

Oracle® Solaris 11.2의 시스템 서비스 관리

ORACLE®

부품 번호: E53842
2014년 7월

Copyright © 2013, 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	13
1 서비스 관리 기능 소개	15
SMF 기능	15
이 릴리스의 새로운 기능	16
SMF 개념 및 구성 요소	16
SMF 서비스	18
서비스 모델	19
서비스 이름	19
서비스 상태	20
서비스 종속성	20
서비스 다시 시작 프로그램	21
서비스 등록 정보 및 등록 정보 그룹	22
서비스 구성 저장소	23
구성 파일 및 SMF 서비스	26
서비스 관리 권한	27
2 서비스에 대한 정보 가져오기	29
시스템의 서비스 나열	29
서비스 상태 표시	29
서비스에 대한 추가 정보 표시	30
선택한 서비스 정보 표시	31
서비스 종속성 표시	32
종속성 그룹	33
서비스가 종속된 인스턴스 나열	33
서비스에 종속된 인스턴스 나열	34
서비스를 자동으로 시작할지 여부 표시	35
서비스 상태에 대한 추가 정보 가져오기	36
서비스 로그 파일 보기	37
서비스 구성 검사	39

등록 정보 및 등록 정보 그룹의 설명 표시	39
서비스 및 인스턴스 등록 정보 값 표시	40
등록 정보 그룹 유형의 등록 정보 표시	42
값이 설정된 계층 표시	43
지정된 스냅샷에 값 표시	44
구성 사용자 정의 표시	45
이벤트 알림 매개변수 표시	45
3 서비스 관리	49
SMF 서비스 인스턴스 관리	49
서비스 시작	49
서비스 중지	52
서비스 다시 시작	54
서비스 구성 다시 읽기	55
서비스 삭제	56
상태 전환 및 FMA 이벤트에 대한 알림 구성	57
4 서비스 구성	59
서비스 구성 명령 사용	59
등록 정보 편집기 호출	60
대화식으로 또는 파일을 사용하여 svccfg 호출	61
등록 정보 값 설정	62
▼ ttymon 등록 정보 값을 수정하는 방법	64
▼ 서비스 프로세스 환경에 대한 환경 변수를 수정하는 방법	65
등록 정보 그룹, 등록 정보 및 등록 정보 값 추가	66
등록 정보 그룹, 등록 정보 및 등록 정보 값 삭제	68
관리 구성 삭제	68
비관리 구성 삭제	70
서비스 인스턴스 추가	71
스냅샷 되돌리기	72
매니페스트와 프로파일 가져오기 및 적용	72
여러 시스템 구성	73
▼ svcbundle을 사용하여 프로파일을 만드는 방법	73
▼ svccfg를 사용하여 프로파일을 만드는 방법	74
inetd로 제어되는 서비스 수정	75
▼ inetd 제어 서비스에 대한 등록 정보 값을 변경하는 방법	75
파일에 의해 구성되는 서비스 수정	76
5 SMF를 사용하여 응용 프로그램 제어	79

SMF 서비스 만들기	79
▼ 서비스 번들 생성기 도구를 사용하여 SMF 서비스를 만드는 방법	80
Oracle 데이터베이스 인스턴스를 시작 또는 중지하는 서비스 만들기	81
서비스, 인스턴스, 등록 정보 그룹 및 등록 정보 이름 지정	87
▼ 실행 제어 스크립트를 SMF 서비스로 변환하는 방법	88
스텐실을 사용하여 구성 파일 만들기	89
▼ 스텐실 서비스를 만드는 방법	89
Oracle Solaris의 스텐실 서비스 예	90
A SMF 모범 사례 및 문제 해결	95
SMF 모범 사례	95
서비스 문제 해결	96
구성 변경 사항 이해	96
성능 저하, 오프라인 또는 유지 관리 상태인 인스턴스 복구	97
인스턴스를 성능 저하 또는 유지 관리 상태로 표시	100
저장소 문제 진단 및 복구	100
시작 메시징 양 지정	103
부트할 SMF 마일스톤 지정	104
SMF를 사용하여 시스템 부트 문제 조사	106
inetd 서비스를 SMF 서비스로 변환	107
색인	109

그림

그림 1-1	서비스 관리 기능 프레임워크	17
그림 2-1	서비스 종속성 관계	32

표

표 2-1	종속성이 중지된 후 자동으로 서비스 다시 시작	36
표 A-1	SMF 시작 메시지 로깅 레벨	104
표 A-2	SMF 부트 마일스톤 및 해당 실행 레벨	105

코드 예

예 2-1	모든 사용으로 설정된 서비스 나열	29
예 2-2	모든 설치된 서비스 나열	30
예 2-3	모든 서비스 인스턴스 나열	30
예 2-4	계약 서비스에서 시작한 프로세스 표시	31
예 2-5	프로세스가 중지한 후 자동으로 다시 시작되는 계약 서비스 표시	31
예 2-6	현재 사용 중인 인스턴스 및 상속된 등록 정보 나열	40
예 2-7	현재 사용 중인 지정된 등록 정보 또는 등록 정보 그룹 나열	40
예 2-8	편집 보기에서 서비스 및 인스턴스 값 나열	41
예 2-9	편집 보기에서 지정된 등록 정보 또는 등록 정보 그룹 나열	42
예 2-10	등록 정보 그룹 및 해당 유형 표시	42
예 2-11	등록 정보 그룹 유형의 등록 정보 나열	43
예 3-1	서비스 인스턴스를 영구적으로 사용으로 설정	51
예 3-2	서비스 인스턴스를 일시적으로 사용으로 설정	51
예 3-3	서비스 인스턴스를 사용 안함으로 설정	53
예 3-4	서비스 상태 이벤트에 대한 전역 알림 구성	57
예 3-5	지정된 서비스 인스턴스에 대한 알림 구성	58
예 3-6	FMA 이벤트에 대한 알림 구성	58
예 3-7	알림 설정 삭제	58
예 4-1	단순 값 설정	62
예 4-2	콜론 문자가 포함된 값 설정	62
예 4-3	포함된 공백이 들어 있는 값 설정	62
예 4-4	값 세트인 값 설정	63
예 4-5	값 추가	63
예 4-6	addpg를 사용하여 새 등록 정보 그룹 만들기	66
예 4-7	setprop를 사용하여 새 등록 정보 만들기	67
예 4-8	addpropvalue를 사용하여 새 등록 정보 만들기	67
예 4-9	모든 등록 정보 값 삭제	68
예 4-10	모든 일치하는 등록 정보 값 삭제	69
예 4-11	등록 정보 삭제	69
예 4-12	등록 정보 그룹 삭제	69

예 4-13	사용자 정의 삭제	70
예 4-14	번들 지원이 포함된 구성 삭제	70
예 4-15	구성 마스킹 해제	71
예 4-16	svcbundle을 사용하여 프로파일 자동 설치	74
예 4-17	inetd 제어 서비스가 시작될 때 실행할 명령 수정	76
예 5-1	생성된 매니페스트 자동 설치	81

이 설명서 사용

- **개요** - Oracle Solaris SMF(서비스 관리 기능)를 사용하는 방법에 대해 설명합니다. SMF는 더 광범위한 Oracle Solaris 예측적 자가 치유 기능의 구성 요소 중 하나입니다.
- **대상** - 시스템 서비스를 관리하는 시스템 관리자
- **필요한 지식** - Oracle Solaris 시스템 관리 경험

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.

서비스 관리 기능 소개

Oracle Solaris SMF(서비스 관리 기능) 프레임워크에서는 시스템 및 응용 프로그램 서비스를 관리합니다. SMF에서는 시스템 작업에 필수적인 중요 시스템 서비스를 관리하고 데이터 베이스나 웹 서버와 같은 응용 프로그램 서버를 관리합니다. SMF는 하드웨어 또는 소프트웨어 오류가 발생하더라도 필수 시스템 및 응용 프로그램 서비스를 지속적으로 실행하게 하여 시스템 가용성을 향상합니다.

SMF는 구성 파일 대신 서비스 관리에 사용되고 시작 응용 프로그램에 사용할 권장 방식입니다. SMF는 `init` 스크립팅 시작 방식, `inetd.conf` 구성 및 대부분 `rc?.d` 스크립트를 대체합니다. SMF는 가능한 한 기존 관리 방식과의 호환성을 유지합니다. 예를 들어 대부분 고객 및 ISV 제공 `rc` 스크립트는 SMF 없이 작동한 것과 같은 방식으로 작동합니다.

SMF 기능

SMF 프레임워크는 Oracle Solaris 11 시스템에서 항상 활성 상태입니다. SMF는 다음 기능을 제공합니다.

- 더 빠르게 부트합니다. SMF는 서비스 종속성에 따라 서비스를 병렬로 시작하여 대규모 시스템의 부트 속도를 높입니다.
- 실패한 서비스를 다시 시작합니다. SMF 서비스에는 다른 서비스와의 종속성 관계가 잘 정의되어 있습니다. 서비스가 실패하면 SMF에서는 영향을 받는 종속 서비스를 보고합니다. SMF는 실패한 서비스를 종속성 순서대로 다시 시작하려고 합니다.
- 서비스를 검사합니다. 서비스와 프로세스 간 관계를 표시합니다. 서비스 등록 정보 값을 표시합니다.
- 서비스를 관리합니다. 서비스를 사용으로 설정, 사용 안함으로 설정 및 다시 시작합니다. 이러한 변경 사항은 업그레이드와 재부트를 거쳐도 지속될 수 있습니다. 그렇지 않으면 사용자가 일시적 변경 사항을 지정할 수 있습니다.
- 서비스를 구성합니다.
 - 서비스 등록 정보 값을 변경합니다.
 - 사용자 정의 등록 정보를 추가하고 삭제합니다.
- 서비스 변경 사항을 감사합니다. SMF는 모든 관리 변경에 대한 Solaris 감사 레코드를 서비스 또는 해당 등록 정보에 씁니다. SMF는 관리자가 등록 정보 값 또는 서비스 상태를 설정했는지를 표시할 수 있습니다.

- 등록 정보를 수정하는 기능과 서비스를 사용으로 설정하거나, 사용 안함으로 설정하거나, 다시 시작하는 기능을 포함하여 루트가 아닌 사용자에게 작업을 안전하게 위임합니다.
- 새 서비스를 만듭니다. 간편하게 기본 서비스의 새 인스턴스를 만들거나, 기존 서비스를 복사하고 수정하거나, 서비스 만들기 도구를 사용합니다.
- 서비스 문제를 디버깅합니다. 간편하게 사용으로 설정된 서비스가 실행 중이 아닌 이유 또는 서비스가 다른 서비스 실행을 차단하는 이유에 대한 설명을 표시합니다.
- 특정 소프트웨어 이벤트나 하드웨어 오류에 대해 사용자에게 알리는 방법을 구성합니다.
- `inetd.conf` 구성을 SMF 서비스로 변환합니다.
- SMF 서비스 등록 정보를 구성 파일로 변환합니다. 이 방식은 SMF에서 관리되지만 구성 파일을 필요로 하는 응용 프로그램과 상호 작용하는 서비스에 대한 브리지를 제공합니다.

이 릴리스의 새로운 기능

다음 SMF 기능은 이 릴리스에서 새로운 기능입니다.

동기적 다시 시작 및 새로 고침 작업

`svcadm` 명령의 `restart` 및 `refresh` 하위 명령에 대한 새 `-s` 옵션은 `enable` 및 `disable` 하위 명령에 대한 기존 `-s` 옵션과 비슷한 동기 작업을 지정합니다.

로그 파일 표시

`svcs` 명령에 대한 새 `-L` 옵션은 지정된 서비스 인스턴스에 대한 전체 로그 파일, 로그 파일의 마지막 몇 행 또는 로그 파일의 전체 경로 이름을 표시합니다.

일련의 서비스 인스턴스에 대한 일괄 처리 작업

새 `libscf(3LIB)` 함수를 호출하면 일반적인 관리 작업을 동기적 또는 비동기적으로 일괄 그룹화하고 실행할 수 있습니다. `maintenance` 또는 `degraded` 상태를 사용으로 설정, 사용 안함으로 설정, 다시 시작, 표시하고 새로 고치고 지우고 일련의 서비스 인스턴스에 대한 마일스톤 간에 전환할 수 있습니다.

SMF 서비스 등록 정보를 구성 파일로 변환

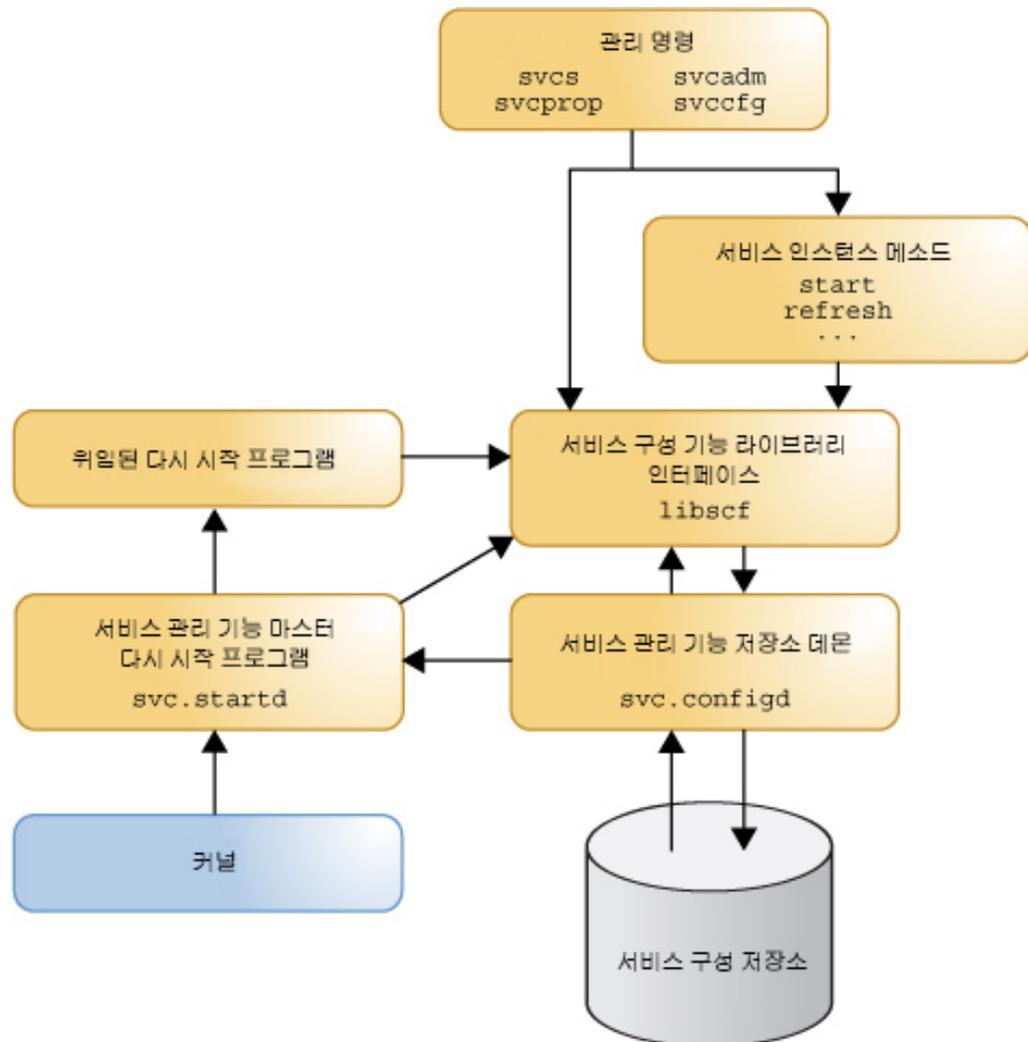
SMF에서 관리되지만 구성 파일을 필요로 하는 응용 프로그램과 상호 작용하는 서비스의 경우 새 스텐실 기능은 서비스 구성 저장소에서 서비스에 대해 정의된 등록 정보 값에서 구성 파일을 만들고 업데이트합니다. 스텐실 파일이라는 새로운 파일에는 SMF에서 생성될 구성 파일의 구조 정의가 포함됩니다. 새로운 `svcio` 유틸리티는 SMF 서비스의 스텐실 파일 및 등록 정보에 있는 정의에서 구성 파일을 생성합니다. 이 기능을 사용하여 기존 응용 프로그램을 최소한으로 변경하고 SMF 구성 관리를 활용할 수 있습니다.

SMF 개념 및 구성 요소

이 절에서는 본 설명서의 나머지 부분에서 사용되는 용어를 정의합니다. 다음 그림은 SMF 프레임워크의 주 구성 요소를 보여줍니다. 이미지를 부트할 때 SMF는 필요한 경우 서비스

구성 저장소를 업데이트하고, 저장소 데이터를 읽고, 사용으로 설정된 서비스 인스턴스를 시작합니다. 다음 그림에서 libscf는 다시 시작 프로그램이 서비스 구성 저장소와 상호 작용하기 위해 사용하는 라이브러리 인터페이스입니다. 서비스 구성 저장소와 libscf 라이브러리 인터페이스 간 상호 작용은 svc.configd 데몬에서 관리됩니다. svcs, svcprop, svcadm 및 svccfg 명령은 관리자가 서비스 구성 저장소와 상호 작용하기 위해 사용하는 인터페이스입니다.

그림 1-1 서비스 관리 기능 프레임워크



SMF 서비스

SMF 서비스는 다음과 같은 시스템 엔티티를 나타내는 영구 실행 응용 프로그램입니다.

- 데이터베이스 또는 웹 서버와 같은 응용 프로그램 서비스
- 필수 시스템 서비스
- 장치의 소프트웨어 상태
- 커널 구성 정보
- 시스템 `init` 상태에 해당하는 마일스톤

서비스 인스턴스는 서비스의 하위이고 응용 프로그램 및 기타 서비스 인스턴스에 기능과 종속성 관계를 제공합니다. 인스턴스만 상태가 있고 시작 및 중지할 수 있습니다. 인스턴스가 하드웨어 또는 소프트웨어 오류와 같은 어떤 이유로 실패하면 SMF는 자동으로 오류를 감지하고 해당 인스턴스와 모든 종속 인스턴스를 다시 시작합니다.

서비스 인스턴스를 사용하여 서비스의 여러 구성을 동시에 실행할 수 있습니다. 서비스 인스턴스는 일반적인 서비스 구성을 상속하고 사용자 정의합니다. 예를 들어 한 인스턴스가 포트 80에서 수신 대기하도록 구성되고 다른 인스턴스가 포트 1008에서 수신 대기하도록 구성된 웹 서버 서비스를 정의할 수 있습니다. 대부분 서비스에는 `default` 인스턴스가 있습니다. SMF를 사용하여 구성을 저장하지만 프로그램을 실행하지 않는 일부 서비스에는 인스턴스가 없습니다. 예를 들어 `x11/x11-server` 서비스에는 인스턴스가 없습니다.

SMF 서비스는 서비스 매니페스트라는 파일에서 설명됩니다. 매니페스트는 서비스 인스턴스, 종속성, 구성 등록 정보 및 메소드를 설명합니다. 서비스 메소드는 서비스 인스턴스를 시작, 중지하고 새로 고칩니다. 메소드는 데몬, 기타 이진 실행 파일 또는 실행 가능한 스크립트일 수 있습니다. 서비스 프로파일 파일을 사용하여 주로 등록 정보를 추가하고 등록 정보 값을 추가 및 대체하는 방식으로 기존 서비스를 사용자 정의할 수 있습니다. “저장소 계층” [24]의 설명대로 새 등록 정보와 값은 매니페스트에 지정된 값 위에 계층화됩니다. 매니페스트 및 프로파일에 대한 자세한 내용은 “서비스 번들” [23]을 참조하십시오. 또한 프로파일은 “여러 시스템 구성” [73]에 설명된 대로 같은 사용자 정의 구성을 여러 시스템에 적용할 수 있는 우수한 도구입니다.

서비스 정보는 *SMF* 데이터베이스라고도 하는 서비스 구성 저장소에 저장됩니다. 서비스 구성 저장소에는 시스템에 있는 각 서비스 인스턴스의 현재 상태와 각 서비스 및 서비스 인스턴스에 대한 구성 데이터가 저장됩니다. 데이터는 “저장소 계층” [24]에 설명된 대로 값이 수정된 방법에 따라 계층에 저장됩니다.

SMF는 사용으로 설정, 사용 안함으로 설정, 새로 고침 및 다시 시작을 포함하여 서비스 인스턴스에서 호출할 수 있는 작업을 제공합니다. 각 서비스 인스턴스는 이러한 관리 작업을 수행하는 다시 시작 프로그램에서 관리됩니다. 일반적으로 다시 시작 프로그램은 메소드를 실행하여 한 상태에서 다른 상태로 서비스 인스턴스를 이동하는 방식으로 작업을 수행합니다. 다시 시작 프로그램에 대한 자세한 내용은 “서비스 다시 시작 프로그램” [21]을 참조하십시오.

마일스톤 서비스는 시스템 준비 레벨을 나타내는 특수 서비스 유형입니다. 마일스톤은 시작을 위해 다른 서비스 인스턴스가 종속된 서비스입니다. 예를 들어 실행 레벨은 `svc:/`

milestone/multi-user-server와 같은 마일스톤 서비스로 나타냅니다. 마일스톤을 사용하여 svc:/milestone/devices, svc:/milestone/network 또는 svc:/milestone/name-services와 같은 서비스 그룹 준비를 표시할 수 있습니다.

서비스 모델

SMF 서비스는 다음 세 가지 모델의 하나입니다.

일시적 서비스

서비스는 몇 가지 작업을 수행한 후 오래 실행되는 프로세스를 시작하지 않고 종료됩니다.

하위 또는 대기 서비스

하위 프로세스가 완전히 종료될 때마다 서비스가 다시 시작됩니다. 완전히 종료되는 하위 프로세스는 오류로 처리되지 않습니다.

계약 또는 데몬 서비스

서비스는 오래 실행되는 데몬을 시작하거나 서비스 계약의 일부로 함께 연결된 여러 관련된 프로세스를 시작합니다. 계약 서비스는 이 서비스에서 시작하는 프로세스와 모든 종속 서비스 및 해당 시작 순서를 관리합니다. 상위 레벨 서비스만 관리하면 됩니다.

서비스 이름

각 서비스와 서비스 인스턴스는 FMRI(Fault Management Resource Identifier)로 나타냅니다. 서비스 인스턴스에 대한 전체 FMRI의 형식은 다음과 같습니다.

```
svc:/service_name:instance_name
```

*service_name*은 network/dns/client 또는 application/pkg/server와 같은 계층적 이름입니다. 마지막 슬래시(/) 앞에 나오는 *service_name*의 구성 요소는 서비스의 범주입니다. application, device, milestone, network, system과 같은 범주를 사용하여 서비스의 목적을 식별할 수 있습니다.

site 범주는 고유한 SMF 서비스를 만들 때 이름 충돌을 피하도록 돕기 위해 예약되어 있습니다. 예를 들어 svc:/site/tool이라는 사이트별 서비스는 svc:/tool이라는 Oracle Solaris 서비스와 충돌하지 않습니다.

서비스 인스턴스 이름은 콜론 문자 뒤의 상위 서비스 이름에 추가됩니다. 예를 들어 svc:/system/identity:node 및 svc:/system/identity:domain은 svc:/system/identity 서비스의 인스턴스입니다.

스크립트에서 모범 사례는 전체 서비스 인스턴스 이름을 사용하는 것입니다. 대화식으로 이름은 이름의 맨 오른쪽 부분으로 축약되어 고유한 이름이 될 수 있습니다. 예를 들어 `svc:/system/identity`는 `identity`로 축약되고 `svc:/system/identity:domain`은 `identity:domain`으로 축약될 수 있습니다. 인스턴스 이름 앞에는 서비스 이름의 일부가 표시되고 뒤에는 콜론 문자가 표시되어야 합니다.

서비스 상태

특정 시점에 SMF 서비스 인스턴스는 다음 상태의 하나에 있습니다.

- `degraded` - 인스턴스가 실행 중이거나 실행할 수 있지만 제한된 기능으로 작동 중입니다.
- `disabled` - 서비스 인스턴스가 사용으로 설정되어 있지 않고 실행 중이 아니거나 실행할 수 없습니다.
- `maintenance` - 인스턴스가 사용으로 설정되어 있지만 실행할 수 없습니다. 관리 작업이 아직 완료되지 않아 인스턴스가 `maintenance` 상태 사이에서 전환되고 있을 수 있습니다. 그렇지 않으면 문제를 해결하는 데 관리 작업이 필요합니다.
- `offline` - 인스턴스가 사용으로 설정되어 있지만 실행 중이 아니거나 실행할 수 없습니다. 예를 들어 사용으로 설정된 서비스의 종속성이 충족되지 않으면 서비스가 `offline` 상태에서 유지됩니다.
- `online` - 인스턴스가 사용으로 설정되어 있고 실행 중이거나 실행할 수 있습니다. `online` 상태는 모든 종속성이 충족되고 제대로 구성된 서비스 인스턴스에 대한 예상된 작동 상태입니다.
- `uninitialized` - 이 상태는 모든 서비스에 대한 초기 상태입니다.

서비스 인스턴스는 관리 작업이나 종속 서비스 상태와 같은 조건에 따라 상태 사이에 전환됩니다. 예를 들어 `disabled` 상태였던 인스턴스를 사용으로 설정하면 새로 사용으로 설정된 인스턴스가 먼저 `offline` 상태로 전환되고, 모든 종속성이 충족되면 `online` 상태로 전환됩니다. 이러한 서비스 상태 및 서비스 인스턴스가 이러한 상태 사이에서 전환되는 방식에 대한 자세한 내용은 [smf\(5\)](#) 매뉴얼 페이지를 참조하십시오.

서비스 종속성

서비스에는 서비스, 서비스 인스턴스 또는 파일에 대한 종속성이 있을 수 있습니다. 서비스 종속성은 서비스 간 관계를 정의합니다.

종속성 관계에 따라 서비스가 시작되고 자동으로 중지되는 시기가 결정됩니다. 사용으로 설정된 서비스의 종속성이 충족되지 않으면 서비스가 `offline` 상태에 있습니다. 사용으로 설정된 서비스의 종속성이 충족되면 서비스가 시작됩니다. 서비스 시작에 성공하면 서비스가 `online` 상태로 전환됩니다.

서비스가 상태 사이에서 전환될 때 서비스 종속성이 재평가됩니다. 충족되는 서비스 종속성이 나중에 충족되지 않을 수 있습니다. 파일 종속성은 한 번만 평가됩니다.

종속성은 필수이거나 선택 사항일 수 있습니다. 서비스 종속성은 실행 중이거나 사용 안함으로 설정되어야 할 수 있습니다. 종속 서비스는 서비스 종속성의 하나가 중지되거나 새로 고쳐지면 다시 시작되거나 다시 시작되지 않도록 구성할 수 있습니다.

종속성 관계는 다음 기능을 허용합니다.

- 확장 가능하고 재생 가능한 초기화 프로세스
- 독립 서비스를 병렬로 시작하여 병렬 기능이 있는 시스템에서 더 빠른 시스템 시작
- 오류가 직접 영향을 미치는 서비스만 다시 시작하고 해당 서비스를 올바른 종속성 순서로 다시 시작하여 정확한 오류 방지 및 오류 복구

서비스 다시 시작 프로그램

각 SMF 서비스 인스턴스는 다시 시작 프로그램에서 관리됩니다. 다시 시작 프로그램은 인스턴스 구성을 검색하고 실행 환경을 제공합니다. 모든 다시 시작 프로그램에 공통적인 정보는 [smf_restarter\(5\)](#)를 참조하십시오.

마스터 다시 시작 프로그램 데몬

svc.startd 데몬은 SMF에 대한 마스터 다시 시작 프로그램 데몬 및 모든 서비스 인스턴스에 대한 기본 다시 시작 프로그램입니다. svc.startd 데몬은 모든 서비스 인스턴스와 해당 종속성에 대한 상태를 관리합니다. 인스턴스가 online 상태로 이동하면 종속성이 충족되므로 마스터 다시 시작 프로그램이 다른 인스턴스의 시작 메소드를 호출하거나 위임된 다시 시작 프로그램에 시작 메소드를 호출하게 지시합니다. 마스터 다시 시작 프로그램은 인스턴스의 종속성이 더는 충족되지 않을 때 서비스 인스턴스를 중지합니다. 인스턴스가 실패하면 다시 시작 프로그램이 인스턴스를 다시 시작하려고 합니다. 인스턴스는 모든 종속성이 충족될 때까지 온라인 상태일 수 없으므로 인스턴스의 종속성을 사용하여 인스턴스의 다시 시작 동작을 결정할 수 있습니다. 각 종속성 선언에 설정된 등록 정보는 종속성이 필요한지 여부와 종속성이 다시 시작되면 인스턴스가 다시 시작될 경우를 정의합니다.

작업 중에 svc.startd 데몬은 적절한 실행 레벨에서 적절한 /etc/rc*.d 스크립트를 시작하고 이는 이전에 init에서 수행된 작업입니다.

다음 예에서는 svc.startd가 network/ipmp:default 서비스 인스턴스에 대한 다시 시작 프로그램을 보여줍니다. 다른 출력은 이 예에서 생략되었습니다.

```
$ svcs -l ipmp:default
restarter    svc:/system/svc/restarter:default
```

restarter 등록 정보가 비어 있거나 `svc:/system/svc/restarter:default`로 설정되면 서비스 인스턴스는 `svc.startd`에서 관리됩니다. `svc.startd` 데몬에 대한 자세한 내용은 [svc.startd\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

위임된 다시 시작 프로그램

일부 서비스의 경우 시작 시 공통적인 일련의 동작이 발생합니다. 위임된 다시 시작 프로그램은 이러한 서비스에 대한 특정 실행 환경 및 응용 프로그램별 다시 시작 동작을 제공할 수 있습니다.

위임된 다시 시작 프로그램의 예로 `inetd`를 들 수 있습니다. 이는 서비스가 항상 실행되도록 하는 것이 아니라 필요에 따라 인터넷 서비스를 시작할 수 있습니다. `inetd` 다시 시작 프로그램은 서비스 인스턴스에 네트워크 연결로 구성된 환경을 입력 및 출력 파일 설명자로 제공합니다. `inetd` 데몬에 대한 자세한 내용은 [inetd\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 다음 예는 `inetd`가 `cups/in-lpd:default` 서비스 인스턴스에 대한 다시 시작 프로그램임을 보여줍니다. 다른 출력은 이 예에서 생략되었습니다.

```
$ svcs -l cups/in-lpd:default
restarter   svc:/network/inetd:default
```

`restarter` 등록 정보에서 지정된 위임된 다시 시작 프로그램은 다시 시작 프로그램이 사용 가능해지면 서비스 인스턴스를 관리합니다.

서비스 등록 정보 및 등록 정보 그룹

종속성, 메소드, 상태, 응용 프로그램 데이터를 포함하여 서비스에 대한 정보는 서비스 구성 저장소에 일련의 등록 정보로 저장됩니다. 등록 정보는 서비스 또는 서비스 인스턴스에서 정의할 수 있습니다. 서비스에 설정된 등록 정보는 해당 서비스의 모든 인스턴스가 상속합니다. 인스턴스에 설정된 등록 정보는 해당 인스턴스에서만 사용됩니다. 서비스 인스턴스는 상속된 등록 정보의 값을 사용자 정의하고 상위 서비스에 대해 정의되지 않은 추가 등록 정보를 정의할 수 있습니다.

등록 정보는 등록 정보 그룹으로 구성됩니다. 다음과 같은 몇 가지 일반 등록 정보 그룹이 있습니다.

- `general` - 인스턴스가 사용으로 설정되는지 등의 정보를 포함합니다.
- `restarter` - 인스턴스의 현재 상태를 포함하여 서비스에 대해 다시 시작 프로그램이 저장한 런타임 정보를 포함합니다.
- `start, refresh, stop` - 서비스를 시작 또는 중지하거나 새로 고치기 위해 실행할 프로그램과 같은 정보를 포함합니다.
- `config` - 서비스 개발자가 응용 프로그램 데이터를 포함하는 데 사용됩니다.

등록 정보 및 등록 정보 그룹에 대한 자세한 내용은 [smf\(5\)](#) 매뉴얼 페이지를 참조하십시오.

서비스 구성 저장소

각 서비스에 대한 정보는 *SMF* 데이터베이스라고도 하는 서비스 구성 저장소에 저장됩니다. 서비스 구성 저장소에는 시스템에 있는 각 서비스 인스턴스의 현재 상태와 각 서비스 및 서비스 인스턴스의 등록 정보 같은 정보가 저장됩니다.

저장소에는 서비스에 대한 *SMF* 런타임 데이터와 영구 구성 정보가 저장됩니다.

- 영구 구성 정보는 데이터 소스에 따라 계층에 저장됩니다. “저장소 계층” [24]을 참조하십시오.
- 런타임 데이터 또는 비영구 구성 정보는 재부트 시 유지되지 않고 저장소에는 비영구 데이터에 대한 계층 정보가 저장되지 않습니다. 비영구 데이터는 일반적으로 활성 프로그램 상태를 포함합니다.

저장소에는 유형, 값, 제약 조건, 등록 정보 설명과 같은 서비스 템플릿 데이터도 저장됩니다. 템플릿 데이터는 서비스 매니페스트에 정의됩니다. 템플릿 데이터에 대한 자세한 내용은 `smf_template(5)` 매뉴얼 페이지를 참조하십시오.

서비스 구성 저장소는 *SMF* 인터페이스를 통해서만 조작하거나 질의할 수 있습니다. `svcs`, `svccprop`, `svccadm`, `svccfg` 명령 또는 `libscf(3LIB)` 매뉴얼 페이지에 나열된 서비스 구성 기능 라이브러리 함수를 사용합니다. 등록 정보 값을 읽고 쓰고 등록 정보 값을 지정된 계층 및 스냅샷에 표시할 수 있습니다. 계층에 대한 자세한 내용은 “저장소 계층” [24]을 참조하십시오. 스냅샷에 대한 자세한 내용은 “저장소 스냅샷” [25]을 참조하십시오. 선택된 서비스 인스턴스나 상위 서비스의 등록 정보만 표시하거나, 등록 정보의 구성된 보기를 표시할 수 있습니다. 구성된 보기에서 상위 서비스에 설정된 등록 정보와 서비스 인스턴스에 설정된 등록 정보가 둘 다 표시됩니다. 표시된 값은 서비스 인스턴스에 설정된 값입니다.

서비스 번들

서비스 번들은 서비스 또는 서비스 인스턴스에 대한 서비스 구성 저장소에 저장된 정보를 포함하는 XML 파일입니다. 서비스 번들에 제공된 정보는 서비스 구성 저장소에 저장되고 저장소에서 내보낼 수 있습니다. 표준 위치의 서비스 번들은 시스템 부트 중에 저장소로 가져옵니다.

두 가지 서비스 번들 유형은 매니페스트와 프로파일입니다.

매니페스트 매니페스트에는 특정 서비스 또는 서비스 인스턴스 세트와 연결된 전체 등록 정보 세트가 포함됩니다.

프로파일 일반적으로 프로파일은 매니페스트에 제공된 정보를 확장하고 대체하는 서비스 또는 서비스 인스턴스의 사용자 정의를 제공합니다. 사용자 정의의 예에는 추가 등록 정보와 변경된 등록 정보 값이 포함됩니다.

매니페스트에 대한 표준 위치는 `/lib/svc/manifest`입니다. 프로파일에 대한 표준 위치는 `/etc/svc/profile`입니다.

시스템이 부트되거나 매니페스트 가져오기 서비스가 다시 시작되면 매니페스트를 가져오고 프로파일이 새롭거나 변경된 경우 해당 프로파일이 적용됩니다. 서비스 번들을 제공하는 IPS 패키지에서 패키지가 설치될 때 매니페스트 가져오기 서비스가 다시 시작되어야 함을 지정할 수 있습니다.

로컬 사용자 정의는 `/etc/svc/profile/site` 디렉토리에서 `.xml` 접미어를 사용하여 프로파일 파일로 제공될 수 있습니다. 여러 매니페스트 또는 프로파일에서 동일한 서비스 또는 인스턴스에 대한 동일한 저장소 계층의 동일한 등록 정보를 정의할 경우 SMF는 사용할 값을 결정할 수 없습니다. 이 유형의 충돌이 감지된 경우 인스턴스는 유지 관리 상태가 됩니다. 계층에 대한 자세한 내용은 [“저장소 계층” \[24\]](#)을 참조하십시오.

서비스를 Oracle Solaris에 제공할 뿐 아니라 서비스 번들은 다양한 시스템에 걸쳐 사용자 정의 구성을 제공할 수 있습니다.

시스템 프로파일 `/etc/svc/profile/generic.xml`은 설치하는 동안 적용됩니다. 이 프로파일을 변경하지 마십시오. 이 시스템 프로파일에 대한 모든 변경 사항은 업그레이드 시 덮어씁니다. 자세한 내용은 [smf_bootstrap\(5\)](#) 매뉴얼 페이지를 참조하십시오.

저장소 계층

서비스 구성 저장소에는 단일 등록 정보에 대한 여러 가지 값을 저장할 수 있습니다. 저장소에서는 데이터가 데이터 소스에 따라 계층에 저장됩니다. 소스는 매니페스트, 시스템 프로파일, 사이트 프로파일, SMF 명령과 라이브러리 인터페이스를 사용하여 생성된 사용자 정의일 수 있습니다. 여러 계층의 값을 보고 실행 중인 구성에서 값의 소스를 이해할 수 있습니다. 즉, 값이 매니페스트 또는 프로파일에 지정되었거나 관리자가 변경했는지를 이해할 수 있습니다.

SMF 명령과 라이브러리 인터페이스를 사용하여 적용된 구성 변경 사항은 `admin` 계층에만 나타납니다. 다른 계층의 구성은 표준 위치의 프로파일 및 매니페스트 파일에 정의됩니다. 파일의 등록 정보가 저장소에 추가될 때 해당 등록 정보에 대한 정보에는 해당 파일의 이름이 포함됩니다.

계층	컨텐츠
admin	SMF 명령 또는 라이브러리 인스턴스를 사용하거나 관리자 또는 응용 프로그램에 의해 적용된 모든 변경 사항. 또한 <code>admin</code> 계층에는 매니페스트를 가져오거나 비표준 위치의 프로파일을 적용하여 적용된 모든 변경 사항이 포함됩니다. 비표준 위치 사용에 대한 주의는 “매니페스트와 프로파일 가져오기 및 적용” [72] 을 참조하십시오.
site-profile	<code>/etc/svc/profile/site</code> 디렉토리에 있는 프로파일이나 레거시 <code>/etc/svc/profile/site.xml</code> 및 <code>/var/svc/profile/site.xml</code> 프로파일의 모든 값. <code>/var/svc/profile</code> 은 표준 위치로서 제거되었으므로 새 프로파일에 사용되지 않아야 합니다.
system-profile	<code>/etc/svc/profile/generic.xml</code> 및 <code>/etc/svc/profile/platform.xml</code> 시스템 프로파일의 모든 값.
manifest	<code>/lib/svc/manifest</code> 및 <code>/var/svc/manifest</code> 디렉토리에 있는 매니페스트의 값. <code>/var/svc/manifest</code> 는 표준 위치로서 제거되었으므로 새 매니페스트에 사용되지 않아야 합니다.

계층 내에서 등록 정보 충돌은 허용되지 않습니다. admin 계층에서 충돌하는 등록 정보는 이전 등록 정보를 덮어씁니다. 같은 등록 정보가 다른 계층의 여러 파일에서 제공되는데 상위 계층에 설정되어 있지 않으면 인스턴스에 충돌 태그가 지정되고 충돌하는 정의가 제거되거나 등록 정보가 상위 계층에 설정될 때까지 인스턴스가 시작되지 않습니다.

보려는 구성 데이터의 계층을 지정하고 이에 따라 관리 사용자 정의에 해당하는 데이터 및 소프트웨어와 함께 제공된 데이터를 확인할 수 있습니다. 클라이언트에서 구성 데이터를 검색할 계층을 지정하지 않으면 맨 위 계층 데이터가 제공됩니다. 맨 위 계층은 맨 위에서 맨 아래까지 admin 계층, site-profile 계층, system-profile 계층, manifest 계층의 우선 순위 순서에 따라 결정됩니다. 등록 정보 값이 admin 계층에 있으면 이 값은 저장소에서 제공하는 값입니다. 이 방법에서는 로컬 사용자 정의가 시스템이 설치될 때 제공된 값보다 우선 적용됩니다.

저장소 스냅샷

저장소에서는 서비스가 성공적으로 시작될 때마다 각 서비스의 읽기 전용 스냅샷을 캡처합니다. 이러한 스냅샷을 사용하여 필요한 경우 간편하게 이전 작업 상태로 돌아갈 수 있습니다. 다음 스냅샷은 제공된 모든 인스턴스에 사용할 수 있습니다.

initial	서비스와 해당 인스턴스를 처음 가져왔을 때 초기 구성. 매니페스트 가져오기를 수행하기 전에 프로파일이 서비스나 인스턴스를 시작하면 initial 스냅샷이 생성되지 않습니다.
previous	이미 제공된 서비스에 대해 매니페스트 가져오기가 수행될 때 캡처된 현재 구성. 서비스가 이미 가져오고 있는 매니페스트나 다른 매니페스트에 의해 제공되었을 수 있습니다.
running	서비스 인스턴스의 실행 중인 구성. 구성 데이터를 변경할 때 <code>svcadm refresh</code> 또는 <code>svccfg refresh</code> 명령을 사용하여 새 값을 실행 중인 스냅샷으로 승격합니다.
start	online 상태로 성공적으로 전환되는 동안 캡처된 구성.

저장소 백업

SMF는 다음과 같이 서비스 구성 저장소의 백업을 자동으로 수행합니다.

- 각 시스템 시작 시 처음으로 저장소를 변경하기 직전에 boot 백업이 수행됩니다.
- 서비스가 새 매니페스트를 가져왔거나 업그레이드 스크립트를 실행한 경우 `svc:/system/early-manifest-import:default` 또는 `svc:/system/manifest-import:default`가 완료되기 전 `manifest_import` 백업이 발생합니다.

각 유형의 네 가지 백업은 시스템에서 유지 관리되고 필요에 따라 가장 오래된 백업이 삭제됩니다.

이러한 백업의 하나에서 저장소를 복원할 수 있습니다. [백업에서 저장소를 복원하는 방법 \[101\]](#)을 참조하십시오.

구성 파일 및 SMF 서비스

SMF는 응용 프로그램을 시작하는 데 사용할 권장 방식입니다. 대부분 경우에 구성 파일 대신 SMF가 서비스 관리에 사용됩니다. 이 섹션에서는 몇 가지 일반적인 레거시 구성 스크립트 및 파일을 처리하는 방법을 설명합니다.

/etc/rc?.d 스크립트

?가 실행 레벨을 나타내는 /etc/rc?.d 디렉토리에는 실행 레벨 전환 시 실행되는 서비스를 관리하기 위한 레거시 초기화 및 종료 스크립트가 포함됩니다. 이전에 /etc/rc?.d 스크립트에 의해 구현된 대부분 서비스는 SMF에서 관리됩니다. 일부 /etc/rc?.d 스크립트는 이러한 서비스를 /etc/rc*.d 스크립트로 예상하는 타사 응용 프로그램을 사용할 수 있도록 유지됩니다. 이러한 스크립트는 /etc/init.d 디렉토리의 파일에 하드 링크됩니다. /etc/rc?.d 스크립트 및 실행 레벨에 대한 자세한 내용은 /etc/init.d/README 파일, /etc/rc?.d 디렉토리의 README 파일 및 [inittab\(4\)](#) 매뉴얼 페이지를 참조하십시오. 실행 제어 스크립트를 변환하는 방법에 대한 자세한 내용은 [실행 제어 스크립트를 SMF 서비스로 변환하는 방법 \[88\]](#)을 참조하십시오. rc?.d 스크립트를 변환한 후 스크립트 이름을 *sscript*에서 *sscript*로 바꾸면 스크립트가 효과적으로 제거됩니다.

/etc/init.d 스크립트

/etc/init.d 디렉토리에는 init 상태를 변경하기 위한 초기화 및 종료 스크립트가 포함됩니다. 일부 이러한 스크립트는 /etc/rc?.d 디렉토리의 스크립트에 하드 링크됩니다. /etc/init.d 스크립트에 대한 자세한 내용은 /etc/init.d/README 및 the [init.d\(4\)](#) 매뉴얼 페이지를 참조하십시오.

레거시 init.d 실행 제어 스크립트는 svc가 아니라 lrc로 시작되는 SMF FMRI로 표시됩니다. 예를 들어 /etc/rc2.d/S47pppd PPP 구성 스크립트는 lrc:/etc/rc2_d/S47pppd 서비스로 표시됩니다. 이러한 lrc 서비스의 상태는 legacy_run입니다. 다음 예와 같이 레거시 서비스의 이름과 시작 시간을 나열할 수 있지만 SMF를 사용하여 이러한 서비스를 관리할 수는 없습니다.

```
$ svcs lrc:\*
STATE          STIME          FMRI
legacy_run     9:34:54       lrc:/etc/rc2_d/S47pppd
legacy_run     9:34:54       lrc:/etc/rc2_d/S89PRESERVE
$ svcs -l lrc:/etc/rc2_d/S47pppd
svcs: Operation not supported for legacy service 'lrc:/etc/rc2_d/S47pppd'
$ svccfg -s lrc:/etc/rc2_d/S47pppd listprop
svccfg: Operation not supported for legacy service 'lrc:/etc/rc2_d/S47pppd'
```

/etc/inittab 항목

/etc/inittab 파일의 항목은 init로 전달하는 프로세스를 제어합니다. /etc/inittab 파일을 직접 편집하지 마십시오. SMF 서비스를 수정하십시오. ttymon에 전달된 매개변수

를 수정하는 방법에 대한 예는 [ttymon 등록 정보 값을 수정하는 방법 \[64\]](#)을 참조하십시오.

/etc/inittab 파일 항목의 형식에 대한 자세한 내용은 [inittab\(4\)](#) 매뉴얼 페이지를 참조하십시오. 실행 레벨에 대한 자세한 내용은 [inittab\(4\)](#) 매뉴얼 페이지 및 /etc/init.d/README를 참조하십시오.

/etc/inetd.conf 파일

이전에 [inetd.conf](#) 파일을 사용하여 구성된 서비스는 이제 SMF를 사용하여 구성됩니다. [inetd.conf](#) 파일의 구성을 사용할 수 있으려면 SMF 서비스로 변환해야 합니다. [“inetd 서비스를 SMF 서비스로 변환” \[107\]](#)을 참조하십시오. 이미 SMF 서비스로 변환된 [inetd](#) 서비스에 대해서는 [“inetd로 제어되는 서비스 수정” \[75\]](#)을 참조하십시오.

/etc/nscd.conf 파일

/etc/nsswitch.conf 파일

/etc/resolv.conf 파일

이러한 파일을 편집하지 마십시오. 편집이 손실됩니다. 이러한 파일은 파일을 구문 분석할 수 있는 응용 프로그램에 대한 이전 버전과의 호환성을 위해 SMF 데이터에서 자동으로 생성됩니다. `svccfg setprop` 명령을 사용하여 [“등록 정보 값 설정” \[62\]](#)에 표시된 대로 등록 정보 값을 수정합니다.

[nscd.conf](#) 파일의 기능은 `svc:/system/name-service-cache` SMF 서비스로 대체되었습니다. [nscd.conf](#) 파일을 편집하는 대신 구성할 `name-service-cache` 등록 정보를 확인하려면 [nscd.conf\(4\)](#) 매뉴얼 페이지를 참조하십시오.

[nsswitch.conf](#) 파일의 기능은 `svc:/system/name-service/switch` SMF 서비스로 대체되었습니다. [nsswitch.conf](#) 파일을 편집하는 대신 구성할 `name-service/switch` 등록 정보를 확인하려면 [nsswitch.conf\(4\)](#) 매뉴얼 페이지를 참조하십시오.

[resolv.conf](#) 파일의 기능은 `svc:/network/dns/client` SMF 서비스로 대체되었습니다. [resolv.conf](#) 파일을 편집하는 대신 구성할 `dns/client` 등록 정보를 확인하려면 [resolv.conf\(4\)](#) 매뉴얼 페이지를 참조하십시오.

이러한 파일은 편집하지 않아야 하는 구성 파일의 예입니다. 이러한 다른 파일이 존재합니다. [“파일에 의해 구성되는 서비스 수정” \[76\]](#)에 설명된 대로 경우에 따라 구성 파일 편집이 구성을 수정하는 올바른 방법입니다. 구성 파일을 편집하기 전에 파일의 모든 주석과 모든 관련 매뉴얼 페이지를 읽고 파일 편집이 관련 서비스에 대한 구성을 수정하는 올바른 방법인지 확인합니다.

서비스 관리 권한

서비스 상태와 구성을 수정하려면 높은 수준의 권한이 필요합니다. 다음 방법의 하나를 사용하여 필요한 권한을 얻습니다. 필요한 프로파일이나 역할을 결정하는 방법 및 권한을 지정하는 방법을 포함하여 권한 부여, 프로파일 및 역할에 대한 자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)을 참조하십시오.

권한 부여

SMF 작업에 필요한 권한 부여에 대한 자세한 내용은 [smf_security\(5\)](#) 매뉴얼 페이지를 참조하십시오. `action_authorization`, `modify_authorization`, `read_authorization` 및 `value_authorization`과 같은 등록 정보에 대한 특정 서비스를 검사할 수도 있습니다.

권한 프로파일

`profiles` 명령을 사용하여 지정 받은 권한 프로파일을 나열합니다. 서비스 관리 권한 프로파일은 `solaris.smf.manage` 및 `solaris.smf.modify` 권한을 부여하고 이 프로파일을 통해 `svcadm` 및 `svccfg` 명령을 사용할 수 있습니다. 서비스 운영자 권한 프로파일은 `solaris.smf.manage` 및 `solaris.smf.modify.framework` 권한을 부여합니다.

역할

`roles` 명령을 사용하여 지정 받은 역할을 나열합니다. `root` 역할이 있는 경우 `su` 명령을 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 2

서비스에 대한 정보 가져오기

이 장에서는 다음과 같은 서비스에 대한 정보를 가져오는 방법을 보여줍니다.

- 서비스 상태, 종속성 및 기타 등록 정보 값
- 계약 서비스에서 시작한 프로세스
- 문제 해결을 위한 로그 파일 위치
- FMA 이벤트 및 서비스 전환 이벤트 알림 설정

시스템의 서비스 나열

`svcs` 명령은 서비스 인스턴스 상태에 대한 주 명령입니다.

서비스 상태 표시

이러한 예에 표시된 상태의 설명은 “서비스 상태” [20]를 참조하십시오.

예 2-1 모든 사용으로 설정된 서비스 나열

옵션이나 인수를 사용하지 않으면 `svcs` 명령은 이 시스템에서 사용으로 설정된 모든 서비스 인스턴스와 일시적으로 사용 안함으로 설정된 인스턴스를 나열합니다.

이 목록에서 `disabled` 상태의 서비스 인스턴스는 다음 시스템 부트 시 사용으로 설정됩니다. `legacy_run` 상태의 인스턴스는 SMF에서 관리되지 않습니다. 이러한 레거시 서비스에 대한 자세한 내용은 “구성 파일 및 SMF 서비스” [26]를 참조하십시오. `maintenance`, `degraded` 또는 `offline` 상태의 서비스가 있으면 “서비스 상태에 대한 추가 정보 가져오기” [36]를 참조하십시오.

STIME 열에는 인스턴스가 나열된 상태로 전환된 시간이 표시됩니다. 인스턴스가 이 상태로 전환된 후 24시간이 경과하면 STIME 열에 날짜가 표시됩니다.

```
$ svcs
STATE          STIME          FMRI
```

```

legacy_run   Sep_09   lrc:/etc/rc2_d/S47pppd
legacy_run   Sep_09   lrc:/etc/rc2_d/S81dodatadm_udaplt
legacy_run   Sep_09   lrc:/etc/rc2_d/S89PRESERVE
disabled     Sep_09   svc:/system/vbiosd:default
online       Sep_09   svc:/system/early-manifest-import:default
online       Sep_09   svc:/system/svc/restarter:default
...

```

예 2-2 모든 설치된 서비스 나열

다음 부트 시 자동으로 사용으로 설정되지 않을 disabled 인스턴스를 포함하여 이 시스템에 설치된 모든 서비스 인스턴스를 나열하려면 `svcs -a` 명령을 사용합니다.

```
$ svcs -a
```

나열된 상태에서 다른 상태로 전환 중인 서비스 인스턴스의 상태에는 별표(*)가 추가됩니다. 예를 들어 `offline*`는 인스턴스에서 시작 메소드를 아직 실행 중임을 의미할 수 있습니다.

상태가 없거나 인식되지 않으면 물음표(?)가 표시됩니다.

예 2-3 모든 서비스 인스턴스 나열

서비스 이름을 지정하면 `svcs` 명령이 모든 서비스 인스턴스를 나열합니다. `-o` 옵션에 대한 자세한 내용은 “[선택한 서비스 정보 표시](#)” [31]를 참조하십시오.

```
$ svcs -Ho inst identity
node
domain
```

서비스에 대한 추가 정보 표시

`svcs -l` 명령은 인스턴스 상태, 인스턴스에 대한 로그 파일 및 구성 파일 경로, 종속성 유형, 종속성 다시 시작 속성 값, 종속성 상태에 대한 자세한 정보가 포함된 각 지정된 서비스 인스턴스에 대한 긴 목록을 보여줍니다. 다음 예에서는 이 서비스 인스턴스의 모든 필수 종속성이 온라인 상태임을 보여줍니다. 사용 안함으로 설정된 하나의 종속성은 선택적 종속성입니다. 종속성 유형 및 재시작 속성 값에 대한 자세한 내용은 “[서비스 종속성 표시](#)” [32]를 참조하십시오. `svcs -l` 출력에서 “[서비스 상태](#)” [20]에 설명된 상태 이외의 상태는 종속성에 대해 가능합니다. 자세한 내용은 `svcs(1)` 매뉴얼 페이지를 참조하십시오. 또한 다음 예에서는 지정된 서비스 인스턴스가 일시적으로 사용으로 설정되고 온라인 상태이며 서비스가 계약 유형 서비스임을 보여줍니다. 서비스 유형에 대한 설명은 “[서비스 모델](#)” [19]을 참조하십시오. 상태 값 뒤에 별표가 표시되어 있으면(예: `offline*`) 인스턴스가 전환 중이고 다음 상태 필드에는 `none` 대신 상태 값이 표시됩니다. `state_time`은 인스턴스가 나열된 상태로 전환된 시간입니다.

```
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
```

```

enabled      true (temporary)
state        online
next_state   none
state_time   September 17, 2013 05:57:26 PM PDT
logfile      /var/svc/log/application-management-net-snmp:default.log
restarter    svc:/system/svc/restarter:default
contract_id  160
manifest     /etc/svc/profile/generic.xml
manifest     /lib/svc/manifest/application/management/net-snmp.xml
dependency   require_all/none svc:/system/filesystem/local (online)
dependency   optional_all/none svc:/milestone/name-services (online)
dependency   optional_all/none svc:/system/system-log (online)
dependency   optional_all/none svc:/network/rpc/rstat (disabled)
dependency   require_all/restart svc:/system/cryptosvc (online)
dependency   require_all/restart svc:/milestone/network (online)
dependency   require_all/refresh file://localhost/etc/net-snmp/snmp/snmpd.conf (online)
dependency   require_all/none svc:/milestone/multi-user (online)

```

예 2-4 계약 서비스에서 시작한 프로세스 표시

svcs -p 명령을 사용하여 계약 서비스 인스턴스에서 시작한 프로세스의 프로세스 ID 및 명령 이름을 표시합니다. net-snmp 서비스는 MIB(Management Information Base) 세트를 통해 시스템에 대한 정보를 수집하는 /usr/sbin/snmpd SNMP 에이전트를 관리합니다.

```

$ svcs -p net-snmp
STATE      STIME      FMRI
online     17:57:26   svc:/application/management/net-snmp:default
           17:57:26   5022 snmpd

```

예 2-5 프로세스가 중지한 후 자동으로 다시 시작되는 계약 서비스 표시

계약이 비어 있으면 계약 서비스 인스턴스가 자동으로 다시 시작됩니다. 또한 SMF는 하드웨어 또는 소프트웨어 오류 이벤트에서 자동으로 복구하는 과정에서 계약 서비스 인스턴스와 연결된 프로세스를 다시 시작하려고 합니다. 다음 예에서는 /usr/sbin/snmpd 프로세스가 종료된 후 새 프로세스 ID를 사용하여 자동으로 다시 시작됨을 보여줍니다. net-snmp:default 인스턴스는 계속 온라인 상태이고 새 시작 시간이 포함됩니다.

```

$ kill 5022
$ svcs -p net-snmp
STATE      STIME      FMRI
online     17:57:59   svc:/application/management/net-snmp:default
           17:57:59   5037 snmpd

```

선택한 서비스 정보 표시

svcs 명령의 출력은 다른 명령에 파이프하거나 스크립트에서 사용할 때 매우 유용할 수 있습니다. svcs 명령의 -o 옵션을 사용하여 원하는 정보 열 및 열 순서를 지정할 수 있습니다. 예를 들어 서비스 이름과 인스턴스 이름을 개별 열에 출력하고 서비스의 현재 상태와 다음 상태

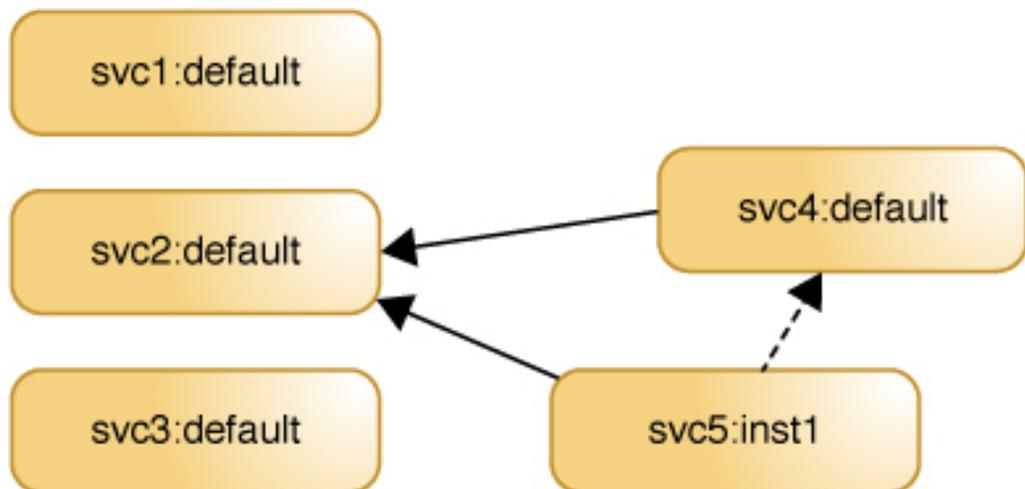
및 계약 ID를 출력할 수 있습니다. -s 및 -s 옵션을 사용하여 하나 이상의 열에 대한 출력의 정렬 순서를 지정할 수 있습니다.

서비스 종속성 표시

종속성 관계는 서비스 인스턴스 상태 전환을 관리합니다. 종속성에 대한 자세한 내용은 “[서비스 종속성](#)” [20]을 참조하십시오. 자세한 설명과 다양한 종속성을 지정하는 방법은 [5장. SMF를 사용하여 응용 프로그램 제어](#)를 참조하십시오.

다음 그림에서 svc1:default, svc2:default 및 svc3:default 서비스 인스턴스에서는 다른 서비스나 파일이 필요하지 않고 다른 리소스를 시작할 필요가 없습니다. 이러한 인스턴스는 병렬로 시작되고 해당하는 시작 메소드를 실행하며 다른 리소스에서 대기하지 않고 온라인 상태로 이동할 수 있습니다. svc4:default 인스턴스는 svc2:default 인스턴스가 온라인 상태가 될 때까지 시작 메소드를 실행할 수 없습니다. svc5:inst1 인스턴스에는 svc2:default 리소스와 svc4:default 리소스가 둘 다 필요합니다. svc5:inst1이 svc4:default에 대해 가지는 종속성은 선택적 종속성이고 svc4:default가 enabled/online 또는 disabled 상태이거나 존재하지 않는 경우 충족됩니다. svc5:inst1 인스턴스는 svc2:default가 온라인 상태가 될 때까지 기다려야 하고, svc4:default가 존재하고 사용으로 설정된 경우 svc5:inst1은 svc4:default가 온라인 상태가 될 때까지 기다려야 합니다. svc4:default가 존재하고 사용 안함으로 설정되거나 존재하지 않는 경우에는 svc5:inst1이 svc4:default를 기다릴 필요가 없습니다.

그림 2-1 서비스 종속성 관계



종속성 그룹

각 종속성은 다음 그룹의 하나에 지정됩니다. 그룹은 해당 그룹의 종속성이 충족되는 방식을 정의합니다.

<code>require_all</code>	<p>이 종속성은 다음 조건이 둘 다 충족될 때 충족됩니다.</p> <ul style="list-style-type: none"> ■ 이 그룹의 모든 서비스 종속성은 <code>online</code> 또는 <code>degraded</code> 상태로 실행 중입니다. ■ 이 그룹의 모든 파일 종속성이 존재합니다.
<code>require_any</code>	<p>이 종속성은 다음 조건의 하나가 충족될 때 충족됩니다.</p> <ul style="list-style-type: none"> ■ 이 그룹의 서비스 종속성 중 하나 이상이 <code>online</code> 또는 <code>degraded</code> 상태로 실행 중입니다. ■ 이 그룹의 파일 종속성 중 하나 이상이 존재합니다.
<code>optional_all</code>	<p>이 종속성은 이 그룹의 모든 서비스 종속성이 다음 조건을 충족할 때 충족됩니다.</p> <ul style="list-style-type: none"> ■ 서비스 인스턴스가 <code>online</code> 또는 <code>degraded</code> 상태로 실행 중입니다. ■ 서비스에서 관리 작업이 실행되도록 요청합니다. 서비스가 존재하지 않거나, 완료되지 않았거나, <code>disabled</code> 또는 <code>maintenance</code> 상태이거나, <code>offline</code> 상태로 관리 작업이 시작되도록 요청하는 종속성을 기다리고 있습니다. <p>이 그룹의 파일 종속성이 존재하거나 존재하지 않을 수 있습니다. 서비스 인스턴스가 전환 중이고 관리 작업이 시작되도록 요청하지 않으면 이 종속성이 충족되지 않습니다. 이 경우 종속 서비스는 이 종속성이 시작되기를 기다리거나 종속성이 관리 작업 없이 시작될 수 있는지 확인하기를 기다립니다.</p>
<code>exclude_all</code>	<p>이 종속성은 다음 조건이 둘 다 충족될 때 충족됩니다.</p> <ul style="list-style-type: none"> ■ 이 그룹의 모든 서비스 종속성이 <code>disabled</code> 또는 <code>maintenance</code> 상태이거나 존재하지 않습니다. ■ 이 그룹의 모든 파일 종속성이 존재하지는 않습니다.

서비스가 종속된 인스턴스 나열

`svcs -d` 명령은 지정된 서비스가 종속된 서비스 인스턴스를 나열합니다.

이 예에서는 `system-repository` 서비스가 종속된 서비스 인스턴스를 보여줍니다.

```
$ svcs -d system-repository
```

```
STATE      STIME      FMRI
online     Sep_09     svc:/milestone/network:default
online     Sep_09     svc:/system/filesystem/local:default
online     Sep_09     svc:/system/filesystem/autofs:default
```

svcs -l 명령은 지정된 서비스가 종속된 서비스도 나열합니다. 종속성의 이름과 상태 이외에 -l 옵션 출력에는 종속성의 유형이나 그룹 및 restart_on 등록 정보 값이 표시됩니다. 이 예에서 종속성의 두 개는 필수이고 하나는 선택 사항입니다. 이러한 그룹의 종속성이 종속 서비스에 영향을 미치는 방식에 대한 자세한 내용은 “종속성 그룹” [33]을 참조하십시오. 종속성의 restart_on 등록 정보에 대한 여러 가지 값이 종속 서비스에 영향을 미치는 방식에 대한 자세한 내용은 “종속성 그룹” [33]을 참조하십시오.

```
$ svcs -l system-repository
fmri      svc:/application/pkg/system-repository:default
name      IPS System Repository
enabled   false
state     disabled
next_state none
state_time Mon Sep 09 18:42:28 2013
restarter svc:/system/svc/restarter:default
manifest  /lib/svc/manifest/application/pkg/pkg-system-repository.xml
dependency require_all/error svc:/milestone/network:default (online)
dependency require_all/none svc:/system/filesystem/local:default (online)
dependency optional_all/error svc:/system/filesystem/autofs:default (online)
```

svccprop 명령을 사용하여 이러한 종속성을 나열할 수도 있습니다. 이 양식에는 종속성의 그룹 및 restart_on 값이 개별 행에 표시되고 종속성 상태는 표시되지 않습니다.

```
$ svccprop -g dependency system-repository:default
network/entities fmri svc:/milestone/network:default
network/grouping astring require_all
network/restart_on astring error
network/type astring service
filesystem-local/entities fmri svc:/system/filesystem/local:default
filesystem-local/grouping astring require_all
filesystem-local/restart_on astring none
filesystem-local/type astring service
autofs/entities fmri svc:/system/filesystem/autofs:default
autofs/grouping astring optional_all
autofs/restart_on astring error
autofs/type astring service
```

서비스에 종속된 인스턴스 나열

svcs -D 명령은 지정된 서비스에 종속된 서비스 인스턴스를 나열합니다.

이 예에서는 system-repository 서비스에 종속된 서비스 인스턴스를 보여줍니다.

```
$ svcs -D system-repository
```

```
STATE          STIME      FMRI
online         16:39:30  svc:/application/pkg/zones-proxyd:default
```

다음 명령은 zones-proxyd가 system-repository에 종속됨을 확인합니다.

```
$ svcs -do svc,desc zones-proxyd
SVC          DESC
application/pkg/system-repository  IPS System Repository
system/filesystem/minimal          minimal file system mounts
milestone/network                   Network milestone
```

다음 명령은 zones-proxyd가 system-repository에 종속되는 방식에 대한 추가 정보를 표시합니다. 이 출력의 마지막 행은 zones-proxyd 서비스에서 system-repository 서비스를 실행 중이도록 요청함을 보여주고 system-repository가 현재 실행 중임을 보여줍니다. 또한 이 출력에는 system-repository 서비스를 새로 고치면 zones-proxyd가 다시 시작될 것임을 보여줍니다.

```
$ svcs -l zones-proxyd
fmri          svc:/application/pkg/zones-proxyd:default
name          Zones Proxy Daemon
enabled       true
state         online
next_state    none
state_time    January 6, 2014 04:39:30 PM PST
restarter     svc:/system/svc/restarter:default
manifest      /lib/svc/manifest/application/pkg/zoneproxyd.xml
dependency    require_any/none svc:/system/filesystem/minimal (online)
dependency    require_any/error svc:/milestone/network (online)
dependency    require_all/restart svc:/application/pkg/system-repository (online)
```

서비스를 자동으로 시작할지 여부 표시

실행 중인 서비스는 종속성의 하나가 중지되거나 새로 고쳐지면 다시 시작되도록 구성할 수 있습니다. 실행 중인 서비스(online 또는 degraded 상태)의 종속성이 충족되지 않으면 서비스가 offline 상태로 전환됩니다. 종속성이 중지되거나 새로 고쳐진 후 서비스가 다시 시작되면 종속성이 다시 충족되고 종속 서비스가 다시 실행 중인 상태로 전환될 수 있습니다.

다음 요소에 따라 require_all, require_any 또는 optional_all 종속성이 중지되거나 새로 고쳐진 후 서비스를 다시 시작할지가 결정됩니다.

- 종속성이 중지되거나 새로 고쳐졌는지를 나타냅니다. 중지된 경우 하드웨어 오류나 코어 덤프와 같은 오류 또는 관리 작업과 같은 다른 이유 때문에 종속성이 중지되었는지를 나타냅니다.
- 종속성의 restart_on 속성 값입니다. 가능한 값은 none, error, restart 및 refresh입니다.

다음 표에 나와 있는 대로 종속성의 restart_on 속성 값이 none이면 종속성이 중지되거나 새로 고쳐질 때 종속 서비스가 다시 시작되지 않습니다. 종속성의 restart_on 속성 값이

refresh이면 종속성이 중지되거나 새로 고쳐질 때 종속 서비스가 항상 다시 시작됩니다. restart_on 값이 error이면 오류 때문에 종속성이 중지된 경우에만 종속 서비스가 다시 시작됩니다. restart_on 값이 restart이면 종속성이 새로 고쳐진 경우에만 종속 서비스가 다시 시작됩니다.

표 2-1 종속성이 중지된 후 자동으로 서비스 다시 시작

require_all, require_any 또는 optional_all 종속성 중지 또는 새로 고침 이벤트	종속성 restart_on 속성 값			
	none	error	restart	refresh
오류 때문에 중지	다시 시작되지 않음	다시 시작	다시 시작되지 않음	다시 시작
기타 중지	다시 시작되지 않음	다시 시작되지 않음	다시 시작되지 않음	다시 시작
새로 고침	다시 시작되지 않음	다시 시작되지 않음	다시 시작	다시 시작

“서비스가 종속된 인스턴스 나열” [33]은 system-repository 서비스에 두 개의 require_all 종속성과 하나의 optional_all 종속성이 있음을 보여줍니다. 다음 명령은 milestone/network 서비스나 system/filesystem/autofs 서비스가 어떤 다른 이유로 중지되거나 새로 고쳐지는 경우가 아니라 오류 때문에 중지되면 system-repository 서비스가 다시 시작될 것임을 보여줍니다. system/filesystem/local 서비스가 어떤 이유로 새로 고쳐지거나 중지되면 system-repository 서비스가 다시 시작되지 않습니다.

```
$ svccfg -s system-repository:default listprop -o propName,propval '**restart_on'
```

```
network/restart_on      astring      error
filesystem-local/restart_on astring      none
autofs/restart_on       astring      error
```

서비스 상태에 대한 추가 정보 가져오기

인수가 없으면 svcs -x 명령은 다음 서비스 인스턴스에 대한 설명 정보를 제공합니다.

- 사용으로 설정되지만 실행 중이 아닌 인스턴스입니다.
- 다른 사용으로 설정된 서비스가 실행되지 않게 차단하는 인스턴스입니다.

모든 사용으로 설정된 서비스가 실행 중이면 svcs -x 명령은 출력을 생성하지 않습니다.

시작 메소드가 실패했기 때문에 다음 예에서 pkg/depot 서비스는 maintenance 상태입니다.

```
$ svcs -x
```

```
svc:/application/pkg/depot:default (IPS Depot)
  State: maintenance since September 11, 2013 01:30:42 PM PDT
  Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
  See: http://support.oracle.com/msg/SMF-8000-KS
  See: pkg.depot-config(1M)
  See: /var/svc/log/application-pkg-depot:default.log
```

Impact: This service is not running.

출력에서는 시작 메소드가 실패한 이유를 확인하기 위해 My Oracle Support에서 제공되는 예측적 자가 치유 기술 자료 문서, 매뉴얼 페이지 및 로그 파일을 참조하도록 제안합니다. 로그 파일을 보는 다양한 방법에 대한 자세한 내용은 “서비스 로그 파일 보기” [37]를 참조하십시오. maintenance 상태인 서비스를 수정하는 방법에 대한 자세한 내용은 “성능 저하, 오프라인 또는 유지 관리 상태인 인스턴스 복구” [97]를 참조하십시오.

다음 예에서 print/server 서비스에는 실행되고 있지 않은 종속 서비스가 있습니다. print/server 서비스가 사용 안함으로 설정되었으므로 종속 서비스를 실행할 수 없습니다.

```
$ svcs -x
svc:/application/print/server:default (LP print server)
  State: disabled since Fri Mar 08 14:42:32 2013
  Reason: Disabled by an administrator.
    See: http://sun.com/msg/SMF-8000-05
    See: lpsched(1M)
  Impact: 2 dependent services are not running. (Use -v for list.)
$ svcs -xv
svc:/application/print/server:default (LP print server)
  State: disabled since Fri Mar 08 14:42:32 2013
  Reason: Disabled by an administrator.
    See: http://sun.com/msg/SMF-8000-05
    See: man -M /usr/share/man -s 1M lpsched
  Impact: 2 dependent services are not running:
    svc:/application/print/rfc1179:default
    svc:/application/print/ipp-listener:default
$ svcs -D print/server
STATE      STIME      FMRI
online     Mar_08     svc:/milestone/multi-user:default
offline    Mar_08     svc:/application/print/ipp-listener:default
offline    Mar_08     svc:/application/print/rfc1179:default
```

svcs -x 명령에 제공된 인수가 이 섹션의 시작 부분에서 언급된 기준을 충족하지 않으면 출력에는 인스턴스 상태에 대한 이유가 아니라 추가 정보에 대한 리소스가 표시됩니다.

```
$ svcs -x smb
svc:/network/smb:default (SMB properties)
  State: online since Thu Sep 12 19:16:56 2013
  See: smb(4)
  See: /var/svc/log/network-smb:default.log
  Impact: None.
```

서비스 로그 파일 보기

SMF는 중요한 다시 시작 프로그램 작업, 메소드 표준 출력, 표준 오류 출력에 대한 정보를 각 서비스 인스턴스에 대한 `/var/svc/log/service:instance.log`에 기록합니다. 로그 파일 이름에서 `service` 이름의 슬래시가 하이픈으로 대체됩니다. svcs 명령을 `-L`, `-l` 또는 `-x` 옵션

과 함께 사용하면 지정된 서비스 인스턴스에 대한 로그 파일의 전체 경로 이름이 표시됩니다. `svcs -xL` 명령은 로그 파일의 마지막 몇 행을 표시하고 `svcs -Lv` 명령을 사용하여 전체 로그 파일을 보도록 알립니다. `svcs -Lv` 명령은 매우 길 수 있는 전체 파일을 표시합니다. 예를 들어 편집기에서 로그 파일을 보거나 마지막 n 개 항목을 보려면 `svcs -L` 명령의 출력을 실행합니다.

다음 예는 로그 파일을 사용하여 `svcs` 목록에 표시된 서비스가 일시적으로 사용 안함으로 설정된 이유를 조사하는 방법을 보여줍니다.

```
$ svcs
STATE      STIME      FMRI
legacy_run Sep_09     lrc:/etc/rc2_d/S47pppd
legacy_run Sep_09     lrc:/etc/rc2_d/S81dodatadm_udadpt
legacy_run Sep_09     lrc:/etc/rc2_d/S89PRESERVE
disabled   Sep_09     svc:/system/vbiosd:default
online     Sep_09     svc:/system/early-manifest-import:default
online     Sep_09     svc:/system/svc/restarter:default
...
$ svcs -x vbiosd
svc:/system/vbiosd:default (BIOS call emulation)
  State: disabled since Mon Sep  9 18:42:37 2013
Reason: Temporarily disabled by service method: "vbiosd is not supported on UEFI systems."
  See: http://support.oracle.com/msg/SMF-8000-15
  See: vbiosd(1M)
  See: /var/svc/log/system-vbiosd:default.log
Impact: This service is not running.
$ svcs -xL vbiosd
svc:/system/vbiosd:default (BIOS call emulation)
  State: disabled since Mon Sep  9 18:42:37 2013
Reason: Temporarily disabled by service method: "vbiosd is not supported on UEFI systems."
  See: http://support.oracle.com/msg/SMF-8000-15
  See: vbiosd(1M)
  See: /var/svc/log/system-vbiosd:default.log
Impact: This service is not running.
  Log:
[ Sep  9 18:42:27 Enabled. ]
[ Sep  9 18:42:37 Executing start method ("/lib/svc/method/svc-vbiosd start"). ]
[ Sep  9 18:42:37 Method "start" exited with status 101. ]
[ Sep  9 18:42:37 "start" method requested temporary disable: "vbiosd is not supported on UEFI systems"
]

  Use: 'svcs -Lv svc:/system/vbiosd:default' to view the complete log.
$ svcs -L vbiosd
/var/svc/log/system-vbiosd:default.log
$ view `svcs -L vbiosd`
```

유용하다고 볼 수 있는 기타 로그 파일에는 마스터 다시 시작 프로그램 데몬에 대한 로그와 시스템 로그가 포함됩니다. 로그 파일 이름을 확인하고 `svc.startd` 다시 시작 프로그램 데몬에 대한 로그 파일을 보려면 서비스 이름 `restarter`를 `svcs` 명령과 함께 사용합니다. `syslogd` 시스템 로그 데몬에 대한 로그 파일을 보려면 서비스 이름 `system-log`를 사용합니다.

시스템 부트 시 표시되는 메시징 양을 변경하는 방법에 대한 자세한 내용은 “[시작 메시징 양 지정](#)” [103]을 참조하십시오. 특정 서비스 상태로 전환되거나 해당 상태에서 벗어날 때 또는 FMA 이벤트가 발생할 때 사용자에게 알리게 서비스를 구성하는 방법에 대한 자세한 내용은 “[상태 전환 및 FMA 이벤트에 대한 알림 구성](#)” [57]을 참조하십시오.

서비스 구성 검사

서비스 구성은 서비스와 서비스 인스턴스에서 설정되고 서비스 구성 저장소의 계층에 저장되는 등록 정보로 표현됩니다. 서비스에 설정된 등록 정보는 해당 서비스의 모든 인스턴스가 상속합니다. 인스턴스에 설정된 등록 정보는 해당 인스턴스에서만 사용됩니다. 서비스 인스턴스는 상속된 등록 정보의 값을 사용자 정의하고 상위 서비스에 대해 정의되지 않은 추가 등록 정보를 정의할 수 있습니다.

이 섹션에서는 등록 정보 값을 검색하는 방법과 값이 서비스에 대해 전역인지, 인스턴스에 특정인지, 소프트웨어와 함께 제공되었는지 또는 관리 사용자 정의인지 확인하는 방법을 보여줍니다.

등록 정보 및 등록 정보 그룹의 설명 표시

`svccfg describe` 명령은 현재 등록 정보 값을 포함하여 서비스의 등록 정보 그룹 및 등록 정보에 대한 설명을 표시합니다. 피연산자가 없으면 `describe`는 선택된 서비스나 서비스 인스턴스의 모든 등록 정보 그룹과 등록 정보에 대한 설명을 표시합니다. `-v` 옵션을 사용하여 현재 값의 설명 및 가능한 값 목록을 포함한 추가 정보를 표시합니다. `-t` 옵션을 사용하여 템플릿 정보를 표시합니다.

```
$ svccfg -s pkg/server describe network/restart_on
network/restart_on astring    none
    Determines whether to restart the service due to a dependency refresh, restart, or failure.
$ svccfg -s pkg/server describe -v network/restart_on
network/restart_on astring    none
    type: astring
    required: true
    Determines whether to restart the service due to a dependency refresh, restart, or failure.
    visibility: readwrite
    minimum number of values: 1
    maximum number of values: 1
    value: none
    value description: Never restart due to dependency refresh, restart, or failure.
    value constraints:
    value name: none
    value name: error
    value name: restart
```

```
value name: refresh
```

서비스 및 인스턴스 등록 정보 값 표시

이 섹션의 예에서는 다양한 보기, 계층 및 스냅샷에서 서비스 및 인스턴스의 등록 정보와 등록 정보 그룹을 보는 방법을 설명합니다.

예 2-6 현재 사용 중인 인스턴스 및 상속된 등록 정보 나열

기본적으로 `svccprop` 명령은 값이 현재 사용 중인 실행 중인 스냅샷에서 등록 정보에 지정된 값을 표시합니다. 기본적으로 `svccprop` 명령은 실행 중인 스냅샷의 구성된 보기를 표시합니다. 이는 인스턴스별 등록 정보와 상속된 등록 정보가 표시됨을 의미합니다. 상속된 등록 정보의 값이 인스턴스에서 사용자 정의되면 인스턴스에 설정된 값이 표시됩니다. 출력에는 각 등록 정보에 대해 슬래시 문자로 구분된 등록 정보 그룹과 등록 정보 이름, 등록 정보 값의 데이터 유형, 등록 정보 값을 표시하는 하나의 행이 나열됩니다. 등록 정보나 그룹 이름을 지정하지 않으면 실행 중인 스냅샷의 모든 등록 정보 값이 표시됩니다.

FMRI 또는 패턴 피연산자가 인스턴스를 지정하지 않으면 서비스에만 설정된 등록 정보가 표시됩니다. 인스턴스에만 설정된 등록 정보가 표시되지 않습니다. 다음 명령은 서비스 종속성, 서비스 유형, 프로파일 및 매니페스트 파일의 경로와 같은 등록 정보를 표시합니다.

```
$ svccprop svc:/system/identity
```

인스턴스를 지정하면 해당 인스턴스에 대해 사용자 정의된 등록 정보 및 상위 서비스에서 상속된 등록 정보의 구성된 보기가 표시됩니다. 다음 명령은 상위 서비스에서 상속된 등록 정보와 이 인스턴스에 특정한 등록 정보를 포함하여 지정된 인스턴스에 대한 실행 중인 스냅샷에 있는 모든 등록 정보를 나열합니다. 값이 이 인스턴스에 대해 사용자 정의된 상속된 등록 정보의 경우 사용자 정의된 값이 표시됩니다. 이 예는 추가 종속성, 이 인스턴스를 시작하는 실행 파일 경로, 이 인스턴스에 대한 로그 파일 경로, 이 인스턴스의 상태 정보와 같은 등록 정보를 보여줍니다.

```
$ svccprop svc:/system/identity:domain
```

예 2-7 현재 사용 중인 지정된 등록 정보 또는 등록 정보 그룹 나열

-p 옵션을 사용하여 특정 등록 정보 또는 특정 등록 정보 그룹의 모든 등록 정보를 표시합니다.

```
$ svccprop -p pkg/port pkg/server
svc:/application/pkg/server:oss:properties/pkg/port count 82
svc:/application/pkg/server:s11:properties/pkg/port count 81
svc:/application/pkg/server:default:properties/pkg/port count 80
$ svccprop -p pkg pkg/server:s11
pkg/inst_root astring /export/ipsrepos/Solaris11
pkg/port count 81
...
```

```
pkg/ssl_cert_file astring ""
pkg/ssl_key_file astring ""
...
```

예 2-8 편집 보기에서 서비스 및 인스턴스 값 나열

옵션을 사용하면 `svcprop` 명령은 실행 중인 스냅샷 대신 편집 보기를 표시할 수 있습니다. 편집 보기에는 가장 최근 변경 사항이 표시됩니다. 인스턴스를 새로 고치거나 다시 시작하여 편집 보기의 변경 사항을 실행 중인 스냅샷으로 커밋하거나 커밋하지 않을 수 있습니다. 다음 명령은 실행 중인 스냅샷과 편집 보기의 차이점을 보여줍니다. 방금 `oss` 및 `s11` 인스턴스가 생성되었고 등록 정보 값이 변경되었지만 인스턴스는 아직 새로 고쳐지지 않았습니다. 첫 번째 명령은 실행 중인 스냅샷의 구성된 보기를 표시합니다. 이러한 인스턴스가 사용자 정의된 후 새로 고쳐지지 않았으므로 표시된 값은 `pkg/server` 서비스의 값입니다. `-c` 옵션은 편집 값의 구성된 보기를 표시합니다. `-c` 옵션은 구성 없이 편집 값을 표시합니다. 이 보기는 구성된 보기가 아니므로 사용자 정의되지 않은 인스턴스에 대한 값을 찾을 수 없습니다.

```
$ svcprop -p pkg/port pkg/server
svc:/application/pkg/server:oss/:properties/pkg/port count 80
svc:/application/pkg/server:s11/:properties/pkg/port count 80
svc:/application/pkg/server:default/:properties/pkg/port count 80
$ svcprop -c -p pkg/port pkg/server
svc:/application/pkg/server:oss/:properties/pkg/port count 82
svc:/application/pkg/server:s11/:properties/pkg/port count 81
svc:/application/pkg/server/:properties/pkg/port count 80
$ svcprop -C -p pkg/port pkg/server
svc:/application/pkg/server:oss/:properties/pkg/port count 82
svc:/application/pkg/server:s11/:properties/pkg/port count 81
svcprop: Couldn't find property 'pkg/port' for instance 'svc:/application/pkg/server:default'.
```

`svccfg` 명령은 기본적으로 실행 중인 스냅샷이 값이 아니라 편집 등록 정보 값을 표시합니다. `svccfg`를 강제로 실행하여 “[지정된 스냅샷에 값 표시](#)” [44]에 표시된 대로 `selectsnap` 하위 명령을 사용하면 실행 중인 스냅샷의 값을 표시할 수 있습니다.

`svccfg` 명령은 상위 서비스를 지정하면 상위 서비스의 값만 표시하고 인스턴스를 지정하면 인스턴스 값만 표시합니다. `svccfg listprop` 명령에서 출력이 반환되지 않으면 지정한 등록 정보가 지정한 상위 서비스 또는 인스턴스에서 설정되지 않을 수 있습니다. 등록 정보를 삭제한 경우 `listcust -M`를 사용하여 “[구성 사용자 정의 표시](#)” [45]에 표시된 대로 마스킹된 값을 확인합니다.

등록 정보 그룹 또는 등록 정보 이름이 지정되지 않았으므로 다음 명령은 지정된 서비스에 대한 모든 편집 등록 정보 값을 나열합니다. `svcprop svc:/system/identity` 명령으로 표시된 출력 이외에 이 출력에는 등록 정보 그룹 이름 및 유형과 템플릿 데이터가 포함됩니다.

```
$ svccfg -s svc:/system/identity listprop
```

다음 명령은 지정된 서비스 인스턴스에 대한 모든 편집 등록 정보 값을 나열합니다. 예를 들어 이 명령은 구성된 보기를 표시하지 않으므로 이 출력에는 프로파일 및 매니페스트 파일의 경로가 표시되지 않습니다.

```
$ svccfg -s svc:/system/identity:domain listprop
```

예 2-9 편집 보기에서 지정된 등록 정보 또는 등록 정보 그룹 나열

다음 명령은 지정된 서비스 인스턴스에 대한 지정된 등록 정보 그룹의 모든 편집 등록 정보 값을 나열합니다. -o 옵션을 사용하여 표시할 열을 선택할 수 있습니다. 유효한 열 이름 목록은 [svccfg\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
$ svccfg -s pkg/server:s11 listprop pkg
pkg                application
pkg/inst_root      astring            /export/ipsrepos/Solaris11
pkg/port           count              81
$ svccfg -s pkg/server:s11 listprop -o propname,value pkg
inst_root          /export/ipsrepos/Solaris11
port               81
```

등록 정보 그룹 유형의 등록 정보 표시

등록 정보 이름이나 등록 정보 그룹 이름별로 등록 정보 값을 표시하고 등록 정보 그룹 유형별로 등록 정보 값을 표시할 수도 있습니다.

예 2-10 등록 정보 그룹 및 해당 유형 표시

svccfg 명령의 listpg 하위 명령은 각 등록 정보 그룹의 이름과 유형을 표시합니다.

```
$ svccfg -s pkg/server listpg
pkg                application
pkg_bui            application
pkg_secure         application
fs                 dependency
autofs             dependency
ntp                dependency
network            dependency
general            framework
manifestfiles     framework
start              method
stop               method
tm_common_name    template
$ svccfg -s pkg/server:s11 listpg
pkg                application
general            framework
restarter         framework          NONPERSISTENT
```

비영구 등록 정보 그룹은 일반적으로 활성 프로그램 상태를 포함합니다. 비영구 등록 정보 그룹의 등록 정보 값은 시스템 부트 중에 지워집니다.

등록 정보 그룹 이름을 지정하여 해당 등록 정보 그룹의 유형만 표시합니다.

```
$ svccfg -s pkg/mirror listpg config
```

```
config application
```

예 2-11 등록 정보 그룹 유형의 등록 정보 나열

svccfg 명령의 `-g` 옵션을 사용하여 특정 등록 정보 그룹 유형의 등록 정보를 표시합니다. 등록 정보 그룹 유형에는 `application`, `dependency`, `method`, `framework` 및 `template`이 포함됩니다.

```
$ svccfg -g method pkg/server:s11
start/exec astring %p{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m
start/timeout_seconds count 0
start/type astring method
stop/exec astring %p{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m %r{restarter/contract}
stop/timeout_seconds count 30
stop/type astring method
$ svccfg -g method -p exec pkg/server:s11
start/exec astring %p{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m
stop/exec astring %p{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m %r{restarter/contract}
```

값이 설정된 계층 표시

서비스 구성 저장소에서는 등록 정보 데이터가 데이터 소스에 따라 계층에 저장됩니다. `svccfg` 및 `svccfg` 명령은 등록 정보 값의 소스인 계층을 표시할 수 있습니다. `svccfg` 및 `svccfg` 명령의 `-l` 옵션에는 정보를 표시하려는 계층을 지정할 인수가 필요합니다. 인수 값은 `manifest`, `system-profile`, `site-profile` 및 `admin`입니다. 출력에는 특정 등록 정보 값이 서비스 매니페스트 또는 프로파일에 설정되었는지, 아니면 관리자가 설정했는지가 표시됩니다. 계층에 대한 자세한 내용은 “저장소 계층” [24]을 참조하십시오. 키워드 `all`은 모든 계층의 별칭입니다. 지정하는 계층이 요청한 등록 정보 값의 소스가 아니면 출력이 표시되지 않습니다.

다음 명령은 일부 등록 정보 값의 소스가 매니페스트이고, 일부는 관리자가 설정했으며, 일부 등록 정보의 값이 두 개 이상의 계층에 있음을 보여줍니다. `pkg/readonly` 등록 정보의 값은 서비스 매니페스트에 설정되고 관리자도 이와 같은 값을 설정합니다. 계층에 따라 값이 다를 수 있습니다.

```
$ svccfg -l all -p pkg pkg/server:s11
pkg/port count admin 81
pkg/inst_root astring admin /export/ipsrepos/Solaris11
pkg/address net_address manifest
pkg/cfg_file astring manifest ""
...
pkg/readonly boolean manifest true
pkg/readonly boolean admin true
...
```

`svccfg listprop` 명령의 `-l` 옵션이 인수 `current`를 사용할 수도 있습니다. `current`를 `-l` 인수로 사용하면 `-l` 옵션을 사용하지 않을 때 표시된 것과 같은 등록 정보 값이 표시됩니다. 출력의 유일한 차이는 계층 이름도 표시된다는 점입니다. 서비스 구성 저장소에는 비영구 데이

터에 대한 계층 정보가 저장되지 않으므로 비영구 데이터에는 계층 이름이 표시되지 않습니다(세번째 열에는 <none>이 표시됨). 비영구 등록 정보 그룹은 일반적으로 활성 프로그램 상태를 포함하고 비영구 등록 정보 그룹의 등록 정보 값은 시스템 부트 중에 지워집니다.

```
$ svccfg -s pkg/server:s11 listprop -l current
pkg                application admin
pkg/inst_root      astring    admin    /export/ipsrepos/Solaris11
pkg/port           count      admin    81
general            framework  admin
general/complete  astring    manifest
general/enabled    boolean    admin    true
restarter          framework  <none>    NONPERSISTENT
restarter/logfile  astring    <none>    /var/svc/log/application-pkg-
server:default.log
restarter/contract count      <none>    121
restarter/start_pid count      <none>    1055
restarter/start_method_timestamp time       <none>    1379605275.329096000
restarter/start_method_waitstatus integer     <none>    0
restarter/auxiliary_state astring    <none>    dependencies_satisfied
restarter/next_state astring    <none>    none
restarter/state    astring    <none>    online
restarter/state_timestamp time       <none>    1379605275.332259000
```

지정된 스냅샷에 값 표시

다음 명령은 이 서비스 인스턴스에 사용할 수 있는 스냅샷을 나열합니다. `svccfg` 또는 `svccfg`와 함께 스냅샷 이름을 사용하여 해당 스냅샷에서 설정된 등록 정보 값을 표시합니다. 인스턴스에만 스냅샷이 있습니다. 서비스에는 스냅샷이 없습니다. 서비스 구성 저장소의 스냅샷에 대한 자세한 내용은 “저장소 스냅샷” [25]을 참조하십시오.

```
$ svccfg -s pkg/server:default listsnap
initial
previous
running
start
$ svccfg -s pkg/server:s11 listsnap
previous
running
start
```

다음 명령은 `pkg/inst_root` 등록 정보 값이 이전 스냅샷의 값과 다름을 보여줍니다.

```
$ svccfg -s pkg/server:s11 listprop pkg/inst_root
/var/share/pkg/repositories/solaris
$ svccfg -s pkg/server:s11
[previous]svc:/application/pkg/server:s11> selectsnap previous
[previous]svc:/application/pkg/server:s11> listprop pkg/inst_root
pkg/inst_root astring    /var/share/pkg/repositories/solaris
[previous]svc:/application/pkg/server:s11> exit
```

구성 사용자 정의 표시

svccfg listcust 명령은 지정된 서비스에 대한 admin 계층의 사용자 정의를 표시합니다. 또한 -L 옵션을 사용하여 site-profile 계층의 사용자 정의를 표시합니다. 다음 명령은 pkg/server 서비스의 pkg/port 및 pkg/inst_root 등록 정보를 관리자가 사용자 정의했음을 보여줍니다.

```
$ svccfg -s pkg/server:s11 listcust
pkg                application admin
pkg/port           count      admin      81
pkg/inst_root     astring   admin      /export/ipsrepos/Solaris11
general           framework admin
general/complete astring   manifest
general/enabled   boolean   admin      true
```

다음 svcprop 명령은 등록 정보 config/nodename의 정의가 manifest 계층에서 제공되지만 값 solaris가 admin 계층에서 설정됨을 보여줍니다.

```
$ svccfg -s identity:node describe config/nodename
config/nodename astring   solaris
Network name of the computer
$ svcprop -p config/nodename -l all svc:/system/identity:node
config/nodename astring manifest ""
config/nodename astring admin solaris
```

또한 svccfg listcust 명령은 모든 마스킹된 엔티티를 표시합니다. -M 옵션을 사용하여 마스킹된 엔티티를 나열합니다. svccfg delcust 명령을 사용하기 전에 svccfg listcust 명령을 사용하여 삭제할 대상을 확인합니다. 마스킹된 엔티티에 대한 자세한 내용은 [“등록 정보 그룹, 등록 정보 및 등록 정보 값 삭제” \[68\]](#) 및 [smf\(5\)](#) 매뉴얼 페이지를 참조하십시오.

이벤트 알림 매개변수 표시

svcs -n 명령은 FMA 이벤트 알림 매개변수, 시스템 전체 SMF 상태 전환 알림 매개변수, 서비스 인스턴스 상태 전환 알림 매개변수를 표시합니다. 이러한 매개변수에 대한 자세한 내용은 [smf\(5\)](#) 매뉴얼 페이지에서 "알림 매개변수"를 참조하십시오.

```
$ svcs -n
Notification parameters for FMA Events
Event: problem-diagnosed
Notification Type: smtp
Active: true
reply-to: root@localhost
to: root@localhost

Notification Type: snmp
Active: true
```

```

Notification Type: syslog
Active: true

Event: problem-repaired
Notification Type: snmp
Active: true

Event: problem-resolved
Notification Type: snmp
Active: true

System wide notification parameters:
svc:/system/svc/global:default:
Event: to-maintenance
Notification Type: smtp
Active: true
to: sysadmins@example.com

svc:/application/pkg/mirror:default:
Event: to-maintenance
Notification Type: smtp
Active: true
to: installteam@example.com

```

세 가지 FMA 이벤트인 `problem-diagnosed`, `problem-repaired` 및 `problem-resolved`가 표시됩니다. 또한 네번째 이벤트인 `problem-updated`에 대한 알림 매개변수를 구성할 수 있습니다.

시스템 전체 상태 전환 알림 설정의 경우 이러한 전역 설정을 저장하는 서비스도 나열됩니다. 이 시스템 전체 설정은 사용자 정의 설정입니다. 시스템 전체 또는 전역 값은 사용자 정의 값이 설정되지 않은 모든 서비스 인스턴스에 적용됩니다.

표시된 마지막 설정은 특정 서비스 인스턴스에 대한 사용자 정의 설정입니다.

`svccfg listnotify` 명령을 사용하여 지정된 이벤트에 대한 알림 매개변수만 표시합니다. 상태 전환 이벤트의 경우 `-g` 옵션을 사용하여 전역 설정을 표시합니다. 출력에는 알림 매개변수 값의 소스도 표시됩니다.

```

$ svccfg listnotify problem-resolved
Event: problem-resolved (source: svc:/system/fm/notify-params:default)
Notification Type: snmp
Active: true
$ svccfg listnotify -g to-maintenance
Event: to-maintenance (source: svc:/system/svc/global:default)
Notification Type: smtp
Active: true
to: sysadmins@example.com
$ svccfg -s pkg/mirror listnotify to-maintenance
Event: to-maintenance (source: svc:/application/pkg/mirror)
Notification Type: smtp
Active: true
to: installteam@example.com

```

이벤트 알림 구성에 대한 자세한 내용은 “[상태 전환 및 FMA 이벤트에 대한 알림 구성](#) [57]을 참조하십시오.

◆◆◆ 3 장 3

서비스 관리

이 장에서는 서비스를 시작, 중지, 다시 시작하는 방법과 서비스 구성을 다시 읽는 방법을 설명합니다. 이 장에서는 사용자에게 FMA 이벤트나 서비스 상태 전환에 대해 알리게 시스템을 구성하는 방법을 설명합니다. 이러한 변경 사항은 “저장소 계층” [24]에 설명된 대로 admin 레벨 사용자 정의입니다.

서비스 상태를 변경하는 명령은 svcadm입니다. svcadm 명령은 서비스 인스턴스에서 작업을 수행합니다. 인스턴스를 지정하지 않고 서비스 이름을 제공하고 해당 서비스에 단일 인스턴스만 있으면 svcadm이 해당 인스턴스에서 작업을 수행합니다. 인스턴스를 지정하지 않고 서비스 이름을 제공하고 해당 서비스에 여러 인스턴스가 있거나, 여러 인스턴스와 일치하는 다른 패턴을 지정하면 svcadm이 오류 메시지를 표시합니다.

SMF 서비스 인스턴스 관리

서비스 인스턴스는 항상 “서비스 상태” [20]에 설명된 상태 중 하나입니다. 이 절에서는 인스턴스를 다른 상태로 전환하는 방법, 업데이트된 등록 정보 값을 실행 중인 스냅샷에 커밋하는 방법, 기본 보기에서 인스턴스를 삭제하는 방법을 설명합니다.

서비스 시작

다음 상태의 서비스 인스턴스는 이미 사용으로 설정되어 있으므로 시작할 필요가 없습니다. degraded, maintenance, offline, online. 시작하려는 인스턴스가 degraded, maintenance 또는 offline 상태이면 “성능 저하, 오프라인 또는 유지 관리 상태인 인스턴스 복구” [97]를 참조하십시오. 시작하려는 인스턴스가 disabled 상태이면 다음 절차에 나와 있는 대로 인스턴스를 사용으로 설정합니다. 인스턴스를 사용으로 설정하면 인스턴스의 다시 시작 프로그램이 인스턴스를 online 상태로 전환하려고 합니다.

▼ 서비스 인스턴스를 사용으로 설정하는 방법

1. 인스턴스 상태와 종속성을 확인합니다.

인스턴스가 현재 사용 안함으로 설정되고 필요한 종속성이 모두 online 또는 degraded 상태에서 실행 중인지 확인합니다.

```
$ svcs -l FMRI
```

2. 인스턴스를 사용으로 설정합니다.

서비스의 다시 시작 프로그램이 지정된 인스턴스를 online 상태로 전환하려고 합니다.

인스턴스가 영구적으로 또는 일시적으로 사용으로 설정될 수 있습니다. 영구적 사용은 시스템을 재부트해도 지속되고 기본값입니다. 일시적 사용은 재부트될 때까지만 지속됩니다.

■ 인스턴스를 영구적으로 사용으로 설정합니다.

```
$ svcadm enable FMRI
```

■ 인스턴스를 일시적으로 사용으로 설정합니다.

-t 옵션을 사용하여 일시적 사용을 지정합니다.

```
$ svcadm enable -t FMRI
```

지금 인스턴스를 실행하지만 다음 재부트 시 실행하지 않으려면 인스턴스가 사용 안함으로 설정되었는지 확인하고 인스턴스를 일시적으로 사용으로 설정합니다. 인스턴스가 일시적으로 사용으로 설정되었는지 확인하려면 `svcs -l` 명령을 사용하고 `enabled` 행을 확인합니다.

```
enabled      true (temporary)
```

■ 인스턴스를 동기적으로 사용으로 설정합니다.

-s 옵션을 지정하면 `svcadm`은 인스턴스를 사용으로 설정하고 돌아가기 전에 인스턴스가 online 또는 degraded 상태로 전환할 때까지 대기합니다. `svcadm` 명령은 인스턴스가 온라인 상태로 전환되거나 인스턴스가 온라인 상태로 전환하는 데 관리자 개입이 필요하다고 결정되면 돌아갑니다.

-T 옵션을 -s 옵션과 함께 사용하여 전환을 수행할 상한(초)을 지정하거나 전환을 수행할 수 없게 결정합니다.

```
$ svcadm enable -sT 10 FMRI
```

3. 인스턴스가 온라인 상태인지 확인합니다.

```
$ svcs FMRI
```

인스턴스가 degraded, maintenance 또는 offline 상태이면 “[성능 저하, 오프라인 또는 유지 관리 상태인 인스턴스 복구](#)” [97]를 참조하십시오.

예 3-1 서비스 인스턴스를 영구적으로 사용으로 설정

다음 명령은 pkg/mirror:default 서비스 인스턴스가 현재 사용 안함으로 설정되고 모든 필요한 종속성이 온라인 상태임을 보여줍니다.

```
$ svcs -l pkg/mirror
fmri          svc:/application/pkg/mirror:default
name          IPS Repository Mirror
enabled       false
state         disabled
next_state    none
state_time    September 17, 2013 07:16:52 AM PDT
restarter     svc:/system/svc/restarter:default
manifest      /lib/svc/manifest/application/pkg/pkg-mirror.xml
dependency    require_all/error svc:/milestone/network:default (online)
dependency    require_all/none svc:/system/filesystem/local:default (online)
dependency    optional_all/error svc:/system/filesystem/autofs:default (online)
dependency    require_all/none svc:/application/pkg/repositories-setup (online)
```

다음 명령은 pkg/mirror:default 인스턴스를 사용으로 설정합니다. 이 경우 svcadm 명령은 pkg/mirror:default 인스턴스가 성공적으로 사용으로 설정되었으므로 돌아옵니다.

```
$ svcadm enable -sT 10 pkg/mirror:default
$ svcs pkg/mirror
STATE      STIME      FMRI
online     22:03:53  svc:/application/pkg/mirror:default
```

예 3-2 서비스 인스턴스를 일시적으로 사용으로 설정

다음 명령은 net-snmp:default 서비스 인스턴스가 현재 사용 안함으로 설정되고 모든 필요한 종속성이 온라인 상태임을 보여줍니다. 사용 안함으로 설정된 하나의 종속성은 선택적 종속성입니다.

```
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
enabled       false
state         disabled
next_state    none
state_time    September 17, 2013 05:56:39 PM PDT
logfile       /var/svc/log/application-management-net-snmp:default.log
restarter     svc:/system/svc/restarter:default
contract_id   /etc/svc/profile/generic.xml
manifest      /lib/svc/manifest/application/management/net-snmp.xml
dependency    require_all/none svc:/system/filesystem/local (online)
dependency    optional_all/none svc:/milestone/name-services (online)
dependency    optional_all/none svc:/system/system-log (online)
dependency    optional_all/none svc:/network/rpc/rstat (disabled)
dependency    require_all/restart svc:/system/cryptosvc (online)
dependency    require_all/restart svc:/milestone/network (online)
dependency    require_all/refresh file://localhost/etc/net-snmp/snmp/snmpd.conf (online)
dependency    require_all/none svc:/milestone/multi-user (online)
```

다음 예와 같이 `-t` 옵션을 사용하여 인스턴스를 사용으로 설정한 후 인스턴스는 일시적으로 사용으로 설정되고 온라인 상태이며, `svcs -p` 명령을 통해 표시된 대로 `snmpd` 프로세스를 시작했으므로 계약 ID를 포함합니다.

```
$ svcadm enable -t net-snmp:default
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
enabled       true (temporary)
state         online
next_state    none
state_time    September 17, 2013 05:57:26 PM PDT
logfile       /var/svc/log/application-management-net-snmp:default.log
restarter     svc:/system/svc/restarter:default
contract_id   160
manifest      /etc/svc/profile/generic.xml
manifest      /lib/svc/manifest/application/management/net-snmp.xml
dependency    require_all/none svc:/system/filesystem/local (online)
dependency    optional_all/none svc:/milestone/name-services (online)
dependency    optional_all/none svc:/system/system-log (online)
dependency    optional_all/none svc:/network/rpc/rstat (disabled)
dependency    require_all/restart svc:/system/cryptosvc (online)
dependency    require_all/restart svc:/milestone/network (online)
dependency    require_all/refresh file://localhost/etc/net-snmp/snmpd.conf (online)
dependency    require_all/none svc:/milestone/multi-user (online)
$ svcs -p net-snmp
STATE        STIME      FMRI
online       17:57:26  svc:/application/management/net-snmp:default
              17:57:26      5022 snmpd
```

서비스 중지

`svcadm disable` 명령을 사용하여 사용으로 설정되거나 일시적으로 사용 안함으로 설정된 서비스 인스턴스를 사용 안함으로 설정합니다. 사용 안함으로 설정된 인스턴스는 다시 시작할 수 없습니다. 먼저 인스턴스를 사용으로 설정해야 합니다.

▼ 서비스 인스턴스를 사용 안함으로 설정하는 방법

1. 다른 서비스가 이 인스턴스에 종속되는지 확인합니다.

- a. 이 인스턴스에 종속된 서비스를 나열합니다.

```
$ svcs -D FMRI
```

- b. 종속 서비스에 이 인스턴스가 필요한지 확인합니다.

`svcs -D` 명령의 각 결과에 대해 `svcs -l` 명령을 사용하여 종속성이 필수 종속성인지 확인합니다.

이 인스턴스가 다른 서비스의 필수 종속성이면 이 인스턴스를 사용 안함으로 설정하지 않아야 합니다.

2. 인스턴스를 사용 안함으로 설정합니다.

서비스의 다시 시작 프로그램이 지정된 인스턴스를 `disabled` 상태로 전환하려고 합니다.

인스턴스가 영구적으로 또는 일시적으로 사용 안함으로 설정될 수 있습니다. 영구적 사용 안함은 시스템을 재부트해도 지속되고 기본값입니다. 일시적 사용 안함은 재부트될 때까지만 지속됩니다.

■ 인스턴스를 영구적으로 사용 안함으로 설정합니다.

```
$ svcadm disable FMRI
```

■ 인스턴스를 일시적으로 사용 안함으로 설정합니다.

`-t` 옵션을 사용하여 일시적 사용 안함을 지정합니다.

```
$ svcadm disable -t FMRI
```

지금 인스턴스를 사용 안함으로 설정하지만 다음 재부트 시 실행하려면 인스턴스가 `online` 또는 `degraded` 상태에서 실행 중인지 확인하고 인스턴스를 일시적으로 사용 안함으로 설정합니다. 인스턴스가 일시적으로 사용 안함으로 설정되었는지 확인하려면 `svcs -l` 명령을 사용하고 `enabled` 행을 확인합니다.

```
enabled      false (temporary)
```

■ 인스턴스를 동기적으로 사용 안함으로 설정합니다.

`-s` 옵션을 지정하면 `svcadm`은 인스턴스를 사용 안함으로 설정하고 돌아가기 전에 인스턴스가 `disabled` 상태로 전환할 때까지 대기합니다. `svcadm` 명령은 인스턴스가 `disabled` 상태로 전환되거나 인스턴스가 `disabled` 상태로 전환하는 데 관리자 개입이 필요하다고 결정되면 돌아갑니다.

`-T` 옵션을 `-s` 옵션과 함께 사용하여 전환을 수행할 상한(초)을 지정하거나 전환을 수행할 수 없게 결정합니다.

```
$ svcadm disable -sT 10 FMRI
```

3. 인스턴스가 사용 안함으로 설정되었는지 확인합니다.

```
$ svcs FMRI
```

예 3-3 서비스 인스턴스를 사용 안함으로 설정

이 예는 `pkg/update:default` 서비스 인스턴스가 처음에 온라인 상태이고 다른 서비스가 이 인스턴스에 종속되지 않았음을 보여줍니다. `svcadm disable` 명령이 성공하고 인스턴스가 현재 `disabled` 상태이며 다시 시작 시도가 실패합니다.

```
$ svcs pkg/update
```

```

STATE      STIME      FMRI
online     7:18:17   svc:/application/pkg/update:default
$ svcs -D pkg/update:default
STATE      STIME      FMRI
$ svcadm disable pkg/update
$ svcs pkg/update
STATE      STIME      FMRI
disabled  22:51:12  svc:/application/pkg/update:default
$ svcadm restart pkg/update:default
$ svcs pkg/update
STATE      STIME      FMRI
disabled  22:51:12  svc:/application/pkg/update:default
    
```

서비스 다시 시작

다시 시작 작업은 현재 online 또는 degraded 상태에서 실행 중인 인스턴스만 다시 시작합니다. 예를 들어 인스턴스가 실행되는 동안 적용할 수 없는 구성 변경을 수행했기 때문에 실행 중인 인스턴스를 다시 시작해야 할 수 있습니다.

서비스 인스턴스를 다시 시작해도 구성이 새로 고쳐지지 않습니다. svcadm restart 명령은 인스턴스의 중지 메소드를 실행한 다음 인스턴스의 시작 메소드를 실행합니다. svcadm restart 명령은 등록 정보 변경 사항을 실행 중인 스냅샷에 커밋하지 않고 인스턴스의 새로 고침 메소드를 실행하지 않습니다. 구성 변경 사항을 실행 중인 스냅샷에 커밋하는 방법에 대한 자세한 내용은 [“서비스 구성 다시 읽기” \[55\]](#)를 참조하십시오.

manifest-import 서비스를 다시 시작하는 것은 특수한 경우입니다. manifest-import 서비스를 다시 시작하면 표준 위치의 변경된 모든 매니페스트 또는 프로파일을 가져오고, 서비스 구성 저장소로 변경 사항이 커밋되고, 새로운 실행 중인 스냅샷이 작성되고, 새로 고침 메소드가 경우 변경된 인스턴스의 새로 고침 메소드가 실행됩니다.

▼ 서비스 인스턴스를 다시 시작하는 방법

1. 인스턴스 상태를 확인합니다.

인스턴스는 online 또는 degraded 상태에 있어야 합니다.

```
$ svcs FMRI
```

2. 인스턴스를 다시 시작합니다.

서비스의 다시 시작 프로그램이 지정된 인스턴스를 online 상태로 전환하려고 합니다. 대부분 다시 시작 프로그램은 중지 작업에 이어서 시작 작업을 실행하여 다시 시작 작업을 구현합니다.

■ 인스턴스를 다시 시작합니다.

```
$ svcadm restart FMRI
```

■ 인스턴스를 동기적으로 다시 시작합니다.

-s 옵션을 지정하면 svcadm은 인스턴스를 다시 시작하고 돌아가기 전에 인스턴스가 online, degraded 또는 maintenance 상태로 전환할 때까지 대기합니다. svcadm 명령은 인스턴스가 이러한 상태로 전환되거나 인스턴스가 이러한 상태 중 하나로 전환하는 데 관리자 개입이 필요하다고 결정되면 돌아갑니다.

-T 옵션을 -s 옵션과 함께 사용하여 전환을 수행할 상한(초)을 지정하거나 전환을 수행할 수 없게 결정합니다.

```
$ svcadm restart -sT 10 FMRI
```

3. 인스턴스가 시작되었는지 확인합니다.

다시 시작에 성공하면 인스턴스는 online, degraded 또는 maintenance 상태입니다. 인스턴스가 degraded 또는 maintenance 상태이면 “[성능 저하, 오프라인 또는 유지 관리 상태인 인스턴스 복구](#)” [97]를 참조하십시오.

```
$ svcs FMRI
```

서비스 구성 다시 읽기

서비스 구성을 변경하면 해당 변경 사항이 실행 중인 스냅샷에 바로 나타나지 않습니다. 해당 변경 사항은 서비스 구성 저장소에 현재 또는 편집 등록 정보 값으로 저장됩니다. 새로 고침 작업은 편집 구성의 값을 사용하여 지정된 서비스 인스턴스의 실행 중인 스냅샷을 업데이트합니다.

svcadm refresh 및 svccfg refresh 명령은 다음 단계를 둘 다 수행합니다.

1. 새로운 실행 중인 스냅샷을 만들어 편집 등록 정보를 실행 중인 스냅샷에 커밋합니다.
2. 새로 고침 메소드가 있고 인스턴스가 online 또는 degraded 상태이면 인스턴스의 새로 고침 메소드를 실행합니다. 새로 고침 메소드는 변경 사항이 적용되었음을 응용 프로그램에 알려야 합니다. 새로 고침 메소드는 실행 중인 스냅샷에서 등록 정보 값을 다시 읽을 수 있습니다. 새로 고침 메소드가 있어도 실행 중인 스냅샷의 구성이 업데이트됩니다.

svcadm refresh 명령은 서비스 인스턴스에서 작업을 수행합니다. svccfg refresh 명령은 서비스 인스턴스 또는 상위 서비스에서 작업을 수행합니다. 서비스를 지정하면 svccfg refresh 명령은 해당 서비스의 모든 인스턴스를 새로 고칩니다. 상위 서비스가 아니라 서비스 인스턴스에 대한 스냅샷만 작성되지만 상위 서비스 등록 정보는 서비스 인스턴스에 상속됩니다. 인스턴스가 변경된 상위 서비스 등록 정보를 대체하지 않으면 해당 변경 사항이 서비스 인스턴스 스냅샷에 나타납니다.

종속성 변경과 같은 일부 변경 사항은 바로 적용됩니다. “[서비스 다시 시작](#)” [54]에 설명된 대로 다른 변경 사항은 서비스가 다시 시작될 때까지 적용되지 않습니다. 응용 프로그램이 실행되는 동안 적용할 수 없는 변경 사항을 적용하려면 새로 고침 후 다시 시작해야 합니다.

다. 응용 프로그램이 실행되는 동안 적용할 수 없는 변경 사항의 예에는 소켓 닫기 또는 열기 나 환경 변수 재설정이 포함됩니다.

-s 옵션을 svcadm refresh 명령과 함께 지정하면 svcadm은 인스턴스를 새로 고치고 돌아가기 전에 인스턴스가 online, degraded 또는 maintenance 상태로 전환할 때까지 대기합니다. svcadm 명령은 인스턴스가 이러한 상태로 전환되거나 인스턴스가 이러한 상태 중 하나로 전환하는 데 관리자 개입이 필요하다고 결정되면 돌아갑니다. -t 옵션을 -s 옵션과 함께 사용하여 전환을 수행할 상한(초)을 지정하거나 전환을 수행할 수 없게 결정합니다.

서비스 삭제

svccfg delete 명령은 시스템에서 서비스 인스턴스를 제거하지 않습니다. 대신 svccfg delete 명령은 인스턴스를 마스킹합니다. svccfg delete 명령을 실행한 후에도 서비스 매니페스트가 /lib/svc/manifest에 있습니다. SMF는 서비스 구성 저장소를 파일 시스템 콘텐츠와 동기화된 상태로 유지합니다. 매니페스트가 표준 위치의 파일 시스템에 있으므로 해당 서비스 정보는 저장소에 저장되고 기본 보기에서만 마스킹됩니다. 모든 관리 사용자 정의가 마스킹된 인스턴스에서 삭제됩니다. 마스킹된 엔티티에 대한 설명은 [smf\(5\)](#) 매뉴얼 페이지를 참조하십시오.

서비스 인스턴스를 지원하는 파일은 해당 서비스 인스턴스가 마스킹되더라도 pkg 명령을 사용할 때 업데이트됩니다. 서비스 인스턴스를 지원하는 파일이 pkg 명령으로 업데이트되면 SMF 데이터 저장소는 서비스가 보기에서 마스킹되어 있더라도 업데이트됩니다. 서비스 인스턴스가 마스킹되지 않으면 해당 서비스 인스턴스는 추가 개입을 필요로 하지 않고 pkg를 통해 제공된 파일에서 이미 업데이트된 것입니다. 서비스 인스턴스를 마스킹 해제하려면 [서비스 인스턴스 삭제 실행 취소 \[57\]](#)를 참조하십시오.

▼ 서비스 인스턴스를 삭제하는 방법

1. 삭제할 인스턴스 종속 항목을 확인합니다.

svcs -D 명령을 사용하여 이 인스턴스에 종속된 인스턴스를 표시합니다. 이 인스턴스를 삭제한 후에는 종속 인스턴스를 실행하지 못할 수 있습니다. svcs -l 명령을 사용하여 이 인스턴스가 종속 인스턴스의 필수 종속성인지 확인합니다.

2. 인스턴스를 마스킹합니다.

svccfg delete 명령을 사용하여 기본 보기에서 인스턴스를 마스킹합니다. svcs 명령을 사용하여 인스턴스 상태를 표시합니다. 인스턴스가 online 또는 degraded 상태에서 실행 중이면 svccfg delete -f 명령을 사용하여 기본 보기에서 인스턴스를 마스킹합니다.

```
$ svcs -H my-svc
disabled          7:25:37 svc:/site/my-svc:default
$ svccfg delete svc:/site/my-svc:default
```

3. 인스턴스가 마스킹되었는지 확인합니다.

svccfg listcust -M 명령을 사용하여 인스턴스가 마스킹되었는지 확인합니다. svcs와 같은 명령은 일치하는 인스턴스를 찾을 수 없다는 오류 메시지를 표시해야 합니다.

```
$ svccfg listcust -M
svc:/site/my-svc:default manifest MASKED
  general                admin    MASKED
  general/complete astring admin    MASKED
  general/enabled boolean  admin    MASKED true
$ svcs -H my-svc
svcs: Pattern 'my-svc' doesn't match any instances
```

▼ 서비스 인스턴스 삭제 실행 취소

1. 인스턴스가 마스킹되었는지 확인합니다.

이전 절차에 표시된 대로 svccfg listcust -M 명령을 사용합니다.

2. 인스턴스를 마스킹 해제합니다.

```
$ svccfg -s svc:/site/my-svc:default delcust
Deleting customizations for instance: default
```

매니페스트를 다시 가져와도 마스크가 제거되지 않습니다.

3. 인스턴스가 마스킹 해제되었는지 확인합니다.

svccfg listcust -M 명령을 사용하여 인스턴스가 마스킹되지 않았는지 확인합니다. svcs 명령은 인스턴스 상태를 표시해야 합니다.

상태 전환 및 FMA 이벤트에 대한 알림 구성

서비스 상태가 변경되거나 FMA 이벤트가 발생할 때 알리도록 시스템을 구성할 수 있습니다. SMTP(Simple Mail Transfer Protocol) 또는 SNMP(Simple Network Management Protocol) 알림을 지정할 수 있습니다.

기본적으로 SNMP 트랩은 유지 관리 전환 시에 보내집니다. 전환 알림에 SNMP를 사용하면 다른 상태 전환에 대해 추가적인 트랩을 구성할 수 있습니다.

다음 예에서는 SMF 및 FMA 이벤트에 대한 알림 매개변수를 설정하는 방법과 알림 매개변수를 삭제하는 방법을 보여줍니다.

예 3-4 서비스 상태 이벤트에 대한 전역 알림 구성

다음 명령은 서비스가 maintenance 상태가 될 경우 전자 메일을 보내는 알림을 만듭니다.

```
$ svccfg setnotify -g to-maintenance mailto:sysadmins@example.com
```

-g	-g 옵션은 사용자 정의 값 세트가 없는 모든 서비스 인스턴스에 대해 이 알림 매개변수를 설정합니다. 모든 수정된 서비스 인스턴스가 새로 고쳐집니다. -g 옵션은 FMA 이벤트 없이 서비스 상태 전환에 대한 알림을 설정할 때만 사용할 수 있습니다.
to-maintenance	to-maintenance 인수는 smf(5) 매뉴얼 페이지의 "알림 매개변수"에 설명된 상태 전환 이벤트입니다. 상태 이름만 지정하는 작업에는 to-state 및 from-state 전환이 둘 다 포함됩니다. 이 이벤트는 심표를 구분된 전환 목록일 수도 있습니다.
mailto:	mailto 인수는 지정된 이벤트에 대해 수신하려는 알림을 지정합니다. 이 인수는 snmp를 지정할 수도 있습니다. snmp 알림 값은 snmp:active 또는 snmp:inactive여야 합니다. mailto 알림 값은 이 예에 표시된 양식 외에 mailto:active 또는 mailto:inactive일 수 있습니다. 알림 매개변수를 설정하면 해당 이벤트에 대한 기존 값을 덮어씁니다. active 및 inactive 설정은 기존 값을 덮어쓰지 않고 지정된 이벤트에 대해 기존 알림을 적용할지를 토글합니다.

예 3-5 지정된 서비스 인스턴스에 대한 알림 구성

다음 명령은 pkg/mirror 서비스가 maintenance 상태로 전환될 때 전자 메일을 보내는 알림을 만듭니다.

```
$ svccfg -s pkg/mirror setnotify to-maintenance mailto:installteam@example.com
```

다음 명령은 http:apache22 서비스가 online 상태에서 다른 상태로 전환될 때 전자 메일을 보내는 알림을 만듭니다.

```
$ svccfg -s http:apache22 setnotify from-online mailto:webservices@example.com
```

예 3-6 FMA 이벤트에 대한 알림 구성

problem-diagnosed 인수는 FMA 이벤트입니다. 이 인수는 심표로 구분된 FMA 이벤트 목록일 수 있습니다. [smf\(5\)](#) 매뉴얼 페이지의 "알림 매개변수"에서 FMA 이벤트 목록을 참조하십시오.

```
$ svccfg setnotify problem-diagnosed mailto:IT@example.com
```

예 3-7 알림 설정 삭제

다음 명령은 이전 예에서 설정된 알림 설정을 삭제합니다.

```
$ svccfg delnotify -g to-maintenance
$ svccfg -s pkg/mirror delnotify to-maintenance
$ svccfg setnotify problem-diagnosed mailto:root@localhost
```

서비스 구성

SMF는 구성 데이터를 서비스 구성 저장소에 저장합니다. SMF 서비스 구성은 구성 저장소에서 데이터를 수정한 다음 수정 사항을 실행 중인 스냅샷에 커밋하는 작업입니다. 이 장에서는 구성 저장소에서 데이터를 수정하는 방법을 설명합니다. 구성 저장소의 데이터를 보려면 “서비스 구성 검사” [39]를 참조하십시오. 실행 중인 스냅샷으로 구성 수정 사항을 커밋하려면 “서비스 구성 다시 읽기” [55]를 참조하십시오.

각 서비스 및 서비스 인스턴스는 구성 데이터를 등록 정보 그룹으로 구성되는 등록 정보에 저장합니다. 구성 저장소의 데이터 수정에는 서비스 등록 정보 값 수정, 사용자 정의 등록 정보 그룹 및 등록 정보 만들기, 서비스의 새 인스턴스 만들기, 프로파일 적용이 포함됩니다. 구성 수정에는 사용자 정의 삭제와 저장소 스냅샷 되돌리기도 포함됩니다.

이 장에서는 inetd 서비스를 수정하는 방법도 보여줍니다.

SMF 구성 변경은 Oracle Solaris 감사 프레임워크를 사용하여 기록할 수 있습니다. 자세한 내용은 *Oracle Solaris 11.2*에서 감사 관리의 “감사 서비스 구성(작업 맵)”을 참조하십시오.

서비스 구성 명령 사용

svccfg 명령은 서비스 구성 저장소에서 데이터를 조작합니다. svccfg 명령으로 변경한 사항은 admin 계층에 기록됩니다. 계층에 대한 자세한 내용은 “저장소 계층” [24]을 참조하십시오. svccfg 명령으로 변경한 사항은 서비스 구성 저장소에 현재 또는 편집 등록 정보 값으로 저장되며, 실행 중인 스냅샷에 바로 나타나지 않습니다. 구성 데이터를 변경할 때 svcadm refresh 또는 svccfg refresh 명령을 사용하여 새 값을 실행 중인 스냅샷에 커밋합니다.

새로 변경된 데이터를 실행 중인 스냅샷과 별도로 유지하면 여러 변경 사항을 만들고 모든 변경 사항을 실행 중인 스냅샷에 함께 커밋할 수 있습니다. 여러 변경 사항을 만드는 프로세스에서 일부 등록 정보 값이 호환되지 않거나 일치하지 않을 수 있지만 실행 중인 스냅샷은 수정되지 않습니다. 변경 작업을 완료하면 새로 고침을 수행합니다.

다음 방법으로 svccfg 명령을 사용할 수 있습니다.

- svccfg editprop 명령을 사용하여 현재 선택된 엔티티의 등록 정보 그룹 및 등록 정보에서 편집기를 호출합니다.
- 명령줄에서 전체 svccfg 명령을 입력하여 setprop 같은 하위 명령을 지정합니다.
- 대화식 세션을 시작하려면 명령줄에 svccfg 또는 svccfg -s *FMRID*만 입력합니다.

- -f 옵션을 지정하여 파일에서 svccfg 명령을 읽습니다.

등록 정보 편집기 호출

다음 예와 같이 svccfg 명령을 호출하면 선택된 엔티티의 등록 정보에서 편집기가 열립니다. svccfg 명령의 이 형식을 사용하면 여러 등록 정보 값을 매우 빠르고 편리하게 수정할 수 있습니다. editprop 하위 명령의 경우 -s 옵션으로 엔티티를 지정해야 합니다.

```
$ svccfg -s pkg/server:s11 editprop
```

지정된 엔티티의 각 등록 정보에 대한 현재 값에 해당하는 setprop 명령 파일이 VISUAL 환경 변수로 지정된 편집기에서 열립니다. VISUAL을 정의하지 않으면 EDITOR로 지정된 편집기가 열립니다. VISUAL 및 EDITOR를 둘 다 정의하지 않으면 등록 정보 파일이 vi에서 열립니다.

파일의 각 행 앞에는 주석 문자가 표시됩니다. svccfg 편집 구성에서 등록 정보 값을 변경하려면 주석 문자를 제거하고 값을 변경한 다음 파일을 저장합니다. 실행 중인 스냅샷에서 등록 정보 값을 변경하려면 refresh 하위 명령인 파일의 마지막 행에서 주석 문자를 제거합니다.

다음 목록은 editprop 하위 명령으로 생성된 파일의 부분적인 예를 보여줍니다.

```
##
## Change property values by removing the leading '#' from the
## appropriate lines and editing the values. svccfg subcommands
## such as delprop can also be added to the script.
##
## Property group "pkg"
## The following properties are defined in the selected instance
## (svc:/application/pkg/server:s11)

# setprop pkg/port = count: 81
# setprop pkg/inst_root = astring: /export/ipsrepos/Solaris11

## The following properties inherit from the parent service
## (svc:/application/pkg/server)

# ...

## Property group "pkg_bui"

# ...

## Property group "pkg_secure"

# ...

## Uncomment to apply these changes to this instance.
# refresh
```

파일 상태의 명령어로 `setprop` 이외에 하위 명령을 추가할 수 있습니다. 예를 들어 `delprop` 명령을 추가할 수 있습니다. `framework` 및 `dependency`와 같은 일부 등록 정보 그룹은 기본적으로 표시되지 않습니다. `editprop -a`를 지정하여 모든 등록 정보를 표시합니다.

이 임시 파일에서 주석 처리되지 않은 명령은 편집 세션을 저장하고 종료할 때 실행됩니다.

대화식으로 또는 파일을 사용하여 `svccfg` 호출

여러 구성 작업을 수행하려면 다음 예와 같이 `svccfg` 명령을 대화식으로 호출하는 것이 편리할 수 있습니다.

```
$ svccfg
svc:> select pkg/server
svc:/application/pkg/server> list
:properties
default
svc:/application/pkg/server> add s11
svc:/application/pkg/server> select s11
svc:/application/pkg/server:s11> setprop pkg/inst_root=/export/ipsrepos/Solaris11
svc:/application/pkg/server:s11> setprop pkg/port=81
svc:/application/pkg/server:s11> unselect
svc:/application/pkg/server> add oss
svc:/application/pkg/server> select oss
svc:/application/pkg/server:oss> setprop pkg/inst_root=/export/ipsrepos/SolarisStudio
svc:/application/pkg/server:oss> setprop pkg/port=82
svc:/application/pkg/server:oss> unselect
svc:/application/pkg/server> list
:properties
default
s11
oss
svc:/application/pkg/server> refresh
svc:/application/pkg/server> select pkg/mirror:default
svc:/application/pkg/mirror:default> listprop config/crontab_period
config/crontab_period astring      "30 2 25 * *"
svc:/application/pkg/mirror:default> setprop config/crontab_period="00 3 25 * *"
svc:/application/pkg/mirror:default> refresh
svc:/application/pkg/mirror:default> exit
$
```

이전 예에서 대화식 프롬프트에 제공된 동일한 명령이 파일에서 제공되고 다음 명령과 같은 명령과 함께 실행될 수도 있습니다.

```
$ svccfg -f cfgpkgrepos
```

등록 정보 값 설정

다음 명령은 등록 정보 값을 설정합니다.

```
svccfg setprop
    등록 정보 값을 변경합니다.
```

```
svccfg addpropvalue
    다중 값 등록 정보에 값을 추가합니다.
```

```
svccfg setenv
    서비스 프로세스 실행 환경에 대한 환경 변수 값을 변경합니다.
```

구성 변경을 실행 중인 스냅샷에 커밋하려면 `svccfg refresh` 명령이나 `svcadm refresh` 명령을 사용해야 합니다.

예 4-1 단순 값 설정

`setprop`를 가장 간단하게 사용하는 방법은 선택된 서비스나 인스턴스에 대해 `pg/name`을 지정하는 것입니다. 여기서 `pg`는 등록 정보 그룹의 이름이고 `name`은 등록 정보 이름이고 같은 기호 뒤에 새 값을 지정합니다. 등록 정보가 이미 있거나 템플릿화된 경우에는 등록 정보 유형을 지정할 필요가 없습니다.

```
$ svccfg -s pkg/server:s11 setprop pkg/port=81
```

예 4-2 콜론 문자가 포함된 값 설정

등록 정보 값이 콜론 문자(:)가 포함되면 유형이 `astring`인 다음 예와 같이 등록 정보 유형을 지정합니다.

```
$ svccfg -s system-repository:default setprop config/http_proxy = astring:
https://proxyURI
```

`listprop` 하위 명령을 사용하여 설정할 등록 정보 유형을 찾습니다.

```
$ svccfg -s system-repository:default listprop config/http_proxy
config/http_proxy astring
```

예 4-3 포함된 공백이 들어 있는 값 설정

큰따옴표를 사용하여 포함된 공백이 들어 있는 값을 설정합니다. 셀에 따라 큰따옴표로 묶인 문자열을 작은따옴표로 묶어야 할 수 있습니다.

```
$ svccfg -s pkg/mirror setprop config/crontab_period = "00 3 25 * *"
$ svccfg -s pkg/mirror setprop config/crontab_period = "'00 3 25 * *'"
```

따옴표를 사용하여 큰따옴표나 백슬래시 문자가 포함된 값을 설정하고, 백슬래시 문자를 사용하여 큰따옴표나 백슬래시 문자를 제어합니다.

예 4-4 값 세트인 값 설정

괄호를 사용하여 값 세트를 단일 값으로 지정합니다. 또한 셀에 따라 값 세트를 작은따옴표로 묶어야 할 수 있습니다.

```
$ svccfg -s dns/client setprop config/nameserver = (10.0.0.1 192.168.0.1)
$ svccfg -s dns/client setprop config/nameserver = '(10.0.0.1 192.168.0.1)'
$ svccfg -s dns/client listprop config/nameserver
config/nameserver net_address 10.0.0.1 192.168.0.1
```

describe 하위 명령을 사용하여 값 세트에서 허용되는 값 개수를 찾습니다.

```
$ svccfg -s dns/client describe -v config/nameserver
config/nameserver net_address 10.0.0.1 192.168.0.1
  type: net_address
  required: false
  The IP address of a DNS nameserver to be used by the resolver.
  visibility: readwrite
  minimum number of values: 1
  maximum number of values: 3
  value: 10.0.0.1
  value: 192.168.0.1
```

예 4-5 값 추가

addpropvalue 하위 명령을 사용하여 선택된 서비스나 서비스 인스턴스의 지정된 등록 정보에 제공된 값을 추가합니다. 새 값은 등록 정보에 대한 기존 등록 정보 값 목록의 끝에 추가됩니다.

```
$ svcprop -p keymap/layout keymap:default
US-English
$ svccfg -s keymap:default addpropvalue keymap:layout UK-English
$ svccfg -s keymap:default listprop keymap:layout
keymap/layout astring "US-English" "UK-English"
```

이전 setprop 예에서는 값 세트의 모든 값을 한 번에 지정해야 합니다. 값을 하나만 지정하면 해당 값이 단일 값의 새 세트가 됩니다. 이 addpropvalue 예에서 추가된 값은 고유합니다. 이 추가된 값에 액세스하려면 libscf 함수 scf_iter_property_values()를 사용하여 값에 걸쳐 반복해야 합니다. listprop는 값을 둘 다 나열하지만 describe는 첫번째 값만 나열하고 이 등록 정보에 대한 최대 허용 값 개수가 1개임을 보고합니다.

```
$ svccfg -s keymap:default describe -v keymap/layout
keymap/layout astring US-English
  type: astring
  required: true
  The keyboard layout
  visibility: readwrite
```

```

    minimum number of values: 1
    maximum number of values: 1
    value: US-English

```

▼ ttymon 등록 정보 값을 수정하는 방법

이 절차에서는 ttymon에 전달되는 매개변수를 수정하는 방법을 보여줍니다.

1. 수정할 서비스를 식별합니다.

ttymon(1M) 매뉴얼 페이지에는 수정할 서비스가 `svc:/system/console-login`이라고 설명됩니다. **ttymon(1M)** 매뉴얼 페이지에는 ttymon 등록 정보 그룹의 등록 정보에 대한 설명도 있습니다.

다음 명령은 이 이미지에 있는 `console-login` 서비스의 여러 인스턴스를 보여주고 default 인스턴스가 현재 온라인 상태의 유일한 인스턴스임을 보여줍니다.

```

$ svcs console-login
STATE          STIME      FMRI
disabled       10:49:43   svc:/system/console-login:terma
disabled       10:49:43   svc:/system/console-login:termb
online         10:50:54   svc:/system/console-login:default

```

2. 수정할 등록 정보를 식별합니다.

다음 명령은 default 인스턴스의 ttymon 등록 정보 그룹에서 각 등록 정보의 이름, 데이터 유형, 값 및 간단한 설명을 보여줍니다.

```

$ svccfg -s console-login:default describe ttymon
ttymon          application
ttymon/device   astring        /dev/console
    The terminal device to be used for the console login prompt.
ttymon/terminal_type astring
    Sets the initial value of the TERM environment variable

```

이전 출력에는 `terminal_type` 등록 정보에 대한 값이 표시되지 않습니다. 다음 명령은 `console-login:default` 인스턴스의 `ttymon/terminal_type` 등록 정보 값이 현재 null임을 확인합니다.

```

$ svcprop -p ttymon/terminal_type console-login:default
""

```

3. 등록 정보 값을 수정합니다.

다음 명령을 입력하여 `console-login:default` 인스턴스의 `ttymon/terminal_type` 등록 정보 값을 `xterm`으로 변경합니다.

```

$ svccfg -s system/console-login:default setprop ttymon/terminal_type=xterm

```

4. 실행 중인 스냅샷에 새 값 커밋

다음 출력은 `terminal_type` 등록 정보 값이 편집 구성에서 변경되었지만 실행 중인 스냅샷에서 변경되지 않았음을 보여줍니다.

```
$ svccfg -s console-login:default listprop ttymon/terminal_type
ttymon/terminal_type astring      xterm
$ svcprop -p ttymon/terminal_type console-login:default
""
```

서비스 인스턴스를 새로 고친 후 등록 정보 값이 실행 중인 스냅샷에서 변경됩니다.

```
$ svcadm refresh console-login:default
$ svcprop -p ttymon/terminal_type console-login:default
xterm
```

▼ 서비스 프로세스 환경에 대한 환경 변수를 수정하는 방법

이 절차는 서비스에서 시작한 프로세스가 실행될 환경에서 환경 변수 값을 설정하는 방법을 설명합니다.

이 절차의 예에서는 디버깅을 위해 `cron` 환경 변수를 수정하는 방법을 보여줍니다.

1. 서비스가 실행 중인지 확인합니다.

다음 출력은 `cron` 서비스가 온라인 상태이고 `cron` 프로세스가 실행 중임을 보여줍니다.

```
$ svcs -p cron
STATE      STIME      FMRI
online     10:24:05  svc:/system/cron:default
           10:24:05   1089 cron
```

2. 환경 변수를 설정합니다.

`setenv` 하위 명령은 서비스나 서비스 인스턴스에서 시작한 프로세스가 실행될 환경에 대한 환경 변수를 설정합니다.

다음 명령을 사용하여 설정할 환경 변수의 현재 값을 확인합니다.

```
$ pargs -e `pgrep -f /usr/sbin/cron`
```

이 예에서 설정되는 환경 변수에는 현재 값이 없습니다.

다음 명령은 `svc:/system/cron:default` 서비스 인스턴스에서 시작한 `/usr/sbin/cron` 프로세스에 대한 `UMEM_DEBUG` 및 `LD_PRELOAD` 환경 변수를 설정합니다.

```
$ svccfg -s system/cron:default setenv UMEM_DEBUG default
$ svccfg -s system/cron:default setenv LD_PRELOAD libumem.so
```

3. 서비스를 새로 고치고 다시 시작합니다.

환경 변수 값을 변경하려면 새로 고치고 다시 시작해야 합니다.

```
$ svcadm refresh system/cron:default
$ svcadm restart system/cron:default
```

4. 변경 작업이 수행되었는지 확인합니다.

다음 출력은 서비스가 다시 시작되었고 프로세스에 새 프로세스 ID가 포함되고 해당 프로세스 환경에 대한 두 개 환경 변수가 설정되었음을 보여줍니다.

```
$ svcs -p cron
STATE      STIME      FMRI
online     9:24:39   svc:/system/cron:default
           9:24:39   5601 cron

$ svcprop -g method -p environment system/cron:default
start/environment astring LD_PRELOAD=libumem.so UMEM_DEBUG=default
$ pargs -e `pgrep -f /usr/sbin/cron`
5601: /usr/sbin/cron
envp[0]: LOGNAME=root
envp[1]: LD_PRELOAD=libumem.so
envp[2]: PATH=/usr/sbin:/usr/bin
envp[3]: SMF_FMRI=svc:/system/cron:default
envp[4]: SMF_METHOD=start
envp[5]: SMF_RESTARTER=svc:/system/svc/restarter:default
envp[6]: SMF_ZONENAME=global
envp[7]: UMEM_DEBUG=default
```

참조 unsetenv 하위 명령은 서비스나 서비스 인스턴스에서 시작한 프로세스에 대한 환경 변수를 설정 해제합니다.

등록 정보 그룹, 등록 정보 및 등록 정보 값 추가

다음 명령은 등록 정보와 등록 정보 그룹을 추가합니다.

```
svccfg setprop
svccfg addpropvalue
```

등록 정보가 없는 경우 값이 설정되고 있는 등록 정보를 추가합니다.

```
svccfg addpg
```

서비스 또는 서비스 인스턴스에 새 등록 정보 그룹을 추가합니다.

구성 변경을 실행 중인 스냅샷에 커밋하려면 svccfg refresh 명령이나 svcadm refresh 명령을 사용해야 합니다.

예 4-6 addpg를 사용하여 새 등록 정보 그룹 만들기

addpg 하위 명령을 사용하여 등록 정보 그룹을 선택된 서비스나 서비스 인스턴스에 추가합니다.

```
svccfg -s FMRI addpg name type [flags]
```

type 규약에 의하면 *type* 값은 대개 application입니다. 등록 정보 유형에 대한 자세한 내용은 5장. SMF를 사용하여 응용 프로그램 제어를 참조하십시오.

flags *flags* 값으로 P를 지정하여 등록 정보와 추가된 등록 정보를 비영구 등록 정보로 저장합니다. P를 지정하면 이 등록 정보 그룹과 포함된 등록 정보가 재부트 시 자동으로 제거됩니다. P 값은 SCF_PG_FLAG_NONPERSISTENT에 대한 별칭입니다. scf_service_add_pg(3SCF) 매뉴얼 페이지를 참조하십시오.

```
$ svccfg -s svc:/site/my-svc addpg config application
$ svccfg -s my-svc listprop config
config application
$ svccfg -s my-svc:default listprop config
$
```

이 예에서 관리자는 config 등록 정보 그룹을 상위 서비스 my-svc에 추가했지만 인스턴스 my-svc:default에는 추가하지 않았습니다. listprop 명령은 config 등록 정보 그룹이 서비스 인스턴스에 존재하지 않음을 보여줍니다.

예 4-7 setprop를 사용하여 새 등록 정보 만들기

setprop 하위 명령을 사용하여 “등록 정보 값 설정” [62]에 설명된 대로 등록 정보 값을 설정합니다. 등록 정보 그룹이 선택된 인스턴스나 인스턴스 서비스에 존재하지 않을 경우 템플릿 정의에 유형과 플래그가 있으면 등록 정보 그룹이 생성됩니다. 등록 정보가 선택된 인스턴스나 인스턴스 서비스에 존재하지 않을 경우 등록 정보 *type*을 지정해야 합니다.

```
$ svccfg -s my-svc:default setprop config/vendor = astring: vendora
$ svccfg -s my-svc:default listprop config/vendor
config/vendor astring       vendora
```

예 4-8 addpropvalue를 사용하여 새 등록 정보 만들기

addpropvalue 하위 명령을 사용하여 “등록 정보 값 설정” [62]에 설명된 대로 등록 정보 값을 추가합니다. 등록 정보 그룹이 선택된 인스턴스나 인스턴스 서비스에 존재하지 않을 경우 템플릿 정의에 유형과 플래그가 있으면 등록 정보 그룹이 생성됩니다. 등록 정보가 선택된 인스턴스나 인스턴스 서비스에 존재하지 않을 경우 등록 정보 *type*을 지정해야 합니다.

```
$ svccfg -s my-svc:default addpropvalue config/vendor astring: vendorb
$ -s my-svc:default addpropvalue config/customer astring: acustomer
$ svccfg -s my-svc:default listprop config
config           application
config/vendor   astring       "vendora" "vendorb"
config/customer astring       acustomer
```

등록 정보 그룹, 등록 정보 및 등록 정보 값 삭제

다음 명령은 등록 정보 값, 등록 정보 및 등록 정보 그룹을 삭제합니다.

```
svccfg setprop
```

모든 등록 정보 값을 삭제합니다.

```
svccfg delpropvalue
```

지정된 패턴과 일치하는 지정된 등록 정보의 모든 값을 삭제합니다.

```
svccfg delprop
```

등록 정보를 삭제합니다.

```
svccfg delpg
```

등록 정보 그룹을 삭제합니다.

```
svccfg delcust
```

관리 사용자 정의를 삭제합니다.

구성 변경을 실행 중인 스냅샷에 커밋하려면 `svccfg refresh` 명령이나 `svcadm refresh` 명령을 사용해야 합니다.

관리 구성 삭제

`svccfg` 명령이나 `libscf` 호출을 사용하여 구성을 수정하면 서비스 구성 저장소의 `admin` 계층만 수정됩니다. 계층에 대한 자세한 내용은 [“저장소 계층” \[24\]](#)을 참조하십시오. `admin` 계층에만 정의되고 기타 계층에 없는 구성을 삭제하면 해당 구성이 사라집니다. `-l` 옵션을 사용하여 서비스 구성 저장소의 모든 계층을 표시할 때도 구성을 표시하는 명령이 더는 삭제된 구성을 표시하지 않습니다. 다른 계층에 있는 구성을 삭제하는 방법에 대한 자세한 내용은 [“비관리 구성 삭제” \[70\]](#)를 참조하십시오.

예 4-9 모든 등록 정보 값 삭제

[“등록 정보 값 설정” \[62\]](#)에 설명된 대로 `setprop` 하위 명령을 사용합니다. 모든 등록 정보 값을 삭제하려면 유형 또는 값을 지정하지 마십시오. 값이 삭제되지만 등록 정보는 계속 존재합니다.

```
$ svccfg -s my-svc:default setprop config/vendor =
$ svccfg -s my-svc:default listprop config/vendor
config/vendor  astring
```

예 4-10 모든 일치하는 등록 정보 값 삭제

delpropvalue 하위 명령을 사용하여 지정된 패턴과 일치하는 이름 지정된 등록 정보의 모든 값을 삭제합니다.

```
$ svccfg -s my-svc:default setprop config/tool = astring: '(hammer tongs wrench)'
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring    acustomer
config/vendor   astring    "vendora" "vendorb"
config/tool     astring    "hammer tongs wrench"
$ svccfg -s my-svc:default delpropvalue config/vendor '*b'
$ svccfg -s my-svc:default delpropvalue config/tool 'tong*'
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring    acustomer
config/vendor   astring    vendora
config/tool     astring    "hammer tongs wrench"
$ # config/tool is a single value that is a value set
$ svccfg -s my-svc:default delpropvalue config/tool '*tong*'
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring    acustomer
config/vendor   astring    vendora
config/tool     astring
```

예 4-11 등록 정보 삭제

delprop 하위 명령을 사용하여 선택된 서비스나 서비스 인스턴스의 이름 지정된 등록 정보를 삭제합니다.

```
$ svccfg -s my-svc:default delprop config/tool
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring    acustomer
config/vendor   astring    vendora
```

예 4-12 등록 정보 그룹 삭제

delpg 및 delprop 하위 명령은 둘 다 등록 정보 그룹을 삭제할 수 있습니다. delpg 하위 명령은 선택된 서비스나 서비스 인스턴스의 이름 지정된 등록 정보 그룹을 삭제합니다. delprop 하위 명령은 이름 지정된 등록 정보가 없으면 이름 지정된 등록 정보 그룹을 삭제합니다.

```
$ svccfg -s my-svc:default delpg config
$ svccfg -s my-svc:default listprop config
$
```

예 4-13 사용자 정의 삭제

delcust 하위 명령은 선택된 서비스나 서비스 인스턴스에서 관리 사용자 정의를 삭제합니다. delcust 하위 명령을 사용하기 전에 listcust 하위 명령을 같은 패턴이나 옵션과 함께 사용하여 삭제될 항목을 확인합니다. 지정되는 패턴은 등록 정보 또는 등록 정보 그룹과 일치해야 합니다.

```
$ svccfg -s my-svc:default listcust
config                application admin
config/customer      astring    admin      acustomer
config/vendor        astring    admin      "vendora" "vendorb"
config/tool           astring    admin      "hammer tongs wrench"
$ svccfg -s my-svc:default listcust '*tool'
config/tool           astring    admin      "hammer tongs wrench"
$ svccfg -s my-svc:default delcust '*tool'
Deleting customizations for property: config/tool
$ svccfg -s my-svc:default listcust '*tool'
$ svccfg -s my-svc:default listcust
config                application admin
config/customer      astring    admin      acustomer
config/vendor        astring    admin      "vendora" "vendorb"
```

비관리 구성 삭제

서비스 구성 저장소의 site-profile, system-profile 및 manifest 계층에 존재하는 구성은 서비스 매니페스트 및 프로파일 파일에 정의됩니다. 계층에 대한 자세한 내용은 “저장소 계층” [24]을 참조하십시오. SMF는 서비스 구성 저장소를 파일 시스템 콘텐츠와 동기화된 상태로 유지합니다. 표준 위치의 매니페스트 또는 프로파일 파일에 정의된 모든 구성은 삭제된 후를 비롯하여 관리 사용자 정의 후에 파일 시스템에 계속 존재하고 서비스 구성 저장소에도 저장되어 있습니다. 매니페스트 또는 프로파일 파일에 정의된 구성에는 번들 지원이 포함된다 합니다. 번들 지원이 포함된 구성을 삭제하면 정보가 파일 시스템에서 삭제되지 않지만, 기본 보기에 표시되지 않도록 마스킹됩니다. 마스킹된 엔티티에 대한 설명은 smf(5) 매뉴얼 페이지를 참조하십시오.

번들 지원이 포함된 구성을 삭제하는 작업은 관리 사용자 정의입니다. 이 경우 delcust 하위 명령은 아무것도 삭제하지 않고 구성을 마스킹 해제합니다. listcust -M 하위 명령을 사용하여 마스킹된 구성을 표시합니다. delcust -M 하위 명령을 사용하여 구성을 마스킹 해제하거나 구성의 삭제 또는 마스킹을 실행 취소합니다.

예 4-14 번들 지원이 포함된 구성 삭제

“관리 구성 삭제” [68]에서 my-svc 서비스의 config 등록 정보 그룹은 admin 계층에만 존재했습니다. 매니페스트나 프로파일에는 config 등록 정보 그룹이 존재하지 않았습니다. 해당 등록 정보를 삭제하면 시스템에서 사라집니다. 이 예에서는 번들 지원이 포함된 구성을 삭제할 때 다른 결과를 보여줍니다.

등록 정보가 서비스 매니페스트에 정의됩니다.

```
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring    admin          /export/ipsrepos/Solaris11
pkg/inst_root astring    manifest      /var/pkgrepo
$ svccfg -s pkg/server delprop pkg/inst_root
```

삭제한 후 listprop를 옵션 없이 사용하면 등록 정보가 표시되지 않습니다. 등록 정보에 변
들 지원이 포함되므로 등록 정보는 서비스 구성 저장소에 계속 존재하고 -l 또는 -M 옵션을
listprop 하위 명령과 함께 사용하여 표시할 수 있습니다.

```
$ svccfg -s pkg/server listprop pkg/inst_root
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring    admin          MASKED /export/ipsrepos/Solaris11
pkg/inst_root astring    manifest      MASKED /var/pkgrepo
$ svccfg -s pkg/server listcust -M
pkg/inst_root astring    admin          MASKED /export/ipsrepos/Solaris11
```

예 4-15 구성 마스킹 해제

등록 정보를 마스킹 해제하면 두 사용자 정의가 모두 사라집니다.

- 등록 정보가 더는 마스킹되거나 숨겨지지 않습니다.
- 등록 정보에 더는 사용자 정의 값이 포함되지 않습니다.

```
$ svccfg -s pkg/server delcust -M
Deleting customizations for property: pkg/inst_root
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring    manifest      /var/pkgrepo
$ svccfg -s pkg/server listprop pkg/inst_root
pkg/inst_root astring    /var/pkgrepo
```

서비스 인스턴스 추가

서비스 인스턴스를 사용하여 서비스의 여러 구성을 동시에 실행할 수 있습니다. 서비스 인스
턴스는 일반적인 서비스 구성을 상속하고 사용자 정의합니다.

add 하위 명령과 지정된 이름을 사용하여 새 엔티티를 선택된 서비스의 하위로 만듭니다.

```
$ svcs -Ho inst pkg/server
default
$ svccfg -s pkg/server add s11
$ svcs -Ho inst pkg/server
default
s11
```

스냅샷 되돌리기

다음 작업은 각각 새로운 실행 중인 스냅샷을 만듭니다.

- `svcadm restart manifest-import`
- `svcadm refresh`
- `svccfg refresh`

`revert` 하위 명령은 `-s` 옵션으로 지정된 인스턴스의 관리 사용자 정의(admin 계층)와 해당 서비스를 이름 지정된 스냅샷이나 현재 선택된 스냅샷에 기록된 값으로 되돌립니다. `listsnap` 하위 명령을 사용하여 이 서비스 인스턴스의 가능한 스냅샷 목록을 표시합니다. `selectsnap` 하위 명령을 사용하여 대화식 모드에서 스냅샷을 선택합니다.

```
$ svcprop -p pkg/inst_root pkg/server:default
pkg/inst_root astring /export/ipsrepos/Solaris11
$ svccfg -s pkg/server:default listsnap
initial
previous
running
start
$ svcprop -s previous -p pkg/inst_root pkg/server:default
pkg/inst_root astring /var/pkgrepo
```

`revert` 하위 명령이 모든 관리 사용자 정의를 되돌리므로 모든 관리 사용자 정의를 나열하고 되돌리기 전에 값을 검토하십시오.

```
$ svcprop -s previous -l admin pkg/server:default
pkg/inst_root astring /var/pkgrepo
$ svccfg -s pkg/server:default revert previous
$ svcadm refresh pkg/server:default
$ svcprop -p pkg/inst_root pkg/server:default
pkg/inst_root astring /var/pkgrepo
```

매니페스트와 프로파일 가져오기 및 적용

`manifest-import` 서비스를 다시 시작하면 표준 위치의 매니페스트를 가져오고 표준 위치의 새롭거나 변경된 프로파일이 적용됩니다. 매니페스트 및 프로파일 표준 위치에 대해서는 “[서비스 번들](#)” [23]을 참조하십시오. 매니페스트를 가져오거나 프로파일을 적용한 결과로 서비스가 시작되거나 중지되면 존재하는 적절한 메소드가 실행됩니다.

표준 위치의 파일을 `svccfg import` 명령으로 지정하면 `manifest-import` 서비스가 다시 시작됩니다.

권장 모범 사례는 매니페스트 및 프로파일 파일을 표준 위치에 저장하고 `svccfg import` 또는 `svccfg apply` 명령을 사용하는 대신 `manifest-import` 서비스를 다시 시작합니다.

```
$ svcadm restart manifest-import
```

manifest-import 서비스를 다시 시작하면 표준 위치에 있는 프로파일과 매니페스트의 구성이 영향을 받는 인스턴스의 manifest, system-profile 또는 site-profile 계층에 적용되고, 영향을 받는 인스턴스가 새로 고쳐지고 검증되며, 새 스냅샷이 생성됩니다.

비표준 위치의 프로파일과 매니페스트를 가져오거나 적용하면 구성이 영향을 받는 인스턴스의 admin 계층에 적용됩니다. 기본 또는 초기 구성 제공에는 비표준 위치를 사용하는 것이 좋습니다. 많은 구성을 변경하려면 비표준 위치에서 가져오거나 적용하는 방법이 많은 명령을 실행하는 방법보다 더 간편할 수 있지만 manifest-import 서비스의 자동화된 관리 방식이라는 장점이 사라집니다. 서비스 제공을 관리하려면 manifest-import 서비스에 알려진 위치와 예상된 상태가 필요합니다.

여러 시스템 구성

여러 시스템에서 같은 구성을 구현하려면 사용으로 설정할 서비스와 등록 정보 값을 지정하는 SMF 프로파일을 만들고 해당 프로파일을 각 시스템의 site 디렉토리에 저장합니다. “[저장소 계층](#)” [24]의 설명대로 로컬 사용자 정의는 시스템이 설치될 때 제공된 값보다 우선 적용되고 site-profile 계층의 사용자 정의는 관리자나 응용 프로그램이 만든 사용자 정의에만 적용됩니다.

프로파일은 기존 서비스와 인스턴스에 대한 등록 정보를 추가 및 설정하고 새 서비스 인스턴스를 지정합니다. 프로파일은 매니페스트가 지정할 수 있는 거의 모든 항목을 지정할 수 있습니다.

다음 메소드의 하나를 사용하여 사이트 프로파일을 만듭니다.

- svcbundle 명령을 bundle-type=profile과 함께 사용하여 새 프로파일 파일을 만듭니다.
- svccfg extract 명령을 사용하여 기존 서비스에서 프로파일 정보를 캡처합니다.

프로파일 파일에서 등록 정보 값을 사용자 정의하고 각 사용자 정의 이유에 대한 주석을 포함합니다. /etc/svc/profile/site에 파일을 복사하고 manifest-import 서비스를 다시 시작합니다.

▼ svcbundle을 사용하여 프로파일을 만드는 방법

svc:/system/rmtmpfiles 서비스는 부트 시 /tmp 디렉토리를 정리하는 역할을 수행합니다. 기본적으로 rmtmpfiles 서비스는 /var/tmp를 정리하지 않습니다. 부트 프로세스 중에 /var/tmp를 정리하려면 options/clean_vartmp 등록 정보를 true로 설정하여 svc:/system/rmtmpfiles 서비스의 동작을 변경합니다. 여러 시스템에서 동작을 이와 같이 변경하는 가장 간편한 방법은 프로파일을 만들고 각 시스템의 /etc/svc/profile/site에 저장하는 것입니다.

1. **프로파일을 만듭니다.**
다음 명령은 /tmp/rmtmpfiles.xml에서 새 프로파일을 만듭니다.

```
$ svcbundle -o /tmp/rmtmpfiles.xml -s service-name=system/rmtmpfiles \  
-s bundle-type=profile -s service-property=options:clean_vartmp:boolean:true
```
2. **필요에 따라 프로파일을 변경합니다.**
3. **프로파일을 올바른 디렉토리로 복사합니다.**

```
$ cp /tmp/rmtmpfiles.xml /etc/svc/profile/site/rmtmpfiles.xml
```
4. **매니페스트 가져오기 서비스를 다시 시작하여 프로파일을 시스템에 적용합니다.**

```
$ svcadm restart manifest-import
```

예 4-16 svcbundle을 사용하여 프로파일 자동 설치

새 프로파일을 변경할 필요가 없으면 -i 옵션을 사용하여 생성된 직후 프로파일을 설치합니다. svcbundle 명령은 프로파일을 /etc/svc/profile/site/rmtmpfiles.xml에 쓰고 manifest-import 서비스를 다시 시작합니다. /etc/svc/profile/site 디렉토리에서 이름이 같은 기존 파일을 덮어씁니다.

```
# svcbundle -i -s service-name=system/rmtmpfiles \  
-s bundle-type=profile -s service-property=options:clean_vartmp:boolean:true
```

▼ svccfg를 사용하여 프로파일을 만드는 방법

1. **프로파일을 만듭니다.**
svccfg extract 명령은 admin 및 site-profile 계층의 지정된 서비스 또는 인스턴스에 대한 서비스 프로파일을 표시합니다. 다른 계층에서 값을 추출하려면 -l 옵션을 사용합니다.

```
$ svccfg extract -l current network/dns/client > dnsclientprofile.xml
```
2. **필요에 따라 프로파일을 변경합니다.**
프로파일 이름을 의미 있는 이름으로 변경합니다. 다음 예와 같이 이름은 기본적으로 extract로 설정됩니다.

```
<service_bundle type='profile' name='extract'>
```
3. **프로파일을 올바른 디렉토리로 복사합니다.**

```
$ cp dnsclientprofile.xml /etc/svc/profile/site/dnsclientprofile.xml
```
4. **매니페스트 가져오기 서비스를 다시 시작하여 프로파일을 시스템에 적용합니다.**

```
$ svcadm restart manifest-import
```

inetd로 제어되는 서비스 수정

inetd로 제어되는 서비스는 `inetd.conf` 파일의 구성에서 변환된 SMF 서비스입니다. `inetd` 명령은 이러한 서비스에 대한 위임된 다시 시작 프로그램입니다.

다음 절차는 `inetd`로 제어되는 서비스의 등록 정보 값을 변경하는 방법을 보여줍니다.

수정할 서비스가 `inetd`로 제어되는지 확인하려면 `inetadm` 명령을 옵션이나 인수 없이 호출하여 모든 `inetd`로 제어되는 서비스를 나열합니다. 다음 예에서는 부분적인 목록만 보여줍니다.

```
$ inetadm
ENABLED STATE FMRI
enabled online svc:/application/cups/in-lpd:default
...
disabled disabled svc:/application/x11/xfs:default
```

`inetadm` 명령의 `-l` 옵션은 `inetd`로 제어되는 서비스의 모든 등록 정보를 나열합니다. 다음 예에서 오류 메시지는 지정된 서비스가 `inetd`로 제어되는 서비스가 아님을 나타냅니다. "No restarter property(다시 시작 프로그램 등록 정보 없음)"는 마스터 다시 시작 프로그램 `svc.startd`가 서비스 인스턴스를 관리함을 의미합니다.

```
$ inetadm -l ssh
Error: Specified service instance "svc:/network/ssh:default" has no
restarter property. inetd is not the delegated restarter of this instance.
```

마찬가지로 다음 예에서 "Couldn't find property 'general/restarter'(등록 정보 'general/restarter'를 찾을 수 없음)"는 기본 다시 시작 프로그램 `svc.startd`가 서비스 인스턴스를 관리함을 나타냅니다.

```
$ svcprop -p general/restarter ssh
svcprop: Couldn't find property 'general/restarter' for instance
'svc:/network/ssh:default'.
```

서비스가 `inetd`로 제어되면 해당 다시 시작 프로그램은 다음 예와 같이 `inetd`입니다.

```
$ svcprop -p general/restarter xfs
svc:/network/inetd:default
```

▼ inetd 제어 서비스에 대한 등록 정보 값을 변경하는 방법

1. 서비스에 대한 등록 정보를 나열합니다.

`inetadm` 명령의 `-l` 옵션을 사용하여 지정된 서비스의 모든 등록 정보를 나열합니다. 등록 정보의 현재 값을 검사합니다.

```
# inetadm -l FMRI
```

2. 등록 정보 값을 변경합니다.

inetadm 명령의 -m 옵션을 사용하여 지정된 등록 정보의 값을 변경합니다. 서비스 등록 정보에 대한 특정 정보는 해당 서비스와 연관된 매뉴얼 페이지에서 다룹니다.

```
# inetadm -m FMRI property-name=value
```

등록 정보 값을 삭제하려면 빈 값을 지정합니다.

```
$ inetadm -m svc property=""
```

3. 등록 정보 값이 변경되었는지 확인합니다.

다시 등록 정보를 나열하여 올바르게 변경되었는지 확인합니다.

```
# inetadm -l FMRI
```

4. 변경 사항이 적용되었는지 확인합니다.

등록 정보 변경이 예상대로 적용되는지 확인합니다.

예 4-17 inetd 제어 서비스가 시작될 때 실행할 명령 수정

이 예에서는 inetd로 제어되는 서비스의 명령줄에 옵션이나 인수를 추가하거나 제거하는 방법을 보여줍니다. 서비스가 시작될 때 실행되는 명령은 exec 등록 정보 값입니다.

exec 등록 정보의 현재 값을 검사할 수 있도록 inetadm 명령의 -l 옵션을 사용하여 지정된 서비스의 모든 등록 정보를 나열합니다. 이 예에서는 X 창 시스템 글꼴 서버인 svc:/application/x11/xfps 서비스를 보여줍니다. 자세한 내용은 xfs(1) 매뉴얼 페이지를 참조하십시오.

```
$ inetadm -l xfs | grep exec
exec="/usr/bin/xfps -inetd"
```

inetadm 명령의 -m 옵션을 사용하여 지정된 서비스의 exec 등록 정보 값을 변경합니다.

```
$ inetadm -m xfs exec="/usr/bin/xfps -inetd -config /opt/site/fs/config"
```

등록 정보 값이 변경되었는지 확인합니다.

```
$ inetadm -l xfs | grep exec
exec="/usr/bin/xfps -inetd -config /opt/site/fs/config"
```

파일에 의해 구성되는 서비스 수정

inetd로 관리되는 몇몇 SMF 서비스는 서비스 등록 정보가 아니라 파일에서 해당 구성의 일부를 가져옵니다. 이 구성을 수정하려면 구성 파일을 편집하고 SMF 명령을 사용하여 서비스를 다시 시작합니다. 이러한 구성 파일은 서비스가 실행되는 동안 변경할 수 있지만 파일의 내용은 서비스가 시작될 때만 읽습니다.

구성 파일을 직접 편집하기 전에 다음 조건을 확인하십시오.

- 구성 파일에 직접 편집하지 않게 알리는 메시지가 포함되지 않았는지 확인합니다.
- 서비스에 configfile 유형의 등록 정보 그룹이 포함되지 않았는지 확인합니다.

```
$ svcprop -g configfile network/ntp
```

서비스에 configfile 유형의 등록 정보 그룹이 포함되면 구성 파일이 아니라 해당 등록 정보 그룹의 등록 정보를 수정합니다. [“스텐실을 사용하여 구성 파일 만들기” \[89\]](#)를 참조하십시오.

예를 들어 새 NTP 서버를 추가하여 NTP 클라이언트를 지원하려면 서버에 대한 새 항목을 /etc/inet/ntp.conf 파일에 추가하고 다음 명령에 표시된 대로 NTP 서비스를 다시 시작합니다.

```
$ svcadm restart svc:/network/ntp:default
```

IKEv2를 사용으로 설정하려면 /etc/inet/ike/ikev2.config 파일을 수정하여 IKEv2 데몬을 구성하고 다음 명령에 표시된 대로 IKEv2 서비스를 사용으로 설정합니다. ikev2.config 파일을 편집하려면 네트워크 IPsec 관리 프로파일이 지정된 사용자로 pfedit 명령을 사용합니다. 이 방법으로 파일을 편집하면 올바른 파일 소유권이 유지됩니다. pfedit 사용에 대한 자세한 내용은 [pfedit\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
$ svcadm enable svc:/network/ipsec/ike:ikev2
```


◆◆◆ 5 장

SMF를 사용하여 응용 프로그램 제어

이 장에서는 등록 정보를 읽는 데 `libscf` 라이브러리 인터페이스를 사용할 수 없는 경우 `stencil`을 사용하여 구성 파일을 만드는 방법을 포함하여 응용 프로그램에 대한 서비스 지원을 제공하는 방법을 설명합니다.

응용 프로그램 구성과 같은 작업을 수행하려고 서비스를 만들고 제공하는 추가적인 예는 “Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”의 7 장, “Automating System Change as Part of Package Installation” 및 “Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”의 8 장, “Advanced Topics For Package Updating”을 참조하십시오.

SMF 서비스 만들기

서비스 매니페스트에는 서비스가 시작되고 중지될 때 실행할 인스턴스, 종속성, 스크립트를 포함하여 특정 서비스와 관련된 전체 등록 정보 세트 및 기본 응용 프로그램 등록 정보 값이 포함됩니다. 매니페스트는 서비스 설명과 같은 템플릿 정보도 제공합니다. SMF 매니페스트의 콘텐츠와 형식에 대한 자세한 내용은 [service_bundle\(4\)](#) 매뉴얼 페이지 및 `/usr/share/lib/xml/dtd/service_bundle.dtd.1` 서비스 번들 DTD를 참조하십시오. 규칙 이름 지정 및 등록 정보 그룹 유형 지정에 대한 자세한 내용은 “서비스, 인스턴스, 등록 정보 그룹 및 등록 정보 이름 지정” [87]을 참조하십시오.

`svcbundle` 서비스 번들 생성기 도구는 간단한 서비스를 만들거나 더 복잡한 서비스를 시작하는 좋은 방법입니다. 자세한 내용은 [svcbundle\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 서비스 번들 DTD 및 기타 서비스 매니페스트를 사용하여 더 복잡한 서비스를 완료할 수 있습니다.

사용자 정의 매니페스트에 대한 표준 위치는 `/lib/svc/manifest/site`입니다. 이 위치에 저장된 매니페스트는 서비스가 시작되기 전 부트 프로세스 중에 `svc:/system/early-manifest-import:default` 서비스를 통해 서비스 구성 저장소로 가져옵니다. 가져오기 프로세스가 일찍 실행되면 저장소는 서비스가 시작되기 전에 최신 매니페스트의 정보를 포함하게 됩니다. 이 표준 위치에 저장된 매니페스트는 `svc:/system/manifest-import` 서비스가 다시 시작될 때도 가져옵니다.

여러 매니페스트를 사용하여 단일 서비스를 설명할 수 있습니다. 이 기능은 예를 들어, 서비스의 기존 매니페스트를 수정하지 않고 서비스의 새 인스턴스를 정의하는 데 유용할 수 있습니다.

니다. 하지만 여러 매니페스트가 동일한 서비스 또는 인스턴스에 대한 동일한 계층의 동일한 등록 정보를 정의할 경우 SMF는 사용할 값을 결정할 수 없습니다. 이 유형의 충돌이 감지된 경우 인스턴스는 유지 관리 상태가 됩니다. 계층에 대한 자세한 내용은 “저장소 계층” [24]을 참조하십시오.

해당 사용자가 `svcs` 및 `svccfg describe` 명령에서 서비스에 대한 정보를 얻을 수 있도록 매니페스트에 이름 및 설명 메타 데이터를 추가합니다. 등록 정보 값의 설명도 추가할 수 있습니다. DTD에서 `value`, `values` 및 `template` 요소를 참조하십시오.

`svccfg validate` 명령을 사용하여 서비스 매니페스트 파일이나 서비스 인스턴스 FMRI를 검증합니다. 표준 위치의 매니페스트, 메소드 및 프로파일 파일을 사용하여 `manifest-import` 서비스를 다시 시작하고 서비스 인스턴스를 설치 및 구성합니다. `svcs` 명령을 사용하여 서비스 인스턴스의 상태를 확인합니다.

이 섹션에서는 사용자 정의 SMF 서비스를 만드는 방법을 보여줍니다. `site` 접두어는 사이트별 사용자 정의용으로 예약되어 있습니다. `svc:/site/service-name` 서비스는 Oracle Solaris 릴리스에서 제공된 서비스와 충돌하지 않습니다.

▼ 서비스 번들 생성기 도구를 사용하여 SMF 서비스를 만드는 방법

이 절차에서는 기존 사용자 정의 스크립트를 시작 메소드로 사용하는 서비스를 만드는 방법을 보여줍니다.

1. 서비스 모델을 확인합니다.

기본적으로 `svcbundle`은 `transient` 서비스를 만듭니다. 이 서비스에 대한 시작 메소드 스크립트가 오래 실행되는 데몬을 시작하고 `contract` 서비스인지 확인합니다. 서비스 모델에 대한 자세한 내용은 “서비스 모델” [19] 및 `svc.startd(1M)` 매뉴얼 페이지를 참조하십시오.

2. 스크립트를 표준 위치에 복사합니다.

이 예에서 이 서비스에 대한 시작 메소드 스크립트가 될 스크립트의 이름은 `ex_svc`입니다. 이 스크립트를 `/lib/svc/method/ex_svc`에 복사합니다.

3. 초기 매니페스트를 만듭니다.

이 예에서 서비스 이름은 `site/ex_svc`입니다. 이는 일시적 서비스이며 중지 메소드가 필요하지 않습니다.

```
$ svcbundle -o /tmp/ex_svc.xml -s service-name=site/ex_svc \
-s start-method=/lib/svc/method/ex_svc
```

4. 필요에 따라 매니페스트를 변경합니다.

`/tmp/ex_svc.xml` 매니페스트의 콘텐츠가 필요한 콘텐츠인지 확인합니다. 필요에 따라 주석을 추가합니다.

5. 매니페스트가 유효한지 확인합니다.

svccfg validate 명령을 사용하여 서비스 매니페스트가 유효한지 확인합니다.

6. 매니페스트를 표준 디렉토리로 복사합니다.

```
$ cp /tmp/ex_svc.xml /lib/svc/manifest/site/ex_svc.xml
```

7. 매니페스트를 가져온 다음 서비스를 시작합니다.

```
$ svcadm restart manifest-import
```

예 5-1 생성된 매니페스트 자동 설치

새 서비스 매니페스트를 변경할 필요가 없으면 -i 옵션을 사용하여 생성된 직후 매니페스트를 설치합니다. svcbundle 명령은 매니페스트를 /lib/svc/manifest/site에 쓰고 manifest-import 서비스를 다시 시작합니다. /lib/svc/manifest/site 디렉토리에서 이름이 같은 기존 파일을 덮어씁니다.

```
# svcbundle -i -s service-name=site/ex_svc \
-s start-method=/lib/svc/method/ex_svc
```

Oracle 데이터베이스 인스턴스를 시작 또는 중지하는 서비스 만들기

이 예에서는 Oracle 데이터베이스를 관리하는 데 도움이 되는 다음 서비스를 제공합니다.

- Oracle 데이터베이스 인스턴스를 시작하거나 중지하는 데이터베이스 서비스
- 데이터베이스 인스턴스에 대한 클라이언트 연결 요청의 수신 트래픽을 관리하는 프로세스인 리스너를 시작하는 리스너 서비스

이 예에서는 파일 지원 저장소를 사용합니다. 파일 지원 저장소 대신 ASM(Automatic Storage Management) 기능을 사용할 수 있습니다. ASM은 Oracle 데이터베이스 파일에 대한 볼륨 관리자 및 파일 시스템입니다.

Oracle 데이터베이스를 각 설치에 대해 다음 환경 변수를 설정해야 합니다.

ORACLE_HOME 데이터베이스가 설치되는 위치입니다. 이 섹션의 예에서 데이터베이스 설치 위치는 /opt/oracle/product/home입니다.

ORACLE_SID 시스템에서 특정 데이터베이스를 고유하게 식별하는 시스템 ID입니다.

이 예에서 이러한 환경 변수는 서비스 매니페스트에서 정의된 후 메소드 스크립트에서 사용 됩니다.

데이터베이스 시작 및 중지 서비스

이 섹션에서는 Oracle 데이터베이스 인스턴스 제어 서비스 매니페스트인 `/lib/svc/manifest/site/oracle.xml`을 보여줍니다. 이 서비스 매니페스트에 대해 살펴볼 몇 가지 기능은 다음과 같습니다.

- `svc:/site/application/database/oracle:default`라는 하나의 서비스 인스턴스가 정의됩니다. 이 인스턴스는 기본적으로 사용으로 설정됩니다.

이 예에서는 기본 인스턴스를 정의하는 두 가지 방법을 보여줍니다. 이 예에서 기본 인스턴스는 매니페스트의 맨 위에 있는 `create_default_instance` 요소에 정의됩니다. 매니페스트의 맨 아래에 있는 `instance` 요소는 이 작업을 수행하는 다른 방법을 보여줍니다.

- 이 서비스를 사용하려면 모든 로컬 파일 시스템이 마운트되고 모든 네트워크 인터페이스가 초기화되어야 합니다.

파일 지원 데이터베이스를 사용하고 있다면 데이터베이스 서비스가 로컬 파일 시스템에 종속되어야 합니다. ASM을 사용하고 있다면 데이터베이스 서비스가 ASM을 관리하는 서비스에 종속되어야 합니다. 원격 클라이언트 연결을 허용하려면 데이터베이스 서비스가 네트워킹에 종속되어야 합니다.

- `method_context` 요소의 `method_environment` 요소는 시작 또는 중지할 데이터베이스 인스턴스를 식별하는 `ORACLE_HOME` 및 `ORACLE_SID` 환경 변수를 정의합니다. 그런 다음 이러한 값이 메소드 스크립트에서 사용하도록 제공됩니다.

이 서비스의 인스턴스를 여러 개 만들면(매니페스트의 맨 아래에 있는 `instance` 요소 참조) 각 인스턴스에는 해당 특정 데이터베이스에 대한 고유한 `ORACLE_HOME` 및 `ORACLE_SID` 값을 정의하는 고유한 `method_context` 요소가 필요할 수 있습니다.

- `method_context` 요소의 속성을 사용하여 이 예에 표시된 프로젝트 및 작업 디렉토리 외에 리소스 풀을 정의할 수 있습니다. `method_context` 요소의 `method_profile` 또는 `method_credential` 요소도 정의할 수 있습니다. `method_credential` 요소는 이 예에 표시된 `user`, `group` 및 `privileges` 값 이외에 `supp_groups` 및 `limit_privileges` 값을 지정할 수 있습니다. 자세한 내용은 DTD를 참조하십시오.

- 시작/중지 메소드 스크립트는 `/lib/svc/method/oracle`입니다. 메소드 시간이 초과되기 전에 경과되는 시간(초)은 기본값부터 증가합니다.

- 이 서비스 인스턴스를 사용 또는 사용 안함으로 설정하려면 사용자에게 `solaris.smf.manage.oracle` 권한을 부여해야 합니다. 이 예에서 사용자 `oracle`에는 `solaris.smf.manage.oracle` 권한이 부여됩니다.

```
<?xml version="1.0"?>

<!--
  Define a service to control the startup and shutdown of a database instance.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type="manifest" name="oracle">
<service name="site/application/database/oracle" type="service" version="1">
  <create_default_instance enabled="true" />

```

```

<!--
    Wait for all local file systems to be mounted.
    Wait for all network interfaces to be initialized.
-->

<dependency type="service"
    name="fs-local"
    grouping="require_all"
    restart_on="none">
    <service_fmri value="svc:/system/filesystem/local" />
</dependency>

<dependency type="service"
    name="network"
    grouping="require_all"
    restart_on="none">
    <service_fmri value="svc:/milestone/network:default" />
</dependency>

<!-- Define the methods. -->

<method_context project=":default" working_directory=":default">
    <method_credential user="oracle" group="dba" privileges=":default" />
    <method_environment>
        <envvar name="ORACLE_HOME" value="/opt/oracle/product/home" />
        <envvar name="ORACLE_SID" value="oracle" />
    </method_environment>
</method_context>

<exec_method type="method"
    name="start"
    exec="/lib/svc/method/oracle start"
    timeout_seconds="120"/>

<exec_method type="method"
    name="stop"
    exec="/lib/svc/method/oracle stop"
    timeout_seconds="120" />

<!--
    What authorization is needed to allow the framework
    general/enabled property to be changed when performing the
    action (enable, disable, etc) on the service.
-->

<property_group name="general" type="framework">
    <propval type="astring"
        name="action_authorization"
        value="solaris.smf.manage.oracle" />
</property_group>

<!-- Define an instance of the database. -->

<!--<instance name="default" enabled="true" />-->

```

```
    <stability value="Evolving" />
</service>
</service_bundle>
```

해당 사용자가 `svcs` 및 `svccfg describe` 명령에서 서비스에 대한 정보를 얻을 수 있도록 매니페스트에 이름 및 설명 메타 데이터를 추가합니다. DTD에서 `template` 요소를 참조하십시오.

`svccfg validate` 명령을 사용하여 서비스 매니페스트가 유효한지 확인합니다.

다음은 Oracle 데이터베이스 인스턴스 제어 서비스에 대한 시작/중지 메소드 스크립트 /`lib/svc/method/oracle`입니다. 이 메소드는 데이터베이스 `dbstart` 및 `dbshut` 명령을 호출합니다.

```
#!/bin/ksh -p

. /lib/svc/share/smf_include.sh

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
export PATH=$PATH:$ORACLE_HOME/bin

function startup
{
    dbstart $ORACLE_HOME
}

function shutdown
{
    dbshut $ORACLE_HOME
}

case $1 in
  start) startup ;;
  stop) shutdown ;;

  *) echo "Usage: $0 { start | stop }" >&2
    exit $SMF_EXIT_ERR_FATAL
    ;;
esac

exit $SMF_EXIT_OK
```

데이터베이스 리스너 서비스

리스너 서비스는 데이터베이스 인스턴스에 대한 클라이언트 연결 요청의 수신 트래픽을 관리하는 프로세스입니다. 리스너 서비스는 관리 중인 클라이언트 연결이 포함된 데이터베이스 서비스 인스턴스에 종속됩니다.

이 섹션에서는 Oracle 데이터베이스 인스턴스 제어 서비스 매니페스트인 `/lib/svc/manifest/site/listener.xml`을 보여줍니다. 이 서비스 매니페스트에 대해 살펴볼 몇 가지 기능은 다음과 같습니다.

- `svc:/site/application/database/listener:default`라는 하나의 서비스 인스턴스가 정의됩니다. 이 인스턴스는 기본적으로 사용으로 설정됩니다.
- 서비스를 사용하려면 Oracle 데이터베이스 인스턴스 제어 서비스인 `svc:/site/application/database/oracle`을 시작해야 합니다. 데이터베이스 인스턴스가 어떤 이유로 다시 시작되면 리스너도 다시 시작됩니다.
- `method_context` 요소의 `method_environment` 요소는 시작 또는 중지할 데이터베이스 인스턴스를 식별하는 `ORACLE_HOME` 및 `ORACLE_SID` 환경 변수를 정의합니다. 그런 다음 이러한 값이 메소드 스크립트에서 사용하도록 제공됩니다.
이 서비스의 인스턴스를 여러 개 만들면(매니페스트의 맨 아래에 있는 `instance` 요소 참조) 각 인스턴스에는 해당 특정 데이터베이스에 대한 고유한 `ORACLE_HOME` 및 `ORACLE_SID` 값을 정의하는 고유한 `method_context` 요소가 필요할 수 있습니다.
- `method_context` 요소의 속성을 사용하여 이 예에 표시된 프로젝트 및 작업 디렉토리 외에 리소스 풀을 정의할 수 있습니다. `method_context` 요소의 `method_profile` 또는 `method_credential` 요소도 정의할 수 있습니다. `method_credential` 요소는 이 예에 표시된 `user`, `group` 및 `privileges` 값 이외에 `supp_groups` 및 `limit_privileges` 값을 지정할 수 있습니다. 자세한 내용은 DTD를 참조하십시오.
- 시작/중지 메소드 스크립트는 `/lib/svc/method/listener`입니다. 메소드 시간이 초과되기 전에 경과되는 시간(초)은 기본값부터 증가합니다.
- 이 서비스 인스턴스를 사용 또는 사용 안함으로 설정하려면 사용자에게 `solaris.smf.manage.oracle` 권한을 부여해야 합니다.
- 서비스는 `transient`입니다. “서비스 모델” [19]을 참조하십시오.

```
<?xml version="1.0"?>
<!--
  Define a service to control the startup and shutdown of a database listener.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type="manifest" name="listener">
<service name="site/application/database/listener" type="service" version="1">
  <create_default_instance enabled="true" />
  <!--<single_instance />-->

  <!-- Wait for the database to be started. -->

  <dependency type="service"
    name="oracle"
    grouping="require_all"
    restart_on="refresh">
    <service_fmri value="svc:/site/application/database/oracle" />
  </dependency>

  <!-- Define the methods. -->
```

```

<method_context project=":default" working_directory=":default">
  <method_credential user="oracle" group="dba" privileges=":default" />
  <method_environment>
    <envvar name="ORACLE_HOME" value="/opt/oracle/product/home" />
    <envvar name="ORACLE_SID" value="oracle" />
  </method_environment>
</method_context>

<exec_method type="method"
  name="start"
  exec="/lib/svc/method/listener start"
  timeout_seconds="150"/>

<exec_method type="method"
  name="stop"
  exec="/lib/svc/method/listener stop"
  timeout_seconds="30" />

<!--
  What authorization is needed to allow the framework
  general/enabled property to be changed when performing the
  action (enable, disable, etc) on the service.
-->

<property_group name="general" type="framework">
  <propval type="astring"
    name="action_authorization"
    value="solaris.smf.manage.oracle" />
</property_group>

<!-- Make the instance transient (since it backgrounds itself). -->

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

<!-- Define an instance of the listener. -->

<!--<instance name="default" enabled="true" />-->

  <stability value="Evolving" />
</service>
</service_bundle>

```

해당 사용자가 `svcs` 및 `svccfg describe` 명령에서 서비스에 대한 정보를 얻을 수 있도록 매니페스트에 이름 및 설명 메타 데이터를 추가합니다. DTD에서 `template` 요소를 참조하십시오.

다음은 Oracle 데이터베이스 인스턴스 리스너 서비스에 대한 시작/중지 메소드 스크립트 / `lib/svc/method/listener`입니다. 이 메소드는 리스너 프로세스 `lsnrctl`을 시작하거나 중지합니다. `lsnrctl`이 시작되면 데이터베이스 서비스의 상태가 테스트됩니다.

```
#!/bin/ksh -p
```

```

. /lib/svc/share/smf_include.sh

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
export PATH=$PATH:$ORACLE_HOME/bin

function startup
{
    lsnrctl start

    # Wait for the listener to report ready.

    i=0
    while ! lsnrctl status | grep -i ready ; do
        ((i = i+1))
        if (( $i == 120 )) ; then
            # It's been *at least* 2 minutes, time to give up.
            echo "The listener failed to report ready." >&2
            exit $SMF_EXIT_ERR_FATAL
        fi

        sleep 1
    done

    # Ping the database once to prove it is now available.

    if ! tnsping $ORACLE_SID ; then
        exit $SMF_EXIT_ERR_FATAL
    fi
}

function shutdown
{
    lsnrctl stop
}

case $1 in
    start) startup ;;
    stop) shutdown ;;

    *) echo "Usage: $0 { start | stop }" >&2
       exit $SMF_EXIT_ERR_FATAL
       ;;
esac

exit $SMF_EXIT_OK

```

서비스, 인스턴스, 등록 정보 그룹 및 등록 정보 이름 지정

서비스 및 인스턴스 이름은 다음 표현식에 맞아야 합니다.

```
(([A-Za-z0-9]_[A-Za-z0-9.-]*,)?[A-Za-z0-9]_[A-Za-z0-9.-]*)
```

서비스 또는 인스턴스 이름은 대소문자를 구분하고 영숫자 문자로 시작되어야 하며 영숫자, 문자, 밑줄(_), 하이픈(-), 점(.)을 포함할 수 있습니다. 이전 버전과의 호환성을 유지 관리하려고 첫 문자와 마지막 문자 사이에 단일 쉼표(,)를 사용할 수 있습니다.

등록 정보 그룹 및 등록 정보 이름은 다음 표현식에 맞아야 합니다.

```
[A-Za-z0-9-._~:/?#\[\]@!$&'(\)*+,;=%]+
```

등록 정보 그룹 및 등록 정보 이름은 대소문자를 구분하고 영숫자 문자, 하이픈(-), 점(.), 밑줄(_), 틸드(~), 콜론(:), 슬래시(/), 물음표(?), 파운드 문자(#), 대괄호([및]), 위치 기호(@), 느낌표(!), 달러 기호(\$), 앰퍼센드(&), 작은따옴표('), 괄호((및)), 별표(*), 더하기 기호(+), 쉼표(,), 세미콜론(;), 등호(=), 공백, 퍼센트 기호(%)를 포함할 수 있습니다.

FMRI에서 등록 정보 그룹 및 등록 정보 이름은 쉼표 문자가 인코딩되지 않는다는 점을 제외하고 [Uniform Resource Identifier \(URI\) Generic Syntax RFC 3986](#)에 따라 인코딩됩니다.

등록 정보 그룹 유형은 이 등록 정보 그룹의 범주입니다. 등록 정보 그룹 유형에는 application, dependency, method, framework, implementation 및 template이 포함됩니다. smf(5)의 확장 이름 지정 규약을 준수하면 추가 등록 정보 그룹 유형을 도입할 수 있습니다. framework, implementation 또는 template을 등록 정보 그룹 유형으로 지정하지 마십시오. framework, implementation 또는 template 유형의 등록 정보 그룹은 SMF에서 특수하게 사용됩니다. application 유형의 등록 정보 그룹은 이 그룹이 연결된 서비스에만 관련되어야 합니다.

▼ 실행 제어 스크립트를 SMF 서비스로 변환하는 방법

이 절차에서는 SMF에서 실행 제어 서비스를 관리할 수 있도록 실행 제어 스크립트를 SMF 서비스 매니페스트로 바꾸는 방법을 보여줍니다. 실행 제어 스크립트를 변환하려면 rc-script 이름을 svcbundle 명령의 -s 옵션과 함께 사용합니다. 자세한 내용은 [svcbundle\(1M\)](#) 매뉴얼 페이지를 참조하거나 svcbundle help rc-script를 입력하십시오.

1. 서비스 모델을 확인합니다.

기본적으로 svcbundle은 transient 서비스를 만듭니다. 이 실행 제어 스크립트가 오래 실행되는 데몬을 시작하고 contract 서비스인지 확인합니다. 서비스 모델에 대한 자세한 내용은 [“서비스 모델” \[19\]](#) 및 [svc.startd\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

2. 초기 매니페스트를 만듭니다.

이 예에서 서비스 이름은 ex_con이고 레벨 2에서 실행되는 계약 서비스입니다.

```
$ svcbundle -o /tmp/ex_con.xml -s service-name=ex_con
-s rc-script=/etc/init.d/ex_con:2 -s model=contract
```

3. **필요에 따라 매니페스트를 변경합니다.**
/tmp/ex_con.xml 매니페스트의 콘텐츠가 필요한 콘텐츠인지 확인합니다. 필요에 따라 주석을 추가합니다.
4. **매니페스트를 표준 디렉토리로 복사합니다.**

```
$ cp /tmp/ex_con.xml /lib/svc/manifest/site/ex_con.xml
```
5. **기존 서비스를 중지합니다.**

```
$ /etc/init.d/ex_con stop
```
6. **실행 제어 스크립트를 사용 안함으로 설정합니다.**
해당 rc n.d 디렉토리에서 실행 제어 스크립트에 대한 모든 링크를 제거합니다.
7. **매니페스트를 가져온 다음 서비스를 시작합니다.**

```
$ svcadm restart manifest-import
```
8. **새 서비스를 나열합니다.**
새 서비스가 있고 예상된 상태인지 확인합니다.

```
$ svcs ex-con
```

스텐실을 사용하여 구성 파일 만들기

응용 프로그램에서 등록 정보를 읽는 데 libscf 라이브러리 인터페이스를 사용할 수 없으면 스텐실을 사용하여 구성 파일을 만들 수 있습니다. 스텐실 서비스는 스텐실 파일과 스텐실 서비스에 정의되는 등록 정보 값을 사용하여 구성 파일을 만듭니다. 스텐실 파일에는 서비스가 현재 SMF에서 관리되더라도 서비스에 필요한 구성 파일의 구조 정의가 포함됩니다. 스텐실 서비스를 사용하여 기존 응용 프로그램을 변경하지 않고 SMF 구성 관리를 활용할 수 있습니다.

스텐실은 서비스 인스턴스 start 메소드가 실행되기 바로 전에 구성 파일을 생성하는 데 사용됩니다. 스텐실된 등록 정보 값을 업데이트하는 경우 서비스를 다시 시작하여 변경 사항을 구성 파일에 통합한 후 응용 프로그램을 시작하고 구성 파일을 읽습니다.

▼ 스텐실 서비스를 만드는 방법

스텐실 파일에는 서비스가 현재 SMF에서 관리되더라도 계속해서 서비스에 필요한 구성 파일의 구조 정의가 포함됩니다. svcio 유틸리티는 SMF 서비스의 스텐실 파일 및 등록 정보에 있는 정의에서 구성 파일을 생성합니다. svcio 유틸리티에 대한 자세한 내용은 svcio(1) 매

뉴얼 페이지를 참조하고 스텐실 파일 형식에 대한 자세한 내용은 `smf_stencil(4)` 매뉴얼 페이지를 참조하십시오.

1. 스텐실 파일을 만듭니다.

스텐실 파일은 구성 파일을 만드는 데 사용할 형식을 `svcio` 유틸리티에 알립니다. `svcio` 유틸리티는 스텐실이라는 템플릿을 기반으로 SMF 등록 정보를 응용 프로그램별 구성 파일로 변환합니다.

2. 서비스에 등록 정보 그룹을 추가합니다.

스텐실 서비스 등록 정보 그룹은 구성 파일을 만드는 데 사용할 경로와 소유권을 `svcio` 유틸리티에 알립니다. SMF는 시작 또는 새로 고침 메소드를 실행하기 전에 모든 스텐실 인식 서비스에 대한 구성을 재생성합니다. `configfile` 유형의 등록 정보 그룹은 구성 파일을 생성하는 방법을 SMF에 알립니다. 각 `configfile` 등록 정보 그룹은 서비스에 대한 단일 구성 파일을 설명하고 SMF 저장소에 저장된 다른 등록 정보에서 이러한 파일을 생성하는 방법을 `svcio`에 알립니다.

서비스에서 스텐실을 인식하게 구성하려면 템플릿 및 결과 구성 파일로 사용할 두 스텐실 파일의 경로가 포함된 각 관리되는 구성 파일에 대한 등록 정보 그룹을 추가합니다. 등록 정보 그룹에는 다음 등록 정보가 있습니다.

<code>path</code>	구성 파일을 쓸 경로입니다(예: <code>/etc/svc.conf</code>).
<code>stencil</code>	<code>/lib/svc/stencils</code> 에 대한 상대적 위치인 사용할 스텐실 파일의 경로입니다. 예를 들어 <code>stencil</code> 등록 정보 값이 <code>svc.stencil</code> 이면 <code>/lib/svc/stencils/svc.stencil</code> 파일이 사용됩니다.
<code>mode</code>	구성 파일(<code>path</code>)에 사용할 모드입니다(예: <code>644</code>).
<code>owner</code>	구성 파일(<code>path</code>)에 대해 설정할 소유자입니다. 이 등록 정보를 설정하지 않을 경우 파일의 소유자는 <code>svcio</code> 를 호출하는 사용자입니다.
<code>group</code>	구성 파일(<code>path</code>)에 대해 설정할 그룹입니다. 이 등록 정보를 설정하지 않으면 해당 그룹이 <code>path</code> 에 대한 기본 그룹이 됩니다.

Oracle Solaris의 스텐실 서비스 예

Puppet 및 Kerberos에 대한 서비스에서는 스텐실을 사용하여 구성 파일을 제공합니다.

Puppet 스텐실 서비스

Puppet은 많은 시스템의 구성을 관리하는 데 사용되는 툴킷입니다. Oracle Solaris에서 Puppet 응용 프로그램은 SMF에서 관리됩니다.

Puppet 서비스의 상위 레벨 보기

system/management/puppet 패키지를 설치하면 두 개의 SMF 서비스 인스턴스인 puppet:master 및 puppet:agent가 제공됩니다. 이러한 인스턴스는 기본적으로 사용 안함으로 설정됩니다.

이러한 인스턴스를 사용으로 설정한 후 다음 명령은 puppet:master 및 puppet:agent가 둘 다 계약 서비스임을 보여줍니다.

```
$ svcs -p puppet
STATE          STIME          FMRI
online         17:19:32      svc:/application/puppet:agent
                17:19:32          2565 puppet
online         17:19:32      svc:/application/puppet:master
                17:19:32          2567 puppet
```

다음 명령은 계약 서비스에서 시작된 프로세스에 대한 추가 정보를 보여줍니다.

```
$ ps -o pid,args -p 2565,2567
PID COMMAND
2565 /usr/ruby/1.9/bin/ruby /usr/sbin/puppet agent --logdest /var/log/puppet/puppet-
2567 /usr/ruby/1.9/bin/ruby /usr/sbin/puppet master --logdest /var/log/puppet/puppet
```

ps 출력에 제시된 대로 puppet은 /var/log/puppet의 로그 파일에 쓰고 있습니다.

```
$ ls /var/log/puppet
puppet-agent.log  puppet-master.log
```

초기 Puppet 구성 파일

Puppet은 /etc/puppet/puppet.conf라는 구성 파일을 사용해야 합니다. /usr/sbin/puppet 응용 프로그램은 /etc/puppet/puppet.conf에서 구성 정보를 읽고 application/puppet 서비스 인스턴스의 등록 정보 세트에서는 읽지 않습니다. 필수 구성 파일을 제공하려고 각 puppet 인스턴스는 스텐실 파일과 configfile 등록 정보 그룹을 제공합니다. configfile 등록 정보 그룹은 지정된 구성 파일을 실행하고 만들도록 svcio 유틸리티에 알립니다. 스텐실 파일은 서비스 등록 정보 값의 데이터를 구성 파일에 올바른 형식으로 쓰는 데 사용됩니다.

다음 명령은 configfile 유형의 등록 정보 그룹에 있는 모든 puppet 서비스 등록 정보를 보여줍니다. 이 출력은 puppet 서비스의 인스턴스에 같은 configfile 등록 정보와 같은 값이 포함됨을 보여줍니다. 각 puppet 서비스 인스턴스는 구성 파일 경로, 구성 파일 모드 및 스텐실 파일 경로를 제공합니다.

```
$ svccprop -g configfile puppet
svc:/application/puppet:master/:properties/puppet_stencil/mode astring 0444
svc:/application/puppet:master/:properties/puppet_stencil/path astring /etc/puppet/puppet.conf
svc:/application/puppet:master/:properties/puppet_stencil/stencil astring puppet.stencil
svc:/application/puppet:agent/:properties/puppet_stencil/mode astring 0444
svc:/application/puppet:agent/:properties/puppet_stencil/path astring /etc/puppet/puppet.conf
svc:/application/puppet:agent/:properties/puppet_stencil/stencil astring puppet.stencil
```

다음 명령은 이러한 인스턴스 등록 정보가 상위 서비스에서 상속됨을 확인합니다.

```
$ svccfg -s puppet listprop -l all puppet_stencil
puppet_stencil      configfile manifest
puppet_stencil/mode astring   manifest      0444
puppet_stencil/path astring   manifest      /etc/puppet/puppet.conf
puppet_stencil/stencil astring   manifest      puppet_stencil
$ svccfg -s puppet:agent listprop -l all puppet_stencil
$ svccfg -s puppet:master listprop -l all puppet_stencil
```

예를 들어 기반구조에 따라 puppet:agent1 및 puppet:agent2 인스턴스가 필요할 수 있습니다. 이 경우 “Puppet 구성 파일 수정” [93]에 표시된 대로 등록 정보 값을 사용자 정의하고 각 인스턴스에 대한 등록 정보를 추가합니다.

구성 파일 /etc/puppet/puppet.conf의 초기 콘텐츠는 다음과 같습니다.

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

Puppet 스텐실 파일

스텐실 파일의 콘텐츠는 구성 파일에 기록되는 등록 정보 및 기타 정보를 사용자에게 알립니다. puppet_stencil/stencil 등록 정보의 값인 puppet_stencil 경로는 /lib/svc/stencils에 대한 상대적 위치입니다. 다음은 스텐실 파일 /lib/svc/stencils/puppet_stencil의 콘텐츠입니다.

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
; walk each instance and extract all properties from the config PG
$%/(svc:/%s:(.*)/:properties)/ {
  $%{${%1}/general/enabled:?
  [%2]
  $%/${%1}/config/(.*)/ {
    %3 = $%{${%1}/config/%3} }
  }
}
```

스텐실 파일에서 svc:/%s:(.*)/:properties(또는 %1)는 svc:/application/puppet:agent/:properties 및 svc:/application/puppet:master/:properties로 확장됩니다. 여기서 .* (또는 %2)는 모든 인스턴스와 일치해야 합니다. 그런 다음 인스턴스 이름은 구성 파일의 블록에 레이블을 지정하는 데 사용됩니다. 다음에 나타나는 .* (또는 %3)는 %1 서비스 인스턴스에 대한 config 등록 정보 그룹의 모든 등록 정보와 일치합니다. 스텐실은 서비스 인스턴스에서 등록 정보 이름과 해당 등록 정보의 값을 구성 파일에 쓰도록 svcio에 알립니다.

Puppet 구성 파일 수정

“초기 Puppet 구성 파일” [91]에서 확인한 대로 처음에는 리터럴 주석 행만 구성 파일에 기록됩니다. 스텐실 파일에서 `general/enabled` 등록 정보의 값을 테스트하면 등록 정보 값이 구성 파일에 기록되지 않습니다. 다음 명령은 기본적으로 `general/enabled` 등록 정보 값이 `false`임을 보여줍니다.

```
$ svcprop -p general/enabled puppet
svc:/application/puppet:master/:properties/general/enabled boolean false
svc:/application/puppet:agent/:properties/general/enabled boolean false
```

`svcadm enable` 명령을 사용하여 인스턴스를 사용으로 설정하면 `general/enabled` 등록 정보 값이 변경되지 않습니다. `general/enabled` 등록 정보 값을 `true`로 변경하고 인스턴스를 다시 시작하면 해당 인스턴스에 대한 `config` 등록 정보 그룹의 모든 등록 정보가 구성 파일에 기록됩니다.

```
$ svccfg -s puppet:agent setprop general/enabled=true
$ svcprop -p general/enabled puppet:agent
false
$ svcadm refresh puppet:agent
$ svcprop -p general/enabled puppet:agent
true
$ svcadm restart puppet:agent
```

다음 명령은 처음에 `config` 등록 정보 그룹의 유일한 등록 정보가 각 인스턴스에 대한 로그 파일 경로임을 보여줍니다.

```
$ svcprop -p config puppet
svc:/application/puppet:master/:properties/config/logdest astring /var/log/puppet/puppet-master.log
svc:/application/puppet:agent/:properties/config/logdest astring /var/log/puppet/puppet-agent.log
```

사용으로 설정된 인스턴스에 대한 `config` 등록 정보는 구성 파일의 인스턴스 이름으로 레이블이 지정된 블록에 추가되었습니다.

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

```
[agent]
```

```
logdest = /var/log/puppet/puppet-agent.log
```

Puppet 구성 설명서에 따르면 구성 파일에는 `[main]`, `[agent]` 및 `[master]` 블록이 포함될 수 있습니다. `[main]` 블록의 구성은 `agent` 및 `master`에 모두 적용됩니다. Puppet `agent`의 경우 `[agent]` 블록의 구성이 `[main]` 블록의 같은 구성을 대체합니다. Puppet `master`의 경우 `[master]` 블록의 구성이 `[main]` 블록의 같은 구성을 대체합니다. `agent` 및 `master`에 공통적인 구성에 대한 `[main]` 블록을 제공하려면 `puppet:main` 인스턴스와 해당 인스턴스에 대한 적절한 `config` 등록 정보를 만듭니다.

다음 명령은 Puppet 구성 파일에 구성을 추가하는 방법을 보여줍니다.

```
$ svccfg -s puppet:agent
svc:/application/puppet:agent> setprop config/report=true
svc:/application/puppet:agent> setprop config/pluginsync=true
svc:/application/puppet:agent> refresh
svc:/application/puppet:agent> exit
$ svcadm restart puppet:agent
$ cat /etc/puppet/puppet.conf
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.

[agent]

logdest = /var/log/puppet/puppet-agent.log
pluginsync = true
report = true
```

비슷한 명령을 사용하여 등록 정보를 제거하고 등록 정보 값을 변경할 수 있습니다. [4장. 서비스 구성](#)을 참조하십시오. main 인스턴스를 추가하려면 “[서비스 인스턴스 추가](#)” [71]에 표시된 대로 svccfg add 명령을 사용합니다.

Kerberos 스텐실 서비스

스텐실을 사용하는 Oracle Solaris 서비스의 다른 예는 Kerberos입니다. 다음 명령은 configfile 등록 정보 그룹이 krb5_conf이고, 스텐실 파일이 /lib/svc/stencils/krb5.conf.stencil이고, 구성 파일이 /etc/krb5/krb5.conf임을 보여줍니다.

```
$ svcprop -g configfile svc:/system/kerberos/install:default
krb5_conf/disabled boolean true
krb5_conf/group astring sys
krb5_conf/mode integer 644
krb5_conf/owner astring root
krb5_conf/path astring /etc/krb5/krb5.conf
krb5_conf/stencil astring krb5.conf.stencil
```

SMF 모범 사례 및 문제 해결

이 부록은 몇 가지 권장 모범 사례를 제공하고 몇몇 SMF 서비스 문제를 해결하는 방법을 보여줍니다.

SMF 모범 사례

대부분 서비스는 구성 정책을 설명합니다. 원하는 구성이 구현되지 않은 경우 서비스를 수정하여 정책 설명을 수정합니다. 서비스 등록 정보 값을 수정하거나 다른 등록 정보 값을 사용하여 새 서비스 인스턴스를 만듭니다. 서비스 인스턴스를 사용 안함으로 설정하지 않고 SMF 서비스로 관리하려는 구성 파일을 편집하십시오.

Oracle 또는 타사 소프트웨어 공급업체에서 제공하는 매니페스트 및 시스템 프로파일을 수정하지 마십시오. 이러한 매니페스트와 프로파일은 시스템을 업그레이드할 때 대체될 수 있고 이후 이러한 파일에 대한 변경 사항이 손실됩니다. 대신에 사이트 프로파일을 만들어 서비스를 사용자 정의하거나 `svccfg` 명령이나 `inetadm` 명령을 사용하여 등록 정보를 직접 조작합니다. 또한 `/lib/svc/manifest/site` 및 `/var/svc/manifest/site` 디렉토리는 사이트별 사용을 위해 예약되어 있습니다. Oracle Solaris에서는 이러한 디렉토리에 매니페스트가 제공되지 않습니다.

같은 사용자 정의 구성을 여러 시스템에 적용하려면 `svcbundle` 명령이나 `svccfg extract` 명령을 사용하여 프로파일 파일을 만듭니다. 해당 파일에서 등록 정보 값을 사용자 정의하고 각 사용자 정의 이유에 대한 주석을 포함합니다. 각 시스템의 `/etc/svc/profile/site`에 파일을 복사하고 각 시스템에서 `manifest-import` 서비스를 다시 시작합니다. [“여러 시스템 구성” \[73\]](#)을 참조하십시오.

사이트 프로파일을 만들 때 정의된 구성이 같은 서비스나 서비스 인스턴스에 대해 다른 사이트 프로파일에서 정의된 구성과 충돌하지 않는지 확인하십시오. SMF가 서비스 구성 저장소의 같은 계층에서 충돌하는 구성을 찾으면 영향을 받는 서비스 인스턴스가 유지 관리 상태로 전환됩니다.

매니페스트 및 프로파일 파일에 대해 비표준 위치를 사용하지 마십시오. 매니페스트 및 프로파일 표준 위치에 대해서는 [“서비스 번들” \[23\]](#)을 참조하십시오.

자체 사용을 위해 서비스를 만들 경우 서비스 이름 시작 부분에 `site`를 사용합니다. `svc:/site/service_name:instance_name`.

“시작 메시징 양 지정” [103]에 설명된 대로 로깅 레벨을 구성하는 경우를 제외하고 마스터 다시 시작 프로그램 서비스, `svc:/system/svc/restarter:default`를 수정하지 마십시오.

`svccfg delcust` 명령을 사용하기 전에 `svccfg listcust` 명령을 같은 옵션과 함께 사용하십시오. `delcust` 하위 명령은 서비스에 대한 모든 관리 사용자 정의를 제거할 수 있습니다. `listcust` 하위 명령을 사용하여 `delcust` 하위 명령으로 삭제될 사용자 정의를 확인합니다.

스크립트에서 전체 서비스 인스턴스 FMRI를 사용합니다.
`svc:/service_name:instance_name.`

서비스 문제 해결

이 절은 다음 내용으로 구성되어 있습니다.

- 실행 중인 스냅샷에 구성 변경 사항 커밋
- 문제가 있다고 보고되는 서비스 수정
- 인스턴스를 `degraded` 또는 `maintenance` 상태로 수동 전환
- 손상된 서비스 구성 저장소 수정
- 시스템 시작 시 표시 또는 저장할 메시징 양 구성
- 지정된 마일스톤으로 전환 또는 부트
- SMF를 사용하여 부트 문제 조사
- `inetd` 서비스를 SMF 서비스로 변환

구성 변경 사항 이해

서비스 구성 저장소에서 SMF는 등록 정보 변경 사항을 실행 중인 스냅샷의 등록 정보와 별도로 저장합니다. 서비스 구성을 변경하면 해당 변경 사항이 실행 중인 스냅샷에 바로 나타나지 않습니다.

새로 고침 작업은 편집 구성의 값을 사용하여 지정된 서비스 인스턴스의 실행 중인 스냅샷을 업데이트합니다.

기본적으로 `svcprop` 명령은 실행 중인 스냅샷의 등록 정보를 표시하고 `svccfg` 명령은 편집 구성의 등록 정보를 표시합니다. 등록 정보 값을 변경했으나 구성 새로 고침을 수행하지 않았으면 `svcprop` 및 `svccfg` 명령은 서로 다른 등록 정보 값을 표시합니다. 구성 새로 고침을 수행한 후 `svcprop` 및 `svccfg` 명령은 같은 등록 정보 값을 표시합니다.

재부트해도 실행 중인 스냅샷이 변경되지 않습니다. `svcadm restart` 명령은 구성을 새로 고치지 않습니다. `svcadm refresh` 또는 `svccfg refresh` 명령을 사용하여 구성 변경 사항을 실행 중인 스냅샷에 커밋합니다.

성능 저하, 오프라인 또는 유지 관리 상태인 인스턴스 복구

svcs -x 명령을 인수 없이 사용하여 다음 설명 중 하나와 일치하는 서비스 인스턴스에 대한 설명 정보를 표시합니다.

- 서비스가 사용으로 설정되었으나 실행되지 않습니다.
- 서비스가 또 다른 사용으로 설정된 서비스를 실행하지 못하게 합니다.

다음 목록에서는 서비스 문제에 접근하는 방법을 요약합니다.

1. 서비스 로그 파일 보기부터 시작하여 문제를 진단합니다.
2. 문제를 해결합니다. 문제 해결 시 서비스 구성을 수정할 경우 서비스를 새로 고칩니다.
3. 영향을 받는 서비스를 실행 상태로 이동합니다.

▼ 유지 관리 상태인 인스턴스를 복구하는 방법

유지 관리 상태인 서비스 인스턴스가 사용으로 설정되었으나 실행할 수 없습니다.

1. 인스턴스가 유지 관리 상태인 이유를 확인합니다.

관리 작업이 아직 완료되지 않아 인스턴스가 maintenance 상태 사이에서 전환되고 있을 수 있습니다. 인스턴스가 전환 중이면 해당 상태는 maintenance*로 표시되어야 합니다.

시작 메소드가 실패했기 때문에 다음 예에서 “State” 및 “Reason” 행은 pkg/depot 서비스가 maintenance 상태임을 보여줍니다.

```
$ svcs -x
svc:/application/pkg/depot:default (IPS Depot)
  State: maintenance since September 11, 2013 01:30:42 PM PDT
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
  See: http://support.oracle.com/msg/SMF-8000-KS
  See: pkg.depot-config(1M)
  See: /var/svc/log/application-pkg-depot:default.log
Impact: This service is not running.
```

Oracle Support 사이트에 로그인하여 참조되는 예측적 자가 치유 지식 문서를 확인합니다. 이 경우 문서에서는 로그 파일에서 시작 메소드가 실패한 이유를 확인하라고 알려줍니다. svcs 출력에는 로그 파일 이름이 제공됩니다. 로그 파일을 보는 방법에 대한 자세한 내용은 “서비스 로그 파일 보기” [37]를 참조하십시오. 이 예에서 로그 파일에는 시작 메소드 호출 및 치명적 오류 메시지가 표시됩니다.

```
[ Sep 11 13:30:42 Executing start method ("/lib/svc/method/svc-pkg-depot start"). ]
pkg.depot-config: Unable to get publisher information:
The path '/export/ipsrepos/Solaris11' does not contain a valid package repository.
```

2. 문제를 해결합니다.

다음 단계의 하나 이상이 필요할 수 있습니다.

■ **서비스 구성 업데이트**

보고된 문제를 해결하는 데 서비스 구성 수정이 필요한 경우에는 구성이 변경된 모든 서비스에 대해 `svccfg refresh` 또는 `svcadm refresh` 명령을 사용합니다. `svccprop` 명령을 사용하여 등록 정보 값을 확인하거나 이 서비스와 관련된 기타 서비스를 통해 실행 중인 스냅샷에서 구성이 업데이트되었는지 확인합니다.

■ **종속성이 실행 중인지 확인합니다.**

때때로 `svcs -x` 출력의 "Impact" 행에서는 maintenance 상태인 서비스에 종속된 서비스가 실행되고 있지 않음을 알려줍니다. `svcs -l` 명령을 사용하여 종속 서비스의 현재 상태를 확인합니다. 모든 필수 종속성이 실행 중인지 확인합니다. `svcs -x` 명령을 사용하여 모든 사용으로 설정된 서비스가 실행 중인지 확인합니다.

■ **계약 프로세스가 중지되었는지 확인합니다.**

maintenance 상태인 서비스가 계약 서비스이면 서비스에 의해 시작된 모든 프로세스가 중지되지 않았는지 확인합니다. 계약 서비스 인스턴스가 유지 관리 상태이면 다음 예와 같이 계약 ID가 비어 있어야 하고 해당 계약과 관련된 모든 프로세스가 중지되어야 합니다. `svcs -l` 또는 `svcs -o ctid`를 사용하여 유지 관리 상태인 서비스 인스턴스에 대한 계약이 없는지 확인합니다. `svcs -p`를 사용하여 이 서비스 인스턴스와 관련된 모든 프로세스가 실행 중인지 확인합니다. 유지 관리 상태인 서비스 인스턴스에 대해 `svcs -p`를 통해 표시된 모든 프로세스가 종료되어야 합니다.

```
$ svcs -l system-repository
fmri          svc:/application/pkg/system-repository:default
name          IPS System Repository
enabled       true
state         maintenance
next_state    none
state_time    September 17, 2013 07:18:19 AM PDT
logfile       /var/svc/log/application-pkg-system-repository:default.log
restarter     svc:/system/svc/restarter:default
contract_id
manifest      /lib/svc/manifest/application/pkg/pkg-system-repository.xml
dependency    require_all/error svc:/milestone/network:default (online)
dependency    require_all/none svc:/system/filesystem/local:default (online)
dependency    optional_all/error svc:/system/filesystem/autofs:default (online)
```

3. **인스턴스가 복구되었음을 다시 시작 프로그램에 알립니다.**

보고된 문제가 해결되면 `svcadm clear` 명령을 사용하여 서비스를 online 상태로 되돌립니다. maintenance 상태 서비스의 경우 `clear` 하위 명령은 서비스가 복구되었음을 해당 서비스에 대한 다시 시작 프로그램에 알립니다.

```
$ svcadm clear pkg/depot:default
```

-s 옵션을 지정하면 `svcadm` 명령은 인스턴스가 online 상태로 전환되거나 관리자 개입이 있어야 인스턴스가 online 상태로 전환될 수 있다고 결정될 때까지 돌아가지 않고 대기합니다.

-T 옵션을 -s 옵션과 함께 사용하여 전환을 수행할 상한(초)을 지정하거나 전환을 수행할 수 없게 결정합니다.

4. 인스턴스가 복구되었는지 확인합니다.

svcs 명령을 사용하여 유지 관리 상태였던 서비스가 현재 온라인 상태인지 확인합니다. svcs -x 명령을 사용하여 모든 사용으로 설정된 서비스가 실행 중인지 확인합니다.

▼ 오프라인 상태인 인스턴스를 복구하는 방법

오프라인 상태인 서비스 인스턴스가 사용으로 설정되었으나 실행되고 있지 않거나 실행할 수 없습니다.

1. 인스턴스가 오프라인 상태인 이유를 확인합니다.

종속성이 아직 충족되지 않아 인스턴스가 offline 상태 사이에서 전환되고 있을 수 있습니다. 인스턴스가 전환 중이면 해당 상태는 offline*로 표시되어야 합니다.

2. 문제를 해결합니다.

■ 서비스 종속성을 사용으로 설정합니다.

필요한 종속성이 사용 안함으로 설정된 경우 다음 명령을 사용하여 사용으로 설정합니다.

```
$ svcadm enable -r FMRI
```

■ 종속성 파일을 수정합니다.

종속성 파일이 누락되었거나 읽을 수 없는 상태일 수 있습니다. pkg fix 또는 pkg revert를 사용하여 이 문제 유형을 해결하려고 할 수 있습니다. pkg(1) 매뉴얼 페이지를 참조하십시오.

3. 필요한 경우 인스턴스를 다시 시작합니다.

필수 종속성이 충족되지 않아 인스턴스가 오프라인 상태인 경우 종속성을 수정하거나 사용으로 설정하면 오프라인 인스턴스가 다시 시작되고 추가적인 관리자 작업 없이 온라인으로 전환될 수 있습니다.

서비스에 다른 수정 사항을 적용한 경우 인스턴스를 다시 시작하십시오.

```
$ svcadm restart FMRI
```

4. 인스턴스가 복구되었는지 확인합니다.

svcs 명령을 사용하여 오프라인 상태였던 인스턴스가 현재 온라인 상태인지 확인합니다. svcs -x 명령을 사용하여 모든 사용으로 설정된 서비스가 실행 중인지 확인합니다.

▼ 성능 저하 상태인 인스턴스를 복구하는 방법

성능 저하 상태인 서비스 인스턴스가 사용으로 설정되고 실행 중이거나 실행할 수 있지만, 제한된 용량으로 작동 중입니다.

1. 인스턴스가 성능 저하 상태인 이유를 확인합니다.
2. 문제를 해결합니다.

3. 인스턴스를 온라인 상태로 전환하게 다시 시작 프로그램에 요청합니다.

보고된 문제가 해결되면 `svcadm clear` 명령을 사용하여 인스턴스를 `online` 상태로 되돌립니다. `degraded` 상태 인스턴스의 경우 `clear` 하위 명령은 해당 인스턴스에 대한 다시 시작 프로그램이 인스턴스를 `online` 상태로 전환하게 요청합니다.

```
$ svcadm clear pkg/depot:default
```

4. 인스턴스가 복구되었는지 확인합니다.

`svcs` 명령을 사용하여 성능 저하 상태였던 인스턴스가 현재 온라인 상태인지 확인합니다. `svcs -x` 명령을 사용하여 모든 사용으로 설정된 서비스가 실행 중인지 확인합니다.

인스턴스를 성능 저하 또는 유지 관리 상태로 표시

서비스 인스턴스를 `degraded` 상태나 `maintenance` 상태로 표시할 수 있습니다. 예를 들어 응용 프로그램이 루프 상태이거나 교착 상태이면 이 작업을 수행하려고 할 수 있습니다. 상태 변경에 대한 정보는 표시된 인스턴스의 종속성에 전파되어 다른 관련 인스턴스를 디버깅하는데 도움이 될 수 있습니다.

`-i` 옵션을 지정하여 즉각적인 상태 변경을 요청합니다.

인스턴스를 `maintenance`로 표시하면 `-t` 옵션을 지정하여 임시 상태 변경을 요청할 수 있습니다. 임시 요청은 재부트될 때까지만 지속됩니다.

`-s` 옵션을 `svcadm mark` 명령과 함께 지정하면 `svcadm`은 인스턴스를 표시하고 돌아가기 전에 인스턴스가 `degraded` 또는 `maintenance` 상태로 전환될 때까지 대기합니다. `-T` 옵션을 `-s` 옵션과 함께 사용하여 전환을 수행할 상한(초)을 지정하거나 전환을 수행할 수 없게 결정합니다.

저장소 문제 진단 및 복구

시스템 시작 시 저장소 데몬 `svc.configd`는 `/etc/svc/repository.db`에 저장된 구성 저장소에 대한 무결성 검사를 수행합니다. `svc.configd` 무결성 검사가 실패할 경우 `svc.configd` 데몬은 다음과 유사한 메시지를 콘솔에 작성합니다.

```
svc.configd: smf(5) database integrity check of:
```

```
  /etc/svc/repository.db
```

```
failed. The database might be damaged or a media error might have
prevented it from being verified. Additional information useful to
your service provider is in:
```

```
  /system/volatile/db_errors
```

```
The system will not be able to boot until you have restored a working
database. svc.startd(1M) will provide a sulogin(1M) prompt for recovery
purposes. The command:
```

```
  /lib/svc/bin/restore_repository
```

```
can be run to restore a backup version of your repository. See
http://support.oracle.com/msg/SMF-8000-MY for more information.
```

그런 다음 svc.configd 데몬이 종료됩니다. 이 종료는 svc.startd 데몬에 의해 감지되고 이후 svc.startd가 sulogin을 시작합니다.

sulogin 프롬프트에서 Ctrl-D를 입력하여 sulogin을 종료합니다. svc.startd 데몬은 sulogin 종료를 인식하고 svc.configd 데몬을 다시 시작하여 저장소를 다시 검사합니다. 추가로 다시 시작된 후에는 문제가 다시 나타날 수 없습니다. svc.configd 데몬을 직접 호출하지 마십시오. svc.startd 데몬은 svc.configd 데몬을 시작합니다.

svc.configd가 다시 무결성 검사 실패를 보고하고 sulogin 프롬프트가 다시 표시되면 필요한 파일 시스템이 가득 차지 않았는지 확인합니다. root 암호를 사용하여 sulogin 프롬프트에서나 원격으로 로그인합니다. root 및 system/volatile 파일 시스템에서 둘 다 공간을 사용할 수 있는지 확인합니다. 파일 시스템의 하나가 가득 차 있으면 정리하고 시스템을 다시 시작합니다. 파일 시스템이 둘 다 가득 차 있지 않으면 [백업에서 저장소를 복원하는 방법 \[101\]](#) 절차에 따릅니다.

다음 중 하나의 원인으로 인해 서비스 구성 저장소가 손상될 수 있습니다.

- 디스크 실패
- 하드웨어 버그
- 소프트웨어 버그
- 우발적인 파일 덮어쓰기

다음 절차에서는 손상된 저장소를 저장소의 백업 복사본으로 바꾸는 방법을 보여줍니다.

▼ 백업에서 저장소를 복원하는 방법

1. 로그인합니다.

root 암호를 사용하여 sulogin 프롬프트에서나 원격으로 로그인합니다.

2. 저장소 복원 명령을 실행합니다.

```
# /lib/svc/bin/restore_repository
```

이 명령을 실행하면 손상되지 않은 백업을 복원하는 데 필요한 단계를 안내합니다. “저장소 백업” [25]의 설명대로 SMF는 저장소의 백업을 자동으로 만듭니다.

SMF는 영구적 및 비영구적 구성 데이터를 유지 관리합니다. 이러한 두 가지 저장소에 대한 자세한 내용은 “서비스 구성 저장소” [23]를 참조하십시오. restore_repository 명령은 영구적 저장소만 복원합니다. 또한 restore_repository 명령은 시스템을 재부트하여 비영구적 구성 데이터를 삭제합니다. 비영구적 데이터는 시스템이 재부트 시 필요하지 않은 런타임 데이터입니다.

시작되면 /lib/svc/bin/restore_repository 명령은 다음과 유사한 메시지를 표시합니다.

```
See http://support.oracle.com/msg/SMF-8000-MY for more information on the use of
this script to restore backup copies of the smf(5) repository.
```

```
If there are any problems which need human intervention, this script will
give instructions and then exit back to your shell.
```

root(/) 파일 시스템이 쓰기 권한으로 마운트된 후 또는 시스템이 로컬 영역인 경우 복원할 저장소 백업을 선택하라는 메시지가 표시됩니다.

```
The following backups of /etc/svc/repository.db exists, from
oldest to newest:
```

```
... list of backups ...
```

백업 이름은 백업이 만들어진 유형 및 시간을 기준으로 제공됩니다. boot로 시작하는 백업은 시스템이 부트된 후 저장소에 처음으로 변경 작업을 수행하기 전에 완료됩니다. manifest_import로 시작하는 백업은 svc:/system/manifest-import:default가 프로세스를 마친 후 완료됩니다. 백업 시간은 YYYYMMDD_HHMMSS 형식으로 표시됩니다.

3. 적당한 응답을 입력합니다.

일반적으로 가장 최근의 백업 옵션이 선택됩니다.

```
Please enter either a specific backup repository from the above list to
restore it, or one of the following choices:
```

CHOICE	ACTION
boot	restore the most recent post-boot backup
manifest_import	restore the most recent manifest_import backup
-seed-	restore the initial starting repository (All customizations will be lost, including those made by the install/upgrade process.)
-quit-	cancel script and quit

```
Enter response [boot]:
```

복원할 백업을 지정하지 않고 Enter 키를 누를 경우 []으로 묵인 기본 응답이 선택됩니다. -quit-을 선택하면 restore_repository 스크립트가 종료되고 셸 스크립트로 돌아갑니다.

참고 - `-seed`를 선택하면 `seed` 저장소가 복원됩니다. 이 저장소는 초기 설치 및 업그레이드 중 사용을 위해 마련되었습니다. 복구 목적으로 `seed` 저장소를 사용하는 것은 최후의 수단이 되어야 합니다.

복원할 백업이 선택되면 검증되고 무결성이 검사됩니다. 문제가 있을 경우 `restore_repository` 명령은 오류 메시지를 출력하고 다른 선택을 요구합니다. 유효한 백업이 선택되면 다음 정보가 출력되고 최종 확인 메시지가 표시됩니다.

After confirmation, the following steps will be taken:

```
svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
  -- renamed --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS
/system/volatile/db_errors
  -- copied --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS_errors
repository_to_restore
  -- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).
```

Proceed [yes/no]?

4. 결함을 임시로 수정하려면 **yes**를 입력합니다.

`restore_repository` 명령이 나열된 모든 작업을 실행한 후 시스템이 재부트됩니다.

시작 메시징 양 지정

기본적으로 시스템 부트 시 시작되는 각 서비스는 콘솔에 메시지를 표시하지 않습니다. 다음 메소드의 하나를 사용하여 콘솔에 나타나는 메시지와 `svc.startd` 로그 파일에만 기록되는 메시지를 변경합니다. `logging-level` 값은 아래 표에 표시된 값의 하나일 수 있습니다.

- SPARC 시스템을 부트할 때 `ok` 프롬프트에서 `-m` 옵션을 `boot` 명령으로 지정합니다. [kernel\(1M\)](#) 매뉴얼 페이지의 "메시지 옵션"을 참조하십시오.

```
ok boot -m logging-level
```

- x86 시스템을 부트할 때 GRUB 메뉴를 편집하여 `-m` 옵션을 지정합니다. [“Oracle Solaris 11.2 시스템 부트 및 종료”](#)의 “부트 시 GRUB 메뉴를 편집하여 커널 인수 추가” 및 [kernel\(1M\)](#) 매뉴얼 페이지의 "메시지 옵션"을 참조하십시오.
- 시스템을 재부트하기 전에 `svccfg` 명령을 사용하여 `options/logging` 등록 정보 값을 변경합니다. 이 등록 정보가 시스템에서 변경된 적이 없으면 등록 정보가 종료되지 않고 사용자가 등록 정보를 추가해야 합니다. 다음 예는 자세한 메시징으로 변경됩니다. 변경 사항은 다음에 `svc.startd` 데몬이 다시 시작될 때 적용됩니다.

```
$ svccfg -s system/svc/restarter:default listprop options/logging
$ svccfg -s system/svc/restarter:default addprop options application
```

```
$ svccfg -s system/svc/restarter:default setprop options/logging=verbose
$ svccfg -s system/svc/restarter:default listprop options/logging
options/logging astring      verbose
```

표 A-1 SMF 시작 메시지 로깅 레벨

로깅 레벨 키워드	설명
quiet	관리 개입이 필요한 오류 메시지를 콘솔에 표시합니다. 또한 이러한 메시지를 syslog 및 /var/svc/log/svc.startd.log에 기록합니다.
verbose	quiet 레벨에서 제공되는 메시징 이외에 시작된 각 서비스에 대한 단일 메시지를 콘솔에 표시하고 관리 개입이 필요하지 않은 오류에 대한 경보를 /var/svc/log/svc.startd.log에 기록합니다.
debug	quiet 레벨에서 제공되는 메시징 이외에 시작된 각 서비스에 대한 단일 메시지를 콘솔에 표시하고 svc.startd 디버그 메시지를 /var/svc/log/svc.startd.log에 기록합니다.

부트할 SMF 마일스톤 지정

시스템을 부트할 때 부트할 SMF 마일스톤을 지정할 수 있습니다.

기본적으로 general/enabled 등록 정보 값이 true인 모든 서비스는 시스템 부트 시 시작됩니다. 시스템을 부트할 마일스톤을 변경하려면 다음 메소드의 하나를 사용합니다. *milestone* 값은 표 A-2. “SMF 부트 마일스톤 및 해당 실행 레벨”에 표시된 마일스톤 서비스의 FMRI 또는 키워드일 수 있습니다.

- SPARC 시스템을 부트할 때 ok 프롬프트에서 -m 옵션을 boot 명령으로 지정합니다. [kernel\(1M\)](#) 매뉴얼 페이지의 -m 옵션을 참조하십시오.

```
ok boot -m milestone=milestone
```

- x86 시스템을 부트할 때 GRUB 메뉴를 편집하여 -m 옵션을 지정합니다. “Oracle Solaris 11.2 시스템 부트 및 종료”의 “부트 시 GRUB 메뉴를 편집하여 커널 인수 추가” 및 [kernel\(1M\)](#) 매뉴얼 페이지의 -m 옵션을 참조하십시오.
- 시스템을 재부트하기 전에 svcadm milestone 명령을 -d 옵션과 함께 사용합니다. -d 옵션을 사용하거나 사용하지 않고 이 명령은 실행 중인 서비스를 바로 제한하고 복원합니다. -d 옵션을 사용하면 이 명령은 지정된 마일스톤을 기본 부트 마일스톤으로 설정합니다. 이 새로운 기본값은 재부트 시에도 지속됩니다.

```
$ svcadm milestone -d milestone
```

이 명령은 시스템의 현재 실행 레벨을 변경하지 않습니다. 시스템의 현재 실행 레벨을 변경하려면 init 명령을 사용합니다.

-s 옵션을 지정하면 svcadm은 마일스톤을 변경하고 돌아가기 전에 지정된 마일스톤에 대한 전환이 완료될 때까지 대기합니다. svcadm 명령은 모든 인스턴스가 지정된 마일스톤에 도달하는 데 필요한 상태로 전환되거나 전환하는 데 관리자 개입이 필요하다고 결정될 때

돌아갑니다. -T 옵션을 -s 옵션과 함께 사용하여 마일스톤 변경 작업을 완료하거나 돌아갈 상한(초)을 지정합니다.

다음 표에서는 해당하는 Oracle Solaris 실행 레벨을 포함하여 SMF 부트 마일스톤에 대해 설명합니다. 시스템의 실행 레벨은 사용자가 사용할 수 있는 서비스와 리소스를 정의합니다. 시스템에는 한 번에 하나의 실행 레벨만 지정할 수 있습니다. 실행 레벨에 대한 자세한 내용은 “Oracle Solaris 11.2 시스템 부트 및 종료”의 “실행 레벨 작동 방식”, `inittab(4)` 매뉴얼 페이지 및 `/etc/init.d/README` 파일을 참조하십시오. SMF 부트 마일스톤에 대한 자세한 내용은 `svcadm(1M)` 매뉴얼 페이지의 `milestone` 하위 명령을 참조하십시오.

표 A-2 SMF 부트 마일스톤 및 해당 실행 레벨

SMF 마일스톤 FMRI 또는 키워드	해당 실행 레벨	설명
none		none 키워드는 마스터 다시 시작 프로그램에 대한 서비스를 제외하고 어떤 서비스도 실행되지 않는 마일스톤을 나타냅니다. none을 지정하면 <code>svc:/system/svc/restarter:default</code> 에 대한 서비스를 제외하고 모든 서비스가 일시적으로 사용 안함으로 설정됩니다. none 마일스톤은 시작 문제를 디버깅할 때 유용할 수 있습니다. 특정 지침에 대해서는 시스템 부트 시 서비스 시작 문제를 조사하는 방법 [106] 을 참조하십시오.
all		all 키워드는 모든 서비스에 종속된 마일스톤을 나타냅니다. all을 지정하면 모든 서비스에 대한 일시적 사용 및 사용 안함 요청을 무시합니다. 이는 <code>svc.startd</code> 에서 사용되는 기본 마일스톤입니다.
<code>svc:/milestone/single-user</code>	s 또는 S	<code>svc:/milestone/single-user:default</code> 및 직접적이나 간접적으로 종속되어 있는 모든 서비스에 대한 일시적 사용 및 사용 안함 요청을 무시합니다. 모든 기타 서비스를 일시적으로 사용 안함으로 설정합니다.
<code>svc:/milestone/multi-user</code>	2	<code>svc:/milestone/multi-user:default</code> 및 직접적이나 간접적으로 종속되어 있는 모든 서비스에 대한 일시적 사용 및 사용 안함 요청을 무시합니다. 모든 기타 서비스를 일시적으로 사용 안함으로 설정합니다.
<code>svc:/milestone/multi-user-server</code>	3	<code>svc:/milestone/multi-user-server:default</code> 및 직접적이나 간접적으로 종속되어 있는 모든 서비스에 대한 일시적 사용 및 사용 안함 요청을 무시합니다. 모든 기타 서비스를 일시적으로 사용 안함으로 설정합니다.

시스템이 현재 부트되어 있는 마일스톤을 확인하려면 `svcs` 명령을 사용합니다. 다음 예에서는 시스템이 실행 레벨 3, `milestone/multi-user-server`로 부트되어 있음을 보여줍니다.

```
$ svcs 'milestone/*'
STATE          STIME      FMRI
online         9:08:05   svc:/milestone/unconfig:default
online         9:08:06   svc:/milestone/config:default
online         9:08:07   svc:/milestone/devices:default
online         9:08:25   svc:/milestone/network:default
online         9:08:31   svc:/milestone/single-user:default
online         9:08:51   svc:/milestone/name-services:default
online         9:09:13   svc:/milestone/self-assembly-complete:default
online         9:09:23   svc:/milestone/multi-user:default
online         9:09:24   svc:/milestone/multi-user-server:default
```

SMF를 사용하여 시스템 부트 문제 조사

이 절에서는 시스템이 부트 중에 멈추거나 주요 서비스가 부트 중에 시작되지 않을 경우 수행할 작업을 설명합니다.

▼ 시스템 부트 시 서비스 시작 문제를 조사하는 방법

시스템 부트 시 서비스를 시작할 때 문제가 발생하면 시스템이 부트 중에 멈출 수도 있습니다. 이 절차는 부트 시 발생하는 서비스 문제를 조사하는 방법을 보여줍니다.

1. 서비스를 시작하지 않고 부트합니다.

다음 명령은 `svc.startd` 데몬이 모든 서비스를 일시적으로 사용 안함으로 설정하고 콘솔에서 `sulogin`을 시작하도록 지시합니다.

```
ok boot -m milestone=none
```

`boot -m` 명령과 함께 사용할 수 있는 SMF 마일스톤 목록은 [“부트할 SMF 마일스톤 지정” \[104\]](#)을 참조하십시오.

2. `root`로 시스템에 로그인합니다.

3. 모든 서비스를 사용으로 설정합니다.

```
# svcadm milestone all
```

4. 부트 프로세스가 어디에서 멈추는지 확인합니다.

부트 프로세스가 멈출 때 `svcs -a`를 실행하여 어떤 서비스가 실행 중이 아닌지 확인합니다. `/var/svc/log`의 로그 파일에서 오류 메시지를 찾습니다.

5. 문제를 수정한 후 모든 서비스가 시작되었는지 확인합니다.

a. 필요한 모든 서비스가 온라인인지 확인합니다.

```
# svcs -x
```

b. `console-login` 서비스 종속성이 충족되었는지 확인합니다.

이 명령은 콘솔에서 `login` 프로세스가 실행되는지 확인합니다.

```
# svcs -l system/console-login:default
```

6. 정상적인 부트 프로세스를 계속합니다.

▼ 로컬 파일 시스템 서비스가 부트 중에 실패하는 경우 단일 사용자 로그인을 강제 수행하는 방법

시스템을 부트하는 데 필요하지 않은 로컬 파일 시스템은 `svc:/system/filesystem/local:default` 서비스로 마운트됩니다. 해당 파일 시스템을 마운트할 수 없는 경우 `filesystem/local` 서비스가 유지 관리 상태로 전환됩니다. 시스템 시작이 계속 진행되며 `filesystem/local`에 종속되지 않은 모든 서비스가 시작됩니다. `filesystem/local` 서비스에 대한 필수 종속성이 있는 서비스는 시작되지 않습니다.

이 절차에서는 서비스 실패 직후 시스템 시작이 계속되도록 두는 대신 `sulogin` 프롬프트가 나타나도록 시스템 구성을 변경하는 방법을 보여줍니다.

1. `system/console-login` 서비스를 수정합니다.

```
$ svccfg -s svc:/system/console-login
svc:/system/console-login> addpg site,filesystem-local dependency
svc:/system/console-login> setprop site,filesystem-local/entities = fmri: svc:/system/filesystem/local
svc:/system/console-login> setprop site,filesystem-local/grouping = astring: require_all
svc:/system/console-login> setprop site,filesystem-local/restart_on = astring: none
svc:/system/console-login> setprop site,filesystem-local/type = astring: service
svc:/system/console-login> end
```

2. 서비스를 새로 고칩니다.

```
$ svcadm refresh console-login
```

`system/filesystem/local:default` 서비스에서 실패가 발생하면 `svcs -vx` 명령을 사용하여 실패를 식별합니다. 실패가 해결된 후 다음 명령을 사용하여 오류 상태를 지우고 시스템 부트를 계속할 수 있습니다.

```
$ svcadm clear filesystem/local
```

inetd 서비스를 SMF 서비스로 변환

시스템의 `inetd.conf` 파일에는 항목이 포함되지 않아야 합니다. `inetd.conf` 파일에는 이 파일에 더는 직접 사용되지 않는 레거시 파일이라는 주석만 포함되어야 합니다. `inetd.conf` 파일에 항목이 포함되어 있으면 이 절의 지침에 따라 이러한 구성을 SMF 서비스로 변환합니다. `inetd.conf` 파일에 구성되어 있지만 SMF 서비스로 구성되지 않은 서비스는 사용할 수 없습니다. `inetd.conf` 파일에 구성되어 있는 서비스는 `inetd` 명령으로 직접 다시 시작되지 않습니다. 대신에 `inetd` 명령이 변환된 서비스에 대한 위임된 다시 시작 프로그램입니다.

초기 시스템 부트 중에 `inetd.conf` 파일의 구성이 SMF 서비스로 자동으로 변환됩니다. 초기 시스템 부트 후에 IPS(Image Packaging System, 이미지 패키징 서비스) 패키지가 제공하지 않는 추가 소프트웨어를 설치하여 `inetd.conf` 파일에 항목을 추가할 수 있습니다. IPS 패

키지가 제공하는 소프트웨어에는 필수 SMF 매니페스트가 포함되고 해당 SMF 매니페스트는 올바른 등록 정보 값을 사용하여 해당 서비스 인스턴스를 인스턴스화합니다.

시스템의 `inetd.conf` 파일에 항목이 포함되어 있으면 `inetconv` 명령을 사용하여 해당 구성을 SMF 서비스로 변환합니다. `inetconv` 명령은 `inetd.conf` 항목을 SMF 서비스 매니페스트 파일로 변환하고 해당 매니페스트를 SMF 저장소로 가져와 서비스 인스턴스를 인스턴스화합니다. 명령 옵션에 대한 자세한 내용과 명령 사용 예를 보려면 [inetconv\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

새 SMF 매니페스트의 이름은 `inetd.conf` 항목의 `service_name`을 통합합니다. `inetd.conf` 파일의 항목은 새 서비스 인스턴스의 등록 정보로 저장됩니다. 새 SMF 매니페스트는 등록 정보 그룹과 등록 정보를 지정하여 `inetd.conf` 항목에 나열된 작업을 정의합니다. `inetconv` 명령을 실행한 후 `svcs` 및 `svccprop` 명령을 사용하여 새 서비스 인스턴스가 생성되고 올바른 등록 정보 값을 포함하는지 확인합니다.

`inetd` 명령은 SMF 인터넷 서비스에 대한 위임된 다시 시작 프로그램입니다. 이러한 서비스를 관리하는 데 `inetd` 명령을 직접 사용하지 마십시오. `inetadm` 명령을 옵션이나 피연산자 없이 사용하여 `inetd`로 제어되는 서비스 목록을 확인합니다. `inetadm`, `svcadm` 및 `svccfg` 명령을 사용하여 이 변환된 서비스를 구성하고 관리합니다.

`inetconv` 명령은 입력 `inetd.conf` 파일을 수정하지 않습니다. `inetconv`를 성공적으로 실행한 후 `inetd.conf` 파일에서 항목을 수동으로 삭제해야 합니다.

이미 SMF 서비스로 변환된 `inetd` 서비스를 구성하는 방법에 대한 자세한 내용은 [“inetd로 제어되는 서비스 수정” \[75\]](#)을 참조하십시오.

색인

번호와 기호

- admin 계층, 24, 45
- ASM, 81
- ASM(Automatic Storage Management), 81
- boot 명령
 - none 마일스톤, 106
 - 마일스톤 옵션, 104
- boot 백업, 25
- config 등록 정보 그룹, 22
- degraded 서비스 상태, 20, 97, 100
- describe 하위 명령
 - svccfg 명령, 39
- disabled 서비스 상태, 20
- DTD, 79
- /etc/inetd.conf 파일, 26, 76, 107
- /etc/init.d 스크립트, 26
- /etc/inittab 항목, 26
- /etc/rc?.d 스크립트, 26
- Fault Management Resource Identifier 살펴볼 내용 FMRI
- FMRI, 19, 88
- general 등록 정보 그룹, 22
- inetadm 명령, 107
 - 예, 75
- inetconv 명령, 107
- inetd 명령, 107
- inetd 서비스
 - SMF 서비스로 변환, 107
 - 등록 정보 수정, 75
 - 명령줄 인수 수정, 75
- inetd.conf 파일, 26, 107
- init.d 스크립트, 26
- initial 스냅샷, 25
- inittab 항목, 26
- Kerberos
 - 스텐실 서비스 예, 94
- legacy_run 서비스 상태, 29
- libscf 라이브러리, 23, 89
 - 일괄 처리 작업 기능, 16
- listcust 하위 명령
 - svccfg 명령, 41, 45
- listpg 하위 명령
 - svccfg 명령, 42
- listprop 하위 명령
 - svccfg 명령, 41
- listsnap 하위 명령
 - svccfg 명령, 44
- lrc 서비스, 26
- maintenance 서비스 상태, 20, 97, 100
- manifest 계층, 24, 45
- manifest-import 백업, 25
- manifest-import 서비스 다시 시작, 54
- offline 서비스 상태, 20, 97
- online 서비스 상태, 20
- Oracle 데이터베이스, 81
 - ASM, 81
 - 데이터베이스 서비스, 82
 - 리스너 서비스, 84
- previous 스냅샷, 25
- Puppet
 - 스텐실 서비스 예, 90
- rc?.d 스크립트, 26
- resolv.conf 파일, 26
- restarter 등록 정보 그룹, 22
- restore_repository 명령, 101
- running 스냅샷, 25
- selectsnap 하위 명령
 - svccfg 명령, 41
- site-profile 계층, 24, 45
- SMF 데이터베이스 살펴볼 내용 서비스 구성 저장소
- start 등록 정보 그룹, 22
- start 스냅샷, 25

- STATE 열, 29
 - sulogin 명령, 101
 - svc.configd 저장소 데몬, 100
 - svc.startd 마스터 다시 시작 프로그램 데몬, 21, 101
 - svcadm 명령, 49
 - clear 하위 명령, 97
 - disable 하위 명령, 52
 - enable 하위 명령, 49
 - mark 하위 명령, 100
 - milestone 하위 명령, 104, 106
 - refresh 하위 명령, 55
 - restart manifest-import 하위 명령, 54, 56
 - restart 하위 명령, 54, 97
 - 동기 옵션, 16
 - svcbundle 명령
 - rc-script 서비스, 88
 - 매니페스트 만들기, 79
 - 자동 설치, 74, 81
 - 프로파일 만들기, 73
 - svccfg 명령
 - add 하위 명령, 71
 - addpg 하위 명령, 66
 - addpropvalue 하위 명령, 62, 66
 - apply 하위 명령, 72
 - delcust -M 하위 명령, 70
 - delcust 하위 명령, 68, 70
 - delete 하위 명령, 56
 - delpg 하위 명령, 68
 - delprop 하위 명령, 68
 - delpropvalue 하위 명령, 68
 - describe 하위 명령, 39
 - editprop 하위 명령, 60
 - extract 하위 명령, 74
 - import 하위 명령, 72
 - listcust -M 하위 명령, 56, 70
 - listcust 하위 명령, 41, 45
 - listpg 하위 명령, 42
 - listprop 하위 명령, 41
 - listsnap 하위 명령, 44
 - refresh 하위 명령, 55
 - revert 하위 명령, 72
 - selectsnap 하위 명령, 41, 72
 - setenv 하위 명령, 62, 65
 - setprop 하위 명령, 62, 66, 68
 - unsetenv 하위 명령, 65
 - validate 하위 명령, 80
 - 대화식 사용, 61
 - 등록 정보 표시, 39
 - 명령 입력 파일, 61
 - svcio 유틸리티, 16, 89
 - svcprop 명령
 - 등록 정보 표시, 39, 40
 - svcs 명령, 29
 - system-profile 계층, 24
 - uninitialized 서비스 상태, 20
- ㄱ
- 계약 서비스, 19
 - 계층, 24
 - admin 계층, 45
 - manifest 계층, 45
 - site-profile 계층, 45
- 구성
- 등록 정보 값 삭제, 68
 - 등록 정보 값 설정, 62
 - 등록 정보 값 추가, 62
 - 등록 정보 그룹 및 등록 정보 추가, 66
 - 등록 정보 그룹 삭제, 68
 - 등록 정보 삭제, 68
 - 마스킹, 70
 - 마스킹 해제, 70
 - 번들 지원 구성 삭제, 70
 - 사용자 정의 삭제, 68, 70
 - 새로 고침, 55
 - 수정, 59
 - 구성 저장소 살펴볼 내용 서비스 구성 저장소
 - 구성 파일, 15, 16, 26, 89
 - 구성된 보기, 23, 40
 - 권한, 27, 27
 - 권한 부여, 27
 - 권한 프로파일, 27
 - 기본 다시 시작 프로그램, 21
- ㄴ
- 다시 시작 프로그램, 18, 21
 - svc.startd 마스터 다시 시작 프로그램 데몬, 21
 - 기본 다시 시작 프로그램, 21
 - 위임된 다시 시작 프로그램, 22

단일 사용자 로그인, 101
 대기 서비스, 19
 데몬 서비스, 19
 등록 정보, 22, 23, 39
 구성된 보기, 23, 40
 비영구, 42
 실행 중인 스냅샷, 40
 이름 지정, 87
 등록 정보 그룹, 22, 39
 비영구, 42
 유형, 42, 88
 이름 지정, 87

ㄹ

로그 파일, 37

ㄴ

마스킹된 구성, 70
 마스킹된 엔티티, 45, 56
 마스터 다시 시작 프로그램 데몬, 21
 마일스톤
 none, 106
 부트, 104
 해당 실행 레벨, 104
 현재 부트됨, 105
 마일스톤 서비스, 18
 매니페스트, 18
 site 디렉토리, 79
 표준 위치, 23, 79
 형식, 79
 메소드, 18

ㄷ

백업, 25
 번들, 23
 번들 지원, 70
 부트
 SMF 마일스톤, 106
 로깅 레벨, 103
 마일스톤, 104
 서비스 시작 제외, 106
 현재 마일스톤, 105

비영구 등록 정보, 43
 비영구 등록 정보 그룹, 42

ㄸ

사용자 정의
 나열, 45
 상태 전환 알림, 57
 서비스, 18
 구성 새로 고침, 55
 구성 수정, 59
 다시 시작, 54
 사용 안함, 52
 사용으로 설정, 49
 삭제, 56
 시작, 49
 이름 지정, 19, 87
 중지, 52
 서비스 계약, 19
 서비스 구성 기능 라이브러리, libscf, 23
 서비스 구성 새로 고침, 55
 서비스 구성 업데이트, 55
 서비스 구성 저장소, 18, 23
 svc.configd 데몬, 100
 계층, 24
 라이브러리 인터페이스, 23
 백업, 25
 복구, 100
 수정, 59
 스냅샷, 25
 템플릿 데이터, 23
 서비스 다시 시작, 54
 서비스 다시 시작 프로그램, 21
 서비스 등록 정보, 22, 23, 39
 서비스 등록 정보 그룹, 22, 39
 서비스 모델, 19
 서비스 번들
 DTD, 79
 표준 위치, 23
 서비스 사용 안함, 52
 서비스 삭제, 56
 서비스 상태
 나열, 29
 설명, 20
 수동 변경, 97, 100
 전환, 20

- 서비스 시작, 49
- 서비스 인스턴스, 18
 - 추가, 71
- 서비스 제거, 56
- 서비스 중지, 52
- 서비스를 사용으로 설정, 49
- 스냅샷, 41, 44
 - previous, 25
 - running, 25
 - start, 25
 - 되돌리기, 72
 - 실행 구성 업데이트, 55
 - 실행 중인 스냅샷, 40
 - 초기, 25
- 스텐실 서비스, 89
 - Kerberos 예, 94
 - Puppet 예, 90
- 스텐실 파일, 16, 89, 89
- 실행 레벨
 - 기본 레벨, 104
 - 해당 SMF 마일스톤, 104
 - 현재, 105
- 실행 제어 스크립트, 26
 - SMF 서비스로 변환, 88
- 실행 중인 스냅샷, 40

○

- 알림 구성, 57
- 알림 매개변수
 - 표시, 45
- 역할, 27
- 오류 로깅, 37
- 위임된 다시 시작 프로그램, 22
- 이벤트 알림, 57
- 인스턴스, 18
 - 이름 지정, 19, 87
 - 추가, 71
- 일시적 서비스, 19

ㄹ

- 작업, 18
- 저장소 살펴볼 내용 서비스 구성 저장소
- 종속성, 20, 20

- 그룹, 33
- 나열, 32
- 서비스가 종속된 인스턴스, 33
- 서비스에 종속된 인스턴스, 34
- 종속 서비스 상태에 대한 영향, 35

ㄷ

- 초기 상태 살펴볼 내용 실행 레벨

ㅌ

- 템플릿 데이터, 23

ㅍ

- 프로파일, 18
 - site 디렉토리, 23, 73
 - 만들기, 73
 - 표준 위치, 23

ㅎ

- 하위 서비스, 19
- 환경 변수
 - 메소드로 수정, 65