

## Oracle® Solaris 11 시스템

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, 디스어셈블리 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 등록 상표입니다.

본 소프트웨어 혹은 하드웨어와 관련 문서(설명서)는 제 3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스와 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

---

머리말 .....	9
<b>제1부 Oracle Solaris 11 설치 옵션 .....</b>	<b>13</b>
<b>1 설치 옵션 개요 .....</b>	<b>15</b>
설치 옵션 비교 .....	15
간단한 사전 설정 설치 .....	16
서버 설정이 필요한 설치 .....	16
추가 옵션 .....	17
<b>제2부 설치 매체를 사용하여 설치 .....</b>	<b>19</b>
<b>2 설치 준비 .....</b>	<b>21</b>
LiveCD 및 텍스트 설치의 시스템 요구 사항 .....	21
다중 운영 체제 설치를 위한 부트 환경 준비 .....	22
시스템 분할 .....	23
설치 전 시스템 분할 지침 .....	23
대화식 설치 중 시스템 분할 지침 .....	23
적절한 장치 드라이버가 있는지 확인 .....	27
▼ Oracle Device Detection Tool 사용 방법 .....	27
장치 드라이버 유틸리티 개요 .....	28
▼ Device Driver Utility 시작 방법 .....	28
▼ 누락된 드라이버 설치 방법 .....	29
▼ HCL에 시스템 나열 방법 .....	31
Oracle Configuration Manager 사용 .....	31

<b>3 LiveCD 사용</b>	33
GUI 설치 프로그램으로 설치	33
GUI 설치 프로그램의 기본 설정	34
▼ GUI 설치 수행 방법	34
시스템이 콘솔 모드로 부트하는 경우 수행할 작업	39
▼ 시스템이 콘솔 모드로 부트되는 경우 LiveCD에서 Oracle Solaris를 설치하는 방법	39
LiveCD 설치 후 소프트웨어 추가	41
<b>4 텍스트 설치 프로그램 사용</b>	43
텍스트 설치 프로그램으로 설치	43
텍스트 설치 프로그램의 네트워킹 구성	44
▼ 텍스트 설치 수행 방법	44
텍스트 설치 후 소프트웨어 추가	50
네트워크를 통해 텍스트 설치 수행	51
▼ 네트워크를 통해 텍스트 설치 수행 방법	51
<b>5 매체에서 부트하는 자동 설치</b>	53
AI 매체를 사용한 설치 개요	53
AI 매체를 사용한 설치	53
AI 매체를 사용한 설치의 시스템 요구 사항	54
▼ AI 매체를 사용한 설치 방법	55
사용자 정의 AI 매니페스트 만들기	56
AI 매체에서 SPARC 시스템 부트	56
AI 매체에서 x86 시스템 부트	57
설치 로그 파일 보기	59
<b>6 Oracle Solaris 인스턴스 구성 해제 또는 재구성</b>	61
기능 그룹	61
sysconfig 유틸리티 사용	62
Oracle Solaris 인스턴스 구성 해제	62
시스템 구성	63
▼ SCI 도구를 사용하여 재구성하는 방법	63
SCI 도구를 사용하여 구성 프로파일 만들기	67

<b>제3부</b>	<b>설치 서버를 사용하여 설치</b>	69
<b>7</b>	<b>다중클라이언트의 자동 설치</b>	71
	자동 설치란?	71
	Automated Installer를 어떻게 사용합니까?	72
	Automated Installer 사용 사례	73
	AI 사용을 위한 최소 요구 사항	73
	설치 지침 사용자 정의	75
	시스템 구성 지침 제공	76
	로컬 IPS 패키지 저장소 제공	77
	사용자 정의 처음 부트 스크립트 제공	78
	추가 AI 설치 서비스 제공	79
<b>8</b>	<b>설치 서버 설정</b>	81
	AI 서버 설정 작업 맵	81
	설치 서버 요구 사항	82
	AI 서버 하드웨어 요구 사항	82
	AI 서버 소프트웨어 요구 사항	82
	AI 설치 도구 설치	83
	설치 서버 구성	85
	멀티홈 설치 서버 구성	85
	웹 서버 호스트 포트 구성	85
	AI 설치 서비스 만들기	86
	DHCP 설정 없이 설치 서비스 만들기	88
	로컬 DHCP 설정을 포함한 로컬 서비스 만들기	90
	설치 서버 유지 관리	92
	설치 서비스 추가, 수정 또는 삭제	92
	클라이언트를 설치 서비스와 연관	95
	클라이언트별 설치 지침을 설치 서비스와 연관	96
	클라이언트별 구성 지침을 설치 서비스와 연관	99
	AI 매니페스트 또는 시스템 구성 프로파일 내보내기	101
	AI 매니페스트 또는 시스템 구성 프로파일에 대한 조건 수정	101
	설치 서비스에 대한 정보 표시	102
	사용자 정의된 설치에 대한 정보 표시	103
	AISMF 서비스 관리	105

<b>9</b>	<b>설치 사용자 정의</b> .....	107
	클라이언트를 설치 및 구성 지침과 일치 .....	107
	AI 매니페스트 선택 .....	108
	시스템 구성 프로파일 선택 .....	109
	선택 조건 .....	109
	기본 AI 매니페스트 .....	113
<b>10</b>	<b>클라이언트 시스템 프로비전</b> .....	117
	XML AI 매니페스트 파일 사용자 정의 .....	118
	클라이언트 설치 타임에 AI 매니페스트 만들기 .....	119
	파생된 매니페스트 스크립트 만들기 .....	120
	파생된 매니페스트 스크립트를 설치 서비스에 추가 .....	133
<b>11</b>	<b>클라이언트 시스템 구성</b> .....	135
	구성 프로파일 제공 .....	135
	시스템 구성 프로파일 만들기 .....	135
	시스템 구성 프로파일 검증 .....	136
	시스템 구성 프로파일을 설치 서비스에 추가 .....	136
	시스템 구성 프로파일에 구성 지정 .....	137
	루트 및 사용자 계정 .....	138
	시스템 신원 .....	140
	시간대 및 로캘 .....	141
	터미널 유형 및 키보드 레이아웃 .....	142
	정적 네트워크 구성 .....	143
	이름 서비스 구성 .....	145
	시스템 구성 프로파일 템플릿 사용 .....	146
	예제 시스템 구성 프로파일 .....	148
	샘플 시스템 구성 프로파일 .....	148
	정적 네트워크 구성 지정 .....	149
	이름 서비스 구성 지정 .....	152
<b>12</b>	<b>영역 설치 및 구성</b> .....	161
	AI의 비전역 영역 설치 방법 .....	161
	전역 영역 AI 매니페스트의 영역 사양 .....	162

비전역 영역 구성 및 설치 데이터 .....	163
비전역 영역 AI 매니페스트 .....	164
비전역 영역 구성 프로파일 .....	166
<b>13 처음 부트 중 사용자 정의 스크립트 실행 .....</b>	<b>169</b>
처음 부트 시 실행할 스크립트 만들기 .....	169
SMF 매니페스트 파일 만들기 .....	171
스크립트 및 서비스에 대한 IPS 패키지 만들기 .....	172
▼ IPS 패키지를 만들어 게시하는 방법 .....	173
AI 클라이언트에 처음 부트 패키지 설치 .....	174
▼ IPS 패키지를 설치하는 방법 .....	174
<b>14 AI 클라이언트 시스템에서 사용할 Oracle Configuration Manager 설정 .....</b>	<b>177</b>
AI 클라이언트에서 Oracle Configuration Manager의 기본 동작 .....	177
사용자 정의 응답 파일 제공 .....	178
▼ 사용자 정의 응답 파일 패키지를 만들고 설치하는 방법 .....	178
Oracle Configuration Manager의 옵트아웃 .....	180
<b>15 클라이언트 시스템 설치 .....</b>	<b>181</b>
클라이언트가 설치되는 방법 .....	181
클라이언트 시스템 요구 사항 .....	182
SPARC 및 x86 클라이언트 시스템 요구 사항 .....	182
추가 SPARC 클라이언트 시스템 요구 사항 .....	182
설치 클라이언트 설정 .....	183
x86 클라이언트 설정 .....	183
SPARC 클라이언트 설정 .....	184
클라이언트를 서비스에서 삭제 .....	184
클라이언트 설치 .....	184
보안 셸을 사용하여 원격으로 설치 모니터 .....	184
SPARC 클라이언트 설치 .....	185
x86 클라이언트 설치 .....	188
클라이언트 설치 메시지 .....	191

<b>16 자동 설치 문제 해결 .....</b>	<b>193</b>
클라이언트 설치 실패 .....	193
설치 로그 및 지침 확인 .....	193
DNS 확인 .....	193
클라이언트 부트 오류 확인 .....	194
설치 시작 없이 설치 환경 부트 .....	201
설치 시작 없이 부트 후에 설치 시작 .....	202



# 머리말

---

**Oracle Solaris 11 시스템**은 다음 방식 중 하나를 사용하여 Oracle Solaris 운영 체제(OS)를 설치 및 구성하기 위한 지침을 제공합니다.

- Oracle Solaris LiveCD 설치 프로그램
- Oracle Solaris 대화식 텍스트 설치 프로그램
- Oracle Solaris AI(Automated Installer) 기능
- Oracle Solaris SCI Tool 대화식 시스템 구성 도구
- sysconfig(1M) 명령줄 시스템 구성 도구

모든 경우 설치를 완료하려면 네트워크의 패키지 저장소에 액세스가 필요합니다.

## 이 책의 대상

본 설명서는 Oracle Solaris 11 OS를 설치할 시스템 관리자를 대상으로 합니다.

## 이 책의 구성

본 설명서는 다음 부와 장으로 구성됩니다.

**제1부**는 사용자 요구에 가장 잘 맞는 방식을 선택하도록 도와주는 대안 설치 방식을 설명합니다.

**제2부:**

- 2 장, “설치 준비”
- 3 장, “LiveCD 사용”
- 4 장, “텍스트 설치 프로그램 사용”
- 5 장, “매체에서 부트하는 자동 설치”
- 6 장, “Oracle Solaris 인스턴스 구성 해제 또는 재구성”

제3부는 자동 설치 및 관련된 프로세스와 도구를 설명합니다.

- 7 장, “다중 클라이언트의 자동 설치”는 AI가 네트워크에서 다중 SPARC 및 x86 클라이언트 시스템의 핸드프리 설치를 수행하는 방법을 설명합니다.
- 8 장, “설치 서버 설정”은 클라이언트 설치를 관리하기 위한 별도의 시스템을 설정하는 방법을 설명합니다.
- 9 장, “설치 사용자 정의”는 여러 클라이언트 시스템이 각자 설치 및 구성되도록 클라이언트 선택 조건을 서로 다른 설치 지침 및 시스템 구성 지침에 적용하는 방법을 설명합니다.
- 10 장, “클라이언트 시스템 프로비전”은 서로 다른 클라이언트에 대한 사용자 정의 설치 지침을 만드는 방법을 설명합니다.
- 11 장, “클라이언트 시스템 구성”은 설치 후에 클라이언트 시스템 구성에 필요한 정보를 지정하는 방법을 설명합니다.
- 12 장, “영역 설치 및 구성”은 AI 클라이언트 설치의 일부로 비전역 영역의 설치 및 구성을 지정하는 방법을 설명합니다.
- 13 장, “처음 부트 중 사용자 정의 스크립트 실행”은 클라이언트 시스템의 추가 설치나 구성을 수행하기 위해 처음 부트 시 실행되는 스크립트를 만드는 방법을 설명합니다.
- 14 장, “AI 클라이언트 시스템에서 사용할 Oracle Configuration Manager 설정”은 Oracle Configuration Manager가 AI 클라이언트 설치에서 작동하도록 보증하는 방법을 설명합니다.
- 15 장, “클라이언트 시스템 설치”는 AI 클라이언트에 대한 시스템 요구 사항을 제공하고 각 클라이언트를 올바른 넷 이미지 및 설치/구성 지침과 연관시키는 방법을 설명합니다.
- 16 장, “자동 설치 문제 해결”은 몇몇 가능한 실패와 복구 방법을 설명합니다.

## 관련 정보

**Oracle Solaris 11 Installation Man Pages** 는 `aimanifest(1M)`, `distro_const(1M)`, `installadm(1M)`, `js2ai(1M)`, `ai_manifest(4)`, `dc_manifest(4)` 매뉴얼 페이지의 복사본을 제공합니다.

**사용자 정의 Oracle Solaris 11 설치 이미지 만들기** 는 Oracle Solaris Distribution Constructor 도구를 사용하여 설치 이미지를 사용자 정의하는 방법을 설명합니다.

**Oracle Solaris 11 부트 환경 만들기 및 관리**에서는 Oracle Solaris 시스템에서 비전역 영역을 비롯한 여러 부트 환경을 관리하는 방법에 대해 설명합니다.

**Oracle Solaris 관리: 일반 작업의 6 장, “서비스 관리(개요)”** describes the Oracle Solaris SMF(서비스 관리 기능)에 대해 설명합니다. SMF 프로파일을 사용하여 시스템을 구성할 수 있습니다.

**Oracle Solaris 11 소프트웨어 패키지 추가 및 업데이트**는 Oracle Solaris IPS(Image Packaging System) 기능과 IPS 패키지를 찾아서 설치하는 방법을 설명합니다. **pkg(5)** 매뉴얼 페이지에서 Image Packaging System을 더 자세히 설명합니다. **pkg(1)** 매뉴얼 페이지는 IPS 패키지를 찾아서 설치, 업데이트 및 확인하는 방법에 대해 자세히 제공합니다.

**Oracle Solaris 11 패키지 저장소 복사 및 만들기**는 Oracle IPS 패키지 저장소의 로컬 복사본을 만드는 방법 또는 고유의 사용자 정의 저장소를 만드는 방법을 설명합니다.

Oracle Solaris 11 시스템 관리 방법에 대한 자세한 내용은 Oracle Solaris 11 System Administration 설명서를 참조하십시오.

ISC(Internet Systems Consortium) DHCP 서버의 다운로드 및 설명서는 [isc.org](http://isc.org) 웹 사이트의 **DHCP** 섹션을 참조하십시오.

**Oracle Solaris 10 JumpStart에서 Oracle Solaris 11 자동 설치 프로그램으로 전환**는 JumpStart에서 AI로 이전하기 위한 정보를 제공합니다. 두 도구는 모두 Oracle Solaris의 자동 설치 기능입니다.

## Oracle Support에 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.

## 활자체 규약

다음 표는 이 책에서 사용되는 활자체 규약에 대해 설명합니다.

표 P-1 활자체 규약

활자체	의미	예
AaBbCc123	명령 및 파일, 디렉토리 이름; 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오.  모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오.  <code>machine_name% you have mail.</code>
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	<code>machine_name% su</code>  Password:
AaBbCc123	새로 나오는 용어, 강조 표시할 용어입니다. 명령줄 변수를 실제 이름이나 값으로 바꾸십시오.	<code>rm filename</code> 명령을 사용하여 파일을 제거합니다.

표 P-1 활자체 규약 (계속)		
활자체	의미	예
AaBbCc123	책 제목, 장, 절	사용자 설명서의 6장을 읽으십시오.  캐시는 로컬로 저장된 복사본입니다.  파일을 저장하면 <b>안 됩니다</b> .  <b>주:</b> 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

## 명령 예의 셸 프롬프트

다음 표에는 Oracle Solaris OS에 포함된 셸의 기본 UNIX 시스템 프롬프트 및 슈퍼유저 프롬프트가 나와 있습니다. 명령 예제에 표시된 기본 시스템 프롬프트는 Oracle Solaris 릴리스에 따라 다릅니다.

표 P-2 셸 프롬프트	
셸	프롬프트
Bash 셸, Korn 셸 및 Bourne 셸	\$
슈퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	#
C 셸	machine_name%
슈퍼유저용 C 셸	machine_name#

## 제 1 부

# Oracle Solaris 11 설치 옵션



## 설치 옵션 개요

Oracle Solaris 소프트웨어는 사용자 요구에 따라 여러 가지 다양한 방법으로 설치할 수 있습니다. 다음과 같은 Oracle Solaris 설치 옵션 개요를 참조하십시오.

### 설치 옵션 비교

다음 차트는 다양한 설치 옵션의 기능을 비교한 것입니다.

표 1-1 설치 옵션

설치 옵션	최소 준비	서버 필요	단일 또는 다중 시스템에 설치	패키지 저장소에서 패키지 설치
x86만 해당: 3 장, “LiveCD 사용”	예	아니오, 매체에서 설치	단일	아니오
4 장, “텍스트 설치 프로그램 사용”	예	아니오, 매체에서 설치	단일	아니오
51 페이지 “네트워크를 통해 텍스트 설치 수행”	아니오	예, 서버에서 설치 이미지 검색	단일	예
5 장, “매체에서 부트하는 자동 설치”	아니오	설치 매체를 사용자 정의하려는 경우 서버가 필요하지만, 설치에는 필요하지 않음	단일	예
7 장, “다중 클라이언트의 자동 설치”	아니오	예, 서버 필요	단일 또는 다중	예

더불어, 사용자 정의 LiveCD 이미지, 텍스트 설치 프로그램 이미지, 자동 설치 이미지를 포함한 **사용자 정의 Oracle Solaris 11 설치 이미지 만들기** 옵션이 있습니다.

## 간단한 사전 설정 설치

LiveCD의 GUI 설치 프로그램과 텍스트 설치 프로그램은 간단한 사전 설정 설치 프로그램 방식입니다.

- 두 설치 프로그램 모두 x86 플랫폼에 Oracle Solaris를 설치하는 데 사용할 수 있습니다. 또한 텍스트 설치 프로그램을 사용하여 SPARC 플랫폼에 Oracle Solaris를 설치할 수 있습니다.
- 두 설치 프로그램 모두 최소의 메모리에서 작동할 수 있습니다. 메모리 요구 사항을 확인하려면 **Oracle Solaris 11 릴리스 정보**를 참조하십시오.
- 두 설치 프로그램을 통해 설치하는 동안 디스크 분할 영역을 선택, 만들기 또는 수정할 수 있습니다.

LiveCD에는 데스크탑 또는 랩탑에 적절한 소프트웨어 세트가 들어 있습니다. 텍스트 설치 매체에는 범용 서버 시스템에 적절한 더 작은 소프트웨어 세트가 들어 있습니다.

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.

---

주 - 텍스트 설치 프로그램으로 설치된 패키지 세트는 `solaris-large-server` 패키지 세트입니다. 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 수정하는 것 외에도, 텍스트 설치 프로그램에서는 Solaris 분할 영역 내에 VTOC 슬라이스를 만들고 수정할 수 있습니다.

간단한 설치 수행에 대한 자세한 내용은 [제2부](#)를 참조하십시오.

## 서버 설정이 필요한 설치

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



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

AI를 사용하려면 먼저 네트워크에 서버를 설정해야 합니다. 클라이언트 시스템을 부트할 때 시스템이 서버에서 설치 사양을 얻고 Oracle Solaris 패키지 저장소에서 소프트웨어 패키지를 검색하면 클라이언트 시스템에 소프트웨어가 설치됩니다.

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

자세한 내용은 제3부를 참조하십시오.

AI 서버를 설정한 후에는 "핸즈프리" 네트워크 설치 이외의 두 가지 추가 설치 옵션이 제시됩니다.

- 네트워크를 통해 대화식 텍스트 설치를 수행하는 옵션이 있습니다. 대화식 설치를 통해 특정 시스템에 대한 설치 사양을 추가로 사용자 정의할 수 있습니다.

자세한 내용은 51 페이지 “네트워크를 통해 텍스트 설치 수행”을 참조하십시오.

- AI용 설정에는 AI 이미지를 다운로드하여 네트워크나 로컬에 저장하는 과정이 포함됩니다. CD, DVD 또는 (x86 설치의 경우) USB 플래시 드라이브와 같은 이동식 매체에 이미지를 구울 수 있습니다. 그런 다음 각 시스템에서 직접 AI 매체를 부트하여 자동 설치를 시작할 수 있습니다. AI 매체를 사용하는 설치는 대화식이 아닙니다.

지침은 5 장, “매체에서 부트하는 자동 설치”를 참조하십시오.

## 추가 옵션

이미 설명된 설치 옵션 외에도, Oracle Solaris 운영 체제를 설치/수정하기 위한 다음 옵션이 있습니다.

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.

설치된 Oracle Solaris 11 시스템 업데이트

자세한 내용은 [사용자 정의 Oracle Solaris 11 설치 이미지 만들기](#)를 참조하십시오.

기존 Oracle Solaris 11 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 소프트웨어 패키지 추가 및 업데이트](#)를 참조하십시오.

## 제 2 부

# 설치 매체를 사용하여 설치

GUI 설치 프로그램 또는 텍스트 설치 프로그램을 사용하여 최소한의 준비로 단일 시스템에 Oracle Solaris 운영 체제를 설치할 수 있습니다. 로컬에서 또는 네트워크를 통해 텍스트 설치를 수행할 수 있습니다. 추가적으로, AI(Automated Installer) 기능을 사용하는 경우 자동 설치 이미지를 만들어서 매체에 구운 후 이 매체를 사용하여 단일 시스템을 설치할 수 있습니다. 설치된 시스템 구성을 해제했다가 다시 구성하는 옵션도 있습니다.

다음은 참조하십시오.

- 2 장, “설치 준비”
- 3 장, “LiveCD 사용”
- 4 장, “텍스트 설치 프로그램 사용”
- 51 페이지 “네트워크를 통해 텍스트 설치 수행”
- 5 장, “매체에서 부트하는 자동 설치”
- 6 장, “Oracle Solaris 인스턴스 구성 해제 또는 재구성”



## 설치 준비

시스템을 설치하기 전에 다음 정보를 검토하십시오.

- 21 페이지 “LiveCD 및 텍스트 설치의 시스템 요구 사항”
- 22 페이지 “다중 운영 체제 설치를 위한 부트 환경 준비”
- 23 페이지 “시스템 분할”
- 27 페이지 “적절한 장치 드라이버가 있는지 확인”
- 31 페이지 “Oracle Configuration Manager 사용”

## LiveCD 및 텍스트 설치의 시스템 요구 사항

다음 표는 LiveCD 설치 이미지 또는 텍스트 설치 이미지를 사용한 Oracle Solaris 11 릴리스 설치 요구 사항을 간략히 설명합니다.

요구 사항	설명
메모리	<p>현재 릴리스의 최소 메모리 요구 사항을 확인하려면 <a href="#">Oracle Solaris 11 Release Notes</a>를 참조하십시오.</p> <p>주 - 텍스트 설치 프로그램은 LiveCD 설치 프로그램보다 적은 메모리가 필요합니다. 정확한 최소 요구 사항은 시스템 사양에 따라 달라집니다. 그러나 시스템에 GUI 설치 프로그램을 실행할 메모리가 부족한 경우 텍스트 설치 프로그램을 대신 사용하십시오.</p>
디스크 공간	<p>현재 릴리스의 디스크 공간 요구 사항을 확인하려면 <a href="#">Oracle Solaris 11 Release Notes</a>를 참조하십시오.</p>

# 다중 운영 체제 설치를 위한 부트 환경 준비

Oracle Solaris를 다중 부트 환경의 일부로 설치하는 경우 다음과 같은 다양한 운영 체제의 사양을 검토합니다.

표 2-1 다중 운영 체제 환경

기존 운영 체제	설명
Microsoft Windows	<p>Oracle Solaris 릴리스 설치를 위한 충분한 디스크 공간을 설정합니다. x86 플랫폼의 모든 Oracle Solaris 버전은 GRUB(GNU Grand Unified Bootloader)를 사용합니다. Oracle Solaris는 Windows를 인식하며, 설치 중에 Windows 분할 영역이 변경되지 않은 상태로 유지되는지 확인합니다. 설치가 완료되고 시스템이 재부트되면 GRUB 메뉴에 Windows 및 Oracle Solaris 부트 항목이 모두 표시됩니다.</p> <p>주 - 이제 x86 시스템의 Oracle Solaris 운영 체제가 RTC(실시간 시계) 또는 하드웨어 시계의 UTC(협정 세계시) 시간을 저장합니다. 이전에는 운영 체제가 x86 시스템에서 RTC의 로컬 시간을 저장했습니다.</p>
Linux, 또는 Windows 및 Linux	<p>Linux 운영 체제 또는 Linux와 Windows 운영 체제 둘 다 x86 기반 시스템에 설치된 경우 Oracle Solaris를 설치하기 전에 menu.lst 파일의 복사본을 저장합니다. 설치 후 menu.lst 파일을 편집하여 이전 설치의 Linux 정보를 추가해야 합니다. 지침은 <a href="#">x86 플랫폼에서 Oracle Solaris 부트 및 종료의 “menu.lst 파일을 편집하여 부트 항목 및 매개변수 수정”</a>을 참조하십시오.</p> <p>주 - Linux 운영 체제도 설치되어 있는 시스템에 Oracle Solaris를 설치하는 경우 Oracle Solaris 분할 영역은 반드시 Linux swap 분할 영역보다 우선해야 합니다.</p>
Solaris 10 OS	<p>LiveCD의 설치 프로그램으로는 Oracle Solaris의 다중 인스턴스를 설치할 수 없습니다. 그러나 텍스트 설치 프로그램은 각 인스턴스가 서로 다른 슬라이스에 있는 경우 동일한 분할 영역에서 Oracle Solaris 운영 체제의 다중 인스턴스를 지원합니다. LiveCD 및 텍스트 설치 프로그램을 사용하면 Oracle Solaris의 다중 인스턴스가 설치된 기존 시스템에서 Solaris 10 1/06 이상 릴리스를 대체할 수 있습니다.</p> <p>주 - 특정 Solaris 볼륨 목차(Volume Table of Contents, VTOC) 슬라이스를 현재 운영 체제에 유지해야 하는 경우 텍스트 설치 프로그램을 사용하십시오.</p>
확장된 분할 영역	<p>확장된 분할 영역에 다른 운영 체제가 있는 경우 설치 중에 기존의 확장된 분할 영역을 변경할 필요는 없습니다. LiveCD GUI 설치 프로그램, 텍스트 설치 프로그램 또는 자동 설치 프로그램을 사용하여 Oracle Solaris를 설치할 때 확장된 분할 영역을 만들기, 크기 조정 또는 삭제할 수 있습니다. 또한 확장된 분할 영역 내 논리적 분할 영역에 Oracle Solaris를 설치할 수도 있습니다.</p>

## 시스템 분할

이 섹션에서는 설치 전에 또는 대화식 설치 도중 시스템을 분할하는 방법을 안내합니다. 또한 x86의 분할 영역과 Solaris VTOC 슬라이스를 설정하는 방법을 설명합니다.

### 설치 전 시스템 분할 지침

LiveCD ISO 이미지 또는 텍스트 설치 프로그램 이미지에서 Oracle Solaris를 설치하는 경우 전체 디스크를 사용하거나 x86 분할 영역에 운영 체제를 설치할 수 있습니다. 더불어, 텍스트 설치 프로그램을 사용하면 SPARC 슬라이스에 운영 체제를 설치할 수 있습니다.

x86 기반 시스템에서 설치 프로그램은 하나 이상의 드라이브에 여러 개의 운영 체제 부트를 지원하는 GRUB를 사용합니다. 설치하는 동안뿐만 아니라 설치 이전에도 Oracle Solaris를 설치할 분할 영역을 만들 수 있습니다. 다양한 운영 체제를 분할하고 설치한 후에는 부트 시 GRUB 메뉴에서 적절한 메뉴 항목을 선택하여 운영 체제 중 하나를 배포할 수 있습니다.



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

fdisk 명령을 사용하여 Oracle Solaris fdisk 분할 영역을 만들거나 수정할 수 있습니다. 지침은 [Oracle Solaris 관리: 장치 및 파일 시스템의 “Solaris fdisk 분할 영역을 만드는 방법”](#)을 참조하십시오. [fdisk\(1M\)](#) 매뉴얼 페이지도 참조하십시오.

다른 방법으로, 상용 제품이나 오픈 소스 도구를 사용하여 하드 드라이브를 분할할 수 있습니다.

**주** - 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. 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 조건에 해당될 경우 설치 시 디스크 전체 레이아웃을 덮어씁니다.
  - 디스크 테이블을 읽을 수 없는 경우
  - 디스크가 이전에 분할된 적이 없는 경우
  - 설치할 전체 디스크를 선택합니다.
- 기존 fdisk 분할 영역이 Oracle Solaris 시스템에 있는 경우 사용자가 기존 분할 영역을 수정하지 않으면 설치 시 fdisk 분할 영역만 덮어씁니다. 따라서 기존의 다른 분할 영역은 변경되지 않습니다.
- 하나의 Solaris 분할 영역만 허용됩니다.
- Solaris 분할 영역을 설치에 사용해야 합니다.
- 기존 Solaris 분할 영역이 있는 경우 해당 분할 영역은 기본적으로 선택되어 있습니다. 분할 영역은 기존의 확장된 분할 영역 내에서 논리적 분할 영역이 될 수 있습니다.
- 디스크 분할 또는 슬라이스에서 변경한 사항은 설치 프로그램 패널 선택을 마치고 설치가 시작될 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 디스크 분할 옵션을 선택하면 선택한 디스크의 기존 fdisk 분할 영역이 패널에 표시됩니다. 디스크에 배치될 때와 동일한 순서로 최대 4개의 기본 분할 영역이 표시됩니다. 이러한 기본 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.
- fdisk 분할 영역이 OS 설치에 사용된다면 2TB보다 클 수 없습니다. 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 권장 및 최소 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 옵션	설명 및 사용자 조치(필요할 경우)
기존 Solaris 분할 영역을 사용합니다.	이 옵션은 현재 크기를 사용하여 기존 Solaris 분할 영역에 Oracle Solaris 11 운영 체제를 설치합니다. 디스크 분할 옵션을 선택합니다. 다른 사항은 변경할 필요가 없습니다.
새 Solaris 분할 영역을 만듭니다.	현재 시스템에 기존 Solaris 분할 영역이 없는 경우 새 Solaris 분할 영역을 만들 수 있습니다. 이렇게 하려면 기본 분할 영역이나 논리적 분할 영역을 선택한 다음 유형을 Solaris로 변경합니다. 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 기본 분할 영역에 Oracle Solaris 11 운영 체제를 설치합니다.	다른 기본 분할 영역에 운영 체제를 설치할 수 있습니다. 이렇게 하려면 먼저 기존의 Solaris 분할 영역 유형을 사용되지 않음으로 변경해야 합니다. 그런 다음 다른 분할 영역을 선택하고 유형을 Solaris로 변경할 수 있습니다. 설치하는 동안 이렇게 수정하면 이전 Solaris 분할 영역과 새 Solaris 분할 영역의 기존 분할 영역 내용이 모두 지워집니다.
확장된 분할 영역 내에 새로운 Solaris 분할 영역을 만듭니다.	확장된 분할 영역 내에 새로운 Solaris 분할 영역을 만들 수 있습니다. Solaris 분할 영역이 이미 있는 경우 해당 유형을 사용되지 않음으로 변경합니다. 그런 다음 확장된 분할 영역을 새로 만들려면 분할 영역 유형을 확장으로 변경합니다. 확장된 분할 영역의 크기를 조정한 다음 확장된 분할 영역 안의 논리적 분할 영역 중 하나를 Solaris 분할 영역으로 변경할 수 있습니다. 또한 논리적 분할 영역의 크기를 해당 논리적 분할 영역을 포함한 확장된 분할 영역의 크기까지 늘릴 수 있습니다.
기존 분할 영역을 삭제합니다.	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.

## 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 Oracle ZFS 루트 풀에 설치되어 있어야 합니다. 기본적으로 루트 풀이 포함된 슬라이스에는 설치 프로그램에서 `rpool`이라는 레이블을 지정합니다. 루트 풀이 **없는** 슬라이스에 운영 체제를 설치하려면 설치 프로그램에서 해당 슬라이스의 유형을 `rpool`로 변경합니다. 설치하는 동안 ZFS 루트 풀이 해당 슬라이스에 만들어집니다.

주 - 하나의 ZFS 풀에만 `rpool`을 지정할 수 있으므로 `rpool`이 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 슬라이스를 사용되지 않음으로 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 11 운영 체제를 설치합니다. 대상 슬라이스를 선택한 다음 유형을 <code>rpool</code> 로 변경합니다.
슬라이스 크기 조정	새로 만든 <code>rpool</code> 슬라이스의 크기만 변경할 수 있습니다. 필드에 새 크기를 입력합니다.
새 슬라이스 만들기	사용되지 않은 슬라이스를 선택하고 해당 유형을 변경합니다. 예를 들어, 사용되지 않음을 <code>rpool</code> 로 변경합니다.
기존 슬라이스 삭제	슬라이스 유형을 사용되지 않음으로 변경합니다. 설치 중 슬라이스가 삭제되므로 인접한 슬라이스 크기 조정에 해당 공간을 사용할 수 있습니다.

## 적절한 장치 드라이버가 있는지 확인

Oracle Solaris를 설치하기 전에 시스템의 장치가 지원되는지 여부를 확인합니다. <http://www.oracle.com/webfolder/technetwork/hcl/index.html>의 하드웨어 호환성 목록(Hardware Compatibility List, HCL)에서는 Oracle Solaris에서 작동하는 것으로 인증되거나 보고된 하드웨어에 대한 정보를 제공합니다. Solaris on x86 Platforms Device Support 도구에서는 다양한 x86 구성 요소를 지원하는 Oracle Solaris 드라이버를 알 수 있습니다.

다음 유틸리티를 사용하여 장치 드라이버가 사용 가능한지 여부를 확인할 수 있습니다.

### ■ Oracle Device Detection Tool

Oracle Device Detection Tool은 최신 릴리스가 시스템에서 감지된 장치를 지원하는지 여부를 보고합니다. 이 도구는 여러 가지 Solaris 10 릴리스, Windows, Linux, Mac OS X 및 FreeBSD를 비롯하여 다양한 시스템에서 실행됩니다. Oracle Device Detection Tool에 대한 링크는 HCL(<http://www.oracle.com/webfolder/technetwork/hcl/index.html>)에 있습니다. 도구 사용에 대한 지침은 27 페이지 “Oracle Device Detection Tool 사용 방법”을 참조하십시오.

### ■ Device Driver Utility

Device Driver Utility는 Oracle Device Detection Tool과 동일한 정보를 제공합니다. 이 유틸리티는 Oracle Solaris 11 시스템의 데스크탑에 사용할 수 있습니다. 그리고 텍스트 설치 프로그램 메뉴 옵션을 통해 사용할 수 있습니다.

---

주 - Device Driver Utility에는 1.5GB 이상의 메모리가 필요할 수 있습니다. 시스템에 설치를 수행하는 데 적합한 보완 장치가 있는 경우 Device Driver Utility를 실행하기 전에 먼저 설치를 완료한 다음 설치된 하드 디스크를 부트합니다. 그러면 유틸리티에서 설치된 시스템의 스왑 공간을 사용할 수 있습니다.

---

## ▼ Oracle Device Detection Tool 사용 방법

Oracle Device Detection Tool을 사용하여 현재 릴리스에 시스템의 모든 장치용 드라이버가 있는지 여부를 확인할 수 있습니다.

- 1 웹 브라우저에서 [http://www.oracle.com/webfolder/technetwork/hcl/hcts/device\\_detect.html](http://www.oracle.com/webfolder/technetwork/hcl/hcts/device_detect.html)로 이동합니다.
- 2 Device Detection Tool 사용 섹션에서 Oracle Device Detection Tool 시작 옵션을 누릅니다.
- 3 라이선스 계약에 동의합니다.
- 4 ddtool 다운로드 링크를 누릅니다.

- 5 JavaWS로 열기 옵션을 선택하고 실행을 선택합니다.  
도구가 실행되지만 시스템에 설치되지는 않습니다.
- 6 드라이버 가용성을 확인할 대상 운영 체제를 선택합니다.

---

참고 - 추가 정보를 보려면 도움말 버튼을 누릅니다.

---

## 장치 드라이버 유틸리티 개요

Oracle Device Detection Tool의 대안으로, Device Driver Utility는 시스템의 장치와 이러한 장치가 관리하는 드라이버에 대한 정보를 제공합니다. 이 유틸리티는 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.

- 28 페이지 “Device Driver Utility 시작 방법”
- 29 페이지 “누락된 드라이버 설치 방법”
- 31 페이지 “HCL에 시스템 나열 방법”

## ▼ Device Driver Utility 시작 방법

Device Driver Utility는 설치 이미지를 부트하면 자동으로 실행됩니다. Oracle Solaris를 설치한 후 Device Driver Utility를 수동으로 시작할 수도 있습니다.

- 다음 방법 중 하나를 사용하여 Device Driver Utility를 시작합니다.

- LiveCD 설치 이미지를 부트합니다.

LiveCD ISO 이미지를 부트하면 Device Driver Utility가 백그라운드로 실행됩니다. 기본 게시자의 IPS 패키지에 누락된 드라이버가 있으면 Device Driver Utility에서 해당 드라이버 패키지를 자동으로 설치합니다. 누락된 다른 드라이버가 있을 경우 보고서를 검토하고 누락된 추가 드라이버를 설치할 수 있도록 유틸리티 창을 표시하라는 메시지가 Device Driver Utility에 표시됩니다.

- LiveCD의 데스크탑에서 Device Driver Utility를 수동으로 시작합니다.

LiveCD의 데스크탑에서 Device Driver Utility를 수동으로 시작하려면 데스크탑에서 Device Driver Utility 아이콘을 두 번 누릅니다. 또는 주 메뉴에서 응용 프로그램 → 시스템 도구 → Device Driver Utility를 선택합니다.

■ **Oracle Solaris 텍스트 설치 프로그램 이미지를 부트합니다.**

텍스트 설치 프로그램에서 Device Driver Utility를 시작하려면 초기 메뉴에서 추가 드라이버 설치를 선택합니다.

주 - 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 Device Driver Utility를 시작하려면 주 메뉴에서 응용 프로그램 → 시스템 도구 → Device Driver Utility를 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. [29 페이지 “누락된 드라이버 설치 방법”](#)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.

---

참고 - 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 표시되면 다음을 수행합니다.

- 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. [31 페이지 “HCL에 시스템 나열 방법”](#)을 참조하십시오.

## ▼ HCL에 시스템 나열 방법

다음과 같이 특정 장치를 작업하면서 찾은 드라이버에 대한 정보를 다른 사용자와 공유할 수 있습니다.

### 1 장치 드라이버 유틸리티를 시작합니다.

[28 페이지 “Device Driver Utility 시작 방법”](#)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.

## Oracle Configuration Manager 사용

LiveCD 설치 또는 텍스트 설치 후에 재부트하는 즉시 익명 시스템 설치 구성 정보가 Oracle Configuration Manager에 의해 Oracle Support로 업로드됩니다. My Oracle Support는 설치된 시스템의 구성 정보를 수신하지만, 고객 정보는 어떤 것도 수집하지 않습니다.

특히, 처음 재부트 중 Oracle Configuration Manager 서비스가 처음으로 실행되고 등록 서버에 시스템을 등록하려고 시도합니다. 이 등록을 성공하면 익명 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. 일정 잡기는 `/usr/sbin/emCCR`을 사용하여 조정할 수 있습니다. `emCCR(1M)` 매뉴얼 페이지와 **Oracle Configuration Manager Installation and Administration Guide**를 참조하십시오.

LiveCD 설치 또는 텍스트 설치를 수행할 때 다음 옵션이 있습니다.

- 구성 정보의 기본 익명 등록이 자동으로 발생하도록 허용
- 시스템을 재부트하기 전에, 설치 끝에 Oracle Configuration Manager 기능 사용 안함  
사용 안함 지침은 다음을 참조하십시오.
  - LiveCD 설치: [단계 13](#)
  - 텍스트 설치: [단계 16](#)

---

주 - 설치 이미지를 만들어서 매체에 굽거나 네트워크를 통해 텍스트 설치를 시작하는 등의 목적으로 설치 서버를 사용하는 경우 Oracle Configuration Manager 관리를 위한 다양한 옵션이 있습니다. [14 장, “AI 클라이언트 시스템에서 사용할 Oracle Configuration Manager 설정](#)”을 참조하십시오.

---

기본 익명 등록을 허용 또는 취소하도록 선택했는지 여부에 따라, Oracle Configuration Manager를 사용하여 나중에 시스템을 등록 또는 재등록하도록 선택할 수 있습니다.

- 이미 익명으로 등록한 경우 지원 서비스를 이용하기 위해 나중에 등록의 일부로 MOS(My Oracle Solaris) 계정 정보를 제공하도록 선택할 수 있습니다. 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.
- 기본 등록을 취소한 경우 익명으로 등록하거나 나중에 전체 고객 등록을 수행할 수 있습니다.
- 네트워크 프록시 요구 사항으로 인해 자동 등록을 완료하지 못한 경우 익명으로 또는 MOS 계정 정보로 등록할 수 있습니다.

대화식 모드로 `configCCR` 유틸리티(`/usr/sbin/configCCR`)를 사용하여 등록할 수 있습니다. 등록을 완료한 후에 다음과 같이 서비스를 사용으로 설정할 수 있습니다.

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

자세한 내용은 다음을 참조하십시오.

- **Oracle Solaris 관리: 일반 작업의 5 장, “Oracle Configuration Manager 작업”**
- `configCCR(1M)` 매뉴얼 페이지
- **Oracle Configuration Manager Installation and Administration Guide**



## LiveCD 사용

---

이 장에서는 LiveCD 이미지를 사용하여 설치를 수행하는 방법을 설명합니다.

### GUI 설치 프로그램으로 설치

Oracle Solaris 11 소프트웨어를 설치할 때 다음 정보를 고려하십시오.

- 21 페이지 “[LiveCD 및 텍스트 설치의 시스템 요구 사항](#)”을 참조하십시오.
- 둘 이상의 운영 체제가 설치된 x86 기반 시스템에 Oracle Solaris를 설치하려는 경우 설치 프로세스 중에 디스크를 분할할 수 있습니다. GUI 설치 프로그램 또는 텍스트 설치 프로그램 사용 지침을 참조하십시오.

---

주 - LiveCD ISO 이미지의 설치 프로그램은 x86 플랫폼에만 사용됩니다.

---

또는 설치 전에 `fdisk` 명령이나 타사 분할 도구를 사용하여 새 분할 영역을 만들거나 기존 분할 영역을 조정할 수 있습니다. 23 페이지 “[설치 전 시스템 분할 지침](#)”을 참조하십시오.

특정 운영 체제의 설치 환경 준비에 대한 자세한 내용은 22 페이지 “[다중 운영 체제 설치를 위한 부트 환경 준비](#)”를 참조하십시오.

- GUI 설치 프로그램은 운영 체제를 업그레이드할 수 없습니다. 그러나 Oracle Solaris 11 릴리스를 설치한 후에는 이미지 패키징 시스템을 사용하여 시스템에서 업데이트를 사용할 수 있는 모든 패키지를 업데이트할 수 있습니다. [Oracle Solaris 11 소프트웨어 패키지 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 덮어씁니다.

## GUI 설치 프로그램의 기본 설정

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

- Oracle Solaris는 도메인 이름 시스템(Domain Name System, DNS) 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 가능합니다.
- NFSv4 도메인은 동적으로 파생됩니다.

## ▼ GUI 설치 수행 방법

### 1 예비 작업을 완료합니다.

#### a. LiveCD가 없으면 LiveCD ISO 이미지를 다운로드합니다.

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

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

이미지를 다운로드한 후 다음 중 하나를 수행합니다.

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

주 - USB 이미지의 경우, USB 플래시 드라이브에 이미지를 복사하려면 `usbcopy` 유틸리티가 필요합니다. `pkg:/install/distribution-creator` 패키지를 설치하여 이 유틸리티를 시스템에 추가할 수 있습니다.

- 시스템에 이미지를 저장하고 가상 머신에서 실행합니다.

- b. 시스템에서 설치 프로그램을 실행하는 데 필요한 요구 사항과 제한 사항을 확인합니다.
  - i. 시스템이 필요한 시스템 요구 사항을 모두 충족하는지 확인합니다.  
21 페이지 “LiveCD 및 텍스트 설치의 시스템 요구 사항”을 참조하십시오.
  - ii. 필요한 장치 드라이버가 모두 있는지 확인합니다.  
27 페이지 “적절한 장치 드라이버가 있는지 확인”을 참조하십시오.
- c. 다음과 같은 Oracle Solaris 11 릴리스 설치 옵션 중 하나를 선택합니다.
  - Oracle Solaris 11 릴리스가 시스템에 설치되는 유일한 운영 체제인 경우 33 페이지 “GUI 설치 프로그램으로 설치”를 참조하십시오.
  - 다중 운영 체제 설치를 지원하는 환경을 설정하는 경우:
    - i. 22 페이지 “다중 운영 체제 설치를 위한 부트 환경 준비”에서 사양을 검토합니다.
    - ii. 시스템을 백업합니다.
    - iii. 설치 전에 시스템을 분할해야 하는 경우 23 페이지 “시스템 분할”을 참조하십시오.

---

주 - Linux 운영 체제를 이미 설치한 경우에는 설치 후 사용할 수 있도록 기존 menu.lst 파일의 복사본을 USB 드라이브에 저장해야 합니다. 지침은 **x86 플랫폼에서 Oracle Solaris 부트 및 종료의 “menu.lst 파일을 편집하여 부트 항목 및 매개변수 수정”**을 참조하십시오.

---

## 2 설치 매체를 삽입하고 시스템을 부트합니다.

LiveCD에서 GRUB 메뉴가 표시될 때 다른 옵션을 선택하지 않으면 기본 항목이 자동으로 사용됩니다.

---

주 - 시스템의 그래픽 카드가 LiveCD에서 지원되지 않거나 시스템에 그래픽 카드가 없는 경우 LiveCD를 삽입하면 시스템이 콘솔 모드로 부트됩니다. 이런 경우 GUI 설치를 수행할 수 없습니다. 39 페이지 “시스템이 콘솔 모드로 부트하는 경우 수행할 작업”을 참조하십시오.

---

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

## 3 키보드 및 언어를 선택하거나 기본 영어 옵션을 적용합니다.

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

#### 4 설치에 필요하지만 누락된 드라이버를 모두 설치합니다.

LiveCD를 부트할 때 드라이브가 누락되어 있으면 프롬프트가 표시됩니다. Device Driver Utility에 액세스하는 지침에 따라 설치에 필요한 드라이버를 찾아 설치합니다.

#### 5 LiveCD 데스크탑에서 Oracle Solaris 설치 아이콘을 두 번 눌러 GUI 설치 프로그램을 시작합니다.

#### 6 시작 패널에서 다음을 선택합니다.

주 - 이 패널에서 현재 릴리스에 대한 릴리스 정보를 검토할 수 있습니다.

#### 7 디스크 패널에서 여러 설치 대상이 표시된 경우 설치 대상을 선택하거나 기본값을 적용합니다. 그런 다음, 운영 체제를 전체 디스크에 또는 디스크의 분할 영역에 설치할지 여부를 지정합니다.

선택적으로 분할 영역 레이아웃을 수정할 수 있습니다. 지침은 [23 페이지 “대화식 설치 중 시스템 분할 지침”](#)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 선택합니다. 그런 다음, 현재 로컬 시간과 일치하도록 날짜 및 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 암호가 필요합니다. 로그인 이름은 문자로 시작해야 하며 문자와 숫자만 포함할 수 있습니다.

주 - 내가 만드는 사용자 계정은 관리 권한을 갖습니다.

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

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

#### 10 설치 사양을 검토합니다.

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

#### 11 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.

#### 12 설치 로그를 검토합니다.

설치 결과 패널에서 설치 로그에 액세스하여 검토할 수 있습니다.

#### 13 (옵션) Oracle Configuration Manager를 사용하여 설치된 시스템의 익명 등록을 취소하려면 다음 단계에 따라 새로 만든 부트 환경을 마운트하고 시스템을 재부트하기 전에 해당 부트 환경에 "opt-out" 파일을 추가합니다.



주의 - 기본적으로 설치된 시스템의 시스템 구성은 Oracle Configuration Manager로 전송됩니다. 이것은 고객 정보가 제공되지 않는 익명 등록입니다.

익명 등록은 초기 설치 후에 재부트 시 자동으로 발생하지만, 설치 후 설치된 시스템을 재부트하기 전에 다음 지침에 따라 등록을 취소할 수 있습니다.

설치 타임에 옵트아웃하지 않으려면 나중에 언제든지 서비스를 일시 중지할 수 있습니다.

자세한 내용은 31 페이지 “Oracle Configuration Manager 사용”을 참조하십시오.

a. 설치된 시스템을 재부트하기 전에 종료를 눌러 설치 프로그램을 끝냅니다.

b. 터미널 창을 엽니다.

c. 루트 역할을 맡습니다.

---

주 - 사용자 패널에서 컴퓨터 이름을 변경했다라도, 새로 설치된 시스템을 재부트하기 전의 기본 루트 암호는 "solaris"입니다.

---

**d. 다음 예와 같이 새로 만든 부트 환경을 마운트합니다.**

```
# beadm mount solaris /a
```

재부트하기 전의 부트 환경 이름은 기본적으로 "solaris"입니다.

**e. 편집기에서 마운트된 부트 환경에 새 파일을 만들고 파일 이름을 /a/etc/svc/profile/site/ocm.xml로 지정합니다.**

예를 들어, 다음 명령어를 입력합니다.

```
# vi /a/etc/svc/profile/site/ocm.xml
```

**f. 다음 내용을 파일에 입력하고 파일을 저장한 후 파일을 종료합니다.**

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='profile' name='ocm'
  xmlns:xi='http://www.w3.org/2003/XInclude'>
  <service name='system/ocm' type='service' version='1'>
    <instance name='default' enabled='false' />
  </service>
</service_bundle>
```

이 파일은 기본 SMF 서비스를 사용 안함으로 설정하고 익명 등록에서 등록 정보를 "opt-out"으로 변경합니다.

**g. 이 샘플 명령에 표시된 대로 부트 환경을 마운트 해제합니다.**

```
# beadm unmount solaris
```

**h. 터미널을 종료합니다.**

---

주 - 설치와 재부트 후에 이 파일을 제거하고 서비스를 사용으로 설정하여 시스템을 등록하도록 선택할 수 있습니다.

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

이 명령은 익명 등록을 수행합니다.

시스템 구성 데이터를 MOS 계정과 연관시키려는 경우 또는 사이트에 네트워크 프로키를 사용해야 할 경우 configCCR 명령을 사용해야 합니다. [31 페이지 “Oracle Configuration Manager 사용”](#)을 참조하십시오.

---

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

성공적으로 설치된 후에는 시스템을 재부트하거나 설치 프로그램을 끝내고 시스템을 종료합니다.

다음 시스템 부트가 시작되면 LiveCD를 꺼냅니다. 또는 GRUB 메뉴에서 “하드 디스크에서 부트” 옵션을 선택합니다.

설치에 실패하면 설치 로그를 확인하고 설치 프로그램을 종료할 수 있습니다.

## 시스템이 콘솔 모드로 부트하는 경우 수행할 작업

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

다음과 같은 두 가지 대안이 있습니다.

- LiveCD ISO 이미지 대신 텍스트 설치 프로그램 이미지를 사용합니다.  
네트워크에 액세스하지 않고 로컬 콘솔에서 텍스트 설치 프로그램을 실행할 수 있습니다. 4 장, “[텍스트 설치 프로그램 사용](#)”을 참조하십시오.
- 다음 절차를 사용하여 원격 설치를 수행합니다.

---

주 - 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 LiveCD에서 Oracle Solaris를 설치하는 방법

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

- 1 설치할 시스템에 LiveCD를 삽입한 다음 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 활성화합니다.**

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

**5 DHCP에서 대상 시스템에 할당된 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 말합니다.**

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

---

주 - 설치 전의 기본 루트 암호는 "solaris"입니다.

---

**8 GUI 설치 프로그램을 실행합니다.**

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

---

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

---

**9 설치가 완료되면 대상 시스템을 재부트합니다.**

**참조** 다음 항목에 대한 내용은 [Oracle Solaris 관리: 일반 작업](#)를 참조하십시오.

- 사용자 계정 및 그룹 관리
- 시스템 부트 및 종료
- 서비스 관리
- 하드웨어 오류 관리
- 시스템 프로세스 관리
- 다음과 같은 일반적인 시스템 문제 해결:
  - 재부트를 실패할 경우 수행할 작업
  - 루트 암호를 잊은 경우 수행할 작업
  - 시스템이 정지될 경우 수행할 작업



## LiveCD 설치 후 소프트웨어 추가

운영 체제를 설치한 후에 소프트웨어 패키지를 추가하려면 **pkg(1)** 매뉴얼 페이지와 **Oracle Solaris 관리: 일반 작업의 12 장**, “소프트웨어 패키지 관리(작업)”에 설명된 대로 **pkg** 명령을 사용합니다. 또는 Oracle Solaris Package Manager GUI 도구를 사용하여 추가 소프트웨어를 설치할 수 있습니다. 데스크탑 메뉴에서 시스템>관리>패키지 관리자로 이동합니다.

**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 식별했고 **pkg install** 명령을 **-nv** 옵션과 함께 사용한 결과를 조사했으면 다음과 비슷한 명령을 실행하여 추가 소프트웨어를 설치합니다.

```
# pkg install --require-new-be --be-name newBEname packagename
```

이 샘플 명령에는 새 부트 환경 만들기에 필요한 옵션이 있고 설치할 패키지를 지정합니다.

GUI 데스크탑이 없는데 Oracle Solaris 데스크탑을 설치하려면 **solaris-desktop** 패키지를 설치합니다.



## 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 11 릴리스를 설치할 때 다음 정보를 고려하십시오.

- 21 페이지 “LiveCD 및 텍스트 설치의 시스템 요구 사항”을 참조하십시오.
- 둘 이상의 운영 체제가 설치된 x86 기반 시스템에 Oracle Solaris를 설치하려는 경우 설치 프로세스 중에 디스크를 분할할 수 있습니다.  
또는 설치 전에 fdisk 명령이나 타사 분할 도구를 사용하여 새 분할 영역을 만들거나 기존 분할 영역을 조정할 수 있습니다. 23 페이지 “설치 전 시스템 분할 지침”을 참조하십시오.  
특정 운영 체제의 설치 환경 준비에 대한 자세한 내용은 22 페이지 “다중 운영 체제 설치를 위한 부트 환경 준비”를 참조하십시오.
- Oracle Solaris 11 설치 프로그램은 운영 체제를 업그레이드할 수 없습니다. 그러나 Oracle Solaris 11 릴리스를 설치한 후에는 이미지 패키징 시스템을 사용하여 시스템에서 업데이트를 사용할 수 있는 모든 패키지를 업데이트할 수 있습니다.  
**Oracle Solaris 11 소프트웨어 패키지 추가 및 업데이트**를 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.

---

- LiveCD에는 데스크탑 또는 랩탑에 적절한 소프트웨어 세트가 들어 있습니다. 텍스트 설치 매체에는 범용 서버 시스템에 적절한 더 작은 소프트웨어 세트가 들어 있습니다. 특히 텍스트 설치 프로그램을 사용하면 GNOME 데스크탑이 설치되지 않습니다.

텍스트 설치 프로그램으로 설치 후에 추가 패키지를 설치하려면 50 페이지 “텍스트 설치 후 소프트웨어 추가”를 참조하십시오.

## 텍스트 설치 프로그램의 네트워킹 구성

텍스트 설치 프로그램의 네트워킹 패널은 사용자에게 다음 옵션을 제공합니다.

- 자동 - LiveCD 설치 프로그램 방식과 비슷한, 자동 NCP로 대상 시스템을 구성합니다.
- 수동 - "DefaultFixed" NCP를 선택하고 한 네트워크 인터페이스(NIC)의 정적 IPv4 구성을 제공합니다. 선택한 NIC에 대해 IPv4 기본 경로와 IPv6 자동 구성이 사용으로 설정됩니다. 이 옵션은 DNS, NIS, LDAP 이름 지정 서비스의 수동 구성도 제공합니다.
- 없음 - "DefaultFixed" NCP를 선택하고 루프백 인터페이스만 구성합니다.

## ▼ 텍스트 설치 수행 방법

### 1 예비 작업을 완료합니다.

#### a. 텍스트 설치 프로그램 이미지가 없으면 이미지를 다운로드합니다.

Oracle Solaris 텍스트 설치 프로그램 ISO 이미지를 다운로드하려면

[http://www.oracle.com/](http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html)

[technetwork/server-storage/solaris11/downloads/index.html](http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html)로 이동합니다.

---

주 - USB 플래시 드라이브에 이미지를 구우려면 USB 이미지를 다운로드합니다.

---

이미지를 다운로드한 후 다음 중 하나를 수행합니다.

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

---

주 - USB 이미지의 경우, USB 플래시 드라이브에 이미지를 복사하려면 `usbcopy` 유틸리티가 필요합니다. `pkg:/install/distribution-creator` 패키지를 설치하여 이 유틸리티를 시스템에 추가할 수 있습니다.

---

- 시스템에 이미지를 저장하고 가상 머신에서 실행합니다.

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

- i. 시스템이 필요한 시스템 요구 사항을 모두 충족하는지 확인합니다.

21 페이지 “LiveCD 및 텍스트 설치의 시스템 요구 사항”을 참조하십시오.

ii. 필요한 장치 드라이버가 모두 있는지 확인합니다.

27 페이지 “적절한 장치 드라이버가 있는지 확인”을 참조하십시오.

c. 다중 운영 체제 설치를 지원하는 환경을 설정하는 경우:

i. 22 페이지 “다중 운영 체제 설치를 위한 부트 환경 준비”에서 사양을 검토합니다.

ii. 시스템을 백업합니다.

iii. 설치 전에 시스템을 분할해야 하는 경우:

2 장, “설치 준비”에서 지침을 검토합니다. 특히 분할 영역이나 슬라이스에 Oracle Solaris를 설치하고 설정하려는 경우 아직 작업을 수행하지 않았으면 23 페이지 “설치 전 시스템 분할 지침”의 정보를 검토합니다.

---

주 - Linux 운영 체제를 이미 설치한 경우에는 설치 후 사용할 수 있도록 기존 menu.lst 파일의 복사본을 USB 드라이브에 저장해야 합니다. 지침은 **x86 플랫폼에서 Oracle Solaris 부트 및 종료의 “menu.lst 파일을 편집하여 부트 항목 및 매개변수 수정”**을 참조하십시오.

---

2 설치 매체를 삽입하고 시스템을 부트한 다음 예비 키보드 및 언어를 선택합니다.

---

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

---

3 (선택 사항) 필수 드라이버 설치의 설치 메뉴에서 옵션 #2를 선택합니다.

Device Driver Utility 사용에 대한 지침은 28 페이지 “Device Driver Utility 시작 방법”을 참조하십시오. 드라이버를 설치한 후에 텍스트 설치를 다시 시작하고 설치 메뉴로 돌아갑니다.

4 설치 메뉴에서 첫번째 옵션을 선택하여 설치를 시작합니다.

Welcome to the Oracle Solaris xxx 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]:

---

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

---

5 시작 패널을 지나서 계속합니다.

- 6 디스크 패널에서 여러 개의 대상 디스크가 나열된 경우 대상 디스크를 선택하거나 기본값을 적용합니다.
- 7 운영 체제를 전체 디스크에 또는 디스크의 분할 영역이나 슬라이스에 설치할지 여부를 선택합니다.
  - 전체 디스크
  - x86 분할 영역
  - SPARC 슬라이스
- 8 (옵션) 일련의 대상 선택 패널에서 분할 영역 또는 슬라이스 레이아웃을 수정할 수 있는 옵션이 있습니다.  
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.

자세한 분할 지침은 23 페이지 “대화식 설치 중 시스템 분할 지침”을 참조하거나 설치 프로그램의 온라인 도움말을 참조하십시오.

- 9 네트워크에서 시스템을 식별하는 컴퓨터 이름을 입력합니다.
- 10 다음 옵션 중 하나를 선택하여 유선 이더넷 네트워크 연결을 어떻게 구성할지 지정합니다.
  - 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. 연결 설정을 입력하거나 설치 프로그램에서 감지하여 제공한 기본 정보를 적용합니다.

---

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

---

- c. 시스템이 DNS 이름 서비스를 사용해야 하는지 여부를 지정합니다.
- d. Configure DNS(DNS 구성)를 선택한 경우 다음 단계를 계속합니다.
  - i. 시스템에서 사용할 DNS 서버에 대한 IP 주소를 하나 이상 입력합니다.
  - ii. DNS 질의를 실행할 때 검색될 도메인 이름을 하나 이상 제공합니다.

- e. 시스템이 LDAP 이름 서비스, NIS 이름 서비스 또는 None(없음)을 사용해야 하는지 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 프로파일 없이 시스템에 LDAP를 구성하려면 LDAP 대신 None(없음)을 선택합니다. 그런 다음, 설치가 완료된 후 수동으로 LDAP를 구성합니다.

---

주 - 선택된 네트워크 이름 지정 서비스가 없으면 /etc/hosts(4)와 같은 표준 이름 소스 파일을 사용하여 네트워크 이름을 분석할 수 있습니다. 자세한 내용은 nsswitch.conf(4) 매뉴얼 페이지를 참조하십시오.

---

- f. 선택한 대체 이름 서비스에 대해 시스템이 상주하는 도메인을 제공합니다.

---

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

---

- g. LDAP를 유일한 이름 서비스로 또는 DNS와 더불어 추가 이름 서비스로 선택한 경우 다음과 같이 LDAP 사양을 제공합니다.

- i. 시스템에 LDAP 이름 서비스를 구성하는 데 사용할 LDAP 프로파일을 지정합니다.

- ii. LDAP 프로파일 서버의 IP 주소를 입력합니다.

- iii. LDAP 검색 기준을 제공하거나 기본 검색 기준을 적용합니다.

- iv. LDAP 프록시 바인드 정보를 제공할지 여부를 지정합니다.

---

주 - 프로파일이 프록시 인증서 레벨을 지정하고 인증 방식이 None(없음)이 아닌 경우 프록시 바인드 정보를 제공해야 합니다. 해당 정보를 생략하면 LDAP가 초기화되지 않습니다.

---

- v. 필요한 경우, LDAP 프록시 바인드 식별 이름 및 프록시 바인드 암호를 제공합니다.

- h. NIS를 유일한 이름 서비스로 또는 DNS와 더불어 추가 이름 서비스로 선택한 경우 NIS 사양을 제공합니다.

소프트웨어에서 이름 서버를 검색하도록 하거나, 직접 이름 서버를 지정할 수 있습니다. 다음 두 가지 옵션 중 하나를 선택합니다.

- Find One(이름 서버 찾기)을 선택합니다.

---

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

---

- **Specify One(이름 서버 지정)**을 선택하고 하위 패널에 이름 서버의 호스트 이름 또는 IP 주소를 입력합니다.

일련의 네트워킹 구성 패널을 완료한 후에 설치 프로그램에 일련의 시간대 패널과 날짜 및 시간 패널이 표시됩니다.

- **설치 중 네트워크가 구성되지 않도록 지정하려면 None(없음)**을 선택합니다.  
설치가 시간대 패널로 진행됩니다.

- 11 **일련의 시간대 패널에서 먼저 시간대를 선택하고, 로컬 시간과 일치하도록 날짜 및 시간을 조정합니다.**

---

주 - 기본값은 GMT 시간대가 구성되도록 하는 것입니다.

---

- 12 **사용자 패널을 완성합니다.**

반드시 사용자 계정을 만들 필요는 없습니다. 그러나 루트 암호를 만들어야 합니다.

- **이 패널에서 사용자 계정을 만들면 사용자 암호와 루트 암호를 모두 제공해야 합니다.**  
이 경우 루트는 사용자에게 할당된 역할이 됩니다.  
사용자 계정을 만들려면 사용자 이름과 암호를 입력합니다. 이름은 문자로 시작해야 하며 문자와 숫자만 포함할 수 있습니다.

- **사용자 계정을 만들지 않으면 계속 루트 암호를 제공해야 합니다.**  
이 경우 루트는 일반 사용자가 됩니다.

- 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.**

Oracle Solaris 설치 프로세스가 시작됩니다.



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

---

- 15 **설치 로그를 검토합니다.**

설치 결과 패널에서 설치 로그에 액세스하여 검토할 수 있습니다.



- 16 (옵션) Oracle Configuration Manager를 사용하여 설치된 시스템의 익명 등록을 취소하려면 다음 단계에 따라 새로 만든 부트 환경을 마운트하고 시스템을 재부트하기 전에 해당 부트 환경에 "opt-out" 파일을 추가합니다.



주의 - 기본적으로 설치된 시스템의 시스템 구성은 Oracle Configuration Manager로 전송됩니다. 이것은 고객 정보가 제공되지 않는 익명 등록입니다.

익명 등록은 초기 설치 후에 재부트 시 자동으로 발생하지만, 설치 후 설치된 시스템을 재부트하기 전에 다음 지침에 따라 등록을 취소할 수 있습니다.

설치 타임에 옵트아웃하지 않으려면 나중에 언제든지 서비스를 일시 중지할 수 있습니다.

자세한 내용은 31 페이지 “Oracle Configuration Manager 사용”을 참조하십시오.

- a. 설치된 시스템을 재부트하기 전에 F9를 눌러 설치 프로그램을 끝냅니다.
- b. 설치 메뉴에서 '3'을 선택하여 셸을 엽니다.
- c. 루트 역할을 말합니다.
- d. 다음 예와 같이 새로 만든 부트 환경을 마운트합니다.

```
# beadm mount solaris /a
```

주 - 설치 프로그램 패널에서 컴퓨터 이름을 변경했다 해도, 재부트하기 전의 부트 환경 이름은 기본적으로 "solaris"입니다.

- e. 편집기에서 마운트된 부트 환경에 새 파일을 만들고 파일 이름을 /a/etc/svc/profile/site/ocm.xml로 지정합니다.

예를 들어, 다음 명령어를 입력합니다.

```
# vi /a/etc/svc/profile/site/ocm.xml
```

- f. 다음 내용을 파일에 입력하고 파일을 저장한 후 파일을 종료합니다.

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='profile' name='ocm'
  xmlns:xi='http://www.w3.org/2003/XInclude'>
  <service name='system/ocm' type='service' version='1'>
    <instance name='default' enabled='false' />
  </service>
</service_bundle>
```

이 파일은 기본 SMF 서비스를 사용 안함으로 설정하고 익명 등록에서 등록 정보를 "opt-out"으로 변경합니다.

g. 이 샘플 명령에 표시된 대로 부트 환경을 마운트 해제합니다.

```
# beadm unmount solaris
```

h. 셸을 종료합니다.

```
# exit
```

주 - 설치와 재부트 후에 이 파일을 제거하고 서비스를 사용으로 설정하여 시스템을 등록하도록 선택할 수 있습니다.

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

이 명령은 익명 등록을 수행합니다.

시스템 구성 데이터를 MOS 계정과 연관시키려는 경우 또는 사이트에 네트워크 프로토콜을 사용해야 할 경우 configCCR 명령을 사용해야 합니다. [31 페이지 “Oracle Configuration Manager 사용”](#)을 참조하십시오.

## 17 재부트하거나 셸로 이동하고 시스템을 종료합니다.

참조 다음 항목에 대한 내용은 [Oracle Solaris 관리: 일반 작업](#)를 참조하십시오.

- 사용자 계정 및 그룹 관리
- 시스템 부트 및 종료
- 서비스 관리
- 하드웨어 오류 관리
- 시스템 프로세스 관리
- 다음과 같은 일반적인 시스템 문제 해결:
  - 재부트를 실패할 경우 수행할 작업
  - 루트 암호를 잊은 경우 수행할 작업
  - 시스템이 정지될 경우 수행할 작업

## 텍스트 설치 후 소프트웨어 추가

운영 체제를 설치한 후에 소프트웨어 패키지를 추가하려면 pkg(1) 매뉴얼 페이지와 [Oracle Solaris 관리: 일반 작업의 12 장, “소프트웨어 패키지 관리\(작업\)”](#)에 설명된 대로 pkg 명령을 사용합니다.

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 식별했고 `pkg install` 명령을 `-nv` 옵션과 함께 사용한 결과를 조사했으면 다음과 비슷한 명령을 실행하여 추가 소프트웨어를 설치합니다.

```
# pkg install packagename
```

`packagename` 변수를 설치할 패키지의 이름으로 바꿉니다.

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

```
# pkg install --require-new-be --be-name newBName packagename
```

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

## 네트워크를 통해 텍스트 설치 수행

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

### ▼ 네트워크를 통해 텍스트 설치 수행 방법

- 1 AI 클라이언트 이미지를 다운로드하고 해당 이미지를 기반으로 설치 서비스를 만듭니다. 지침은 [제3부](#)를 참조하십시오.
- 2 다음과 같이 네트워크를 통해 클라이언트 시스템을 부트합니다.

- SPARC 클라이언트의 경우 OBP 프롬프트에 다음을 입력합니다.

```
# boot net:dhcp
```

- x86 클라이언트의 경우 설치 메뉴에서 1을 선택합니다.

Welcome to the Oracle Solaris xxx 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 [44 페이지](#) “[텍스트 설치 수행 방법](#)”을 참조하십시오.

---

주 - 텍스트 설치 프로그램으로 설치된 패키지 세트는 `solaris-large-server` 패키지 세트입니다. 그러나 네트워크를 통한 텍스트 설치 프로그램은 실제로 자동 설치입니다. 자동 설치는 IPS 저장소에서 필요한 소프트웨어를 가능한 많이 다운로드하도록 설계되어 있습니다. 네트워크를 통해 텍스트 설치 프로그램을 사용할 경우 더 작은 패키지 세트인 `solaris-auto-install`이 기본적으로 설치됩니다.

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

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

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

---

## 매체에서 부트하는 자동 설치

---

네트워크를 통한 부트가 아닌 매체의 AI 이미지를 부트하여 SPARC 시스템 또는 X86 시스템에 Oracle Solaris 11 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 설치 서버에 배치될 시스템을 설치합니다.
- WAN 부트 기능이 없는 SPARC 시스템을 설치합니다.
- 고장난 시스템 문제를 해결합니다. 이동식 매체에서 시스템을 부트하고 설치된 시스템을 검사하고 진단을 실행합니다.

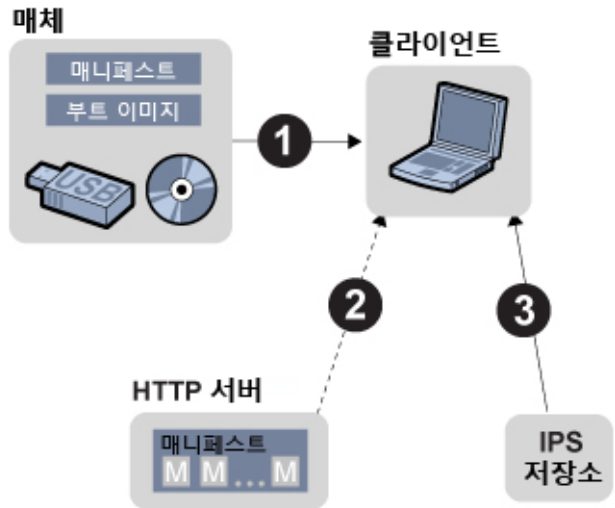
AI 매체를 사용한 설치는 다음 특징이 있습니다.

- 설치 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. [56 페이지 “사용자 정의 AI 매니페스트 만들기”](#)에 설명된 대로 기본 AI 매니페스트를 검토합니다.

그림 5-1 매체를 사용한 AI 설치



# AI 매체를 사용한 설치의 시스템 요구 사항

SPARC 및 x86 시스템 모두 다음 요구 사항을 충족해야 합니다.

표 5-1 AI 매체를 사용한 설치의 시스템 요구 사항

요구 사항	사양
메모리	현재 릴리스의 최소 메모리 요구 사항을 확인하려면 <a href="#">Oracle Solaris 11 Release Notes</a> 를 참조하십시오.
디스크 공간	현재 릴리스의 디스크 공간 요구 사항을 확인하려면 <a href="#">Oracle Solaris 11 Release Notes</a> 를 참조하십시오.
네트워크 액세스	설치될 시스템은 설치 중 다음 리소스에 액세스할 수 있어야 합니다. <ul style="list-style-type: none"><li>■ 네트워크 구성 정보를 제공하는 DHCP 서버</li><li>■ 클라이언트 시스템에 설치될 패키지를 포함하는 IPS 저장소</li></ul> 사용자 정의 AI 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 다음 인터넷 위치로 이동합니다.

[http://www.oracle.com/](http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html)

[technetwork/server-storage/solaris11/downloads/index.html](http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html)

- SPARC 시스템 – SPARC AI .iso 파일을 다운로드합니다.
- x86 시스템 – x86 AI .iso 파일 또는 x86 AI .usb 파일을 다운로드합니다.

### 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. [56 페이지 “사용자 정의 AI 매니페스트 만들기”](#)를 참조하십시오.

### 3 부트 가능한 매체를 만듭니다.

- SPARC 및 x86 ISO 이미지 – .iso 파일을 CD나 DVD에 굽습니다.
- x86 USB 이미지 – USB 플래시 드라이브에 이미지를 복사하려면 **usbcopy** 유틸리티를 사용합니다.

---

주 - pkg:/install/distribution-creator 패키지를 설치하여 이 유틸리티를 시스템에 추가할 수 있습니다.

---

### 4 매체에서 부트합니다.

부트 이미지가 포함된 장치에서 시스템을 부트합니다. 기본 AI 매니페스트 또는 사용자 정의 AI 매니페스트를 지정하는 방법은 [56 페이지 “AI 매체에서 SPARC 시스템 부트”](#) 및 [57 페이지 “AI 매체에서 x86 시스템 부트”](#)를 참조하십시오.

"핸즈프리" 설치가 수행됩니다. 설치 후에 SCI 도구가 시작되고 시스템에 대한 구성 정보를 제공하도록 요청합니다.

### 5 SCI 도구 패널에서 구성 정보를 제공합니다.

[67 페이지 “SCI 도구를 사용하여 구성 프로파일 만들기”](#)를 참조하십시오.

## 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 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/default.xml에 있습니다.

```
# lofi_dev=$(/usr/sbin/lofiadm -a /tmp/sol-11-ai-sparc.iso)
# /usr/sbin/mount -o ro -F hsfs ${lofi_dev} /mnt
# cp /mnt/auto_install/manifest/default.xml /home/username/custom.xml
# /usr/sbin/umount /mnt
# /usr/sbin/lofiadm -d ${lofi_dev}
```

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

다른 방법으로, 사용자 정의 매니페스트를 만들기 위한 기초로 [113 페이지 “기본 AI 매니페스트”](#)에 표시된 매니페스트를 사용할 수 있습니다.

대상 디스크나 추가로 설치할 패키지과 같은 설치 사양을 변경하려면 ai\_manifest(4) 매뉴얼 페이지를 참조하십시오.

---

주 - Oracle Configuration Manager 설정을 사전 설정하는 옵션도 있습니다. [14 장, “AI 클라이언트 시스템에서 사용할 Oracle Configuration Manager 설정”](#)을 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.xml일 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 매니페스트 사용

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 매니페스트 사용

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.xml을 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 0.97 (639K lower / 2078660K upper memory)
```

```
Oracle Solaris 11 Automated Install custom
Oracle Solaris 11 Automated Install
Oracle Solaris 11 Automated Install custom ttya
Oracle Solaris 11 Automated Install custom ttyb
Oracle Solaris 11 Automated Install ttya
Oracle Solaris 11 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 매니페스트 사용

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

```
Oracle Solaris 11 Automated Install
Oracle Solaris 11 Automated Install ttya
Oracle Solaris 11 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 매니페스트 사용

사용자 정의 AI 매니페스트를 사용하려면 다음 옵션 중 하나를 선택합니다.

```
Oracle Solaris 11 Automated Install custom
Oracle Solaris 11 Automated Install custom ttya
Oracle Solaris 11 Automated Install custom ttyb
```

이러한 사용자 정의 옵션 중 하나를 선택할 때 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 x86 이미지 부트

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

일반적으로, 사용 중인 GRUB 항목의 커널 라인에 **install=true**가 지정된 경우 설치가 자동으로 시작합니다. 자동 설치를 즉시 시작하지 않고 x86 시스템을 부트하려면 선택할 GRUB 메뉴 항목을 조사합니다. 해당 GRUB 항목의 커널 라인에 **install=true**가 지정된 경우 라인을 편집하여 **install=true**를 제거합니다. 그러면 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.
- 설치를 성공한 경우 시스템을 재부트하기 전에 `/system/volatile/install_log`에서, 재부트한 후에 `/var/sadm/system/logs/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

Oracle Solaris 인스턴스를 구성 해제 또는 재구성할 때 여러 가지 미리 정의된 하위 시스템이 영향을 받습니다. 이러한 하위 시스템을 기능 그룹이라고 합니다.

인스턴스의 전체 그룹은 "시스템"이라고 합니다.

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

표 6-1 기능 그룹

그룹화	구성 요소	구성되지 않은 상태
system	전체 시스템	아래의 컴파일
identity	시스템 노드 이름	알 수 없음
kdb_layout	키보드	미국 영어
network	네트워크	네트워크 없음
location	시간대	UTC
	로캘	C 로캘
users	루트	빈 루트 암호
	초기 사용자 계정	사용자 계정 제거
naming_services	DNS, NIS, LDAP 클라이언트, nsswitch	네트워크 이름 지정 서비스 없음

## sysconfig 유틸리티 사용

sysconfig 유틸리티를 사용하여 Oracle Solaris 인스턴스에 다음 구성 작업을 수행할 수 있습니다.

- 전역 또는 비전역 영역에서 Oracle Solaris 인스턴스 구성을 해제하고 구성되지 않은 상태로 두려면 `sysconfig unconfigure` 명령을 사용합니다.

62 페이지 “Oracle Solaris 인스턴스 구성 해제”를 참조하십시오.

- 전역 또는 비전역 영역에서 Oracle Solaris 인스턴스를 재구성하려면 `sysconfig configure` 명령을 사용합니다.
  - 기존 구성 프로파일을 명령에 지정하면 비대화식 구성이 수행됩니다.
  - 기존 구성 프로파일을 명령에 지정하지 않으면 대화식 SCI 도구가 실행됩니다. SCI 도구를 사용하여 해당 Solaris 인스턴스에 대한 특정 구성 정보를 제공할 수 있습니다.

63 페이지 “시스템 구성”을 참조하십시오.

- `sysconfig create-profile` 명령을 사용하여 새 시스템 구성 프로파일을 만들 수 있습니다. 67 페이지 “SCI 도구를 사용하여 구성 프로파일 만들기”를 참조하십시오.

sysconfig 명령은 Solaris 인스턴스의 모든 기능 그룹에 영향을 줍니다. 자세한 지침은 `sysconfig(1M)` 매뉴얼 페이지를 참조하십시오.

---

주 - sysconfig 유틸리티를 사용하려면 루트 역할을 맡아야 합니다.

---

## Oracle Solaris 인스턴스 구성 해제

이전에 구성된 Solaris 인스턴스 구성을 해제하고 구성되지 않은 상태로 두려면 `unconfigure` 명령을 사용합니다. 모든 기능 그룹의 구성이 해제됩니다.

다음 예와 같이 `sysconfig unconfigure` 명령을 사용합니다.

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

이 예는 인스턴스 구성을 해제합니다.

---

주 - -g 옵션이 지정되지 않은 경우 시스템 구성을 해제하기 전에 확인을 요청합니다.

---

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

```
# sysconfig unconfigure -s
```

자세한 내용은 `sysconfig(1M)` 매뉴얼 페이지를 참조하십시오.

# 시스템 구성

`sysconfig configure` 명령을 사용하여 전역 또는 비전역 영역에서 Oracle Solaris 인스턴스를 구성하거나 재구성할 수 있습니다. 이 구성은 대화식으로 또는 비대화식으로 발생할 수 있습니다.

- `sysconfig configure` 유틸리티에 `-c` 옵션을 사용하여 기존 시스템 구성 프로파일을 지정할 수 있습니다. 해당 옵션과 함께 유틸리티를 실행하면 유틸리티가 기존 프로파일에서 구성 사양을 읽고 이러한 사양을 사용하여 시스템을 비대화식으로 구성합니다.

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

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

---

주 - `-c` 옵션은 프로파일 또는 프로파일의 디렉토리를 지정합니다. 모든 프로파일은 `.xml` 파일 확장자를 포함해야 합니다.

---

시스템 구성(SC) 프로파일에 대한 내용은 [11 장, “클라이언트 시스템 구성”](#)을 참조하십시오.

- `sysconfig configure` 명령을 지정된 프로파일 없이 호출하면 SCI 도구가 자동으로 활성화됩니다.

SCI 도구는 새로 설치되거나 구성되지 않은 시스템의 구성을 지원합니다. 이 도구를 사용하여 새로 만든 비전역 영역 또는 다른 구성되지 않은 시스템에 대한 시스템 구성을 제공할 수 있습니다. SCI 도구는 구성 정보를 묻는 일련의 대화식 텍스트 패널로 구성됩니다. [63 페이지 “SCI 도구를 사용하여 재구성하는 방법”](#)을 참조하십시오.

---

주 - SCI 도구의 일련의 화면은 텍스트 설치의 일부로 자동으로 실행되기도 합니다.

---

다른 방법으로, SCI 도구를 실행하여 SCI 도구 화면에 입력한 구성 사양을 기반으로 새 시스템 구성 프로파일을 만들 수 있습니다. [67 페이지 “SCI 도구를 사용하여 구성 프로파일 만들기”](#)를 참조하십시오.

## ▼ SCI 도구를 사용하여 재구성하는 방법

- 1 루트 역할로 전환합니다.

---

주 - 비전역 영역에서 작업하는 경우 다음과 같이 루트 역할로 영역에 로그인합니다.

```
# zlogin -C -e ^ZONENAME
```

---

**2 프로파일을 지정하지 않고 `sysconfig configure` 명령을 실행합니다.**

```
# sysconfig configure
```

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

---

주 - 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 입력합니다.**

**5 다음 옵션 중 하나를 선택하여 유선 이더넷 네트워크 연결을 어떻게 구성할지 지정합니다.**

- DHCP를 사용하여 네트워크 연결을 구성하려면 **Automatically(자동)**를 선택합니다.  
SCI 도구가 시간대 패널로 진행됩니다.
- 네트워킹 사양을 제공하려면 **Manually(수동)**를 선택하고 다음과 같이 계속합니다.
  - a. 인터페이스가 여러 개 있는 경우 구성할 연결을 선택합니다.
  - b. 연결 설정을 입력하거나 SCI 도구에서 감지하여 제공한 기본 정보를 적용합니다.

---

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

---

- c. 시스템이 DNS 이름 서비스를 사용해야 하는지 여부를 지정합니다.
- d. **Configure DNS(DNS 구성)**를 선택한 경우 다음 단계를 계속합니다.
  - i. 시스템에서 사용할 DNS 서버에 대한 IP 주소를 하나 이상 입력합니다.
  - ii. DNS 질의를 실행할 때 검색될 도메인 이름을 하나 이상 제공합니다.



- e. 시스템이 LDAP 이름 서비스, NIS 이름 서비스 또는 None(없음)을 사용해야 하는지 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 프로파일 없이 시스템에 LDAP를 구성하려면 LDAP 대신 None(없음)을 선택합니다. 그런 다음, SCI 도구 프로세스가 완료된 후 수동으로 LDAP를 구성합니다.

---

주 - 선택된 네트워크 이름 지정 서비스가 없으면 /etc/hosts(4)와 같은 표준 이름 소스 파일을 사용하여 네트워크 이름을 분석할 수 있습니다. 자세한 내용은 nsswitch.conf(4) 매뉴얼 페이지를 참조하십시오.

---

- f. 선택한 대체 이름 서비스에 대해 시스템이 상주하는 도메인을 제공합니다.

---

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

---

- g. LDAP를 유일한 이름 서비스로 또는 DNS와 더불어 추가 이름 서비스로 선택한 경우 다음과 같이 LDAP 사양을 제공합니다.

- i. 시스템에 LDAP 이름 서비스를 구성하는 데 사용할 LDAP 프로파일을 지정합니다.

- ii. LDAP 프로파일 서버의 IP 주소를 입력합니다.

- iii. LDAP 검색 기준을 제공하거나 기본 검색 기준을 적용합니다.

- iv. LDAP 프록시 바인드 정보를 제공할지 여부를 지정합니다.

---

주 - 프로파일이 프록시 인증서 레벨을 지정하고 인증 방식이 None(없음)이 아닌 경우 프록시 바인드 정보를 제공해야 합니다. 해당 정보를 생략하면 LDAP가 초기화되지 않습니다.

---

- v. 필요한 경우, LDAP 프록시 바인드 식별 이름 및 프록시 바인드 암호를 제공합니다.

- h. NIS를 유일한 이름 서비스로 또는 DNS와 더불어 추가 이름 서비스로 선택한 경우 NIS 사양을 제공합니다.

소프트웨어에서 이름 서버를 검색하도록 하거나, 직접 이름 서버를 지정할 수 있습니다. 다음 두 가지 옵션 중 하나를 선택합니다.

- **Find One(이름 서버 찾기)**을 선택합니다.

---

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

---

- **Specify One(이름 서버 지정)**을 선택하고 하위 패널에 이름 서버의 호스트 이름 또는 IP 주소를 입력합니다.

일련의 네트워킹 구성 패널을 완료한 후에 SCI 도구에 일련의 시간대 패널과 날짜 및 시간 패널이 표시됩니다.

- **설치 중 네트워크가 구성되지 않도록 지정하려면 None(없음)**을 선택합니다.  
SCI 도구가 시간대 패널로 진행됩니다.

- 6 일련의 시간대 패널에서 먼저 시간대를 선택하고, 로컬 시간과 일치하도록 날짜 및 시간을 조정합니다.

---

주 - 기본값은 GMT 시간대가 구성되도록 하는 것입니다.

---

- 7 사용자 패널을 완성합니다.

반드시 사용자 계정을 만들 필요는 없습니다. 그러나 루트 암호를 만들어야 합니다.

- 이 패널에서 사용자 계정을 만들면 사용자 암호와 루트 암호를 모두 제공해야 합니다.  
이 경우 루트는 사용자에게 할당된 역할이 됩니다.

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

- 사용자 계정을 만들지 않으면 계속 루트 암호를 제공해야 합니다.  
이 경우 루트는 일반 사용자가 됩니다.

- 8 구성 설정을 검토합니다.

- 설정이 맞으면 시스템에 구성을 적용합니다.
- 설정이 틀리면 필요한 횟수만큼 Back 키를 눌러 틀린 정보가 있는 패널로 돌아가서 내용을 변경하고 다시 패널을 계속합니다.

## SCI 도구를 사용하여 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 구성 프로파일을 만들려면 `sysconfig create-profile` 명령을 사용합니다. 프로파일이 만들어지지만, 구성은 시스템에 적용되지 않습니다.

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

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

`-g` 옵션은 구성할 특정 기능 그룹을 지정하는 데 사용됩니다. 이 예에서 전체 시스템이 구성됩니다. 기능 그룹 목록은 [표 6-1](#)을 참조하십시오.

다음 예는 `sysconfig create-profile -o` 옵션을 사용하여 프로파일을 만들 때 다른 출력 파일 위치를 지정합니다. 그런 다음, `sysconfig configure -c` 옵션이 해당 프로파일 위치에 시스템을 재구성합니다.

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

---

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

자세한 내용은 `sysconfig(1M)` 매뉴얼 페이지를 참조하십시오. [11 장](#), “클라이언트 시스템 구성”도 참조하십시오.

---



## 제 3 부

# 설치 서버를 사용하여 설치

이 섹션은 네트워크를 통한 클라이언트 시스템의 자동 설치를 설명합니다.



## 다중 클라이언트의 자동 설치

---

AI(Automated Installer)를 사용하여 네트워크의 여러 클라이언트 시스템에 Oracle Solaris 11 운영 체제(OS)를 설치할 수 있습니다. AI는 SPARC 및 x86 시스템의 핸드프리 설치를 수행합니다. 모든 설치에서 네트워크의 소프트웨어 패키지 저장소에 액세스할 수 있어야 합니다.

### 자동 설치란?

AI는 네트워크를 통한 SPARC 및 x86 클라이언트의 Oracle Solaris 11 OS 설치를 자동화합니다. 디스크 레이아웃, 소프트웨어 선택과 같은 설치 매개변수나 호스트 이름, 네트워크 구성, 사용자 계정과 같은 시스템 구성 매개변수로 클라이언트를 사용자 정의할 수 있습니다. 사용자 정의는 클라이언트 단위로 진행되며 대규모 환경에 맞게 조정할 수 있습니다.

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

1. 클라이언트 시스템이 네트워크를 통해 부트하고 DHCP 서버에서 네트워크 구성과 설치 서버의 위치를 얻습니다.
2. 설치 서버가 클라이언트에 부트 이미지를 제공합니다.
3. 클라이언트 특성에 따라 어떤 설치 지침과 어떤 시스템 구성 지침을 클라이언트 설치에 사용할지 결정합니다.
4. Oracle Solaris 11 OS가 AI 설치 서비스의 설치 지침에 지정된 패키지 저장소에서 패키지를 가져와서 클라이언트에 설치됩니다.

## Automated Installer를 어떻게 사용합니까?

AI를 사용하여 네트워크를 통해 클라이언트 시스템을 설치하려면 DHCP를 설정하고 설치 서버에 AI 설치 서비스를 설정해야 합니다. 8 장, “설치 서버 설정”을 참조하십시오. AI는 DHCP를 사용하여 IP 주소, 서브넷 마스크, 라우터, DNS 서버 및 설치 서버의 위치를 설치될 클라이언트 시스템에 제공합니다. DHCP 서버와 AI 설치 서버는 동일한 시스템일 수도 있고 두 개의 다른 시스템일 수도 있습니다.

설치할 클라이언트 시스템에서 Oracle Solaris IPS(Image Packaging System) 소프트웨어 패키지 저장소에 액세스할 수 있어야 합니다. IPS 패키지 저장소는 설치 서버에, 로컬 네트워크의 다른 서버에 또는 인터넷에 있을 수 있습니다.

AI 설치 서비스에는 SPARC 또는 x86 네트워크 부트 이미지(넷 이미지), 1개 이상의 설치 지침 파일(AI 매니페스트), 0개 이상의 시스템 구성 지침 SMF 프로파일 파일이 포함됩니다. 넷 이미지는 완전한 설치가 아닙니다. 클라이언트 시스템이 설치를 완료하려면 IPS 패키지 저장소에 액세스해야 합니다. 각 클라이언트는 하나의 AI 매니페스트만 사용합니다. 서로 다른 클라이언트는 서로 다른 AI 매니페스트를 사용할 수 있습니다. AI 매니페스트는 클라이언트가 설치를 완료하는 데 필요한 패키지를 검색하는 하나 이상의 IPS 패키지 저장소를 지정합니다. AI 매니페스트는 추가로 설치할 패키지 이름과 대상 장치 및 분할 영역과 같은 정보도 포함합니다. 클라이언트를 부트하기 전에 또는 클라이언트 설치 타임에 동적으로 AI 매니페스트를 사용자 정의하는 방법은 10 장, “클라이언트 시스템 프로비전”을 참조하십시오. 클라이언트 구성 지침을 지정할 수도 있습니다. 시스템 구성 프로파일에 대한 내용은 11 장, “클라이언트 시스템 구성”을 참조하십시오. 클라이언트의 처음 부트 시, 추가 설치 및 구성을 수행하는 방법은 13 장, “처음 부트 중 사용자 정의 스크립트 실행”을 참조하십시오.

두 클라이언트 시스템이 서로 다른 구조를 갖거나 서로 다른 버전의 Oracle Solaris 11 OS로 설치해야 하는 경우 두 개의 AI 설치 서비스를 만들고 각 설치 서비스를 설치할 구조 및 OS 버전에 적절한 이미지 소스와 연관시킵니다. 특정 구조의 첫 번째 설치 서비스가 설치 서버에 만들어질 때 해당 서비스의 복사본인 default-i386 또는 default-sparc이 자동으로 생성됩니다. create-client 하위 명령을 사용하여 다른 설치 서비스와 명시적으로 연관되지 않은 경우 이 기본 서비스가 해당 구조의 모든 클라이언트 설치에 사용됩니다.

두 클라이언트 시스템을 동일한 버전의 Oracle Solaris 11 OS로 설치하되, 다른 방법으로 설치해야 하는 경우 AI 설치 서비스별로 두 개의 AI 매니페스트를 만듭니다. 서로 다른 AI 매니페스트는 설치 대상으로 서로 다른 설치 패키지나 서로 다른 슬라이스를 지정할 수 있습니다.

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.

클라이언트를 부트할 때 설치가 시작됩니다. 클라이언트를 부트하면 DHCP는 AI 설치 서버로 클라이언트를 지정하고, 클라이언트는 올바른 설치 서비스와 해당 서비스 내의



올바른 AI 매니페스트와 시스템 구성 프로파일에 액세스합니다. 15 장, “클라이언트 시스템 설치”는 클라이언트가 특정 설치 서비스와 연관되는 방법을 설명합니다. 9 장, “설치 사용자 정의”는 클라이언트가 올바른 AI 매니페스트와 시스템 구성 프로파일을 식별하는 방법을 설명합니다.

적절한 시스템 구성 지침이 제공되지 않은 경우 설치 후 처음 부트 시, 대화식 도구에서 시스템 구성 정보를 묻습니다. 시스템 구성 프로파일의 정보와 예제는 11 장, “클라이언트 시스템 구성”을 참조하십시오. 대화식 구성 도구에 대한 내용은 63 페이지 “시스템 구성”을 참조하십시오.

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 12 장, “영역 설치 및 구성”을 참조하십시오.

또한 처음 부트 시, Oracle Configuration Manager가 시스템 구성 정보를 수집하여 Oracle Support로 정보를 보내려고 시도합니다. 14 장, “AI 클라이언트 시스템에서 사용할 Oracle Configuration Manager 설정”을 참조하십시오.

## Automated Installer 사용 사례

다음 사용 사례는 주요 AI 사용 방법을 설명합니다. 이러한 사용 사례는 서로 관련이 없습니다. 대신, 각 사례는 별도의 AI 기능을 설명하며 해당 기능에 속하지 않는 모든 동작은 최소 사례와 동일합니다. 아마도 이러한 사용 사례에 설명된 기능을 조합하여 사용하게 됩니다.

### AI 사용을 위한 최소 요구 사항

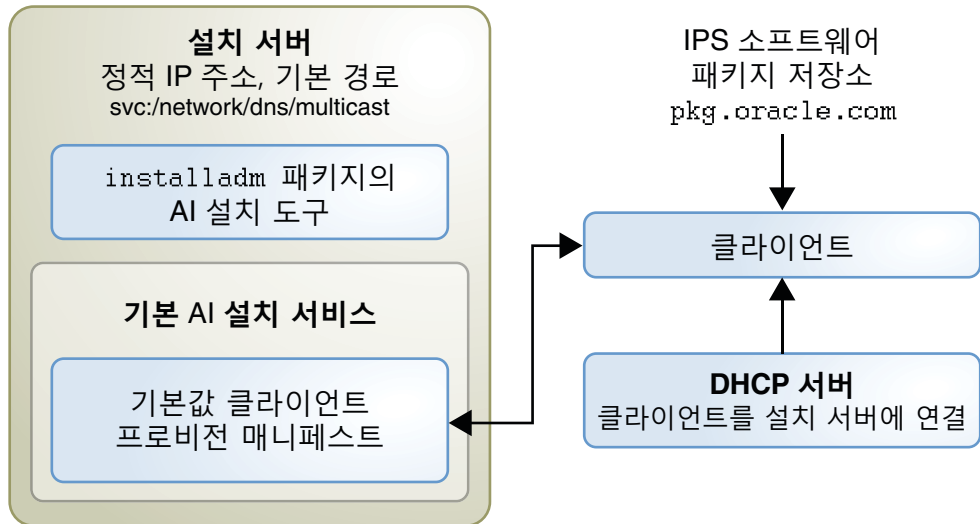
AI 사용을 위한 최소한의 작업은 하나의 설치 서비스를 만드는 것입니다. 이 최소 시나리오에서 모든 클라이언트는 동일한 구조를 갖고 동일한 버전의 Oracle Solaris OS로 설치됩니다. 설치에 기본 AI 매니페스트를 사용하므로 인터넷의 기본 IPS 패키지 저장소에서 사용 가능한 가장 최근 버전의 OS를 지정합니다.

1. 설치 서버가 정적 IP 주소와 기본 라우터를 사용하는지 확인합니다.
2. 설치 도구 패키지 `install/installadm`를 설치합니다.
3. `installadm create-service` 명령을 실행합니다.

특정 구조의 첫번째 설치 서비스가 설치 서버에 만들어질 때 해당 서비스의 복사본인 `default-i386` 또는 `default-sparc`이 자동으로 생성됩니다. `create-client` 하위 명령을 사용하여 다른 설치 서비스와 명시적으로 연관되지 않은 경우 이 기본 서비스가 해당 구조의 모든 클라이언트 설치에 사용됩니다.

4. 클라이언트가 DHCP 서버에 액세스할 수 있는지 확인합니다.
5. 서비스를 부트하는 데 필요한 정보가 DHCP 구성에 제공되었는지 확인합니다.

6. 클라이언트가 IPS 소프트웨어 패키지 저장소에 액세스할 수 있는지 확인합니다. 기본 IPS 패키지 저장소를 사용하려면 클라이언트가 인터넷에 액세스할 수 있어야 합니다.
7. 클라이언트를 네트워크 부트합니다.



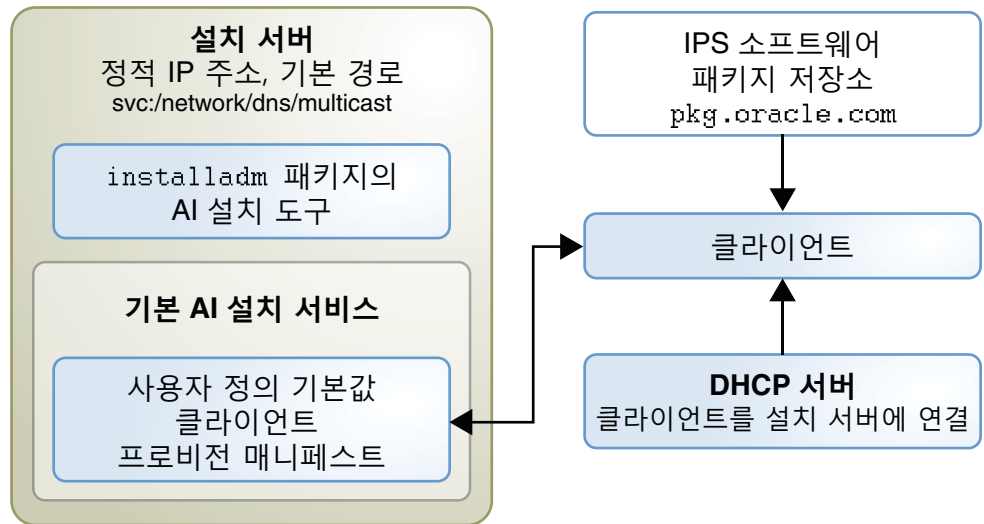
클라이언트를 네트워크 부트할 때 다음 단계가 수행됩니다.

1. 클라이언트가 DHCP 서버에서 설치 서버 주소를 얻습니다.
2. 클라이언트가 구조가 일치하면 `default-arch` 설치 서비스를 사용합니다.
3. 클라이언트가 `default-arch` 설치 서비스의 기본 AI 매니페스트를 사용합니다.  
네트워크를 통해 IPS 패키지 저장소에서 소프트웨어 패키지를 설치하게 됩니다.
4. 클라이언트가 설치 후 부트할 때, 제공된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다.

## 설치 지침 사용자 정의

설치 대상 디스크, 분할 영역, 미러 구성과 같은 설치 매개변수나 추가로 설치할 소프트웨어 패키지를 지정하려면 사용자 정의된 AI 매니페스트를 제공합니다. 최소 필수 단계와 더불어, 클라이언트를 부트하기 전에 다음 단계를 수행합니다.

1. 새 AI 매니페스트를 만들거나, 클라이언트 설치 타임에 사용자 정의 AI 매니페스트를 동적으로 생성하는 스크립트를 작성합니다. 10 장, “클라이언트 시스템 프로비전”을 참조하십시오.
2. `installadm create-manifest` 명령을 실행하여 새 매니페스트나 스크립트를 `default-arch` 설치 서비스에 추가합니다. 클라이언트가 이 매니페스트나 스크립트를 선택하기 위한 조건을 지정합니다.



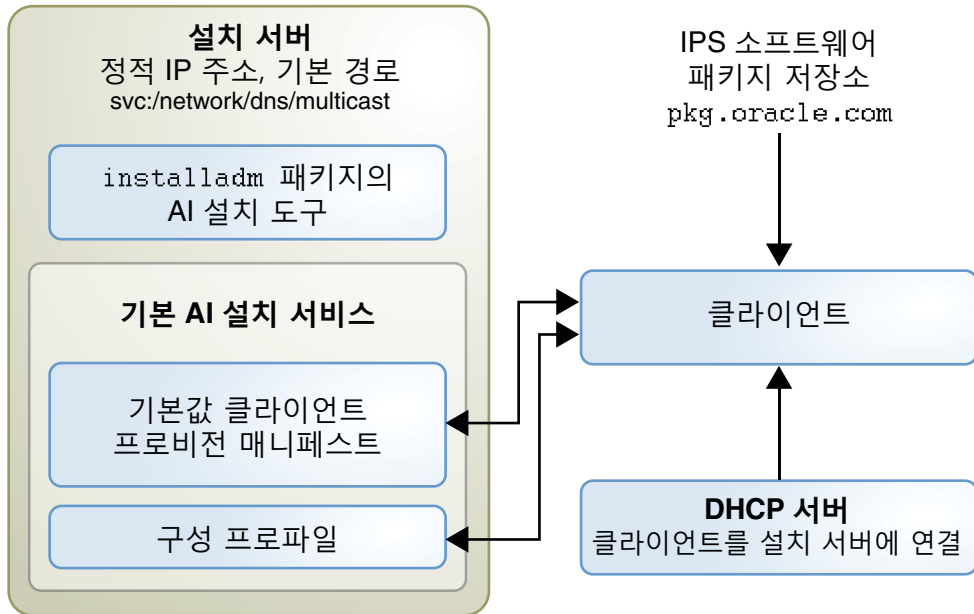
클라이언트를 네트워크 부트할 때 다음 단계가 수행됩니다.

1. 클라이언트가 DHCP 서버에서 설치 서버 주소를 얻습니다.
2. 클라이언트가 구조가 일치하면 `default-arch` 설치 서비스를 사용합니다.
3. 클라이언트가 `create-manifest`에 지정된 조건에 따라 올바른 AI 매니페스트로 지정됩니다. 조건이 일치하지 않으면 클라이언트는 이 서비스에 기본 매니페스트를 사용합니다.
4. 클라이언트가 선택한 AI 매니페스트에 따라 프로비전됩니다.
5. 클라이언트가 설치 후 부트할 때, 제공된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다.

## 시스템 구성 지침 제공

시간대, 사용자 계정, 네트워킹과 같은 시스템 구성 매개변수를 지정하려면 SMF(서비스 관리 기능) 시스템 구성 프로파일 파일을 제공합니다. 최소 필수 단계와 더불어, 클라이언트를 부트하기 전에 다음 단계를 수행합니다.

1. 11 장, “클라이언트 시스템 구성”에 설명된 대로 시스템 구성 프로파일을 만듭니다.
2. `installadm create-profile` 명령을 실행하여 프로파일을 검증하고, `default-arch` 설치 서비스에 프로파일을 추가하고, 이 시스템 구성 프로파일을 사용할 클라이언트를 선택하기 위한 조건을 지정합니다. 지정된 조건이 없는 경우 서비스의 모든 클라이언트가 해당 프로파일을 사용합니다.



클라이언트를 네트워크 부트할 때 다음 단계가 수행됩니다.

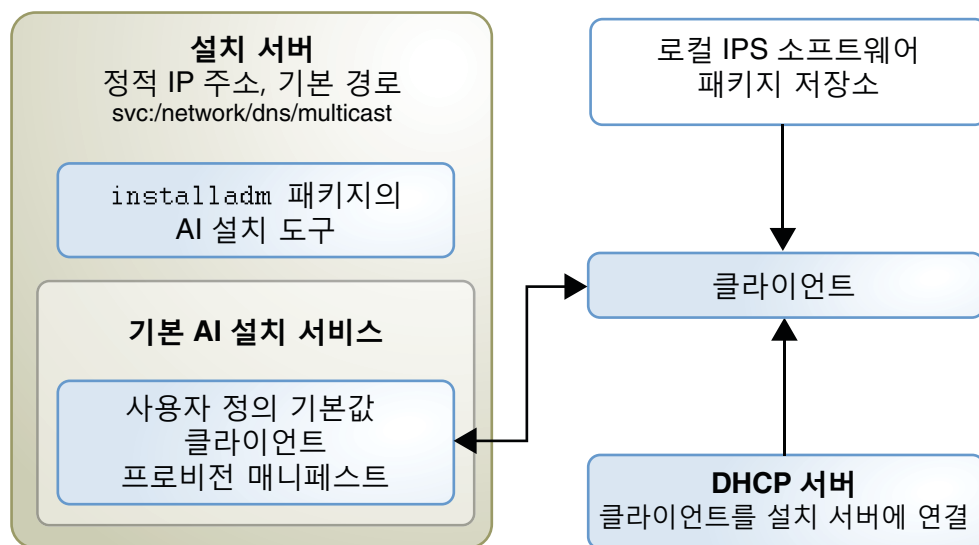
1. 클라이언트가 DHCP 서버에서 설치 서버 주소를 얻습니다.
2. 클라이언트가 구조가 일치하면 `default-arch` 설치 서비스를 사용합니다.
3. 클라이언트가 `default-arch` 설치 서비스의 기본 AI 매니페스트를 사용합니다.  
네트워크를 통해 IPS 패키지 저장소에서 소프트웨어 패키지를 설치하게 됩니다.
4. 클라이언트가 `default-arch` 설치 서비스의 `create-profile`에 지정된 조건에 따라 올바른 시스템 구성 프로파일로 지정됩니다.

- 클라이언트가 선택한 구성 프로파일에 따라 구성됩니다. 조건이 일치하지 않아서 선택된 구성 프로파일이 없는 경우 대화식 구성 도구가 시작됩니다.

## 로컬 IPS 패키지 저장소 제공

클라이언트가 인터넷에 액세스할 수 없거나 기타 이유 등으로 인터넷 패키지 저장소가 아닌 로컬 패키지 저장소를 사용하면 데이터 전송 성능이 향상될 수 있습니다. 최소 필수 단계와 더불어, 클라이언트를 부트하기 전에 다음 단계를 수행합니다.

- IPS 패키지 저장소의 로컬 복사본을 만들고 저장소에서 클라이언트 시스템에 액세스 가능하도록 만듭니다. 지침은 [Oracle Solaris 11 패키지 저장소 복사 및 만들기](#)를 참조하십시오.
- 기본 AI 매니페스트를 사용자 정의하여 새 저장소를 소프트웨어 소스로 지정합니다. 기본 매니페스트를 내보내어 편집하고, `installadm update-manifest` 명령을 실행하여 `default-arch` 설치 서비스의 기본 AI 매니페스트를 편집된 매니페스트로 바꿉니다. 지침은 [10 장, “클라이언트 시스템 프로비전”](#)을 참조하십시오.



클라이언트를 네트워크 부트할 때 다음 단계가 수행됩니다.

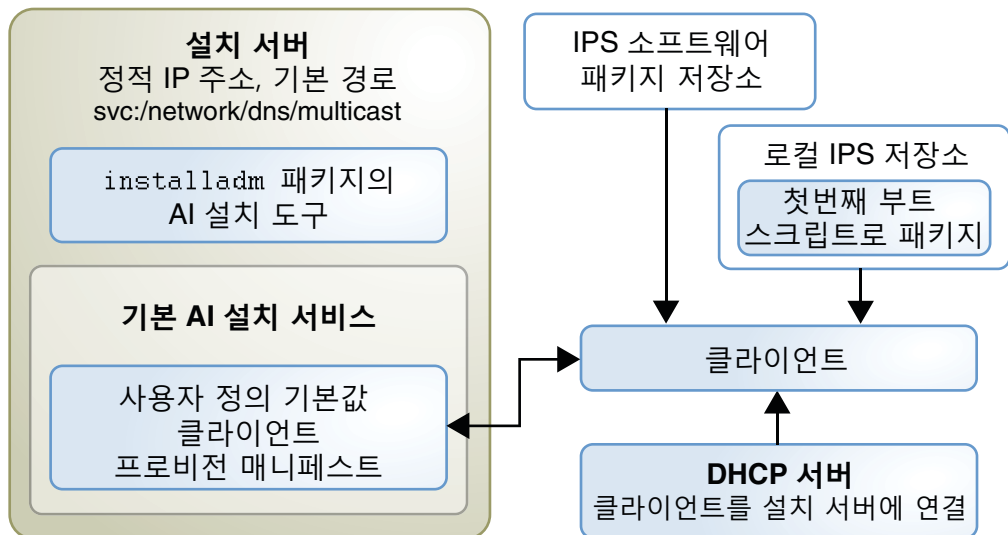
- 클라이언트가 DHCP 서버에서 설치 서버 주소를 얻습니다.
- 클라이언트가 구조가 일치하면 `default-arch` 설치 서비스를 사용합니다.
- 클라이언트가 사용자 정의된 AI 매니페스트에 따라 프로비전되고, 로컬 IPS 패키지 저장소가 사용됩니다.

- 클라이언트가 설치 후 부트할 때, 제공된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다.

## 사용자 정의 처음 부트 스크립트 제공

AI 매니페스트나 시스템 구성 프로파일로 표현할 수 없는 구성을 포함하려면 처음 부트 시 실행할 스크립트를 포함할 수 있습니다. 최소 필수 단계와 더불어, 클라이언트를 부트하기 전에 다음 단계를 수행합니다. 이러한 단계에 대한 자세한 내용은 13 장, “처음 부트 중 사용자 정의 스크립트 실행”을 참조하십시오.

- 클라이언트의 처음 부트 시 실행할 스크립트를 만듭니다.
- 스크립트를 실행할 일회용 SMF 서비스를 만듭니다.
- 서비스와 스크립트에 대한 IPS 패키지를 만들어서 로컬 IPS 저장소에 추가합니다.
- 저장소에서 클라이언트 시스템에 액세스 가능하도록 만듭니다.
- 기본 AI 매니페스트를 사용자 정의하여 새 저장소를 소프트웨어 소스로 지정하고 설치될 새 패키지를 지정합니다. 기본 매니페스트를 내보내어 편집하고, `installadm update-manifest` 명령을 실행하여 `default-arch` 설치 서비스의 기본 AI 매니페스트를 편집된 매니페스트로 바꿉니다. 지침은 10 장, “클라이언트 시스템 프로비전”을 참조하십시오.



클라이언트를 네트워크 부트할 때 다음 단계가 수행됩니다.

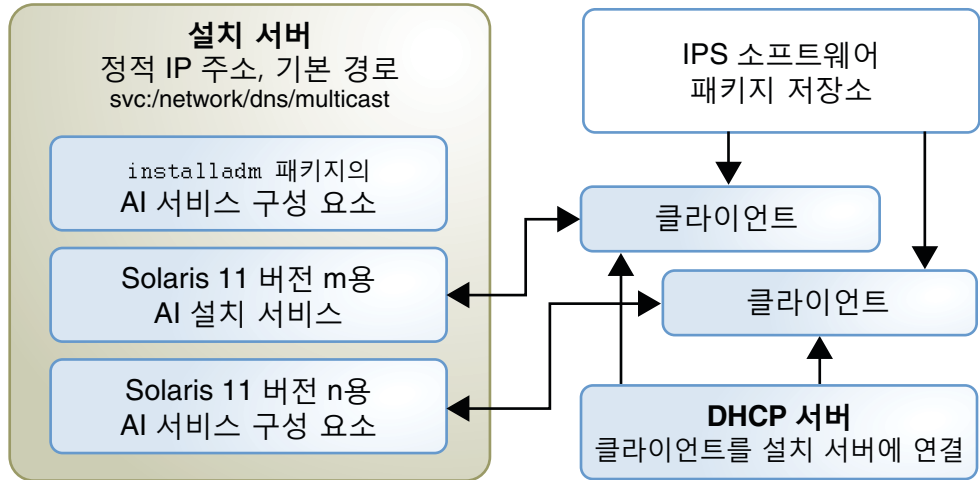
1. 클라이언트가 DHCP 서버에서 설치 서버 주소를 얻습니다.
2. 클라이언트가 구조가 일치하면 `default-arch` 설치 서비스를 사용합니다.
3. 클라이언트가 사용자 정의된 AI 매니페스트에 따라 프로비전되고, 처음 부트 스크립트로 사용자 정의 패키지가 설치됩니다.
4. 클라이언트가 설치 후 부트할 때, 제공된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다.
5. 클라이언트가 설치 후 부트할 때 사용자 정의 일회용 처음 부트 서비스가 실행되고 사용자 정의 스크립트를 실행합니다.

## 추가 AI 설치 서비스 제공

서로 다른 클라이언트 구조에 설치하거나 서로 다른 버전의 Oracle Solaris 11 OS를 설치하려면 8 장, “설치 서버 설정”에 설명된 대로 추가 AI 설치 서비스를 만듭니다. 최소 필수 단계와 더불어, 클라이언트를 부트하기 전에 다음 단계를 수행합니다.

1. `installadm create-service` 명령을 실행하고 설치할 구조 및 OS 버전에 해당하는 소스를 지정합니다.
2. 서로 다른 구조에 대한 첫번째 설치 서비스인 경우 해당 서비스의 복사본인 `default-arch`가 자동으로 생성됩니다. `create-client` 하위 명령을 사용하여 다른 설치 서비스와 명시적으로 연관되지 않은 경우 이 기본 서비스가 해당 구조의 모든 클라이언트 설치에 사용됩니다.

이 새로운 설치 서비스가 기존 설치 서비스와 동일한 구조인 경우 `installadm create-client` 명령을 실행하여 이 구조에 대한 기본 서비스 대신 이 새로운 설치 서비스로 클라이언트를 지정합니다.



클라이언트를 네트워크 부트할 때 다음 단계가 수행됩니다.

1. 클라이언트가 DHCP 서버에서 설치 서버 주소를 얻습니다.
2. 클라이언트가 `create-client`에 의해 새 설치 서비스로 지정되거나, 이 클라이언트에 `create-client`가 실행되지 않은 경우 기본 설치 서비스로 지정됩니다.
3. 클라이언트가 선택한 설치 서비스에 대한 기본 AI 매니페스트에 따라 프로비전됩니다.
4. 클라이언트가 설치 후 부트할 때, 제공된 시스템 구성 프로파일이 없으므로 대화식 도구에서 시스템 구성 정보를 묻습니다.



## 설치 서버 설정

네트워크를 통해 클라이언트를 설치하려면 AI에서 설치 서버로 사용될 별도의 시스템이 필요합니다. 설치 서버에 넷 이미지를 제공하는 AI 설치 서비스를 만들고 서로 다른 클라이언트에 Oracle Solaris 11 OS를 설치하기 위한 지침을 제공할 수 있습니다.

### AI 서버 설정 작업 맵

다음 작업 맵은 AI 설치 서버의 설정 단계를 요약한 것입니다.

표 8-1 작업 맵

작업	참고
AI 설치 서버가 되기 위한 최소 하드웨어 요구 사항을 충족하는지 여부를 확인합니다.	82 페이지 “AI 서버 하드웨어 요구 사항”을 참조하십시오.
AI 설치 서버가 정적 IP 주소 및 기본 경로를 사용하도록 구성합니다. 선택적으로 <code>svc:/network/dns/multicast</code> SMF 서비스를 사용으로 설정합니다. AI 설치 서버가 IPS 소프트웨어 패키지 저장소에 액세스할 수 있는지 확인합니다.	82 페이지 “AI 서버 소프트웨어 요구 사항”을 참조하십시오.
AI 도구 세트를 설치합니다.	83 페이지 “AI 설치 도구 설치”를 참조하십시오.
설치 서비스를 설정합니다.	86 페이지 “AI 설치 서비스 만들기”를 참조하십시오. 설치할 구조마다, 그리고 설치할 서로 다른 버전의 운영 체제마다 별도의 설치 서비스가 필요합니다.

## 설치 서버 요구 사항

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

## AI 서버 하드웨어 요구 사항

다음 요구 사항은 Oracle Solaris 11 OS가 이미 설치되어 있다고 가정합니다. AI 설치 서버에 Oracle Solaris 11 OS를 설치하거나 업데이트해야 하는 경우 메모리 및 디스크 공간 요구 사항은 4 장, “텍스트 설치 프로그램 사용” 및 3 장, “LiveCD 사용”을 참조하십시오.

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

## AI 서버 소프트웨어 요구 사항

- 운영 체제** AI 서버에 Oracle Solaris 11 OS를 설치합니다. AI 서버에 Oracle Solaris 11을 설치하려면 제2부를 참조하십시오.
- 정적 IP 주소** AI 서버가 정적 IP 주소를 사용하도록 구성합니다. **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “IP 인터페이스를 구성하는 방법”**을 참조하십시오.
- 기본 라우터** 네트워크 상태를 표시하는 `netstat(1M)` 명령을 사용하여 AI 서버에 기본 경로가 설정되었는지 확인합니다. AI 서버에 기본 경로가 설정되지 않은 경우 `/etc/defaultrouter(4)` 파일을 서버 네트워크에 대한 정적 기본 경로의 IP 주소로 채워서 정적 기본 경로를 설정할 수 있습니다.
- 멀티캐스트 DNS** 선택 사항: `svc:/network/dns/multicast` SMF 서비스를 사용으로 설정합니다. `svcs(1)` 명령을 사용하여 서비스 상태를 확인하고, 필요한 경우 `svcadm(1M)` 명령을 사용하여 서비스를 사용으로 설정합니다. 예 8-1을 참조하십시오.

**소프트웨어 패키지 저장소**

설치 서버가 IPS 소프트웨어 패키지 저장소에 액세스할 수 있는지 확인합니다. AI에 `install/installadm` 패키지가 필요합니다.

**DHCP**

DHCP를 설정합니다. AI 서버가 DHCP 서버일 수도 있습니다. 다른 방법으로, 이 네트워크에 이미 설정된 DHCP 서버를 사용할 수 있습니다. 각 클라이언트 구조마다 서로 다른 DHCP 구성이 필요합니다. [90 페이지 “로컬 DHCP 설정을 포함한 로컬 서비스 만들기”](#)는 설치 서버에 DHCP 설정의 예를 보여줍니다. DHCP 구성에 대한 자세한 내용은 [Oracle Solaris 관리: IP 서비스의 제II부](#), “DHCP”를 참조하십시오.

**예 8-1 멀티캐스트 DNS 사용**

다음 명령은 `svc:/network/dns/multicast` SMF 서비스의 상태를 확인하고 서비스를 사용으로 설정합니다.

```
# svcs /network/dns/multicast
STATE      STIME      FMRI
disabled   10:19:28   svc:/network/dns/multicast:default
# svcadm enable /network/dns/multicast
# svcs /network/dns/multicast
STATE      STIME      FMRI
online     13:28:30   svc:/network/dns/multicast:default
```

## AI 설치 도구 설치

AI 설치 도구 패키지는 AI 설치 서비스를 만들고 유지 관리할 수 있는 `installadm(1M)` 명령을 제공합니다.

`installadm` 명령으로 다음 작업을 수행할 수 있습니다.

- 설치 서비스를 만들고 사용으로 설정합니다.
- DHCP 서버를 설정하고 업데이트합니다.
- 사용자 정의 클라이언트 설치 및 구성 지침을 추가합니다.
- 클라이언트가 사용자 정의 설치 및 구성 지침을 사용하기 위한 조건을 설정합니다.

`installadm` 명령에 대한 자세한 내용은 [92 페이지 “설치 서버 유지 관리”](#) 및 [Oracle Solaris 11 Installation Man Pages](#)를 참조하십시오.

도구 패키지를 설치하려면 AI 설치 서버에서 Oracle Solaris IPS(Image Packaging System) 소프트웨어 패키지 저장소에 액세스할 수 있어야 합니다. 인터넷이나 `install/installadm` 패키지가 포함된 로컬 IPS 패키지 서버에 연결되어 있는지 확인합니다.

`pkg list` 명령을 사용하여 `installadm` 패키지가 이 시스템에 이미 설치되어 있는지 여부를 확인합니다.

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

`-a` 옵션을 사용하여 IPS 패키지 저장소에 `installadm` 패키지가 있는지 확인합니다.

```
$ pkg list -a installadm
NAME (PUBLISHER)                                VERSION                                IFO
install/installadm                             0.5.11-0.175.0.0.0.0.1345 ---
```

이 이미지에 대해 여러 개의 게시자가 정의된 경우 `-v` 옵션을 사용하여 어떤 게시자가 `installadm` 패키지를 제공하는지 표시합니다.

```
$ pkg list -av installadm
FMRI                                                                IFO
pkg://solaris/install/installadm@0.5.11,5.11-0.175.0.0.0.0.1345:20110815T024057Z ---
```

`pkg publisher` 명령을 사용하여 게시자의 원본을 표시합니다. 이 예에는 `solaris` 저장소의 로컬 복사본이 만들어졌습니다.

```
# pkg publisher
PUBLISHER          TYPE    STATUS  URI
solaris            origin online  file:///export/Solaris11/
example.com (non-sticky) origin online  http://pkg.example.com/
```

`pkg install` 명령을 사용하여 `installadm` 패키지를 설치합니다.

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

DOWNLOAD	PKGS	FILES	XFER (MB)
Completed	1/1	66/66	0.3/0.3

PHASE	ACTIONS
Install Phase	119/119

PHASE	ITEMS
Package State Update Phase	1/1
Image State Update Phase	2/2

PHASE	ITEMS
Reading Existing Index	8/8
Indexing Packages	1/1
Deleting content cache	

```
$ pkg info installadm
Name: install/installadm
Summary: installadm utility
Description: Automatic Installation Server Setup Tools
```

```

Category: System/Administration and Configuration
State: Installed
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.175.0.0.0.1345
Packaging Date: Mon Aug 15 02:40:57 2011
Size: 967.86 kB
FMRI: pkg://solaris/install/installadm@0.5.11,5.11-0.175.0.0.0.1345:20110815T024057Z

```

## 설치 서버 구성

이 섹션에서는 AI 클라이언트 설치 준비를 위해 설치 서버에 수행할 수 있는 구성 중 일부를 설명합니다.

## 멀티홈 설치 서버 구성

기본적으로 AI 설치 서버가 멀티홈인 경우 서버가 연결된 모든 네트워크에 설치 클라이언트를 제공하도록 구성됩니다. 이 구성을 수정하려면

`svc:/system/install/server:default` SMF 서비스의 `all_services/networks` 및 `all_services/exclude_networks` 등록 정보를 조정합니다.

`all_services/networks` 등록 정보 값은 CIDR 형식의 네트워크 목록입니다(예:

`192.168.56.0/24`). `all_services/exclude_networks` 등록 정보 값은

`all_services/networks` 등록 정보의 처리 방법을 지정하는 부울 `true/false`입니다.

`exclude_networks`가 `false`인 경우 AI 설치 서버가 `networks` 등록 정보에 나열된

네트워크만 제공합니다. `exclude_networks`가 `true`인 경우 AI 설치 서버가 `networks` 등록 정보에 나열된 네트워크를 제공하지 않습니다.

다음 명령은 세 개의 네트워크에 연결된 AI 설치 서버를 재구성하여 하나의

네트워크에만 설치를 제공합니다. 이 예에서 멀티홈 AI 설치 서버는 `192.168.56.0/24`,

`205.10.11.0/24`, `205.10.12.0/24`라는 세 개의 네트워크에 연결됩니다. 다음 명령을 실행하여

`192.168.56.0/24` 네트워크에만 설치를 제공합니다.

```

# svccfg -s system/install/server:default \
  setprop all_services/networks = 192.168.56.0/24
# svcadm refresh system/install/server:default

```

## 웹 서버 호스트 포트 구성

AI 서버는 웹 서버를 사용하여 설치 서비스를 호스트합니다. 기본적으로 웹 서버는 포트 5555에 호스트됩니다. 설치 서비스 웹 서버를 호스트하는 포트를 사용자 정의하려면

svc:/system/install/server:default SMF 서비스의 all\_services/port 등록 정보를 구성합니다. 다음 명령은 AI 서버가 포트 7000에서 설치 서비스를 호스트하도록 구성합니다.

```
# svccfg -s system/install/server:default setprop all_services/port = 7000
# svccfg refresh system/install/server:default
```

주 - 설치 서비스를 만들기 전에 port 등록 정보를 사용자 정의합니다. 설치 서비스를 만든 후에 port 등록 정보가 수정된 경우 기존 설치 서비스가 더 이상 제대로 작동하지 않으므로 삭제 후 다시 만들어야 합니다.

## AI 설치 서비스 만들기

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

installadm create-service 명령을 사용하여 AI 설치 서비스를 만듭니다. 서비스에 의미 있는 이름을 부여하고 서비스가 만들어질 경로를 지정합니다. 네트워크 부트 이미지(넷 이미지) 패키지 또는 ISO 파일의 소스를 지정합니다.

AI 설치 서비스를 만들 때 AI SMF 서비스인 system/install/server가 사용으로 설정됩니다(아직 사용으로 설정되지 않은 경우). 설치 서비스 이미지가 /etc/netboot/svcname에 마운트됩니다. SPARC 설치 서비스의 경우 wanboot.conf 파일이 설치 서비스 이미지의 루트에 있습니다. x86 설치 서비스의 경우 menu.lst GRUB 메뉴가 설치 서비스 이미지의 루트에 있습니다.

특정 구조의 첫번째 설치 서비스가 설치 서버에 만들어질 때 해당 서비스의 별칭인 default-i386 또는 default-sparc이 자동으로 생성됩니다. 이 기본 서비스는 고유의 매니페스트와 프로파일을 가진 완전한 서비스입니다. create-client 하위 명령을 사용하여 다른 설치 서비스와 명시적으로 연관되지 않은 경우 이 기본 서비스가 해당 구조의 모든 클라이언트 설치에 사용됩니다.

default-arch 서비스의 별칭으로 삼을 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 [94 페이지 “설치 서비스 등록 정보 수정”](#)을 참조하십시오.

default-arch 별칭이 새 설치 서비스로 변경되고 로컬 ISC DHCP 구성이 발견된 경우 이 기본 별칭 부트 파일이 해당 구조에 대한 기본 DHCP 서버급 부트 파일로 설정됩니다.

새로운 `default-arch` 별칭을 만들 때 로컬 ISC DHCP 서버가 이미 구성된 경우 이 구조에 대한 기본 부트 파일이 이 새로운 별칭의 부트 파일로 설정됩니다.

`default-arch` 서비스를 포함한 각 서비스에는 `imagepath/auto_install/manifest`에 기본 AI 매니페스트가 있습니다. 이 매니페스트를 다른 파일로 복사했다가 `create-manifest` 하위 명령을 사용하여 편집 후 설치 서비스에 추가할 수 있습니다. [118 페이지 “XML AI 매니페스트 파일 사용자 정의”](#)를 참조하십시오.

`installadm create-service` 명령은 포트 5555에 실행 중인 웹 서버에 넷 이미지를 제공합니다. 예를 들어, 웹 서버 주소가 `http://aiserver:5555/export/aiserver/s11-ai-x86/s11-x86`일 수 있습니다.

모든 옵션에 대한 내용은 [92 페이지 “설치 서비스 만들기”](#) 또는 `installadm(1M)` 매뉴얼 페이지를 참조하십시오.

```
installadm create-service [-n svcname] [-s FMRI_or_ISO] [-d imagepath]
```

- `-n svcname`      설치 서비스에 이름을 제공하지 않으면 기본 이름이 할당됩니다.
- `-s FMRI_or_ISO`      *FMRI*는 IPS AI 넷 이미지 패키지의 식별자로, Oracle Solaris 11 릴리스에서 `install-image/solaris-auto-install`입니다. *ISO*는 AI 넷 이미지 ISO 파일의 경로 이름입니다.  
  
*FMRI\_or\_ISO*를 지정하지 않으면 패키지를 제공하는 `pkg publisher` 목록의 첫번째 게시자로부터 최신 버전의 `install-image/solaris-auto-install` 패키지가 설치됩니다.  
  
서로 다른 버전의 패키지를 설치하거나 서로 다른 게시자로부터 패키지를 설치하려면 *FMRI*에 버전 또는 게시자를 지정합니다. 예를 들어, `pkg://publisher/install-image/solaris-auto-install` 또는 `pkg://publisher/install-image/solaris-auto-install@version`을 지정합니다. `-p` 옵션을 사용하여 특정 패키지 저장소를 지정합니다.
- `-d imagepath`      *imagepath*는 새 설치 서비스의 위치입니다. `install-image/solaris-auto-install` 패키지가 이 위치에 설치되거나, 지정된 ISO 파일이 이 위치에 확장됩니다.  
  
*imagepath*를 지정하지 않으면 `/export/auto_install/ svcname`에 서비스가 만들어지고, 자동으로 생성된 위치를 사용할 것인지 확인하는 프롬프트가 나타납니다. 이 프롬프트를 표시하지 않으려면 `-y` 옵션을 지정합니다.

`create-service` 명령은 [90 페이지 “로컬 DHCP 설정을 포함한 로컬 서비스 만들기”](#)에 표시된 대로 AI 설치 서버에 DHCP를 설정할 수 있습니다. 별도의 DHCP 서버를 설정하거나 기존 DHCP 서버가 AI와 함께 사용되도록 구성하려면 [Oracle Solaris 관리: IP 서비스의 제II부, “DHCP”](#)를 참조하십시오. DHCP 서버는 설치될 시스템에 DNS 정보를 제공할 수 있어야 합니다.

## DHCP 설정 없이 설치 서비스 만들기

이 섹션의 예에서 DHCP는 여러 서버에 이미 설정되어 있거나 나중에 설정됩니다. `create-service` 명령이 이 서버에 ISC DHCP가 실행 중임을 감지하지 못하면 명령 출력에 DHCP 구성 지침이 표시됩니다. 이 예에서 `create-service` 명령은 DHCP 구성에 필요한 부트 파일을 제공합니다.

## ISO 파일을 사용하여 SPARC 설치 서비스 만들기

이 예는 ISO 파일의 넷 이미지를 사용하여 SPARC 클라이언트용 AI 설치 서비스를 만듭니다.

```
# installadm create-service -n s11-sparc \
-s /var/tmp/images/sparc/sol-11-dev-170-ai-sparc.iso \
-d /install/images/s11-sparc
```

Creating service: s11-sparc

Setting up the target image at /install/images/s11-sparc ...  
Service discovery fallback mechanism set up  
Creating SPARC configuration file  
Refreshing install services

Creating default-sparc alias.

No local DHCP configuration found. This service is the default alias for all SPARC clients. 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
```

Service discovery fallback mechanism set up  
Creating SPARC configuration file  
Refreshing install services

위의 `installadm create-service` 명령의 실행 결과로 다음 작업이 수행됩니다.

1. 설치 서비스 이름이 `s11-sparc`으로 지정됩니다.
2. 설치 서비스 대상 디렉토리 `/install/images/s11-sparc`이 만들어집니다.
3. ISO 파일 `/var/tmp/images/sparc/sol-11-dev-170-ai-sparc.iso`가 넷 이미지 위치 `/install/images/s11-sparc`으로 압축이 풀립니다.
4. 이 서비스의 `wanboot.conf` 파일이 `/install/images/s11-sparc/wanboot.conf`에 생성됩니다.
5. AI SMF 서비스 `system/install/server`를 새로 고쳐서 `/install/images/s11-sparc`을 `/etc/netboot/s11-sparc`으로 마운트합니다.
6. 이 설치 서버에 생성된 첫번째 SPARC 설치 서비스인 경우 `default-sparc` 서비스 별칭이 자동으로 생성됩니다. `s11-sparc`의 이미지가 별칭에 사용되므로 `/install/images/s11-sparc`이 `/etc/netboot/default-sparc`으로 마운트됩니다.



7. 구성 파일 `/etc/netboot/wanboot.conf`가  
`/etc/netboot/default-sparc/wanboot.conf`로 심볼릭 링크됩니다. 구성 파일  
`/etc/netboot/system.conf`가 `/etc/netboot/default-sparc/system.conf`로 심볼릭  
 링크됩니다.
8. DHCP 구성에 필요한 부트 파일 `http://10.80.238.5:5555/cgi-bin/wanboot-cgi`가  
 제공됩니다.
9. 로컬 ISC DHCP 서버가 이미 구성된 경우 새로운 `default-sparc` 별칭의 부트 파일이  
 모든 SPARC 클라이언트의 기본 부트 파일로 설정됩니다. 이는 `-i` 및 `-c` 옵션의 사용  
 여부에 관계없이 적용됩니다.

## IPS 패키지를 사용하여 x86 설치 서비스 만들기

이 예는 IPS 패키지의 넷 이미지를 사용하여 x86 클라이언트용 AI 설치 서비스를  
 만듭니다. 이 명령은 설치 서비스 이름 옵션만 제공하므로 옵션이 지정되지 않았을 때  
 기본 동작을 보여줍니다. DHCP 구성에 필요한 부트 파일 외에, DHCP 구성에 필요한  
 부트 서버 IP도 제공합니다.

```
# installadm create-service -n s11-i386 -y

Creating service from: pkg:/install-image/solaris-auto-install
Download: install-image/solaris-auto-install ... Done
Install Phase ... Done
Package State Update Phase ... Done
Image State Update Phase ... Done
Reading Existing Index ... Done
Indexing Packages ... Done

Creating service: s11-i386

Image path: /export/auto_install/s11-i386

Refreshing install services

Creating default-i386 alias.

No local DHCP configuration found. This service is the default
alias for all PXE clients. 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

Refreshing install services
```

위의 `installadm create-service` 명령의 실행 결과로 다음 작업이 수행됩니다.

1. 설치 서비스 이름이 `s11-i386`으로 지정됩니다.
2. 지정된 넷 이미지 소스 옵션이 없으므로 패키지를 제공하는 설치 서버 게시자 목록의  
 첫번째 게시자로부터 최신 버전의 `install-image/solaris-auto-install` 패키지를  
 얻습니다.

3. `-d` 옵션으로 지정된 넷 이미지 대상이 없으므로 기본 디렉토리 `/export/auto_install/s11-i386`에 이미지가 만들어집니다. `y` 옵션이 지정되었으므로 이 기본 대상을 수락할 것인지 확인하는 프롬프트가 표시되지 않습니다.
4. `install-image/solaris-auto-install` 패키지가 넷 이미지 위치 `/export/auto_install/s11-i386`으로 설치됩니다.  
기본적으로 설치된 `install-image/solaris-auto-install` 패키지의 변형은 AI 설치 서버의 구조와 일치합니다. 이 예에서 설치 서버는 x86 시스템입니다. 이 서버에 SPARC 설치 서비스를 만들려면 `-a` 옵션을 사용해야 합니다. `-a` 옵션에 대한 내용은 92 페이지 “설치 서비스 만들기”를 참조하십시오.
5. `pxegrub` 메뉴가 `/export/auto_install/s11-i386/menu.lst`에 만들어집니다.
6. AI SMF 서비스 `system/install/server`를 새로 고쳐서 `/export/auto_install/s11-i386`을 `/etc/netboot/s11-i386`으로 마운트합니다.
7. 이 설치 서버에 생성된 첫 번째 x86 설치 서비스이므로 `default-i386` 서비스 별칭이 자동으로 생성됩니다. `s11-i386`의 이미지가 별칭에 사용되므로 `/export/auto_install/s11-i386`이 `/etc/netboot/default-i386`으로 마운트됩니다.
8. DHCP 구성에 필요한 부트 서버 IP가 제공됩니다. DHCP 구성에 필요한 부트 파일 `default-i386/boot/grub/pxegrub`가 제공됩니다.
9. 로컬 ISC DHCP 서버가 이미 구성되었으므로 새로운 `default-i386` 별칭의 부트 파일이 모든 x86 클라이언트의 기본 부트 파일로 설정됩니다. 이는 `-i` 및 `-c` 옵션의 사용 여부와 관계없이 적용됩니다.

## 로컬 DHCP 설정을 포함한 로컬 서비스 만들기

`installadm create-service` 명령을 사용하여 이 AI 설치 서버에 DHCP 서버를 설정할 수 있습니다. 다음 예는 호스트 이름을 분석하는 DNS를 사용하여 네트워크가 단일 서브넷으로 구성되고 설치 서버가 네트워크의 DHCP 서버 역할도 하는 x86 클라이언트용 설치 서비스를 만듭니다. 이 설치 서비스는 10.80.239.150(`-i`)부터 시작하여 20개 IP 주소(`-c`)를 제공합니다. DHCP 서버가 아직 구성되지 않은 경우 ISC DHCP 서버가 구성됩니다. ISC DHCP 서버가 이미 구성된 경우 해당 DHCP 서버가 업데이트됩니다.

`-i` 및 `-c` 인수가 제공되고 DHCP가 구성된 경우 생성 중인 설치 서비스와 IP 범위 사이에 바인딩이 존재하지 않습니다. `-i` 및 `-c`를 전달할 때 IP 범위가 설정되고, 필요한 경우 새 DHCP 서버가 만들어지고, 모든 설치 서비스와 모든 사용할 클라이언트에 대해 해당 DHCP 서버가 계속 작동 및 실행 중입니다. DHCP 서버에 제공된 네트워크 정보는 생성 중인 서비스와 특별한 관계가 없습니다.

요청된 IP 범위가 설치 서버가 직접 연결된 서브넷에 없고 설치 서버가 멀티홈인 경우 부트 파일 서버 주소(대개 이 시스템의 IP 주소)를 제공하는 `-B` 옵션이 사용됩니다. 이는

반드시 다중 IP 주소가 설치 서버에 구성되고 DHCP 릴레이가 사용될 때만 필요합니다. 다른 모든 구성에서는 소프트웨어가 이를 자동으로 결정할 수 있습니다.

```
# installadm create-service -n s11-x86 \
-s /var/tmp/images/i386/sol-11-dev-171-ai-x86.iso \
-d /install/images/s11-x86 \
-i 10.80.239.150 -c 20
```

```
Creating service from: /var/tmp/images/i386/sol-11-dev-171-ai-x86.iso
Setting up the image ...
```

```
Creating service: s11-x86
```

```
Image path: /install/images/s11-x86
```

```
Starting DHCP server...
```

```
Adding IP range to local DHCP configuration
```

```
Refreshing install services
```

```
Creating default-i386 alias.
```

```
Setting the default PXE bootfile in the local DHCP configuration to
'default-i386/boot/grub/pxegrub'
```

```
Refreshing install services
```

위의 `installadm create-service` 명령의 실행 결과로 다음 작업이 수행됩니다.

1. 설치 서비스 이름이 `s11-x86`으로 지정됩니다.
2. 설치 서비스 대상 디렉토리 `/install/images/s11-x86`이 만들어집니다.
3. ISO 파일 `/var/tmp/images/i386/sol-11-dev-171-ai-x86.iso`가 넷 이미지 위치 `/install/images/s11-x86`으로 압축이 풀립니다.
4. `pxegrub` 메뉴가 `/install/images/s11-x86/menu.lst`에 만들어집니다.
5. AI SMF 서비스 `system/install/server`를 새로 고쳐서 `/install/images/s11-x86`을 `/etc/netboot/s11-x86`으로 마운트합니다.
6. 이 설치 서버에 생성된 첫번째 `x86` 설치 서비스이므로 `default-i386` 서비스 별칭이 자동으로 생성됩니다. `s11-x86`의 이미지가 별칭에 사용되므로 `/install/images/s11-x86`이 `/etc/netboot/default-i386`으로 마운트됩니다.
7. 필요한 경우 DHCP 서비스가 만들어지고, IP 주소가 10.80.239.150부터 10.80.239.169까지 프로비전됩니다. DHCP 서비스가 이 서버에 이미 설정된 경우 `-i` 및 `-c` 옵션이 DHCP 서버를 이 서비스의 새 IP 주소로 업데이트합니다.
8. `default-i386/boot/grub/pxegrub` 부트 파일이 PXE 클라이언트의 기본 부트 파일로 로컬 DHCP 구성에 추가됩니다.

## 설치 서버 유지 관리

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

- 92 페이지 “설치 서비스 추가, 수정 또는 삭제”
- 95 페이지 “클라이언트를 설치 서비스와 연관”
- 96 페이지 “클라이언트별 설치 지침을 설치 서비스와 연관”
- 102 페이지 “설치 서버의 모든 설치 서비스 나열”
- 103 페이지 “설치 서비스와 연관된 클라이언트 나열”
- 104 페이지 “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”

## 설치 서비스 추가, 수정 또는 삭제

설치할 서로 다른 클라이언트 구조마다, 그리고 클라이언트 시스템에 설치할 서로 다른 버전의 Oracle Solaris 11 OS마다 별도의 설치 서비스가 필요합니다.

### 설치 서비스 만들기

다음 명령을 사용하여 설치 서비스를 만듭니다. 예제는 [86 페이지 “AI 설치 서비스 만들기”](#)를 참조하십시오.

```
installadm create-service [-n svcname] [-s FMRI_or_ISO] [-p prefix=origin] [-a architecture] [-d imagepath] [-y] [-t exist]
```

**-n svcname**

*svcname*은 영숫자, 밑줄(\_), 하이픈(-)으로 구성할 수 있습니다. *svcname*의 첫 글자는 하이픈일 수 없습니다. 설치 서비스에 이름을 제공하지 않으면 기본 이름이 할당됩니다.

**-s FMRI\_or\_ISO**

이 옵션은 넷 부트 이미지의 소스를 지정합니다. *FMRI*는 IPS AI 넷 이미지 패키지의 식별자로, Oracle Solaris 11 릴리스에서 `install-image/solaris-auto-install`입니다. AI 넷 이미지 ISO 파일을 사용하는 경우 넷 이미지 ISO 파일의 경로 이름을 지정합니다.

*FMRI\_or\_ISO*를 지정하지 않으면 패키지를 제공하는 pkg publisher 목록의 첫번째 게시자로부터 최신 버전의 `install-image/solaris-auto-install` 패키지가 설치됩니다.

서로 다른 버전의 패키지를 설치하거나 서로 다른 게시자로부터 패키지를 설치하려면 *FMRI*에 버전 또는 게시자를 지정합니다. 예를 들어,  
`pkg://publisher/install-image/solaris-auto-install`  
 또는

	pkg://publisher/install-image/solaris-auto-install@version을 지정합니다. -p 옵션을 사용하여 특정 패키지 저장소를 지정합니다.
-p prefix= origin	이 옵션은 install-image/solaris-auto-install 패키지를 검색하려는 IPS 패키지 저장소를 지정합니다. prefix는 게시자 이름이고 origin은 solaris=http://pkg.oracle.com/solaris/release/와 같은 URI입니다.
	-s 및 -p가 지정되지 않은 경우 패키지를 제공하는 pkg publisher 목록의 첫번째 게시자로부터 최신 버전의 install-image/solaris-auto-install 패키지가 설치됩니다.
-a architecture	이 옵션은 넷 이미지 소스가 IPS 패키지일 때만 사용됩니다. architecture는 설치될 클라이언트의 구조를 지정합니다. i386 또는 sparc을 지정할 수 있습니다.
	IPS 패키지에서 서비스를 만들 때 기본적으로 설치되는 패키지의 변형은 서비스가 생성될 시스템의 구조와 일치하는 변형입니다. 예를 들어, AI 설치 서버가 x86인 경우 create-service가 기본적으로 설치하는 solaris-auto-install 패키지의 변형은 i386 변형입니다. SPARC 클라이언트를 설치할 서비스를 만드는 경우 -a sparc을 지정하여 설치 서비스에 solaris-auto-install 패키지의 sparc 변형을 설치합니다.
-d imagepath	imagepath는 새 설치 서비스의 위치입니다. imagepath를 지정하지 않으면 /export/auto_install/svcname에 서비스가 만들어지고, 자동으로 생성된 위치를 사용할 것인지 확인하는 프롬프트가 나타납니다. 이 프롬프트를 표시하지 않으려면 -y 옵션을 지정합니다.
-y	자동으로 생성된 imagepath를 사용할 것인지 확인하는 프롬프트를 표시하지 않으려면 -y 옵션을 지정합니다.
-t existing_service	existing_service 서비스의 넷 이미지를 공유하지만 고유의 매니페스트, 프로파일, 클라이언트를 가진 새 서비스를 별칭으로 지정합니다.
-i dhcp_ip_start	이 옵션은 로컬 DHCP 구성에 추가할 범위의 시작 IP 주소를 지정합니다. IP 주소 수는 -c 옵션으로 제공됩니다. 로컬 ISC DHCP 구성이 없는 경우 ISC DHCP 서버가 시작됩니다.

- c *count\_of\_ipaddr* DHCP 구성의 총 IP 주소 수가 *count\_of\_ipaddr* 값과 같도록 설정합니다. 첫번째 IP 주소는 -i 옵션으로 제공된 *dhcp\_ip\_start* 값입니다.
- b *boot\_property= value,...* x86 서비스만 해당합니다. 이 옵션은 서비스 이미지에서 서비스별 *menu.lst* 파일의 등록 정보 값을 설정합니다. 이 옵션을 사용하여 이 서비스에 특정한 부트 등록 정보를 설정합니다. 이 옵션은 콤마로 구분된 *boot\_property= value* 쌍을 받아들일 수 있습니다.
- B *server\_ipaddr* 이 옵션을 사용하여 어떤 클라이언트가 부트 파일을 요청할 부트 서버의 IP 주소를 제공합니다. 이 옵션은 다른 방법으로 이 IP 주소를 결정할 수 없는 경우에만 필요합니다.

## 설치 서비스 등록 정보 수정

`installadm set-service` 명령을 사용하여 *svcname* 설치 서비스에 설정할 등록 정보 및 값을 지정합니다.

```
installadm set-service -o prop=value svcname
```

*prop=value* 쌍은 다음 중 하나여야 합니다.

*aliasof=another\_svcname*  
*svcname* 서비스의 별칭으로 삼을 설치 서비스를 변경합니다.

이 등록 정보를 설정하면 *svcname* 서비스가 *another\_svcname* 서비스의 별칭이 되도록 변경됩니다. *svcname* 서비스는 이미 별칭이어야 합니다. `default-arch` 설치 서비스는 별칭입니다. `create-service`의 -t 옵션으로 만든 서비스는 별칭입니다. [102 페이지 “설치 서버의 모든 설치 서비스 나열”](#)에 표시된 대로 `installadm list` 명령을 사용하여 *svcname*가 별칭인지 확인합니다.

*svcname* 또는 *another\_svcname*에 추가된 매니페스트, 프로파일, 클라이언트 바인딩은 별칭을 재설정 한 후에도 그대로 유지됩니다. *svcname* 서비스가 사용하는 넷 이미지만 변경됩니다.

별칭을 설정하기 전에 *svcname*에 추가된 매니페스트와 프로파일은 별칭을 재설정할 때 다시 검증됩니다. 새로운 넷 이미지와 연관된 AI 및 SMF DTD가 서로 다를 수 있기 때문입니다. 이 검증은 아래 설명된 `create-manifest` 및 `create-profile`로 수행된 것과 동일한 검증입니다.

```
default-manifest=manifest_or_script_name
```

주어진 서비스에 이미 등록된 특정 매니페스트나 스크립트가 해당 서비스의 기본 매니페스트나 스크립트가 되도록 지정합니다. 다음 명령을 사용하여 이 서비스에 등록된 매니페스트 및 스크립트 목록을 표시합니다.

```
$ installadm list -n svcname -m
```

## 설치 서비스 이름 바꾸기

다음 명령을 사용하여 *svcname* 이름을 *newsvcname*으로 바꿉니다.

```
installadm rename-service svcname newsvcname
```

*newsvcname*은 영숫자, 밑줄(\_), 하이픈(-)으로 구성할 수 있습니다. *newsvcname*의 첫 글자는 하이픈일 수 없습니다.

## 설치 서비스 사용 또는 사용 안함

다음 명령을 사용하여 *svcname* 설치 서비스를 사용으로 설정합니다.

```
installadm enable svcname
```

다음 명령을 사용하여 *svcname* 설치 서비스를 사용 안함으로 설정합니다.

```
installadm disable svcname
```

## 설치 서비스 삭제

다음 명령을 사용하여 *svcname* 설치 서비스를 삭제합니다.

```
installadm delete-service [-r] [-y] svcname
```

이 명령은 *svcname* 설치 서비스에 대한 AI 매니페스트와 시스템 구성 프로파일, 넷 이미지 및 웹 서버 구성을 삭제합니다. 서비스가 기본 별칭이고 로컬 ISC DHCP 구성이 존재할 경우 이 서비스와 연관된 부트 파일이 ISC DHCP 구성에서 제거됩니다.

-r 옵션을 사용하여 이 서비스와 연관된 클라이언트 및 이 서비스로 별칭 지정된 서비스를 제거합니다. -y 옵션을 사용하여 확인 프롬프트를 표시하지 않습니다.

## 클라이언트를 설치 서비스와 연관

`installadm create-client` 명령은 클라이언트를 특정 설치 서비스와 연관시킵니다. 추가 예제와 샘플 출력은 [183 페이지 “설치 클라이언트 설정”](#)을 참조하십시오.

## 클라이언트를 설치 서비스에 추가

`installadm create-client` 명령을 사용하여 *macaddr* 클라이언트를 *svcname* 설치 서비스와 연관시키고 x86 클라이언트용 사용자 정의 클라이언트 설정을 제공합니다. 시스템의 MAC 주소를 찾으려면 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화](#) 및 [dladm\(1M\)](#) 매뉴얼 페이지에 설명된 대로 `dladm` 명령을 사용하십시오.

```
installadm create-client [-b property=value,...] -e macaddr -n svcname
```

클라이언트가 x86 시스템이고 로컬 ISC DHCP 구성이 존재할 경우 ISC DHCP 구성에 클라이언트가 구성됩니다.

x86 클라이언트 시스템의 경우 **-b** 옵션을 사용하여 `/etc/netboot`에서 클라이언트별 `menu.lst` 파일의 부트 등록 정보를 설정합니다.

다음 명령은 MAC 주소 `00:14:4f:a7:65:70`의 클라이언트를 `s11-sparc` 설치 서비스에 추가합니다.

```
# installadm create-client -e 00:14:4f:a7:65:70 -n s11-sparc
```

다음 예는 x86 클라이언트를 추가하고 설치 출력을 직렬 콘솔로 재지정합니다.

```
# installadm create-client -e c0ffec0ffee -n s11-x86 -b 'console=ttya'
```

## 클라이언트를 서로 다른 설치 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.

## 클라이언트를 설치 서비스에서 삭제

`installadm delete-client` 명령을 사용하여 `macaddr` 클라이언트를 연관된 설치 서비스에서 삭제합니다.

```
installadm delete-client macaddr
```

클라이언트가 x86 시스템이고 로컬 ISC DHCP 구성이 존재할 경우 ISC DHCP 구성에서 클라이언트 구성이 해제됩니다.

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

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

## 클라이언트별 설치 지침을 설치 서비스와 연관

각 설치 서비스에 대해 여러 세트의 설치 지침을 지정할 수 있고, 각 클라이언트에 사용할 명령 집합을 지정할 수 있습니다.

## AI 매니페스트 추가

`installadm create-manifest` 명령을 사용하여 `manifest_or_script_filename` 사용자 정의 AI 매니페스트를 `svcname` 설치 서비스에 추가합니다.

```
installadm create-manifest -n svcname -f manifest_or_script_filename [-m manifest_or_script_name] [-c criteria=value|list|range]
```



*manifest\_or\_script\_filename*은 AI 매니페스트 XML 파일일 수도 있고, 파생된 매니페스트 스크립트일 수도 있습니다. 10 장, “클라이언트 시스템 프로비전”을 참조하십시오. `create-manifest` 하위 명령은 XML 매니페스트 파일을 설치 서비스에 추가하기 전에 검증합니다. 파생된 매니페스트 스크립트 파일을 검증하려면 133 페이지 “파생된 매니페스트 스크립트를 설치 서비스에 추가”에 표시된 대로 `aimanifest validate` 명령을 사용합니다.

*manifest\_or\_script\_name*은 `installadm list` 명령으로 표시된 이름입니다. 104 페이지 “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”을 참조하십시오.

*manifest\_or\_script\_name*이 제공되지 않은 경우 *manifest\_or\_script\_name*은 `ai_instance` 요소의 `name` 속성 값이거나 *manifest\_or\_script\_filename*의 기본 이름입니다.

`-d` 옵션은 이 AI 매니페스트를 기본 AI 매니페스트로 만듭니다. 기본 매니페스트는 이 설치 서비스의 다른 매니페스트에 대해 지정된 조건과 일치하지 않는 클라이언트에서 사용되는 매니페스트입니다. `-d`가 지정된 경우 매니페스트 선택 목적으로 `-c` 및 `-C`가 무시됩니다. 이 서비스에 대한 이전의 기본 AI 매니페스트는 클라이언트 조건이 없는 경우 비활성이 됩니다. 이전의 기본 매니페스트에 조건이 있는 경우 활성을 유지하고 연관된 조건이 효력을 발휘합니다.

`-d`가 지정되지 않은 경우 어떤 클라이언트가 이 AI 매니페스트를 사용하여 설치를 완료할지 정의하려면 `-c` 또는 `-C`를 지정해야 합니다. `-d`, `-c`, `-C`가 모두 지정되지 않은 경우 이 매니페스트가 서비스에 추가되지만 비활성입니다. 즉, 클라이언트가 사용할 수 없습니다.

특정 클라이언트가 이 AI 매니페스트를 사용하도록 하려면 먼저 해당 클라이언트가 이 `create-manifest` 명령에 지정된 설치 서비스를 사용할지 확인합니다. `create-client` 명령을 사용하여 특정 설치 서비스와 명시적으로 연관되지 않은 클라이언트 시스템은 적절한 `default-arch` 설치 서비스를 사용합니다. 사용자 정의된 AI 매니페스트를 `default-arch` 설치 서비스에 추가할 수도 있고, 사용자 정의된 AI 매니페스트를 서로 다른 서비스에 추가한 다음 `create-client`를 통해 클라이언트가 해당 서비스를 사용하도록 만들 수도 있습니다.

`-c` 옵션은 명령줄에 클라이언트 선택 조건을 지정합니다. `-c` 옵션은 XML 파일에 조건을 지정합니다. *criteriafile* 값은 전체 경로 및 파일 이름입니다. 명령줄의 조건 키워드와 파일 예제 목록은 9 장, “설치 사용자 정의”를 참조하십시오.

`installadm create-manifest` 명령은 동일한 유형의 조건이 겹치지 않는지 확인합니다. 예를 들어, 한 조건 사양이 IP 주소 10.0.0.0부터 10.255.255.255까지 일치하는 경우 IP 주소 10.10.10.10과 일치하는 조건 사양을 추가하려고 시도하면 `installadm`이 오류와 함께 종료됩니다. 조건 사양에 대한 자세한 내용은 9 장, “설치 사용자 정의”를 참조하십시오.

다음 명령은 `manifest_t200.xml` 매니페스트를 `s11-sparc` 설치 서비스에 추가합니다. `-c` 옵션은 이 설치 서비스를 사용 중인 Sun Fire T200 서버로 식별된 클라이언트가 `manifest_t200.xml` 설치 지침에 할당되도록 지정합니다.

```
# installadm create-manifest -f ./mymanifests/manifest_t200.xml \
-m t200 -n s11-sparc -c platform="SUNW,Sun-Fire-T200"
```

다음 명령은 `criteria_t200.xml` 파일의 내용이 아래와 같을 경우 앞의 명령과 동등한 기능을 합니다.

```
# installadm create-manifest -f ./mymanifests/manifest_t200.xml \
-m t200 -n s11-sparc -C ./mymanifests/criteria_t200.xml
```

다음은 `criteria_t200.xml` 파일의 내용입니다.

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,Sun-Fire-T200</value>
  </ai_criteria>
</ai_criteria_manifest>
```

## AI 매니페스트 업데이트

`installadm update-manifest` 명령을 사용하여 `manifest_or_script_name` AI 매니페스트의 내용을 `svcname` 설치 서비스의 `manifest_or_script_filename` AI 매니페스트로 바꿉니다. 조건, 기본 상태 및 `manifest_or_script_name`은 업데이트 결과로 바뀌지 않습니다.

```
installadm update-manifest -n svcname
-f manifest_or_script_filename [-m manifest_or_script_name]
```

`update-manifest` 하위 명령은 XML 매니페스트 파일을 설치 서비스에 추가하기 전에 검증합니다. 파생된 매니페스트 스크립트 파일을 검증하려면 [133 페이지 “파생된 매니페스트 스크립트를 설치 서비스에 추가”](#)에 표시된 대로 `aimanifest validate` 명령을 사용합니다.

`manifest_or_script_name` 매니페스트는 `svcname` 서비스에 이미 존재해야 합니다. `installadm list` 명령을 사용하여 확인합니다. [104 페이지 “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”](#)을 참조하십시오.

`manifest_or_script_name`이 지정되지 않은 경우 다음 방법 중 하나로 교체할 매니페스트를 식별할 수 있습니다.

- `manifest_or_script_filename` 매니페스트에서 `ai_instance` 요소의 `name` 속성. 이 속성이 지정되고 이 속성 값이 이 설치 서비스에 대한 기존 매니페스트의 `manifest_or_script_name`과 일치하는 경우입니다.
- `manifest_or_script_filename` 매니페스트의 기본 이름. 이 이름이 이 설치 서비스에 대한 기존 매니페스트의 `manifest_or_script_name`과 일치하는 경우입니다.

다음 명령은 `s11-sparc` 서비스에서 `t200` 매니페스트의 내용을 `./mymanifests/manifest_newt200.xml`의 내용으로 업데이트합니다. `installadm list`의 매니페스트 이름은 여전히 `t200`입니다.

```
# installadm update-manifest -n s11-sparc \
-f ./mymanifests/manifest_newt200.xml -m t200
```

## AI 매니페스트 삭제

`installadm delete-manifest` 명령을 사용하여 *manifest\_or\_script\_name* AI 매니페스트를 *svcname* 설치 서비스에서 제거합니다. *manifest\_or\_script\_name*은 `installadm list` 명령이 반환하는 매니페스트 이름입니다. [104 페이지 “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”](#)을 참조하십시오.

```
installadm delete-manifest -m manifest_or_script_name -n svcname
```

기본 AI 매니페스트는 삭제할 수 없습니다.

다음 명령은 t200 AI 매니페스트를 s11-sparc 설치 서비스에서 제거합니다.

```
# installadm delete-manifest -m t200 -n s11-sparc
```

## 클라이언트별 구성 지침을 설치 서비스와 연관

각 설치 서비스에 대해 여러 세트의 시스템 구성 지침을 지정할 수 있습니다. 각 클라이언트와 여러 시스템 구성 프로파일이 연관될 수 있습니다.

### 시스템 구성 프로파일 추가

`installadm create-profile` 명령을 사용하여 *profile\_filename* 시스템 구성 프로파일을 *svcname* 설치 서비스에 추가합니다.

```
installadm create-profile -n svcname
  -f profile_filename... [-p profile_name]
  [-c criteria=value|list|range... | -C criteriafile]
```

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

`create-profile` 하위 명령은 시스템 구성 프로파일을 설치 서비스에 추가하기 전에 검증합니다. 개발 중인 프로파일을 검증하려면 아래 `validate` 하위 명령을 참조하십시오.

*profile\_filename*은 `create-profile` 명령에 지정된 조건이나 환경 변수로부터 값을 구하는 교체 태그를 포함할 수 있습니다. [11 장, “클라이언트 시스템 구성”](#)을 참조하십시오.

*profile\_name*은 `installadm list` 명령으로 표시된 이름입니다. [104 페이지 “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”](#)을 참조하십시오. *profile\_name*이 제공되지 않은 경우 *profile\_name*은 *profile\_filename*의 기본 이름입니다. `-p` 옵션은 여러 개의 *profile\_filename*이 지정된 경우 유효하지 않습니다.

-c 옵션은 명령줄에 클라이언트 선택 조건을 지정합니다. -c 옵션은 XML 파일에 조건을 지정합니다. *criteriafile* 값은 전체 경로 및 파일 이름입니다. 명령줄의 조건 키워드와 파일 예제 목록은 9 장, “설치 사용자 정의”를 참조하십시오.

특정 클라이언트가 이 시스템 구성 프로파일을 사용하도록 하려면 먼저 해당 클라이언트가 이 `create-profile` 명령에 지정된 설치 서비스를 사용할지 확인합니다. `create-client` 명령을 사용하여 특정 설치 서비스와 명시적으로 연관되지 않은 클라이언트 시스템은 적절한 `default-arch` 설치 서비스를 사용합니다. 사용자 정의된 시스템 구성 프로파일을 `default-arch` 설치 서비스에 추가할 수도 있고, 사용자 정의된 구성 프로파일을 서로 다른 서비스에 추가한 다음 `create-client`를 통해 클라이언트가 해당 서비스를 사용하도록 만들 수도 있습니다.

다음 명령은 `profile_t200.xml` 프로파일을 `s11-sparc` 설치 서비스에 추가합니다. -c 옵션은 이 설치 서비스를 사용 중인 Sun Fire T200 서버로 식별된 클라이언트가 `profile_t200.xml` 시스템 구성 지침에 할당되도록 지정합니다.

```
# installadm create-profile -f ./mymanifests/profile_t200.xml \
-p t200 -n s11-sparc -c platform="SUNW,Sun-Fire-T200"
```

## 시스템 구성 프로파일 검증

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

```
installadm validate -n svcname -P profile_filename... | -p profile_name...
```

-p 옵션을 사용하여 설치 서비스에 추가되지 않은 프로파일을 검증합니다. *profile\_filename*은 파일에 대한 전체 경로 이름입니다.

-p 옵션을 사용하여 104 페이지 “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”에 표시된 대로 `create-profile` 하위 명령을 통해 *svcname* 설치 서비스에 이미 추가된 프로파일을 검증합니다. `create-profile` 하위 명령은 시스템 구성 프로파일을 설치 서비스에 추가하기 전에 검증합니다. `validate -p` 하위 명령은 프로파일이 추가된 이후 손상되지 않았는지 확인합니다.

*svcname*은 *profile\_filename* 및 *profile\_name* 프로파일에 모두 필요합니다. 아직 설치 서비스에 추가되지 않은 프로파일에는 서비스 이름이 필요합니다. `service_bundle(4)` DTD가 여러 버전의 OS마다 서로 다를 수 있기 때문입니다. 설치 서버를 실행 중인 버전이 아닌 다른 버전의 OS를 설치하도록 설치 서비스를 정의할 수도 있습니다. 설치 중인 클라이언트에 사용될 DTD에 대해 프로파일을 검증해야 합니다.

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

## 시스템 구성 프로파일 삭제

`installadm delete-profile` 명령을 사용하여 *profile\_name* 시스템 구성 프로파일을 *svcname* 설치 서비스에서 제거합니다. *profile\_name*은 `installadm list` 명령이 반환하는 프로파일 이름입니다. [104 페이지 “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”](#)을 참조하십시오.

```
installadm delete-profile -p profile_name... -n svcname
```

다음 명령은 **t200** 시스템 구성 프로파일을 **s11-sparc** 설치 서비스에서 제거합니다.

```
# installadm delete-profile -p t200 -n s11-sparc
```

## AI 매니페스트 또는 시스템 구성 프로파일 내보내기

`installadm export` 명령을 사용하여 *svcname* 설치 서비스에서 지정된 AI 매니페스트나 시스템 구성 프로파일의 내용을 *pathname* 파일이나 디렉토리로 복사합니다.

```
installadm export -n svcname
  -m manifest_or_script_name... -p profile_name...
  [-o pathname]
```

*pathname*이 지정되지 않은 경우 매니페스트와 프로파일 내용이 `stdout`로 이동합니다. 하나의 입력 파일만 지정된 경우 *pathname*은 파일 이름일 수 있습니다. 여러 개의 입력 파일이 지정된 경우 *pathname*은 디렉토리여야 합니다.

*manifest\_or\_script\_name*은 XML AI 매니페스트 또는 파생된 매니페스트 스크립트일 수 있습니다. 매니페스트 및 파생된 매니페스트 스크립트 생성에 대한 내용은 [10 장, “클라이언트 시스템 프로비전”](#)을 참조하십시오.

`installadm export` 명령을 사용하여 다음을 수행합니다.

- 매니페스트 및 프로파일의 사양을 확인합니다.
- 기존 매니페스트나 프로파일을 수정합니다. 기존 매니페스트나 프로파일을 새 매니페스트나 프로파일을 생성하기 위한 기초로 사용합니다.

## AI 매니페스트 또는 시스템 구성 프로파일에 대한 조건 수정

`installadm set-criteria` 명령을 사용하여 `create-manifest` 또는 `create-profile`로 *svcname* 설치 서비스에 이미 추가된 AI 매니페스트나 시스템 구성 프로파일과 연관된 클라이언트 조건을 업데이트합니다.

```
installadm set-criteria -m manifest_or_script_name -p profile_name... -n svcname
  -c criteria=value|list|range... | -C criteriafile |
  -a criteria=value|list|range...
```

0개 이상의 매니페스트를 0개 또는 원하는 수의 프로파일과 함께 동일한 `set-criteria` 명령줄에 지정할 수 있습니다. `manifest_or_script_name` 및 `profile_name` 이름은 `installadm list` 명령이 반환하는 이름입니다. [104 페이지 “모든 AI 매니페스트 및 시스템 구성 프로파일 나열”](#)을 참조하십시오.

`-c` 또는 `-C` 옵션을 사용하여 이러한 기존 매니페스트 및 프로파일의 조건을 새로 지정된 조건으로 바꿉니다. `-a` 옵션을 사용하여 기존 조건을 유지하고 지정된 조건을 추가합니다. 조건 지정에 대한 자세한 내용은 [9 장, “설치 사용자 정의”](#)를 참조하십시오.

다음 명령은 처음에 플랫폼 조건 사양과 함께 이 서비스에 추가된 메모리 조건 사양을 매니페스트에 추가합니다.

```
# installadm set-criteria -m t200 -n s11-sparc -a mem="4096-unbounded"
```

`create-manifest`로 지정되고 `set-criteria`로 추가된 조건의 결과, 이 설치 서비스를 사용 중인 클라이언트에서 사용되는 매니페스트는 Sun Fire T200 서버이고 최소한 4GB의 메모리를 가집니다.

다음 `criteria_t200.xml` 파일에서 `-a` 옵션 대신 `-C` 옵션을 사용하면 이와 동일한 결과를 얻을 수 있습니다.

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,Sun-Fire-T200</value>
  </ai_criteria>
  <ai_criteria name="mem">
    <range>
      4096
      unbounded
    </range>
  </ai_criteria>
</ai_criteria_manifest>
```

## 설치 서비스에 대한 정보 표시

`installadm list` 명령을 사용하여 설치 서비스에 대한 정보를 표시합니다.

```
installadm list [-n svcname] [-c] [-m] [-p]
```

### 설치 서버의 모든 설치 서비스 나열

다음 명령은 이 서버의 설치 서비스를 모두 표시합니다. 이 예에서 두 개의 사용으로 설정된 설치 서비스가 있습니다. 사용 안함으로 설정된 서비스는 상태 값이 `off`입니다. 주어진 구조에 생성된 첫번째 서비스가 해당 구조의 클라이언트의 기본 서비스임을 기억하십시오. [92 페이지 “설치 서비스 추가, 수정 또는 삭제”](#)를 참조하십시오.

```
$ installadm list
```

```
Service Name  Alias Of  Status Arch  Image Path
```

```

-----
default-i386  s11-x86  on    x86  /install/images/s11_x86
default-sparc s11-sparc on    Sparc /install/images/s11_sparc
s11-sparc      on    Sparc /install/images/s11_sparc
s11-x86        on    x86  /install/images/s11_x86

```

## 지정된 설치 서비스에 대한 정보 표시

다음 명령은 -n 옵션으로 지정된 설치 서비스에 대한 정보를 표시합니다.

```
$ installadm list -n s11-sparc
```

```

Service Name  Alias Of  Status Arch  Image Path
-----
s11-sparc      on       Sparc  /install/images/s11_sparc

```

## 설치 서비스와 연관된 클라이언트 나열

다음 명령은 이 설치 서버의 설치 서비스와 연관된 클라이언트를 모두 나열합니다. 클라이언트는 `installadm create-client` 명령을 사용하여 설치 서비스와 연관되었습니다. [95 페이지 “클라이언트를 설치 서비스에 추가”](#)를 참조하십시오.

```
$ installadm list -c
```

```

Service Name Client Address      Arch  Image Path
-----
s11-sparc    00:14:4F:A7:65:70 Sparc  /install/images/s11_sparc
s11-x86      08:00:27:8B:BD:71 x86    /install/images/s11_x86
              01:C2:52:E6:4B:E0 x86    /install/images/s11_x86

```

## 특정 설치 서비스와 연관된 클라이언트 나열

다음 명령은 지정된 설치 서비스에 추가된 클라이언트를 모두 나열합니다. 다음 예에서 하나의 클라이언트가 s11-sparc 설치 서비스와 연관됩니다.

```
$ installadm list -c -n s11-sparc
```

```

Service Name Client Address      Arch  Image Path
-----
s11-sparc    00:14:4f:a7:65:70 Sparc  /install/images/s11_sparc

```

## 사용자 정의된 설치에 대한 정보 표시

이 섹션의 명령은 특정 설치 서비스와 연관된 AI 매니페스트와 시스템 구성 프로파일을 보여줍니다. 또한 어떤 클라이언트 조건이 각 매니페스트 및 프로파일과 연관되는지 보여줍니다.



## 모든 AI 매니페스트 및 시스템 구성 프로파일 나열

다음 명령은 이 설치 서버의 모든 설치 서비스에 대한 모든 AI 매니페스트, 파생된 매니페스트 스크립트, 시스템 구성 프로파일을 나열합니다. Manifest/Profile 열은 매니페스트, 스크립트 또는 프로파일의 내부 이름을 표시합니다.

```
# installadm list -m -p
```

```
Service Name  Manifest
-----
s11-sparc     t200
s11-x86       ipv4
               mem1
```

```
Service Name  Profile
-----
s11-sparc     mac1
               t200
s11-x86       mac2
               mac3
               ipv4
               mem1
```

## 지정된 설치 서비스와 연관된 매니페스트 및 프로파일 나열

다음 예는 설치 서비스 `s11-sparc`와 연관된 모든 AI 매니페스트, 파생된 매니페스트 스크립트, 시스템 구성 프로파일을 보여줍니다. Manifest/Profile 열은 매니페스트, 스크립트 또는 프로파일의 내부 이름을 표시합니다. 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 `-d` 옵션과 함께 사용할 경우 이 서비스에 대한 새 기본 매니페스트가 됩니다. `mem1`이 기본 매니페스트이므로 해당 조건은 무시됩니다. 다른 매니페스트가 기본 매니페스트로 만들어진 경우 클라이언트가 이 매니페스트를 사용하도록 선택하기 위해 `mem1` 조건이 사용됩니다. 원래의 기본 매니페스트는 사용할 클라이언트를 결정하기 위해 연관된 조건이 없으므로 비활성입니다. 기본 매니페스트만 연관된 조건이 없습니다. 다른 매니페스트의 사용 조건과 일치하지 않는 클라이언트는 기본 매니페스트를 사용합니다. AI 매니페스트 선택에 대한 자세한 내용은 9장, “설치 사용자 정의”를 참조하십시오.

```
# installadm list -m -p -n s11-sparc
```

```
Manifest      Status      Criteria
-----
orig_default  Inactive    None
mem1          Default     (Ignored: mem = 2048 - 4095)
t200                                     platform = SUNW,Sun-Fire-T200
                                               mem = 4096-unbounded
```

```
Profile  Criteria
-----
mac1     mac  = 01:C2:52:E6:4B:E0
```



```

hostname = server1
ipv4 = 192.168.168.251
platform = SUNW,Sun-Fire-T200
mem = 4096-unbounded

```

## AISMF 서비스 관리

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

### 예 8-2 AISMF 서비스 사용

`installadm create-service` 명령을 실행할 때 AISMF 서비스가 사용으로 설정됩니다. 또한 기존 설치 서비스에 영향을 주는 다른 `installadm` 명령을 실행할 때도 AISMF 서비스가 사용으로 설정됩니다. AISMF 서비스를 수동으로 사용으로 설정하려면 다음 명령을 실행합니다.

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

현재 설치 서버에 사용으로 설정된 설치 서비스가 없거나 주의가 필요한 문제가 발생한 경우 AISMF 서비스가 유지 관리 모드로 진입합니다.

### 예 8-3 AISMF 서비스 사용 안함

AISMF 서비스를 사용 안함으로 설정하려면 다음 명령을 실행합니다.

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

사용으로 설정된 AI 설치 서비스가 하나라도 있으면 AISMF 서비스를 사용 안함으로 설정하지 마십시오. 어떤 설치 서비스가 사용으로 설정되었는지 확인하는 방법은 [102 페이지 “설치 서버의 모든 설치 서비스 나열”](#)을 참조하십시오.



## 설치 사용자 정의

설치를 사용자 정의하려면 설치 지침 및 시스템 구성 지침을 사용자 정의합니다. 그런 다음, 클라이언트 조건을 지정하여 사용자 정의된 설치 및 구성 지침을 해당 조건으로 식별된 클라이언트와 일치시킵니다.

AI 설치 서비스에는 1개 이상의 설치 지침 파일(AI 매니페스트)과 0개 이상의 구성 지침 파일(SMF 시스템 구성 프로파일)이 포함됩니다. 각 클라이언트는 단 하나의 AI 매니페스트만 사용합니다. 각 클라이언트는 원하는 수의 시스템 구성 프로파일을 사용할 수 있습니다. 클라이언트 시스템이 시스템 구성 프로파일을 사용하지 않으면, 클라이언트 설치 후에 처음 부트 시 해당 클라이언트에 대화식 도구가 열려서 클라이언트 구성을 완료합니다.

## 클라이언트를 설치 및 구성 지침과 일치

AI를 사용할 때 먼저 DHCP 서버와 설치 서버를 설정합니다. 설치 서버에는 적어도 하나의 AI 부트 이미지 및 해당 부트 이미지와 연관된 AI 설치 서비스가 있습니다. 클라이언트를 부트할 때 DHCP가 클라이언트를 설치 서버로 지정합니다.

클라이언트는 해당 클라이언트 구조의 기본 설치 서비스를 사용하거나, 할당된 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 할당하려면 해당 클라이언트를 설치 서비스에 추가합니다. 15 장, “클라이언트 시스템 설치”를 참조하십시오. 클라이언트의 MAC 주소 및 이 클라이언트가 사용할 설치 서비스의 이름을 지정합니다. 이 MAC 주소의 클라이언트를 부트할 때 DHCP가 클라이언트를 설치 서버로 지정하고 클라이언트가 지정된 설치 서비스를 사용합니다. 시스템의 MAC 주소를 찾으려면 **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화** 및 `dladm(1M)` 매뉴얼 페이지에 설명된 대로 `dladm` 명령을 사용하십시오.

하나의 넷 이미지에 대해 여러 유형의 설치를 정의하려면 추가 AI 매니페스트를 만들고 시스템 구성 프로파일을 만듭니다. 새 AI 매니페스트와 구성 프로파일을 해당 넷 이미지에 대한 AI 설치 서비스에 추가합니다. 어떤 클라이언트가 어떤 AI 매니페스트와 어떤 시스템 구성 프로파일을 사용할지 정의하는 조건을 지정합니다. [96 페이지 “클라이언트별 설치 지침을 설치 서비스와 연관”](#)을 참조하십시오.

사용자 정의 AI 매니페스트를 만들려면 [10 장, “클라이언트 시스템 프로비전”](#)을 참조하십시오. 시스템 구성 프로파일을 만들려면 [11 장, “클라이언트 시스템 구성”](#)을 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 [113 페이지 “기본 AI 매니페스트”](#)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](#)에 표시된 순서대로 클라이언트 특성이 평가됩니다. `installadm` 도구는 동일한 유형의 조건이 겹치지 않는지 확인합니다. [96 페이지 “AI 매니페스트 추가”](#)를 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 `mac`가 `ipv4`보다 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.

`manifest_x86.xml` AI 매니페스트가 클라이언트 구조를 지정하는 다음 조건 파일과 함께 서비스에 추가되었습니다.

```
<ai_criteria_manifest>
  <ai_criteria name="arch">
    <value>i86pc</value>
```

## 예 9-1 클라이언트를 AI 매니페스트와 일치 (계속)

```
</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>
```

x86 클라이언트를 설치 중인 경우 manifest\_x86.xml에 할당됩니다.

MAC 주소 00:14:4f:a7:65:70의 SPARC 클라이언트를 설치 중인 경우 manifest\_mac1.xml에 할당됩니다.

다른 MAC 주소의 SPARC 시스템을 설치 중인 경우 기본 AI 매니페스트에 할당됩니다.

## 시스템 구성 프로파일 선택

AI 매니페스트 선택에 사용된 것과 동일한 조건 키워드가 클라이언트의 시스템 구성 프로파일 선택에 사용됩니다. hostname 조건 키워드는 AI 매니페스트가 아닌 구성 프로파일에만 사용할 수 있습니다. 표 9-1을 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 모든 일치하는 구성 프로파일이 시스템 구성에 적용됩니다. 예를 들어, 한 조건 사양이 클라이언트의 호스트 이름과 일치하고 다른 조건 사양이 동일 클라이언트의 메모리 크기와 일치하는 경우 양쪽 구성 프로파일이 클라이언트 구성에 사용됩니다.

## 선택 조건

다음 표는 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.

-c 옵션을 사용하여 명령줄에 조건 키워드 및 값을 지정합니다.

```
-c criteria=value|list|range
-c mac="aa:bb:cc:dd:ee:ff"
-c zonename="zone1 zone2"
-c mem="2048-unbounded"
```

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 이름	설명	명령줄 및 XML 파일 예제
mac	16진수 MAC 주소를 콜론(:) 구분자로 표시, 또는 MAC 주소 범위	<b>CLI, 단일 MAC 주소:</b>  <code>-c mac="0:14:4F:20:53:97"</code>  <b>CLI, MAC 주소 범위:</b>  <code>-c mac=0:14:4F:20:53:94-0:14:4F:20:53:A0</code>  <b>XML, 단일 MAC 주소:</b>  <code>&lt;ai_criteria name="mac"&gt;   &lt;value&gt;0:14:4F:20:53:97&lt;/value&gt; &lt;/ai_criteria&gt;</code>  <b>XML, MAC 주소 범위:</b>  <code>&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;</code>
ipv4	IP 버전 4 네트워크 주소, 또는 IP 주소 범위	<b>CLI, 단일 IP 주소:</b>  <code>-c ipv4="10.6.68.127"</code>  <b>CLI, IP 주소 범위:</b>  <code>-c ipv4="10.6.68.1-10.6.68.200"</code>  <b>XML, 단일 IP 주소:</b>  <code>&lt;ai_criteria name="ipv4"&gt;   &lt;value&gt;10.6.68.127&lt;/value&gt; &lt;/ai_criteria&gt;</code>  <b>XML, IP 주소 범위:</b>  <code>&lt;ai_criteria name="ipv4"&gt;   &lt;range&gt;     10.6.68.1     10.6.68.200   &lt;/range&gt; &lt;/ai_criteria&gt;</code>

표 9-1 조건 키워드 및 조건 계층 (계속)

조건 이름	설명	명령줄 및 XML 파일 예제
platform	uname -i로 반환된 플랫폼 이름 포함 값: i86pc SUNW, SPARC-Enterprise SUNW, Sun-Fire-T200	<b>CLI:</b>  -c platform="SUNW,Sun-Fire-T200"  <b>XML:</b>  <ai_criteria name="platform"> <value>SUNW,Sun-Fire-T200</value> </ai_criteria>
arch	uname -m으로 반환된 구조 값: i86pc, sun4u 또는 sun4v	<b>CLI:</b>  -c arch="i86pc"  <b>XML:</b>  <ai_criteria name="arch"> <value>i86pc</value> </ai_criteria>
cpu	uname -p로 반환된 CPU 클래스 값: i386 또는 sparc	<b>CLI:</b>  -c cpu="sparc"  <b>XML:</b>  <ai_criteria name="cpu"> <value>sparc</value> </ai_criteria>
network	IP 버전 4 네트워크 번호, 또는 네트워크 번호 범위	<b>CLI, 단일 IP 주소:</b>  -c network="10.0.0.0"  <b>CLI, IP 주소 범위:</b>  -c network="11.0.0.0-12.0.0.0"  <b>XML, 단일 IP 주소:</b>  <ai_criteria name="network"> <value>10.0.0.0</value> </ai_criteria>  <b>XML, IP 주소 범위:</b>  <ai_criteria name="network"> <range> 11.0.0.0 12.0.0.0 </range> </ai_criteria>

표 9-1 조건 키워드 및 조건 계층 (계속)

조건 이름	설명	명령줄 및 XML 파일 예제
mem	prtcnf로 반환된 MB 단위의 메모리 크기, 또는 메모리 크기 범위  unbounded 키워드는 범위에 상한이 없음을 나타냅니다.	<b>CLI, 하나의 메모리 크기:</b> - c mem="4096" <b>CLI, 메모리 크기 범위:</b> - c mem="2048-unbounded" <b>XML, 하나의 메모리 크기:</b> <pre>&lt;ai_criteria name="mem"&gt;   &lt;value&gt;4096&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <b>XML, 메모리 크기 범위:</b> <pre>&lt;ai_criteria name="mem"&gt;   &lt;range&gt;     2048     unbounded   &lt;/range&gt; &lt;/ai_criteria&gt;</pre>
zonename	zoneadm list로 표시된 zones(5) 영역의 이름 또는 이름 목록. 12 장, “영역 설치 및 구성”을 참조하십시오.	<b>CLI, 단일 영역 이름:</b> - c zonename="myzone" <b>CLI, 영역 이름 목록:</b> - c zonename="zoneA zoneB zoneC" <b>XML, 단일 영역 이름:</b> <pre>&lt;ai_criteria name="zonename"&gt;   &lt;value&gt;myzone&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <b>XML, 영역 이름 목록:</b> <pre>&lt;ai_criteria name="zonename"&gt;   &lt;value&gt;zoneA zoneB zoneC&lt;/value&gt; &lt;/ai_criteria&gt;</pre>



표 9-1 조건 키워드 및 조건 계층 (계속)

조건 이름	설명	명령줄 및 XML 파일 예제
hostname	클라이언트 호스트 이름 또는 클라이언트 호스트 이름 목록.  이 조건 키워드는 AI 매니페스트가 아닌 시스템 구성 프로파일에만 사용할 수 있습니다.	<b>CLI, 단일 호스트 이름:</b>  <code>-c hostname="host3"</code>  <b>CLI, 호스트 이름 목록:</b>  <code>-c hostname="host1 host2 host6"</code>  <b>XML, 단일 호스트 이름:</b>  <pre>&lt;ai_criteria name="hostname"&gt;   &lt;value&gt;host3&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <b>XML, 호스트 이름 목록:</b>  <pre>&lt;ai_criteria name="hostname"&gt;   &lt;value&gt;host1 host2 host6&lt;/value&gt; &lt;/ai_criteria&gt;</pre>

## 기본 AI 매니페스트

새 설치 서비스를 만들 때 `install_service_image_path` `/auto_install/manifest/default.xml`이 해당 설치 서비스의 초기 기본 AI 매니페스트입니다.

이 기본 AI 매니페스트가 아래에 표시됩니다. 이 기본 매니페스트는 설치 이미지마다 약간 다를 수 있습니다.

기본 매니페스트의 **target** 섹션은 만들려는 ZFS 파일 시스템 또는 데이터 집합을 정의합니다. 기본 매니페스트는 설치할 대상 디스크를 정의하지 않습니다. 매니페스트에 지정된 대상 디스크가 없을 때 설치할 기본 대상 위치가 어떻게 결정되는지 알아보려면 [ai\\_manifest\(4\)](#) 매뉴얼 페이지를 참조하십시오.

**destination** 섹션은 설치할 로컬을 지정하는 데 사용할 수 있습니다. 페이지 사양을 매니페스트에 사용하여 설치할 로컬을 제한하면 모든 로컬이 필요하지 않은 경우 시간과 공간을 절약할 수 있습니다. 지정된 페이지가 없는 경우 모든 로컬의 페이지가 `true`로 기본 설정됩니다. 이미지 페이지 및 속성 설정에 대한 자세한 내용은 [ai\\_manifest\(4\)](#) 매뉴얼 페이지를 참조하십시오.

소프트웨어 설치 지침은 기본 IPS 패키지 저장소를 지정하고 다음 두 패키지를 설치합니다.

- **entire** 패키지는 필수입니다. 이 통합 패키지는 시스템 패키지가 호환 가능한 버전으로 설치되도록 제약합니다. 이 통합의 존재 여부에 따라 적절한 시스템 업데이트와 올바른 패키지 선택이 결정됩니다. AI 매니페스트에서 이 패키지 설치를 제거하지 마십시오. 그리고 설치 후에 이 패키지 설치를 해제하지 마십시오.
- **solaris-large-server** 패키지는 대부분의 환경에 필요한 도구 및 장치 드라이버의 그룹 패키지입니다. 이 패키지는 수많은 네트워크 및 저장소 드라이버, Python 라이브러리, Perl 등을 설치합니다. **solaris-large-server** 그룹 패키지에 포함될 전체 패키지 목록은 [Oracle Solaris 11 소프트웨어 패키지 추가 및 업데이트의 “그룹 패키지의 설치 가능한 모든 패키지 나열”](#)에 설명된 대로 `pkg contents` 명령을 사용하십시오.

설치할 다른 패키지의 이름을 찾는 방법은 [Oracle Solaris 11 소프트웨어 패키지 추가 및 업데이트](#)를 참조하십시오.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

Copyright (c) 2008, 2011, Oracle and/or its affiliates. All rights reserved.

```
-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.@DTD_VERSION_AI@">
<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">
      <destination>
        <image>
          <!-- Specify locales to install -->
          <facet set="false">facet.locale.*</facet>
          <facet set="true">facet.locale.de</facet>
          <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>
<source>
    <publisher name="solaris">
        <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
</source>
<!--
    By default the latest build available, in the specified IPS
    repository, is installed.  If another build is required, the
    build number has to be appended to the 'entire' package in the
    following form:

        <name>pkg:/entire@0.5.11-0.build#</name>
-->
<software_data action="install">
    <name>pkg:/entire@latest</name>
    <name>pkg:/group/system/solaris-large-server</name>
</software_data>
</software>
</ai_instance>
</auto_install>

```



## 클라이언트 시스템 프로비전

---

AI 설치 서비스를 만들 때 클라이언트의 프로비전 방법을 지정하는 기본 AI 매니페스트를 가져옵니다. AI 매니페스트는 어디에 운영 체제를 설치하고 어떤 소프트웨어 패키지를 설치할지 지정하는 XML 파일입니다. 스트라이핑, 미러링, 분할과 같은 디스크 구성을 지정할 수도 있습니다. AI 매니페스트의 XML 요소에 대한 내용은 [ai\\_manifest\(4\)](#) 매뉴얼 페이지와 `install_service_image_path/auto_install/manifest/ai_manifest.xml` 샘플 AI 매니페스트를 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.
- 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.

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 매니페스트나 스크립트는 매니페스트나 스크립트를 설치 서비스에 추가할 때 지정된 조건에 따라 다릅니다. 클라이언트가 사용자 정의 매니페스트나 스크립트의 사용 조건과 일치하지 않으면 기본 매니페스트가 사용됩니다. 서비스의 매니페스트나 스크립트를 해당 서비스의 기본값이 되도록 지정할 수 있습니다.

## XML AI 매니페스트 파일 사용자 정의

사용자 정의 XML AI 매니페스트 파일을 만들고 적용하려면 다음 단계를 따릅니다.

1. 기존 AI 매니페스트를 복사합니다. AI 설치 서비스를 만들 때 기본 AI 매니페스트를 가져옵니다. 설치 서비스 만들기에 대한 내용은 [8 장](#), “[설치 서버 설정](#)”을 참조하십시오.

`list` 하위 명령을 사용하여 특정 설치 서비스와 이미 연관된 AI 매니페스트가 무엇인지 확인합니다.

```
$ installadm list -m -n s11-x86
```

Manifest	Status	Criteria
-----	-----	-----
orig_default	Default	None

`installadm export` 명령을 사용하여 이 기본 매니페스트나 기타 이 서비스에 추가된 AI 매니페스트를 복사합니다.

```
# installadm export -n s11-x86 -m orig_default -o mem1.xml
```

`orig_default`의 복사본이 지금 `mem1.xml` 파일에 있습니다.

2. `mem1.xml`을 수정합니다. [ai\\_manifest\(4\)](#) 매뉴얼 페이지의 정보에 따라 태그 및 값을 추가하면 됩니다.
3. 새 AI 매니페스트를 적절한 AI 설치 서비스에 추가합니다. 어떤 클라이언트가 이러한 설치 지침을 사용할지 정의하는 조건을 지정하면 됩니다.

```
# installadm create-manifest -n s11-x86 -f ./mem1.xml -m mem1 \
-c mem="2048-unbounded"
```

여러 `-c` 옵션이나 하나의 `-c` 파일을 지정할 수 있습니다. 클라이언트 조건 지정에 대한 내용은 [9 장](#), “[설치 사용자 정의](#)” 및 `set-criteria` 하위 명령을 참조하십시오.

```
$ installadm list -n s11-x86 -m
```

Manifest	Status	Criteria
-----	-----	-----
orig_default	Default	None
mem1		mem = 2048 MB - unbounded

특정 매니페스트 파일이나 파생된 매니페스트 스크립트가 서비스의 기본 매니페스트나 스크립트가 되도록 지정할 수 있습니다. 이미 서비스에 추가된 매니페스트와 스크립트 중에서 기본값을 변경하려면 `set-service` 하위 명령과 함께 `-o` 옵션을 사용합니다.

```
# installadm set-service -o default-manifest=mem1 s11-x86
# installadm list -n s11-x86 -m
```

Manifest	Status	Criteria
-----	-----	-----
orig_default	Inactive	None
mem1	Default	(Ignored: mem = 2048 MB - unbounded)

이 예에서 원래의 기본값은 사용할 클라이언트를 지정하는 조건이 없으므로 지금 비활성입니다. 기본 매니페스트나 스크립트만 클라이언트 선택 조건이 없으므로 계속 활성일 수 있습니다.

이 서비스에 대한 새 기본 매니페스트나 스크립트를 추가하려면 `create-manifest`와 함께 `-d` 옵션을 사용하고 클라이언트 조건은 지정하지 마십시오.

```
# installadm create-manifest -n s11-x86 -d \
-f ./region1.xml -m region1
# installadm list -n s11-x86 -m
```

Manifest	Status	Criteria
orig_default	Inactive	None
mem1		mem = 2048 MB - unbounded
region1	Default	None

이미 설치 서비스에 추가된 매니페스트나 스크립트의 내용을 변경하려면 `installadm update-manifest` 명령을 사용합니다. 조건, 기본 상태 및 `manifest_or_script_name`은 업데이트 결과로 바뀌지 않습니다.

```
# installadm update-manifest -n s11-x86
-f ./newregion1.xml -m region1
```

`create-manifest` 및 `update-manifest` 하위 명령은 XML 매니페스트 파일을 설치 서비스에 추가하기 전에 검증합니다. AI가 클라이언트 설치 타임에 AI 매니페스트를 구문상 검증합니다.

---

주 - 잘못된 매니페스트가 클라이언트에 제공되면 자동 설치가 중단됩니다. 검증 실패의 원인을 조사하려면 클라이언트의 `/system/volatile/install_log`를 확인하십시오.

---

`list`, `export`, `create-manifest`, `set-criteria`, `update-manifest`, `set-service` 하위 명령에 대한 자세한 내용은 [92 페이지 “설치 서버 유지 관리”](#)를 참조하십시오.

## 클라이언트 설치 타임에 AI 매니페스트 만들기

클라이언트 설치 전에 사용자 정의 AI 매니페스트를 만드는 대안은, 클라이언트 설치 타임에 각 클라이언트에 대한 AI 매니페스트를 동적으로 생성하는 스크립트를 작성하는 것입니다. 스크립트로 환경 변수 및 다른 클라이언트 구성 정보를 질의하여 각 클라이언트에 대한 사용자 정의 AI 매니페스트를 만들 수 있습니다. 매니페스트는 설치 타임에 발견된 각 클라이언트의 속성을 기반으로 하므로 **파생된 매니페스트**라고 합니다.

파생된 매니페스트는 많은 수의 시스템을 거의 동일하게 설치할 수 있으므로 이러한 시스템의 AI 매니페스트가 비교적 작은 차이를 보이는 경우 특히 유용합니다. 이 시스템 그룹에 공통되는 설치 매개변수를 지정하는 AI 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 AI 매니페스트를 수정하여 각 클라이언트에 대한 사용자 정의 디스크 레이아웃을 지정할 수 있습니다.

파생된 매니페스트 스크립트를 만들고 적용하려면 다음 단계를 따릅니다.

1. 수정용 기본 매니페스트로 사용할 기존 AI 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. 설치 중인 클라이언트의 속성을 기반으로 설치 타임에 기본 매니페스트를 동적으로 수정하는 스크립트를 작성합니다.
3. 파생된 매니페스트 스크립트를 적절한 AI 설치 서비스에 추가합니다. 설치 타임에 어떤 클라이언트가 이 스크립트를 사용하여 설치 지침을 만들지 정의하는 조건을 지정하면 됩니다.

AI가 클라이언트 설치 타임에 스크립트를 실행하여 AI 매니페스트의 인스턴스를 생성합니다. AI가 결과 매니페스트를 구문상 검증합니다.

---

주 - 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/sadm/system/logs/derived/manifest.xml`로 복사되고, 매니페스트를 파생시키는 데 사용된 스크립트가 `/var/sadm/system/logs/derived/manifest_script`로 복사됩니다.

## 파생된 매니페스트 스크립트 만들기

일반적으로, 파생된 매니페스트 스크립트는 클라이언트에서 정보를 검색하고 해당 정보를 토대로 기본 AI 매니페스트를 수정하여 이 클라이언트에 대한 사용자 정의 AI 매니페스트를 만듭니다. 파생된 매니페스트 스크립트는 여러 개의 부분 AI 매니페스트를 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. AI가 `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, 프로파일, 권한에 대한 내용은 [Oracle Solaris 관리: 보안 서비스의 제III부, “역할, 권한 프로파일 및 권한”](#)을 참조하십시오.

시스템 속성을 읽는 명령을 사용하는 것 외에도, 다음 표에 표시된 환경 변수를 통해 클라이언트 속성을 사용할 수 있습니다.

표 10-1 클라이언트 속성 환경 변수

환경 변수 이름	설명
<code>SI_ARCH</code>	설치될 클라이언트의 구조입니다. <code>uname -p</code> 의 출력과 같습니다.
<code>SI_CPU</code>	설치될 클라이언트의 ISA 또는 프로세서 유형입니다. <code>uname -p</code> 의 출력과 같습니다.
<code>SI_NUMDISKS</code>	클라이언트의 디스크 수입니다.
<code>SI_DISKNAME_#</code>	클라이언트에서 발견된 디스크의 <code>ctds</code> 이름을 나타내는 플랫폼 형식의 변수 집합입니다. <code>SI_DISKNAME_#</code> 변수의 <code>SI_NUMDISKS</code> 번호가 존재합니다. 여기서 <code>#</code> 은 1부터 시작해서 최대 <code>SI_NUMDISKS</code> 까지의 정수로 바뀝니다. 이 변수 집합은 <code>SI_DISKSIZE_#</code> 에 설명된 변수 집합과 상관이 있습니다.
<code>SI_DISKSIZE_#</code>	클라이언트에서 발견된 디스크의 디스크 크기를 나타내는 플랫폼 형식의 변수 집합입니다. <code>SI_DISKSIZE_#</code> 변수의 <code>SI_NUMDISKS</code> 번호가 존재합니다. 여기서 <code>#</code> 은 1부터 시작해서 최대 <code>SI_NUMDISKS</code> 까지의 정수로 바뀝니다. 이 변수 집합은 <code>SI_DISKNAME_#</code> 에 설명된 변수 집합과 상관이 있습니다. 크기는 MB 단위의 정수입니다.
<code>SI_HOSTADDRESS</code>	설치 환경에 설정된 클라이언트의 IP 주소입니다.
<code>SI_HOSTNAME</code>	설치 환경에 설정된 클라이언트의 호스트 이름입니다.
<code>SI_KARCH</code>	클라이언트의 커널 구조입니다. <code>uname -m</code> 의 출력과 같습니다.
<code>SI_INSTALL_SERVICE</code>	매니페스트 스크립트를 얻기 위해 사용된 설치 서비스의 이름입니다. 이 환경 변수는 매체 부트가 아닌 네트워크 부트에만 값을 가집니다.
<code>SI_MANIFEST_SCRIPT</code>	매니페스트 스크립트의 URL입니다.
<code>SI_MEMSIZE</code>	클라이언트의 물리적 메모리 양입니다. 크기는 MB 단위의 정수입니다.

표 10-1 클라이언트 속성 환경 변수 (계속)

환경 변수 이름	설명
SI_MODEL	클라이언트의 모델 이름입니다. <code>uname -i</code> 의 출력과 같습니다.
SI_NATISA	클라이언트의 고유의 명령 집합 구조입니다. <code>isainfo -n</code> 의 출력과 같습니다.
SI_NETWORK	클라이언트의 네트워크 번호입니다. 네트워크 번호는 (IP_ADDR & netmask)입니다.
SI_PLATFORM	클라이언트의 플랫폼입니다. <code>uname -i</code> 의 출력과 같습니다.

## AI 매니페스트 사용자 정의

AI 매니페스트의 XML 요소를 추가하거나 수정하려면 `/usr/bin/aimanifest` 명령을 사용합니다.

`aimanifest`가 수정할 수 있는 최소 파일은 다음 조각을 모두 포함해야 합니다.

- 개발 중인 XML 매니페스트에 유효한 DTD에 대한 `!DOCTYPE` 참조
- 이 DTD에 대한 루트 요소

다음 예는 AI 매니페스트에 대한 최소 기본 매니페스트 파일을 보여주며, 여기에는 이 파생된 매니페스트 스크립트를 추가할 설치 서비스에 대한 AI DTD 파일 지정이 포함됩니다.

```
<!DOCTYPE auto_install SYSTEM "file:///image_path/auto_install/ai.dtd.#">
<auto_install/>
```

`#`은 1과 같은 정수입니다. `image_path`는 다음 명령으로 반환된 경로입니다. 여기서 `service_name`은 이 파생된 매니페스트 스크립트를 추가할 설치 서비스의 이름입니다.

```
$ installadm list -n service_name
```

`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 `image_path/auto_install/manifest/`에서 매니페스트를 로드할 수 있습니다.

이 장의 예에서는 `/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 안 됩니다.

`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. `aimanifest` 명령 사용의 예는 `aimanifest(1M)` 매뉴얼 페이지와 아래의 예제 스크립트를 참조하십시오.

`aimanifest` 명령에 지정된 값이 슬래시(/)나 작은/큰 따옴표와 같은 특수 문자를 포함하는 경우 해당 값을 작은/큰 따옴표로 둘러싸야 합니다. 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
```

---

**참고** - 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 baseAImanifest.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
```

## 예제 파생된 매니페스트 스크립트

이 섹션은 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. `aimanifest`를 `load`, `add`, `set` 하위 명령과 함께 호출할 때마다 `AIM_MANIFEST` 파일이 열기, 수정, 저장됩니다. `AIM_MANIFEST`가 설정되지 않으면 `aimanifest` 명령을 실패합니다.

2. AIM\_LOGFILE 환경 변수를 스크립트가 상세 정보 및 오류 메시지를 작성할 수 있는 위치로 설정합니다.

aimanifest 명령은 각 aimanifest 호출의 하위 명령 이름, 인수 값, 반환 상태를 화면과 \$AIM\_MANIFEST\_LOG 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 10-1에 표시된 환경 변수의 값을 파생된 매니페스트 스크립트에 사용할 수 있습니다.

#### 예 10-1 디스크 크기에 준하여 디스크 분할 지정

이 예는 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.

```
#!/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

# Assume there is only one disk on the system for this example.
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/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
fi
exit $SCRIPT_SUCCESS
```

일부 클라이언트의 경우 다음 요소가 \$AIM\_MANIFEST에 추가됩니다.

## 예 10-1 디스크 크기에 준하여 디스크 분할 지정 (계속)

```
<target>
  <disk>
    <partition action="use_existing_solaris2"/>
  </disk>
</target>
```

다른 클라이언트의 경우 `SI_DISKSIZE_1` 값에 따라 다음과 비슷한 요소가 `$AIM_MANIFEST`에 추가됩니다.

```
<target>
  <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 AI 매니페스트를 사용자 정의합니다. 스크립트를 실행하기 전에 `SI_NUMDISKS` 및 `SI_DISKNAME_1`을 설정합니다. `SI_NUMDISKS`에 설정된 값에 따라 `SI_DISKNAME_2`, `SI_DISKNAME_3` 및 기타 등등을 설정합니다. 이 환경 변수는 AI 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
```

## 예 10-2 추가 디스크의 존재 여부에 준하여 루트 풀 레이아웃 지정 (계속)

```

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 디스크로 구성된 시스템의 경우 이 예제의 출력은 다음 XML과 같습니다.

```

<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">
      <vdev name="mirror_vdev" redundancy="mirror"/>
    </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(적어도 두 개의 200GB)를 사용합니다. 스크립트를 실행하기 전에 테스트 환경에 SI\_NUMDISKS, SI\_DISKNAME\_1, SI\_DISKSIZE\_1을 설정합니다. 또한 SI\_NUMDISKS에 설정된 값에 따라 SI\_DISKNAME\_2, SI\_DISKSIZE\_2 및 기타 등등을 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 aimanifest의 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
}

```

예 10-3 지정된 크기의 디스크가 적어도 두 개 있는 경우 미러링된 구성 지정 (계속)

```

}

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 Kornshell 버전의 Python 버전입니다.

```

import os
from subprocess import call, check_call, CalledProcessError

SCRIPT_SUCCESS = 0
SCRIPT_FAILURE = 1

def main():

```

## 예 10-3 지정된 크기의 디스크가 적어도 두 개 있는 경우 미러링된 구성 지정 (계속)

```

# Find the disks first.
found_1 = ""
found_2 = ""

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:
    subprocess.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:
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                                "target/logical/zpool/vdev@redundancy", "mirror"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                                "target/logical/zpool/vdev[@redundancy='mirror']@name", "mirror_vdev"])

        subprocess.check_call(["/usr/bin/aimanifest", "add",
                                "target/disk/disk_name@name", "found_1"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                                "target/disk/disk_name[@name='" + found_1 + "',]" + "@name_type", "ctd"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                                "target/disk[disk_name@name='" + found_1 + "',]" + "@in_vdev", "mirror_vdev"])

        subprocess.check_call(["/usr/bin/aimanifest", "add",
                                "target/disk/disk_name@name", "found_2"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                                "target/disk/disk_name[@name='" + found_2 + "',]" + "@name_type", "ctd"])
        subprocess.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 AI



## 예 10-4 IP 주소에 준하여 설치할 패키지 지정 (계속)

매니페스트를 사용자 정의합니다. 스크립트를 실행하기 전에 테스트 환경에 `SI_HOSTADDRESS`를 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

# 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 대상디스크가 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_1`을 설정합니다. 또한 `SI_NUMDISKS`에 설정된 값에 따라 `SI_DISKNAME_2`, `SI_DISKSIZE_2` 및 기타 등등을 설정합니다. 이 환경 변수는 AI 설치 중에 파생된 매니페스트 스크립트에 설정되어 사용할 수 있습니다.

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
```

예 10-5 대상 디스크가 특정 크기가 되도록 지정 (계속)

```
{
    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 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
```

## 예 10-6 부정확한 매니페스트 사양의 스크립트 (계속)

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 지정됩니다. 요소 값은 아닙니다.
3. `pkg:/driver/pcmcia` 값이 따옴표로 둘러싸여야 합니다. 슬래시나 다른 특수 문자가 포함된 값은 따옴표를 붙여야 합니다.

이 예의 `set` 줄을 다음 줄로 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 줄을 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 실행합니다.

1. 수정할 스크립트에 대한 기본 AI 매니페스트를 설정합니다.
  - a. 스크립트의 첫번째 `aimanifest` 명령이 `aimanifest load` 명령인지 확인합니다. 대상 설치 서비스에 대한 AI 매니페스트 검증에 사용할 적절한 DTD를 지정하는 `<!DOCTYPE>` 정의가 로드 중인 파일에 있는지 확인합니다. 다음 예는 AI 매니페스트에 대한 최소 기본 매니페스트 파일을 보여주며, 여기에는 이 파생된 매니페스트 스크립트를 추가할 설치 서비스에 대한 AI DTD 파일 지정이 포함됩니다.

```
<!DOCTYPE auto_install SYSTEM "file:///image_path/auto_install/ai.dtd.#">
<auto_install/>
```

#은 1과 같은 정수입니다. `image_path`는 다음 명령으로 반환된 경로입니다. 여기서 `service_name`은 이 파생된 매니페스트 스크립트를 추가할 설치 서비스의 이름입니다.

---

```
$ installadm list -n service_name
```

- b. AIM\_MANIFEST를 스크립트가 AI 매니페스트를 개발할 위치로 설정합니다. 이 위치는 비권한 사용자 **aiuser**가 쓰기 가능해야 합니다.

---

주 - 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 **aiuser**가 쓰기 가능해야 합니다.

---

주 - 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. 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.

5. 테스트 환경의 환경 변수를 이 파생된 매니페스트 스크립트로 설치될 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 설치를 진행 중일 때 표 10-1에 표시된 환경 변수의 값을 파생된 매니페스트 스크립트에 사용할 수 있습니다.

의도한 클라이언트 시스템이 설치 서버나 기타 파생된 매니페스트 스크립트를 테스트할 수 있는 시스템과 아주 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. 위에 설명된 대로 테스트 환경에 환경 변수를 설정하면 이러한 차이점 중 일부가 해결됩니다.

다음 방식을 사용하여 의도한 클라이언트 시스템 중 하나에서 파생된 매니페스트 스크립트를 테스트합니다.

1. 클라이언트 시스템에서 "Text Installer and command line" 모드로 AI 이미지를 부트합니다.
2. 설치 프로그램 초기 메뉴에서 "셸"을 선택합니다.
3. `wget` 또는 `sftp`를 사용하여 AI 설치 서버에서 스크립트를 복사합니다.
4. 다음 방식 중 하나를 사용하여 스크립트를 디버그합니다.
  - 스크립트를 수동으로 실행합니다.
  - 다음 명령을 사용하여 AI를 테스트 모드로 실행합니다.

```
$ auto-install -m script -i
```

AI 로그 파일 `/system/volatile/install.log`를 검사합니다. 스크립트 검증 결과를 나타내는 다음 줄이 로그 파일에 있어야 합니다.

```
Derived Manifest Module: XML validation completed successfully
```

5. 스크립트를 설치 서버로 다시 복사합니다.

## 파생된 매니페스트 스크립트를 설치 서비스에 추가

XML 매니페스트를 설치 서비스에 추가하는 것과 동일한 방법으로 AI 설치 서비스에 스크립트를 추가합니다. 동일한 옵션을 사용하여 이 스크립트로 설치 매니페스트를 만들 클라이언트를 선택하기 위한 조건을 지정합니다. XML 매니페스트를 업데이트하는 것처럼 스크립트를 업데이트할 수 있습니다. 스크립트가 서비스의 기본 매니페스트가 되도록 설정할 수 있습니다. 서비스와 연관된 매니페스트를 나열할 때 스크립트가 표시됩니다. XML 매니페스트를 내보내는 것처럼 스크립트의 내용을 내보낼 수 있습니다.

XML 매니페스트를 설치 서비스에 추가할 때 매니페스트가 검증됩니다. 스크립트를 설치 서비스에 추가할 때는 스크립트가 검증되지 않습니다.

스크립트를 설치 서비스에 추가하기 전에 파생된 AI 매니페스트를 검증합니다.

1. 의도한 클라이언트 시스템과 비슷한 환경에서 스크립트를 실행합니다.
2. 결과 매니페스트에 `validate` 하위 명령을 사용합니다.

```
$ /usr/bin/aimanifest validate
```

검증을 실패한 경우에만 메시지가 표시됩니다.

적절한 AI 설치 서비스에 스크립트를 추가합니다. 이러한 설치 지침을 사용할 클라이언트를 정의하는 조건을 지정하면 됩니다.

```
# installadm create-manifest -n s11-x86 -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 9 장, “설치 사용자 정의”를 참조하십시오.

`list`, `export`, `create-manifest`, `set-criteria`, `update-manifest`, `set-service` 하위 명령에 대한 내용은 92 페이지 “설치 서버 유지 관리”를 참조하십시오.

## 클라이언트 시스템 구성

---

이 장에서는 설치 후에 클라이언트 시스템 구성에 필요한 정보를 지정하는 방법을 설명합니다. smf(5) 등록 정보를 통해 구성 가능한 모든 구성을 지정할 수 있습니다.

### 구성 프로파일 제공

시스템 구성 프로파일은 SMF(서비스 관리 기능) 프로파일 형식의 구성 매개변수 집합으로 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 할당할 수 있습니다. 더욱 광범위하게 적용 가능한 등록 정보 값을 설정하는 다른 프로파일에 동일한 클라이언트와 많은 다른 클라이언트를 할당할 수 있습니다.

특정 클라이언트에 제공된 시스템 구성 프로파일이 없으면 해당 클라이언트에 대화식 구성 도구가 열립니다. 구성 도구를 대화식으로 사용하는 방법은 [63 페이지 “시스템 구성”](#)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 프로파일 파일을 만드는 방법은 67 페이지 “SCI 도구를 사용하여 구성 프로파일 만들기”를 참조하십시오.

- 등록 정보 사양을 사용하여 시스템 구성 프로파일을 수동으로 만듭니다. 137 페이지 “시스템 구성 프로파일에 구성 지정” 및 148 페이지 “예제 시스템 구성 프로파일”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.

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 설치 서비스에 추가할 때 유효한 값으로 대체될 변수를 포함할 수 있습니다. 146 페이지 “시스템 구성 프로파일 템플릿 사용”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. `validate` 하위 명령에 대한 자세한 내용은 100 페이지 “시스템 구성 프로파일 검증”을 참조하십시오.

프로파일 템플릿을 검증하려면 146 페이지 “시스템 구성 프로파일 템플릿 사용”에 설명된 대로 환경 변수에 변수 값을 제공합니다.

```
# export AI_HOSTNAME=server1;export AI_IPV4=10.0.0.1;export AI_NETWORK=10.0.0.0
# installadm validate -n svc1 -P /export/hostIPnet.xml
```

## 시스템 구성 프로파일을 설치 서비스에 추가

`installadm create-profile` 명령을 사용하여 시스템 구성 프로파일을 설치 서비스에 추가합니다. `create-profile` 하위 명령은 프로파일을 설치 서비스에 추가하기 전에 검증합니다.

적절한 클라이언트가 해당 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 집합을 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.

`create-profile`, `list`, `set-criteria`, `export` 하위 명령에 대한 자세한 내용은 [92 페이지](#) “설치 서버 유지 관리” 및 [installadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 시스템 구성 프로파일에 구성 지정

`smf(5)` 등록 정보를 통해 구성 가능한 모든 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 DNS(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, 제약조건 및 기타 가능한 설정 선택에 대한 설명을 포함하여 모든 사용 가능한 정보를 제공합니다.

`-t` 옵션은 선택한 템플릿 데이터만 보여주고([smf\\_template\(5\)](#) 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

sysconfig create-profile 명령을 users 그룹과 함께 사용하여 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.

**참고** - Oracle Solaris OS에 대한 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` 값으로 추가하는 것입니다.

## 루트 계정 구성

`root_account` 등록 정보 그룹은 다음 등록 정보를 포함합니다.

표 11-1 `root_account` 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
password	astring	yes	암호화된 루트 암호. 루트 암호를 제공하지 않으면 루트 암호가 비어 있습니다.
type	astring	no	계정 유형: <code>normal</code> 또는 <code>role</code> . 기본값은 <code>normal</code> 입니다.
expire	string	no	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

`user_account` 등록 정보 그룹은 다음 등록 정보를 포함합니다.

표 11-2 user\_account 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
login	astring	yes	사용자의 로그인.
password	astring	yes	암호화된 사용자 암호.
description	astring	no	대개 사용자의 전체 이름.
shell	astring	no	로그인 시 사용자의 셸로 사용된 프로그램의 전체 경로 이름.
uid	count	no	새 사용자의 UID. 기본 UID는 101입니다.
gid	count	no	사용자의 기본 그룹 멤버십. 기본 GID는 10입니다.
type	astring	no	계정 유형: normal 또는 role. 기본 값은 normal입니다.
profiles	astring	no	prof_attr(4)에 정의된 하나 이상의 콤마로 구분된 실행 프로파일.
roles	astring	no	user_attr(4)에 정의된 하나 이상의 콤마로 구분된 역할.
sudoers	astring	no	sudoers(4) 파일에 login과 함께 추가된 항목.
expire	astring	no	로그인 만료일. 0으로 설정하면 다음 로그인 시 암호를 강제로 변경하도록 합니다.
home_zfs_dataset	astring	no	사용자의 홈 디렉토리 ZFS 데이터 집합. 기본 값은 root_pool/export/home/login입니다.
home_mountpoint	astring	no	사용자의 홈 디렉토리 마운트 포인트. 기본 값은 /export/home/login입니다.

## 시스템 신원

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 등록 정보를 포함합니다.

표 11-3 identity 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
nodename	astring	no	시스템 호스트 이름. 기본값은 unknown입니다.

**예 11-2 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>
```

## 시간대 및 로캘

sysconfig create-profile 명령을 location 그룹과 함께 사용하여 시간대 및 로캘을 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 등록 정보를 포함합니다.

표 11-4 timezone 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
localtime	astring	no	시스템 시간대. 기본값은 UTC입니다.

**예 11-3 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 로캘을 설정합니다.

environment 등록 정보 그룹은 다음 환경 변수를 정의할 수 있습니다. 환경 변수에 대한 내용은 [environ\(5\)](#) 매뉴얼 페이지를 참조하십시오.

표 11-5 environment 등록 정보 그룹의 등록 정보

환경 변수	유형	필수	기본값
LC_CTYPE	aststring	no	C
LC_NUMERIC	aststring	no	C
LC_TIME	aststring	no	C
LC_COLLATE	aststring	no	C
LC_MONETARY	aststring	no	C
LC_MESSAGES	aststring	no	C
LC_ALL	aststring	no	C
LANG	aststring	no	C

## 예 11-4 로캘 구성

이 예는 로캘을 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>
```

## 터미널 유형 및 키보드 레이아웃

## 예 11-5 터미널 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 11-6 키보드 레이아웃 구성**

`sysconfig create-profile` 명령을 `kdb layout` 그룹과 함께 사용하여 키보드 레이아웃을 구성하는 유효한 프로파일을 생성합니다.

```
# sysconfig create-profile -g kdb_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 `Czech`로 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>
```

## 정적 네트워크 구성

`sysconfig create-profile` 명령을 `network` 그룹과 함께 사용하여 네트워크를 구성하는 유효한 프로파일을 생성합니다.

```
# 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.

`svc:/network/install` 서비스는 하나의 IPv4 인터페이스와 하나의 IPv6 인터페이스, 그리고 선택적으로 이러한 인터페이스에서 연결 가능한 기본 경로의 구성을 지원합니다. 서비스가 IPv4 인터페이스용 그룹과 IPv6 인터페이스용 그룹의 두 등록 정보 그룹을 정의합니다. 서비스가 해당 등록 정보와 `ipadm(1M)`을 사용하여 네트워크 인터페이스를 구성합니다. 마찬가지로, 서비스가 해당 등록 정보와 `route(1M)`을 사용하여 기본 경로를 정의합니다.

149 페이지 “정적 네트워크 구성 지정”에서 예제를 참조하십시오.

`install_ipv4_interface` 등록 정보 그룹은 다음 등록 정보를 포함합니다.

표 11-6 `install_ipv4_interface` 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
<code>name</code>	<code>aststring</code>	<code>yes</code>	네트워크 인터페이스의 이름.

표 11-6 install\_ipv4\_interface 등록 정보 그룹의 등록 정보 (계속)

등록 정보	유형	필수	설명
address_type	astring	yes	ipadm(1M) create-addr 하위 명령에 대한 -T 옵션을 구성하는 데 사용된 값입니다. 유효한 값은 static 또는 dhcp입니다.
static_address	net_address_v4	no	address_type이 static인 경우에만 필요합니다. ipadm(1M) create-addr 하위 명령에 대한 로컬 주소를 구성하는 데 사용됩니다.
dhcp_wait	astring	no	address_type이 dhcp인 경우에만 적용됩니다. 정의된 경우, 이 등록 정보는 ipadm(1M) create-addr 하위 명령의 -wseconds(또는 forever) 부분을 구성하는 데 사용됩니다.
default_route	net_address_v4	no	route(1M)을 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 그룹은 다음 등록 정보를 포함합니다.

표 11-7 install\_ipv6\_interface 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
name	astring	yes	네트워크 인터페이스의 이름.
address_type	astring	yes	ipadm(1M) create-addr 하위 명령에 대한 -T 옵션을 구성하는 데 사용된 값입니다. 유효한 값은 static 또는 addrconf입니다.
static_address	net_address_v6	no	address_type이 static인 경우에만 필요합니다. ipadm(1M) create-addr 하위 명령에 대한 로컬 주소를 구성하는 데 사용됩니다.
interface_id	net_address_v6	no	address_type이 addrconf인 경우에만 적용됩니다. ipadm(1M) create-addr 하위 명령의 -i interface_id 부분을 구성하는 데 사용됩니다.
stateless	astring	no	address_type이 addrconf인 경우에만 적용됩니다. ipadm(1M) create-addr 하위 명령의 -p stateless=yes no 부분을 구성하는 데 사용됩니다.



표 11-7 install\_ipv6\_interface 등록 정보 그룹의 등록 정보 (계속)

등록 정보	유형	필수	설명
stateful	astring	no	address_type이 addrconf인 경우에만 적용됩니다. ipadm(1M) create-addr 하위 명령의 -p stateful=yes no 부분을 구성하는 데 사용됩니다.
default_route	net_address_v6	no	route(1M)을 사용하여 기본 경로를 정의하는 데 사용됩니다.  # /usr/sbin/route \ -p add default default-route \ -ifp ifname ifname 값은 name 등록 정보의 인터페이스 이름 부분입니다.

svc:/network/dns/client 서비스는 DNS 클라이언트의 구성을 지원합니다. 서비스가 하나의 등록 정보 그룹 config를 정의합니다. 서비스가 해당 등록 정보를 사용하여 DNS [resolv.conf\(4\)](#) 파일을 구성합니다.

config 등록 정보 그룹은 다음 등록 정보를 포함합니다.

표 11-8 config 등록 정보 그룹의 등록 정보

등록 정보	유형	필수	설명
도메인	astring	no	로컬 도메인 이름. resolv.conf(4)에서 domain 지시어를 구성하는 데 사용됩니다.
nameserver	net_address_list	yes	IPv4 및 IPv6 주소 목록. resolv.conf(4)에서 nameserver 지시어를 구성하는 데 사용됩니다.
search	astring_list	no	호스트 이름 조회용 검색 목록을 위한 도메인 값 목록. resolv.conf(4)에서 search 지시어를 구성하는 데 사용됩니다.

## 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
```

svc:/network/dns/client SMF 서비스는 초기 DNS 클라이언트 구성을 구성합니다. 이 서비스는 시스템 구성을 유발하지 않는 등록 정보 값을 사용하여 초기에 사용 안함으로 설정됩니다. [152 페이지 “이름 서비스 구성 지정”](#)에서 예제를 참조하십시오.

## 시스템 구성 프로파일 템플릿 사용

프로파일은 설치 서버 환경에서 또는 `create-profile` 하위 명령에 지정된 조건에서 구성 매개변수를 검색하는 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 6-1을 참조하십시오.

다음 예제 프로파일 `hostIPnet.xml`에서 `AI_HOSTNAME`은 클라이언트 시스템의 호스트 이름에 대한 자리 표시자이고, `AI_IPV4`는 클라이언트 시스템의 IP 주소에 대한 자리 표시자이고, `AI_NETWORK`는 클라이언트 시스템의 IP 버전 4 네트워크 번호에 대한 자리 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="{AI_NETWORK}"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

다음 명령은 입력 `hostandIP.xml` 파일을 변경하지 않고 설치 서비스에서 사용자 정의된 시스템 구성 프로파일을 만듭니다.

```
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server1 -c hostname="server1" -c ipv4="10.0.0.1" -c network="10.0.0.0"
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server2 -c hostname="server2" -c ipv4="10.0.0.2" -c network="10.0.0.0"
$ installadm list -n svc1 -p
Profile  Criteria
-----  -
server1  hostname = server1
         ipv4 = 10.0.0.1
         network = 10.0.0.0
server2  hostname = server2
         ipv4 = 10.0.0.2
         network = 10.0.0.0
```

`hostandIP.xml` 파일이 변경되지 않은 채, `svc1` 설치 서비스 내부의 `server1` 및 `server2` 프로파일이 사용자 정의됩니다. 예를 들어, `server1` 프로파일의 내용은 다음과 같습니다.

```
# installadm export -n svc1 -p server1
<?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="10.0.0.1/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.0"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

이와 동일한 결과를 환경 변수를 설정하여 얻을 수 있습니다. MAC 주소 또는 호스트 이름과 같이 이 클라이언트를 고유하게 식별하는 하나의 -c 옵션을 지정해야 합니다. 그래야 이러한 구성 값이 해당 클라이언트에만 적용됩니다.

```
# export AI_HOSTNAME=server1;export AI_IPV4=10.0.0.1;export AI_NETWORK=10.0.0.0
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server1 -c mac="aa:bb:cc:dd:ee:ff"
```

다음 표는 템플릿 프로파일에서 자리 표시자로 사용할 수 있는 변수를 보여줍니다. 이러한 변수는 환경 변수로 지정할 수도 있습니다.

표 11-9 시스템 구성 템플릿 프로파일의 변수

변수 이름	조건 이름	설명
AI_ARCH	arch	uname -m의 커널 구조
AI_CPU	cpu	uname -p의 프로세서 유형
AI_HOSTNAME	hostname	클라이언트 DNS 이름
AI_IPV4	ipv4	IP 버전 4 네트워크 주소, 또는 IP 주소 범위
AI_MAC	mac	16진수 MAC 주소를 콜론(:) 구분자로 표시, 또는 MAC 주소 범위
AI_MEM	mem	prtcnf로 반환된 MB 단위의 메모리 크기, 또는 메모리 크기 범위
AI_NETWORK	network	IP 버전 4 네트워크 식별자, 또는 네트워크 식별자 범위
AI_SERVICE		설치 서비스 이름
AI_ZONENAME	zonename	zoneadm list로 표시된 zones(5) 영역의 이름

## 예제 시스템 구성 프로파일

이 섹션의 예는 `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 후에 이 샘플 시스템 구성 프로파일은 `image_path/auto_install/sc_profiles/sc_sample.xml`에서 사용할 수 있습니다.

```
<?xml version='1.0'?>
<!--
Copyright (c) 2011, 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
* Timezone set to UTC
* Network configuration is automated with Network Auto-magic
* DNS name service client is enabled

See installadm(1M) 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"/>
        <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>
      <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='default' enabled='true'>
      <property_group name='environment'>
        <propval name='LC_ALL' 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 프로파일은 `/usr/share/auto_install/sc_profiles/static_network.xml`에서 사용할 수 있습니다.

이 예제 프로파일은 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.com을 로컬 DNS 도메인 이름으로 사용
- example2.com 및 example3.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 예제
8	255.0.0.0
16	255.255.0.0
24	255.255.255.0

```
<!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>
      <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>
```

```

    </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='default' enabled='true'>
      <property_group name='environment'>
        <propval name='LC_ALL' 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>

  <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='net0/v4'>
        <propval name='address_type' type='astring' value='static'>
        <propval name='static_address' type='net_address_v4' value='x.x.x.x/n'>
        <propval name='default_route' type='net_address_v4' value='x.x.x.x'>
      </property_group>

      <property_group name='install_ipv6_interface' type='application'>
        <propval name='name' type='astring' value='net0/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='x.x.x.x'>
        </net_address_list>

```

```

    </property>
    <property name='search'>
      <astring_list>
        <value_node value='example.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>

```

## 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. sysconfig를 사용하여 시스템 구성 프로파일을 만드는 방법은 67 페이지 “SCI 도구를 사용하여 구성 프로파일 만들기” 및 sysconfig(1M) 매뉴얼 페이지를 참조하십시오.

## 이름 서비스 NIS 구성

예 11-7 지정된 도메인에 대해 NIS 사용

이 예제 프로파일은 다음 구성을 수행합니다.

- my.domain.com에 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>
</service_bundle>

```



## 예 11-7 지정된 도메인에 대해 NIS 사용 (계속)

```

<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='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>

```

## 예 11-8 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">

```

## 예 11-8 NIS 구성 및 DNS 사용 안함 (계속)

```

    <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 11-9 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 -->

```

## 예 11-9 NIS 구성 (계속)

```

    <service version="1" type="service" name="network/nis/client">
      <instance enabled="true" name="default"/>
    </service>
  </service_bundle>

```

## 예 11-10 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' />
    </property_group>
  </service>

```

## 예 11-10 지정된 도메인에 대해 NIS 및 DNS 사용 (계속)

```

        <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 11-11 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>
  </service>

```

## 예 11-11 검색 목록으로 DNS 구성 (계속)

```

    <instance enabled="true" name="default"/>
  </service>
</service_bundle>

```

## 이름 서비스 LDAP 구성

## 예 11-12 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 11-13 보안LDAP 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 식별 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="ldapserver.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 AI 클라이언트 설치의 일부로 비전역 영역의 설치 및 구성을 지정하는 방법을 설명합니다.

### AI의 비전역 영역 설치 방법

비전역 영역은 전역 영역이 설치된 후에 처음 재부트 시 설치 및 구성됩니다.

1. AI를 사용하여 시스템을 설치할 때 AI 매니페스트의 `configuration` 요소를 사용하여 해당 시스템에 비전역 영역을 설치할 수 있습니다. `configuration` 요소에 대한 내용은 [162 페이지 “전역 영역 AI 매니페스트의 영역 사양”](#)을 참조하십시오.
2. 시스템이 전역 영역 설치 후에 처음 부트할 때 영역 셀프 어셈블리 SMF 서비스(`svc:/system/zones-install:default`)가 전역 영역 AI 매니페스트에 정의된 각 비전역 영역을 구성 및 설치합니다. 비전역 영역 설치에 사용된 데이터에 대한 내용은 [163 페이지 “비전역 영역 구성 및 설치 데이터”](#)를 참조하십시오.
3. 영역이 `autoboot=true`로 구성된 경우 `system/zones-install` 서비스가 설치 후에 영역을 부트합니다.

`system/zones-install` 서비스를 사용하여 레이블이 있는 영역을 만들고 설치할 수 있습니다. 레이블이 있는 영역은 `autoboot=true`로 구성되고 전역 영역에도 레이블이 붙은 경우에만 자동 부트됩니다. AI가 전역 영역을 설치하고 `system/zones-install` 서비스가 레이블이 있는 비전역 영역을 만들고 설치한 후에 필요한 내용을 변경하여 전역 영역에 레이블을 붙일 수 있습니다. 시스템 재부트 시, `svc:/system/zones:default` 서비스가 `autoboot=true`로 설정된 레이블이 있는 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 -cv`의 출력을 모니터링합니다.

다음 오류가 발생할 경우 영역이 설치되지 않습니다.

- 영역 config 파일이 구문상 정확하지 않습니다.
- 설치될 영역 집합에서 영역 이름, 영역 경로 또는 위임된 ZFS 데이터 집합 사이에 충돌이 발생합니다.
- 필요한 데이터 집합이 전역 영역에 구성되지 않았습니다.

## 전역 영역 AI 매니페스트의 영역 사양

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	필요함. 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	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="#">164 페이지 “비전역 영역 AI 매니페스트”</a> 를 참조하십시오.  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.  비전역 영역용 사용자 정의 AI 매니페스트를 제공하지 않으면 <a href="#">예 12-1</a> 에 표시된 대로 영역용 기본 AI 매니페스트가 사용됩니다.
구성 프로파일	선택 사항. 비전역 영역에 대한 0개 이상의 구성 파일을 제공할 수 있습니다. 이러한 구성 프로파일은 전역 영역 구성용 시스템 구성 프로파일과 비슷합니다. 시스템 구성 프로파일 파일에 대한 내용은 <a href="#">11 장, “클라이언트 시스템 구성”</a> 을 참조하십시오. 구성 프로파일 파일을 제공하여 사용자 및 영역 관리자의 루트 암호화 같은 영역 구성을 지정할 수 있습니다. 비전역 영역용 예제 구성 프로파일은 <a href="#">166 페이지 “비전역 영역 구성 프로파일”</a> 을 참조하십시오.  영역용 구성 프로파일 파일을 제공하려면 전역 영역을 설치 중인 설치 서비스에 구성 프로파일을 추가합니다. create-profile 명령에서 이 구성 프로파일을 사용할 모든 영역의 이름과 함께 zonename 조건 키워드를 지정합니다.  구성 프로파일 파일을 제공하지 않으면 영역의 처음 부트 시 시스템 구성 대화식 도구가 실행되어 필요한 데이터를 질의합니다. 대화식 구성 도구 사용에 대한 내용은 <a href="#">63 페이지 “시스템 구성”</a> 을 참조하십시오.

다음 예는 /tmp/zmanifest.xml AI 매니페스트를 s11-sparc 설치 서비스에 추가하고 zone1 및 zone2가 이 매니페스트를 사용하도록 지정합니다.

```
# installadm create-manifest -n s11-sparc -f /tmp/zmanifest.xml \
-m zmanifest -c zonename="zone1 zone2"
```

다음 예는 /tmp/z1profile.xml 구성 프로파일을 s11-sparc 설치 서비스에 추가하고 zone1 및 zone2가 이 프로파일을 사용하도록 지정합니다.

```
# installadm create-profile -n s11-sparc -f /tmp/z1profile.xml \
-p z1profile -c zonename="zone1 zone2"
```

다음 예는 /tmp/z2profile.xml 구성 프로파일을 s11-sparc 설치 서비스에 추가하고 zone2가 이 프로파일을 사용하도록 지정합니다.

```
# installadm create-profile -n s11-sparc -f /tmp/z2profile.xml \
-p z2profile -c zonename=zone2
```

다음 예는 s11-sparc 설치 서비스에 추가된 AI 매니페스트 및 구성 프로파일을 보여줍니다.

```
# installadm list -n s11-sparc -m -p
```

Manifest	Status	Criteria
orig_default	Default	None
line1-netra2000		mac = 00:14:4F:2D:7A:DC
zmanifest		zonename = zone1 zone2

Profile	Criteria
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 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 섹션은 추가 파일 시스템 또는 데이터 집합을 정의합니다.

logical 요소의 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 영역용 기본 AI 매니페스트를 보여줍니다. 영역용 사용자 정의 AI 매니페스트를 제공하지 않으면 이 매니페스트가 사용됩니다. 이 매니페스트는 /usr/share/auto\_install/manifest/zone\_default.xml에서 사용할 수 있습니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

Copyright (c) 2011, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.@DTD_VERSION_AI@">

<auto_install>
  <ai_instance name="zone_default">
    <target>
      <logical>
        <zpool name="rpool">
          <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>
          <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>
```

예 12-1 기본 영역 AI 매니페스트 (계속)

```

        </image>
      </destination>
      <software_data action="install">
        <name>pkg:/group/system/solaris-small-server</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>

```

## 비전역 영역 구성 프로파일

영역용 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. DNS를 구성할 수 있습니다.

영역에 허용되지 않는 구성을 지정하면 이러한 등록 정보 설정이 무시됩니다.

다음 파일은 비전역 영역용 샘플 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 완성할 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 실행할 스크립트 만들기

171 페이지 “SMF 매니페스트 파일 만들기”에 표시된 SMF 서비스 매니페스트의 위쪽 부근에 있는 다음 줄을 통해 서비스가 사용으로 설정됩니다.

```
<create_default_instance enabled='true' />
```

처음 부트 스크립트가 끝나면 서비스를 사용 안함으로 설정하고 패키지를 제거합니다. 그래야 처음 부트 스크립트가 한번만 실행됩니다.

```
#!/bin/sh
```

```
svcadm disable svc:/site/first-boot-script-svc:default
pkg uninstall pkg:/first-boot-script
```

```
exit $SMF_EXIT_OK
```

이 예제에서 first-boot-script-svc는 171 페이지 “SMF 매니페스트 파일 만들기”에서 만든 SMF 서비스이며, first-boot-script는 172 페이지 “스크립트 및 서비스에 대한 IPS 패키지 만들기”에서 만든 IPS 패키지입니다.

## 참고 -

- 여러 스크립트의 여러 명령이 서로 충돌하는 것을 피하려면 처음 부트 스크립트를 하나만 사용하십시오.
- 처음 부트 스크립트로 재부트해야 하는 경우 재부트를 스크립트의 마지막 동작으로 만듭니다.

### 예 13-1 샘플 처음 부트 스크립트

이 예는 `/opt/site/first-boot-script.sh`라는 샘플 처음 부트 스크립트를 보여줍니다. 이 스크립트는 방금 AI 설치에서 만든 부트 환경(BE)의 복사본을 먼저 저장합니다. 처음 부트 스크립트를 수정하기 전에 BE의 복사본을 저장하면, 저장된 BE로 부트하면서 스크립트에서 소개된 문제를 쉽게 복구할 수 있습니다.

```
#!/bin/sh

# Load SMF shell support definitions
. /lib/svc/share/smf_include.sh

echo "Save original boot environment first."
# 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

# Add support for faster serial console
echo "Setting up support for faster serial console"
! grep console115200 >/dev/null /etc/ttydefs && \
    echo "console115200:115200 hupcl opost onlcr:115200::console115200" \
    >>/etc/ttydefs

echo "Configure ssh server for root login and X11 forwarding"
ed - << EOF
r /etc/ssh/sshd_config
/PermitRootLogin/
c
PermitRootLogin yes
.
/X11Forwarding/
c
X11Forwarding yes
.
w
q
EOF
svcadm refresh ssh

# Set up coreadm
echo "Setting core file configuration"
coreadm -G default -g /var/cores/%f.%u.%p.%t.core
coreadm -e global
coreadm -e process
coreadm -e proc-setid
```

## 예 13-1 샘플 처음 부트 스크립트 (계속)

```
coreadm -e log

# Disable service and uninstall package
svcadm disable svc:/site/first-boot-script-svc:default
pkg uninstall pkg:/first-boot-script

echo "Site first-boot script done. Rebooting in 5 seconds." > /dev/console

sleep 5 && reboot -p || reboot &

exit $SMF_EXIT_OK
```

## SMF 매니페스트 파일 만들기

스크립트를 실행하는 일시적 서비스를 정의하는 SMF 매니페스트 파일을 만듭니다.

- 아래의 예제 서비스 매니페스트 끝 부분에 `duration` 등록 정보가 `transient` 값을 가집니다. 일시적 서비스는 `start` 메소드를 한번 실행하고, `$SMF_EXIT_OK`로 메소드를 종료하면 다시 실행하지 않습니다. `svc.startd` 데몬은 스크립트의 첫번째 실행 후에 다시 시작을 시도하지 않습니다.
- 서비스의 `start` 메소드가 처음 부트 스크립트를 실행합니다.
- 이 예에서 서비스의 이름은 `site/first-boot-script-svc`입니다. 클라이언트를 부트한 후에 다음 명령의 출력에서 서비스를 볼 수 있습니다.

```
$ svcs -a|grep first-boot-script
STATE      STIME      FMRI
disabled   13:51:42   svc:/site/first-boot-script-svc:default
```

- 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 예상한 방법으로 시스템을 구성하기 전에 스크립트가 실행될 수 있습니다.

---

**참고** - 스크립트의 종속성을 평가하고 만족스러운 후에 서비스가 스크립트를 실행하도록 구성합니다.

---

## 예 13-2 샘플 SMF 서비스 매니페스트

다음 파일은 `first-boot-script-svc` 서비스에 대한 SMF 매니페스트 파일인 `first-boot-script-svc-manifest.xml`입니다. 이 서비스는 기본적으로 사용으로 설정되고 다시 시작하지 않습니다. 이 예에서 서비스가 실행하는 스크립트는 `/opt/site/first-boot-script.sh`입니다.

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type='manifest' name='first-boot-script:site-first-boot-script-svc'>
```

## 예 13-2 샘플 SMF 서비스 매니페스트 (계속)

```

<service
  name='site/first-boot-script-svc'
  type='service'
  version='1'>
  <create_default_instance enabled='true' />
  <single_instance />

  <!-- Run the script late in the startup sequence after first boot. -->
  <dependency name='multi-user' grouping='require_all' restart_on='none' type='service'>
    <service_fmri value='svc:/milestone/multi-user:default' />
  </dependency>

  <exec_method
    type='method'
    name='start'
    exec='/opt/site/first-boot-script.sh'
    timeout_seconds='360'>
    <method_context>
      <method_credential user='root' />
    </method_context>
  </exec_method>

  <exec_method
    type='method'
    name='stop'
    exec=':true'
    timeout_seconds='60'
  />

  <property_group name='startd' type='framework'>
    <propval name='duration' type='astring' value='transient' />
  </property_group>

</service>
</service_bundle>

```

## 스크립트 및 서비스에 대한 IPS 패키지 만들기

다음은 포함하는 IPS 패키지를 만듭니다.

- 171 페이지 “SMF 매니페스트 파일 만들기”에서 만든 서비스 매니페스트 파일
- 169 페이지 “처음 부트 시 실행할 스크립트 만들기”에서 만든 처음 부트 스크립트
- 설치 서버와 같은 다른 위치에서 제공할 수 없는, 스크립트에 필요한 파일

## ▼ IPS 패키지를 만들어 게시하는 방법

### 1 디렉토리 계층을 만듭니다.

이 예에서 서비스 매니페스트가 `/lib/svc/manifest/site`로 설치되고 처음 부트 스크립트가 `/opt/site`로 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.p5m`이라는 파일을 만듭니다.

```
set name=pkg.fmri value=first-boot-script@1.0,5.11
set name=pkg.summary value="AI first boot script"
set name=pkg.description value="Script that runs at first boot after AI installation"
file path=lib/svc/manifest/site/first-boot-script-svc-manifest.xml mode=0444 \
  owner=root group=sys
dir path=opt/site mode=0755 owner=root group=sys
file path=opt/site/first-boot-script.sh mode=0555 owner=root group=sys
```

처음 부트 스크립트가 무엇인지에 따라, 종속성을 지정해야 할 수 있습니다. 이 매니페스트를 수정하면 `pkglint`를 사용하여 새 매니페스트가 올바른지 확인합니다. 경고를 무시할 수 있습니다.

```
# pkglint first-boot-script.p5m
```

### 3 패키지 저장소를 만듭니다.

이 예는 `firstboot`를 게시자로 사용하여 로컬 디렉토리에 저장소를 만듭니다.

---

주 - 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@1.0,5.11:20111101T024901Z
PUBLISHED
```

클라이언트가 `firstbootrepo` 저장소에서 패키지를 설치할 수 있습니다. 다음 섹션에 표시된 대로 `firstbootrepo` 원본과 함께 `firstboot` 게시자가 AI 매니페스트에 정의됩니다.

패키지를 나열하여 패키지가 사용 가능한지 확인합니다.

```
$ pkg list -g ./firstbootrepo first-boot-script
NAME (PUBLISHER)          VERSION  IFO
first-boot-script (firstboot)  1.0      ---
```

패키지의 테스트 설치를 시도해 볼 수 있습니다. -n 옵션은 패키지가 실제로 설치되지 않음을 의미합니다.

```
# pkg set-publisher -g ./firstbootrepo firstboot
# pkg install -nv first-boot-script
    Packages to install:      1
    Estimated space available: 111.26 GB
    Estimated space to be consumed: 72.42 MB
    Create boot environment:  No
    Create backup boot environment: No
    Rebuild boot archive:     No

Changed packages:
firstboot
first-boot-script
None -> 1.0,5.11:20111101T024901Z
```

다음 순서 NFS 공유 또는 HTTP를 통해 새 저장소에서 클라이언트 시스템에 액세스 가능하도록 만드는 방법은 [Oracle Solaris 11 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 [10 장, “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 sharenfs`를 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.**

AI로 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.





## AI 클라이언트 시스템에서 사용할 Oracle Configuration Manager 설정

---

Oracle Configuration Manager를 사용하여 시스템 구성을 My Oracle Support에 기록할 수 있습니다. 이를 위해 Oracle은 사전 예방적 맞춤형 지원을 제공할 수 있습니다.

이 장에서는 Oracle Configuration Manager가 AI 클라이언트 설치에서 작동하도록 보증하는 방법을 설명합니다.

### AI 클라이언트에서 Oracle Configuration Manager의 기본 동작

클라이언트 처음 부트가 network 마일스톤에 도달하면 system/ocm SMF 서비스가 시작됩니다.

system/ocm 서비스는 opt\_out 등록 정보가 지정되었는지 여부를 검사합니다. opt\_out 등록 정보는 기본적으로 지정되지 않습니다.

- opt\_out 등록 정보가 지정된 경우 서비스가 스스로 사용 안함으로 설정되어 종료됩니다. opt\_out 등록 정보를 지정하려면 180 페이지 “Oracle Configuration Manager의 옵트아웃”을 참조하십시오.
- opt\_out 등록 정보가 지정되지 않은 경우 system/ocm 서비스가 response\_file\_pkg\_name 등록 정보가 지정되었는지 여부를 검사합니다. response\_file\_pkg\_name 등록 정보는 기본적으로 지정되지 않습니다.
  - response\_file\_pkg\_name 등록 정보가 지정되지 않은 경우 Oracle Configuration Manager가 기본 응답 파일을 사용합니다.  
Oracle Configuration Manager가 익명 등록을 수행하고 시스템 데이터를 수집하여 인터넷 위치 <https://ccr.oracle.com/>으로 보내기 시작합니다.
  - response\_file\_pkg\_name 등록 정보가 지정된 경우 Oracle Configuration Manager가 해당 패키지에 전달된 사용자 정의 응답 파일을 사용합니다. 파일 이름이 아닌, 패키지 이름이 이 등록 정보에 제공됩니다. response\_file\_pkg\_name 등록 정보를 지정하려면 178 페이지 “사용자 정의 응답 파일 제공”을 참조하십시오.

이 사용자 정의 응답 파일을 사용하여 내 지원 ID를 이 시스템 구성 정보와 연관짓고, 인터넷 액세스가 없는 시스템에 대해 프록시나 Oracle 지원 허브를 지정할 수 있습니다.

## 사용자 정의 응답 파일 제공

다음 정보를 지정하려면 사용자 정의 응답 파일을 제공해야 합니다.

- 내 고객 지원 ID를 통해 이 데이터를 회사와 연관지을 수 있습니다.
- 프록시나 Oracle 지원 허브를 통해 데이터 수집 후 인터넷 액세스가 없는 시스템에 보낼 수 있습니다.

### ▼ 사용자 정의 응답 파일 패키지를 만들고 설치하는 방법

다음 단계에 따라 사용자 정의 응답 파일에 사용자 정의 IPS 패키지를 제공합니다.

#### 1 사용자 정의 응답 파일을 만듭니다.

기존 Oracle Configuration Manager 설치를 사용하여 응답 파일을 만듭니다. 사용자 정의 응답 파일 만들기에 대한 내용은 **Oracle Configuration Manager Installation and Administration Guide**의 "[Creating a Response File \(http://download.oracle.com/docs/cd/E23562\\_01/doc.1035/e22050/admin.htm#BAJICIFE\)](http://download.oracle.com/docs/cd/E23562_01/doc.1035/e22050/admin.htm#BAJICIFE)"을 참조하십시오. 응답 파일 이름은 ocm.rsp여야 합니다.

#### 2 사용자 정의 IPS 패키지를 만듭니다.

사용자 정의 응답 파일을 포함하는 IPS 패키지를 만듭니다. 서로 다른 시스템에 서로 다른 사용자 정의 응답 파일을 만들려면 각 응답 파일을 별도의 패키지에 넣습니다. 경로 및 파일 이름은 각 사용자 정의 응답 파일마다 같습니다. 패키지 이름은 서로 다를 수 있습니다. Oracle Configuration Manager 사용자 정의 응답 파일 패키지 하나만 AI 매니페스트에 포함할 수 있습니다.

##### a. 디렉토리 계층을 만듭니다.

사용자 정의 응답 파일은 AI 클라이언트 시스템에 /var/ocm/.rsp/ocm.rsp로 설치해야 합니다.

```
$ mkdir -p proto/var/ocm/.rsp
$ cp ocm.rsp proto/var/ocm/.rsp
```

##### b. 패키지 매니페스트를 만듭니다.

다음과 같이 ocsresp.p5m이라는 파일을 만듭니다.

```
set name=pkg.fmri value=ocsresp@1.0,5.11
set name=pkg.summary value="Oracle Configuration Manager custom response file"
set name=pkg.description value="Oracle Configuration Manager custom response file"
```

```
depend fmri=pkg:/system/ocm type=require
file var/ocm/.rsp/ocm.rsp group=root mode=0444 owner=root path=var/ocm/.rsp/ocm.rsp
```

depend 및 file 줄만 필수입니다. summary 및 description 줄은 이 패키지를 저장소에서 찾거나 보려는 경우 유용합니다.

### c. 패키지 저장소를 만듭니다.

이 예는 ocm을 게시자로 사용하여 로컬 디렉토리에 저장소를 만듭니다.

---

주 - 설치할 때 AI 클라이언트에서 액세스 가능한 디렉토리에 저장소를 만듭니다.

---

```
$ pkgrepo create customocm
# pkgrepo -s customocm add-publisher ocm
```

### d. 패키지를 게시합니다.

```
$ pkgsend publish -d ./proto -s ./customocm ocsresp.p5m
pkg://ocm/ocsresp@1.0,5.11:20110708T174359Z
PUBLISHED
```

클라이언트가 customocm 저장소에서 패키지를 설치할 수 있습니다. 다음 단계에 표시된 대로 customocm 원본과 함께 ocm 게시자가 AI 매니페스트에 정의됩니다.

NFS 공유 또는 HTTP를 통해 새 저장소에서 클라이언트 시스템에 액세스 가능하도록 만드는 방법은 [Oracle Solaris 11 패키지 저장소 복사 및 만들기](#)를 참조하십시오.

## 3 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 [10 장, “클라이언트 시스템 프로비전”](#)을 참조하십시오.

installadm export 명령을 사용하여 하나 이상의 기존 AI 매니페스트의 내용을 검색합니다. 다음 예는 추가할 XML 요소를 보여줍니다. Oracle Configuration Manager 사용자 정의 응답 파일 패키지 하나만 AI 매니페스트에 포함할 수 있습니다.

```
<software type="IPS">
  <source>
    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
    <publisher name="ocm">
      <origin name="file:///net/host/export/customocm"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg://ocm/ocsresp</name>
  </software_data>
</software>
```

원본이 클라이언트가 AI 설치 중 액세스할 수 있는 URI인지 확인합니다. `zfs set sharenfs`를 사용하여 저장소를 내보내면 클라이언트가 로컬 저장소에 액세스할 수 있습니다.

#### 4 AI 설치 서비스에서 수정된 AI 매니페스트를 업데이트합니다.

`installadm update-manifest` 명령을 사용하여 AI 매니페스트 내용을 사용자 정의 Oracle Configuration Manager 응답 파일 패키지가 포함된 내용으로 바꿉니다. 업데이트 이후의 매니페스트나 스크립트에 어떤 조건이나 기본 상태는 그대로 남습니다.

#### 5 사용자 정의 응답 파일 패키지 등록 정보를 시스템 구성 프로파일에 추가합니다.

클라이언트가 원하는 수의 시스템 구성 프로파일을 사용할 수 있으므로 이 등록 정보 설정을 별도의 프로파일에 넣을 수 있습니다. 그러면 일부 또는 모든 클라이언트에 대해 이 설정을 쉽게 추가할 수 있습니다.

```
<service name='system/ocm' type='service' version='1' >
  <instance name='default' enabled='true' >
    <property_group name='reg' type='framework'>
      <propval name='response_file_pkg_name' type='astring' value='pkg:/ocsresp' />
    </property_group>
  </instance>
</service>
```

#### 6 수정된 시스템 구성 프로파일을 AI 설치 서비스에 추가합니다.

`installadm` 명령의 `create-profile` 하위 명령을 사용하여 신규 또는 사용자 정의된 시스템 구성 프로파일을 AI 설치 서비스에 추가합니다.

다음 예에서 `filename`은 구성 프로파일의 전체 경로 이름입니다. `criteria.xml` 파일은 어떤 클라이언트가 이 프로파일을 사용할지 지정합니다. 이 프로파일이 응답 파일 패키지가 이름 설정만 포함하고 동일한 사용자 정의 응답 파일을 모든 클라이언트가 사용해야 하는 경우 `-C` 또는 `-c` 옵션을 생략합니다.

```
# installadm create-profile -n svcname -f filename -C criteria.xml
```

## Oracle Configuration Manager의 옵트아웃

일부 클라이언트에 대해 Oracle Configuration Manager를 사용하지 않도록 선택하려면 다음 요소를 해당 클라이언트의 시스템 구성 프로파일에 추가합니다.

```
<service name='system/ocm' type='service' version='1' >
  <instance name='default' enabled='false' >
    <property_group name='reg' type='framework'>
      <propval name='opt_out' type='astring' value='true' />
    </property_group>
  </instance>
</service>
```

해당 클라이언트의 식별 조건이 포함된 AI 설치 서비스에 신규 또는 수정된 구성 프로파일을 추가합니다.

## 클라이언트 시스템 설치

이 장에서는 AI 클라이언트에 대한 시스템 요구 사항을 제공하고 각 클라이언트를 올바른 AI 설치 서비스와 연관시키는 방법을 설명합니다.

### 클라이언트가 설치되는 방법

설치 서버를 설정할 때 각 클라이언트 구조마다, 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 클라이언트 IP 주소, 부트 파일 및 설치 서버의 IP 주소를 검색합니다.
3. 클라이언트 시스템이 다음 소스 중 하나에서 넷 이미지를 로드합니다.
  - `installadm create-client` 명령으로 이 클라이언트에 할당된 설치 서비스
  - 이 구조에 대한 기본 설치 서비스
4. 클라이언트 시스템이 108 페이지 “AI 매니페스트 선택”에 설명된 대로 결정된 AI 매니페스트를 사용하여 설치를 완료합니다.
5. `auto_reboot`가 AI 매니페스트에 설정된 경우 클라이언트 시스템이 재부트되고, 또는 시스템 관리자가 클라이언트를 재부트합니다.
6. 재부트 중, 다음 방법 중 하나로 클라이언트 시스템이 구성됩니다.
  - 109 페이지 “시스템 구성 프로파일 선택”에 설명된 대로 결정된 시스템 구성 프로파일 사용
  - 대화식 시스템 구성 도구에서 관리자의 응답 사용

AI 클라이언트 설치를 마치면 "Automated Installation succeeded(자동 설치 성공)" 메시지가 화면에 표시되고, `/system/volatile/install_log` 로그 파일에 완료 메시지가 표시되고, 해당 클라이언트의 `svc:/application/auto-installer` SMF 서비스가 online 상태에 도달합니다.

## 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 설치 클라이언트로 사용할 수 있습니다.

### SPARC 및 x86 클라이언트 시스템 요구 사항

네트워크를 통한 AI 설치의 SPARC 및 x86 클라이언트는 다음 요구 사항을 충족해야 합니다.

<b>메모리</b>	1GB 최소
<b>디스크 공간</b>	13GB 최소
<b>네트워크 액세스</b>	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>■ AI 설치 서버</li> <li>■ 클라이언트 시스템에 설치될 패키지를 포함하는 IPS 저장소</li> </ul>

### 추가 SPARC 클라이언트 시스템 요구 사항

<b>펌웨어</b>	최신 WAN 부트 지원을 포함하는 현재 버전의 OBP(Open Boot PROM)를 포함하도록 SPARC 클라이언트의 펌웨어를 업데이트해야 합니다.
<b>WAN 부트</b>	네트워크를 통한 AI 설치의 SPARC 클라이언트는 WAN 부트를 지원해야 합니다.

네트워크를 통해 부트하려면 AI에서 SPARC 클라이언트에 WAN 부트 지원이 필요합니다. 클라이언트 OBP(Open Boot PROM)가 WAN 부트를 지원하는지 여부를 확인할 수 있습니다. `network-boot-arguments`가 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. 5 장, “매체에서 부트하는 자동 설치”를 참조하십시오.

```
# eeprom | grep network-boot-arguments
```

## 설치 클라이언트 설정

설치 서버에 `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 선택적으로 부트 등록 정보를 지정할 수 있습니다.

## x86 클라이언트 설정

다음 예는 MAC 주소 `0:e0:81:5d:bf:e0`의 x86 클라이언트를 `s11-x86` 설치 서비스와 연관시킵니다. 이 명령에 의한 DHCP 구성 출력을 DHCP 서버에 추가해야 합니다. 이 DHCP 구성이 완료되지 않으면 클라이언트가 `s11-x86` 설치 서비스를 부트할 수 없습니다.

```
# installadm create-client -n s11-x86 -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           : 0100E0815DBFE0
```

`/etc/netboot` 디렉토리에서 `installadm create-client` 명령의 결과를 볼 수도 있습니다. 이 목록은 이 클라이언트의 설치 서비스 파일만 보여줍니다. `menu.lst` 파일의 줄은 가독성을 위해 구분한 것입니다.

```
# cd /etc/netboot
# ls -l
lrwxrwxrwx  Aug 26 08:27 0100E0815DBFE0 -> ./s11-x86/boot/grub/pxegrub
-rw-r--r--  Aug 26 08:27 menu.lst.0100E0815DBFE0
drwxr-xr-x  Aug 26 08:26 s11-x86

# cat menu.lst.0100E0815DBFE0
default=0
timeout=30
```

```

min_mem64=0
title Oracle Solaris 11 11/11 Text Installer and command line
    kernel$ /s11-x86/platform/i86pc/kernel/$ISADIR/unix -B
install_media=http://$serverIP:5555//export/auto_install/s11-x86,install_service=s11-x86,
install_svc_address=$serverIP:5555
    module$ /s11-x86/platform/i86pc/$ISADIR/boot_archive

title Oracle Solaris 11 11/11 Automated Install
    kernel$ /s11-x86/platform/i86pc/kernel/$ISADIR/unix -B install=true,
install_media=http://$serverIP:5555//export/auto_install/s11-x86,install_service=s11-x86,
install_svc_address=$serverIP:5555,livemode=text
    module$ /s11-x86/platform/i86pc/$ISADIR/boot_archive

```

## SPARC 클라이언트 설정

다음 예는 MAC 주소 00:14:4f:a7:65:70의 SPARC 클라이언트를 s11-sparc 설치 서비스와 연관시킵니다.

```
# installadm create-client -n s11-sparc -e 00:14:4f:a7:65:70
```

SPARC wanboot-cgi 부트 파일이 create-service로 이미 구성되었으므로 DHCP 서버에 구성이 필요 없습니다. [86 페이지 “AI 설치 서비스 만들기”](#)를 참조하십시오.

## 클라이언트를 서비스에서 삭제

installadm delete-client 명령을 사용하여 클라이언트를 설치 서비스에서 삭제합니다.

```
# installadm delete-client macaddr
```

클라이언트는 하나의 설치 서비스에만 연관될 수 있으므로 서비스 이름을 지정할 필요가 없습니다.

## 클라이언트 설치

설치를 시작하려면 클라이언트를 부트합니다. 이 섹션은 SPARC 또는 x86 클라이언트를 부트하는 방법을 정확히 보여줍니다. 또한 설치 진행률을 원격으로 모니터링할 수 있는 방법도 설명합니다.

## 보안 셸을 사용하여 원격으로 설치 모니터

ssh를 사용하여 자동 설치 클라이언트에 대한 네트워크 액세스를 사용으로 설정할 수 있습니다. 이 액세스를 사용하여 설치 진행 상황을 원격으로 관찰할 수 있습니다.



설치 구성 파일에서 `livessh` 옵션을 `enable`로 설정하여 원격 액세스를 사용으로 설정합니다. 이 액세스가 사용으로 설정된 경우 사용자 이름 `jack`과 암호 `jack`을 사용하여 AI 클라이언트에 로그인할 수 있습니다.

## x86 클라이언트 설치 모니터

x86 시스템의 경우 `menu.lst` 구성 파일이 다음 방법으로 생성됩니다.

- `installadm create-client` 명령을 사용한 경우 파일 이름이 `/etc/netboot/menu.lst.01MAC_address`입니다. 여기서 `MAC_address`는 `installadm create-client` 명령에 지정된 MAC 주소입니다.
- `installadm create-client` 명령을 사용하지 않은 경우 파일 이름이 `/etc/netboot/service_name/menu.lst`입니다. 여기서 `service_name`은 `installadm create-service` 명령으로 만든 설치 서비스의 이름입니다.

이 파일에서 옵션이 커널 매개변수로 제공됩니다. 다음 예에서 `livessh` 및 `install_debug` 옵션이 `enable`로 설정됩니다.

```
kernel$ ... -B install_media=...,livessh=enable,install_debug=enable
```

## 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 액세스할 수 있습니다.

`system.conf` 파일에서 옵션은 이름-값 쌍으로 정의됩니다. 다음 예에서 `livessh` 옵션이 `enable`로 설정됩니다.

```
$ cat /etc/netboot/svc1/system.conf
...
livessh=enable
...
```

## SPARC 클라이언트 설치

다음 명령을 사용하여 OBP 프롬프트에서 SPARC 클라이언트를 부트합니다.

```
ok boot net:dhcp - install
```

## SPARC 클라이언트 네트워크 부트 시퀀스

다음 이벤트가 SPARC 클라이언트의 AI 부트 중 발생합니다.

1. 클라이언트가 부트하고 DHCP 서버에서 IP 주소와 wanboot-cgi 파일의 위치를 가져옵니다.
2. wanboot-cgi 프로그램이 wanboot.conf를 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.zlib이 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.

## 샘플 SPARC 네트워크 부트 출력

다음과 같은 부트 시퀀스의 출력이 표시됩니다.

```
{0} ok boot net:dhcp - install
```

```
SPARC Enterprise T5120, No Keyboard
Copyright 2008 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.29.1, 16256 MB memory available, Serial #81036844.
Ethernet address 0:14:4f:d4:86:2c, Host ID: 84d4862c.
```

```
Boot device: /pci@0/pci@0/pci@1/pci@0/pci@2/network@0:dhcp File and args: - install
1000 Mbps full duplex Link up
Timed out waiting for BOOTP/DHCP reply
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: configuring /pci@0/pci@0/pci@1/pci@0/pci@2/network@0:dhcp
```

```
1000 Mbps full duplex Link up
<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
Fri Aug 26 14:34:53 wanboot progress: miniroot: Read 221327 of 221327 kB (100%)
Fri Aug 26 14:34:53 wanboot info: miniroot: Download complete
SunOS Release 5.11 Version snv_175 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-08-26 13:47:31-- http://host1:5555/install/images/sparc_snv175//solaris.zlib
```

```

idn_decode failed (9): 'System iconv failed'
Resolving host1... 10.80.238.5
idn_decode failed (9): 'System iconv failed'
Connecting to host1|10.80.238.5|:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 135977984 (130M) [text/plain]
Saving to: '/tmp/solaris.zlib'

100%[=====>] 135,977,984 22.3M/s   in 6.0s

2011-08-26 13:47:37 (21.7 MB/s) - '/tmp/solaris.zlib' saved [135977984/135977984]

Downloading solarismisc.zlib
--2011-08-26 13:47:37-- http://host1:5555/install/images/sparc_snv175//solarismisc.zlib
idn_decode failed (9): 'System iconv failed'
Resolving host1... 10.80.238.5
idn_decode failed (9): 'System iconv failed'
Connecting to host1|10.80.238.5|:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18826752 (18M) [text/plain]
Saving to: '/tmp/solarismisc.zlib'

100%[=====>] 18,826,752 21.7M/s   in 0.8s

2011-08-26 13:47:37 (21.7 MB/s) - '/tmp/solarismisc.zlib' saved [18826752/18826752]

Downloading .image_info
--2011-08-26 13:47:37-- http://host1:5555/install/images/sparc_snv175//.image_info
idn_decode failed (9): 'System iconv failed'
Resolving host1... 10.80.238.5
idn_decode failed (9): 'System iconv failed'
Connecting to host1|10.80.238.5|:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 82 [text/plain]
Saving to: '/tmp/.image_info'

100%[=====>] 82          --.-K/s   in 0s

2011-08-26 13:47:37 (2.22 MB/s) - '/tmp/.image_info' saved [82/82]

Done mounting image
Configuring devices.
Hostname: solaris
Service discovery phase initiated
Service name to look up: default-sparc
Service discovery finished successfully
Process of obtaining install manifest initiated
Using the install manifest obtained via service discovery

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.

solaris console login: 13:48:35   Install Log: /system/volatile/install_log
13:48:35   Using XML Manifest: /system/volatile/ai.xml
13:48:35   Using profile specification: /system/volatile/profile
13:48:35   Using service list file: /var/run/service_list

```

```

13:48:35 Starting installation.
13:48:35 0% Preparing for Installation
13:48:36 100% manifest-parser completed.
13:48:36 0% Preparing for Installation
13:48:36 1% Preparing for Installation
13:48:37 2% Preparing for Installation
13:48:37 3% Preparing for Installation
13:48:37 4% Preparing for Installation
13:48:47 7% target-discovery completed.
13:48:47 === Executing Target Selection Checkpoint ==
13:48:48 Selected Disk(s) : c3t0d0
13:48:48 13% target-selection completed.
13:48:48 17% ai-configuration completed.
13:49:01 19% target-instantiation completed.
13:49:02 19% Beginning IPS transfer
13:49:02 Creating IPS image
13:50:58 Installing packages from:
13:50:58     solaris
13:50:58         origin: http://pkg.example.com/solaris/
14:48:40 21% generated-transfer-1491-1 completed.
14:48:41 23% initialize-smf completed.
14:48:43 Installing SPARC bootblk to root pool devices: ['/dev/rdisk/c3t0d0s0']
14:48:43 Setting openprom boot-device
14:48:44 33% boot-configuration completed.
14:48:44 35% update-dump-adm completed.
14:48:45 37% setup-swap completed.
14:48:45 40% set-flush-ips-content-cache completed.
14:48:47 42% device-config completed.
14:48:49 44% apply-sysconfig completed.
14:48:49 46% transfer-zpool-cache completed.
14:49:03 87% boot-archive completed.
14:49:04 89% transfer-ai-files completed.
14:49:04 99% create-snapshot completed.
14:49:05 Automated Installation succeeded.
14:49:05 You may wish to reboot the system at this time.
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/sadm/system/logs/install_log

```

## 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.

## x86 클라이언트 네트워크 부트 시퀀스

다음 이벤트가 x86 클라이언트의 AI 부트 중 발생합니다.

1. 클라이언트가 부트하여 IP 주소를 가져오고, DHCP 서버에서 제공된 위치로부터 부트 파일 pxegrub가 다운로드됩니다.
2. pxegrub 부트 파일이 로드되고 menu.lst 파일을 읽습니다.
3. pxegrub 부트 파일이 boot\_archive 파일을 가져오고 Oracle Solaris OS가 TFTP를 사용하여 부트됩니다.
4. 넷 이미지 아카이브 solaris.zlib 및 solarismisc.zlib가 GRUB 메뉴에서 제공된 대로 HTTP를 사용하여 다운로드됩니다.
5. mDNS 조회에서 또는 부트된 GRUB 메뉴 항목에서 지정된 AI 설치 서비스로부터 AI 매니페스트와 시스템 구성 프로파일이 다운로드됩니다.
6. AI 설치 프로그램이 AI 매니페스트로 호출되어 설치를 수행합니다.

## 샘플 x86 네트워크 부트 출력

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 11/11 Text Installer and command line
Oracle Solaris 11 11/11 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 202 페이지 "설치 시작 없이 부트 후에 설치 시작"에 표시된 대로 메뉴가 표시됩니다. 이 메뉴를 사용하여 시스템을 조사하거나 설치합니다.

일단 이미지를 선택하면 다음 메시지가 표시됩니다.

```
Remounting root read/write snv_175 64-bit
Probing for device nodes ...acle and/or its affiliates. All rights reserved.
Preparing network image for use
Downloading solaris.zlib
--2011-08-26 07:35:13-- http://10.80.238.5:5555/install/images/i386_snv175/solaris.zlib
Connecting to 10.80.238.5:5555... connected.
```

HTTP request sent, awaiting response... 200 OK  
 Length: 130032128 (124M) [text/plain]  
 Saving to: '/tmp/solaris.zlib'

100%[=====>] 130,032,128 16.4M/s in 8.3s

2011-08-26 07:35:22 (14.9 MB/s) - '/tmp/solaris.zlib' saved [130032128/130032128]

Downloading solarismisc.zlib  
 --2011-08-26 07:35:22-- http://10.80.238.5:5555//install/images/i386\_snv175/solarismisc.zlib  
 Connecting to 10.80.238.5:5555... connected.  
 HTTP request sent, awaiting response... 200 OK  
 Length: 18758144 (18M) [text/plain]  
 Saving to: '/tmp/solarismisc.zlib'

100%[=====>] 18,758,144 21.7M/s in 0.8s

2011-08-26 07:35:23 (21.7 MB/s) - '/tmp/solarismisc.zlib' saved [18758144/18758144]

Downloading .image\_info  
 --2011-08-26 07:35:23-- http://10.80.238.5:5555//install/images/i386\_snv175/.image\_info  
 Connecting to 10.80.238.5:5555... connected.  
 HTTP request sent, awaiting response... 200 OK  
 Length: 241 [text/plain]  
 Saving to: '/tmp/.image\_info'

100%[=====>] 241 --.-K/s in 0s

2011-08-26 07:35:23 (29.3 MB/s) - '/tmp/.image\_info' saved [241/241]

Done mounting image  
 Configuring devices.  
 Hostname: solaris  
 Service discovery phase initiated  
 Service name to look up: default-i386  
 Service discovery finished successfully  
 Process of obtaining install manifest initiated  
 Using the install manifest obtained via service discovery

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.

solaris console login: 07:35:35 Install Log: /system/volatile/install\_log  
 07:35:35 Using XML Manifest: /system/volatile/ai.xml  
 07:35:35 Using profile specification: /system/volatile/profile  
 07:35:35 Using service list file: /var/run/service\_list  
 07:35:36 Starting installation.  
 07:35:36 0% Preparing for Installation  
 07:35:36 100% manifest-parser completed.  
 07:35:36 0% Preparing for Installation  
 07:35:36 1% Preparing for Installation  
 07:35:36 2% Preparing for Installation  
 07:35:36 3% Preparing for Installation  
 07:35:37 4% Preparing for Installation  
 07:35:51 7% target-discovery completed.  
 07:35:51 === Executing Target Selection Checkpoint ==

```

07:35:51 Selected Disk(s) : c7t0d0
07:35:51 13% target-selection completed.
07:35:51 17% ai-configuration completed.
07:36:23 19% target-instantiation completed.
07:36:23 19% Beginning IPS transfer
07:36:23 Creating IPS image
07:36:41 Installing packages from:
07:36:41     solaris
07:36:41     origin: http://pkg.example.com/solaris/
07:53:29 21% generated-transfer-1006-1 completed.
07:53:29 23% initialize-smf completed.
07:53:30 Setting console boot device property to ttya
07:53:30 Disabling graphical console in boot loader
07:53:30 Creating Legacy GRUB config directory:
/rpool/boot/grub
07:53:30 Installing boot loader to devices: ['/dev/rdisk/c7t0d0s0']
07:53:31 33% boot-configuration completed.
07:53:31 35% update-dump-adm completed.
07:53:31 37% setup-swap completed.
07:53:31 40% set-flush-ips-content-cache completed.
07:53:32 42% device-config completed.
07:53:46 44% apply-sysconfig completed.
07:53:46 46% transfer-zpool-cache completed.
07:54:08 87% boot-archive completed.
07:54:08 89% transfer-ai-files completed.
07:54:09 99% create-snapshot completed.
07:54:09 Automated Installation succeeded.
07:54:09 You may wish to reboot the system at this time.
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/sadm/system/logs/install_log

```

## 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. 일단 IPS의 패키지 설치가 시작되면 장시간이 로그 파일의 업데이트를 볼 수 없습니다.

## 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/sadm/system/logs/install_log
```

AI 매니페스트에 자동 재부트를 설정한 경우 이 시점에 시스템이 재부트됩니다. 설치 성공 후에 자동 재부트를 지정하려면 <ai\_instance> 태그의 `auto_reboot` 속성을 `true`로 설정합니다. 기본값은 `false`이며, 클라이언트가 설치 성공 후에 자동으로 재부트하지 않습니다.



## 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 시스템의 설치를 실패한 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.

#### 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/dhccpinfo 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.

- 194 페이지 “SPARC 네트워크 부팅 오류 및 가능한 원인”
- 197 페이지 “x86 네트워크 부팅 오류 및 가능한 원인”
- 199 페이지 “SPARC 및 x86 오류 메시지”

## SPARC 네트워크 부팅 오류 및 가능한 원인

이 섹션은 네트워크를 통해 SPARC 클라이언트를 부트할 때 볼 수 있는 오류/문제와 가능한 원인을 설명합니다.

- 194 페이지 “BOOTP/DHCP 릴레이 대기 중 시간 초과”
- 194 페이지 “부트 로드 실패”
- 195 페이지 “내부 서버 오류 또는 WAN 부트 경보”
- 195 페이지 “오류 메시지 403: 사용 권한 없음 또는 404: 찾을 수 없음”
- 196 페이지 “Automated Installer 사용 안함”

## 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
Timed out waiting for BOOTP/DHCP reply
Timed out waiting for BOOTP/DHCP reply
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

AI 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 정보

AI 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
Tue Aug 5 20:46:43 wanboot alert: miniinfo: Request returned code 500
Tue Aug 5 20:46:44 wanboot alert: Internal Server Error \
(root filesystem image missing)
```

- AI 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
Tue Aug 5 20:53:02 wanboot alert: miniroot: Request returned code 403
Tue 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.

## 오류 메시지 403: 사용 권한 없음 또는 404: 찾을 수 없음

AI 클라이언트가 boot\_archive를 성공적으로 다운로드하고 Oracle Solaris 커널을 부트했지만 이미지 아카이브 중 하나를 얻지 못한 경우 "오류 403: 사용 권한 없음" 및 "오류 404: 찾을 수 없음" 메시지가 표시됩니다. 어떤 파일이 문제를 일으키는 지 나타내는 오류 메시지가 표시됩니다. 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
Fri Aug 26 16:26:52 wanboot progress: miniroot: Read 221327 of 221327 kB (100%)
Fri Aug 26 16:26:53 wanboot info: miniroot: Download complete
Warning: 'SUNW,UltraSPARC-IIii' support will be removed in a future release of Solaris.
SunOS Release 5.11 Version snv_175 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
WARNING: i2c_0 failed to add interrupt.
WARNING: i2c_0 operating in POLL MODE only

Hardware watchdog enabled
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-08-26 23:19:57-- http://10.134.125.136:5555/export/auto_install/175s//solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2011-08-26 23:19:57 ERROR 404: Not Found.

```

Could not obtain http://10.134.125.136:5555/export/auto\_install/175s//solaris.zlib from install server  
Please verify that the install server is correctly configured and reachable from the client  
Requesting System Maintenance Mode

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 대상 디렉토리의 내용을 확인합니다. WAN 부트 구성을 확인합니다.

## Automated Installer 사용 안함

클라이언트 시스템에서 Oracle Solaris OS를 설치할 때 설치를 시작하려면 다음과 같이 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. 다음 메시지가 표시됩니다.

```

Auto-installer disabled. Enable the auto-installer service
by running the following command:
svcadm enable svc:/application/auto-installer:default

```

자동 설치를 시작하려면 로그인하여 메시지에 표시된 대로 설치 서비스를 사용으로 설정하거나, `install` 인수로 위에 표시된 명령을 사용하여 시스템을 재부트할 수 있습니다.

## x86 네트워크 부팅 오류 및 가능한 원인

이 섹션은 네트워크를 통해 x86 클라이언트를 부트할 때 볼 수 있는 오류/문제와 가능한 원인을 설명합니다.

- 197 페이지 “DHCP 또는 프록시 DHCP 제공이 수신되지 않음”
- 197 페이지 “TFTP 오류 또는 GATEWAY 메시지 후에 시스템 중단”
- 198 페이지 “GRUB 메뉴 항목을 선택한 후에 시스템 중단”
- 198 페이지 “HTTP 요청 전송 결과에 403 사용 권한 없음 또는 404 찾을 수 없음”
- 198 페이지 “Automated Installer 사용 안함”

### DHCP 또는 프록시 DHCP 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 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 AI 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 AI 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.

설치 서버에서 이 클라이언트의 `menu.lst` 파일이 유효한 부트 아카이브를 가리키는 지 여부를 확인합니다. 서버 이미지의 부트 디렉토리는 `df -k`에서 발췌한 이 샘플에 표시된 대로 `/etc/netboot` 디렉토리 아래에 마운트된 루프백이어야 합니다.

```
/install/images/x86_snv175 \
28046887      8432439      19614448      31%      /etc/netboot/x86_snv175
```

`installadm create-service` 명령에서 사용한 대상 디렉토리의 이름을 알고 있으면 해당 정보를 사용하여 대상 디렉토리가 마운트되는지 여부를 확인할 수 있습니다.

## HTTP 요청 전송 결과에 403 사용 권한 없음 또는 404 찾을 수 없음

설치 서버에서 설치 프로그램 중 하나에 액세스할 수 없거나 `/etc/netboot` 아래의 `menu.lst` 파일에 지정된 위치에 존재하지 않으면 클라이언트가 부트할 수 있지만 해당 파일을 다운로드할 수 없습니다. 어떤 파일이 문제를 일으키는 지 나타내는 오류 메시지가 표시됩니다. 예를 들어, x86 클라이언트의 다음 출력에서 `solaris.zlib` 파일이 지정된 위치에 존재하지 않습니다.

```
SunOS Release 5.11 Version snv_175 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-08-18 20:02:26-- http://10.134.125.136:5555//export/auto_install/s11-x86/solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2011-08-18 20:02:26 ERROR 404: Not Found.
```

Could not obtain http://10.134.125.136:5555//export/auto\_install/s11-x86/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.

## Automated Installer 사용 안함

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 11/11 Text Installer and command line
Oracle Solaris 11 11/11 Automated Install
```

첫번째 GRUB 메뉴 항목을 선택하거나 프롬프트의 시간 초과가 허용된 경우 시스템이 자동 설치 부트 이미지로 부트하지만 설치가 시작되지 않습니다. 다음과 같은 메시지가 표시됩니다.

```
Auto-installer disabled. Enable the auto-installer service
by running the following command:
svcadm enable svc:/application/auto-installer:default
```

자동 설치를 시작하려면 로그인하여 메시지에 표시된 대로 설치 서비스를 사용으로 설정하거나, 시스템을 재부트하고 두번째 메뉴 항목을 선택할 수 있습니다.

## SPARC 및 x86 오류 메시지

다음 오류는 SPARC 및 x86 설치 양쪽에 공통됩니다.

- 199 페이지 “자동 설치 실패 메시지”
- 199 페이지 “유효한 패키지 서버에 연결할 수 없음”
- 200 페이지 “패키지를 찾을 수 없음”

## 자동 설치 실패 메시지

설치 중 실패하면 다음 메시지가 표시됩니다.

```
Automated Installation failed. Please refer to /system/volatile/install_log file
for details
Apr 9 14:28:09 solaris svc.startd[7]: application/auto-installer:default
failed fatally: transitioned to maintenance (see 'svcs -xv' for details)
```

## 유효한 패키지 서버에 연결할 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 Checkpoint execution error:
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 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 31 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 31 15:54:47 UTC 2011
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://sun.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.

- ping(1M)을 사용하여 실패한 클라이언트 시스템에서 패키지 서버에 연결해 봅니다.
- DNS를 사용 중인 경우 DNS가 AI 클라이언트에 올바르게 구성되었는지 확인합니다.  
[193 페이지 “DNS 확인”](#)을 참조하십시오.
- 로컬 저장소를 사용 중인 경우 저장소에서 모든 클라이언트에 액세스 가능한지 확인합니다. [Oracle Solaris 11 패키지 저장소 복사 및 만들기의 3 장, “저장소에 대한 액세스 제공”](#)을 참조하십시오.
- AI 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

AI 매니페스트에 지정된 패키지 중 하나를 IPS 저장소에서 찾을 수 없는 경우 디스크에 패키지를 설치하기 전에 설치 프로그램을 실패합니다. 다음 예에서 설치 프로그램이 IPS 저장소에서 entity 패키지를 찾을 수 없습니다. 다음 출력은 콘솔에 표시되는 내용의 예입니다.

```
14:04:02 Failed Checkpoints:
14:04:02
14:04:02 ips
```



```

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:/entirity
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:/entirity
```

문제의 패키지가 유효한 패키지인지 여부를 확인합니다. 이 패키지를 서로 다른 IPS 저장소에서 사용할 수 있는 경우 다른 publisher 요소를 source 요소에 추가하여 AI 매니페스트에 해당 IPS 저장소를 추가합니다.

## 설치 시작 없이 설치 환경 부트

다음 방식 중 하나를 사용하여 자동 설치를 시작하지 않고 설치 환경을 부트합니다. 클라이언트를 부트할 때 202 페이지 “설치 시작 없이 부트 후에 설치 시작”에 표시된 대로 메뉴가 표시됩니다. 이 메뉴를 사용하여 시스템을 조사하거나 설치합니다.

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 11/11 Text Installer and command line
Oracle Solaris 11 11/11 Automated Install

```

기본 항목인 "Text Installer and command line"은 핸즈프리 자동 설치를 시작하지 않고 이미지를 부트합니다.

항목의 커널 줄에 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 줄에서 `install=true` 부트 등록 정보를 제거합니다.

일반적으로 x86 설치의 경우 `install=true` 부트 등록 정보가 부트 중인 GRUB 항목의 커널 줄에 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 부트 항목의 커널 줄을 편집하고 등록 정보를 제거합니다.

## 설치 시작 없이 부트 후에 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 sun)
- 5 Reboot

이 메뉴에는 기본 선택이 없습니다.

셸을 열려면 옵션 3을 선택합니다.

다음 명령을 사용하여 자동 설치를 시작합니다.

```
# svcadm enable manifest-locator:default
# svcadm enable svc:/application/auto-installer:default
```