configuration Manager 작업 .....</b>	<b>245</b>
Oracle Configuration Manager 소개 .....	245
Oracle Configuration Manager 중앙 수집기 정보 .....	246
Oracle Configuration Manager 관리(작업) .....	247
<b>B Device Driver Utility 사용 .....</b>	<b>251</b>
장치 드라이버 유틸리티 개요 .....	251
<b>색인 .....</b>	<b>255</b>



## 그림

---

그림 5-1	매체를 사용한 AI 설치 .....	56
그림 7-1	AI 네트워크 예 .....	74
그림 7-2	아키텍처 하나와 OS 하나를 지원하는 AI 서버 .....	80
그림 7-3	아키텍처 2개를 지원하는 AI 서버 .....	81
그림 7-4	아키텍처 하나와 디스크 레이아웃 2개를 지원하는 AI 서버 .....	82
그림 7-5	아키텍처 하나와 디스크 구성 2개를 지원하는 AI 서버 .....	83
그림 7-6	아키텍처 하나와 릴리스 2개를 지원하는 AI 서버 .....	84
그림 7-7	일부 클라이언트에 대한 추가 구성이 포함된 아키텍처 하나를 지원하는 AI 서버 .....	85
그림 7-8	여러 구성 변경을 지원하는 AI 서버 .....	86



## 표

---

표 1-1	설치 옵션 .....	19
표 2-1	다중 운영 체제 환경 .....	25
표 2-2	대화식 설치 중 디스크 분할 옵션 .....	28
표 2-3	텍스트 설치 중 VTOC 슬라임을 수정하는 옵션 .....	29
표 6-1	기능 그룹 .....	63
표 9-1	조건 키워드 및 조건 계층 .....	136
표 10-1	클라이언트 속성 환경 변수 .....	144
표 11-1	root_account 등록 정보 그룹의 등록 정보 .....	169
표 11-2	user_account 등록 정보 그룹의 등록 정보 .....	170
표 11-3	config 등록 정보 그룹의 등록 정보 .....	172
표 11-4	timezone 등록 정보 그룹의 등록 정보 .....	173
표 11-5	environment 등록 정보 그룹의 등록 정보 .....	173
표 11-6	IPv4 네트워크 인터페이스에 대한 등록 정보 그룹 등록 정보 .....	175
표 11-7	IPv6 네트워크 인터페이스에 대한 등록 정보 그룹 등록 정보 .....	176
표 11-8	svc:/system/name-service/switch 등록 정보 그룹의 config 등록 정보 .....	177
표 11-9	config 등록 정보 그룹의 등록 정보 .....	177
표 11-10	시스템 구성 템플릿 프로파일의 변수 .....	179



## 코드 예

---

예 6-1	시스템 구성 해제 .....	64
예 6-2	SMF 데이터 구성 해제 .....	64
예 6-3	특정 기능 그룹 구성 해제 .....	64
예 6-4	시스템 구성 해제 후 재부트 .....	64
예 6-5	시스템 구성 프로파일을 사용하여 시스템 재구성 .....	65
예 6-6	기본 프로파일 사용 .....	69
예 6-7	프로파일 만들기 및 사용 .....	69
예 6-8	프로파일을 만들고 기능 그룹 구성에 사용 .....	69
예 7-1	아키텍처 하나와 OS 하나를 지원하는 AI 서버 .....	80
예 7-2	아키텍처 2개를 지원하는 AI 서버 .....	81
예 7-3	아키텍처 하나와 디스크 구성 2개를 지원하는 AI 서버 .....	82
예 7-4	아키텍처 하나와 시간대 2개를 지원하는 AI 서버 .....	83
예 7-5	아키텍처 하나와 릴리스 2개를 지원하는 AI 서버 .....	84
예 7-6	일부 클라이언트에 대한 추가 구성이 포함된 아키텍처 하나를 지원하는 AI 서버 .....	85
예 7-7	여러 구성 변경을 지원하는 AI 서버 .....	86
예 8-1	네트워크에서 AI 지원 사용 안함 .....	94
예 8-2	AI 서버에서 지원하는 네트워크 포함 .....	94
예 8-3	AI 웹 서버 호스트 포트 구성 .....	94
예 8-4	보안 AI 웹 서버 호스트 포트 구성 .....	94
예 8-5	기본 이미지 경로 구성 .....	94
예 8-6	AI 서버용 AI 클라이언트의 IP 주소 및 번호 정의 .....	94
예 8-7	AI 서버에서 로컬 DHCP 서비스의 자동 업데이트 사용 안함 .....	95
예 8-8	AI 서버에서 DHCP 업데이트 사용 .....	95
예 8-9	AI 서버에서 DHCP가 사용으로 설정된 상태에서 ISO 파일을 사용하여 SPARC 설치 서비스 만들기 .....	97
예 8-10	IPS 패키지를 사용하여 x86 설치 서비스 만들기 .....	98
예 8-11	다른 아키텍처용 설치 서비스 만들기 .....	99
예 8-12	자동으로 X86 클라이언트를 설치하는 서비스 만들기 .....	99
예 8-13	SPARC 클라이언트를 서비스와 연관 .....	103

예 8-14	x86: X86 클라이언트를 서비스에 연관시키고 출력을 직렬 회선으로 재 지정 .....	103
예 8-15	x86: X86 클라이언트의 부트 등록 정보 변경 .....	103
예 8-16	클라이언트 조건을 매니페스트와 연관 .....	105
예 8-17	클라이언트 조건을 스크립트와 연관 .....	105
예 8-18	설치 서비스용 기본 매니페스트 만들기 .....	105
예 8-19	클라이언트 조건을 시스템 구성 프로파일과 연관 .....	106
예 8-20	AI SMF 서비스 사용 .....	106
예 8-21	AI SMF 서비스 사용 안함 .....	106
예 8-22	사용자 제공 자격 증명을 사용하여 AI 서버 자격 증명 생성 .....	110
예 8-23	설치 중 AI 서버 인증 요구 .....	112
예 8-24	x86: 설치 중 암호화 요구 .....	112
예 8-25	특정 클라이언트에 대해 사용자가 제공한 자격 증명 사용 .....	113
예 8-26	특정 설치 서비스의 클라이언트용 자격 증명 .....	113
예 8-27	기본 클라이언트 자격 증명 .....	114
예 8-28	단일 클라이언트용 인증서 삭제 .....	116
예 8-29	CA 인증서 삭제 .....	116
예 8-30	서버 보안 인증서 삭제 .....	116
예 8-31	Kerberos 클라이언트 배포 중 기존 키 다운로드 .....	119
예 8-32	Kerberos 클라이언트 배포 중 새 키 만들기 .....	119
예 8-33	AI 클라이언트를 MS AD 도메인에 자동 연결 .....	120
예 8-34	AI 서버의 모든 설치 서비스 나열 .....	121
예 8-35	지정된 설치 서비스에 대한 정보 표시 .....	121
예 8-36	설치 서비스와 연관된 클라이언트 나열 .....	122
예 8-37	특정 설치 서비스와 연관된 클라이언트 나열 .....	122
예 8-38	모든 AI 매니페스트 및 시스템 구성 프로파일 나열 .....	122
예 8-39	지정된 설치 서비스와 연관된 매니페스트 및 프로파일 나열 .....	123
예 8-40	서버 보안 정보 나열 .....	124
예 8-41	클라이언트 보안 정보 나열 .....	125
예 8-42	설치 서비스 별칭 만들기 .....	126
예 8-43	설치 서비스 별칭 수정 .....	127
예 8-44	설치 서비스 생성 시 기본 AI 매니페스트 설정 .....	127
예 8-45	기존 설치 서비스를 수정하여 기본 AI 매니페스트 설정 .....	127
예 8-46	새 설치 서비스용 이미지 경로 설정 .....	127
예 8-47	기존 설치 서비스용 이미지 경로 설정 .....	127
예 8-48	설치 서비스 업데이트 .....	128
예 8-49	설치 서비스 업데이트 시 다른 저장소 사용 .....	128
예 8-50	설치 서비스 업데이트 시 다른 넷 이미지 패키지 사용 .....	129
예 9-1	클라이언트를 AI 매니페스트와 일치 .....	134

예 10-1	디스크 크기에 준하여 디스크 분할 지정 .....	146
예 10-2	추가 디스크의 존재 여부에 준하여 루트 폴 레이아웃 지정 .....	148
예 10-3	지정된 크기의 디스크가 적어도 두 개 있는 경우 미러링된 구성 지정 ....	150
예 10-4	IP 주소에 준하여 설치할 패키지 지정 .....	152
예 10-5	대상 디스크가 최소한 특정 크기가 되도록 지정 .....	153
예 10-6	시스템 구성 프로파일 추가 .....	154
예 10-7	부정확한 매니페스트 사양의 스크립트 .....	154
예 10-8	시 매니페스트 마법사 사용 안함으로 설정 .....	158
예 10-9	시 서버에 매니페스트 파일을 저장하도록 허용 .....	159
예 11-1	만료된 암호로만 루트 계정 구성 .....	169
예 11-2	SSH 키 구성 .....	171
예 11-3	호스트 이름 구성 .....	172
예 11-4	노드 이름 매핑 사용 안함 .....	172
예 11-5	시간대 구성 .....	173
예 11-6	로케일 구성 .....	173
예 11-7	단말기 유형 구성 .....	174
예 11-8	키보드 레이아웃 구성 .....	174
예 11-9	지정된 도메인에 대해 NIS 사용 .....	186
예 11-10	NIS 구성 및 DNS 사용 안함 .....	187
예 11-11	NIS 구성 .....	188
예 11-12	지정된 도메인에 대해 NIS 및 DNS 사용 .....	188
예 11-13	검색 목록으로 DNS 구성 .....	190
예 11-14	LDAP 및 LDAP 검색 기준 구성 .....	191
예 11-15	보안 LDAP 서버로 LDAP 구성 .....	191
예 12-1	기본 영역 시 매니페스트 .....	200
예 13-1	템플릿 처음 부트 스크립트 .....	206
예 13-2	다중 IP 인터페이스를 구성하는 처음 부트 스크립트 .....	208
예 13-3	생성된 SMF 서비스 매니페스트 .....	209
예 13-4	사용자 정의 서비스 매니페스트: 스크립트 실행에 허용된 시간 늘리기 .....	211
예 13-5	사용자 정의 서비스 매니페스트: 비전역 영역 설치 후 스크립트 실행 확 인 .....	212



## 이 설명서 사용

---

- **개요** - Oracle Solaris 11.2 릴리스를 설치하고 자동 설치 서버를 구성하는 방법을 설명합니다.
- **대상** - 기술자, 시스템 관리자 및 공인 서비스 공급자
- **필요한 지식** - Oracle Solaris 사용 경험이 필요합니다.

## 제품 설명서 라이브러리

이 제품에 대한 최신 정보 및 알려진 문제는 설명서 라이브러리(<http://www.oracle.com/pls/topic/lookup?ctx=E56343>)에서 확인할 수 있습니다.

## Oracle 지원 액세스

Oracle 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

## 피드백

<http://www.oracle.com/goto/docfeedback>에서 이 설명서에 대한 피드백을 보낼 수 있습니다.



부 I

## Oracle Solaris 11.2 설치 옵션



## 설치 옵션 개요

Oracle Solaris 소프트웨어는 사용자 요구에 따라 여러 가지 다양한 방법으로 설치할 수 있습니다. 이 장에서는 Oracle Solaris 설치 옵션에 대해 설명합니다.

### 설치 옵션 비교

다음 표에서는 다양한 설치 옵션의 기능을 비교합니다.

표 1-1 설치 옵션

설치 옵션	자세한 정보	최소 준비	서버 필요	패키지 저장소에서 패키지 설치
x86 전용: GUI 설치 프로그램	<a href="#">3장. 라이브 매체 사용</a>	예	아니오, 매체에서 설치	아니오
텍스트 설치 프로그램	<a href="#">4장. 텍스트 설치 프로그램 사용</a>	예	아니오, 매체에서 설치	아니오
설치 서버에서 부트 지원을 사용하여 텍스트 설치	<a href="#">네트워크를 통해 텍스트 설치를 시작하는 방법 [51]</a>	아니오	예, 서버에서 설치 이미지 검색	아니오
매체를 사용하는 자동 설치 프로그램	<a href="#">5장. 매체에서 부트하는 자동 설치</a>	아니오	설치 매체를 사용자 정의하려는 경우 서버가 필요하지만, 설치에는 필요하지 않음	예
다중 클라이언트를 위한 자동 설치 프로그램	<a href="#">7장. 다중 클라이언트의 자동 설치</a>	아니오	예, 서버 필요	예

또한 사용자 정의 라이브 매체 이미지, 텍스트 설치 프로그램 이미지 및 자동 설치 이미지를 비롯한 설치 이미지를 만드는 옵션이 있습니다. [“사용자 정의 Oracle Solaris 11.2 설치 이미지 만들기”](#)를 참조하십시오.

### 간단한 설치

라이브 매체의 GUI 설치 프로그램과 텍스트 설치 프로그램은 간단한 설치 방법입니다.

- 이 방법 중 하나를 사용하여 x86 플랫폼에 Oracle Solaris를 설치할 수 있습니다. 텍스트 설치 프로그램을 사용하여 SPARC 플랫폼에 Oracle Solaris를 설치할 수도 있습니다.

- 두 설치 프로그램 모두 최소 메모리로 작동할 수 있습니다. 메모리 요구 사항은 “[Oracle Solaris 11.2 릴리스 노트](#)”를 참조하십시오.
- 두 설치 프로그램을 통해 설치하는 동안 디스크 분할 영역을 선택, 만들기 또는 수정할 수 있습니다.

Live Media에는 데스크탑 또는 랩탑에 적절한 소프트웨어 세트가 들어 있습니다. 텍스트 설치 프로그램은 범용 서버 시스템에 적절한 더 작은 소프트웨어 세트를 설치합니다.

텍스트 설치 프로그램은 GUI 설치 프로그램에 비해 다음과 같은 장점이 있습니다.

- SPARC 또는 x86 기반 시스템에 운영 체제를 설치할 수 있습니다.
- 그래픽 카드가 없거나 필요하지 않은 시스템에서 사용할 수 있습니다.
- 시스템의 사양에 따라 GUI 설치 프로그램보다 적은 메모리가 필요할 수 있습니다.
- 네트워크 및 이름 지정 서비스의 수동 구성이 가능합니다.
- 네트워크가 자동 설치를 수행하도록 설정된 경우 네트워크를 통해 텍스트 설치를 수행할 수 있습니다. 네트워크에 설치 서비스를 설정하고 클라이언트 시스템을 부트할 때 텍스트 설치를 선택하면 됩니다.

---

**참고** - 텍스트 설치 프로그램은 `solaris-large-server` 패키지 세트를 설치합니다. 그러나 네트워크를 통해 텍스트 설치 프로그램을 사용할 경우 이와 달리 더 작은 패키지 세트인 `solaris-auto-install`이 설치됩니다. 설치된 시스템으로 부트한 후에 `solaris-large-server` 패키지 세트를 설치해야 합니다.

---

- 분할 영역을 수정할 수 있을 뿐 아니라, 텍스트 설치 프로그램에서는 Solaris 분할 영역 내에 VTOC 슬라임을 만들고 수정을 사용으로 설정할 수 있습니다.

간단한 설치 수행에 대한 자세한 내용은 [설치 매체를 사용하여 설치 \[23\]](#)를 참조하십시오.

## 서버 설정이 필요한 설치

AI(자동 설치 프로그램) 기능을 사용하여 단일/다중 클라이언트 시스템에 Oracle Solaris 소프트웨어의 "핸즈프리" 설치를 수행할 수 있습니다.

아래에 설명한 AI 매체에서 부트하는 경우가 아니라면 먼저 네트워크에서 서버를 설정해야 AI를 사용할 수 있습니다. 클라이언트 시스템을 부트할 때 시스템이 서버에서 설치 사양을 얻고 Oracle Solaris 패키지 저장소에서 소프트웨어 패키지를 검색하면 클라이언트 시스템에 소프트웨어가 설치됩니다.

---

**참고** - 설치 프로세스 중에 네트워크로 연결된 저장소의 패키지를 검색하므로 각 시스템에서 네트워크에 액세스할 수 있어야 합니다.

---

AI는 x86 및 SPARC 기반 클라이언트 시스템에 "핸즈프리" 자동 네트워크 설치를 수행할 수 있습니다. AI 클라이언트는 아키텍처, 디스크/메모리 용량 및 기타 특성이 서로 다를 수 있습니다. 설치는 네트워크 구성, 설치된 패키지 및 기타 사양에 따라 다를 수 있습니다.

자세한 내용은 [설치 서버를 사용하여 설치 \[71\]](#)를 참조하십시오.

"핸즈프리" 네트워크 설치 외에도 네트워크를 통해 대화식 텍스트 설치를 수행할 수 있습니다. 대화식 설치를 통해 특정 시스템에 대한 설치 사양을 추가로 사용자 정의할 수 있습니다. 자세한 내용은 [네트워크를 통해 텍스트 설치를 시작하는 방법 \[51\]](#)을 참조하십시오.

AI 이미지에 액세스할 수 있는 경우 AI 서버를 구성하지 않았더라도 이미지를 다운로드하여 네트워크나 로컬에 저장할 수 있습니다. 그런 다음 CD, DVD 또는 (x86 설치의 경우) USB 플래시 드라이브와 같은 이동식 매체에 이미지를 구울 수 있습니다. 그러면 각각의 시스템에서 직접 AI 매체를 부트할 수 있습니다. AI 매체를 사용하는 설치는 대화식이 아닙니다. 지침은 [5장. 매체에서 부트하는 자동 설치](#)를 참조하십시오.

## 추가 설치 옵션

이미 설명한 설치 옵션 외에도, Oracle Solaris 운영 체제를 설치 및 수정하기 위한 다음 옵션이 있습니다

Creating custom installation images(사용자 정의 설치 이미지 만들기)

배포 생성자 도구를 사용하여 미리 구성된 Oracle Solaris 설치 이미지를 만들 수 있습니다. 이 도구는 사용자 정의된 XML 매니페스트 파일을 입력으로 사용하고 매니페스트 파일에 지정된 매개변수를 기반으로 설치 이미지를 작성합니다. 기본 설치 이미지 중 하나를 기반으로 사용자 정의 이미지를 작성할 수 있습니다. 예를 들어, 사용자 정의 텍스트 설치 프로그램 이미지 또는 사용자 정의 GUI 설치 프로그램 이미지를 작성할 수 있습니다. 자세한 내용은 ["사용자 정의 Oracle Solaris 11.2 설치 이미지 만들기"](#)를 참조하십시오.

Updating an installed Oracle Solaris system(설치된 Oracle Solaris 시스템 업데이트)

기존 Oracle Solaris 설치 시스템을 업데이트하는 데 설치 프로그램을 사용할 수 없습니다. 대신, pkg 유틸리티를 사용하여 패키지 저장소에 액세스하고 시스템에 해당하는 신규 또는 업데이트된 소프트웨어 패키지를 다운로드해야 합니다. 자세한 내용은 ["Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트"](#)를 참조하십시오.

## 설치의 새로운 기능

이 절에서는 기존 고객을 위해 이 릴리스의 설치와 관련된 중요한 새 기능을 중심으로 설명합니다.

- 기능 그룹을 사용하여 실행 중인 시스템 구성의 선택된 부분을 재구성하거나 구성 해제할 수 있습니다. ["기능 그룹 개요" \[63\]](#)를 참조하십시오.
- `installadm` 명령이 향상되어 설치 서비스의 구성이 쉬워졌습니다. 또한 이 명령은 이제 대화식입니다. 명령에 옵션을 포함하지 않을 경우 간단한 프로세스로 `installadm` 명령을

여러 번 입력할 수 있는 프롬프트가 표시됩니다. 명령 예제는 8장. [AI 서버 설정](#)에 나와 있습니다.

- AI 및 보안 IPS 저장소를 사용하여 클라이언트를 안전하게 설치할 수 있습니다. [“자동 설치의 보안 수준 향상” \[106\]](#)을 참조하십시오.
- 기본 AI 매니페스트는 파생된 매니페스트 스크립트가 되어 AI 프로세스에 기존 시스템 구성을 사용하여 AI 구성 단계 중 일부를 단순화할 수 있게 되었습니다. [“기본 AI 매니페스트” \[164\]](#)를 참조하십시오.
- 다음 작업을 지원하도록 AI 매니페스트를 구성할 수 있게 되었습니다.
  - 기존 분할 영역 또는 슬라이스 다시 사용. [“기존의 디스크 슬라이드 또는 분할 영역 다시 사용” \[164\]](#)을 참조하십시오.
  - 여러 SVR4 패키지 설치. [“여러 SVR4 패키지 설치” \[163\]](#)를 참조하십시오.

또한 파생된 매니페스트 스크립트를 작성해 기존 클라이언트의 정보를 사용하여 시스템 구성 프로파일을 만들 수 있으므로 클라이언트 데이터를 수동으로 관리할 필요가 없습니다. [예 10-6. “시스템 구성 프로파일 추가”](#)을 참조하십시오.

- BUI(브라우저 사용자 인터페이스)를 사용하여 AI 매니페스트를 만들 수 있으므로 모든 구성 정보에 대해 XMP 파일을 편집할 필요가 없습니다. [“AI 매니페스트 마법사를 사용하여 AI 매니페스트 만들기” \[158\]](#)를 참조하십시오.
- 이제 시스템 구성 프로파일에 다음 항목이 포함됩니다.
  - 다중 인터페이스구성. [“여러 IPv4 인터페이스 구성” \[185\]](#)을 참조하십시오.
  - 사용자의 SSH 키. [“SSH 키 구성” \[171\]](#)을 참조하십시오.
  - Kerberos 클라이언트 구성 정보. [AI를 사용한 Kerberos 클라이언트 구성 방법 \[117\]](#)을 참조하십시오.

또한 프로파일에 변수를 포함하여 AI 서비스가 설치 클라이언트에서 현재 구성 정보를 수집할 수 있게 되었습니다. [“시스템 구성 프로파일 템플릿 사용” \[178\]](#)을 참조하십시오.

부 II

## 설치 매체를 사용하여 설치



# ◆◆◆ 2 장 2

## 설치 준비

---

시스템을 설치하기 전에 설치를 위한 시스템 요구 사항, 시스템 분할 영역을 위한 제안, Oracle Configuration Manager 관련 정보를 비롯하여 이 장에 나온 정보를 검토하십시오.

### 라이브 매체 및 텍스트 설치의 시스템 요구 사항

라이브 매체 설치 이미지 또는 텍스트 설치 이미지를 사용하여 Oracle Solaris 11.2 릴리스를 설치하는 데 필요한 최소 메모리 및 디스크 공간 요구 사항을 확인하려면 [“Oracle Solaris 11.2 릴리스 노트”](#)를 참조하십시오.

---

**참고** - 텍스트 설치 프로그램은 라이브 매체 설치 프로그램보다 적은 메모리가 필요합니다. 정확한 최소 요구 사항은 시스템 사양에 따라 달라집니다. 시스템에 GUI 설치 프로그램을 실행할 메모리가 부족한 경우 텍스트 설치 프로그램을 대신 사용하십시오.

---

### 다중 운영 체제 설치를 위한 시스템 준비

Oracle Solaris를 다중 부트 환경 시스템의 일부로 설치하는 경우 여러 가지 운영 체제에 대한 다음 사항을 검토하십시오.

표 2-1 다중 운영 체제 환경

기존 운영 체제	설명
Microsoft Windows	Oracle Solaris 릴리스 설치를 위한 충분한 디스크 공간을 설정합니다. 이 릴리스에서 x86 플랫폼용 Oracle Solaris는 새 버전의 GRand Unified Bootloader(GRUB 2)를 사용합니다. Oracle Solaris는 Windows를 인식하며, 설치 중에 Windows 분할 영역이 변경되지 않은 상태로 유지되는지 확인합니다. 설치가 완료되고 시스템이 재부트되면 GRUB 2 메뉴에 Windows 및 Oracle Solaris 부트 항목이 모두 표시됩니다.  GRUB 2에 대한 자세한 내용은 <a href="#">“Oracle Solaris 11.2 시스템 부트 및 종료”</a> 의 <a href="#">“GRUB 2 소개”</a> 를 참조하십시오.
Solaris 10 OS	라이브 매체 설치 프로그램으로는 Oracle Solaris 운영 체제의 다중 인스턴스를 설치할 수 없습니다. 그러나 텍스트 설치 프로그램은 각 인스턴스

기존 운영 체제	설명
	가 서로 다른 슬라이스에 있는 경우 동일한 분할 영역에서 Oracle Solaris 운영 체제의 다중 인스턴스를 지원합니다. 라이브 매체 및 텍스트 설치 프로그램을 사용하여 Oracle Solaris의 다중 인스턴스가 설치된 기존 시스템에서 Solaris 10 1/06 이상의 릴리스를 대체할 수 있습니다. 참고 - 특정 Solaris 볼륨 목차(Volume Table of Contents, VTOC) 슬라이스를 현재 운영 체제에 유지해야 하는 경우 텍스트 설치 프로그램을 사용하십시오.
확장된 분할 영역	확장된 분할 영역에 다른 운영 체제가 있는 경우 설치 중에 기존의 확장된 분할 영역을 변경할 필요는 없습니다. 라이브 매체 설치 프로그램, 텍스트 설치 프로그램 또는 자동 설치 프로그램을 사용하여 Oracle Solaris를 설치할 때 확장된 분할 영역을 만들거나 크기를 조정하거나 삭제할 수 있습니다. 또한 확장된 분할 영역 내 논리적 분할 영역에 Oracle Solaris를 설치할 수도 있습니다.

## 시스템 분할 영역

이 섹션에서는 설치 전에 또는 대화식 설치 도중 시스템을 분할하는 방법을 안내합니다.

설치 프로그램은 전체 디스크 또는 포맷되지 않은 디스크에 설치할 때 GPT 형식을 사용합니다. 단, 기존 GPT 분할 영역 또는 DOS 분할 영역은 기본적으로 유지되며 설치 프로그램에 표시되므로 기존 분할 영역을 유지하고 해당 분할 영역에 설치할 수 있습니다.

참고 - 지원되는 SPARC 기반 시스템에서 GPT 인식 펌웨어를 적용하는 방법에 대한 자세한 내용은 [SPARC: GPT Labeled Disk Support](#)를 참조하십시오.

이 절에서는 Solaris VTOC 슬라이스를 설정하는 방법도 설명합니다.

## 설치 전 시스템 분할 지침



주의 - 하드 드라이브를 분할하기 전에 시스템을 백업하는 것을 잊지 마십시오.

라이브 매체 ISO 이미지 또는 텍스트 설치 프로그램 이미지에서 Oracle Solaris를 설치하는 경우 전체 디스크를 사용하거나 분할 영역에 운영 체제를 설치할 수 있습니다. 또한 SPARC 클라이언트에서는 텍스트 설치 프로그램이 슬라이스에 설치할 수 있습니다.

상용 제품이나 오픈 소스 도구를 사용하여 설치 전 Oracle Solaris를 설치하기 위한 분할 영역을 만들 수 있습니다. 또는 Oracle Solaris 설치 중 분할 영역을 만들 수 있습니다. x86 기반 시스템에서 Oracle Solaris 설치 프로그램은 하나 이상의 드라이브에 여러 개의 운영 체제 부트를 지원하는 GRUB 2를 사용합니다. 다양한 운영 체제를 분할하고 설치한 후에는 부트 시 GRUB 2 메뉴에서 적절한 메뉴 항목을 선택하여 운영 체제 중 하나를 배포할 수 있습니다.

GRUB 2에 대한 자세한 내용은 “Oracle Solaris 11.2 시스템 부트 및 종료”의 “GRUB 2 소개”를 참조하십시오.

---

참고 - Linux 스왑 분할 영역을 만드는 경우 Oracle Solaris에 사용된 것과 동일한 분할 영역 ID가 Linux 스왑에 사용됩니다. 설치 중 디스크 분할 단계에서 Linux 스왑 분할 영역을 Oracle Solaris 분할 영역으로 변경할 수 있습니다.

---

## 대화식 설치 중 시스템 분할 지침

x86 기반 시스템에서 GUI 설치 또는 텍스트 설치 중 분할 영역을 선택, 만들기 또는 수정할 수 있습니다. 설치 프로그램은 전체 디스크 또는 포맷되지 않은 디스크에 설치할 때 GPT 형식을 사용합니다. 단, 기존 GPT 분할 영역 또는 DOS 분할 영역은 기본적으로 유지되며 설치 프로그램에 표시되므로 기존 분할 영역을 유지하고 해당 분할 영역에 설치할 수 있습니다. 더불어, 텍스트 설치 프로그램인 경우에만 대화식 설치 중에 VTOC 슬라이스를 선택, 만들기 또는 수정할 수 있습니다.

Oracle Solaris를 설치할 때 디스크 분할에 대해 다음과 같은 중요한 정보를 고려하십시오.

- 다음 분할 사양을 확인합니다.
  - 디스크에 기존 DOS 분할 영역이 있을 경우 최대 4개의 DOS 기본 분할 영역이 표시됩니다. DOS 확장 분할 영역이 존재하는 경우 논리적 분할 영역도 확장된 분할 영역 내에 디스크 레이아웃 순서로 표시됩니다. Solaris 분할 영역은 하나만 허용되며 해당 Solaris 분할 영역을 설치에 사용해야 합니다. Solaris 분할 영역은 확장된 분할 영역 내 논리적 분할 영역일 수 있습니다.
  - 디스크에 기존 GPT 분할 영역이 있을 경우 GPT 분할 영역이 표시됩니다. 최대 7개의 GPT 분할 영역이 지원됩니다. 설치 중 하나 이상의 Solaris 분할 영역을 만들 수 있지만 하나의 Solaris 분할 영역을 설치 대상으로 선택해야 합니다. 기존 Solaris GPT 분할 영역이 여러 개인 경우 적합한 첫번째 Solaris GPT 분할 영역이 기본적으로 설치 대상으로 선택됩니다.
- 다음 조건에 해당될 경우 Oracle Solaris 설치가 전체 디스크 레이아웃을 덮어씁니다.
  - 디스크 테이블을 읽을 수 없는 경우
  - 디스크가 이전에 분할된 적이 없는 경우
  - 설치할 전체 디스크를 선택합니다.
- 기존 Oracle Solaris 분할 영역이 있는 경우 사용자가 기존 분할 영역을 수정하지 않으면 설치 기본값이 Oracle Solaris 분할 영역만 덮어씁니다. 해당 분할 영역은 기존의 확장된 분할 영역 내에서 논리적 분할 영역이 될 수 있습니다. 따라서 기존의 다른 분할 영역은 변경되지 않습니다.
- Solaris 분할 영역을 설치에 사용해야 합니다.
- 디스크 분할 또는 슬라이스에서 변경한 사항은 설치 프로그램 패널 선택을 마치고 설치가 시작된 다음에야 구현됩니다. 설치 전 언제든지 변경을 취소하고 원래 설정을 복원할 수 있습니다.
- 기존 분할 영역 테이블을 읽을 수 없는 경우 분할 제안 정보가 표시됩니다.



주의 - 이 경우 설치하는 동안 디스크의 기존 데이터가 모두 삭제됩니다.

- 설치하는 동안 Partition the Disk(디스크 분할) 옵션을 선택하면 선택한 디스크의 기존 분할 영역이 디스크에 배치될 때와 동일한 순서로 패널에 표시됩니다. 이러한 분할 영역에 대해 사용되지 않은 디스크 공간이 표시됩니다. 각 분할 영역에 대해 분할 영역 유형, 현재 크기 및 사용 가능한 최대 디스크 공간도 표시됩니다. 확장된 분할 영역이 있는 경우 논리적 분할 영역도 확장된 분할 영역 내에 디스크 레이아웃 순서로 표시됩니다.
- 설치를 제대로 수행하기에 공간이 부족한 디스크나 분할 영역에는 레이블이 표시됩니다.

### x86: 대화식 설치 중 분할 영역 설정

x86 플랫폼에 설치하는 경우 설치 화면에서 항목을 직접 편집하여 디스크 분할을 변경할 수 있습니다. 설치를 계속 진행하면 소프트웨어 설치에 필요한 최소 및 권장 크기도 표시됩니다.

다음 표는 디스크 분할 옵션을 설명합니다. 이 표를 사용하여 사용자의 요구에 가장 잘 맞는 옵션을 결정할 수 있습니다.

표 2-2 대화식 설치 중 디스크 분할 옵션

분할 영역 옵션	설명 및 사용자 조치(필요할 경우)
기존 Solaris 분할 영역을 사용합니다.	이 옵션은 현재 크기를 사용하여 기존 Solaris 분할 영역에 Oracle Solaris 운영 체제를 설치합니다. 디스크 분할 옵션을 선택합니다. 다른 사항은 변경할 필요가 없습니다.
Solaris 분할 영역이 없는 경우 새 Solaris 분할 영역을 만들어야 합니다.	현재 시스템에 기존 Solaris 분할 영역이 없는 경우 기본 분할 영역 또는 논리적 분할 영역을 선택한 후 해당 유형을 Solaris로 변경하여 새 Solaris 분할 영역을 만들어야 합니다. 설치 중에 이렇게 수정하면 기존 분할 영역 내용이 지워집니다.
Solaris 분할 영역에 할당되는 공간을 늘리고 해당 분할 영역에 설치합니다.	사용 가능한 디스크 공간이 충분한 경우 Solaris 분할 영역에 할당되는 크기를 늘린 후 해당 분할 영역에 소프트웨어를 설치할 수 있습니다. 선택한 분할 영역 앞이나 뒤의 사용되지 않는 인접 공간이 사용 가능한 공간에 포함됩니다. 분할 영역을 확대하면 분할 영역 뒤의 사용되지 않는 공간이 먼저 사용됩니다. 그런 다음 분할 영역 앞의 사용되지 않는 공간이 사용되고 선택한 분할 영역의 시작 실린더가 변경됩니다.
다른 Solaris 분할 영역에 Oracle Solaris 운영 체제를 설치합니다.	다른 Solaris 분할 영역에 운영 체제를 설치할 수 있습니다. 다른 분할 영역을 선택하고 유형을 Solaris로 변경합니다. 설치하는 동안 이렇게 수정하면 이전 Solaris 분할 영역과 새 Solaris 분할 영역의 기존 분할 영역 내용이 모두 지워집니다. 참고 - 시스템에 기존 DOS 분할 영역이 있는 경우 Solaris 분할 영역이 하나만 허용됩니다. 새 Solaris 분할 영역을 만들기 전에 먼저 기존 Solaris 분할 영역 유형을 Unused(사용되지 않음)로 변경해야 합니다.
확장된 분할 영역 내에 새로운 Solaris 분할 영역을 만듭니다.	확장된 분할 영역 내에 새로운 Solaris 분할 영역을 만들 수 있습니다. 분할 영역 유형을 Extended(확장)로 변경합니다. 확장된 분할 영역의 크기를 조정하면 다음 확장된 분할 영역 안의 논리적 분할 영역 중 하나를 Solaris 분할 영역으로 변경할 수 있습니다. 또한 논

분할 영역 옵션	설명 및 사용자 조치(필요할 경우)
	<p>리적 분할 영역의 크기를 해당 논리적 분할 영역을 포함한 확장된 분할 영역의 크기까지 늘릴 수 있습니다.</p> <p>참고 - 시스템에 기존 DOS 분할 영역이 있는 경우 Solaris 분할 영역이 하나만 허용됩니다. 확장된 분할 영역 내에 Solaris 분할 영역을 만들기 전에 먼저 기존 Solaris 분할 영역 유형을 Unused(사용되지 않음)로 변경해야 합니다.</p>
기존 분할 영역을 삭제합니다.	<p>유형을 사용되지 않음으로 변경하여 기존 분할 영역을 삭제할 수 있습니다. 설치 중 분할 영역이 삭제되므로 인접한 분할 영역 크기를 조정하면 해당 공간을 사용할 수 있습니다.</p>

## 텍스트 설치 중 VTOC 슬라이스 설정

SPARC 플랫폼에서 텍스트 설치를 수행하는 경우 설치 도중에 VTOC 슬라이스를 수정할 수 있습니다. x86 플랫폼에서 텍스트 설치를 수행하면 설치 도중 분할 영역이 이미 수정되지 않은 경우 해당 분할 영역 내에서 슬라이스를 수정할 수 있습니다.

VTOC 슬라이스를 설정할 때는 다음 사항을 기억해 두십시오.

- 설치 프로그램에 기존 슬라이스가 표시됩니다. 슬라이스는 배치된 순서대로 표시됩니다. 현재 크기 및 각 슬라이스에 사용 가능한 최대 크기도 표시됩니다.
- Oracle Solaris는 ZFS 루트 풀에 설치되어야 합니다. 기본적으로 루트 풀이 포함된 슬라이스에는 설치 프로그램에서 rpool이라는 레이블을 지정합니다. 루트 풀이 없는 슬라이스에 운영 체제를 설치하려면 설치 프로그램에서 해당 슬라이스의 유형을 rpool로 변경합니다. 설치하는 동안 ZFS 루트 풀이 해당 슬라이스에 만들어집니다.

참고 - ZFS 풀 하나에만 rpool을 지정할 수 있으므로 이름이 rpool인 풀이 이미 장치에 있는 경우 설치 프로그램이 rpool# 형식을 사용하여 새 풀의 이름을 지정합니다.

- 슬라이스의 크기는 사용 가능한 최대 크기까지 늘릴 수 있습니다. 사용 가능한 공간을 더 확보하려면 인접 슬라이스의 유형을 Unused(사용되지 않음)로 변경할 수 있습니다. 그러면 인접한 슬라이스에 사용 가능한 공간을 만들 수 있습니다.
- 슬라이스를 명시적으로 변경하지 않으면 슬라이스의 내용이 설치하는 동안 그대로 유지됩니다.

다음 표에서는 텍스트 설치 중 슬라이스를 수정하는 옵션에 대해 설명합니다.

표 2-3 텍스트 설치 중 VTOC 슬라이스를 수정하는 옵션

옵션	설명 및 사용자 조치(필요할 경우)
기존 슬라이스 사용	이 옵션은 현재 크기를 사용하여 기존 VTOC 슬라이스에 Oracle Solaris 운영 체제를 설치합니다. 대상 슬라이스를 선택한 다음 유형을 rpool로 변경합니다.
슬라이스 크기 조정	새로 만든 rpool 슬라이스의 크기만 변경할 수 있습니다. 필드에 새 크기를 입력합니다.

옵션	설명 및 사용자 조치(필요할 경우)
새 슬라이스 만들기	사용되지 않은 슬라이스를 선택하고 해당 유형을 변경합니다. 예를 들어, 사용되지 않음을 rpool로 변경합니다.
기존 슬라이스 삭제	슬라이스 유형을 Unused(사용되지 않음)로 변경합니다. 설치 중 슬라이스가 삭제되므로 인접한 슬라이스 크기 조정에 해당 공간을 사용할 수 있습니다.

## 적절한 장치 드라이버가 있는지 확인

Oracle Solaris OS를 설치하기 전에 시스템의 장치가 지원되는지 여부를 확인해야 합니다. <http://www.oracle.com/webfolder/technetwork/hcl/index.html>에서 HCL(하드웨어 호환성 목록)을 참조하십시오. HCL은 Oracle Solaris 운영 체제에서 작동한다고 인증되거나 보고된 하드웨어에 대한 정보를 제공합니다.

## Oracle Configuration Manager 사용

이 Oracle Solaris 릴리스에서 대화식 설치 중, Oracle Configuration Manager 및 Oracle Auto Service Request 유틸리티를 시스템에 설치하려는 경우 설치된 시스템에 이러한 서비스를 구성하라는 메시지가 나타납니다.

- Oracle Configuration Manager는 Oracle Support 조직에 시스템의 소프트웨어 구성을 설명하는 데이터를 정기적으로 보냅니다.
- Oracle Auto Service Request는 하드웨어 및 소프트웨어 문제를 지칭하는 FMA(Fault Management Architecture) 이벤트가 발생할 때 Oracle Support 조직에 데이터를 보냅니다.

---

참고 - 모든 데이터는 보안 모드로 전송됩니다.

---

대화식 설치를 수행할 때 다음 옵션이 있습니다.

- 고객 정보를 식별하지 않고 My Oracle Support로 익명 시스템 구성을 보내려면 기본 Support Registration(지원 등록) 설치 프로그램 패널 익명 등록 주소 또는 암호가 없는 다른 전자 메일 주소를 사용합니다.
- My Oracle Support에서 고객 정보를 확인하고 보안 업데이트를 받으려면 Support Configuration(지원 구성) 패널의 익명 전자 메일 주소를 My Oracle Support 로그인 ID로 바꾸고 My Oracle Support 암호를 추가합니다. 이 옵션을 사용하면 Oracle Auto Service Request도 시작됩니다.

고객 구성 데이터가 정기적으로 업로드될 경우 고객 지원 담당자가 이 데이터를 분석하여 향상된 서비스를 제공할 수 있습니다. 예를 들어, 서비스 요청을 작성하면 지원 담당자가

해당 서비스 요청과 구성 데이터를 직접 연관지을 수 있습니다. 그러면 고객 지원 담당자가 사용자의 시스템 목록을 확인하고 이에 따라 문제를 해결할 수 있습니다.

- 데이터를 자동으로 My Oracle Support에 보내지 않으려면 Support Configuration(지원 구성) 패널의 익명 전자 메일 주소를 삭제하고 필드를 비워 둡니다. 연결 해제 모드에서 Oracle Configuration Manager가 시작됩니다. 이 모드에서 Oracle Configuration Manager는 데이터를 전송하기 위해 수동으로 활성화할 수 있습니다. 예를 들어, 기술 지원 담당자가 사용자의 시스템 데이터를 요청할 경우 Oracle Configuration Manager를 수동으로 사용하여 해당 데이터를 제공할 수 있습니다.

Oracle Configuration Manager가 비연결 모드가 아닌 한, 처음 재부트 중 Oracle Configuration Manager 서비스가 실행되고 등록 서버에 시스템을 등록하려고 시도합니다. 이 등록을 성공하면 구성 정보의 업로드가 수행됩니다. 또한 등록 성공 시 내부 스케줄러가 시작됩니다. 그 후에는 구성 데이터가 스케줄러 통제하에 업로드됩니다. 차후 재부트 시에는 구성 데이터가 서비스 시작의 일부로 전송되지 않습니다. 시스템이 이미 등록되었음을 인식하고 간단히 스케줄러를 실행합니다. `/usr/sbin/emCCR`을 사용하여 일정을 조정할 수 있습니다. [emCCR\(1M\)](#) 매뉴얼 페이지와 [Oracle Configuration Manager Installation and Administration Guide](#)를 참조하십시오.

등록을 허용하도록 선택했는지 여부에 관계없이, 지원 서비스를 이용하려면 Oracle Configuration Manager를 사용하여 나중에 시스템을 등록 또는 재등록하도록 선택할 수 있습니다.

다음과 같은 상황에서 등록 또는 재등록하도록 선택할 수 있습니다.

- 이전에 익명으로 등록했습니다.
- 이전에 Oracle Configuration Manager 연결을 해제했습니다.
- My Oracle Support 자격 증명을 입력할 때 Oracle에 연결할 수 없어서 검증받지 못했습니다. 예를 들어, 네트워크 프록시 요구 사항 때문에 자동 등록을 완료할 수 없었습니다.

등록하거나 재등록하려면 대화식 모드에서 `configCCR` 유틸리티(`/usr/sbin/configCCR`)를 사용하십시오. 예를 들어, 다음 명령을 실행하여 기존 구성 사양을 제거합니다.

```
# /usr/lib/ocm/ccr/bin/configCCR -r
```

그리고 다음 명령을 사용하여 Oracle Configuration Manager를 수동으로 구성합니다.

```
# /usr/lib/ocm/ccr/bin/configCCR -a
```

등록을 완료한 후에 다음과 같이 서비스를 사용으로 설정할 수 있습니다.

```
# svcadm enable system/ocm
```

서비스가 사용으로 설정된 후에는 시스템을 재부트할 때 Oracle Configuration Manager 클라이언트가 다시 시작됩니다.

Oracle Configuration Manager 및 Oracle Auto Service Request에 대한 자세한 내용은 다음을 참조하십시오.

- [부록 A. Oracle Configuration Manager 작업](#)
- `configCCR(1M)` 매뉴얼 페이지

- *Oracle Configuration Manager Installation and Administration Guide*
- <http://www.oracle.com/support/policies.html>
- Oracle Auto Service Request 설명서 - <http://www.oracle.com/asr>

## 라이브 매체 사용

---

이 장에서는 라이브 매체 이미지를 사용하여 설치를 수행하는 방법을 설명합니다.

### GUI 설치 프로그램으로 설치

Oracle Solaris 소프트웨어를 설치할 때 다음 정보를 고려하십시오.

- “라이브 매체 및 텍스트 설치의 시스템 요구 사항” [25]을 참조하십시오.
- 라이브 매체 ISO 이미지의 설치 프로그램은 x86 플랫폼에만 사용됩니다.
- 둘 이상의 운영 체제가 설치된 시스템에 Oracle Solaris를 설치하려는 경우 설치 프로세스 중에 디스크를 분할할 수 있습니다.

다음 사항에 유의하십시오.

- 설치 프로그램은 전체 디스크 또는 포맷되지 않은 디스크에 설치할 때 GPT 형식을 사용합니다. 단, 기존 GPT 분할 영역 또는 DOS 분할 영역은 기본적으로 유지되며 설치 프로그램에 표시되므로 기존 분할 영역을 유지하고 해당 분할 영역에 설치할 수 있습니다. 자세한 내용은 “대화식 설치 중 시스템 분할 지침” [27]을 참조하십시오.
- 이 릴리스에서 x86 플랫폼용 Oracle Solaris는 새 버전의 GRand Unified Bootloader(GRUB 2)를 설치합니다. GRUB 2에 대한 자세한 내용은 “Oracle Solaris 11.2 시스템 부트 및 종료”의 “GRUB 2 소개”를 참조하십시오.

원하는 경우 타사 또는 오픈 소스 분할 도구를 사용하여 새 분할 영역을 만들거나 설치 전에 기존 분할 영역을 조정할 수 있습니다. “설치 전 시스템 분할 지침” [26]을 참조하십시오.

- 이 릴리스에서 GUI 설치 프로그램을 사용하여 Oracle Solaris 운영 체제를 iSCSI 대상에 설치할 수 있습니다. iSCSI 대상이 부트 디스크로 작동할 수 있고 시스템에서 iSCSI 부팅에 필요한 조건을 지원해야 합니다.

시스템에서 iSCSI 디스크 자동 검색을 지원하는 경우 설치 프로그램이 해당 옵션을 제공합니다. 다른 방법으로, 설치 화면에서 수동으로 값을 입력하여 iSCSI 대상을 지정할 수 있습니다.

자세한 내용은 이 장의 [GUI 설치 수행 방법 \[35\]](#) 설치 절차를 참조하십시오.

[iscsiadm\(1M\)](#) 매뉴얼 페이지도 참조하십시오.

- GUI 설치 프로그램은 운영 체제를 업그레이드할 수 없습니다. 그러나 Oracle Solaris 운영 체제를 설치한 후에는 이미지 패키징 시스템을 사용하여 시스템에서 업데이트를 사용

할 수 있는 모든 패키지를 업데이트할 수 있습니다. “Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”를 참조하십시오.

- GUI 설치 프로그램은 전체 디스크에 또는 디스크의 Oracle Solaris x86 분할 영역에 초기 설치를 수행할 수 있습니다.



---

주의 - 설치 시 대상 장치의 모든 소프트웨어와 데이터를 덮어씹습니다.

---

## GUI 설치 프로그램의 기본 설정

라이브 매체의 GUI 설치 프로그램에서 사용되는 기본 네트워크 및 보안 설정은 다음과 같습니다.

- Oracle Solaris는 DNS(도메인 이름 시스템) 해석과 함께 DHCP를 사용하여 자동으로 네트워크에 연결됩니다.  
DNS 도메인 및 서버 인터넷 프로토콜(Internet Protocol, IP) 주소는 DHCP 서버에서 검색됩니다.
- 자동 네트워킹으로 활성 인터페이스에 IPv6 자동 구성이 사용으로 설정됩니다.
- NFSv4 도메인은 동적으로 파생됩니다.

## ▼ GUI 설치를 준비하는 방법

GUI 설치를 수행하기 전에 이 절차의 작업을 완료하십시오.

1. 라이브 매체가 없는 경우 라이브 매체 ISO 이미지를 다운로드합니다.

Oracle Solaris 라이브 매체 ISO 이미지를 다운로드하려면 <http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html>로 이동합니다.

---

참고 - 다른 방법으로, USB 플래시 드라이브에 이미지를 구우려면 USB 이미지를 다운로드합니다.

---

이미지를 다운로드한 후 CD, DVD 또는 USB 플래시 드라이브와 같은 이동식 매체에 이미지를 복사합니다.

---

참고 - USB 이미지의 경우, USB 플래시 드라이브에 이미지를 복사하려면 `usbcopy` 유틸리티가 필요합니다. 이 유틸리티를 시스템에 추가하려면 `pkg:/install/distribution-creator` 패키지를 설치합니다.

---

2. 시스템에서 설치 프로그램을 실행하는 데 필요한 요구 사항과 제한 사항을 확인합니다.

- a. 시스템이 필요한 시스템 요구 사항을 모두 충족하는지 확인합니다.  
“라이브 매체 및 텍스트 설치의 시스템 요구 사항” [25]을 참조하십시오.
- b. 필요한 장치 드라이버가 모두 있는지 확인합니다.  
“적절한 장치 드라이버가 있는지 확인” [30]을 참조하십시오.
3. 여러 운영 체제를 설치하려면 필요한 환경을 설정합니다.
  - a. “다중 운영 체제 설치를 위한 시스템 준비” [25]에서 사양을 검토합니다.
  - b. 시스템을 백업합니다.
  - c. 설치 전에 시스템을 분할하려면 “시스템 분할 영역” [26]을 참조하십시오.

다음 순서 [GUI 설치 수행 방법 \[35\]](#)을 참조하십시오.

## ▼ GUI 설치 수행 방법

1. 설치 매체를 삽입하고 시스템을 부트합니다.  
GRUB2 메뉴가 표시될 때 다른 옵션을 선택하지 않으면 기본 항목이 자동으로 사용됩니다.

---

참고 - 시스템의 그래픽 카드가 라이브 매체에서 지원되지 않거나 시스템에 그래픽 카드가 없는 경우 매체를 삽입하면 시스템이 콘솔 모드로 부트됩니다. 이런 경우 GUI 설치를 수행할 수 없습니다. “시스템이 콘솔 모드로 부트하는 경우 수행할 작업” [39]을 참조하십시오.

---

- 로그인하라는 메시지가 나타나면 사용자 이름과 암호에 모두 jack을 입력합니다.
- root 암호는 solaris입니다.

2. 키보드 및 언어를 선택하거나 기본 영어 옵션을 적용합니다.

---

참고 - 언어 및 키보드 선택은 설치 프로그램과 설치된 시스템의 기본값으로 설정됩니다. 설치된 시스템의 로그인 패널에서 로케를 수정할 수 있습니다.

---

3. 설치에 필요하지만 누락된 드라이버를 모두 설치합니다.  
라이브 매체를 부트할 때 드라이버가 누락되어 있으면 프롬프트가 표시됩니다. Device Driver Utility에 액세스하는 지침에 따라 설치에 필요한 드라이버를 찾아 설치합니다.
4. 라이브 매체 데스크탑에서 Oracle Solaris 설치 아이콘을 두 번 눌러 GUI 설치 프로그램을 시작합니다.

5. 시작 패널에서 다음을 선택합니다.
6. Disk Discovery(디스크 검색) 패널에서 설치 프로그램이 검색할 디스크 유형을 선택합니다.
  - Local Disks(로컬 디스크) - 내부 및 외부 하드 디스크를 포함하여 컴퓨터에 연결된 디스크의 기본 옵션입니다.
  - iSCSI - 설치 프로그램이 iSCSI 표준을 사용하여 네트워크를 통해 액세스할 수 있는 원격 디스크를 검색하려면 이 옵션을 선택합니다. 다음과 같은 추가 필드가 표시됩니다.
    - Use DHCP autodiscovery(DHCP 자동 검색 사용) - 시스템에서 iSCSI 디스크 자동 검색을 지원하는 경우 이 옵션이 사용으로 설정됩니다. 이 옵션을 선택하면 자동 검색으로 반환된 값이 조건 필드에 채워집니다. 이러한 값을 더 세분화하려면 Specify search criteria 옵션을 선택할 수 있습니다.
    - Specify search criteria(검색 조건 지정) - 이 옵션을 선택하고 iSCSI 검색 값을 수동으로 제공할 수 있습니다.

Target IP(대상 IP)	iSCSI 대상의 IP 주소입니다. 0-255 범위의 숫자를 제공합니다. 이 IP 주소의 시스템은 온라인 상태이고 액세스할 수 있어야 합니다. 필수 필드입니다.
LUN	제공된 IP 주소에 있는 iSCSI 장치의 논리 장치 번호입니다. LUN은 0, 1과 같은 숫자 값인 경우가 많습니다. 선택적 필드입니다.
Target Name(대상 이름)	IQN(iSCSI Qualified Name) 형식의 iSCSI 대상 이름입니다. 선택적 필드입니다.
Port(포트)	제공된 IP 주소와 함께 iSCSI 장치 검색에 사용되는 포트 번호입니다. 기본값 3260은 일반적으로 iSCSI에 사용되는 포트입니다. 선택적 필드입니다.
Initiator Name(개시자 이름)	iSCSI 검색 세션에 설정할 개시자 노드 이름입니다. iSCSI 부팅의 경우 개시자 노드 이름을 수정할 수 없으므로 이 필드가 숨겨집니다. 선택적 필드입니다.
Use CHAP(CHAP 사용)	CHAP(Challenge-Handshake Authentication Protocol) 인증 세부 정보를 입력하려면 이 옵션을 선택합니다.
Name(이름)	인증에 사용할 CHAP 이름입니다. 선택적 필드입니다.
Password(암호)	인증을 위한 CHAP 보안 값입니다. 제공된 경우 이 값은 12-16자 사이여야 합니다. 선택적 필드입니다.

iSCSI 옵션을 선택할 경우 제공된 세부 정보가 검증되는 동안 Next를 선택할 때 지연이 발생할 수 있습니다. iSCSI LUN을 검색할 수 없을 경우 오류가 표시됩니다. 유효한 조건을 입력하거나 iSCSI의 선택을 해제하여 문제를 해결할 때까지는 작업을 계속할 수 없습니다.

7. **Disk Selection(디스크 선택) 패널에서 여러 설치 대상이 표시된 경우 설치 대상을 선택하거나 기본값을 적용합니다. 그런 다음, 운영 체제를 전체 디스크에 또는 디스크의 분할 영역에 설치할지 여부를 지정합니다.**

설치 프로그램은 전체 디스크 또는 포맷되지 않은 디스크에 설치할 때 GPT 형식을 사용합니다. 단, 기존 GPT 분할 영역 또는 DOS 분할 영역은 기본적으로 유지되며 설치 프로그램에 표시되므로 기존 분할 영역을 유지하고 해당 분할 영역에 설치할 수 있습니다.

다음 사항에 유의하십시오.

- 디스크에 기존 DOS 분할 영역이 있을 경우 최대 4개의 DOS 기본 분할 영역이 표시됩니다. DOS 확장 분할 영역이 존재하는 경우 논리적 분할 영역도 확장된 분할 영역 내에 디스크 레이아웃 순서로 표시됩니다. Solaris 분할 영역은 하나만 허용되며 해당 Solaris 분할 영역을 설치에 사용해야 합니다. Solaris 분할 영역은 확장된 분할 영역 내 논리적 분할 영역일 수 있습니다.
- 디스크에 기존 GPT 분할 영역이 있을 경우 GPT 분할 영역이 표시됩니다. 최대 7개의 GPT 분할 영역이 지원됩니다. 설치 중 하나 이상의 Solaris 분할 영역을 만들 수 있지만 하나의 Solaris 분할 영역을 설치 대상으로 선택해야 합니다. 기존 Solaris GPT 분할 영역이 여러 개인 경우 적합한 첫번째 Solaris GPT 분할 영역이 기본적으로 설치 대상으로 선택됩니다.

분할 영역 레이아웃을 수정할 수 있는 옵션이 있습니다. 지침은 [“대화식 설치 중 시스템 분할 지침” \[27\]](#)을 참조하십시오.

이 설치 단계 중 언제든지 원래 설정으로 되돌릴 수 있습니다.



주의 - 기존 분할 영역 테이블을 읽을 수 없는 경우 패널에 분할 제안이 표시됩니다. 이 경우 설치하는 동안 디스크의 모든 데이터가 삭제됩니다.

8. **대상 시간대를 선택하고 현재 로컬 시간과 일치하도록 날짜와 시간을 조정합니다.**

설치 프로그램은 가능한 경우 시스템 내부 설정의 시간대를 초기 기본값으로 사용합니다. 맵에서 위치를 선택하면 설치 프로그램이 해당 정보를 사용하여 날짜, 시간 및 시간대를 설정합니다.

9. **사용자 설정을 완료합니다.**

- 사용자 이름과 암호를 입력합니다.

사용자 계정 설정을 완료하려면 로그인 이름과 암호를 제공해야 합니다. 로그인 이름은 문자로 시작해야 하며 문자와 숫자만 포함할 수 있습니다.

참고 - 내가 만드는 사용자 계정은 관리 권한을 갖습니다.

설치된 시스템에서 초기 root 암호의 기본값은 여기서 제공한 사용자 계정 암호가 됩니다. 처음으로 root 암호를 사용하면 암호를 변경하라는 메시지가 표시됩니다.

- 컴퓨터 이름을 입력하거나 기본값을 적용합니다. 이 필드는 비워 둘 수 없습니다.

**10. Support Configuration(지원 구성) 패널에서 OCM 및 ASR에 대해 등록을 구성하는 방법을 결정합니다.**

기본 Support Configuration(지원 구성) 설치 프로그램 패널은 익명 등록 주소를 제공합니다. 이 익명 주소를 암호 없이 사용할 경우 MOS(My Oracle Support)는 설치된 시스템의 구성에 대한 정보를 받지만 Oracle Support 조직에 시스템 구성을 업로드할 때 고객 정보는 전혀 받지 않습니다.

다른 방법으로, 다음과 같이 보안 업데이트에 등록하거나 OCM 연결을 해제할 수 있습니다.

- Support Configuration(지원 구성) 패널의 익명 전자 메일 주소를 My Oracle Support 로그인 ID로 바꾸고 My Oracle Support 암호를 추가할 수 있습니다. My Oracle Support에서 고객 정보를 보고 보안 업데이트를 받고자 하는 경우 이 옵션을 사용하십시오. 이 옵션을 사용하면 ASR도 시작됩니다.
- Support Configuration(지원 구성) 패널의 익명 전자 메일 주소를 삭제하고 해당 필드를 비워 둘 경우 OCM은 연결 해제 모드로 시작됩니다. My Oracle Support에 데이터가 전송되지 않습니다. 또는 익명 전자 메일 주소를 삭제하고 MOS 로그인 ID 이외의 다른 전자 메일 주소로 바꿀 경우 OCM은 비인증 모드로 Oracle Support에 데이터를 전송합니다.

자세한 내용은 [“Oracle Configuration Manager 사용” \[30\]](#)을 참조하십시오.

**11. 설치 사양을 검토합니다.**

설치 요약 패널에서 사양을 검토합니다. 필요한 경우 뒤로 이동하여 설치를 시작하기 전에 필요에 따라 변경합니다.

**12. 제공한 사양을 사용하여 시스템을 설치합니다.**

Oracle Solaris 설치 프로세스가 시작됩니다.



**주의** - 설치가 진행되는 동안 중단하지 마십시오. 설치가 완료되지 않으면 디스크가 불확정 상태로 유지될 수 있습니다.

**13. 설치 로그를 검토합니다.**

설치 결과 패널에서 설치 로그에 액세스하여 검토할 수 있습니다.

**14. 시스템을 재부트하거나 설치 프로그램을 끝내고 시스템을 종료합니다.**

성공적으로 설치된 후에는 시스템을 재부트하거나 설치 프로그램을 끝내고 시스템을 종료합니다.

다음 시스템 부트가 시작되면 매체를 꺼냅니다. 또는 GRUB 메뉴에서 Boot from Hard Disk 옵션을 선택합니다.

설치에 실패하면 설치 로그를 확인하고 설치 프로그램을 종료할 수 있습니다.

## 시스템이 콘솔 모드로 부트하는 경우 수행할 작업

시스템의 그래픽 카드가 라이브 매체에서 지원되지 않거나 시스템에 그래픽 카드가 없는 경우 매체를 삽입하면 시스템이 콘솔 모드로 부트됩니다. 이런 경우 GUI 설치를 수행할 수 없습니다.

다음 두 가지 대안이 있습니다.

- 라이브 매체 ISO 이미지 대신 텍스트 설치 프로그램 이미지를 사용합니다.  
네트워크에 액세스하지 않고 로컬 콘솔에서 텍스트 설치 프로그램을 실행할 수 있습니다. [4장. 텍스트 설치 프로그램 사용](#)을 참조하십시오.
- 시스템이 콘솔 모드로 부트되는 경우 라이브 매체에서 Oracle Solaris를 설치하는 방법 [39]에 설명된 대로 원격 설치를 수행합니다.

---

참고 - 이 옵션을 사용하는 경우 텍스트 설치 프로그램 이미지를 다운로드할 필요가 없습니다. 그러나 이 옵션을 사용하려면 원격 ssh 액세스 및 X 서버가 실행 중인 대상 시스템이 필요합니다.

---

### ▼ 시스템이 콘솔 모드로 부트되는 경우 라이브 매체에서 Oracle Solaris를 설치하는 방법

**시작하기 전에** 이 절차에서는 네트워크로 연결된 두 개의 시스템, 즉 라이브 매체가 부트된 시스템(대상 시스템) 및 설치를 수행할 원격 시스템이 필요합니다. 두 시스템은 모두 네트워크에 액세스할 수 있어야 하지만 두 시스템이 동일한 서브넷에 있을 필요는 없습니다. 그러나 대상 시스템은 원격 시스템에서 연결 가능해야 합니다. 또한 원격 시스템은 그래픽 데스크탑을 지원하는 OS를 실행 중이어야 합니다.

1. 설치할 시스템에 매체를 삽입한 후 시스템을 부트합니다.
2. 콘솔 로그인 시 기본 로그인과 암호를 입력합니다.  
Oracle Solaris의 기본 사용자 로그인 및 암호는 jack입니다.
3. root 사용자로 전환합니다.  

```
$ su root
Password: solaris
```

 root 암호는 solaris입니다.
4. ssh 원격 로그인 프로그램에 대한 서비스를 사용으로 설정합니다.

```
# svcadm enable ssh:default
```

5. DHCP에서 대상 시스템에 지정한 IP 주소를 표시합니다.

```
# ifconfig -a
```

6. 원격 시스템에서 터미널 창을 열고 다음을 입력합니다.

```
$ ssh -X IP-address-of-target -l jack
```

여기서 *IP-address-of-target*은 대상 시스템에서 실행한 `ifconfig -a` 명령에 대한 출력입니다.

원격 시스템에서 이 명령을 실행하면 보안 셸이 열려 GUI 설치 프로그램을 사용할 수 있도록 대상 시스템에 액세스할 수 있게 됩니다.

7. **root** 역할을 맡습니다.

```
$ su root  
Password: solaris
```

8. GUI 설치 프로그램을 실행합니다.

```
# /usr/bin/gui-install
```

---

참고 - 이 방식으로는 설치 프로그램 그래픽 디스플레이가 완벽하지 않을 수 있습니다.

---

9. 설치가 완료되면 대상 시스템을 재부트합니다.

## 라이브 매체 설치 후 소프트웨어 추가

운영 체제를 설치한 후 소프트웨어 패키지를 추가하려면 [pkg\(1\)](#) 매뉴얼 페이지에 설명된 대로 `pkg` 명령을 사용합니다. 또는 Oracle Solaris Package Manager GUI 도구를 사용하여 추가 소프트웨어를 설치할 수 있습니다. 데스크탑 메뉴에서 System(시스템)->Administration(관리)->Package Manager(패키지 관리자)로 이동합니다.

---

참고 - 패키지 설치, 업데이트 및 제거 작업을 수행하려면 높은 수준의 권한이 필요합니다. 자세한 내용은 [“Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”](#)의 [“설치 권한”](#)을 참조하십시오.

---

`pkg` 명령 또는 Package Manager 도구를 사용하여 설치할 패키지의 이름을 찾아서 패키지에 대한 정보를 얻고 패키지를 설치할 수 있습니다.

선택적으로 새 부트 환경으로 설치할 수 있습니다. 그러면 새 설치에 문제가 있을 경우 현재 이미지를 계속 사용할 수 있습니다.

먼저 `pkg install` 명령과 함께 `-nv` 옵션을 사용하여 실제로 패키지를 설치하기 전에 패키지 설치가 어떻게 보이는지 확인해야 합니다. 설치할 패키지를 식별하고 `pkg install` 명령을 `-nv` 옵션과 함께 사용하여 얻은 출력을 확인한 후 다음 예제와 비슷한 명령을 실행하여 추가 소프트웨어를 설치합니다.

```
# pkg install --be-name new-BE-name package-name
```

이 샘플 명령은 새 부트 환경을 만드는 데 필요한 옵션을 포함하며 설치할 패키지를 지정합니다.

GUI 데스크탑이 없는데 Oracle Solaris Desktop을 설치하려면 `solaris-desktop` 패키지를 설치합니다.



# ◆◆◆ 4 장

## 텍스트 설치 프로그램 사용

---

개별 SPARC 및 x86 클라이언트 시스템에 대화식 텍스트 설치를 수행할 수 있습니다. 추가적으로, 네트워크를 자동 설치용으로 설정한 경우 네트워크를 통해 텍스트 설치를 수행할 수 있습니다.

### 텍스트 설치 프로그램으로 설치

Oracle Solaris 운영 체제를 설치할 때 다음 정보를 고려하십시오.

- [“라이브 매체 및 텍스트 설치의 시스템 요구 사항” \[25\]](#)을 참조하십시오.
- 둘 이상의 운영 체제가 설치된 x86 기반 시스템에 Oracle Solaris를 설치하려는 경우 설치 프로세스 중에 디스크를 분할할 수 있습니다.
  - 설치 프로그램은 전체 디스크 또는 포맷되지 않은 디스크에 설치할 때 GPT 형식을 사용합니다. 단, 기존 GPT 분할 영역 또는 DOS 분할 영역은 기본적으로 유지되며 설치 프로그램에 표시되므로 기존 분할 영역을 유지하고 해당 분할 영역에 설치할 수 있습니다. 자세한 내용은 [“대화식 설치 중 시스템 분할 지침” \[27\]](#)을 참조하십시오.
  - 이 릴리스에서 Oracle Solaris 설치 프로그램은 x86 시스템용 GRUB 2를 사용합니다. GRUB 2는 하나 이상의 드라이브에 여러 개의 운영 체제 부트를 지원합니다. GRUB 2에 대한 자세한 내용은 [“Oracle Solaris 11.2 시스템 부트 및 종료”의 “GRUB 2 소개”](#)를 참조하십시오.

또한 오픈 소스 또는 타사 분할 도구를 사용하여 새 분할 영역을 만들거나 설치 전에 기존 분할 영역을 조정할 수 있는 옵션이 있습니다. [“설치 전 시스템 분할 지침” \[26\]](#)을 참조하십시오.

- Oracle Solaris 설치 프로그램은 운영 체제를 업그레이드할 수 없습니다. 그러나 Oracle Solaris 운영 체제를 설치한 후에는 이미지 패키징 시스템을 사용하여 시스템에서 업데이트를 사용할 수 있는 모든 패키지를 업데이트할 수 있습니다. [“Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”](#)를 참조하십시오.
- 텍스트 설치 프로그램을 사용하여 Oracle Solaris 운영 체제를 iSCSI 대상에 설치할 수 있습니다. iSCSI 대상이 부트 디스크로 작동할 수 있고 시스템에서 iSCSI 부트에 필요한 조건을 지원해야 합니다. 시스템에서 iSCSI 디스크 자동 검색을 지원하는 경우 설치 프로그램이 해당 옵션을 제공합니다. 다른 방법으로, 설치 화면에서 수동으로 값을 입력하여

iSCSI 대상을 지정할 수 있습니다. iSCSI를 사용하려면 설치 프로세스를 시작하기 전에 시스템의 네트워크 인터페이스가 정적 IP 주소로 구성되어 있어야 합니다.

자세한 내용은 [텍스트 설치 수행 방법 \[46\]](#)을 참조하십시오. [iscsiadm\(1M\)](#) 매뉴얼 페이지도 참조하십시오.

- 텍스트 설치 프로그램은 전체 디스크, Oracle Solaris x86 분할 영역 또는 SPARC 슬라이스에 초기 설치를 수행할 수 있습니다.



주의 - 설치 시 대상 장치의 모든 소프트웨어와 데이터를 덮어씁니다.

- 라이브 매체에는 데스크탑 또는 랩탑에 적절한 소프트웨어 세트가 들어 있습니다. 텍스트 설치 프로그램은 범용 서버 시스템에 적절한 더 작은 소프트웨어 세트를 설치합니다. 특히 텍스트 설치 프로그램을 사용하면 GNOME 데스크탑이 설치되지 않습니다. 텍스트 설치 프로그램으로 설치를 수행한 후 추가 패키지를 설치하려면 [“텍스트 설치 후 소프트웨어 추가” \[52\]](#)를 참조하십시오.

## 텍스트 설치 프로그램의 네트워킹 구성

텍스트 설치 프로그램의 네트워킹 패널에 다음 옵션이 제공됩니다.

- Automatically(자동) - 라이브 매체 설치 프로그램 방식과 비슷하게 자동 NCP(네트워크 구성 프로파일)로 대상 시스템을 구성합니다.
- Manually(수동) - DefaultFixed NCP를 선택하고 네트워크 인터페이스(NIC) 하나의 정적 IPv4 구성을 제공합니다. 해당 NIC에 대해 IPv4 기본 경로와 IPv6 자동 구성이 사용으로 설정됩니다. 이름 지정 서비스의 수동 구성에도 이 옵션을 사용할 수 있습니다.
- None(없음) - DefaultFixed NCP를 선택하고 루프백 인터페이스만 구성합니다.

## 텍스트 설치 프로그램 내에서 이동하는 방법

각 패널의 맨 아래에 나열된 기능 키를 사용하여 다른 패널로 이동할 수 있습니다. 화살표 키를 사용하여 지정된 패널의 필드 간에 이동합니다. 키보드에 기능 키가 없거나 키가 응답하지 않는 경우 ESC 키를 누르면 탐색을 위한 대체 키가 표시됩니다.

설치하는 동안 언제든지 이전 패널로 돌아갈 수 있습니다.

## 텍스트 설치 작업

이 절에서는 다음과 같은 작업을 다룹니다.

- [텍스트 설치를 준비하는 방법 \[45\]](#)

- 텍스트 설치 수행 방법 [46]
- 네트워크를 통해 텍스트 설치를 시작하는 방법 [51]
- “텍스트 설치 후 소프트웨어 추가” [52]

## ▼ 텍스트 설치를 준비하는 방법

텍스트 설치를 수행하기 전에 이 절차의 작업을 완료하십시오.

1. 텍스트 설치 프로그램 이미지가 없으면 이미지를 다운로드합니다.

Oracle Solaris 텍스트 설치 프로그램 ISO 이미지를 다운로드하려면 <http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html>로 이동합니다.

---

참고 - USB 플래시 드라이브에 이미지를 구우려면 USB 이미지를 다운로드합니다.

---

이미지를 다운로드한 후 CD, DVD 또는 USB 플래시 드라이브와 같은 이동식 매체에 이미지를 복사합니다.

---

참고 - USB 이미지의 경우, USB 플래시 드라이브에 이미지를 복사하려면 `usbcopy` 유틸리티가 필요합니다. 이 유틸리티를 시스템에 추가하려면 `pkg:/install/distribution-creator` 패키지를 설치합니다.

---

2. 시스템에서 설치 프로그램을 실행하는 데 필요한 요구 사항과 제한 사항을 확인합니다.

- a. 시스템이 필요한 시스템 요구 사항을 모두 충족하는지 확인합니다.  
“라이브 매체 및 텍스트 설치의 시스템 요구 사항” [25]을 참조하십시오.
- b. 필요한 장치 드라이버가 모두 있는지 확인합니다.  
“적절한 장치 드라이버가 있는지 확인” [30]을 참조하십시오.

3. 여러 운영 체제를 설치하려면 필요한 환경을 설정합니다.

- a. “다중 운영 체제 설치를 위한 시스템 준비” [25]에서 사양을 검토합니다.
- b. 시스템을 백업합니다.
- c. 설치 전에 시스템을 분할하려면 2장. 설치 준비에서 지침을 검토합니다.  
특히 분할 영역이나 슬라이스에 Oracle Solaris를 설치하고 설정하려는 경우 아직 작업을 수행하지 않았으면 “설치 전 시스템 분할 지침” [26]의 정보를 검토합니다.

다음 순서 [텍스트 설치 수행 방법 \[46\]](#)을 참조하십시오.

## ▼ 텍스트 설치 수행 방법

1. 텍스트 설치 매체를 삽입하고 시스템을 부트합니다. 요청할 경우 예비 키보드 및 언어를 선택합니다.

x86 설치 프로세스 중에 키보드 및 언어 선택을 요청합니다. SPARC 설치 프로세스에 대해서는 이 값이 사전 설정됩니다.

---

참고 - 언어 및 키보드 선택은 설치 프로그램과 설치된 시스템의 기본값으로 설정됩니다.

---

2. (선택 사항) 필수 드라이버를 설치하려면 설치 메뉴에서 옵션 2를 선택합니다.

Device Driver Utility 사용에 대한 지침은 [Device Driver Utility 시작 방법 \[251\]](#)을 참조하십시오. 드라이버를 설치한 후에 텍스트 설치를 다시 시작하고 설치 메뉴로 돌아갑니다.

3. (선택 사항) iSCSI 디스크 검색을 사용하려면 옵션 3을 선택합니다.

셸 프롬프트에서 단계에 따라 네트워크 인터페이스를 구성합니다. 자세한 내용은 “[Oracle Solaris 11.2 네트워크 구성 요소의 구성 및 관리](#)”의 3 장, “[Oracle Solaris에서 IP 인터페이스와 주소 구성 및 관리](#)”를 참조하십시오. 네트워크 인터페이스를 구성한 후 Control-D를 눌러 셸을 종료합니다.

4. 설치 메뉴에서 첫번째 옵션을 선택하여 설치를 시작합니다.

```
Welcome to the Oracle Solaris 11.2 installation menu
```

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot
```

```
Please enter a number [1]:
```

"Continue" 기능 키를 사용하여 다음 패널로 이동합니다.

---

참고 - 키보드를 사용하여 설치 프로그램 패널을 탐색합니다. 마우스는 사용할 수 없습니다. 각 패널에 나열된 키 명령을 확인하고, 자세한 내용은 온라인 도움말을 참조하십시오.

---

5. Discovery Selection(검색 선택) 패널에서 시스템을 설치할 디스크의 검색 방법을 선택합니다.

- Local Disks(로컬 디스크) - 내부 및 외부 하드 디스크를 포함하여 컴퓨터에 연결된 디스크의 기본 옵션입니다.

- iSCSI - 설치 프로그램이 iSCSI 표준을 사용하여 네트워크를 통해 액세스할 수 있는 원격 디스크를 검색하려면 이 옵션을 선택합니다. iSCSI를 사용하려면 3단계를 완료해야 합니다. 추가 패널에서 다음 정보를 묻습니다.

Target IP(대상 IP)	iSCSI 대상의 IP 주소입니다. 0-255 범위의 숫자 4개를 제공합니다. 이 IP 주소의 시스템은 온라인 상태이고 액세스할 수 있어야 합니다. 필드입니다.
Target LUN(대상 LUN)	제공된 IP 주소에 있는 iSCSI 장치의 논리 장치 번호입니다. LUN은 0, 1과 같은 숫자 값인 경우가 많습니다. 선택적 필드입니다.
Target Name(대상 이름)	IQN(iSCSI Qualified Name) 형식의 iSCSI 대상 이름입니다. 선택적 필드입니다.
Port(포트)	제공된 IP 주소와 함께 iSCSI 장치 검색에 사용되는 포트 번호입니다. 기본값 3260은 일반적으로 iSCSI에 사용되는 포트입니다. 선택적 필드입니다.
Initiator Name(개시자 이름)	iSCSI 검색 세션에 설정할 개시자 노드 이름입니다. iSCSI 부팅의 경우 개시자 노드 이름을 수정할 수 없으므로 이 필드가 숨겨집니다. 이 필드는 사용자에게 대해 생성됩니다.
CHAP Name(CHAP 이름)	인증에 CHAP를 사용하는 경우 인증에 사용할 CHAP(Challenge-Handshake 인증 프로토콜) 이름입니다. 선택적 필드입니다.
CHAP Password(CHAP 암호)	인증을 위한 CHAP 보안 값입니다. 제공된 경우 이 값은 12-16자 사이여야 합니다. 선택적 필드입니다.

iSCSI 옵션을 선택할 경우 제공된 세부 정보가 검증되는 동안 Next를 선택할 때 지연이 발생할 수 있습니다. iSCSI LUN을 검색할 수 없을 경우 오류가 표시됩니다. 유효한 조건을 입력하거나 iSCSI의 선택을 해제하여 문제를 해결할 때까지는 작업을 계속할 수 없습니다.

6. **Disks(디스크) 패널에서 OS를 설치할 디스크를 선택합니다.**  
대상 디스크가 2개 이상 나열될 경우 디스크 하나를 선택하거나 기본값을 적용합니다.
7. **Partitions(분할 영역) 패널에서 운영 체제를 전체 디스크에 설치할지 디스크의 일부분에 설치할지 선택합니다.**  
다음 중에서 선택할 수 있습니다.
  - Use the entire disk(전체 디스크 사용)
  - Use a GPT partition(GPT 분할 영역 사용)

---

참고 - SPARC 설치 중에 분할 영역이 아닌 슬라이스에 대한 정보를 묻는 메시지가 나타납니다.

---

8. (옵션) Partition Selection(분할 영역 선택) 패널에서 분할 영역 레이아웃을 수정합니다. 설치 패널을 완성하면서 언제든지 원래 설정으로 되돌릴 수 있습니다.



주의 - 기존 분할 영역 테이블을 읽을 수 없는 경우 패널에 분할 제안이 표시됩니다. 이 경우 설치하는 동안 디스크의 모든 데이터가 삭제됩니다.

설치 프로그램은 전체 디스크 또는 포맷되지 않은 디스크에 설치할 때 GPT 형식을 사용합니다. 단, 기존 GPT 분할 영역 또는 DOS 분할 영역은 기본적으로 유지되며 설치 프로그램에 표시되므로 기존 분할 영역을 유지하고 해당 분할 영역에 설치할 수 있습니다.

다음 사항에 유의하십시오.

- 디스크에 기존 DOS 분할 영역이 있을 경우 최대 4개의 DOS 기본 분할 영역이 표시됩니다. DOS 확장 분할 영역이 존재하는 경우 논리적 분할 영역도 확장된 분할 영역 내에 디스크 레이아웃 순서로 표시됩니다. Solaris 분할 영역은 하나만 허용되며 해당 Solaris 분할 영역을 설치에 사용해야 합니다. Solaris 분할 영역은 확장된 분할 영역 내 논리적 분할 영역일 수 있습니다.
- 디스크에 기존 GPT 분할 영역이 있을 경우 GPT 분할 영역이 표시됩니다. 최대 7개의 GPT 분할 영역이 지원됩니다. 설치 중 하나 이상의 Solaris 분할 영역을 만들 수 있지만 하나의 Solaris 분할 영역을 설치 대상으로 선택해야 합니다. 기존 Solaris GPT 분할 영역이 여러 개인 경우 적합한 첫번째 Solaris GPT 분할 영역이 기본적으로 설치 대상으로 선택됩니다.

SPARC 설치 프로세스에서 디스크 슬라이스에 대한 정보를 묻습니다.

자세한 분할 지침은 “대화식 설치 중 시스템 분할 지침” [27]을 참조하거나 설치 프로그램의 온라인 도움말을 참조하십시오.

9. System Identity(시스템 ID) 패널에서는 네트워크에서 시스템을 식별할 컴퓨터 이름을 제공합니다.
10. Network(네트워크) 패널에서 유선 이더넷 네트워크 연결을 구성하는 방법을 지정합니다.
- 설치 중 네트워크가 구성되지 않도록 지정하려면 None(없음)을 선택합니다. 설치 프로그램이 시간대 패널로 진행됩니다.
  - DHCP를 사용하여 네트워크 연결을 구성하려면 Automatically(자동)를 선택합니다. 설치 프로그램이 시간대 패널로 진행됩니다.
  - 네트워킹 사양을 제공하려면 Manually(수동)를 선택하고 다음과 같이 계속합니다.
    - a. 인터페이스가 여러 개 있는 경우 구성할 연결을 선택합니다.
    - b. Manually Configure(수동 구성) 패널에서 연결 설정을 입력하거나 설치 프로그램에서 찾아 제공한 기본 정보를 적용합니다.

---

참고 - IP 주소 및 넷마스크는 필수 필드입니다. 라우터는 선택적 필드입니다.

---

- c. DNS Name Service(DNS 이름 서비스) 패널에서 시스템에 DNS 이름 서비스를 사용하도록 선택할 경우 다음을 수행합니다.
  - i. DNS Server Addresses(DNS 서버 주소) 패널에서 DNS 서버의 IP 주소를 하나 이상 입력합니다.
  - ii. DNS Search List(DNS 검색 목록) 패널에서 DNS 질의를 만들 때 검색할 도메인 이름을 하나 이상 제공합니다.
- d. Alternate Name Service(대체 이름 서비스) 패널에서 시스템에 LDAP 이름 서비스, NIS 이름 서비스 또는 None(없음)을 사용할지 여부를 지정합니다.
  - 이전 단계에서 DNS를 선택한 경우 LDAP 또는 NIS가 DNS와 더불어 대체 이름 서비스로 설정됩니다.
  - 이전 단계에서 DNS를 선택하지 않은 경우 LDAP 또는 NIS가 유일한 이름 서비스로 설정됩니다.
  - LDAP 프로파일을 사용하지 않고 시스템에 LDAP를 구성하려면 LDAP 대신 None을 선택합니다. 그런 다음, 설치가 완료된 후 수동으로 LDAP를 구성합니다.
  - 선택된 네트워크 이름 지정 서비스가 없으면 /etc/hosts와 같은 표준 이름 소스 파일을 사용하여 네트워크 이름을 분석할 수 있습니다. 자세한 내용은 [nsswitch.conf\(4\)](#) 매뉴얼 페이지를 참조하십시오.
- e. Domain Name(도메인 이름) 패널에서 선택한 대체 이름 서비스에 대해 시스템이 상주하는 도메인을 제공합니다.

---

참고 - 도메인 이름을 확인하려면 시스템 관리자에게 문의하십시오. 또는 이전에 설치된 시스템에 `domainname` 명령을 사용하십시오.

---

- f. LDAP Profile(LDAP 프로파일) 패널에서는 Alternate Name Service(대체 이름 서비스) 패널에서 LDAP를 선택한 경우 다음과 같이 LDAP 구성 사양을 제공합니다.
  - 시스템에서 LDAP 이름 서비스를 구성하는 데 사용할 LDAP 프로파일
  - LDAP 프로파일 서버의 IP 주소
  - LDAP 검색 기준
  - LDAP Proxy(LDAP 프록시) 패널에서 LDAP 프록시 바인드 정보를 제공할지 여부를 지정합니다.  
필요할 경우 LDAP 프록시 바인드 고유 이름 및 프록시 바인드 암호를 제공합니다.

- g. NIS Name Server(NIS 이름 서버) 패널에서는 Alternate Name Service(대체 이름 서비스) 패널에서 NIS를 선택한 경우 NIS 사양을 제공합니다.

소프트웨어에서 이름 서버를 검색하도록 하거나, 직접 이름 서버를 지정할 수 있습니다. 다음 둘 중 하나를 선택합니다.

- Find One(이름 서버 찾기)

---

참고 - 이름 서버가 로컬 서브넷에 있는 경우에만 소프트웨어가 해당 서버를 찾을 수 있습니다.

---

- Specify One(이름 서버 지정) - 이름 서버의 호스트 이름이나 IP 주소를 하위 패널에 입력합니다.

- 11. Time Zone(시간대) 패널에서 지역, 위치 및 시간대를 선택합니다.

---

참고 - 기본값은 GMT 시간대가 구성되도록 하는 것입니다.

---

- 12. Locale(로케일) 패널에서 언어와 언어 사용 지역을 선택합니다.

- 13. 다음 패널에서 날짜와 시간을 설정합니다.

- 14. 다음 패널에서 키보드 레이아웃을 선택합니다.

- 15. User(사용자) 패널에서 계정을 만듭니다.

반드시 사용자 계정을 만들 필요는 없지만, 루트 암호를 만들어야 합니다.

- 이 패널에서 사용자 계정을 만들면 사용자 암호와 루트 암호를 모두 제공해야 합니다.

이 경우 root는 사용자에게 지정된 역할이 됩니다.

사용자 계정을 만들려면 사용자 이름과 암호를 입력합니다. 이름은 문자로 시작해야 하며 문자와 숫자만 포함할 수 있습니다.

- 사용자 계정을 만들지 않으면 계속 root 암호를 제공해야 합니다.

이 경우 root는 일반 사용자가 됩니다.

- 16. Support - Registration(지원 - 등록) 패널에서 Oracle Configuration Manager를 사용하거나 Oracle Auto Service Request를 시작할지 여부와 방법을 결정합니다.

자세한 내용은 “Oracle Configuration Manager 사용” [30]을 참조하십시오.

- 17. Support - Network Configuration(지원 - 네트워크 구성) 패널에서 OCM 및 ASR에 대한 액세스 방법을 선택합니다.

사용 가능한 옵션은 다음과 같습니다.

- No proxy(프록시 사용 안함)
- Proxy(프록시) - 보안 프록시를 사용하는 경우 다음 패널에서 프록시 호스트 이름, 포트 번호 및 사용자 이름과 암호를 묻습니다.
- Aggregation Hubs(통합 허브) - 다음 패널에서 OCM 허브 URL 및 ASR Manager URL 을 묻습니다.

18. **설치 사양을 검토합니다.**

설치 요약 패널에서 사양을 검토합니다. 필요한 경우 뒤로 이동하여 설치를 시작하기 전에 필요에 따라 변경합니다.

19. **제공한 사양을 사용하여 시스템을 설치합니다.**

다시 시작 기능 키를 사용하여 Oracle Solaris 설치 프로세스를 시작합니다.



주의 - 설치가 진행되는 동안 중단하지 마십시오. 설치가 완료되지 않으면 디스크가 불확정 상태로 유지될 수 있습니다.

20. **설치 로그를 검토합니다.**

설치 결과 패널에서 설치 로그에 액세스하여 검토할 수 있습니다.

21. **재부트하거나 셸로 이동하고 시스템을 종료합니다.**

## ▼ 네트워크를 통해 텍스트 설치를 시작하는 방법

네트워크를 통해 자동 설치를 수행하도록 시스템을 설정한 경우 네트워크를 통해 클라이언트를 부트하고 대화식 텍스트 설치를 시작하는 옵션도 주어집니다. 이 옵션으로 한번에 하나의 시스템만 설치할 수 있지만, 대화식 선택을 통해 설치 사양을 수정하여 각 설치를 사용자 정의할 수 있는 기회가 생깁니다.

1. **AI 클라이언트 이미지를 다운로드하고 해당 이미지를 기반으로 설치 서비스를 만듭니다.**

자세한 내용은 “[설치 서비스 만들기](#)” [96]를 참조하십시오.

2. **네트워크를 통해 클라이언트 시스템을 부트합니다.**

- SPARC 클라이언트의 경우 OBP 프롬프트에 다음 명령을 입력합니다.

```
# boot net:dhcp
```

- x86 클라이언트의 경우 설치 메뉴에서 1을 선택합니다.

Welcome to the Oracle Solaris 11.2 installation menu

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot
```

Please enter a number [1]:

### 3. 클라이언트 시스템의 텍스트 설치를 완료합니다.

지침은 [텍스트 설치 수행 방법 \[46\]](#)을 참조하십시오.

---

**참고** - 텍스트 설치 프로그램으로 설치된 패키지 세트는 solaris-large-server 패키지 세트입니다. 부트 후 네트워크를 통해 텍스트 설치 프로그램을 사용할 경우 더 작은 패키지 세트인 solaris-auto-install이 기본적으로 설치됩니다.

이처럼 설치된 시스템은 아주 최소한입니다. 설치된 시스템으로 부트한 후에 solaris-large-server 패키지 세트를 설치하고, 선택적으로 다음과 같이 데스크탑을 설치해야 합니다.

패키지 설치, 업데이트 및 제거 작업을 수행하려면 높은 수준의 권한이 필요합니다. 자세한 내용은 [“Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”](#)의 [“설치 권한”](#)을 참조하십시오.

```
# pkg install solaris-desktop
```

```
# pkg install solaris-large-server
```

---

## 텍스트 설치 후 소프트웨어 추가

운영 체제를 설치한 후 소프트웨어 패키지를 추가하려면 [pkg\(1\)](#) 매뉴얼 페이지에 설명된 대로 pkg 명령을 사용합니다. pkg 명령 또는 Package Manager 도구를 사용하여 설치할 패키지의 이름을 찾아서 패키지에 대한 정보를 얻고 패키지를 설치할 수도 있습니다.

---

**참고** - 패키지 설치, 업데이트 및 제거 작업을 수행하려면 높은 수준의 권한이 필요합니다. 자세한 내용은 [“Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”](#)의 [“설치 권한”](#)을 참조하십시오.

---

선택적으로 새 부트 환경으로 설치할 수 있습니다. 그러면 새 설치에 문제가 있을 경우 현재 이미지를 계속 사용할 수 있습니다.

먼저 pkg install 명령과 함께 -nv 옵션을 사용하여 실제로 패키지를 설치하기 전에 패키지 설치가 어떻게 보이는지 확인해야 합니다. 설치할 패키지를 식별하고 pkg install 명령을 -nv 옵션과 함께 사용하여 얻은 출력을 확인한 후 다음 예제와 비슷한 명령을 실행하여 추가 소프트웨어를 설치합니다.

```
# pkg install package-name
```

다른 방법으로, 다음 샘플 명령을 사용하여 새 백업 부트 환경을 만들고 설치할 패키지를 지정할 수 있습니다.

```
# pkg install --be-name new-BE-name package-name
```

GUI 데스크탑이 없는데 Oracle Solaris Desktop을 설치하려면 `solaris-desktop` 패키지를 설치합니다.



# ◆◆◆ 5 장

## 매체에서 부트하는 자동 설치

---

네트워크를 통한 부트가 아닌 매체의 AI 이미지를 부트하여 SPARC 시스템 또는 x86 시스템에 Oracle Solaris OS의 자동 설치를 시작할 수 있습니다. 이 장에서는 매체에서 AI 클라이언트를 부트하는 이유와 해당 모드에서 설치를 수행하는 방법을 설명합니다.

### AI 매체를 사용한 설치 개요

AI 매체를 사용한 설치를 통해 다음 선택적 작업을 수행할 수 있습니다.

- AI 서버가 될 시스템을 설치합니다.
- WAN 부트 기능이 없는 SPARC 시스템을 설치합니다.
- 고장난 시스템 문제를 해결합니다. 이동식 매체에서 시스템을 부트하고 설치된 시스템을 검사하며 진단을 실행할 수 있습니다.

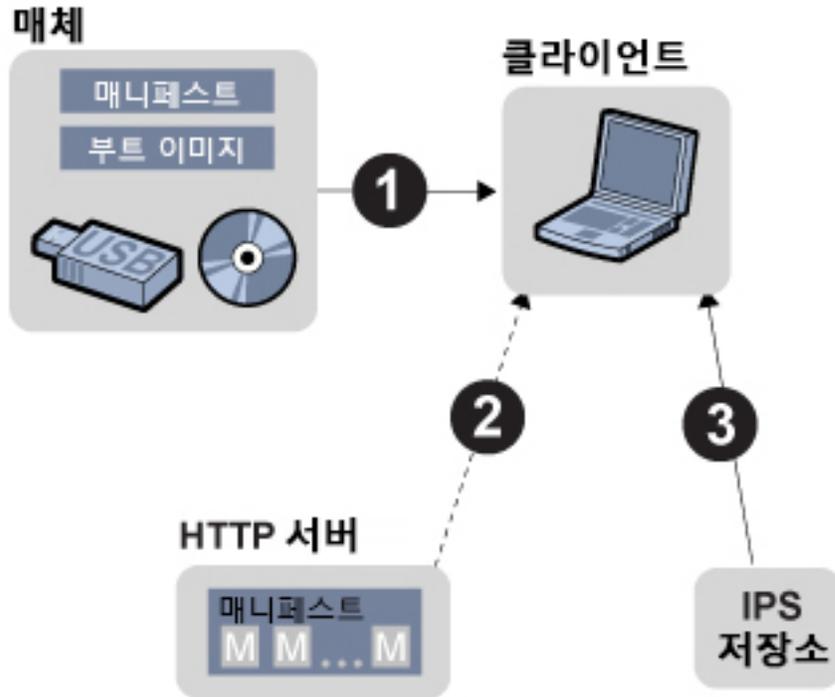
AI 매체를 사용한 설치는 다음 특징이 있습니다.

- AI 서버 또는 설치 서비스를 설정할 필요가 없습니다.
- 시스템이 네트워크를 통해 부트할 수 없어도 됩니다.

### AI 매체를 사용한 설치

CD, DVD 또는 USB 장치에서 AI 이미지를 부트하여 해당 시스템만 핸드프리 설치를 시작할 수 있습니다. AI 매니페스트는 설치 지침을 제공합니다. 설치될 시스템은 네트워크에 액세스할 수 있어야 합니다. 설치를 완료하려면 인터넷의 또는 로컬 네트워크의 IPS 저장소에서 소프트웨어 패키지가 검색됩니다. **“사용자 정의 AI 매니페스트 만들기” [58]**에 설명된 대로 기본 AI 매니페스트를 검토합니다.

그림 5-1 매체를 사용한 SI 설치



## SI 매체를 사용한 설치의 시스템 요구 사항

SPARC 및 x86 시스템 모두 다음 요구 사항을 충족해야 합니다.

- 메모리 - 현재 릴리스의 최소 메모리 요구 사항을 확인하려면 “[Oracle Solaris 11.2 릴리스 노트](#)”를 참조하십시오.
- 디스크 공간 - 현재 릴리스의 디스크 공간 요구 사항을 확인하려면 “[Oracle Solaris 11.2 릴리스 노트](#)”를 참조하십시오.
- 네트워크 액세스 - 설치할 시스템은 클라이언트 시스템에 설치할 패키지가 포함된 IPS 저장소에 액세스할 수 있어야 합니다. 또한 사용자 정의 SI 매니페스트를 만들면 시스템에서 HTTP 서버의 해당 매니페스트에 액세스할 수 있어야 합니다.

## ▼ AI 매체를 사용한 설치 방법

1. AI 부트 이미지를 다운로드합니다.  
AI 부트 이미지를 다운로드하려면 <http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html>로 이동합니다.
2. 기본 AI 매니페스트를 검토합니다.  
AI 이미지에 제공된 기본 매니페스트를 사용할 수도 있고, 또는 사용자 정의 매니페스트를 만들어서 클라이언트를 부트할 때 이 사용자 정의 매니페스트의 위치를 제공할 수도 있습니다. [“사용자 정의 AI 매니페스트 만들기” \[58\]](#)를 참조하십시오.
3. 부트 가능한 매체를 만듭니다.
  - ISO 이미지 - .iso 파일을 CD나 DVD에 굽습니다.
  - USB 이미지 - `usbcopy` 유틸리티를 사용하여 USB 플래시 드라이브에 이미지를 복사합니다.

---

참고 - 이 유틸리티를 시스템에 추가하려면 `pkg:/install/distribution-creator` 패키지를 설치합니다.

---

4. 매체에서 부트합니다.  
부트 이미지가 포함된 장치에서 시스템을 부트합니다. 기본 AI 매니페스트 또는 사용자 정의 AI 매니페스트를 지정하는 방법은 [“AI 매체에서 SPARC 시스템 부트” \[59\]](#) 및 [“AI 매체에서 x86 시스템 부트” \[60\]](#)를 참조하십시오.  
“핸즈프리” 설치가 수행됩니다. 설치 후에 SCI 도구가 시작되고 시스템에 대한 구성 정보를 제공하도록 요청합니다.
5. SCI 도구 패널에서 구성 정보를 제공합니다.  
[“SCI 도구를 사용하여 시스템 구성 프로파일 만들기” \[69\]](#)를 참조하십시오.

## ▼ SPARC 시스템에서 USB 플래시 드라이브의 지속 장치 별칭을 만드는 방법

SPARC 시스템에서 USB 플래시 드라이브의 지속 장치 별칭을 만들려면 OBP를 사용해야 하므로 시스템을 종료해야 합니다. 별칭이 생성된 후에는 같은 포트를 다시 사용하는 한 별칭을 다시 만들 필요가 없습니다.

1. 시스템을 종료하고 부트 프롬프트에 둡니다.
2. 시스템에서 사용 가능한 디스크를 식별합니다.

이 예제에서는 두번째 장치의 선택을 보여줍니다.

```
{0} ok show-disks
a) /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@3/disk
b) /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk
c) /pci@400/pci@0/pci@1/scsi@0/disk
d) /iscsi-hba/disk
q) NO SELECTION

Enter Selection, q to quit: b
/pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev
      for creating devalias mydev for
/pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk
```

### 3. USB 플래시 드라이브의 별칭을 설정합니다.

```
{0} ok nvalias usbdrive ^Y
```

### 4. USB 플래시 드라이브에서 부트합니다.

```
{0} ok boot usbdrive
Boot device: /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk File
and args:
```

## 사용자 정의 AI 매니페스트 만들기

AI 부트 이미지에 제공된 AI 매니페스트의 설치 사양을 사용하여 시스템을 설치할 수도 있고, 또는 사용자 정의 설치 사양을 만들 수도 있습니다. 사용자 정의 AI 매니페스트를 만들면 HTTP 서버에 매니페스트를 저장했다가 설치될 시스템을 부트할 때 매니페스트의 위치를 제공합니다.

.iso AI 이미지를 다운로드하면 다음 샘플 명령을 사용하여 해당 이미지의 AI 매니페스트를 검사할 수 있습니다. 이 예에서 /tmp는 AI 이미지를 다운로드한 디렉토리이고, /home/username은 AI 매니페스트를 복사 및 편집하려는 디렉토리입니다. AI 매니페스트는 이미지의 auto\_install/manifest/default.xml에 있습니다.

```
# /usr/sbin/mount -o ro -F hsfs /tmp/sol-11_2-20-ai-x86.iso /mnt
# cp /mnt/auto_install/manifest/default.xml /home/username/custom.xml
# umount /mnt
```

기본 매니페스트 파일의 복사본(이 예제에서는 /home/username/custom.xml)을 검토하고 이러한 사양이 이 설치에 적합한지 여부를 결정합니다.

다른 방법으로, 사용자 정의 매니페스트를 만들기 위한 기초로 “[기본 AI 매니페스트](#)” [164]에 표시된 매니페스트를 사용할 수 있습니다.

대상 디스크나 추가로 설치할 패키지 등의 설치 사양을 변경하는 방법을 알아보려면 [aimanifest\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

AI 매니페스트 수정을 마쳤으면 사용자 정의 매니페스트를 HTTP 서버로 복사합니다. 사용자 정의 AI 매니페스트의 URL을 적어 두십시오. 그래야 설치될 시스템을 부트할 때 해당 URL을 제공할 수 있습니다. 예를 들어, URL은 `http://example.com/custom.xml`일 수 있습니다.

## AI 매체에서 SPARC 시스템 부트

AI 매체에서 시스템을 부트할 때 기본 AI 매니페스트 또는 사용자 정의 AI 매니페스트를 지정할 수 있습니다.

### 기본 AI 매니페스트를 사용하여 AI 매체에서 SPARC 시스템 부트

AI 부트 이미지에 있는 기본 AI 매니페스트를 사용하려면 OBP 프롬프트에 다음 명령을 입력합니다.

```
ok> boot cdrom - install
```

기본 매니페스트의 사양을 사용하여 자동 설치가 진행됩니다.

### 사용자 정의 AI 매니페스트를 사용하여 AI 매체에서 SPARC 시스템 부트

사용자 정의 AI 매니페스트를 사용하려면 OBP 프롬프트에 다음 명령을 입력합니다.

```
ok> boot cdrom - install aimanifest=prompt
```

다음 프롬프트가 표시됩니다.

```
Enter the URL for the AI manifest [HTTP, default]:
```

사용자 정의 매니페스트의 URL을 입력합니다. 예를 들어, `http://example.com/custom.xml`을 입력합니다.

사용자 정의 매니페스트의 사양을 사용하여 자동 설치가 진행됩니다.

### 설치 없이 SPARC 이미지 부트

설치 없이 매체에서 부트하고 싶을 수 있습니다. 예를 들어, 문제를 해결하거나 시스템을 조사하려는 경우입니다.

AI 이미지를 부트하되, 자동 설치를 시작하지 않으려면 다음 명령을 사용합니다.

```
ok> boot cdrom
```

시스템이 부트되고 로그인 패널이 표시되지만 설치가 시작되지 않습니다.

## AI 매체에서 x86 시스템 부트

x86 시스템에서, GRUB 메뉴에서 자동 설치 옵션을 선택합니다. 여기서 사용한 GRUB 메뉴 선택이나 부트 명령에 따라 매체의 기본 매니페스트를 사용할지, HTTP 서버에 저장된 사용자 정의 매니페스트를 사용할지 지정됩니다.

GRUB 메뉴 선택은 다음 예와 비슷해야 합니다.

```
GNU GRUB version 1.99.5.11.0.175.2.0.0.20.0
```

```
Oracle Solaris 11.2 Automated Install custom
Oracle Solaris 11.2 Automated Install
Oracle Solaris 11.2 Automated Install custom ttya
Oracle Solaris 11.2 Automated Install custom ttyb
Oracle Solaris 11.2 Automated Install ttya
Oracle Solaris 11.2 Automated Install ttyb
Boot from Hard Disk
```

Use the arrow keys to select which entry is highlighted.  
Press enter to boot the selected OS, 'e' to edit the  
commands before booting, or 'c' for a command-line.

## 기본 AI 매니페스트를 사용하여 AI 매체에서 x86 시스템 부트

AI 부트 이미지에 있는 기본 AI 매니페스트를 사용하려면 화살표 키를 사용하여 다음 옵션 중 하나를 선택합니다.

```
Oracle Solaris 11.2 Automated Install
Oracle Solaris 11.2 Automated Install ttya
Oracle Solaris 11.2 Automated Install ttyb
```

ttya 옵션은 설치 중 화면 출력을 직렬 콘솔 ttya (COM1)로 보냅니다. ttyb 옵션은 설치 중 화면 출력을 직렬 콘솔 ttyb (COM2)로 보냅니다.

기본 매니페스트의 사양을 사용하여 자동 설치가 진행됩니다.

## 사용자 정의 AI 매니페스트를 사용하여 AI 매체에서 x86 시스템 부트

사용자 정의 AI 매니페스트를 사용하려면 다음 옵션 중 하나를 선택합니다.

```
Oracle Solaris 11.2 Automated Install custom
Oracle Solaris 11.2 Automated Install custom ttya
Oracle Solaris 11.2 Automated Install custom ttyb
```

이러한 사용자 정의 옵션 중 하나를 선택할 때 다음 프롬프트가 표시됩니다.

Enter the AI manifest location [URL, /filepath, 'default']:

사용자 정의 매니페스트의 URL을 입력합니다. 예를 들어, `http://example.com/custom.xml`을 입력합니다.

사용자 정의 매니페스트의 사양을 사용하여 자동 설치가 진행됩니다.

## 설치 없이 x86 이미지 부트

문제 해결, 시스템 검사 등을 위해 설치하지 않고 매체에서 부트할 수 있습니다.

사용 중인 GRUB2 항목에 대해 `$multiboot`로 시작하는 라인에 `install=true`가 지정된 경우 설치가 자동으로 시작됩니다. 자동 설치를 즉시 시작하지 않고 x86 시스템을 부트하려는 경우 사용할 GRUB2 항목의 커널 라인에 `install=true`가 지정되어 있으면 라인을 편집하여 `install=true`를 제거합니다. 해당 옵션을 선택할 때 시스템이 부트되고 로그인 화면이 표시되지만 설치가 시작되지 않습니다.

## 설치 로그 파일 보기

자동 설치가 완료되면 출력에 설치 성공 또는 실패 여부가 나타납니다.

- 설치를 실패한 경우 `/system/volatile/install_log`에서 설치 로그를 검토할 수 있습니다.
- 설치에 성공한 경우 시스템을 재부트하기 전에 `/system/volatile/install_log`에서, 시스템을 재부트한 후에 `/var/log/install/install_log`에서 로그를 찾을 수 있습니다.



## Oracle Solaris 인스턴스 구성 해제 또는 재구성

Oracle Solaris 인스턴스는 설치 중에 만들어지고 구성됩니다. Oracle Solaris 인스턴스는 전역 또는 비전역 영역에서 부트 환경으로 정의됩니다. 이 장에서는 Oracle Solaris 인스턴스를 구성 해제 및 재구성하는 방법을 설명합니다.

### 기능 그룹 개요

Oracle Solaris 인스턴스를 구성 해제 또는 재구성할 때 전체 시스템이나 특정 부속 시스템의 구성 데이터를 변경할 수 있습니다. 이러한 부속 시스템을 기능 그룹이라고 합니다. system 기능 그룹은 시스템의 모든 기능 그룹을 변경합니다. 또는 특정 구성의 구성 요소만 변경하도록 하나 이상의 기능 그룹을 지정할 수 있습니다.

다음 표는 Oracle Solaris 인스턴스에 존재하는 구성 가능한 기능 그룹을 나열합니다.

표 6-1 기능 그룹

그룹	구성 요소	구성 해제 상태
date_time	시스템 날짜 및 시간	해당 없음
identity	시스템 노드 이름	알 수 없음
keyboard	키보드	미국 영어
location	시간대	UTC
	로케일	C 로케일
naming_services	DNS, NIS, LDAP 클라이언트, nsswitch	네트워크 이름 지정 서비스 없음
network	네트워크	네트워크 없음
support	OCM 및 ASR 지원	기본 설정은 OCM 및 ASR에 익명 등록입니다.
system	전체 시스템	"system" 그룹은 다른 모든 그룹을 포함합니다.
users	루트	빈 루트 암호
	초기 사용자 계정	사용자 계정 제거

## Oracle Solaris 인스턴스 구성 해제

전에 구성된 Solaris 인스턴스의 구성을 해제하고 구성 해제 상태로 두려면 `unconfigure` 하위 명령을 사용하십시오. 모든 기능 그룹 또는 일부를 구성 해제하도록 선택할 수 있습니다. 자세한 내용은 [sysconfig\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### 예 6-1 시스템 구성 해제

다음 예제와 같이 `sysconfig unconfigure` 명령을 사용하여 시스템에서 모든 구성 데이터를 제거합니다.

```
# sysconfig unconfigure -g system
```

### 예 6-2 SMF 데이터 구성 해제

`/etc/svc/profile/site` 및 관리 계층에 저장된 데이터는 기능 그룹에 포함되지 않습니다. 이 데이터를 제거하고 시스템 구성을 해제하려면 다음 명령을 사용하십시오.

```
# sysconfig unconfigure -g system --include-site-profile
```

제거된 XML 프로파일은 `/etc/svc/profile/sysconfig/site-profile.tar`라는 tar 파일로 아카이브됩니다.

### 예 6-3 특정 기능 그룹 구성 해제

이 예제에서는 시스템의 날짜 및 시간 데이터를 구성합니다.

```
# sysconfig unconfigure -g date_time
```

---

**참고** - `-g` 옵션이 지정되지 않은 경우 시스템 구성을 해제하기 전에 확인을 요청합니다.

---

### 예 6-4 시스템 구성 해제 후 재부트

다른 방법으로, 다음과 같이 시스템 구성을 해제하고 시스템을 종료할 수 있습니다.

```
# sysconfig unconfigure -s
```

시스템을 재구성하기 위해 시스템이 재부트된 후 SCI 도구가 실행됩니다.

## 시스템 재구성

`sysconfig configure` 명령을 사용하여 전역 또는 비전역 영역에서 Oracle Solaris 인스턴스를 구성하거나 재구성할 수 있습니다. 비대화식 또는 대화식으로 이 구성이 이루어질 수 있습니다. `-c` 옵션을 `sysconfig configure` 명령과 함께 사용하여 기존 시스템 구성 프로파일을 지정할 수 있습니다. 해당 옵션과 함께 명령을 실행하면 유틸리티가 기존 프로파일에서 구성 사양을 읽고 이러한 사양을 사용하여 시스템을 비대화식으로 구성합니다. `-c`를 사용하지 않고 명령을 실행하면 SCI 도구가 자동으로 활성화됩니다.

---

참고 - `sysconfig reconfigure` 명령은 `sysconfig configure`에 대한 별칭입니다.

---

예 6-5            시스템 구성 프로파일을 사용하여 시스템 재구성

다음 명령은 `myprofile.xml`이라는 기존 구성 프로파일을 사용하여 시스템이 구성되도록 지정합니다.

```
# sysconfig configure -c myprofile.xml
```

### ▼ SCI 도구를 사용하여 재구성하는 방법

1. 루트 역할로 전환합니다.
2. 프로파일을 지정하지 않고 `sysconfig configure` 명령을 실행합니다.

```
# sysconfig configure
```

SCI 도구가 표시됩니다. 다음 단계는 SCI 도구에서 일련의 대화식 패널을 완성하기 위한 지침을 제공합니다.

---

참고 - 기능 키를 사용하여 SCI 도구 패널을 탐색합니다. 마우스는 사용할 수 없습니다. 각 패널의 기능 키 참조를 참고하고, 필요에 따라 온라인 도움말을 참조하십시오.

---

3. 초기 시작 패널을 지나서 계속합니다.
4. 네트워크에서 시스템을 식별하는 이름을 제공합니다.
5. 다음 옵션 중 하나를 선택하여 유선 이더넷 네트워크 연결을 어떻게 구성할지 지정합니다.
  - 설치 중 네트워크가 구성되지 않도록 지정하려면 None(없음)을 선택합니다. 설치 프로그램이 시간대 패널로 진행됩니다.
  - DHCP를 사용하여 네트워크 연결을 구성하려면 Automatically(자동)를 선택합니다.

설치 프로그램이 시간대 패널로 진행됩니다.

- 네트워킹 사양을 제공하려면 Manually(수동)를 선택하고 다음과 같이 계속합니다.
  - a. 인터페이스가 여러 개 있는 경우 구성할 연결을 선택합니다.
  - b. Manually Configure(수동 구성) 패널에서 연결 설정을 입력하거나 설치 프로그램에서 찾아 제공한 기본 정보를 적용합니다.

---

참고 - IP 주소 및 넷마스크는 필수 필드입니다. 라우터는 선택적 필드입니다.

---

- c. DNS Name Service(DNS 이름 서비스) 패널에서 시스템에 DNS 이름 서비스를 사용할지 여부를 지정합니다.
- d. Configure DNS(DNS 구성)를 선택한 경우 다음 패널에서 정보를 업데이트합니다.
  - i DNS Server Addresses(DNS 서버 주소) 패널에서 DNS 서버의 IP 주소를 하나 이상 입력합니다.
  - ii DNS Search List(DNS 검색 목록) 패널에서 DNS 질의를 만들 때 검색할 도메인 이름을 하나 이상 제공합니다.
- e. Alternate Name Service(대체 이름 서비스) 패널에서 시스템에 LDAP 이름 서비스, NIS 이름 서비스 또는 None(없음)을 사용할지 여부를 지정합니다.
  - 이전 단계에서 DNS를 선택한 경우 LDAP 또는 NIS가 DNS와 더불어 대체 이름 서비스로 설정됩니다.
  - 이전 단계에서 DNS를 선택하지 않은 경우 LDAP 또는 NIS가 유일한 이름 서비스로 설정됩니다.
  - LDAP 프로파일을 사용하지 않고 시스템에 LDAP를 구성하려면 LDAP 대신 None을 선택합니다. 그런 다음, 설치가 완료된 후 수동으로 LDAP를 구성합니다.  
선택된 네트워크 이름 지정 서비스가 없으면 /etc/hosts와 같은 표준 이름 소스 파일을 사용하여 네트워크 이름을 분석할 수 있습니다. 자세한 내용은 nsswitch.conf(4) 매뉴얼 페이지를 참조하십시오.
- f. Domain Name(도메인 이름) 패널에서 선택한 대체 이름 서비스에 대해 시스템이 상주하는 도메인을 제공합니다.

---

참고 - 도메인 이름을 확인하려면 시스템 관리자에게 문의하십시오. 또는 이전에 설치된 시스템에 domainname 명령을 사용하십시오.

---

- g. LDAP Profile(LDAP 프로파일) 패널에서는 Alternate Name Service(대체 이름 서비스) 패널에서 LDAP를 선택한 경우 다음과 같이 LDAP 구성 사양을 제공합니다.
- 시스템에서 LDAP 이름 서비스를 구성하는 데 사용할 LDAP 프로파일
  - LDAP 프로파일 서버의 IP 주소
  - LDAP 검색 기준
  - LDAP Proxy(LDAP 프록시) 패널에서 LDAP 프록시 바인드 정보를 제공할지 여부를 지정합니다.  
필요할 경우 LDAP 프록시 바인드 고유 이름 및 프록시 바인드 암호를 제공합니다.
- h. NIS Name Server(NIS 이름 서버) 패널에서는 Alternate Name Service(대체 이름 서비스) 패널에서 NIS를 선택한 경우 NIS 사양을 제공합니다.  
소프트웨어에서 이름 서버를 검색하도록 하거나, 직접 이름 서버를 지정할 수 있습니다. 다음 둘 중 하나를 선택합니다.
- Find One(이름 서버 찾기)

---

참고 - 이름 서버가 로컬 서브넷에 있는 경우에만 소프트웨어가 해당 서버를 찾을 수 있습니다.

---

- Specify One(이름 서버 지정) - 이름 서버의 호스트 이름이나 IP 주소를 하위 패널에 입력합니다.

6. Time Zone(시간대) 패널에서 지역, 위치 및 시간대를 선택합니다.

---

참고 - 기본값은 GMT 시간대가 구성되도록 하는 것입니다.

---

7. Locale(로케일) 패널에서 언어와 언어 사용 지역을 선택합니다.
8. 다음 패널에서 날짜와 시간을 설정합니다.
9. 다음 패널에서 키보드 레이아웃을 선택합니다.
10. 사용자 패널을 완성합니다.  
반드시 사용자 계정을 만들 필요는 없지만, 루트 암호를 만들어야 합니다.
- 이 패널에서 사용자 계정을 만들면 사용자 암호와 루트 암호를 모두 제공해야 합니다. 이 경우 root는 사용자에게 지정된 역할이 됩니다.  
사용자 계정을 만들려면 사용자 이름과 암호를 입력합니다. 이름은 문자로 시작해야 하며 문자와 숫자만 포함할 수 있습니다.

- 사용자 계정을 만들지 않으면 계속 root 암호를 제공해야 합니다.  
이 경우 root는 일반 사용자가 됩니다.

11. **Support - Registration(지원 - 등록) 패널에서 My Oracle Support 전자 메일 주소와 암호를 입력합니다.**

기본 Support - Registration(지원 - 등록) 패널에서는 익명 등록 주소를 제공합니다. 이 익명 주소를 암호 없이 사용할 경우 MOS(My Oracle Support)는 설치된 시스템의 구성에 대한 정보를 받지만 Oracle Support 조직에 시스템 구성을 업로드할 때 고객 정보는 전혀 받지 않습니다.

또는 다음과 같이 보안 업데이트에 등록하거나 OCM(Oracle Configuration Manager) 연결을 해제할 수 있습니다.

- 패널의 익명 전자 메일 주소를 My Oracle Support 로그인 ID로 바꾸고 My Oracle Support 암호를 추가할 수 있습니다. My Oracle Support에서 고객 정보를 보고 보안 업데이트를 받고자 하는 경우 이 옵션을 사용하십시오. 이 옵션을 사용하면 ASR(Auto Service Request)도 시작됩니다.
- 패널의 익명 전자 메일 주소를 삭제하고 해당 필드를 비워 둘 경우 OCM이 연결 해제 모드로 시작됩니다. My Oracle Support에 데이터가 전송되지 않습니다. 또는 익명 전자 메일 주소를 삭제하고 MOS 로그인 ID 이외의 다른 전자 메일 주소로 바꿀 경우 OCM은 비인증 모드로 Oracle Support에 데이터를 전송합니다.

자세한 내용은 [“Oracle Configuration Manager 사용” \[30\]](#)을 참조하십시오.

12. **Support - Network Configuration(지원 - 네트워크 구성) 패널에서 OCM 및 ASR에 대한 액세스 방법을 선택합니다.**

사용 가능한 옵션은 다음과 같습니다.

- No proxy(프록시 사용 안함)
- Proxy(프록시) - 보안 프록시를 사용하는 경우 다음 패널에서 프록시 호스트 이름, 포트 번호 및 사용자 이름과 암호를 묻습니다.
- Aggregation Hubs(통합 허브) - 다음 패널에서 OCM 허브 URL 및 ASR Manager URL을 묻습니다.

13. **설치 사양을 검토합니다.**

설치 요약 패널에서 사양을 검토합니다. 필요한 경우 뒤로 이동하여 설치를 시작하기 전에 필요에 따라 변경합니다.

14. **제공한 사양을 사용하여 시스템을 설치합니다.**

설정이 맞으면 시스템에 구성을 적용합니다.

## SCI 도구를 사용하여 시스템 구성 프로파일 만들기

SCI 도구를 실행하여 SCI 도구 패널에 입력한 구성 사양을 기반으로 새 시스템 구성 프로파일을 생성할 수 있습니다. 새 프로파일의 기본 위치는 `/system/volatile/profile/sc_profile.xml`입니다.

새 시스템 구성 프로파일을 만들려면 `sysconfig create-profile` 명령을 사용합니다. 프로파일이 만들어지지만, 구성은 시스템에 적용되지 않습니다.

---

**참고** - 프로파일을 재구성에 성공적으로 사용하려면 해당 프로파일에 `.xml` 확장자를 포함해야 합니다.

자세한 내용은 [sysconfig\(1M\)](#) 매뉴얼 페이지를 참조하십시오. [11장. 클라이언트 시스템 구성도](#) 참조하십시오.

---

### 예 6-6 기본 프로파일 사용

SCI Tool(SCI 도구) 패널에 제공된 사양을 기반으로 SCI 도구가 새 시스템 구성 프로파일을 만듭니다. 새 프로파일은 기본 위치에 저장됩니다. 새 프로파일을 사용하여 다음 예에 표시된 대로 시스템을 구성할 수 있습니다.

```
# sysconfig configure -g system -c /etc/system/profile/sysconfig/sc_profile.xml
```

### 예 6-7 프로파일 만들기 및 사용

`-g` 옵션은 구성할 특정 기능 그룹을 지정하는 데 사용됩니다. 이 예에서 전체 시스템이 구성됩니다. 기능 그룹 목록은 [표 6-1. "기능 그룹"](#)을 참조하십시오. 다음 예제에서는 프로파일을 만든 다음 이 프로파일을 사용하여 비대화식으로 시스템을 재구성합니다.

```
# sysconfig create-profile -o /tmp/myprofile.xml
# sysconfig configure -g system -c /tmp/myprofile.xml
```

### 예 6-8 프로파일을 만들고 기능 그룹 구성에 사용

다음 예제에서는 `network` 및 `naming_services` 기능 그룹의 프로파일을 만듭니다. 그런 다음 기능 그룹을 비대화식으로 재구성하는 데 프로파일이 사용됩니다.

```
# sysconfig create-profile -g network,naming_services -o /tmp/myprofile.xml
# sysconfig configure -g network,naming_services -c /tmp/myprofile.xml
```



### 부 III

## 설치 서버를 사용하여 설치

이 섹션은 네트워크를 통한 클라이언트 시스템의 자동 설치를 설명합니다.



## 다중 클라이언트의 자동 설치

---

AI(자동 설치 프로그램)를 사용하여 네트워크의 여러 클라이언트 시스템에 Oracle Solaris OS(운영 체제)를 설치할 수 있습니다. AI는 SPARC 및 x86 시스템의 "핸즈프리" 설치를 수행합니다. 모든 AI 설치에서 네트워크의 소프트웨어 패키지 저장소 또는 Oracle Solaris Unified Archive에 액세스할 수 있어야 합니다.

### 자동 설치란?

AI는 네트워크를 통한 SPARC 및 x86 클라이언트의 Oracle Solaris OS 설치를 자동화합니다. 디스크 레이아웃 및 소프트웨어 패키지 선택을 정의하는 설치 전 지침을 사용하여 클라이언트 설치를 사용자 지정할 수 있습니다. 호스트 이름, 네트워크 구성, 사용자 계정 및 그 밖의 설치 후 클라이언트별 지침을 정의하는 사용자 정의 시스템 구성 매개변수를 제공할 수도 있습니다. 모든 사용자 정의는 클라이언트 단위로 진행되며 대규모 환경에 맞게 조정할 수 있습니다.

네트워크를 통한 클라이언트의 자동 설치는 다음과 같은 높은 레벨의 단계로 구성됩니다.

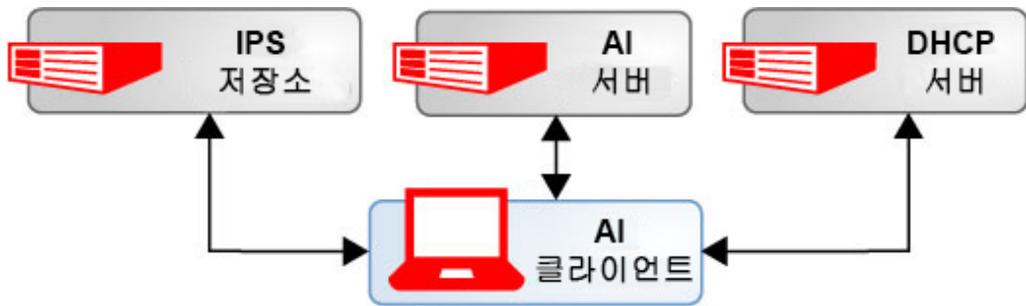
1. 클라이언트 시스템이 네트워크를 통해 부트하고 DHCP 서버에서 네트워크 구성과 AI 서버의 위치를 얻습니다. SPARC 클라이언트는 선택적으로 OBP(Open Boot PROM)에 설정된 `network-boot-arguments` 변수에서 네트워크 구성 정보와 AI 서버의 위치를 얻을 수 있습니다.
2. 설치 서비스가 클라이언트에 부트 이미지를 제공합니다.
3. 클라이언트의 특성에 따라 클라이언트 설치에 사용되는 설치 지침 및 시스템 구성 지침이 결정됩니다.
4. Oracle Solaris 11 OS가 클라이언트에 설치됩니다. 설치 프로그램이 패키지 저장소에서 패키지를 가져오거나 설치 서비스의 설치 지침에 지정된 아카이브에서 시스템 이미지를 가져옵니다.

### 자동 설치 프로그램의 구성 요소

AI를 사용하는 네트워크는 다음 구성 요소로 이루어져 있습니다.

- 호스트 정보와 함께 AI 클라이언트를 제공하는 DHCP 서버
- 클라이언트에 설치할 소프트웨어 패키지를 제공하는 IPS(이미지 패키징 시스템) 저장소 하나 이상. 네트워크에 있는 아카이브에서 클라이언트의 시스템 이미지가 만들어질 수도 있습니다.
- AI 클라이언트의 구성 지침이 포함된 AI 서버
- AI 클라이언트 하나 이상

그림 7-1 AI 네트워크 예



DHCP 서버, IPS 서버 및 AI 서버를 별도의 시스템에서 호스트할 필요가 없습니다. 특히 AI 서버와 DHCP 서버를 같은 시스템에 설치하면 DHCP 서비스가 설치 서비스와 함께 있을 경우 `installadm` 명령이 DHCP 서비스를 업데이트하므로 관리 단계가 간편해집니다.

AI 서버 자체는 다음 구성 요소를 포함할 수 있습니다.

- 설치 서비스 하나 이상. 각 서비스는 클라이언트에 설치할 아키텍처 및 OS와 일치하도록 구성됩니다.
- AI 매니페스트 하나 이상. AI 매니페스트는 사용할 디스크 레이아웃이나 추가할 패키지와 같은 클라이언트 설치 지침을 제공합니다.
- 선택적 시스템 구성 프로파일. 이 프로파일은 사용할 시간대나 이름 서비스와 같은 시스템 구성 정보를 제공합니다.

또한 IPS 패키지를 만들어 설치 프로세스 중에 처음 부트 스크립트를 클라이언트에 전달할 수 있습니다. 이 스크립트로 AI 클라이언트에 타사 유틸리티를 추가하는 등 AI 매니페스트나 시스템 구성 프로파일로는 수행할 수 없는 추가 설치 또는 구성 단계를 수행할 수 있습니다.

## AI를 지원하는 DHCP 서버

가장 간단한 구성에서 AI는 DHCP를 사용하여 IP 주소, 서브넷 마스크, 라우터, 이름 서비스 서버 및 AI 서버의 위치를 클라이언트 시스템에 제공합니다. AI 서버와 같은 시스템에서 실행

되도록 DHCP 서버를 구성할 수 있습니다. 이 구성은 가장 관리하기 쉬운 구성입니다. DHCP 서버가 별도의 시스템에 있을 경우 새 클라이언트를 추가할 때 수동으로 DHCP 서버를 업데이트해야 할 수 있습니다.

클라이언트가 DHCP를 사용하지 않고 AI 서버를 찾을 수 있도록 SPARC 클라이언트를 구성할 수는 있지만 x86 클라이언트의 경우 그럴 수 없습니다. DHCP를 실행하지 않으려면 [“x86 클라이언트 설치” \[226\]](#)에서 x86 클라이언트 설치 대안을 참조하십시오. SPARC 클라이언트에서 DHCP 없이 AI 서버를 사용할 수 있도록 OBP를 사용해 필요한 정보를 제공하기 위한 지침은 [“SPARC 클라이언트 설치” \[224\]](#)를 참조하십시오.

## AI를 지원하는 IPS 저장소

설치하려는 클라이언트 시스템에서 Oracle Solaris IPS(이미지 패키징 시스템) 소프트웨어 패키지 저장소 또는 Oracle Solaris Unified Archive에 액세스할 수 있어야 합니다. 저장소는 소프트웨어 패키지가 검색되는 위치입니다. 이 위치는 URI(Universal Resource Identifier)로 지정됩니다. IPS 패키지 저장소는 AI 서버, 로컬 네트워크의 다른 서버 또는 인터넷상의 다른 곳에 있을 수 있습니다. 패키지 저장소 액세스에 대한 자세한 내용은 [“Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”의 “게시자 구성”](#)을 참조하십시오. 또한 IPS 서버는 클라이언트를 완전히 구성하는 데 필요한 처음 부트 스크립트를 제공할 수 있습니다.

## AI 서버

AI를 사용하여 네트워크를 통해 AI 클라이언트 시스템을 설치하려면 먼저 AI 서버에서 AI 설치 서비스를 설정해야 합니다. 전체 절차는 [8장. AI 서버 설정](#)을 참조하십시오. 서버의 IP 주소가 각 클라이언트에 대해 생성된 파일에 들어 있으므로 AI 서버의 정적 네트워크 주소를 만드는 방법이 절차에 포함됩니다. 서버의 IP 주소가 변경되면 모든 클라이언트의 구성 파일을 다시 만들어야 합니다.

## 설치 서비스

각 설치 서버는 설치 서비스를 하나 이상 포함할 수 있습니다. 지원할 OS 버전 및 클라이언트 아키텍처마다 설치 서비스를 만들어야 합니다. 예를 들어, Oracle Solaris 11.1을 부트하는 SPARC 클라이언트용 설치 서비스와 Oracle Solaris 11.2를 부트하는 SPARC 클라이언트용 설치 서비스를 두고, 2개를 더 만들어 x86 클라이언트에 같은 서비스를 제공할 수 있습니다. 각 설치 서비스에는 SPARC 또는 x86 부트 이미지, 1개 이상의 설치 지침 파일(AI 매니페스트) 및 선택적 시스템 구성 프로파일이 포함됩니다. [“설치 서비스 만들기” \[96\]](#)에서는 설치 서비스를 만들고 유지하기 위한 지침을 제공합니다.

AI 서버에서 제공하는 부트 이미지는 완전한 설치가 아닙니다. 부트 이미지는 설치를 실행할 수 있는 클라이언트에 구성을 만듭니다. 설치를 완료하려면 클라이언트 시스템이 IPS 패키지 저장소나 아카이브에 액세스해야 합니다.

## 시 매니페스트

시 매니페스트는 클라이언트 프로비저닝 또는 설치 지침을 포함합니다. 여러 클라이언트가 매니페스트 하나를 공유할 수 있지만 클라이언트마다 시 매니페스트 하나만 사용됩니다. 시 매니페스트는 클라이언트가 설치를 완료하는 데 필요한 패키지를 검색하는 하나 이상의 IPS 패키지 저장소를 지정합니다. IPS 패키지 대신 아카이브를 사용할 수 있습니다. 시 매니페스트는 추가로 설치할 패키지 이름과 대상 설치 장치 및 분할 영역과 같은 정보도 포함할 수 있습니다. Oracle Solaris 11 OS 버전이 같은 두 클라이언트 시스템을 서로 다른 방식으로 설치하려면 시 설치 서비스 하나와 연관된 시 매니페스트 2개를 만듭니다. 서로 다른 시 매니페스트는 설치 대상으로 서로 다른 설치 패키지나 서로 다른 슬라이스를 지정할 수 있습니다. 클라이언트를 부트하기 전에 또는 클라이언트 설치 타임에 동적으로 시 매니페스트를 만들고 사용자 정의하는 방법은 [10장. 클라이언트 시스템 프로비전](#)을 참조하십시오.

## 시스템 구성 프로파일

클라이언트 시스템에 서로 다른 구성을 적용해야 하는 경우 설치 서비스별로 여러 시스템 구성 프로파일을 만듭니다. 서로 다른 시스템 구성 프로파일은 서로 다른 네트워크 또는 로컬 설정이나 고유한 호스트 이름 및 IP 주소를 지정할 수 있습니다. 시간대를 설정하는 프로파일이 여러 시 클라이언트에 사용될 수 있습니다. 프로파일에 대한 자세한 내용은 [11장. 클라이언트 시스템 구성](#)을 참조하십시오.

클라이언트에 프로파일이 구성되지 않은 경우 설치가 끝나고 클라이언트가 부트된 후 대화식 도구에서 시스템 구성 정보를 묻습니다. 대화식 구성 도구에 대한 자세한 내용은 “[시스템 재구성](#)” [65]을 참조하십시오.

## 처음 부트 스크립트

시 매니페스트나 시스템 구성 프로파일로 표현할 수 없는 구성을 포함하려면 처음 부트 시 실행할 스크립트를 포함할 수 있습니다. 자세한 내용은 [13장. 처음 부트 중 사용자 정의 스크립트 실행](#)을 참조하십시오.

## 시 클라이언트

클라이언트를 부트하면 설치가 시작됩니다. 클라이언트를 부트할 때 클라이언트가 시 서버로 지정되고, 클라이언트는 올바른 설치 서비스와 해당 서비스에 연관된 올바른 시 매니페스트 및 시스템 구성 프로파일에 액세스합니다. [9장. 설치 사용자 정의](#)에서는 클라이언트 설치 시 시 서버가 사용할 올바른 시 매니페스트 및 시스템 구성 프로파일을 식별하는 방법을 설명합니다.

## AI 보안

TLS(전송 계층 보안) 프로토콜로 자동 설치를 보호할 수 있습니다. 개인 인증서 및 키 쌍과 CA(인증 기관) 인증서를 AI 서버와 클라이언트에 지정할 수 있습니다. OBP HMAC 및 암호 키를 사용하여 SPARC 클라이언트 보안을 강화할 수 있습니다. 자세한 내용은 [“자동 설치의 보안 수준 향상” \[106\]](#)을 참조하십시오.

## AI 및 영역

비전역 영역의 설치를 지정한 경우 설치 후 처음 부트 시, 이러한 영역이 설치 및 구성됩니다. AI 클라이언트 설치의 일부로 비전역 영역의 설치 및 구성을 지정하는 방법은 [12장. 영역 설치 및 구성](#)을 참조하십시오.

## AI 구성 프로세스 개요

이 절에서는 주요 AI 사용 방법을 설명합니다. 이 작업 중 일부는 자동으로 수행할 사용자 정의 양에 따라 선택적입니다. [“자동 설치 프로그램 사용 사례” \[80\]](#)에서는 상황에 맞는 작업에 대해 자세히 설명합니다.

1. 필요한 구성을 제공하여 AI 클라이언트를 지원하도록 AI 서버를 구성합니다. 자세한 내용은 [AI 서버를 설정하는 방법 \[91\]](#)을 참조하십시오. 서버가 구성되면 다음이 수행됩니다.

- 올바른 OS가 설치됩니다.
- 정적 IP 주소를 사용하여 네트워크 인터페이스가 구성됩니다.
- `installadm` 패키지가 설치됩니다.

선택적으로 멀티캐스트 DNS가 사용으로 설정될 수 있습니다.

2. 설치 서비스를 만들어 클라이언트에서 사용할 부트 이미지를 만듭니다. 설치할 클라이언트 아키텍처 및 OS 버전마다 설치 서비스를 만들어야 합니다. AI 서버에서 특정 아키텍처의 첫번째 설치 서비스를 만들 때 해당 서비스의 별칭인 `default-i386` 또는 `default-sparc`가 자동으로 생성됩니다. `create-client` 하위 명령을 사용하여 다른 설치 서비스와 명시적으로 연관되지 않은 경우 이 기본 서비스가 해당 구조의 모든 클라이언트 설치에 사용됩니다. 자세한 내용은 [“설치 서비스 만들기” \[96\]](#)를 참조하십시오.
3. (선택 사항) 클라이언트를 서비스와 연관시킵니다. [“클라이언트를 서비스와 연관” \[103\]](#)을 참조하십시오. 여러 버전의 OS로 클라이언트 아키텍처 하나를 지원하는 경우 이 단계가 필요합니다.
4. (선택 사항) AI 매니페스트를 만들어 클라이언트 설치에 사용되는 패키지 저장소 또는 아카이브의 위치를 정의합니다. 자세한 내용은 [10장. 클라이언트 시스템 프로비전](#)을 참조하십시오. AI 매니페스트를 만들 때 기준을 지정하여 매니페스트를 특정 클라이언트

엔트에 연관시킬 수 있습니다. 자세한 내용은 “클라이언트를 설치 및 구성 지침과 일치” [133]를 참조하십시오. SI 매니페스트를 만든 후에는 매니페스트를 설치 서비스와 연관시켜야 합니다. 자세한 내용은 “클라이언트별 설치 지침을 설치 서비스와 연관” [104]을 참조하십시오. 클라이언트에 대한 설치를 사용자 정의하려면 반드시 매니페스트를 만들어 설치 서비스와 연관시켜야 합니다. 사용자 정의할 수 있는 몇 가지 정보는 다음과 같습니다.

- 디스크 레이아웃
- IPS 저장소 또는 아카이브 위치
- 설치할 로케일
- 설치할 패키지

5. (선택 사항) 시스템 구성 프로파일을 한 개 이상 만들어 클라이언트의 추가 구성 정보를 제공합니다. 자세한 내용은 11장, 클라이언트 시스템을 참조하십시오. 시스템 구성 파일을 만들 때 기준을 지정하여 프로파일을 특정 클라이언트에 연관시킬 수 있습니다. 자세한 내용은 “클라이언트를 설치 및 구성 지침과 일치” [133]를 참조하십시오. 시스템 구성 프로파일을 만든 후에는 프로파일을 설치 서비스와 연관시켜야 합니다. 자세한 내용은 “클라이언트별 구성 지침을 설치 서비스와 연관” [105]을 참조하십시오. 다음을 자동으로 구성하려면 반드시 하나 이상의 시스템 구성 프로파일을 만들어 설치 서비스와 연관시켜야 합니다.

- SMF 서비스(등록 정보 사용, 사용 안함 및 설정 포함)
- 사용자 계정 및 그룹
- 노드 이름
- 시간대 및 로케일
- 터미널 유형 및 키보드 레이아웃
- 네트워크 인터페이스
- 이름 서비스
- Oracle Configuration Manager 및 Oracle ASR(Auto Service Request)

시스템 구성 프로파일을 만들어 이 항목 중 일부나 전부를 구성하도록 선택할 수 있습니다. 클라이언트가 설치된 후 시스템 구성 프로파일에 지정되지 않은 항목은 수동으로 구성해야 합니다. 프로파일을 사용하지 않도록 선택할 경우 설치가 완료된 후 변경하라는 메시지가 표시됩니다.

6. (선택 사항) 처음 부트 스크립트를 만들어 SI 매니페스트나 프로파일에 추가할 수 없는 구성 요구 사항을 제공합니다. 자세한 내용은 13장, 처음 부트 중 사용자 정의 스크립트 실행을 참조하십시오.
7. 필요할 경우 클라이언트가 DHCP 서버 및 IPS 서버 또는 아카이브용 호스트에 액세스할 수 있는지 확인합니다.
8. 설치 서비스를 사용하여 네트워크를 통해 클라이언트를 부트합니다.

## SI 클라이언트 부트

SI 클라이언트를 네트워크 부트할 때 다음 작업이 수행됩니다.

1. 클라이언트가 DHCP 서버에서 AI 서버 주소를 가져옵니다.  
SPARC 클라이언트는 선택적으로 OBP에 설정된 `network-boot-arguments` 변수에서 AI 서버 주소를 가져올 수 있습니다.
2. 클라이언트가 AI 서버에서 부트 프로그램을 가져오고 부트 프로그램을 로드합니다.
  - SPARC 클라이언트가 AI 서버에서 `wanboot` 부트 프로그램을 가져옵니다.
  - X86 클라이언트가 GRUB 메뉴(설치 서비스 이름 포함)와 `pxegrub` 부트 프로그램을 AI 서버에서 가져옵니다.
3. 클라이언트가 AI 서버에서 부트 아카이브를 다운로드하고 커널을 로드합니다.
4. 클라이언트가 HTTP를 사용하여 설치 프로그램을 다운로드합니다.  
SPARC 클라이언트가 설치 프로그램을 사용하여 설치 서비스 이름을 가져옵니다.
5. AI 서버가 선택 조건에 따라 매니페스트를 선택합니다. 선택한 매니페스트를 기반으로 클라이언트에서 설치 변경이 수행됩니다.
6. AI 서버가 클라이언트 선택 조건에 따라 프로파일을 선택합니다.
7. AI 설치 프로그램이 IPS 저장소에서 패키지를 설치하거나 아카이브에서 이미지를 설치합니다.
8. 로컬 디스크에서 이미지를 사용하기 위해 클라이언트가 재부트됩니다.
9. 시스템 구성 프로파일이 사용되지 않은 경우 시스템 구성 정보를 묻는 메시지가 나타납니다.
10. 해당하는 경우 선택한 프로파일에 주어진 정보를 기반으로 클라이언트에서 시스템 구성이 변경됩니다.
11. 해당되는 경우 처음 부트 스크립트가 실행됩니다.

## AI 서버 계획

AI 서버를 설치하기 전에 이 절에서 설명한 대로 몇 가지 구성이 필요합니다.

### AI 서버의 네트워크 인터페이스 구성

여러 인터페이스가 있는 시스템에서 AI 서버를 구성할 수 있습니다. 기본적으로 AI 서버는 모든 네트워크 인터페이스에서 클라이언트를 지원하도록 구성됩니다. AI 서버를 호스트하는 시스템에 여러 네트워크 인터페이스가 있을 경우 [예 8-1. “네트워크에서 AI 지원 사용 안함”](#)에 나온 대로 서버에서 지원하지 않을 네트워크에 대해 설치 서비스를 사용 안함으로 설정할 수 있습니다.

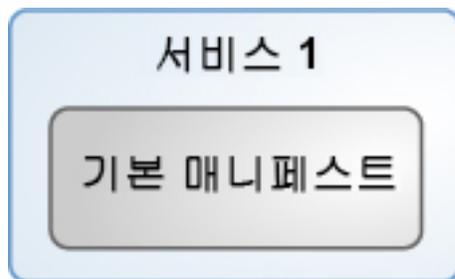
## 필요한 설치 인스턴스 식별

각 AI 서버에는 설치 서비스가 하나 이상 포함됩니다. 각 설치 서비스는 아키텍처(SPARC 또는 x86) 및 제공된 OS 릴리스(Oracle Solaris 11.1 또는 11.2 등)를 지원하도록 설정됩니다. 가능한 옵션 중 1개나 2개만 필요할 수 있습니다. 예를 들어, x86 클라이언트만 있는데 Oracle Solaris 11.1 또는 11.2 설치 옵션이 필요할 경우 OS 버전마다 하나씩, 설치 서비스 2개를 만들어야 합니다. x86 클라이언트와 SPARC 클라이언트가 모두 있고 Oracle Solaris 11.2 설치만 지원하려면 아키텍처마다 설치 서비스를 만듭니다. 자세한 내용은 “[설치 서비스 만들기](#)” [96]를 참조하십시오.

## 자동 설치 프로그램 사용 사례

이 절의 예에서는 AI 서버에서 서비스를 구성하는 주요 단계를 보여줍니다. 사례는 가장 간단한 서비스 구성에서 시작하여 고급 구성으로 발전합니다. 설치 서비스를 설정하는 구성 단계의 대부분은 선택 사항이며 각자의 환경에 따라 사용 여부가 결정됩니다.

그림 7-2 아키텍처 하나와 OS 하나를 지원하는 AI 서버



예 7-1 아키텍처 하나와 OS 하나를 지원하는 AI 서버

이 예에서는 OS 하나를 사용하여 클라이언트 아키텍처 하나만 지원하도록 AI 서버가 구성됩니다. 이 환경을 만들려면 다음을 수행합니다.

1. AI 서버를 구성합니다. [AI 서버를 설정하는 방법](#) [91]을 참조하십시오.
2. 설치 서비스 만들기. “[설치 서비스 만들기](#)” [96]를 참조하십시오.

이 예에서는 기본 AI 매니페스트를 포함하는 설치 서비스가 AI 서버에 포함됩니다. 클라이언트가 설치 후 부트될 때, 구성된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템

구성 정보를 묻습니다. 추가 구성 단계는 수동으로 수행해야 합니다. 시스템 구성 프로파일을 추가하여 시스템 구성 단계를 자동화할 수 있습니다.

그림 7-3 아키텍처 2개를 지원하는 AI 서버



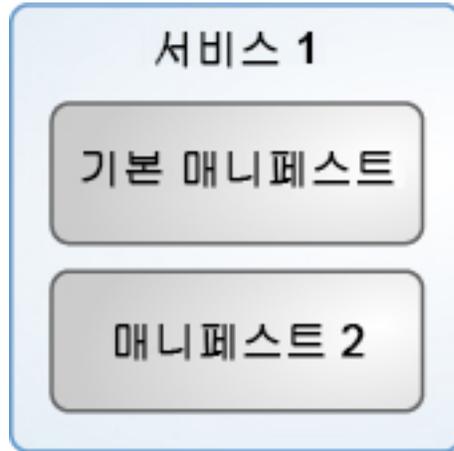
예 7-2 아키텍처 2개를 지원하는 AI 서버

이 예에서는 클라이언트 아키텍처 2개를 지원하도록 AI 서버가 구성됩니다. 이 환경을 만들려면 다음을 수행합니다.

1. AI 서버를 구성합니다. [AI 서버를 설정하는 방법 \[91\]](#)을 참조하십시오.
2. x86 클라이언트용 설치 서비스를 만듭니다. [“설치 서비스 만들기” \[96\]](#)를 참조하십시오.
3. SPARC 클라이언트용 설치 서비스를 만듭니다.

이 예에서는 클라이언트 아키텍처마다 하나씩, 설치 서비스 2개가 AI 서버에 포함됩니다. 각 서비스는 기본 매니페스트를 포함합니다. 클라이언트가 설치 후 부트될 때, 구성된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다. 추가 구성 단계는 수동으로 수행해야 합니다. 프로파일을 추가하여 시스템 구성 단계를 자동화할 수 있습니다.

그림 7-4 아키텍처 하나와 디스크 레이아웃 2개를 지원하는 AI 서버



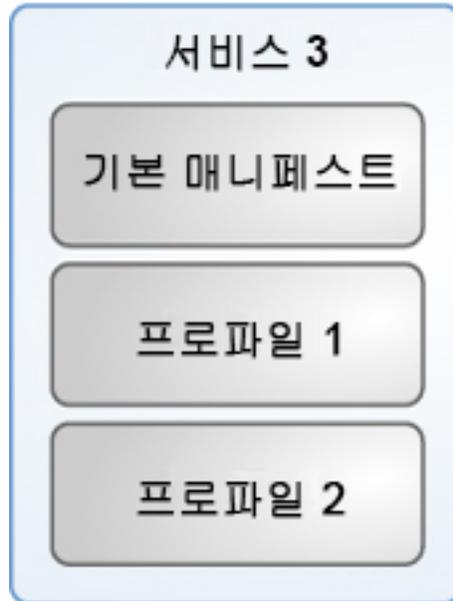
예 7-3 아키텍처 하나와 디스크 구성 2개를 지원하는 AI 서버

이 예에서는 클라이언트 디스크 레이아웃에 대한 구성이 2개인 클라이언트 아키텍처 하나만 지원하도록 AI 서버가 구성됩니다. 이 환경을 만들려면 다음을 수행합니다.

1. AI 서버를 구성합니다. [AI 서버를 설정하는 방법 \[91\]](#)을 참조하십시오.
2. 클라이언트용 설치 서비스를 만듭니다. [“설치 서비스 만들기” \[96\]](#)를 참조하십시오.
3. 두번째 디스크 레이아웃의 AI 매니페스트를 만듭니다. [10장. 클라이언트 시스템 프로비전](#)을 참조하십시오.
4. 두번째 매니페스트를 설치 서비스와 연관시킵니다. [“클라이언트별 설치 지침을 설치 서비스와 연관” \[104\]](#)을 참조하십시오.

이 예에서는 AI 매니페스트가 2개 있는 설치 서비스 하나가 AI 서버에 포함됩니다. 기본 매니페스트에 디스크 레이아웃 하나가 포함됩니다. 두번째 매니페스트에 두번째 디스크 레이아웃이 포함됩니다. 두번째 매니페스트는 이 매니페스트를 사용할 클라이언트의 식별 조건에 연관되어 있습니다. 다른 모든 클라이언트에서는 기본 매니페스트를 사용합니다. 클라이언트가 설치 후 부트될 때, 구성된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다. 추가 구성 단계는 수동으로 수행해야 합니다. 프로파일을 추가하여 시스템 구성 단계를 자동화할 수 있습니다.

그림 7-5 아키텍처 하나와 디스크 구성 2개를 지원하는 SI 서버



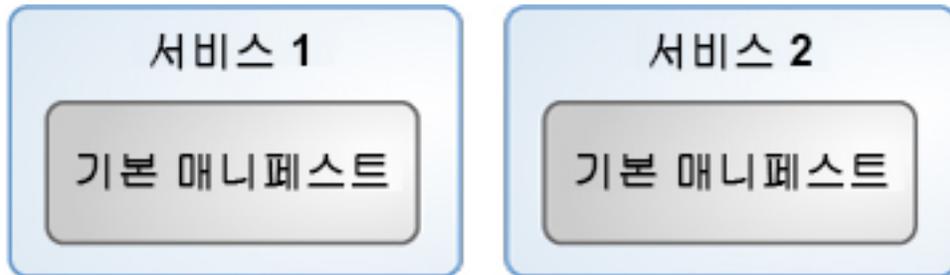
예 7-4 아키텍처 하나와 시간대 2개를 지원하는 SI 서버

이 예에서는 시간대를 설정하는 시스템 구성이 2개인 클라이언트 아키텍처 하나만 지원하도록 SI 서버가 구성됩니다. 이 환경을 만들려면 다음을 수행합니다.

1. SI 서버를 구성합니다. [SI 서버를 설정하는 방법 \[91\]](#)을 참조하십시오.
2. 클라이언트용 설치 서비스를 만듭니다. [“설치 서비스 만들기” \[96\]](#)를 참조하십시오.
3. 시간대 하나에 대한 시스템 구성 프로파일을 만듭니다. [11장. 클라이언트 시스템 구성](#)을 참조하십시오.
4. 두번째 시간대에 대한 시스템 구성 프로파일을 만듭니다.
5. 두 시스템 구성 프로파일을 모두 설치 서비스와 연관시킵니다. [“클라이언트별 구성 지침을 설치 서비스와 연관” \[105\]](#)을 참조하십시오.

이 예에서는 기본 매니페스트 및 시간대에 서로 다른 값을 설정하는 프로파일 2개를 포함하는 설치 서비스 하나가 SI 서버에 포함됩니다. 각 프로파일은 어느 클라이언트가 어느 프로파일을 사용할지 식별하는 조건에 연관되어 있습니다. 클라이언트가 설치 후 부트될 때, 시스템 구성 프로파일에 포함되지 않은 추가 구성 정보는 모두 수동으로 구성해야 합니다.

그림 7-6 아키텍처 하나와 릴리스 2개를 지원하는 AI 서버



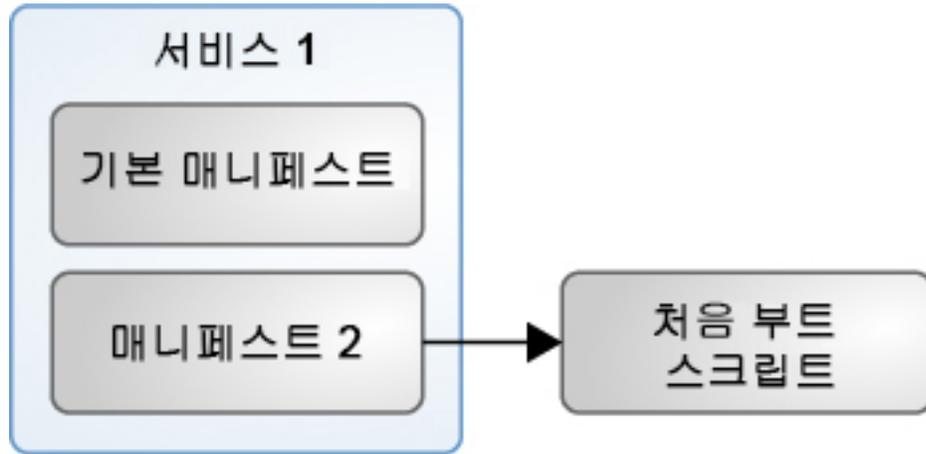
예 7-5 아키텍처 하나와 릴리스 2개를 지원하는 AI 서버

이 예에서는 Oracle Solaris OS 버전이 2개인 클라이언트 아키텍처 하나만 지원하도록 AI 서버가 구성됩니다. 이 환경을 만들려면 다음을 수행합니다.

1. AI 서버를 구성합니다. [AI 서버를 설정하는 방법 \[91\]](#)을 참조하십시오.
2. 첫번째 OS에 대해 설치 서비스를 만듭니다. [“설치 서비스 만들기” \[96\]](#)를 참조하십시오.
3. 두번째 OS에 대해 두번째 설치 서비스를 만듭니다. [“설치 서비스 만들기” \[96\]](#)를 참조하십시오.
4. 클라이언트를 적합한 서비스에 연관시키도록 AI 클라이언트를 정의합니다. [“클라이언트를 서비스와 연관” \[103\]](#)을 참조하십시오.

이 예에서는 설치 서비스 2개가 AI 서버에 포함됩니다. 클라이언트 정의에 따라 어느 클라이언트가 어느 서비스를 사용할지 결정됩니다. 클라이언트가 설치 후 부트될 때, 구성된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다. 추가 구성 단계는 수동으로 수행해야 합니다. 프로파일을 추가하여 시스템 구성 단계를 자동화할 수 있습니다.

그림 7-7 일부 클라이언트에 대한 추가 구성이 포함된 아키텍처 하나를 지원하는 AI 서버



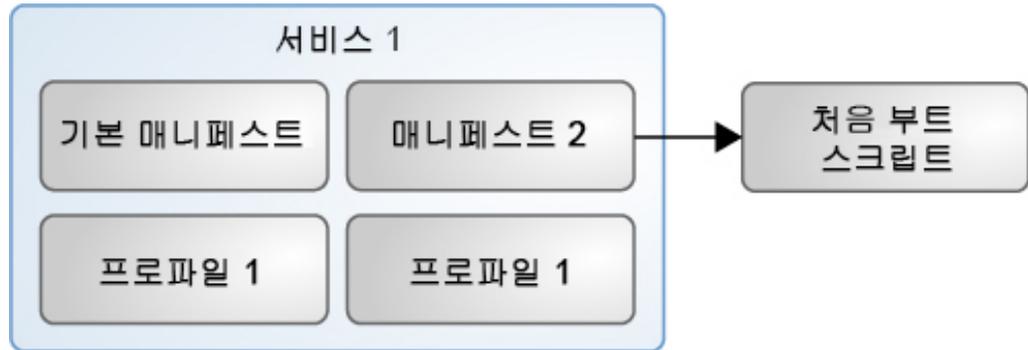
예 7-6 일부 클라이언트에 대한 추가 구성이 포함된 아키텍처 하나를 지원하는 AI 서버

이 예에서는 클라이언트 아키텍처 하나만 지원하도록 AI 서버가 구성되지만 클라이언트 세트 하나에는 매니페스트나 프로파일에서 수행할 수 없는 추가 구성이 필요합니다. 이 환경을 만들려면 다음 단계를 수행하십시오.

1. AI 서버를 구성합니다. [AI 서버를 설정하는 방법 \[91\]](#)을 참조하십시오.
2. 클라이언트용 설치 서비스를 만듭니다. [“설치 서비스 만들기” \[96\]](#)를 참조하십시오.
3. 처음 부트 스크립트를 만듭니다. [13장. 처음 부트 중 사용자 정의 스크립트 실행](#)을 참조하십시오.
4. 처음 부트 스크립트가 포함된 패키지를 만들고 패키지 저장소에 패키지를 추가합니다. [“스크립트 및 서비스에 대한 IPS 패키지 만들기” \[213\]](#)를 참조하십시오.
5. 처음 부트 패키지가 포함된 두번째 매니페스트를 만듭니다. [“클라이언트별 설치 지침을 설치 서비스와 연관” \[104\]](#)을 참조하십시오.
6. 두번째 매니페스트를 설치 서비스와 연관시킵니다. [“클라이언트별 설치 지침을 설치 서비스와 연관” \[104\]](#)을 참조하십시오.

이 예에서는 매니페스트가 2개 있는 설치 서비스 하나가 AI 서버에 포함됩니다. 두번째 매니페스트에는 클라이언트가 재부트된 후 처음 부트 스크립트를 실행하는 처음 부트 서비스 패키지를 설치하기 위한 지침이 포함됩니다. 두번째 매니페스트는 이 매니페스트를 사용할 클라이언트의 식별 조건에 연관되어 있습니다. 다른 모든 클라이언트에서는 기본 매니페스트를 사용합니다. 설치가 완료된 후 클라이언트가 부트될 때 처음 부트 스크립트가 실행됩니다. 그런 다음 구성된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다. 추가 구성 단계는 수동으로 수행해야 합니다. 프로파일을 추가하여 시스템 구성 단계를 자동화할 수 있습니다.

그림 7-8 여러 구성 변경을 지원하는 SI 서버



예 7-7 여러 구성 변경을 지원하는 SI 서버

이 예에서는 구성이 2개인 아키텍처 하나를 지원하도록 SI 서버가 구성됩니다. 첫번째 구성은 매우 기본적인입니다. 두번째 구성을 사용하여 다른 디스크 레이아웃을 선택하고 추가 구성을 수행할 수 있습니다. 두 구성은 같은 시스템 구성 프로파일을 사용하여 시간대를 구성합니다. 이 환경을 만들려면 다음을 수행하십시오.

1. SI 서버를 구성합니다. [SI 서버를 설정하는 방법 \[91\]](#)을 참조하십시오.
2. 클라이언트용 설치 서비스를 만듭니다. [“설치 서비스 만들기” \[96\]](#)를 참조하십시오.
3. 처음 부트 스크립트를 만듭니다. [13장. 처음 부트 중 사용자 정의 스크립트 실행](#)을 참조하십시오.
4. 처음 부트 스크립트가 포함된 패키지를 만들고 패키지 저장소에 패키지를 추가합니다. [“스크립트 및 서비스에 대한 IPS 패키지 만들기” \[213\]](#)를 참조하십시오.
5. 처음 부트 패키지가 포함된 두번째 매니페스트를 만들고 두번째 디스크 레이아웃을 정의합니다. [10장. 클라이언트 시스템 프로비전](#)을 참조하십시오.
6. 두번째 매니페스트를 설치 서비스와 연관시킵니다. [“클라이언트별 설치 지침을 설치 서비스와 연관” \[104\]](#)을 참조하십시오.
7. 시스템 구성 프로파일을 만들어 시간대를 설정합니다. [11장. 클라이언트 시스템 구성](#)을 참조하십시오.
8. 시스템 구성 프로파일을 설치 서비스와 연관시킵니다. [“클라이언트별 구성 지침을 설치 서비스와 연관” \[105\]](#)을 참조하십시오.

이 예에서는 매니페스트가 2개 있는 설치 서비스 하나가 SI 서버에 포함됩니다. 두번째 매니페스트에는 두번째 디스크 레이아웃에 대한 정보와 처음 부트 스크립트를 포함하는 처음 부트 서비스 패키지를 설치하기 위한 지침이 포함됩니다. 클라이언트가 재부트된 후 처음 부트 스크립트가 실행됩니다. 두번째 매니페스트는 이 매니페스트를 사용할 클라이언트의 식별 조건에 연관되어 있습니다. 다른 모든 클라이언트에서는 기본 매니페스트를 사용합니다. 두

매니페스트 중 하나를 사용하는 클라이언트는 프로파일에 연관된 조건에 따라 프로파일도 사용하도록 구성될 수 있습니다. 설치가 완료된 후 클라이언트가 부트될 때 처음 부트 스크립트가 실행됩니다. 시스템 구성 프로파일에 포함되지 않은 추가 구성 정보는 모두 수동으로 구성해야 합니다.



## AI 서버 설정

---

네트워크를 통해 AI 클라이언트를 설치하려면 AI에서 AI 서버로 작동할 별도의 시스템이 필요합니다. AI 서버에서 설치 서비스를 만들어 다양한 클라이언트에 원하는 Oracle Solaris 11 릴리스를 설치하기 위한 넷 이미지 및 지침을 제공합니다.

### AI 서버 설정 작업

AI 서버를 설정하기 위한 높은 레벨의 단계는 다음과 같습니다.

- 서버가 AI 서버가 되기 위한 최소 요구 사항을 충족하는지 여부를 확인합니다. 자세한 내용은 [“AI 서버 요구 사항” \[89\]](#)을 참조하십시오.
- AI 명령을 사용할 수 있게 만들기 위해 사용할 방법을 결정합니다. 전체 설명은 [“설치 서비스 작업 권한” \[90\]](#)을 참조하십시오.
- 정적 IP 주소 및 기본 경로를 사용하도록 AI 서버를 구성하고, AI 패키지를 설치하고, 필요한 경우 svc:/network/dns/multicast SMF 서비스를 사용으로 설정합니다. 자세한 내용은 [“AI 서버 구성” \[91\]](#)을 참조하십시오.
- 필요한 경우 설치 서비스를 사용으로 설정할 네트워크, AI 웹 서버 호스트 포트 번호 및 모든 이미지의 기본 이미지 경로와 같은 AI 서버에 대한 추가 설정을 변경합니다. 자세한 내용은 [“AI 서버의 구성 변경” \[93\]](#)을 참조하십시오.

### AI 서버 요구 사항

이 절에 설명된 요구 사항을 충족하는 모든 시스템(랩탑, 데스크탑, 가상 머신, 엔터프라이즈 서버 등)을 AI 서버로 사용할 수 있습니다. AI 서버는 x86 시스템 또는 SPARC 시스템일 수 있습니다. x86 AI 서버는 SPARC 및 x86 클라이언트를 모두 설치할 수 있고, SPARC AI 서버는 SPARC 및 x86 클라이언트를 모두 설치할 수 있습니다.

#### 운영 체제

AI 서버에 Oracle Solaris 11.2 OS를 설치합니다. 자세한 내용은 [설치 매체를 사용하여 설치 \[23\]](#)를 참조하십시오. 기존 서버에서 소프트웨어를 업데이트하는 방법은 [“Oracle Solaris 11.2로 업데이트”](#)를 참조

하십시오. AI 서버에서 최신 OS 버전을 사용하면 이전 버전 설치를 계속 지원하면서 새 기능을 모두 사용할 수 있습니다.

---

**참고** - 모든 Oracle Solaris 11 릴리스(업데이트 및 SRU 포함)를 11.2 클라이언트를 설치하는 AI 서버에서 설치된 OS로 사용할 수 있습니다.

---

<b>메모리</b>	최소 요구 사항은 1GB의 메모리입니다.
<b>디스크 공간</b>	AI 서버로 작동하는 데 필요한 추가 디스크 공간은 설정하는 설치 서비스 수에 따라 다릅니다. 설치할 서로 다른 클라이언트 아키텍처마다, 그리고 클라이언트 시스템에 설치할 서로 다른 버전의 Oracle Solaris 11 OS마다 별도의 설치 서비스가 필요합니다. 각각의 넷 이미지는 대략 300-400MB입니다.

## 설치 서비스 작업 권한

자동 설치에 사용되는 대부분의 명령에는 높은 수준의 권한이 필요합니다. 추가 권한을 얻으려면 다음 방법 중 하나를 사용하십시오.

<b>권한 프로파일</b>	profiles 명령을 사용하여 지정 받은 권한 프로파일을 나열합니다.  소프트웨어 설치 소프트웨어 설치 권한 프로파일이 있으면 pfexec 명령을 사용하여 패키지를 설치하고 업데이트할 수 있습니다.  <code>\$ pfexec pkg install install/installadm</code>
	설치 서비스 관리 설치 서비스 관리 권한 프로파일이 있으면 pfexec 명령을 사용하여 설치 서비스를 만들고 시스템 구성 프로파일을 설치 서비스에 추가할 수 있습니다.  <code>\$ pfexec installadm create-service</code>
	서비스 관리 서비스 관리 권한 프로파일이 있으면 SMF 서비스를 구성하고 사용으로 설정할 수 있습니다. 서비스 관리 권한 프로파일에는 pfexec가 필요 없습니다.  <code>\$ svcadm refresh system/install/server:default</code>
<b>sudo</b>	사이트의 보안 정책에 따라 사용자 암호와 함께 sudo 명령을 사용하여 권한이 있는 명령을 실행할 수 있는 경우도 있습니다.  <code>\$ sudo pkg install install/installadm</code>

**역할** roles 명령을 사용하여 지정 받은 역할을 나열합니다. 루트 역할이 있는 경우 루트 암호와 함께 su 명령을 사용하여 루트 역할을 가정할 수 있습니다.

## AI 서버 구성

이 절에서는 AI 클라이언트 설치 준비를 위해 AI 서버에서 수행할 수 있는 구성 중 일부를 설명합니다.

### ▼ AI 서버를 설정하는 방법

#### 1. Oracle Solaris 11.2를 설치합니다.

모든 버전의 Oracle Solaris 11을 AI 서버에 설치할 수 있지만 Oracle Solaris 11.2 릴리스에는 해당 버전이 서버에 설치된 경우에만 사용할 수 있는 여러 서버측 향상 기능이 포함되어 있습니다. 기존 Oracle Solaris 11 11/11 또는 11.1 AI 서버가 있는 경우의 AI 서버 구성 지침은 해당 릴리스의 설명서를 참조하십시오.

#### 2. 관리자가 됩니다.

자세한 내용은 [“How to Use Your Assigned Administrative Rights” in “Oracle Solaris 11.2 Administration: Security Services”](#)을 참조하십시오.

#### 3. 네트워크 인터페이스 IP 주소가 정적인지 확인합니다.

이 예제에서 net0은 정적 IP 주소를 사용하여 구성됩니다.

```
# ipadm
NAME                CLASS/TYPE STATE      UNDER  ADDR
net0                 ip         ok         --     --
  net0/v4            static    ok         --     129.144.83.5/24
lo0                  loopback  ok         --     --
  lo0/v4             static    ok         --     127.0.0.1/24
  lo0/v6             static    ok         --     ::1/128
```

주소가 정적이 아닌 경우 다음 단계를 따르십시오.

#### a. 사용으로 설정된 네트워크 구성 프로파일을 확인합니다.

```
# netadm list
TYPE      PROFILE      STATE
ncp       DefaultFixed online
ncp       Automatic    disabled
loc       Automatic    offline
loc       NoNet        offline
loc       User         offline
```

```
loc          DefaultFixed  online
```

- b. 필요한 경우 DefaultFixed 네트워크 구성 프로파일을 사용하여 설정합니다.

```
# netadm enable -p ncp DefaultFixed
```

- c. IP 인터페이스를 만듭니다.

```
# ipadm create-ip net0
```

- d. 인터페이스에서 정적 IP를 구성합니다.

```
# ipadm create-addr -T static -a local=129.144.83.5/24 net0
```

- e. 구성을 확인합니다.

```
# ipadm
      NAME          CLASS/TYPE    STATE    UNDER    ADDR
      net0          loopback     ok       --       --
      net0/v4       static       ok       --       129.144.83.5/24
```

4. (옵션) AI 서버의 기본 경로를 설정합니다.

```
# route -p add default 192.144.83.1
```

5. AI 패키지를 설치합니다.

- a. AI 패키지가 아직 설치되지 않았는지 확인합니다.

```
# pkg list installadm
pkg list: no packages matching 'installadm' installed
```

- b. IPS 패키지 저장소에 AI 패키지가 포함되어 있는지 확인합니다.

```
# pkg list -a installadm
NAME (PUBLISHER)                                VERSION                                IFO
install/installadm                             0.5.11-0.175.1.0.0.24.0             ---
```

- c. AI 패키지를 설치합니다.

```
# pkg install install/installadm
Packages to install: 1
Create boot environment: No
Create backup boot environment: No
Services to change: 2

DOWNLOAD                                PKGS    FILES    XFER (MB)    SPEED
Completed                                1/1     72/72     0.3/0.3      0B/s
```

PHASE	ITEMS
Installing new actions	138/138
Updating package state database	Done
Updating image state	Done
Creating fast lookup database	Done
Reading search index	Done
Updating search index	1/1

## 6. mDNS(멀티캐스트 DNS)를 사용합니다.

mDNS 서비스를 사용하면 AI 클라이언트가 동일 서브넷의 다른 AI 서버에서 복제된 설치 서비스를 찾을 수 있습니다. mDNS에 대한 자세한 내용은 [“Oracle Solaris 11.2의 이름 지정 및 디렉토리 서비스 작업: DNS 및 NIS”](#)의 [“멀티캐스트 DNS 및 서비스 검색에 대한 설명”](#)을 참조하십시오.

### a. 필요한 경우 mDNS 패키지를 설치합니다.

```
# pkg install pkg:/service/network/dns/mdns
```

### b. 이름 서비스 스위치 정보를 업데이트합니다.

로컬 호스트를 확인할 수 있으려면 mdns를 소스로 포함하도록 name-service/switch 서비스의 config/host 등록 정보를 변경합니다. 예를 들면 다음과 같습니다.

```
# /usr/sbin/svccfg -s svc:/system/name-service/switch
svc:/system/name-service/switch> setprop config/host = astring: "files dns mdns"
svc:/system/name-service/switch> select system/name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch> quit
```

### c. mDNS 서비스를 사용으로 설정합니다.

```
# svcadm enable svc:/network/dns/multicast:default
```

이런 방식으로 mDNS를 사용으로 설정하면 업그레이드 및 재부트 후에도 변경 사항이 유지됩니다. 자세한 내용은 [svcadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## AI 서버의 구성 변경

installadm set-server 명령을 사용하여 AI 서버의 여러 구성 설정을 변경할 수 있습니다. 이러한 구성 설정에는 다음이 포함됩니다.

- AI 지원에서 제외할 네트워크
- AI 지원을 포함할 네트워크
- AI 웹 서버 포트

이 명령을 사용하려면 관리자여야 합니다. 자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)의 [“지정된 관리 권한 사용”](#)을 참조하십시오.

**예 8-1**            네트워크에서 AI 지원 사용 안함

기본적으로 AI 서버는 서버가 연결된 모든 네트워크에 AI 클라이언트를 제공하도록 구성됩니다. 이 예제에서 192.168.56.0/24 네트워크 인터페이스는 더 이상 AI 요청을 제공하지 않습니다.

```
# installadm set-server -L 192.168.56.0/24
```

**예 8-2**            AI 서버에서 지원하는 네트워크 포함

경우에 따라 제외할 네트워크를 나열하는 것보다 설치 서비스를 지원할 네트워크를 나열하는 것이 더 쉽습니다. 다음 명령은 두 네트워크에서 설치 서비스를 허용하는 방법을 보여줍니다.

```
# installadm set-server -l 205.10.11.0/24, 205.10.12.0/24
```

**예 8-3**            AI 웹 서버 호스트 포트 구성

AI 서버는 웹 서버를 사용하여 설치 서비스를 호스트합니다. 기본적으로 AI 웹 서버는 포트 5555에서 호스트됩니다. http://localhost:5555에서 설치 서비스 파일을 볼 수 있습니다. 웹 서버에 사용되는 포트 번호를 변경할 수 있습니다. 다음 명령은 AI 서버가 포트 7000에서 설치 서비스를 호스트하도록 구성합니다.

```
# installadm set-server -p 7000
```

**예 8-4**            보안 AI 웹 서버 호스트 포트 구성

보안 AI 서버는 웹 서버를 사용하여 설치 서비스를 호스트합니다. 기본적으로 보안 AI 웹 서버는 포트 5556에서 호스트됩니다. http://localhost:5556에서 설치 서비스 파일을 안전하게 볼 수 있습니다. 웹 서버에 사용되는 포트 번호를 변경할 수 있습니다. 다음 명령은 AI 서버가 포트 7001에서 보안 설치 서비스를 호스트하도록 구성합니다.

```
# installadm set-server -P 7001
```

**예 8-5**            기본 이미지 경로 구성

기본적으로 이미지는 /export/auto\_install의 *service-name* 디렉토리에 생성됩니다. 따라서 기본적으로 *service-name* 서비스의 넷 이미지는 /export/auto\_install/*service-name*에 생성됩니다. 다음 명령은 AI 서버가 기본적으로 /export/aiimages/*service-name*에 새 설치 서비스를 만들도록 구성합니다.

```
# installadm set-server -d /export/aiimages
```

**예 8-6**            AI 서버용 AI 클라이언트의 IP 주소 및 번호 정의

다음 예제에서는 설치 서버가 네트워크의 DHCP 서버 역할을 하도록 설정합니다. DHCP 서버는 10.80.239.150(-i)부터 시작하여 20개의 IP 주소(-c)를 제공하도록 설정됩니다. DHCP

서버가 아직 구성되지 않은 경우 ISC DHCP 서버가 구성됩니다. ISC DHCP 서버가 이미 구성된 경우 해당 DHCP 서버가 업데이트됩니다.

요청된 IP 범위가 설치 서버가 직접 연결된 서브넷에 없고 설치 서버가 멀티홈인 경우 `-B` 옵션을 사용하여 부트 파일 서버 주소(대개 이 시스템의 IP 주소)를 제공합니다. 이 옵션은 반드시 다중 IP 주소가 설치 서버에 구성되고 DHCP 릴레이가 사용될 때만 필요합니다. 다른 구성에서는 소프트웨어가 이를 자동으로 결정할 수 있습니다.

```
# installadm set-server -i 10.80.239.150 -c 20
```

**예 8-7** AI 서버에서 로컬 DHCP 서비스의 자동 업데이트 사용 안함

기본적으로 AI 서버에서 클라이언트 및 서비스 구성이 수정되면 로컬 ISC DHCP 구성이 자동으로 업데이트됩니다. 로컬 ISC DHCP 구성을 자동으로 유지 관리하지 않으려면 다음 명령을 사용합니다.

```
# installadm set-server -M
Changed Server
Disabling SMF service svc:/network/dhcp/server:ipv4
Refreshing SMF service svc:/system/install/server:default
```

**예 8-8** AI 서버에서 DHCP 업데이트 사용

ISC DHCP 구성 자동 업데이트가 사용 안함으로 설정된 경우 다음 명령을 사용하여 사용으로 설정할 수 있습니다.

```
# installadm set-server -m
Warning: AI server will now manage DHCP
Changed Server
Enabling SMF service svc:/network/dhcp/server:ipv4
```

## 웹 서버 사용자 파일 디렉토리 구성

AI 웹 서버는 `installadm` 명령을 사용하여 추가된 넷 이미지, AI 매니페스트 및 시스템 구성 프로파일을 제공합니다. AI 웹 서버는 AI 사용자 또는 관리자가 제공한 파일을 제공할 수도 있습니다.

보호할 필요가 없는 사용자 파일은 `svc:/system/install/server:default` SMF 서비스의 `all_services/webserver_files_dir` 등록 정보로 지정한 디렉토리에 저장할 수 있습니다. 이 등록 정보에는 기본값이 없습니다. 이 등록 정보의 값을 지정하는 경우 값이 로컬 시스템의 디렉토리여야 합니다. 이 디렉토리는 다음 URL에서 AI 웹 서버를 통해 볼 수 있습니다. 여기서 `server`는 AI 서버의 호스트 이름 또는 IP 주소이고 `port`는 [예 8-3. "AI 웹 서버 호스트 포트 구성"](#)에 설명된 AI 웹 서버 포트 번호입니다.

```
http://server:port/files
```

보호할 필요가 없는 사용자 파일은 `all_services/webserver_secure_files_dir` 등록 정보로 지정한 디렉토리에 저장할 수 있습니다. 이 등록 정보에는 기본값이 없습니다. 이 등록 정보의 값을 지정하는 경우 값이 로컬 시스템의 디렉토리여야 합니다. 이 디렉토리는 다음 URL에서 SI 웹 서버를 통해 볼 수 있습니다. 여기서 `server`는 SI 서버의 호스트 이름 또는 IP 주소이고 `secure-port`는 위의 예 8-4. “보안 SI 웹 서버 호스트 포트 구성”에 설명된 보안 SI 웹 서버 포트 번호입니다.

`https://server:secure-port/secure_files`

SI 매니페스트가 인증서 및 키를 요구하는 IPS 패키지 저장소를 지정할 경우 해당 게시자 자격 증명을 저장한 다음 SI 매니페스트에 이 URI를 지정할 수 있습니다. 보안 자격 증명이 지정된 클라이언트만 이 디렉토리에 액세스할 수 있습니다.

---

**작은 정보** - 보안성을 높이기 위해 `webserver_secure_files_dir` 디렉토리의 파일은 사용자 `webservd` 및 그룹 `webservd`가 소유해야 하며 전세계적 액세스 권한이 없어야 합니다.

---

## 설치 서비스 작업

SI 서버를 설정한 후에 다음 작업 중 일부를 수행할 수 있습니다. [installadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

- “설치 서비스 만들기” [96]
- “클라이언트를 설치 서비스와 연관” [102]
- “클라이언트별 설치 지침을 설치 서비스와 연관” [104]
- “클라이언트별 구성 지침을 설치 서비스와 연관” [105]
- 자동 설치에 대한 보안 구성 방법 [108]
- SI를 사용한 Kerberos 클라이언트 구성 방법 [117]
- “설치 서비스에 대한 정보 표시” [120]

## 설치 서비스 만들기

SI 서버는 여러 개의 설치 서비스를 가질 수 있습니다. 각 클라이언트 하드웨어 아키텍처마다, 그리고 설치할 서로 다른 버전의 Oracle Solaris 11 OS마다 별도의 설치 서비스를 만듭니다.

### ▼ 설치 서비스를 만드는 방법

설치하려는 각 클라이언트 아키텍처(SPARC 또는 x86)와 각 운영 체제(Oracle Solaris 11, 11.1 또는 11.2)에 대한 설치 서비스를 만듭니다.

### 1. 관리자가 됩니다.

자세한 내용은 [“How to Use Your Assigned Administrative Rights”](#) in [“Oracle Solaris 11.2 Administration: Security Services”](#)을 참조하십시오.

### 2. DHCP 서비스를 확인합니다.

로컬 또는 원격 DHCP 지원이 적절하게 설정되었는지 확인합니다.

### 3. 설치 서비스를 만듭니다.

사용할 수 있는 모든 옵션은 [installadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 예를 들면 다음과 같습니다.

```
installadm create-service -s source -y
```

<i>source</i>	넷 이미지의 데이터 소스를 지정합니다. 값은 IPS AI 넷 이미지 패키지의 FMRI 식별자일 수 있으며, 기본값은 <code>install-image/solaris-auto-install</code> 입니다. 값은 AI ISO 이미지 파일의 전체 경로 이름일 수도 있습니다.
<i>-y</i>	자동 생성된 이미지 경로의 사용을 확인하는 프롬프트를 표시하지 않습니다.

#### 예 8-9 AI 서버에서 DHCP가 사용으로 설정된 상태에서 ISO 파일을 사용하여 SPARC 설치 서비스 만들기

이 예제에서는 네트워크가 단일 서브넷으로 구성되고 AI 서버가 네트워크의 DHCP 서버 역할도 하는 SPARC 클라이언트용 설치 서비스를 만듭니다. DHCP 서버가 아직 구성되지 않은 경우 ISC DHCP 서버가 구성됩니다. ISC DHCP 서버가 이미 구성된 경우 해당 DHCP 서버가 업데이트됩니다.

AI 클라이언트가 AI 서버가 직접 연결된 서브넷에 없고 AI 서버가 멀티홈인 경우 `-b` 옵션을 사용하여 부트 파일 서버 주소(대개 이 시스템의 IP 주소)를 제공합니다. 이 옵션은 다중 IP 주소가 AI 서버에 구성되고 DHCP 릴레이가 사용될 때만 필요합니다. 다른 구성에서는 소프트웨어가 이를 자동으로 결정할 수 있습니다.

이 명령을 x86 ISO 파일과 함께 실행하는 경우 출력의 차이점은 ISO 파일의 이름, 만든 서비스의 이름과 아키텍처 유형 및 만든 부트 파일의 설명뿐입니다. x86 출력은 [예 8-10. “IPS 패키지를 사용하여 x86 설치 서비스 만들기”](#)을 참조하십시오.

```
# installadm create-service -s /var/tmp/images/sparc/sol-11_2-ai-sparc.iso -y
0% : Service svc:/network/dns/multicast:default is not online. Installation services will
not be advertised via multicast DNS.
0% : Creating service from: /var/tmp/images/sparc/sol-11_2-ai-sparc.iso
36% : Transferring contents
36% : Creating sparc service: solaris11_2-sparc
36% : Image path: /export/auto_install/solaris11_2-sparc
36% : Setting "solaris" publisher URL in default manifest to:
36% : http://pkg.oracle.com/solaris/release/
36% : Creating default-sparc alias
```

```

36% : Setting "solaris" publisher URL in default manifest to:
36% : http://pkg.oracle.com/solaris/release/
36% : Setting the default SPARC bootfile(s) in the local DHCP configuration to:
36% : http://10.80.239.2:5555/cgi-bin/wanboot-cgi
100% : Created Service: 'solaris11_2-sparc'
100% : Refreshing SMF service svc:/system/install/server:default
100% : Restarting SMF service svc:/network/dhcp/server:ipv4
# installadm list
Service Name      Status Arch  Type Secure Alias Aliases Clients Profiles Manifests
-----
default-sparc     on      sparc iso  no    yes  0      0      0      1
solaris11_2-sparc on      sparc iso  no    no   1      0      0      1
    
```

**예 8-10** IPS 패키지를 사용하여 x86 설치 서비스 만들기

이 예제에서는 로컬 DHCP 없이 x86 AI 서버를 사용하며, IPS 패키지의 넷 이미지를 사용하여 x86 클라이언트용 설치 서비스를 만듭니다. 이 명령은 옵션이 지정되지 않았을 때 기본 동작을 보여줍니다. IPS 패키지를 사용할 때 다른 정보가 제공되지 않는 경우 AI 클라이언트의 아키텍처가 AI 서버의 아키텍처와 일치하는 것으로 간주됩니다. 이 AI 서버가 SPARC 시스템인 경우 -a i386 옵션을 제공하여 x86 설치 서비스를 만들 것을 지정해야 합니다.

DHCP 구성에 필요한 부트 파일 외에, 이 명령 출력은 DHCP 구성에 필요한 부트 서버 IP도 제공합니다.

이 명령을 실행하여 SPARC를 만든 경우 출력의 차이점은 서비스의 이름과 아키텍처 유형 및 만든 부트 파일의 설명뿐입니다. SPARC 출력은 [예 8-9. "AI 서버에서 DHCP가 사용으로 설정된 상태에서 ISO 파일을 사용하여 SPARC 설치 서비스 만들기"](#)를 참조하십시오.

```

# installadm create-service -y
0% : Creating service from: pkg:/install-image/solaris-auto-install
0% : Using publisher(s):
0% : solaris: http://pkg.oracle.com/solaris/release/
5% : Refreshing Publisher(s)
7% : Startup Phase
15% : Planning Phase
61% : Download Phase
90% : Actions Phase
91% : Finalize Phase
91% : Creating i386 service: solaris11_2-i386
91% : Image path: /export/auto_install/solaris11_2-i386
91% : Setting "solaris" publisher URL in default manifest to:
91% : http://pkg.oracle.com/solaris/release/
91% : DHCP is not being managed by install server.
91% : Creating default-i386 alias
91% : Setting "solaris" publisher URL in default manifest to:
91% : http://pkg.oracle.com/solaris/release/
91% : DHCP is not being managed by install server.
91% : No local DHCP configuration found. This service is the default
91% : alias for all PXE clients. If not already in place, the following should
91% : be added to the DHCP configuration:
91% : Boot server IP: 10.80.239.2
    
```

```

91% : Boot file(s):
91% :   bios clients (arch 00:00): default-i386/boot/grub/pxegrub2
91% :   uefi clients (arch 00:07): default-i386/boot/grub/grub2netx64.efi
91% :
100% : Created Service: 'solaris11_2-i386'
100% : Refreshing SMF service svc:/system/install/server:default
# installadm list
Service Name      Status Arch  Type Secure Alias Aliases Clients Profiles Manifests
-----
default-i386      on      i386  pkg  no    yes  0      0      0      1
solaris11_2-i386  on      i386  pkg  no    no   1      0      0      1

```

#### 예 8-11 다른 아키텍처용 설치 서비스 만들기

기본적으로 설치 서비스를 만들 때 아키텍처는 AI 서버와 동일합니다. 다른 아키텍처의 서비스를 만들려면 `-a` 옵션을 사용하십시오. 다음 예제에서는 SPARC AI 서버에서 x86 서비스를 만듭니다.

```
# installadm create-service -n solaris11_2-i386 -a i386 -y
```

#### 예 8-12 자동으로 X86 클라이언트를 설치하는 서비스 만들기

x86 클라이언트에서 GRUB 메뉴의 기본 항목은 자동으로 AI를 시작하지 않습니다. 설치가 자동으로 시작되도록 GRUB 메뉴를 사용자 정의하려면 다음 명령을 사용합니다.

```
# installadm create-service -s /var/tmp/images/i386/sol-11_2-ai-x86.iso -y -b
install=true
```

## 설치 서비스를 만들 때 발생하는 동작

설치 서비스를 만들 때 AI SMF 서비스인 `system/install/server`가 사용으로 설정됩니다 (아직 사용으로 설정되지 않은 경우). 설치 서비스 이미지가 `/etc/netboot/svcname`에 마운트됩니다. SPARC 설치 서비스의 경우 `wanboot.conf` 파일이 설치 서비스 이미지의 루트에 있습니다. x86 설치 서비스의 경우 GRUB 메뉴가 설치 서비스 이미지의 루트에 있습니다.

특정 아키텍처의 첫번째 설치 서비스가 AI 서버에 생성될 때 해당 서비스의 별칭인 `default-i386` 또는 `default-sparc`가 자동으로 생성됩니다. 이 기본 서비스는 고유의 매니페스트와 프로파일을 가진 완전한 서비스이지만, 이 기본 서비스는 명시적으로 만든 서비스와 넷 이미지를 공유합니다. 이 기본 서비스는 `create-client` 하위 명령을 사용하여 다른 설치 서비스와 명시적으로 연관되지 않은 해당 아키텍처의 모든 클라이언트 설치에 사용됩니다.

`default-arch` 서비스 별칭의 서비스를 변경하려면 `set-service` 하위 명령을 사용하여 `aliasof` 등록 정보를 설정합니다. 서비스에 추가된 매니페스트와 프로파일은 별칭을 재설정 한 후에도 그대로 유지됩니다. 서비스가 사용하는 넷 이미지만 변경됩니다. `aliasof` 등록 정보 설정에 대한 자세한 내용은 “[설치 서비스 관리](#)” [126]를 참조하십시오. `default-arch` 서비스의 별칭으로 삼을 서비스의 넷 이미지를 업데이트하려면 “[기존 설치 서비스 업데이트](#)” [128]에 표시된 대로 `update-service` 하위 명령을 사용합니다.

default-arch 별칭이 새 설치 서비스로 변경되고 로컬 ISC DHCP 구성이 발견된 경우 all\_services/manage\_dhcp 등록 정보의 값이 true이면 이 기본 별칭 부트 파일이 해당 구조에 대한 기본 DHCP 서버급 부트 파일로 설정됩니다. all\_services/manage\_dhcp 등록 정보에 대한 자세한 내용은 예 8-7. “AI 서버에서 로컬 DHCP 서비스의 자동 업데이트 사용 안함”을 참조하십시오.

installadm create-service 명령은 포트 5555에 실행 중인 웹 서버에 넷 이미지를 제공합니다. 예를 들어, 웹 서버 주소가 http://10.80.238.5:5555/solaris11\_2-i386일 수 있습니다. 다른 포트를 사용하려면 예 8-3. “AI 웹 서버 호스트 포트 구성”을 참조하십시오.

installadm create-service 명령의 실행 결과로 다음 작업이 수행됩니다.

1. 설치 서비스 이름을 정의하지 않으면 이름이 자동으로 생성됩니다. 설치 서비스를 만들 때 명령줄에서 -n 옵션을 포함하여 서비스 이름을 지정할 수 있습니다.
2. 지정된 넷 이미지 소스 옵션이 없으면 이 패키지를 제공하는 AI 서버 게시자 목록의 첫 번째 게시자로부터 최신 버전의 install-image/solaris-auto-install 패키지가 검색됩니다.
3. 기본 설치 서비스 넷 이미지 디렉토리가 생성됩니다. 디렉토리 이름에는 /export/auto\_install/solaris11\_2-sparc 또는 /export/auto\_install/solaris11\_2-i386과 같은 서비스 이름이 포함됩니다. 확인 프롬프트를 표시하지 않으려면 -y 옵션을 지정합니다.
4. 넷 이미지의 소스에 따라 다음 두 작업 중 하나가 발생합니다.
  - a. 넷 이미지 소스 옵션을 지정하지 않은 경우 install-image/solaris-auto-install 패키지가 넷 이미지 디렉토리에 설치됩니다.  
기본적으로 설치된 install-image/solaris-auto-install 패키지의 변형은 AI 서버의 아키텍처와 일치합니다. AI 서버가 x86 시스템이고 이 서버에 SPARC 설치 서비스를 만들려는 경우 -a 옵션을 사용해야 합니다. -a 옵션에 대한 자세한 내용은 예 8-11. “다른 아키텍처용 설치 서비스 만들기”을 참조하십시오.
  - b. 넷 이미지 소스 옵션을 지정한 경우 이미지 파일의 압축이 해제되거나 이미지 파일이 넷 이미지 디렉토리에 설치됩니다.
5. 설치 서비스 아키텍처에 따라 파일이 생성됩니다.
  - SPARC 클라이언트의 경우: 이 서비스에 대한 wanboot.conf 파일이 /etc/netboot/wanboot.conf에 생성됩니다.
  - x86 클라이언트의 경우: GRUB 메뉴가 /etc/netboot/solaris11\_2-i386/grub.cfg에 마운트됩니다.
6. AI SMF 서비스 system/install/server를 새로 고쳐서 /export/auto\_install/service-name을 /etc/netboot/service-name으로 마운트합니다.
7. 이 AI 서버에 생성되는 첫 번째 SPARC 설치 서비스인 경우 default-sparc 서비스 별칭이 자동으로 생성됩니다. 또한 /export/auto\_install/service-name이 /etc/netboot/default-sparc로 마운트됩니다.  
첫 번째 x86 설치 서비스의 경우 default-i386 서비스 별칭이 생성되고 /etc/netboot/default-i386 마운트 지점이 생성됩니다.
8. SPARC 클라이언트의 경우 구성 파일 /etc/netboot/wanboot.conf가 /etc/netboot/default-sparc/wanboot.conf와 심볼릭 링크로 연결됩니다. 또한 구성 파일 /etc/

netboot/system.conf가 /etc/netboot/default-sparc/system.conf와 심볼릭 링크로 연결됩니다.

9. 필요한 경우 DHCP 서비스가 생성되고 IP 주소가 프로비전됩니다. DHCP 서비스가 이 서버에 이미 설정된 경우 `-i` 및 `-c` 옵션이 DHCP 서버를 이 서비스의 새 IP 주소로 업데이트합니다. `svc:/network/dhcp/server` 서비스는 online입니다.
10. 로컬 DHCP 서비스를 사용하지 않는 구성의 경우:
  - SPARC 클라이언트의 경우: DHCP 구성에 필요한 부트 파일 `http://10.80.238.5:5555/cgi-bin/wanboot-cgi`가 제공됩니다.
  - x86 클라이언트의 경우: DHCP 구성에 필요한 부트 서버 IP가 제공됩니다. DHCP 구성에 필요한 부트 파일 `default-i386/boot/grub/pxegrub2` 및 `default-i386/boot/grub/grub2netx64.efi`도 제공됩니다.
11. 로컬 ISC DHCP 서버가 이미 구성된 경우 새로운 `default-sparc` 또는 `default-i386` 별칭의 부트 파일이 일치하는 모든 클라이언트의 기본 부트 파일로 설정됩니다. 이 지정은 `-i` 및 `-c` 옵션의 사용 여부에 관계없이 발생합니다.

## AI 클라이언트를 지원하기 위한 예제 DHCP 구성 파일

이 절에서는 `installadm`을 사용하여 ISC DHCP 구성을 위해 DHCP 구성 파일에 정보를 추가하는 방법을 보여줍니다. ISC DHCP 구성에 대한 자세한 내용은 [“Oracle Solaris 11.2의 DHCP 작업”의 2 장, “ISC DHCP 서비스 관리”](#)를 참조하십시오.

### Oracle Solaris 11.2 i386 설치 서비스용 ISC DHCP 구성

다음 예제에서는 이전에 만든 Oracle Solaris 11.2 i386 설치 서비스용 ISC DHCP 구성을 위해 `installadm`을 사용하여 `-i` 및 `-c` 옵션으로 지정된 IP 주소를 `/etc/inet/dhcpd4.conf` 파일에 추가하는 방법을 보여줍니다.

```
subnet 10.80.239.0 netmask 25.255.255.0 {
    range 10.80.239.150 10.80.239.169;
    option broadcast-address 10.80.239.255;
    option routers 10.80.239.1;
    next-server 10.80.239.170;
}
```

다음 예제에서는 이전에 만든 `default-i386` Oracle Solaris 11.2 i386 설치 서비스용 ISC DHCP 구성을 위해 `installadm`을 사용하여 `/etc/inet/dhcpd4.conf` 파일에 기본 PXE 부트 파일을 설정하는 방법을 보여줍니다.

```
class "PXEBoot" {
    match if (substring(option vendor-class-identifier, 0, 9) = "PXEClient");
    if option arch = 00:00 {
        filename "default-i386/boot/grub/pxegrub2";
    } else if option arch = 00:07 {
```

```
        filename "default-i386/boot/grub/grub2netx64.efi";
    }
}
```

## Oracle Solaris 11 i386 설치 서비스용 ISC DHCP 구성

Oracle Solaris 11.2 서비스 대신 Oracle Solaris 11 i386 설치 서비스를 만드는 경우 다음 예제와 비슷한 출력이 표시됩니다.

```
If not already in place, the following should be added to the DHCP configuration:
    Boot server IP      : 10.134.125.136
    Boot file           : default-i386/boot/grub/pxegrub
```

다음 예제에서는 Oracle Solaris 11 i386 설치 서비스용 ISC DHCP 구성을 위해 `installadm`을 사용하여 `/etc/inet/dhcpd4.conf` 파일에 기본 PXE 부트 파일을 설정하는 방법을 보여줍니다.

```
class "PXEBoot" {
    match if (substring(option vendor-class-identifier, 0, 9) = "PXEClient");
    if option arch = 00:00 {
        filename "default-i386/boot/grub/pxegrub";
    }
}
```

## Oracle Solaris 11.2 sparc 설치 서비스용 ISC DHCP 구성

i386 서비스 대신 sparc 설치 서비스를 만든 경우 다음 예제와 비슷한 출력이 나타납니다.

```
If not already in place, the following should be added to the DHCP configuration:
Boot file: http://10.80.238.5:5555/cgi-bin/wanboot-cgi
```

다음 예제에서는 Oracle Solaris 11.2 sparc 설치 서비스용 ISC DHCP 구성을 위해 `installadm`을 사용하여 `/etc/inet/dhcpd4.conf` 파일에 기본 부트 파일을 설정하는 방법을 보여줍니다.

```
class "SPARC" {
    match if not (substring(option vendor-class-identifier, 0, 9) = "PXEClient");
    filename "http://10.80.238.5:5555/cgi-bin/wanboot-cgi";
}
```

## 클라이언트를 설치 서비스와 연관

`installadm create-client` 명령은 클라이언트를 특정 설치 서비스와 연관시킵니다. x86 클라이언트용 사용자 정의 클라이언트 설정도 제공할 수 있습니다. 추가 예제와 샘플 출력은 [“시 클라이언트 설정” \[221\]](#)을 참조하십시오.

`installadm delete-client` 명령은 설치 서비스에 대한 클라이언트 연관을 제거합니다.

## 클라이언트를 서비스와 연관

클라이언트는 하나의 설치 서비스와만 연관될 수 있습니다. `installadm create-client` 명령을 한번 이상 실행하고 매번 동일한 MAC 주소를 지정하면 마지막 지정된 설치 서비스에만 해당 클라이언트가 연관됩니다. 이 명령을 실행하려면 관리자여야 합니다.

시스템의 MAC 주소를 찾으려면 `dladm` 명령을 사용합니다. 자세한 내용은 `dladm(1M)` 매뉴얼 페이지를 참조하십시오.

### 예 8-13 SPARC 클라이언트를 서비스와 연관

다음 명령은 MAC 주소가 `00:14:4f:a7:65:70`인 클라이언트를 `solaris11_2-sparc` 설치 서비스에 추가합니다.

```
# installadm create-client -e 00:14:4f:a7:65:70 -n solaris11_2-sparc
```

### 예 8-14 x86: X86 클라이언트를 서비스에 연관시키고 출력을 직렬 회선으로 재지정

다음 예제에서는 x86 클라이언트를 추가하고 클라이언트별 `/etc/netboot/grub.cfg` 파일의 부트 등록 정보를 변경합니다. 이 예제에서는 설치 출력이 직렬 콘솔 장치로 재지정됩니다.

```
# installadm create-client -e c0ffec0ffee -n solaris11_2-i386 -b console=ttya
```

### 예 8-15 x86: X86 클라이언트의 부트 등록 정보 변경

x86 클라이언트 시스템의 경우 `-g` 옵션을 사용하여 클라이언트 부트 시 사용할 사용자 정의 GRUB2 메뉴를 지정할 수 있습니다. 이 예제에서는 이름이 `/etc/netboot/grub.custom.cfg`인 사용자 정의 GRUB2 메뉴를 지정합니다.

```
# installadm create-client -e c0ffec0ffee -n solaris11_2-i386 -G /etc/netboot/grub.custom.cfg
```

`-b` 및 `-g` 옵션은 동시에 사용할 수 없습니다.

## x86 클라이언트용 자동 AI 설치

AI 설치를 자동으로 시작하려면 `create-client` 하위 명령을 사용하여 클라이언트를 만들 때 `-b install=true` 옵션을 사용하십시오. 이 설정을 서비스의 모든 클라이언트에 적용하려면 `create-service` 하위 명령을 사용하여 서비스를 만들 때 이 옵션을 사용합니다.

## 클라이언트를 설치 서비스에서 삭제

`installadm delete-client` 명령을 사용하여 `macaddr` 클라이언트를 설치 서비스에서 연관 해제합니다.

```
installadm delete-client -e mac-addr
```

다음 명령은 MAC 주소가 `00:14:4f:a7:65:70`인 클라이언트를 삭제합니다. 클라이언트는 하나의 설치 서비스에만 연관될 수 있으므로 서비스 이름을 지정할 필요가 없습니다.

```
# installadm delete-client -e 00:14:4f:a7:65:70
```

## 설치 지침 사용자 정의

SI 매니페스트 또는 파생된 매니페스트 스크립트를 사용하여 설치 서비스에 대한 특정 클라이언트 설치 지침을 제공할 수 있습니다. 시스템 구성 프로파일은 구성 지침을 제공합니다. 여러 시스템 구성 프로파일을 한 서비스 또는 한 SI 클라이언트와 연관시킬 수 있습니다. 또 한 여러 서비스 간에 프로파일을 공유할 수 있습니다.

## 클라이언트별 설치 지침을 설치 서비스와 연관

`installadm create-manifest` 명령을 사용하여 사용자 정의 SI 매니페스트를 특정 설치 서비스와 연관시킵니다. 파생된 매니페스트 스크립트를 설치 서비스에 추가할 수도 있습니다. 각 설치 서비스에 여러 SI 매니페스트 또는 파생된 매니페스트 스크립트가 연관되어 있을 수 있습니다. 지정된 서비스에 대한 기본 매니페스트로 구성되지 않은 매니페스트 또는 스크립트에는 클라이언트 조건이 정의되어야 각 클라이언트에 대해 올바른 지침이 사용됩니다.

다음 예제를 사용하기 전에 먼저 SI 매니페스트를 만들어야 합니다. SI 매니페스트 생성 지침은 [10장. 클라이언트 시스템 프로비전](#)을 참조하십시오.

명령의 구문은 다음과 같습니다.

```
installadm create-manifest -n service -f filename
```

**service**                    매니페스트 또는 파생된 매니페스트 스크립트를 연관시킬 서비스를 지정합니다.

**filename**                서비스와 연관시킬 매니페스트 또는 파생된 매니페스트 스크립트의 경로를 식별합니다.

**예 8-16**          클라이언트 조건을 매니페스트와 연관

이 예제에서는 manifest-sparc-ent.xml 매니페스트를 solaris11\_2-sparc 설치 서비스에 추가합니다. -c 옵션은 이 설치 서비스를 사용 중이며 M5000 또는 M4000 서버로 식별되는 모든 클라이언트에 manifest-sparc-ent.xml 설치 지침이 지정되도록 합니다. -m 옵션은 매니페스트의 AI 인스턴스 이름을 sparc-ent로 설정합니다.

```
# installadm create-manifest -n solaris11_2-sparc -f ./manifest-sparc-ent.xml \
-m sparc-ent -c platform="SUNW,SPARC-Enterprise"
```

**예 8-17**          클라이언트 조건을 스크립트와 연관

이 예제에서는 manifest-sparc-ent.xml 매니페스트를 solaris11\_2-sparc 설치 서비스에 추가합니다. 클라이언트 조건은 criteria-sparc-ent.xml 파일에서 정의됩니다.

```
# installadm create-manifest -n solaris11_2-sparc -f ./manifest-sparc-ent.xml \
-m sparc-ent -C ./criteria-sparc-ent.xml
```

criteria-sparc-ent.xml 파일의 내용은 다음과 같습니다.

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,SPARC-Enterprise</value>
  </ai_criteria>
</ai_criteria_manifest>
```

**예 8-18**          설치 서비스용 기본 매니페스트 만들기

이 예제에서는 -d 옵션을 사용하여 명명된 매니페스트 또는 스크립트가 이 서비스의 새 기본값임을 지정합니다. 매니페스트와 연관된 모든 클라이언트 조건이 저장되지만 이 매니페스트 또는 스크립트가 기본값인 경우 무시됩니다.

```
# installadm create-manifest -n solaris11_2-sparc -f ./manifest-sparc-ent.xml -d
```

## 클라이언트별 구성 지침을 설치 서비스와 연관

단일 클라이언트에서 여러 프로파일을 사용할 수 있으므로 한 create-profile 명령에서 여러 시스템 구성 프로파일을 지정할 수 있습니다. 여러 프로파일에 대해 동일한 클라이언트 선택 조건, 겹치는 조건 또는 조건 없음을 지정할 수 있습니다. 조건 없음이 지정된 경우 이 설치 서비스를 사용하는 모든 클라이언트에서 해당 프로파일을 사용합니다.

먼저 시스템 구성 프로파일을 만들어야 합니다. 지침은 [11장. 클라이언트 시스템 구성](#)을 참조하십시오.

명령의 구문은 다음과 같습니다.

```
# installadm create-profile -n service -f filename
```

`service`                   프로파일을 연관시킬 서비스를 지정합니다.

`filename`                   서비스와 연관시킬 프로파일의 경로 이름을 식별합니다.

**예 8-19**                   클라이언트 조건을 시스템 구성 프로파일과 연관

다음 명령은 `profile-sparc-ent.xml` 프로파일을 `solaris11_2-sparc` 설치 서비스에 추가합니다. `-c` 옵션은 이 설치 서비스를 사용 중이며 M4000 또는 M5000 서버로 식별되는 모든 클라이언트에 `profile-sparc-ent.xml` 시스템 구성 정보가 지정되도록 합니다. `-p` 옵션은 프로파일의 이름을 `sparc-ent`로 설정합니다.

```
# installadm create-profile -n solaris11_2-sparc -f ./profile-sparc-ent.xml \
-p sparc-ent -c platform="SUNW,SPARC-Enterprise"
```

## AI SMF 서비스 관리

AI 서버에서 SMF 서비스 `svc:/system/install/server:default`는 AI 서버 응용 프로그램과 모든 설치 서비스의 전체적인 상태를 나타냅니다.

**예 8-20**                   AI SMF 서비스 사용

`installadm create-service` 명령을 실행할 때 AI SMF 서비스가 사용으로 설정됩니다. 기존 설치 서비스에 영향을 주는 다른 `installadm` 명령을 실행할 때도 AI SMF 서비스가 사용으로 설정됩니다. 수동으로 AI SMF 서비스를 사용으로 설정하려면 다음을 수행하십시오.

```
$ svcadm enable svc:/system/install/server:default
```

**예 8-21**                   AI SMF 서비스 사용 안함

AI SMF 서비스를 사용 안함으로 설정하려면 다음을 수행하십시오.

```
$ svcadm disable svc:/system/install/server:default
```

## 자동 설치의 보안 수준 향상

TLS(전송 계층 보안) 프로토콜로 자동 설치를 보호할 수 있습니다. TLS로 인증하려면 AI 서버 및 각 AI 클라이언트에 개인 인증서 및 키 쌍을 지정해야 합니다. 또한 인증서를 생성하고 서명하는 데 사용되는 CA(인증 기관) 인증서를 제공해야 합니다. SPARC 클라이언트용 보안을 사용으로 설정하려면 각 클라이언트에 대해 OBP HMAC 키와 암호 키를 생성해야 합니다. 이러한 키는 초기 네트워크 부트 파일의 다운로드도 보호합니다.

x86 클라이언트의 보안도 사용으로 설정할 수 있지만 x86 클라이언트는 PXEBoot를 사용하므로 초기 네트워크 부트 단계는 보호되지 않습니다. x86 클라이언트의 보안을 사용으로 설정하려면 CA 인증서와 클라이언트 인증서 및 키 파일을 포함하는 사용자 정의 SI 이미지에서 x86 설치 서비스를 만들어야 합니다. 보안 인증서를 포함하는 사용자 정의 SI 매체 빌드 방법은 “[사용자 정의 Oracle Solaris 11.2 설치 이미지 만들기](#)”의 3 장, “[이미지 빌드](#)”를 참조하십시오. 이 이미지에서 설치 서비스를 만든 후에는 해당 SI 이미지 작성 중 사용한 것과 동일한 보안 인증서를 사용하여 설치 서비스에 보안을 설정해야 합니다.

다음 방법으로 자동 설치를 보호할 수 있습니다.

- 서버 인증: 서버의 ID를 확인할 수 있습니다.
- 클라이언트 인증: 클라이언트의 ID를 확인할 수 있습니다.
- 자동 설치에 대한 액세스 제어
- 서버 데이터에 대한 액세스 제어
- 모든 클라이언트에 대해 또는 지정한 클라이언트에 대해 별도로 클라이언트 데이터 보호
- 네트워크를 통해 읽을 수 없도록 데이터 암호화
- 보호된 IPS 패키지 저장소에 액세스
- 웹 서버를 통해 사용자가 지정한 디렉토리를 안전하게 게시. 이 디렉토리에 액세스하려면 클라이언트 인증이 필요합니다.

SI 프로세스를 보호하는 것 외에 SI를 사용하여 SI 클라이언트에서 Kerberos를 프로비전하여 네트워크 내에서의 보안을 향상시킬 수 있습니다. 지침은 [SI를 사용한 Kerberos 클라이언트 구성 방법 \[117\]](#)을 참조하십시오.

## 보안 자격 증명 구성

`installadm` 명령을 사용하여 SI 서버, 지정된 SI 클라이언트, 지정된 설치 서비스의 클라이언트 및 아직 자격 증명이 없는 모든 클라이언트에 대한 보안 자격 증명을 구성합니다. TLS를 위해서는 웹 서버에 서버 자격 증명 필요하므로 클라이언트 인증 전에 SI 서버 인증을 구성합니다.

다음 작업을 완료하려면 `installadm` 명령을 사용합니다.

- **자동으로 자격 증명 생성.** 사용자가 제공한 자격 증명을 사용하지 않는 경우 `-g` 옵션을 사용하여 개인 X.509 인증서 및 키 쌍, X.509 CA 인증서 및 OBP 키를 자동으로 생성할 수 있습니다. 자세한 내용은 “[SI 서버 자격 증명 구성](#)” [110]을 참조하십시오.
- **사용자가 제공한 자격 증명 입력.** 사용자가 제공한 자격 증명을 사용하는 경우 `-c`, `-k` 및 `-a` 옵션을 사용하여 이러한 사용자 제공 자격 증명을 지정할 수 있습니다.  
CA 인증서만 지정하고(`-a` 옵션) 개인 인증서 및 키를 별도로 지정하거나(`-c` 및 `-k` 옵션) 세 옵션을 모두 한 명령에서 지정할 수 있습니다. `-c` 및 `-k` 옵션만 지정하는 경우 연관된 CA 인증서(`-a` 옵션)를 미리 지정해야 합니다. `-c` 및 `-k` 옵션은 둘 중 하나만 지정할 수 없으며 쌍으로 지정해야 합니다.  
`-c` 옵션의 인수는 PEM 인코딩 X.509 인증서 파일에 대한 경로입니다.

-k 옵션의 인수는 PEM 인코딩 X.509 개인 키 파일에 대한 경로입니다. 이 키 파일에서 문장암호는 제거되어야 합니다.

-A 옵션의 인수는 PEM 인코딩 X.509 CA(인증 기관) 인증서 파일에 대한 경로입니다. CA 인증서에는 고유한 주체 행이 있어야 합니다. 각 CA 트러스트 체인은 한 번만 지정하면 됩니다. CA 체인에 둘 이상의 CA 인증서 파일이 포함된 경우 한 `installadm` 명령에서 별도의 -A 옵션을 사용합니다.

OBP 키가 없는 경우 새로 생성됩니다. OBP 키가 생성된 경우 이러한 키를 설정하는 OBP 명령이 표시됩니다.

- **OBP 키 생성.** -g, -c, -k 또는 -A 옵션을 사용할 때 OBP 키가 없는 경우 자동으로 생성됩니다. -E 및 -H 옵션 사용에 대한 자세한 내용은 “[SPARC 클라이언트용 OBP 보안 키](#)” [114]를 참조하십시오.
- **자격 증명 표시.** 언제든지 `installadm list` 명령을 사용하여 AI 서버, 설치 서비스 또는 특정 클라이언트에 대한 현재 자격 증명 세트를 표시할 수 있습니다.

## 보안 우선 순위

적용할 보안 설정과 사용할 자격 증명을 결정할 때의 우선 순위는 다음과 같습니다.

1. AI 서버에 자격 증명 없음. 기본 상태입니다. 추가된 보안이 없고 클라이언트용 펌웨어 키가 모두 지워져야 합니다.
2. 서버측 보안 사용 안함. 추가된 보안이 없고 클라이언트용 펌웨어 키가 모두 지워져야 합니다.
3. 모든 클라이언트 서비스 정책이 `disable`으로 설정됩니다. 추가된 보안이 없고 클라이언트용 펌웨어 키가 모두 지워져야 합니다.
4. `set-client` 하위 명령으로 생성된 사용자 정의 클라이언트 자격 증명 사용
5. `set-service` 하위 명령으로 생성된 설치 서비스 자격 증명 사용
6. `set-server -D` 하위 명령으로 생성된 기본 클라이언트 자격 증명 사용
7. 클라이언트에 대한 클라이언트 자격 증명 없이 서비스 정책이 `require-server-auth`인 경우 기본 클라이언트 OBP 키 사용

## ▼ 자동 설치에 대한 보안 구성 방법

1. 관리자가 됩니다.  
자세한 내용은 “[How to Use Your Assigned Administrative Rights](#)” in “[Oracle Solaris 11.2 Administration: Security Services](#)”를 참조하십시오.
2. AI 서버에 대한 보안 자격 증명을 생성합니다.  
다음 명령은 X.509 루트 CA 인증서와 서명 CA 인증서, 서버 인증서 및 개인 키 및 AI 서버 인증용 OBP 키를 자동으로 생성합니다. CA 인증서와 OBP 키는 기존에 없는 경우에만 생성됩니다. OBP 키가 생성된 경우 이러한 키를 설정하는 OBP 명령이 표시됩니다.

```
# installadm set-server -g
The root CA certificate has been generated.
The CA signing certificate request has been generated.
The signing CA certificate has been generated.
A new certificate key has been generated.
A new certificate has been generated.
Generating new encryption key...
To set the OBP encryption key for server authentication only, enter
this OBP command:
set-security-key wanboot-aes 8d210964e95f2a333c5e749790633273
Generating new hashing key (HMAC)...
To set the OBP hashing (HMAC) key for server authentication only,
enter this OBP command:
set-security-key wanboot-hmac-sha1 4088861239fa3f3bed22f8eb885bfa476952fab4
Configuring web server security.
Changed Server
```

AI 서버 자격 증명 구성에 대한 자세한 내용은 “AI 서버 자격 증명 구성” [110]을 참조하십시오.

### 3. (옵션) 설치 서비스 보안 정책을 설정합니다.

다음 예제에서는 설치 서비스를 사용하기 위해 클라이언트 인증이 필요한 보안 설정을 지정합니다. 특정 설치 서비스와 연관된 모든 데이터 및 모든 클라이언트를 보호하려면 `require-client-auth` 설치 서비스 보안 설정을 사용하여 서버 및 클라이언트 인증을 모두 사용하여 모든 클라이언트가 보호되도록 합니다. 이 예에서 클라이언트가 `svcname` 설치 서비스 데이터에 액세스하려면 X.509 자격 증명이 있어야 합니다.

```
# installadm set-service -p require-client-auth -n svcname
```

설치 서비스 보안 정책 구성에 대한 자세한 내용은 “보안 설치 서비스 구성” [111]을 참조하십시오.

### 4. AI 클라이언트에 대한 자격 증명을 생성합니다.

다음 예제에서는 개인 X.509 인증서 및 키 쌍, 지정한 클라이언트 인증을 위한 X.509 CA 인증서를 자동으로 생성합니다. 여기에서 `02:00:00:00:00:00`은 클라이언트의 MAC 주소입니다. MAC 주소를 사용하여 지정한 클라이언트 자격 증명은 각 클라이언트마다 고유합니다. CA 인증서는 기존에 없는 경우에만 생성됩니다. 클라이언트가 SPARC 시스템이고 OBP 키가 기존에 없어 생성된 경우 이러한 키를 설정하는 OBP 명령이 표시됩니다.

```
# installadm set-client -e 02:00:00:00:00:00 -g
Generating credentials for client 02:00:00:00:00:00...
A new certificate key has been generated.
A new certificate has been generated.
Generating new encryption key...
To set the OBP encryption key, enter this OBP command:
set-security-key wanboot-aes 030fd11c98afb3e434576e886a094c1c
Generating new hashing key (HMAC)...
To set the OBP hashing (HMAC) key, enter this OBP command:
set-security-key wanboot-hmac-sha1 e729a742ae4ba977254a2cf89c2060491e7d86eb
Changed Client: '02:00:00:00:00:00'
```

클라이언트 자격 증명 구성에 대한 자세한 내용은 “클라이언트 자격 증명 구성” [112]을 참조하십시오.

5. SPARC 클라이언트용 OBP 키를 설정합니다.

보안 자격 증명에 지정된 SPARC 클라이언트의 경우 AI 설치용 클라이언트를 부트할 때 OBP 보안 키(해싱 키 및 암호 키)를 설정해야 합니다. 다음 예제에서는 SPARC 클라이언트 콘솔에서 OBP AES 암호 키를 설정합니다.

```
ok set-security-key wanboot-aes 030fd11c98afb3e434576e886a094c1c
```

다음 예제에서는 SPARC 클라이언트 콘솔에서 OBP 해싱(HMAC) 키를 설정합니다.

```
ok set-security-key wanboot-hmac-sha1 e729a742ae4ba977254a2cf89c2060491e7d86eb
```

자세한 내용과 예제는 “보안 다운로드를 사용하여 SPARC 클라이언트 설치” [224]를 참조하십시오.

6. 보안 IPS 저장소에서 설치할 AI 매니페스트를 수정합니다.

AI 매니페스트에서 보안 원본이 있는 게시자를 지정한 경우 credentials sub-element of the publisher 요소에서 키 및 인증서를 지정합니다. 자세한 내용은 ai\_manifest(4) 매뉴얼 페이지의 소프트웨어 절을 참조하십시오. image 요소의 속성에서 SSL 키 및 인증서를 지정할 수 있지만 이 키 및 인증서는 매니페스트에서 지정한 첫번째 게시자에만 적용됩니다. image 요소 및 credentials 요소에서 모두 키와 인증서를 지정한 경우 credentials 요소에서 지정한 자격 증명이 사용됩니다. 키 및 인증서 파일은 AI 웹 서버의 사용자 지정 디렉토리에 두는 것이 좋습니다. 자세한 내용은 “웹 서버 사용자 파일 디렉토리 구성” [95]을 참조하십시오.

## AI 서버 자격 증명 구성

AI 서버 보안은 다음과 같은 이점을 제공합니다.

- AI 클라이언트에서 AI 서버의 ID를 확인할 수 있습니다.
- AI 클라이언트에서 TLS를 통해 자동으로 암호화된 데이터를 수신하므로 네트워크 트래픽 모니터링을 통해 읽을 수 없습니다.

-C, -K 및 -A 옵션 사용에 대한 자세한 지침은 “보안 자격 증명 구성” [107]을 참조하십시오.

**예 8-22** 사용자 제공 자격 증명을 사용하여 AI 서버 자격 증명 생성

다음 예제에서는 사용자가 제공한 자격 증명을 지정합니다. OBP 키가 없는 경우 새로 생성됩니다. OBP 키가 생성된 경우 이러한 키를 설정하는 OBP 명령이 표시됩니다.

```
# installadm set-server -C server.crt -K server.key -A cacert.pem
```

CA 자격 증명을 지정하지 않은 경우 이러한 클라이언트 자격 증명을 생성하는 데 사용되는 CA 인증서를 미리 지정해야 합니다.

클라이언트의 설치 서비스에 정책 `require-server-auth`가 있고 클라이언트 또는 서비스 자격 증명에 지정되지 않은 경우 기본 클라이언트 OBP 키가 사용됩니다. 또한 기본 클라이언트 자격 증명에 사용 중이어야 합니다.

## 보안 설치 서비스 구성

`installadm create-service` 명령을 사용하여 설치 서비스를 만들 때 구성합니다. 자세한 내용은 [설치 서비스를 만드는 방법 \[96\]](#)을 참조하십시오. `installadm set-service` 명령을 사용하여 기존 설치 서비스를 재구성합니다. 이 절에서는 설치 서비스에 대한 보안 정책을 설정하는 방법을 설명합니다.

각 설치 서비스에는 하나의 보안 정책을 설정할 수 있습니다. 사용 가능한 선택 항목은 다음과 같습니다.

### `require-client-auth`

AI 클라이언트의 ID를 확인합니다. 지정한 서비스의 모든 클라이언트에 대해 클라이언트 및 서버 인증이 필요합니다. 또한 이 옵션에는 암호화도 필요합니다.

서비스의 모든 클라이언트가 클라이언트 인증으로 인증되어야 합니다. 지정한 서비스의 모든 클라이언트에는 자격 증명에 지정되어야 하고 이 서비스의 모든 SPARC 클라이언트에는 OBP 키가 정의되어야 합니다. 클라이언트 인증을 위해 구성되지 않은 서비스의 클라이언트는 이 설치 서비스를 사용할 수 없습니다.

### `require-server-auth`

AI 서버의 ID를 확인합니다. 지정한 서비스의 모든 클라이언트가 서버 인증을 수행해야 합니다. 또한 이 옵션에는 암호화도 필요합니다.

지정한 설치 서비스에 액세스하려면 최소한 AI 서버 인증이 필요합니다. 클라이언트 인증은 선택 사항이지만 지정된 클라이언트 자격 증명을 제공해야 합니다. 이 서비스의 모든 SPARC 클라이언트용 OBP 키도 정의해야 합니다.

### `optional`

인증되거나 인증되지 않은 클라이언트 모두에 대해 설치 서비스 액세스를 허용합니다. 서버에 자격 증명에 있는 경우에는 암호화도 필요합니다. 이 옵션이 기본 동작입니다.

지정된 클라이언트 자격 증명을 제공해야 합니다. 지정된 자격 증명에 없는 클라이언트는 OBP 키 또는 서버 인증을 사용하지 않습니다. 서버 인증은 클라이언트 인증을 위해 구성된 클라이언트의 경우에만 제공해야 합니다.

### `encr-only`

x86 클라이언트의 경우: 인증을 요구하지 않고 SSL/TLS 종단간 암호화를 사용으로 설정합니다. 인증이 없으면 클라이언트 및 서버의 ID를 보장할 수 없습니다. 타사가 네트워크를 통해 전송 중인 데이터를 읽을 수 없습니다.

disable

지정한 서비스의 모든 클라이언트에 대한 모든 보안을 사용 안함으로 설정합니다.

이 서비스의 클라이언트가 인증되지 않습니다. 자격 증명이 발행되지 않습니다. 이 서비스의 클라이언트는 “웹 서버 사용자 파일 디렉토리 구성” [95]에서 설명한 `webserver_secure_files_dir` 디렉토리에 액세스할 수 없습니다. 이 설정을 사용할 때는 이전에 인증에 의해 보호된 설치 서비스 파일은 더 이상 보호되지 않는다는 점에 주의하십시오. 클라이언트 데이터가 무단 액세스로부터 보호되지 않습니다. 인증을 다시 사용으로 설정하려면 다른 보안 정책 값으로 `set-service` 하위 명령을 다시 지정하십시오.

예 8-23 설치 중 AI 서버 인증 요구

이 예에서는 설치 서비스를 사용하기 위해 서버 인증이 필요한 보안 설정을 지정합니다. 최소한 AI 서버를 인증하기 위해 지정된 서비스의 클라이언트를 요구하려면 `require-server-auth` 설치 서비스 보안 설정을 사용합니다.

```
# installadm set-service -p require-server-auth -n install-service
```

예 8-24 x86: 설치 중 암호화 요구

이 예에서는 암호화를 사용하지만 인증을 요구하지 않는 보안 설정을 지정합니다. x86 클라이언트에서 특정 설치 서비스의 데이터 전송을 보호하지만 클라이언트 또는 서버 인증을 요구하지 않으려면 `encr-only` 보안 설정을 사용합니다. 서버 인증서는 여전히 필요합니다. 데이터가 네트워크를 통한 스누핑으로부터 보호되지만 AI 서버는 적절한 요청을 발행하는 모든 클라이언트에 데이터를 제공합니다.

```
# installadm set-service -p encr-only -n install-service
```

## 클라이언트 자격 증명 구성

클라이언트 AI 서버 보안은 다음과 같은 이점을 제공합니다.

- AI 서버에서 AI 클라이언트의 ID를 확인할 수 있습니다.
- 데이터가 네트워크를 통해 암호화됩니다.
- 사용자 정의 자격 증명에 있는 클라이언트의 경우 해당 클라이언트의 게시된 파일을 다른 클라이언트에서 읽을 수 없습니다.
- 인증된 클라이언트만 “웹 서버 사용자 파일 디렉토리 구성” [95]에서 설명한 사용자 지정 보안 디렉토리에 액세스할 수 있습니다.

특정 클라이언트, 특정 설치 서비스의 클라이언트 또는 자격 증명에 없는 클라이언트에 대해 자격 증명을 생성하거나 지정할 수 있습니다. 생성된 OBP 키는 양방향(클라이언트 및 서버) 인증을 위해 생성됩니다. SPARC 클라이언트에 보안 인증서를 지정한 경우 AI 설치용 클라이언트를 부트할 때 OBP 키를 제공해야 합니다. “보안 다운로드를 사용하여 SPARC 클라이언트 설치” [224]를 참조하십시오.

**참고** - create-client 하위 명령을 사용하여 설치 서비스 간에 클라이언트를 이동할 수 있습니다. 기존 클라이언트에서 보안 자격 증명과 함께 사용된 하위 명령은 해당 클라이언트의 자격 증명에서 유효하지 않습니다.

**예 8-25**            특정 클라이언트에 대해 사용자가 제공한 자격 증명 사용

이 예에서는 사용자가 제공한 자격 증명을 지정합니다. 클라이언트가 SPARC 시스템이고 OBP 키가 기존에 없는 경우 새로 생성됩니다. OBP 키가 생성된 경우 이러한 키를 설정하는 OBP 명령이 표시됩니다.

```
# installadm set-client -e 02:00:00:00:00:00 -C client.crt -K client.key -A cacert.pem
```

-C, -K 및 -A 옵션에 대한 자세한 내용은 “[보안 자격 증명 구성](#)” [107]을 참조하십시오. CA 자격 증명을 지정하지 않은 경우 이러한 클라이언트 자격 증명을 생성하는 데 사용되는 CA 인증서를 미리 지정해야 합니다.

-E 및 -H 옵션 사용에 대한 자세한 내용은 “[SPARC 클라이언트용 OBP 보안 키](#)” [114]를 참조하십시오.

**예 8-26**            특정 설치 서비스의 클라이언트용 자격 증명

이 예에서는 solaris11\_2-sparc 설치 서비스에 지정되고 자격 증명이 아직 지정되지 않은 클라이언트에 대해 자격 증명을 제공합니다.

```
# installadm set-service -g -n solaris11_2-sparc
Generating credentials for service solaris11_2-sparc...
A new certificate key has been generated.
A new certificate has been generated.
Generating new encryption key...
To set the OBP encryption key, enter this OBP command:
  set-security-key wanboot-aes 34bc980ccc8dfee478f89b5acbfd51b4
Generating new hashing key (HMAC)...
To set the OBP hashing (HMAC) key, enter this OBP command:
  set-security-key wanboot-hmac-sha1 b8a9f0b3472e8c3b29443daf7c9d448faad14fee
```

이 설치 서비스는 SPARC 설치 서비스이므로 OBP 키도 생성되고 이러한 키를 설정하는 OBP 명령이 표시됩니다.

이러한 클라이언트에 MAC 주소를 지정하여 자격 증명을 지정하지 않으면 이후 solaris11\_2-sparc 설치 서비스에 지정되는 클라이언트도 이 자격 증명을 사용합니다.

이 옵션은 여러 클라이언트에서 통일된 응용 프로그램 세트를 사용하려는 경우 유용합니다. 그러나 MAC 주소를 사용하여 자격 증명을 지정하지 않은 이 설치 서비스의 모든 클라이언트는 동일한 자격 증명을 갖게 되며 서로 설치 데이터를 볼 수 있습니다.

-C, -K 및 -A 옵션에 대한 자세한 내용은 “[보안 자격 증명 구성](#)” [107]을 참조하십시오.

-E 및 -H 옵션 사용에 대한 자세한 내용은 “[SPARC 클라이언트용 OBP 보안 키](#)” [114]를 참조하십시오.

**예 8-27** 기본 클라이언트 자격 증명

이 예에서는 자격 증명이 지정되지 않은 클라이언트에 대한 기본 자격 증명 세트를 제공합니다.

```
# installadm set-server -D -g
Generating default client credentials...
A new certificate key has been generated.
A new certificate has been generated.
Generating new encryption key...
To set the OBP encryption key, enter this OBP command:
    set-security-key wanboot-aes 7cdbda5b8fc4b10ffbd29fa19d13af77
Generating new hashing key (HMAC)...
To set the OBP hashing (HMAC) key, enter this OBP command:
    set-security-key wanboot-hmac-sha1 14effe2c515da4940ef1db165791e92790163004
```

일부 클라이언트가 SPARC 클라이언트일 수 있으므로 OBP 키도 생성되고 이러한 키를 설정하는 OBP 명령이 표시됩니다.

기본 클라이언트 자격 증명이 지정되면 모든 클라이언트가 클라이언트 및 서버 인증을 수행하고 SI 서버의 모든 SPARC 클라이언트에 대해 펌웨어 키가 필요하게 됩니다. 또한 여러 클라이언트가 동일한 자격 증명을 갖게 되므로 서로의 설치 데이터를 볼 수 있습니다.

-C, -K 및 -A 옵션에 대한 자세한 내용은 “[보안 자격 증명 구성](#)” [107]을 참조하십시오.

-E 및 -H 옵션 사용에 대한 자세한 내용은 “[SPARC 클라이언트용 OBP 보안 키](#)” [114]를 참조하십시오.

## SPARC 클라이언트용 OBP 보안 키

SPARC 클라이언트가 향상된 보안 기능의 혜택을 받으려면 SI 설치용 클라이언트를 부트할 때 OBP 보안 키를 설정해야 합니다.

installadm 명령을 set-server, set-service 또는 set-client 하위 명령과 함께 사용하여 TLS 자격 증명을 생성하거나 지정할 때 해싱(HMAC) 키 및 암호 키가 기존에 없는 경우 자동으로 생성되어 표시됩니다. 동일한 명령을 반복할 경우 이러한 펌웨어 키는 자동으로 재생성되지 않습니다.

-E 및 -H 옵션을 사용하여 OBP 키를 재생성할 수 있습니다. OBP 키가 없는 경우 -E 또는 -H 옵션을 지정하지 마십시오. 기존 암호 키 또는 HMAC는 무효화되고 대체됩니다. 암호 키를 재생성하려면 -E 옵션을 사용합니다. 해싱 키를 재생성하려면 -H 옵션을 사용합니다. -E 및 -H 옵션 모두, -E 옵션만 또는 -H 옵션만 지정할 수 있습니다. 명령을 실행하면 기존 OBP 키가 무효화되고 새로 생성된 값으로 대체됩니다. 이러한 키를 설정하는 OBP 명령이 표시됩니다.

나중에 OBP 보안 키를 설정하는 OBP 명령을 표시하려면 다음 예제와 같이 list 하위 명령에 -v 옵션을 사용합니다.

```
# installadm list -v -e mac-addr
```

이 명령은 TLS 자격 증명이 설치 서비스 이름을 사용하는 클라이언트 MAC 주소를 사용하여 지정되었는지 아니면 기본 클라이언트 자격 증명인지 여부에 관계없이 이 클라이언트에 대한 올바른 OBP 키를 표시합니다. list 하위 명령의 출력은 예 8-41. “클라이언트 보안 정보 나열”에서처럼 OBP 키가 이 특정 클라이언트, 지정한 설치 서비스 또는 기본 클라이언트 중 어떤 항목에 대해 정의되었는지 보여줍니다.

## 보안 사용 및 사용 안함

이 절에서는 보안 구성을 삭제하지 않고 보안 요구 사항을 사용 안함으로 설정한 다음 이전에 구성된 서버 및 클라이언트 인증 설정을 사용하여 보안 요구 사항을 다시 사용으로 설정하는데 사용할 수 있는 옵션에 대해 설명합니다.

보안은 기본적으로 사용으로 설정됩니다. 보안을 사용 안함으로 설정하면 클라이언트에 대해 자격 증명이 발행되지 않고 클라이언트에서 자격 증명을 요구하지도 않습니다. 보안을 사용 안함으로 설정하면 SI 클라이언트에 제공되는 모든 SI 파일에 대해 HTTPS 네트워크 보호가 제공되지 않습니다. 보안을 사용 안함으로 설정한 경우 SI 웹 서버에서 제공하는 사용자 지정 보안 파일(“웹 서버 사용자 파일 디렉토리 구성” [95] 참조)에 액세스할 수 없습니다.

보안을 사용 안함으로 설정한 경우에도 보안을 계속 구성할 수 있습니다. 보안을 다시 사용으로 설정하면 모든 변경 사항이 적용됩니다.

다음 명령을 사용하여 서버 차원의 보안 적용을 사용 안함으로 설정합니다.

```
# installadm set-server -S
Refreshing web server.
Automated Installer security has been disabled.
```

설치 서비스가 이미 구성된 시스템의 보안을 사용 안함으로 설정할 때는 주의하십시오. 보호된 설치 서비스 데이터에 액세스할 때 인증이 필요하지 않으며 인증되지 않은 클라이언트가 SI를 통해 Oracle Solaris를 설치할 수 있습니다.

set-security --disable을 사용하여 보안 적용을 사용 안함으로 설정한 다음 보안 적용을 다시 사용으로 설정하려면 다음 명령을 사용하십시오.

```
# installadm set-security -s
Configuring web server security.
Refreshing web server.
Warning: client 02:00:00:00:00:00 of service solaris11_2-i386
is required to have credentials but has none.
Automated Installer security has been enabled.
```

## 인증서 삭제

installadm 명령을 사용하여 보안 인증서를 삭제합니다. set-server, set-service 및 set-client 하위 명령을 사용하여 보안 인증서를 삭제할 수 있습니다.

보안 인증서는 `delete-client` 또는 `delete-service` 하위 명령을 실행하여 제거할 수도 있습니다. `delete-client` 명령은 모든 클라이언트별 인증서를 제거합니다. `delete-service` 하위 명령은 모든 서비스별 인증서와 해당 서비스 및 모든 별칭 서비스의 모든 클라이언트에 대한 클라이언트별 인증서를 제거합니다.

**주의** - 삭제된 인증서는 복구할 수 없으며 서버 인증서 없이는 TLS 보안 프로토콜이 작동하지 않습니다. 서버 인증서를 삭제하기 전에 SI 보안이 사용 안함으로 설정됩니다.

**예 8-28**            단일 클라이언트용 인증서 삭제

이 예에서는 MAC 주소를 사용하여 클라이언트에 지정한 개인 키와 인증서, CA 인증서 및 OBP 키를 삭제합니다. 클라이언트 펌웨어에서 OBP 키가 설정된 경우 **“해시 키 및 암호 키 삭제” [224]**의 설명에 따라 설정 해제하십시오.

```
# installadm set-client -e mac-addr -x
```

**예 8-29**            CA 인증서 삭제

이 예에서는 CA 인증서를 사용하는 모든 클라이언트용으로 지정된 CA 인증서를 삭제합니다. **예 8-41. “클라이언트 보안 정보 나열”**에서 보는 것처럼 `--hash` 옵션 인수 값은 `list` 하위 명령에 의해 표시된 인증서 X.509 주체의 해시 값입니다. 지정한 CA 인증서를 사용하는 모든 클라이언트의 수가 계산되고 계속할지를 확인하는 프롬프트와 함께 표시됩니다.

```
$ installadm set-client -x --hash b99588cf
Identifier hash: b99588cf
Subject: /C=CZ/O=Oracle Czech s.r.o./OU=install/CN=genca
Issuer: /C=CZ/O=Oracle Czech s.r.o./OU=install/CN=genca
Valid from Apr 27 13:12:27 2012 GMT to Apr 27 13:12:27 2015 GMT
This CA has the following uses:
    WARNING: this is the server CA certificate
Deleting this Certificate Authority certificate can prevent
    credentials from validating.
Do you want to delete this Certificate Authority certificate [y|N]: y
Deleting all references to Certificate Authority with hash value b99588cf
```

**주의** -이 예에서 이 CA 인증서를 사용하는 모든 클라이언트에 대해 이 CA 인증서의 모든 인스턴스가 삭제되고 해당 클라이언트는 더 이상 인증을 받을 수 없습니다. 지정한 CA 인증서를 사용하여 인증서를 생성하면 `installadm` 명령으로 더 이상 인증서를 생성할 수 없습니다.

**예 8-30**            서버 보안 인증서 삭제

이 예에서는 서버의 서버 인증 전용 개인 키 및 인증서, CA 인증서 및 OBP 키를 삭제합니다.

```
# installadm set-server -x
```

## ▼ AI를 사용한 Kerberos 클라이언트 구성 방법

이 절차에서는 클라이언트의 keytab 파일이 이미 만들어져서 AI 서버에 저장되어 있습니다. 이 예에서는 자동 등록 기능을 통해 기존 인증서 또는 새 주체를 사용하여 Kerberos 클라이언트를 구성합니다. 자동 등록 프로세스는 각 클라이언트에 대한 keytab 파일을 만들고 인코딩할 필요가 없으므로 더 간편합니다.

### 1. 관리자가 됩니다.

자세한 내용은 [“How to Use Your Assigned Administrative Rights”](#) in [“Oracle Solaris 11.2 Administration: Security Services”](#)을 참조하십시오.

### 2. 필요한 경우 설치 서비스를 만듭니다.

```
# installadm create-service -n krb-sparc \
  -d /export/auto_install/krb-sparc \
  -s /export/auto_install/iso/sol-11_2-ai-sparc.iso
```

```
Creating service from:
/export/auto_install/iso/sol-11_2-ai-sparc.iso
Setting up the image ...
Creating sparcs service: krb-sparc
Image path: /export/auto_install/krb-sparc
Refreshing install services
```

### 3. 클라이언트를 서비스에 연결합니다.

Kerberos를 실행하여 설치해야 할 모든 클라이언트에 대해 이 단계를 반복합니다. 이 예에서는 11:11:11:11:11:11 주소를 사용하는 클라이언트가 krb-sparc 설치 서비스와 연결됩니다.

```
# installadm create-client -n krb-sparc -e 11:11:11:11:11:11
Adding host entry for 11:11:11:11:11:11 to local DHCP configuration.
```

### 4. 클라이언트용 인증서를 만듭니다.

```
# installadm set-client -c 11:11:11:11:11:11 -g
Generating credentials for client 11:11:11:11:11:11...
A new certificate key has been generated.
A new certificate has been generated.
```

### 5. Kerberos 구성 파일의 내용을 정의하는 시스템 구성 프로파일을 만듭니다.

이 예에서는 kclient 명령을 대화식으로 실행하여 프로파일을 만듭니다. 또는 명령줄 옵션 또는 입력 프로파일을 사용하여 명령을 호출할 수 있습니다. 자세한 내용은 [kclient\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

이 예에서는 KDC가 MIT 서버에서 실행 중입니다. Solaris KDC의 샘플 출력을 보려면 [예 8-31. “Kerberos 클라이언트 배포 중 기존 키 다운로드”](#)을 참조하십시오. AD 클라이언트

의 샘플 출력을 보려면 예 8-33. “AI 클라이언트를 MS AD 도메인에 자동 연결”을 참조하십시오.

```
# kclient -x /root/krb-sc.xml
Starting client setup
-----
Is this a client of a non-Solaris KDC ? [y/n]: y
Which type of KDC is the server:
  ms_ad: Microsoft Active Directory
  mit: MIT KDC server
  heimdal: Heimdal KDC server
  shishi: Shishi KDC server
Enter required KDC type: mit
Do you want to use DNS for kerberos lookups ? [y/n]: n
  No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the master KDCs for the above realm using a comma-separated list: kdc.example.com
Do you have any slave KDC(s) ? [y/n]: y
Enter a comma-separated list of slave KDC host names: kdc2.example.com
Do you have multiple domains/hosts to map to a realm ? [y/n]: n
  No action performed.
Setting up /root/krb-sc.xml.
```

6. (옵션) 클라이언트의 이진 keytab 파일을 XML 프로파일로 변환합니다.

자동 등록을 통해 키를 가져올 수 있거나 클라이언트에 키가 없는 경우 이 단계가 필요하지 않습니다. 클라이언트에서는 keytab 파일을 만들어야 하는데 이 작업은 주로 클라이언트를 처음 구성할 때 KDC 관리자가 수행합니다.

```
# kclient-kt2prof -k ./host1.keytab -p /root/host1.xml
```

7. 클라이언트 프로파일을 만들어 클라이언트의 나머지를 구성합니다.

이 절차에서는 프로파일을 사용해야 하므로 시스템 구성 프로파일을 사용하여 클라이언트 수 만큼 프로파일을 구성합니다.

8. (옵션) 프로파일에 대한 보안 정책을 설정합니다.

클라이언트 프로파일에 keytab이 포함된 경우 인증된 클라이언트만 keytab 파일을 다운로드할 수 있도록 서비스에 require-client-auth 보안 정책을 지정해야 합니다.

```
# installadm set-service -p require-client-auth -n krb-sparc
```

9. 클라이언트 프로파일을 클라이언트 서비스와 연결합니다.

Kerberos 구성 파일, 클라이언트 keytab 파일 및 설치 서비스에 생성한 기타 프로파일을 연결합니다.

```
# installadm create-profile -n krb-sparc -f /root/krb-sc.xml
Profile krb-sc.xml added to database.
# installadm create-profile -n krb-sparc -f /root/host1.xml -c mac="11:11:11:11:11:11"
Profile host1.xml added to database.
```

## 10. 클라이언트를 부트하여 AI 프로세스를 시작합니다.

### 예 8-31 Kerberos 클라이언트 배포 중 기존 키 다운로드

자동 등록은 KDC가 Solaris KDC 또는 MS AD인 경우에만 사용할 수 있습니다. KDC가 MIT, Heimdal 또는 Shishi인 경우 미리 생성된 keytab 전송만 가능합니다.

자동 등록을 사용하여 기존 키를 다운로드하려면 먼저 c 및 i 관리 권한을 사용하여 KDC에 관리 주체를 만들어야 합니다. 이 예에서 주체의 이름은 download/admin입니다. 자세한 지침은 “Oracle Solaris 11.2의 Kerberos 및 기타 인증 서비스 관리”의 “새 Kerberos 주체 만들기” 및 “Oracle Solaris 11.2의 Kerberos 및 기타 인증 서비스 관리”의 “주체의 Kerberos 관리 권한 수정”을 참조하십시오.

이 예에서 KDC는 Oracle Solaris를 실행 중입니다. 또한 클라이언트용 키가 이미 생성되어 있습니다.

이 예에서는 Kerberos 구성 파일에 대한 시스템 구성 프로파일을 만들 때 download/admin 주체를 추가하는 방법을 보여줍니다. download/admin 주체는 Kerberos 클라이언트를 배포할 때 KDC 서버에서 기존 키를 전송하는 데 사용하는 특수 관리 주체입니다.

```
# kclient -x /root/krb-sc.xml
Starting client setup
-----
Is this a client of a non-Solaris KDC ? [y/n]: n
    No action performed.
Do you want to use DNS for kerberos lookups ? [y/n]: n
    No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the master KDCs for the above realm using a comma-separated
list: kdc.example.com
Do you have any slave KDC(s) ? [y/n]: y
Enter a comma-separated list of slave KDC host names: kdc2.example.com
Do you have multiple domains/hosts to map to realm ? EXAMPLE.COM [y/n]: n
    No action performed.
Should the client automatically join the realm ? [y/n]: y
Enter the krb5 administrative principal to be used: download/admin
Password for download/admin: xxxxxxxx
Do you plan on doing Kerberized nfs ? [y/n]: n
    No action performed.
Is this client a member of a cluster that uses a logical host name ? [y/n]: n
    No action performed.
Do you have multiple DNS domains spanning the Kerberos realm EXAMPLE.COM ? [y/n]: n
    No action performed.
Setting up /root/krb-sc.xml.
```

### 예 8-32 Kerberos 클라이언트 배포 중 새 키 만들기

자동 등록은 KDC가 Solaris KDC 또는 MS AD인 경우에만 사용할 수 있습니다. KDC가 MIT, Heimdal 또는 Shishi인 경우 미리 생성된 keytab 전송만 가능합니다.

자동 등록을 사용하여 새 키를 다운로드하려면 먼저 a, c 및 i 관리 권한을 사용하여 KDC에 관리 주체를 만들어야 합니다. 이 예에서 주체의 이름은 create/admin입니다. 자세한 지침은 “Oracle Solaris 11.2의 Kerberos 및 기타 인증 서비스 관리”의 “새 Kerberos 주체 만들기” 및 “Oracle Solaris 11.2의 Kerberos 및 기타 인증 서비스 관리”의 “주체의 Kerberos 관리 권한 수정”을 참조하십시오.

이 예에서 KDC는 Oracle Solaris를 실행 중입니다. 이 예에서는 Kerberos 구성 파일에 대한 시스템 구성 프로파일을 만들 때 create/admin 주체를 추가합니다. create/admin 주체는 Kerberos 클라이언트를 배포할 때 KDC 서버에서 새 키를 전송하는 데 사용하는 특수 관리 주체입니다. 이 명령에는 더 많은 옵션이 포함되어 있으므로 더 적은 질문을 묻습니다.

```
# kclient -x /root/krb-sc.xml -R EXAMPLE.COM -a create/admin -d none -m kdc.example.com
Starting client setup
-----
Do you have multiple domains/hosts to map to realm ? EXAMPLE.COM [y/n]: n
    No action performed.
Should the client automatically join the realm ? [y/n]: y
Password for create/admin: xxxxxxxx
Setting up /root/krb-sc.xml.
```

#### 예 8-33 AI 클라이언트를 MS AD 도메인에 자동 연결

이 예에서 클라이언트는 AD 도메인을 연결합니다. 다음 명령을 사용하여 Kerberos 구성 파일에 대한 시스템 구성 프로파일을 만들 때 Administrator 주체를 추가합니다.

```
# kclient -x /root/krb-sc.xml
Starting client setup
-----
Is this a client of a non-Solaris KDC ? [y/n]: y
Which type of KDC is the server:
    ms_ad: Microsoft Active Directory
    mit: MIT KDC server
    heimdal: Heimdal KDC server
    shishi: Shishi KDC server
Enter required KDC type: ms_ad
Should the client automatically join AD domain ? [y/n]: y
Enter the Kerberos realm: EXAMPLE.COM
Enter the krb5 administrative principal to be used: Administrator
Password for Administrator: xxxxxxxx
Setting up /root/krb-sc.xml.
```

## 설치 서비스에 대한 정보 표시

installadm list 명령을 사용하여 설치 서비스뿐 아니라 서비스와 연관된 클라이언트, AI 매니페스트 및 시스템 구성 프로파일에 대한 정보를 보여줍니다. 이 절은 다음과 같이 구성되어 있습니다.

예 8-34. “AI 서버의 모든 설치 서비스 나열”에서는 AI 서버에 대한 설치 서비스를 모두 나열하는 방법을 보여줍니다.

예 8-35. “지정된 설치 서비스에 대한 정보 표시”에서는 특정 설치 서비스에 대한 정보를 나열하는 방법을 보여줍니다.

예 8-36. “설치 서비스와 연관된 클라이언트 나열”에서는 설치 서비스와 연관된 클라이언트를 나열하는 방법을 보여줍니다.

예 8-37. “특정 설치 서비스와 연관된 클라이언트 나열”에서는 특정 설치 서비스와 연관된 클라이언트를 나열하는 방법을 보여줍니다.

예 8-38. “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”에서는 모든 AI 매니페스트 및 시스템 구성 프로파일을 나열하는 방법을 보여줍니다.

예 8-39. “지정된 설치 서비스와 연관된 매니페스트 및 프로파일 나열”에서는 특정 설치 서비스와 연관된 AI 매니페스트 및 시스템 구성 프로파일을 나열하는 방법을 보여줍니다.

예 8-41. “클라이언트 보안 정보 나열”에서는 클라이언트 보안 정보를 나열하는 방법을 보여줍니다.

예 8-40. “서버 보안 정보 나열”에서는 AI 서버 보안 정보를 나열하는 방법을 보여줍니다.

#### 예 8-34 AI 서버의 모든 설치 서비스 나열

이 예에서는 이 서버의 설치 서비스를 모두 표시합니다. 이 예에서 네 개의 사용으로 설정된 설치 서비스가 있습니다. 사용 안함으로 설정된 서비스는 상태 값이 off입니다.

```
$ /usr/sbin/installadm list
```

Service Name	Status	Arch	Type	Alias	Aliases	Clients	Profiles	Manifests
default-i386	on	i386	iso	yes	0	0	0	1
default-sparc	on	sparc	iso	yes	0	0	0	1
solaris11_2-i386	on	i386	iso	no	1	0	1	1
solaris11_2-sparc	on	sparc	iso	no	1	0	2	1

default-i386 서비스는 이 서버에 처음 i386 서비스를 만들 때 자동으로 생성되었습니다. default-i386 서비스는 create-client 하위 명령을 사용하여 solaris11\_2-i386 서비스와 연관되지 않은 x86 클라이언트에 의해 사용됩니다. default-i386 및 solaris11\_2-i386 서비스는 설치 이미지를 공유하지만 AI 매니페스트와 시스템 구성 프로파일은 서로 다릅니다.

default-sparc 서비스는 이 서버에 처음 sparc 서비스를 만들 때 자동으로 생성되었습니다. default-sparc 서비스는 create-client 하위 명령을 통해 solaris11\_2-sparc 서비스와 연관되지 않은 모든 SPARC 클라이언트에서 사용됩니다. default-sparc 및 solaris11\_2-sparc 서비스는 설치 이미지를 공유하지만 AI 매니페스트와 시스템 구성 프로파일은 서로 다릅니다.

#### 예 8-35 지정된 설치 서비스에 대한 정보 표시

이 예에서는 -n 옵션으로 지정된 설치 서비스에 대한 정보를 표시합니다.

```
$ /usr/sbin/installadm list -n solaris11_2-sparc
```

Service Name	Status	Arch	Type	Alias	Aliases	Clients	Profiles	Manifests
solaris11_2-sparc	on	sparc	iso	no	1	0	2	1

**예 8-36** 설치 서비스와 연관된 클라이언트 나열

이 예에서는 이 AI 서버의 설치 서비스와 연관된 클라이언트를 모두 나열합니다. 클라이언트는 `installadm create-client` 명령을 사용하여 설치 서비스와 연관되었습니다. [“클라이언트를 서비스와 연관” \[103\]](#)을 참조하십시오.

```
$ /usr/sbin/installadm list -c
```

Service Name	Client Address	Arch	Secure	Custom	Args	Custom	Grub
solaris11_2-sparc	00:14:4F:A7:65:70	sparc	no	no		no	
solaris11_2-i386	08:00:27:8B:BD:71	i386	no	no		no	
	01:C2:52:E6:4B:E0	i386	no	no		no	

**예 8-37** 특정 설치 서비스와 연관된 클라이언트 나열

이 예에서는 지정된 설치 서비스에 추가된 클라이언트를 모두 나열합니다. 다음 예제에서 하나의 클라이언트가 `solaris11_2-sparc` 설치 서비스와 연관됩니다.

```
$ /usr/sbin/installadm list -c -n solaris11_2-sparc
```

Service Name	Client Address	Arch	Secure	Custom	Args	Custom	Grub
solaris11_2-sparc	00:14:4f:a7:65:70	sparc	no	no		no	

**예 8-38** 모든 AI 매니페스트 및 시스템 구성 프로파일 나열

이 예에서는 이 AI 서버의 모든 설치 서비스에 대한 모든 AI 매니페스트, 파생된 매니페스트 스크립트, 시스템 구성 프로파일을 나열합니다. Service, Manifest Name 및 Profile Name 열에는 매니페스트, 스크립트 또는 프로파일의 내부 이름이 표시됩니다. Status 열은 각 서비스에 대한 기본 매니페스트와 비활성 매니페스트를 식별합니다. 매니페스트에 연관된 조건이 없고 기본값도 아닌 경우 매니페스트는 비활성입니다. Criteria 열은 연관된 클라이언트 조건을 보여줍니다.

`orig_default` 매니페스트는 설치 서비스를 만들 때 설치 서비스의 일부였던 원래의 기본 AI 매니페스트입니다. `mem1` 매니페스트는 메모리 조건으로 만들어졌고 이 서비스에 대한 새 기본 매니페스트로 지정됩니다. `mem1`이 기본 매니페스트이므로 해당 조건은 무시됩니다. 다른 매니페스트가 기본 매니페스트로 만들어진 경우 클라이언트가 `mem1` 매니페스트를 사용하도록 선택하기 위해 `mem1` 조건이 사용됩니다. 원래의 기본 매니페스트는 사용할 클라이언트를 결정하기 위해 연관된 조건이 없으므로 비활성입니다. 기본 매니페스트만 연관된 조건이 없습니다. 서비스와 연관된 다른 매니페스트를 사용하기 위한 기준과 일치하지 않는 클라이언트는 기본 매니페스트를 사용하는데 이 경우에는 `mem1`입니다. AI 매니페스트 선택에 대한 자세한 내용은 [9장. 설치 사용자 정의](#)를 참조하십시오.

```
$ installadm list -m -p
```

```

Service Name          Manifest Name      Type   Status  Criteria
-----
default-i386         orig_default      derived default one
default-sparc        orig_default      derived default none
solaris11_2-i386     ipv4              xml    active  ipv4 = 10.6.68.1 -
10.6.68.200
MB - 4095 MB)
solaris11_2-sparc    mem1              derived default (Ignored: mem = 2048
unbounded            sparc-ent         xml    active  mem = 4096 MB -
platform = SUNWSPARC-
Enterprise
MB - 4095 MB)
mem1                  derived default (Ignored: mem = 2048
orig_default         derived inactive none

Service Name          Profile Name       Criteria
-----
solaris11_2-i386     mac2              mac = 08:00:27:8B:BD:71
hostname = server2
mac3                 mac = 01:C2:52:E6:4B:E0
hostname = server3
ipv4                  ipv4 = 10.0.2.100 - 10.0.2.199
mem1                  mem = 2048 MB - 4095 MB
solaris11_2-sparc    mac1              mac = 01:C2:52:E6:4B:E0
hostname = server1
ipv4                  ipv4 = 192.168.168.251
sparc-ent            platform = SUNWSPARC-Enterprise
mem = 4096-unbounded

```

이 명령을 올바른 프로파일과 함께 실행한 경우 매니페스트 목록의 추가 열에서 매니페스트 유형이 xml인지 아니면 derived인지 식별합니다.

#### 예 8-39 지정된 설치 서비스와 연관된 매니페스트 및 프로파일 나열

이 예에서는 설치 서비스 solaris11\_2-sparc와 연관된 모든 SI 매니페스트, 파생된 매니페스트 스크립트 및 시스템 구성 프로파일을 보여줍니다.

```

$ installadm list -m -p -n solaris11_2-sparc
Service Name          Manifest Name Type   Status  Criteria
-----
solaris11_2-sparc    sparc-ent     xml    active  mem = 4096 MB - unbounded
platform = SUNWSPARC-
Enterprise
mem1                  derived default Ignored:
mem = 2048 MB - 4095 MB)
orig_default         derived inactive none

Service Name          Profile Name       Criteria
-----
solaris11_2-sparc    mac1              mac = 01:C2:52:E6:4B:E0
hostname = server1

```

```

sparc-ent          ipv4 = 192.168.168.251
                   platform = SUNWSPARC-Enterprise
                   mem = 4096-unbounded
    
```

**예 8-40**          서버 보안 정보 나열

list 하위 명령과 -v 및 -s 옵션을 함께 실행하면 다음과 같은 서버 정보가 표시됩니다.

- 현재 보안 상태: 사용 또는 사용 안함
- 서버 인증서 X.509 주체 및 발행자 문자열
- 서버 인증서의 유효 일자
- 서버 인증서 검증 결과
- 서버 CA 인증서 해시 값, X.509 주체 및 발행자
- 클라이언트 인증을 위한 클라이언트 CA 인증서
- 기본 클라이언트 인증서

```

# installadm list -v -s
AI Server Parameter Value
-----
Hostname ..... install-svr
Architecture ..... i386
Active Networks .... 10.134.125.170
Http Port ..... 5555
Secure Port ..... 5556
Image Path Base Dir .... /export/auto_install
Multi-Homed? ..... no
Managing DHCP? ..... yes
DHCP IP Range ..... 192.168.100.240 - 192.168.100.249
Boot Server ..... 192.168.100.45
Web UI Enabled? ..... yes
Wizard Saves to Server? no
Security Enabled? ..... yes
Security Key? ..... yes
Security Cert:
    Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=osol-inst
    Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
    Source : Server Certificate
    Valid from: Jan 24 22:53:00 2014 GMT
               to: Jan 24 22:53:00 2024 GMT
    Validates?: yes
CA Certificates:
    d09051e4 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
            Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
            Source : Server CA Certificate
            Valid from: Jan 24 22:53:00 2014 GMT
                       to: Jan 24 22:53:00 2024 GMT
    f9d73b41 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
            Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
            Source : Server CA Certificate
            Valid from: Jan 24 22:53:00 2014 GMT
                       to: Jan 24 22:53:00 2024 GMT
    
```

```

Def Client Sec Key? .... yes
Def Client Sec Cert:
    Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Client default
    Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
    Source : Default Client Certificate
    Valid from: Jul 15 19:33:00 2013 GMT
    to: Jul 13 19:33:00 2023 GMT
Def Client CA Certs .... none
Def Client FW Encr Key . adcc858c58ecae04c02282e7245c235c
Def Client FW HMAC Key . cb7bc6213512c8fa3dc7d7283a9e056dc2791f98
Number of Services ..... 102
Number of Clients ..... 37
Number of Manifests .... 108
Number of Profiles ..... 92

```

#### 예 8-41 클라이언트 보안 정보 나열

list 하위 명령과 -v 및 -e 옵션을 함께 실행하면 다음과 같은 클라이언트 보안 정보가 표시됩니다.

- 클라이언트에 사용되는 자격 증명
- 클라이언트 자격 증명의 소스
- 클라이언트 인증서의 유효성

```

# installadm list -v -e 00:14:4F:83:3F:4A
Service Name      Client Address    Arch  Secure Custom Args Custom Grub
-----
solaris11_2-sparc 00:14:4F:A7:65:70 sparc yes    no        no

Client Credentials? yes
Security Key? ..... yes
Security Cert:
    Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=CID 01020000000000
    Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
    Valid from: Jan 24 10:20:00 2014 GMT
    to: Jan 24 10:20:00 2024 GMT

CA Certificates:
    d09051e4 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
    Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
    Source : Default CA Certificate
    Valid from: Jan 24 22:53:00 2014 GMT
    to: Jan 24 22:53:00 2024 GMT

FW Encr Key (AES) . 23780bc444636f124ba3ff61bdac32d1
FW HMAC Key (SHA1) 1093562559ec45a5bb5235b27c1d0545ff259d63
Boot Args ..... none

```

export 하위 명령은 클라이언트에 지정된 TLS 자격 증명을 보여줍니다. -c를 추가하면 x.509 TLS 인증서가 표시됩니다.

```

# installadm export -e 00:14:4F:83:3F:4A -C
----- certificate: client_00:14:4F:83:3F:4A_cert_de22916b -----

```

```
-----BEGIN CERTIFICATE-----
MIICFDCCAX+gAwIBAgIBGTALBgkqhkiG9w0BAQswUDELMAkGA1UEBhMCVVMxDzAN
....
UiZDA6G0dvE=
-----END CERTIFICATE-----
```

-K 옵션은 X.509 개인 키를 보여줍니다.

```
# installadm export -e 00:14:4F:83:3F:4A -K
----- key: client_00:14:4F:83:3F:4A_key -----
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDCCJbC5Bd0uMQ0A0k4LLlQqWiQwqkx9lpIhHl31tF1/WxHi74A
....
SYoBeKAOPSo7Evund+bHAR0L0H4QnbSjgl1UDuZr3T3h
-----END RSA PRIVATE KEY-----
```

## 설치 서비스 관리

installadm set-service 명령을 사용하여 기존 설치 서비스를 재구성합니다. 이 절의 예에서는 설치 서비스 별칭, 기본 SI 매니페스트 또는 설치 서비스용 이미지 경로를 설정하는 방법과 설치 서비스 업데이트 방법을 보여줍니다.

### 설치 서비스 별칭 설정

설치 서비스 별칭을 사용하여 새 서비스를 만들 때 수행해야 하는 재구성의 양을 최소화할 수 있습니다. 예를 들어 default-arch 설치 서비스는 별칭입니다. 서비스를 생성할 때 create-service 하위 명령을 -t 옵션과 함께 사용하여 별칭을 만들 수 있습니다. set-service 하위 명령을 -t 옵션과 함께 사용하면 지정한 서비스가 다른 서비스의 별칭으로 변경됩니다. set-service 하위 명령을 사용할 때는 지정한 서비스가 이미 별칭이어야 합니다.

서비스 또는 별칭에 추가된 매니페스트, 프로파일 및 클라이언트 조건은 별칭 재설정 후에도 동일하게 유지됩니다. 지정한 서비스가 사용하는 넷 이미지만 변경됩니다.

별칭을 설정하기 전에 서비스에 추가된 매니페스트와 프로파일은 별칭을 재설정할 때 다시 검증됩니다. 새로운 설치 이미지와 연관된 AI DTD 및 SMF DTD가 서로 다를 수 있기 때문입니다. 이 검증은 create-manifest 및 create-profile 명령으로 수행된 것과 동일한 검증입니다.

**예 8-42**            설치 서비스 별칭 만들기

이 예에서는 새 install-sparc 서비스를 기존 solaris11\_2-sparc 설치 서비스에 대한 별칭으로 만듭니다.

```
# installadm create-service -t solaris11_2-sparc -n install-sparc
```

**예 8-43**          설치 서비스 별칭 수정

이 예에서는 이전에 solaris11\_2-i386 설치 서비스 및 install-i386 설치 서비스 별칭을 생성했어야 합니다. 다음 예제에서는 install-i386 설치 서비스를 solaris11\_2-i386 설치 서비스에 대한 별칭으로 설정합니다.

```
# installadm set-service -t solaris11_2-i386 -n install-i386
```

## 설치 서비스용 기본 AI 매니페스트 설정

이 예에서는 특정 매니페스트 또는 파생된 매니페스트 스크립트를 새 설치 서비스 및 기존 설치 서비스 모두에 대한 기본 매니페스트로 지정하는 방법을 보여줍니다.

**예 8-44**          설치 서비스 생성 시 기본 AI 매니페스트 설정

이 예에서는 mem1 매니페스트를 새 install-sparc 서비스에 대한 기본 매니페스트로 설정합니다. 다른 클라이언트 조건과 일치하지 않고 이 서비스와 연관된 모든 클라이언트는 기본적으로 이 매니페스트를 사용합니다.

```
# installadm create-service -M /tmp/mem1 -n install-sparc
```

**예 8-45**          기존 설치 서비스를 수정하여 기본 AI 매니페스트 설정

이 예에서는 기존에 등록된 mem1 매니페스트를 기존 install-i386 서비스에 대한 기본 매니페스트로 설정합니다. 다른 클라이언트 조건과 일치하지 않고 이 서비스와 연관된 모든 클라이언트는 기본적으로 이 매니페스트를 사용합니다.

```
# installadm set-service -M mem1 -n install-i386
```

## 설치 서비스용 이미지 경로 설정

이 예에서는 지정한 서비스의 설치 이미지 경로를 정의 또는 재설정하는 방법을 보여줍니다.

**예 8-46**          새 설치 서비스용 이미지 경로 설정

이 예에서는 서비스 및 넷 이미지를 만드는 동안 solaris11\_2-i386 서비스에 대한 설치 이미지 경로를 정의합니다.

```
# installadm set-service -d /export/ai-images/solaris11_2.i386 -n solaris11_2-i386
```

**예 8-47**          기존 설치 서비스용 이미지 경로 설정

이 예에서는 solaris11\_2-i386 서비스에 대한 설치 이미지 경로를 재배치합니다.

```
# installadm set-service -d /export/ai-images/solaris11_2.i386 -n solaris11_2-i386
```

## 기존 설치 서비스 업데이트

update-service 하위 명령을 사용하여 IPS AI 넷 이미지 패키지를 사용하여 생성된 서비스의 별칭과 연관된 이미지를 업데이트합니다. 업데이트된 이미지로 새 서비스가 생성되고 새 서비스를 사용하도록 별칭이 변경됩니다.

서비스를 업데이트할 때 다른 저장소를 사용하려면 update-service 하위 명령에 -p 옵션을 추가합니다. -p 옵션이 지정되지 않은 경우 별칭이 *svc-name*인 서비스의 이미지를 만드는 데 사용된 게시자가 사용됩니다.

-s 옵션이 지정되지 않은 경우 게시자로부터 사용 가능한 최신 버전의 install-image/solaris-auto-install 패키지가 사용됩니다.

**예 8-48**            설치 서비스 업데이트

이 예에서는 새 서비스를 만들고 이 새 서비스를 사용하도록 default-i386 별칭을 변경합니다.

```
# installadm update-service -n default-i386
```

**예 8-49**            설치 서비스 업데이트 시 다른 저장소 사용

이 예에서는 solaris11\_2-i386 서비스와 연관된 게시자를 식별하는 방법을 보여줍니다. 먼저 installadm list 하위 명령을 사용하여 서비스에 대한 이미지 경로를 지정합니다. 그런 다음 이미지 경로를 사용하여 사용 중인 게시자를 지정할 수 있습니다.

```
$ installadm list -v -n solaris11_2-i386
Service Name      Status Arch    Type Alias Aliases Clients Profiles Manifests
-----
solaris11_2-i386 on     i386   iso   no     1       0       1       1

Image Path ..... /export/auto_install/solaris11_2-i386
....

$ pkg -R /export/auto_install/solaris11_2-i386 publisher
PUBLISHER        TYPE    STATUS    URI
solaris          origin online    http://pkg.oracle.com/solaris/release/
```

이 예에서는 설치 서비스를 업데이트할 때 example.com/solaris/mybuild의 게시자를 사용하도록 지정합니다.

```
# installadm update-service -n default-i386 -p solaris=http://example.com/solaris/mybuild
```

예 8-50 설치 서비스 업데이트 시 다른 넷 이미지 패키지 사용

이 예에서는 특정 넷 이미지 패키지를 지정합니다.

```
# installadm update-service -n default-i386 -s FMRI
```

## AI 매니페스트 관리

이 절에서는 AI 매니페스트를 업데이트, 삭제, 검증 또는 내보내는 방법을 보여줍니다.

### AI 매니페스트 업데이트

`installadm update-manifest` 명령을 사용하여 지정한 AI 매니페스트 또는 파생된 매니페스트 스크립트 파일을 지정한 설치 서비스의 매니페스트 또는 스크립트 파일로 대체합니다. 조건, 기본 상태 및 매니페스트 이름은 업데이트 결과로 변경되지 않습니다.

`update-manifest` 하위 명령은 XML 매니페스트 파일을 설치 서비스에 추가하기 전에 검증합니다.

매니페스트는 지정한 서비스에 이미 있어야 합니다. [예 8-38. “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”](#)에 표시된 것처럼 `installadm list` 명령을 사용하여 확인합니다.

매니페스트가 지정되지 않은 경우 다음 방법 중 하나로 교체할 매니페스트를 식별할 수 있습니다.

- 이 속성 값이 지정되고 이 속성 값이 이 설치 서비스에 대한 기존 매니페스트의 이름과 일치하는 경우 지정한 매니페스트에서 `ai_instance` 요소의 `name` 속성
- 이 이름이 이 설치 서비스에 대한 기존 매니페스트의 이름과 일치하는 경우 지정한 파일 이름의 기본 이름

이 예에서는 `solaris11_2-sparc` 서비스에 있는 `sparc-ent` 매니페스트의 내용을 `./mymanifests/manifest-new-sparc-ent.xml`의 내용으로 업데이트합니다. `installadm list`의 매니페스트 이름은 여전히 `sparc-ent`입니다.

```
# installadm update-manifest -n solaris11_2-sparc \
-f ./mymanifests/manifest-new-sparc-ent.xml -m sparc-ent
```

### AI 매니페스트 검증

`installadm validate` 명령을 사용하여 AI 매니페스트의 구문 정확성을 검증합니다.

-m 옵션을 사용하여 설치 서비스에 추가되지 않은 매니페스트를 검증합니다. -m 인수의 값은 매니페스트의 경로 이름입니다.

-m 옵션을 사용하여 지정한 설치 서비스에 이미 추가된 매니페스트를 검증합니다. 예 8-38. “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”에 표시된 것처럼 `installadm list` 명령을 사용하여 매니페스트 이름에 대해 가능한 값을 표시합니다. `create-manifest` 하위 명령은 AI 매니페스트를 설치 서비스에 추가하기 전에 검증합니다. `validate -m` 하위 명령은 매니페스트가 추가된 이후 손상되지 않았는지 확인합니다.

설치 서비스에 추가된 매니페스트와 아직 추가되지 않은 매니페스트의 서비스 이름을 지정해야 합니다. DTD가 OS 버전마다 다를 수 있기 때문에 아직 설치 서비스에 추가되지 않은 매니페스트에는 서비스 이름이 필요합니다. AI 서버를 실행 중인 버전이 아닌 다른 버전의 OS를 설치하도록 설치 서비스를 정의할 수도 있습니다. 설치 중인 클라이언트에 사용될 DTD에 대해 매니페스트를 검증해야 합니다. 자세한 내용은 [ai\\_manifest\(4\)](#) 매뉴얼 페이지를 참조하십시오.

검증된 매니페스트는 `stdout`에 출력됩니다. 오류는 `stderr`에 나열됩니다.

## AI 매니페스트 삭제

`installadm delete-manifest` 명령을 사용하여 지정된 AI 매니페스트나 파생된 매니페스트 스크립트를 지정된 설치 서비스에서 제거합니다. 예 8-38. “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”에 표시된 것처럼 매니페스트 이름은 `installadm list` 명령에서 반환한 것과 같은 이름입니다.

기본 AI 매니페스트는 삭제할 수 없습니다.

다음 명령은 `solaris11_2-sparc` 설치 서비스에서 `sparc-ent` AI 매니페스트를 제거합니다.

```
# installadm delete-manifest -m sparc-ent -n solaris11_2-sparc
```

## 시스템 구성 프로파일 관리

이 절에서는 시스템 구성 프로파일을 업데이트, 삭제, 검증 및 내보내는 방법을 제공합니다.

### 시스템 구성 프로파일 업데이트

`installadm update-profile` 명령을 사용하여 지정된 설치 서비스에서 지정된 프로파일을 명령된 파일의 내용으로 대체합니다. 업데이트 이후의 프로파일에 어떤 조건은 그대로 남습니다.

서비스가 포함된 설치 서비스 및 프로파일의 이름을 지정할 수 있습니다. 설치 서비스에 프로파일이 지정되지 않은 경우 업데이트할 프로파일 이름이 파일의 기본 이름이 됩니다.

다음 명령은 solaris11\_2-sparc 서비스에서 sparc-ent 프로파일의 내용을 ./myprofiles/profile-new-sparc-ent.xml의 내용으로 업데이트합니다.

```
# installadm update-profile -n solaris11_2-sparc \
-f ./myprofiles/profile-new-sparc-ent.xml -p sparc-ent
```

## 시스템 구성 프로파일 검증

installadm validate 명령을 사용하여 구문 정확성을 위해 시스템 구성 프로파일을 검증합니다.

-p 옵션을 사용하여 설치 서비스에 추가되지 않은 프로파일을 검증합니다. -p 인수의 값은 프로파일의 경로 이름입니다.

-p 옵션을 사용하여 지정된 설치 서비스에 추가되지 않은 프로파일을 검증합니다. [예 8-38. “모든 SI 매니페스트 및 시스템 구성 프로파일 나열”](#)에 표시된 것처럼 installadm list 명령을 사용하여 프로파일 이름에 대해 가능한 값을 표시합니다. create-profile 하위 명령은 시스템 구성 프로파일을 설치 서비스에 추가하기 전에 검증합니다. validate -p 하위 명령은 프로파일이 추가된 이후 손상되지 않았는지 확인합니다.

설치 서비스에 추가된 프로파일과 아직 추가되지 않은 프로파일의 서비스 이름을 지정해야 합니다. DTD가 OS 버전마다 다를 수 있기 때문에 아직 설치 서비스에 추가되지 않은 프로파일에는 서비스 이름이 필요합니다. SI 서버를 실행 중인 버전이 아닌 다른 버전의 OS를 설치하도록 설치 서비스를 정의할 수도 있습니다. 설치 중인 클라이언트에 사용될 DTD에 대해 프로파일을 검증해야 합니다. 자세한 내용은 [service\\_bundle\(4\)](#) 매뉴얼 페이지를 참조하십시오.

검증된 프로파일은 stdout에 출력됩니다. 오류는 stderr에 나열됩니다.

## 시스템 구성 프로파일 삭제

installadm delete-profile 명령을 사용하여 *profile* 시스템 구성 프로파일을 *svcname* 설치 서비스에서 제거합니다. *profile* 인수 값은 installadm list 명령이 반환하는 프로파일 이름입니다. [예 8-38. “모든 SI 매니페스트 및 시스템 구성 프로파일 나열”](#)을 참조하십시오.

```
installadm delete-profile -p profile... -n svcname
```

다음 명령은 solaris11\_2-sparc 설치 서비스에서 sparc-ent 시스템 구성 프로파일을 제거합니다.

```
# installadm delete-profile -p sparc-ent -n solaris11_2-sparc
```

## AI 매니페스트 또는 시스템 구성 프로파일 내보내기

`installadm export` 명령을 사용하여 지정된 설치 서비스에서 지정된 AI 매니페스트나 시스템 구성 프로파일의 내용을 명명된 파일이나 디렉토리로 복사합니다.

`-o` 옵션이 지정되지 않은 경우 매니페스트와 프로파일 내용이 `stdout`로 이동합니다. 하나의 입력 파일만 지정된 경우 `pathname` 인수 값은 파일 이름일 수 있습니다. 여러 개의 입력 파일이 지정된 경우 `pathname`은 디렉토리여야 합니다.

지정한 매니페스트는 XML AI 매니페스트 또는 파생된 매니페스트 스크립트의 이름일 수 있습니다. 매니페스트 및 파생된 매니페스트 스크립트 생성에 대한 내용은 [10장. 클라이언트 시스템 프로비전](#)을 참조하십시오.

다음 작업을 위해 `installadm export` 명령을 사용합니다.

- 매니페스트 및 프로파일의 사양을 확인합니다.
- 기존 매니페스트나 프로파일을 수정합니다.
- 기존 매니페스트나 프로파일을 새 매니페스트나 프로파일을 생성하기 위한 기초로 사용합니다.

## 설치 사용자 정의

---

설치를 사용자 정의하려면 먼저 SI 매니페스트의 설치 지침 및 시스템 구성 프로파일의 시스템 구성 지침을 사용자 정의하십시오. 그런 다음, 클라이언트 조건을 지정하여 사용자 정의된 설치 및 구성 지침을 해당 조건으로 식별된 클라이언트와 일치시킵니다.

SI 설치 서비스에는 설치 지침이 있는 1개 이상의 SI 매니페스트와 구성 지침이 있는 0개 이상의 시스템 구성 프로파일이 포함됩니다. 각 클라이언트는 단 하나의 SI 매니페스트만 사용합니다. 각 클라이언트는 원하는 수의 시스템 구성 프로파일을 사용할 수 있습니다. 클라이언트 시스템에서 프로파일을 사용하지 않으면 클라이언트 설치 후 처음 부트 시 해당 클라이언트에 대화식 도구가 열려서 클라이언트 구성이 완료됩니다.

### 클라이언트를 설치 및 구성 지침과 일치

SI를 사용할 때 먼저 SI 서버를 설정합니다. 클라이언트가 네트워크를 통해 부트될 때 SI 서버에서 설치 서비스를 사용합니다.

클라이언트는 해당 클라이언트 구조의 기본 설치 서비스 또는 지정된 설치 서비스를 사용합니다. 설치 서비스는 이 장에 설명된 방식을 사용하여 클라이언트를 올바른 설치 및 구성 지침과 일치시킵니다.

서로 다른 부트 이미지(SPARC 이미지와 x86 이미지 또는 서로 다른 Oracle Solaris 버전)를 사용하는 설치를 정의하려면 이미지마다 별도의 서비스를 만듭니다.

클라이언트를 특정 설치 서비스에 지정하려면 해당 클라이언트를 설치 서비스에 추가합니다(14장, [클라이언트 시스템 설치](#) 참조). 클라이언트의 MAC 주소 및 이 클라이언트가 사용할 설치 서비스의 이름을 지정합니다. 이 MAC 주소의 클라이언트를 부트할 때 클라이언트가 SI 서버로 지정되고 지정된 설치 서비스를 사용합니다. 시스템의 MAC 주소를 찾으려면 [dladm\(1M\)](#) 매뉴얼 페이지의 설명에 따라 `dladm` 명령을 사용합니다.

하나의 설치 서비스에 대해 여러 유형의 설치를 정의하려면 추가 SI 매니페스트 및 시스템 구성 프로파일을 만듭니다. 새 SI 매니페스트 및 프로파일을 SI 설치 서비스에 추가합니다. 어떤 클라이언트가 어떤 SI 매니페스트와 어떤 시스템 구성 프로파일을 사용할지 정의하는 조건을 지정합니다. [“클라이언트별 설치 지침을 설치 서비스와 연관” \[104\]](#)을 참조하십시오.

사용자 정의 AI 매니페스트를 만드는 방법은 10장. 클라이언트 시스템 프로비전을 참조하십시오. 시스템 구성 프로파일을 만드는 방법은 11장. 클라이언트 시스템 구성을 참조하십시오.

## AI 매니페스트 선택

각 클라이언트는 단 하나의 AI 매니페스트만 사용하여 설치를 완료합니다. 다음 알고리즘에 따라 클라이언트의 AI 매니페스트가 선택됩니다.

- 이 설치 서비스에 대해 사용자 정의 AI 매니페스트가 정의되지 않은 경우 기본 AI 매니페스트가 사용됩니다. 기본 AI 매니페스트는 어떤 클라이언트 조건과도 연관되지 않습니다. 기본 AI 매니페스트의 예는 “기본 AI 매니페스트” [164]를 참조하십시오.
- 이 설치 서비스에 대해 사용자 정의 AI 매니페스트가 정의되었지만 클라이언트가 사용자 정의 AI 매니페스트의 조건과 일치하지 않는 경우 기본 AI 매니페스트를 사용합니다.
- 클라이언트가 사용자 정의 AI 매니페스트에 대해 지정된 조건과 일치하는 경우 해당 사용자 정의 매니페스트를 사용합니다.

클라이언트 특성이 여러 AI 매니페스트의 조건과 일치하면 설치할 매니페스트를 선택하기 위해 표 9-1. “조건 키워드 및 조건 계층”에 표시된 순서대로 클라이언트 특성이 평가됩니다. installadm 도구는 동일한 유형의 조건이 겹치지 않는지 확인합니다. 자세한 내용은 “클라이언트별 설치 지침을 설치 서비스와 연관” [104]을 참조하십시오.

아래 표에 지정된 순서대로 여러 겹치지 않는 조건이 사용됩니다. 예를 들어, 한 조건 사양이 클라이언트의 MAC 주소와 일치하고 다른 조건 사양이 동일 클라이언트의 IP 주소와 일치하는 경우 mac이 ipv4보다 선택 우선 순위가 높으므로 MAC 주소 조건 사양과 연관된 매니페스트가 사용됩니다.

### 예 9-1 클라이언트를 AI 매니페스트와 일치

다음 예에서 두 개의 사용자 정의 AI 매니페스트가 동일한 설치 서비스에 추가되었습니다. 이러한 매니페스트와 연관된 클라이언트 조건은 다음과 같습니다. sparc-ent.xml AI 매니페스트가 클라이언트 플랫폼을 지정하는 다음 조건 파일과 함께 서비스에 추가되었습니다.

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,SPARC-Enterprise</value>
  </ai_criteria>
</ai_criteria_manifest>
```

manifest\_mac1.xml AI 매니페스트가 클라이언트 MAC 주소를 지정하는 다음 조건 파일과 함께 서비스에 추가되었습니다.

```
<ai_criteria_manifest>
  <ai_criteria name="mac">
    <value>00:14:4f:a7:65:70</value>
  </ai_criteria>
```

```
</ai_criteria_manifest>
```

MAC 주소가 00:14:4f:a7:65:70인 AI 클라이언트를 설치 중인 경우 여기에 manifest\_mac1.xml이 지정됩니다.

AI 클라이언트가 M4000 또는 M5000인 경우 여기에 sparc-ent.xml이 지정됩니다.

AI 클라이언트가 어떤 AI 매니페스트의 조건과도 일치하지 않으면 설치 서비스용 기본 매니페스트가 클라이언트에 지정됩니다.

## 시스템 구성 프로파일 선택

AI 매니페스트 선택에 사용된 것과 동일한 조건 키워드가 클라이언트의 시스템 구성 프로파일 선택에 사용됩니다. 표 9-1. “조건 키워드 및 조건 계층”을 참조하십시오.

여러 개의 시스템 구성 프로파일을 특정 클라이언트에 대해 선택할 수 있습니다. 선택 범위를 하나의 프로파일로 좁히기 위해 필요한 알고리즘은 없습니다.

클라이언트 특성이 여러 시스템 구성 프로파일의 조건과 일치하면 일치하는 모든 프로파일이 시스템 구성 시 적용됩니다. 예를 들어, 한 조건 사양이 클라이언트의 호스트 이름과 일치하고 다른 조건 사양이 동일 클라이언트의 메모리 크기와 일치하는 경우 두 프로파일 모두 해당 클라이언트 구성 시 사용됩니다.

## 선택 조건

표 9-1. “조건 키워드 및 조건 계층”은 어떤 클라이언트가 특정 AI 매니페스트나 시스템 구성 프로파일을 사용할지 지정하는 조건 키워드를 보여줍니다. 예제 열은 몇몇 가능한 값을 보여줍니다. 조건 키워드 및 값은 installadm 하위 명령인 create-manifest, create-profile, set-criteria와 함께 사용할 수 있습니다.

ipv4, mac, mem, network 사양은 하이픈(-)으로 구분된 범위 값으로 표현할 수 있습니다. 범위 한쪽 끝에 제한 없음을 지정하려면 unbounded를 사용합니다. 아래 mem 예제를 참조하십시오.

arch, cpu, hostname, platform, zonename 사양은 공백으로 구분된 인용 값 목록으로 표현할 수 있습니다. 아래 zonename 예제를 참조하십시오.

-c 옵션을 사용하여 명령줄에 조건 키워드 및 값을 지정합니다.

```
-c criteria=value|list|range
-c mac="aa:bb:cc:dd:ee:ff"
-c mem="2048-unbounded"
-c zonename="zone1 zone2"
```

조건은 XML 파일의 `ai_criteria` 요소에 지정할 수도 있습니다. 이 파일의 내용이 유일한 조건 사양이어야 합니다. `-c` 옵션을 사용하여 명령줄에 조건 파일 이름을 지정합니다. 예제가 테이블에 표시됩니다.

표 9-1 조건 키워드 및 조건 계층

조건 키워드	설명	명령줄 및 XML 파일 예제
arch	uname -m으로 반환된 구조 값: i86pc, sun4u 또는 sun4v	<p>CLI:</p> <pre>-c arch="i86pc"</pre> <p>XML:</p> <pre>&lt;ai_criteria name="arch"&gt;   &lt;value&gt;i86pc&lt;/value&gt; &lt;/ai_criteria&gt;</pre>
cpu	uname -p로 반환된 CPU 클래스 값: i386 또는 sparc	<p>CLI:</p> <pre>-c cpu="sparc"</pre> <p>XML:</p> <pre>&lt;ai_criteria name="cpu"&gt;   &lt;value&gt;sparc&lt;/value&gt; &lt;/ai_criteria&gt;</pre>
hostname	클라이언트 호스트 이름 또는 클라이언트 호스트 이름 목록	<p>CLI, 단일 호스트 이름:</p> <pre>-c hostname="host3"</pre> <p>CLI, 호스트 이름 목록:</p> <pre>-c hostname="host1 host2 host6"</pre> <p>XML, 단일 호스트 이름:</p> <pre>&lt;ai_criteria name="hostname"&gt;   &lt;value&gt;host3&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p>XML, 호스트 이름 목록:</p> <pre>&lt;ai_criteria name="hostname"&gt;   &lt;value&gt;host1 host2 host6&lt;/value&gt; &lt;/ai_criteria&gt;</pre>
ipv4	IP 버전 4 네트워크 주소, 또는 IP 주소 범위	<p>CLI, 단일 IP 주소:</p> <pre>-c ipv4="10.6.68.127"</pre> <p>CLI, IP 주소 범위:</p> <pre>-c ipv4="10.6.68.1-10.6.68.200"</pre> <p>XML, 단일 IP 주소:</p> <pre>&lt;ai_criteria name="ipv4"&gt;</pre>

조건 키워드	설명	명령줄 및 XML 파일 예제
mac	16진수 MAC 주소를 콜론(:) 구분자로 표시, 또는 MAC 주소 범위	<pre>&lt;value&gt;10.6.68.127&lt;/value&gt; &lt;/ai_criteria&gt;  XML, IP 주소 범위:  &lt;ai_criteria name="ipv4"&gt;   &lt;range&gt;     10.6.68.1     10.6.68.200   &lt;/range&gt; &lt;/ai_criteria&gt;</pre> <p>CLI, 단일 MAC 주소:</p> <pre>-c mac="0:14:4F:20:53:97"</pre> <p>CLI, MAC 주소 범위:</p> <pre>-c mac=0:14:4F:20:53:94-0:14:4F:20:53:A0</pre> <p>XML, 단일 MAC 주소:</p> <pre>&lt;ai_criteria name="mac"&gt;   &lt;value&gt;0:14:4F:20:53:97&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p>XML, MAC 주소 범위:</p> <pre>&lt;ai_criteria name="mac"&gt;   &lt;range&gt;     0:14:4F:20:53:94     0:14:4F:20:53:A0   &lt;/range&gt; &lt;/ai_criteria&gt;</pre>
mem	prtcnf로 반환된 MB 단위의 메모리 크기, 또는 메모리 크기 범위  unbounded 키워드는 범위에 상한이 없음을 나타냅니다.	<p>CLI, 하나의 메모리 크기:</p> <pre>-c mem="4096"</pre> <p>CLI, 메모리 크기 범위:</p> <pre>-c mem="2048-unbounded"</pre> <p>XML, 하나의 메모리 크기:</p> <pre>&lt;ai_criteria name="mem"&gt;   &lt;value&gt;4096&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p>XML, 메모리 크기 범위:</p> <pre>&lt;ai_criteria name="mem"&gt;   &lt;range&gt;     2048     unbounded   &lt;/range&gt;</pre>

조건 키워드	설명	명령줄 및 XML 파일 예제
network	IP 버전 4 네트워크 번호, 또는 네트워크 번호 범위	<pre>&lt;/ai_criteria&gt;</pre> <p>CLI, 단일 IP 주소:</p> <pre>-c network="10.0.0.0"</pre> <p>CLI, IP 주소 범위:</p> <pre>-c network="11.0.0.0-12.0.0.0"</pre> <p>XML, 단일 IP 주소:</p> <pre>&lt;ai_criteria name="network"&gt;   &lt;value&gt;10.0.0.0&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p>XML, IP 주소 범위:</p> <pre>&lt;ai_criteria name="network"&gt;   &lt;range&gt;     11.0.0.0     12.0.0.0   &lt;/range&gt; &lt;/ai_criteria&gt;</pre>
platform	x86 시스템의 경우 <code>uname -i</code> , SPARC 시스템의 경우 <code>prtconf -b</code> 로 반환된 플랫폼 이름 포함 값: i86pc M4000 및 M5000 서버의 경우 SUNW,SPARC-Enterprise T4 서버의 경우 ORCL, SPARC-T4-2	<p>CLI:</p> <pre>-c platform="SUNW,SPARC-Enterprise"</pre> <p>XML:</p> <pre>&lt;ai_criteria name="platform"&gt;   &lt;value&gt;SUNW,SPARC-Enterprise&lt;/value&gt; &lt;/ai_criteria&gt;</pre>
zonename	zoneadm list로 표시된 영역의 이름 또는 이름 목록. <a href="#">12장. 영역 설치 및 구성</a> 을 참조하십시오.	<p>CLI, 단일 영역 이름:</p> <pre>-c zonename="myzone"</pre> <p>CLI, 영역 이름 목록:</p> <pre>-c zonename="zoneA zoneB zoneC"</pre> <p>XML, 단일 영역 이름:</p> <pre>&lt;ai_criteria name="zonename"&gt;   &lt;value&gt;myzone&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p>XML, 영역 이름 목록:</p> <pre>&lt;ai_criteria name="zonename"&gt;   &lt;value&gt;zoneA zoneB zoneC&lt;/value&gt; &lt;/ai_criteria&gt;</pre>

## 클라이언트 시스템 프로비전

---

설치 서비스를 만들 때 클라이언트의 프로비전 방법을 지정하는 기본 AI 매니페스트를 가져옵니다. 기본 AI 매니페스트는 운영 체제를 설치할 위치와 설치할 소프트웨어 패키지를 지정하는 파생된 매니페스트입니다. 스트라이핑, 미러링, 분할과 같은 디스크 구성을 지정할 수도 있습니다. AI 매니페스트의 XML 요소에 대한 자세한 내용은 [ai\\_manifest\(4\)](#) 매뉴얼 페이지를 참조하십시오.

이 장에서는 특정 클라이언트에 대한 사용자 정의 AI 매니페스트를 만드는 방법을 설명합니다.

- 사용자 정의 XML AI 매니페스트 파일을 만듭니다. 이 방식은 몇몇 시스템만 사용자 정의 프로비저닝이 필요한 환경에 가장 적합합니다. 설치될 대부분의 시스템은 하드웨어가 같거나 비슷하고 동일하게 프로비전됩니다. [XML AI 매니페스트 파일을 사용자 정의하는 방법 \[140\]](#)을 참조하십시오.
- 설치 타임에 각 클라이언트에 대한 AI 매니페스트를 동적으로 생성하는 스크립트를 작성합니다. 이 방식은 설치 타임에 발견된 클라이언트 특성을 기반으로 각 클라이언트에 대한 사용자 정의 설치를 만듭니다. [파생된 매니페스트 스크립트를 만들고 적용하는 방법 \[142\]](#)을 참조하십시오.
- XML을 편집할 필요 없이 AI 매니페스트 마법사를 사용하여 매니페스트를 만듭니다. [AI 매니페스트 마법사를 사용하여 AI 매니페스트를 만드는 방법 \[159\]](#)을 참조하십시오.

특정 설치 서비스에 XML 매니페스트 파일과 매니페스트 파일 생성을 위한 스크립트를 모두 포함할 수 있습니다. 특정 클라이언트는 정적 또는 스크립트로 생성된, 하나의 AI 매니페스트만 사용합니다. 특정 클라이언트가 사용하는 AI 매니페스트는 매니페스트를 설치 서비스에 추가할 때 지정된 조건에 따라 다릅니다. 클라이언트가 사용자 정의 AI 매니페스트의 사용 조건과 일치하지 않으면 기본 매니페스트가 사용됩니다. 서비스의 AI 매니페스트를 해당 서비스의 기본값이 되도록 지정할 수 있습니다.

### XML AI 매니페스트 파일 사용자 정의

다음 절차를 사용하여 사용자 정의 XML AI 매니페스트 파일을 만들고 적용합니다.

## ▼ XML AI 매니페스트 파일을 사용자 정의하는 방법

### 1. 관리자가 됩니다.

자세한 내용은 “How to Use Your Assigned Administrative Rights” in “Oracle Solaris 11.2 Administration: Security Services”을 참조하십시오.

### 2. 기존 AI 매니페스트를 복사합니다.

AI 설치 서비스를 만들 때 해당 설치 서비스에 기본 AI 매니페스트가 있습니다. 설치 서비스 만들기에 대한 자세한 내용은 [8장. AI 서버 설정](#)을 참조하십시오.

#### a. 기존 매니페스트를 나열합니다.

`installadm list` 명령을 사용하여 특정 설치 서비스와 이미 연관된 AI 매니페스트가 무엇인지 확인합니다.

```
$ installadm list -m -n solaris11_2-i386
Service Name      Manifest Name Type   Status  Criteria
-----
solaris11_2-i386  orig_default  derived default none
```

#### b. 특정 매니페스트의 복사본을 검색합니다.

`installadm export` 명령을 사용하여 이 기본 매니페스트나 이 서비스에 추가된 다른 AI 매니페스트의 내용을 추출합니다.

```
# installadm export -n solaris11_2-i386 -m orig_default -o mem1
```

이제 `orig_default`의 복사본이 `mem1` 파일에 있습니다.

### 3. 매니페스트 복사본을 수정합니다.

`mem1`을 수정합니다. [ai\\_manifest\(4\)](#) 매뉴얼 페이지에 있는 정보에 따라 태그 및 값을 추가하면 됩니다.

### 4. 새 매니페스트를 설치 서비스에 추가합니다.

새 AI 매니페스트를 적절한 AI 설치 서비스에 추가합니다. 어떤 클라이언트가 이러한 설치 지침을 사용할지 정의하는 조건을 지정하면 됩니다.

```
# installadm create-manifest -n solaris11_2-i386 -f ./mem1 -m mem1 \
-c mem="2048-unbounded"
```

`-c` 옵션을 여러 개 지정할 수 있습니다. 또는 `-c`를 사용하여 여러 클라이언트 조건이 포함된 파일을 사용합니다. 클라이언트 조건 지정에 대한 내용은 [9장. 설치 사용자 정의](#) 및 `set-criteria` 하위 명령을 참조하십시오.

이 명령이 실행된 후 `list` 하위 명령이 다음 내용을 보여줍니다.

```
# installadm list -m -n solaris11_2-i386
Service Name      Manifest Name      Type   Status  Criteria
-----

```

```
solaris11_2-i386          mem1          derived active   mem = 2048 MB -
unbounded

orig_default            derived default  none
```

#### ■ 새 매니페스트를 기본값으로 만듭니다.

매니페스트 파일 또는 파생된 매니페스트 스크립트가 서비스의 기본 매니페스트나 스크립트가 되도록 지정할 수 있습니다. 이미 서비스에 추가된 매니페스트와 스크립트 중에서 기본값을 변경하려면 `set-service` 하위 명령과 함께 `-M` 옵션을 사용합니다.

```
# installadm set-service -M mem1 -n solaris11_2-i386
# installadm list -m -n solaris11_2-i386
Service Name          Manifest Name      Type   Status          Criteria
-----
solaris11_2-i386     mem1              derived default / active mem =
2048 MB - unbounded

orig_default          derived inactive   none
```

이 예에서 원래의 기본값은 사용할 클라이언트를 지정하는 조건이 없으므로 지금 비활성입니다. 기본 매니페스트나 스크립트만 클라이언트 선택 조건이 없으므로 계속 활성화될 수 있습니다.

#### ■ 새 매니페스트를 기본값으로 추가합니다.

이 서비스에 대한 새 기본 매니페스트나 스크립트를 추가하려면 `create-manifest`와 함께 `-d` 옵션을 사용합니다. 지정된 조건이 있으면 저장되고 다른 매니페스트가 기본값이 될 때까지 무시됩니다.

```
# installadm create-manifest -n solaris11_2-i386 -d \
-f ./region1.xml -m region1
# installadm list -m -n solaris11_2-i386
Service Name          Manifest Name      Type   Status          Criteria
-----
solaris11_2-i386     mem1              derived active   mem = 2048 MB -
unbounded

region1              xml    default none
orig_default          derived inactive none
```

#### ■ 기존 매니페스트를 사용자 정의합니다.

`installadm update-manifest` 명령을 사용하여 새 매니페스트나 스크립트를 추가하지 않고 기존 매니페스트나 스크립트의 내용을 변경합니다. 조건, 기본 상태 및 매니페스트 이름이나 스크립트 이름은 업데이트 결과로 바뀌지 않습니다.

```
# installadm update-manifest -n solaris11_2-i386
-f ./newregion1.xml -m region1
```

### 5. 사용자 정의된 매니페스트를 검증합니다.

`create-manifest` 및 `update-manifest` 하위 명령은 XML 매니페스트 파일을 설치 서비스에 추가하기 전에 구문상 검증합니다. 시는 클라이언트 설치 타임에 시 매니페스트를 의미상 검증합니다.

---

**참고** - 잘못된 매니페스트가 클라이언트에 제공되면 자동 설치가 중단됩니다. 검증 실패의 원인을 조사하려면 클라이언트의 `/system/volatile/install_log`를 확인하십시오.

---

`installadm list, export, create-manifest, set-criteria, update-manifest` 및 `set-service` 하위 명령에 대한 자세한 내용은 “[설치 서비스 작업](#)” [96]을 참조하십시오.

## 클라이언트 설치 타임에 SI 매니페스트 만들기

클라이언트 설치 전에 사용자 정의 SI 매니페스트를 만드는 대안은, 클라이언트 설치 타임에 각 클라이언트에 대한 SI 매니페스트를 동적으로 생성하는 스크립트를 작성하는 것입니다. 스크립트로 환경 변수 및 다른 클라이언트 구성 정보를 질의하여 각 클라이언트에 대한 사용자 정의 SI 매니페스트를 만들 수 있습니다. 매니페스트는 설치 타임에 발견된 각 클라이언트의 속성을 기반으로 하므로 파생된 매니페스트라고 합니다.

파생된 매니페스트는 많은 수의 시스템을 거의 동일하게 설치할 수 있으므로 이러한 시스템의 SI 매니페스트가 비교적 작은 차이를 보이는 경우 특히 유용합니다. 이 시스템 그룹에 공통되는 설치 매개변수를 지정하는 SI 매니페스트를 만듭니다. 이 공통 매니페스트를 기초로 사용하여, 각 클라이언트를 설치할 때 클라이언트마다 다른 매개변수를 공통 매니페스트에 추가하는 파생된 매니페스트 스크립트를 만듭니다. 예를 들어, 파생된 매니페스트 스크립트는 각 클라이언트 시스템에 연결된 디스크의 개수와 크기를 감지하고, 클라이언트 설치 타임에 SI 매니페스트를 수정하여 각 클라이언트에 대한 사용자 정의 디스크 레이아웃을 지정할 수 있습니다.

### ▼ 파생된 매니페스트 스크립트를 만들고 적용하는 방법

**1. 수정할 매니페스트를 선택합니다.**

수정용 기본 매니페스트로 사용할 기존 SI 매니페스트를 식별합니다.

스크립트를 개발하고 테스트하려면 로컬 복사본으로 작업할 수 있습니다. 설치 타임에, 이 파생된 매니페스트 스크립트를 사용할 각 클라이언트에서 기본 매니페스트에 액세스할 수 있어야 합니다.

**2. 매니페스트를 수정하는 스크립트를 작성합니다.**

설치 중인 클라이언트의 속성을 기반으로 설치 타임에 기본 매니페스트를 동적으로 수정하는 스크립트를 작성합니다.

**3. 스크립트를 설치 서비스에 추가합니다.**

파생된 매니페스트 스크립트를 적절한 SI 설치 서비스에 추가합니다. 설치 타임에 어떤 클라이언트가 이 스크립트를 사용하여 설치 지침을 만들지 정의하는 조건을 지정하면 됩니다. 클

라이언트 선택 조건을 지정하지 않으려면 이 스크립트를 서비스의 기본 SI 매니페스트로 추가할 수 있습니다.

SI가 클라이언트 설치 타임에 스크립트를 실행하여 SI 매니페스트의 인스턴스를 생성합니다. SI가 결과 매니페스트를 구문상 검증합니다.

---

**참고** - 매니페스트가 생성되지 않거나 파생된 매니페스트가 검증되지 않으면 클라이언트 설치가 중단됩니다. 검증 실패의 원인을 조사하려면 클라이언트의 `/system/volatile/install_log`를 확인하십시오.

---

클라이언트 설치를 성공하면 파생된 매니페스트가 클라이언트의 `/var/log/install/derived/manifest.xml`로 복사되고, 매니페스트를 파생시키는 데 사용된 스크립트가 `/var/log/install/derived/manifest_script`로 복사됩니다.

## 파생된 매니페스트 스크립트 만들기

일반적으로 파생된 매니페스트 스크립트는 클라이언트에서 정보를 검색하고 해당 정보를 토대로 기본 SI 매니페스트를 수정하여 이 클라이언트에 대한 사용자 정의 SI 매니페스트를 만듭니다. 파생된 매니페스트 스크립트는 부분적 SI 매니페스트 여러 개를 결합할 수도 있습니다. 최종의 파생된 매니페스트는 완전해야 하고 검증을 통과해야 합니다.

파생된 매니페스트 스크립트는 이미지에서 지원되는 일종의 스크립트일 수 있습니다. 예를 들어, `ksh93` 및 `python`은 기본적으로 이미지에 있습니다. 다른 종류의 스크립트를 사용하려면 이미지에서 필요한 사항을 지원하는지 확인하십시오.

## 클라이언트 속성 검색

파생된 매니페스트 스크립트는 시스템 속성을 읽는 명령을 실행할 수 있습니다. SI가 `aiuser` 역할로 스크립트를 실행합니다. `aiuser` 역할에는 비권한 사용자의 모든 권한과 더불어 다음 추가 권한이 있습니다.

```
solaris.network.autoconf.read
solaris.smf.read.*
```

`aiuser` 역할은 다른 비권한 사용자보다 더 많은 정보를 읽을 수 있다는 점을 제외하면 권한이 없는 사용자입니다. `aiuser` 역할은 시스템을 변경할 수 없습니다.

역할, 프로필 및 권한에 대한 자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)을 참조하십시오.

시스템 속성을 읽는 명령을 사용하는 것 외에도, 다음 표에 나오는 환경 변수를 통해 클라이언트 속성을 사용할 수 있습니다.

표 10-1 클라이언트 속성 환경 변수

환경 변수 이름	설명
SI_ARCH	설치될 클라이언트의 구조입니다. <code>uname -p</code> 의 출력과 같습니다.
SI_CONFIG_PROFILE_DIR	사용자가 제공한 시스템 구성 프로파일이 저장되고 설치 서비스에 사용되는 디렉토리입니다.
SI_CPU	설치될 클라이언트의 ISA 또는 프로세서 유형입니다. <code>uname -p</code> 의 출력과 같습니다.
SI_DISKNAME_#	클라이언트에서 발견된 디스크의 이름을 나타내는 플랫폼 형식의 변수 세트입니다. <code>SI_DISKNAME_#</code> 변수의 <code>SI_NUMDISKS</code> 번호가 존재합니다. 여기서 #은 1부터 시작해서 최대 <code>SI_NUMDISKS</code> 까지의 경수로 바뀝니다. 이 변수 세트는 <code>SI_DISKSIZE_#</code> 에 설명된 변수 세트와 상관이 있습니다.
SI_DISKSIZE_#	클라이언트에서 발견된 디스크의 디스크 크기를 나타내는 플랫폼 형식의 변수 세트입니다. <code>SI_DISKSIZE_#</code> 변수의 <code>SI_NUMDISKS</code> 번호가 존재합니다. 여기서 #은 1부터 시작해서 최대 <code>SI_NUMDISKS</code> 까지의 경수로 바뀝니다. 이 변수 세트는 <code>SI_DISKNAME_#</code> 에 설명된 변수 세트와 상관이 있습니다. 크기는 MB 단위의 경수입니다.
SI_HOSTADDRESS	설치 환경에 설정된 클라이언트의 IP 주소입니다.
SI_HOSTNAME	설치 환경에 설정된 클라이언트의 호스트 이름입니다.
SI_INSTALL_SERVICE	매니페스트 스크립트를 얻기 위해 사용된 설치 서비스의 이름입니다. 이 환경 변수는 매체 부트가 아닌 네트워크 부트에만 값을 가집니다.
SI_KARCH	클라이언트의 커널 구조입니다. <code>uname -m</code> 의 출력과 같습니다.
SI_MEMSIZE	클라이언트의 물리적 메모리 양입니다. 크기는 MB 단위의 경수입니다.
SI_NATISA	클라이언트의 고유의 명령 세트 구조입니다. <code>isainfo -n</code> 의 출력과 같습니다.
SI_NETWORK	클라이언트의 네트워크 번호입니다. 네트워크 번호는 (IP_ADDR & netmask)입니다.
SI_NUMDISKS	클라이언트의 디스크 수입니다.
SI_PLATFORM (or SI_MODEL)	클라이언트의 플랫폼입니다. x86 시스템의 경우 <code>uname -i</code> , SPARC 시스템의 경우 <code>prtconf -b</code> 의 출력과 같습니다.
SI_SYSPKG	클라이언트에 있는 Oracle Solaris 통합 패키지의 릴리스입니다(현재 이름은 <code>entire</code> ). 클라이언트의 전체 패키지가 <code>pkg://solaris/entire@0.5.11,5.11-0.175.0.0.0.2.0:20111020T143822Z</code> 일 경우 <code>SI_SYSPKG</code> 의 값은 <code>pkg:/entire@0.5.11-0.175.0</code> 입니다. 업데이트 릴리스 또는 <code>sru</code> 의 경우 클라이언트의 전체 패키지가 <code>pkg://solaris/entire@0.5.11,5.11-0.175.1.19.0.6.0:20140508T221351Z</code> 이면 <code>SI_SYSPKG</code> 의 값이 <code>pkg:/entire@0.5.11-0.175.1</code> 입니다.

## AI 매니페스트 사용자 정의

AI 매니페스트의 XML 요소를 추가하거나 수정하려면 `/usr/bin/aimanifest` 명령을 사용하십시오.

`aimanifest`로 수정할 파일은 최소한 다음 조각을 포함해야 합니다.

- 개발 중인 XML 매니페스트에 유효한 DTD에 대한 `!DOCTYPE` 참조
- 이 DTD에 대한 루트 요소

다음 예제에서는 AI 매니페스트에 대한 최소 기본 매니페스트 파일을 보여주며, 이 파생된 매니페스트 스크립트를 추가할 설치 서비스의 AI DTD 파일을 지정합니다.

```
<!DOCTYPE auto_install SYSTEM "file:///imagepath/auto_install/ai.dtd.1">
<auto_install/>
```

*imagepath* 인수 값은 다음 명령으로 반환된 경로입니다. 여기서 *svcname*은 이 파생된 매니페스트 스크립트를 추가할 설치 서비스의 이름입니다.

```
$ installadm list -v -n svcname
```

---

**참고** - AI 클라이언트를 설치하기 위해 스크립트를 사용하기 전에 *imagepath*를 다시 *///usr/share*로 변경하십시오.

---

*aimanifest* 명령의 *load* 하위 명령을 사용하여 파생된 매니페스트 스크립트에서 다른 *aimanifest* 호출 전에 기본 매니페스트를 로드합니다. 로드할 파일은 클라이언트 설치 타임에 클라이언트에서 액세스할 수 있어야 합니다. 예를 들어, 대상 설치 서비스의 *imagepath/**auto\_install/manifest/*에서 매니페스트를 로드할 수 있습니다.

이 장의 예제에서는 */usr/share/auto\_install/manifest/default.xml* 파일을 로드합니다. */usr/share/auto\_install/manifest/*의 샘플 매니페스트는 대상 설치 서비스의 매니페스트와 다를 수 있습니다. 실제 운용에는 */usr/share/auto\_install/manifest/*에서 매니페스트를 로드하면 안됩니다.

*load* 하위 명령은 부분 매니페스트를 로드/삽입하는 데 사용할 수도 있습니다.

*add* 하위 명령을 사용하여 새 요소를 추가합니다. *set* 하위 명령을 사용하여 요소 속성을 추가하거나 요소 또는 속성 값을 변경합니다. 자세한 내용은 [aimanifest\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 이 매뉴얼 페이지와 이어지는 예제 스크립트에서 *aimanifest* 명령 사용의 예를 제공합니다.

---

**참고** - *aimanifest* 명령에 지정된 값에 다음 문자 중 하나가 있을 경우 해당 값을 작은 따옴표나 큰 따옴표로 묶어야 문자가 XML 경로 이름의 일부로 해석되지 않습니다.

```
/'"@[]=
```

사용된 셸 규칙에 따라, 셸이 따옴표를 제거하거나 해석하지 않도록 선행 백슬래시 문자(`\`)로 따옴표를 이스케이프 처리해야 할 수도 있습니다.

---

다음 예는 패키지 이름 *pkg:/entire*가 포함된 *software\_data* 요소의 동작을 반환합니다. 이 예에서 슬래시가 특수 문자이므로 *pkg:/entire* 둘레에 따옴표가 필요합니다. 이 명령이 *ksh93* 스크립트와 같은 셸 스크립트에서 호출될 경우 따옴표를 이스케이프 처리하기 위해 백슬래시 문자가 필요합니다.

```
# /usr/bin/aimanifest get software_data[name="\pkg:/entire"]@action
```

---

**작은 정보** - 최적의 사용법으로, 오류 시 중지하는 트랩을 설정합니다.

---

다음 부분 스크립트는 파생된 매니페스트 스크립트의 좋은 모델입니다.

```
#!/bin/ksh93
```

```
SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1
```

```
function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load baseAlmanifest.xml

# Customize AI manifest. For example:
/usr/bin/aimanifest load -i manifest_fragment.xml
/usr/bin/aimanifest set origin@name file:///net/myserver/myrepo/repo.redist

exit $SCRIPT_SUCCESS
```

## 파생된 매니페스트 스크립트의 예제

이 절에서는 파생된 매니페스트 스크립트를 작성하여 클라이언트 속성을 확인하고 해당 정보를 토대로 AI 매니페스트를 사용자 정의하는 방법을 보여줍니다. 이러한 예제가 유효한 AI 매니페스트 생성에 필요한 모든 정보를 반드시 포함하지는 않습니다.

이러한 예제를 사용하려면 다음 설정 단계를 수행합니다.

1. AIM\_MANIFEST 환경 변수를 스크립트가 AI 매니페스트를 개발할 위치로 설정합니다.  
aimanifest 명령이 \$AIM\_MANIFEST 파일을 수정할 때마다 파일이 재작성됩니다. aimanifest를 load, add, delete 또는 set 하위 명령과 함께 호출할 때마다 AIM\_MANIFEST 파일이 열리고, 수정되고, 저장됩니다. AIM\_MANIFEST가 설정되지 않으면 aimanifest 명령이 실패합니다.
2. AIM\_LOGFILE 환경 변수를 스크립트가 상세 정보 및 오류 메시지를 작성할 수 있는 위치로 설정합니다.  
설정하면 aimanifest 명령이 각 aimanifest 호출의 하위 명령 이름, 인수 값, 반환 상태를 화면과 \$AIM\_LOGFILE 파일에 기록합니다.
3. 스크립트를 실행한 시스템에서 aimanifest 명령이 사용 가능한지 확인합니다.  
aimanifest 명령을 사용할 수 없으면 auto-install-common 패키지를 설치합니다.
4. 환경 변수를 설정합니다. 이러한 예제는 환경 변수를 사용하여 클라이언트에 대한 정보를 검색하는 방법을 보여줍니다. 이러한 예제를 사용하려면 환경 변수에 대한 값을 설정해야 합니다.  
AI를 사용하여 시스템을 설치할 때 표 10-1. “클라이언트 속성 환경 변수”에 표시된 환경 변수의 값을 파생된 매니페스트 스크립트에 사용할 수 있습니다.

### 예 10-1 디스크 크기에 준하여 디스크 분할 지정

이 예는 Oracle Solaris fdisk 분할 영역의 대상 디스크의 크기가 1TB보다 큰 경우 그 크기의 절반만 사용하도록 AI 매니페스트를 사용자 정의합니다. 이 스크립트를 실행할 때 SI\_DISKSIZE\_1을 1TB보다 작게 설정했다가 1TB보다 크게 설정해 봅니다. 또한 스크립트를

실행하기 전에 `SI_NUMDISKS` 및 `SI_DISKNAME_1`을 설정합니다. 지정된 분할 영역은 x86 클라이언트에만 적용되므로 이 스크립트는 x86 클라이언트에만 사용할 수 있습니다.

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# Check that there is only one disk on the system.
if [[ $SI_NUMDISKS -gt "1" ]] ; then
    print -u2 "System has too many disks for this script."
    exit $SCRIPT_FAILURE
fi

/usr/bin/aimanifest add \
    /auto_install/ai_instance/target/disk/disk_name@name $SI_DISKNAME_1

if [[ $SI_DISKSIZE_1 -gt "1048576" ]] ; then
    typeset -i PARTN_SIZE=$SI_DISKSIZE_1/2

    # Default action is to create.
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk[disk_name@name=\"$SI_DISKNAME_1\"]/
partition@name 1
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk/partition[@name=1]/size@val \
        ${PARTN_SIZE}mb
else
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk[disk_name@name=\"$SI_DISKNAME_1\"]/
partition@action \
        use_existing_solaris2
fi
exit $SCRIPT_SUCCESS
```

`SI_DISKSIZE_1` 값이 1048576보다 작거나 같은 클라이언트의 경우 `$AIM_MANIFEST`에 다음과 같은 요소가 추가됩니다.

```
<target>
  <disk>
    <disk_name name="/dev/dsk/c0t0d0s0"/>
    <partition action="use_existing_solaris2"/>
  </disk>
  <!-- <logical> section -->
</target>
```

SI\_DISKSIZE\_1 값이 1048576보다 큰 클라이언트의 경우 SI\_DISKSIZE\_1 값에 따라 다음과 비슷한 요소가 \$AIM\_MANIFEST에 추가됩니다.

```
<target>
  <disk>
    <disk_name name="/dev/dsk/c0t0d0s0"/>
    <partition name="1">
      <size val="524288mb"/>
    </partition>
  </disk>
  <!-- <logical> section -->
</target>
```

분할 영역에 대한 별도의 디스크 사양을 만들지 않기 위해 분할 영역을 추가하는 명령에 disk\_name이 지정됩니다. 이 예제의 스크립트는 분할 영역이 \$SI\_DISKNAME\_1 디스크에 있고 다른 디스크에는 없도록 지정합니다. 이 예제의 해당 라인을 다음 라인으로 바꾸면 의도한 결과를 얻지 못합니다.

```
    /usr/bin/aimanifest add \
      /auto_install/ai_instance/target/disk/partition@name 1
  /usr/bin/aimanifest add \
    /auto_install/ai_instance/target/disk/partition[@name=1]/size@val \
      ${PARTN_SIZE}mb
else
  /usr/bin/aimanifest add \
    /auto_install/ai_instance/target/disk/partition@action \
      use_existing_solaris2
```

위에 표시된 출력 대신, 이 스크립트는 다음과 같은 잘못된 출력을 제공합니다.

```
<target>
  <disk>
    <disk_name name="c0t0d0s0"/>
  </disk>
  <disk>
    <partition name="1">
      <size val="524288mb"/>
    </partition>
  </disk>
</target>
```

#### 예 10-2 추가 디스크의 존재 여부에 준하여 루트 풀 레이아웃 지정

이 예는 두번째 디스크가 존재할 경우 루트 풀의 미러를 구성하고 세번째 디스크가 존재할 경우 3방향 미러를 구성하도록 SI 매니페스트를 사용자 정의합니다. 스크립트를 실행하기 전에 SI\_NUMDISKS 및 SI\_DISKNAME\_1을 설정합니다. SI\_NUMDISKS에 설정된 값에 따라 필요할 경우 SI\_DISKNAME\_2, SI\_DISKNAME\_3 등을 설정합니다. 이 환경 변수는 SI 설치 중에 파생된 매니페스트 스크립트에 설정되어 사용할 수 있습니다.

이 예는 aimanifest 반환 경로(-r 옵션) 사용 방법을 보여줍니다. 반환 경로에 대한 자세한 내용은 [aimanifest\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# Use the default if there is only one disk.
if [[ $SI_NUMDISKS -ge 2 ]] ; then
    typeset -i disk_num

    # Turn on mirroring. Assumes a root zpool is already set up.
    vdev=$(/usr/bin/aimanifest add -r \
        target/logical/zpool[@name=rpool]/vdev@name mirror_vdev)
    /usr/bin/aimanifest set ${vdev}@redundancy mirror

    for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
        eval curr_disk="$SI_DISKNAME_${disk_num}"
        disk=$(/usr/bin/aimanifest add -r target/disk@in_vdev mirror_vdev)
        /usr/bin/aimanifest set ${disk}@in_zpool rpool
        /usr/bin/aimanifest set ${disk}@whole_disk true
        disk_name=$(/usr/bin/aimanifest add -r \
            ${disk}/disk_name@name $curr_disk)
        /usr/bin/aimanifest set ${disk_name}@name_type ctd
    done
fi
exit $SCRIPT_SUCCESS
```

c0t0d0 및 c0t1d0이라는 두 디스크가 있는 시스템의 경우 이 예제에서 다음 XML 요소가 출력됩니다.

```
<target>
  <disk in_vdev="mirror_vdev" in_zpool="rpool" whole_disk="true">
    <disk_name name="c0t0d0" name_type="ctd"/>
  </disk>
  <disk in_vdev="mirror_vdev" in_zpool="rpool" whole_disk="true">
    <disk_name name="c0t1d0" name_type="ctd"/>
  </disk>
  <logical>
    <zpool name="rpool" is_root="true">
      <vdev name="mirror_vdev" redundancy="mirror"/>
      <filesystem name="export" mountpoint="/export"/>
      <filesystem name="export/home"/>
      <be name="solaris"/>
    </zpool>
  </logical>
</target>
```

**예 10-3** 지정된 크기의 디스크가 적어도 두 개 있는 경우 미러링된 구성 지정

이 예는 시스템에 적어도 두 개의 200GB 디스크가 있는 경우 미러링된 구성을 지정하도록 AI 매니페스트를 사용자 정의합니다. 처음 발견된 두 디스크(최소한 200GB)를 사용합니다. 스크립트를 실행하기 전에 테스트 환경에 `SI_NUMDISKS`, `SI_DISKNAME_1`, `SI_DISKSIZE_1`을 설정합니다. 또한 `SI_NUMDISKS`에 설정된 값에 따라 필요할 경우 `SI_DISKNAME_2`, `SI_DISKSIZE_2` 등을 설정합니다. 이 환경 변수는 AI 설치 중에 파생된 매니페스트 스크립트에 설정되어 사용할 수 있습니다.

이 예는 동일한 경로를 가진 노드가 여러 개 존재할 때 노드를 수정하는 방법을 보여줍니다. 셸 구현은 `aimanifest`의 반환 경로(`-r`) 옵션을 사용하여 특정 노드로 경로를 반환하고 해당 경로를 통해 동일한 노드를 추가로 수정합니다. Python 구현은 하위 경로 사용(노드 경로 안에 `[]` 사용)을 통해 동일한 노드를 추가로 수정하는 방법을 보여줍니다.

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

# Find the disks first.
typeset found_1
typeset found_2
typeset -i disk_num

for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
    eval curr_disk="$SI_DISKNAME_${disk_num}"
    eval curr_disk_size="$SI_DISKSIZE_${disk_num}"
    if [[ $curr_disk_size -ge "204800" ]] ; then
        if [ -z $found_1 ] ; then
            found_1=$curr_disk
        else
            found_2=$curr_disk
            break
        fi
    fi
done

# Now, install them into the manifest.
# Let the installer take the default action if two large disks are not found.

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

if [[ -n $found_2 ]] ; then
    # Turn on mirroring.
    vdev=$(/usr/bin/aimanifest add -r \
        /auto_install/ai_instance/target/logical/zpool/vdev@redundancy mirror)
```

```

/usr/bin/aimanifest set ${vdev}@name mirror_vdev

disk=$(/usr/bin/aimanifest add -r \
    /auto_install/ai_instance/target/disk@in_vdev mirror_vdev)
disk_name=$(/usr/bin/aimanifest add -r ${disk}/disk_name@name $found_1)
/usr/bin/aimanifest set ${disk_name}@name_type ctd

disk=$(/usr/bin/aimanifest add -r \
    /auto_install/ai_instance/target/disk@in_vdev mirror_vdev)
disk_name=$(/usr/bin/aimanifest add -r ${disk}/disk_name@name $found_2)
/usr/bin/aimanifest set ${disk_name}@name_type ctd
fi

exit $SCRIPT_SUCCESS

```

다음 스크립트는 위의 Korn 셸 버전의 Python 버전입니다.

```

#!/usr/bin/python2.6

import os
import sys

from subprocess import check_call, CalledProcessError

SCRIPT_SUCCESS = 0
SCRIPT_FAILURE = 1

def main():

    # Find the disks first.
    found_1 = ""
    found_2 = ""

    si_numdisks = int(os.environ["SI_NUMDISKS"])
    for disk_num in range(1, si_numdisks + 1):
        curr_disk_var = "SI_DISKNAME_" + str(disk_num)
        curr_disk = os.environ[curr_disk_var]
        curr_disk_size_var = "SI_DISKSIZE_" + str(disk_num)
        curr_disk_size = os.environ[curr_disk_size_var]
        if curr_disk_size >= "204800":
            if not len(found_1):
                found_1 = curr_disk
            else:
                found_2 = curr_disk
            break

    # Now, write the disk specifications into the manifest.
    # Let the installer take the default action if two large disks are not found.

    try:
        check_call(["/usr/bin/aimanifest", "load",
            "/usr/share/auto_install/manifest/default.xml"])
    except CalledProcessError as err:
        sys.exit(err.returncode)

```

```

if len(found_2):
    try:
        check_call(["/usr/bin/aimanifest", "add",
                    "target/logical/zpool[@name=rpool]/vdev@redundancy", "mirror"])
        check_call(["/usr/bin/aimanifest", "set",
                    "target/logical/zpool/vdev[@redundancy='mirror']@name", "mirror_vdev"])

        check_call(["/usr/bin/aimanifest", "add",
                    "target/disk/disk_name@name", found_1])
        check_call(["/usr/bin/aimanifest", "set",
                    "target/disk/disk_name[@name='" + found_1 + "'" + "@name_type", "ctd"])
        check_call(["/usr/bin/aimanifest", "set",
                    "target/disk[disk_name@name='" + found_1 + "'" + "@in_vdev", "mirror_vdev"])

        check_call(["/usr/bin/aimanifest", "add",
                    "target/disk/disk_name@name", found_2])
        check_call(["/usr/bin/aimanifest", "set",
                    "target/disk/disk_name[@name='" + found_2 + "'" + "@name_type", "ctd"])
        check_call(["/usr/bin/aimanifest", "set",
                    "target/disk[disk_name@name='" + found_2 + "'" + "@in_vdev", "mirror_vdev"])
    except CalledProcessError as err:
        sys.exit(err.returncode)

    sys.exit(SUCCESS)

if __name__ == "__main__":
    main()

```

#### 예 10-4 IP 주소에 준하여 설치할 패키지 지정

이 예는 클라이언트의 IP 주소가 지정된 범위에 있는 경우 하나의 패키지를 설치하고, 클라이언트의 IP 주소가 다른 범위에 있는 경우 다른 패키지를 설치하도록 SI 매니페스트를 사용자 정의합니다. 스크립트를 실행하기 전에 테스트 환경에 SI\_HOSTADDRESS를 설정합니다. 이 환경 변수는 SI 설치 중에 파생된 매니페스트 스크립트에 설정되어 사용할 수 있습니다.

```

#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# First determine which range the host IP address of the client is in.
echo $SI_HOSTADDRESS | sed 's/\./ /g' | read a b c d

# Assume all systems are on the same class A and B subnets.

```

```
# If the system is on class C subnet = 100, then install the /pkg100 package.
# If the system is on class C subnet = 101, then install the /pkg101 package.
# Otherwise, do not install any other additional package.

if ((c == 100)) ; then
    /usr/bin/aimanifest add \
        software/software_data[@action='install']/name pkg:/pkg100
fi
if ((c == 101)) ; then
    /usr/bin/aimanifest add \
        software/software_data[@action='install']/name pkg:/pkg101
fi

exit $SCRIPT_SUCCESS
```

**예 10-5** 대상 디스크가 최소한 특정 크기가 되도록 지정

이 예는 최소한 50GB인 디스크에만 설치하도록 AI 매니페스트를 사용자 정의합니다. 더 작은 디스크는 무시합니다. 스크립트를 실행하기 전에 테스트 환경에 SI\_NUMDISKS, SI\_DISKNAME\_1, SI\_DISKSIZE\_1을 설정합니다. 또한 SI\_NUMDISKS에 설정된 값에 따라 필요할 경우 SI\_DISKNAME\_2, SI\_DISKSIZE\_2 등을 설정합니다. 이 환경 변수는 AI 설치 중에 파생된 매니페스트 스크립트에 설정되어 사용할 수 있습니다.

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

typeset found
typeset -i disk_num
for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
    eval curr_disk="$SI_DISKNAME_${disk_num}"
    eval curr_disk_size="$SI_DISKSIZE_${disk_num}"
    if [[ $curr_disk_size -ge "512000" ]] ; then
        found=$curr_disk
        /usr/bin/aimanifest add \
            /auto_install/ai_instance/target/disk/disk_name@name $found
        break
    fi
done

if [[ -z $found ]] ; then
    exit $SCRIPT_FAILURE
```

```
fi

exit $SCRIPT_SUCCESS
```

**예 10-6**            시스템 구성 프로파일 추가

때때로 각 클라이언트의 시스템 구성을 변경해야 합니다. 각 클라이언트에 대해 AI 서버에 개별 시스템 구성 프로파일을 만들기보다는 파생된 매니페스트 스크립트를 구성하여 프로파일을 만들 수 있습니다. 프로파일을 /system/volatile/profile에 저장해야 설치 서비스에 사용됩니다. 이 예제에서는 클라이언트를 재구성할 때 로컬 기본 라우터에 대한 설정이 사용됩니다.

```
ROUTER-CONFIG=/system/volatile/profile/router-config.xml
ROUTER=`netstat -rn | grep "^default" | awk '{print $2}'`

cat<<EOF>${ROUTER-CONFIG}
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="router">
  <service name="network/install" version="1" type="service">
    <instance name="default" enabled="true">
      <property_group name="install_ipv4_interface" type="application">
        <propval name="default_route" type="net_address_v4" value="${ROUTER}"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
EOF
```

**예 10-7**            부정확한 매니페스트 사양의 스크립트

이 예의 스크립트는 오류를 포함합니다.

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
  exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

/usr/bin/aimanifest set \
  software[@type="IPS"]/software_data/name pkg:/driver/pcmcia
/usr/bin/aimanifest set \
  software/software_data[@name=pkg:/driver/pcmcia]@action uninstall

return $SCRIPT_SUCCESS
```

이 예는 \$AIM\_MANIFEST 작성과 관련한 세 가지 문제가 있습니다.

1. `aimanifest`의 `set` 하위 명령은 기존 요소나 속성의 값을 변경하거나 새 속성을 만들 수 있습니다. `set` 하위 명령은 새 요소를 만들 수 없습니다. 첫번째 `set` 하위 명령은 새 패키지 이름을 만드는 대신 매니페스트에서 기존 패키지 이름을 수정하려고 시도합니다. 여러 개의 패키지 이름이 매니페스트에 존재할 경우 수정할 패키지를 결정할 수 없어서 모호한 오류가 발생합니다. 이 예의 첫번째 `set` 하위 명령은 `add` 하위 명령이었어야 합니다.
2. 이 예의 두번째 `set` 하위 명령에서 `pkg:/driver/pcmcia` 값의 요소 `name`이 선행 `@` 기호로 지정됩니다. 속성 값은 선행 `@` 기호로 지정되지만, 요소 값은 그렇지 않습니다.
3. `pkg:/driver/pcmcia` 값이 따옴표로 둘러싸여야 합니다. 슬래시나 다른 특수 문자가 포함된 값은 따옴표를 붙여야 합니다.

이 예의 `set` 라인을 다음 라인으로 바꿔야 합니다.

```
/usr/bin/aimanifest add \  
    software[@type="IPS"]/software_data[action uninstall]  
/usr/bin/aimanifest add \  
    software/software_data[@action=uninstall]/name pkg:/driver/pcmcia
```

이러한 두 `add` 하위 명령에서 작성 중인 매니페스트의 `software` 섹션 끝에 다음 라인을 추가합니다.

```
<software_data action="uninstall">  
    <name>pkg:/driver/pcmcia</name>  
</software_data>
```

## 파생된 매니페스트 스크립트 테스트

파생된 매니페스트 스크립트를 테스트하려면 AI 설치 환경과 비슷한 환경에서 스크립트를 실행하십시오.

1. 수정할 스크립트에 대한 기본 AI 매니페스트를 설정합니다.
  - a. 스크립트의 첫번째 `aimanifest` 명령이 `aimanifest load` 명령인지 확인합니다. 대상 설치 서비스의 AI 매니페스트 검증에 사용할 수 있는 DTD를 지정하는 `<!DOCTYPE>` 정의가 로드 중인 파일에 있는지 확인합니다. 다음 예제에서는 AI 매니페스트에 대한 최소 기본 매니페스트 파일을 보여주며, 이 파생된 매니페스트 스크립트를 추가할 설치 서비스의 AI DTD 파일을 지정합니다.

```
<!DOCTYPE auto_install SYSTEM "file:///imagepath/auto_install/ai.dtd.1">  
<auto_install/>
```

`imagepath` 인수 값은 다음 명령으로 반환된 경로입니다. 여기서 `svcname`은 이 파생된 매니페스트 스크립트를 추가할 설치 서비스의 이름입니다.

**참고** - 클라이언트에서 스크립트를 사용하기 전에 imagepath를 기본 경로인 `///usr/share`로 재설정해야 합니다.

---

```
$ installadm list -v -n svcname | grep Image
```

- b. AIM MANIFEST를 스크립트가 AI 매니페스트를 개발할 위치로 설정합니다. 이 위치는 비권한 사용자 aiuser가 쓰기 가능해야 합니다.
- 

**참고** - AI가 설치를 진행 중일 때 AIM\_MANIFEST를 설정할 필요가 없습니다. AI가 기본값을 설정합니다.

---

2. AIM\_LOGFILE을 스크립트가 상세 정보 및 오류 메시지를 작성할 수 있는 위치로 설정합니다. 이 위치는 비권한 사용자 aiuser가 쓰기 가능해야 합니다.
- 

**참고** - AI가 설치를 진행 중일 때 AIM\_LOGFILE을 설정할 필요가 없습니다. 이 로그 정보는 더 큰 설치 로그인 `/system/volatile/install_log`의 일부입니다.

---

3. 스크립트를 테스트하는 시스템에서 aimanifest 명령이 사용 가능한지 확인합니다. aimanifest 명령을 사용할 수 없으면 auto-install-common 패키지를 설치합니다.
4. 테스트 환경의 환경 변수를 이 파생된 매니페스트 스크립트를 사용하여 설치할 클라이언트 시스템을 나타내는 값으로 설정합니다. 샘플 파일 `/usr/share/auto_install/derived_manifest_test_env.sh`를 템플릿으로 사용할 수 있습니다. 값을 적절히 변경합니다.  
AI가 설치하는 동안 표 10-1. “클라이언트 속성 환경 변수”에 표시된 환경 변수의 값을 파생된 매니페스트 스크립트에 사용할 수 있습니다.
5. 루트 역할을 맡을 수 있는지 확인합니다. 루트 역할로부터 암호 지정 없이 aiuser 역할을 맡을 수 있습니다.

```
$ su
Password:
# su aiuser -c ./script
#
```

AI가 aiuser 역할로 파생된 매니페스트 스크립트를 실행합니다. AI 설치 환경에 가까우려면 aiuser 역할로 스크립트를 실행합니다. aiuser 역할이 아닌 다른 권한을 가진 사용자로 스크립트를 실행하면 스크립트의 일부 작업 결과가 다를 수 있습니다.

6. 결과 매니페스트에 validate 하위 명령을 사용합니다.

```
$ /usr/bin/aimanifest validate
```

검증을 실패한 경우에만 메시지가 표시됩니다.

의도한 클라이언트 시스템이 AI 서버나 기타 파생된 매니페스트 스크립트를 테스트할 수 있는 시스템과 전혀 다를 수 있습니다. 스크립트에서 호출하는 명령을 사용할 수 없거나 다른 버전의 다른 동작으로 나타날 수 있습니다. 시스템의 구조가 서로 다르거나 디스크 개수와 크기가 서로 다를 수 있습니다. 설명된 대로 테스트 환경에 환경 변수를 설정하면 이러한 차이점 중 일부가 해결됩니다.

## ▼ 설치 환경에서 파생된 매니페스트 스크립트를 테스트하는 방법

이 절차에서는 전체 설치 프로세스를 실행하지 않고 의도한 클라이언트 시스템 중 하나에서 파생된 매니페스트 스크립트를 테스트하는 방법을 설명합니다.

1. 해당 클라이언트 시스템에서 AI 이미지를 부트합니다.  
클라이언트 시스템에서 "Text Installer and command line" 모드로 AI 이미지를 부트합니다.
2. 설치 프로그램 초기 메뉴에서 Shell(셸)을 선택합니다.
3. AI 서버에서 스크립트를 복사합니다.  
wget 또는 sftp를 사용하여 AI 서버에서 스크립트를 복사합니다.
4. 스크립트를 디버그합니다.  
다음 방식 중 하나를 사용하여 스크립트를 디버그합니다.

### ■ 스크립트를 수동으로 실행합니다.

### ■ AI를 테스트 모드로 실행합니다.

다음 명령을 사용하여 AI를 테스트 모드로 실행합니다.

```
$ auto-install -m script -i
```

AI 로그 파일 /system/volatile/install\_log를 검사합니다. 스크립트 검증 결과를 나타내는 다음 라인이 로그 파일에 있어야 합니다.

```
Derived Manifest Module: XML validation completed successfully
```

5. 스크립트를 AI 서버로 다시 복사합니다.  
내용을 변경한 경우 스크립트를 AI 서버로 다시 복사합니다.

## 설치 서비스에 파생된 매니페스트 스크립트 추가

XML 매니페스트를 설치 서비스에 추가하는 것과 동일한 방법으로 AI 설치 서비스에 파생된 매니페스트 스크립트를 추가합니다. 동일한 옵션을 사용하여 이 스크립트로 설치 매니페스트를 만들 클라이언트를 선택하기 위한 조건을 지정합니다. 또한 XML 매니페스트를 업데이트

트하는 것처럼 스크립트를 업데이트할 수 있습니다. 스크립트가 서비스의 기본 매니페스트가 되도록 설정할 수 있습니다. 서비스와 연관된 매니페스트를 나열하면 스크립트 및 XML 매니페스트가 둘 다 표시됩니다. 매니페스트 유형 `derived`와 함께 스크립트가 나열됩니다. XML 매니페스트를 내보내는 것처럼 스크립트 내용을 내보낼 수 있습니다.

XML 매니페스트를 설치 서비스에 추가할 때 매니페스트가 검증됩니다. 파생된 매니페스트 스크립트를 설치 서비스에 추가할 때는 스크립트가 검증되지 않습니다.

의도한 클라이언트 시스템과 비슷한 환경에서 스크립트를 실행합니다. 자세한 내용은 [“파생된 매니페스트 스크립트 테스트” \[155\]](#)를 참조하십시오.

적절한 SI 설치 서비스에 스크립트를 추가합니다. 이러한 설치 지침을 사용할 클라이언트를 정의하는 조건을 지정하면 됩니다. 클라이언트 선택 조건을 지정하지 않으려면 `-d` 옵션을 사용하여 이 스크립트를 서비스의 기본 SI 매니페스트로 추가할 수 있습니다.

```
# installadm create-manifest -n solaris11_2-i386 -f ./mac1.ksh -m mac1 \
-c mac=BB:AA:AA:AA:AA:AA
```

여러 `-c` 옵션이나 하나의 `-c` 파일을 지정할 수 있습니다. `set-criteria` 하위 명령도 참조하십시오. 클라이언트 조건 지정에 대한 내용은 [9장. 설치 사용자 정의](#)를 참조하십시오.

`installadm list`, `export`, `create-manifest`, `set-criteria`, `update-manifest` 및 `set-service` 하위 명령에 대한 자세한 내용은 [“설치 서비스 작업” \[96\]](#)을 참조하십시오.

## SI 매니페스트 마법사를 사용하여 SI 매니페스트 만들기

SI 매니페스트 마법사는 수동으로 XML 파일을 편집할 필요 없이 SI 매니페스트를 만드는 데 사용할 수 있는 BUI(브라우저 사용자 인터페이스) 웹 응용 프로그램입니다. SI 서버의 URL을 사용하여 이 응용 프로그램에 연결할 수 있습니다. 이 마법사는 사용자가 SI 매니페스트의 여러 섹션을 구성할 수 있는 8개의 주 화면으로 이루어져 있습니다.

## SI 매니페스트 마법사에 대해 SI 서버 구성

다음 예제에서는 SI 매니페스트 마법사를 사용 안함으로 설정하는 방법과 사용자가 SI 서버에서 매니페스트 파일을 저장할 수 있게 허용하는 방법을 보여줍니다.

**예 10-8** SI 매니페스트 마법사 사용 안함으로 설정

기본적으로 SI 서버가 사용으로 설정되어 있으면 SI 매니페스트 마법사가 사용으로 설정됩니다. SI 매니페스트 마법사를 사용 안함으로 설정하려면 다음 명령을 사용하십시오.

```
# installadm set-server -U
```

**예 10-9** AI 서버에 매니페스트 파일을 저장하도록 허용

AI 매니페스트 마법사로 만든 AI 매니페스트는 기본적으로 AI 서버에 저장할 수 없습니다. 이 파일을 사용자 데스크탑에 저장해야 합니다. AI 서버에 파일을 저장할 수 있도록 하려면 다음 명령을 사용하십시오.

```
# installadm set-server -z
```

매니페스트가 저장되면 파일이 `/var/ai/wizard-manifest/`에 저장됩니다. 그러면 `installadm` 명령을 실행하여 이 매니페스트를 설치 서비스에 연관시킬 수 있습니다.

## ▼ AI 매니페스트 마법사를 사용하여 AI 매니페스트를 만드는 방법

AI 서버에서 `/usr/bin/ai-wizard`를 실행하여 이 마법사를 시작할 수도 있습니다.

시작하기 전에 AI 매니페스트 마법사로 만든 매니페스트를 더 쉽게 추가하기 위해 생성된 매니페스트를 AI 서버의 임시 위치에 저장하도록 사용으로 설정할 수 있습니다. 자세한 내용은 [예 10-9. “AI 서버에 매니페스트 파일을 저장하도록 허용”](#)를 참조하십시오.

1. **AI 매니페스트 마법사를 시작합니다.**  
AI 서버의 URL을 사용하여 이 응용 프로그램에 연결할 수 있습니다. 기본적으로 이름이 `ai-server`인 AI 서버의 URL은 `http://ai-server.domain:5555`입니다.
2. **시작 화면에서 매니페스트에 연관시킬 서비스를 식별합니다.**  
구성된 모든 설치 서비스와 각 설치 서비스의 상태 및 클라이언트 아키텍처가 이 화면에 나열됩니다. 목록의 첫번째 항목은 AI 서버 자체와 관련이 있으며 항상 표시되므로 구성된 서비스가 없으면 이 항목을 선택하여 매니페스트를 만들 수 있습니다. AI 매니페스트와 연관시킬 설치 서비스를 선택하고 `Start`를 누릅니다.
3. **Introduction(소개) 화면에서 매니페스트 이름과 대상을 선택합니다.**  
AI 매니페스트 이름을 입력하거나 `default`를 사용하도록 선택합니다. AI 매니페스트가 전역 영역용인지 비전역 영역용인지 선택합니다. 이 마지막 값이 대상입니다. `Next`를 눌러 계속합니다.
4. **Root Pool(루트 풀) 화면에서 루트 풀에 대한 정보를 입력합니다.**  
이 화면에서 루트 풀 이름, 부트 환경 이름을 설정하고 루트 풀이 미러링되는지 여부를 선택하며 스왑 및 덤프 장치 구성 매개변수를 정의할 수 있습니다. `Next`를 눌러 계속합니다.
5. **Data Pools(데이터 풀) 화면에서 추가 ZFS 저장소 풀에 대한 정보를 입력합니다.**

데이터 풀을 5개까지 지정할 수 있습니다. 데이터 풀마다 풀 이름과 마운트 지점을 지정해야 합니다. 데이터 풀마다 None, Mirror, Raid-Z, Raid-Z1, Raid-Z2 또는 Raid-Z3 중복성 레벨을 선택할 수 있습니다. Next를 눌러 계속합니다.

**6. Disks(디스크) 화면에서 루트 및 저장소 풀의 디스크를 할당합니다.**

이전 화면에서 정의한 모든 풀이 이 화면에 나열됩니다. 풀마다 다음 중 하나를 구성할 수 있습니다.

- 디스크 키워드, Auto 또는 Boot Disk
- Device Size 또는 Device Type과 같은 디스크 등록 정보
- CTD Name 또는 Volume Id와 같은 디스크 이름

이 등록 정보를 한 번에 2개 이상 구성해야 할 경우 그 중 하나를 선택하여 마법사를 사용할 때 추가하고 매니페스트가 생성된 후 파일을 편집하거나 aimanifest 명령을 사용하여 등록 정보를 더 추가합니다. Next를 눌러 계속합니다.

**7. Repositories(저장소) 화면에서 IPS 저장소를 정의합니다.**

기본 Oracle Solaris 지원 및 릴리스 저장소가 자동으로 정의됩니다. 클라이언트에 추가할 패키지가 포함된 추가 저장소 5개 이내에 대한 정보를 입력할 수도 있습니다. 각 저장소에는 저장소 이름과 원본 URI가 필요합니다.

각 저장소에 대해 Add Details를 누른 후 SSL 인증서 파일, SSL 키 파일, 저장소의 백업 원본 URI를 지정할 수 있습니다. Next를 눌러 계속합니다.

**8. Software(소프트웨어) 화면에서 설치할 소프트웨어 패키지를 선택합니다.**

필요에 따라 큰 서버, 작은 서버 및 데스크탑 그룹을 설치하도록 선택할 수 있습니다. 또한 클라이언트가 생성되면 추가할 추가 또는 사용자 정의 패키지의 패키지 FMRI를 추가할 수 있습니다. Next를 눌러 계속합니다.

**9. Zones(영역) 화면에서 영역 이름을 정의하고 영역 구성 파일 URI를 정의합니다.**

클라이언트가 생성되면 추가할 영역마다 영역 이름과 영역 구성 파일 URI를 입력합니다. Next를 눌러 계속합니다.

**10. Review(검토) 화면에서 입력한 정보를 확인하고 save를 눌러 매니페스트를 만듭니다.**

서버측 저장을 허용하도록 SI 서버가 구성된 경우 매니페스트가 서버에 저장되며 로컬로 파일을 저장하는 옵션도 주어집니다. 그렇지 않으면 로컬로만 매니페스트를 저장할 수 있습니다.

## 예제 SI 매니페스트

이 절의 예제는 완성된 SI 매니페스트가 언급된 결과를 얻기 위해 갖춰야 할 XML 요소를 보여줍니다. 이러한 매니페스트는 XML을 직접 편집하거나 파생된 매니페스트 스크립트를 사용하여 만들 수 있습니다.

이 절에 나온 모든 매니페스트는 `/image-path/auto_install/manifest/default.xml`에 있는 샘플 기본 XML 파일을 기반으로 하며 필요에 따라 수정되었습니다. `software` 요소의 `destination` 요소는 간결하게 생략합니다.

## iSCSI 대상 장치 지정

이 예제에서 설치 대상은 iSCSI 장치입니다. `target` 요소의 `disk` 요소의 `iscsi` 요소를 사용합니다. `disk` 요소의 `whole_disk` 속성은 iSCSI 디스크의 표준 값인 `true`로 설정됩니다. `target_name`, `target_lun` 및 `target_ip` 속성 설명은 [ai\\_manifest\(4\)](#) 매뉴얼 페이지를 참조하십시오.

```
<auto_install>
  <ai_instance name="default">
    <target>
      <disk whole_disk="true">
        <iscsi target_name="iqn.1986-03.com.sun:02:1234567890abcdef" \
          target_lun="1" target_ip="129.158.144.200"/>
      </disk>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@0.5.11-0.175.2</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

## RAID 구성 지정

이 예제에서는 두 디스크 `c0t0d0` 및 `c0t1d0`을 사용하여 RAID 구성을 지정합니다. 이 매니페스트는 [예 10-3. "지정된 크기의 디스크가 적어도 두 개 있는 경우 미러링된 구성 지정"](#)에 표시된 미러링된 구성의 매니페스트와 비슷합니다. 두 매니페스트 간의 한 가지 차이점은

redundancy 속성 값이 mirror 대신 raidz라는 것입니다. 중복성 유형에 대한 자세한 내용은 zpool(1M) 매뉴얼 페이지를 참조하십시오. 또 다른 차이점은 rpool은 루트 풀을 암시하기 때문에 ZFS 풀 이름이 rpool로 지정되지 않는다는 것입니다. 이 예제에서 지정을 생략할 수 있도록 기본적으로 zpool 요소의 is\_root 속성 값은 false입니다. 지정된 루트 풀이 없으므로 이 설치에 대한 초기 사용자를 구성하지 않습니다.

```
<auto_install>
  <ai_instance name="default">
    <target>
      <disk in_vdev="raid_vdev" in_zpool="raidpool" whole_disk="true">
        <disk_name name="c0t0d0" name_type="ctd"/>
      </disk>
      <disk in_vdev="raid_vdev" in_zpool="raidpool" whole_disk="true">
        <disk_name name="c0t1d0" name_type="ctd"/>
      </disk>
      <logical>
        <zpool name="raidpool" is_root="false">
          <vdev name="raid_vdev" redundancy="raidz"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@0.5.11-0.175.2</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

## SVR4 패키지 설치

이 예제는 SVR4 패키지를 설치하는 방법을 보여줍니다. SVR4 패키지는 SVR4 유형의 software 요소에 이름이 지정되어야 합니다. 게시자 원본의 이름 속성 값은 SVR4 패키지 하위 디렉토리 또는 SVR4 패키지 데이터 스트림 파일을 포함하는 디렉토리입니다. SVR4 패키지 하위 디렉토리의 원본 이름은 전체 파일 디렉토리 경로 또는 파일 URI일 수 있습니다. SVR4 패키지 데이터 스트림 파일의 원본 이름은 전체 파일 디렉토리 경로, 파일 URI 또는 HTTP URI일 수 있습니다.

---

**작은 정보** - 설치의 일부로 사용자 입력이 필요한 패키지를 설치하지 마십시오.

---

```

<auto_install>
  <ai_instance name="default">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home" />
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@0.5.11-0.175.2</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
    <software type="SVR4">
      <source>
        <publisher>
          <origin name="/net/host2/usr/dist"/>
        </publisher>
      </source>
      <software_data>
        <name>SUNWpackage</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>

```

## 여러 SVR4 패키지 설치

여러 SVR4 패키지를 설치하려면 아래에 나온 대로 패키지마다 소프트웨어 태그를 지정해야 합니다.

```

<software type="SVR4">
  <source>
    <publisher>
      <origin name="/net/192.168.56.2/svr4/app1.pkg"/>
    </publisher>
  </source>
  <software_data>
    <name>application1</name>
  </software_data>
</software>
<software type="SVR4">

```

```
<source>
  <publisher>
    <origin name="/net/192.168.56.2/svr4/app2.pkg"/>
  </publisher>
</source>
<software_data>
  <name>application2</name>
</software_data>
</software>
```

## 기존의 디스크 슬라이드 또는 분할 영역 다시 사용

이 샘플에서는 SPARC 클라이언트에 기존 디스크 슬라이드를 사용하도록 지정하는 방법을 보여줍니다. 기존 슬라이드의 규격(start\_sector and size)이 디스크 슬라이드에 다시 사용 됩니다. 구성 프로세스에서는 이미 설치된 Solaris 버전이 있는지 확인하기 위해 슬라이드를 검색하지 않습니다.

```
<disk>
  <disk_name name="c1t0d0" name_type="ctd"/>
  <slice name="0" action="use_existing" force="true" in_zpool="rpool">
</disk>
```

다음 예제에서는 x86 클라이언트에 대해 AI 프로세스 중 디스크에 있는 기존 분할 영역이 다시 사용되어야 함을 지정하는 방법을 보여줍니다. 명명된 슬라이드의 기존 규격을 분할 영역에 다시 사용해야 합니다. 이 경우 다시 사용할 분할 영역이 구성 프로세스 중에 자동으로 확인됩니다.

```
<partition action="use_existing_solaris2">
  <slice action="use_existing" name="0" force="true"/>
</partition>
```

## 기본 AI 매니페스트

설치 서비스의 기본 AI 매니페스트는 파생된 매니페스트입니다. 설치 서비스를 만들 때 서비스에 대해 기본 매니페스트인 `orig_default`가 생성됩니다. 다른 설치 이미지의 경우 생성된 기본 매니페스트가 약간 다를 수 있습니다. 이 파일의 XML 콘텐츠에 대한 자세한 내용은 [ai\\_manifest\(4\)](#) 매뉴얼 페이지를 참조하십시오.

## 클라이언트 시스템 구성

---

이 장에서는 설치 후에 클라이언트 시스템 구성에 필요한 정보를 지정하는 방법을 설명합니다. SMF(서비스 관리 기능) 등록 정보를 사용하여 구성 가능한 모든 구성을 지정할 수 있습니다.

### 구성 프로파일 제공

시스템 구성 프로파일은 SMF 프로파일 형식의 구성 매개변수 세트로 클라이언트 시스템 구성을 지정합니다. 시스템 구성 프로파일은 적절한 SMF 서비스에 대해 SMF 등록 정보를 설정합니다.

AI 설치 후에 시스템의 처음 부트 중에 시스템 구성 프로파일이 적용됩니다. 특정 구성 영역을 담당하는 SMF 서비스가 SMF 등록 정보를 처리하고 이에 따라 시스템을 구성합니다.

각 클라이언트는 원하는 수의 시스템 구성 프로파일을 사용할 수 있습니다. 예를 들어, 클라이언트에 대한 호스트 이름 및 IP 주소를 제공하는 하나의 프로파일에 클라이언트를 지정할 수 있습니다. 더욱 광범위하게 적용 가능한 등록 정보 값을 설정하는 다른 프로파일에 동일한 클라이언트와 많은 다른 클라이언트를 지정할 수 있습니다.

특정 클라이언트에 대해 제공된 시스템 구성 프로파일이 없으면 설치 후 처음 부트할 때 해당 클라이언트에서 대화식 구성 도구가 열립니다. 구성 도구를 대화식으로 사용하는 방법에 대한 자세한 내용은 “[시스템 재구성](#)” [65]을 참조하십시오.

### 시스템 구성 프로파일 만들기

다음 방식 중 하나를 사용하여 시스템 구성 프로파일을 만듭니다.

- 대화식 구성 도구를 실행하고 출력을 파일에 저장합니다. 다음 명령은 대화식으로 입력한 응답으로부터 sc.xml에 유효한 프로파일을 만듭니다.

```
# sysconfig create-profile -o sc.xml
```

구성 도구를 사용하여 프로파일을 만드는 방법에 대한 자세한 내용은 “[SCI 도구를 사용하여 시스템 구성 프로파일 만들기](#)” [69]를 참조하십시오.

- 등록 정보 사양을 사용하여 시스템 구성 프로파일을 수동으로 만듭니다. “[시스템 구성 프로파일에 구성 지정](#)” [167] 및 “[예제 시스템 구성 프로파일](#)” [180]에 표시된 내용을 참조하십시오.

다음 라인을 모든 시스템 구성 프로파일에 포함합니다.

```
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- service, property_group, property, and propval specifications -->
</service_bundle>
```

적용되지 않는 서비스나 등록 정보를 지정하면 해당 사양이 무시됩니다.

특정 등록 정보를 한번 이상 지정하지 마십시오.

- 파생된 매니페스트 스크립트가 실행 중일 때 이 스크립트로 시스템 구성 프로파일을 만들 수 있습니다. 예 10-6. “[시스템 구성 프로파일 추가](#)”을 참조하십시오.

시스템 구성 프로파일은 두 가지 방법으로 등록 정보 및 속성 값을 표현할 수 있습니다. 하나의 프로파일이 양쪽 방식을 사용할 수 있습니다.

- 이 장에 표시된 등록 정보 사양을 사용하여 설치 서비스에 프로파일을 추가하기 전에 값을 명시적으로 입력할 수 있습니다.
- 시스템 구성 프로파일을 사용하여 클라이언트 시스템을 설치할 때 유효한 값으로 대체될 변수를 포함할 수 있습니다. “[시스템 구성 프로파일 템플릿 사용](#)” [178]을 참조하십시오.

## 시스템 구성 프로파일 검증

`installadm validate` 명령을 사용하여 구문 정확성을 위해 개발 중인 시스템 구성 프로파일을 검증합니다. 이 프로파일을 추가할 설치 서비스가 이미 존재해야 합니다. “[시스템 구성 프로파일 검증](#)” [131] 하위 명령에 대한 자세한 내용은 Validating a System Configuration Profile을 참조하십시오.

## 시스템 구성 프로파일을 설치 서비스에 추가

`installadm create-profile` 명령을 사용하여 시스템 구성 프로파일을 설치 서비스에 추가합니다. 프로파일을 설치 서비스에 추가하기 전에 `create-profile` 하위 명령이 프로파일을 검증합니다.

적절한 클라이언트가 해당 시스템 구성 프로파일을 선택하도록 조건을 지정합니다. 지정된 조건이 없는 경우 모든 클라이언트가 이 프로파일을 사용합니다.

단일 클라이언트가 일치하는 여러 개의 시스템 구성 프로파일을 사용할 수 있습니다. 특정 등록 정보가 한번 이상 지정된 프로파일 세트를 사용하는 클라이언트가 없는지 확인합니다. 클라이언트가 특정 등록 정보에 대해 여러 개의 사양을 받는 경우 등록 정보 값이 각 사양마다 같더라도 구성 중인 SMF 서비스의 동작이 정의되지 않습니다.

클라이언트가 설치 서비스의 시스템 구성 프로파일에 지정된 조건과 일치하지 않으면 해당 클라이언트에 대화식 구성 도구가 열립니다.

`installadm list` 명령을 사용하여 주어진 설치 서비스에 추가된 프로파일을 나열하고 각 프로파일에 지정된 조건을 나열합니다.

`installadm set-criteria` 명령을 사용하여 프로파일에 지정된 클라이언트 선택 조건을 변경하거나 추가할 수 있습니다.

`installadm export` 명령을 사용하여 설치 서비스에 추가된 프로파일 내용의 복사본을 검색합니다. 해당 복사본을 수정하여 다른 프로파일을 만들 수 있습니다.

`installadm update-profile` 명령을 사용하여 설치 서비스에 이미 추가된 프로파일의 내용을 바꿉니다.

`create-profile`, `update-profile`, `list`, `set-criteria` 및 `export` 하위 명령에 대한 자세한 내용은 “설치 서비스 작업” [96] 및 `installadm(1M)` 매뉴얼 페이지를 참조하십시오.

## 시스템 구성 프로파일에 구성 지정

SMF 등록 정보를 사용하여 구성 가능한 모든 시스템 특성의 구성을 지정할 수 있습니다. 예를 들어, 시스템 구성 프로파일은 루트 계정, 초기 사용자, 키보드 레이아웃, 터미널 유형, IPv4 네트워크 인터페이스(`static` 또는 `DHCP`)와 기본 경로, IPv6 네트워크 인터페이스(`static` 또는 `addrconf`)와 기본 경로, 그리고 이름 서비스(이름 서버 목록, 검색 목록, 도메인)를 구성할 수 있습니다. 적용되지 않는 서비스나 등록 정보를 지정하면 해당 사양이 무시됩니다. 특정 등록 정보를 한번 이상 지정하지 마십시오.

지정할 SMF 등록 정보가 무엇인지 확실히 모르면 `svccfg` 명령의 `describe` 하위 명령을 사용하여 서비스의 등록 정보 그룹 및 등록 정보에 대한 설명을 가능한 설정과 함께 표시할 수 있습니다. `svccfg(1M)` 매뉴얼 페이지의 "Property Inspection and Modification Subcommands"를 참조하십시오.

```
svccfg -s FMRI describe [-v] [-t] [propertygroup/property]
```

등록 정보 그룹 이름을 지정하거나 슬래시(/)로 구분된 등록 정보 그룹 이름 및 등록 정보 이름을 인수로 지정하여 등록 정보 그룹이나 특정 등록 정보를 질의할 수 있습니다.

`-v` 옵션은 현재 설정, 제약 조건 및 기타 가능한 설정 선택에 대한 설명을 포함하여 모든 사용 가능한 정보를 제공합니다.

-t 옵션은 선택한 템플릿 데이터만 보여주고([smf\\_template\(5\)](#) 매뉴얼 페이지 참조) 등록 정보 그룹 및 등록 정보에 대한 현재 설정을 표시하지 않습니다.

```
$ svccfg -s name-service/switch describe config
config                                application
    Name service switch configuration data as described in nsswitch.conf(4).
config/value_authorization            astring                                solaris.smf.value.name-service.switch
config/default                        astring                                files
    Default configuration database entry.
config/host                            astring                                "files dns mdns"
    Override configuration for host database lookups. (both IPv4 and IPv6 hosts)
config/printer                         astring                                "user files"
    Override configuration for printer database lookups.
$ svccfg -s name-service/switch describe -v config
config                                application
    name: config
    type: application
    required: true
    target: this
    description: Name service switch configuration data as described in nsswitch.conf(4).
config/value_authorization            astring                                solaris.smf.value.name-service.switch
config/default                        astring                                files
    type: astring
    required: true
    Default configuration database entry.
    visibility: readwrite
    minimum number of values: 1
    maximum number of values: 1
    value: files
...
$ svccfg -s name-service/switch describe -t config
name: config
type: application
    Name service switch configuration data as described in nsswitch.conf(4).
    name: default
    type: astring
    Default configuration database entry.
    name: host
    type: astring
    Override configuration for host database lookups. (both IPv4 and IPv6 hosts)
    name: password
    type: astring
    Override configuration for passwd database lookups. Also used with the shadow and
    user_attr databases.
    name: group
    type: astring
    Override configuration for group database lookups.
    name: network
    type: astring
    Override configuration for network database lookups.
...
$ svccfg -s system/config-user describe root_account
root_account                          application
root_account/expire                   astring
```

```

root_account/password          astring
root_account/read_authorization astring          solaris.smf.read.system-config
root_account/stability         astring          Evolving
root_account/type              astring
    
```

## 루트 및 사용자 계정 구성

다음 `sysconfig create-profile` 명령을 `users` 그룹과 함께 입력하여 루트 사용자 및 초기 사용자를 구성하는 유효한 프로파일을 생성합니다.

```
# sysconfig create-profile -g users -o sc_users.xml
```

`svc:/system/config-user` SMF 서비스는 사용자 및 루트 계정을 구성합니다. 이 서비스는 두 등록 정보 그룹을 인식합니다.

- `root_account` 등록 정보 그룹은 루트 계정을 구성하는 SMF 등록 정보를 포함합니다.
- `user_account` 등록 정보 그룹은 사용자 계정을 구성하는 SMF 등록 정보를 포함합니다.

---

**작은 정보** - Oracle Solaris OS용 암호화된 암호를 생성하는 한 가지 방법은 의도한 이름과 암호로 사용자를 만들고 사용자 레코드의 첫번째와 두번째 콜론 사이에 `/etc/shadow` 파일의 암호를 복사하고 해당 정보를 매니페스트의 `password` 값에 추가하는 것입니다.

---

## 루트 계정 구성

`root_account` 등록 정보 그룹은 다음 표에 나열된 등록 정보를 포함합니다.

표 11-1 `root_account` 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
<code>password</code>	<code>astring</code>	필수	암호화된 루트 암호입니다. 루트 암호를 제공하지 않으면 루트 암호가 비어 있습니다.
<code>type</code>	<code>astring</code>	선택	계정 유형: <code>normal</code> 또는 <code>role</code> . 기본값은 <code>normal</code> 입니다.
<code>expire</code>	<code>string</code>	선택	로그인 만료일. 0으로 설정하면 다음 로그인 시 루트 암호를 강제로 변경하도록 합니다.

예 11-1 만료된 암호로만 루트 계정 구성

```

<service name="system/config-user" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="root_account" type="application">
      <propval name="password" value="encrypted_password"/>
      <propval name="type" value="normal"/>
      <propval name="expire" value="0"/>
    </property_group>
  </instance>
    
```

</service>

## 사용자 계정 구성

이 절에는 다음과 같은 정보가 포함되어 있습니다.

- “자동 마운트에 종속되지 않고 사용자 계정 만들기” [170]
- “사용자 계정 등록 정보” [170]
- “다중 초기 사용자 구성” [171]

### 자동 마운트에 종속되지 않고 사용자 계정 만들기

기본적으로 초기 사용자 계정을 만들 때 홈 디렉토리는 자동 마운트로 관리되고 `/home/login` 디렉토리에서 액세스할 수 있습니다. 자동 마운트에 종속되지 않고 초기 사용자 계정을 만들려면 시스템 구성 프로파일에서 `user_account/autohome` 등록 정보를 빈 문자열("")로 설정합니다.

`user_account/autohome` 등록 정보를 빈 문자열로 설정하면 다음과 같은 효과가 있습니다.

- `/etc/passwd` 파일의 홈 디렉토리 항목이 `/home/login`이 아닌, 홈 ZFS 데이터 세트의 마운트 지점으로 설정됩니다. 홈 ZFS 데이터 세트의 기본 마운트 지점은 `/export/home/login`입니다.
- `/etc/auto_home` 파일에 추가된 매핑 항목은 없습니다.

### 사용자 계정 등록 정보

`user_account` 등록 정보 그룹은 다음 표에 나열된 등록 정보를 포함합니다.

표 11-2 user\_account 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
login	astring	required	사용자의 로그인입니다.
password	astring	required	암호화된 사용자 암호입니다.
description	astring	optional	대개 사용자의 전체 이름
shell	astring	optional	로그인 시 사용자의 셸로 사용된 프로그램의 전체 경로 이름입니다.
uid	count	optional	새 사용자의 UID. 기본 UID는 101입니다.
gid	count	optional	사용자의 기본 그룹 멤버십. 기본 GID는 10입니다.
type	astring	optional	계정 유형: <code>normal</code> 또는 <code>role</code> . 기본값은 <code>normal</code> 입니다.
profiles	astring	optional	<code>prof_attr(4)</code> 매뉴얼 페이지에 정의된 하나 이상의 심프로 구분된 실행 프로파일입니다.
roles	astring	optional	<code>user_attr(4)</code> 매뉴얼 페이지에 정의된 하나 이상의 심프로 구분된 역할입니다.

등록 정보	유형	필수	설명
sudoers	astring	optional	sudoers 파일에 login과 함께 추가된 항목입니다.
expire	astring	optional	로그인 만료일. 0으로 설정하면 다음 로그인 시 암호를 강제로 변경하도록 합니다.
home_zfs_dataset	astring	optional	사용자의 홈 디렉토리 ZFS 데이터 세트. 기본값은 <i>root_pool/export/home/login</i> 입니다.
home_mountpoint	astring	optional	사용자의 홈 디렉토리 마운트 지점. 기본값은 <i>/export/home/login</i> 입니다.
autohome	astring	optional	사용자의 자동 홈 디렉토리 마운트 지점. 구성된 사용자에 대해 <i>/etc/auto_home</i> 파일에 값이 입력됩니다. 기본값은 <i>localhost:/export/home/login</i> 입니다. <i>autohome</i> 등록 정보가 빈 문자열("")로 설정된 경우 자동 마운트에 종속되지 않고 사용자 계정이 만들어집니다.

## 다중 초기 사용자 구성

새로 설치된 시스템에 다중 사용자를 구성하려면 스크립트에서 `useradd` 명령을 사용하여 사용자를 지정합니다. 그런 다음 일회용 SMF 서비스를 사용하여 처음 부트 시 스크립트를 실행합니다. 지침은 [13장. 처음 부트 중 사용자 정의 스크립트 실행](#)을 참조하십시오.

## SSH 키 구성

`ssh_public_keys` 등록 정보 그룹에는 미리 생성된 ssh 키가 있습니다. 클라이언트가 구성될 때 이 키가 사용자 `$HOME/.ssh/authorized_keys` 파일에 작성됩니다.

### 예 11-2 SSH 키 구성

```
<property_group name="user_account" type="application">
  <...>
  <property type="astring" name="ssh_public_keys">
    <astring_list>
      <value_node value='[<options>] <key-type> <base64-encoding-key> [<comment>]'
        <value_node value='[<options>] <key-type> <base64-encoding-key> [<comment>]'
      </value_node>
    </astring_list>
  </property>
</property_group>
```

## 시스템 신원 설정

`sysconfig create-profile` 명령을 `identity` 그룹과 함께 사용하여 시스템 노드 이름을 구성하는 유효한 프로파일을 생성합니다.

```
# sysconfig create-profile -g identity -o sc_identity.xml
```

svc:/system/identity:node SMF 서비스는 시스템 호스트 이름을 설정합니다. node는 svc:/system/identity의 인스턴스입니다.

identity 등록 정보 그룹은 다음 표에 나열된 등록 정보를 포함합니다.

표 11-3 config 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
nodename	astring	선택	시스템 호스트 이름입니다. 기본값은 solaris입니다.
enable_mapping	boolean	선택	노드 이름 매핑을 사용 안함으로 설정하는 값입니다. 기본값은 true입니다.
loopback	astring	선택	루프백에 매핑된 호스트 이름. 기본값은 solaris입니다.

예 11-3 호스트 이름 구성

이 예는 시스템 호스트 이름을 solaris로 설정합니다.

```
<service name="system/identity" version="1" type="service">
  <instance name="node" enabled="true">
    <property_group name="config" type="application">
      <propval name="nodename" value="solaris"/>
    </property_group>
  </instance>
</service>
```

예 11-4 노드 이름 매핑 사용 안함

Oracle Solaris 11 OS 또는 Oracle Solaris 11 업데이트 릴리스를 설치할 때 기본적으로 시스템 노드 이름은 루프백에 매핑되거나 설치의 일부로 구성된 인터페이스의 IP 주소에 매핑 됩니다. 다음 예제에 표시된 대로 enable\_mapping 등록 정보를 false로 설정하여 이 기본 매핑을 사용 안함으로 설정할 수 있습니다.

```
<service name="system/identity" version="1" type="service">
  <instance name="node" enabled="true">
    <property_group name="config" type="application">
      <propval name="nodename" value="solaris"/>
      <propval name="enable_mapping" value="false"/>
    </property_group>
  </instance>
</service>
```

## 시간대 및 로케일 설정

sysconfig create-profile 명령을 location 그룹과 함께 사용하여 시간대 및 로케일을 구성하는 유효한 프로파일을 생성합니다.

```
# sysconfig create-profile -g location -o sc_location.xml
```

svc:/system/timezone SMF 서비스는 시스템의 시간대를 설정합니다.  
 timezone 등록 정보 그룹은 다음 표에 나열된 등록 정보를 포함합니다.

표 11-4 timezone 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
localtime	astring	선택	시스템 시간대입니다. 기본값은 UTC입니다.

예 11-5 시간대 구성

이 예는 시간대를 Central European Time/Prague, CZ로 설정합니다.

```
<service name='system/timezone' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='timezone'>
      <propval name='localtime' value='Europe/Prague' />
    </property_group>
  </instance>
</service>
```

svc:/system/environment:init SMF 서비스는 시스템의 로케일을 설정합니다.

environment 등록 정보 그룹은 다음 환경 변수를 정의할 수 있습니다. 환경 변수에 대한 자세한 내용은 [environ\(5\)](#) 매뉴얼 페이지를 참조하십시오.

표 11-5 environment 등록 정보 그룹의 등록 정보

환경 변수	유형	필수	기본값
LC_CTYPE	astring	선택	C
LC_NUMERIC	astring	선택	C
LC_TIME	astring	선택	C
LC_COLLATE	astring	선택	C
LC_MONETARY	astring	선택	C
LC_MESSAGES	astring	선택	C
LC_ALL	astring	선택	C
LANG	astring	선택	C

예 11-6 로케일 구성

이 예는 로케일을 Czech 언어(cs) 및 Czech Republic(cz)으로 설정합니다.

```
<service name='system/environment' version='1'>
  <instance name='init' enabled='true'>
    <property_group name='environment'>
      <propval name='LC_ALL' value='cs_CZ.UTF-8' />
    </property_group>
  </instance>
</service>
```

```
</instance>
</service>
```

## 터미널 유형 및 키보드 레이아웃 설정

### 예 11-7 단말기 유형 구성

svc:/system/console-login SMF 서비스는 터미널 유형을 구성합니다. 관련된 SMF 등록 정보의 정의는 [ttymon\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

이 예는 터미널 유형을 vt100으로 설정합니다.

```
<service name="system/console-login" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="ttymon" type="application">
      <propval name="terminal_type" value="vt100"/>
    </property_group>
  </instance>
</service>
```

### 예 11-8 키보드 레이아웃 구성

sysconfig create-profile 명령을 kbd\_layout 그룹과 함께 사용하여 키보드 레이아웃을 구성하는 유효한 프로파일을 생성합니다.

```
# sysconfig create-profile -g kbd_layout -o sc_kdb.xml
```

svc:/system/keymap SMF 서비스는 키보드 레이아웃을 구성합니다. 관련된 SMF 등록 정보의 정의는 [kbd\(1\)](#) 매뉴얼 페이지를 참조하십시오.

이 예는 키보드 레이아웃을 Czech로 설정합니다.

```
<service name='system/keymap' version='1' type='service'>
  <instance name='default' enabled='true'>
    <property_group name='keymap' type='system'>
      <propval name='layout' value='Czech' />
    </property_group>
  </instance>
</service>
```

## 네트워크 인터페이스 구성

sysconfig create-profile 명령을 network 그룹과 함께 사용하여 네트워크를 구성하는 유효한 프로파일을 생성합니다. 이 명령은 SCL 도구를 시작하며 인터페이스를 구성하는 데 필요한 정보를 묻습니다.

```
# sysconfig create-profile -g network -o sc_network.xml
```

svc:/network/install SMF 서비스는 초기 물리적 네트워크 인터페이스를 구성합니다. 이 서비스는 시스템 구성을 유발하지 않는 등록 정보 값을 사용하여 초기에 사용 안함으로 설정됩니다.

**참고** - 설치 대상이 iSCSI 장치인 경우 해당 설치의 어떤 시스템 구성 프로파일에도 네트워크 인터페이스를 구성하지 마십시오. iSCSI 부트의 경우 iSCSI 장치의 네트워크 인터페이스가 클라이언트 부트 프로세스에서 초기에 구성됩니다. 동일한 인터페이스를 다시 구성하면 네트워크의 network/install 서비스가 유지 관리 상태가 됩니다.

다중 네트워크 인터페이스를 구성하려면 스크립트에 구성을 지정하고 일회용 SMF 서비스를 사용하여 처음 부트 시 스크립트를 실행합니다. 지침 및 샘플 스크립트는 [13장. 처음 부트 중 사용자 정의 스크립트 실행](#)을 참조하십시오.

svc:/network/install 서비스는 여러 IPv4 및 IPv6 인터페이스를 지원하며 선택적으로 이 인터페이스에서 연결할 수 있는 기본 경로를 지원합니다. 이 서비스를 사용하여 IPv4 및 IPv6 인터페이스를 구성할 수 있습니다. 서비스에서 해당 등록 정보와 ipadm 명령을 사용하여 네트워크 인터페이스를 구성합니다. 마찬가지로 서비스에서는 해당 등록 정보와 route 명령을 사용하여 기본 경로를 정의합니다.

“정적 네트워크 구성 지정” [\[182\]](#)에서 예제를 참조하십시오.

install\_ipv4\_interface 등록 정보 그룹을 사용하면 인터페이스를 하나만 구성할 수 있지만 ipv4\_interface 등록 정보 그룹을 사용하면 여러 인터페이스를 구성할 수 있습니다. 두 IPv4 등록 정보 그룹에 모두 다음 표에 나온 등록 정보가 포함됩니다.

표 11-6 IPv4 네트워크 인터페이스에 대한 등록 정보 그룹 등록 정보

등록 정보	유형	필수	설명
name	astring	required	네트워크 인터페이스의 이름
address_type	astring	required	ipadm create-addr 하위 명령에 대한 -t 옵션을 구성하는 데 사용된 값입니다. 유효한 값은 static 또는 dhcp입니다.
static_address	net_address_v4	optional	address_type이 static인 경우에만 필요합니다. ipadm create_addr 하위 명령에 대한 로컬 주소를 구성하는 데 사용됩니다.
dhcp_wait	astring	optional	address_type이 dhcp인 경우에만 적용됩니다. 정의된 경우 이 등록 정보는 ipadm create-addr 하위 명령의 -w seconds(또는 forever) 부분을 구성하는 데 사용됩니다.
default_route	net_address_v4	optional	route명령을 사용하여 기본 경로를 정의하는 데 사용됩니다.

```
# /usr/sbin/route \
-p add default default-route \
-ifp ifname
```

ifname 값은 name 등록 정보의 인터페이스 이름 부분입니다.

`install_ipv6_interface` 등록 정보 그룹을 사용하면 인터페이스를 하나만 구성할 수 있지만 `ipv6_interface` 등록 정보 그룹을 사용하면 여러 인터페이스를 구성할 수 있습니다. IPv6 인터페이스에 대한 등록 정보 그룹에는 다음 표에 나온 등록 정보가 포함됩니다.

표 11-7 IPv6 네트워크 인터페이스에 대한 등록 정보 그룹 등록 정보

등록 정보	유형	필수	설명
<code>name</code>	<code>astring</code>	<code>required</code>	네트워크 인터페이스의 이름
<code>address_type</code>	<code>astring</code>	<code>required</code>	<code>ipadm create-addr</code> 하위 명령에 대한 <code>-t</code> 옵션을 구성하는 데 사용된 값입니다. 유효한 값은 <code>static</code> 또는 <code>addrconf</code> 입니다.
<code>static_address</code>	<code>net_address_v6</code>	<code>optional</code>	<code>address_type</code> 이 <code>static</code> 인 경우에만 필요합니다. <code>ipadm create-addr</code> 하위 명령에 대한 로컬 주소를 구성하는 데 사용됩니다.
<code>interface_id</code>	<code>net_address_v6</code>	<code>optional</code>	<code>address_type</code> 이 <code>addrconf</code> 인 경우에만 적용됩니다. <code>ipadm create-addr</code> 하위 명령의 <code>-i interface_id</code> 부분을 구성하는 데 사용됩니다.
<code>stateless</code>	<code>astring</code>	<code>optional</code>	<code>address_type</code> 이 <code>addrconf</code> 인 경우에만 적용됩니다. <code>ipadm create-addr</code> 하위 명령의 <code>-p stateless=yes no</code> 부분을 구성하는 데 사용됩니다.
<code>stateful</code>	<code>astring</code>	<code>optional</code>	<code>address_type</code> 이 <code>addrconf</code> 인 경우에만 적용됩니다. <code>ipadm create-addr</code> 하위 명령의 <code>-p stateful=yes no</code> 부분을 구성하는 데 사용됩니다.
<code>default_route</code>	<code>net_address_v6</code>	<code>optional</code>	<code>route</code> 명령을 사용하여 기본 경로를 정의하는 데 사용됩니다.  <pre># /usr/sbin/route \ -p add default default-route \ -ifp ifname</pre> <code>ifname</code> 값은 <code>name</code> 등록 정보의 인터페이스 이름 부분입니다.

## 이름 서비스 구성

`sysconfig create-profile` 명령을 `naming_services` 그룹과 함께 사용하여 DNS, NIS, LDAP 클라이언트 및 이름 서비스 스위치를 구성하는 유효한 프로파일을 생성합니다.

```
# sysconfig create-profile -g naming_services -o sc_ns.xml
```

`naming_services` 그룹에는 SMF 서비스 2개가 포함됩니다.

- `svc:/system/name-service/switch` 서비스는 이름 지정 서비스를 관리합니다.
- `svc:/network/dns/client` 서비스는 DNS 서비스를 관리합니다.

`svc:/system/name-service/switch` SMF 서비스는 이름 서비스 스위치를 구성합니다. 이 서비스는 시스템 구성을 유발하지 않는 등록 정보 값을 사용하여 초기에 사용 안함으로 설

정됩니다. “이름 서비스 구성 지정” [186]에서 예제를 참조하십시오. `svc:/system/name-service/switch` 서비스의 `config` 등록 정보 그룹에는 다음 표에 나온 등록 정보가 포함됩니다.

표 11-8 `svc:/system/name-service/switch` 등록 정보 그룹의 `config` 등록 정보

등록 정보	유형	필수	설명
default	astring	선택	기본 구성 데이터베이스 항목
bootparam	astring	선택	bootparams 데이터베이스의 구성 대체
ethers	astring	선택	ethers 데이터베이스의 구성 대체
group	astring	선택	group 데이터베이스의 구성 대체
host	astring	선택	host 데이터베이스의 구성 대체
netmask	astring	선택	netmask 데이터베이스의 구성 대체
network	astring	선택	network 데이터베이스의 구성 대체
password	astring	선택	passwd 데이터베이스의 구성 대체
protocol	astring	선택	protocol 데이터베이스의 구성 대체
rpc	astring	선택	rpc 데이터베이스의 구성 대체

`svc:/network/dns/client` 서비스는 DNS 클라이언트의 구성을 지원합니다. 서비스가 하나의 등록 정보 그룹 `config`를 정의합니다. 서비스에서 해당 등록 정보를 사용하여 DNS 서비스의 구성 정보를 구성합니다. “이름 서비스 구성 지정” [186]에서 예제를 참조하십시오.

`config` 등록 정보 그룹에는 다음 표에 나온 등록 정보가 포함됩니다.

표 11-9 `config` 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
domain	astring	optional	로컬 도메인 이름입니다. <code>resolv.conf</code> 에서 <code>domain</code> 지시어를 구성하는 데 사용됩니다.
nameserver	net_address_list	required	IPv4 및 IPv6 주소 목록. <code>resolv.conf</code> 에서 <code>nameserver</code> 지시어를 구성하는 데 사용됩니다.
search	astring_list	optional	호스트 이름 조회용 검색 목록을 위한 도메인 값 목록. <code>resolv.conf</code> 에서 <code>search</code> 지시어를 구성하는 데 사용됩니다.

## Kerberos 구성

`kclient` 명령으로 클라이언트의 Kerberos 구성 정보가 포함된 시스템 구성 프로파일을 만들어야 합니다. 프로파일을 볼 수 있지만 직접 파일을 편집하지는 마십시오. 자세한 내용은 [AI를 사용한 Kerberos 클라이언트 구성 방법 \[117\]](#)을 참조하십시오.

## Oracle Configuration Manager 및 Oracle Auto Service Request 설정

Oracle Configuration Manager는 My Oracle Support로 시스템 구성을 기록할 수 있고, Oracle Auto Service Request는 특정 하드웨어 결함에 대한 서비스 요청을 자동으로 생성할 수 있습니다.

sysconfig create-profile 명령을 support 그룹과 함께 사용하여 Oracle Configuration Manager 및 Oracle Auto Service Request를 구성하는 유효한 프로파일을 생성합니다.

```
# sysconfig create-profile -g support -o sc_support.xml
```

출력 프로파일은 첫번째 등록 단계를 설정하며, 다음 조건과 일치하는 모든 클라이언트에 대해 동일합니다.

- 시스템이 동일한 My Oracle Support 자격 증명을 등록에 사용합니다. 이 프로파일을 사용하는 모든 클라이언트 시스템은 동일한 방법으로 My Oracle Support에 등록합니다. 이러한 클라이언트의 모든 데이터는 동일한 My Oracle Support 계정과 연관됩니다.
- 시스템이 동일한 네트워크 구성을 통해 My Oracle Support에 액세스합니다. 이 프로파일을 사용하는 모든 클라이언트 시스템은 동일한 프록시 서버와 집계 허브를 통해 My Oracle Support에 액세스합니다.

SI 클라이언트 시스템의 여러 그룹에 대해 추가 프로파일을 만들려면 기존 프로파일을 복사하고 편집할 것이 아니라, sysconfig create-profile 명령을 재실행해야 합니다. 프록시 서버에 사용자 이름과 암호가 있을 경우 암호가 암호화되어 있으므로 sysconfig create-profile을 재실행해야 합니다.

## 시스템 구성 프로파일 템플릿 사용

프로파일은 설치 프로세스 중 클라이언트 설치 환경의 값으로 대체될 변수를 포함할 수 있습니다. 이렇게 하면 단일 프로파일 파일이 서로 다른 클라이언트에 서로 다른 구성 매개변수를 설정할 수 있습니다. 사용할 수 있는 변수 목록은 표 11-10. “시스템 구성 템플릿 프로파일의 변수”를 참조하십시오.

다음 예제 프로파일 hostIPnet.xml에서 AI\_HOSTNAME은 클라이언트 시스템의 호스트 이름에 대한 위치 표시자이고, AI\_IPV4는 클라이언트 시스템의 IP 주소에 대한 위치 표시자입니다.

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service name="system/identity" version="1" type="service">
    <instance name="node" enabled="true">
      <property_group name="config" type="application">
        <propval name="nodename" value="{{AI_HOSTNAME}}"/>
      </property_group>
      <property_group name="install_ipv4_interface" type="application">
```

```

    <propval name="name" value="net0/v4"/>
    <propval name="address_type" value="static"/>
    <propval name="static_address" type="net_address_v4" value="{{AI_IPV4}}/8"/>
    <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
  </property_group>
</instance>
</service>
</service_bundle>

```

다음 명령은 입력 `hostandIP.xml` 파일을 변경하지 않고 각 설치 클라이언트에 대해 설치 서비스에서 사용자 정의할 시스템 구성 프로파일을 만듭니다.

```
# installadm create-profile -n solaris11_2-i386 -f /export/hostIPnet.xml
```

`hostandIP.xml` 파일이 변경되지 않은 채, 클라이언트에 적용된 프로파일이 사용자 정의됩니다. 예를 들어, 호스트 이름 `server1`의 클라이언트를 설치할 때 `hostandIP.xml` 프로파일의 내용은 다음과 같습니다.

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service name="system/identity" version="1" type="service">
    <instance name="node" enabled="true">
      <property_group name="config" type="application">
        <propval name="nodename" value="server1"/>
      </property_group>
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" value="net0/v4"/>
        <propval name="address_type" value="static"/>
        <propval name="static_address" type="net_address_v4" value="10.0.0.2/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
      </property_group>
    </instance>
  </service>
</service_bundle>

```

다음 표는 템플릿 프로파일에서 위치 표시자로 사용할 수 있는 변수를 보여줍니다.

**참고** - 영역 프로파일에서는 프로파일 템플릿 변수가 지원되지 않습니다.

표 11-10 시스템 구성 템플릿 프로파일의 변수

변수 이름	설명
AI_ARCH	uname -m의 커널 아키텍처
AI_CPU	uname -p의 프로세서 유형
AI_HOSTNAME	클라이언트 DNS 이름
AI_IPV4	IP 버전 4 네트워크 주소
AI_IPV4_PREFIXLEN	IPv4 네트워크 주소의 접두어 길이
AI_MAC	16진수 MAC 주소를 콜론(:) 구분자로 표시
AI_MEM	prtconf로 반환된 MB 단위의 메모리 크기

변수 이름	설명
AI_NETLINK_DEVICE	네트워크 인터페이스 물리적 장치의 이름
AI_NETLINK_VANITY	네트워크 인터페이스의 기본 배니티 이름
AI_NETWORK	IP 버전 4 네트워크 식별자
AI_ROUTER	클라이언트의 기본 라우터에 대한 IP 버전 4 네트워크 주소

## 예제 시스템 구성 프로파일

이 절의 예제는 `installadm create-profile` 명령을 사용하여 설치 서비스에 추가할 수 있는 완전한 시스템 구성 프로파일입니다.

## 샘플 시스템 구성 프로파일

이 섹션은 수정할 기초로 사용할 수 있는 샘플 시스템 구성 프로파일을 보여줍니다. 이 샘플은 `/usr/share/auto_install/sc_profiles/sc_sample.xml`에서 사용할 수 있습니다. 설치 서비스를 만든 후 `image-path/auto_install/sc_profiles/sc_sample.xml`에서 이 샘플 프로파일을 사용할 수 있습니다.

```
<?xml version="1.0"?>
<!--
Copyright (c) 2011, 2012, Oracle and/or its affiliates. All rights reserved.
-->

<!--
Sample system configuration profile for use with Automated Installer

Configures the following:
* User account name 'jack', password 'jack', GID 10, UID 101, root role, bash shell
* 'root' role with password 'solaris'
* Keyboard mappings set to US-English
* Time zone set to UTC
* Network configuration is automated with Network Auto-magic
* DNS name service client is enabled

See the installadm(1M) man page for usage of 'create-profile' subcommand.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="system configuration">
  <service name="system/config-user" version="1">
    <instance name="default" enabled="true">
      <property_group name="user_account">
        <propval name="login" value="jack"/>
        <propval name="password" value="9Nd/cwBcNWFZg"/>
        <propval name="description" value="default_user"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

```

        <propval name="shell" value="/usr/bin/bash"/>
        <propval name="gid" value="10"/>
        <propval name="uid" value="101"/>
        <propval name="type" value="normal"/>
        <propval name="roles" value="root"/>
        <propval name="profiles" value="System Administrator"/>
    </property_group>
    <property_group name="root_account">
        <propval name="password" value="encrypted_password"/>
        <propval name="type" value="role"/>
    </property_group>
</instance>
</service>

<service version="1" name="system/identity">
    <instance enabled="true" name="node">
        <property_group name="config">
            <propval name="nodename" value="solaris"/>
        </property_group>
    </instance>
</service>

<service name="system/console-login" version="1">
    <instance name="default" enabled="true">
        <property_group name="ttymon">
            <propval name="terminal_type" value="sun"/>
        </property_group>
    </instance>
</service>

<service name="system/keymap" version="1">
    <instance name="default" enabled="true">
        <property_group name="keymap">
            <propval name="layout" value="US-English"/>
        </property_group>
    </instance>
</service>

<service name="system/timezone" version="1">
    <instance name="default" enabled="true">
        <property_group name="timezone">
            <propval name="localtime" value="UTC"/>
        </property_group>
    </instance>
</service>

<service name="system/environment" version="1">
    <instance name="init" enabled="true">
        <property_group name="environment">
            <propval name="LANG" value="en_US.UTF-8"/>
        </property_group>
    </instance>
</service>

```

```
<service name="network/physical" version="1">
  <instance name="default" enabled="true">
    <property_group name="netcfg" type="application">
      <propval name="active_ncp" type="astring" value="Automatic"/>
    </property_group>
  </instance>
</service>
</service_bundle>
```

## 정적 네트워크 구성 지정

이 샘플 프로파일의 버전은 /usr/share/auto\_install/sc\_profiles/static\_network.xml에서 사용할 수 있습니다. 아래 표시된 이 프로파일의 버전을 수정하여 다음 매개변수를 구성합니다.

- bge0에 IPv4 정적 주소 10.0.0.10 및 넷마스크 255.0.0.0 사용
- 10.0.0.1 IPv4 기본 경로
- bge1에 IPv6 addrconf 주소 유형 사용
- DNS 8.8.8.8 이름 서버
- example1.com 및 example2.com을 호스트 이름 조회용 DNS 검색 목록으로 사용

넷마스크는 *IPaddress/netmask* 표기법으로 지정됩니다. 여기서 *netmask*는 넷마스크의 상위 비트 수를 지정하는 숫자입니다.

<i>netmask</i> 의 값	넷마스크 예
8	255.0.0.0
16	255.255.0.0
24	255.255.255.0

```
<?xml version="1.0"?>
<!--
Copyright (c) 2010, 2011, Oracle and/or its affiliates. All rights reserved.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="system configuration">
  <service name="system/config-user" version="1">
    <instance name="default" enabled="true">
      <property_group name="user_account">
        <propval name="login" value="jack"/>
        <propval name="password" value="9Nd/cwBcNWFZg"/>
        <propval name="description" value="default_user"/>
        <propval name="shell" value="/usr/bin/bash"/>
        <propval name="gid" value="10"/>
        <propval name="type" value="normal"/>
        <propval name="roles" value="root"/>
        <propval name="profiles" value="System Administrator"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

```

    </property_group>
    <property_group name="root_account">
      <propval name="password" value="$5$dnRfcZse
$Hx4aBQ161Uvn9ZxJFKMdRiy8tCf4gMT2s2rtkFba2y4"/>
      <propval name="type" value="role"/>
    </property_group>
  </instance>
</service>

<service version="1" name="system/identity">
  <instance enabled="true" name="node">
    <property_group name="config">
      <propval name="nodename" value="solaris"/>
    </property_group>
  </instance>
</service>

<service name="system/console-login" version="1">
  <instance name="default" enabled="true">
    <property_group name="ttymon">
      <propval name="terminal_type" value="sun"/>
    </property_group>
  </instance>
</service>

<service name="system/keymap" version="1">
  <instance name="default" enabled="true">
    <property_group name="keymap">
      <propval name="layout" value="US-English"/>
    </property_group>
  </instance>
</service>

<service name="system/timezone" version="1">
  <instance name="default" enabled="true">
    <property_group name="timezone">
      <propval name="localtime" value="UTC"/>
    </property_group>
  </instance>
</service>

<service name="system/environment" version="1">
  <instance name="init" enabled="true">
    <property_group name="environment">
      <propval name="LANG" value="en_US.UTF-8"/>
    </property_group>
  </instance>
</service>

<service name="network/physical" version="1">
  <instance name="default" enabled="true">
    <property_group name="netcfg" type="application">
      <propval name="active_ncp" type="astring" value="DefaultFixed"/>
    </property_group>
  </instance>
</service>

```

```

        </instance>
    </service>

    <service name="network/install" version="1" type="service">
        <instance name="default" enabled="true">
            <property_group name="install_ipv4_interface" type="application">
                <propval name="name" type="astring" value="bge0/v4"/>
                <propval name="address_type" type="astring" value="static"/>
                <propval name="static_address" type="net_address_v4" value="10.0.0.10/8"/>
                <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
            </property_group>

            <property_group name="install_ipv6_interface" type="application">
                <propval name="name" type="astring" value="bge1/v6"/>
                <propval name="address_type" type="astring" value="addrconf"/>
                <propval name="stateless" type="astring" value="yes"/>
                <propval name="stateful" type="astring" value="yes"/>
            </property_group>
        </instance>
    </service>

    <service name="network/dns/client" version="1">
        <property_group name="config">
            <property name="nameserver">
                <net_address_list>
                    <value_node value="8.8.8.8"/>
                </net_address_list>
            </property>
            <property name="search">
                <astring_list>
                    <value_node value="example1.com example2.com"/>
                </astring_list>
            </property>
        </property_group>
        <instance name="default" enabled="true"/>
    </service>

    <service version="1" name="system/name-service/switch">
        <property_group name="config">
            <propval name="default" value="files"/>
            <propval name="host" value="files dns mdns"/>
            <propval name="printer" value="user files"/>
        </property_group>
        <instance enabled="true" name="default"/>
    </service>

    <service version="1" name="system/name-service/cache">
        <instance enabled="true" name="default"/>
    </service>
</service_bundle>

```

## 여러 IPv4 인터페이스 구성

이 예제에서는 다중 인터페이스를 구성할 수 있게 해주는 `ipv4_interface` 등록 정보 그룹을 사용합니다. `install_ipv4_interface` 정의에서 변경되는 라인은 등록 정보 그룹 이름 및 유형을 정의하는 라인뿐입니다. `ipv6_interface` 등록 정보 그룹을 사용할 수도 있습니다.

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
  <service_bundle type="profile" name="sysconfig">
    <service version="1" type="service" name="network/install">
      <instance enabled="true" name="default">
        <property_group type="ipv4_interface" name="install_ipv4_interface_0">
          <propval type="net_address_v4" name="static_address" value="10.0.0.10/8"/>
          <propval type="astring" name="name" value="net0/v4"/>
          <propval type="astring" name="address_type" value="static"/>
        </property_group>
        <property_group type="ipv4_interface" name="install_ipv4_interface_1">
          <propval type="net_address_v4" name="static_address" value="10.0.0.11/8"/>
          <propval type="astring" name="name" value="net1/v4"/>
          <propval type="astring" name="address_type" value="static"/>
        </property_group>
      </instance>
    </service>
  </service_bundle>
  <service name="network/install" version="1" type="service">
    <instance name="default" enabled="true">
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" type="astring" value="bge0/v4"/>
        <propval name="address_type" type="astring" value="static"/>
        <propval name="static_address" type="net_address_v4" value="10.0.0.10/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
      </property_group>
```

## 사용자 SSH 키 추가

이 예제에서는 자동 설치 세션 중에 `ssh_public_keys`를 사용하여 사용자의 ssh 키를 추가하는 방법을 보여줍니다. 명명된 사용자의 `$HOME/.ssh/authorized_keys`에 각 키가 추가됩니다.

```
<property_group type="application" name="user_account">
  <...>
    <property type="astring" name="ssh_public_keys">
      <astring_list>
        <value_node value='[<options>] <key-type>
<base64-encoding-key> [<comment>]' />
        <value_node value='[<options>] <key-type>
<base64-encoding-key> [<comment>]' />
      </astring_list>
```

```

    </property>
  </property_group>

```

## 이름 서비스 구성 지정

이 섹션의 샘플 프로파일을 템플릿으로 사용하여 고유의 프로파일을 만들거나, `sysconfig` 도구를 `naming_services` 그룹과 함께 사용하여 프롬프트할 응답에 따라 프로파일을 만들 수 있습니다. `sysconfig`를 사용하여 시스템 구성 파일을 만드는 방법은 “[SCI 도구를 사용하여 시스템 구성 프로파일 만들기](#)” [69] 및 [sysconfig\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 이름 서비스 NIS 구성

**예 11-9** 지정된 도메인에 대해 NIS 사용

이 예제 프로파일은 다음 구성을 수행합니다.

- `my.domain.com`에 대해 NIS를 사용으로 설정합니다.
- NIS 서버를 검색하는 브로드캐스팅을 사용합니다(동일한 서브넷에 있어야 함).
- 이름 서비스 캐시 서비스를 사용으로 설정합니다(필수).

```

<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
-->
<service_bundle type='profile' name='default'>
  <service name='network/nis/domain' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='domainname' type='hostname' value='my.domain.com' />
    </property_group>
  </service>
  <instance name='default' enabled='true' />
  </service>
  <service name='network/nis/client' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='use_broadcast' type='boolean' value='true' />
    </property_group>
  </service>
  <instance name='default' enabled='true' />
  </service>
  <service name='system/name-service/switch' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='default' type='astring' value='files nis' />
      <propval name='printer' type='astring' value='user files nis' />
      <propval name='netgroup' type='astring' value='nis' />
    </property_group>
  </service>

```

```

</property_group>
<instance name='default' enabled='true' />
  </service>
  <service name='system/name-service/cache' type='service' version='1'>
<instance name='default' enabled='true' />
  </service>
</service_bundle>

```

#### 예 11-10 NIS 구성 및 DNS 사용 안함

이 예제 프로파일은 다음 구성을 수행합니다.

- NIS 서버에 대한 자동 브로드캐스팅으로 이름 서비스 NIS를 구성합니다(동일한 서브넷에 있어야 함).
- NIS 도메인 my.domain.com을 구성합니다.
- 이름 서비스 캐시 서비스를 사용으로 설정합니다(필수).
- DNS 이름 서비스를 사용 안함으로 설정합니다.

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- service name-service/switch below for NIS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="printer" value="user files nis"/>
      <propval type="astring" name="netgroup" value="nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- service name-service/cache must be present along with name-service/switch -->
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <!-- if no DNS, must be explicitly disabled to avoid error msgs -->
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
      <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- configure the NIS client service to broadcast the subnet for a NIS server -->
  <service version="1" type="service" name="network/nis/client">
    <property_group type="application" name="config">
      <propval type="boolean" name="use_broadcast" value="true"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
</service_bundle>

```

## 예 11-11 NIS 구성

다음 프로파일은 서버 IP 주소 10.0.0.10 및 도메인 mydomain.com으로 이름 서비스 NIS를 구성합니다. 서버 IP 주소가 명시적으로 지정된 경우 NIS 서버가 동일한 서브넷에 있을 필요가 없습니다.

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- name-service/switch below for NIS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="printer" value="user files nis"/>
      <propval type="astring" name="netgroup" value="nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- name-service/cache must be present along with name-service/switch -->
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <!-- if no DNS, must be explicitly disabled to avoid error msgs -->
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
      <propval type="hostname" name="domainname" value="mydomain.com"/>
      <!-- Note: use property with net_address_list and value_node as below -->
      <property type="net_address" name="ypservers">
        <net_address_list>
          <value_node value="10.0.0.10"/>
        </net_address_list>
      </property>
    </property_group>
    <!-- configure default instance separate from property_group -->
    <instance enabled="true" name="default"/>
  </service>
  <!-- enable the NIS client service -->
  <service version="1" type="service" name="network/nis/client">
    <instance enabled="true" name="default"/>
  </service>
</service_bundle>
```

## 예 11-12 지정된 도메인에 대해 NIS 및 DNS 사용

이 예는 DNS 및 NIS 이름 서비스를 모두 구성합니다.

- 다중 DNS 이름 서버를 지정합니다.
- DNS 도메인 검색 목록을 지정합니다.
- NIS 도메인을 지정합니다.

- NIS 서버를 검색하는 브로드캐스팅을 지정합니다.

```

<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
  Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
-->
<service_bundle type='profile' name='default'>
  <service name='network/dns/client' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='domain' type='astring' value='us.oracle.com' />
      <property name='nameserver' type='net_address'>
        <net_address_list>
          <value_node value='130.35.249.52' />
          <value_node value='130.35.249.41' />
          <value_node value='130.35.202.15' />
        </net_address_list>
      </property>
      <property name='search' type='astring'>
        <astring_list>
          <value_node value='us.oracle.com oracle.com oraclecorp.com' />
        </astring_list>
      </property>
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='network/nis/domain' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='domainname' type='hostname' value='mydomain.com' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='network/nis/client' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='use_broadcast' type='boolean' value='true' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='system/name-service/switch' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='default' type='astring' value='files nis' />
      <propval name='host' type='astring' value='files dns' />
      <propval name='printer' type='astring' value='user files nis' />
      <propval name='netgroup' type='astring' value='nis' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='system/name-service/cache' type='service' version='1'>
    <instance name='default' enabled='true' />
  </service>
</service_bundle>

```

## 이름 서비스 DNS 구성

예 11-13          검색 목록으로 DNS 구성

다음 예제 프로파일은 다음 매개변수를 구성합니다.

- 이름 서비스 DNS
- 서버 IP 주소 1.1.1.1 및 2.2.2.2
- 도메인 dom.ain.com

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- name-service/switch below for DNS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- name-service/cache must be present along with name-service/switch -->
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <property_group type="application" name="config">
      <!-- Note: use property with net_address_list and value_node as below -->
      <property type="net_address" name="nameserver">
        <net_address_list>
          <value_node value="1.1.1.1"/>
          <value_node value="2.2.2.2"/>
        </net_address_list>
      </property>
      <!-- Note: use property with astring_list and value_node,
        concatenating search names, as below -->
      <property type="astring" name="search">
        <astring_list>
          <value_node value="dom.ain.com ain.com"/>
        </astring_list>
      </property>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
</service_bundle>
```

## 이름 서비스 LDAP 구성

예 11-14 LDAP 및 LDAP 검색 기준 구성

이 예제 프로파일은 다음 매개변수를 구성합니다.

- 이름 서비스 LDAP에 서버 IP 주소 10.0.0.10 사용
- system/nis/domain 서비스에 지정된 도메인 my.domain.com
- LDAP 검색 기준(필수), dc=my,dc=domain,dc=com

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="printer" value="user files ldap"/>
      <propval type="astring" name="netgroup" value="ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
  <service version="1" type="service" name="network/ldap/client">
    <property_group type="application" name="config">
      <propval type="astring" name="profile" value="default"/>
      <property type="host" name="server_list">
        <host_list>
          <value_node value="10.0.0.10"/>
        </host_list>
      </property>
      <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
      <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
</service_bundle>
```

예 11-15 보안 LDAP 서버로 LDAP 구성

이 예제 프로파일은 다음 매개변수를 구성합니다.

- 이름 서비스 LDAP에 서버 IP 주소 10.0.0.10 사용
- system/nis/domain 서비스에 지정된 도메인 my.domain.com
- LDAP 검색 기준(필수), dc=my,dc=domain,dc=com
- LDAP 프록시 바인드 고유 이름 cn=proxyagent,ou=profile,dc=my,dc=domain,dc=com
- LDAP 프록시 바인드 암호, 보안 조치로 암호화됨. 다음 방식 중 하나를 사용하여 암호화된 값을 찾을 수 있습니다.
  - sysconfig create-profile에서 bind\_passwd 등록 정보 값을 가져옵니다.
  - LDAP 서버의 SMF 구성에서 값을 가져옵니다.

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="printer" value="user files ldap"/>
      <propval type="astring" name="netgroup" value="ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
  <service version="1" type="service" name="network/ldap/client">
    <property_group type="application" name="config">
      <propval type="astring" name="profile" value="default"/>
      <property type="host" name="server_list">
        <host_list>
          <value_node value="10.0.0.10"/>
        </host_list>
      </property>
      <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>
    </property_group>
    <property_group type="application" name="cred">
      <propval type="astring" name="bind_dn"
value="cn=proxyagent,ou=profile,dc=my,dc=domain,dc=com"/>
      <!-- note that the password below is encrypted -->
      <propval type="astring" name="bind_passwd" value="{NS1}c2ab873ae7c5ceefa4b9"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
      <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
</service_bundle>
```

## DNS를 LDAP와 함께 사용

DNS 이름 서비스를 LDAP 이름 서비스와 함께 사용할 수 있습니다. 전형적인 사용법은, DNS로 노드 이름(LDAP 서버 이름 포함)을 분석하고 LDAP로 기타 다른 이름을 분석하는 것입니다. 이 예의 첫번째 service 요소에 표시된 대로, system/name-service/switch 서비스에서 노드 이름 검색에 DNS를 지정하고 LDAP로 다른 이름을 분석할 수 있습니다.

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <property_group type="application" name="config">
      <property type="net_address" name="nameserver">
        <net_address_list>
          <value_node value="10.0.0.10"/>
        </net_address_list>
      </property>
      <propval type="astring" name="domain" value="my.domain.com"/>
      <property type="astring" name="search">
        <astring_list>
          <value_node value="my.domain.com"/>
        </astring_list>
      </property>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/ldap/client">
    <property_group type="application" name="config">
      <propval type="astring" name="profile" value="default"/>
      <property type="host" name="server_list">
        <host_list>
          <!-- here, DNS is expected to resolve the LDAP server by name -->
          <value_node value="ldapservers.my.domain.com"/>
        </host_list>
      </property>
      <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
```

```

        <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

## NIS를 DNS와 함께 사용

비슷한 방법으로 NIS를 DNS와 함께 사용할 수 있습니다.

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <property_group type="application" name="config">
      <property type="net_address" name="nameserver">
        <net_address_list>
          <value_node value="10.0.0.10"/>
        </net_address_list>
      </property>
      <propval type="astring" name="domain" value="my.domain.com"/>
      <property type="astring" name="search">
        <astring_list>
          <value_node value="my.domain.com"/>
        </astring_list>
      </property>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
      <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/client">
    <property_group type="application" name="config">
      <propval type="boolean" name="use_broadcast" value="true"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>

```

```
</service_bundle>
```



## 영역 설치 및 구성

---

이 장에서는 SI 클라이언트 설치의 일부로 비전역 영역의 설치 및 구성을 지정하는 방법을 설명합니다.

### SI의 비전역 영역 설치 방법

비전역 영역은 전역 영역이 설치된 후에 처음 재부트 시 설치 및 구성됩니다.

1. SI를 사용하여 시스템을 설치할 때 SI 매니페스트의 configuration 요소를 사용하여 해당 시스템에 비전역 영역을 설치할 수 있습니다. configuration 요소에 대한 자세한 내용은 “전역 영역 SI 매니페스트에서 비전역 영역 지정” [198]을 참조하십시오.
2. 시스템이 전역 영역 설치 후에 처음 부트할 때 영역 셀프 어셈블리 SMF 서비스(svc:/system/zones-install:default)가 전역 영역 SI 매니페스트에 정의된 각 비전역 영역을 구성 및 설치합니다. 비전역 영역 설치에 사용된 데이터에 대한 내용은 “비전역 영역 구성 및 설치 데이터” [198]를 참조하십시오.
3. 영역이 autoboot=true로 구성된 경우 system/zones-install 서비스가 설치 후에 영역을 부트합니다.

system/zones-install 서비스는 온라인을 유지하지만 다시 시작 전까지 새 구성 정보를 처리하지 않습니다. system/zones-install 서비스를 사용 또는 사용 안함으로 설정하면 안됩니다. 이 서비스를 무조건 다시 시작해야 합니다.

비전역 영역 설치를 모니터링하려면 system/zones-install 서비스나 zoneadm list -cv의 출력을 모니터링합니다.

다음 오류가 발생할 경우 영역이 설치되지 않습니다.

- 영역 config 파일이 구문상 정확하지 않습니다
- 설치될 영역 세트에서 영역 이름, 영역 경로 또는 위임된 ZFS 데이터 세트 사이에 충돌이 발생합니다.
- 필요한 데이터 세트가 전역 영역에 구성되지 않았습니다

## 전역 영역 AI 매니페스트에서 비전역 영역 지정

클라이언트 시스템에 대한 AI 매니페스트의 configuration 요소를 사용하여 비전역 영역을 지정합니다. configuration 요소의 name 속성을 사용하여 영역 이름을 지정합니다. source 속성을 사용하여 영역에 대한 config 파일의 위치를 지정합니다. source 위치는 클라이언트가 설치 중 액세스할 수 있는 http:// 또는 file:// 위치일 수 있습니다.

다음 샘플 AI 매니페스트는 두 개의 비전역 영역을 지정합니다.

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance>
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@latest</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>

    <configuration type="zone" name="zone1" source="http://server/zone1/config"/>
    <configuration type="zone" name="zone2" source="file:///net/server/zone2/config"/>

  </ai_instance>
</auto_install>
```

## 비전역 영역 구성 및 설치 데이터

다음 파일이 비전역 영역의 구성 및 설치에 사용됩니다.

config 파일	필수. config 파일은 zonecfg export 명령의 출력에서 가져온 파일 형식의 영역 구성입니다.  config 파일의 위치는 AI 매니페스트의 configuration 요소의 source 속성으로 지정됩니다. AI가 이 config 파일을 영역 구성에 사용할 설치된 클라이언트 시스템으로 복사합니다.
-----------	---

AI 매니페스트	<p>선택 사항. 이 영역 설치용 AI 매니페스트는 영역에 설치될 패키지를 게시자 정보 및 인증서/키 파일과 함께 지정합니다. 영역용 사용자 정의 AI 매니페스트 만들기에 대한 내용은 <a href="#">“비전역 영역 AI 매니페스트” [200]</a>를 참조하십시오.</p> <p>영역용 사용자 정의 AI 매니페스트를 제공하려면 전역 영역을 설치 중인 설치 서비스에 매니페스트를 추가합니다. create-manifest 명령에서 이 AI 매니페스트를 사용할 모든 영역의 이름과 함께 zonename 조건 키워드를 지정합니다.</p> <p>비전역 영역용 사용자 정의 AI 매니페스트를 제공하지 않으면 <a href="#">예 12-1. “기본 영역 AI 매니페스트”</a>에 표시된 대로 영역용 기본 AI 매니페스트가 사용됩니다.</p>
시스템 구성 프로파일	<p>선택 사항. 비전역 영역용 시스템 구성 프로파일을 0개 이상 제공할 수 있습니다. 이러한 프로파일은 전역 영역 구성용 프로파일과 비슷합니다. 시스템 구성 프로파일 파일에 대한 내용은 <a href="#">11장. 클라이언트 시스템 구성</a>을 참조하십시오. 프로파일을 제공하여 사용자 및 영역 관리자의 루트 암호와 같은 영역 구성을 지정할 수 있습니다. 비전역 영역용 예제 프로파일은 <a href="#">“비전역 영역 시스템 구성 프로파일” [202]</a>을 참조하십시오.</p> <p>영역용 시스템 구성 프로파일을 제공하려면 전역 영역을 설치 중인 설치 서비스에 프로파일을 추가합니다. create-profile 명령에서 이 프로파일을 사용할 모든 영역의 이름과 함께 zonename 조건 키워드를 지정합니다.</p> <p>시스템 구성 프로파일 파일을 제공하지 않으면 영역의 처음 부트 시 시스템 구성 대화식 도구가 실행되어 필요한 데이터를 질의합니다. 대화식 구성 도구 사용에 대한 자세한 내용은 <a href="#">“시스템 재구성” [65]</a>을 참조하십시오.</p>

다음 예제에서는 /tmp/zmanifest.xml AI 매니페스트를 solaris11\_2-sparc 설치 서비스에 추가하고 zone1 및 zone2가 이 매니페스트를 사용하도록 지정합니다.

```
# installadm create-manifest -n solaris11_2-sparc -f /tmp/zmanifest.xml \
-m zmanifest -c zonename="zone1 zone2"
```

다음 예제에서는 /tmp/z1profile.xml 프로파일을 solaris11\_2-sparc 설치 서비스에 추가하고 zone1 및 zone2가 이 프로파일을 사용하도록 지정합니다.

```
# installadm create-profile -n solaris11_2-sparc -f /tmp/z1profile.xml \
-p z1profile -c zonename="zone1 zone2"
```

다음 예제에서는 /tmp/z2profile.xml 프로파일을 solaris11\_2-sparc 설치 서비스에 추가하고 zone2가 이 프로파일을 사용하도록 지정합니다.

```
# installadm create-profile -n solaris11_2-sparc -f /tmp/z2profile.xml \
-p z2profile -c zonename=zone2
```

다음 예제에서는 solaris11\_2-sparc 설치 서비스에 추가된 AI 매니페스트 및 시스템 구성 프로파일을 보여줍니다.

```
$ installadm list -n solaris11_2-sparc -m -p

Service Name      Manifest Name    Type    Status  Criteria
-----
solaris11_2-sparc line1-netra2000 xml     active  mac = 00:14:4F:2D:7A:DC
                  zmanifest       xml     active  zonename = zone1,zone2
                  orig_default    derived default none

Service Name      Profile Name     Criteria
-----
solaris11_2-sparc z1profile       zonename = zone1,zone2
                  z2profile       zonename = zone2
```

## 비전역 영역 AI 매니페스트

이 비전역 영역 설치용 AI 매니페스트는 전역 영역 설치용 AI 매니페스트와 비슷합니다. AI 매니페스트 요소 및 속성에 대한 자세한 내용은 [ai\\_manifest\(4\)](#) 매뉴얼 페이지를 참조하십시오.

다음 요소나 속성은 비전역 영역 AI 매니페스트에 사용하지 마십시오.

- ai\_instance 요소의 auto\_reboot 속성
- ai\_instance 요소의 http\_proxy 속성
- target 요소의 disk 자식 요소
- logical 요소의 noswap 속성
- logical 요소의 nodump 속성
- configuration 요소

target 요소의 logical 자식 요소만 비전역 영역 AI 매니페스트에 사용할 수 있습니다. logical 요소에서는 하나의 zpool 자식 요소만 지정할 수 있습니다.

zpool 요소에서 filesystem 및 be 자식 요소만 비전역 영역 AI 매니페스트에 사용할 수 있습니다.

software 요소의 type 속성에 지원되는 유일한 값은 IPS이며, 이것이 기본값입니다.

### 예 12-1 기본 영역 AI 매니페스트

다음 파일은 비전역 영역의 기본 AI 매니페스트를 보여줍니다. 영역용 사용자 정의 AI 매니페스트를 제공하지 않으면 이 매니페스트가 사용됩니다. 이 매니페스트는 `/usr/share/auto_install/manifest/zone_default.xml`에서 사용할 수 있습니다.

target 섹션은 영역에 대한 ZFS 파일 시스템을 정의합니다. destination 섹션은 설치할 로컬을 지정합니다. software\_data 섹션은 solaris-small-server 패키지 설치를 지정합니다. solaris-small-server 패키지는 대부분의 비전역 영역에 필요한 도구 및 장치 드라이버의 그룹 패키지입니다. solaris-small-server 그룹 패키지에 포함된 전체 패키지 목록을 보러

면 “Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”의 “그룹 패키지의 설치 가능한 모든 패키지 나열”에 설명된 대로 pkg contents 명령을 사용하십시오.

지정된 패키지 소스는 없습니다. 시스템 저장소에 대한 내용은 pkg.sysrepo(1M)를 참조하십시오.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

Copyright (c) 2011, 2013, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">

<auto_install>
  <ai_instance name="zone_default">
    <target>
      <logical>
        <zpool name="rpool">
          <!--
            Subsequent <filesystem> entries instruct an installer
            to create following ZFS datasets:

                <root_pool>/export      (mounted on /export)
                <root_pool>/export/home (mounted on /export/home)

            Those datasets are part of standard environment
            and should be always created.

            In rare cases, if there is a need to deploy a zone
            without these datasets, either comment out or remove
            <filesystem> entries. In such scenario, it has to be also
            assured that in case of non-interactive post-install
            configuration, creation of initial user account is
            disabled in related system configuration profile.
            Otherwise the installed zone would fail to boot.
          -->
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris">
            <options>
              <option name="compression" value="on"/>
            </options>
          </be>
        </zpool>
      </logical>
    </target>

    <software type="IPS">
      <destination>
        <image>
          <!-- Specify locales to install -->
          <facet set="false">facet.locale.*</facet>
          <facet set="true">facet.locale.de</facet>
        </image>
      </destination>
    </software>
  </ai_instance>
</auto_install>
```

```

        <facet set="true">facet.locale.de_DE</facet>
        <facet set="true">facet.locale.en</facet>
        <facet set="true">facet.locale.en_US</facet>
        <facet set="true">facet.locale.es</facet>
        <facet set="true">facet.locale.es_ES</facet>
        <facet set="true">facet.locale.fr</facet>
        <facet set="true">facet.locale.fr_FR</facet>
        <facet set="true">facet.locale.it</facet>
        <facet set="true">facet.locale.it_IT</facet>
        <facet set="true">facet.locale.ja</facet>
        <facet set="true">facet.locale.ja_*</facet>
        <facet set="true">facet.locale.ko</facet>
        <facet set="true">facet.locale.ko_*</facet>
        <facet set="true">facet.locale.pt</facet>
        <facet set="true">facet.locale.pt_BR</facet>
        <facet set="true">facet.locale.zh</facet>
        <facet set="true">facet.locale.zh_CN</facet>
        <facet set="true">facet.locale.zh_TW</facet>
    </image>
</destination>
<software_data action="install">
    <name>pkg:/group/system/solaris-small-server</name>
</software_data>
</software>
</ai_instance>
</auto_install>

```

## 비전역 영역 시스템 구성 프로파일

영역용 시스템 구성 프로파일을 제공하여 언어, 로케일, 시간대, 터미널, 사용자 및 영역 관리자의 루트 암호와 같은 영역 매개변수를 구성할 수 있습니다. 시간대를 구성할 수 있지만 시간은 설정할 수 없습니다. 이름 서비스를 구성할 수 있습니다.

영역에 허용되지 않는 구성을 지정하면 이러한 등록 정보 설정이 무시됩니다.

다음 파일에서는 비전역 영역용 샘플 시스템 구성 프로파일 파일을 보여줍니다.

```

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/config-user">
    <instance enabled="true" name="default">
      <property_group type="application" name="root_account">
        <propval type="astring" name="login" value="root"/>
        <propval type="astring" name="password" value="encrypted_password"/>
        <propval type="astring" name="type" value="normal"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/timezone">
    <instance enabled="true" name="default">

```

```
<property_group type="application" name="timezone">
  <propval type="astring" name="localtime" value="UTC"/>
</property_group>
</instance>
</service>
<service version="1" type="service" name="system/environment">
  <instance enabled="true" name="init">
    <property_group type="application" name="environment">
      <propval type="astring" name="LC_ALL" value="C"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="system/identity">
  <instance enabled="true" name="node">
    <property_group type="application" name="config">
      <propval type="astring" name="nodename" value="z2-test"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="system/keymap">
  <instance enabled="true" name="default">
    <property_group type="system" name="keymap">
      <propval type="astring" name="layout" value="US-English"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="system/console-login">
  <instance enabled="true" name="default">
    <property_group type="application" name="ttymon">
      <propval type="astring" name="terminal_type" value="vt100"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="network/physical">
  <instance enabled="true" name="default">
    <property_group type="application" name="netcfg"/>
  </instance>
</service>
</service_bundle>
```



## 처음 부트 중 사용자 정의 스크립트 실행

AI 매니페스트나 시스템 구성 프로파일에서 완료할 수 없는 추가 설치 및 구성을 수행하려면 일회용 SMF 서비스를 통해 처음 부트 시 실행되는 스크립트를 만들 수 있습니다.

1. 처음 부트 스크립트를 만듭니다.
2. 처음 부트 시 한번 스크립트를 실행하는 SMF 서비스의 매니페스트를 만듭니다.
3. 서비스 매니페스트 및 스크립트를 포함하는 IPS 패키지를 만듭니다.
4. IPS 패키지 저장소에 패키지를 추가합니다.
5. AI 매니페스트에 해당 패키지를 지정하여 AI 설치 중 패키지를 설치합니다.

서비스가 AI 설치 후에 처음 재부트 시 스크립트를 실행합니다.

### 처음 부트 시 한번 실행 컨트롤 구현

다음 절차는 스크립트가 새로 설치된 시스템의 처음 부트 시에만 실행하고, 스크립트가 한번만 실행하도록 보증하는 방법을 보여줍니다.

#### ▼ 처음 부트 시 한번 실행을 보증하는 방법

1. 스크립트를 실행할 서비스를 만듭니다.  
이 간단한 서비스를 만드는 가장 쉬운 방법은 Using the Manifest Creation Tool에 표시된 대로 “매니페스트 만들기 도구 사용” [209] 명령을 사용하는 것입니다.
2. 스크립트를 실행하기 전에 스크립트 완료 플래그를 설정합니다.  
서비스 매니페스트에 부울 완료 등록 정보를 정의하고 값을 false로 설정합니다. Example 13-3에서 매니페스트의 예 13-3. “생성된 SMF 서비스 매니페스트” 등록 정보를 참조하십시오.
3. 스크립트 끝에 스크립트 완료 플래그를 설정합니다.  
스크립트 끝에 svccfg 명령을 사용하여 completed 등록 정보를 true로 설정합니다. svcadm 명령을 사용하여 새 등록 정보 값으로 서비스를 새로 고칩니다. 예 13-1. “템플릿 처음 부트 스크립트”에서 샘플 스크립트 끝을 참조하십시오.

#### 4. 스크립트가 완료되면 서비스를 사용 안함으로 설정합니다.

서비스 매니페스트에서 기본 서비스 인스턴스가 만들어지고 사용으로 설정됩니다. 스크립트에서 서비스가 사용 안함으로 설정됩니다. 처음 부트 스크립트를 종료할 때 SMF\_EXIT\_TEMP\_DISABLE 종료 코드를 사용하여 서비스의 start 메소드를 종료하고 서비스를 일시적으로 사용 안함으로 설정합니다. 서비스가 사용 안함으로 설정되고 서비스의 stop 메소드가 실행되지 않습니다.

서비스를 일시적으로 사용 안함으로 설정하는 것이 영구적으로 사용 안함으로 설정하는 것보다 선호되는데, 더 쉽게 다시 사용으로 설정할 수 있기 때문입니다. 어떤 상황에서 영역 복제나 마이그레이션과 같이 완료된 구성 작업을 업데이트하려면 스크립트(및 서비스)를 다시 실행해야 합니다. 서비스가 영구적으로 사용 안함으로 설정된 경우 서비스를 다시 사용으로 설정하려면 svcadm enable 명령을 실행해야 합니다.

또한 서비스를 일시적으로 사용 안함으로 설정하는 것은 온라인 상태로 두는 것보다 선호됩니다. 온라인 상태의 서비스는 재부트할 때마다 작업 중인 것으로 나타날 수 있습니다. 이 예제에서 서비스의 이름은 site/first-boot-script-svc입니다. 클라이언트를 부트한 후에 서비스가 disabled 상태임을 볼 수 있습니다.

```
$ svcs first-boot-script-svc
STATE          STIME          FMRI
disabled       8:24:16       svc:/site/first-boot-script-svc:default
```

## 처음 부트 시 실행할 스크립트 만들기

스크립트에 사용할 수 있는 소스를 알려면 처음 부트 시 클라이언트 시스템에 설치된 도구를 알아야 합니다. solaris-large-server 패키지가 기본적으로 설치됩니다. 해당 그룹 패키지를 설치한 경우 Python, bash, ksh 및 기타 처음 부트 시 사용 가능한 도구가 제공됩니다. solaris-large-server 그룹 패키지에 포함된 전체 패키지 목록을 보려면 [“Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”](#)의 [“그룹 패키지의 설치 가능한 모든 패키지 나열”](#)에 설명된 대로 pkg contents 명령을 사용하십시오. solaris-large-server 패키지에 제공되지 않은 스크립트의 소스를 사용하려면 필요한 패키지를 식별하고 SI 매니페스트에 지정하십시오. 설치할 다른 패키지의 이름을 찾는 방법은 [“Oracle Solaris 11.2의 소프트웨어 추가 및 업데이트”](#)를 참조하십시오.

- 여러 스크립트의 여러 명령이 서로 충돌하는 것을 피하려면 처음 부트 스크립트를 하나만 사용하십시오.
- 처음 부트 스크립트로 재부트하지 마십시오.

예 13-1            템플릿 처음 부트 스크립트

이 예는 처음 부트 스크립트에서 수행할 작업을 보여줍니다.

- SMF 메소드 종료 코드와 같은 정의를 사용하려면 처음 부트 스크립트가 `/lib/svc/share/smf_include.sh`를 로드해야 합니다.
- 스크립트가 이전 부트 시 이미 실행되었는지 여부를 테스트해야 합니다. `completed` 등록 정보가 이미 `true`로 설정된 경우 `start` 메소드를 종료하고 서비스를 일시적으로 사용 안함으로 설정합니다.

스크립트의 다음 라인은 `site/first-boot-script-svc:default` 서비스 인스턴스에서 `config` 등록 정보 그룹의 `completed` 등록 정보 값을 가져와서 로컬 `completed` 변수에 지정합니다.

```
completed=`svccfg -p config/completed site/first-boot-script-svc:default`
```

스크립트의 다음 라인은 `SMF_EXIT_TEMP_DISABLE` 종료 코드를 서비스 `start` 메소드로 보냅니다. 단문 종료 이유로 `method_completed`, 장문 종료 이유로 "Configuration completed"를 함께 사용합니다.

```
smf_method_exit $SMF_EXIT_TEMP_DISABLE script_completed "Configuration completed"
```

- 처음 부트 스크립트가 방금 SI 설치에서 만든 부트 환경(BE)의 복사본을 저장해야 합니다. 처음 부트 스크립트를 수정하기 전에 BE의 복사본을 저장하면, 저장된 BE로 부트하면서 스크립트에서 소개된 문제를 쉽게 복구할 수 있습니다.
- 스크립트가 작업을 마치면 `completed` 등록 정보 값을 `true`로 설정하고, 새 등록 정보 값으로 서비스를 새로 고치고, `start` 메소드를 종료하고, 서비스를 일시적으로 사용 안함으로 설정해야 합니다. `svccfg` 명령을 사용하여 `completed` 등록 정보를 `true`로 설정하고 `svcadm` 명령을 사용하여 서비스를 새로 고칩니다.

기본적으로 `sh`는 `ksh93`임을 잊지 마십시오.

```
#!/bin/sh

# Load SMF shell support definitions
. /lib/svc/share/smf_include.sh

# If nothing to do, exit with temporary disable
completed=

$(svccfg -p config/completed site/first-boot-script-svc:default
)
[ "${completed}" = "true" ] && \
    smf_method_exit $SMF_EXIT_TEMP_DISABLE completed "Configuration completed"

# Obtain the active BE name from beadm: The active BE on reboot has an R in
# the third column of 'beadm list' output. Its name is in column one.
bename=

$(beadm list -Hd|nawk -F ';' '{ $3 ~ /R/ {print $1}'
)
beadm create ${bename}.orig
echo "Original boot environment saved as ${bename}.orig"

# Place your one-time configuration tasks here
```

```
# Record that this script's work is done
svccfg -s site/first-boot-script-svc:default setprop config/completed = true
svcadm refresh site/first-boot-script-svc:default

smf_method_exit $SMF_EXIT_TEMP_DISABLE method_completed "Configuration completed"
```

---

작은 정보 - 스크립트의 구문 오류를 확인하려면 -n 옵션을 사용합니다.

```
$ ksh -n first-boot-script.sh
```

---

예 13-2 다중 IP 인터페이스를 구성하는 처음 부트 스크립트

이 예제에서는 두 IP 인터페이스에 주소를 구성하고 기본 경로를 추가하는 first-boot-script.sh라는 처음 부트 스크립트를 보여줍니다.

```
#!/bin/sh

# Load SMF shell support definitions
. /lib/svc/share/smf_include.sh

# If nothing to do, exit with temporary disable
completed=`svccprop -p config/completed site/first-boot-script-svc:default`
[ "${completed}" = "true" ] && \
    smf_method_exit $SMF_EXIT_TEMP_DISABLE completed "Configuration completed"

# Obtain the active BE name from beadm: The active BE on reboot has an R in
# the third column of 'beadm list' output. Its name is in column one.
bename=`beadm list -Hd|nawk -F ';' '{ $3 ~ /R/ {print $1} }`
beadm create ${bename}.orig
echo "Original boot environment saved as ${bename}.orig"

# Create and configure addresses on two IP interfaces
/usr/sbin/ipadm create-ip net0
/usr/sbin/ipadm create-ip net1
/usr/sbin/ipadm create-addr -a 10.153.125.222/24 net0
/usr/sbin/ipadm create-addr -a 169.254.182.77/24 net1

# Add a default route with net0 as the gateway
/usr/sbin/route add default 10.153.125.1 -ifp net0

# Record that this script's work is done
svccfg -s site/first-boot-script-svc:default setprop config/completed = true
svcadm refresh site/first-boot-script-svc:default

smf_method_exit $SMF_EXIT_TEMP_DISABLE method_completed "Configuration completed"
```

처음 부트 스크립트의 또 다른 용도는 useradd 명령을 사용하여 시스템에서 여러 초기 사용자를 구성하는 것입니다.

## SMF 매니페스트 파일 만들기

스크립트를 실행하는 서비스를 정의하는 SMF 매니페스트 파일을 만듭니다.

- 서비스의 `start` 메소드가 처음 부트 스크립트를 실행합니다.
- 이 예는 `multi-user` 종속성을 지정하여 처음 부트 후에 시작 시퀀스에서 늦게 스크립트가 실행되도록 합니다. 처음 부트 스크립트가 무엇인지에 따라, 해당 종속성이 필요하지 않을 수 있습니다. 해당 종속성을 지정하지 않으면 시스템을 정확히 구성하기 전에 스크립트가 실행될 수 있습니다.

---

작은 정보 - 스크립트의 종속성을 평가하고 만족스러운 후에 서비스가 스크립트를 실행하도록 구성합니다.

---

- `completed` 등록 정보는 `false` 값으로 정의됩니다.

## 매니페스트 만들기 도구 사용

`svcbundle` 명령을 사용하여 유효한 서비스 매니페스트를 생성할 수 있습니다. 다음 예에서 기본적으로 `svcbundle` 명령으로 생성된 매니페스트는 일시적 서비스를 지정하고 `multi-user` 종속성을 지정합니다.

예 13-3          생성된 SMF 서비스 매니페스트

다음 명령에서 “[처음 부트 시 실행할 스크립트 만들기](#)” [206]에 표시된 스크립트의 이름은 `start-method` 값으로 지정됩니다. [Creating an IPS Package for the Script and Service](#)에서 만든 패키지가 “[스크립트 및 서비스에 대한 IPS 패키지 만들기](#)” [213] 스크립트를 `/opt/site/first-boot-script.sh`로 설치하므로 스크립트의 이름은 `/opt/site/first-boot-script.sh`로 지정됩니다.

다음 명령에서 `completed` 등록 정보는 등록 정보 그룹 이름, 등록 정보 이름, 등록 정보 유형, 초기 등록 정보 값을 콜론으로 구분된 목록으로 지정합니다.

```
$ svcbundle -s service-name=site/first-boot-script-svc \
-s start-method=/opt/site/first-boot-script.sh \
-s instance-property=config:completed:boolean:false \
> first-boot-script-svc-manifest.xml
```

아래 표시된 생성된 서비스 매니페스트에서 처음 부트 스크립트 `/opt/site/first-boot-script.sh`는 `start` 메소드의 `exec` 속성 값입니다. `completed` 등록 정보는 이 서비스의 기본 인스턴스인 `first-boot-script-svc:default`를 정의하는 `instance` 요소에 지정됩니다.

```
<?xml version="1.0" ?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
```

```
<!--
  Manifest created by svcbundle (2014-Jan-14 16:39:30-0700)
-->
<service_bundle type="manifest" name="site/first-boot-script-svc">
  <service version="1" type="service" name="site/first-boot-script-svc">
    <!--
      The following dependency keeps us from starting until the
      multi-user milestone is reached.
    -->
    <dependency restart_on="none" type="service"
      name="multi_user_dependency" grouping="require_all">
      <service_fmri value="svc:/milestone/multi-user"/>
    </dependency>
    <exec_method timeout_seconds="60" type="method" name="start"
      exec="/opt/site/first-boot-script.sh"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute to stop the service. See smf_method(5) for more
      details.
    -->
    <exec_method timeout_seconds="60" type="method" name="stop"
      exec=":true"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute when the service is refreshed. Services are
      typically refreshed when their properties are changed in the
      SMF repository. See smf_method(5) for more details. It is
      common to retain the value of :true which means that SMF will
      take no action when the service is refreshed. Alternatively,
      you may wish to provide a method to reread the SMF repository
      and act on any configuration changes.
    -->
    <exec_method timeout_seconds="60" type="method" name="refresh"
      exec=":true"/>
    <property_group type="framework" name="startd">
      <propval type="astring" name="duration" value="transient"/>
    </property_group>
    <instance enabled="true" name="default">
      <property_group type="application" name="config">
        <propval type="boolean" name="completed" value="false"/>
      </property_group>
    </instance>
    <template>
      <common_name>
        <loctext xml:lang="C">
          <!--
            Replace this comment with a short name for the
            service.
          -->
        </loctext>
      </common_name>
      <description>
        <loctext xml:lang="C">
          <!--
```

```

        Replace this comment with a brief description of
        the service
        -->
    </loctext>
</description>
</template>
</service>
</service_bundle>

```

## 생성된 매니페스트 사용자 정의

svcbundle 명령으로 생성된 서비스 매니페스트는 수정할 필요 없이 사용자 요구를 충족했을 수 있습니다. 다음 예는 서비스 매니페스트의 수정을 보여줍니다.

서비스 매니페스트를 수정할 경우 svccfg validate 명령을 사용하여 매니페스트가 여전히 유효한지 확인합니다.

**예 13-4** 사용자 정의 서비스 매니페스트: 스크립트 실행에 허용된 시간 늘리기

생성된 서비스 매니페스트의 다음 복사본에서 exec\_method 시간 초과 기본값인 60초가 start 메소드에 대해 증가했습니다. start 메소드에 처음 부트 스크립트를 실행할 충분한 시간이 있는지 확인하십시오.

```

<?xml version="1.0" ?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<!--
  Manifest created by svcbundle (2014-Jan-14 16:39:30-0700)
-->
<service_bundle type="manifest" name="site/first-boot-script-svc">
  <service version="1" type="service" name="site/first-boot-script-svc">
    <!--
      The following dependency keeps us from starting until the
      multi-user milestone is reached.
    -->
    <dependency restart_on="none" type="service"
      name="multi_user_dependency" grouping="require_all">
      <service_fmri value="svc:/milestone/multi-user"/>
    </dependency>
    <!--
      Make sure the start method has adequate time to run the script.
    -->
    <exec_method timeout_seconds="360" type="method" name="start"
      exec="/opt/site/first-boot-script.sh"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute to stop the service. See smf_method(5) for more
      details.
    -->
  
```

```

<exec_method timeout_seconds="60" type="method" name="stop"
  exec=":true"/>
<!--
  The exec attribute below can be changed to a command that SMF
  should execute when the service is refreshed. Services are
  typically refreshed when their properties are changed in the
  SMF repository. See smf_method(5) for more details. It is
  common to retain the value of :true which means that SMF will
  take no action when the service is refreshed. Alternatively,
  you may wish to provide a method to reread the SMF repository
  and act on any configuration changes.
-->
<exec_method timeout_seconds="60" type="method" name="refresh"
  exec=":true"/>
<property_group type="framework" name="startd">
  <propval type="astring" name="duration" value="transient"/>
</property_group>
<instance enabled="true" name="default">
  <property_group type="application" name="config">
    <propval type="boolean" name="completed" value="false"/>
  </property_group>
</instance>
<template>
  <common_name>
    <loctext xml:lang="C">
      <!--
        Replace this comment with a short name for the
        service.
      -->
    </loctext>
  </common_name>
  <description>
    <loctext xml:lang="C">
      <!--
        Replace this comment with a brief description of
        the service
      -->
    </loctext>
  </description>
</template>
</service>
</service_bundle>

```

```
$ svccfg validate first-boot-script-svc-manifest.xml
```

**예 13-5** 사용자 정의 서비스 매니페스트: 비전역 영역 설치 후 스크립트 실행 확인

다음 서비스 매니페스트 발췌 부분에서는 비전역 영역이 모두 설치된 후 처음 부트 스크립트가 실행되도록 svc:/milestone/multi-user에 대한 종속성이 svc:/system/zones-install에 대한 종속성으로 변경됩니다.

```
<!--
```

```

    The following dependency keeps us from starting until all
    non-global zones are installed.
-->
<dependency restart_on="none" type="service"
  name="ngz_dependency" grouping="require_all">
  <service_fmri value="svc:/system/zones-install"/>
</dependency>

```

## 스크립트 및 서비스에 대한 IPS 패키지 만들기

다음은 포함하는 IPS 패키지를 만듭니다.

- “SMF 매니페스트 파일 만들기” [209]에서 만든 서비스 매니페스트 파일
- “처음 부트 시 실행할 스크립트 만들기” [206]에서 만든 처음 부트 스크립트
- AI 서버와 같이 다른 위치에서 제공할 수 없는 스크립트에 필요한 모든 파일

### ▼ IPS 패키지를 만들어 게시하는 방법

#### 1. 디렉토리 계층을 만듭니다.

이 예에서 서비스 매니페스트가 /lib/svc/manifest/site로 설치되고 처음 부트 스크립트가 /opt/site로 설치됩니다.

```

$ mkdir -p proto/lib/svc/manifest/site
$ mkdir -p proto/opt/site
$ cp first-boot-script-svc-manifest.xml proto/lib/svc/manifest/site
$ cp first-boot-script.sh proto/opt/site

```

#### 2. 패키지 매니페스트를 만듭니다.

다음과 같이 first-boot-script.p5m이라는 파일을 만듭니다.

```

set name=pkg.fmri value=first-boot-script@1.0,5.11-0
set name=pkg.summary value="AI first-boot script"
set name=pkg.description value="Script that runs at first boot after AI installation"
set name=info.classification value=\
  "org.opensolaris.category.2008:System/Administration and Configuration"
file lib/svc/manifest/site/first-boot-script-svc-manifest.xml \
  path=lib/svc/manifest/site/first-boot-script-svc-manifest.xml owner=root \
  group=sys mode=0444
dir path=opt/site owner=root group=sys mode=0755
file opt/site/first-boot-script.sh path=opt/site/first-boot-script.sh \
  owner=root group=sys mode=0555

```

처음 부트 스크립트가 무엇인지에 따라, 종속성을 지정해야 할 수 있습니다. 이 매니페스트를 수정할 경우 새 매니페스트가 올바른지 확인합니다. 경고를 무시할 수 있습니다. pkgdepend, pkgmogrify 및 pkglint 명령에 대한 정보를 포함하여 패키지를 만드는 방법에 대한 자세

한 내용은 “Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2 ”의 2 장, “Packaging Software With IPS”를 참조하십시오.

### 3. 패키지 저장소를 만듭니다.

이 예는 firstboot를 게시자로 사용하여 로컬 디렉토리에 저장소를 만듭니다.

---

참고 - 설치할 때 AI 클라이언트에서 액세스 가능한 디렉토리에 저장소를 만듭니다.

---

```
$ pkgrepo create firstbootrepo
$ pkgrepo -s firstbootrepo add-publisher firstboot
```

### 4. 패키지를 게시합니다.

```
$ pkgsend publish -d ./proto -s ./firstbootrepo first-boot-script.p5m
pkg://firstboot/first-boot-script@1.0,5.11-0:20140114T022508Z
PUBLISHED
```

클라이언트가 firstbootrepo 저장소에서 패키지를 설치할 수 있습니다. 다음 섹션에 표시된 대로 firstbootrepo 원본과 함께 firstboot 게시자가 AI 매니페스트에 정의됩니다.

### 5. 패키지가 사용 가능한지 확인합니다.

패키지를 나열하여 패키지가 사용 가능한지 확인합니다.

```
$ pkg list -g ./firstbootrepo first-boot-script
NAME (PUBLISHER)          VERSION  IFO
first-boot-script (firstboot)  1.0-0   ---
```

### 6. (선택 사항) 패키지의 설치를 테스트합니다.

-n 옵션은 패키지가 설치되지 않음을 나타냅니다.

```
# pkg set-publisher -g ./firstbootrepo firstboot
# pkg publisher
PUBLISHER TYPE    STATUS P LOCATION
solaris   origin  online F http://http://pkg.oracle.com/solaris/release/
firstboot origin  online F file:///home/user1/firstboot/firstbootrepo/
# pkg list -af first-boot-script
NAME (PUBLISHER)          VERSION  IFO
first-boot-script (firstboot)  1.0-0   ---
# pkg install -nv first-boot-script
Packages to install:      1
Estimated space available: 50.68 GB
Estimated space to be consumed: 64.66 MB
Create boot environment: No
Create backup boot environment: No
Rebuild boot archive:    No

Changed packages:
firstboot
first-boot-script
None -> 1.0,5.11-0:20140114T022508Z
```

```

Planning linked: 0/2 done; 1 working: zone:z2
Linked image 'zone:z2' output:
| Estimated space available: 50.68 GB
| Estimated space to be consumed: 62.07 MB
| Rebuild boot archive: No
\

Planning linked: 1/2 done; 1 working: zone:z1
Linked image 'zone:z1' output:
| Estimated space available: 50.67 GB
| Estimated space to be consumed: 62.07 MB
| Rebuild boot archive: No
\
    
```

다음 순서 NFS 공유 또는 HTTP를 통해 클라이언트 시스템이 새 저장소에 액세스할 수 있게 만드는 방법은 [“Oracle Solaris 11.2 패키지 저장소 복사 및 만들기”](#)를 참조하십시오.

## AI 클라이언트에 처음 부트 패키지 설치

사용자 정의 AI 매니페스트 파일을 만들고 새 패키지, 게시자 및 저장소 정보를 추가합니다.

### ▼ IPS 패키지를 설치하는 방법

#### 1. AI 매니페스트에 패키지를 추가합니다.

AI 매니페스트의 소프트웨어 설치 섹션에 패키지를 추가합니다. AI 매니페스트 XML 파일을 사용자 정의하거나 이러한 요소를 추가하는 파생된 매니페스트 스크립트를 작성합니다. AI 매니페스트 사용자 정의에 대한 내용은 [10장. 클라이언트 시스템 프로비전](#)을 참조하십시오.

installadm export 명령을 사용하여 하나 이상의 기존 AI 매니페스트의 내용을 검색합니다. 다음 예는 추가할 XML 요소를 보여줍니다.

```

<software type="IPS">
  <source>
    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
    <publisher name="firstboot">
      <origin name="file:///net/host1/export/firstbootrepo"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg:/first-boot-script</name>
  </software_data>
</software>
    
```

원본이 클라이언트가 AI 설치 중 액세스할 수 있는 URI인지 확인합니다. zfs set sharenfs를 사용하여 저장소를 내보내면 클라이언트가 로컬 저장소에 액세스할 수 있습니다.

2. **AI 설치 서비스에서 수정된 AI 매니페스트를 업데이트합니다.**

installadm update-manifest 명령을 사용하여 AI 매니페스트 내용을 처음 부트 스크립트 패키지가 포함된 내용으로 바꿉니다. 업데이트 이후의 매니페스트나 스크립트에 어떤 조건이나 기본 상태는 그대로 남습니다.

3. **클라이언트를 네트워크 부트합니다.**

AI를 사용하도록 클라이언트를 네트워크 부트하여 Oracle Solaris 11 OS 및 사용자 정의 first-boot-script 패키지를 설치합니다. 클라이언트를 설치 후에 부트할 때 서비스가 처음 부트 스크립트를 실행합니다.

## 처음 부트 서비스 테스트

AI 설치를 테스트하기 전에 서비스를 테스트하려면 간단히 테스트 시스템에 패키지를 설치하고 해당 테스트 시스템을 재부트하면 됩니다.

```
# pkg install first-boot-script
    Packages to install: 1
    Create boot environment: No
    Create backup boot environment: No

DOWNLOAD                                PKGS      FILES    XFER (MB)   SPEED
Completed                                1/1       2/2      0.0/0.0     0B/s

PHASE                                     ITEMS
Installing new actions                    7/7
Updating package state database           Done
Updating image state                      Done
Creating fast lookup database             Done
Reading search index                     Done
# pkg list first-boot-script
NAME (PUBLISHER)                          VERSION    IFO
first-boot-script (firstboot)             1.0-0     i--
# pkg info first-boot-script
    Name: first-boot-script
    Summary: AI first-boot script
    Description: Script that runs at first boot after AI installation
    Category: System/Administration and Configuration
    State: Installed
    Publisher: firstboot
    Version: 1.0
    Build Release: 5.11
    Branch: 0
    Packaging Date: Dec 23, 2013 02:50:31 PM
    Size: 3.89 kB
    FMRI: pkg://firstboot/first-boot-script@1.0,5.11-0:20131223T145031Z
```

테스트 시스템을 재부트합니다. 위에 표시된 대로 스크립트가 새 부트 환경을 만든 경우 새 부트 환경으로 부트해야 합니다.

스크립트가 /opt/site 디렉토리에 있고 스크립트의 결과가 올바른지 확인합니다.

서비스의 상태를 확인합니다. 스크립트를 마치고 올바르게 종료된 경우 서비스가 disabled 상태여야 합니다.

```
# svcs first-boot-script-svc
STATE      STIME      FMRI
disabled   8:24:16   svc:/site/first-boot-script-svc:default
```

다음 명령 중 하나를 사용하여 completed 등록 정보의 값을 확인합니다.

```
# svcprop first-boot-script-svc:default
config/completed boolean true
# svcprop -p config/completed first-boot-script-svc:default
true
```

서비스 로그 파일을 검토하려면 다음 명령을 사용하여 로그 파일의 위치를 찾습니다.

```
# svcs -x first-boot-script-svc
svc:/site/first-boot-script-svc:default (?)
  State: disabled since Dec 23, 2013 08:24:16 AM PDT
Reason: Temporarily disabled by service method: "Configuration completed."
  See: http://support.oracle.com/msg/SMF-8000-15
  See: /var/svc/log/site-first-boot-script-svc:default.log
Impact: This service is not running.
```

로그 파일은 다음 정보를 포함합니다.

```
[ Jul 23 08:22:57 Enabled. ]
[ Jul 23 08:24:14 Executing start method ("/opt/site/first-boot-script.sh"). ]
[ Jul 23 08:24:16 Method "start" exited with status 101. ]
[ Jul 23 08:24:16 "start" method requested temporary disable: "Configuration completed"
]
[ Jul 23 08:24:16 Rereading configuration. ]
```

## ▼ 스크립트 또는 서비스를 업데이트하는 방법

스크립트 또는 서비스 매니페스트를 변경하려면 이 절차를 사용하여 업데이트를 설치합니다.

1. 업데이트된 파일을 prototype 디렉토리로 복사합니다.

```
$ cp first-boot-script-svc-manifest.xml proto/lib/svc/manifest/site
$ cp first-boot-script.sh proto/opt/site
```

2. 패키지 버전을 증분합니다.

패키지 매니페스트에서 pkg.fmri 속성 값을 다음과 같이 변경합니다.

```
first-boot-script@1.0,5.11-0.1
```

3. **새 버전을 게시합니다.**

패키지의 새 버전을 저장소로 게시합니다.

```
$ pkgsend publish -d ./proto -s ./firstbootrepo first-boot-script.p5m
pkg://firstboot/first-boot-script@1.0,5.11-0.1:2013123T231948Z
PUBLISHED
```

4. **패키지를 업데이트합니다.**

pkg list -af 명령을 사용하여 새 버전에 액세스할 수 있는지 확인합니다. 패키지 목록을 업데이트하려면 pkg refresh firstboot 명령을 사용해야 할 수 있습니다. pkg update 명령을 사용하여 패키지를 업데이트합니다.

5. **테스트 시스템을 재부트합니다.**

## 클라이언트 시스템 설치

---

이 장에서는 SI 클라이언트에 대한 시스템 요구 사항을 제공하고 각 클라이언트를 올바른 SI 설치 서비스와 연관시키는 방법을 설명합니다.

### 클라이언트가 설치되는 방법

SI 서버를 설정할 때 각 클라이언트 아키텍처마다, 그리고 설치할 각 버전의 Oracle Solaris OS마다 적어도 하나의 설치 서비스를 만들었습니다. 각 설치 서비스를 만들 때 필요에 따라 서로 다른 클라이언트에 대해 사용자 정의된 설치 지침 및 시스템 구성 지침을 만들었습니다. 자동 설치를 시작하려면 클라이언트를 부트하면 됩니다.

클라이언트를 네트워크 부트한 후에 설치 서비스에서 제공된 넷 이미지, 설치 사양, 시스템 구성 사양을 사용하여 클라이언트의 설치 및 구성을 완료합니다.

1. 관리자가 클라이언트를 네트워크 부트합니다.
2. 클라이언트 시스템이 DHCP 서버에 연결하여 클라이언트의 네트워크 구성과 SI 서버의 위치를 검색합니다. SPARC 클라이언트는 선택적으로 OBP에 설정된 `network-boot-arguments` 변수를 사용하여 이 정보를 얻을 수 있습니다.
3. 클라이언트 시스템이 다음 소스 중 하나에서 넷 이미지를 로드합니다.
  - `installadm create-client` 명령으로 이 클라이언트에 지정된 설치 서비스
  - 이 구조에 대한 기본 설치 서비스
4. 클라이언트 시스템이 “**SI 매니페스트 선택**” [134]에 설명된 대로 결정된 SI 매니페스트를 사용하여 설치를 완료합니다.
5. `auto_reboot`가 SI 매니페스트에 설정된 경우 클라이언트 시스템이 재부트되고, 또는 시스템 관리자가 클라이언트를 재부트합니다.
6. 재부트 중, 다음 방법 중 하나로 클라이언트 시스템이 구성됩니다.
  - “**시스템 구성 프로파일 선택**” [135]에 설명된 대로 결정된 시스템 구성 프로파일 사용
  - 대화식 시스템 구성 도구에서 관리자의 응답 사용
7. 재부트 후 클라이언트용으로 설정된 처음 부트 스크립트가 실행됩니다.

SI 클라이언트 설치를 마치면 Automated Installation succeeded 메시지가 화면에 표시되고, /system/volatile/install\_log 로그 파일에 완료 메시지가 표시되고, 해당 클라이언트의 svc:/application/auto-installer SMF 서비스가 online 상태에 도달합니다.

## SPARC 및 x86 클라이언트 시스템 요구 사항

자동 설치용 클라이언트 시스템은 다음 요구 사항을 충족해야 합니다. 이러한 요구 사항을 충족하는 시스템(랩탑, 데스크탑, 가상 머신, 엔터프라이즈 서버 등)을 자동 SI 클라이언트로 사용할 수 있습니다.

네트워크를 통한 SI 설치의 SPARC 및 x86 클라이언트는 다음 요구 사항을 충족해야 합니다.

메모리	최소 1GB
디스크 공간	최소 13GB
네트워크 액세스	클라이언트 시스템은 설치 중 다음 리소스에 액세스할 수 있어야 합니다. <ul style="list-style-type: none"> <li>■ 네트워크 구성 정보를 제공하는 DHCP 서버</li> <li>■ SI 서버</li> <li>■ 클라이언트 시스템에 설치될 패키지를 포함하는 IPS 저장소</li> </ul>

SPARC 클라이언트 시스템에서는 최신 WAN 부트 지원을 포함하는 현재 버전의 OBP(Open Boot PROM)를 포함하도록 펌웨어를 업데이트해야 합니다.

네트워크를 통해 부트하려면 SI에서 SPARC 클라이언트에 대한 WAN 부트를 지원해야 합니다. 클라이언트 OBP(Open Boot PROM)가 WAN 부트를 지원하는지 여부를 확인할 수 있습니다. network-boot-arguments가 eeprom에 설정할 수 있는 유효한 변수인지 확인하면 됩니다.

network-boot-arguments 변수를 표시하거나 명령이 network-boot-arguments: data not available 출력을 반환할 경우 OBP가 WAN 부트를 지원하므로 네트워크를 통해 클라이언트를 설치할 수 있습니다.

```
# eeprom | grep network-boot-arguments
network-boot-arguments: data not available
```

명령 결과에 출력이 없으면 WAN 부트가 지원되지 않으므로 네트워크를 통해 클라이언트를 설치할 수 없습니다. [5장. 매체에서 부트하는 자동 설치](#)를 참조하십시오.

```
# eeprom | grep network-boot-arguments
```

## AI 클라이언트 설정

AI 서버에서 `installadm create-client` 명령을 사용하여 특정 클라이언트를 특정 설치 서비스와 연관시킵니다.

`installadm create-client` 명령에 다음 정보가 필요합니다.

- 클라이언트의 MAC 주소
- 설치에 사용할 클라이언트에 대한 설치 서비스의 이름

x86 클라이언트의 경우 `installadm create-client` 명령에 `-b` 옵션을 사용하여 선택적으로 부트 등록 정보를 지정할 수 있습니다. SPARC 클라이언트의 경우 부트 시 입력하는 부트 명령에 클라이언트 부트 시 사용할 부트 인수를 포함합니다.

## SPARC 클라이언트 설정

다음 예제에서는 MAC 주소가 `00:14:4f:a7:65:70`인 SPARC 클라이언트를 `solaris11_2-sparc` 설치 서비스와 연관시킵니다.

```
# installadm create-client -n solaris11_2-sparc -e 00:14:4f:a7:65:70
```

SPARC `wanboot-cgi` 부트 파일이 `create-service`로 이미 구성되었으므로 DHCP 서버에 구성이 필요 없습니다. 자세한 내용은 “[설치 서비스 만들기](#)” [96]를 참조하십시오.

다음과 같은 `installadm create-client` 명령의 결과가 `/etc/netboot` 디렉토리에 나타납니다.

```
dr-xr-x---  2 webservd webservd   4 Apr  9 08:53 0100144FA76570
```

## x86 클라이언트 설정

다음 예제에서는 MAC 주소가 `0:e0:81:5d:bf:e0`인 x86 클라이언트를 `solaris11_2-i386` 설치 서비스와 연관시킵니다. 이 명령에 의한 DHCP 구성 출력을 DHCP 서버에 추가해야 합니다. 이 DHCP 구성이 완료되지 않으면 클라이언트가 `solaris11_2-i386` 설치 서비스를 부트할 수 없습니다.

```
# installadm create-client -n solaris11_2-i386 -e 0:e0:81:5d:bf:e0
```

```
No local DHCP configuration found. If not already configured, the
following should be added to the DHCP configuration:
```

```
  Boot server IP      : 10.80.239.5
  Boot file(s)       :
    bios clients (arch 00:00): 0100E0815DBFE0.bios
    uefi clients (arch 00:07): 0100E0815DBFE0.uefi
```

다음 예제에서는 `installadm`으로 이 클라이언트에 대한 PXE 부트 항목을 설정하는 방법을 보여줍니다.

```

host 00E0815DBFE0 {
  hardware ethernet 00:E0:81:5D:BF:E0;
  if option arch = 00:00 {
    filename "0100E0815DBFE0.bios";
  } else if option arch = 00:07 {
    filename "0100E0815DBFE0.uefi";
  }
}

```

다음과 같은 `installadm create-client` 명령의 결과가 `/etc/netboot` 디렉토리에 나타납니다.

```

lrwxrwxrwx 1 root root 21 May 6 10:32 0100E0815DBFE0 -> ./0100E0815DBFE0.bios
lrwxrwxrwx 1 root root 44 May 6 10:32 0100E0815DBFE0.bios -> ./solaris11_2-i386/boot/grub/
pxegrub2
lrwxrwxrwx 1 root root 51 May 6 10:32 17:49 0100E0815DBFE0.uefi -> ./solaris11_2-i386/
boot/grub/grub2netx64.efi
-rw-r--r-- 1 root root 1744 May 6 10:32 17:49 grub.cfg.0100E0815DBFE0
-rw-r--r-- 1 root root 1204 May 6 10:32 17:49 menu.conf.0100E0815DBFE0

```

## 클라이언트를 서비스에서 삭제

`installadm delete-client` 명령을 사용하여 클라이언트를 설치 서비스에서 삭제합니다.

```
$ installadm delete-client -e macaddr
```

클라이언트는 하나의 설치 서비스에만 연관될 수 있으므로 서비스 이름을 지정할 필요가 없습니다.

## 클라이언트 설치

설치를 시작하려면 클라이언트를 부트합니다. 이 절에서는 SPARC 또는 x86 클라이언트를 부트하는 방법을 설명합니다. 클라이언트 콘솔에서 설치 진행률을 모니터링할 수 있습니다. 설치 중 발생할 수 있는 오류도 클라이언트 콘솔에 표시됩니다. 또한 설치 진행률을 원격으로 모니터링할 수 있는 방법도 설명합니다.

## 보안 셸을 사용하여 원격으로 설치 모니터

ssh를 사용하여 자동 SI 클라이언트에 대한 네트워크 액세스를 사용으로 설정할 수 있습니다. 이 액세스를 사용하여 `/system/volatile/install_log` 설치 로그 파일에서 진행률을 모니터링하면 설치 진행 상황을 원격으로 관찰할 수 있습니다.

특정 설치 서비스의 모든 클라이언트에 대한 원격 액세스를 사용으로 설정하려면 설치 구성 파일에서 `livessh` 옵션을 `enable`로 설정합니다. 이 액세스가 사용으로 설정된 경우 사용자 이름 `jack`과 암호 `jack`을 사용하여 SI 클라이언트에 로그인할 수 있습니다.

개별 클라이언트에서 부트 명령줄에 이 옵션을 설정할 수도 있습니다.

## x86 클라이언트 설치 모니터

x86 시스템의 경우, 다음 예제에 표시된 대로 `create-service` 하위 명령과 함께 `-b` 옵션을 사용하여 해당 서비스를 사용하는 모든 클라이언트에 대한 부트 등록 정보를 설정합니다.

```
# installadm create-service -a i386 -b livessh=enable
```

다음 발췌 부분은 `/etc/netboot/svcname/grub.cfg` 파일에 등록 정보가 어떻게 나타나는지 보여줍니다.

```
$multiboot $kern /platform/i86pc/kernel/amd64/unix -B livessh=enable,...
```

부트 명령줄에 `livessh`를 지정하여 단일 x86 클라이언트에 대해 `ssh`를 사용으로 설정할 수 있습니다. 지침은 [“Oracle Solaris 11.2 시스템 부트 및 종료”의 “부트 시 GRUB 메뉴를 편집하여 커널 인수 추가”](#)를 참조하십시오.

## SPARC 클라이언트 설치 모니터

SPARC 시스템의 경우 `/etc/netboot` 디렉토리 아래에 마운트된 서비스의 넷 이미지 디렉토리인 `/etc/netboot/svcname/system.conf`를 통해 `system.conf` 파일에 액세스합니다.

`system.conf` 파일에서 옵션은 이름-값 쌍으로 정의됩니다. 다음 예에서 `livessh` 옵션이 `enable`로 설정됩니다.

```
$ cat /etc/netboot/solaris11_2-sparc/system.conf
...install_service=solaris11_2-sparc
install_svc_address=$serverIP:5555
livessh=enable
...
```

부트 명령줄에 `livessh`를 지정하여 단일 SPARC 클라이언트에 대해 `ssh`를 사용으로 설정할 수 있습니다. 다음 예는 이 인수를 지정하는 두 가지 다른 방법을 보여줍니다.

```
ok boot net:dhcp - livessh
ok boot net:dhcp - livessh=enable
```

부트 명령줄에 지정한 `livessh` 사양은 서비스의 `system.conf` 파일에 지정된 설정을 대체합니다. 예를 들어, `system.conf` 파일에 `livessh=enable`로 지정된 경우 부트 명령줄에서 `livessh=disable`을 지정하면 특정 클라이언트에서 `livessh`를 사용 안함으로 설정할 수 있습니다.

```
ok boot net:dhcp - livessh=disable
```

## SPARC 클라이언트 설치

OBP 프롬프트에서 SPARC 클라이언트를 네트워크 부트합니다. 보안 다운로드를 사용할지, 아니면 DHCP를 사용할지를 결정합니다.

### 보안 다운로드를 사용하여 SPARC 클라이언트 설치

자격 증명으로 보호되는 SPARC AI 클라이언트 시스템의 경우 보안 키로 구성된 SPARC OBP 펌웨어를 통해 네트워크에서 넷 부트 파일 및 부트 파일 시스템을 안전하게 다운로드할 수 있습니다. 다운로드된 부트 파일 및 파일 시스템을 검증하려면 OBP에서 펌웨어 키를 지정해야 합니다.

해싱 다이제스트(HMAC)가 SHA1 알고리즘으로 계산되고 AES 암호화 방법이 사용됩니다.

#### 해싱 키 및 암호 키 설정

OBP 명령 프롬프트에서 HMAC 및 암호 키를 설정할 수 있습니다.

다음 예제 명령에서는 SPARC 클라이언트 콘솔의 OBP HMAC를 AI 생성 SHA1 값으로 설정합니다.

```
ok set-security-key wanboot-hmac-sha1 767280bd72bca8cef3d679815dfca54638691ec5
```

다음 예제 명령에서는 SPARC 클라이언트 콘솔에서 OBP AES 암호 키를 설정합니다.

```
ok set-security-key wanboot-aes 38114ef74dc409a161099775f437e030
```

#### 해싱 키 및 암호 키 재설정

서버의 구성에서 클라이언트의 OBP 키가 재생성되는 경우 인증된 AI 설치를 수행하려면 영향을 받는 SPARC 클라이언트에서 키를 업데이트해야 합니다. 기존 OBP 키를 무효화하고 새 OBP 키를 생성하려면 `installadm` 명령에 `-H` 및 `-E` 옵션을 사용하십시오. 서버 인증 전용, 특정 클라이언트용, 특정 설치 서비스용 및 기본 클라이언트용 OBP 키 생성에 대한 자세한 내용은 [“SPARC 클라이언트용 OBP 보안 키” \[114\]](#)를 참조하십시오.

#### 해싱 키 및 암호 키 삭제

HMAC 키 및 암호 키를 삭제하면 해당 클라이언트에서 더 이상 인증이 필요하지 않게 되거나 인증이 시도되지 않습니다. AI를 사용하여 `sec` 등록 정보가 `require-client-auth` 또는 `require-server-auth`로 설정된 설치 서비스를 통해 클라이언트를 설치할 수 없게 됩니다.

OBP 명령 프롬프트에서 HMAC 키 및 암호 키를 삭제하려면 키를 설정할 때 사용하는 것과 같은 명령을 사용하되 값을 제공하지 마십시오.

```
ok set-security-key wanboot-hmac-sha1
ok set-security-key wanboot-aes
```

## DHCP를 사용하여 SPARC 클라이언트 설치

DHCP를 사용하는 경우 다음 네트워크 부트 명령을 사용합니다.

```
ok boot net:dhcp - install
```

## DHCP를 사용하지 않고 SPARC 클라이언트 설치

DHCP를 사용하지 않는 경우 OBP에서 다음 명령을 사용하여 `network-boot-arguments` 변수를 설정하십시오. 이 변수는 OBP에 지속적으로 설정됩니다.

```
ok setenv network-boot-arguments host-ip=client-ip,
router-ip=router-ip,subnet-mask=subnet-mask,hostname=hostname,
file=wanboot-cgi-file
```

다음 명령을 사용하여 클라이언트를 네트워크 부트합니다.

```
ok boot net - install
```

---

참고 - `network-boot-arguments` 변수를 사용하는 경우 SPARC 클라이언트에 DNS 구성 정보가 없습니다. 이 클라이언트와 함께 사용된 AI 매니페스트에서 IPS 패키지 저장소의 위치와 매니페스트의 다른 URI에 대해 호스트 이름 대신 IP 주소를 지정하는지 확인하십시오.

---

## SPARC 클라이언트 네트워크 부트 시퀀스

다음 이벤트가 SPARC 클라이언트의 AI 부트 중 발생합니다.

1. 클라이언트가 부트하고, DHCP 서버에서 또는 OBP에 설정된 `network-boot-arguments` 변수에서 네트워크 구성과 `wanboot-cgi` 파일의 위치를 가져옵니다.
2. `wanboot-cgi` 프로그램에서 `wanboot.conf`를 읽고 WAN 부트 이진의 위치를 클라이언트로 보냅니다.
3. WAN 부트 이진이 HTTP를 사용하여 다운로드되고 클라이언트가 WAN 부트 프로그램을 부트합니다.
4. WAN 부트가 `boot_archive` 파일을 가져오고 Oracle Solaris OS가 부트됩니다.
5. 이미지 아카이브 `solaris.zlib` 및 `solarismisc.zlib`가 HTTP를 사용하여 다운로드됩니다.
6. mDNS 조회에서 또는 `system.conf` 파일에서 지정된 AI 설치 서비스로부터 AI 매니페스트와 시스템 구성 프로파일이 다운로드됩니다.

7. AI 설치 프로그램이 AI 매니페스트로 호출되어 클라이언트에 Oracle Solaris OS의 설치를 수행합니다.

## x86 클라이언트 설치

다음과 같은 네트워크 부트 방식 중 하나를 사용하여 x86 클라이언트 설치를 시작합니다.

- 적절한 기능 키를 누릅니다. 예를 들어, 일부 시스템은 F12를 사용하여 네트워크에서 부트합니다.
- BIOS에서 부트 순서를 변경합니다.

클라이언트를 부트할 때 부트를 시작할 네트워크 장치를 선택합니다.

다음 이벤트가 x86 클라이언트의 AI 부트 중 발생합니다.

1. 클라이언트가 부트하여 IP 주소를 가져오고, DHCP 서버에서 제공된 위치로부터 부트 파일이 다운로드됩니다.
2. 부트 파일이 로드되고 GRUB 메뉴 파일을 읽습니다.
3. 사용자가 GRUB 메뉴에서 두번째 옵션 "Oracle Solaris 11.2 Automated Install(Oracle Solaris 11.1 자동 설치)"을 선택합니다.
4. 부트 파일이 부트 아카이브 파일을 가져오고 Oracle Solaris OS가 TFTP를 사용하여 부트됩니다.
5. 넷 이미지 아카이브 solaris.zlib 및 solarismisc.zlib가 GRUB 메뉴에서 제공된 대로 HTTP를 사용하여 다운로드됩니다.
6. mDNS 조회에서 또는 부트된 GRUB 메뉴 항목에서 지정된 AI 설치 서비스로부터 AI 매니페스트와 시스템 구성 프로파일이 다운로드됩니다.
7. AI 설치 프로그램이 AI 매니페스트로 호출되어 설치를 수행합니다.

시스템이 성공적으로 PXE 부트된 경우 GRUB 메뉴를 표시하기 전에 다음 메시지가 잠시 표시됩니다.

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation
```

```
CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
```

GRUB 메뉴가 두 메뉴 항목으로 나타납니다. 두번째 항목을 선택하여 자동 설치를 시작합니다.

```
Oracle Solaris 11.2 Text Installer and command line
Oracle Solaris 11.2 Automated Install
```

기본 GRUB 메뉴 항목 "Text Installer and command line"은 핸드프리 자동 설치를 시작하지 않고 이미지를 부트합니다. GRUB 메뉴의 두번째 항목 "Automated Install"을 선택하여 자동 설치를 시작합니다. 첫번째 메뉴 항목을 선택하면 클라이언트를 부트할 때 **명령줄에서**

“자동 설치 시작” [241]에 표시된 대로 메뉴가 표시됩니다. 이 메뉴를 사용하여 시스템을 조사하거나 설치합니다.

## 클라이언트 설치 메시지

다음 메시지는 SPARC 및 x86 설치 양쪽에 공통됩니다.

### 자동 설치 시작 메시지

클라이언트가 성공적으로 부트하고 설치 파일을 다운로드할 수 있으면 다음 메시지가 표시됩니다.

```
Automated Installation started
The progress of the Automated Installation will be output to the console
Detailed logging is in the logfile at /system/volatile/install_log
Press RETURN to get a login prompt at any time.
```

root에 암호 solaris로 로그인하여 /system/volatile/install\_log에서 설치 메시지를 모니터링할 수 있습니다.

### 자동 설치 성공 메시지

다음 메시지가 보이면 설치를 성공한 것입니다.

```
Automated Installation finished successfully
The system can be rebooted now
Please refer to the /system/volatile/install_log file for details
After reboot it will be located at /var/log/install/install_log
```

Reboot to start the installed system

SI 매니페스트에 자동 재부트를 설정한 경우 이 시점에 시스템이 재부트됩니다. 설치 성공 후에 자동 재부트를 지정하려면 <ai\_instance> 태그의 auto\_reboot 속성을 true로 설정합니다. 기본값은 false이며, 클라이언트가 설치 성공 후에 자동으로 재부트하지 않습니다.



# ◆◆◆ 15 장

## 자동 설치 문제 해결

---

이 장에서는 몇몇 가능한 실패와 복구 방법을 설명합니다.

### 클라이언트 설치 실패

이 섹션은 클라이언트 설치를 실패할 경우 취할 조치를 제안합니다.

### 설치 로그 및 지침 확인

클라이언트 시스템에 설치하는 데 실패할 경우 `/system/volatile/install_log`에서 로그를 찾을 수 있습니다.

이 클라이언트에 사용된 AI 매니페스트는 `/system/volatile/ai.xml`에 있습니다. 이 클라이언트에 사용된 시스템 구성 프로파일은 `/system/volatile/profile/*`에 있습니다.

또한 `/var/ai/image-server/logs/*`에서 AI 서버의 웹 서버 로그를 확인할 수 있습니다. 클라이언트가 잘못되거나 예상치 못한 AI 매니페스트 또는 시스템 구성 프로파일을 수신하는 경우 또는 초기 부트 단계 후 클라이언트가 파일을 다운로드할 수 없는 경우 이 로그가 유용합니다.

### DNS 확인

비어 있지 않은 `/etc/resolv.conf` 파일이 존재하는지 확인하여 DNS가 클라이언트에 구성되었는지 여부를 검사합니다.

`/etc/resolv.conf`가 존재하지 않거나 비어 있으면 DHCP 서버가 DNS 서버 정보를 클라이언트에 제공하는지 확인합니다.

```
# /sbin/dhcpinfo DNSserv
```

이 명령이 아무것도 반환하지 않으면 DHCP 서버가 DNS 서버 정보를 클라이언트에 제공하도록 설정되지 않았습니다. 이 문제를 해결하려면 DHCP 관리자에게 문의하십시오.

/etc/resolv.conf 파일이 존재하고 올바르게 구성된 경우 다음 가능한 문제가 있는지 확인하고 시스템 관리자에게 해결책을 문의하십시오.

- DNS 서버가 IPS 저장소 서버 이름을 분석 중이 아닐 수 있습니다.
- DNS 서버에 연결할 기본 경로가 없습니다.

## 클라이언트 부트 오류 확인

클라이언트 시스템을 부트할 때 발생할 수 있는 오류에 대한 다음 추가 정보를 검토합니다.

- [“SPARC 네트워크 부팅 오류 및 가능한 원인” \[231\]](#)
- [“x86 네트워크 부팅 오류 및 가능한 원인” \[234\]](#)
- [“SPARC 및 x86 오류 메시지” \[237\]](#)

## 부트 디스크를 찾을 수 없음

자동 설치 중에 부트 디스크를 찾을 수 없는 경우 부트 디스크를 확인하고 AI 매니페스트를 수정합니다.

1. SPARC OBP 또는 x86 BIOS에서 부트 장치를 명시적으로 선택합니다.
2. 시스템을 재부트합니다.
3. 설치할 시스템에 로그인합니다.
4. 설치 중에 사용할 장치를 식별합니다. format 명령에 표시된 대로 SYS/HDD\* 콘센트 이름 또는 CTD 디스크 이름을 통해 장치를 식별할 수 있습니다.
5. /system/volatile/ai.xml 매니페스트를 수정하고 "boot\_disk" 값을 바꿉니다. 예를 들면 다음과 같습니다.

```
<disk_keyword key="SYS/HDD1" name_type="receptacle"/>
```

```
<disk_keyword key="c0t5000CCA012B2A254d0" name_type="ctd"/>
```

6. 설치 서비스를 새로 고칩니다.

```
# svcadm clear auto-installer
```

## SPARC 네트워크 부팅 오류 및 가능한 원인

이 섹션은 네트워크를 통해 SPARC 클라이언트를 부트할 때 볼 수 있는 오류/문제와 가능한 원인을 설명합니다.

- “BOOTP/DHCP 릴레이 대기 중 시간 초과” [231]
- “부트 로드 실패” [231]
- “내부 서버 오류 또는 WAN 부트 경보” [232]
- “ERROR 403: Forbidden 또는 ERROR 404: Not Found” [232]
- “자동 설치 프로그램 시작되지 않음” [233]
- “잘못된 HMAC 값” [233]

### BOOTP/DHCP 릴레이 대기 중 시간 초과

DHCP 서버가 SPARC 클라이언트의 요청에 응답하지 않으면 다음 메시지가 표시됩니다.

```
...
OpenBoot 4.23.4, 8184 MB memory available, Serial #69329298.
Ethernet address 0:14:4f:21:e1:92, Host ID: 8421e192.
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
Timed out waiting for BOOTP/DHCP reply
```

시간 초과 메시지는 클라이언트가 DHCP 요청을 보냈는데 해당 요청에 응답하지 않았음을 나타냅니다. 이 오류는 DHCP 구성 문제 때문인 것 같습니다. 클라이언트가 DHCP 서버에 올바르게 구성되었는지 여부를 확인합니다.

### 부트 로드 실패

SI 클라이언트가 boot\_archive 다운로드를 시작했지만 "부트 로드 실패" 오류와 함께 실패한 경우 클라이언트 DHCP 정보가 잘못 구성되었음을 나타냅니다.

```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
HTTP: Bad Response: 500 Internal Server Error
Evaluating:

Boot load failed
```

이 오류는 다른 DHCP 서버가 클라이언트에 응답 중인 경우 발생할 수 있습니다. 이 클라이언트에 대한 DHCP 구성을 확인합니다. 구성이 정확한 것 같으면 서브넷에 다른 DHCP 서버가 있는지 확인합니다.

## 내부 서버 오류 또는 WAN 부트 경보

SI 클라이언트가 부트 아카이브 다운로드를 시작할 IP 주소 및 초기 매개변수를 얻은 후에 클라이언트가 boot\_archive를 찾거나 다운로드하지 못할 수 있습니다.

- 클라이언트가 boot\_archive를 찾을 수 없으면 다음 오류가 표시됩니다.

```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 366 of 366 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Mon Aug 5 20:46:43 wanboot alert: miniinfo: Request returned code 500
Mon Aug 5 20:46:44 wanboot alert: Internal Server Error \
(root filesystem image missing)
```

- SI 클라이언트가 boot\_archive 파일을 찾았지만 파일에 액세스할 수 없으면 다음 오류가 표시됩니다.

```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 366 of 366 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Mon Aug 5 20:53:02 wanboot alert: miniroot: Request returned code 403
Mon Aug 5 20:53:03 wanboot alert: Forbidden
```

이러한 양쪽 문제에 대해 이 클라이언트용으로 구성된 boot\_archive 파일을 수정합니다. \$IMAGE/boot/boot\_archive에서 boot\_archive의 경로 이름 및 권한을 확인합니다.

## ERROR 403: Forbidden 또는 ERROR 404: Not Found

SI 클라이언트가 boot\_archive를 성공적으로 다운로드하고 Oracle Solaris 커널을 부트했지만 이미지 아카이브 중 하나를 가져오지 못한 경우 ERROR 403: Forbidden 및 ERROR 404: Not Found 메시지가 표시됩니다. 어떤 파일이 문제를 일으키는지 나타내는 오류 메시지가 표

시됩니다. 예를 들어, SPARC 클라이언트의 다음 출력에서 `solaris.zlib` 파일이 존재하지 않거나 지정된 위치에 액세스할 수 없습니다.

```
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 368 of 368 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Mon May 5 18:57:36 wanboot progress: miniroot: Read 235737 of 235737 kB (100%)
Mon May 5 18:57:36 wanboot info: miniroot: Download complete
SunOS Release 5.11 Version 11.2 64-bit
Copyright (c) 1983, 2014, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2014-05-05 18:52:30-- http://10.134.125.136:5555/export/auto_install/11_2_sparc/
solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2014-05-05 18:52:30 ERROR 404: Not Found.

Could not obtain http://10.134.125.136:5555/export/auto_install/11_2_sparc/solaris.zlib from
install server
Please verify that the install server is correctly configured and reachable from the client
```

이 문제는 다음 조건 중 하나로 유발될 수 있습니다.

- WAN 부트에 구성된 이미지 경로가 부정확합니다.
- 이미지 경로가 존재하지 않거나 불완전합니다.
- 권한 문제로 인해 액세스가 거부되었습니다.

DHCP 구성과 `installadm create-service`를 실행할 때 지정된 넷 이미지의 내용을 확인합니다. WAN 부트 구성을 확인합니다.

## 자동 설치 프로그램 시작되지 않음

클라이언트 시스템에서 Oracle Solaris OS를 설치할 때 설치를 시작하려면 부트 시 `install` 인수를 포함해야 합니다.

```
ok boot net:dhcp - install
```

`install` 부트 인수 없이 부트하면 SPARC 클라이언트가 자동 설치 부트 이미지로 부트하지만 설치가 시작되지 않습니다. 이 지점에서 자동 설치를 시작하는 방법은 [“명령줄에서 자동 설치 시작” \[241\]](#)을 참조하십시오.

## 잘못된 HMAC 값

SPARC AI 클라이언트를 부트한 직후 SPARC 콘솔에 `Invalid HMAC value`라는 메시지가 표시되고 시스템에서 `ok` 프롬프트를 반환할 경우 다음 조건 중 하나가 문제의 원인입니다.

- 인증을 통해 AI 클라이언트가 보안되지만 OBP 키를 설정하지 않았습니다. 클라이언트 펌웨어에서 OBP 키를 설정하여 해결합니다. 인증에 대한 자세한 내용은 [“자동 설치의 보안 수준 향상” \[106\]](#)을 참조하십시오. OBP 키 설정에 대한 자세한 내용은 [“보안 다운로드를 사용하여 SPARC 클라이언트 설치” \[224\]](#)를 참조하십시오.
- 해당 AI 클라이언트는 보안되지 않지만 OBP 키가 설정되어 있습니다. 클라이언트 펌웨어에서 OBP 키를 설정 해제하여 해결합니다. [“해싱 키 및 암호 키 재설정” \[224\]](#)을 참조하십시오.
- 클라이언트의 설치 서비스에 클라이언트 인증을 요구하는 정책이 있지만 클라이언트에 해당되는 자격 증명이 지정되었습니다. `require-client-auth` 정책이 있는 서비스의 모든 클라이언트에 사용할 수 있는 자격 증명이 있어야 합니다.

다음 단계에서는 문제를 파악하는 방법을 보여줍니다.

1. AI 서버에 대해 보안이 사용 안함으로 설정되지 않았는지 확인합니다. `installadm list -sv`를 사용하여 보안이 사용으로 설정되었는지 확인합니다.
2. 클라이언트의 설치 서비스에 대해 보안이 사용 안함으로 설정되지 않았는지 확인합니다. `installadm list -vn <svcname>`을 사용하여 보안이 사용 안함으로 설정되지 않았는지 확인합니다.
3. 클라이언트에서 사용자 정의 자격 증명을 사용하는 경우 `installadm list -ve <macaddr>`을 사용하여 펌웨어 키 값을 가져옵니다.
4. 클라이언트가 사용자 정의 클라이언트가 아닐 경우 `installadm list -vn default-sparc`를 사용하여 `default-sparc` 서비스에 정의된 펌웨어 키가 있는지 확인합니다.
5. `installadm list -vn <svcname>`으로 클라이언트의 서비스 정책을 확인합니다.
6. `default-sparc` 서비스에 대한 자격 증명이 없을 경우 `installadm list -sv` 명령을 사용하여 기본 클라이언트 자격 증명을 찾습니다. 기본 클라이언트 자격 증명이 있을 경우 기본 클라이언트에 대해 나열된 펌웨어 키를 사용합니다.
7. 기본 클라이언트 자격 증명이 없을 경우 `installadm list -vn default-sparc`를 사용하여 서비스 정책이 `require-server-auth`로 설정되었는지 확인합니다. 그렇다면 `installadm list -sv`에서 기본 클라이언트에 대해 나열된 펌웨어 키를 사용합니다.

## x86 네트워크 부팅 오류 및 가능한 원인

이 섹션은 네트워크를 통해 x86 클라이언트를 부트할 때 볼 수 있는 오류/문제와 가능한 원인을 설명합니다.

- [“DHCP 또는 ProxyDHCP 제공이 수신되지 않음” \[235\]](#)
- [“TFTP 오류 또는 GATEWAY 메시지 후에 시스템 중단” \[235\]](#)
- [“GRUB 메뉴 항목을 선택한 후에 시스템 중단” \[236\]](#)
- [“HTTP 요청 전송 결과에 403 사용 권한 없음 또는 404 찾을 수 없음” \[236\]](#)
- [“자동 설치 프로그램 시작되지 않음” \[237\]](#)

## DHCP 또는 ProxyDHCP 제공이 수신되지 않음

DHCP 서버가 x86 클라이언트의 요청에 응답하지 않으면 다음 메시지가 표시됩니다.

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation

CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
DHCP..... No DHCP or ProxyDHCP offers were received
PXE-MOF: Exiting Intel Boot Agent
```

시간 초과 메시지는 클라이언트가 DHCP 요청을 보냈는데 응답하지 않았음을 나타냅니다. 이 문제는 DHCP 구성 오류 때문인 것 같습니다. 클라이언트가 DHCP 서버에 올바르게 구성되었는지 여부를 확인합니다.

## TFTP 오류 또는 GATEWAY 메시지 후에 시스템 중단

DHCP 서버는 DHCP 응답의 일부로 초기 부트 프로그램의 IP 주소 및 위치를 제공합니다.

- 부트 프로그램이 존재하지 않으면 SI 클라이언트 부트를 계속할 수 없습니다. 다음과 같은 메시지가 표시됩니다.

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation

CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
TFTP.
PXE-T02: Access Violation
PXE-E3C: TFTP Error - Access violation
PXE-MOF: Exiting Intel Boot Agent
```

- 부트 프로그램이 존재하지만 잘못된 프로그램인 경우 이 메시지가 표시된 후에 SI 클라이언트가 중단됩니다.

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation

CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
```

## GRUB 메뉴 항목을 선택한 후에 시스템 중단

클라이언트가 초기 부트를 수행할 수 있지만 커널을 부트할 수 없는 경우 GRUB 메뉴에서 항목을 선택한 후에 시스템이 중단됩니다.

시 서버에서 이 클라이언트의 grub.cfg 또는 menu.lst 파일이 유효한 부트 아카이브를 가리키는지 여부를 확인합니다. installadm list로 표시된 이미지 경로에 대해 df -k에서 발췌한 이 샘플에 표시된 대로, 서버 이미지의 부트 디렉토리는 /etc/netboot 디렉토리 아래에 마운트된 루프백이어야 합니다.

```
Filesystem      1K-blocks      Used Available Use% Mounted on
/export/auto_install/solaris11_2-i386
    92052473  36629085  55423388  40% /etc/netboot/default-i386
/export/auto_install/solaris11_2-i386
    92052473  36629085  55423388  40% /etc/netboot/solaris11_2-i386
```

## HTTP 요청 전송 결과에 403 사용 권한 없음 또는 404 찾을 수 없음

시 서버에서 설치 프로그램 중 하나에 액세스할 수 없거나 /etc/netboot 아래의 grub.cfg 또는 menu.lst 파일에 지정된 위치에 존재하지 않으면 클라이언트가 부트할 수 있지만 해당 파일을 다운로드할 수 없습니다. 어떤 파일이 문제를 일으키는지 나타내는 오류 메시지가 표시됩니다. 예를 들어, x86 클라이언트의 다음 출력에서 solaris.zlib 파일이 지정된 위치에 존재하지 않습니다.

```
SunOS Release 5.11 Version 11.2 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2015-05-05 20:02:26-- http://10.134.125.136:5555/export/auto_install/solaris11_2-i386/
solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2015-05-05 20:02:26 ERROR 404: Not Found.
```

```
Could not obtain http://10.134.125.136:5555/export/auto_install/solaris11_2-i386/solaris.zlib
from install server
Please verify that the install server is correctly configured and reachable from the client
```

```
Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
Console login service(s) cannot run
```

installadm create-service 명령을 실행할 때 지정된 대상 디렉토리의 내용을 확인합니다.

## 자동 설치 프로그램 시작되지 않음

네트워크를 통해 부트하는 설치의 x86 클라이언트 시스템에 Oracle Solaris OS를 설치할 때 자동 설치를 시작하려면 GRUB 부트 메뉴에서 두번째 항목을 선택해야 합니다. 일반적으로 메뉴 항목은 다음과 같이 표시됩니다.

```
Oracle Solaris 11.2 Text Installer and command line
Oracle Solaris 11.2 Automated Install
```

첫번째 GRUB 메뉴 항목을 선택하거나 프롬프트의 시간 초과가 허용된 경우 시스템이 자동 설치 부트 이미지로 부트하지만 설치가 시작되지 않습니다. 이 지점에서 자동 설치를 시작하는 방법은 [“명령줄에서 자동 설치 시작” \[241\]](#)을 참조하십시오.

## SPARC 및 x86 오류 메시지

다음 오류는 SPARC 및 x86 설치 양쪽에 공통됩니다.

- [“Automated Installation Failed 메시지” \[237\]](#)
- [“IPS 서버를 사용할 수 없음” \[237\]](#)
- [“패키지를 찾을 수 없음” \[239\]](#)

### Automated Installation Failed 메시지

설치 중 실패가 발생하면 다음 메시지가 표시됩니다.

```
21:43:34 Automated Installation Failed. See install log at /system/volatile/install_log
Automated Installation failed
Please refer to the /system/volatile/install_log file for details
Jul 6 21:43:34 solaris svc.startd[9]: application/auto-installer:default failed fatally:
transitioned to maintenance (see 'svcs -xv' for details)
```

### IPS 서버를 사용할 수 없음

설치 클라이언트가 Oracle Solaris OS를 설치하려면 AI 매니페스트에 정의된 IPS 패키지 저장소에 연결해야 합니다. 클라이언트가 패키지 저장소에 액세스할 수 없으면 설치를 실패하고 application/auto-installer 서비스가 유지 관리 모드로 진입합니다. 다음 출력은 콘솔에 표시되는 내용의 예입니다.

```
15:54:46 Creating IPS image
15:54:46 Error occurred during execution of 'generated-transfer-1341-1' checkpoint.
15:54:47 Failed Checkpoints:
15:54:47 generated-transfer-1341-1
15:54:47
15:54:47 Checkpoint execution error:
```

```

15:54:47
15:54:47      Framework error: code: 6 reason: Couldn't resolve host 'pkg.example.com'
15:54:47      URL: 'http://pkg.example.com/solaris/release/versions/0/'.
15:54:47
15:54:47      Automated Installation Failed. See install log at /system/volatile/install_log
Automated Installation failed
Please refer to the /system/volatile/install_log file for details
Aug 21 15:54:47 line2-v445 svc.startd[8]: application/auto-installer:default failed fatally:
transitioned to maintenance (see 'svcs -xv' for details)
...
SUNW-MSG-ID: SMF-8000-YX, TYPE: defect, VER: 1, SEVERITY: major
EVENT-TIME: Wed Aug 21 15:54:47 UTC 2013
PLATFORM: SUNW,Sun-Fire-V445, CSN: -, HOSTNAME: line2-v445
SOURCE: software-diagnosis, REV: 0.1
EVENT-ID: c8a5b809-ece4-4399-9646-d8c64d78aac7
DESC: A service failed - a start, stop or refresh method failed.
AUTO-RESPONSE: The service has been placed into the maintenance state.
IMPACT: svc:/application/auto-installer:default is unavailable.
REC-ACTION: Run 'svcs -xv svc:/application/auto-installer:default' to determine the generic
reason
why the service failed, the location of any logfiles, and a list of other services impacted.
Please
refer to the associated reference document at http://support.oracle.com/msg/SMF-8000-YX for
the latest service
procedures and policies regarding this diagnosis.

```

/system/volatile/install\_log 파일에서 다음과 비슷한 메시지를 확인합니다.

```

TransportFailures: Framework error: code: 6 reason: Couldn't resolve host
'pkg.example.com'
URL: 'http://pkg.example.com/solaris/versions/0/'

TransportFailures: Framework error: code: 7 reason: Failed connect to
pkg.example.com:80; Connection refused
URL: 'http://pkg.example.com/solaris/versions/0/'

TransportFailures: http protocol error: code: 404 reason: Not Found
URL: 'http://pkg.oracle.com/mysolaris/versions/0/'

```

보고 있는 메시지에 따라 다음 가능한 해결법을 시도하십시오.

- ping을 사용하여 실패한 클라이언트 시스템에서 패키지 서버에 연결해 봅니다.
- DNS를 사용 중인 경우 DNS가 SI 클라이언트에 올바르게 구성되었는지 확인합니다. “DNS 확인” [229]을 참조하십시오.
- 로컬 저장소를 사용 중인 경우 저장소에서 모든 클라이언트에 액세스 가능한지 확인합니다. “Oracle Solaris 11.2 패키지 저장소 복사 및 만들기”의 3 장, “저장소에 대한 액세스 제공”을 참조하십시오.
- SI 매니페스트의 URI에 철자 오류가 없는지 확인합니다.
- 다음과 같은 명령을 사용하여 패키지 저장소가 유효한지 여부를 확인합니다.

```
$ pkg list -g http://pkg.example.com/solaris/ entire
```

카탈로그를 새로 고치거나 인덱스를 재구축해야 할 수 있습니다.

## 패키지를 찾을 수 없음

SI 매니페스트에 지정된 패키지 중 하나를 IPS 저장소에서 찾을 수 없는 경우 디스크에 패키지를 설치하기 전에 설치 프로그램을 실패합니다. 다음 예에서 설치 프로그램이 IPS 저장소에서 mypkg 패키지를 찾을 수 없습니다. 다음 출력은 콘솔에 표시되는 내용의 예입니다.

```
14:04:02    Failed Checkpoints:
14:04:02
14:04:02          generated-transfer-1230-1
14:04:02
14:04:02    Checkpoint execution error:
14:04:02
14:04:02          The following pattern(s) did not match any allowable packages.  Try
14:04:02          using a different matching pattern, or refreshing publisher information:
14:04:02
14:04:02          pkg:/mypkg
14:04:02
14:04:02    Automated Installation Failed.  See install log at /system/volatile/install_log
```

다음 출력은 /system/volatile/install\_log 로그 파일의 일부 예입니다.

```
PlanCreationException: The following pattern(s) did not match any allowable packages.
Try using a different matching pattern, or refreshing publisher information:
```

```
pkg:/mypkg
```

문제의 패키지가 유효한 패키지인지 여부를 확인합니다. 이 패키지를 서로 다른 IPS 저장소에서 사용할 수 있는 경우 다른 publisher 요소를 source 요소에 추가하여 SI 매니페스트에 해당 IPS 저장소를 추가합니다.

## 보안 클라이언트의 부트 오류

SI 클라이언트를 부트할 때 다음과 유사한 메시지가 나타나면 TLS 인증서가 아직 유효하지 않다는 의미입니다.

```
SSL3_GET_RECORD:wrong version number - secure HTTPS GET REQUEST to unsecured HTTP port
```

클라이언트의 시스템 시간이 인증서가 생성된 시간보다 먼저여서 발생한 문제일 수 있습니다. 클라이언트의 시스템 시간을 확인하십시오. 보안 자격 증명을 생성하고 지정하는 방법에 대한 자세한 내용은 [“자동 설치의 보안 수준 향상” \[106\]](#)을 참조하십시오.

## 보안 관련 SI 오류

[“자동 설치의 보안 수준 향상” \[106\]](#)에서 설명한 대로 SI 서버와 클라이언트를 보안한 경우 클라이언트를 부트하거나 설치하는 동안 문제가 발생하면 다음 단계에 따라 인증서 오류를 확인해 보십시오.

- 클라이언트의 /var/ai/image-server/logs/에서 Apache access\_log 및 error\_log를 확인합니다.
- AI 클라이언트의 콘솔에 로그인합니다. /tmp/install\_log 파일과 /system/volatile/의 SMF 로그를 검사합니다.
- AI 클라이언트에서 부트 아카이브가 로드된 후 인증이 실패할 경우 이미지 파일, AI 매니페스트 또는 시스템 구성 프로파일을 가져오려고 시도하면 일시적으로 네트워크가 중단될 수 있습니다. AI 서버가 제대로 작동하는지 확인하고 클라이언트 설치를 다시 시작합니다.
- openssl s\_client 명령을 사용하여 연결을 테스트합니다.

```
$ openssl s_client -key client-key -cert client-certificate \
-CAcert server-CA-certificate -connect AI-server-address:port
```

- installadm list -s -v 명령을 사용하여 AI 서버에서 보안의 사용 또는 사용 안함 상태를 표시합니다. 예 8-40. “서버 보안 정보 나열”을 참조하십시오.
- installadm list -v -n <svcname> 명령을 사용하여 클라이언트의 서비스 정책을 확인합니다

- CA 인증서에 대해 지정된 자격 증명을 확인합니다. -k 및 -c 옵션을 installadm list 하위 명령에 사용하여 지정된 키와 인증서를 나열합니다. diff와 같은 문자 비교 유틸리티를 사용하여 키 및 인증서를 필요한 키 및 인증서와 비교합니다.
- 클라이언트의 /var/ai/ai-webserver/tls.key/server.key에서 문자암호가 제거되었는지 확인합니다. X.509 개인 키 파일에서 문자암호가 제거되어야 합니다.
- 다음 예제에 표시된 대로 wget 명령을 사용하여 적절한 키, 인증서 및 CA 인증을 사용하는 AI 이미지의 파일을 인출합니다.

```
$ wget --private-key=client-key --certificate=client-certificate \
--ca-certificate=server-CA-certificate \
http://AI-server-address:5555/path-to-file-in-image
```

## 설치 시작 없이 설치 환경 부트

다음 방식 중 하나를 사용하여 자동 설치를 시작하지 않고 설치 환경을 부트합니다.

네트워크를 통한 SPARC 클라이언트 부팅

다음 명령을 사용하여 자동 설치를 시작하지 않고 네트워크를 통해 SPARC 클라이언트를 부트합니다.

```
ok boot net:dhcp
```

부트 인수로 install 플래그를 지정하지 마십시오.

매체에서 SPARC 클라이언트 부팅

다음 명령을 사용하여 설치를 시작하지 않고 매체에서 SPARC 클라이언트를 부트합니다.

```
ok boot cdrom
```

부트 인수로 `install` 플래그를 지정하지 마십시오.

네트워크를 통한 x86 클라이언트 부팅

네트워크를 통해 부트하는 x86 설치의 경우 다음 GRUB 메뉴가 표시됩니다.

```
Oracle Solaris 11.2 Text Installer and command line
Oracle Solaris 11.2 Automated Install
```

기본 항목인 "Text Installer and command line"은 핸즈프리 자동 설치를 시작하지 않고 이미지를 부트합니다.

항목의 커널 라인에 `install=true` 부트 등록 정보가 지정되지 않았는지 확인합니다.

매체에서 x86 클라이언트 부팅

매체에서 x86 시스템을 부트하고 설치를 시작하지 않으려면 GRUB 메뉴를 편집하고 부트할 항목의 커널 라인에서 `install=true` 부트 등록 정보를 제거합니다.

일반적으로 x86 설치의 경우 `install=true` 부트 등록 정보가 부트 중인 GRUB 항목의 커널 라인에 지정되면 설치가 자동으로 시작됩니다. 자동 설치를 시작하지 않고 x86 기반 시스템을 부트하려면 GRUB 부트 항목이 `install=true` 부트 등록 정보를 지정하지 않는지 확인합니다. 등록 정보가 지정된 경우 [“Oracle Solaris 11.2 시스템 부트 및 종료”](#)의 [“부트 시 GRUB 메뉴를 편집하여 커널 인수 추가”](#)에 설명된 대로 부트 항목을 편집하고 등록 정보를 제거합니다.

클라이언트를 부트할 때 [“명령줄에서 자동 설치 시작” \[241\]](#)에 나온 대로 메뉴가 표시됩니다. 이 메뉴를 사용하여 시스템을 조사하거나 설치합니다.

## 명령줄에서 자동 설치 시작

설치를 시작하지 않는 부트 옵션을 선택한 경우 다음 메뉴가 표시됩니다.

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently xterm)
5 Reboot
```

Please enter a number [1]:

셸을 열려면 옵션 3을 선택합니다.

셸이 실행되면 다음 명령을 사용하여 자동 설치를 시작합니다.

```
$ svcadm enable svc:/application/manifest-locator:default  
$ svcadm enable svc:/application/auto-installer:default
```

## 부 IV

### 관련 작업 수행





## Oracle Configuration Manager 작업

---

이 장에서는 Oracle Configuration Manager의 개요와 Oracle Solaris 릴리스를 실행하는 시스템에서 이 서비스를 사용하는 방법에 대한 지침을 제공합니다.

### Oracle Configuration Manager 소개

Oracle Configuration Manager는 시스템의 구성 정보를 수집하고 Oracle 저장소에 업로드하는 데 사용됩니다. 이 정보 수집기를 서버의 모든 제품에 대한 정보를 수집하는 중앙 수집기로 구성하거나, 개별 수집 사이트에서 정보를 수집하도록 구성할 수 있습니다. 자세한 내용은 [“Oracle Configuration Manager 중앙 수집기 정보” \[246\]](#)를 참조하십시오.

고객 지원 담당자는 이 정보를 사용하여 더 나은 서비스를 제공할 수 있습니다. Oracle Configuration Manager를 사용할 경우의 몇 가지 이점은 다음과 같습니다.

- 지원 문제 해결을 위한 시간이 단축됩니다.
- 사전 예방적으로 문제를 방지합니다.
- 모범 사례 및 Oracle 기술 자료 문서에 더 쉽게 액세스합니다.
- 고객 비즈니스 요구를 더 잘 이해하고 일관된 대응과 서비스를 제공합니다.

Oracle Configuration Manager는 연결 및 연결 해제 모드 중 하나로 실행할 수 있습니다. 연결 해제 모드는 시스템이 인터넷에 연결되지 않았고 Oracle Support Hub를 구성할 수 없는 경우에만 필요합니다. 이 모드에서는 수동으로 구성 정보를 수집하고 서비스 요청을 통해 정보를 Oracle에 업로드할 수 있습니다.

연결 모드에서는 다음과 같이 Oracle Configuration Manager를 여러 네트워크 구성에서 실행할 수 있습니다.

- 시스템을 인터넷에 직접 연결할 수 있습니다.
- 시스템을 프록시 서버를 통해 인터넷에 연결할 수 있습니다.
- 시스템은 인터넷에 직접 액세스할 수 없지만 인트라넷 프록시 서버에 액세스할 수 있고 여기서 Oracle Support Hub를 통해 인터넷에 연결됩니다.
- 시스템은 인터넷에 직접 액세스할 수 없지만 Oracle Support Hub에 액세스할 수 있고 여기서 프록시 서버를 통해 인터넷에 연결됩니다.

Oracle Configuration Manager를 설정 및 구성하는 방법에 대한 자세한 내용은 [Oracle Configuration Manager Installation and Administration Guide](#)를 참조하십시오. 이 문

서의 나머지 부분에서는 Oracle Configuration Manager와 연관된 Oracle Solaris 작업에 초점을 맞춥니다.

---

**참고** - 프록시 또는 Oracle Support Hub를 사용하도록 Oracle Configuration Manager를 구성하려면 대화식 모드에서 `configCCR` 명령을 실행해야 합니다. 자세한 내용은 [Oracle Support Hub](#)를 참조하십시오.

---

Oracle Solaris 11을 설치하는 동안 Oracle 저장소에 익명 연결의 설정이 시도됩니다. 연결이 설정되면 다른 정보를 묻지 않고 이 연결을 통해 설치가 진행됩니다. 이상적인 경우라면 시스템이 완전히 설치된 후에 등록 또는 네트워크 구성을 변경해야 합니다. 익명으로 로드된 데이터는 어떤 조직에도 연결되어 있지 않습니다. 소프트웨어가 Oracle 저장소에 연결할 수 없는 경우 시스템을 수동으로 등록한 다음 Oracle Configuration Manager 서비스를 사용하여 설정할 수 있습니다.

## Oracle Configuration Manager 중앙 수집기 정보

Oracle Solaris 운영 체제의 일부로 설치된 Oracle Configuration Manager 수집기는 중앙 수집기로 구성 및 지정됩니다. 개인화된 지원 경험, 신속한 지원 문제 해결, 사전 예방적 문제 방지와 같은 Oracle Configuration Manager 수집기의 이점을 활용하려면 각 Oracle 설치마다 구성 데이터를 수집하여 업로드해야 합니다. 이것은 대개 Oracle 홈에 설치된 수집기의 작업입니다. 그러나 Oracle 홈의 수집기가 구성되지 않았거나 연결 해제되었을 수 있습니다. 중앙 수집기의 목적은 이러한 Oracle 홈을 수집하여 고유의 MOS(My Oracle Support) 자격 증명 하에 업로드하는 것입니다. 중앙 수집기의 특징은 다음과 같습니다.

- 중앙 수집기는 다음을 수집합니다.
  - 상주한 Oracle 홈
  - 구성된 수집기가 없는 호스트의 Oracle 홈
  - 수집기가 연결 해제 모드에 있는 Oracle 홈
  - 수집기가 인증된 등록을 받은 Oracle 홈

Oracle 홈의 수집기가 `ORACLE_CONFIG_HOME` 지정으로 구성된 경우 중앙 수집기가 해당 홈을 수집하지 않습니다.

- `root` 역할을 사용하여 `setupCCR` 및 `configCCR` 명령에 `-c` 옵션을 지정하면 중앙 수집기로 사용할 수집기 설치를 지정할 수 있습니다. 그 이후 `configCCR` 명령을 `-c` 옵션 없이 사용하면 중앙 수집기 지정이 철회됩니다. `setupCCR` 및 `configCCR` 명령을 `-c` 옵션과 함께 실행하면 해당 수집기가 중앙 수집기로 지정됩니다. Oracle Solaris 운영 체제의 일부로 설치된 수집기는 `root` 권한으로 설치되므로 호스트의 중앙 수집기로 작동합니다.
- Oracle Universal Installer 중앙 인벤토리는 중앙 수집기가 일련의 수집 후보 Oracle 홈을 얻게 되는 소스입니다. 설명서에 기술된 대로 Installer에서 중앙 인벤토리를 검색합니다. Oracle Solaris 운영 체제의 Installer 중앙 인벤토리 포인터의 기본 위치는 `/var/opt/oracle/oraInst.loc`입니다. Oracle 설치 인벤토리를 다른 위치에 놓도록 선택하면 중앙 인벤토리가 정보를 찾아서 수집할 수 없습니다.

- 이 릴리스에서는 Oracle Solaris OS의 구성 정보 외에도, 단일 Oracle 데이터베이스와 Oracle WebLogic을 사용하는 Oracle Fusion Middleware 기반 제품이 중앙 수집기를 통해 수집됩니다.
- Oracle 홈에서 중앙 수집기가 수집한 모든 구성 정보는 중앙 수집기의 My Oracle Support 자격 증명을 사용하여 업로드됩니다.

## Oracle Configuration Manager 관리(작업)

다음 작업 맵에서는 Oracle Solaris 시스템에서 Oracle Configuration Manager를 사용하는 것과 관련된 여러 절차가 포함됩니다.

작업	설명	수행 방법
Oracle Configuration Manager 서비스를 사용으로 설정합니다.	구성 변경을 수행한 후 Oracle Configuration Manager 서비스를 사용으로 설정합니다.	<a href="#">Oracle Configuration Manager 서비스를 사용으로 설정하는 방법 [247]</a>
Oracle Configuration Manager 서비스를 사용 안함으로 설정합니다.	중요한 구성 변경을 수행하기 전에 Oracle Configuration Manager 서비스를 사용 안함으로 설정합니다.	<a href="#">Oracle Configuration Manager 서비스를 사용 안함으로 설정하는 방법 [247]</a>
Oracle 저장소에 시스템을 수동으로 등록합니다.	등록 자격 증명을 변경합니다.	<a href="#">Oracle 저장소에 시스템을 수동으로 등록하는 방법 [248]</a>
데이터 수집 시간을 변경	데이터 수집 빈도와 시간을 재설정합니다.	<a href="#">Oracle Configuration Manager의 데이터 수집 시간 또는 빈도를 변경하는 방법 [248]</a>

### ▼ Oracle Configuration Manager 서비스를 사용으로 설정하는 방법

1. 관리자가 됩니다.  
자세한 내용은 “[How to Use Your Assigned Administrative Rights](#)” in “[Oracle Solaris 11.2 Administration: Security Services](#)”을 참조하십시오.
2. Oracle Configuration Manager 서비스를 사용으로 설정합니다.

```
# svcadm enable system/ocm
```

### ▼ Oracle Configuration Manager 서비스를 사용 안함으로 설정하는 방법

1. 관리자가 됩니다.

자세한 내용은 “[How to Use Your Assigned Administrative Rights](#)” in “[Oracle Solaris 11.2 Administration: Security Services](#)”을 참조하십시오.

2. Oracle Configuration Manager 서비스를 사용 안함으로 설정합니다.

```
# svcadm disable system/ocm
```



주의 - Oracle Solaris 시스템에서 `emCCR stop` 명령을 실행하지 마십시오. 서비스에 대한 모든 변경은 Oracle Solaris의 SMF(서비스 관리 기능)를 사용하여 수행해야 합니다.

## ▼ Oracle 저장소에 시스템을 수동으로 등록하는 방법

1. 관리자가 됩니다.

자세한 내용은 “[How to Use Your Assigned Administrative Rights](#)” in “[Oracle Solaris 11.2 Administration: Security Services](#)”을 참조하십시오.

2. 사용자 등록을 변경합니다.

```
# configCCR
```

Oracle Configuration Manager 소프트웨어에 전자 메일 계정과 암호를 묻는 메시지가 표시됩니다. 일반적으로 My Oracle Support ID와 연관된 전자 메일 계정을 사용합니다.

시스템이 등록 서버와 직접 통신할 수 있는 경우 통신이 수행됩니다. 그렇지 않은 경우 Oracle Support Hub의 URL을 묻는 메시지가 표시됩니다. 사이트에 사용되는 URL이 있는 경우 여기에서 지정합니다. Oracle Support Hub의 URL을 지정하지 않거나 여전히 등록 서버와 통신할 수 없는 경우 네트워크 프록시를 묻는 메시지가 표시됩니다.

등록이 완료된 후 데이터 수집이 시작됩니다.

참조 자세한 내용은 [configCCR\(1M\)](#) 매뉴얼 페이지 또는 [Oracle Configuration Manager Installation and Administration Guide](#)를 참조하십시오. `configCCR` 명령을 사용하는 대화식 세션에 대한 전체 예는 [configCCR](#) 페이지를 참조하십시오.

## ▼ Oracle Configuration Manager의 데이터 수집 시간 또는 빈도를 변경하는 방법

1. 관리자가 됩니다.

자세한 내용은 “[How to Use Your Assigned Administrative Rights](#)” in “[Oracle Solaris 11.2 Administration: Security Services](#)”을 참조하십시오.

2. 데이터 수집 빈도를 재설정합니다.

이 예는 매주 월요일 아침 6시에 발생하도록 데이터 수집 시간을 재설정합니다.

```
# emCCR set collection_interval=FREQ=WEEKLY\; BYDAY=MON\; BYHOUR=6
```

참조 자세한 내용은 [emCCR\(1M\)](#) 매뉴얼 페이지 또는 [Oracle Configuration Manager Installation and Administration Guide](#)를 참조하십시오.



## Device Driver Utility 사용

---

Oracle DDU(Device Driver Utility)는 최신 릴리스가 설치된 시스템에서 감지된 장치를 지원하는지 여부를 보고합니다.

### 장치 드라이버 유틸리티 개요

Device Driver Utility는 설치된 시스템의 장치와 이러한 장치가 관리하는 드라이버에 대한 정보를 제공합니다. DDU는 현재 부트된 운영 체제에 시스템에서 감지된 모든 장치에 대한 드라이버가 있는지 여부를 보고합니다. 장치에 연결된 드라이버가 없으면 Device Driver Utility에서 설치할 드라이버 패키지를 권장합니다.

Device Driver Utility를 사용하여 HCL(<http://www.oracle.com/webfolder/technetwork/hcl/index.html>)에 시스템 정보를 제출할 수도 있습니다. 그러면 시스템 및 해당 구성 요소가 "보고 사용"으로 HCL에 나열됩니다.

이 섹션에서는 다음 작업을 설명합니다.

- [Device Driver Utility 시작 방법 \[251\]](#)
- [누락된 드라이버 설치 방법 \[252\]](#)
- [HCL에 시스템 나열 방법 \[254\]](#)

### ▼ Device Driver Utility 시작 방법

Device Driver Utility는 설치된 시스템을 부트할 때 자동으로 실행됩니다. Oracle Solaris OS를 설치한 후 Device Driver Utility를 수동으로 시작할 수도 있습니다.

- 다음 방법 중 하나를 사용하여 Device Driver Utility를 시작합니다.
  - Oracle Solaris 텍스트 설치 프로그램 이미지를 부트합니다.  
텍스트 설치 프로그램에서 Device Driver Utility를 시작하려면 초기 메뉴에서 추가 드라이버 설치를 선택합니다.

---

**참고** - 텍스트 설치 프로그램을 부트할 때 자동 네트워킹이 기본적으로 설정됩니다. DHCP를 사용하는 경우 Device Driver Utility를 사용하여 추가로 네트워크를 설정할 필요가 없습니다. DHCP를 사용하지 않는 경우 초기 메뉴에서 셀 옵션을 선택한 다음 Device Driver Utility를 사용하기 전에 적절한 명령을 사용하여 네트워크 설정을 수동으로 구성합니다.

---

■ **설치된 시스템에서 Device Driver Utility를 시작합니다.**

설치된 시스템의 데스크탑에서 장치 드라이버 유틸리티를 시작하려면 주 메뉴에서 Applications(응용 프로그램) -> System Tools(시스템 도구) -> Device Driver Utility(장치 드라이버 유틸리티)를 선택합니다.

Device Driver Utility에서 시스템을 스캔하고 감지되는 장치 목록을 표시합니다. 감지되는 각 장치에 대한 목록에는 제조업체, 모델 및 현재 장치를 관리하는 드라이버 이름과 같은 정보가 표시됩니다.

다음 순서 유틸리티에서 연결된 드라이버가 없는 장치를 감지하면 해당 장치가 장치 목록에서 선택됩니다. 장치에 대한 자세한 정보를 표시하고 누락된 드라이버를 설치할 수 있습니다. [누락된 드라이버 설치 방법 \[252\]](#)을 참조하십시오.

## ▼ 누락된 드라이버 설치 방법

유틸리티에서 연결된 드라이버가 없는 장치를 감지하면 해당 장치가 장치 목록에서 선택됩니다. 장치에 대한 자세한 정보를 표시하고 누락된 드라이버를 설치할 수 있습니다.

1. **Device Driver Utility 목록에서 장치 이름을 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 세부 정보 표시를 선택합니다.**

장치 및 드라이버 세부 정보 창이 표시됩니다. 장치 이름, 공급업체 이름, 노드 이름, 드라이버 이름 및 기타 장치에 대한 세부 정보가 표시됩니다.

2. **누락된 드라이버에 대한 자세한 정보를 표시하려면 선택한 장치에 대한 정보 링크를 누릅니다.**

현재 장치를 관리하는 드라이버가 없으면 장치 목록의 드라이버 열에 해당 장치의 드라이버 상태가 표시됩니다. 누락된 드라이버는 다음 범주 중 하나에 속하도록 표시됩니다.

- IPS - 구성된 IPS 패키지 저장소의 하나
- SVR4 - SVR4(System V Revision 4) 패키지
- DU - DU 패키지
- UNK - Device Driver Utility는 이 장치의 Oracle Solaris 드라이버를 찾을 수 없습니다.

---

작은 정보 - 추가 정보를 보려면 도움말 버튼을 누릅니다.

---

3. **누락된 드라이버를 설치합니다.**

■ IPS 드라이버의 경우:

- a. 표의 해당 행에서 정보 링크를 눌러 장치 드라이버가 포함된 IPS 패키지에 대한 정보를 표시합니다.  
패키지 라디오 버튼의 텍스트 필드가 관련 패키지 정보로 채워져 있습니다. 올바른 게시자가 지정됩니다.
- b. 패키지를 설치하려면 설치 버튼을 누릅니다.

■ 정보 링크에 구성되지 않은 게시자의 IPS 패키지가 나열된 경우:

- i 저장소 메뉴에서 저장소 추가를 선택합니다.  
저장소 관리자 창이 표시됩니다.
- ii 새 저장소의 이름과 URI를 추가하고 추가를 누릅니다.

■ 패키지 필드가 채워지지 않으면 정보 링크에서 IPS 패키지 이름을 입력한 다음 설치를 누릅니다.

■ SVR4 또는 DU 드라이버의 경우:

- 패키지 URL이 제공되면 파일/URL 필드에 해당 URL을 입력하고 설치를 누릅니다.
- 시스템에 패키지 복사본이 있는 경우 찾아보기 버튼을 누르고 패키지를 선택한 다음 설치를 누릅니다.

■ 드라이버 상태가 UNK로 표시되는 경우:

- a. 이 드라이버에서 관리할 장치의 이름을 선택합니다.
- b. 패키지 필드나 파일/URL 필드에 관련 패키지 정보를 입력한 다음 설치를 누릅니다.
- c. (옵션) 장치에서 작동하는 드라이버에 대한 정보를 공유하려면 제출 버튼을 누릅니다.

다음 순서 Device Driver Utility로 작업하는 경우 특정 장치를 작업하면서 찾은 드라이버에 대한 정보를 다른 사용자와 공유할 수 있습니다. [HCL에 시스템 나열 방법 \[254\]](#)을 참조하십시오.

## ▼ HCL에 시스템 나열 방법

특정 장치를 작업하면서 찾은 드라이버에 대한 정보를 다른 사용자와 공유할 수 있습니다.

1. 장치 드라이버 유틸리티를 시작합니다.

[Device Driver Utility 시작 방법 \[251\]](#)을 참조하십시오.

2. 시스템 및 구성 요소를 "보고 사용"으로 HCL에 나열하려면 제출 버튼을 누릅니다.

하드웨어 호환성 목록에 정보 제출 창이 열립니다. 이 창에는 시스템에 대해 수집된 모든 정보가 표시됩니다.

a. 시스템 유형을 선택합니다.

b. 자동으로 채워지지 않은 필드에 적절한 정보를 입력합니다.

- 제조업체 이름 - 시스템 제조업체(예: Toshiba, Hewlett-Packard 또는 Dell)의 이름입니다.

- 전체 모델 번호.

BIOS/펌웨어 제조업체는 시스템을 부트하는 동안 일반적으로 표시되는 BIOS 설정 화면에 있는 정보입니다.

- CPU 유형 - CPU 제조업체 이름입니다.

c. 이름 및 전자 메일 주소를 제공합니다.

d. 일반 메모 필드에 설명을 추가하고 저장을 누릅니다. 저장된 파일을 `device-detect-feedback_ww@oracle.com`으로 보냅니다.

## 색인

---

### 번호와 기호

/etc/auto\_home 파일, 170

/etc/passwd 파일, 170

/etc/resolv.conf 구성 파일, 229

/etc/svc/profile/site

정보 제거, 64

/etc/svc/profile/sysconfig/site-profile.tar 파일, 64

/system/volatile

AI 실패 문제 해결, 239

/system/volatile/install\_log 로그 파일, 220

/system/volatile/install\_log 파일, 229

자동 설치, 61

/tmp/install\_log

AI 실패 문제 해결, 239

/usr/bin/ai-wizard, 158

/usr/sbin/configCCR 명령

수동 등록, 248

/usr/sbin/emCCR 명령

데이터 수집 변경, 248

/var/ai/image-server/log 파일, 229

/var/ai/image-server/logs

AI 실패 문제 해결, 239

/var/log/install/install\_log파일

자동 설치, 61

16진수 MAC 주소

조건 키워드, 136

AI

개요, 73

사용 사례, 80

설치 사용자 정의, 133

AI 매니페스트

AI 매니페스트 마법사로 만들기, 158

개요, 76

기본 AI 매니페스트, 164

기존 매니페스트 수정, 139

매니페스트 검증, 129

매니페스트 복사, 132

매니페스트 선택 조건, 135

매니페스트 업데이트, 129

사용자 정의 IPS 패키지 설치, 215

선택 알고리즘, 134

설치 서비스에 추가, 104

설치 서비스에서 삭제, 130

설치 시작 전에 변경, 139

영역 configuration 요소, 197

예제

iSCSI 대상, 161

RAID 구성, 161

SVR4 패키지 설치, 162

다중 SVR4 패키지, 163

클라이언트 설치 타임에 만들기 살펴볼 내용 파생된 매니페스트

AI 매니페스트 마법사

AI 매니페스트 만들기, 158

사용 안함, 158

서버에서 파일 저장, 159

AI 보안

개요, 77

AI 사용자 정의, 133

AI 서버

계획, 79

관리자 권한, 90

구성, 91

멀티캐스트 DNS, 93

멀티홈, 94

보안 웹 서버 호스트 포트, 94

웹 서버 파일 디렉토리, 95

웹 서버 호스트 포트, 94

인증, 106

구성 요소, 73

보안 인증서 및 키, 107

- 설정, 89
- 설정 작업 요약, 89
- 요구 사항, 89
- 인증, 107
- 인증서 삭제, 116
- AI 설치 서비스
  - 설치 서비스에서 클라이언트 삭제, 222
  - 클라이언트 연관, 221
- AI 시스템 구성 프로파일
  - 설치 서비스에 추가, 166
  - 프로파일 검증, 166
  - 프로파일 만들기, 165
  - 프로파일 선택 조건, 135
- AI 시작되지 않음
  - SPARC 설치 문제 해결, 233
  - x86 설치 문제 해결, 237
- AI 클라이언트
  - /system/volatile/install\_log 파일, 229
  - CA 인증서 삭제, 116
  - hostname 조건 키워드, 136
  - SPARC 설치 문제 해결, 231
  - x86 설치 문제 해결, 234
  - 개요, 73
  - 명령줄에서 설치 시작, 241
  - 보안 인증서 삭제, 116
  - 부트 오류
    - 문제 해결, 230
  - 선택 조건, 136
  - 설치 문제 해결, 229
  - 자동 설치를 시작하지 않고 부트, 240
- AI 클라이언트 구성 살펴볼 내용 시스템 구성
- AI 클라이언트 설치
  - /system/volatile/install\_log 로그 파일, 220
  - 개요, 219
  - ssh 명령을 사용하여 모니터, 222
  - 네트워크 부트
    - network-boot-arguments OBP 변수, 225
    - SPARC WAN 부트 지원, 220
  - 네트워크 부팅
    - SPARC 클라이언트, 224
    - x86 클라이언트, 226
  - 보안, 112
  - 설치 메시지
    - 설치 성공, 227
    - 설치 시작, 227
  - 클라이언트 요구 사항, 220
- AI(자동 설치 프로그램) 살펴볼 내용 AI
  - aimanifest 명령
    - add 하위 명령, 145
    - load 하위 명령, 145
    - set 하위 명령, 145
    - validate 하위 명령, 157
  - aiuser 역할, 143
  - all\_services 등록 정보 그룹
    - exclude\_networks 속성, 94
    - manage\_dhcp 등록 정보, 97
    - networks 등록 정보, 94
    - webserver\_files\_dir 등록 정보, 95
    - webserver\_secure\_files\_dir 등록 정보, 95
  - Apache 로그 파일
    - AI 실패 문제 해결, 239
  - arch 조건 키워드, 136
  - ASR 살펴볼 내용 Oracle Auto Service Request Automated Installation Failed
    - 문제 해결, 237
  - boot\_archive 파일
    - SPARC 설치 문제 해결, 232
  - c 옵션
    - sysconfig unconfigure 명령, 65
  - CA 인증서
    - AI 클라이언트용 삭제, 116
  - configCCR 명령
    - c 옵션, 246
    - 수동 등록, 248
  - configure 하위 명령
    - sysconfig 명령, 65
  - cpu 조건 키워드, 136
  - d 옵션
    - installadm set-server 명령, 94
  - date\_time 기능 그룹
    - 설명, 63
  - DHCP
    - /etc/inet/dhcpd4.conf 구성 파일, 101, 102, 102
    - dhcpinfo 명령, 229
    - svc:/network/dhcp/server SMF 서비스, 101
    - 구성 파일, 101
    - 자동 구성, 95, 97
  - DHCP 서버
    - AI 및, 74
  - DHCP 서버가 응답하지 않음
    - AI 클라이언트 설치 문제 해결, 231

- x86 설치 문제 해결, 235
- dhcpinfo 명령, 229
- DNS
  - AI 클라이언트 설치 문제 해결, 229
- DNS 선택
  - 텍스트 설치 프로그램, 46
- emCCR 명령
  - 데이터 수집 변경, 248
- ERROR 403: Forbidden
  - SPARC 설치 문제 해결, 232
  - x86 설치 문제 해결, 236
- ERROR 404: Not Found
  - SPARC 설치 문제 해결, 232
  - x86 설치 문제 해결, 236
- Forbidden 오류
  - SPARC 설치 문제 해결, 232
  - x86 설치 문제 해결, 236
- g 옵션
  - sysconfig unconfigure 명령, 64, 64
- GPT 분할 영역
  - 설치 중 선택 및 수정, 27
- GPT 형식
  - 텍스트 설치 프로그램 및, 43
- GRUB 2
  - 시스템 분할 영역, 26
  - 텍스트 설치 프로그램 및, 43
- GRUB 메뉴, 100
- grub.cfg 파일, 100
  - x86 설치 문제 해결, 236
- grub2netx64.efi 파일, 101
- GUI 설치
  - 시스템 요구 사항, 25
  - 예, 35
- GUI 설치 프로그램
  - GUI 설치 준비, 34
  - 분할 영역 지침, 33
  - 설치 중 지원되지 않거나 누락된 그래픽 카드로 설  
치, 39
  - 설치 후 추가 패키지 추가, 40
  - 설치에 사용되는 기본 네트워크 및 보안 설정, 34
  - 지원되는 플랫폼, 33
- HMAC 해싱 키, 114, 224
  - SPARC 설치 문제 해결, 233
- hostname 조건 키워드, 136
- i86 값
  - cpu 조건 키워드, 136
- i86pc 값
  - 조건 키워드, 136
- identity 기능 그룹
  - 설명, 63
- include-site-profile 옵션
  - sysconfig unconfigure 명령, 64
- installadm 명령
  - create-client 하위 명령, 103, 221
  - create-manifest 하위 명령, 104
  - create-profile 하위 명령, 105
  - create-service 하위 명령, 96
  - delete-client 하위 명령, 104, 222
  - delete-manifest 하위 명령, 130
  - delete-profile 하위 명령, 131
  - export 하위 명령, 132
  - list 하위 명령, 120
  - set-client 하위 명령
    - hash 옵션, 116
    - x 옵션, 116
  - set-server 하위 명령
    - x 옵션, 116
  - set-service 하위 명령, 111, 126
  - update-manifest 하위 명령, 129
  - update-profile 하위 명령, 130
  - update-service 하위 명령, 128
  - validate 하위 명령, 129, 131, 166
- 보안 인증서 삭제, 115
- 보안 향상, 107
- 하위 명령 예제, 94, 94
- IP 주소 변수
  - 파생된 매니페스트, 144
- IPS 서버를 사용할 수 없음
  - 문제 해결, 237
- IPS 저장소
  - AI 및, 75
- IPv4 등록 정보 그룹
  - SMF 등록 정보, 174
- ipv4 조건 키워드, 136
- IPv6 등록 정보 그룹
  - SMF 등록 정보, 176
- ISA 변수
  - 파생된 매니페스트, 144
- iSCSI
  - 텍스트 설치 프로그램과 함께 사용, 43
- iSCSI 대상
  - AI 매니페스트, 161

- iSCSI 디스크 검색
  - GUI 설치, 35
  - 텍스트 설치 프로그램, 46
- keyboard 기능 그룹
  - 설명, 63
- LDAP 선택
  - 텍스트 설치 프로그램, 46
- Linux 스왑 분할 영역, 26
- location 기능 그룹
  - 설명, 63
- M 옵션
  - installadm set-server 명령, 95
- mac 조건 키워드, 136
- mDNS(멀티캐스트 DNS), 93
- mem 조건 키워드, 136
- menu.lst 파일
  - x86 설치 문제 해결, 236
- Microsoft Windows
  - Oracle Solaris 설치, 25
- MOS 살펴볼 내용 My Oracle Support
- My Oracle Support
  - AI 설치, 178
  - 자격 증명
    - Oracle Configuration Manager, 246
- naming\_services 기능 그룹
  - 설명, 63
- network 기능 그룹
  - 설명, 63
- network 조건 키워드, 136
- network-boot-arguments OBP 변수, 225
- Not Found 오류
  - SPARC 설치 문제 해결, 232
  - x86 설치 문제 해결, 236
- OBP 보안 키
  - 암호 키, 114, 224
  - 해싱 키(HMAC), 114, 224, 233
- OCM 살펴볼 내용 Oracle Configuration Manager
- openssl 명령
  - AI 실패 문제 해결, 239
- Oracle Auto Service Request
  - AI 설치용 구성, 178
  - GUI 설치, 35
  - 구성, 30
  - 텍스트 설치 프로그램, 46
- Oracle Configuration Manager
  - AI 설치용 구성, 178
  - GUI 설치, 35
  - Oracle Universal Installer, 246
  - 개요, 245
  - 구성, 30
  - 데이터 수집, 248
  - 사용, 247
  - 사용 안함, 247
  - 수동 등록, 248
  - 중앙 수집기, 246
  - 텍스트 설치 프로그램, 46
- Oracle Solaris 설치
  - 시스템 요구 사항, 25
- Oracle Solaris 인스턴스
  - 정의, 63
- Oracle Universal Installer
  - Oracle Configuration Manager, 246
- ORCL, SPARC-T4-2 값
  - platform 조건 키워드, 136
- OS 분할 영역
  - 설치 중 선택 및 수정, 27
- OUI 살펴볼 내용 Oracle Universal Installer
- p 옵션
  - installadm set-server 명령, 94
- P 옵션
  - installadm set-server 명령, 94
- PEM 형식의 X.509 인증서 및 키, 107
- pkg 명령
  - 기존 설치를 업데이트하기 위해 사용, 21
  - 라이브 매체 또는 GUI 설치 후 적용, 40
  - 텍스트 설치 후 소프트웨어 추가, 52
- platform 조건 키워드, 136
- pxegrub2 파일, 101
- RAID 구성
  - AI 매니페스트, 161
- reconfigure 하위 명령
  - sysconfig 명령, 65
- s 옵션
  - sysconfig unconfigure 명령, 64
- SCI 도구
  - 재구성, 65
- setupCCR 명령
  - c 옵션, 246
- SI\_\* 변수
  - 파생된 매니페스트, 143
- SMF 등록 정보

- AI 클라이언트 구성, 165
- all\_services 등록 정보 그룹, 91
- config 등록 정보 그룹, 171, 172, 177
- enable\_mapping 등록 정보, 172
- environment 등록 정보 그룹, 172
- IPv4 등록 정보 그룹, 174
- IPv6 인터페이스 등록 정보 그룹, 176
- root\_account 등록 정보 그룹, 169
- timezone 등록 정보 그룹, 172
- user\_account 등록 정보 그룹, 170
- 표시, 167
- SMF 서비스
  - svc:/application/auto-installer, 220
  - svc:/network/dhcp/server, 101
  - svc:/network/dns/client, 176
  - svc:/network/dns/multicast, 93
  - svc:/network/install, 174
  - svc:/system/config-user, 169
  - svc:/system/console-login, 174
  - svc:/system/environment:init, 172
  - svc:/system/identity, 171
  - svc:/system/install/server, 91, 106
  - svc:/system/keymap, 174
  - svc:/system/name-service/switch, 193
  - svc:/system/timezone, 172
  - svc:/system/zones-install, 197
  - 처음 부트 시 한 번 실행, 205
- SMF 서비스 로그
  - AI 실패 문제 해결, 239
- SMF 서비스 매니페스트
  - svcbundle 명령, 209
  - 만들기, 209
  - 매니페스트 만들기 도구 살펴볼 내용 svcbundle 명령
    - 사용자 정의
      - start 메소드 시간 초과, 211
      - 종속성, 212
    - 처음 부트 시 한번 실행 서비스 예제, 209
- SMF 정보
  - 제거, 64
- SMF(서비스 관리 기능)
  - AI 클라이언트 구성, 165
- sparc 값
  - cpu 조건 키워드, 136
- ssh 명령
  - AI 클라이언트 설치 모니터, 222
- sun4u 값
  - arch 조건 키워드, 136
- sun4v 값
  - arch 조건 키워드, 136
- SUNW,SPARC-Enterprise 값
  - platform 조건 키워드, 136
- support 기능 그룹
  - 설명, 63
- svcbundle 명령, 209
- svccfg 명령
  - 등록 정보 표시, 167
- SVR4 패키지 설치
  - AI 매니페스트, 162
- sysconfig 명령
  - configure 하위 명령, 65
  - reconfigure 하위 명령, 65
  - unconfigure 하위 명령, 64
- sysconfig create-profile 명령, 165
- system 기능 그룹
  - 모든 구성 데이터 제거, 64
  - 설명, 63
- system.conf 파일, 100
- TFTP Error
  - x86 설치 문제 해결, 235
- TLS 인증서가 유효하지 않음
  - 문제 해결, 239
- TLS(전송 계층 보안) 프로토콜, 107
- unbounded 키워드, 136
- unconfigure 하위 명령
  - c 옵션, 65
  - g 옵션, 64, 64
  - include-site-profile 옵션, 64
  - s 옵션, 64
- sysconfig 명령, 64
- USB 이미지
  - 텍스트 설치 프로그램을 위해 구하기, 45
- users 기능 그룹
  - 설명, 63
- VTOC 슬라이스
  - rpool 및 ZFS 루트 풀, 29
  - 설치 중 선택 및 수정, 27
  - 수정 옵션, 29
- WAN 부트 지원, 220
- wanboot 경보
  - SPARC 설치 문제 해결, 232
- wanboot.conf 파일, 100

webservd 사용자 및 그룹, 95  
 wget 명령  
   시 실패 문제 해결, 239  
 X.509 인증서 및 키, 107  
 x86  
   시스템 분할 영역, 28  
 x86 시스템 분할 영역  
   옵션, 28, 29  
 zonename 조건 키워드, 136

## ㄱ

계획  
   시 서버, 79  
 고유 ISA 변수  
   파생된 매니페스트, 144  
 관리 계층  
   정보 제거, 64  
 관리자 권한  
   시 서버, 90  
 구성 살펴볼 내용 시스템 구성  
   시  
   개요, 77  
   시 매니페스트 마법사, 158  
   Oracle Configuration Manager, 30  
 구성 데이터  
   모두 제거, 64  
   선택된 데이터 제거, 64  
 구성 프로파일 살펴볼 내용 시스템 구성 프로파일  
 구성 해제  
   인스턴스, 64  
   종료, 64  
 권한  
   sudo 명령, 90  
   권한 프로파일, 90  
   역할, 91  
 기능 그룹  
   개요, 63  
 기본 디렉토리  
   시 넷 이미지, 94  
 기본 암호  
   라이브 매체 또는 GUI 설치, 35  
   텍스트 설치 프로그램, 46

## ㄴ

날짜  
   GUI 설치 도중 설정, 35  
   텍스트 설치 프로그램 중에 설정, 46  
 네트워크 구성  
   텍스트 설치 프로그램 및, 44  
 네트워크 그룹화  
   SMF 등록 정보, 174  
 네트워크 번호  
   조건 키워드, 136  
 네트워크 번호 변수  
   파생된 매니페스트, 144  
 네트워크 인터페이스  
   시 서버, 79  
   구성, 174  
 네트워크 주소  
   조건 키워드, 136

## ㄷ

다중 운영 체제  
   설치 요구 사항, 25  
 대화식 설치  
   시스템 분할 영역(x86), 28  
 데이터 수집  
   Oracle Configuration Manager, 248  
 도구를 사용하여 장치 드라이버 찾기, 251  
 드라이버  
   찾기, 30, 251  
 디스크 공간 요구 사항  
   설치, 25  
 디스크 수 변수  
   파생된 매니페스트, 144  
 디스크 이름 변수  
   파생된 매니페스트, 144  
 디스크 크기 변수  
   파생된 매니페스트, 144

## ㄹ

라이브 매체, 33  
   살펴볼 다른 내용  
   살펴볼 다른 내용 GUI 설치 프로그램  
   GUI 설치 준비, 34  
   분할 영역 지침, 33

- 설치 중 지원되지 않거나 누락된 그래픽 카드로 설치, 39
  - 설치 후 추가 패키지 추가, 40
  - 설치에 사용되는 기본 네트워크 및 보안 설정, 34
  - 시스템 요구 사항, 25
  - 예, 35
  - 지원되는 플랫폼, 33
  - 콘솔 모드로 설치, 39
  - 로그 파일
    - 자동 설치, 61
- 
- 만들기
    - 파생된 매니페스트, 142
  - 매니페스트 살펴볼 내용 AI 매니페스트 멀티홈
    - AI 서버, 79
  - 멀티홈 AI 서버, 94
  - 메모리 메가바이트
    - 조건 키워드, 136
  - 메모리 변수
    - 파생된 매니페스트, 144
  - 메모리 요구 사항
    - 설치, 25
  - 명령줄
    - 자동 설치 시작, 241
  - 문제 해결
    - AI 설치, 229
  - 물리적 메모리 변수
    - 파생된 매니페스트, 144
- ㅂ
- 배포 생성자
    - 설명, 21
  - 변경
    - AI 매니페스트
      - 수동, 139
  - 보안 인증서
    - AI 서버용 삭제, 116
    - 삭제, 115
  - 보안 클라이언트 부트 오류
    - 문제 해결, 239
  - 부트
    - AI 클라이언트
      - 개요, 78
      - 자동 설치를 시작하지 않음, 240
    - 부트 디스크를 찾을 수 없음
      - AI 클라이언트 설치 문제 해결, 230
    - 부트 로드 실패
      - SPARC 설치 문제 해결, 231
    - 부트 아카이브
      - x86 설치 문제 해결, 236
    - 부트 오류
      - AI 클라이언트 설치 문제 해결, 230
    - 부트 파일
      - SPARC wanboot-cgi 파일, 101, 221
      - x86 grub2netx64.efi 파일, 101
      - x86 pxegrub2 파일, 101
      - x86 클라이언트 .bios 파일, 221
      - x86 클라이언트 .uefi 파일, 221
  - 분할
    - 텍스트 설치 프로그램, 43
  - 비전역 영역 살펴볼 내용 영역
- ㅅ
- 사용
    - Oracle Configuration Manager, 247
  - 사용 사례
    - AI, 80
  - 사용 안함
    - AI 매니페스트 마법사, 158
    - Oracle Configuration Manager, 247
  - 사용자 구성
    - GUI 설치, 35
    - 텍스트 설치 프로그램, 46
  - 삭제
    - AI 서버 인증서, 116
    - 단일 CA 인증서, 116
    - 보안 인증서, 115
    - 단일 클라이언트, 116
  - 선택
    - 설치 중 분할 영역, 27
  - 선택 알고리즘
    - 매니페스트 및 프로파일, 133
  - 설치
    - AI(자동 설치 프로그램) 사용 살펴볼 내용 AI 클라이언트 설치
      - 네트워크를 통해 텍스트 설치 프로그램 사용, 51
      - 다중 운영 체제, 25

- 설치 전의 기본 root 암호, 39
- 옵션 개요, 19
- 추가 옵션, 21
- 텍스트 설치 프로그램, 43
- 텍스트 설치 프로그램 및 USB 이미지 사용, 45
- 설치 디스크 선택
  - GUI 설치, 35
  - 텍스트 설치 프로그램, 46
- 설치 로그
  - SI 실패 문제 해결, 239
- 설치 서버 살펴볼 내용 SI 서버
- 설치 서비스
  - default-arch 서비스 별칭 변경, 99
  - DHCP 구성, 95, 97
  - GRUB 메뉴, 100
  - grub.cfg 파일, 100
  - system.conf 파일, 100
  - wanboot.cgi 파일, 101
  - wanboot.conf 파일, 100
  - 개요, 75
  - 넷 이미지
    - IPS 패키지, 98
    - ISO 파일, 97
    - 기본 대상, 100
    - 기본 소스, 100
    - 기본값 기본 디렉토리, 94
  - 등록 정보 수정
    - default-manifest 등록 정보, 127
  - 보안 구성, 106
  - 보안 웹 서버 호스트 포트, 94
  - 부트 파일
    - SPARC wanboot.cgi 파일, 101
    - x86 grub2netx64.efi 파일, 101
    - x86 pxegrub2 파일, 101
  - 설치 서비스 만들기, 96
  - 설치 서비스 업데이트, 128
  - 설치 서비스에 대한 정보 표시, 120
  - 설치 지침 살펴볼 내용 SI 매니페스트
    - 수정
      - 보안 등록 정보, 111
    - 웹 서버 파일, 95
    - 웹 서버 호스트 포트, 94
    - 작업 목록, 96
    - 클라이언트 구성 지침 살펴볼 내용 시스템 구성 프로파일
    - 클라이언트 삭제, 104
    - 클라이언트 설치 지침 살펴볼 내용 SI 매니페스트
      - 클라이언트 연관, 103
  - 설치 서비스 넷 이미지
    - IPS 패키지, 98
    - 기본 대상, 100
    - 기본 소스, 100
    - 기본값 기본 디렉토리, 94
  - 설치 서비스 변수
    - 파생된 매니페스트, 144
  - 설치 요구 사항, 25
  - 설치 인스턴스
    - SI 서버, 80
  - 설치 준비
    - 텍스트 설치 프로그램, 45
  - 설치 프로그램
    - x86 설치 문제 해결, 236
  - 수동 등록
    - Oracle Configuration Manager, 248
  - 수정
    - 설치 중 분할 영역, 27
  - 시간
    - GUI 설치 도중 설정, 35
    - 텍스트 설치 프로그램 중에 설정, 46
  - 시간대 선택
    - GUI 설치, 35
    - 텍스트 설치 프로그램, 46
  - 시스템 구성, 165
    - 클라이언트 설치 타임, 178
    - Oracle Auto Service Request, 178
    - Oracle Configuration Manager, 178
    - sysconfig create-profile 명령, 165
    - 구성 프로파일 검증, 166
    - 기본 영역 SI 매니페스트, 200
    - 네트워크 인터페이스, 174
    - 노드 이름 매핑, 172
    - 단말기 유형, 174
    - 사용자
      - /etc/auto\_home 파일, 170
      - /etc/passwd 파일, 170
      - autohome 등록 정보, 170
      - 다중 사용자 계정, 171
      - 루트 사용자 계정, 169
      - 암호화된 암호, 169
      - 자동 마운트 종속성, 170
      - 초기 사용자 계정, 169
    - 사용자 정의 IPS 패키지, 213

- 설치 서비스에 프로파일 추가, 166
- 시간대, 172
- 시스템 ID, 171
- 시스템 구성 프로파일 만들기, 165
- 시스템 로케일, 172
- 영역 시스템 구성 프로파일, 202
- 예제 프로파일, 180
  - DNS 프로파일과 함께 LDAP 사용, 193
  - DNS 프로파일과 함께 NIS 사용, 194
  - DNS와 검색 목록 프로파일, 190
  - LDAP 프로파일, 191
  - NIS 및 DNS 프로파일, 188
  - NIS 서비스 프로파일, 186
  - 보안 LDAP 프로파일, 191
  - 이름 서비스 프로파일, 186
  - 정적 네트워크 프로파일, 182
- 이름 서비스, 176
- 처음 부트 스크립트, 205
  - 만들기, 206
  - 템플릿, 206
  - 다중 IP 인터페이스 구성, 208
- 키보드 레이아웃, 174
- 호스트 이름, 172
- 시스템 구성 파일 디렉토리 변수
  - 파생된 매니페스트, 144
- 시스템 구성 프로파일, 105
  - 살펴볼 다른 내용 설치 서비스에 추가
    - /usr/share/auto\_install/sc\_profiles 프로파일, 180
    - 개요, 76
    - 선택 알고리즘, 135
    - 설치 서비스에서 삭제, 131
    - 예제, 180
    - 재구성, 65
    - 프로파일 검증, 131
    - 프로파일 복사, 132
    - 프로파일 선택 조건, 105
    - 프로파일 업데이트, 130
- 시스템 구성 프로파일 템플릿, 178
  - 변수, 179
- 시스템 분할 영역
  - GPT, 33
  - GRUB 2, 33
  - GUI 설치 프로그램 또는 라이브 매체 ISO 이미지, 33
  - VTOC 슬라이스, 29
- 설치 중 선택 및 수정, 27
- 지침, 26
- 시스템 요구 사항
  - 설치, 25
  - 자동 설치, 56
- 시스템 중단
  - x86 설치 문제 해결, 236
- 시작
  - 명령줄에서 자동 설치, 241
- 
- 암호화된 암호
  - /etc/shadow 파일에서 복사, 169
- 언어 선택
  - GUI 설치, 35
  - 텍스트 설치 프로그램, 46
- 영역
  - AI 매니페스트, 198, 200
    - configuration 요소, 197
    - 기본값, 200
  - AI 및, 77
  - AI 클라이언트에 설치, 197
  - config 파일, 198
  - 설치 서비스에 매니페스트 추가, 199
  - 설치 서비스에 프로파일 추가, 199
  - 시스템 구성 프로파일, 202
- 예제
  - AI 매니페스트, 160
- 웹 서버
  - 보안 호스트 포트, 94
  - 파일 디렉토리, 95
  - 호스트 포트, 94
- 이름 서비스 선택
  - 텍스트 설치 프로그램, 46
- 이미지 아카이브
  - SPARC 설치 문제 해결, 232
- 인증 실패
  - 문제 해결, 239
- 인증서
  - AI 서버용 삭제, 116
- ㄹ
- 자동 설치

- 개요, 55
  - 사용자 정의 매니페스트, 58
  - 설치 개요, 57
  - 시스템 요구 사항, 56
  - 잘못된 HMAC 값
    - SPARC 설치 문제 해결, 233
  - 장치 드라이버
    - 장치 드라이버 유틸리티 사용, 251
    - 정보 찾기, 30, 251
  - 재구성
    - SCI 도구 사용, 65
    - 시스템 구성 프로파일 사용, 65
    - 인스턴스, 65
  - 적용
    - 파생된 매니페스트, 142
  - 정의
    - Oracle Solaris 인스턴스, 63
  - 제거
    - 모든 구성 데이터, 64
    - 선택된 구성 데이터, 64
  - 조건 키워드
    - 시 클라이언트 선택, 136
  - 종료
    - 구성 해제 후, 64
  - 중앙 수집기
    - Oracle Configuration Manager, 246
  - 지침
    - 시스템 분할 영역, 26
- ㄱ**
- 처음 부트 스크립트
    - 개요, 76
  - 추가
    - GUI설치 후 추가 패키지, 40
    - 텍스트 설치 후 추가 패키지, 52
- ㅋ**
- 커널 아키텍처 변수
    - 파생된 매니페스트, 144
  - 콘솔 모드
    - 라이브 매체 설치, 39
  - 클라이언트 속성 환경 변수
    - 파생된 매니페스트, 143, 143
  - 클라이언트 아키텍처 변수
    - 파생된 매니페스트, 144
  - 키보드 선택
    - GUI 설치, 35
    - 텍스트 설치 프로그램, 46
- ㅅ**
- 테스트
    - 파생된 매니페스트 스크립트, 155
  - 텍스트 설치
    - VTOC 슬라이스 수정, 29
    - 시스템 요구 사항, 25
  - 텍스트 설치 프로그램
    - GUI 설치 프로그램 대비 장점, 20
    - SCSI 사용, 43
    - USB 이미지 구하기, 45
    - 기본 패키지 세트, 20
    - 네트워크를 통해 설치 시작, 51
    - 설치, 43
    - 설치 준비, 45
    - 설치 후 소프트웨어 추가, 52
- ㅈ**
- 파생된 매니페스트
    - AIM\_LOGFILE 환경 변수, 146
    - AIM\_MANIFEST 환경 변수, 146
    - aimanifest 명령, 144
    - aiuser 역할, 143
    - 만들기 및 적용, 142
    - 설치 서비스에 추가, 157
    - 수정할 초기 매니페스트, 144
    - 스크립트 검증, 157
    - 스크립트 테스트, 155
    - 예제 스크립트, 146
    - 클라이언트 속성 환경 변수, 143
  - 패키지 변수
    - 파생된 매니페스트, 144
  - 패키지를 찾을 수 없음
    - 문제 해결, 239
  - 프로세서 유형 변수
    - 파생된 매니페스트, 144
  - 프로파일 살펴볼 내용 시스템 구성 프로파일
  - 플랫폼 변수

파생된 매니페스트, 144

**ㅎ**

호스트 이름 변수

    파생된 매니페스트, 144

확장된 분할 영역

    Oracle Solaris 설치, 25

환경 변수

    AIM\_LOGFILE, 146

