

Oracle® Solaris 11.1 패키지 저장소 복사 및 만들기

Copyright © 2011, 2013, 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, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

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

목차

머리말	5
1 이미지 패키징 시스템 패키지 저장소	9
로컬 IPS 저장소	9
저장소 호스트 시스템 준비	10
시스템 요구 사항	10
저장소 관리 권한	10
ZFS 데이터 세트 만들기	11
2 IPS 패키지 저장소 복사	13
인터넷에서 저장소 복사	13
로컬 저장소에 대한 기반구조 만들기	13
저장소 복사	13
파일에서 저장소 복사	14
패키지 저장소 파일 가져오기	15
저장소 파일의 내용을 사용 가능하도록 설정	15
저장소 파일 복사	16
이미지 마운트 해제	16
검색 인덱스 작성 및 저장소 스냅샷 작성	16
3 저장소에 대한 액세스 제공	19
파일 인터페이스를 사용하여 패키지 검색	19
NFS 공유 구성	19
게시자 원본을 파일 저장소 URI로 설정	20
HTTP 인터페이스를 사용하여 패키지 검색	20
저장소 서버 서비스 구성	20
저장소 서비스 시작	21

게시자 원본을 HTTP 저장소 URI로 설정	21
4 로컬 IPS 패키지 저장소 유지 관리	23
로컬 저장소 업데이트	23
저장소 등록 정보 확인 및 설정	24
로컬 저장소 사용자 정의	26
여러 저장소 서버 인스턴스를 사용하여 여러 저장소 제공	27
저장소 서버 Apache 구성	28
저장소 서버에 대한 캐싱 구성	28
웹 프록시 뒤에서 저장소 서버 실행	29
Apache 구성 예	30

머리말

Oracle Solaris 11.1 패키지 저장소 복사 및 만들기에서는 Oracle Solaris 이미지 패키징 시스템(IPS) 기능을 사용하여 소프트웨어 패키지 저장소를 만드는 방법을 설명합니다. IPS 도구를 사용하면 기존 저장소를 쉽게 복사하거나 고유 패키지에 대해 고유한 저장소를 만들고, 저장소에서 패키지를 쉽게 업데이트할 수 있습니다. 저장소 사용자에게 파일 인터페이스 또는 HTTP 인터페이스를 제공할 수 있습니다.

이 설명서의 대상

본 설명서는 소프트웨어를 설치 및 관리하거나 소프트웨어의 설치 및 관리를 지원하는 시스템 관리자를 위해 만들어졌습니다.

이 설명서의 구성

- 1 장, “**이미지 패키징 시스템 패키지 저장소**”에서는 로컬 IPS 패키지 저장소를 제공하는 데 다른 이점을 설명하고 저장소에 대해 ZFS 파일 시스템을 만드는 방법을 보여 줍니다.
- 2 장, “**IPS 패키지 저장소 복사**”에서는 파일에서 저장소를 복사하고 인터넷 위치에서 저장소를 복사하는 방법을 설명합니다.
- 3 장, “**저장소에 대한 액세스 제공**”에서는 클라이언트가 저장소에서 패키지를 보고 설치할 수 있도록 설정하는 방법을 설명합니다.
- 4 장, “**로컬 IPS 패키지 저장소 유지 관리**”에서는 다음과 같은 작업을 수행하는 방법을 설명합니다.
 - 저장소에 업데이트된 패키지 추가
 - 저장소의 등록 정보 값 변경
 - 서로 다른 원본의 패키지를 저장소에 추가
 - 하나의 서버에서 여러 저장소에 대한 액세스 제공
 - 저장소 서버 구성

관련 문서

- **Oracle Solaris 11.1 관리: ZFS 파일 시스템**
- **Oracle Solaris 11.1에서 서비스 및 결함 관리의 1 장, “서비스 관리(개요)”에서는 Oracle Solaris SMF(서비스 관리 기능) 기능에 대해 설명합니다.**
- **Apache HTTP Server Version 2.2 Documentation**
- **Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.1**
- **Oracle Solaris 11.1 소프트웨어 패키지 추가 및 업데이트**
- **Oracle Solaris 11.1 시스템**

Oracle Support에 액세스

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

활자체 규약

다음 표는 이 설명서에서 사용되는 활자체 규약에 대해 설명합니다.

표 P-1 활자체 규약

활자체 또는 기호	설명	예제
AaBbCc123	명령, 파일, 디렉토리 이름 및 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오. 모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오. <code>machine_name% you have mail.</code>
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	<code>machine_name%su</code> Password:
AaBbCc123	위치 표시자: 실제 이름이나 값으로 바뀝니다.	<code>rm filename</code> 명령을 사용하여 파일을 제거합니다.

표 P-1 활자체 규약 (계속)

활자체 또는 기호	설명	예제
AaBbCc123	설명서 제목, 새 용어, 강조 표시할 용어입니다.	<p>사용자 설명서의 6장을 읽으십시오.</p> <p>캐시는 로컬로 저장된 복사본입니다.</p> <p>파일을 저장하면 안 됩니다.</p> <p>주: 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.</p>

명령 예의 셸 프롬프트

다음 표에서는 Oracle Solaris OS에 포함된 셸의 UNIX 시스템 프롬프트 및 슈퍼유저 프롬프트를 보여줍니다. 명령 예에서 셸 프롬프트는 일반 사용자 또는 권한이 있는 사용자가 명령을 실행할지를 나타냅니다.

표 P-2 셸 프롬프트

셸	프롬프트
Bash 셸, Korn 셸 및 Bourne 셸	\$
슈퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	#
C 셸	machine_name%
슈퍼유저용 C 셸	machine_name#

이미지 패키징 시스템 패키지 저장소

Oracle Solaris 11 소프트웨어는 이미지 패키징 시스템(IPS) 패키지로 배포됩니다. IPS 패키지는 IPS 게시자가 채우는 IPS 패키지 저장소에 저장됩니다.

이 설명서는 IPS 패키지 저장소를 만드는 방법을 설명합니다. 이 장에서는 내부 용도로 로컬 IPS 패키지 저장소를 만들어야 하는 경우에 대해 설명합니다.

로컬 IPS 저장소

로컬 IPS 저장소가 필요할 수 있는 이유는 다음과 같습니다.

- **성능 및 보안.** 클라이언트 시스템이 인터넷으로 이동하여 새 소프트웨어 패키지를 검색하거나 기존 패키지를 업데이트하지 못하도록 해야 할 수 있습니다.
- **복제.** 오늘 수행한 설치 작업을 내년에도 동일하게 수행할 수 있도록 해야 할 수 있습니다.
- **사용자 정의 패키지.** Oracle Solaris 패키지와 동일한 저장소에 고유한 IPS 패키지를 포함해야 할 수 있습니다.

IPS는 원본 저장소와 미러 저장소의 두 가지 유형의 저장소를 지원합니다. 위에서 언급한 성능 및 보안 목표를 달성하려면 사용자가 만든 로컬 저장소가 원본 저장소여야 합니다. **원본** 저장소는 하나 이상의 패키지에 대한 내용(파일) 및 모든 메타 데이터(예: 카탈로그, 매니페스트 및 검색 인덱스)를 포함합니다. **미러** 저장소는 패키지 내용(파일)만 포함합니다. 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.

이 설명서에서 설명하는 두 가지 저장소 복사 방법 모두 원본 저장소를 만듭니다. 원본 저장소는 `pkgrecv` 명령을 사용하여 패키지 저장소를 복사할 때 암시적으로 만들어지며, Oracle에서 제공하는 저장소 ISO 파일이 원본 저장소를 제공합니다.

저장소 호스트 시스템 준비

이 단원에서는 IPS 패키지 저장소를 호스트하는 시스템에 대한 요구 사항 및 권장 사항에 대해 설명합니다. 또한 패키지 저장소를 만들고 구성하는 데 필요한 권한에 대해서도 설명합니다.

시스템 요구 사항

IPS 패키지 저장소를 호스트하는 시스템은 x86 기반 시스템 또는 SPARC 기반 시스템일 수 있습니다.

운영 체제 IPS 저장소 서버는 Oracle Solaris 11 11/11 OS 또는 최신 버전의 Oracle Solaris OS를 실행해야 합니다. Oracle Solaris 11 11/11을 실행하는 저장소 서버는 모든 Oracle Solaris 11 업데이트 패키지를 지원합니다.

디스크 공간 Oracle Solaris 11.1 릴리스 저장소의 복사본을 호스트하려면 저장소 서버에 15GB의 여유 공간이 있어야 합니다.

하나의 시스템에 두 개 이상의 IPS 저장소가 호스트되는 경우 각 저장소를 개별적으로 롤백하고 복구할 수 있도록 각 저장소를 개별적인 ZFS 파일 시스템으로 만듭니다.

저장소 관리 권한

패키지 저장소를 만들고 구성하는 데 필요한 권한을 얻으려면 다음 방법 중 하나를 사용합니다.

권한 프로파일 `profiles` 명령을 사용하여 지정 받은 권한 프로파일을 나열합니다.

ZFS File System Management

ZFS File System Management 권한 프로파일이 있으면 `pfexec` 명령을 사용하여 `zfs` 명령을 실행할 수 있습니다.

```
$ pfexec zfs create rpool/export/repoSolaris11
```

Software Installation

Software Installation 권한 프로파일이 있으면 `pfexec` 명령을 사용하여 `pkg` 명령을 실행할 수 있습니다.

```
$ pfexec pkg set-publisher \  
-g http://localhost:80/ solaris
```

Service Management

Service Management 권한 프로파일이 있으면 서비스 명령을 실행할 수 있습니다. 이 경우 `pfexec` 명령은 필요하지 않습니다.

```
$ svcadm enable application/pkg/server
```

- 역할** roles 명령을 사용하여 지정 받은 역할을 나열합니다. 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.

ZFS 데이터 세트 만들기

권장되는 최적 방법은 로컬 패키지 저장소에 대해 개별적인 ZFS 파일 시스템을 만드는 것입니다. 개별적인 ZFS 파일 시스템을 사용하면 다음과 같은 이점을 얻을 수 있습니다.

- 향상된 성능을 얻을 수 있습니다.
- 파일 시스템 특성을 개별적으로 설정할 수 있습니다.
- 직접 스냅샷을 작성하고 지정된 파일 시스템을 복구할 수 있습니다.

zfs list 명령을 사용하여 현재 ZFS 데이터 세트를 볼 수 있습니다.

```
$ zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                75.2G  108G   5.00G   /rpool
rpool/ROOT                           23.0G  108G    31K   legacy
rpool/ROOT/solaris                    44.8G  108G   3.52G   /
rpool/dump                             1.97G  108G   1.97G   -
rpool/export                          43.0G  108G   30.5G   /export
rpool/export/home                     12.6G  108G    32K   /export/home
rpool/export/home/bob                 12.6G  108G   12.6G   /export/home/bob
rpool/swap                             2.09G  108G   1.97G   -
```

루트 풀에서 패키지 저장소에 대해 ZFS 파일 시스템을 만듭니다.

```
$ pfexec zfs create rpool/export/repoSolaris11
$ zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                75.2G  108G   5.00G   /rpool
rpool/export/repoSolaris11          31K   108G    31K   /export/repoSolaris11
...
```

참고 - 저장소를 업데이트할 때 성능 향상을 위해 atime을 off로 설정합니다.

```
$ pfexec zfs set atime=off rpool/export/repoSolaris11
```

atime 등록 정보는 파일을 읽을 때 파일의 액세스 시간을 업데이트할지 여부를 제어합니다. 이 등록 정보를 off로 설정하면 파일을 읽을 때 쓰기 트래픽이 발생하지 않습니다.

IPS 패키지 저장소 복사

이 장에서는 Oracle Solaris 11.1 릴리스 IPS 패키지 저장소의 복사본을 만드는 두 가지 방법을 설명합니다. 복사본을 만들려면 매체 또는 Oracle Solaris 11.1에서 저장소 파일을 사용하거나 인터넷에서 저장소를 검색하면 됩니다.

인터넷에서 저장소 복사

이 절에서는 인터넷 위치로부터 저장소를 복사하여 Oracle Solaris 11.1 릴리스 패키지 저장소의 로컬 복사본을 만드는 방법을 설명합니다.

로컬 저장소에 대한 기반구조 만들기

저장소를 복사할 수 있도록 필요한 pkg(5) 저장소 기반구조를 만듭니다. pkg(5) 및 pkgrepo(1) 매뉴얼 페이지를 참조하십시오.

```
$ pkgrepo create /export/repoSolaris11
```

저장소 복사

pkgrecv 명령을 사용하여 저장소를 복사합니다. 이 작업은 네트워크 성능에 영향을 줄 수 있습니다. 이 작업을 완료하는 데 필요한 시간은 네트워크 대역폭 및 연결 속도에 따라 달라집니다. Oracle Solaris 11.1 릴리스 저장소를 복사하려면 약 7GB의 데이터가 전송됩니다.

참고 - 성능 향상을 위해 대량의 메모리를 사용하는 응용 프로그램을 달고 ZFS 저장소 풀 용량이 80% 미만인지 확인하십시오.

zpool list 명령을 사용하여 풀 용량을 확인합니다.

```
$ zpool list
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool    186G  75.2G  111G  40%  1.00x  ONLINE  -
```

다음 명령은 `-s` 옵션으로 지정된 패키지 저장소에서 모든 패키지의 최신 버전을 `-d` 옵션으로 지정된 저장소로 가져옵니다. 자세한 내용은 `pkgrecv(1)` 매뉴얼 페이지를 참조하십시오.

```
$ pkgrecv -s http://pkg.oracle.com/solaris/release/ -d /export/repoSolaris11 '*'
Processing packages for publisher solaris ...
Retrieving and evaluating 4400 package(s)...
Download Manifests (4400/4400)
PROCESS                ITEMS          GET (MB)        SEND (MB)
developer/build/cmake  446/4400      332.1/4589.7   1000.2/14511.8
...
Completed                4400/4400    4589.7/4589.7  14511.8/14511.8
```

저장소를 복사한 후에는 프로세스가 몇 가지 종료 작업을 수행합니다. "Completed(완료됨)" 라인이 표시되면 프롬프트가 표시될 때까지 몇 분 정도 기다립니다.

이 저장소를 나중에 업데이트할 경우 변경 사항만 복사되고 프로세스 시간이 줄어들 수 있습니다.

`pkgrecv` 작업이 중단된 경우 `-c` 옵션을 사용하여 이미 다운로드한 콘텐츠를 검색하고 콘텐츠를 다운로드를 재개합니다. `cache_dir` 값은 다음 예에 표시된 것처럼 전송이 중단되었을 때 정보 메시지에 제공됩니다.

```
PROCESS                ITEMS          GET (MB)        SEND (MB)
...
pkgrecv: http protocol error: code: 503 reason: Service Unavailable
URL: 'http://pkg.oracle.com/solaris/release/file/file_hash

pkgrecv: Cached files were preserved in the following directory:
/var/tmp/pkgrecv-f0GaIg
Use pkgrecv -c to resume the interrupted download.
$ pkgrecv -c /var/tmp/pkgrecv-f0GaIg \
-s http://pkg.oracle.com/solaris/release/ -d /export/repoSolaris11 '*'
Processing packages for publisher solaris ...
Creating Plan
Retrieving and evaluating 156 package(s)...
PROCESS                ITEMS          GET (MB)        SEND (MB)
desktop/compiz        1/156          0/395.0         0/1100.2
```

파일에서 저장소 복사

이 단원에서는 매체에 있거나 Oracle Solaris 11.1 다운로드 사이트에서 제공되는 저장소 파일로부터 Oracle Solaris 11.1 릴리스 패키지 저장소의 로컬 복사본을 만드는 방법을 보여 줍니다.

패키지 저장소 파일 가져오기

시스템 설치 이미지를 다운로드한 곳과 동일한 위치에서 Oracle Solaris 11.1 IPS 패키지 저장소 .iso 파일을 다운로드하거나 매체 팩킷에서 저장소 DVD를 찾습니다. 저장소는 두 파일로 되어 있으며 총 용량이 약 7GB입니다.

저장소 .iso 파일 외에도 다른 두 개의 파일이 제공됩니다.

- 체크섬 파일. Downloads(다운로드) 페이지 위쪽 근처의 "MD5 checksum(MD5 체크섬)" 링크를 누릅니다. 두 저장소 파일, 그리고 이러한 두 파일을 연결한 것에 대해 체크섬이 제공됩니다. 다음 명령의 출력을 체크섬 파일의 해당 값과 비교하여 다운로드가 성공했는지 확인합니다.

```
$ digest -a md5 iso_file
```

- README 파일. README 파일에는 저장소를 USB 또는 DVD 매체에 복사하는 방법과 같은 추가 정보와 함께 이 절의 정보가 포함됩니다.

마지막 단계에서 만든 파일 시스템에 저장소 파일을 복사합니다. 파일들을 하나의 파일로 연결합니다.

```
$ cat sol-11_1-repo-full.iso-a sol-11_1-repo-full.iso-b > \
sol-11_1-repo-full.iso
$ ls /export/repoSolaris11
sol-11_1-repo-full.iso
```

저장소 파일의 내용을 사용 가능하도록 설정

저장소 .iso 파일의 내용을 사용 가능하도록 설정합니다.

```
$ pfexec mount -F hsfs /export/repoSolaris11/sol-11_1-repo-full.iso /mnt
$ ls /mnt
COPYRIGHT  NOTICES  README  repo
```

mount 명령으로부터 오류 메시지를 수신할 경우, .iso 파일에 대한 전체 절대 경로를 지정했는지 확인합니다.

작업을 확인합니다.

```
$ df -k /mnt
Filesystem                                1K-blocks    Used Available Use% Mounted on
/export/repoSolaris11/sol-11_1-repo-full.iso 6778178 6778178      0 100% /mnt
```

저장소 서버 시스템이 다시 시작될 때마다 .iso 이미지를 다시 마운트해야 합니다. 시스템이 다시 시작될 때마다 .iso를 다시 마운트할 필요가 없도록 하려면 다음 절의 설명에 따라 저장소 파일을 복사합니다.

저장소 파일 복사

각 저장소 액세스 성능을 향상시키고 시스템이 다시 시작될 때마다 .iso 이미지를 다시 마운트할 필요가 없도록 하려면 /mnt/repo/에서 저장소 파일을 ZFS 파일 시스템으로 복사합니다. 이 복사 작업은 rsync 또는 tar을 통해 수행할 수 있습니다.

- rsync 명령을 사용할 경우 repo 디렉토리에 있는 파일과 하위 디렉토리를 복사하려면 /mnt/repo가 아니라 /mnt/repo/(후행 슬래시 문자 포함)로 지정해야 합니다. rsync(1) 매뉴얼 페이지를 참조하십시오.

```
$ rsync -aP /mnt/repo/ /export/repoSolaris11
```

- 다음 예에 표시된 것처럼 tar 명령을 사용하면 마운트된 파일 시스템에서 저장소 ZFS 파일 시스템으로 저장소를 빠르게 이동할 수 있습니다.

```
$ cd /mnt/repo; tar cf - . | (cd /export/repoSolaris11; tar xfp -)
$ cd /export/repoSolaris11
```

작업을 확인합니다.

```
$ ls /export/repoSolaris11
pkg5.repository          README
publisher                sol-11_1-repo-full.iso
$ df -k /export/repoSolaris11
Filesystem                1K-blocks      Used Available Use% Mounted on
rpool/export/repoSolaris11 191987712    13733450   75787939   16% /export/repoSolaris11
```

이미지 마운트 해제

이미지를 마운트 해제합니다.

```
$ pfexec umount /mnt
```

검색 인덱스 작성 및 저장소 스냅샷 작성

저장소 만들기 명령은 기본적으로 검색 인덱스를 작성하지 않습니다. 클라이언트가 로컬 저장소에서 패키지를 검색할 수 있도록 하려면 다음 명령을 사용하여 저장소에 있는 패키지를 카탈로그화하고 검색 인덱스를 업데이트합니다.

```
$ pkgrepo -s /export/repoSolaris11 refresh
Initiating repository refresh.
```

이 패키지 저장소에 대해 개별 ZFS 파일 시스템을 만들었으므로 저장소 파일 시스템에 대한 ZFS 스냅샷을 만들 수 있습니다. 스냅샷을 사용하면 나중에 파일 시스템을 쉽게 롤백하거나 파일 시스템을 신속하게 복사할 수 있습니다. 저장소 내용을 변경할 때마다 저장소 파일 시스템을 스냅샷으로 작성해야 합니다. 다음 명령을 사용하여 새 저장소의 스냅샷을 만듭니다.

```
$ pfexec zfs snapshot rpool/export/repoSolaris11@initial
```

`zfs clone` 명령을 사용하여 저장소 파일 시스템의 복사본을 만들 수 있습니다. 예를 들어, 패키지의 원본 세트를 사용해서 하나의 저장소를 유지 관리하면서 새 버전의 패키지로 다른 복사본을 주기적으로 업데이트할 수 있습니다.

ZFS 복제, 스냅샷 및 롤백에 대한 자세한 내용은 [Oracle Solaris 11.1 관리: ZFS 파일 시스템](#)을 참조하십시오.

저장소에 대한 액세스 제공

이 장에서는 클라이언트가 파일 인터페이스 또는 HTTP 인터페이스를 사용하여 로컬 저장소에 있는 패키지를 검색할 수 있도록 설정하는 방법에 대해 설명합니다. 하나의 저장소에 두 가지 액세스 유형을 모두 설정할 수 있습니다.

파일 인터페이스를 사용하여 패키지 검색

이 절에서는 로컬 네트워크의 디렉토리에서 로컬 저장소 패키지를 제공하는 방법을 설명합니다.

NFS 공유 구성

클라이언트가 NFS를 통해 로컬 저장소에 액세스하도록 설정하려면 공유를 만들고 게시하도록 `sharenfs` 등록 정보를 설정합니다.

```
$ pfexec zfs create -o mountpoint=/export/repoSolaris11 rpool/repoSolaris11
$ pfexec zfs set share=name=s11repo,path=/export/repoSolaris11,prot=nfs rpool/repoSolaris11
name=s11repo,path=/export/repoSolaris11,prot=nfs
$ pfexec zfs set sharenfs=on rpool/repoSolaris11
```

다음 테스트 중 하나를 사용하여 공유가 게시되었는지 확인합니다.

- 공유 파일 시스템 테이블에서 저장소를 검색합니다.

```
$ grep repo /etc/dfs/sharetab
/export/repoSolaris11 s11repo nfs sec=sys,rw
```

- 원격 시스템에서 저장소에 액세스할 수 있는지 확인합니다.

```
$ dfshares solaris
RESOURCE          SERVER ACCESS  TRANSPORT
solaris:/export/repoSolaris11  solaris -      -
```

게시자 원본을 파일 저장소 URI로 설정

클라이언트 시스템이 로컬 파일 저장소에서 패키지를 가져올 수 있도록 하려면 게시자에 대한 원본을 설정합니다. 다음 명령을 사용하여 새 저장소에서 패키지의 게시자 이름을 확인합니다.

```
$ pkgrepo info -s /export/repoSolaris11
PUBLISHER PACKAGES STATUS          UPDATED
solaris   4400    online        2012-07-25T23:40:03.496688Z
```

각 클라이언트에서 다음 명령을 실행하여 solaris 게시자에 대한 원본을 재설정합니다.

```
$ pfxec pkg set-publisher -G '*' -M '*' -g /net/host1/export/repoSolaris11/ solaris
-G '*'    solaris 게시자에 대한 모든 기존 원본을 제거합니다.
-M '*'    solaris 게시자에 대한 모든 기존 미러를 제거합니다.
-g        새로 만든 로컬 저장소의 URI를 solaris 게시자에 대한 새 원본으로
          추가합니다.
```

게시자 구성에 대한 자세한 내용은 [Oracle Solaris 11.1 소프트웨어 패키지 추가 및 업데이트의 “게시자 구성”](#)을 참조하십시오.

HTTP 인터페이스를 사용하여 패키지 검색

이 절에서는 패키지 저장소 서버를 사용하여 로컬 저장소 패키지를 제공하는 방법을 설명합니다.

서로 다른 포트에서 실행 중인 여러 pkg.depotd 데몬을 사용하여 여러 저장소를 제공하는 방법은 [27 페이지 “여러 저장소 서버 인스턴스를 사용하여 여러 저장소 제공”](#)을 참조하십시오. 서로 다른 접두어로 하나의 도메인 이름 아래에 여러 저장소를 실행하는 방법은 [31 페이지 “하나의 도메인 아래의 다중 저장소”](#)를 참조하십시오.

저장소 서버 서비스 구성

클라이언트가 HTTP를 통해 로컬 저장소에 액세스하도록 하려면 application/pkg/server SMF(서비스 관리 기능) 서비스를 사용으로 설정합니다.

```
$ svccfg -s application/pkg/server setprop pkg/inst_root=/export/repoSolaris11
$ svccfg -s application/pkg/server setprop pkg/readonly=true
```

작업을 확인합니다.

```
$ svcprop -p pkg/inst_root application/pkg/server
/export/repoSolaris11
```

pkg.depotd를 사용하여 저장소를 클라이언트에 제공합니다. 기본적으로 pkg.depotd는 포트 80에서 연결을 수신합니다. pkg/port 등록 정보를 재설정하여 포트를 변경할 수 있습니다.

```
$ svccfg -s application/pkg/server setprop pkg/port=port_number
```

application/pkg/server 등록 정보의 전체 목록을 보려면 pkg.depotd(1M) 매뉴얼 페이지를 참조하십시오.

여러 서비스 등록 정보를 설정하려면 다음 명령을 사용하여 모든 등록 정보를 한 번에 편집합니다.

```
$ svccfg -s pkg/server editprop
```

변경한 행의 시작 부분에서 주석 표시자(#)를 제거하는 것에 주의하십시오.

저장소 서비스 시작

pkg.depotd 저장소 서비스를 다시 시작합니다.

```
$ svcadm refresh application/pkg/server
$ svcadm enable application/pkg/server
```

저장소 서버가 작동 중인지 확인하려면 localhost 위치에서 브라우저 창을 엽니다. 기본적으로 pkg.depotd는 포트 80에서 연결을 수신합니다. 포트를 변경한 경우 localhost:port_number 위치에서 브라우저 창을 엽니다.

게시자 원본을 HTTP 저장소 URI로 설정

클라이언트 시스템이 로컬 파일 저장소에서 패키지를 가져올 수 있도록 하려면 게시자에 대한 원본을 설정합니다. 다음 명령을 사용하여 새 저장소에서 패키지의 게시자 이름을 확인합니다.

```
$ pkgrepo info -s /export/repoSolaris11
PUBLISHER PACKAGES STATUS UPDATED
solaris 4400 online 2012-07-25T23:40:03.496688Z
```

각 클라이언트에서 다음 명령을 실행하여 solaris 게시자에 대한 원본을 재설정합니다.

```
$ pfxec pkg set-publisher -G '*' -M '*' -g http://localhost:port_number/ solaris
-G '*' solaris 게시자에 대한 모든 기존 원본을 제거합니다.
-M '*' solaris 게시자에 대한 모든 기존 미러를 제거합니다.
```

- g 새로 만든 로컬 저장소의 URI를 solaris 게시자에 대한 새 원본으로 추가합니다.

게시자 구성에 대한 자세한 내용은 [Oracle Solaris 11.1 소프트웨어 패키지 추가 및 업데이트](#)의 “게시자 구성”을 참조하십시오.

로컬 IPS 패키지 저장소 유지 관리

이 장에서는 IPS 저장소에서 패키지를 업데이트하고, 저장소의 등록 정보를 설정하거나 업데이트하고, 보조 원본의 저장소에 패키지를 추가하는 방법을 설명합니다.

이 장에서는 또한 캐싱 및 로드 밸런싱을 포함하여 저장소 서버의 Apache 구성을 설명합니다.

로컬 저장소 업데이트

더 새로운 패키지를 로컬 저장소에 전송하기 전에 사용자가 복사하려는 패키지가 구축된 서버와 동일하거나 더 새로운 버전의 Oracle Solaris 11 OS가 저장소 서버에서 실행 중인지 확인합니다. 예를 들어, 서버가 Oracle Solaris 11 OS를 실행 중이고 저장소를 Oracle Solaris 11.1 저장소로 업데이트하려는 경우, 저장소를 업데이트하기 전에 서버를 Oracle Solaris 11.1로 업데이트합니다. 시스템 업데이트에 대한 자세한 내용은 [Oracle Solaris 11.1 소프트웨어 패키지 추가 및 업데이트](#)의 “이미지 업데이트”를 참조하십시오.

로컬 IPS 패키지 저장소를 만들기 위해 `pkgrecv` 명령 또는 `.iso` 파일을 사용했는지 여부에 따라 `pkgrecv(1)` 명령을 사용하여 저장소를 업데이트합니다. 변경된 패키지만 업데이트됩니다. [13 페이지 “저장소 복사”](#)의 성능 팁을 참조하십시오.

저장소 내용을 변경하기 전에 저장소 파일 시스템을 스냅샷으로 작성해야 합니다. SRU(Support Repository Update) 번호를 포함하는 등, 스냅샷 이름을 의미 있는 이름으로 지정합니다. ZFS 스냅샷 및 롤백에 대한 자세한 내용은 [Oracle Solaris 11.1 관리: ZFS 파일 시스템](#)을 참조하십시오.

```
$ pkgrecv -s http://pkg.oracle.com/solaris/support/ -d /export/repoSolaris11 \
--key /var/pkg/ssl/Oracle_Solaris_11_Support.key.pem \
--cert /var/pkg/ssl/Oracle_Solaris_11_Support.certificate.pem '*'
```

이 업데이트를 정기적으로 수행하려는 경우 `PKG_SRC` 및 `PKG_DEST` 환경 변수를 사용해야 할 수 있습니다.

```
$ export PKG_SRC=http://pkg.oracle.com/solaris/support/
$ export PKG_DEST=/export/repoSolaris11
$ pkgrecv '*'
```

저장소를 업데이트한 후에는 다음 명령을 사용하여 저장소에서 발견된 모든 새로운 패키지를 카탈로그화하고 모든 검색 인덱스를 업데이트합니다.

```
$ pkgrepo rebuild -s /export/repoSolaris11
```

HTTP 인터페이스를 통해 저장소를 제공하는 경우, SMF 서비스를 다시 시작합니다.

```
$ svcadm restart application/pkg/server:default
```

저장소를 두 개 이상 만든 경우, 서비스를 다시 시작할 때 적합한 서비스 인스턴스를 지정합니다.

저장소 등록 정보 확인 및 설정

이 절에서는 IPS 저장소에 대한 정보를 표시하고 저장소 및 게시자 등록 정보를 설정하는 방법을 설명합니다. [pkgrepo\(1\)](#) 매뉴얼 페이지를 참조하십시오.

다음 명령은 로컬 저장소에 의해 알려진 패키지 게시자의 목록을 표시합니다. STATUS(상태) 열은 게시자의 패키지 데이터를 현재 처리 중인지 여부를 알려줄 수 있습니다.

```
$ pkgrepo info -s /export/repoSolaris11
PUBLISHER PACKAGES STATUS          UPDATED
solaris   4400      online      2012-07-25T23:40:03.496688Z
```

다음 명령은 로컬 저장소에 대한 등록 정보를 표시합니다.

```
$ pkgrepo get -s /export/repoSolaris11
SECTION    PROPERTY    VALUE
publisher  prefix     solaris
repository description This\ repository\ serves\ a\ copy\ of\ the\ Oracle\ Solaris\ 11.1\
Package\ Repository.
repository name      Oracle\ Solaris\ 11.1\ Package\ Repository
repository version   4
```

게시자 접두어 값은 다음과 같은 경우에 `solaris`를 사용하도록 지정합니다.

- 두 개 이상의 게시자의 패키지가 제공되었고 `pkg` 명령에서 패키지 이름에 게시자가 지정되지 않은 경우
- 패키지가 저장소에 게시되었고 게시자가 지정되지 않은 경우

기본적으로 버전 4 저장소가 만들어집니다. 버전 4 저장소는 다중 게시자에 대한 패키지 저장을 지원합니다.

`set` 하위 명령을 사용하여 새 등록 정보 값을 지정합니다.

```

$ pkgrepo set -s /export/repoSolaris11 \
repository/description="Local copy of the Oracle Solaris 11.1 repository" \
repository/name="Oracle Solaris 11.1 Package Repository"
$ pkgrepo get -s /export/repoSolaris11
SECTION    PROPERTY    VALUE
publisher  prefix      solaris
repository description Local\ copy\ of\ the\ Oracle\ Solaris\ 11.1\ repository
repository name Oracle\ Solaris\ 11.1\ Package\ Repository
repository version 4

```

다음 명령은 로컬 저장소에서 solaris 게시자에 대한 등록 정보를 표시합니다. 괄호는 특정 값이 값 목록일 수 있음을 나타냅니다. 여러 로컬 저장소를 만들 경우, 구성 중에 저장소 이름, 경로, 포트 번호 및 Apache 프록시 기준을 변경합니다. 31 페이지 “단순 접두어가 지정된 프록시 구성”에서는 pkg/proxy_base 설정을 보여줍니다.

```

$ pkgrepo get -p solaris -s /export/repoSolaris11
PUBLISHER SECTION    PROPERTY    VALUE
solaris  publisher alias
solaris  publisher prefix      solaris
solaris  repository collection-type core
solaris  repository description Local\ copy\ of\ the\ Oracle\ Solaris\ 11.1\ repository
solaris  repository legal-uris ()
solaris  repository mirrors ()
solaris  repository name Oracle\ Solaris\ 11.1\ Package\ Repository
solaris  repository origins ()
solaris  repository refresh-seconds
solaris  repository registration-uri ""
solaris  repository related-uris ()

```

collection-type core 모음 유형은 저장소에서 패키지로 선언된 모든 종속성이 저장소에 포함됨을 나타냅니다.

legal-uris legal-uris는 저장소에 대한 법적 정보를 제공하는 문서의 위치 목록입니다.

origins origins는 이 저장소의 패키지 메타 데이터 및 콘텐츠에 대한 전체 복사본을 포함하는 저장소의 위치 목록입니다.

related-uris related-uris는 사용자가 관심있어 할 수 있는 패키지가 포함된 저장소의 위치 목록입니다.

다른 게시자 및 저장소 등록 정보에 대한 설명은 [pkgrepo\(1\)](#) 매뉴얼 페이지를 참조하십시오.

다음 명령은 pkg.oracle.com 저장소에서 지정된 *section/property*에 대한 정보를 표시합니다.

```

$ pkgrepo get -p solaris -s http://pkg.oracle.com/solaris/release \
repository/name repository/description
PUBLISHER SECTION    PROPERTY    VALUE
solaris  repository description Local\ copy\ of\ the\ Oracle\ Solaris\ 11.1\ repository
solaris  repository name Oracle\ Solaris\ 11.1\ Package\ Repository

```

로컬 저장소의 solaris 게시자에 대해 저장소 설명 및 저장소 이름 등록 정보 값이 설정되지 않았습니다. 게시자 등록 정보의 값을 제공하려면 위에 표시된 것처럼 set 하위 명령을 사용합니다. 이때 게시자 이름도 지정합니다. 게시자 repository/name 값은 브라우저 인터페이스에서 페이지 위쪽 근처에 그리고 페이지 제목으로 표시됩니다. 게시자 repository/description 값은 브라우저 인터페이스에서 이름 바로 아래의 About(정보) 절에 표시됩니다.

```
$ pkgrepo set -p solaris -s /export/repoSolaris11 \
repository/description="Local copy of the Oracle Solaris 11.1 repository" \
repository/name="Oracle Solaris 11.1 Package Repository"
$ pkgrepo get -p solaris -s /export/repoSolaris11
PUBLISHER SECTION PROPERTY VALUE
solaris publisher alias
solaris publisher prefix solaris
solaris repository collection-type core
solaris repository description Local\ copy\ of\ the\ Oracle\ Solaris\ 11.1\ repository
solaris repository legal-uris ()
solaris repository mirrors ()
solaris repository name Oracle\ Solaris\ 11.1\ Package\ Repository
solaris repository origins ()
solaris repository refresh-seconds
solaris repository registration-uri ""
solaris repository related-uris ()
```

로컬 저장소 사용자 정의

원본 저장소의 하위 세트인 저장소를 만들 수 있습니다. 다음 명령은 group/feature/amp 패키지의 모든 버전 및 해당 버전의 모든 종속성을 amprepo 저장소에 복사합니다. amprepo 저장소는 이전에 pkgrepo create 명령을 사용하여 만들어졌습니다.

```
$ pkgrecv -s http://pkg.oracle.com/solaris/release/ -d /export/amprepo \
-m all-versions -r group/feature/amp
```

다른 게시자의 패키지를 자신의 저장소에 추가할 수 있습니다. 다음 pkgrecv 명령은 ISVproducts.p5p 패키지 아카이브의 모든 패키지를 로컬 저장소에 추가합니다. pkg list 출력에서는 게시자가 이 이미지에서 검색 순서가 가장 높은 게시자가 아니기 때문에 표시됩니다.

```
$ pkg list -g /tmp/ISVproducts.p5p
NAME (PUBLISHER) VERSION IFO
isvtool (isvpub) 1.0 ---
$ pkgrecv -s /tmp/ISVproducts.p5p -d /export/repoSolaris11 '*'
Processing packages for publisher isvpub ...
Retrieving and evaluating 1 package(s)...
PROCESS ITEMS GET (MB) SEND (MB)
Completed 1/1 0.0/0.0 0.0/0
$ pkg list -g /export/repoSolaris11 isvtool
NAME (PUBLISHER) VERSION IFO
isvtool (isvpub) 1.0 ---
```

패키지 아카이브는 게시자 정보와 해당 게시자가 제공한 하나 이상의 패키지가 포함된 파일입니다. **Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.1**의 “Publish as a Package Archive”를 참조하십시오.

여러 저장소 서버 인스턴스를 사용하여 여러 저장소 제공

이 절에서는 20 페이지 “HTTP 인터페이스를 사용하여 패키지 검색”에 제공된 정보를 확장하여 동일한 저장소 서버의 서로 다른 포트에서 실행 중인 여러 `pkg.depotd` 데몬을 사용하여 여러 저장소를 제공하도록 지원하는 방법을 보여 줍니다.

이 예에서는 `repoSolaris11` 저장소 외에도 `dev_repo` 저장소가 존재합니다. `repoSolaris11` 저장소는 포트 80을 사용하여 `http://localhost/`에서 액세스할 수 있습니다.

`dev_repo` 저장소에 게시자 접두어가 설정되어 있는지 확인합니다.

```
$ pkgrepo set -s /export/dev_repo publisher/prefix=dev
```

`pkg/server` 서비스의 새 인스턴스를 추가합니다.

```
$ svccfg -s pkg/server add dev
$ svccfg -s pkg/server:dev setprop pkg/port=81
$ svccfg -s pkg/server:dev setprop pkg/inst_root=/export/dev_repo
```

새 인스턴스가 추가되었는지 확인합니다.

```
$ svccfg -s pkg/server list
:properties
default
dev
```

새 서비스를 시작합니다.

```
$ svcadm refresh application/pkg/server:dev
$ svcadm enable application/pkg/server:dev
```

`http://localhost:81/`에서 저장소를 찾아봅니다.

서로 다른 접두어로 하나의 도메인 이름 아래에 여러 저장소를 실행하는 방법은 31 페이지 “하나의 도메인 아래의 다중 저장소”를 참조하십시오.

저장소 서버 Apache 구성

이 절에서는 Apache 웹 서버 인스턴스 뒤에서 저장소 서버를 실행하여 얻을 수 있는 다음과 같은 이점에 대해 설명합니다.

- 콘텐츠 캐싱 및 로드 밸런싱을 통해 성능을 향상시킬 수 있습니다.
- 하나의 도메인 이름으로 여러 저장소를 호스트할 수 있습니다.

저장소 서버에 대한 캐싱 구성

캐싱 프록시 뒤에서 저장소 서버를 설정하는 최소 구성이 필요합니다. 카탈로그 속성 파일 및 저장소 검색 결과(아래 설명 참조)를 제외하고는 제공된 모든 파일이 고유하고, 따라서 필요한 경우 무제한으로 안전하게 캐시할 수 있습니다. 또한 모든 저장소 응답에는 캐시 파일이 실수로 사용되지 않는 것을 방지하기 위해 적합한 HTTP 헤더를 포함합니다.

Apache를 캐싱 프록시로 구성하는 방법에 대한 자세한 내용은 [Caching Guide](#)를 참조하십시오.

CacheRoot 지시어를 사용하여 디렉토리가 캐시된 파일을 포함하도록 지정합니다. 지정된 디렉토리를 Apache 프로세스가 쓸 수 있는지 확인합니다. Apache가 이 디렉토리에 쓸 수 없는 경우 명시적인 오류 메시지가 출력되지 않습니다.

```
CacheRoot /tank/proxycache
```

Apache는 특정 디렉토리에 대해 캐싱을 사용으로 설정합니다. 다음 지시어에 표시된 것처럼 저장소 서버가 서버에서 모든 콘텐츠를 캐시하도록 해야 할 수 있습니다.

```
CacheEnable disk /
```

CacheMaxFileSize 지시어를 사용하여 캐시할 최대 파일 크기를 설정합니다. 1MB의 Apache 기본값은 대부분의 저장소에서 너무 작을 수 있습니다. 다음 지시어는 캐시되는 파일의 최대 크기를 1GB로 설정합니다.

```
CacheMaxFileSize 1000000000
```

기본 파일 시스템에서 성능을 최적화하기 위해 디스크 내장 캐시의 디렉토리 구조를 조정합니다. ZFS 데이터 세트에서 다중 디렉토리 레벨은 하나의 디렉토리에 있는 파일 수보다 성능에 더 많은 영향을 줍니다. 따라서 각 디렉토리에 파일 수가 많은 하나의 디렉토리 레벨을 구성하십시오. CacheDirLevels 및 CacheDirLength 지시어를 사용하여 디렉토리 구조를 제어합니다. CacheDirLevels를 1로 설정합니다. CacheDirLength를 디렉토리 수와 디렉토리당 파일 수 간에 적절한 균형을 이룰 수 있는 값으로 설정합니다. 아래에 설정된 값 2는 4096개의 디렉토리를 생성합니다. 자세한 내용은 [Disk-based Caching](#) 설명서를 참조하십시오.

CacheDirLevels 1
CacheDirLength 2

카탈로그 속성 파일에 대한 캐시 고려 사항

저장소 카탈로그 속성 파일(catalog.attrs)에는 저장소 카탈로그의 현재 상태가 포함됩니다. 이 파일은 캐싱을 보장하도록 충분히 클 수 있습니다. 하지만 백엔드 저장소의 카탈로그가 변경된 경우 이 파일이 사용되지 않을 수 있습니다. 다음 두 가지 방법 중 하나를 사용하여 이 문제를 해결할 수 있습니다.

- 이 파일을 캐시하지 않습니다. 이 솔루션은 추가 트래픽이 중요한 고려 사항이 아닌 고대역폭 환경에서 저장소 서버를 실행하는 경우에 가장 적합합니다. 다음 httpd.conf 파일 부분은 catalog.attrs 파일을 캐시하지 않도록 지정하는 방법을 보여 줍니다.

```
<LocationMatch ".*catalog.attrs">
    Header set Cache-Control no-cache
</LocationMatch>
```

- 백엔드 저장소의 카탈로그가 업데이트될 때마다 캐시에서 이 파일을 정렬합니다.

검색을 위한 캐시 고려 사항

패키지 저장소를 검색하면 요청에 따라 사용자 정의 응답이 생성됩니다. 따라서 검색 결과는 캐시하는 데 적합하지 않습니다. 저장소 서버는 검색 결과가 캐시에서 사용되지 않는 것을 방지하기 위해 적합한 HTTP 헤더를 설정합니다. 하지만 캐싱으로 얻을 수 있는 예상되는 대역폭 절감 효과는 크지 않습니다. 다음 httpd.conf 파일 부분은 검색 결과를 캐시하지 않도록 지정하는 방법을 보여 줍니다.

```
<LocationMatch ".*search/\d/.*">
    Header set Cache-Control no-cache
</LocationMatch>
```

웹 프록시 뒤에서 저장소 서버 실행

pkg(5) 저장소 서버는 로컬 네트워크 또는 인터넷에서 저장소에 대한 액세스를 쉽게 제공할 수 있게 해줍니다. 하지만 저장소 서버는 하나의 도메인 이름 또는 정교한 접두어 아래에서 여러 저장소를 제공하도록 지원하지 않습니다. 하나의 도메인 이름 아래에서 여러 저장소를 호스트하려면 웹 프록시 뒤에서 저장소 서버를 실행하십시오. 웹 프록시 뒤에서 저장소 서버를 실행하면 여러 저장소 간에 로드 밸런싱을 사용으로 설정하고 콘텐츠 캐싱을 사용으로 설정하여 서버의 성능을 향상시킬 수도 있습니다.

이 절의 예에서는 Apache 웹 서버를 프록시 소프트웨어로 사용합니다. Oracle Solaris 11.1 OS에는 web/server/apache-22 패키지의 Apache 웹 서버와 /etc/apache2/2.2의 기본 httpd.conf 파일이 포함됩니다. Apache 웹 서버는 svc:/network/http:apache22 서비스를 사용으로 설정하여 활성화됩니다. 자세한 내용은 [Apache HTTP Server Version 2.2 Documentation](#)을 참조하십시오.

이 예에 표시된 원칙을 모든 프록시 서버 소프트웨어에 적용할 수 있어야 합니다.

권장되는 일반 Apache 구성 설정

다음 설정은 성능과 보안에 영향을 줍니다.

Apache DEFLATE 필터를 사용으로 설정합니다.

HTTP 클라이언트는 HTTP 요청에서 압축된 데이터를 수신할 수 있음을 서버에 알릴 수 있습니다. Apache DEFLATE 필터를 사용으로 설정하면 카탈로그 및 매니페스트와 같은 메타 데이터의 전송 크기를 크게 줄일 수 있습니다. 카탈로그 및 매니페스트와 같은 메타 데이터는 종종 90%까지 압축됩니다.

```
AddOutputFilterByType DEFLATE text/html application/javascript text/css text/plain
```

인코딩된 슬래시를 디코딩하지 않습니다.

패키지는 URL 인코딩된 슬래시를 포함할 수 있습니다. 이러한 슬래시가 디렉토리 구분 기호로 해석되지 않도록 하려면 Apache에 이를 디코딩하지 않도록 지시합니다.

```
AllowEncodedSlashes NoDecode
```

주 - 이 설정을 생략하면 검색 기능에 부정적인 영향을 줄 수 있습니다.

더 많은 파이프라인된 요청을 허용합니다.

클라이언트가 연결을 닫지 않고도 더 많은 수의 파이프라인된 요청을 수행하도록 허용하려면 MaxKeepAliveRequests 값을 늘립니다. Apache의 기본값인 100은 너무 낮은 값입니다.

```
MaxKeepAliveRequests 10000
```

응답의 최대 대기 시간을 설정합니다.

프록시 시간 초과는 백엔드 저장소가 응답할 때까지 Apache가 기다리는 시간을 설정합니다. 대부분의 작업에서는 30초 정도가 적합합니다. 검색할 때 결과 수가 너무 많으면 시간이 상당히 오래 걸릴 수 있습니다. 이러한 검색을 수용하기 위해서는 시간 초과 값을 더 높게 설정해야 할 수 있습니다.

```
ProxyTimeout 30
```

전달 프록싱을 사용 안함으로 설정합니다.

전달 프록싱이 사용 안함으로 설정되었는지 확인합니다.

```
ProxyRequests Off
```

Apache 구성 예

이 절에서는 다중 저장소, 로드 밸런싱되지 않은 설정 및 로드 밸런싱된 설정을 보여 줍니다.

단순 접두어가 지정된 프록시 구성

이 예에서는 로드 밸런싱되지 않은 저장소 서버에 대한 기본 구성을 보여 줍니다. 이 예에서는 `http://pkg.example.com/myrepo`를 `internal.example.com:10000`에 연결합니다.

이 예제에 설명되지 않은 다른 등록 정보 설정에 대한 지침은 [27 페이지 “여러 저장소 서버 인스턴스를 사용하여 여러 저장소 제공”](#)을 참조하십시오.

저장소 서버를 액세스할 수 있는 URL의 이름을 지정하는 `pkg/proxy_base` 설정을 사용하여 저장소 서버를 구성해야 합니다. 다음 명령을 사용하여 `pkg/proxy_base`를 설정합니다.

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

`pkg(5)` 클라이언트는 네트워크 작업을 수행할 때 저장소 서버에 대해 20개의 병렬 연결을 엽니다. 저장소 스레드 수가 특정 시점에 서버에 대한 예상 연결 수와 일치하는지 확인합니다. 다음 명령을 사용하여 저장소당 스레드 수를 설정합니다.

```
$ svccfg -s pkg/server:repo "setprop pkg/threads = 200"
$ svcadm refresh pkg/server:repo
$ svcadm restart pkg/server:repo
```

`nocanon`을 사용하여 URL의 정규화를 억제합니다. 이 설정은 검색이 올바르게 작동하게 하는 데 중요합니다. 또한 백엔드 연결 수를 저장소 서버가 제공하는 스레드 수로 제한합니다. 다음 `httpd.conf` 파일 부분은 하나의 저장소 서버를 프록시하는 방법을 보여 줍니다.

```
Redirect /myrepo http://pkg.example.com/myrepo/
ProxyPass /myrepo/ http://internal.example.com:10000/ nocanon max=200
```

하나의 도메인 아래의 다중 저장소

프록시 뒤에서 저장소 서버를 실행하는 가장 중요한 이유는 하나의 도메인 이름에서 서로 다른 접두어로 여러 저장소를 쉽게 실행하기 위해서입니다. [31 페이지 “단순 접두어가 지정된 프록시 구성”](#)의 예는 다중 저장소를 지원하도록 쉽게 확장해서 사용할 수 있습니다.

이 예에서는 하나의 도메인 이름에서 세 개의 서로 다른 접두어가 세 개의 서로 다른 패키지 저장소에 연결되어 있습니다.

- `http://pkg.example.com/repo_one`은 `internal.example.com:10000`에 연결됩니다.
- `http://pkg.example.com/repo_two`는 `internal.example.com:20000`에 연결됩니다.
- `http://pkg.example.com/xyz/repo_three`는 `internal.example.com:30000`에 연결됩니다.

pkg(5) 저장소 서버는 SMF 관리 서비스입니다. 따라서 동일한 호스트에서 여러 저장소 서버를 실행하려면 단순히 새로운 서비스 인스턴스를 만들기만 하면 됩니다.

```
$ svccfg -s pkg/server add repo1
$ svccfg -s pkg/server:repo1 setprop pkg/property=value
$ ...
```

이전 예와 같이 각 저장소 서버는 200개의 스레드로 실행됩니다.

```
Redirect /repo_one http://pkg.example.com/repo_one/
ProxyPass /repo_one/ http://internal.example.com:10000/ nocanon max=200

Redirect /repo_two http://pkg.example.com/repo_two/
ProxyPass /repo_two/ http://internal.example.com:20000/ nocanon max=200

Redirect /xyz/repo_three http://pkg.example.com/xyz/repo_three/
ProxyPass /xyz/repo_three/ http://internal.example.com:30000/ nocanon max=200
```

로드 밸런싱된 구성

Apache 로드 밸런서 뒤에서 저장소 서버를 실행해야 할 수 있습니다. 이 예에서는 `http://pkg.example.com/myrepo`를 `internal1.example.com:10000` 및 `internal2.example.com:10000`에 연결합니다.

31 페이지 “단순 접두어가 지정된 프록시 구성”에 표시된 것과 같이 적합한 `proxy_base` 설정을 사용하여 저장소 서버를 구성합니다.

각 저장소가 실행 중인 스레드 수를 로드 밸런서 설정에 있는 저장소 수로 나눈 값으로 백엔드 연결 수를 제한합니다. 그렇지 않으면 Apache가 저장소에 대해 사용 가능한 것보다 많은 수의 연결을 열고 작동이 정지되어 성능이 저하될 수 있습니다. `max=` 매개변수를 사용하여 각 저장소에 대한 최대 병렬 연결 수를 지정합니다. 아래의 예에서는 각각 200개의 스레드를 실행하는 두 개의 저장소를 보여 줍니다. 저장소 스레드 수를 설정하는 방법에 대한 예는 31 페이지 “단순 접두어가 지정된 프록시 구성”을 참조하십시오.

```
<Proxy balancer://pkg-example-com-myrepo>
  # depot on internal1
  BalancerMember http://internal1.example.com:10000 retry=5 max=100

  # depot on internal2
  BalancerMember http://internal2.example.com:10000 retry=5 max=100
</Proxy>

Redirect /myrepo http://pkg.example.com/myrepo/
ProxyPass /myrepo/ balancer://pkg-example-com-myrepo/ nocanon
```

완전한 로드 밸런싱 예

다음 예에서는 로드 밸런싱된 저장소 서버 설정 및 로드 밸런싱되지 않은 저장소 서버 설정을 호스트하는 저장소 서버에 대해 `httpd.conf` 파일에 추가하기 위해 필요한 모든 지시어가 포함되어 있습니다.

이 예에서는 하나의 도메인 이름에서 두 개의 서로 다른 접두어가 세 개의 서로 다른 패키지 저장소에 연결되어 있습니다.

- `http://pkg.example.com/repo_one`은 `internal1.example.com:10000` 및 `internal2.example.com:10000`에 연결됩니다.
- `http://pkg.example.com/repo_two`는 `internal1.example.com:20000`에 연결됩니다.

```
AddOutputFilterByType DEFLATE text/html application/javascript text/css text/plain
AllowEncodedSlashes NoDecode

MaxKeepAliveRequests 10000

ProxyTimeout 30

ProxyRequests Off

<Proxy balancer://pkg-example-com-repo_one>
    # depot on internal1
    BalancerMember http://internal1.example.com:10000 retry=5 max=100

    # depot on internal2
    BalancerMember http://internal2.example.com:10000 retry=5 max=100
</Proxy>

Redirect /repo_one http://pkg.example.com/repo_one/
ProxyPass /repo_one/ balancer://pkg-example-com-repo_one/ nocanon
Redirect /repo_two http://pkg.example.com/repo_two/
ProxyPass /repo_two/ http://internal.example.com:20000/ nocanon max=200
```

