사용자 정의 Oracle® Solaris 11 설치 이미지 만들기

부품 번호: E25772
2011년 11월 E25772–01

ORACLE

Copyright © 2008, 2011, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이센스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이센스 계약서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이센스, 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 주기 바랍니다.

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

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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, 라이센스 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 라이센스에 의거하여 사용되며SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록상표입니다.

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

120410@25097

목차

머리말 5

1 사용자 정의 설치 이미지 만들기 개요 9
배포판 구성자 정보 9
Oracle Solaris 이미지 유형 10
이미지 만들기 프로세스 11
SPARC 및 x86 아카이브 차이점 11

2 사용자 정의 설치 이미지 설계 13
이미지 빌드를 위한 시스템 요구 사항 13
이미지 사용자 정의 14
샘플 매니페스트 14
▼ 사용자 정의 이미지를 만들고 빌드하는 방법 15
매니페스트 컨텐츠 수정 15
사용자 정의 스크립트 만들기 및 사용 22

3 이미지 빌드 25
`distro_const` 명령 25
▼ 이미지를 한 단계로 빌드하는 방법 26
▼ 이미지를 여러 단계로 빌드하는 방법 26

머리말

사용자 정의 Oracle Solaris 11 설치 이미지 만들기는 Oracle Solaris DC(배포판 구성자) 도구를 사용하여 사용자 정의 Oracle Solaris 설치 이미지를 빌드하기 위한 지침을 제공합니다.

관련 정보

Oracle Solaris 11 시스템는 다음 방법을 사용하여 Oracle Solaris OS(운영 체제)를 설치하고 구성하기 위한 지침을 제공합니다.

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

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

Oracle Solaris 관리: 일반 작업의 6 장, "서비스 관리(개요)"에서는 Oracle Solaris SMF(Service Management Facility) 기능에 대해 설명합니다. SMF 프로파일을 사용하여 시스템을 구성할 수 있습니다.

pkg(5) 매뉴얼 페이지에서는 설치할 소프트웨어 패키지를 저장하고 검색할 수 있도록 해주는 Oracle Solaris IPS(Image Packaging System) 기능에 대해 설명합니다. pkg(1) 매뉴얼 페이지에서는 IPS 패키지를 설치하는 방법에 대해 설명합니다.

Oracle Solaris 11 시스템을 관리하는 방법은 Oracle Solaris 11 시스템 관리 설명서를 참조하십시오.

Oracle Solaris 10 JumpStart에서 Oracle Solaris 11 자동 설치 프로그램으로 전환에서는 둘 다 Oracle Solaris의 자동 설치 기능인 JumpStart에서 AI로 마이그레이션하는 데 도움이 되는 정보를 제공합니다.

Oracle Support에 액세스

Oracle 고객은 My Oracle Support를 통해 전자 지원에 액세스할 수 있습니다. 자세한 내용은 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info를 참조하거나, 청각 장애가 있는 경우 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs를 방문하십시오.

활자체 규약

다음 표는 이 책에서 사용되는 활자체 규약에 대해 설명합니다.

표 P–1 활자체 규약

활자체 또는 기호	의미	예제
`AaBbCc123`	명령 및 파일, 디렉토리 이름; 컴퓨터 화면에 출력되는 내용입니다.	`.login` 파일을 편집하십시오. 모든 파일 목록을 보려면 `ls -a` 명령을 사용하십시오. `machine_name% you have mail.`
`AaBbCc123`	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	`machine_name%` **`su`** `Password:`
AaBbCc123	새로 나오는 용어, 강조 표시할 용어입니다. 명령줄 변수를 실제 이름이나 값으로 바꾸십시오.	`rm` *filename* 명령을 사용하여 파일을 제거합니다.
AaBbCc123	책 제목, 장, 절	**사용자 설명서**의 6장을 읽으십시오. **캐시**는 로컬로 저장된 복사본입니다. 파일을 저장하면 **안 됩니다**. **주:** 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

명령 예의 셸 프롬프트

다음 표에는 Oracle Solaris OS에 포함된 셸의 기본 UNIX 시스템 프롬프트 및 수퍼유저 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	`$`
수퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	`#`
C 셸	`machine_name%`
수퍼유저용 C 셸	`machine_name#`

◆ ◆ ◆ 1 장

1 사용자 정의 설치 이미지 만들기 개요

시스템 관리자와 응용 프로그램 개발자는 배포판 구성자 도구를 사용하여 사용자 정의 Oracle Solaris 설치 이미지를 빌드할 수 있습니다.

- 사용자 정의 설치 이미지를 만든 적이 없는 경우 9 페이지 "배포판 구성자 정보"를 읽어 보십시오.
- 사용자 정의 이미지를 빌드할 준비가 된 경우 13 페이지 "이미지 빌드를 위한 시스템 요구 사항"으로 이동하십시오.

배포판 구성자 정보

배포판 구성자는 사전 구성된 Oracle Solaris 이미지를 빌드하는 명령줄 도구입니다. 이 도구는 XML 매니페스트 파일을 입력으로 사용하며 매니페스트 파일에 지정된 매개변수를 기반으로 이미지를 빌드합니다.

배포판 구성자는 ISO 이미지를 빌드할 수 있는데, 이 이미지는 ISO(International Organization for Standardization)에서 정의한 형식의 광 디스크 아카이브 파일(디스크 이미지라고도 함)입니다. 또한 생성된 ISO 이미지를 기반으로 하는 USB 이미지도 만들 수 있습니다. 그러나 USB 이미지는 ISO 이미지와 달리 x86 시스템에서 만들어 사용할 수 있습니다.

다음 사항에 유의하십시오.

- 이미지 구성에 따라 ISO 또는 USB 이미지는 부트 가능할 수 있습니다.
- ISO 이미지와 USB 이미지 모두 시스템에 설치하거나 라이브 매체 환경에서 실행할 수 있습니다.
- ISO 이미지는 CD 또는 DVD로 구울 수 있습니다.
- USB 이미지는 플래시 드라이브로 복사할 수 있습니다.
- ISO 이미지와 USB 이미지 모두 인터넷에 게시할 수 있습니다.

다양한 유형의 플래시 메모리 장치가 Oracle Solaris 릴리스에서 제공하는 드라이버 지원을 제공하는 경우 배포판 구성자는 이러한 장치에서 작동하는 USB 이미지를 만듭니다. `usbcopy` 유틸리티는 USB 이미지를 USB 플래시 드라이브로 복사하는 데 사용됩니다. 이 `usbcopy` 유틸리티는 `distribution-constructor` 패키지에서 제공합니다.

Oracle Solaris 이미지 유형

배포판 구성자를 사용하여 다음과 같은 유형의 Oracle Solaris 이미지를 만들 수 있습니다.

- **Oracle Solaris x86 LiveCD** – Oracle Solaris 릴리스로 배포되는 LiveCD 이미지와 비교되는 x86 ISO 이미지를 만들 수 있습니다. 패키지를 추가하거나 제거하여 이 ISO 이미지의 컨텐츠를 수정할 수도 있습니다. 그 결과로 발생하는 부트된 환경에서 사용자 정의 ISO 이미지 또는 USB 이미지를 만들기 위해 기본 설정을 수정할 수 있습니다.

  주 – LiveCD 이미지에 포함된 패키지의 크기에 따라 LiveCD 이미지가 대신 LiveDVD 이미지가 될 수도 있습니다.

  LiveCD 설치에 대한 자세한 내용은 **Oracle Solaris 11 시스템**의 3 장, “LiveCD 사용”를 참조하십시오.

- **Oracle Solaris x86 또는 SPARC 텍스트 설치 이미지** – Oracle Solaris 운영 체제의 텍스트 설치를 수행하는 데 사용할 수 있는 SPARC 또는 x86 ISO 이미지를 만들 수 있습니다. 텍스트 설치 프로그램은 그래픽 카드가 필요하지 않은 시스템에서 사용할 수 있습니다.

  주 – 텍스트 설치의 경우 LiveCD에서 설치할 때 포함된 소프트웨어 패키지가 모두 설치되지는 **않습니다**. 예를 들어 텍스트 설치 프로그램은 데스크탑을 설치하지 않습니다. 텍스트 설치 후 `solaris-desktop` 패키지와 같은 추가 패키지를 추가할 수 있습니다.

  텍스트 설치에 대한 자세한 내용은 **Oracle Solaris 11 시스템**의 4 장, “텍스트 설치 프로그램 사용”를 참조하십시오.

- **자동 설치를 위한 x86 또는 SPARC ISO 이미지** – Oracle Solaris 릴리스에는 Automated Installer 도구가 포함되어 있습니다. AI(Automated Installer)는 네트워크를 통해 하나 이상의 SPARC 및 x86 시스템에 Oracle Solaris OS를 자동으로 설치하는 데 사용됩니다. 설치마다 아키텍처, 설치된 패키지, 디스크 용량 및 기타 매개변수가 다를 수 있습니다. Automated Installer는 SPARC 또는 x86 AI ISO 이미지를 사용하여 Oracle Solaris OS를 클라이언트 시스템에 설치할 수 있습니다. 배포판 구성자를 사용하면 SPARC 클라이언트에 Oracle Solaris OS를 설치하는 데 사용할 수 있는 SPARC AI ISO 이미지를 만들거나, x86 클라이언트에 Oracle Solaris OS를 설치하는 데 사용할 수 있는 x86 AI ISO 이미지를 만들 수 있습니다.

Automated Installer에 대한 자세한 내용은 **Oracle Solaris 11 시스템**의 제III부, "설치 서버를 사용하여 설치"를 참조하십시오.

이미지 만들기 프로세스

배포판 구성자는 **매니페스트 파일**이라고 하는 XML 파일에 지정된 설정을 기반으로 이미지를 만듭니다. 매니페스트 파일에는 배포판 구성자를 사용하여 만드는 ISO 이미지의 컨텐츠 및 매개변수에 대한 지정 사항이 포함되어 있습니다. 배포판 구성자에는 사용자 정의 LiveCD, x86/SPARC AI ISO 이미지 또는 x86/SPARC 텍스트 설치 이미지를 만드는 데 사용할 수 있는 샘플 매니페스트가 포함되어 있습니다.

각 매니페스트 파일의 모든 필드는 필요한 이미지 유형을 만드는 미리 설정된 기본값을 제공합니다. 매니페스트 파일의 필드를 편집하여 생성되는 이미지를 추가로 사용자 정의할 수 있습니다. 예를 들어 매니페스트의 대상 요소를 편집하여 이미지를 구성할 수 있는 빌드 영역에 대해 다른 위치를 지정할 수 있습니다. 또한 지정된 게시자를 확인한 다음 이미지를 빌드하는 데 필요한 패키지를 다운로드하도록 사용 중인 시스템에서 해당 게시자에게 연결할 수 있는지 확인할 수 있습니다. 필요한 경우 소프트웨어 이름 요소를 편집하여 다른 게시자 저장소 위치를 지정할 수 있습니다. 지침은 14 페이지 "이미지 사용자 정의"를 참조하십시오.

사용자 정의 스크립트를 만들어 설치 이미지를 수정할 수도 있습니다. 그런 다음 매니페스트 파일에 체크포인트를 추가하여 이러한 스크립트를 실행할 수 있습니다. 자세한 내용은 22 페이지 "사용자 정의 스크립트 만들기 및 사용"을 참조하십시오.

배포판 구성자 패키지에는 매니페스트 지정 사항을 해석하여 이미지를 만드는 명령줄 유틸리티인 `distro_const` 명령도 포함되어 있습니다. 자신의 요구 사항에 적합하도록 매니페스트 파일에서 이미지 블루프린트 편집을 완료한 후에는 `distro_const` 명령을 실행하여 이미지를 빌드할 수 있습니다. 자세한 내용은 3 장, "이미지 빌드"를 참조하십시오.

`distro_const` 명령에 제공된 옵션으로 이미지 생성 프로세스의 여러 단계에서 빌드 프로세스를 중지 및 다시 시작하여 빌드 중인 이미지를 검사하고 디버그할 수 있습니다. 빌드 프로세스 중 이와 같이 중지 및 다시 시작하는 프로세스를 **체크포인팅**이라고 합니다. 체크포인팅은 선택적입니다. 기본 체크포인트는 각 매니페스트 파일에 지정됩니다. 지침은 26 페이지 "이미지를 여러 단계로 빌드하는 방법" 또는 `distro_const`(1M) 매뉴얼 페이지를 참조하십시오.

SPARC 및 x86 아카이브 차이점

x86 이미지의 루트 아카이브는 SPARC 이미지의 루트 아카이브와 다릅니다. x86 이미지의 전체 루트 아카이브인 `boot_archive`는 `lzma`를 사용하여 압축된 UFS 파일 시스템입니다. SPARC 플랫폼에서는 이와 같은 방식의 전체 루트 아카이브 압축을 지원하지 않습니다. 대신 SPARC 루트 아카이브는 각 파일을 개별적으로 압축하는

DCFS를 사용합니다. 개별적으로 압축된 파일은 매니페스트에서 특정 처리가 필요합니다. 지침은 dc_manifest(4) 매뉴얼 페이지의 <boot_archive_contents> 필드를 참조하십시오.

◆ ◆ ◆ 2 장

2 사용자 정의 설치 이미지 설계

시스템 요구 사항을 검토한 다음 이 장에 설명된 대로 사용자 정의 설치 이미지를 설계하십시오.

이미지 빌드를 위한 시스템 요구 사항

배포판 구성자를 사용하려면 시스템에 다음이 설정되어 있어야 합니다.

표 2–1 시스템 요구 사항

요구 사항	설명
디스크 공간	배포판 구성자 작업 공간으로 권장되는 최소 크기는 8GB입니다. 배포판 구성자를 사용할 수 있는 충분한 시스템 공간이 있는지 확인하십시오.
Oracle Solaris 릴리스	시스템에 Oracle Solaris OS(운영 체제)가 설치되어 있어야 합니다. 다음 사항을 고려하십시오. ■ 설치된 시스템에서 네트워크 액세스가 가능해야 합니다. 배포판 구성자는 네트워크를 통해 ISO 이미지에 대한 패키지를 검색하는 IPS(Image Packaging System) 저장소에 액세스합니다. 따라서 매니페스트 파일에 지정한 저장소에 대한 네트워크 액세스가 가능해야 합니다. ■ 배포판 구성자를 사용할 경우 SPARC 시스템에서는 SPARC 이미지만 만들 수 있고, x86 시스템에서는 x86 이미지만 만들 수 있습니다. ■ 시스템의 Oracle Solaris 릴리스 버전은 배포판 구성자에서 사용하는 이미지의 릴리스 버전과 동일해야 합니다. **주 –** 배포판 구성자를 실행하려면 루트 역할이어야 합니다.
필요한 패키지	배포판 구성자 응용 프로그램이 포함된 `distribution-constructor` 패키지

이미지 사용자 정의

배포판 구성자는 **매니페스트 파일**이라고 하는 XML 파일에 지정된 설정을 기반으로 이미지를 만듭니다. 매니페스트 파일에는 배포판 구성자를 사용하여 만드는 ISO 이미지의 컨텐츠 및 매개변수에 대한 지정 사항이 포함되어 있습니다. `distribution-constructor` 패키지는 사용자 정의 LiveCD, x86/SPARC AI ISO 이미지 또는 x86/SPARC 텍스트 설치 이미지를 만드는 데 사용할 수 있는 샘플 매니페스트를 제공합니다.

각 매니페스트 파일의 요소는 필요한 ISO 이미지 유형을 만드는 미리 설정된 기본값을 제공합니다. 매니페스트 파일의 이러한 미리 설정된 요소를 수동으로 편집하여 생성되는 이미지를 사용자 정의할 수 있습니다. 또한 사용자 정의 스크립트를 만들어 설치 이미지를 추가로 수정할 수 있습니다. 그런 다음 매니페스트 파일에서 새 스크립트를 참조하십시오.

샘플 매니페스트

`distribution-constructor` 패키지에서 제공하는 샘플 매니페스트 파일은 다음과 같습니다.

표 2–2 샘플 매니페스트

매니페스트 유형	매니페스트 위치	설명
x86 LiveCD ISO 이미지	`/usr/share/distro_const/dc_livecd.xml`	Oracle Solaris LiveCD와 비교되는 ISO 이미지를 만드는 데 사용됩니다.
x86 텍스트 설치 이미지	`/usr/share/distro_const/dc_text_x86.xml`	x86 Oracle Solaris 운영 체제의 텍스트 설치를 수행하는 데 사용할 수 있는 ISO 이미지를 만드는 데 사용됩니다.
SPARC 텍스트 설치 이미지	`/usr/share/distro_const/dc_text_sparc.xml`	SPARC Oracle Solaris 운영 체제의 텍스트 설치를 수행하는 데 사용할 수 있는 ISO 이미지를 만드는 데 사용됩니다.
x86 AI ISO 이미지	`/usr/share/distro_const/dc_ai_x86.xml`	x86 클라이언트에 Oracle Solaris OS를 자동으로 설치하기 위한 x86 AI ISO 이미지를 만드는 데 사용됩니다.
SPARC AI ISO 이미지	`/usr/share/distro_const/dc_ai_sparc.xml`	SPARC 클라이언트에 Oracle Solaris OS를 자동으로 설치하기 위한 SPARC AI ISO 이미지를 만드는 데 사용됩니다.

▼ 사용자 정의 이미지를 만들고 빌드하는 방법

1 **배포판 구성자 응용 프로그램 및 샘플 매니페스트가 포함된 `distribution-constructor` 패키지를 다운로드합니다.**

패키지 관리자 도구를 사용하여 필요한 패키지를 설치할 수 있습니다. 패키지 관리자는 Oracle Solaris 운영 체제의 데스크탑에 있는 메뉴 표시줄에서 사용할 수 있습니다. 메뉴 표시줄에서 시스템>관리>패키지 관리자로 이동합니다.

또는 다음과 같은 IPS 명령을 사용하여 이 패키지를 설치합니다.

```
# pkg install distribution-constructor
```

2 **샘플 매니페스트 중 하나를 복사한 다음 새 파일 이름의 사용자 정의 매니페스트 파일을 만듭니다.**

`distro_const` 명령을 사용하여 이미지를 만들 경우 매니페스트 파일을 이름으로 참조합니다.

주 – 원본 매니페스트 파일과 기본 스크립트를 복사하기 전에 항상 이들을 백업하십시오.

---

3 **자신의 요구에 알맞도록 매니페스트 요소를 편집합니다.**

예를 들어 매니페스트의 대상 요소를 편집하여 이미지를 구성할 수 있는 빌드 영역에 대해 다른 위치를 지정할 수 있습니다. 또한 게시자를 확인하여 이미지를 빌드하는 데 필요한 패키지를 다운로드하도록 시스템에서 해당 게시자에게 연결할 수 있는지 확인할 수 있습니다. 필요한 경우 소프트웨어 이름 요소를 편집하여 다른 게시지 ? 저장소 위치를 지정할 수 있습니다.

자세한 내용은 15 페이지 “매니페스트 컨텐츠 수정” 및 `dc_manifest`(4) 매뉴얼 페이지를 참조하십시오.

4 **(옵션) 사용자 정의 스크립트를 만들어 이미지를 추가로 수정합니다.**

새 스크립트를 만드는 경우 매니페스트 파일의 실행 섹션에서 스크립트 참조를 업데이트합니다.

지침은 22 페이지 “사용자 정의 스크립트 만들기 및 사용”을 참조하십시오.

5 **`distro_const` 유틸리티를 실행하여 이미지를 만듭니다.**

지침은 3 장, “이미지 빌드”를 참조하십시오.

매니페스트 컨텐츠 수정

각 매니페스트 파일의 모든 필드는 필요한 ISO 이미지 유형을 만드는 미리 설정된 기본값을 제공합니다. 매니페스트 파일의 이러한 미리 설정된 필드를 수동으로 편집하여 생성되는 이미지를 추가로 사용자 정의할 수 있습니다.

선택한 샘플 매니페스트에 따라 기본 요소는 다음과 같습니다.

표 2–3 매니페스트 요소

요소	설명
`<distro name="Oracle_Solaris_Text_X86" add_timestamp="false">`	선택적 시간 기록과 함께 이미지 이름을 지정합니다.
`<boot_mods>`	이미지에 대한 GRUB 메뉴 수정 사항을 지정합니다.
`<target>`	이미지가 빌드된 ZFS 빌드 데이터 세트를 정의합니다.
`<software name="transfer-ips-install" type="IPS">`	설치할 소프트웨어 패키지의 소스를 지정합니다.
`<software_data action="install">`	설치할 패키지를 나열합니다.
`<software_data action="uninstall">`	설치 제거할 패키지를 나열합니다.
`<software name="set-ips-attributes">`	설치가 완료된 후 IPS에 대해 다른 속성을 설정합니다.
`<software name="ba-init">`	부트 아카이브 컨텐츠를 지정합니다. **주의 –** 주의해서 수정하십시오. 부트 아카이브가 올바르지 않을 경우 설치된 시스템이 부트되지 않습니다.
`<execution stop_on_error="true">` `<checkpoint name="transfer-ips-install"/>`	빌드 체크포인트를 나열합니다.
`<configuration name="pre-pkg-img-mod" type="sysconf" source="/etc/svc/profile/generic_limited_net.xml">`	빌드 중 매체에 적용될 SMF 서비스를 지정합니다. **주의 –** 되도록이면 수정하지 마십시오.

이미지 제목 제공

다음 요소를 사용하여 빌드하려는 이미지의 사용자 정의 또는 기본 이름을 제공하십시오.

```
<distro name="Oracle_Solaris_Text_X86" add_timestamp="false">
```

일련의 이미지 빌드를 수행하고 증분 이미지를 보존하려는 경우, 시간 기록 변수를 “true”로 변경하십시오. 그러면 시간 기록이 자동으로 각 이미지의 이름에 추가됩니다.

HTTP 프록시를 지정해야 하는 경우 프록시 변수를 포함하는 `distro name` 요소의 주석 처리를 해제한 다음 프록시 위치를 입력하십시오.

부트 메뉴 수정

이 부트 요소는 이미지에 적용될 부트 메뉴 수정 사항을 지정합니다.

다음 예에서는 제목이 “boot1”인 특수화된 부트 메뉴가 이미지에 적용됩니다. 시간 초과 속성은 기본 부트 항목이 자동으로 활성화되기 이전 시간을 지정합니다.

```
<boot_mods title="boot1" timeout="5">
```

부트 메뉴 요소 내에서 새 항목마다 새 `boot_entry` 요소를 추가하여 개별 부트 메뉴 항목을 추가할 수 있습니다. 각 부트 항목에 대한 “start” 또는 “end”의 insert_at 속성 값에 따라 순차적으로 부트 메뉴에 항목이 추가됩니다.

주 – 새 항목은 기존 "with magnifier” 항목 앞에 추가하십시오.

개별 `boot_entry` 요소의 다음 예를 참조하십시오.

```
<boot_entry>
   <title_suffix>with screen reader</title_suffix>
   <kernel_args>-B assistive_tech=reader</kernel_args>
</boot_entry>
```

자세한 내용은 `dc_manifest`(4) 매뉴얼 페이지를 참조하십시오.

빌드 영역 지정

대상 요소는 사용자 정의할 수 있습니다. 이 요소는 빌드에 사용될 ZFS 빌드 데이터 세트를 정의합니다. 이 데이터 세트는 이미지가 생성되는 영역입니다. 유효한 데이터 세트 위치를 입력해야 합니다. 빌드로 시스템에 보존해야 하는 컨텐츠가 삭제되지 않는지 기본 빌드 영역을 확인해야 합니다. 필요한 경우 빌드 영역을 수정하십시오.

주 – 파일 시스템 이름에 zpool의 이름이 포함되어서는 안됩니다.

다음 예를 보십시오.

```
 <target>
   <logical>
     <zpool action="use_existing" name="rpool">
       <dataset>
         <filesystem name="dc/sample-dataset-location"
         action="preserve"/>
       </dataset>
     </zpool>
   </logical>
 </target>
```

게시자 지정

다음 요소는 배포판 구성자가 이미지를 빌드하기 위해 다운로드하고 사용할 패키지를 가져올 수 있는 게시자를 지정합니다.

```
<software name="transfer-ips-install">
```

이 섹션의 소스 요소에서 게시자 이름 및 원본 이름 요소를 편집하여 사용할 게시자 및 패키지 저장소의 위치를 지정하십시오. 게시자가 여러 개 나열될 수 있습니다. 배포판 구성자가 설치할 패키지를 찾으려고 하면 여기에 나열된 순서로 게시자가 검색됩니다.

게시자에 대한 미러를 지정해야 하는 경우 미러 이름 요소의 주석 처리를 해제한 다음 이 요소를 편집하십시오.

다음 예를 보십시오.

```
<source>
   <publisher name="publisher1">
      <origin name="http://example.oracle.com/primary-pub"/>
      <mirror name="mirror.example.com"/>
   </publisher>
   <publisher name="publisher2">
       <origin name="http://example2.com/dev/solaris"></origin>
   </publisher>
   <publisher name="publisher3.org">
       <origin name="http://example3.com/dev"></origin>
    </publisher>
</source>
```

게시자 사용에 대한 자세한 내용은 **Oracle Solaris 11 소프트웨어 패키지 추가 및 업데이트**를 참조하십시오.

설치할 패키지 나열

`install` 속성을 포함하는 `software_data` 요소는 사용 중인 매니페스트에 따라 특정 유형의 이미지를 빌드하기 위해 설치할 패키지 세트를 나열합니다. 예를 들어 `dc_livecd.xml` 매니페스트는 LiveCD 이미지를 빌드하는 데 필요한 패키지를 나열합니다. 각 이름 태그는 하나의 패키지 이름을 나열하거나 여러 개의 패키지를 포함하는 그룹 패키지의 이름을 나열합니다.

```
<software_data action="install">
   <name>pkg:/group/system/solaris-desktop</name>
   <name>pkg:/system/install/gui-install</name>
   <name>pkg:/system/install/media/internal</name>
</software_data>
```

이미지에 추가할 패키지가 있는 경우 각 패키지에 대해 이름 태그를 추가하여 패키지 이름을 추가하십시오.

기본적으로 지정된 저장소에서 사용 가능한 가장 최근 패키지 버전이 나열됩니다. 다른 버전이 필요할 경우 다음 형식을 사용하여 패키지 참조에 버전 번호를 추가하십시오.

```
<name>pkg:/group/system/solaris-desktop@0.5.11-0.build#</name>
```

주 – 설치된 버전과 충돌하는 다른 패키지가 있을 경우 특정 버전이 지정된 패키지가 설치되지 않을 수 있습니다. 자세한 내용은 pkg(5) 매뉴얼 페이지를 참조하십시오.

예 2–1 패키지 및 추가 게시지 추가

이 예에서는 또 다른 게시자인 mypublisher가 지정되었습니다. 또한 추가 패키지 mypackage1 및 mypackage2도 지정되었습니다.

빌드 프로세스 중 게시자는 나열된 순서로 확인됩니다. 첫번째 게시자에서 패키지를 찾을 수 없는 경우 다음 게시자에서 지정된 패키지를 검색합니다.

```
<software name="transfer-ips-install" type="IPS">
   <destination>
      <xi:include xmlns:xi="http://www.w3.org/2003/XInclude"
            href="/usr/share/distro_const/lang_facets.xml"/>
   </destination>
   <source>
      <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
      </publisher>
      <publisher name="mypublisher">
         <origin name="http://mypublisher.company.com"/>
      </publisher>
   </source>
   <software_data action="install">
        <name>pkg:/group/system/solaris-large-server</name>
        <name>pkg:/system/install/text-install</name>
        <name>pkg:/system/install/media/internal</name>
        <name>pkg:/mypackage1</name>
        <name>pkg:/mypackage2</name>
   </software_data>
</software>
```

설치 제거할 패키지 나열

uninstall 속성을 포함하는 software_data 요소를 사용하여 개별 패키지를 제거하거나 그룹 패키지 정의를 제거할 수 있습니다.

다음 예에서 solaris-desktop은 여러 개의 개별 패키지를 포함하는 그룹 패키지의 이름입니다.

```
<software_data action="uninstall">
   <name>pkg:/group/system/solaris-desktop</name>
</software_data>
```

그룹 패키지는 제거할 수 있습니다. 그룹 패키지를 제거하면 그룹 정의만 실제로 제거됩니다. 해당 그룹의 일부로 이전에 설치되었던 개별 패키지는 제거되지 않습니다. 그러나 그룹 패키지를 제거하지 않고도 이러한 개별 패키지를 제거할 수 있습니다. 그룹 패키지를 보존하는 것이 진행 중인 참조에 유용할 수 있습니다. 또한 이름 태그를 사용하여 개별 패키지를 제거할 수도 있습니다. 설치 제거 섹션의 끝에 제거할 추가 패키지를 추가하십시오.

설치된 시스템의 게시자 지정

배포판 구성자를 사용하여 만든 이미지로 시스템이 설치된 후 시스템에 영향을 주는 요소는 다음과 같습니다.

```
<software name="set-ips-attributes">
```

게시자 이름과 선택적 미러 이름을 제공하여 설치된 시스템이 다운로드 및 설치할 추가 패키지에 액세스할 수 있는 위치를 지정하십시오.

이 요소에 IPS 속성을 설정할 수도 있습니다. pkg(1) 매뉴얼 페이지 IPS 등록 정보를 참조하십시오.

빌드 체크포인트 설정

매니페스트의 실행 요소는 이미지 구성 프로세스 중에 실행되는 일련의 체크포인트를 나열합니다. 체크포인트는 이 섹션에 나열된 순서로 실행됩니다. 기본 설치 이미지를 빌드하는 데 필요한 기본 체크포인트는 각 매니페스트 파일에 포함되어 있습니다.

각 체크포인트 이름 태그에는 체크포인트 스크립트의 위치를 지정하는 mod-path 속성이 포함되어 있습니다.

일부 기본 체크포인트 태그에는 기본값이 제공된 인수가 포함되어 있습니다. dc_ai_sparc.xml 샘플 매니페스트의 다음 체크포인트 예는 이미지 빌드에 대한 부트 아카이브를 만들며 해당 작업을 수행할 스크립트를 가리킵니다. 체크포인트 예에도 각 인수에 대해 특정 값이 제공된 인수 필드가 포함되어 있습니다.

```
<checkpoint name="ba-arch"
    desc="Boot Archive Archival"
    mod_path="solaris_install/distro_const/checkpoints/
    boot_archive_archive"
    checkpoint_class="BootArchiveArchive">
    <kwargs>
        <arg name="size_pad">0</arg>
        <arg name="bytes_per_inode">0</arg>
        <arglist name="uncompressed_files">
             <argitem>etc/svc/repository.db</argitem>
             <argitem>etc/name_to_major</argitem>
             <argitem>etc/minor_perm</argitem>
             <argitem>etc/driver_aliases</argitem>
             <argitem>etc/driver_classes</argitem>
             <argitem>etc/path_to_inst</argitem>
             <argitem>etc/default/init</argitem>
             <argitem>etc/nsswitch.conf</argitem>
             <argitem>etc/passwd</argitem>
             <argitem>etc/shadow</argitem>
             <argitem>etc/inet/hosts</argitem>
        </arglist>
     </kwargs>
   </checkpoint>
```

이 예에 표시된 것과 같이, <kwargs> 요소에는 빌드 중 체크포인트에 전달해야 하는 키워드 인수가 있습니다. <kwargs> 요소 내에는 체크포인트에 전달할 개별 키워드를

지정하는 데 사용할 수 있는 <arg name> 요소가 있습니다. 또한 <arglist> 요소에는 체크포인트에 전달할 여러 <argitem> 값 목록이 포함되어 있습니다. 이 예의 경우 <arglist> 요소에서 압축되지 않은 파일 목록이 포함되어 있습니다.

각 <kargs> 목록 항목은 큰 따옴표로 묶여 있습니다. 큰 따옴표를 사용하지 않았거나 전체 문자열이 큰 따옴표로 묶인 경우, 공백과 줄바꿈을 포함한 전체 문자열이 하나의 인수로 해석됩니다. 인수를 쉼표로 구분하지 마십시오.

이미지 빌드 중에 사용할 사용자 정의 스크립트를 만든 경우 스크립트 위치를 가리키는 체크포인트 요소를 추가해야 합니다. 사용자 정의 스크립트에 대한 체크포인트에는 사용자 정의 스크립트 위치를 가리키는 <args> 요소만 필요합니다. 자세한 내용 및 예제는 22 페이지 "사용자 정의 스크립트 만들기 및 사용"을 참조하십시오.

distro_const 명령 옵션을 사용하면 특정 체크포인트에서 빌드 프로세스의 일시 중지 및 다시 시작을 제어할 수 있습니다. 26 페이지 "이미지를 여러 단계로 빌드하는 방법"을 참조하십시오.

예 2–2 SVR4 패키지 추가

이 예에서는 새 체크포인트가 매니페스트에 추가되었습니다. 이 새 체크포인트는 이미지에 추가할 SVR4 패키지 및 패키지의 위치를 나열합니다. 그런 다음 이 새 체크포인트는 실행 섹션에서 참조됩니다.

먼저 새 software 요소를 추가하여 새 체크포인트가 생성됩니다. 이 체크포인트는 SVR4를 소프트웨어 유형으로 지정합니다. 이 체크포인트는 패키지를 찾고 설치하는 위치입니다.

또한 설치할 특정 SVR4 패키지는 software_data 요소에 나열됩니다.

```
<software name=transfer-svr4-install type="SVR4">
   <destination>
       <dir path={PKG_IMAGE_PATH}/>
   </destination>
   <source>
    <dir path="/path/to/packages"/>
   </source>
   <software_data action="install">
      <name>SUNWpackage1</name>
      <name>SUNWpackage2</name>
   </software_data>
</software>
```

체크포인트에 포함된 경우 {PKG_IMAGE_PATH} 및 {BOOT_ARCHIVE} 값은 distro_const 유틸리티를 통해 각각 <ZFS Dataset>/build_data/pkg_image 및 <ZFS Dataset>/build_data/boot_archive로 대체됩니다. 이 예에서 SVR4 패키지는 <ZFS Dataset>/build_data/pkg_image 에 설치됩니다.

끝으로 새 체크포인트가 실행 섹션에서 참조됩니다.

예 2–2 SVR4 패키지 추가 *(계속)*

```
<execution stop_on_error="true">
   <checkpoint name="transfer-ips-install"
       desc="Transfer pkg contents from IPS"
       mod_path="solaris_install/transfer/ips"
       checkpoint_class="TransferIPS"/>
   <checkpoint name="set-ips-attributes"
       desc="Set post-install IPS attributes"
       mod_path="solaris_install/transfer/ips"
       checkpoint_class="TransferIPS"/>
   <checkpoint name="transfer-svr4-install"
      desc="Transfer pkg contents from SVR4 packages"
      mod_path="solaris_install/transfer/svr4"
      checkpoint_class="TransferSVR4"/>
```

소프트웨어 이름은 체크포인트 이름과 일치해야 합니다. 이 예에서는 둘 다 “transfer-svr4–install”입니다.

사용자 정의 스크립트 만들기 및 사용

배포판 구성자를 사용하여 빌드하려는 이미지의 유형을 기반으로 사용자 정의하는 데 사용할 수 있는 추가 스크립트를 지정할 수 있습니다. 매니페스트 파일은 스크립트를 가리키며, 스크립트는 일반 이미지를 매체별 배포로 변환합니다. 이러한 스크립트는 매니페스트 파일의 실행 섹션에서 참조됩니다. 원하는 수의 사용자 정의 스크립트 체크포인트를 지정할 수 없습니다.

주 – 스크립트에 대한 지원은 응용 프로그램 패키지와 함께 제공되는 수정되지 않은 기본 스크립트로 제한됩니다. 이러한 스크립트를 사용자 정의하도록 선택한 경우 먼저 원본 스크립트를 백업하십시오.

▼ 사용자 정의 스크립트를 만들고 사용하는 방법

시작하기 전에 사용자 정의 스크립트를 만드는 경우 다음 사항에 유의하십시오.

- 스크립트는 Python 프로그램, 셸 스크립트 또는 이진 파일일 수 있습니다.
- 스크립트는 매니페스트 파일의 실행 섹션에 나열된 순서로 실행됩니다.
- 스크립트(셸 및 python 모듈 모두) 내에서 실행된 명령의 표준 출력(`stdout`) 및 오류 출력 (`stderr`)은 완료되거나 시도된 빌드에 대해 보고하는 로그 파일에 캡처됩니다.

1 새 스크립트를 만듭니다.

2 새 스크립트를 홈 디렉토리 또는 시스템이나 네트워크의 특정 위치에 추가합니다.
루트 역할을 수행하는 사용자가 이 스크립트를 실행할 수 있는지 확인합니다.

3 **해당 매니페스트 파일의 실행 섹션에 체크포인트를 추가하여 새 스크립트를 참조합니다.**

스크립트의 전체 경로를 지정해야 합니다. 체크포인트는 매니페스트의 실행 섹션에 나열된 순서로 실행됩니다.

매니페스트 파일의 실행 섹션에 새 스크립트에 대한 참조를 추가할 경우 이 스크립트가 해당 작업을 수행하기 전이나 후에 이미지 빌드를 일시 중지하는 데 사용할 수 있는 체크포인트 이름을 지정해야 합니다. 선택적으로 체크포인트 이름과 연관된 사용자 정의 메시지를 포함시킬 수 있습니다. 이 메시지를 생략할 경우 스크립트의 경로가 기본 체크포인트 메시지로 사용됩니다. 체크포인트 메시지는 빌드 프로세스 중 체크포인트가 실행될 때 표시됩니다.

주 – 체크포인트 이름으로 숫자 대신 의미 있는 이름을 사용하십시오. 새 스크립트가 추가되면 해당 새 스크립트에 대한 새 체크포인트가 숫자로 지정된 체크포인트 순서를 방해합니다.

다음 체크포인트 예는 사용자 정의 스크립트인 “my-script”를 참조합니다.

```
<checkpoint name="my-script"
        desc="my new script"
        mod_path="solaris_install/distro_const/checkpoints/custom_script"
        checkpoint_class="CustomScript">
        <args>/tmp/myscript.sh</args>
    </checkpoint>
```

4 **(옵션) 다음과 같이 체크포인트의 일부로 빌드 프로세스를 지정합니다.**

여기서 `{PKG_IMAGE_PATH}`는 인수 섹션의 빌드 매개변수로 지정되었습니다.

```
<checkpoint name="my-script"
        desc="my new script"
        mod_path="solaris_install/distro_const/checkpoints/my_script"
        checkpoint_class="CustomScript">
        <args>/tmp/myscript.sh {PKG_IMAGE_PATH}</args>
    </checkpoint>
```

체크포인트에 포함된 경우 `{PKG_IMAGE_PATH}` 및 `{BOOT_ARCHIVE}` 값은 `distro_const` 유틸리티를 통해 각각 `<ZFS Dataset>/build_data/pkg_image` 및 `<ZFS Dataset>/build_data/boot_archive`로 대체됩니다.

5 **이미지를 빌드합니다.**

이미지는 한 단계로 빌드할 수 있습니다. 또는 체크포인트에서 빌드를 중지했다가 다시 시작하여 빌드 상태를 확인할 수 있습니다.

지침은 3 장, “이미지 빌드”를 참조하십시오.

6 **(옵션) 빌드가 완료되면 빌드 프로세스에 대해 보고하는 로그 파일을 볼 수 있습니다.**

빌드 출력에 로그 파일의 위치가 표시됩니다.

◆ ◆ ◆ 3 장

3 이미지 빌드

사용하려는 매니페스트 파일을 설정하고 원하는 경우 finalizer 스크립트를 사용자 정의했으면 `distro_const` 명령을 실행하여 이미지를 빌드할 준비가 된 것입니다.

`distro_const` 명령을 사용하여 다음 방법 중 하나로 이미지를 빌드할 수 있습니다.

- 한 단계로
- 필요에 따라 빌드를 일시 중지했다가 다시 시작하여 빌드 프로세스 중 이미지 컨텐츠 검사 및 스크립트 디버그

distro_const 명령

`distro_const` 명령의 전체 구문은 다음과 같습니다.

`Syntax: distro_const build [-v] [-r` *checkpoint_name*`] [-p` *checkpoint_name*`] [-l]` *manifest*

다음 명령 옵션을 검토하십시오.

표 3–1 distro_const 명령 옵션

명령 옵션	설명
`distro_const build` *manifest*	지정된 매니페스트 파일을 사용하여 이미지를 한 단계로 빌드합니다.
`distro_const build -v`	Verbose 모드
`distro_const build -l` *manifest*	이미지 빌드를 일시 중지했다가 다시 시작할 수 있는 유효한 체크포인트를 모두 나열합니다.
`distro_const build -p` *checkpoint_name manifest*	지정된 체크포인트에서 이미지 빌드를 일시 중지합니다.
`distro_const build -r` *checkpoint_name manifest*	지정된 체크포인트에서 이미지 빌드를 다시 시작합니다.

표 3–1 distro_const 명령 옵션 *(계속)*

명령 옵션	설명
`distro_const build -h`	명령에 대한 도움말을 표시합니다.

주 – `distro_const` 명령을 사용하려면 사용자가 루트 역할이어야 합니다.

▼ 이미지를 한 단계로 빌드하는 방법

시작하기 전에 `distribution-constructor` 패키지를 다운로드한 다음 이미지에 대한 매니페스트를 선택하십시오. 필요한 경우 매니페스트를 사용자 정의하고 사용자 정의 스크립트를 추가하십시오.

1 루트 역할로 전환합니다.

2 일시 중지하지 않고 이미지 전체 빌드를 실행하려면 다음과 같이 옵션 없이 기본 `distro_const` 명령을 사용합니다.

```
# distro_const build manifest
```

주 – `build` 하위 명령은 필수입니다.

*manifest*를 이미지에 대한 블루프린트로 사용할 매니페스트 파일의 이름으로 대체합니다.

예를 들어, 다음 명령어를 입력합니다.

```
# distro_const build /usr/share/distro_const/dc_livecd.xml
```

3 배포판 구성자가 이미지에 필요한 패키지를 가져옵니다.

4 배포판 구성자가 매니페스트 파일에 지정한 사항에 따라 이미지를 빌드합니다.

5 (옵션) 빌드가 완료되면 빌드 프로세스에 대해 보고하는 로그 파일을 볼 수 있습니다.

빌드 출력에 로그 파일의 표시됩니다.

▼ 이미지를 여러 단계로 빌드하는 방법

`distro_const` 명령에 제공된 옵션으로 이미지 생성 프로세스의 여러 단계에서 빌드 프로세스를 중지 및 다시 시작하여 빌드 중인 이미지의 파일, 패키지 및 스크립트 선택 사항을 검사하고 디버그할 수 있습니다. 이 프로세스는 다음 기본 지침에 설명된 것과 같이 `distro_const` 명령에 사용할 수 있는 체크포인팅 옵션을 사용합니다.

1 **루트 역할로 전환합니다.**

2 **이미지를 빌드하기 전에 빌드를 일시 중지하거나 다시 시작할 수 있는 유효한 체크포인트를 확인합니다.**

```
# distro_const build -l manifest.xml
```

주 – `build` 하위 명령은 필수입니다.

이 명령은 이미지 빌드를 일시 중지하거나 다시 시작할 수 있는 유효한 체크포인트를 표시합니다. 이 명령으로 제공된 체크포인트 이름을 다른 체크포인팅 명령 옵션의 유효한 값으로 사용합니다.

예를 들어 다음 명령은 매니페스트 파일 `dc_livecd.xml`이 제공된 경우 사용 가능한 체크포인트를 확인합니다.

```
# distro_const build -l /usr/share/distro_const/dc_livecd.xml
```

명령이 실행된 후 유효한 체크포인트가 표시됩니다. 예를 들어 체크포인트에 다음이 포함될 수 있습니다.

```
Checkpoint        Resumable Description
--------------    --------- -------------
transfer-ips-install X Transfer package contents from IPS
set-ips-attributes   X Set post-installation IPS attributes
pre-pkg-img-mod      X Pre-package image modification
ba-init                Boot archive initialization
ba-config              Boot archive configuration
ba-arch                Boot archive archiving
grub-setup             Set up the GRUB menu
pkg-img-mod            Package image area modifications
create-iso             ISO image creation
```

주 – 이 샘플 명령 출력에서 Resumable 필드의 "X"는 해당 체크포인트에서 빌드를 다시 시작할 수 있음을 나타냅니다.

---

3 **이미지를 빌드하고 지정된 체크포인트에서 이미지 빌드를 일시 중지합니다.**

```
# distro_const build -p checkpoint_name manifest
```

주 – `build` 하위 명령과 *checkpoint_name* 및 *manifest* 필드는 필수입니다.

예를 들어 다음 명령은 이미지 빌드를 시작하고 `ba-arch`가 이미지 영역을 수정하기 전에 빌드를 일시 중지합니다.

```
# distro_const build -p ba-arch /usr/share/distro_const/dc_livecd.xml
```

4 **지정된 체크포인트에서 이미지 빌드를 다시 시작합니다.**

```
# distro_const build -r checkpoint_name  manifest
```

주 – 지정된 체크포인트는 이전 빌드가 실행을 중지한 체크포인트이거나 이전 체크포인트여야 합니다. 이후 체크포인트는 유효하지 않습니다. *checkpoint_name* 및 *manifest* 필드와 `build` 하위 명령은 필수입니다.

예를 들어 다음 명령은 `ba-arch` 단계에서 이미지 빌드를 다시 시작합니다.

```
# distro_const build -r ba-arch /usr/share/distro_const/dc_livecd.xml
```

주 – `build` 명령에서 일시 중지 및 다시 시작 옵션을 결합할 수 있습니다.

---

5 **(옵션) 빌드가 완료되면 빌드 프로세스에 대해 보고하는 로그 파일을 볼 수 있습니다.**

빌드 출력에 로그 파일의 표시됩니다.