

## Oracle® Solaris 11.2의 소프트웨어 추가 및 업데이트

ORACLE®

부품 번호: E53747  
2014년 7월

Copyright © 2007, 2014, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

---

이 설명서 사용 .....	9
<b>1 이미지 패키징 시스템 소개 .....</b>	<b>11</b>
이미지 패키징 시스템 .....	11
IPS 개념 .....	11
IPS 패키지 .....	12
FMRI(Fault Management Resource Identifier) .....	13
게시자, 저장소 및 패키지 아카이브 .....	15
저장소 원본 및 미러 .....	15
이미지 및 부트 환경 .....	16
패키지 페이지 및 변형 .....	17
설치 권한 .....	17
<b>2 소프트웨어 패키지 정보 보기 .....</b>	<b>19</b>
패키지 설치 상태 정보 표시 .....	19
설치된 패키지 .....	19
설치 가능한 패키지 .....	20
최신 패키지 .....	20
사용 가능한 업데이트가 있는 패키지 .....	21
모든 사용 가능한 패키지 .....	21
이름이 바뀐 패키지와 오래된 패키지 .....	21
특정 버전에 고정된 패키지 .....	22
패키지 설명 또는 라이선스 표시 .....	23
패키지 설명, 크기, 전체 FMRI 표시 .....	23
패키지 라이선스 표시 .....	24
패키지 매니페스트의 정보 표시 .....	24
패키지를 통해 설치된 파일 나열 .....	24
패키지를 통해 설치된 파일의 속성 표시 .....	25
기타 파일 시스템 객체 및 속성 표시 .....	26
그룹 패키지의 설치 가능한 모든 패키지 나열 .....	26

라이선스 요구 사항 표시 .....	27
패키지 검색 .....	27
pkg search 및 pkg contents 명령 비교 .....	27
검색 질의 지정 .....	28
지정한 파일을 제공하는 패키지 식별 .....	29
지정한 SMF 서비스를 제공하는 패키지 식별 .....	29
분류 또는 범주별로 패키지 나열 .....	30
종속 패키지 표시 .....	31
그룹 패키지의 모든 패키지 나열 .....	31
<b>3 소프트웨어 패키지 설치 및 업데이트 .....</b>	<b>33</b>
작업 미리 보기 .....	33
패키지 설치 및 업데이트 .....	35
공통 설치 옵션 .....	35
새 패키지 설치 .....	39
새 부트 환경에 패키지 설치 .....	41
패키지 거부 .....	42
패키지 업데이트 .....	43
패키지 다운그레이드 .....	43
설치된 패키지의 문제 수정 .....	44
pkg fix 및 pkg revert 명령 비교 .....	44
패키지 검증 및 검증 오류 수정 .....	44
파일 복원 .....	46
패키지 제거 .....	48
이미지 다시 설치 .....	48
비전역 영역 작업 .....	49
전역 영역과 비전역 영역의 관계 .....	50
시스템 저장소 및 프록시 서비스 .....	51
여러 비전역 영역을 동시에 업데이트 .....	53
<b>4 Oracle Solaris 이미지 업데이트 또는 업그레이드 .....</b>	<b>55</b>
이미지 업데이트 개요 .....	55
이미지 업데이트 모범 사례 .....	56
사용 가능한 버전 확인 .....	56
업데이트 작업 미리 보기 .....	56
새 부트 환경 지정 .....	57
설치할 버전 지정 .....	58
업데이트 전 버전 제약 조건 지정 .....	58
사용자 정의 통합 설치 .....	59

사용자 정의 통합 패키지 만들기 .....	59
업그레이드 제어 패키지 설치 .....	62
업그레이드 제어 패키지 업데이트 .....	63
이미지 업그레이드 .....	64
이미지 다운그레이드 .....	65
<b>5 설치된 이미지 구성 .....</b>	<b>67</b>
게시자 구성 .....	67
게시자 정보 표시 .....	67
패키지 게시자 추가, 수정 또는 제거 .....	69
프록시 지정 .....	72
선택적 구성 요소의 설치 제어 .....	74
변형 및 페이지 값이 패키지 설치에 영향을 미치는 방식 .....	75
변형 및 페이지 값의 예 .....	75
변형 값 표시 및 변경 .....	76
페이지 값 표시 및 변경 .....	77
지정한 버전으로 패키지 잠금 .....	78
통합에서 지정한 버전 제약 조건 완화 .....	79
기본 응용 프로그램 구현 지정 .....	81
중개의 참가자 식별 .....	81
기본 응용 프로그램 변경 .....	82
그룹 패키지의 일부 패키지 설치 방지 .....	83
이미지 및 게시자 등록 정보 구성 .....	86
부트 환경 정책 이미지 등록 정보 .....	86
패키지 서명에 필요한 등록 정보 .....	87
추가 이미지 등록 정보 .....	90
이미지 등록 정보 설정 .....	91
이미지 만들기 .....	92
작업 내역 보기 .....	93
<b>A 패키지 설치 및 업데이트 문제 해결 .....</b>	<b>97</b>
초기 문제 해결 단계 .....	97
설치된 버전의 pkg:/entire 확인 .....	97
구성된 게시자 원본의 내용 확인 .....	98
설치 재시도 .....	100
게시자나 저장소에 액세스할 수 없음 .....	101
패키지 저장소에 액세스할 수 없음 .....	101
SSL 인증서 문제 .....	102
위치를 찾을 수 없음 .....	103

서비스를 사용할 수 없음 .....	104
사용 가능한 업데이트가 없음 .....	105
패키지를 설치할 수 없음 .....	105
제약 조건을 충족할 수 없음 .....	106
통합의 제약을 받는 패키지 업데이트 .....	106
적절한 종속성을 찾을 수 없을 때 통합 업데이트 .....	110
설치된 종속성을 허용할 수 없을 때 통합 업데이트 .....	112
필요한 패키지를 찾을 수 없음 .....	112
필요한 패키지가 거부됨 .....	113
패키지가 예상한 대로 업데이트되지 않음 .....	114
동기 링크된 패키지를 설치할 수 없음 .....	115
비전역 영역을 설치할 수 없음 .....	116
이미지를 수정할 수 없음 .....	117
파일이 회수됨 .....	117
저장된 이미지 메타 데이터 최소화 .....	118
패키지 설치 성능 향상 .....	118
<b>B IPS 그래픽 사용자 인터페이스 .....</b>	<b>121</b>
패키지 관리자 사용 .....	121
패키지 관리자 명령줄 옵션 .....	121
웹 설치 사용 .....	122
업데이트 관리자 사용 .....	124
업데이트 관리자 명령줄 옵션 .....	126
<b>색인 .....</b>	<b>127</b>

## 코드 예

---

예 5-1	새 게시자 지정 .....	69
예 5-2	게시자 구성 가져오기 .....	69
예 5-3	게시자 키와 인증서 지정 .....	71
예 5-4	게시자 키와 인증서 취소 .....	72
예 5-5	무시 목록에서 패키지 추가 및 패키지 제거 .....	84
예 A-1	Java Runtime Environment 잠금 해제 및 업데이트 .....	107
예 A-2	종속성을 잠금 해제하고 별도로 업데이트할 때 pkg:/entire 업데이트 .....	110





## 이 설명서 사용

---

- **개요** - Oracle Solaris 이미지 패키징 시스템(IPS)의 소프트웨어 설치 기능에 대해 설명합니다. IPS 명령을 사용하여 소프트웨어 패키지를 나열 및 검색하고, 소프트웨어를 설치 및 제거하고, 새로운 Oracle Solaris 운영 체제 릴리스로 업그레이드할 수 있습니다.
- **대상** - 소프트웨어를 설치 및 관리하고 시스템 이미지를 관리하는 시스템 관리자를 대상으로 합니다.
- **필요한 지식** - Oracle Solaris 시스템 관리 경험이 필요합니다.

## 제품 설명서 라이브러리

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

## Oracle 지원 액세스

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

## 피드백

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



## 이미지 패키징 시스템 소개

---

Oracle Solaris 이미지 패키징 시스템(IPS)은 소프트웨어 패키지를 나열 및 검색하고, 소프트웨어를 설치 및 제거하고, 새로운 Oracle Solaris 운영 체제 릴리스로 업그레이드할 수 있도록 지원하는 프레임워크입니다. IPS 명령을 사용하면 설치할 수 있는 패키지 또는 패키지 버전을 제한할 수 있습니다.

### 이미지 패키징 시스템

Oracle Solaris 11 소프트웨어는 IPS 패키지로 배포됩니다. IPS 패키지는 IPS 게시자가 채우는 IPS 패키지 저장소에 저장됩니다. IPS 패키지는 Oracle Solaris 11 이미지에 설치됩니다. IPS 명령줄 인터페이스를 통해 사용할 수 있는 기능 중 일부는 패키지 관리자 그래픽 사용자 인터페이스를 통해 사용할 수 있습니다.

IPS 도구가 제공하는 기능은 다음과 같습니다. 게시자, 저장소 등의 용어에 대한 정의는 [“IPS 개념” \[11\]](#)을 참조하십시오.

- 소프트웨어 패키지를 나열, 검색, 설치, 업데이트 및 제거하고 설치를 제한합니다.
- 패키지 게시자를 나열, 추가 및 제거합니다. 검색 우선 순위, 고착성 같은 게시자 속성을 변경합니다. 서명 정책 같은 게시자 등록 정보를 설정합니다.
- 이미지를 새 운영 체제 릴리스로 업그레이드합니다.
- 기존 IPS 패키지 저장소의 복사본을 만듭니다. 새 패키지 저장소를 만듭니다. [“Oracle Solaris 11.2 패키지 저장소 복사 및 만들기”](#)를 참조하십시오.
- 패키지를 만들고 게시합니다. [“Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”](#)을 참조하십시오.
- 부트 환경 및 기타 이미지를 만듭니다.

IPS를 사용하려면 Oracle Solaris 11 OS를 실행해야 합니다. Oracle Solaris 11 OS를 설치하려면 [“Oracle Solaris 11.2 시스템 설치”](#)를 참조하십시오.

### IPS 개념

이 절에서는 IPS와 관련된 용어와 개념을 정의합니다.

## IPS 패키지

IPS 패키지는 매니페스트라는 텍스트 파일로 정의할 수 있습니다. 패키지 매니페스트는 키/값 쌍과 데이터 페이로드라는 정의된 형식으로 패키지 작업에 대해 설명합니다. 패키지 작업에는 파일, 디렉토리, 링크, 드라이버, 종속성, 그룹, 사용자 및 라이선스 정보가 포함됩니다. 패키지 작업은 패키지의 설치 가능한 객체를 나타냅니다. set이라는 작업은 분류, 요약, 설명 같은 패키지 메타 데이터를 정의합니다.

패키지 작업과 작업 키를 지정하여 패키지를 검색할 수 있습니다. 패키지 작업에 대한 설명은 [“Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”](#)의 [“Package Content: Actions”](#) 또는 pkg(5) 매뉴얼 페이지를 참조하십시오.

통합 패키지와 그룹 패키지는 파일과 같은 콘텐츠를 제공하지 않습니다. 통합 및 그룹 패키지는 관련 패키지 세트의 설치를 돕는 종속성을 지정합니다.

## 통합 패키지

통합 패키지는 설치할 수 있는 다른 패키지의 버전을 지정합니다. 통합 패키지를 설치해도 다른 패키지가 설치되지는 않습니다. 통합 패키지는 incorporate 종속성 패키지를 설치할 경우 미리 정해진 버전의 종속 패키지만 설치할 수 있도록 보장합니다. 예를 들어, 설치된 통합 패키지에서 incorporate 종속성으로 지정된 패키지의 버전 값이 1.4.3인 경우, 버전 값이 1.4.3보다 작거나 1.4.4보다 크거나 같은 패키지 버전은 설치할 수 없습니다. 버전 값이 1.4.3.7인 패키지 버전은 설치할 수 있습니다.

통합 패키지는 종종 많은 incorporate 종속성을 지정하여 호환되는 패키지 버전 공간에 표면을 정의합니다. 이러한 incorporate 종속성 세트를 포함하는 패키지를 흔히 통합이라고 합니다. 통합은 대개 개별적으로 버전이 지정된 것이 아니라 종합적으로 구축된 소프트웨어 패키지 세트를 정의하는 데 사용됩니다. incorporate 종속성은 Oracle Solaris에서 호환되는 소프트웨어 버전을 함께 설치할 때 자주 사용됩니다.

통합 패키지에 incorporate 종속성으로 명명된 패키지는 그 자체로 통합 패키지가 될 수 있습니다. 이 방법으로 통합 패키지의 매니페스트에 이름이 없더라도 많은 패키지가 통합 패키지의 영향을 받을 수 있습니다. 설치 시 통합 패키지의 영향을 받는 패키지는 해당 통합 패키지의 제약을 받습니다. A-incorporation 패키지에 통합된 B-incorporation을 업데이트하면 A-incorporation은 물론 A-incorporation의 제약을 받는 다른 패키지도 업데이트됩니다.

통합은 제약된 패키지를 강제로 동시에 업그레이드하여 작동 중인 지원 가능한 이미지 상태가 유지되도록 도와줍니다. 일반적으로 통합 패키지의 제약을 받는 패키지를 설치하거나 업데이트하면 안됩니다. 대신, 통합 패키지를 업데이트해야 합니다. 제약된 패키지는 제거는 가능하지만 제약된 패키지를 설치하거나 업데이트할 경우 버전이 제약됩니다. 관련 정보는 [“통합에서 지정한 버전 제약 조건 완화” \[79\]](#)를 참조하십시오.

pkg://solaris/entire 패키지는 다른 많은 통합 패키지에 incorporate 종속성을 지정하는 특수 통합으로, 이미지에 설치된 대부분의 시스템 소프트웨어의 버전을 제약합니다.



주의 - entire라는 패키지는 제거하지 마십시오. entire 패키지는 시스템 패키지 버전을 제한하므로 결과 패키지 세트는 지원 가능한 이미지가 됩니다. 적절한 시스템 업데이트와 올바른 패키지 선택은 이 통합에 달려 있습니다. entire 패키지를 제거하면 지원되지 않는 시스템이 됩니다.

## 그룹 패키지

그룹 패키지는 기능이나 도구를 구성하는 패키지 세트를 지정합니다. 그룹 패키지를 설치하면 해당 그룹 패키지의 모든 group 종속성 패키지가 설치됩니다. 그룹 패키지에 group 종속성으로 지정된 패키지는 패키지 버전을 지정하지 않습니다. 그룹 패키지는 버전 관리 도구가 아니라 콘텐츠 관리 도구입니다.

그룹 패키지는 group 종속성에 명명된 패키지가 이미 설치되었거나 무시 목록에 있지 않는 한, 이러한 패키지를 제공합니다. 이미지의 무시 목록에 대한 자세한 내용은 [“그룹 패키지의 일부 패키지 설치 방지” \[83\]](#)를 참조하십시오.

예를 들어, group/feature/storage-server 패키지는 아직 설치되지 않은 경우 저장소 관련 드라이버, 서비스, 파일 시스템, I/O 구성 요소, 라이브러리, 유틸리티를 제공합니다. group/system/solaris-minimal-server 패키지는 최소 지원 Oracle Solaris 환경에 필요한 패키지 세트를 제공합니다. 그룹 패키지에서 제공하는 모든 패키지를 나열하는 방법에 대한 예는 [“그룹 패키지의 설치 가능한 모든 패키지 나열” \[26\]](#)을 참조하십시오.

그룹 패키지를 제거해도 group 종속성에 명명된 패키지가 모두 제거되는 것은 아닙니다. 여전히 설치되어 있는 다른 소프트웨어에 필요한 패키지는 그룹 패키지를 제거할 때 제거되지 않습니다.

## FMRI(Fault Management Resource Identifier)

각 패키지는 FMRI(Fault Management Resource Identifier)로 나타냅니다. 패키지의 전체 FMRI는 다음 형식의 체계, 게시자, 패키지 이름 및 버전 문자열로 구성됩니다.

```
scheme://publisher/name@version:datetimez
```

체계, 게시자 및 버전 문자열은 선택 사항입니다. IPS 명령 피연산자에서 패키지를 고유하게 식별하는 패키지 이름의 가장 작은 부분을 사용할 수 있으며, ? 및 \* 문자를 glob(3C) 스타일의 와일드카드로 사용하여 하나 이상의 패키지를 일치시킬 수 있습니다.

다음 예제 패키지 FMRI는 suri 저장소 라이브러리용입니다.

```
pkg://solaris/system/library/storage/suri@0.5.11,5.11-0.175.2.0.0.34.0:20140303T145535Z
```

체계

pkg

게시자

solaris

게시자를 지정할 경우 게시자 이름 앞에 pkg:// 또는 //를 붙여야 합니다.

패키지 이름

system/library/storage/suri

패키지 이름은 임의의 수의 구성 요소가 슬래시(/) 문자로 구분된 계층 구조입니다. IPS 명령에서 명령에 사용되는 패키지 이름으로 패키지가 고유하게 식별되는 경우 패키지 이름의 선행 구성 요소를 생략할 수 있습니다. 전체 패키지 이름을 지정하되 게시자를 생략할 경우 전체 패키지 이름 앞에 pkg:// 또는 //가 아니라 pkg:/ 또는 /를 붙일 수 있습니다. 패키지 이름을 약어로 지정하는 경우 패키지 이름 왼쪽에 다른 문자를 사용하지 마십시오.

버전

패키지 버전은 네 부분으로 구성됩니다.

구성 요소 버전: 0.5.11

운영 체제에 긴밀하게 바인딩된 구성 요소의 경우 일반적으로 해당 운영 체제 버전의 `uname -r` 값이 구성 요소 버전에 포함됩니다. 고유의 개발 수명 주기가 있는 구성 요소의 경우 구성 요소 버전은 점으로 구분된 릴리스 번호(예: 2.4.10)입니다.

빌드 버전: 5.11

빌드 버전 앞에는 쉼표(,)가 나와야 합니다. 빌드 버전은 패키지 콘텐츠가 생성된 운영 체제의 버전을 지정합니다.

분기 버전: 0.175.2.0.0.34.0

분기 버전 앞에는 하이픈(-)이 나와야 합니다. 분기 버전은 공급업체 관련 정보를 제공합니다.

Oracle Solaris 패키지에서는 패키지 FMRI 버전 문자열의 분기 버전 부분에 다음 정보를 표시합니다.

주 릴리스 번호: 0.175

주 또는 마케팅 개발 릴리스 빌드 번호입니다. 이 예제에서 0.175는 Oracle Solaris 11을 나타냅니다.

업데이트 릴리스 번호: 2

이 Oracle Solaris 릴리스의 업데이트 릴리스 번호입니다. 업데이트 값은 Oracle Solaris 릴리스의 첫번째 고객 출하인 경우 0, 해당 릴리스의 첫번째 업

데이트인 경우 1, 해당 릴리스의 두번째 업데이트인 경우 2 등입니다. 이 예제에서 1은 Oracle Solaris 11.1을 나타냅니다.

SRU 번호: 0

이 업데이트 릴리스에 대한 SRU(Support Repository Update) 번호입니다. SRU에는 버그 수정만 포함되며 새 기능은 포함되지 않습니다. Oracle Support Repository는 지원 계약 중인 시스템에만 제공됩니다.

예약됨: 0

이 필드는 Oracle Solaris 패키지에 현재 사용되지 않습니다.

SRU 빌드 번호: 34

SRU의 빌드 번호 또는 주 릴리스의 래스핀 번호입니다.

야간 빌드 번호: 0

개별 야간 빌드에 대한 빌드 번호입니다.

시간 기록: 20140303T145535Z

시간 기록 앞에는 콜론(:)이 나와야 합니다. 시간 기록은 패키지가 게시된 시간을 나타내며 ISO-8601 기본 형식 `YYYYMMDDTHHMMSSZ`로 표시됩니다.

## 게시자, 저장소 및 패키지 아카이브

게시자는 하나 이상의 패키지를 제공하는 개인이나 조직을 나타냅니다. 게시자는 패키지 저장소나 패키지 아카이브를 사용하여 패키지를 배포할 수 있습니다. 기본 검색 순서로 게시자를 구성할 수 있습니다. 패키지 사양에 게시자 이름을 포함하지 않고 패키지 설치 명령을 실행하면 해당 패키지에 대해 검색 순서의 첫번째 게시자가 검색됩니다. 지정된 패키지 FMRI 패턴의 일치 항목을 찾지 못하면 검색 순서의 두번째 게시자가 검색되고, 패키지를 찾거나 모든 게시자를 검색할 때까지 이런 식으로 다음 게시자가 검색됩니다.

저장소는 패키지가 게시되고 검색되는 위치입니다. 이 위치는 URI(Universal Resource Identifier)로 지정됩니다. 카탈로그는 저장소에 있는 모든 패키지의 목록입니다.

패키지 아카이브는 게시자 정보와 해당 게시자가 제공한 하나 이상의 패키지가 포함된 파일입니다.

## 저장소 원본 및 미러

원본은 패키지 메타 데이터(예: 카탈로그, 매니페스트, 검색 색인)와 패키지 콘텐츠(파일)가 모두 포함된 패키지 저장소입니다. 이미지의 지정된 게시자에 대해 원본이 여러 개 구성된

경우 IPS 클라이언트는 패키지 데이터를 검색할 수 있는 최상의 원본을 선택하려고 시도합니다.

미러는 패키지 콘텐츠만 포함된 패키지 저장소입니다. 미러 저장소로부터 패키지를 설치 및 업데이트하는 클라이언트는 원본 저장소로부터 메타 데이터를 다운로드해야 합니다. IPS 클라이언트는 패키지 콘텐츠를 미러에서 다운로드하는 경우에도 게시자의 카탈로그를 구하기 위해 원본에 액세스합니다. 게시자에 대해 하나의 미러가 구성된 경우 IPS 클라이언트는 해당 미러를 패키지 콘텐츠 검색용으로 선택합니다. 이미지의 지정된 게시자에 대해 미러가 여러 개 구성된 경우 IPS 클라이언트는 패키지 콘텐츠를 검색할 수 있는 최상의 미러를 선택하려고 시도합니다. 모든 미러에 연결할 수 없거나 필요한 콘텐츠가 없거나 미러가 너무 느린 경우 IPS 클라이언트는 원본에서 콘텐츠를 검색합니다. 자세한 내용은 [pkg\(5\)](#) 매뉴얼 페이지의 “Publishers and Repositories”를 참조하십시오.

---

참고 - 미러 저장소로 지정된 저장소에 콘텐츠와 메타 데이터가 모두 완료된 경우에도, 사용자는 동일한 게시자의 원본 저장소에도 동일 패키지의 동일 버전이 존재하지 않으면 미러 저장소의 콘텐츠에 액세스할 수 없습니다.

---

## 이미지 및 부트 환경

이미지는 IPS 패키지를 설치할 수 있으며 다른 IPS 작업을 수행할 수 있는 위치입니다.

부트 환경(BE)은 이미지의 부트 가능한 인스턴스입니다. 물리적/가상 시스템에서 여러 BE를 유지 관리할 수 있으며, 각 BE에 서로 다른 운영 체제 버전을 포함한 서로 다른 소프트웨어 버전을 설치할 수 있습니다. 시스템을 부트할 때 시스템에 있는 BE 중 하나로 부트하는 옵션이 제공됩니다. 패키지 작업의 결과로 새 BE가 자동으로 생성될 수 있습니다. 새 BE가 자동으로 만들어지는지 여부는 “부트 환경 정책 이미지 등록 정보” [86]에 설명된 대로 이미지 정책에 따라 달라집니다. “부트 환경 옵션” [35]에 설명된 옵션을 지정하여 새 BE를 명시적으로 만들 수도 있습니다. `beadm` 명령을 사용하여 새 BE를 만드는 방법에 대한 자세한 내용은 “Oracle Solaris 11.2 부트 환경 만들기 및 관리”를 참조하십시오.

BE에 속하는 파일 시스템에만 패키지를 설치할 수 있습니다. 예를 들어, 기본 Oracle Solaris 11 설치에서 `rpool/ROOT/BEname/` 아래의 데이터 세트만 패키지 작업에 지원됩니다.

Oracle Solaris 영역은 이미지의 또 다른 예입니다. 비전역 영역은 전역 영역이라는 Oracle Solaris 운영 체제의 인스턴스 내에 만들어진 가상화된 운영 체제 환경입니다. 전역 영역은 부모 이미지이고, 해당 전역 영역 내의 비전역 영역은 전역 영역의 자식 이미지입니다. IPS 명령 출력에서 비전역 영역은 부모 전역 영역 이미지에 링크되어 있으므로 링크된 이미지라고도 합니다.

전역 영역에서 실행된 IPS 명령은 “비전역 영역 작업” [49]에 설명된 대로 비전역 영역에 영향을 미칠 수 있습니다. 전역 영역에서 실행된 IPS 명령은 커널 영역(`solaris-kz` 브랜드 영역) 또는 Oracle Solaris 10 영역(`solaris10` 브랜드 영역)에 영향을 미치지 않습니다. 이 설명서에서 “비전역 영역”은 `solaris` 브랜드의 Oracle Solaris 11 비전역 영역을 의미합니다. 영역에 대한 자세한 내용은 “Oracle Solaris 영역 소개”를 참조하십시오.



## 패키지 페이지 및 변형

소프트웨어에는 선택적 구성 요소와 상호 배타적인 구성 요소가 포함될 수 있습니다. 선택적 구성 요소의 예로는 로케일 설명서가 있습니다. 상호 배타적 구성 요소의 예로는 SPARC 또는 x86 및 디버그 또는 비디버그 이진이 있습니다. IPS에서는 선택적 구성 요소를 페이지시라 하고 상호 배타적 구성 요소를 변형이라고 합니다.

페이지 및 변형은 이미지에 설정된 특수한 등록 정보입니다. 페이지 및 변형은 패키지 매니페스트의 작업에 설정된 태그이기도 합니다. 작업의 페이지 및 변형 태그 값을 이미지에 설정된 페이지 및 변형 값과 비교한 결과에 따라, 패키지 작업을 설치할 수 있는지 여부가 결정됩니다. 예를 들어 이미지에서 특정 로케일 페이지시를 false로 설정하는 경우 해당 페이지시를 지정하는 파일 작업은 설치되지 않고 해당 페이지시를 지정하는 현재 설치된 파일 작업은 제거됩니다.

대부분의 변형은 임의의 값을 가질 수 있습니다. 작업에 설정된 페이지 태그는 true 값만 가질 수 있습니다. 이미지에 설정된 페이지 등록 정보는 true 또는 false 값만 가질 수 있습니다.

다음 알고리즘에서는 이미지에 설정된 페이지 및 변형이 특정 작업의 설치 여부에 어떤 영향을 주는지 설명합니다.

- 페이지 또는 변형 태그가 없는 작업은 항상 설치됩니다.
- 페이지 태그가 있는 작업은 태그와 일치하는 모든 페이지 또는 페이지 패턴이 이미지에 false로 설정되지 않는 한 설치됩니다. 어떤 페이지가 true로 설정되거나 명시적으로 설정되지 않은 경우(true가 기본값) 작업이 설치됩니다.
- 변형 태그가 있는 작업은 모든 변형 태그의 값이 이미지에 설정된 값과 동일한 경우에만 설치됩니다.
- 페이지 및 변형 태그가 모두 있는 작업은 페이지와 변형이 모두 작업 설치를 허용하는 경우 설치됩니다.

이미지에 설정된 페이지와 변형의 값을 보거나 수정하려면 [“선택적 구성 요소의 설치 제어” \[74\]](#)를 참조하십시오.

## 설치 권한

**2장. 소프트웨어 패키지 정보 보기**에 설명되어 있는 명령은 특별한 권한이 없어도 사용할 수 있습니다. IPS 패키지 설치, 운영 체제 업그레이드, 게시자 및 이미지 등록 정보 구성 등의 작업을 수행하려면 추가 권한이 필요합니다. 다음 방법 중 하나를 사용하여 필요한 권한을 얻을 수 있습니다. 필요한 프로파일 또는 역할을 확인하는 방법을 포함하여 프로파일 및 역할에 대한 자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)을 참조하십시오.

#### 권한 프로파일

`profiles` 명령을 사용하여 지정 받은 권한 프로파일을 나열합니다. 예를 들어, Software Installation 권한 프로파일이 있으면 `pkg` 및 `beadm` 명령을 실행하여 패키지를 설치 및 업데이트하고 부트 환경을 관리할 수 있습니다.

#### 역할

`roles` 명령을 사용하여 지정 받은 역할을 나열합니다. 예를 들어, `root` 역할이 있으면 `su` 명령과 `root` 암호를 사용하여 `root` 역할을 맡을 수 있습니다.

#### sudo 명령

사이트의 보안 정책에 따라 사용자 암호와 함께 `sudo` 명령을 사용하여 권한이 있는 명령을 실행할 수 있는 경우도 있습니다.

# ◆◆◆ 2 장

## 소프트웨어 패키지 정보 보기

---

이 장에서는 다음과 같은 패키지 정보를 제공하는 명령에 대해 설명합니다.

- 패키지가 설치되었는지 여부 또는 설치/업데이트가 가능한지 여부
- 패키지에 대한 설명, 크기 및 버전
- 그룹 패키지에 포함된 패키지
- 지정한 파일을 제공하는 패키지
- 지정한 SMF 서비스를 제공하는 패키지
- 지정한 패키지에 종속된 패키지
- 특정 범주에 해당하는 패키지

구성된 계시자에 대한 패키지 저장소의 내용이 변경된 경우, 가장 최신 정보를 받을 수 있도록 세션 시작 시 사용 가능한 패키지 목록을 업데이트합니다. 패키지 목록을 업데이트하려면 `pkg refresh` 명령을 실행합니다. 예를 들어, 저장소가 새 패키지로 업데이트된 경우 사용 가능한 패키지를 나열하면 `pkg refresh`를 실행할 때까지 새 패키지가 표시되지 않을 수 있습니다.

이 장에 설명된 명령에 대한 모든 옵션의 전체 목록은 [pkg\(1\)](#) 매뉴얼 페이지를 참조하십시오.

### 패키지 설치 상태 정보 표시

`pkg list` 명령은 패키지가 현재 이미지에 설치되었는지 여부와 업데이트를 사용할 수 있는지 여부를 보여줍니다. 옵션이나 피연산자가 없는 경우 `pkg list` 명령은 현재 이미지에 설치된 모든 패키지를 나열합니다. 결과 범위를 좁히려면 하나 이상의 패키지 이름을 제공합니다. 패키지 이름에 와일드카드를 사용할 수 있습니다. 인수가 `pkg`로 직접 전달되고 셸이 인수를 확장하지 않도록 와일드카드를 따옴표로 묶습니다.

### 설치된 패키지

다음 예제에 표시된 대로, `pkg list` 명령은 각 일치 패키지에 대해 한 줄로 정보를 표시합니다. 1 열의 "i"는 이러한 패키지가 이 이미지에 설치되어 있음을 나타냅니다.

```
$ pkg list '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
isvtoolkit (isvpub)       1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.34.0 i--
```

괄호 안의 게시자 이름은 isvpub 게시자가 이 이미지의 게시자 검색 순서에서 첫번째 게시자가 아님을 나타냅니다. 이 이미지에 설치되어 있는 dtrace-toolkit 패키지는 검색 순서의 첫번째 게시자가 게시했습니다.

## 설치 가능한 패키지

설치되어 있는 패키지와 이 이미지에 설치되어 있지 않지만 설치가 가능한 패키지의 최신 버전을 나열하려면 -a 옵션을 사용합니다.

```
$ pkg list -a '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
image/nvidia/cg-toolkit  3.0.15-0.175.2.0.0.17.0 ---
isvtoolkit (isvpub)       1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.34.0 i--
```

이 출력은 image/nvidia/cg-toolkit 패키지가 사용 가능하고 이 이미지에 설치할 수 있음을 나타냅니다.

## 최신 패키지

이 이미지에 설치할 수 없는 패키지를 포함하여 모든 일치 패키지의 최신 버전을 나열하려면 -n 옵션을 사용합니다.

```
$ pkg list -n '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
developer/dtrace/toolkit  0.99-0.173.0.0.0.1.0  --r
image/nvidia/cg-toolkit  3.0.15-0.175.2.0.0.17.0 ---
isvtoolkit (isvpub)       1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.35.0 ---
```

developer/dtrace/toolkit 패키지와 system/dtrace/dtrace-toolkit 패키지는 이 이미지에 설치할 수 없습니다. 알다시피 이 패키지는 -a 옵션으로 나열되지 않았기 때문입니다. developer/dtrace/toolkit 패키지는 이름이 바뀌었습니다. 자세한 내용은 [“이름이 바뀐 패키지와 오래된 패키지” \[21\]](#)를 참조하십시오.

현재 이 이미지에 설치된 버전보다 최신 버전의 dtrace-toolkit 패키지를 구성된 게시자에서 사용할 수 있습니다. 다음 예제에서 “Reason” 행에 표시된 대로, dtrace-toolkit 패키지가 종속된 다른 패키지도 업데이트하는 경우 최신 버전을 설치할 수 있습니다. 다음 명령 출력에서 간략하게 보이기 위해 시간 기록을 생략했습니다. pkg update 및 pkg install 명령은 [3장. 소프트웨어 패키지 설치 및 업데이트](#)에서 설명합니다.

```
$ pkg update -nv dtrace-toolkit
No updates are available for this image.
$ pkg install -nv dtrace-toolkit@0.99-0.175.2.0.0.35
pkg install: No matching version of system/dtrace/dtrace-toolkit can be installed:
  Reject: pkg://solaris/system/dtrace/dtrace-toolkit@0.99,5.11-0.175.2.0.0.35.0
  Reason: This version is excluded by installed incorporation
pkg://solaris/consolidation/osnet/osnet-incorporation@0.5.11,5.11-0.175.2.0.0.34.0
```

## 사용 가능한 업데이트가 있는 패키지

-u 옵션은 설치된 일치 패키지 중 최신 버전이 있는 패키지를 모두 나열합니다. “최신 패키지” [20]에 예시된 대로, 최신 버전이 있는 패키지의 수가 이 이미지에서 업데이트할 수 있는 패키지의 수보다 클 수 있습니다. 패키지는 설치된 패키지 종속성 및 게시자 구성에 따라 이미지에 부여된 제약 조건에서 허용하는 버전으로만 업데이트할 수 있습니다.

```
$ pkg list -u '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
system/dtrace/dtrace-toolkit  0.99-0.175.2.0.0.34.0  i--
```

## 모든 사용 가능한 패키지

이 이미지에 설치할 수 없는 패키지를 포함하여 모든 일치 패키지의 모든 사용 가능한 버전을 나열하려면 -af 옵션을 사용합니다. -f 옵션은 -a 옵션 없이 사용할 수 없습니다. 결과 범위를 좁히려면 버전 문자열의 일부를 지정할 수 있습니다. 특수 버전 문자열 @latest를 지정하면 -n 옵션과 동일한 결과가 표시됩니다.

```
$ pkg list -af '*toolkit@0.99-0.175.2'
$ pkg list -af '*toolkit@latest'
```

## 이름이 바뀐 패키지와 오래된 패키지

다음 예제에서 O 열의 “r”은 패키지 이름이 바뀌었음을 나타내고, O 열의 “o”는 패키지가 오래되었음을 나타냅니다.

```
$ pkg list -n developer/dtrace/toolkit database/mysql-50 web/amp
NAME (PUBLISHER)          VERSION          IFO
database/mysql-50        5.0.91-0.171    --o
developer/dtrace/toolkit  0.99-0.173.0.0.0.1.0  --r
web/amp                  0.5.11-0.174.0.0.0.0.0  --r
```

이러한 패키지는 아무것도 설치할 수 없습니다. 오래된 패키지를 설치하려고 시도하면 이 이미지에 필요한 업데이트가 없다는 메시지와 함께 설치를 실패합니다. 이름이 바뀐 패키지를 설치하려고 시도하면 시스템은 바뀐 이름으로 패키지를 설치하려고 합니다.

pkg info 명령을 사용하여 이름이 바뀐 패키지의 새 이름을 확인합니다. 패키지가 설치되지 않았으므로 -r 옵션을 사용하여 구성된 패키지 저장소를 질의합니다. 다음 예제에 표시된 대로 “Renamed to” 행을 살펴보십시오.

```
$ pkg info -r web/amp
      Name: web/amp
      Summary:
      State: Not installed (Renamed)
      Renamed to: group/feature/amp@0.5.11-0.174.0.0.0.0
                 consolidation/ips/ips-incorporation
      Publisher: solaris
      Version: 0.5.11
      Build Release: 5.11
      Branch: 0.174.0.0.0.0
      Packaging Date: September 21, 2011 07:15:02 PM
      Size: 5.45 kB
      FMRI: pkg://solaris/web/amp@0.5.11,5.11-0.174.0.0.0.0:20110921T191502Z
```

web/amp 패키지를 설치하려고 시도하면 패키지가 아직 설치되지 않았고 이 이미지에 설치할 수 있는 경우 group/feature/amp 패키지가 대신 설치됩니다.

다음 예제에서 “Renamed to” 패키지가 이미 설치되었으므로 패키징 시스템은 필요한 업데이트가 없다고 보고합니다.

```
$ pkg info -r developer/dtrace/toolkit
      Name: developer/dtrace/toolkit
      Summary:
      State: Not installed (Renamed)
      Renamed to: pkg:/system/dtrace/dtrace-toolkit@0.99,5.11-0.173.0.0.0.0
                 consolidation/osnet/osnet-incorporation
      Publisher: solaris
      Version: 0.99
      Build Release: 5.11
      Branch: 0.173.0.0.0.1.0
      Packaging Date: August 26, 2011 02:55:51 PM
      Size: 5.45 kB
      FMRI: pkg://solaris/developer/dtrace/
             toolkit@0.99,5.11-0.173.0.0.0.1.0:20110826T145551Z
$ pkg list dtrace-toolkit
NAME (PUBLISHER)          VERSION                IFO
system/dtrace/dtrace-toolkit  0.99-0.175.2.0.0.34.0  i--
$ pkg install developer/dtrace/toolkit
No updates necessary for this image.
```

## 특정 버전에 고정된 패키지

F 열의 "f"는 패키지가 고정되어 있음을 나타냅니다. 패키지가 고정된 경우 고정된 버전과 일치하는 패키지만 설치하거나 해당 패키지로만 업데이트할 수 있습니다. 패키지 고정에 대한 자세한 내용은 [“지정한 버전으로 패키지 잠금” \[78\]](#)을 참조하십시오.

```
$ pkg list openssl
NAME (PUBLISHER)          VERSION          IFO
library/security/openssl 1.0.1.6-0.175.2.0.0.34.0 if-
```

## 패키지 설명 또는 라이선스 표시

`pkg info` 명령은 이름, 설명, 설치 상태, 버전, 패키징 날짜, 패키지 크기, 전체 FMRI 등의 패키지 정보를 표시합니다. 옵션이나 피연산자가 없는 경우 `pkg info` 명령은 현재 이미지에 설치된 모든 패키지에 대한 정보를 표시합니다. 결과 범위를 좁히려면 하나 이상의 패키지 이름을 제공합니다. 패키지 이름에 와일드카드를 사용할 수 있습니다. 인수가 `pkg`로 직접 전달되고 셸이 인수를 확장하지 않도록 와일드카드를 따옴표로 묶습니다.

`info`와 `list` 하위 명령은 모두 패키지 이름, 게시자 및 버전 정보를 표시합니다. `pkg list` 명령은 패키지의 업데이트가 있는지 여부, 업데이트를 이 이미지에 설치할 수 있는지 여부 및 패키지가 오래되었거나 이름이 바뀌었거나 고정되었는지 여부를 보여줍니다. 또한 `pkg list` 명령은 패키지 요약과 전체 FMRI를 표시할 수 있습니다. `pkg info` 명령은 패키지 요약, 설명, 범주, 크기를 표시하며 라이선스 정보만 별도로 표시할 수도 있습니다.

## 패키지 설명, 크기, 전체 FMRI 표시

`pkg list -s` 명령을 사용하여 패키지 요약을 표시할 수 있습니다.

```
$ pkg list -s entire
NAME (PUBLISHER)  SUMMARY
entire            Incorporation to lock all system packages to the same build
```

`pkg list -v` 명령은 전체 패키지 FMRI를 나열합니다.

```
$ pkg list -v entire
FMRI          IFO
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z ---
```

`pkg info` 명령은 자세한 정보를 표시합니다.

```
$ pkg info entire
Name: entire
Summary: Incorporation to lock all system packages to the same build
Description: This package constrains system package versions to the same
build. WARNING: Proper system update and correct package
selection depend on the presence of this incorporation.
Removing this package will result in an unsupported system.
Category: Meta Packages/Incorporations
State: Installed
Publisher: solaris
Version: 0.5.11
Branch: 0.175.2.0.0.34.0
```

```
Packaging Date: March  3, 2014 06:26:43 PM
Size: 5.46 kB
FMRI: pkg://solaris/entire@0.5.11-0.175.2.0.0.34.0:20140303T182643Z
```

“이름이 바뀐 패키지와 오래된 패키지” [21]에 표시된 대로, pkg info 명령을 사용하여 이름이 바뀐 패키지의 새 이름을 찾을 수 있습니다.

## 패키지 라이선스 표시

--license 옵션을 사용하여 일치 패키지의 라이선스 텍스트를 표시합니다. 이 정보는 내용이 매우 길 수 있습니다. pkg info 명령에 --license 옵션이 지정되지 않은 경우 다른 정보는 표시되지 않습니다.

```
$ pkg info --license osnet-incorporation
You acknowledge that your use of this Oracle Solaris software product
is subject to (i) the license terms that you accepted when you obtained
the right to use Oracle Solaris software; or (ii) the license terms that
you agreed to when you placed your Oracle Solaris software order with
Oracle; or (iii) the Oracle Solaris software license terms included with
the hardware that you acquired from Oracle; or, if (i), (ii) or (iii)
are not applicable, then, (iv) the OTN License Agreement for Oracle
Solaris (which you acknowledge you have read and agree to) available at
http://www.oracle.com/technetwork/licenses/solaris-cluster-express-license-167852.html.
Note: Software downloaded for trial use or downloaded as replacement
media may not be used to update any unsupported software.
```

## 패키지 매니페스트의 정보 표시

pkg contents 명령은 패키지의 파일 시스템 콘텐츠를 표시합니다. 옵션이나 피연산자가 없는 경우 이 명령은 현재 이미지에 설치된 모든 패키지에 대한 경로 정보를 표시합니다. 명령 옵션을 사용하여 표시할 특정 패키지 콘텐츠를 지정합니다. 결과 범위를 좁히려면 하나 이상의 패키지 이름을 제공합니다. 패키지 이름에 와일드카드를 사용할 수 있습니다. 인수가 pkg로 직접 전달되고 셸이 인수를 확장하지 않도록 와일드카드를 따옴표로 묶습니다.

contents와 search 하위 명령은 모두 패키지의 콘텐츠를 질의합니다. pkg contents 명령은 패키지의 작업과 속성을 표시합니다. pkg search 명령은 질의와 일치하는 패키지를 나열합니다.

## 패키지를 통해 설치된 파일 나열

다음 예제는 pkg contents 명령의 기본 동작을 보여줍니다. 즉, 이 이미지에 설치할 수 있는 각 파일 시스템 객체에 대한 path 속성 값을 표시합니다.



```

$ pkg contents entire
pkg: This package delivers no filesystem content, but may contain metadata. Use
the -o option to specify fields other than 'path', or use the -m option to show
the raw package manifests.
$ pkg contents zip
PATH
usr/bin/zip
usr/bin/zipcloak
usr/bin/zipnote
usr/bin/zipsplit
usr/share/man/man1/zip.1
usr/share/man/man1/zipcloak.1
usr/share/man/man1/zipnote.1
usr/share/man/man1/zipsplit.1

```

pkg contents 명령은 이 이미지에 설치할 수 있는 콘텐츠만 표시합니다. 패키지 매니페스트를 보면(-m 옵션 사용) zip 패키지에 12개의 파일 작업이 있음을 알 수 있습니다. 이 출력에 표시되지 않은 4개 파일은 이 이미지에 설치할 수 없는 파일입니다. 이 이미지는 x86 아키텍처입니다. 표시되지 않은 파일은 SPARC 아키텍처의 /usr/bin 파일 4개입니다. 변형 및 페이지에 대한 자세한 내용은 “선택적 구성 요소의 설치 제어” [74]를 참조하십시오.

## 패키지를 통해 설치된 파일의 속성 표시

경로 외에 설치된 파일에 대한 자세한 정보를 나열하려면 pkg contents 명령에 -t 및 -o 옵션을 사용합니다.

-t 옵션은 표시할 작업의 유형을 지정합니다. 심표로 구분된 목록 형태를 사용하거나 -t 옵션을 여러 번 사용하여 여러 가지 유형을 지정할 수 있습니다.

-o 옵션은 표시할 작업의 속성을 지정합니다. 심표로 구분된 목록 형태를 사용하거나 -o 옵션을 여러 번 사용하여 여러 가지 속성을 지정할 수 있습니다. 패키지 작업 및 속성 목록은 [pkg\(5\)](#) 매뉴얼 페이지를 참조하십시오.

```

$ pkg contents -t file -o owner,group,mode,pkg.size,path zip
OWNER GROUP MODE PKG.SIZE PATH
root bin 0555 231260 usr/bin/zip
root bin 0555 110852 usr/bin/zipcloak
root bin 0555 104960 usr/bin/zipnote
root bin 0555 109340 usr/bin/zipsplit
root bin 0444 86192 usr/share/man/man1/zip.1
root bin 0444 2705 usr/share/man/man1/zipcloak.1
root bin 0444 2396 usr/share/man/man1/zipnote.1
root bin 0444 1837 usr/share/man/man1/zipsplit.1

```

기본적으로 경로를 기준으로 또는 -o 옵션에 지정된 첫번째 속성을 기준으로 출력 결과가 정렬됩니다. -s 옵션을 사용하여 다른 속성을 정렬 키로 지정할 수 있습니다. -s 옵션은 여러 번 지정할 수 있습니다.

## 기타 파일 시스템 객체 및 속성 표시

다음 예제는 지정된 패키지를 통해 설치된 링크의 경로 및 대상을 보여줍니다. pkg(5) 매뉴얼 페이지에 표시된 속성 외에도 여러 의사 속성을 사용할 수 있습니다. 의사 속성 목록은 [pkg\(1\)](#) 매뉴얼 페이지를 참조하십시오.

다음 예제에서 pkg.name 의사 속성은 지정된 작업을 제공하는 패키지의 이름을 보여줍니다. 이 예제에서 Python 2.6.8 및 Python 2.7.3이 설치되었고 명령은 /usr/bin/python 링크에 의존하지 않을 경우 특정 버전을 액세스하는 데 사용할 경로를 보여줍니다. 여러 버전 중개에 대한 자세한 내용은 “[기본 응용 프로그램 구현 지정](#)” [81]을 참조하십시오.

```
$ pkg contents -t link -a path=/usr/bin/python -o path,target,pkg.name
PATH          TARGET      PKG.NAME
usr/bin/python python2.6 runtime/python-26
usr/bin/python python2.7 runtime/python-27
```

## 그룹 패키지의 설치 가능한 모든 패키지 나열

Oracle Solaris는 여러 가지 시스템 설치 그룹 패키지를 제공합니다. Oracle Solaris 11 GUI 설치 프로그램은 solaris-desktop 그룹 패키지를 설치합니다. 텍스트 설치 프로그램 및 자동 설치 프로그램 설치의 기본 SI 매니페스트는 solaris-large-server 그룹 패키지를 설치합니다. 비전역 영역의 기본 설치 매니페스트는 solaris-small-server 그룹 패키지를 설치합니다. solaris-minimal-server 그룹 패키지는 Oracle Solaris를 실행하는 데 필요한 최소 지원 패키지 세트를 설치합니다.

다음 명령을 사용하여 각 그룹에 포함된 패키지 세트를 표시할 수 있습니다.

```
$ pkg contents -Hro fmri -t depend -a type=group solaris-minimal-server
network/ping
service/network/ssh
shell/tcsh
shell/zsh
system/network
```

-t 옵션은 패키지의 depend 작업과 일치합니다. -a 옵션은 group 유형의 depend 작업과 일치합니다. -o 옵션은 그룹 depend 작업의 fmri 속성만 표시합니다. 그룹 패키지는 파일과 같은 콘텐츠를 지정하는 것이 아니라 그룹의 일부인 다른 패키지를 지정한다는 점을 유념하십시오. 그룹 패키지에 대한 자세한 내용은 “[그룹 패키지](#)” [13]를 참조하십시오.

각 패키지에 대한 요약 설명도 표시하려면 pkg list -s 명령을 사용합니다.

```
$ pkg list -Has `pkg contents -Hro fmri -t depend -a type=group solaris-minimal-server`
network/ping          Ping command
service/network/ssh  Secure Shell protocol server
shell/tcsh            Tenex C-shell (tcsh)
shell/zsh            Z Shell (zsh)
system/network       Core Network Infrastructure
```

## 라이선스 요구 사항 표시

다음 예제는 패키지 라이선스에 동의하는 데 필요한 모든 패키지를 표시합니다.

```
$ pkg contents -rt license -a must-accept=true -o license,pkg.name '*'
LICENSE PKG.NAME
BCL     developer/java/jdk-7
BCL     runtime/java/jre-7
LICENSE developer/java/jdk-6
LICENSE library/java/java-demo-6
LICENSE runtime/java/jre-6
lic_OTN consolidation/osnet/osnet-incorporation
lic_OTN install-image/solaris-auto-install
```

라이선스 텍스트를 표시하는 방법에 대한 자세한 내용은 “[패키지 라이선스 표시](#)” [24]를 참조하십시오.

## 패키지 검색

`pkg search` 명령을 사용하여 지정한 패턴과 일치하는 데이터가 포함된 패키지를 검색할 수 있습니다.

## pkg search 및 pkg contents 명령 비교

`pkg contents` 명령과 마찬가지로 `pkg search` 명령은 패키지의 콘텐츠를 검사합니다. `pkg contents` 명령은 콘텐츠를 반환하고 `pkg search` 명령은 검색 질의와 일치하는 패키지의 이름을 반환합니다. 다음 표는 두 명령 사이의 유사점과 차이점을 보여줍니다.

pkg contents	<ul style="list-style-type: none"> <li>■ 설치된 패키지를 검사합니다. <code>-r</code> 옵션을 사용하여 이 이미지에 대해 구성된 모든 게시자와 연관된 저장소의 패키지를 검사합니다.</li> <li>■ <code>-g</code> 옵션을 사용하여 검사할 저장소의 URI를 지정합니다.</li> <li>■ <code>-t</code> 옵션을 사용하여 작업을 지정합니다.</li> <li>■ <code>-a</code> 옵션을 사용하여 속성 및 속성값을 지정합니다.</li> <li>■ <code>-o</code> 옵션을 사용하여 결과 열을 지정합니다.</li> <li>■ <code>-s</code> 옵션을 사용하여 결과를 정렬합니다.</li> </ul>
pkg search	<ul style="list-style-type: none"> <li>■ 이 이미지에 대해 구성된 모든 게시자와 연관된 저장소의 패키지를 검색합니다. <code>-t</code> 옵션을 사용하여 설치된 패키지만 검색합니다.</li> <li>■ <code>-s</code> 옵션을 사용하여 검색할 저장소의 URI를 지정합니다.</li> <li>■ 검색 질의를 사용하여 작업을 지정합니다.</li> <li>■ 검색 질의를 사용하여 속성 및 속성값을 지정합니다.</li> <li>■ <code>-o</code> 옵션을 사용하여 결과 열을 지정합니다.</li> </ul>

---

**작은 정보** - 지정한 패키지의 콘텐츠를 표시하려면 `pkg contents` 명령을 사용하고 질의와 일치하는 패키지를 표시하려면 `pkg search` 명령을 사용합니다. 원하는 콘텐츠를 제공하는 패키지를 확인하려면 `pkg contents` 명령을 사용합니다.

---

## 검색 질의 지정

기본적으로 검색 질의는, 대소문자를 제외하고 정확히 일치하는 일련의 용어입니다. 대소문자 구분 검색을 지정하려면 `-I` 옵션을 사용합니다.

질의어에 `?` 및 `*` 와일드카드를 사용할 수 있습니다. 작은따옴표나 큰따옴표를 사용하여 구문을 검색할 수 있습니다. 와일드카드 또는 따옴표를 사용할 때는 셸을 고려해야 합니다.

질의어를 둘 이상 지정할 수 있습니다. 기본적으로 여러 질의어는 AND로 결합됩니다. OR을 사용하여 두 질의어를 명시적으로 결합할 수 있습니다.

다음과 같은 구조적 형태로 검색 질의를 표현할 수 있습니다.

*package:action:index:token*

<i>package</i>	검색할 패키지의 이름이거나 여러 패키지와 일치하는 패턴입니다.
<i>action</i>	<b>pkg(5)</b> 매뉴얼 페이지의 “Actions” 섹션에 나열된 작업의 이름입니다.
<i>index</i>	<b>pkg(5)</b> 매뉴얼 페이지의 “Actions” 섹션에 나열된 대로 <i>action</i> 속성의 이름입니다.
<i>token</i>	<i>index</i> 값이거나 <i>index</i> 값과 일치하는 패턴입니다.

누락된 필드는 암시적으로 와일드카드로 처리됩니다.

일부 속성은 검색할 수 없습니다. 예를 들어, `mode`는 `file` 작업의 속성이지만 `mode`는 *index*에 대한 유효한 값이 아닙니다.

일부 *index* 값은 다른 속성에서 파생됩니다. 예를 들어 *index*는 `file` 또는 `dir` 작업에 대한 `path` 속성의 마지막 구성 요소인 `basename`일 수 있습니다. 유용한 *index* 값의 예로는 `file` 및 `dir` 작업의 `basename` 및 `path`, `depend` 작업의 종속성 유형(예: `require` 또는 `group`), `driver` 작업의 `driver_name` 및 `alias` 등이 있습니다.

*token* 값은 *index*로 이름이 지정된 속성 값과 비교됩니다. 예를 들어 다음 부분 `driver` 작업에서 `alias`는 *index*에 지정할 수 있는 속성 이름이고 *token*에는 `pci108e*`를 지정할 수 있습니다.

```
driver alias=pci108e,1647 alias=pci108e,16a7
```

`set` 작업의 구문은 약간 다릅니다. `set` 작업의 두 가지 속성은 `name`과 `value`입니다. 이 경우 *index* 값은 `name` 속성 값이고 *token* 값은 일치하는 `value` 속성 값과 비교됩니다. 다음 예제는 드라이버 패키지에 대한 부분 `set` 작업을 보여줍니다.

```
set name=pkg.summary value="Broadcom NetXtreme II 10GbE NIC Driver"
```

다음 예제는 *action*에 *set*, *index*에 *pkg.summary*, *token*에 *Broadcom*을 지정합니다. *search.match* 및 *pkg.name* 열 지정자는 의사 속성입니다. [pkg\(1\)](#) 매뉴얼 페이지를 참조하십시오.

```
$ pkg search -o search.match, pkg.name pkg.summary:Broadcom
SEARCH.MATCH                                PKG.NAME
Broadcom NetXtreme II 10GbE NIC Driver      driver/network/ethernet/bnxe
Broadcom 57xx 1GbE NIC Driver               driver/network/ethernet/bge
Broadcom NetXtreme II 1GbE NIC Driver       driver/network/ethernet/bnx
Broadcom BCM4401 NIC Driver                 driver/network/ethernet/bfe
Broadcom HT1000 SATA driver                 driver/storage/bcm_sata
```

몇몇 제대로 정의된 *set* 작업 *name* 속성 값에는 *pkg.fmri*, *info.classification*, *pkg.description*, *pkg.summary*가 있습니다. [pkg\(5\)](#) 매뉴얼 페이지의 “Set Actions”를 참조하십시오.

기본적으로 현재 설치된 패키지 또는 최신 패키지 버전에 대해서만 일치하는 항목이 표시됩니다. 일치하는 버전을 모두 표시하려면 *-f* 옵션을 사용합니다.

기본적으로 일치하는 모든 작업에 대해 결과가 표시되므로 한 패키지에 대해 여러 행의 결과가 생성될 수 있습니다. 일치하는 각 패키지를 한 번만 나열하려면 *-p* 옵션을 사용합니다.

## 지정한 파일을 제공하는 패키지 식별

다음 예에서는 *libpower* 라이브러리가 *system/kernel/power* 패키지에 있음을 보여줍니다.

```
$ pkg search -Hlo pkg.name /lib/libpower.so.1
system/kernel/power
$ pkg search -lo path, pkg.name libpower.so.1
PATH                                PKG.NAME
lib/libpower.so.1 system/kernel/power
$ pkg search -Hlo path, pkg.name basename:libpower.so.1
lib/libpower.so.1 system/kernel/power
$ pkg search -Hlo path, pkg.name 'path:*libpower.so.1'
lib/libpower.so.1 system/kernel/power
```

## 지정한 SMF 서비스를 제공하는 패키지 식별

특정 SMF 서비스를 제공하는 패키지를 표시하려면 *org.opensolaris.smf.fmri* 속성의 값인 서비스 이름을 검색합니다.

```
$ pkg search -o pkg.name, search.match 'org.opensolaris.smf.fmri:*network/http*'
PKG.NAME                                SEARCH.MATCH
```

```

web/java-servlet/tomcat svc:/network/http
web/proxy/squid         svc:/network/http
web/proxy/privoxy      svc:/network/http
web/server/lighttpd-14 svc:/network/http
web/server/apache-22   svc:/network/http
web/server/apache-22   svc:/network/http:apache22
web/server/lighttpd-14 svc:/network/http:lighttpd14
web/proxy/privoxy      svc:/network/http:privoxy
web/proxy/squid        svc:/network/http:squid
web/java-servlet/tomcat svc:/network/http:tomcat6

```

이 경우 각 속성의 값은 두 개로, 인스턴스 이름이 지정된 서비스 이름과 지정되지 않은 서비스 이름입니다. 다음 예에서는 패키지 매니페스트에서 이 속성을 지정하는 방법을 보여줍니다.

```
set name=org.opensolaris.smf.fmri value=svc:/network/http value=svc:/network/http:apache22
```

다음 예제는 한 번만 나열된 각 패키지에 이와 동일한 정보를 보여줍니다. 콜론 문자는 이스케이프 처리하므로 다른 검색 질의 필드가 아닌 *token*의 일부로 해석됩니다.

```

$ pkg search -o pkg.name,search.match 'org.opensolaris.smf.fmri:*network/http\:*'
PKG.NAME          SEARCH.MATCH
web/server/apache-22   svc:/network/http:apache22
web/server/lighttpd-14 svc:/network/http:lighttpd14
web/proxy/privoxy     svc:/network/http:privoxy
web/proxy/squid       svc:/network/http:squid
web/java-servlet/tomcat svc:/network/http:tomcat6

```

## 분류 또는 범주별로 패키지 나열

다음 예제는 `info.classification` 속성 값에 “Source Code Management”가 있는 모든 설치된 패키지를 식별합니다.

```

$ pkg search -Hlo pkg.shortfmri info.classification:'source code management'
pkg:/developer/versioning/sccs@0.5.11-0.175.2.0.0.8.0
pkg:/developer/versioning/git@1.7.9.2-0.175.2.0.0.34.0
pkg:/developer/versioning/mercurial-27@2.2.1-0.175.2.0.0.34.0

```

다음 예제는 이 검색으로 일치된 패키지 메타 데이터를 보여줍니다.

```
set name=info.classification value="org.opensolaris.category.2008:Development/Source Code Management"
```

이 정보는 `pkg info` 명령 출력에서 “Category” 행에 표시됩니다.

기타 검색할 수 있는 분류는 [“Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”](#)의 [“Classification Values”](#)을 참조하십시오.

다음 예제와 같이 패키지 이름의 구성 요소 중 하나에 추측으로 `pkg list` 명령을 사용할 수도 있습니다.

```
$ pkg list '*storage*'
$ pkg list -a '*database*'
```

## 종속 패키지 표시

아래의 예에서는 지정한 패키지에 종속된 패키지를 보여줍니다.

다음 예제는 system/kernel/power 패키지에 대해 require 종속성을 갖는 패키지를 보여줍니다.

```
$ pkg search -Hlo pkg.name require:system/kernel/power
system/kernel/dynamic-reconfiguration/i86pc
system/hal
```

다음 pkg contents 명령은 검색 결과를 확인합니다. 요청된 출력 action.raw는 패키지 매니페스트에 나타난 대로 정확히 작업을 표시하는 의사 속성입니다.

```
$ pkg contents -rt depend -a fmri='*power*' -o pkg.name,action.raw i86pc system/hal
PKG.NAME                                ACTION.RAW
system/hal                               depend fmri=pkg:/system/kernel/power
@0.5.11-0.175.2.0.0.34.0 type=require variant.opensolaris.zone=global
system/kernel/dynamic-reconfiguration/i86pc depend fmri=pkg:/system/kernel/power
type=require
```

다음 예제는 많은 패키지에 pkg:/x11/server/xorg@1.14.99에 대한 exclude 종속성이 있음을 보여줍니다.

```
$ pkg search -lo pkg.name, fmri 'depend:exclude:*xorg*'
PKG.NAME                                FMRI
x11/server/xorg/driver/xorg-video-ati    pkg:/x11/server/xorg@1.14.99
x11/server/xvnc                           pkg:/x11/server/xorg@1.14.99
x11/server/xserver-common                 pkg:/x11/server/xorg@1.14.99
x11/server/xorg/driver/xorg-input-vmouse  pkg:/x11/server/xorg@1.14.99
x11/server/xephyr                         pkg:/x11/server/xorg@1.14.99
...
```

## 그룹 패키지의 모든 패키지 나열

Oracle Solaris 11 GUI 설치 프로그램은 solaris-desktop 그룹 패키지를 설치합니다. 텍스트 설치 프로그램 및 자동 설치 프로그램 설치의 기본 SI 매니페스트는 solaris-large-server 그룹 패키지를 설치합니다. 비전역 영역의 기본 설치 매니페스트는 solaris-small-server 그룹 패키지를 설치합니다. solaris-minimal-server 그룹 패키지는 Oracle Solaris를 실행하는 데 필요한 최소 지원 패키지 세트를 설치합니다. 다음 검색 폼을 사용하여 각 그룹에 포함된 패키지 세트를 표시할 수 있습니다.

```
$ pkg search -Ho fmri '*/solaris-minimal-server:depend:group:*
```

```
network/ping
service/network/ssh
shell/tcsh
shell/zsh
system/network
```

이 예제에서 `-o pkg.name`은 질의의 *package* 필드에 지정된 패키지의 이름만 반환합니다.

```
group/system/solaris-minimal-server
```

`-o fmri` 옵션은 `solaris-minimal-server` 패키지에서 `group` 유형 종속성으로 지정된 패키지의 `fmri` 속성 값을 반환합니다.

`pkg search` 명령은 지정된 패키지에서 작업의 속성 값을 반환합니다. 이 예에서는 속성 값이 패키지 이름입니다. 이 명령의 결과 수가 [“그룹 패키지의 설치 가능한 모든 패키지 나열” \[26\]](#)에 표시된 비슷한 `pkg contents` 명령의 결과 수보다 많을 수 있습니다. 이러한 검색 결과에는 설치 가능한 패키지뿐 아니라 지정된 패키지에서 `group` 유형의 `depend` 작업에 지정된 모든 패키지의 이름이 포함되기 때문입니다. 예를 들어, 이 이미지에 설치할 수 없는 패키지 변형과 페이지가 포함될 수도 있습니다. 이 차이점을 보려면 양쪽 예제를 `solaris-large-server` 패키지로 실행해 보십시오.



# ◆◆◆ 3 장 3

## 소프트웨어 패키지 설치 및 업데이트

---

패키지 설치 및 업데이트는 일부 패키지를 특정 버전으로 제한하거나 게시자 검색 순서를 구성하거나 패키지 서명 등록 정보를 설정하는 등의 이미지 구성의 영향을 받습니다. 이미지 구성에 대한 설명은 [5장. 설치된 이미지 구성](#)을 참조하십시오.

이미 설치된 패키지, 설치 가능한 패키지 및 업데이트가 있는 패키지를 확인하는 방법은 [2장. 소프트웨어 패키지 정보 보기](#)를 참조하십시오.

이 장에서는 다음 작업을 수행하는 방법을 보여줍니다.

- 시험 설치를 실행하여 설치가 성공할지 여부 및 어떤 항목이 설치되는지를 확인합니다.
- 패키지를 설치, 업데이트 및 제거합니다.
- 패키지를 검증합니다.
- 설치된 패키지의 문제를 수정합니다.
- 설치된 파일을 원래 콘텐츠로 복원합니다.
- 패키지를 제거합니다.

“[비전역 영역 작업](#)” [49]에서는 Oracle Solaris 영역에 고유한 패키지 작업의 측면에 대해 설명합니다.

패키지 설치, 업데이트 및 제거 작업을 수행하려면 높은 수준의 권한이 필요합니다. 자세한 내용은 “[설치 권한](#)” [17]을 참조하십시오.

이 장에 설명된 명령에 대한 모든 옵션의 전체 목록은 [pkg\(1\)](#) 매뉴얼 페이지를 참조하십시오.

### 작업 미리 보기

이 장과 [5장. 설치된 이미지 구성](#)에 나오는 대부분의 명령에는 실제로 변경하지 않고 명령으로 어떤 작업이 수행될지 미리 보는 데 사용할 수 있는 `-n` 옵션이 있습니다.

---

작은 정보 - 가능하면 항상 `-n` 옵션을 사용하는 것이 좋습니다. `-n` 옵션과 하나 이상의 상세 정보 표시 옵션(`-nv`, `-nvv`)을 함께 사용하여 먼저 명령의 결과를 검토한 후 `-n` 옵션을 생략하고 명령을 실행할 수 있습니다.

---

다음 예에서는 실제로 수행되지 않은 패키지 설치에 대한 정보를 보여줍니다.

```
$ pkg install -nv group/feature/amp
    Packages to install:      6
    Mediators to change:     1
    Services to change:      2
    Estimated space available: 22.70 GB
    Estimated space to be consumed: 751.08 MB
    Create boot environment:  No
    Create backup boot environment:  No
    Rebuild boot archive:    No

Changed mediators:
  mediator mysql:
    version: None -> 5.1 (system default)

Changed packages:
solaris
  database/mysql-51
    None -> 5.1.37,5.11-0.175.2.0.0.34.0:20140303T160611Z
  database/mysql-common
    None -> 5.11,5.11-0.175.2.0.0.34.0:20140303T161628Z
  group/feature/amp
    None -> 0.5.11,5.11-0.175.2.0.0.33.0:20140217T134747Z
  web/server/apache-22/module/apache-dtrace
    None -> 0.3.1,5.11-0.175.2.0.0.34.0:20140303T175456Z
  web/server/apache-22/module/apache-fcgid
    None -> 2.3.9,5.11-0.175.2.0.0.34.0:20140303T175502Z
  web/server/apache-22/module/apache-php5
    None -> 5.2.17,5.11-0.175.1.0.0.18:20120611T210317Z
Services:
  restart_fmri:
    svc:/system/manifest-import:default
    svc:/system/rbac:default
```

이 출력에서 이 설치 작업은 새 BE가 아닌 현재 BE에서 수행되고 현재 BE의 백업을 만들지 않습니다. 새 BE나 백업 BE에 필요한 옵션이나 이미지 등록 정보를 지정할 수 있습니다. “Changed packages” 섹션에서 amp 그룹 패키지가 설치되고 해당하는 5개 그룹 종속성이 설치됨을 알 수 있습니다. 출력은 각 패키지의 어떤 버전이 설치될지 보여줍니다. 토큰 None은 이러한 패키지가 현재 설치되지 않았으므로 업데이트되지 않음을 나타냅니다.

다음 명령은 대량의 출력 결과를 생성하는데 이는 영향을 받는 패키지가 너무 많기 때문입니다. 이 페이지를 설정하면 모든 패키지의 모든 현지화된 콘텐츠가 설치됩니다. 이 미리 보기 명령을 실행하면 이 작업 일정을 잡는 방법과 새 로케일을 덜 추가할지 여부에 대한 결정을 바꿀 수 있습니다. 이 출력에서 새 BE는 기본적으로 만들지 않지만 백업 BE가 만들어집니다.

```
$ pkg change-facet -nv 'facet.locale.*=true'
    Packages to change:      130
    Variants/Facets to change:  1
    Estimated space available: 22.70 GB
    Estimated space to be consumed: 3.45 GB
    Create boot environment:  No
    Create backup boot environment:  Yes
```

```

Rebuild boot archive:      No
Changed variants/facets:
  facet locale.* (local): False -> True
Changed packages:
solaris
...

```

## 패키지 설치 및 업데이트

다음 표는 `pkg install` 및 `pkg update` 명령 사이의 유사점과 차이점을 보여줍니다.

<code>pkg install</code>	<ul style="list-style-type: none"> <li>■ 하나 이상의 패키지 이름을 피연산자로 요구합니다.</li> <li>■ 현재 설치되지 않은 패키지를 설치합니다.</li> <li>■ 이미 설치된 패키지를 업데이트합니다.</li> <li>■ 패키지를 다운그레이드하지 않습니다. 설치된 패키지보다 낮은 버전을 지정하면 시스템은 해당 패키지를 설치하지 않습니다.</li> </ul>
<code>pkg update</code>	<ul style="list-style-type: none"> <li>■ 이미 설치된 패키지에서 0개 이상의 이름을 피연산자로 취합니다.</li> <li>■ 설치된 패키지를 업데이트합니다.</li> <li>■ 패키지 이름을 지정하지 않거나 '*'를 지정하면 이미지에 설치된 모든 패키지가 업데이트됩니다.</li> <li>■ 설치된 패키지를 FMRI에 지정된 버전으로 다운그레이드합니다.</li> <li>■ 아직 설치되지 않은 패키지를 설치하지 않습니다. 아직 설치되지 않은 패키지를 지정하면 시스템은 해당 패키지를 설치하지 않습니다.</li> </ul>

[pkg\(5\)](#) 매뉴얼 페이지에서 `file` 작업의 `preserve` 및 `overlay` 속성을 참조하여 설치 및 업데이트 중 이러한 속성을 가진 파일이 어떻게 처리되는지 확인할 수 있습니다.

## 공통 설치 옵션

이 절에서는 여러 설치 관련 명령에 공통되는 옵션을 설명합니다. 중개자 설정 또는 설정 해제, 변형 또는 페이지 변경, 패키지 수정, 파일 되돌리기 등에도 패키지 설치, 업데이트 또는 제거가 포함될 수 있습니다.

## 부트 환경 옵션

패키지를 설치, 업데이트 또는 제거할 때 새 BE나 백업 BE가 자동으로 만들어질 수 있습니다. BE 관련 이미지 정책이 허용하는 제약 조건 내에서 아래에 설명된 옵션을 사용하여 새 BE 및 백업 BE 생성을 제어할 수 있습니다. 새 BE와 백업 BE 및 BE 관련 이미지 정책 설정에 대한 자세한 내용은 [“부트 환경 정책 이미지 등록 정보” \[86\]](#)를 참조하십시오.

다음 BE 옵션을 사용하여 새 BE 또는 백업 BE를 강제로 만들거나 만들지 않도록 설정하고, BE에 사용자 정의 이름을 제공하고, 새 BE가 활성화되지 않도록 지정할 수 있습니다. 이 옵션은 `install`, `exact-install`, `uninstall`, `update`, `revert`, `set-mediator`, `unset-mediator`, `change-variant`, `change-facet` 하위 명령에 사용할 수 있습니다.

`--no-be-activate`

BE가 생성된 경우 해당 BE가 다음 부트 시 활성 BE가 되도록 설정하지 않습니다.

명령 출력에서 새 부트 환경이 생성되었다는 메시지를 확인하십시오. 새 부트 환경이 만들어지고 활성화된 경우 `--no-be-activate` 옵션을 지정하지 않으면 다음 재부트 시 기본적으로 해당 BE가 부트됩니다.

[beadm\(1M\)](#) 명령을 사용하여 `pkg` 명령과 별도로 활성 BE를 표시하고 변경할 수 있습니다.

`--no-backup-be`

백업 BE를 만들지 않습니다.

`--require-backup-be`

새 BE가 생성되지 않을 경우 백업 BE를 만듭니다. 이 옵션을 생략할 경우 백업 BE가 생성되는지 여부는 이미지 정책에 따라 다릅니다. 백업 BE가 자동으로 생성되는 경우에 대한 설명은 [“부트 환경 정책 이미지 등록 정보” \[86\]](#)를 참조하십시오.

`--backup-be-name name`

백업 BE가 생성되면 기본 이름 대신 `name`이라는 이름을 지정합니다. `--backup-be-name`을 사용하면 `--require-backup-be`가 지정된 것을 의미합니다.

`--deny-new-be`

새 BE를 만들지 않습니다. 새 BE가 필요한 경우 설치, 업데이트, 제거 또는 되돌리기 작업이 수행되지 않습니다.

`--require-new-be`

새 BE를 만듭니다. 이 옵션을 생략할 경우 BE가 생성되는지 여부는 이미지 정책에 따라 다릅니다. BE가 자동으로 생성되는 경우에 대한 설명은 [“부트 환경 정책 이미지 등록 정보” \[86\]](#)를 참조하십시오. 이 옵션은 `--require-backup-be`와 함께 사용할 수 없습니다.

`--be-name name`

BE가 생성되면 기본 이름 대신 `name`이라는 이름을 지정합니다. `--be-name`을 사용하면 `--require-new-be`가 지정된 것을 의미합니다. 이 옵션을 사용하는 것이 작업을 수행하는 가장 안전한 방법입니다.

## 비전역 영역에서 작동하는 옵션

[“비전역 영역 작업” \[49\]](#)에 설명된 대로, 전역 영역에서 수행된 일부 패키지 설치, 제거, 업데이트 작업만 자동으로 비전역 영역에 영향을 미칩니다. `-r` 옵션은 전역 영역에서 입력

한 것과 동일한 pkg 작업을 비전역 영역에서 수행하므로 -r을 사용하지 않았을 때보다 더 많은 패키지에 영향을 미칠 수 있습니다. 이 옵션은 install, uninstall, update, change-variant, change-facet 하위 명령에 사용할 수 있습니다.

-r

이 작업은 전역 영역과 모든 설치된 solaris 브랜드 비전역 영역에서도 실행합니다. 비전역 영역에 미치는 영향은 각 비전역 영역에 로그인하여 직접 명령을 실행하는 것과 비슷합니다.

이 옵션을 사용하지 않으면 전역 영역에서 pkg 명령을 실행할 때 “비전역 영역 작업” [49]에 설명된 대로 비전역 영역은 전역 영역과 호환성 유지에 필요한 정도까지만 수정됩니다. 이 옵션을 사용하면 pkg 작업이 -z 및 -Z 옵션으로 제한된 경우를 제외하고 모든 설치된 비전역 영역에 적용됩니다. -z 및 -Z 옵션으로 제외된 영역은 전역 영역과 동기화를 유지하기 위해 업데이트가 필요한 경우 계속 수정될 수 있습니다.

-z zone

이 작업은 지정된 비전역 영역에서만 실행합니다. -z 옵션은 여러 번 지정할 수 있습니다. -z 옵션은 -r 옵션과 함께 사용해야 합니다. -z 옵션은 -Z 옵션과 함께 사용할 수 없습니다.

-Z zone

이 작업은 지정된 영역을 제외한 모든 비전역 영역에서 실행합니다. -Z 옵션은 여러 번 지정할 수 있습니다. -Z 옵션은 -r 옵션과 함께 사용해야 합니다. -Z 옵션은 -z 옵션과 함께 사용할 수 없습니다.

다음 옵션은 전역 영역과 동시에 업데이트할 비전역 영역 수를 지정합니다. 이 옵션은 install, exact-install, uninstall, update, change-variant, change-facet 하위 명령에 사용할 수 있습니다.

-c n

최대  $n$ 개의 설치된 solaris 브랜드 비전역 영역을 전역 영역과 병렬로 업데이트합니다.  $n$ 이 0 또는 음수이면 모든 비전역 영역이 전역 영역과 동시에 업데이트됩니다.

환경 변수 PKG\_CONCURRENCY를  $n$  값으로 설정할 수도 있습니다. -c 옵션은 PKG\_CONCURRENCY 설정을 대체합니다. -c 옵션이 지정된 경우 PKG\_CONCURRENCY는 무시됩니다.

## 서비스 작업 옵션

패키지를 설치하거나 업데이트할 때 지정된 서비스를 다시 시작하거나 새로 고치는 등의 SMF 서비스 작업을 지정할 수 있습니다. 다수의 패키지에 작업하는 경우 모든 서비스 작업을 마치기 전에 pkg 작업이 끝날 수 있습니다. 그러면 연관된 서비스를 아직 이용할 수 없기 때문에 새로 설치된 소프트웨어를 사용할 수 없습니다.

이 문제를 피하려면 다음 옵션 중 하나를 사용하여 pkg 명령과 동시에 SMF 액추에이터를 실행하십시오. 이 옵션은 install, uninstall, update, change-variant, change-facet 하위 명령에 사용할 수 있습니다.

--sync-actuators

이 옵션을 지정하면 pkg 명령이 실행된 영역(전역 영역 또는 비전역 영역)에서 모든 SMF 액추에이터를 마칠 때까지 pkg 명령이 반환되지 않습니다.

--sync-actuators-timeout *timeout*

이 옵션을 지정하면 모든 SMF 액추에이터를 마치거나 *timeout* 기간에 도달할 때까지(둘 중 짧은 쪽) pkg 명령이 반환되지 않습니다. 주어진 *timeout* 시간(초) 내에 액추에이터를 마치지 않으면 pkg 명령이 계속 작동하고 반환 코드 8로 종료됩니다.

## 라이선스 옵션

패키지를 설치하거나 업데이트하기 전에 라이선스 동의를 필요할 수 있습니다. 다음 옵션을 사용하여 필요한 라이선스를 확인하고 동의할 수 있습니다. 이 옵션은 install, exact-install, update, fix, change-variant, change-facet 하위 명령에 사용할 수 있습니다.

--licenses

--licenses 옵션을 사용하여 이 작업의 일부로 설치되거나 업데이트되는 패키지의 모든 라이선스를 표시합니다. 이 작업을 계속하기 위해 동의가 필요한 라이선스뿐만 아니라 모든 패키지의 라이선스가 표시됩니다. 작업을 계속하려면 라이선스 동의가 필요한 경우 --licenses 옵션을 지정하지 않더라도 해당 라이선스가 표시됩니다. 다른 작업을 시작하지 않고 패키지의 라이선스를 보려면 “[패키지 라이선스 표시](#)” [24]에 표시된 대로 pkg list 명령을 사용합니다. 동의가 필요한 라이선스 목록을 표시하려면 “[라이선스 요구 사항 표시](#)” [27]에 표시된 대로 pkg info 명령을 사용합니다.

--accept

--accept 옵션을 사용하여 업데이트되거나 설치된 패키지의 라이선스 계약 조건에 동의함을 나타냅니다. 이 옵션을 제공하지 않고 동의해야 할 패키지 라이선스가 있을 경우 필요한 라이선스가 표시되고 설치 작업이 실패합니다.

## 기타 설치 옵션

--no-index

기본적으로 패키지를 설치, 업데이트, 제거할 때 검색 색인이 업데이트됩니다. --no-index 옵션을 지정하면 이러한 작업이 성공적으로 완료된 후 검색 색인이 업데이트되지 않습니다. 패키지를 다수 설치할 경우 이 옵션을 지정하면 상당 시간을 절약할 수 있습니다. 모든 설치, 업데이트, 제거 작업이 완료되면 pkg refresh를 사용하여 지정한 각 게시자의 게시자 메타 데이터 및 사용 가능한 패키지 목록을 업데이트할 수 있습니다. 게시자를 지정하지 않으면 모든 게시자에 대해 새로 고침 작업이 수행됩니다. 이 옵션은 install, exact-install, uninstall, update 하위 명령에 사용할 수 있습니다.

```
--no-refresh
```

--no-refresh 옵션을 지정하면 사용 가능한 패키지 및 기타 메타 데이터의 최신 목록을 검색하기 위해 이미지 게시자 저장소에 연결하지 않습니다. 이 옵션은 `install`, `exact-install`, `update` 하위 명령에 사용할 수 있습니다.

## 새 패키지 설치

기본적으로 나머지 이미지와 호환되는 패키지의 가장 최신 버전이 게시자 검색 순서의 패키지를 제공한 첫번째 게시자에서 설치됩니다. 최신 버전을 명시적으로 요청하려면 패키지 FMRI의 버전 부분에 `latest`를 사용합니다.

패키지가 이미 설치되어 있으면 나머지 이미지와 호환되는 패키지의 가장 최신 버전을 현재 설치된 버전을 제공한 게시자에서 설치하여 패키지가 업데이트됩니다.

여러 개의 패키지가 지정되었지만 지정된 패키지 중 하나라도 이 이미지에 설치할 수 없으면 지정된 패키지 중 아무것도 설치되지 않습니다.

패키지가 무시 목록에 있을 경우 패키지를 설치하면 해당 목록에서 패키지가 제거됩니다. 무시 목록에 대한 자세한 내용은 [“그룹 패키지의 일부 패키지 설치 방지” \[83\]](#)를 참조하십시오.

## 설치 가능한 패키지 식별 및 지정

이미지에 대해 사용으로 설정된 게시자가 둘 이상인 경우 게시자 고착성과 검색 순서를 설정하거나 패키지 FMRI에 게시자를 지정하여 패키지를 제공하는 게시자를 제어할 수 있습니다. 설치할 버전도 패키지 FMRI에 지정할 수 있습니다. 패키지 FMRI에 대한 설명은 [“FMRI\(Fault Management Resource Identifier\)” \[13\]](#)를 참조하십시오. 게시자 고착성 및 검색 순서 설정에 대한 자세한 내용은 [“게시자 구성” \[67\]](#)을 참조하십시오.

패키지 이름에 게시자를 지정하지 않으면 일치하는 패키지를 제공하는 첫번째 게시자가 설치 소스로 사용됩니다. 해당 게시자가 이 이미지에 설치할 수 있는 패키지 버전을 제공하지 않으면 설치 작업이 실패합니다. 이 이미지에 설치할 수 있는 패키지 버전을 제공하는 게시자를 확인하려면 `pkg list -a` 명령을 사용합니다.

다음 명령은 구성된 게시자에서 `atool` 패키지의 설치할 수 있는 버전을 사용할 수 있지만 검색 순서의 처음에 있는 게시자가 이 이미지에 설치할 수 없는 버전임을 보여줍니다. `pkg list` 명령 옵션에 대한 자세한 내용은 [“패키지 설치 상태 정보 표시” \[19\]](#)를 참조하십시오.

```
$ pkg list -a atool
NAME (PUBLISHER)    VERSION    IFO
atool (isvpub)      2.0       ---
$ pkg list -af atool
NAME (PUBLISHER)    VERSION    IFO
```

```
atool          1.1      ---
atool (isvpub) 2.0      ---
```

이 경우 다음 설치 명령은 실패합니다. 패키징 시스템이 검색 순서의 처음에 있는 게시자에서 패키지 이름 atool의 일치 항목을 찾지만 해당 패키지는 설치할 수 없습니다.

```
$ pkg install atool
```

이 패키지를 설치하려면 다음 예제에 표시된 대로 패키지 이름을 더 구체적으로 지정해야 합니다.

```
$ pkg install //isvpub/atool
$ pkg install atool@2.0
```

-nv 옵션을 사용하여 실제 설치를 수행하기 전에 설치될 항목을 확인합니다. 오류 메시지가 나타나면 [부록 A. 패키지 설치 및 업데이트 문제 해결](#)을 참조하십시오.

## 패키지의 소스 지정

지정한 패키지 저장소나 패키지 아카이브를 이미지의 임시로 패키지 데이터를 검색할 소스 목록에 추가하려면 -g 옵션을 사용합니다. SSL 인증서가 필요한 저장소는 이 옵션과 함께 사용할 수 없습니다. 자식 이미지가 있는 이미지(비전역 영역)에서는 이 옵션을 사용할 수 없습니다. 이 이미지에 비전역 영역이 설치되는 경우 pkg set-publisher 명령을 사용하여 이 게시자 및 원본을 추가합니다. 이 옵션은 여러 번 지정할 수 있습니다.

-g 옵션을 지정하면 패키지를 검색할 때 이미지에서 사용으로 설정된 게시자가 우선합니다.

- 지정한 패키지 이름 또는 패키지 이름 패턴과 일치하는 패키지를 이미지에 사용으로 설정된 게시자에서 사용할 수 있고 동일한 게시자가 -g 옵션에 지정한 위치에 없는 경우 패키징 시스템은 이미지에 사용으로 설정된 게시자에서 패키지를 설치하려고 합니다. install 또는 update 후 이미지에 구성되지 않은 게시자가 제공한 패키지는 원본 없이 이미지 구성에 추가됩니다.
- 지정한 패키지 이름 또는 패키지 이름 패턴과 일치하는 패키지를 이미지에 사용으로 설정된 게시자에서 사용할 수 있고 동일한 게시자가 -g 옵션에 지정한 위치에 패키지를 게시하는 경우 패키징 시스템은 -g 옵션에 지정된 위치에서 패키지를 설치하려고 합니다.

다음 예제에서 btool 패키지는 이미지에 구성된 solaris 게시자에서 사용할 수 있습니다. btool 패키지는 저장소 원본이 http://pkg.example1.com/인 devtool 게시자에서도 사용할 수 있지만 devtool 게시자는 이미지에 구성되어 있지 않습니다. 구성된 게시자에서 패키지를 사용할 수 있는 경우 이미지에 구성된 게시자가 -g 소스보다 우선하므로 명령은 solaris 게시자에서 패키지를 설치하려고 합니다.

```
$ pkg install -g http://pkg.example1.com/ btool
```

devtool 게시자에서 패키지를 설치하려면 패키지 이름에 게시자 이름을 지정합니다.

```
$ pkg install -g http://pkg.example1.com/ //devtool/btool
```



다음 예제에서 isvpub는 이미지에 구성되어 있고 원본이 /export/IPSpkgrepos/isvrepo인 게시자입니다. isvpub 게시자도 패키지를 http://pkg.example2.com/의 저장소에 게시하지만 해당 원본은 이미지에 구성된 게시자에 지정되어 있지 않습니다. 다음 명령은 동일한 게시자가 두 위치 모두에서 패키지를 제공하므로 http://pkg.example2.com/ 위치에서 패키지를 설치하려고 합니다.

```
$ pkg install -g http://pkg.example2.com/ atool
```

“패키지 게시자 추가, 수정 또는 제거” [69]에서 게시자 고착성에 대한 설명도 참조하십시오.

## 새 부트 환경에 패키지 설치

작은 정보 - 가장 안전한 설치 또는 업데이트 방법은 새 BE를 명시적으로 지정하는 것입니다. BE 생성 시점에 대한 자세한 내용은 “부트 환경 정책 이미지 등록 정보” [86]를 참조하십시오.

새 BE는 지정한 설치, 제거 또는 업데이트 변경 사항이 적용된 현재 BE의 복제본입니다. 현재 BE는 수정되지 않습니다. 시스템은 자동으로 다시 시작되지 않습니다. 새 BE는 다음에 시스템을 다시 시작할 때 기본적으로 선택되는 부트 옵션이지만 현재 BE로 부트하는 것도 가능합니다.

--no-be-activate 옵션을 지정할 경우 다음 재부트 시 새로운 BE가 기본 부트 옵션으로 선택되지 않습니다.

--be-name 옵션을 사용하여 새 BE가 강제로 만들어지도록 설정하거나 새 BE가 기본적으로 만들어지는 경우 새 BE에 의미 있는 이름을 지정합니다.

“작업 미리 보기” [33]의 예제에서는 group/feature/amp 패키지를 설치할 때 새 BE가 기본적으로 만들어지지 않도록 설정했습니다. 다음 부분 출력에서는 --be-name 옵션이 지정되었으므로 새 BE가 만들어집니다.

```
$ pkg install -v --be-name s1lamp group/feature/amp
Packages to install:      6
Mediators to change:    1
Estimated space available: 22.70 GB
Estimated space to be consumed: 751.08 MB
Create boot environment: Yes
Activate boot environment: Yes
Create backup boot environment: No
Rebuild boot archive:   No
```

다음 메시지가 설치 작업의 끝에 표시됩니다.

```
A clone of s11 exists and has been updated and activated.
On the next boot the Boot Environment s1lamp will be
```

mounted on '/'. Reboot when ready to switch to this updated BE.

pkg list 명령은 현재 BE에 group/feature/amp 패키지가 설치되어 있지 않기 때문에 group/feature/amp 패키지가 설치되지 않았음을 보고합니다. group/feature/amp 패키지는 새 s11amp BE에 설치되어 있습니다.

```
$ pkg list group/feature/amp
pkg list: no packages matching 'group/feature/amp' installed
```

시스템에 s11amp라는 새로운 활성 BE가 있는지 확인하려면 beadm list 명령을 사용합니다. “N” BE가 현재 부트되어 있고 “R” BE가 재부트 시 기본값입니다. 재부트 시의 기본 BE를 변경하려면 beadm activate 명령을 사용합니다.

```
$ beadm list
BE           Active Mountpoint Space   Policy Created
--           -
s11          N      /           30.92M static 2014-03-05 08:51
s11amp       R      -           25.75G static 2014-03-26 10:45
```

group/feature/amp 패키지가 새 BE에 설치되어 있는지 확인합니다. 새 BE를 마운트하고 -R 옵션을 사용하여 마운트된 BE에서 작동합니다. l 열에 있는 "i"는 group/feature/amp 패키지가 설치되었다는 것을 나타냅니다.

```
$ beadm mount s11amp /mnt
$ beadm list
BE           Active Mountpoint Space   Policy Created
--           -
s11          N      /           30.92M static 2014-03-05 08:51
s11amp       R      /mnt        25.75G static 2014-03-26 10:45
$ pkg -R /mnt list group/feature/amp
NAME (PUBLISHER)      VERSION          IFO
group/feature/amp    0.5.11-0.175.2.0.0.33.0  i--
```

s11amp BE를 반드시 마운트 해제해야 합니다.

```
$ beadm unmount s11amp
```

## 패키지 거부

pkg install 명령의 --reject 옵션을 사용하여 지정된 패키지가 설치되는 것을 막을 수 있습니다. 일치하는 패키지가 이미 설치되어 있으면 이 작업의 일부로 제거됩니다.

그룹 종속성을 가진 거부된 패키지는 무시 목록에 놓입니다. 무시 목록에 대한 자세한 내용은 [“그룹 패키지의 일부 패키지 설치 방지” \[83\]](#)를 참조하십시오.

다음 예제는 cvs 패키지를 제외한 developer-gnu 패키지의 모든 그룹 종속성을 설치합니다.

```
$ pkg install --reject developer/versioning/cvs group/feature/developer-gnu
```

## 패키지 업데이트

`install` 또는 `update` 하위 명령을 사용하여 설치된 패키지를 나머지 이미지와 호환되는 패키지의 최신 버전으로 업데이트할 수 있습니다. 이때 현재 설치된 버전을 제공한 게시자를 사용하여 업데이트가 이루어집니다. `pkg update` 명령을 사용하여 패키지를 업데이트하면 아직 설치되지 않은 패키지를 실수로 설치하는 것을 방지할 수 있습니다.

이미지에 대해 사용으로 설정된 게시자가 둘 이상인 경우 게시자 고착성과 검색 순서를 설정하거나 패키지 FMRI에 게시자를 지정하여 패키지를 제공하는 게시자를 제어할 수 있습니다. 설치할 버전도 패키지 FMRI에 지정할 수 있습니다. 패키지의 최신 버전을 명시적으로 요청하려면 패키지 이름의 버전 부분에 `latest` 키워드를 사용합니다. 패키지 FMRI에 대한 설명은 “[FMRI\(Fault Management Resource Identifier\)](#)” [13]를 참조하십시오. 게시자 고착성 및 검색 순서 설정에 대한 자세한 내용은 “[게시자 구성](#)” [67]을 참조하십시오.

업데이트할 패키지에 속한 보존된 구성 파일은, 파일의 `preserve` 속성 값과 파일이 변경되었는지 여부에 따라 설치되거나 저장되거나 이름이 바뀝니다. 패키지 업데이트 중 파일이 보존되는 방법에 대한 자세한 내용은 [pkg\(5\)](#) 매뉴얼 페이지에서 “File Actions” 섹션의 `preserve` 속성을 참조하십시오.

게시자 고착성 및 검색 순서와 `-g` 옵션 사용에 대한 자세한 내용은 “[새 패키지 설치](#)” [39]를 참조하십시오.

현재 설치되지 않은 패키지를 업데이트하려고 시도하면 어떤 패키지도 업데이트하지 않고 `pkg update` 작업이 종료됩니다. `--ignore-missing` 옵션을 사용하여 설치되지 않은 패키지를 무시하고 일부 업데이트할 패키지가 현재 설치되지 않은 경우 `pkg update`가 실패하지 않도록 막을 수 있습니다.

패키지 FMRI 또는 패턴이 지정되지 않았거나 지정된 패턴이 별표 문자(\*)인 경우 `pkg update` 명령의 특수 동작에 대한 자세한 내용은 [4장. Oracle Solaris 이미지 업데이트 또는 업그레이드](#)를 참조하십시오.

## 패키지 다운그레이드

`pkg update` 명령을 사용하여 패키지 업그레이드는 물론 다운그레이드도 수행할 수 있습니다. 패키지를 다운그레이드하려면 현재 설치된 버전보다 이전 버전으로 패키지 FMRI를 지정합니다. 패키지 FMRI에 대한 설명은 “[FMRI\(Fault Management Resource Identifier\)](#)” [13]를 참조하십시오. `pkg list` 명령을 사용하여 현재 설치된 패키지 버전과 구성된 게시자에서 사용 가능한 버전을 볼 수 있습니다.

다운그레이드할 패키지에 속한 보존된 구성 파일은, 파일의 `preserve` 속성 값과 파일이 변경되었는지 여부에 따라 설치되거나 이름이 바뀝니다. 패키지 다운그레이드 중 파일이 보존되는 방법에 대한 자세한 내용은 [pkg\(5\)](#) 매뉴얼 페이지에서 “File Actions” 섹션의 `preserve` 속성을 참조하십시오.

`-g` 옵션 사용에 대한 자세한 내용은 “[새 패키지 설치](#)” [39]를 참조하십시오.

## 설치된 패키지의 문제 수정

IPS는 설치된 패키지가 올바르게 설치되었는지 검증하고, 검증 문제를 수정하고, 설치된 파일을 패키지된 상태로 복원하는 작업을 제공합니다.

### pkg fix 및 pkg revert 명령 비교

pkg fix 명령과 pkg revert 명령은 모두 설치된 패키지의 구성 요소를 다시 설치합니다. 다음 표는 두 명령 사이의 유사점과 차이점을 보여줍니다.

pkg fix	<ul style="list-style-type: none"> <li>■ 패키지에 작동합니다. 하나 이상의 패키지 이름이나 패키지 이름과 일치하는 패턴을 피연산자로 취합니다.</li> <li>■ pkg verify가 실패한 패키지에서만 작동합니다.</li> <li>■ pkg verify가 보고한 오류만 수정합니다. 패키지에서 다른 콘텐츠나 메타 데이터는 다시 전달하지 않습니다.</li> </ul>
pkg revert	<ul style="list-style-type: none"> <li>■ 파일에 작동합니다. 하나 이상의 파일 이름이나 태그 이름을 피연산자로 취합니다.</li> <li>■ 피연산자로 식별된 파일을 다시 전달합니다. 패키지에서 다른 콘텐츠나 메타 데이터는 다시 전달하지 않습니다.</li> </ul>

### 패키지 검증 및 검증 오류 수정

pkg verify 명령을 사용하여 이미지의 패키지 설치를 검증할 수 있습니다. 관련 게시자에 대한 현재 서명 정책이 ignore가 아니면 정책을 기반으로 각 패키지의 서명이 검증됩니다. 서명 정책을 적용하는 방법에 대한 설명은 [“서명된 패키지에 필요한 이미지 등록 정보” \[88\]](#)를 참조하십시오. 설치된 패키지 콘텐츠에 대한 검증은 사용자 정의 콘텐츠 분석을 기반으로 수행되며 사용자 정의 콘텐츠 분석은 다른 프로그램과는 다른 결과를 반환할 수 있습니다.

패키지 이름을 제공하지 않으면 모든 설치된 패키지를 검사합니다. -v 옵션은 각 설치된 패키지마다 한 줄 이상 정보 메시지를 제공합니다. 다음 예제는 작은 출력 샘플만 보여줍니다. pkg/depot 패키지 설치에 오류가 있습니다.

```
$ pkg verify -v
PACKAGE                                     STATUS
pkg://solaris/archiver/gnu-tar              OK
pkg://solaris/audio/audio-utilities         OK
pkg://solaris/benchmark/x1lperf            OK
...
pkg://solaris/package/pkg/depot             ERROR
  dir: var/cache/pkg/depot
      Group: 'pkg5srv (97)' should be 'bin (2)'
  file: var/log/pkg/depot/access_log
      editable file has been changed
```

```

        file: var/log/pkg/depot/error_log
            editable file has been changed
    ...
pkg://solaris/security/sudo                                OK
        file: etc/sudoers
            editable file has been changed
    ...
pkg://solaris/x11/xlock                                    OK
pkg://solaris/x11/xmag                                     OK
pkg://solaris/x11/xvidtune                                 OK

```

pkg fix 명령을 사용하여 pkg verify 명령이 보고한 패키지 오류를 수정할 수 있습니다.

pkg verify 출력에는 설치된 sudo 패키지의 구성 요소가 패키지된 구성 요소와 다르다고 나타나지만 이러한 차이점이 검증 오류로 보고되지 않았습니다. pkg fix는 아무것도 변경하지 않습니다. /etc/sudoers 파일이 교체되지 않습니다.

```

$ pkg fix pkg://solaris/security/sudo
No repairs for this image.

```

/etc/sudoers 파일을 제거하면 패키지 검증을 실패하고 pkg fix가 파일을 교체합니다.

```

$ pkg fix pkg://solaris/security/sudo
Verifying: pkg://solaris/security/sudo                    ERROR
        file: etc/sudoers
            Missing: regular file does not exist
Created ZFS snapshot: 2014-03-13-22:05:42
Repairing: pkg://solaris/security/sudo
Creating Plan (Evaluating mediators):

DOWNLOAD          PKGS          FILES    XFER (MB)   SPEED
Completed         1/1           1/1      0.0/0.0    990B/s

PHASE              ITEMS
Updating modified actions      1/1
Updating package state database Done
Updating package cache         0/0
Updating image state           Done
Creating fast lookup database  Done

```

한 파일을 다운로드하고 한 작업(file 작업을 수정한다고 언급했듯이, 누락된 파일만 교체합니다. 다른 sudo 패키지 내용은 건드리지 않습니다. 이 작업은 복구를 수행하기 전에 현재 설치의 스냅샷을 저장했습니다. pkg fix 출력에서 “Created ZFS snapshot” 행을 살펴보세요. 현재 이미지에 복구가 수행되었습니다.

```

$ zfs list -r rpool/ROOT/s11
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool/ROOT/s11      16.3G 22.5G 26.1G /
rpool/ROOT/s11@2014-03-13-23:52:19 249M  -    26.1G -

```

pkg verify 출력에서 설치된 pkg/depot 패키지의 디렉토리 소유권에 오류가 있음을 알 수 있습니다. pkg fix 출력은 “Verifying” 섹션의 오류만 보여줍니다. 패키지된 구성 요소와 다른 차이점은 표시되지 않습니다.

```
$ ls -ld /var/cache/pkg/depot
drwxr-xr-x  3 pkg5srv pkg5srv      3 Dec  2 19:47 /var/cache/pkg/depot/
$ pkg fix pkg://solaris/package/pkg/depot
Verifying: pkg://solaris/package/pkg/depot          ERROR
           dir: var/cache/pkg/depot
                Group: 'pkg5srv (97)' should be 'bin (2)'
Created ZFS snapshot: 2014-03-13-22:18:52
Repairing: pkg://solaris/package/pkg/depot
Creating Plan (Evaluating mediators):
```

PHASE	ITEMS
Updating modified actions	1/1
Updating package state database	Done
Updating package cache	0/0
Updating image state	Done
Creating fast lookup database	Done

다음 출력은 오류가 수정되었음을 보여줍니다. 설치된 구성 요소와 패키지로 구성된 구성 요소 간의 다른 차이점은 그대로입니다.

```
$ ls -ld /var/cache/pkg/depot
drwxr-xr-x  3 pkg5srv bin          3 Dec  2 19:47 /var/cache/pkg/depot/
$ pkg verify -v pkg://solaris/package/pkg/depot
PACKAGE                                     STATUS
pkg://solaris/package/pkg/depot           OK
   file: var/log/pkg/depot/access_log
         editable file has been changed
   file: var/log/pkg/depot/error_log
         editable file has been changed
```

## 파일 복원

pkg revert 명령을 사용하여 파일을 패키지로 된 상태로 복원할 수 있습니다. 파일 소유권과 보호도 함께 복원됩니다.



주의 - 몇몇 편집 가능한 파일을 되돌리면 시스템을 부트할 수 없게 되거나 다른 오작동이 발생할 수 있습니다.

주요 편집 가능한 파일을 되돌릴 때는 --require-backup-be 옵션을 사용합니다.

## 명명된 파일 되돌리기

다음 예제는 pkg/depot 패키지에서 설치된 파일 2개가 패키지로 된 버전과 다른데, 이 중 하나를 지정합니다.

```
$ pkg revert -v /var/log/pkg/depot/access_log
Packages to fix:      1
```

```

Estimated space available: 21.08 GB
Estimated space to be consumed: 460.87 MB
Create boot environment: No
Create backup boot environment: No
Rebuild boot archive: No

```

Changed packages:

```

solaris
package/pkg/depot
0.5.11,5.11-0.175.2.0.0.33.0:20140217T134751Z
DOWNLOAD          PKGS      FILES    XFER (MB)   SPEED
Completed          1/1       1/1      0.0/0.0     50B/s

```

```

PHASE              ITEMS
Updating modified actions  1/1
Updating package state database  Done
Updating package cache      0/0
Updating image state        Done
Creating fast lookup database  Done

```

지정된 파일은 패키지로된 버전으로 교체되었습니다. pkg.depot 패키지의 다른 구성 요소는 변경되지 않았습니다.

## 태그된 파일 및 디렉토리 되돌리기

--tagged 옵션을 사용하여 다음 작업을 수행할 수 있습니다.

- 지정된 태그 이름이 붙은 파일을 모두 되돌립니다.
- 지정된 태그 이름이 붙은 디렉토리 아래에, 지정된 패턴과 일치하는 패키지되지 않은 파일이나 디렉토리를 제거합니다.

자세한 내용은 [“Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”의 “File Actions”](#) 및 [“Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”의 “Directory Actions”](#)에서 revert-tag 속성 설명을 참조하십시오.

다음 예제는 dev-init 태그 이름이 붙은 파일을 포함하는 일부 패키지를 보여줍니다.

```

$ pkg contents -o pkg.name,path -s pkg.name -t file -a revert-tag=dev-init '*'
PKG.NAME          PATH
system/device-administration  etc/mpxio/devid_path.cache
system/device-administration  etc/dev/chassis_aliases
system/device-administration  etc/dev/.chassis_aliases
system/device-administration  etc/dev/reserved_devnames
system/kernel          etc/path_to_inst
system/network         etc/dladm/dataalink.conf

```

다음 명령은 dev-init 태그 이름이 붙은 파일을 모두 되돌리는 작업의 미리 보기를 제공합니다. 부트 아카이브가 재구성됩니다. 새 부트 환경이나 백업 부트 환경을 만드는 옵션을 사용하는 것이 좋습니다.

```
$ pkg revert -nv --tagged dev-init
      Packages to fix:      6
      Estimated space available: 22.39 GB
      Estimated space to be consumed: 468.60 MB
      Create boot environment: No
      Create backup boot environment: No
      Rebuild boot archive: Yes

Changed packages:
solaris
system/core-os
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T144208Z
system/device-administration
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T144459Z
system/io/usb
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145048Z
system/kernel
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145214Z
system/kernel/platform
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145112Z
system/network
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T150218Z
```

## 패키지 제거

`pkg uninstall` 명령을 사용하여 설치된 패키지를 제거할 수 있습니다.

패키지 설치 작업을 실행 취소하려면 `pkg history` 명령을 사용하여 설치된 내용을 정확히 찾을 수 있습니다. 자세한 내용은 “[작업 내역 보기](#)” [93]를 참조하십시오.

현재 설치되지 않은 패키지를 제거하려고 시도하면 어떤 패키지도 제거하지 않고 `pkg uninstall` 작업이 종료됩니다. `--ignore-missing` 옵션을 사용하여 설치되지 않은 패키지를 무시하고 일부 제거할 패키지가 현재 설치되지 않은 경우 `pkg uninstall`가 실패하지 않도록 막을 수 있습니다.

그룹 종속성을 가진 패키지의 경우 패키지를 제거하면 무시 목록에 놓입니다. 무시 목록에 대한 자세한 내용은 “[그룹 패키지의 일부 패키지 설치 방지](#)” [83]를 참조하십시오.

## 이미지 다시 설치

원하는 최종 결과가 무엇인지 정확히 알고 있고 그 결과를 이루려면 다수의 패키지 변경이 필요한 경우(예: 다수의 패키지 제거) `pkg exact-install` 명령을 사용할 수 있습니다. `pkg exact-install` 명령의 결과는, 지정된 패키지와 해당 종속성만 설치된 이미지입니다. 현재



설치된 패키지 중에서 `pkg exact-install` 명령줄에 지정되지 않고 지정된 패키지에 종속되지 않은 패키지는 제거됩니다.

`pkg exact-install` 명령은 무시 목록에 있는 패키지를 설치하지 않는다는 제한 사항을 무시합니다. 패키지가 무시 목록에 있을 경우 패키지를 설치하면 해당 목록에서 패키지가 제거됩니다. 무시 목록에 대한 자세한 내용은 [“그룹 패키지의 일부 패키지 설치 방지” \[83\]](#)를 참조하십시오. `pkg exact-install` 명령은 고정 목록에 있는 패키지를 업데이트하지 않는다는 제한 사항을 무시합니다. 고정된 패키지에 대한 자세한 내용은 [“지정한 버전으로 패키지 잠금” \[78\]](#)을 참조하십시오.

그렇지 않으면, `exact-install` 하위 명령은 `install` 하위 명령과 동일한 방법으로 작동합니다. 이미지 변형 및 페이지 설정, 이미지 등록 정보 설정, 게시자 설정은 유지됩니다. 하나라도 패키지를 이 이미지에 설치할 수 없으면 지정된 패키지 중 아무것도 설치되지 않습니다. [“비전역 영역 작업” \[49\]](#)에 설명된 대로 비전역 영역은 패키지 업데이트 또는 제거 결과의 영향을 받습니다. `-r` 옵션은 `exact-install`에 사용할 수 없습니다.

게시자 고착성 및 검색 순서와 `-g` 옵션 사용에 대한 자세한 내용은 [“새 패키지 설치” \[39\]](#)를 참조하십시오.

다음은 `pkg exact-install` 명령 사용 시 권장되는 방법입니다.

- [“설치 가능한 패키지” \[20\]](#)에 표시된 대로 `pkg list -a`를 사용하여 구성된 게시자에서 사용 가능한 패키지의 버전을 확인합니다. `exact-install`을 사용하여 현재 버전을 다시 설치하려는데 사용 가능한 최신 버전이 있는 경우, 설치할 패키지 목록에서 패키지 FMRI의 버전 부분을 지정해야 합니다.
- 설치할 패키지 목록에 `entire` 통합 패키지를 포함합니다.
- 설치할 패키지 목록에 `solaris-minimal-server` 패키지와 같은 시스템 그룹 패키지 중 하나를 포함합니다.
- 먼저 `-nv` 또는 `-nvv` 옵션으로 명령을 실행하여 무엇이 설치되고 무엇이 제거될지 정확히 확인합니다.
- `--be-name` 옵션을 사용하여 새 BE에 의미 있는 이름으로 설치합니다.

다음 예제는 현재 이미지와 동일한 버전으로 최소 설치의 새 이미지를 만듭니다.

```
$ pkg list -Hv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z
$ pkg exact-install --be-name s11.2 entire@0.5.11,5.11-0.175.2.0.0.34 solaris-minimal-server
```

## 비전역 영역 작업

비전역 영역에서는 대부분의 IPS 명령을 전역 영역에서와 동일하게 사용할 수 있습니다. 영역에 대한 소개 정보는 [“이미지 및 부트 환경” \[16\]](#)을 참조하십시오.

패키지 설치와 관련하여 전역 영역과 비전역 영역은 “[전역 영역과 비전역 영역의 관계](#)” [50] 및 “[여러 비전역 영역을 동시에 업데이트](#)” [53]에 설명된 대로 부모-자식 관계를 갖습니다.

전역 영역과 비전역 영역의 중요한 차이는 “[시스템 저장소 및 프록시 서비스](#)” [51]에 설명된 대로 패키지 게시자를 사용한다는 것입니다.

## 전역 영역과 비전역 영역의 관계

설치된 solaris 브랜드 비전역 영역은 전역 영역에서 이루어지는 패키지 설치, 업데이트 및 제거의 영향을 받을 수 있습니다.

페이싯 및 변형을 변경하면 패키지가 설치 및 제거되고 비전역 영역에 영향을 미칠 수 있습니다.

비전역 영역은 부트하지 않아도 전역 영역에서 업데이트할 수 있습니다. 비전역 영역을 설치하기만 하면 전역 영역에서 행하는 패키지 변경의 영향을 받을 수 있습니다.

전역 영역에서 설치 및 업데이트 명령을 실행할 때 기본적으로 전역 영역과 각 설치된 비전역 영역이 순차적으로 업데이트되고, 비전역 영역은 전역 영역과 호환성 유지에 필요한 정도까지만 수정됩니다.

- 비전역 영역에서 최소 필요한 업데이트만 수행하는 대신, 전역 영역에서 수행한 동일 작업을 비전역 영역에서 수행하려면 “[비전역 영역에서 작동하는 옵션](#)” [36]에 설명된 대로 `-r` 옵션을 사용합니다.
- 비전역 영역을 전역 영역과 동시에 업데이트하려면 “[비전역 영역에서 작동하는 옵션](#)” [36]에 설명되고 “[여러 비전역 영역을 동시에 업데이트](#)” [53]에 표시된 대로 `-c` 옵션을 사용합니다.

---

**작은 정보** - 전역 영역뿐 아니라 비전역 영역에서 변경되는 사항을 검토하려면 `-nv` 옵션을 사용합니다.

---

비전역 영역에 로그인한 동안 패키지 명령을 실행하면 해당 비전역 영역만 영향을 받습니다. 다음 방법으로 비전역 영역을 부모 전역 영역과 다르게 설정할 수 있습니다.

- 다른 패키지를 설치할 수 있습니다.
- 결과가 전역 영역과 호환되면 동일한 패키지의 다른 버전을 설치할 수 있습니다.
- 다른 패키지를 무시 목록에 놓을 수 있습니다.
- 다른 패키지를 다른 버전에 고정할 수 있습니다.
- 다른 기본 구현을 선택하도록 중개자를 설정할 수 있습니다.
- 다른 페이싯을 설정할 수 있습니다.

비전역 영역에 설치된 패키지 버전은 전역 영역에 설치된 버전으로 제한될 수 있습니다. 일부 패키지는 비전역 영역과 전역 영역에서 동일한 버전이어야 하므로 비전역 영역에서 업데이트

트하거나 다운그레이드할 수 없습니다. 예를 들어, `entire`라는 패키지는 각 비전역 영역에서 전역 영역과 동일한 버전이어야 합니다.

비전역 영역에 설치된 패키지가 `parent` 종속성을 가진 경우 해당 패키지를 전역 영역에서 업데이트하면 비전역 영역에서도 업데이트됩니다. `parent` 종속성을 가진 패키지에 종속된 패키지도 영향을 받습니다.

`parent` 종속성의 영향을 받지 않는 패키지는 전역 영역에 설치된 버전과 다른 버전으로 비전역 영역에 설치할 수 있습니다. 비전역 영역에 다른 버전을 설치하려면 `pkg install` 명령에 버전을 지정하거나 원하는 버전으로 고정하십시오.

비전역 영역에 패키지 설치와 관련된 도움이 필요하면 [“동기 링크된 패키지를 설치할 수 없음” \[115\]](#) 및 [“비전역 영역을 설치할 수 없음” \[116\]](#)을 참조하십시오.

## 시스템 저장소 및 프록시 서비스

비전역 영역에서 시스템 저장소는 전역 영역에 구성된 패키지 저장소에 대한 액세스를 제공합니다. 전역 영역에서 변경한 게시자 구성은 시스템 저장소를 통해 모든 비전역 영역에 즉시 표시됩니다.

비전역 영역에 구성된 게시자 원본/미러는 해당 위치가 전역 영역 게시자 목록에 구성되지 않았더라도 전역 영역에서 액세스할 수 있어야 합니다. 예를 들어, `localsw` 게시자가 비전역 영역에 구성되고 전역 영역에 구성되지 않은 경우 `localsw` 게시자의 모든 원본/미러는 여전히 전역 영역에서 액세스할 수 있어야 합니다.

시스템 저장소는 `http`, `https`, `v4` 파일 저장소와 `.p5p` 아카이브 저장소를 프록시할 수 있습니다.

영역 프록시는 `pkg` 명령을 영역 내에서 실행하여 전역 영역에서 실행 중인 시스템 저장소와 통신할 수 있는 서비스입니다. 영역 프록시는 두 부분으로 구성됩니다. 전역 영역에서는 다음 서비스가 실행됩니다.

```
svc:/application/pkg/zones-proxyd:default
```

비전역 영역에서는 다음 서비스가 실행됩니다.

```
svc:/application/pkg/zones-proxy-client:default
```

시스템 저장소 및 영역 프록시 서비스에 대한 자세한 내용은 [pkg.sysrepo\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 예에서는 전역 영역의 게시자를 표시합니다.

```
global:~$ pkg publisher
PUBLISHER      TYPE      STATUS P LOCATION
solaris        origin   online F http://pkg.oracle.com/solaris/release/
```

```
solaris          origin  online F file:///export/IPSpkgrepos/Solaris/
devtool (disabled) origin  online F http://pkg.example1.com/
isvpub          origin  online F http://pkg.example2.com/
```

다음 예에서는 비전역 영역에 로그인했을 때 이와 동일한 게시자가 표시되는 방법을 보여줍니다.

```
z1:~$ pkg publisher
PUBLISHER      TYPE      STATUS P LOCATION
solaris (syspub) origin  online T <system-repository>
isvpub  (syspub) origin  online T <system-repository>
z1:~$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB  ENABLED TYPE      STATUS URI
PROXY
solaris   true   true   true   origin  online  http://pkg.oracle.com/solaris/release/
http:
//localhost:1008
isvpub   true   true   true   origin  online  http://pkg.example2.com/
http:
//localhost:1008
```

비전역 영역에서 시스템 저장소는 항상 프록시로 표시됩니다. 이것은 비전역 영역이 전역 영역의 시스템 저장소와 통신하는 데 사용하는 프록시입니다.

비전역 영역에서는 시스템 저장소를 다시 구성할 수 없습니다. 예를 들어, 게시자의 원본 또는 등록 정보를 변경하거나 위치가 <system-repository>인 게시자의 게시자 검색 순서를 변경할 수 없습니다. 전역 영역에서 게시자를 추가하거나 재구성하면 이러한 변경 사항이 비전역 영역에 즉시 표시됩니다. 전역 영역에서 게시자 설정을 해제하면 비전역 영역에 해당 게시자가 설치한 패키지가 없는 한, 비전역 영역에서도 설정이 해제됩니다.

---

**작은 정보** - 전역 영역에서 게시자 설정을 해제하기 전에 비전역 영역에서 해당 게시자의 패키지를 제거하십시오.

---

게시자에 연결할 수 없는 경우 [“프록시 지정” \[72\]](#)에 설명된 대로 전역 영역에서 프록시를 설정할 수 있습니다. 비전역 영역이 있을 때 `http_proxy` 및 `https_proxy` 환경 변수를 사용하는 시기와 방법을 포함한 프록시 설정에 대한 자세한 내용은 [“Oracle Solaris 영역 만들기 및 사용”](#)의 [“설치된 영역이 있는 시스템의 프록시 구성”](#)을 참조하십시오.

전역 영역에 이미 구성된 게시자에 대해 다음 `pkg list` 명령은 전역 영역과 비전역 영역 모두에서 동일한 결과를 제공합니다.

```
z1:~$ pkg list -a isvtool
NAME (PUBLISHER)  VERSION  IFO
isvtool (isvpub)  2.0     ---
isvtool (isvpub)  1.0     ---
```

저장소가 전역 영역에 구성되지 않았더라도 이러한 저장소는 비전역 영역에서 액세스할 수 있는 네트워크 또는 파일 시스템일 수 있습니다. 비전역 영역 게시자 구성은 전역 영역 게시자 구성과 일치하거나 전역 영역 게시자 구성의 수퍼 세트와 일치해야 합니다. 예를 들어, `localsw` 게시자는 비전역 영역에 원본 `file:///export/IPSpkgrepos/localrepo`로 구성할

수 있습니다. `localsw` 게시자가 전역 영역에 구성되지 않았더라도 전역 영역에서 해당 위치에 액세스할 수 있기 때문입니다.

## 여러 비전역 영역을 동시에 업데이트

기본적으로 전역 영역에서 `pkg update` 명령을 사용하면 패키징 시스템은 전역 영역과 각 비전역 영역을 차례로 업데이트합니다. 여러 비전역 영역을 동시에 업데이트하려면 `-c` 옵션을 사용하거나 전역 영역에서 `PKG_CONCURRENCY` 환경 변수를 설정합니다. 자세한 내용은 [“비전역 영역에서 작동하는 옵션” \[36\]](#)을 참조하십시오.

다음 예제에서 두 비전역 영역을 전역 영역과 동시에 업데이트합니다. 출력에서 비전역 영역은 부모 전역 영역 이미지에 링크되어 있으므로 링크된 이미지라고 합니다.

```
global:~$ pkg update -C 0 --be-name s11.2
Startup: Linked image publisher check ... Done
Startup: Refreshing catalog 'solaris' ... Done
Startup: Refreshing catalog 'isvpub' ... Done
Startup: Checking that pkg(5) is up to date ... Done
Planning: Solver setup ... Done
Planning: Running solver ... Done
Planning: Finding local manifests ... Done
Planning: Package planning ... Done
Planning: Merging actions ... Done
Planning: Checking for conflicting actions ... Done
Planning: Consolidating action changes ... Done
Planning: Evaluating mediators ... Done
Planning: Planning completed in 39.00 seconds
      Packages to remove: 2
      Packages to install: 1
      Packages to update: 640
      Create boot environment: Yes
      Create backup boot environment: No

Planning: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Planning: Linked image 'zone:z1' output:
| Packages to install: 1
| Packages to update: 161
| Services to change: 2
`

Planning: Linked images: 1/2 done; 1 working: zone:z2
Planning: Linked image 'zone:z2' output:
| Packages to install: 1
| Packages to update: 161
| Services to change: 2
`

Planning: Finished processing linked images.
Download: 0/12068 items 0.0/350.9MB 0% complete
...
Download: 11664/12068 items 336.1/350.9MB 95% complete
Download: Completed 350.91 MB in 187.08 seconds (0B/s)
```

```
Download: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Download: Linked images: 1/2 done; 1 working: zone:z1
Download: Finished processing linked images.
  Actions: 1/23382 actions (Removing old actions)
  Actions: 3867/23382 actions (Installing new actions)
  Actions: 8192/23382 actions (Updating modified actions)
...
  Actions: 23266/23382 actions (Updating modified actions)
  Actions: Completed 23382 actions in 96.16 seconds.
Finalize: Updating package state database ... Done
Finalize: Updating package cache ... Done
Finalize: Updating image state ... Done
Finalize: Creating fast lookup database ... Done
Finalize: Reading search index ... Done
Finalize: Building new search index ... Done
Finalize: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Finalize: Linked images: 1/2 done; 1 working: zone:z2
Finalize: Finished processing linked images.
```

```
A clone of s11 exists and has been updated and activated.
On the next boot the Boot Environment s11u1 will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

# ◆◆◆ 4 장 4

## Oracle Solaris 이미지 업데이트 또는 업그레이드

---

3장. 소프트웨어 패키지 설치 및 업데이트에서 명령줄에 명명된 하나 이상의 패키지를 설치, 업데이트, 수정, 제거하는 방법을 설명했습니다. 이 장에서는 Oracle Solaris 이미지를 다음 지원 업데이트나 다음 릴리스로 업그레이드하는 방법을 설명합니다. 시스템 업그레이드는 부트 가능한 이미지를 업데이트하는 것을 의미합니다. “[이미지 및 부트 환경](#)” [16]에 언급된 대로, 물리적/가상 시스템은 여러 개의 부트 가능한 이미지를 가질 수 있습니다.

업데이트를 수행하려면 높은 수준의 권한이 필요합니다. 자세한 내용은 “[설치 권한](#)” [17]을 참조하십시오.

pkg update 명령에 대한 모든 옵션의 전체 목록은 [pkg\(1\)](#) 매뉴얼 페이지를 참조하십시오.

### 이미지 업데이트 개요

pkg update 명령에 패키지 FMRI 또는 패턴이 지정되지 않았거나 별표 문자(\*)를 패턴으로 사용할 때, 사용 가능한 업데이트가 있는 설치된 모든 패키지를 설치된 패키지 종속성 및 게시자 구성에 따라 시스템에 적용된 제약 조건에서 허용하는 최신 버전으로 업데이트할 수 있습니다.

- 업데이트된 설치 패키지에 필요한 종속성이 아닌 한, 새 패키지가 설치되지 않습니다.
- 구성된 게시자가 업데이트 중인 이미지의 제약 조건에 맞는 업데이트된 버전을 제공하는 경우에만 설치된 패키지가 업데이트됩니다. 5장. [설치된 이미지 구성](#)에 설명된 대로 사용자가 제어할 수 있는 다음과 같은 구성과 패키지 종속성에 따라 제약 조건이 적용됩니다.
  - 특정 버전에 잠긴 패키지
  - 이미지에 설정된 페이스 및 변형
  - 구성된 패키지 서명 등록 정보
  - 게시자 검색 순서 및 고착성
- 하나라도 필요한 패키지를 설치할 수 없으면 아무 패키지도 업데이트되거나 설치되지 않습니다. [부록 A. 패키지 설치 및 업데이트 문제 해결](#)을 참조하십시오.

현재 이미지에 비전역 영역이 설치되어 있으면 이러한 영역도 업데이트됩니다. “[비전역 영역 작업](#)” [49]을 참조하십시오.

## 이미지 업데이트 모범 사례

업데이트 전에 다음 단계를 수행하십시오.

- 릴리스 노트를 읽습니다.
- 구성된 게시자 원본에서 사용 가능한 패키지 버전을 확인합니다. 게시자에서 `pkg refresh` 명령을 실행하거나 원본 위치에서 `pkgrepo refresh` 명령을 실행해야 할 수 있습니다.
- `pkg update` 명령을 `-nv` 옵션과 함께 사용하여 실제로 업데이트를 수행하지 않고 업데이트되는 패키지 목록을 표시합니다.

업데이트할 때 `--be-name` 또는 `--require-new-be` 옵션을 사용하여 현재 부트 환경이 아닌 새 부트 환경에서 변경을 수행합니다. 자세한 내용은 [“부트 환경 옵션” \[35\]](#) 및 [“부트 환경 정책 이미지 등록 정보” \[86\]](#)를 참조하십시오.

## 사용 가능한 버전 확인

운영 체제 릴리스를 업데이트하려면 `pkg:/entire` 통합 패키지의 사용 가능한 버전을 확인합니다. 다음 명령을 실행하면 Oracle Solaris 11 11/11 SRU 10이 설치되어 있고, Oracle Solaris 11 11/11 SRUs 11, 12 및 13을 사용할 수 있으며, 현재 구성된 `solaris` 게시자로 부터 Oracle Solaris 11.1이 제공됨을 알 수 있습니다. FMRI의 필드에 대한 자세한 내용은 [“FMRI\(Fault Management Resource Identifier\)” \[13\]](#)를 참조하십시오.

```
$ pkg list -af entire
NAME (PUBLISHER)  VERSION                                IFO
entire            0.5.11,5.11-0.175.1.0.0.24.2         ---
entire            0.5.11,5.11-0.175.0.13.0.4.0         ---
entire            0.5.11,5.11-0.175.0.12.0.4.0         ---
entire            0.5.11,5.11-0.175.0.11.0.4.1         ---
entire            0.5.11,5.11-0.175.0.10.0.5.0         i--
```

이러한 버전 중 원하는 버전이 없으면 `solaris` 게시자 원본을 다른 패키지 저장소 위치로 설정해야 합니다.

기본적으로 각 패키지는 현재 설치된 버전을 제공한 게시자에서 업데이트됩니다. 게시자 고착성 및 검색 순서를 지정하여 패키지를 제공하는 게시자를 제어할 수 있습니다. [“패키지 게시자 추가, 수정 또는 제거” \[69\]](#)를 참조하십시오.

## 업데이트 작업 미리 보기

다음 명령은 업데이트로 인해 실제로 설치되는 패키지를 보여줍니다(있는 경우). `-v` 옵션이 지정되었으므로 이 명령은 업데이트되는 627개 패키지, 제거되는 3개 패키지, 설치되는 새로운 1개 패키지 모두의 전체 FMRI(버전 포함)를 표시합니다. 이 예에서는 출력 중 대부분을



생략하고 `entire` 패키지만 보여줍니다. `-n` 옵션이 지정되었으므로 실제로는 업데이트가 수행되지 않습니다. `-n` 옵션 없이 업데이트를 수행하기 전에 이 출력을 검토하십시오.

```
$ pkg update -nv
    Packages to remove:      3
    Packages to install:    1
    Packages to update:     627
    Estimated space available: 48.43 GB
    Estimated space to be consumed: 3.14 GB
    Create boot environment: Yes
    Activate boot environment: Yes
    Create backup boot environment: No
    Rebuild boot archive:   Yes

Changed packages:
solaris
...
entire
  0.5.11,5.11-0.175.0.10.0.5.0:20120803T182627Z ->
  0.5.11,5.11-0.175.1.0.0.24.2:20120919T190135Z
...
```

앞의 예제는 Oracle Solaris 11.1의 `pkg:/entire` 통합 패키지가 설치됨을 보여줍니다.

- `entire` 통합의 제약을 받는 모든 설치된 패키지는 그에 따라 업데이트됩니다.
- 패키지 FMRI가 지정되지 않았으므로 `entire` 통합의 제약을 받지 않는 설치된 패키지에 대한 업데이트도 수행됩니다. 설치된 모든 패키지는 설치된 패키지 종속성 및 게시자 구성에 따라 시스템에 적용된 제약 조건에서 허용되는 최신 버전으로 업데이트됩니다.
- 설치된 패키지는 제거할 수 있으며 업데이트된 설치 패키지가 새로운 종속성을 지정할 경우 새 패키지를 설치할 수 있습니다.

## 새 부트 환경 지정

“[업데이트 작업 미리 보기](#)” [56]의 예제에서 `-n` 옵션 없이 이 명령을 실행할 경우 이 업데이트로 새로운 BE가 만들어진다는 것을 알 수 있습니다. `-n` 옵션 없이 이 명령을 실행하면 업데이트 출력의 마지막에 다음 메시지가 표시됩니다.

```
A clone of currentBE exists and has been updated and activated.
On the next boot the Boot Environment newBE will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

현재 BE는 수정되지 않습니다. 모든 변경 사항은 새로운 BE에 적용됩니다.

가장 안전한 설치 또는 업데이트 방법은 새 BE를 명시적으로 지정하는 것입니다. BE 생성 시점에 대한 자세한 내용은 “[부트 환경 정책 이미지 등록 정보](#)” [86]를 참조하십시오. `--be-name` 옵션을 사용하여 새 BE에 의미 있는 이름을 지정할 수도 있습니다. 새로운 BE가 활성화되고 다음에 시스템을 부트하면 기본적으로 이 새로운 환경이 부트됩니다. 다음 재부트 시 새로운 BE가 기본값이 되지 않도록 하려면 `pkg update` 명령을 `--no-be-activate` 옵션과

함께 사용합니다. 언제든지 `beadm activate` 명령을 사용하여 기본 부트 BE를 변경할 수 있습니다.

새 BE에 만족하면 이전 것을 삭제할 수 있습니다.

---

**작은 정보** - 각 운영 체제 릴리스마다 이전 BE를 보관하십시오. 필요한 경우, 이전 BE로 다시 부트하여 설치된 구 버전과 다음 최신 버전 사이에 버전 업데이트를 수행할 수 있습니다.

---

## 설치할 버전 지정

허용되는 최신 버전으로 업데이트하지 않으려면 `pkg update` 명령에 버전 문자열 부분을 포함하여 패키지 이름을 지정할 수 있습니다. 다음 예에서는 더 새로운 버전이 허용되더라도 Oracle Solaris 11 11/11 SRU 13으로 업데이트하도록 `entire` 통합의 버전을 지정하는 방법을 보여줍니다. `-nv` 옵션을 다시 사용하고 출력을 다시 확인하십시오.

```
$ pkg update -nv entire@0.5.11,5.11-0.175.0.13
Packages to remove:      2
Packages to install:     1
Packages to update:     486
Estimated space available: 48.39 GB
Estimated space to be consumed: 2.50 GB
Create boot environment: Yes
Activate boot environment: Yes
Create backup boot environment: No
Rebuild boot archive:   Yes

Changed packages:
solaris
...
entire
  0.5.11,5.11-0.175.0.10.0.5.0:20120803T182627Z ->
  0.5.11,5.11-0.175.0.13.0.4.0:20121106T194623Z
...
```

일부 설치된 패키지는 `entire` 통합의 제약을 받는 패키지에 종속될 수 없습니다. 이러한 패키지는 `entire` 통합만 업데이트해서는 업데이트되지 않습니다. 이러한 패키지는 동일한 `pkg update` 명령에 이름으로 추가할 수 있습니다.

## 업데이트 전 버전 제약 조건 지정

모든 Oracle Solaris 11 11/11 버전으로의 업데이트를 허용하지만 Oracle Solaris 11.1로의 업데이트는 허용하지 않으려는 경우 다음 명령에 표시된 것처럼 `entire` 통합을 고정시킬 수 있습니다. `0.175.0`를 지정하면 `entire` 패키지를 `0.175.10`이 아니라 `0.175.0.13` 등으로 업데이트할 수 있음을 의미합니다.

```

$ pkg freeze -c "Keep this image at 11 11/11." entire@0.5.11,5.11-0.175.0
entire was frozen at 0.5.11,5.11-0.175.0
$ pkg freeze
NAME      VERSION          DATE              COMMENT
entire    0.5.11,5.11-0.175.0 30 Jan 2013 15:50:01 PST Keep this image at 11 11/11.
$ pkg list entire
NAME (PUBLISHER) VERSION          IFO
entire           0.5.11,5.11-0.175.0.10.0.5.0 if-

```

패키지 고정에 대한 자세한 내용은 “[지정한 버전으로 패키지 잠금](#)” [78]을 참조하십시오.

## 사용자 정의 통합 설치

이전 절에 표시된 대로 `pkg freeze` 명령을 사용한 것과 비슷하게, 고유의 사용자 정의 통합 패키지를 사용하여 원하는 제약 조건을 지정할 수 있습니다. Oracle Solaris 통합 패키지와의 그 사용법에 대한 자세한 내용은 “[통합 패키지](#)” [12]를 참조하십시오.

통합 패키지를 만들고 로컬 IPS 패키지 저장소나 패키지 아카이브 파일에서 패키지를 설치합니다. 제약 조건을 변경하려면 사용자 정의 통합 패키지를 수정 후 다시 전달하고 `pkg update`를 사용하여 새 통합을 설치합니다.

사용자 정의 통합을 사용하여 설치할 수 있는 소프트웨어 버전을 제어하면 복수의 패키지 저장소를 유지하지 않고도 여러 시스템에서 다른 버전의 Oracle Solaris를 쉽게 유지 관리할 수 있습니다. 각 이미지마다 다른 버전의 사용자 정의 업데이트 제어 통합 패키지를 설치할 수 있습니다. 모든 시스템에서 필요한 소프트웨어의 모든 버전을 포함하는 동일한 패키지 저장소를 공유합니다.

## 사용자 정의 통합 패키지 만들기

이미지에 설치할 수 있는 핵심 운영 체제 패키지의 버전을 `pkg:/entire` 통합 패키지로 제어할 수 있습니다. 시스템 업그레이드를 제어하려면 특정 버전의 `pkg:/entire` 패키지를 `incorporate` 종속성으로 지정하는 패키지를 만듭니다.

## 사용자 정의 통합 패키지 매니페스트 만들기

다음 예제는 설치할 수 있는 `pkg:/entire` 패키지의 버전을 제어하는 사용자 정의 통합 패키지에 대한 매니페스트 `upgradectl.p5m`을 보여줍니다. 이 매니페스트의 일부 설정을 아래에 설명합니다.

```

set name=pkg.fmri value=upgradectl@1.0
set name=pkg.summary value="Incorporation to constrain the version of the OS"
set name=pkg.description value="This package controls the version of \

```

```
pkg://solaris/entire that can be installed."
set name=info.classification value="org.opensolaris.category.2008:Meta Packages/
Incorporations"
set name=pkg.depend.install-hold value=core-os
set name=variant.opensolaris.zone value=global value=nonglobal
set name=variant.arch value=sparc value=i386
depend fmri=feature/package/dependency/self type=parent variant.opensolaris.zone=nonglobal
depend fmri=pkg://solaris/entire type=require
depend fmri=pkg://solaris/entire@0.5.11,5.11-0.175.1.0 type=incorporate
```

pkg.depend.install-hold

사용자가 pkg update upgradectl 명령을 입력하면 pkg:/entire 패키지도 자동으로 업데이트됩니다.

variant.opensolaris.zone

이 패키지는 전역 영역과 비전역 영역에 모두 설치할 수 있습니다. parent 종속성에 대한 설명을 참조하십시오.

variant.arch

이 패키지는 SPARC 및 x86 시스템에 모두 설치할 수 있습니다.

parent 종속성

이 패키지는 전역 영역에 이미 설치된 경우에만 비전역 영역에 설치할 수 있습니다.

require 종속성

upgradectl 패키지는 pkg://solaris/entire 패키지가 이미 설치된 경우에만 설치할 수 있습니다. 또는 이와 동일한 작업으로 설치할 수 있습니다.

incorporate 종속성

pkg://solaris/entire 패키지를 지정된 버전에 설치해야 합니다. 지정된 정확성 자릿수에 따라 여러 개의 버전이 incorporate 종속성을 충족할 수 있습니다. 이 예제에서 0.175.1.0은 Oracle Solaris 11.1 SRU 0을 지정합니다. 이 업그레이드 제어 패키지는 지원 업데이트 없이 Oracle Solaris 11.1에 시스템을 유지합니다. 그러나 이 업그레이드 제어 패키지는 pkg:/entire 통합의 제약을 받지 않는 패키지가 업데이트되도록 허용합니다.

## 업그레이드 제어 패키지 게시

upgradectl 패키지를 로컬 파일 기반 저장소에 게시합니다. 이 저장소는 새 패키지 개발 및 테스트용으로 사용됩니다. 범용 저장소를 만들려면 저장소에 개별 파일 시스템을 만드는 등의 추가 단계를 거쳐야 합니다. 범용 패키지 저장소 만들기에 대한 자세한 내용은 [“Oracle Solaris 11.2 패키지 저장소 복사 및 만들기”](#)를 참조하십시오.

시스템에 패키지 개발 저장소를 만듭니다. pkgrepo 명령에 대한 자세한 내용은 [pkgrepo\(1\)](#) 매뉴얼 페이지를 참조하십시오.

```
$ pkgrepo create myrepo
```

이 저장소의 기본 게시자를 설정합니다. 기본 게시자는 저장소의 publisher/prefix 등록 정보 값입니다.

```
$ pkgrepo -s myrepo set publisher/prefix=site
```

upgradectl 패키지를 개발 저장소에 게시합니다.

```
$ pkgsend -s myrepo publish upgradectl.p5m
pkg://site/upgradectl@1.0,5.11:20131104T072336Z
PUBLISHED
```

저장소 기본 게시자가 패키지 FMRI에 적용되었습니다.

저장소를 검사하여 패키지가 게시되었는지 확인합니다.

```
$ pkgrepo -s myrepo list
PUBLISHER NAME          0 VERSION
site      upgradectl    1.0,5.11:20131104T072336Z
$ pkg list -vg myrepo
FMRI                                IFO
pkg://site/upgradectl@1.0,5.11:20131104T072336Z  ---
```

공유 위치에 있는 개별 ZFS 파일 시스템의 로컬 저장소로 패키지를 제공합니다.

```
$ pkgrecv -s myrepo -d /export/IPSpkgrepos/Solaris upgradectl
Processing packages for publisher site ...
Retrieving and evaluating 1 package(s) ...
PROCESS  ITEMS  GET (MB)  SEND (MB)
Completed  1/1    0.0/0.0    0.0/0.0
```

저장소의 패키지가 통합되는 pkg:/entire의 버전을 확인합니다.

```
$ pkg info -g /export/IPSpkgrepos/Solaris upgradectl
Name: upgradectl
Summary: Incorporation to constrain the version of the OS
Description: This package controls the version of pkg://solaris/entire that
             can be installed.
Category: Meta Packages/Incorporations
State: Not installed
Publisher: site
Version: 1.0
Build Release: 5.11
Branch: None
Packaging Date: November 20, 2013 01:01:05 AM
Size: 0.00 B
FMRI: pkg://site/upgradectl@1.0,5.11:20131120T010105Z
$ pkg contents -hro fmri -t depend -a type=incorporate upgradectl
pkg://solaris/entire@0.5.11,5.11-0.175.1.0
```

IPS 패키지 만들기 및 제공에 대한 자세한 내용은 [“Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”](#)의 [“Creating and Publishing a Package”](#)를 참조하십시오.

## 게시자 원본 설정

site 게시자의 원본을 설정합니다. 이 정보로 시스템 저장소가 자동으로 업데이트되므로 비전역 영역에서 site 게시자의 패키지에 액세스할 수 있습니다.

```
$ pkg set-publisher -g /export/IPSpkgrepos/Solaris site
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F https://pkg.oracle.com/solaris/support/
site              origin  online F file:///export/IPSpkgrepos/Solaris/
```

## 업그레이드 제어 패키지 설치

패키지를 설치합니다. 이 경우 pkg:/entire의 설치된 버전이 업그레이드 제어 패키지에 통합된 버전과 같기 때문에 변경 사항이 거의 없어야 합니다. 패키지가 비전역 영역에도 설치되었습니다.

```
$ pkg list -v entire
FMRI                                                    IFO
pkg://solaris/entire@0.5.11,5.11-0.175.1.0.0.24.2:20120919T190135Z  i--
$ zoneadm list
global
z1
$ pkg install upgradectl
          Packages to install: 1
          Create boot environment: No
          Create backup boot environment: No

Planning linked: 0/1 done; 1 working: zone:z1
Planning linked: 1/1 done
Downloading linked: 0/1 done; 1 working: zone:z1
Downloading linked: 1/1 done
PHASE                ITEMS
Installing new actions      9/9
Updating package state database      Done
Updating image state        Done
Creating fast lookup database      Done
Reading search index        Done
Updating search index        1/1
Executing linked: 0/1 done; 1 working: zone:z1
Executing linked: 1/1 done
```

다음 명령은 pkg:/entire의 설치된 버전보다 최신 버전을 구성된 solaris 게시자에서 사용할 수 있지만, 새로 설치된 업그레이드 제어 패키지에 의해 업그레이드 시도가 통제됨을 보여줍니다.

```
$ pkg list -af entire
NAME (PUBLISHER)          VERSION          IFO
entire                   0.5.11-0.175.1.13.0.6.0  ---
entire                   0.5.11-0.175.1.12.0.5.0  ---
```

```

entire                                0.5.11-0.175.1.11.0.4.0    ---
entire                                0.5.11-0.175.1.10.0.6.0    ---
entire                                0.5.11-0.175.1.10.0.5.0    ---
...
$ pkg update
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to latest available versions.
This may indicate an overly constrained set of packages are installed.
latest incorporations:
...
Try specifying expected results to obtain more detailed error messages.
$ pkg update -nv entire@0.5.11-0.175.1.13.0.6.0
pkg update: No matching version of entire can be installed:
  Reject: pkg://solaris/entire@0.5.11,5.11-0.175.1.13.0.6.0:20131108T211557Z
  Reason: This version is excluded by installed incorporation pkg://site/
upgradectl@1.0,5.11:20131120T010105Z

```

## 업그레이드 제어 패키지 업데이트

사용자의 시스템을 새 버전으로 업데이트할 준비가 되었으면 upgradectl.p5m 매니페스트를 업데이트하고 새 업데이트 제어 패키지를 다시 게시하고 다시 전달합니다. 다음 매니페스트에서 업데이트 제어 패키지의 버전과 entire 통합의 버전이 업데이트됩니다. 사용자를 돕기 위해 업그레이드 제어 패키지의 버전 1.10이 pkg:/entire 패키지의 업데이트된 버전 0.175.1.10과 일치하도록 설정됩니다.

```

set name=pkg.fmri value=upgradectl@1.10
set name=pkg.summary value="Incorporation to constrain the version of the OS"
set name=pkg.description value="This package controls the version of \
pkg://solaris/entire that can be installed."
set name=info.classification value="org.opensolaris.category.2008:Meta Packages/
Incorporations"
set name=pkg.depend.install-hold value=core-os
set name=variant.opensolaris.zone value=global value=nonglobal
set name=variant.arch value=sparc value=i386
depend fmri=feature/package/dependency/self type=parent variant.opensolaris.zone=nonglobal
depend fmri=pkg://solaris/entire type=require
depend fmri=pkg://solaris/entire@0.5.11,5.11-0.175.1.10 type=incorporate

```

다음 명령은 업데이트 제어 패키지를 다시 게시하고 다시 전달합니다.

```

$ pkgsend -s myrepo publish upgradectl.p5m
pkg://site/upgradectl@1.10,5.11:20131120T021902Z
PUBLISHED
$ pkgrepo -s myrepo list
PUBLISHER NAME                                O VERSION
site      upgradectl                          1.10,5.11:20131120T021902Z
site      upgradectl                          1.0,5.11:20131120T010105Z
$ pkgrecv -s myrepo -d /export/IPSpkgrepos/Solaris upgradectl
Processing packages for publisher site ...
Retrieving and evaluating 1 package(s)...
PROCESS                                     ITEMS      GET (MB)   SEND (MB)

```

```
Completed                               1/1    0.0/0.0    0.0/0.0
$ pkg refresh site
$ pkg list -af pkg://site/upgradectl
NAME (PUBLISHER)                       VERSION                               IFO
upgradectl (site)                       1.10                                  ---
upgradectl (site)                       1.0                                    i--
```

## 이미지 업그레이드

다음 pkg update 명령은 지정된 패키지가 없기 때문에 모든 패키지를 사용 가능한 최신 버전으로 업데이트합니다. 명령은 업그레이드 제어 패키지를 사용 가능한 최신 버전으로 업데이트합니다. 그러면 upgradectl 패키지의 pkg.depend.install-hold 설정에 따라 upgradectl 패키지를 업데이트할 때 pkg:/entire 패키지가 업데이트되므로 이미지를 업그레이드할 수 있습니다. 이미지는 새로운 upgradectl 통합에 지정된 pkg:/entire 통합의 버전으로 업그레이드됩니다.

```
$ pkg update --be-name s11u1_10
    Packages to remove: 1
    Packages to update: 186
    Mediators to change: 1
    Create boot environment: Yes
    Create backup boot environment: No

Planning linked: 0/1 done; 1 working: zone:z1
Linked image 'zone:z1' output:
| Packages to remove: 1
| Packages to install: 3
| Packages to update: 73
| Mediators to change: 1
| Services to change: 3
`
Planning linked: 1/1 done
DOWNLOAD                               PKGS      FILES    XFER (MB)   SPEED
Completed                               187/187   16139/16139  507.9/507.9 562k/s

Downloading linked: 0/1 done; 1 working: zone:z1
Downloading linked: 1/1 done
PHASE                                     ITEMS
Removing old actions                       1473/1473
Installing new actions                     3451/3451
Updating modified actions                  16378/16378
Updating package state database             Done
Updating package cache                     187/187
Updating image state                       Done
Creating fast lookup database              Done
Reading search index                      Done
Building new search index                  851/851
Executing linked: 0/1 done; 1 working: zone:z1
Executing linked: 1/1 done

A clone of s11u1_0 exists and has been updated and activated.
```



On the next boot the Boot Environment `s11u1_10` will be mounted on `'/'`. Reboot when ready to switch to this updated BE.

현재 BE가 변경되지 않았고 새 BE에 업데이트된 패키지가 있는지 확인합니다.

```
$ pkg list entire upgradectl
NAME (PUBLISHER)          VERSION          IFO
entire                   0.5.11-0.175.1.0.0.24.2  i--
upgradectl (site)       1.0              i--
$ beadm mount s11u1_10 /mnt
$ pkg -R /mnt list entire upgradectl
NAME (PUBLISHER)          VERSION          IFO
entire                   0.5.11-0.175.1.10.0.6.0  i--
upgradectl (site)       1.10             i--
$ beadm unmount s11u1_10
```

## 이미지 다운그레이드

운영 체제 릴리스를 다운그레이드하려면 다운그레이드하려는 버전보다 오래된 BE로 부트하고 여기에서 업그레이드합니다. 예를 들어, Oracle Solaris 11 11/11 SRU 10에서 Oracle Solaris 11 11/11 SRU 13으로 업데이트한 상태에서 다시 SRU 12 이미지가 필요한 경우에는 SRU 10 BE로 재부트한 후에 여기에서 SRU 12로 업데이트하십시오.



# ◆◆◆ 5 장

## 설치된 이미지 구성

---

이 장에서는 패키지 게시자 구성, 설치 가능한 패키지 제한, 패키지 서명 정책 설정, BE(부트 환경) 정책 구성 등 전체 이미지에 적용되는 특성을 구성하는 방법을 설명합니다.

- 원본, 검색 순서, 키와 인증서, 프록시 설정을 포함한 게시자 구성
- 변형 및 페이지를 설정하여 선택적 구성 요소의 설치 제어
- 지정한 버전으로 패키지 잠금
- 통합에서 지정한 버전 제약 조건 완화
- 중개를 사용하여 응용 프로그램의 기본 구현 지정
- 그룹 패키지의 일부 패키지 설치 방지
- BE 만들기 정책과 패키지 서명 정책을 포함한 이미지 및 게시자 등록 정보 구성
- 이미지 만들기
- 패키지 작업 내역 보기

이러한 작업은 대부분 높은 수준의 권한이 필요합니다. 자세한 내용은 “[설치 권한](#)” [17]을 참조하십시오.

이 장에 설명된 명령에 대한 모든 옵션의 전체 목록은 [pkg\(1\)](#) 매뉴얼 페이지를 참조하십시오.

## 게시자 구성

소프트웨어를 설치하고 업데이트하려면 pkg 클라이언트가 패키지 저장소에 연결할 수 있어야 합니다.

## 게시자 정보 표시

pkg publisher 명령을 사용하여 이 이미지에 대해 구성된 패키지 게시자에 대한 정보를 표시할 수 있습니다. 패키지 FMRI에 게시자가 지정되지 않은 경우에는 패키지를 찾기 위해 게시자가 검색되는 순서로 게시자가 나열됩니다.

새로 설치한 Oracle Solaris 11 시스템에는 기본적으로 solaris 게시자가 구성되어 있습니다. pkg publisher 명령을 사용하여 게시자 원본을 확인할 수 있습니다.

```
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
isvpub             (non-sticky) origin  online F file:///export/IPSpkgrepos/isvrepo/
devtool           (disabled) origin  online F http://pkg.example1.com/
```

TYPE 열은 LOCATION 값이 원본인지 또는 미러인지를 나타냅니다. 설명은 “[저장소 원본 및 미러](#) [15]를 참조하십시오.

STATUS와 LOCATION 열 사이에 있는 P 열은 위치가 프록시되었는지 여부를 지정합니다. 이 열의 값은 true(T) 또는 false(F)입니다. 파일 저장소는 프록시되지 않습니다. 값이 T인 HTTP 저장소는 원본이 pkg set-publisher 명령으로 추가된 경우 --proxy 옵션에 지정된 프록시를 사용하여 프록시됩니다. pkg publisher에 -F tsv 옵션을 지정하면 PROXY 열에는 해당 위치에 대해 설정된 프록시가 표시됩니다.

```
$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE    STATUS URI                                PROXY
solaris   true  false true   origin online http://pkg.oracle.com/solaris/release/ -
isvpub    false false true   origin online file:///export/IPSpkgrepos/isvrepo/ -
devtool   true  false false  origin online http://pkg.example1.com/      -
```

P 열의 F 또는 PROXY 열의 -는 해당 위치가 pkg set-publisher 명령을 사용하여 프록시되지 않았음을 나타냅니다. http\_proxy 환경 변수를 설정하여 위치가 프록시된 경우 pkg publisher 출력에 여전히 F 또는 -가 표시됩니다. 다양한 프록시 설정 방법에 대한 자세한 내용은 “[프록시 지정](#) [72]을 참조하십시오.

해당 게시자에 대한 자세한 구성 정보를 표시하려면 게시자 이름을 지정합니다.

```
$ pkg publisher solaris
Publisher: solaris
Alias:
Origin URI: http://pkg.oracle.com/solaris/release/
SSL Key: None
SSL Cert: None
Client UUID: e15e3228-eada-11df-80ab-8023183d954b
Catalog Updated: March 4, 2014 11:48:02 PM
Enabled: Yes
Properties:
    proxied-urls = []
```

게시자 검색 순서의 첫번째 게시자만 표시하려면 -P 옵션을 사용합니다.

```
$ pkg publisher -P
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
```

사용으로 설정된 게시자만 표시하려면 -n 옵션을 사용합니다.

```
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
```

```
solaris          origin online F http://pkg.oracle.com/solaris/release/
isvpub          (non-sticky) origin online F file:///export/IPSpkgrepos/isvrepo/
```

## 패키지 게시자 추가, 수정 또는 제거

`pkg set-publisher` 명령을 사용하여 다음 작업을 수행할 수 있습니다.

- 새 게시자를 구성합니다
- 게시자 원본 및 미러를 설정합니다
- 게시자 고착성을 설정합니다.
- 게시자 검색 순서를 설정합니다
- 게시자 등록 정보를 설정 및 해제하고, 게시자 등록 정보 값을 추가 및 제거합니다
- 게시자의 SSL 키와 인증서를 지정합니다
- 게시자 프록시를 설정합니다.
- 게시자를 사용으로 설정하거나 사용 안함으로 설정합니다
- 게시자를 제거합니다.

`pkg set-publisher` 명령의 형식은 두 가지입니다. 자세한 내용은 [pkg\(1\)](#) 매뉴얼 페이지를 참조하십시오.

- 한 형식에서 게시자 이름은 필수 피연산자입니다.
- 다른 형식에서 저장소 URI가 `-p` 옵션의 인수로 제공되며 지정된 저장소에서 게시자 정보가 검색됩니다. 게시자 이름은 선택적 피연산자이므로 여러 게시자가 해당 저장소로 패키지를 게시하는 경우 명명된 게시자만 구성할 수 있습니다.

### 게시자 추가

아래 예제는 게시자를 추가하는 두 가지 방법을 보여줍니다.

예 5-1 새 게시자 지정

다음 명령은 `-g` 옵션을 사용하여 원본 URI가 지정된 `devtool`이라는 새 게시자를 추가하고 이 게시자를 검색 순서의 첫번째 게시자로 설정합니다. 지정한 게시자를 검색 순서의 첫번째 게시자로 설정하려면 `-p` 옵션이나 `--search-first` 옵션을 사용합니다.

```
$ pkg set-publisher -P -g http://pkg.example1.com/release/ devtool
```

예 5-2 게시자 구성 가져오기

지정한 저장소 URI에서 게시자 구성 정보를 검색하려면 `-p` 옵션을 사용합니다. 게시자를 지정하면 일치하는 게시자만 추가되거나 업데이트됩니다. 게시자를 지정하지 않으면 모든 게시자가 추가되거나 업데이트됩니다.

```
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
$ pkg set-publisher -p /export/IPSpkgrepos/myrepo
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
site               origin  online F file:///export/IPSpkgrepos/myrepo/
```

## 게시자 원본 추가 및 변경

다음 명령은 solaris 게시자에 원본을 추가하는 방법을 보여줍니다. 이미지의 지정된 게시자에 대해 원본이 여러 개 구성된 경우 IPS 클라이언트는 패키지 데이터를 검색할 수 있는 최상의 원본을 선택하려고 시도합니다.

```
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
$ pkg set-publisher -g /export/IPSpkgrepos/Solaris solaris
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
solaris            origin  online F file:///export/IPSpkgrepos/Solaris/
```

URI를 지정한 게시자의 원본으로 제거하려면 -G 옵션을 사용합니다.

게시자의 원본 URI를 변경하려면 새 URI를 추가하고 기존 URI를 제거합니다.

```
$ pkg set-publisher -G '*' -g file:///export/IPSpkgrepos/isvrepo/ isvpub
```

## 게시자 미러 추가 및 변경

URI를 지정한 게시자의 미러로 추가하려면 -m 옵션을 사용합니다. 원본과 미러의 차이점에 대한 설명은 “저장소 원본 및 미러” [15]를 참조하십시오. 동일한 게시자의 원본 저장소에도 동일 패키지의 동일 버전이 존재하지 않으면 미러 저장소의 콘텐츠에 액세스할 수 없습니다.

```
$ pkg set-publisher -m http://pkg.example3.com/ devtool
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
devtool            origin  online F http://pkg.example1.com/
devtool            mirror  online F http://pkg.example3.com/
```

URI를 지정한 게시자의 미러로 제거하려면 -M 옵션을 사용합니다.

게시자의 미러 URI를 변경하려면 새 URI를 추가하고 기존 URI를 제거합니다.

## 게시자 검색 순서 및 고착성 설정

새로 추가한 게시자는 기본적으로 고정되어 있습니다. 게시자가 비고정인 경우 이 게시자에서 설치된 패키지를 다른 게시자에서 업데이트할 수 있습니다. `--sticky` 및 `--non-sticky` 옵션을 사용하여 게시자 고착성을 설정할 수 있습니다.

새로 추가한 게시자는 기본적으로 검색 순서에서 마지막 게시자가 됩니다. 게시자 검색 순서는 설치할 패키지를 찾는 데 사용됩니다. 패키지 설치에 사용된 원본 게시자가 비고정인 경우에는 업데이트할 패키지를 찾는 데 게시자 검색 순서가 사용됩니다. 게시자 검색을 변경하려면 `--search-before`, `--search-after`, `--search-first` 옵션을 사용합니다. `-P` 옵션은 `--search-first` 옵션의 동의어입니다.

일치하는 패키지를 제공하는 첫번째 게시자는 설치 소스로 사용됩니다. 해당 게시자가 이 이미지에 설치할 수 있는 패키지 버전을 제공하지 않으면 설치 작업이 실패합니다. 검색 순서에서 더 아래쪽에 있는 게시자에서 설치하려면 패키지 FMR에 게시자 이름 또는 패키지 버전 문자열과 같은 추가 정보를 제공합니다.

## 게시자 등록 정보 구성

다음 옵션을 사용하여 게시자 등록 정보를 설정 및 해제하고, 게시자 등록 정보 값을 추가 및 제거합니다.

- `--set-property property=value`
- `--add-property-value property=value`
- `--remove-property-value property=value`
- `--unset-property property`

`publisher-search-order` 및 `signature-required-names` 등록 정보는 여러 값을 취할 수 있습니다.

“패키지 서명 등록 정보 구성” [89]에서 `pkg set-publisher` 예제를 참조하십시오.

## 게시자 키와 인증서 구성

예 5-3            게시자 키와 인증서 지정

클라이언트 SSL 키를 지정하려면 `-k` 옵션을 사용합니다. 클라이언트 SSL 인증서를 지정하려면 `-c` 옵션을 사용합니다. 지정된 인증서를 신뢰하는 CA 인증서로 추가하려면 `--approve-ca-cert` 옵션을 사용합니다. 사용자가 승인한 CA 인증서의 해시는 이 게시자에 대한 `pkg publisher` 명령의 출력 결과에 나열됩니다. “게시자 정보 표시” [67]를 참조하십시오.

```
$ pkg set-publisher -k /root/creds/example.key -c /root/creds/example.cert \
--approve-ca-cert /tmp/example_file.pem isvpub
```

**예 5-4**            게시자 키와 인증서 취소

지정한 인증서를 취소된 것으로 처리하려면 `--revoke-ca-cert` 옵션을 사용합니다. 사용자가 취소한 CA 인증서의 해시는 이 게시자에 대한 `pkg publisher` 명령의 출력 결과에 나열됩니다.

지정한 인증서를 승인된 인증서 목록 및 취소된 인증서 목록에서 제거하려면 `--unset-ca-cert` 옵션을 사용합니다.

## 게시자 프록시 구성

지정된 원본 또는 미래의 콘텐츠를 검색할 영구 프록시 URI를 지정하려면 `--proxy` 옵션을 사용합니다. 프록시 값은 `protocol://host[:port]`입니다. 여기서 `protocol`은 `http` 또는 `https`이고 `:port`는 선택 사항입니다. 다양한 프록시 설정 방법에 대한 자세한 내용은 “[프록시 지정](#)” [72]을 참조하십시오.

## 게시자 사용 및 사용 안함

새로 추가한 게시자는 기본적으로 사용으로 설정됩니다. 사용 안함으로 설정된 게시자는 패키지 목록을 채울 때나 `install`, `uninstall`, `update` 패키지 작업 중에 사용되지 않습니다. 사용 안함으로 설정된 게시자에 대해 등록 정보를 설정하거나 볼 수는 있습니다. 사용으로 설정된 게시자가 하나뿐인 경우 해당 게시자를 사용 안함으로 설정할 수 없습니다.

다음 명령은 `isvpub` 게시자를 사용으로 설정하고 검색 순서에서 `devtool`의 앞에 나오도록 설정합니다.

```
$ pkg set-publisher --enable --search-before devtool isvpub
```

`--disable` 옵션을 사용하여 게시자를 사용 안함으로 설정합니다. 예를 들어, 게시자 원본에 일시적으로 연결할 수 없는 경우 게시자를 사용 안함으로 설정할 수 있습니다. 게시자에 연결할 수 없으면 패키지 설치 및 업데이트 작업을 실패합니다.

## 게시자 제거

게시자를 제거하려면 `pkg unset-publisher` 명령을 사용합니다.

```
$ pkg unset-publisher devtool
```

## 프록시 지정

프록시 설정 방법마다 다른 효과와 장점이 있습니다. 예를 들어, `pkg set-publisher` 명령은 게시자 구성의 일부로 프록시를 저장하는 반면, `http_proxy` 환경 변수는 인증된 프록시를 설정할 수 있습니다.



## pkg set-publisher 명령을 사용하여 프록시 설정

pkg set-publisher 명령의 --proxy 옵션은 지정된 게시자 원본 및 미러 URI에 대한 영구 프록시 URI를 설정합니다. 프록시 값은 게시자 구성의 일부로 저장됩니다. 게시자 구성의 일부로 프록시 값을 저장하면 자식 이미지에 사용된 시스템 저장소가 자동으로 업데이트됩니다. 또한 게시자 구성의 일부로 프록시 값을 저장하면 다른 게시자마다 다른 프록시를 사용할 수 있습니다.

```
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F file:///export/IPSpkgrepos/Solaris/

$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE      STATUS URI                                PROXY
solaris   true   false true   origin online file:///export/IPSpkgrepos/Solaris/ -

$ pkg set-publisher -g http://pkg.oracle.com/solaris/release/ --proxy proxyURI solaris
$ pkg publisher solaris
  Publisher: solaris
  Alias:
  Origin URI: file:///export/IPSpkgrepos/Solaris/
  SSL Key: None
  SSL Cert: None
  Origin URI: http://pkg.oracle.com/solaris/release/
  Proxy: proxyURI
  SSL Key: None
  SSL Cert: None
  Client UUID: e15e3228-eada-11df-80ab-8023183d954b
  Catalog Updated: July 11, 2013 11:32:46 PM
  Enabled: Yes
  Properties:
    proxied-urls = []

$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F file:///export/IPSpkgrepos/Solaris/
solaris            origin   online T http://pkg.oracle.com/solaris/release/

$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE      STATUS URI                                PROXY
solaris   true   false true   origin online file:///export/IPSpkgrepos/Solaris/ -
solaris   true   false true   origin online http://pkg.oracle.com/solaris/release/ proxyURI
```

이 이미지에 비전역 영역이 있으면 이 프록시 정보로 시스템 저장소가 자동으로 업데이트되므로 system-repository 서비스에 등록 정보를 설정할 필요가 없습니다. 비전역 영역에서 게시자 프록시를 검사하면 전역 영역에 보이는 프록시 URI가 보이지 않습니다. 전역 영역에서 시스템 저장소는 프록시 URI를 사용합니다. 비전역 영역에서 시스템 저장소는 자체가 프록시 역할을 하므로 비전역 영역에서 전역 영역의 시스템 저장소와 통신할 수 있습니다. “[전역 영역과 비전역 영역의 관계](#)” [50]는 시스템 저장소 게시자가 비전역 영역에서 어떻게 보이는지 예시합니다.

pkg set-publisher 명령의 --proxy 옵션은 인증된 프록시를 설정하는 데 사용할 수 없습니다. --proxy 옵션 값은 `protocol://user:password@host` 형식을 가질 수 없습니다.

## 환경 변수를 사용하여 프록시 설정

프록시 환경 변수 값은 해당 프로토콜의 모든 URI에 적용됩니다. 런타임에 `http_proxy` 환경 변수 값은 `pkg set-publisher` 명령의 `--proxy` 옵션으로 설정된 값을 대체합니다. 프록시 환경 변수에 대한 자세한 내용은 `curl(1)` 매뉴얼 페이지의 ENVIRONMENT 절을 참조하십시오.

비전역 영역이 있는 이미지에 `http_proxy` 환경 변수를 설정하는 경우, 전역 영역에서 `svc:/application/pkg/system-repository` SMF 서비스의 프록시 등록 정보를 이와 같은 값으로 설정하고 서비스를 새로 고칩니다.

```
$ svccfg -s system-repository:default setprop config/http_proxy = astring: proxyURI
$ svccfg -s system-repository:default listprop config/*proxy
config/https_proxy astring
config/http_proxy astring proxyURI
$ svcprop system-repository:default | grep proxy
config/https_proxy astring ""
config/http_proxy astring ""
$ svcadm refresh system-repository:default
$ svcprop system-repository:default | grep proxy
config/https_proxy astring ""
config/http_proxy astring proxyURI
```

`pkg publisher` 명령은 환경 변수나 SMF 서비스 등록 정보로 설정된 프록시를 표시하지 않습니다.

`http_proxy` 환경 변수 값을 변경하면 `system-repository` 서비스 등록 정보를 업데이트하고 서비스를 새로 고쳐야 합니다.

## 선택적 구성 요소의 설치 제어

소프트웨어에는 선택적 구성 요소와 상호 배타적인 구성 요소가 포함될 수 있습니다. 선택적 구성 요소의 예로는 로켈과 설명서가 있습니다. 상호 배타적 구성 요소의 예로는 SPARC 또는 x86 및 디버그 또는 비디버그 이진이 있습니다. IPS에서는 선택적 구성 요소를 페이스잇이라 하고 상호 배타적 구성 요소를 변형이라고 합니다.

현재 이미지에 설정된 변형 및 페이스잇의 값을 표시할 수 있으며 현재 이미지의 변형 및 페이스잇을 변경할 수 있습니다. 이미지에 설정된 페이스잇과 변형의 현재 값을 보려면 `pkg facet` 및 `pkg variant` 명령을 사용합니다. 이미지에 설정된 페이스잇과 변형의 값을 수정하려면 `pkg change-facet` 및 `pkg change-variant` 명령을 사용합니다. [pkg\(1\)](#) 매뉴얼 페이지와 아래 예제를 참조하십시오. 변형과 페이스잇을 변경하면 다수의 패키지가 업데이트될 수 있으며 이 경우 새로운 BE가 필요할 수도 있습니다. 변경하기 전에 어떻게 변경될지 검토하려면 `-nv`를 사용합니다.

## 변형 및 페이지 값이 패키지 설치에 영향을 미치는 방식

페이지 및 변형은 이미지에 설정된 특수한 등록 정보이고 패키지 매니페스트의 작업에 설정된 태그입니다. 작업의 페이지 및 변형 태그 값을 이미지에 설정된 페이지 및 변형 값과 비교한 결과에 따라, 패키지 작업을 설치할 수 있는지 여부가 결정됩니다.

각 페이지 및 변형 태그에는 이름과 값이 있습니다. 단일 작업에 페이지 및 변형 태그를 여러 개 지정할 수 있습니다. 페이지 및 변형 태그를 여러 개 가진 구성 요소의 예로는 개발자가 사용하는 아키텍처 관련 헤더 파일이나 SPARC 전역 영역 전용 구성 요소가 있습니다.

대부분의 변형 태그는 값이 다양할 수 있습니다. 변형 값을 지정하는 패키지가 설치되도록 하려면 변형 태그의 값을 이미지에 설정해야 합니다. arch 및 zone 변형은 이미지를 만들어 초기 콘텐츠를 설치하는 프로그램에 의해 설정됩니다. debug.\* 변형은 기본적으로 이미지에서 false입니다.

작업에 설정된 페이지 태그는 true 값만 가질 수 있습니다. 이미지에서 특정 페이지 값을 false로 설정하는 경우 해당 페이지를 지정하는 파일 또는 기타 작업은 설치되지 않고 해당 페이지를 지정하는 현재 설치된 파일은 제거됩니다.

다음 알고리즘에서는 이미지에 설정된 페이지 및 변형이 특정 작업의 설치 여부에 어떤 영향을 주는지 설명합니다.

- 페이지 또는 변형 태그가 없는 작업은 항상 설치됩니다.
- 페이지 태그가 있는 작업은 태그와 일치하는 모든 페이지 또는 페이지 패턴이 이미지에 false로 설정되지 않는 한 설치됩니다. 어떤 페이지가 true로 설정되거나 명시적으로 설정되지 않은 경우(true가 기본값) 작업이 설치됩니다.
- 변형 태그가 있는 작업은 모든 변형 태그의 값이 이미지에 설정된 값과 동일한 경우에만 설치됩니다.
- 페이지 및 변형 태그가 모두 있는 작업은 페이지와 변형이 모두 작업 설치를 허용하는 경우 설치됩니다.

## 변형 및 페이지 값의 예

대부분의 변형은 원하는 수의 값을 가질 수 있습니다. 예를 들어, arch 변형을 i386, sparc, ppc, arm 또는 배포판이 지원하는 어떤 구조로든 설정할 수 있습니다. (i386 및 sparc만 Oracle Solaris에 사용됩니다.) 한가지 예외는 debug 변형입니다. debug 변형은 true 또는 false로만 설정할 수 있습니다. 다른 값은 정의되지 않은 동작을 일으킵니다. file 작업에 비디버그 및 디버그 버전이 모두 있는 경우, 다음 예제에 표시된 대로 양쪽 버전에 적용 가능한 debug 변형을 명시적으로 설정해야 합니다.

```
file group=sys mode=0644 overlay=allow owner=root \
  path=etc/motd pkg.csize=115 pkg.size=103 preserve=true \
  variant.debug.osnet=true
```

```
file group=sys mode=0644 overlay=allow owner=root \
  path=etc/motd pkg.csize=68 pkg.size=48 preserve=true \
  variant.debug.osnet=false
```

다음은 Oracle Solaris에서 흔히 사용되는 변형 태그입니다.

변형 이름	가능한 값
variant.arch	sparc, i386
variant.opensolaris.zone	global, nonglobal
variant.debug.*	true, false

페이킷은 부울이므로 true(사용) 또는 false(사용 안함)로만 설정할 수 있습니다. 기본적으로 모든 페이킷은 이미지에 true로 설정된 것으로 간주됩니다. 작업의 페이킷 태그는 true 값만 가져야 합니다. 다른 값은 정의되지 않은 동작을 일으킵니다.

이미지에 설정된 페이킷은 doc.man과 같은 전체 페이킷이거나 locale.\*와 같은 패턴일 수 있습니다. 이러한 유연성은 페이킷 이름 공간의 일부분을 사용 안함으로 설정하고 그 안의 개별 페이킷만 사용으로 설정하려는 경우 유용합니다. 예를 들어, 다음 예제에 표시된 대로 모든 로컬을 사용 안함으로 설정한 후 한두 개의 특정 로컬만 사용으로 설정할 수 있습니다.

```
$ pkg change-facet 'locale.*=false'
[output about packages being updated]
$ pkg change-facet locale.en_US=true
[output about packages being updated]
```

다음 목록은 Oracle Solaris에서 사용되는 페이킷 태그의 작은 샘플을 보여줍니다.

```
facet.devel          facet.doc
facet.doc.html       facet.doc.info
facet.doc.man        facet.doc.pdf
facet.locale.de      facet.locale.en_GB
facet.locale.en_US   facet.locale.fr
facet.locale.ja_JP   facet.locale.zh_CN
```

## 변형 값 표시 및 변경

설정된 변형의 값을 표시하려면 pkg variant 명령을 사용합니다.

```
$ pkg variant
VARIANT          VALUE
variant.arch     i386
variant.opensolaris.zone global
```

-v 옵션에는 설치된 패키지에 설정할 수 있는 가능한 변형 값이 포함됩니다.

```
$ pkg variant -v
VARIANT          VALUE
variant.arch     i386
variant.arch     sparc
variant.debug.osnet false
variant.debug.osnet true
variant.opensolaris.zone global
```

```
variant.opensolaris.zone nonglobal
```

pkg change-variant 명령을 사용하여 변형의 값을 변경할 수 있습니다. pkg variant -v 명령을 사용하여 설정할 값을 선택할 수 있습니다.

다음 명령은 대량의 출력 결과를 생성하는데 이는 영향을 받는 패키지가 너무 많기 때문입니다. 새 BE는 기본적으로 생성되지 않지만 대신 백업 BE가 생성됩니다. BE 생성 시점에 대한 자세한 내용은 “부트 환경 정책 이미지 등록 정보” [86]를 참조하십시오. -n 옵션은 -n 없이 작업을 수행할 경우 어떻게 변경될지 보여주지만, 명령이 실제로 변경하는 것은 없습니다.

```
$ pkg change-variant -nv --accept 'variant.debug.*=true'
      Packages to update:      851
      Variants/Facets to change:  3
      Estimated space available: 49.88 GB
      Estimated space to be consumed: 270.57 MB
      Create boot environment:    No
      Create backup boot environment: Yes
      Rebuild boot archive:      No

Changed variants/facets:
  variant variant.debug.*: true
  facet facet.locale.en_US: None
  facet facet.locale.*: None

Changed packages:
solaris
...
```

## 페이셋 값 표시 및 변경

pkg facet 명령을 사용하여 pkg change-facet 명령을 통해 이 이미지에 로컬로 설정되었거나 부모 이미지로부터 상속받은 모든 페이셋의 현재 값과 소스를 표시할 수 있습니다. 예를 들어, 비전역 영역은 전역 영역으로부터 페이셋 설정을 상속받습니다.

```
$ pkg facet
FACETS          VALUE
facet.locale.en_US  True
facet.locale.en    True
facet.locale.*     False
```

pkg change-facet 명령을 사용하여 페이셋의 값을 변경할 수 있습니다.

페이셋 값이 None으로 설정되어 있으면 페이셋 사양이 현재 이미지에서 제거됩니다.

다음 명령은 대량의 출력 결과를 생성하는데 이는 영향을 받는 패키지가 너무 많기 때문입니다. 새 BE는 기본적으로 생성되지 않지만 대신 백업 BE가 생성됩니다. BE 생성 시점에 대한 자세한 내용은 “부트 환경 정책 이미지 등록 정보” [86]를 참조하십시오. -n 옵션은 -n 없이 작업을 수행할 경우 어떻게 변경될지 보여주지만, 명령이 실제로 변경하는 것은 없습니다.

```
$ pkg change-facet -nv 'facet.locale.*=true'
      Packages to update:      851
```

```

Variants/Facets to change:      1
Estimated space available: 49.88 GB
Estimated space to be consumed: 3.13 GB
Create boot environment:       No
Create backup boot environment: Yes
Rebuild boot archive:         No

Changed variants/facets:
facet facet.locale.*: True
Changed packages:
solaris
...

```

## 지정한 버전으로 패키지 잠금

패키지 버전을 제한하려면 `pkg freeze` 명령을 사용합니다.

패키지 피연산자에 버전을 제공하지 않은 경우 명명된 패키지가 설치되어야 하며 시스템에 설치된 버전으로 제약됩니다. 패키지 피연산자에 버전을 제공한 경우 이 제약 조건이나 고정 의 동작은 `incorporate` 종속성이 설치되었을 때와 같습니다(`fmri` 속성 값이 지정된 패키지 버전일 경우).

고정된 패키지가 설치되거나 업데이트되는 경우 고정된 시점의 버전과 일치하는 버전으로 설치 또는 업데이트되어야 합니다. 예를 들어, 패키지가 1.2로 고정된 경우 1.2.1, 1.2.9, 1.2.0.0.1 등으로는 업데이트될 수 있지만 1.3이나 1.1은 될 수 없습니다.

일치하는 패키지를 찾는 데는 패키지 피연산자에 지정된 게시자가 사용됩니다. 그러나 게시자 정보는 고정의 일부로 기록되지 않습니다. 패키지는 게시자가 아니라 해당 버전만 고려하여 고정됩니다.

이미 고정된 패키지를 고정하면 고정된 버전은 새로 지정한 버전으로 교체됩니다.

패키지를 지정하지 않으면 패키지 이름, 고정된 버전, 패키지가 고정된 시간, 패키지가 고정된 이유 등 현재 고정된 패키지에 대한 정보가 표시됩니다.

패키지를 고정해도 패키지를 제거할 수 있습니다. 패키지를 제거하면 별도의 경고가 표시되지 않습니다.

다음 예에서는 패키지가 현재 설치된 버전에서 고정됩니다. `-c` 옵션 인수는 패키지가 고정된 이유를 나타냅니다. 고정으로 인해 설치나 업데이트가 실패할 경우 이유가 표시됩니다. 패키지 목록에서 "f"는 패키지가 고정되었음을 나타냅니다.

```

$ pkg freeze -c "Downgrade to avoid bug" library/security/openssl
library/security/openssl was frozen at 1.0.0.10-0.175.1.0.0.18.0:20120611T201116Z
$ pkg freeze
NAME                                VERSION                                DATE
COMMENT
library/security/openssl 1.0.0.10-0.175.1.0.0.19.0:20120625T171753Z 29 Jul 2012 17:45:44 PDT
Downgrade to

```

```
avoid bug
$ pkg list library/security/openssl
NAME (PUBLISHER)          VERSION          IFO
library/security/openssl 1.0.0.10-0.175.1.0.0.18.0 if-
```

고정된 패키지의 다른 버전을 설치하려고 하면 고정에 대한 메시지가 표시됩니다.

```
$ pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.20.0
Creating Plan (Solver setup): -
pkg update: No matching version of library/security/openssl can be installed:
  Reject: pkg://solaris/library/security/
openssl@1.0.0.10,5.11-0.175.1.0.0.20.0:20120709T180243Z
  Reason: This version is excluded by a freeze on library/security/openssl at version
1.0.0.10,5.11-0.175.1.0.0.18.0:20120611T201116Z.
  The reason for the freeze is: Downgrade to avoid bug
```

고정은 패키징 시스템에 의해 자동으로 시작되지 않습니다. pkg unfreeze 명령을 사용하여 지정한 패키지에서 고정에 따른 제약 조건을 제거할 수 있습니다. 제공한 버전은 모두 무시됩니다.

## 통합에서 지정한 버전 제약 조건 완화

통합 패키지는 어떤 버전의 패키지를 설치할 수 있는지 지정합니다. 이러한 버전 제약 조건은 시스템이 업데이트 간에 지원 가능한 상태로 유지되도록 도와줍니다. 통합 패키지 및 제약 조건에 대한 자세한 내용은 [“통합 패키지” \[12\]](#)를 참조하십시오.

일부 통합된 패키지는 통합에서 지정한 버전과 다른 버전으로 다운그레이드하거나 업그레이드해도 됩니다. 버전 제약 조건은 통합 패키지에 지정된 `version-lock.package` 페이지셋으로 표현됩니다. `version-lock.package` 페이지셋의 기본값은 true입니다. 패키지에 대한 버전 제약 조건을 완화하려면 `version-lock.` 페이지셋 값을 false로 설정합니다.

다음 예제에서 패키지의 이전 버전으로 다운그레이드하려고 합니다. pkg update 명령은 패키지를 업그레이드할 뿐 아니라 다운그레이드하기도 합니다.

```
$ pkg list -af library/security/openssl
NAME (PUBLISHER)          VERSION          IFO
library/security/openssl 1.0.1.5-0.175.2.0.0.24.0 i--
library/security/openssl 1.0.1.5-0.175.2.0.0.23.0 ---
$ pkg update library/security/openssl@1.0.1.5-0.175.2.0.0.23.0
Creating Plan (Solver setup): |
pkg update: No matching version of library/security/openssl can be installed:
  Reject: pkg://solaris/library/security/
openssl@1.0.1.5,5.11-0.175.2.0.0.23.0:20130916T191702Z
  Reason: This version is excluded by installed incorporation
pkg://solaris/consolidation/userland/userland-
incorporation@0.5.11,5.11-0.175.2.0.0.24.0:20131001T160408Z
```

pkg contents 명령은 이 버전 제약 조건이 어떻게 설정되었는지 보여줍니다. 이 패키지에 대한 버전 제약 조건을 완화하려면 `version-lock` 페이지셋을 false로 설정합니다. 그런 다음 다

시 다운그레이드해 봅니다. 새 BE가 생성되지 않고 백업 BE가 생성됩니다. BE 생성 시점에 대한 자세한 내용은 “부트 환경 정책 이미지 등록 정보” [86]를 참조하십시오.

```
$ pkg contents -m userland-incorporation | grep 'library/security/openssl'
depend facet.version-lock.library/security/openssl=true
fmri=pkg:/library/security/openssl@1.0.1.5-0.175.2.0.0.24.0 type=incorporate
$ pkg change-facet facet.version-lock.library/security/openssl=false
    Packages to update: 850
    Variants/Facets to change: 1
    Create boot environment: No
Create backup boot environment: Yes
```

```
PHASE                                ITEMS
Removing old actions                  1/1
Updating image state                  Done
Creating fast lookup database         Done
Reading search index                 Done
Building new search index             850/850
```

```
$ pkg update library/security/openssl@1.0.1.5-0.175.2.0.0.23.0
    Packages to update: 1
    Create boot environment: No
Create backup boot environment: Yes
```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	1/1	10/10	1.6/1.6	0B/s

```
PHASE                                ITEMS
Removing old actions                  3/3
Installing new actions                3/3
Updating modified actions             14/14
Updating package state database       Done
Updating package cache                1/1
Updating image state                  Done
Creating fast lookup database         Done
Reading search index                  Done
Updating search index                 1/1
```

```
$ pkg list library/security/openssl
NAME (PUBLISHER)                      VERSION                                IFO
library/security/openssl              1.0.1.5-0.175.2.0.0.23.0             i--
```

이 패키지가 다운그레이드나 업그레이드되지 않게 하려면 패키지를 현재 버전에 고정합니다. 패키지 목록에서 "f"는 패키지가 고정되었음을 나타냅니다.

```
$ pkg freeze -c "Downgrade to avoid bug" library/security/openssl
library/security/openssl was frozen at 1.0.1.5,5.11-0.175.2.0.0.23.0:20130916T191702Z
$ pkg list library/security/openssl
NAME (PUBLISHER)                      VERSION                                IFO
library/security/openssl              1.0.1.5-0.175.2.0.0.23.0             if-
```

다운그레이드 또는 업그레이드를 다시 사용으로 설정하려면 pkg unfreeze 명령을 사용하여 버전 고정을 제거합니다. 패키지가 통합 패키지에 지정된 버전보다 낮은 버전으로 설치된 경우 이 패키지의 version-lock 페이지를 true로 설정하면 통합 패키지에 지정된 버전이 설치됩니다.



설치된 다른 패키지가 다운그레이드 또는 업그레이드할 패키지와 require 종속성 관계가 있는 경우 이러한 관련 패키지에 대한 버전 제약 조건도 해제해야 합니다. 다음 예제에서 hexedit 패키지에 대한 버전 제약 조건이 제거되었지만 system/library 패키지에 대한 버전 제약 조건 때문에 설치가 거부됩니다.

```
$ pkg install editor/hexedit@1.2.12-0.175.2.0.0.25.0
Creating Plan (Solver setup): -
pkg install: No matching version of editor/hexedit can be installed:
  Reject: pkg://solaris/editor/hexedit@1.2.12-0.175.2.0.0.25.0:20131014T170634Z
  Reason: All versions matching 'require' dependency
pkg://system/library@0.5.11,5.11-0.175.2.0.0.24.0 are rejected
  Reject: pkg://solaris/system/library@0.5.11,5.11-0.175.2.0.0.24.0:20131001T152820Z

pkg://solaris/system/library@0.5.11,5.11-0.175.2.0.0.25.0:20131014T161136Z
  Reason: This version is excluded by installed incorporation
pkg://solaris/consolidation/osnet/osnet-incorporation@0.5.11,5.11-0.175.2.0.0.24.0:20131001T150429Z
```

개별 구성 요소 패키지 외에 통합에 대한 버전 제약 조건도 해제할 수 있습니다. 이 경우 version-lock 페이지를 false 로 설정하면 나머지 시스템에서 통합의 잠금을 해제할 수 있습니다. 통합 패키지의 잠금이 해제되어도 여기에 통합되는 패키지는 계속 동기화됩니다.

## 기본 응용 프로그램 구현 지정

여러 버전의 응용 프로그램 또는 도구를 동일한 이미지에 사용할 수 있습니다. 각 응용 프로그램 버전은 전체 경로를 지정하여 사용자에게 제공됩니다. 한 버전을 기본 버전이라 부르고 /usr/bin과 같은 공통 디렉토리에서 쉽게 사용할 수 있습니다. 아래 설명된 대로 모든 버전이 동일한 중개에 참여하는 경우 기본 버전으로 삼을 버전을 쉽게 재설정할 수 있습니다. 이 관리 선택 사항은 패키지 업데이트 사이에 유지됩니다.

중개란, 각 링크의 링크 경로는 같지만 각 링크의 대상이 다른 링크 세트입니다. 예를 들어, 링크 경로가 /usr/bin/myapp이고 링크 대상은 /usr/myapp/myapp1/bin/myapp 및 /usr/myapp/myapp2/bin/myapp일 수 있습니다. 중개의 각 링크를 중개의 참가자라고 합니다. /usr/bin/myapp이 현재 myapp1을 호출하는 경우, /usr/bin/myapp이 myapp2를 호출하도록 선택을 쉽게 변경할 수 있습니다. 현재 링크의 대상인 소프트웨어 버전이 기본 버전입니다.

## 중개의 참가자 식별

pkg mediator 명령을 사용하여 이미지의 모든 중개 링크의 기본 버전을 표시할 수 있습니다.

다음 출력에서 MEDIATOR는 동일한 기본 링크 경로를 공유하는 링크 세트의 이름입니다. VER. SRC. 및 IMPL. SRC.는 기본 버전이 시스템에서 선택되었는지, 지정된 우선 순위에서 선택되었는지(vendor 또는 site), 관리자가 설정했는지(local) 여부를 보여줍니다. VERSION은 선택한 중개 참가자의 버전으로, 링크가 가리키는 소프트웨어 버전과 비슷해야 합니다.

VERSION은 패키지 개발자가 설정합니다. IMPLEMENTATION은 버전 문자열과 함께 또는 버전 문자열 대신 패키지 개발자가 설정할 수 있는 문자열입니다.

```
$ pkg mediator
MEDIATOR  VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
gcc-runtime system 4.7 system
java      system 1.7 system
php       system 5.2 system
python    vendor 2.6 vendor
ruby      system 1.9 system
```

-a 옵션은 모든 중개 참가자를 보여줍니다. 다른 기본 버전을 선택하려면 이 옵션을 사용하여 선택 사항을 표시합니다. 다음 예제는 java 중개의 모든 참가자를 보여줍니다. system 키워드는 이 중개의 기본 버전이 패키지 우선 순위 설정으로 지정되지 않았고, 관리자가 설정한 것도 아니며, 패키징 시스템이 더 높은 VERSION 값을 선호 버전으로 선택했음을 나타냅니다.

```
$ pkg mediator -a java
MEDIATOR  VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      system 1.7 system
java      system 1.6 system
```

다음 출력에서 두 개의 다른 버전의 Java Runtime Environment가 이 이미지에 설치되었고 1.7.0\_51 버전이 현재 선택된 기본 버전임을 알 수 있습니다.

```
$ pkg list -s '*jre*'
NAME (PUBLISHER)      SUMMARY
runtime/java/jre-6   Java(TM) Platform Standard Edition Runtime Environment (1.6.0_71-b12)
runtime/java/jre-7   Java Platform Standard Edition Runtime Environment (1.7.0_51-b13)
$ java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
Java HotSpot(TM) Server VM (build 24.51-b03, mixed mode)
```

jre-6 및 jre-7 패키지는 모두 /usr/bin/java 경로를 가진 심볼릭 링크를 정의합니다. jre-6 패키지에서 링크의 대상은 jdk1.6.0입니다. jre-7 패키지에서 링크의 대상은 jdk1.7.0입니다. 위의 pkg mediator 및 java -version 명령은 버전 1.7이 현재 기본 버전이자 /usr/bin/java 링크의 대상임을 보여줍니다.

## 기본 응용 프로그램 변경

기본 버전인 지정된 중개의 버전을 재설정하려면 pkg set-mediator 명령을 사용합니다.

pkg mediator -a의 출력을 사용하여 -v 인수에 버전 또는 -I 인수에 구현을 선택합니다. 맞출법 오류가 있거나 현재 사용할 수 없는 중개자 버전이나 구현을 지정할 경우, 지정된 중개자를 사용하는 링크가 제거됩니다.

set-mediator 하위 명령에 -n 옵션을 함께 사용하여 백업 BE가 만들어지는지 여부를 확인할 수 있습니다. 백업 BE가 만들어지지 않으면 set-mediator 하위 명령에 --require-backup-be 옵션을 지정할 수 있습니다. 중개자 변경은 현재 BE에서 수행됩니다. 중개자 변경 후 현재

BE에 아무 문제가 없는 것을 확인했으면 `beadm destroy`를 사용하여 백업 BE를 삭제할 수 있습니다.

이전 출력에서는 버전 1.7이 java 중개에서 현재 선택된 기본 버전이었습니다. 다음 명령은 버전 1.6을 기본 버전으로 설정하는 것을 보여줍니다. 즉, `/usr/bin/java`를 호출하면 JRE 버전 1.6이 호출됩니다. 해당 버전에 대한 전체 경로를 지정하는 경우 JRE 버전 1.7을 계속 사용할 수 있습니다. 두 `pkg mediator` 명령의 출력을 비교하십시오. 중개의 기본 버전을 변경하면 `VER. SRC.`도 `local`로 바뀌면서 관리자가 선택을 지정했음을 나타냅니다. 이 선택 사항은 재부트와 패키지 업데이트 사이에 유지됩니다.

```
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      system  1.7      system
$ pkg mediator -a java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      system  1.7      system
java      system  1.6      system
$ pkg set-mediator -V 1.6 java
    Packages to update: 3
    Mediators to change: 1
    Create boot environment: No
    Create backup boot environment: No

PHASE                                ITEMS
Removing old actions                  2/2
Updating modified actions             3/3
Updating image state                  Done
Creating fast lookup database         Done
Reading search index                  Done
Updating search index                 3/3
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      local   1.6      system
```

선택한 구현이 더 이상 설치되지 않더라도 이 관리 선택 사항은 패키지 업데이트 사이에 유지됩니다. 선택한 구현이 더 이상 설치되지 않으면 중개 링크의 대상이 존재하지 않습니다. 다음 방법 중 하나를 사용하여 기본 구현을 재설정할 수 있습니다.

- `pkg set-mediator` 명령을 다시 사용하여 `pkg mediator -a`로 표시된 업데이트된 목록에서 다른 구현을 선택합니다.
- `pkg unset-mediator` 명령을 사용하여 시스템이 새 구현을 선택하도록 허용합니다.

```
$ pkg unset-mediator java
```

## 그룹 패키지의 일부 패키지 설치 방지

지정한 패키지가 `group` 종속성의 대상인 경우 해당 패키지 설치를 방지하려면 `pkg avoid` 명령을 사용합니다. 이미지와 호환되는 패키지는 무시 목록에 있더라도 언제나 명시적으로 설

치할 수 있습니다. 무시 목록에 있는 패키지를 설치하면 해당 패키지가 무시 목록에서 제거됩니다. `pkg avoid` 명령을 사용하면 그룹 패키지를 설치할 때 해당 그룹 패키지의 일부인 지정된 패키지 설치를 방지할 수 있습니다.

인수를 지정하지 않으면 `pkg avoid` 명령은 무시된 각 패키지를 해당 패키지에 대해 그룹 종속성을 갖는 패키지와 함께 표시합니다.

지정된 패키지로 `pkg avoid` 명령은 지정한 패턴과 일치하는 패키지 이름을 무시 목록에 놓습니다. 현재 설치되어 있지 않은 패키지만 무시할 수 있습니다. 패키지가 그룹 종속성의 대상일 경우 패키지를 제거하면 무시 목록에 놓입니다.

필요한 경우 `require` 종속성을 충족하기 위해 무시 목록에 있는 패키지가 설치되고, 그러면 무시 목록에서 패키지가 제거됩니다. 해당 `require` 종속성이 제거되면 패키지가 제거되고 무시 목록에 다시 놓입니다.

지정한 패키지를 무시 목록에서 명시적으로 제거하려면 `pkg unavoid` 명령을 사용합니다.

설치된 패키지의 그룹 종속성과 일치하는 무시 목록의 패키지는 `unavoid` 하위 명령을 사용하여 무시 목록에서 제거할 수 없습니다. 그룹 종속성과 일치하는 패키지를 무시 목록에서 제거하려면 패키지를 설치하십시오.

**예 5-5** 무시 목록에서 패키지 추가 및 패키지 제거

다음 명령 출력은 `group/feature/amp` 그룹 패키지가 설치되지 않았음을 보여줍니다. 해당 그룹 패키지에 속한 패키지 중 일부가 명시적으로 설치되었거나 다른 패키지의 `require` 종속성으로 설치되었습니다. 지정된 패키지가 설치되지 않았으므로 `pkg contents` 명령과 함께 `-r` 옵션을 사용합니다.

```
$ pkg list -a group/feature/amp
NAME (PUBLISHER)                VERSION                IFO
group/feature/amp              0.5.11-0.175.2.0.0.33.0  ---
$ pkg list -a `pkg contents -o fmri -Hrt depend -a type=group group/feature/amp`
NAME (PUBLISHER)                VERSION                IFO
database/mysql-51              5.1.37-0.175.2.0.0.34.0  ---
web/php-52                      5.2.17-0.175.2.0.0.34.0  i--
web/php-52/extension/php-apc    3.0.19-0.175.2.0.0.34.0  i--
web/php-52/extension/php-mysql  5.2.17-0.175.2.0.0.34.0  i--
web/server/apache-22           2.2.26-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-dtrace  0.3.1-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-fcgid   2.3.9-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-php5    5.2.17-0.175.1.0.0.18   --r
```

다음 명령은 이 그룹 패키지에 속하되 설치되지 않은 패키지 중 하나를 무시 목록에 놓습니다. 그룹 패키지는 설치되지 않았기 때문에 무시 목록에 나타나지 않습니다.

```
$ pkg avoid apache-fcgid
$ pkg avoid
web/server/apache-22/module/apache-fcgid
```

다음 명령은 그룹 패키지가 설치될 때 무시된 패키지가 설치되지 않음을 보여줍니다. 지정된 패키지가 설치되었으므로 `pkg contents` 명령과 함께 `-r` 옵션이 사용되지 않습니다.

```
$ pkg install group/feature/amp
$ pkg list -a `pkg contents -o fmri -Ht depend -a type=group group/feature/amp`
NAME (PUBLISHER)                VERSION                IFO
database/mysql-51                5.1.37-0.175.2.0.0.34.0  i--
web/php-52                        5.2.17-0.175.2.0.0.34.0  i--
web/php-52/extension/php-apc      3.0.19-0.175.2.0.0.34.0  i--
web/php-52/extension/php-mysql    5.2.17-0.175.2.0.0.34.0  i--
web/server/apache-22             2.2.26-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-dtrace 0.3.1-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-fcgid  2.3.9-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-php5  5.2.17-0.175.1.0.0.18    i-r
```

그룹 패키지가 설치된 후 그룹 패키지는 무시 목록에 표시됩니다.

```
$ pkg avoid
web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

pkg unavoid 명령은 패키지가 설치된 그룹 패키지의 일부인 경우 무시 목록에서 해당 패키지를 제거하지 않습니다. 무시 목록에서 이러한 패키지를 제거하려면 패키지를 설치합니다.

```
$ pkg unavoid apache-fcgid
pkg unavail: The following packages are a target of group dependencies; use install to unavail these:
web/server/apache-22/module/apache-fcgid
$ pkg install apache-fcgid
$ pkg avoid
$
```

패키지가 이미 설치된 경우 해당 패키지를 무시 목록에 넣을 수 없습니다. 패키지를 제거하면 패키지가 무시 목록에 놓입니다.

```
$ pkg avoid apache-fcgid
pkg avoid: The following packages are already installed in this image; use uninstall to avoid these:
web/server/apache-22/module/apache-fcgid
$ pkg uninstall apache-fcgid
$ pkg avoid
web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

그룹 패키지의 일부인 패키지를 제거하면 자동으로 패키지가 무시 목록에 놓입니다.

```
$ pkg uninstall database/mysql-51
$ pkg avoid
database/mysql-51 (group dependency of 'group/feature/amp')
web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

그룹 패키지를 제거하면 무시된 패키지가 무시 목록에 그대로 유지되지만 이 패키지와 그룹 패키지의 연관이 무시 목록에 더 이상 표시되지 않습니다.

```
$ pkg uninstall group/feature/amp
$ pkg avoid
database/mysql-51
web/server/apache-22/module/apache-fcgid
$ pkg unavail database/mysql-51 apache-fcgid
```

```
$ pkg avoid
$
```

## 이미지 및 게시자 등록 정보 구성

이미지 정책을 구현하려면 이미지 등록 정보를 설정하면 됩니다. 이 절에서는 이미지 및 게시자 등록 정보와 이러한 등록 정보를 설정하는 방법에 대해 설명합니다. 이미지 등록 정보에 대한 설명은 [pkg\(1\)](#) 매뉴얼 페이지의 “Image Properties”를 참조하십시오.

### 부트 환경 정책 이미지 등록 정보

이미지는 IPS 패키지를 설치할 수 있으며 다른 IPS 작업을 수행할 수 있는 위치입니다. BE(부트 환경)는 이미지의 부트 가능한 인스턴스입니다. 시스템에서 여러 BE를 유지 관리할 수 있으며 각 BE에 서로 다른 소프트웨어 버전을 설치할 수 있습니다. 시스템을 부트할 때 시스템에 있는 BE 중 하나로 부트하는 옵션이 제공됩니다.

패키지 작업의 결과로 새 BE가 자동으로 생성될 수 있습니다. 새 BE를 명시적으로 만들 수도 있습니다. 새 BE가 생성되는지 여부는 이 절에 설명된 대로 이미지 정책에 따라 달라집니다.

기본적으로 다음 작업 중 하나를 수행하면 새 BE가 자동으로 생성됩니다.

- 일부 드라이버 및 기타 커널 구성 요소 같은 특정한 주요 시스템 패키지를 설치하거나 업데이트합니다. 변형/패치를 변경할 때와 패키지를 설치, 제거, 업데이트할 때 주요 시스템 구성 요소를 업데이트할 수 있습니다.
- `--be-name`, `--require-new-be`, `--backup-be-name`, `--require-backup-be` 등의 옵션을 지정합니다.
- `be-policy` 이미지 정책을 `always-new`로 설정합니다. 이 정책에서는 모든 패키지 작업이 다음 부트 시 활성화되는 새 BE에서 수행됩니다.

새 BE가 생성되면 시스템에서 다음 단계를 수행합니다.

1. 현재 BE의 복제본을 만듭니다.  
복제 BE에는 원본 BE의 주 루트 데이터 세트 아래에 있는 계층 구조 내 모든 항목이 포함됩니다. 공유된 파일 시스템은 루트 데이터 세트 아래에 없으므로 복제되지 않습니다. 대신 새 BE에서는 원래 공유 파일 시스템에 액세스합니다.
2. 복제 BE에서 패키지를 업데이트합니다. 현재 BE의 패키지는 업데이트하지 않습니다.  
현재 BE에 비전역 영역이 구성되어 있는 경우 새 BE에서 이러한 기존 영역이 구성됩니다.
3. `--no-be-activate`를 지정하지 않을 경우 다음에 시스템을 부트할 때 새 BE를 기본 부트 옵션으로 설정합니다. 현재 BE는 대체 부트 옵션으로 유지됩니다.

백업 BE가 만들어지면 시스템에서 다음 단계를 수행합니다.

1. 현재 BE의 복제본을 만듭니다.
2. 현재 BE에서 패키지를 업데이트합니다. 복제 BE의 패키지는 업데이트하지 않습니다.

새 BE가 필요하지만 만드는 데 필요한 공간이 부족하면 필요 없는 기존 BE를 삭제하면 됩니다. BE에 대한 자세한 내용은 “Oracle Solaris 11.2 부트 환경 만들기 및 관리”를 참조하십시오.

다음 이미지 등록 정보를 설정하는 방법에 대한 자세한 내용은 “이미지 등록 정보 설정” [91]을 참조하십시오.

#### be-policy

패키지 작업 중 BE가 생성되는 시점을 지정합니다. 사용할 수 있는 값은 다음과 같습니다.

##### default

기본 BE 생성 정책(create-backup)을 적용합니다.

##### always-new

모든 패키지 작업을 다음 부트 시 활성화되는 새 BE에서 수행하므로 재부트가 필요합니다. 백업 BE는 명시적으로 요청하지 않으면 생성되지 않습니다.

이 정책은 가장 안전하지만 재부트하지 않으면 패키지를 추가할 수 없기 때문에 대부분의 사이트 요구 사항에 비해 다소 엄격합니다.

#### create-backup

재부트가 필요한 패키지 작업의 경우 이 정책은 다음 부트 시 활성화되는 새 BE를 만듭니다. 패키지가 수정되거나 커널에 영향을 줄 수 있는 콘텐츠가 설치되고 작업이 라이브 BE에 영향을 주게 되면 백업 BE가 생성되지만 활성 BE로 설정되지는 않습니다. 백업 BE를 명시적으로 요청할 수도 있습니다.

이 정책은 새로 설치한 소프트웨어가 시스템 불안정을 야기하는 경우에 한해 위험을 가질 수 있습니다. 이와 같은 경우는 거의 발생하지 않지만 가능성은 있습니다.

#### when-required

재부트가 필요한 패키지 작업의 경우 이 정책은 다음 부트 시 활성화되는 새 BE를 만듭니다. 백업 BE는 명시적으로 요청하지 않으면 생성되지 않습니다.

이 정책은 많은 위험을 야기할 수 있습니다. 패키징 작업으로 인해 라이브 BE가 변경되어 추가 변경이 불가능하게 되었는데 최신 풀백 BE가 존재하지 않을 수도 있기 때문입니다.

## 패키지 서명에 필요한 등록 정보

서명된 패키지를 설치하는 경우 이 절에 설명된 이미지 등록 정보 및 게시자 등록 정보를 설정하여 패키지 서명을 확인합니다.

## 서명된 패키지에 필요한 이미지 등록 정보

서명된 패키지를 사용하려면 다음 이미지 등록 정보를 구성합니다.

### signature-policy

이 등록 정보 값은 이미지에 패키지를 설치 및 업데이트하거나 이미지의 패키지를 수정 또는 확인할 때 매니페스트에 대해 수행될 검사를 결정합니다. 패키지에 적용되는 최종 정책은 이미지 정책과 게시자 정책의 조합에 따라 달라집니다. 두 정책을 조합할 경우 수준은 둘 중 더 엄격한 정책을 적용했을 때와 같습니다. 기본적으로 패키지 클라이언트는 인증서가 취소되었는지 여부를 확인하지 않습니다. 이러한 검사를 사용으로 설정하려면 `check-certificate-revocation` 이미지 등록 정보를 `true`로 설정합니다. 이 경우 클라이언트가 외부 웹 사이트에 연결해야 할 수도 있습니다. 사용할 수 있는 값은 다음과 같습니다.

### ignore

모든 매니페스트에 대해 서명을 무시합니다.

### verify

서명이 있는 모든 매니페스트가 유효하게 서명되었는지 확인하지만 설치된 모든 패키지에 대해 서명을 요구하지는 않습니다.

이것이 기본값입니다.

### require-signatures

새로 설치된 모든 패키지가 유효한 서명을 적어도 하나 이상 포함하도록 요구합니다. 또한 `pkg fix` 및 `pkg verify` 명령은 설치된 패키지에 유효한 서명이 없을 경우 경고 메시지를 표시합니다.

### require-names

`require-signatures`와 동일한 요구 사항을 따르지만 `signature-required-names` 이미지 등록 정보에 나열되는 문자열이 서명의 트러스트 체인을 확인하는 데 사용되는 인증서의 공통 이름으로 나타나도록 요구합니다.

### signature-required-names

이 등록 정보의 값은 패키지 서명을 검증하는 동안 인증서의 공통 이름으로 나타나야 할 이름 목록입니다.

## 서명된 패키지에 대한 게시자 등록 정보

특정 게시자의 서명된 패키지를 사용하려면 다음 게시자 등록 정보를 구성합니다.

### signature-policy

이 등록 정보는 지정한 게시자의 패키지에만 적용되는 점을 제외하면 `signature-policy` 이미지 등록 정보와 기능이 같습니다.



signature-required-names

이 등록 정보는 지정한 게시자의 패키지에만 적용되는 점을 제외하면 signature-required-names 이미지 등록 정보와 기능이 같습니다.

## 패키지 서명 등록 정보 구성

패키지 서명 등록 정보를 구성하려면 set-property, add-property-value, remove-property-value, unset-property 하위 명령을 사용합니다.

set-publisher 하위 명령의 --set-property, --add-property-value, --remove-property-value, --unset-property 옵션을 사용하여 서명 정책과 특정 게시자에 필요한 이름을 지정할 수 있습니다.

다음 예에서는 모든 패키지에 서명을 요구하도록 이 이미지를 구성합니다. 또한 이 예에서는 "oracle.com"이라는 문자열이 트러스트 체인의 인증서 중 하나에 대해 공통 이름으로 나타나도록 요구합니다.

```
$ pkg set-property signature-policy require-names oracle.com
```

다음 예에서는 서명된 모든 패키지를 확인하도록 이 이미지를 구성합니다.

```
$ pkg set-property signature-policy verify
```

다음 예에서는 example.com이라는 게시자에서 설치된 모든 패키지에 대해 서명을 요구하도록 이 이미지를 구성합니다.

```
$ pkg set-publisher --set-property signature-policy=require-signatures example.com
```

다음 예에서는 필요한 서명 이름을 추가합니다. 이 예에서는 서명의 트러스트 체인에서 유효한 것으로 나타나야 할 이미지의 공통 이름 목록에 trustedname이라는 문자열을 추가합니다.

```
$ pkg add-property-value signature-required-names trustedname
```

다음 예에서는 필요한 서명 이름을 제거합니다. 이 예에서는 서명의 트러스트 체인에서 유효한 것으로 나타나야 할 이미지의 공통 이름 목록에서 trustedname이라는 문자열을 제거합니다.

```
$ pkg remove-property-value signature-required-names trustedname
```

다음 예에서는 지정한 게시자에 대해 필요한 서명 이름을 추가합니다. 이 예에서는 서명의 트러스트 체인에서 유효한 것으로 나타나야 example.com 게시자의 공통 이름 목록에 trustedname이라는 문자열을 추가합니다.

```
$ pkg set-publisher --add-property-value \
signature-required-names=trustedname example.com
```

## 추가 이미지 등록 정보

### ca-path

SSL 작업을 위해 CA 인증서가 보관된 디렉토리를 가리키는 경로 이름을 지정합니다. 이 디렉토리의 형식은 기본 SSL 구현에 따라 다릅니다. 인증된 CA 인증서를 보관하는 데 다른 위치를 사용하려면 다른 디렉토리를 가리키도록 이 값을 변경합니다. CA 디렉토리에 대한 요구 사항은 SSL\_CTX\_load\_verify\_locations(3openssl)의 CApath 부분을 참조하십시오.

기본값은 /etc/openssl/certs입니다.

### check-certificate-revocation

true로 설정되면 패키지 클라이언트는 인증서 발급 후 인증서가 취소되었는지 확인하기 위해 서명 확인에 사용되는 인증서의 CRL 배포 지점에 연결합니다.

기본값은 false입니다.

### content-update-policy

패키지 시스템이 패키징 작업 중 언제 편집 불가능 파일을 업데이트할지 지정합니다. 사용할 수 있는 값은 다음과 같습니다.

#### default

항상 기본 콘텐츠 업데이트 정책을 적용합니다.

#### always

항상 변경된 편집 불가능 파일을 다운로드하고 업데이트합니다.

#### when-required

패키지 시스템이 업데이트가 필요하다고 결정했을 때만 변경된 편집 불가능 파일을 다운로드하고 업데이트합니다.

기본값은 always입니다.

### flush-content-cache-on-success

true로 설정되면 패키지 클라이언트는 이미지 수정 작업이 성공적으로 완료될 때 해당 콘텐츠 캐시에서 파일을 제거합니다. BE 만들기 작업의 경우 소스와 대상 BE에서 모두 콘텐츠가 제거됩니다.

이 등록 정보는 디스크 공간이 제한된 시스템에서 콘텐츠 캐시 크기를 작게 유지하는 데 사용할 수 있습니다. 이 등록 정보는 작업이 완료되는 데 걸리는 시간을 늘릴 수 있습니다.

기본값은 true입니다.

### mirror-discovery

이 등록 정보는 mDNS 및 DNS-SD를 사용하여 링크-로컬 콘텐츠 미러를 검색하도록 클라이언트에 알려줍니다. 이 등록 정보가 true로 설정되면 클라이언트는 동적으로 검색하

는 미러에서 패키지 콘텐츠를 다운로드합니다. mDNS를 통해 콘텐츠를 알리는 미러를 실행하려면 `pkg.depotd(1M)`을 참조하십시오.

기본값은 `False`입니다.

`send-uuid`

네트워크 작업을 수행할 때 이미지의 UUID(Universally Unique Identifier)를 보냅니다. 사용자가 이 옵션을 사용 안함으로 설정할 수는 있지만 일부 네트워크 저장소의 경우 UUID를 제공하지 않는 클라이언트와의 통신을 거부할 수도 있습니다.

기본값은 `True`입니다.

`trust-anchor-directory`

이 등록 정보의 값은 이미지의 트러스트 앵커가 포함된 디렉토리의 경로 이름입니다. 이 경로는 이미지의 상대 경로입니다.

기본값은 `ignore`입니다.

`use-system-repo`

이 등록 정보는 이미지가 이미지 및 게시자 구성의 소스로, 그리고 제공된 게시자와의 통신을 위한 프록시로 시스템 저장소를 사용해야 하는지 여부를 나타냅니다. 시스템 저장소에 대한 자세한 내용은 `pkg.sysrepo(1M)`를 참조하십시오.

기본값은 `ignore`입니다.

## 이미지 등록 정보 설정

`pkg property` 명령을 사용하여 이미지 등록 정보 설정을 볼 수 있습니다. 이미지 등록 정보를 구성하려면 `set-property`, `add-property-value`, `remove-property-value`, `unset-property` 하위 명령을 사용합니다.

## 이미지 등록 정보 값 표시

`pkg property` 명령을 사용하여 이미지 등록 정보를 볼 수 있습니다.

```
$ pkg property
PROPERTY                                VALUE
be-policy                                default
ca-path                                  /etc/openssl/certs
check-certificate-revocation             False
flush-content-cache-on-success           False
mirror-discovery                          False
preferred-authority                       solaris
publisher-search-order                   ['solaris', 'isvpub']
send-uuid                                  True
signature-policy                          verify
signature-required-names                  []
```

```
trust-anchor-directory      etc/certs/CA
use-system-repo             False
```

pkg set-publisher 명령의 검색 순서 옵션을 사용하여 publisher-search-order 등록 정보를 설정할 수 있습니다. “[게시자 검색 순서 및 고착성 설정](#)” [71]을 참조하십시오.

## 이미지 등록 정보 값 설정

pkg set-property 명령을 사용하여 이미지 등록 정보 값을 설정하거나 등록 정보를 추가 및 설정할 수 있습니다.

다음 예에서는 mirror-discovery 등록 정보의 값을 설정합니다.

```
$ pkg set-property mirror-discovery true
$ pkg property -H mirror-discovery
mirror-discovery True
```

## 이미지 등록 정보 값 재설정

pkg unset-property 명령을 사용하여 지정한 등록 정보의 값을 기본값으로 재설정할 수 있습니다.

```
$ pkg unset-property mirror-discovery
$ pkg property -H mirror-discovery
mirror-discovery False
```

# 이미지 만들기

이미지는 IPS 패키지 및 관련 파일, 디렉토리, 링크 및 종속성을 설치할 수 있으며 다른 IPS 작업을 수행할 수 있는 위치입니다.

pkg image-create 명령을 사용하여 만든 이미지는 부트할 수 없습니다. 부트 가능한 이미지를 만들려면 pkg 명령에 --be-name 또는 --require-new-be 옵션을 함께 사용하거나, beadm 또는 zonecfg 및 zoneadm 명령을 사용합니다. pkg image-create 명령은 패키지 유지 관리나 운영 체제 배포와 같은 작업에 사용됩니다.

pkg image-create 명령에는 피연산자로 이미지가 만들어질 디렉토리가 필요합니다. 만들어진 이미지의 기본 유형은 사용자 이미지입니다. 다음 이미지 유형 중 하나를 지정할 수 있습니다.

전체                          전체 시스템을 제공할 수 있는 전체 이미지. 전체 이미지에서는 모든 종속성이 이미지 자체 내에서 확인되고 IPS가 일관된 방식으로 종속성을 유지 관리합니다. Oracle Solaris OS 설치를 완료하면 루트 파일 시스

템과 해당 콘텐츠가 전체 이미지에 포함됩니다. `-f` 또는 `--full` 옵션을 사용하여 전체 이미지를 지정합니다.

**부분** 부분 이미지는 주어진 `dir` 경로를 둘러싸는 전체 이미지(부모 이미지)에 링크됩니다. 부분 이미지는 자체 내에서 전체 시스템을 제공하지 않습니다. `-P` 또는 `--partial` 옵션을 사용하여 부분 이미지를 지정합니다. 비전역 영역은 부분 이미지입니다. 비전역 영역 컨텍스트에서 이미지를 사용하려면 `-z` 또는 `--zone` 옵션을 지정하여 적절한 변형을 설정합니다. 영역 이미지에서는 IPS가 패키지의 종속성에 정의된 대로, 비전역 영역이 해당 전역 영역과 일치하도록 유지 관리합니다. 영역에 대한 자세한 내용은 [“이미지 및 부트 환경” \[16\]](#)을 참조하십시오.

**사용자** 위치 조정이 가능한 패키지만 포함하는 사용자 이미지 이미지 유형을 지정하지 않을 경우 만들어진 이미지의 기본 유형입니다. `-u` 또는 `--user` 옵션을 사용하여 사용자 이미지를 지정합니다.

`-p` 또는 `--publisher` 옵션을 사용하여 패키지 저장소 URI를 제공합니다. 게시자 이름도 제공한 경우 이미지가 생성될 때 해당 게시자만 추가됩니다. 게시자 이름을 제공하지 않은 경우 지정된 저장소에서 인식하는 모든 게시자가 이미지에 추가됩니다. 초기 생성 작업이 끝나면 이 게시자와 관련된 카탈로그가 검색됩니다.

클라이언트 SSL 인증을 사용하는 게시자의 경우 `-c` 및 `-k` 옵션을 사용하여 클라이언트 키와 클라이언트 인증서를 등록합니다. 이 키와 인증서는 이미지 생성 중 추가된 모든 게시자에 대해 사용됩니다.

`--variant`, `--facet`, `--set-property` 옵션을 사용하여 변형 값, 페이지 값, 이미지 등록 정보 값을 설정합니다.

## 작업 내역 보기

`pkg history` 명령을 사용하여 `pkg` 명령 내역을 볼 수 있습니다. 기본적으로 다음 정보가 표시됩니다.

- 작업의 시작 시간
- 작업의 이름. 예: `install`
- 클라이언트. 예: `pkg`
- 작업의 결과: `Succeeded` 또는 `Failed`

자세한 정보나 더 정밀한 정보를 표시하려면 다음 옵션을 사용합니다.

`-l`

기본 정보 외에 다음 정보를 표시합니다.

- 클라이언트의 버전

- 작업을 수행한 사용자의 이름
- 새 BE가 만들어졌는지 여부
- 작업이 완료된 시간
- 실행된 전체 명령
- 명령을 실행하는 중 발생한 오류
- update와 같은 작업을 위해 변경된 패키지의 전체 FMRI

-n *number*

가장 최근 작업을 지정된 개수만 표시합니다.

```
$ pkg history -n4
START                OPERATION            CLIENT              OUTCOME
2013-08-06T16:32:03  fix                  pkg                 Succeeded
2013-08-06T16:41:47  revert              pkg                 Succeeded
2013-08-06T17:56:22  set-property        pkg                 Succeeded
2013-08-06T17:56:53  unset-property      pkg                 Succeeded
```

-o *column[,column]...*

지정된 심표로 구분된 열 이름 목록을 사용하여 출력을 표시합니다. [pkg\(1\)](#)에서 열 이름 목록을 참조하십시오.

```
$ pkg history -o start,time,operation,outcome -n4
START                TIME                OPERATION            OUTCOME
2013-08-06T16:32:03  0:00:27            fix                  Succeeded
2013-08-06T16:41:47  0:00:43            revert              Succeeded
2013-08-06T17:56:22  0:00:00            set-property        Succeeded
2013-08-06T17:56:53  0:00:00            unset-property      Succeeded
```

-t *time | time-time[,time | time-time]...*

%Y-%m-%dT%H:%M:%S 형식의 심표로 구분된 시간 기록 목록에 대한 레코드를 기록합니다 (strftime(3C) 매뉴얼 페이지 참조). 시간 범위를 지정하려면 시작 시간 기록과 끝 시간 기록 사이에 하이픈(-)을 사용합니다. now 키워드는 현재 시간의 별칭입니다. 지정한 시간 기록에 중복되는 시간 기록이나 겹치는 날짜 범위가 포함되어 있으면 각 중복 내역 이벤트의 한 인스턴스만 표시됩니다.

-N

작업에 대한 릴리스 노트 텍스트를 표시하려면 -N 옵션을 사용합니다. -N 옵션은 -o 옵션과 함께 사용할 수 없습니다. 설치 중인 일부 패키지에 릴리스 노트가 포함된 설치/업데이트 작업에서 -v 옵션을 지정하면 작업 출력에 릴리스 노트가 표시됩니다. 새 BE로 설치되는 작업의 경우 작업 출력에 현재 BE의 /tmp에 릴리스 노트 파일의 경로가 제공됩니다. 새 BE로 부트할 때 릴리스 노트가 /usr/share/doc/release-notes에 있습니다. 아니면 다음 명령에 표시된 대로 -N 옵션을 사용하여 릴리스 노트를 볼 수 있습니다.

```
$ pkg history -N -n 1
```

릴리스 노트를 설치한 작업이 이 BE에서 수행한 마지막 pkg 작업이 아니라면 -n 인수에 더 큰 수를 사용하거나, 아니면 다음 명령에 표시된 대로 -t 옵션을 사용하여 릴리스 노트를 설치한 pkg 작업을 식별합니다.

```
$ pkg history -N -t 2013-07-17T08:31:23
```

명령 기록 정보를 모두 삭제하려면 `pkg purge-history` 명령을 사용합니다.

```
$ pkg purge-history
```







## 패키지 설치 및 업데이트 문제 해결

---

이 부록은 패키지를 설치하거나 업데이트할 때 발생할 수 있는 오류를 처리하는 방법을 보여 줍니다. 또한 이 부록은 성능을 향상하고 저장된 메타 데이터를 최소화하기 위한 팁을 제공합니다.

### 초기 문제 해결 단계

설치할 패키지가 구성된 게시자에서 사용 가능하고 이 이미지에 설치할 수 있는지 여부를 확인합니다. 필요한 정보는 설치된 버전의 `pkg:/entire`와 이 이미지에 구성된 게시자 원본입니다. `pkg:/entire` 패키지를 업데이트해야 할 수 있습니다. 게시자 원본을 수정해야 할 수 있습니다.

거의 모든 패키지 설치 문제에 다음 두 가지 검사를 먼저 수행해야 합니다.

- 어떤 버전의 `pkg:/entire` 통합 패키지가 설치되어 있는지 확인합니다.
- 패키지 게시자 원본을 확인합니다.

필요한 패키지가 구성된 게시자에서 사용 가능한 것을 확인했으면 다음 단계에 따라 설치를 진행하십시오.

- 설치 또는 업데이트를 수행할 때마다 `-nv` 옵션을 사용하여 적용할 변경 사항(예: 설치 또는 업데이트할 패키지의 버전 및 새 BE를 만들지 여부)을 확인하십시오. `-v` 옵션은 이와 같은 특정 설치 또는 업데이트 작업에 적용되는 릴리스 노트를 보여 줍니다.
- 더 자세한 오류 메시지를 받으려면 버전이나 게시자 등 설치할 패키지의 FMRI를 더 많이 지정합니다.

### 설치된 버전의 `pkg:/entire` 확인

`pkg list` 명령을 사용하여 현재 설치된 `pkg:/entire` 통합 패키지의 버전을 확인합니다.

```
$ pkg list entire
NAME (PUBLISHER)                                VERSION                                IFO
```

```
entire                                0.5.11-0.175.2.0.0.34.0    i--
$ pkg list -Hv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z    i--
```

pkg:/entire 통합 패키지는 다른 많은 패키지의 버전을 제약하여 작동 중인 지원 가능한 이미지 상태가 유지되도록 도와줍니다. 통합 패키지의 제약을 받는 패키지에 대한 자세한 내용은 [“통합 패키지” \[12\]](#)를 참조하십시오.

pkg:/entire 통합 패키지의 제약을 받는 패키지를 직접 설치하거나 업데이트할 수 없습니다. pkg:/entire의 제약을 받는 패키지를 설치하거나 업데이트하려면 pkg:/entire 패키지를 업데이트해야 합니다. 자세한 내용은 [“제약 조건을 충족할 수 없음” \[106\]](#)을 참조하십시오. 어떤 경우 [“통합의 제약을 받는 패키지 업데이트” \[106\]](#)에 표시된 대로 제약 조건을 제거할 수 있습니다.

## 구성된 게시자 원본의 내용 확인

pkg publisher 명령을 사용하여 패키지 게시자 원본을 확인합니다.

```
$ pkg publisher
PUBLISHER  TYPE    STATUS P LOCATION
solaris    origin  online F http://pkg.oracle.com/solaris/release/
```

보안 URI의 경우 필요한 키와 인증서가 올바르게 설치되었는지 확인하고, 게시자를 구성할 때 -k 및 -c 옵션을 사용합니다.

사이트에 외부 위치용 프록시가 필요한 경우 pkg set-publisher 명령의 --proxy 옵션을 사용하여 해당 프록시를 설정합니다. 지침은 [“프록시 지정” \[72\]](#)을 참조하십시오.

pkg publisher *publisher* 명령을 사용하여 키, 인증서, 프록시 등 게시자에 대한 자세한 정보를 봅니다.

사용으로 설정된 게시자의 원본 URI에 연결할 수 없으면 필요한 위치에 도달할 수 있더라도 설치/업데이트 작업을 실패합니다. 위치에 도달할 수 없는 문제를 해결할 수 없으면 pkg set-publisher -g를 사용하여 연결할 수 없는 원본을 제거하거나 pkg set-publisher --disable를 사용하여 게시자를 사용 안함으로 설정할 수 있습니다. 이 게시자가 더 이상 필요하지 않으면 pkg unset-publisher를 사용하여 게시자를 제거합니다.

패키지 게시자 원본에 필요한 패키지가 있는지 여부를 확인합니다. 예를 들어, solaris 게시자 원본이 공개 릴리스 저장소로 설정된 경우 지원 저장소에서만 제공되는 버전으로 패키지를 업데이트할 수 없습니다.

## 필요한 설치된 패키지가 사용 가능한지 여부 확인

설치된 패키지를 업데이트하거나 설치된 패키지에 종속된 패키지를 설치하거나 비전역 영역을 설치하려면 게시자 원본으로 설정된 저장소에는 현재 이미지에 설치된 것과 동일한 소프

트웨어가 있어야 합니다. 저장소에 이전 또는 최신 소프트웨어가 포함될 수도 있지만 이미지에 설치된 것과 동일한 소프트웨어가 있어야 합니다.

설치된 패키지를 확인할 때 `pkg list` 명령이 아닌 `pkgrepo list` 명령을 사용합니다. `pkg list` 명령은 구성된 게시자 원본에서 패키지를 사용할 수 없더라도 항상 설치된 패키지를 보여줍니다.

다음 명령은 지정된 저장소가 이 이미지에 적절한 게시자 원본이 아님을 보여줍니다. 설치된 버전의 `pkg:/entire`를 원본에서 사용할 수 없기 때문입니다.

```
$ pkg list entire
NAME (PUBLISHER)          VERSION          IFO
entire                0.5.11-0.175.2.0.0.34.0  i--

$ pkg publisher
PUBLISHER  TYPE    STATUS P LOCATION
solaris    origin online F http://pkg.oracle.com/solaris/release/

$ pkgrepo list -Hs http://pkg.oracle.com/solaris/release entire@0.5.11-0.175.2.0.0.34.0
pkgrepo list: The following pattern(s) did not match any packages:
entire@0.5.11-0.175.2.0.0.34.0
```

필요한 패키지가 나열되지 않으면 `pkgrepo refresh` 명령을 실행한 후 `pkgrepo list` 명령을 재시도하십시오.

다음 명령은 설치된 버전의 `pkg:/entire`를 지정된 저장소에서 사용할 수 있음을 보여줍니다.

```
$ pkgrepo list -Hs /export/IPSpkgrepos/Solaris entire@0.5.11-0.175.2.0.0.34.0
solaris  entire                0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z
```

필요한 패키지를 구성된 게시자에서 사용할 수 없지만 다른 저장소 원본에서 사용할 수 있는 경우, 다음 조치 중 하나를 취하십시오.

- `pkg set-publisher` 명령의 `-g` 옵션을 사용하여 이 원본을 `solaris` 게시자에 추가합니다.
- `pkg set-publisher` 명령의 `-g` 및 `-G` 옵션을 사용하여 `solaris` 게시자의 원본을 변경합니다.
- 필요한 패키지를 다른 게시자가 제공하면 `pkg set-publisher` 명령을 사용하여 해당 게시자를 추가합니다.
- 설치 명령(`install`, `uninstall`, `update`, `change-variant`, `change-facet`)의 `-g` 옵션을 사용하여 검색할 저장소 목록 끝에 저장소를 일시적으로 추가합니다.

## 설치할 패키지가 사용 가능한지 여부 확인

다음 명령을 사용하여 설치할 패키지가 구성된 게시자에서 사용 가능한지 여부를 확인합니다. 패키지를 업데이트하는 경우 현재 설치된 패키지의 버전과 업데이트하려는 버전이 모두 사용 가능해야 합니다.

```
$ pkg list -af package
```

필요한 패키지가 나열되지 않으면 `pkg refresh` 명령을 실행한 후 `pkg list` 명령을 재시도하십시오.

필요한 패키지가 여전히 나열되지 않으면 새 게시자나 새 게시자 원본을 추가합니다.

## 설치할 패키지를 이 이미지에 설치할 수 있는지 여부 확인

-af 옵션을 사용할 때 원하는 패키지 버전이 나열되면 -f 옵션 없이 동일한 명령을 다시 사용합니다.

```
$ pkg list -a package
```

원하는 버전이 여전히 나열되면 이 패키지는 제약을 받지 않으므로 다른 패키지를 설치/업데이트하지 않고 이 패키지를 설치할 수 있어야 합니다.

원하는 버전이 나열되지 않으면 이 버전은 구성된 게시자에서 사용할 수 있지만 이 이미지에 설치할 수 없습니다. 패키지를 설치할 수 없는 이유는 다음과 같습니다.

- 패키지가 변형 또는 페이지 설정의 제약을 받습니다.
- 패키지의 버전이 통합 패키지의 제약을 받습니다. 제약하는 패키지를 업데이트하거나, 어떤 경우 제약 조건을 완화할 수 있습니다. 자세한 내용은 “[제약 조건을 충족할 수 없음](#)” [106]을 참조하십시오.
- 패키지의 버전이 고정 작업의 제약을 받습니다. pkg freeze 명령을 실행합니다. require 종속성을 가진 패키지와 설치할 패키지를 다른 버전에 고정할 수 있으며 양쪽 버전을 동시에 설치할 수 없습니다.

## 설치 재시도

설치/업데이트할 패키지를 지정하고 게시자를 지정하지 않으면 패키지 FMRI 또는 패턴과 일치하는 패키지를 제공하는 게시자 중에서 검색 순서의 첫번째 게시자가 설치 소스로 사용됩니다. 해당 게시자가 이 이미지에 설치할 수 있는 패키지 버전을 제공하지 않으면 다른 사용으로 설정된 게시자가 이 이미지에 설치할 수 있는 패키지 버전을 제공하더라도 설치 작업이 실패합니다. 이 문제를 해결하려면 다음 조치 중 하나를 취하십시오.

- 패키지 FMRI에 게시자를 지정합니다. 예를 들어, 전체 패키지 이름 앞에 pkg://solaris/를 지정합니다.
- pkg set-publisher 명령의 -P 옵션을 사용하여 원하는 패키지 버전을 제공하는 게시자를 검색 순서의 첫번째 게시자로 설정합니다.

설치 또는 업데이트를 수행할 때마다 -nv 옵션을 사용하여 적용할 변경 사항(예: 설치 또는 업데이트할 패키지의 버전 및 새 BE를 만들지 여부)을 확인하십시오. -v 옵션은 이와 같은 특정 설치 또는 업데이트 작업에 적용되는 릴리스 노트를 보여 줍니다.

- -nv 옵션을 사용할 때 오류 메시지가 표시되지 않으면 -n 옵션 없이 명령을 다시 실행하여 실제로 설치 또는 업데이트를 수행합니다. 새 BE에 설치를 수행하는 옵션을 지정할지 또는 새 BE나 백업 BE가 기본적으로 만들어지지 않는 경우 백업 BE를 만드는 옵션을 지정할지 고려하십시오.

- 오류 메시지가 나타나면 다음 조치를 취하십시오.
  - 패키지 FMRI에 원하는 버전을 더 많이 지정하여 진단 및 문제 해결에 도움이 되는 자세한 정보를 얻습니다.
  - 더 많은 `-v` 옵션을 지정합니다(예: `-nvv`).
  - `pkg history` 명령을 사용합니다. `-l` 옵션은 변경된 패키지의 전체 FMRI를 제공합니다. “[작업 내역 보기](#)” [93]를 참조하십시오.

설치/업데이트할 패키지를 여러 개 지정하거나 업데이트 작업에 패키지 지정을 생략할 경우 하나라도 패키지를 이 이미지에 설치할 수 없으면 설치/업데이트 작업을 실패합니다. 한 패키지를 설치할 수 없으면 아무 패키지도 설치되지 않습니다. 자세한 정보를 보려면 명령을 다시 호출합니다. 설치할 수 없는 패키지만 지정하고 해당 패키지의 전체 FMRI를 지정하고 하나 이상의 `-v` 옵션을 제공하면 됩니다.

## 게시자나 저장소에 액세스할 수 없음

이 절에 설명된 오류는 게시자의 URI에 액세스할 수 없는 문제와 관련되어 있습니다.

### 패키지 저장소에 액세스할 수 없음

오류 메시지:

- Couldn't resolve host
- Unable to contact any configured publishers
- Unable to contact valid package repository
- Origin URIs do not appear to point to a valid pkg repository

`pkg publisher` 명령을 사용하여 게시자의 URI를 표시합니다. 다음 예제에서 LOCATION 열이나 URI 행을 살펴보세요.

```
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F http://pkg.oracle.com/solaris/release/
$ pkg publisher solaris
    Publisher: solaris
    Alias:
    Origin URI: http://pkg.oracle.com/solaris/release/
```

더 이상 사용하지 않는 게시자가 나열되면 해당 게시자를 사용 안함으로 설정하거나 제거합니다.

```
$ pkg set-publisher --disable publisher
$ pkg unset-publisher publisher
```

사용 중인 게시자의 경우 URI가 정확한지 확인합니다.

- 각 게시자 원본 위치를 브라우저에서 보거나 해당 위치를 ping하십시오.
- `pkgrepo list` 명령을 사용하여 해당 원본에 패키지를 나열해 보십시오.

원본 위치가 정확하지 않으면 `pkg set-publisher` 명령의 `-G` 및 `-g` 옵션을 함께 사용하여 URI를 변경합니다.

게시자에 여러 개의 원본이 있으면 모든 원본 위치에 액세스할 수 있어야 합니다. 하나라도 원본 위치에 액세스할 수 없으면 `pkg set-publisher` 명령의 `-G` 옵션을 사용하여 해당 원본을 제거합니다.

게시자가 비전역 영역에 구성된 경우 해당 게시자가 전역 영역에 구성되지 않았더라도 전역 영역에서 게시자의 모든 위치에 액세스할 수 있어야 합니다.

## SSL 인증서 문제

오류 메시지: SSL certificate problem, verify that the CA cert is OK

`pkg` 명령에 SSL 인증서 문제에 대한 메시지가 표시되면 다음 조치 중 하나 이상을 취하십시오.

- 필요한 키와 인증서가 설치되어 있는지 확인합니다. 키와 인증서를 얻는 방법에 대한 정보가 보안 저장소에 대한 다른 정보와 함께 포함되어야 합니다.
- `pkg set-publisher` 명령에 `-k` 및 `-c` 옵션을 함께 사용하여 키와 인증서가 설치되는 위치를 지정합니다. `pkg publisher` 명령은 게시자가 키와 인증서를 찾을 위치를 보여줍니다. 키와 인증서가 다른 곳에 설치되어 있으면 `-k` 및 `-c` 옵션에 올바른 값을 지정하여 게시자를 재구성합니다.

다음 예제에서 출력은 키와 인증서 파일 경로를 보여주고 abc 게시자에 대해 구성된 원본의 인증서가 만료되었음을 보여줍니다.

```
$ pkg publisher abc
Publisher: abc
Origin URI: https://pkg.oracle.com/abc/release/
  SSL Key: /var/pkg/ssl/keyfile
  SSL Cert: /var/pkg/ssl/certfile
Certificate '/var/pkg/ssl/certfile' has expired.
Please install a valid certificate.
```

- 시스템에 날짜 및 시간이 정확한지 확인합니다.
- `ca-certificates` SMF 서비스를 새로 고칩니다.

```
$ svcadm refresh svc:/system/ca-certificates:default
```

- 원본 URI에 `oc-mgmt`가 포함되어 있으면 Ops Center 지원 담당자에게 문의하십시오.

---

참고 - 클라이언트 SSL 인증서가 필요한 저장소는 `pkg` 설치 명령에 `-g` 옵션을 사용하여 지정할 수 없습니다.

---

## 위치를 찾을 수 없음

오류 메시지: http protocol error: code: 404 reason: Not Found

“패키지 저장소에 액세스할 수 없음” [101]에 설명된 대로 게시자 URI를 확인합니다. 위치를 성공적으로 보거나 핑할 수 있으면 `pkgrepo list` 명령을 사용하여 저장소의 패키지 중 하나를 표시해 보십시오.

URI가 파일 기반 저장소인 경우 `pkg5srv` 사용자가 파일과 디렉토리를 읽을 수 있는지 확인합니다. `pkgrepo verify` 명령을 사용하여 `pkg5srv` 사용자가 저장소를 읽을 수 있는지 여부를 확인할 수 있습니다.

웹 서버 구성을 확인합니다. 자세한 내용은 “Oracle Solaris 11.2 패키지 저장소 복사 및 만들기”의 5 장, “웹 서버 뒤에서 저장소 서버 실행”을 참조하십시오.

- Apache 웹 서버 인스턴스 뒤에서 패키지 저장소를 실행하는 경우 인코딩된 슬래시를 디코딩하지 않도록 `httpd.conf` 파일에 다음 설정을 포함합니다.

```
AllowEncodedSlashes NoDecode
```

- 저장소 서버 `pkg/proxy_base`를 Apache 서버의 저장소 URL로 설정합니다.

```
$ svccfg -s pkg/server:repo setprop pkg/proxy_base = astring: http://pkg.example.com/myrepo
$ svcadm refresh pkg/server:repo
```

비전역 영역에서 문제가 발생하면 다음 문제 해결 단계를 따르십시오. 비전역 영역은 시스템 저장소라는 특수 패키지 저장소를 사용한다는 것을 기억하십시오. 시스템 저장소에 대한 자세한 내용은 `pkg.sysrepo(1M)` 매뉴얼 페이지를 참조하십시오.

- 비전역 영역이 있는 이미지에 `-g` 옵션을 사용할 수 없습니다. 대신, `pkg set-publisher` 명령을 사용하여 해당 게시자와 원본을 명시적으로 추가합니다.
- 파일 기반 저장소의 파일과 디렉토리를 `pkg5srv` 사용자가 읽을 수 있는지 확인합니다. `pkg5srv` 사용자는 `system-repository` Apache 인스턴스를 실행합니다. 시스템 저장소의 위치를 찾는 방법을 보여주는 예제는 “전역 영역과 비전역 영역의 관계” [50]를 참조하십시오.
- 사이트에 외부 위치에 액세스하는 프록시가 필요한 경우 전역 영역에서 게시자에 대해 프록시가 올바르게 지정되었는지 확인합니다. `pkg set-publisher` 명령의 `--proxy` 옵션을 사용하여 프록시를 지정합니다. 지침은 “프록시 지정” [72]을 참조하십시오. 프록시를 검사하는 한 가지 방법은 `pkg refresh --full` 명령에서 액세스 오류 메시지가 없는지 확인하는 것입니다.
- `svc:/application/pkg/system-repository:default` 서비스가 전역 영역에서 온라인인지 확인합니다.
- `svc:/application/pkg/zones-proxyd:default` 서비스가 전역 영역에서 온라인이고 `svc:/application/pkg/zones-proxy-client:default` 서비스가 비전역 영역에서 온라인인지 확인합니다.
- 전역 영역에서 `/var/log/pkg/sysrepo/*`의 로그 파일을 확인하여 파일을 읽으려고 시도할 때 보고된 권한 오류가 있는지 확인합니다. `/var/log/pkg/sysrepo/access_log`에 보

고된 404 또는 503 오류가 있는지 확인합니다. /var/log/pkg/sysrepo/error\_log에 보고된 오류가 있는지 확인합니다.

- 전역 영역에서 /etc/hosts 파일에 localhost가 127.0.0.1로 설정되어 있는지 확인합니다. /system/volatile/pkg/sysrepo/sysrepo\_httpd.conf 파일에서 Listen이 127.0.0.1:1008로 설정되고 ServerName이 127.0.0.1로 설정되어 있는지 확인합니다.
- 전역 영역에서 /system/volatile/pkg/sysrepo/sysrepo\_httpd.conf 파일에 다음 형식의 Alias 행이 포함되어 있는지 확인합니다.

```
$ grep Alias /system/volatile/pkg/sysrepo/sysrepo_httpd.conf
WSGIScriptAlias /wsgi_p5p /etc/pkg/sysrepo/sysrepo_p5p.py
```

sysrepo\_httpd.conf 파일에 Alias 행이 없으면 sysrepo 서비스를 다시 시작합니다.

```
$ svcadm restart svc:/application/pkg/system-repository:default
```

## 서비스를 사용할 수 없음

오류 메시지: http protocol error: code: 503 reason: Service Unavailable

pkg publisher 명령을 통해 사용하려는 패키지 저장소의 위치를 찾고 해당 시스템에서 SMF 서비스를 검사합니다. 다음 명령을 통해 사용으로 설정되었지만 실행 중인 아닌 패키지 저장소 SMF 서비스 인스턴스가 있는지, 다른 사용으로 설정된 인스턴스가 실행되지 못하게 막는 인스턴스가 있는지 확인합니다.

```
$ svcs -xv pkg/server
svc:/application/pkg/server: default (image packaging repository)
  State: online since July 25, 2013 07:53:50 AM PDT
    See: /var/svc/log/application-pkg-server:default.log
  Impact: None.
```

서비스에 문제가 보고되면 svcs 출력에 나열된 로그 파일을 확인하여 구체적인 문제를 확인합니다.

inst\_root 등록 정보, port 등록 정보 및 기타 등록 정보가 올바르게 설정되었는지 확인합니다.

```
$ svcprop -p pkg pkg/server:default
$ svcprop -p pkg/inst_root -p pkg/port pkg/server:default
/export/IPSpkgrepos/Solaris
80
```

필요한 경우, 다음 예제에 표시된 대로 svccfg 명령을 사용하여 등록 정보 값을 재설정합니다.

```
$ svccfg -s pkg/server:default setprop pkg/port=1008
```

svcadm 명령을 사용하여 필요에 따라 서비스 인스턴스를 지우기, 새로 고침, 다시 시작, 사용으로 설정합니다.



## 사용 가능한 업데이트가 없음

오류 메시지: No updates available for this image

특정 패키지를 업데이트하는 경우 다음 명령을 사용하여 해당 패키지의 어떤 버전이 현재 이 이미지에 설치되어 있는지 확인합니다. 모든 설치된 패키지를 업데이트하는 경우(pkg update에 지정된 패키지가 없거나 패키지 이름에 '\*'가 지정됨) 이 명령에서 *package*에 *pkg:/entire*를 사용합니다.

```
$ pkg list -v package
```

현재 설치된 버전이 없는 경우 `pkg update`가 아닌 `pkg install` 명령을 사용합니다.

*package*의 버전이 현재 설치된 경우 다음 명령을 사용하여 어떤 버전의 *package*를 구성된 게시자에서 사용할 수 있는지 확인합니다.

```
$ pkg list -afv package
```

가장 높은 버전 번호를 가진 패키지가 이미 설치된 경우 최신 버전이 없는 것입니다.

최신 버전이 있으면 최신 버전을 사용할 수 있는 패키지 저장소 위치를 확인하고, `pkg set-publisher` 명령을 사용하여 원본 URI를 재설정하거나 적절한 게시자에 원본 URI를 추가합니다. 필요한 경우, 필요한 키와 인증서를 설치하고 `-k` 및 `-c` 옵션을 사용하여 지정합니다. `pkgrepo list` 명령을 사용하여 현재 설치된 버전의 패키지를 구성된 게시자에서도 사용할 수 있는지 확인합니다.

`pkg update` 명령을 다시 실행합니다. 다음 예제에 표시된 대로 `-nv` 옵션을 지정하고 설치할 패키지의 FMRI에 버전(또는 `latest` 키워드)을 포함하면 됩니다. 패키지 이름에 자세한 정보를 제공하면 대개 오류 출력에도 자세한 정보가 표시됩니다.

```
$ pkg update -nv package@latest
```

## 패키지를 설치할 수 없음

오류 메시지: No matching version of *package* can be installed

다음 명령을 사용하여 어떤 버전의 *package*를 구성된 패키지 게시자에서 사용할 수 있는지 확인합니다.

```
$ pkg list -afv package
```

설치할 패키지의 FMRI를 더 많이 지정합니다. 처음 발견되는 일치 항목은 이 이미지에 설치할 수 없지만, 특정 버전은 설치할 수 있습니다. 더 구체적인 FMRI를 여전히 설치할 수 없는 경우 FMRI를 더 많이 지정하면 패키지를 설치할 수 없는 이유가 자세히 표시됩니다.

설치/업데이트할 패키지의 버전이 고정되지 않았는지 확인합니다. `pkg freeze` 명령을 인수 없이 사용하여 고정된 버전을 가진 모든 패키지 목록을 표시합니다.

다음 명령을 사용하여 어떤 버전의 pkg:/entire 패키지가 설치되었는지 확인합니다.

```
$ pkg list -v entire
```

pkg:/entire 통합 패키지나 다른 통합 패키지의 제약을 받는 패키지를 설치하거나 업데이트할 수 없습니다. 통합 패키지를 업데이트해야 합니다. 자세한 내용은 “[제약 조건을 충족할 수 없음](#)” [106]을 참조하십시오.

## 제약 조건을 충족할 수 없음

오류 메시지: No solution was found to satisfy constraints

이 메시지는 통합 패키지의 제약을 받는 패키지의 버전과 일치하지 않는 패키지 버전을 설치하려고 시도했을 때 나타납니다. 통합 패키지 및 제약 조건에 대한 자세한 내용은 “[통합 패키지](#)” [12]를 참조하십시오.

통합은 패키지 세트를 연동되는 버전으로 제약하여 지원 가능한 이미지 상태가 유지되도록 도와줍니다. 이러한 이유로 통합의 제약을 받는 패키지 하나만 업데이트하면 안됩니다. 대신, 통합 패키지를 업데이트해야 합니다. 그러면 모든 제약된 패키지가 새로운 종합 테스트된 버전 세트로 업데이트됩니다.

패키지가 수정(업데이트) 중일 때 pkg 클라이언트는 관련 패키지와 해당 종속성을 검사합니다. 하나라도 종속 패키지를 설치하거나 업데이트할 수 없으면, 설치하거나 업데이트할 수 없는 패키지에 종속성을 가진 패키지마다 별도의 오류 메시지가 생성됩니다. 대량의 오류 메시지를 처리하는 가장 효과적인 방법은, 먼저 가장 안쪽에 있는 오류 메시지를 검사하는 것입니다.

## 통합의 제약을 받는 패키지 업데이트

오류 메시지:

- No suitable version of installed package *package* found
- All versions matching 'incorporate' dependency *package* are rejected
- This version excluded by specified installation version
- This version is excluded by installed incorporation

통합의 제약을 받는 패키지의 경우 최적의 사용법은 통합 패키지를 업데이트하는 것입니다. 그러면 모든 제약된 패키지가 종합 테스트된 세트로 유지됩니다.

그래도 통합 중에서 패키지 하나만 업데이트하려면 패키지에 version-lock 페이지가 true로 설정되었는지 확인합니다. 패키지에 연관된 version-lock 페이지가 있으면 통합으로부터 해당 패키지의 잠금을 해제할 수 있습니다. version-lock 페이지를 false로 설정하여 제약 조건을 제거하고, 패키지를 다시 설치하거나 업데이트해 보십시오. -nv 옵션을 지정하고 패

지 이름의 FMRI에 원하는 패키지 버전을 지정합니다. [“통합에서 지정한 버전 제약 조건 완화” \[79\]](#)를 참조하십시오.

**예 A-1** Java Runtime Environment 잠금 해제 및 업데이트

다음 예제는 runtime/java/jre-7 패키지를 업데이트하는 방법을 보여줍니다. jre-7 패키지는 consolidation/java/java-incorporation 패키지의 제약을 받고, 차례로 java-incorporation 패키지는 pkg:/entire 통합 패키지의 제약을 받습니다.

다음 명령은 jre-7의 0.175.2.0.0.9.0 버전이 현재 설치되어 있고 구성된 패키지 저장소에서 최신 버전을 사용할 수 있음을 보여줍니다.

```
$ pkg list -af runtime/java/jre-7
NAME (PUBLISHER)                VERSION                IFO
runtime/java/jre-7             1.7.0.21-0.175.2.0.0.13.0 ---
runtime/java/jre-7             1.7.0.17-0.175.2.0.0.9.0  i--
```

-f 옵션을 제거하면 어떤 버전을 업데이트할 수 있는지 보여줍니다. 다음 pkg list 출력은 이 이미지에 설치할 수 있는 최신 버전이 없음을 보여주고 pkg update 명령 출력은 이 상태를 확인합니다. -n 옵션은 어떻게 변경될지 보여주지만 실제로 아무것도 변경하지 않습니다.

```
$ pkg list -a runtime/java/jre-7
NAME (PUBLISHER)                VERSION                IFO
runtime/java/jre-7             1.7.0.17-0.175.2.0.0.9.0  i--
$ pkg update -nv runtime/java/jre-7
No updates available for this image.
```

이 패키지를 업데이트할 수 없는 이유를 자세히 표시하려면 업데이트할 버전을 지정합니다. 다음 예제에 표시된 출력은 설치된 java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0 패키지가 jre-7@1.7.0.21-0.175.2.0.0.13.0 패키지의 설치를 허용하지 않음을 나타냅니다. java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0 패키지는 jre-7@1.7.0.21-0.175.2.0.0.13.0 패키지의 설치를 허용하지만, 설치된 entire@0.5.11,5.11-0.175.2.0.0.12.0 통합 패키지는 jre-7@1.7.0.21-0.175.2.0.0.13.0 패키지의 설치를 허용하지 않습니다.

```
$ pkg update -nv runtime/java/jre-7@1.7.0.21-0.175.2.0.0.13.0
pkg update: No solution was found to satisfy constraints
```

```
maintained incorporations:
[output omitted]
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z
```

```
Plan Creation: dependency error(s) in proposed packages:
[output omitted]
No suitable version of required package pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z found:
  Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z
  Reason: All versions matching 'incorporate' dependency pkg://runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0 are rejected
  Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20
```

```
130304T214022Z
  Reason: This version excluded by specified installation version
  Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:
20130304T214022Z
  Reason: This version excluded by specified installation version
  Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0
.175.2.0.0.13.0:20130429T145534Z
  Reason: This version is excluded by installed incorporation pkg://solaris
/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z

Plan Creation: Errors in installed packages due to proposed changes:
  [output omitted]
  No suitable version of installed package pkg://solaris/consolidation/java/java
-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z found
  Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.1
75.2.0.0.9.0:20130304T213946Z
  Reason: All versions matching 'incorporate' dependency pkg://runtime/java/jr
e-7@1.7.0.17,5.11-0.175.2.0.0.9.0 are rejected
  Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20
130304T214022Z
  Reason: This version excluded by specified installation version
  Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.1
75.2.0.0.13.0:20130429T145534Z
  Reason: This version is excluded by installed incorporation pkg://solaris/e
ntire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z
```

최적의 사용법은 entire 패키지를 업데이트하는 것입니다. entire 패키지를 업데이트하면 java-incorporation 패키지가 업데이트되고, 차례로 jre-7 패키지가 업데이트됩니다. 이 예제에서 Java 패키지를 업데이트해야 하므로 해당 이미지를 업데이트된 버전의 entire 앞으로 이동할 수 없습니다.

Java 통합 패키지에 version-lock 페이지를 설정하여 설치할 수 있는 Java 소프트웨어의 버전을 제약할 수 있습니다. 다른 소프트웨어를 업데이트하지 않고 Java 소프트웨어를 업데이트하려면 Java 통합 패키지의 version-lock 페이지 잠금을 해제하고 Java 통합 패키지를 업데이트합니다. version-lock 페이지에 대한 자세한 내용은 [“통합에서 지정한 버전 제약 조건 완화” \[79\]](#)를 참조하십시오.

다음 명령은 설치된 java-incorporation 패키지의 version-lock 페이지 값을 false로 변경합니다. 이미지의 각 패키지마다 이 페이지를 검사하므로 업데이트할 패키지 수는 이 이미지에 설치된 패키지 수입니다.

```
$ pkg change-facet \  
facet.version-lock.consolidation/java/java-incorporation=false  
  Packages to update: 856  
  Variants/Facets to change: 1  
  Create boot environment: No  
  Create backup boot environment: Yes  
  
Planning linked: 1/1 done  
PHASE  
Removing old actions  
Updating image state  
Creating fast lookup database  
ITEMS  
1/1  
Done  
Done
```

```
Reading search index           Done
Building new search index      856/856
```

다음 명령은 페이지 값이 변경되었음을 보여줍니다.

```
$ pkg facet
FACETS                               VALUE
facet.version-lock.consolidation/java/java-incorporation False
```

다음 명령은 -n 옵션을 지정하므로 이 명령은 어떻게 변경될지 보여주지만 실제로 이미지를 변경하지는 않습니다.

```
$ pkg update -nv java-incorporation
Packages to update:           2
Estimated space available:   80.91 GB
Estimated space to be consumed: 687.28 MB
Create boot environment:     No
Create backup boot environment: Yes
Rebuild boot archive:        No
```

```
Changed packages:
solaris
consolidation/java/java-incorporation
  0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z -> 0.5.11,5.11-0.175.2.0.0.13.0
:20130429T145534Z
runtime/java/jre-7
  1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z -> 1.7.0.21,5.11-0.175.2.0.0.
13.0:20130429T145626Z
```

다음 명령은 실제 업데이트를 수행합니다. 이 명령은 현재 이미지에서 업데이트를 수행합니다. --be-name 옵션을 사용하여 새 부트 환경에서 업데이트를 수행할 수 있습니다.

```
$ pkg update -v java-incorporation
Packages to update:           2
Estimated space available:   80.91 GB
Estimated space to be consumed: 687.28 MB
Create boot environment:     No
Create backup boot environment: Yes
Rebuild boot archive:        No
```

```
Changed packages:
solaris
consolidation/java/java-incorporation
  0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z -> 0.5.11,5.11-0.175.2.0.0.13.0
:20130429T145534Z
runtime/java/jre-7
  1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z -> 1.7.0.21,5.11-0.175.2.0.0.
13.0:20130429T145626Z
```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	2/2	171/171	61.9/61.9	0B/s

PHASE	ITEMS
Removing old actions	7/7
Installing new actions	6/6

```
Updating modified actions          170/170
Updating package state database    Done
Updating package cache            2/2
Updating image state               Done
Creating fast lookup database      Done
Reading search index              Done
Updating search index              2/2
```

다음 명령은 jre-7 패키지가 이 이미지에서 업데이트되었는지 확인합니다. 새 부트 환경에서 업데이트를 수행하는 경우 beadm mount 및 pkg -R을 사용하여 다음 검사를 새 부트 환경에서 수행합니다.

```
$ pkg list jre-7
NAME (PUBLISHER)                VERSION                                IFO
runtime/java/jre-7              1.7.0.21-0.175.2.0.0.13.0          i--
```

## 적절한 종속성을 찾을 수 없을 때 통합 업데이트

오류 메시지: A version for 'incorporate' dependency cannot be found

통합 패키지와 해당 incorporate 종속성에 대한 자세한 내용은 [“통합 패키지” \[12\]](#)를 참조하십시오.

다음 설치 상태가 통합 패키지의 incorporate 종속성에 나타날 때 통합 패키지가 업데이트를 실패할 수 있습니다.

- 종속 패키지가 다른 버전에 고정되어 있습니다.
- 종속 패키지가 이미 상위 버전에 설치되어 있습니다.
- 종속 패키지가 다른 게시자에서 설치되었고, 해당 게시자가 고정되어 있습니다.

예 A-2            종속성을 잠금 해제하고 별도로 업데이트할 때 pkg:/entire 업데이트

다음 예제는 지정된 패키지 이름이 없으므로 모든 설치된 패키지를 업데이트하려고 시도합니다. 이 작업에서 업데이트하려고 시도한 설치된 패키지 중 하나는 pkg:/entire 통합 패키지입니다. 이 예제는 이미 상위 버전에 설치된 pkg:/entire의 incorporate 종속성을 보여줍니다.

```
$ pkg update --be-name s11.2
Creating Plan (Solver setup): /
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to latest available versions.
This may indicate an overly constrained set of packages are installed.
```

```
latest incorporations:
[output omitted]
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
```

The following indicates why the system cannot update to the latest version:

```
No suitable version of required package pkg://solaris/entire@0.5.11,5.11-0.175.2
.0.0.10.0:20130318T181506Z found:
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
Reason: A version for 'incorporate' dependency on pkg:/consolidation/ub_javavm
/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0 cannot be found
```

이 메시지는 시스템에서 업데이트하려고 시도한 pkg:/entire 통합 패키지의 버전이 설치할 수 없는 ub\_javavm-incorporation 패키지의 버전을 가리킨다고 말합니다. 한 패키지를 설치할 수 없기 때문에 아무 패키지도 설치되지 않고 업데이트를 실패합니다.

다음 방법으로 ub\_javavm-incorporation 패키지를 설치할 수 없는 이유를 자세히 제공할 수 있습니다.

- -v 옵션을 사용합니다. 예를 들어, -v 또는 -vv를 사용하여 상세 정보 출력을 받습니다.
- 업데이트할 패키지를 지정합니다. 자세한 입력을 제공할수록 더 자세한 메시지가 표시됩니다. 예를 들어, 패키지 이름 외에 패키지 FMRI에 버전을 포함합니다.

다음 명령은 -v 옵션을 포함하고 위의 메시지에서 복사한 entire@0.5.11,5.11-0.175.2.0.0.10.0 패키지로 업데이트하도록 지정합니다. 또한 이 명령은 --be-name 옵션 대신 -n 옵션을 지정합니다. -n 옵션은 어떻게 변경될지 보여주지만 실제로 이 이미지를 변경하지는 않습니다.

```
$ pkg update -nv entire@0.5.11,5.11-0.175.2.0.0.10.0
Creating Plan (Solver setup): /
pkg update: No matching version of entire can be installed:
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
Reason: All versions matching 'require' dependency pkg:/consolidation/ub_javavm/ub_javavm-incorporation are rejected
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.151.0.1:20101105T053418Z
pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0.0.0.2.0:20111019T144756Z
pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0.10.1.0.0:20120920T143020Z
Reason: Excluded by proposed incorporation 'entire'
Newer version pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z is already installed
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213739Z
Reason: Newer version pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z is already installed
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z
Reason: Excluded by proposed incorporation 'entire'
```

이 메시지는 현재 이 이미지에 설치된 ub\_javavm-incorporation 패키지의 버전이 업데이트 작업으로 설치될 pkg:/entire 통합 패키지에서 지정한 버전보다 최신이라고 말합니다.

다음 명령은 현재 설치된 ub\_javavm-incorporation 패키지의 버전을 보여줍니다.

```
$ pkg list ub_javavm-incorporation
```

```
NAME (PUBLISHER)                                VERSION                                IFO
consolidation/ub_javvm/ub_javvm-incorporation  0.5.11-0.175.2.0.0.13.0            i--
```

다음 명령은 시스템에서 업데이트하려고 시도한 pkg:/entire 통합 패키지에서 지정한 ub\_javvm-incorporation 패키지의 버전을 보여줍니다. 시스템에서 업데이트하려고 시도한 pkg:/entire 통합 패키지의 버전은 위의 출력의 첫번째 “Reject” 메시지에서 복사한 것입니다. -r 옵션은 설치된 이미지가 아니라 구성된 패키지 저장소에서 이 패키지를 찾습니다.

```
$ pkg contents -Hrt depend \  
-a facet.version-lock.consolidation/ub_javvm/ub_javvm-incorporation=true \  
-o fmri entire@0.5.11,5.11-0.175.2.0.0.10.0 \  
consolidation/ub_javvm/ub_javvm-incorporation@0.5.11-0.175.2.0.0.9.0
```

이 문제를 해결하려면 업데이트 작업에서 특별히 지정된 패키지를 제외한 모든 설치된 패키지를 업데이트하도록 지시하면 됩니다. pkg update 명령에 하나 이상의 --reject 옵션을 사용하여 --reject 옵션에 지정된 패키지를 업데이트하지 않으면서 업데이트를 수행할 수 있습니다. --reject 인수에 와일드카드를 사용할 수 있습니다. 다음 명령에서 거부할 패키지는 위의 “Reject” 메시지에서 “Reason: Newer version is already installed” 메시지 앞에 오는 패키지를 복사한 것입니다.

```
$ pkg update -v --be-name s11.2 \  
--reject 'consolidation/ub_javvm/ub_javvm-incorporation@0.5.11,5.11-0.151*' \  
--reject 'consolidation/ub_javvm/ub_javvm-incorporation@0.5.11,5.11-0.175.0*' \  
--reject 'consolidation/ub_javvm/ub_javvm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0*'
```

---

작은 정보 - --reject 인수를 지정할 때 주의하십시오. --reject 인수와 일치하는 패키지는 이미 설치된 것이더라도 제거됩니다.

---

## 설치된 종속성을 허용할 수 없을 때 통합 업데이트

오류 메시지:

- The installed package *package* is not permissible
- Excluded by proposed incorporation

이 오류 메시지가 나타나면 *package*가 아마 다른 핵심 운영 체제 패키지와 동기화를 유지해야 하는 핵심 운영 체제 패키지일 것입니다. pkg facet 명령을 사용하여 facet.version-lock.*package* 페이지의 값을 확인합니다. version-lock 페이지 값이 false이면 pkg change-facet 명령을 사용하여 이 페이지 값을 true로 변경하고 업데이트 작업을 다시 시도합니다.

## 필요한 패키지를 찾을 수 없음

오류 메시지: A version for 'require' dependency cannot be found



다음과 같이 필요한 패키지를 찾을 수 없다는 메시지가 나타나면 구성된 게시자에서 패키지를 사용할 수 있는지 여부를 확인합니다.

```
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to
latest available versions.
This may indicate an overly constrained set of packages are installed.
[output omitted]
No suitable version of required package package1 found:
Reject: package1
Reason: A version for 'require' dependency package2 cannot be found
```

다음 명령을 사용하여 구성된 패키지 게시자에서 *package2*를 사용할 수 있는지 여부를 확인합니다.

```
$ pkg list -afv package2
```

`pkg publisher` 명령을 사용하여 게시자 원본 위치를 확인합니다. 이 패키지의 게시자는 전체 FMRI에서 `pkg://` 뒤에 옵니다. 게시자 원본 위치를 변경해야 할 수 있습니다. 위치가 로컬 패키지 저장소인 경우 저장소를 업데이트해야 할 수 있습니다.

## 필요한 패키지가 거부됨

오류 메시지:

- No solution was found to satisfy constraints
- All versions matching 'require' dependency *package* are rejected

`pkg update` 명령의 다음 출력에서 초기 오류 메시지는 필요한 패키지 `desktop-incorporation`의 적절한 버전을 찾을 수 없다고 말합니다. `desktop-incorporation` 패키지가 적절치 않은 이유는 해당 종속 패키지 중 하나를 찾을 수 없기 때문입니다. 적절한 버전을 찾을 수 없으므로 `desktop-incorporation` 패키지가 거부됩니다. `desktop-incorporation` 패키지가 `pkg:/entire` 통합에 필요하기 때문에 `pkg update`를 실패합니다. 다음 명령은 `desktop-incorporation` 패키지가 `pkg:/entire` 통합에 필요함을 보여줍니다.

```
$ pkg search -Hlo pkg.name require:consolidation/desktop/desktop-incorporation
```

그 다음 오류 메시지는 필요한 패키지 `python-extra-26`의 적절한 버전을 찾을 수 없다고 말합니다. `python-extra-26` 패키지가 부적절한 이유는 `python-extra-26` 패키지에 `desktop-incorporation` 패키지가 필요한데 `desktop-incorporation`의 적절한 버전을 찾을 수 없기 때문입니다.

이 정보에 따라 이 업데이트 실패의 해결 방법은 `desktop-incorporation` 패키지의 필요한 버전을 설치할 방법을 찾는 것입니다.

```
pkg update: No solution was found to satisfy constraints

maintained incorporations:
```

[output omitted]

Plan Creation: dependency error(s) in proposed packages:

[output omitted]

No suitable version of required package pkg://solaris/consolidation/desktop/desktop-incorporation@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z found:

Reject: pkg://solaris/consolidation/desktop/desktop-incorporation@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z

Reason: A version for 'incorporate' dependency on pkg://library/python-2/python-sexy-26@0.1.9-0.175.0.0.0.1.0 cannot be found

No suitable version of required package pkg://solaris/library/python-2/python-extra-26@2.6.4-0.175.1.0.0.15.0:201205014T200156Z found:

Reject: pkg://solaris/library/python-2/python-extra-26@2.6.4-0.175.1.0.0.15.0:201205014T200156Z

Reason: All versions matching 'require' dependency pkg://consolidation/desktop/desktop-incorporation are rejected

Reject: pkg://solaris/consolidation/desktop/desktop-incorporation@0.5.11,5.11-0.175.0.0.0.2.0:20111019T132128Z

[output omitted]

pkg://solaris/consolidation/desktop/desktop-incorporation@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z

다음 명령은 필요한 desktop-incorporation 패키지를 위해 찾을 수 없는 패키지가 필수는 아님을 보여줍니다. python-sexy-26 패키지는 desktop-incorporation 패키지에 포함되어 있지만 필수는 아닙니다.

```
$ pkg search -Hlo pkg.name require:library/python-2/python-sexy-26
$ pkg search -Hlo pkg.name incorporate:library/python-2/python-sexy-26
consolidation/desktop/desktop-incorporation
```

incorporate 종속성은 python-sexy-26 패키지가 설치될 경우 지정된 버전에 설치해야 한다고 말합니다. 그러나 어떤 패키지도 python-sexy-26 패키지를 요구하지 않기 때문에 python-sexy-26 패키지를 설치할 필요가 없습니다. 따라서 이 업데이트 실패의 한 가지 해결 방법은 python-sexy-26 패키지를 제거하는 것입니다. 이 패키지의 다른 버전이 현재 설치되어 있지만 업데이트 시 desktop-incorporation 통합에서 지정한 버전을 찾을 수 없었습니다. 패키지를 제거하면 업데이트 프로세스에서 업데이트된 패키지를 찾을 필요가 없습니다.

python-sexy-26 패키지가 필요해서 제거하고 싶지 않으면 pkg://library/python-2/python-sexy-26@0.1.9-0.175.0.0.0.1.0을 제공하는 패키지 저장소를 찾습니다. pkg set-publisher 명령을 사용하여 해당 저장소를 게시자 원본 위치에 추가하거나, pkgrecv 명령을 사용하여 해당 패키지를 현재 설정된 게시자 원본에 추가합니다.

## 패키지가 예상한 대로 업데이트되지 않음

오류 메시지: pkg update: The installed package *package* is not permissible.

업데이트 작업에 와일드카드를 사용하면 업데이트했어야 하는 패키지가 업데이트되지 않았더라도 오류 메시지가 표시되지 않을 수 있습니다. 와일드카드 없이 패키지 이름을 지정하면 오류 메시지를 볼 수 있습니다.

예를 들어, 다음 작업은 오류 없이 완료되지만 어떤 패키지도 업데이트하지 않습니다. 그러나 구성된 게시자에서 최신 패키지를 사용할 수 있는 것을 확인했습니다.

```
$ pkg update '*'
```

자세한 정보를 얻으려면 와일드카드를 사용하는 대신, 업데이트할 pkg:/entire 버전을 지정합니다.

```
$ pkg list -Hafv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.9.0:20130304T214506Z ---
$ pkg update -nv pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.9.0:20130304T214506Z
Creating Plan (Solver setup): -
pkg update: The installed package compress/zip is not permissible.
  Reject:  pkg://solaris/compress/zip@3.0,5.11-0.175.2.0.0.7.0:20121119T070339Z
  Reason:  Excluded by proposed incorporation 'consolidation/userland/userland-incorporation'
```

이 예제에서 compress/zip 패키지는 userland-incorporation 통합 패키지의 업데이트를 막고, 차례로 pkg:/entire 통합의 업데이트를 막습니다. 다음 명령은 compress/zip 패키지에 대한 자세한 정보를 표시합니다.

```
$ pkg list compress/zip
NAME (PUBLISHER)  VERSION                                IFO
compress/zip     3.0-5.11-0.175.2.0.0.7.0             if-
```

이 출력은 compress/zip 패키지가 고정되었음을 보여줍니다. 패키지가 고정되었으므로 업데이트할 수 없습니다.

```
$ pkg unfreeze compress/zip
compress/zip was unfrozen.
```

compress/zip 패키지의 고정을 해제한 채, 원래 pkg update '\*' 작업으로 이미지에서 사용 가능한 업데이트가 있는 모든 패키지를 업데이트해야 합니다. pkg freeze 및 pkg unfreeze 명령에 대한 자세한 내용은 “[지정한 버전으로 패키지 잠금](#)” [78]을 참조하십시오.

“[동기 링크된 패키지를 설치할 수 없음](#)” [115]에 표시된 pkg sync-linked 오류는 이와 비슷하지만 비전역 영역에서 패키지의 업데이트를 막습니다.

## 동기 링크된 패키지를 설치할 수 없음

오류 메시지: pkg sync-linked: The installed package *package* is not permissible.

sync-linked 패키지는 비전역 영역의 패키지입니다. 부모 이미지와 버전이 일치하지 않는다는 이유로 거부된 패키지를 볼 수 있습니다.

```
Linked progress: -pkg: update failed (linked image exception(s)):
```

```
A 'sync-linked' operation failed for child 'zone:z1' with an unexpected
return value of 1 and generated the following output:
```

```
pkg sync-linked: The installed package package is not permissible.  
Reject: package  
Reason: Parent image has a incompatible newer version: package
```

다음 이유로 이 비호환성 메시지가 나타날 수 있습니다.

- 핵심 운영 체제 패키지는 전역 영역과 비전역 영역에서 버전이 같아야 합니다. 이 패키지를 비전역 영역에서 별도로 업데이트할 수 없습니다. 마찬가지로, 이 패키지를 비전역 영역에서 고정하면 전역 영역과 모든 비전역 영역에서 업데이트를 실패합니다.

다음 명령은 전역 영역과 비전역 영역 간에 동기화를 유지해야 하는 패키지 목록을 표시합니다.

```
$ pkg search -o pkg.name :depend:parent:
```

- 구성된 영역이 있는 BE를 업데이트할 때 ABE(대체 BE)의 게시자 구성이 현재 부트된 BE의 게시자 구성과 다르면, 해당 BE를 마운트해서 `pkg -R` 명령을 사용하여 ABE를 업데이트할 수 없습니다. ABE의 비전역 영역은 현재 활성 BE의 게시자 구성을 사용합니다.

## 비전역 영역을 설치할 수 없음

오류 메시지: The following pattern(s) did not match any allowable packages. Try using a different matching pattern, or refreshing publisher information

핵심 운영 체제 패키지는 전역 영역과 비전역 영역에서 버전이 같아야 합니다. 이 이미지의 solaris 게시자 원본으로 설정된 패키지 저장소에 전역 영역에 설치된 것과 동일한 시스템 패키지 버전이 없는 경우, 비전역 영역을 설치하려고 시도하면 다음 오류가 발생합니다.

```
$ zoneadm -z myzone install
```

The following ZFS file system(s) have been created:

```
    rpool/export/zones/myzone
```

Progress being logged to /var/log/zones/zoneadm.20131104T181301Z.myzone.install

```
    Image: Preparing at /export/zones/myzone/root.
```

```
AI Manifest: /tmp/manifest.xml.9daq.i
```

```
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
```

```
    Zonename: myzone
```

```
Installation: Starting ...
```

```
    Creating IPS image
```

```
Startup linked: 1/1 done
```

```
    Installing packages from:
```

```
        solaris
```

```
            origin: http://pkg.oracle.com/solaris/release/
```

```
    Error occurred during execution of 'generated-transfer-4606-1' checkpoint.
```

```
    Failed Checkpoints:
```

```
    Checkpoint execution error:
```

```
        The following pattern(s) did not match any allowable packages. Try  
        using a different matching pattern, or refreshing publisher information:
```

```
Installation: Failed. See install log at /system/volatile/install.4606/install_log
ERROR: auto-install failed.
```

비전역 영역을 설치하려면 solaris 게시자 원본으로 설정된 저장소에는 전역 영역에 설치된 것과 동일한 시스템 소프트웨어가 있어야 합니다. 저장소에는 더 오래되거나 최신인 소프트웨어가 포함될 수도 있지만, 전역 영역에 설치된 것과 동일한 소프트웨어가 있어야 합니다. 다음 명령은 /export/IPSpkgrepos/Solaris 저장소가 이 전역 영역에 적절한 게시자 원본임을 보여줍니다. 이 저장소에는 전역 영역에 설치된 것과 동일한 pkg:/entire 패키지 버전이 있기 때문입니다.

```
$ pkg list entire
NAME (PUBLISHER)      VERSION                IFO
entire                0.5.11-0.175.2.0.0.26.0  i--
$ pkgrepo list -H -s /export/IPSpkgrepos/Solaris \
entire@0.5.11-0.175.2.0.0.26.0
solaris               entire                0.5.11-0.175.2.0.0.26.0:20131028T190148Z
$ pkg set-publisher -G '*' -M '*' -g /export/IPSpkgrepos/Solaris/ solaris
```

## 이미지를 수정할 수 없음

오류 메시지: pkg: The image cannot be modified as it is currently in use by another package client

다음 예제에서 표시된 대로 오류 메시지에는 이미지가 잠긴 클라이언트의 이름과 pid가 포함되어야 합니다.

```
pkg: The image cannot be modified as it is currently in use by another package
client: pkg on cbusl0406l, pid 26604.
```

현재 패키지 프로세스(이 예제에서 26604 프로세스)가 종료된 후에 pkg 명령을 다시 시도합니다. 프로세스가 빨리 종료되지 않으면 ptree -a를 사용하여 프로세스를 검사하기 시작합니다.

현재 이 이미지에서 작동 중인 패키지 프로세스는 사용 가능한 업데이트가 있는지 여부를 검사하는 업데이트 관리자일 수 있습니다. 업데이트 관리자 cron 작업과 pkg/update 서비스에 대한 자세한 내용은 [“업데이트 관리자 사용” \[124\]](#)을 참조하십시오.

## 파일이 회수됨

다음 정보 메시지에 이어서 회수된 파일의 경로와 파일이 이동된 임시 위치가 표시됩니다.

```
The following unexpected or editable files and directories were
salvaged while executing the requested package operation; they
have been moved to the displayed location in the image:
```

이것은 정보 메시지이므로 필요에 따라 무시하거나 조치를 취할 수 있습니다.

디렉토리는 IPS에서 참조 카운트됩니다. 명시적 또는 암시적으로 디렉토리를 참조하는 이미지에 설치된 마지막 패키지가 더 이상 디렉토리를 참조하지 않을 경우 해당 디렉토리가 제거됩니다. 해당 디렉토리에 패키지화되지 않은 파일 시스템 객체가 포함된 경우 해당 항목은 `$IMAGE_META/lost+found`로 이동됩니다. 패키지화되지 않은 파일 시스템 객체는 IPS 패키지로 제공되지 않는 파일과 디렉토리입니다. `IMAGE_META` 값은 일반적으로 `/var/pkg`입니다. `IMAGE_META` 디렉토리에 대한 자세한 내용은 `pkg(5)` 매뉴얼 페이지의 “Files” 섹션을 참조하십시오.

## 저장된 이미지 메타 데이터 최소화

`/var/pkg` 디렉토리는 이미지의 메타 데이터를 보유하고 있습니다. 이 디렉토리는 상당히 커질 수 있습니다. `/var/pkg` 디렉토리에서 아무것도 수동으로 제거하지 마십시오.

`flush-content-cache-on-success` 이미지 등록 정보 값이 `true`로 설정되어 있는지 확인합니다. `flush-content-cache-on-success` 등록 정보 값은 기본적으로 `true`입니다. `flush-content-cache-on-success` 값이 `true`일 때 `pkg install` 및 `pkg update` 작업이 성공적으로 완료되면 캐시된 파일이 제거됩니다. `flush-content-cache-on-success` 등록 정보가 `false`로 설정된 경우, 다음 예제에 표시된 명령을 사용하여 값을 `true`로 재설정할 수 있습니다.

```
$ pkg property flush-content-cache-on-success
PROPERTY          VALUE
flush-content-cache-on-success False
$ pkg set-property flush-content-cache-on-success true
$ pkg property -H flush-content-cache-on-success
flush-content-cache-on-success True
```

컨텐츠 캐시를 비우면(`flush-content-cache-on-success`를 `true`로 설정) 일부 `pkg` 작업을 완료하는 데 시간이 오래 걸릴 수 있습니다.

비전역 영역마다 다른 캐시를 가질 수 있도록 `/usr/lib/pkg.sysrepo`의 `-c` 옵션을 설정할 수 있습니다. `-s` 옵션으로 이 캐시의 최대 크기를 설정할 수 있습니다. [pkg.sysrepo\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 패키지 설치 성능 향상

다음 단계는 패키지 설치 및 업데이트 성능이 향상되도록 도와줍니다.

- ZFS 저장소 풀 용량이 80% 미만인지 확인합니다.

```
$ zpool list
NAME  SIZE ALLOC FREE CAP DEDUP HEALTH ALROOT
```

rpool 186G 75.2G 111G 40% 1.00x ONLINE -

- 로컬 패키지 저장소를 사용합니다. [“Oracle Solaris 11.2 패키지 저장소 복사 및 만들기”](#)를 참조하십시오.
- http\_proxy가 설정된 경우 프록시의 성능을 확인합니다.





# ◆◆◆ 부록 B

## IPS 그래픽 사용자 인터페이스

---

IPS에는 두 개의 GUI(그래픽 사용자 인터페이스) 도구가 포함되어 있습니다.

- 패키지 관리자는 대부분의 패키지 및 게시자 작업과 몇몇 BE(부트 환경) 작업을 제공합니다. Oracle Solaris OS 및 IPS 기술을 처음 접하는 경우 패키지 관리자를 사용하여 신속하게 패키지를 식별하고 설치할 수 있습니다.
- 업데이트 관리자는 업데이트가 있는 이미지의 모든 패키지를 업데이트합니다.

### 패키지 관리자 사용

패키지 관리자는 명령줄에서 수행할 수 있는 작업 중 일부를 제공합니다.

- 패키지 나열, 검색, 설치, 업데이트, 제거
- 패키지 소스 추가 및 구성
- BE 활성화, 이름 바꾸기 및 제거

다음 방법 중 하나로 패키지 관리자를 시작합니다.

도구 모음	도구 모음에서 패키지 관리자 아이콘을 누릅니다. 패키지 관리자 아이콘은 원모양으로 회전하는 화살표가 있는 상자입니다.
데스크탑 아이콘	데스크탑에서 패키지 관리자 아이콘을 두 번 누릅니다.
메뉴 표시줄	System(시스템)->Administration(관리)->Package Manager(패키지 관리자)를 선택합니다.
명령줄	<code>\$ packagemanager &amp;</code>

전체 패키지 관리자 설명서를 보려면 패키지 관리자 메뉴 표시줄에서 Help(도움말)->Contents(목차)를 선택합니다.

### 패키지 관리자 명령줄 옵션

다음은 `packagemanager(1)` 명령에 지원되는 옵션입니다.

표 B-1 패키지 관리자 명령 옵션

옵션	설명
--image-dir 또는 -R <i>dir</i>	<i>dir</i> 에 루트 지정된 이미지에 대해 작업을 수행합니다. 기본 동작은 현재 이미지에 대해 작업을 수행하는 것입니다.  다음 명령은 /aux0/example_root에 저장된 이미지에 대해 작업을 수행합니다.  <b>\$ packagemanager -R /aux0/example_root</b>
--update-all 또는 -U	사용 가능한 업데이트가 있는 설치된 패키지를 모두 업데이트합니다. 이 옵션을 지정하는 것은 패키지 관리자 GUI에서 Updates(업데이트) 옵션을 선택하는 것과 같습니다. 모든 패키지 업데이트에 대한 자세한 내용은 “ <a href="#">업데이트 관리자 사용</a> ” [124]을 참조하십시오.
--info-install 또는 -i <i>file.p5i</i>	패키지 관리자를 웹 설치 모드로 실행하려면 .p5i 파일을 지정합니다. 지정한 파일은 확장자가 .p5i여야 합니다. 자세한 내용은 “ <a href="#">웹 설치 사용</a> ” [122]을 참조하십시오.
--help 또는 -h	명령 사용 정보를 표시합니다.

## 웹 설치 사용

웹 설치 프로세스에 대한 자세한 내용은 패키지 관리자 도움말을 참조하십시오.

패키지 관리자는 한 번만 누르는 간단한 웹 설치 프로세스를 사용한 패키지 설치를 지원합니다. 웹 설치 프로세스에서는 .p5i 파일을 사용합니다. .p5i 파일에는 게시자 및 이러한 게시자에서 설치할 수 있는 패키지를 추가하는 데 필요한 정보가 들어 있습니다. .p5i 파일의 정보는 웹 설치 프로세스에서 읽고 사용됩니다.

### ▼ 웹 설치 파일을 만드는 방법

시스템에 설치한 패키지를 다른 사용자도 설치할 수 있도록 하려는 경우 웹 설치 프로세스를 사용하여 해당 패키지 파일에 대한 설치 지침을 내보내면 됩니다. 웹 설치 프로세스에서는 설치할 패키지 및 게시자에 대한 설치 지침으로 이루어진 .p5i 파일을 만듭니다.

#### 1. 게시자를 선택합니다.

패키지 관리자 Publisher(게시자) 드롭다운 메뉴에서 패키지를 .p5i 파일에 포함시킬 게시자를 선택합니다.

#### 2. 패키지를 선택합니다.

패키지 관리자 패키지 목록 창에서 설치 지침을 배포할 패키지를 선택합니다.

#### 3. 선택 항목을 내보냅니다.

File(파일)->Export Selections(선택 항목 내보내기)를 선택하여 Export Selections Confirmation(선택 항목 내보내기 확인) 창을 표시합니다.

4. **선택 항목을 확인합니다.**  
OK(확인) 버튼을 눌러 선택 항목을 확인합니다. Export Selections(선택 항목 내보내기) 창이 표시됩니다.
5. **(선택 사항) 파일 이름을 변경합니다.**  
.p5i 파일의 기본 이름은 자동으로 제공됩니다. 이 파일 이름을 변경할 수 있습니다.  
.p5i 확장자는 변경하지 마십시오.
6. **(선택 사항) 파일 위치를 변경합니다.**  
.p5i 파일의 기본 위치는 제공됩니다. 이 위치를 변경할 수 있습니다.
7. **웹 설치 파일을 저장합니다.**  
Save(저장) 버튼을 눌러 파일 이름과 위치를 저장합니다.

## ▼ 웹 설치를 사용하여 게시자를 추가하고 패키지를 설치하는 방법

웹 설치 프로세스를 사용하면 .p5i 파일을 통해 패키지를 설치할 수 있습니다. 이 파일은 데스크탑이나 웹 사이트에 있을 수 있습니다.

1. **패키지 관리자를 웹 설치 모드로 시작합니다.**
  - 데스크탑에서 .p5i 파일을 선택합니다.
  - 명령줄에서 패키지 관리자를 시작하고 .p5i 파일을 지정합니다.  

```
$ packagemanager ./wifile.p5i
```
  - .p5i 파일에 대한 링크가 있는 URL 위치로 이동합니다.
    - .p5i 파일이 이 MIME 형식이 등록된 웹 서버에 있으면 .p5i 파일에 대한 링크만 누르면 됩니다.
    - .p5i 파일이 이 MIME 형식이 등록되지 않은 웹 서버에 있으면 .p5i 파일을 데스크탑에 저장한 후 선택합니다.

Install/Update(설치/업데이트) 창이 표시됩니다. 창 위쪽에 "Package Manager Web Installer/The following will be added to your system.(패키지 관리자 웹 설치 프로그램/다음의 내용들이 여러분의 시스템에 추가됩니다.)"라는 레이블이 표시됩니다. 설치할 게시자 및 패키지가 나열됩니다.
2. **Proceed(진행) 버튼을 눌러 설치를 계속합니다.**
  - 필요한 경우, 게시자를 추가합니다.
  - 필요한 경우,

### 3. 필요한 경우, 게시자를 추가합니다.

지정한 패키지 게시자가 시스템에 아직 구성되어 있지 않으면 Add Publisher(게시자 추가) 창이 표시됩니다. 게시자의 이름과 URI는 이미 입력되어 있습니다.

추가할 게시자가 보안 게시자이면 SSL 키와 인증서를 제공해야 합니다. 시스템에서 SSL 키와 SSL 인증서를 찾아봅니다.

게시자가 성공적으로 추가되면 Adding Publisher Complete(게시자 추가 완료) 대화 상자가 표시됩니다.

### 4. 확인(OK) 버튼을 눌러 설치를 계속합니다.

### 5. 사용 안함으로 설정된 게시자를 사용으로 설정합니다.

.psi 파일에 사용 안함으로 설정된 게시자의 패키지가 포함되어 있으면 웹 설치에서는 Enable Publisher(게시자 사용) 대화 상자가 열립니다. 이 대화 상자를 사용하여 패키지 설치가 가능하도록 게시자를 사용으로 설정할 수 있습니다.

이제 Install/Update(설치/업데이트) 창이 패키지 관리자 Install/Update(설치/업데이트) 옵션을 선택할 때와 같습니다.

모든 패키지가 설치되면 응용 프로그램이 닫힙니다.

## 업데이트 관리자 사용

업데이트 관리자는 설치된 모든 패키지를 설치된 패키지 종속성 및 게시자 구성에 따라 시스템에 적용된 제약 조건에서 허용하는 최신 버전으로 업데이트합니다. 이 기능은 다음 기능과 같습니다.

- 패키지 관리자 GUI에서 Updates(업데이트) 버튼을 선택하거나 Package(패키지)->Updates(업데이트) 메뉴 옵션을 선택합니다.
- `packagemanager` 명령을 사용합니다.

```
$ packagemanager --update-all
```

- `pkg` 명령을 사용합니다.

```
$ pkg update
```

다음 방법 중 하나로 업데이트 관리자를 시작합니다.

상태 표시줄                    업데이트가 있으면 상태 표시줄에 알림이 표시됩니다. 알림에 표시된 위치를 누릅니다. 업데이트 관리자 아이콘은 세 개의 상자가 쌓여 있는 모양입니다.

메뉴 표시줄                    System(시스템)->Administration(관리)->Update Manager(업데이트 관리자)를 선택합니다.

명령줄	\$ <b>pm-updatemanager</b>
자동	업데이트 관리자 패키지 package/pkg/update-manager는 cron 작업 /usr/lib/update-manager/update-refresh.sh를 제공합니다.  30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh  SMF 서비스 svc:/application/pkg/update가 온라인 상태이면 이 cron 작업은 구성된 게시자에서 사용할 수 있는 업데이트된 패키지가 있는지 정기적으로 확인합니다(다음 프로세스의 처음 두 단계). 업데이트된 패키지를 사용할 수 있는 경우 데스크탑 도구 모음에 알림이 표시됩니다. 알림 아이콘을 선택하여 업데이트 관리자 GUI를 엽니다.

Updates(업데이트) 창이 표시되고 업데이트 프로세스가 시작됩니다.

1. 시스템에서 모든 카탈로그를 새로 고칩니다.
2. 시스템에서 설치된 모든 패키지를 평가하여 업데이트가 있는 패키지를 확인합니다.
  - 업데이트가 있는 패키지가 없으면 "No Updates Available(사용 가능한 업데이트가 없음)"이라는 메시지가 표시되고 처리가 중지됩니다.
  - 패키지 업데이트가 있으면 검토를 위해 업데이트 대상 패키지가 나열됩니다. 이때가 Cancel(취소) 버튼을 눌러 업데이트를 중단할 수 있는 마지막 기회입니다.

업데이트를 계속하려면 Proceed(진행) 버튼을 누릅니다.

3. 모든 패키지 업데이트가 다운로드되어 설치됩니다.  
다음 패키지는 업데이트가 있는 경우 가장 먼저 업데이트되고 그런 다음 다른 패키지가 업데이트됩니다.

```
package/pkg
package/pkg/package-manager
package/pkg/update-manager
```

기본적으로 각 패키지는 설치 시 원래 사용되었던 게시자에서 업데이트됩니다. 원래 게시자가 비고정인 경우 이 이미지와 호환되는 패키지의 최신 버전을 다른 게시자에서도 설치할 수 있습니다. 게시자를 고정 또는 비고정으로 설정하려면 패키지 관리자 Manage Publishers(게시자 관리) 창이나 pkg set-publisher 명령을 사용합니다.

업데이트 프로세스 도중 오류가 발생하면 Details(세부 정보) 패널이 확장되고 오류 세부 정보가 표시됩니다. 실패한 단계 옆에 오류 상태 표시기가 나타납니다.

4. 업데이트되는 패키지 및 이미지 정책에 따라 새 BE가 생성될 수 있습니다.  
시스템에서 업데이트 전에 새 BE를 만든 경우 기본 BE 이름을 편집할 수 있습니다.  
새 BE로 부트하려면 다시 시작해야 합니다. 새 BE가 기본 부트 옵션으로 나타납니다. 현재 BE는 대체 부트 옵션으로 사용할 수 있습니다.
  - Restart Now(지금 다시 시작) 버튼을 눌러 시스템을 즉시 다시 시작합니다.
  - 나중에 시스템을 다시 시작하려면 Restart Later(다음에 다시 시작) 버튼을 누릅니다.

## 업데이트 관리자 명령줄 옵션

다음은 `pm-updatemanager(1)` 명령에 지원되는 옵션입니다.

표 B-2 업데이트 관리자 명령 옵션

옵션	설명
<code>--image-dir</code> 또는 <code>-R dir</code>	<code>dir</code> 에 루트 지정된 이미지에 대해 작업을 수행합니다. 기본 동작은 현재 이미지에 대해 작업을 수행하는 것입니다.  다음 명령은 <code>/aux0/example_root</code> 에 있는 이미지를 업데이트합니다.  <code>\$ pm-updatemanager -R /aux0/example_root</code>
<code>--help</code> 또는 <code>-h</code>	명령 사용 정보를 표시합니다.

## 색인

---

### 번호와 기호

- BE, 16
  - pkg 명령 옵션, 35
  - 이름 지정, 35
  - 이미지 정책 등록 정보 설정, 86
  - 패키지 설치 중 만들기, 41
  - 필수 지정, 35
  - 활성화, 35
- BE(부트 환경), 16
- beadm 명령, 42
- entire 통합 패키지, 13
- FMRI, 13
- FMRI(Fault Management Resource Identifier), 13
- Oracle Solaris 10 영역, 16
- p5i 파일, 122
- pkg 명령 미리 보기, 33
- pkg 명령
  - 내역 보기, 93
  - 미리 보기, 33
- pkg avoid 명령, 83
- pkg change-facet 명령, 77, 79
- pkg change-variant 명령, 76
- pkg contents 명령, 24
  - pkg search와 비교, 27
- pkg exact-install 명령, 48
- pkg facet 명령, 77
- pkg fix 명령, 44
  - pkg revert와 비교, 44
- pkg freeze 명령, 78, 115
- pkg history 명령, 93
- pkg image-create 명령, 92
- pkg info 명령, 23
  - pkg list와 비교, 23
- pkg install 명령, 39
  - be-name 옵션, 41
  - reject 옵션, 42, 112
  - pkg update와 비교, 35
- pkg list 명령, 19, 39
  - pkg info와 비교, 23
- pkg mediator 명령, 81
- pkg publisher 명령, 67
- pkg purge-history 명령, 93
- pkg refresh 명령, 19, 38
- pkg revert 명령, 46
  - pkg fix와 비교, 44
- pkg search 명령, 27
  - pkg contents와 비교, 27
- pkg set-mediator 명령, 82
- pkg set-publisher 명령, 69, 73
  - 등록 정보 추가 및 제거, 89
- pkg unavoid 명령, 83
- pkg unfreeze 명령, 78, 115
- pkg uninstall 명령, 48
- pkg unset-mediator 명령, 82
- pkg update 명령
  - pkg install과 비교, 35
  - 이미지 업그레이드, 55
  - 패키지 다운그레이드, 43
  - 패키지 업데이트 또는 업그레이드, 43
- pkg variant 명령, 76
- pkg verify 명령, 44, 44
- pkg:/entire 패키지, 97, 106
- pkg.sysrepo 명령, 51
- pkgrepo 명령, 60
- SMF 서비스, 37
- SMF 프록시 서비스, 51
- solaris 브랜드 영역, 16
- solaris-kz 브랜드 영역, 16
- solaris10 브랜드 영역, 16
- URI(Universal Resource Identifier), 15

## ㄱ

검색 색인, 19, 38  
 게시자, 15, 67  
   검색 순서, 20  
   구성, 69  
   등록 정보, 86  
   서명된 패키지 사용, 88  
   추가 및 제거, 89  
   원본, 67  
   프록시, 72  
 그룹, 13, 26, 31, 83  
 그룹 패키지, 13, 26, 31, 83

## ㄷ

등록 정보, 86  
 BE 만들기 정책, 86  
 서명된 패키지 사용, 87  
 설정, 92  
 추가 및 제거, 89  
 표시, 91

## ㄹ

링크된 이미지, 16, 36, 53  
   pkg uninstall, 48  
   pkg update, 43

## ㅁ

무시 목록, 83  
   제거, 39, 49  
   추가, 42, 48  
 미러 저장소, 15

## ㅂ

버전 문자열 구성 요소, 13  
 버전 제약 조건, 12, 106  
   완화, 79  
 변형, 17, 74  
 부모 이미지, 16  
 비전역 영역, 16, 36, 49  
   업데이트, 53

패키지 게시자, 51, 102  
 패키지 설치, 50

## ㅅ

설치 제약 조건, 12, 106  
 소프트웨어 저장소, 11  
 시스템 저장소, 51

## ㅇ

업데이트 관리자, 121  
 업데이트 제약 조건, 106  
   entire 통합 패키지, 12  
 영역, 36, 49  
   Oracle Solaris 10 영역, 16  
   비전역 영역, 16  
   전역 영역, 16  
   커널 영역, 16  
 영역 프록시 서비스  
   zones-proxy-client, 51  
   zones-proxyd, 51  
 원본 저장소, 15  
 웹 설치, 122  
 응용 프로그램 버전 중개, 81  
 이미지, 16  
   다시 설치, 48  
   등록 정보, 86  
   BE 만들기 정책, 86  
   서명된 패키지 사용, 88  
   추가 및 제거, 89  
   만들기, 92  
   업그레이드, 55  
   정책, 86

## ㅈ

자식 이미지, 16  
 저장소, 11, 15, 40  
   내용 업데이트, 19  
   미러, 15  
   원본, 15  
 전역 영역, 16, 36, 49  
 제약 조건, 12, 106  
   완화, 79



- 중개, 81
- 중개자, 81
  
- ㄱ**
- 커널 영역, 16
  
- ㄷ**
- 통합, 12, 79, 106
- 통합 패키지, 12, 79, 97, 106
  - 사용자 정의, 59
  
- ㅍ**
- 파일
  - 편집 가능한 파일 보존 또는 중첩, 35
- 패키지, 12
  - FMRI, 13, 39
  - parent 종속성, 50
  - pkg:/entire, 97, 106
  - SMF 서비스 액추에이터, 37
  - version-lock. 페이지, 79
  - 검색, 27
  - 게시, 11
  - 게시자, 15, 39
  - 고정됨, 22, 49, 78, 115
  - 그룹, 13, 26, 31, 83
  - 나열, 19
  - 다운그레이드, 43
  - 라이선스, 24, 27, 38
  - 만들기, 11
  - 모두 업데이트, 55
  - 모든 사용 가능, 21
  - 무시 목록, 83
    - 제거, 39, 49
    - 추가, 42, 48
  - 버전이 제약됨, 12, 78, 106, 115
  - 변형, 17, 74
  - 사용자 정의 통합 패키지, 59
  - 상호 배타적 구성 요소, 17, 74
  - 서명 등록 정보, 87
  - 서비스 제공, 29
  - 선택적 구성 요소, 17, 74
  - 설명, 23
  - 설치, 39
  - 설치 가능, 20, 39
  - 설치 소스, 40
  - 설치 중 거부, 42, 112
  - 설치되지 않은 패키지 무시, 43, 48
  - 설치된 파일 수정, 46
  - 설치된 패키지 검증, 44
  - 설치된 패키지 수정, 44
  - 설치됨, 19
  - 식별자, 13
  - 업데이트 또는 업그레이드, 43
  - 업데이트 사용 가능, 21
  - 오래됨, 21
  - 이름, 13
  - 이름이 바뀜, 21
  - 이미지 정리, 48
  - 저장소, 15
  - 전체 FMRI, 23
  - 제거, 48, 48
  - 제약됨, 12, 106
  - 종속성, 31
  - 최신, 20
  - 통합, 12, 106
  - 파일 시스템 콘텐츠, 24
  - 파일 제공, 29
  - 페이지, 17, 74
  - 패키지 게시자 살펴볼 내용 게시자
  - 패키지 관리자, 121
  - 패키지 아카이브, 15, 40
  - 패키지 저장소 살펴볼 내용 저장소
  - 패키지 정보 업데이트, 19, 38
  - 페이지, 17, 74, 79
  - 표면, 12, 106
  - 프록시
    - SMF 서비스 등록 정보, 74
    - 패키지 게시자, 72
    - 환경 변수, 74
  - 프록시 서비스, 51

