

Oracle® VM Server for SPARC 2.2 관리 설명서

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

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.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

목차

머리말	15
제1부 Oracle VM Server for SPARC 2.2 소프트웨어	19
1 Oracle VM Server for SPARC 소프트웨어 개요	21
하이퍼바이저 및 Logical Domains	22
Logical Domains Manager	24
도메인의 역할	24
명령줄 인터페이스	25
가상 입/출력	25
리소스 구성	26
지속적 구성	26
Oracle VM Server for SPARC Physical-to-Virtual 변환 도구	27
Oracle VM Server for SPARC Configuration Assistant	27
Oracle VM Server for SPARC Management Information Base	28
2 소프트웨어 설치 및 사용	29
새 시스템에 Oracle VM Server for SPARC 소프트웨어 설치	30
Oracle Solaris OS 업데이트	30
시스템 펌웨어 업그레이드	31
Logical Domains Manager 다운로드	31
Logical Domains Manager 설치	32
Logical Domains Manager 데몬 사용	34
Oracle VM Server for SPARC를 이미 사용 중인 시스템 업그레이드	35
Oracle Solaris OS 업그레이드	35
Logical Domains Manager 및 시스템 펌웨어 업그레이드	37
Oracle VM Server for SPARC 2.2 소프트웨어로 업그레이드	37

출하 시 기본 구성과 Logical Domains 사용 안함	40
▼ 모든 게스트 도메인을 제거하는 방법	40
▼ 모든 논리적 도메인 구성을 제거하는 방법	40
▼ 출하 시 기본 구성을 복원하는 방법	41
▼ Logical Domains Manager를 사용 안함으로 설정하는 방법	41
▼ Logical Domains Manager를 제거하는 방법	42
▼ 서비스 프로세서에서 출하 시 기본 구성을 복원하는 방법	42
3 Oracle VM Server for SPARC 보안	43
RBAC를 사용하여 논리적 도메인의 관리 위임	43
권한 프로파일 및 역할 사용	44
Logical Domains Manager 프로파일 콘텐츠	46
RBAC를 사용하여 도메인 콘솔에 대한 액세스 제어	47
▼ 역할을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법	48
▼ 권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법	50
▼ 역할을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법	51
▼ 권한 프로파일을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법	53
감사를 사용으로 설정한 후 사용	53
▼ 감사를 사용으로 설정하는 방법	54
▼ 감사를 사용 안함으로 설정하는 방법	55
▼ 감사 레코드를 검토하는 방법	56
▼ 감사 로그를 교체하는 방법	56
4 서비스 및 컨트롤 도메인 설정	57
출력 메시지	57
기본 서비스 만들기	58
▼ 기본 서비스를 만드는 방법	58
컨트롤 도메인의 초기 구성	59
▼ 컨트롤 도메인을 설정하는 방법	60
Logical Domains를 사용하도록 재부트	61
▼ 재부트하는 방법	61
컨트롤/서비스 도메인과 다른 도메인 사이의 네트워킹 사용	61
▼ 가상 스위치를 주 인터페이스로 구성하는 방법	62
가상 네트워크 터미널 서버 데몬 사용	62
▼ 가상 네트워크 터미널 서버 데몬을 사용으로 설정하는 방법	63

5	게스트 도메인 설정	65
	게스트 도메인 만들기 및 시작	65
	▼ 게스트 도메인을 만들고 시작하는 방법	65
	게스트 도메인에 Oracle Solaris OS 설치	68
	▼ DVD에서 게스트 도메인에 Oracle Solaris OS를 설치하는 방법	68
	▼ Oracle Solaris ISO 파일에서 게스트 도메인에 Oracle Solaris OS를 설치하는 방법	70
	▼ Oracle Solaris 10 게스트 도메인에서 Oracle Solaris JumpStart 기능을 사용하는 방법	71
6	I/O 도메인 설정	73
	I/O 도메인 개요	73
	일반적인 I/O 도메인 만들기 지침	74
	PCIe 버스 지정	74
	▼ PCIe 버스를 지정하여 I/O 도메인을 만드는 방법	76
	PCIe 끝점 장치 지정	79
	직접 I/O 하드웨어 및 소프트웨어 요구 사항	81
	현재 직접 I/O 기능 제한 사항	81
	PCIe 끝점 장치 구성 계획	82
	primary 도메인 재부트	83
	PCIe 하드웨어 변경	84
	▼ PCIe 끝점 장치를 지정하여 I/O 도메인을 만드는 방법	85
	PCIe SR-IOV 가상 기능 사용	90
	SR-IOV 개요	90
	SR-IOV 하드웨어 및 소프트웨어 요구 사항	92
	현재 SR-IOV 기능 제한 사항	92
	PCIe SR-IOV 가상 기능 사용 계획	93
	가상 기능 만들기, 수정 및 삭제	94
	I/O 도메인에서 가상 기능 추가 및 제거	98
	SR-IOV: primary 도메인 재부트	99
	SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기	99
	고급 SR-IOV 항목	102
7	가상 디스크 사용	107
	가상 디스크 소개	107
	가상 디스크 관리	108

▼ 가상 디스크를 추가하는 방법	108
▼ 가상 디스크 백엔드를 여러 번 내보내는 방법	109
▼ 가상 디스크 옵션을 변경하는 방법	110
▼ 시간 초과 옵션을 변경하는 방법	110
▼ 가상 디스크를 제거하는 방법	110
가상 디스크 식별자 및 장치 이름	110
가상 디스크 표시	111
전체 디스크	111
단일 슬라이스 디스크	111
가상 디스크 백엔드 옵션	112
읽기 전용(ro) 옵션	112
배타적(excl) 옵션	112
슬라이스(slice) 옵션	113
가상 디스크 백엔드	114
물리적 디스크 또는 디스크 LUN	114
▼ 물리적 디스크를 가상 디스크로 내보내는 방법	114
물리적 디스크 슬라이스	115
▼ 물리적 디스크 슬라이스를 가상 디스크로 내보내는 방법	115
▼ 슬라이스 2를 내보내는 방법	116
파일 및 볼륨	116
가상 디스크 다중 경로 구성	120
가상 디스크 다중 경로 및 가상 디스크 시간 초과	121
▼ 가상 디스크 다중 경로를 구성하는 방법	121
CD, DVD 및 ISO 이미지	122
▼ CD 또는 DVD를 서비스 도메인에서 게스트 도메인으로 내보내는 방법	123
▼ primary 도메인에서 ISO 이미지를 내보내서 게스트 도메인을 설치하는 방법	124
가상 디스크 시간 초과	126
가상 디스크 및 SCSI	127
가상 디스크 및 format 명령	127
가상 디스크에서 ZFS 사용	127
서비스 도메인에서 ZFS 풀 구성	128
ZFS를 사용하여 디스크 이미지 저장	128
디스크 이미지의 스냅샷 만들기	129
복제본을 사용하여 새 도메인 프로비전	130
Logical Domains 환경에서 볼륨 관리자 사용	131
볼륨 관리자 위에서 가상 디스크 사용	131

가상 디스크 위에서 볼륨 관리자 사용	133
8 가상 네트워크 사용	135
가상 네트워크에 대한 소개	136
Oracle Solaris 10 네트워킹 개요	136
Oracle Solaris 11 네트워킹 개요	138
가상 스위치	141
가상 네트워크 장치	142
Vnet 간 LDC 채널	142
가상 장치 식별자 및 네트워크 인터페이스 이름	144
▼ Oracle Solaris OS 네트워크 인터페이스 이름을 찾는 방법	145
자동 또는 수동 MAC 주소 지정	146
Logical Domains에 지정된 MAC 주소의 범위	147
자동 지정 알고리즘	147
중복된 MAC 주소 감지	148
해제된 MAC 주소	148
Logical Domains에서 네트워크 어댑터 사용	149
▼ GLDv3 준수 네트워크 어댑터인지 확인하는 방법(Oracle Solaris 10)	149
NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성	150
Oracle Solaris 10 시스템에서 NAT 구성	150
Oracle Solaris 11 시스템에서 NAT 구성	152
Logical Domains 환경에서 IPMP 구성	154
도메인에서 가상 네트워크 장치를 IPMP 그룹으로 구성	154
서비스 도메인에서 IPMP 구성 및 사용	155
Logical Domains 가상 네트워킹에서 링크 기반 IPMP 사용	156
Logical Domains 1.3 이전 릴리스에서 IPMP 구성 및 사용	160
VLAN 태그 지정 사용	162
PVID(포트 VLAN ID)	162
VID(VLAN ID)	163
▼ 가상 스위치 및 가상 네트워크 장치에 VLAN을 지정하는 방법	163
▼ 설치 서버가 VLAN에 있을 때 게스트 도메인을 설치하는 방법	165
NIU 하이브리드 I/O 사용	165
▼ NIU 네트워크 장치로 가상 스위치를 구성하는 방법	168
▼ 하이브리드 모드를 사용으로 설정하는 방법	169
▼ 하이브리드 모드를 사용 안함으로 설정하는 방법	169

가상 스위치에서 링크 통합 사용	169
점보 프레임 구성	171
▼ 점보 프레임을 사용하도록 가상 네트워크 및 가상 스위치 장치를 구성하는 방법 ..	171
vnet 및 vsw 드라이버의 이전(점보 미인식) 버전과의 호환성(Oracle Solaris 10)	174
Oracle Solaris 11 네트워킹 관련 기능 차이점	175
9 도메인 마이그레이션	177
도메인 마이그레이션 소개	178
마이그레이션 작업 개요	178
소프트웨어 호환성	179
마이그레이션 작업 보안	179
도메인 마이그레이션	180
Dry Run 수행	180
비대화식 마이그레이션 수행	180
활성 도메인 마이그레이션	181
CPU에 대한 도메인 마이그레이션 요구 사항	181
메모리에 대한 마이그레이션 요구 사항	182
물리적 I/O 장치에 대한 마이그레이션 요구 사항	183
가상 I/O 장치에 대한 마이그레이션 요구 사항	183
NIU 하이브리드 I/O에 대한 마이그레이션 요구 사항	184
암호화 장치에 대한 마이그레이션 요구 사항	184
활성 도메인의 재구성 지연	185
활성 도메인에 전원 관리 탄력적 정책이 적용되는 동안 마이그레이션	185
다른 도메인에 대한 작업	185
OpenBoot PROM의 도메인 또는 커널 디버거에서 실행 중인 도메인 마이그레이션	185
바인딩된 도메인 또는 비활성 도메인 마이그레이션	186
가상 I/O 장치에 대한 마이그레이션 요구 사항	186
PCIe 끝점 장치에 대한 마이그레이션 요구 사항	186
진행 중인 마이그레이션 모니터	186
진행 중인 마이그레이션 취소	187
실패한 마이그레이션 복구	188
마이그레이션 예	188

10 리소스 관리	191
리소스 재구성	191
동적 재구성	191
지연된 재구성	192
리소스 할당	193
CPU 할당	193
▼ 전체 코어 제약 조건을 적용하는 방법	193
전체 코어 제약 조건과 다른 도메인 기능 사이의 상호 작용	195
SPARC CPU를 조정하여 SPARC T4 시스템에서 작업 로드 성능 최적화	197
CPU 스레드 모드 및 작업 로드	197
스레드 제어 제한 사항	199
하드 분할을 사용하여 시스템 구성	200
도메인의 구성 확인	200
CPU 전체 코어로 도메인 구성	201
다른 Oracle VM Server for SPARC 기능과의 상호 작용	204
도메인에 물리적 리소스 지정	206
제어 도메인에서 물리적 리소스 관리	208
도메인에서 물리적 리소스 관리 제한 사항	208
메모리 동적 재구성 사용	209
메모리 추가	209
메모리 제거	210
일부 메모리 DR 요청	210
제어 도메인의 메모리 재구성	210
동적 및 지연된 재구성	211
메모리 정렬	211
메모리 DR 예	213
전원 관리 사용	216
전원 관리되는 CPU 스레드 및 가상 CPU 나열	218
동적 리소스 관리 사용	221
도메인 리소스 나열	224
시스템에서 읽을 수 있는 출력	224
플래그 정의	224
사용률 통계 정의	225
다양한 목록 보기	225
제약 조건 나열	229

11 도메인 구성 관리	231
나중에 재작성을 위해 도메인 구성 저장	231
▼도메인 구성을 저장하는 방법	232
▼XML 파일에서 도메인 구성을 복원하는 방법(ldm add-domain)	232
▼XML 파일에서 도메인 구성을 복원하는 방법(ldm init-system)	233
Logical Domains 구성 관리	234
▼자동 복구 정책을 수정하는 방법	235
12 기타 관리 작업 수행	237
CLI에서 이름 입력	237
파일 이름(file) 및 변수 이름(var-name)	237
가상 디스크 서버 backend 및 가상 스위치 장치 이름	237
구성 이름(config-name)	238
기타 모든 이름	238
네트워크를 통해 게스트 콘솔에 연결	238
콘솔 그룹 사용	239
▼하나의 그룹에 다중 콘솔을 결합하는 방법	239
부하가 높은 도메인을 중지할 때 시간 초과가 발생할 수 있음	239
Oracle VM Server for SPARC와 함께 Oracle Solaris OS 작동	240
Oracle Solaris OS가 시작된 후 OpenBoot 펌웨어를 사용할 수 없음	240
서버 전원 껐다 켜기	240
전원 관리 도메인에서 활성 CPU에 대해 psradm(1M) 명령을 사용하지 않아야 함 ..	241
Oracle Solaris OS 중단 결과	241
컨트롤 도메인 정지 또는 재부트 결과	241
서비스 프로세서와 함께 Logical Domains 사용	242
▼도메인 구성을 기본 또는 다른 구성으로 재설정하는 방법	242
도메인 종속성 구성	242
도메인 종속성 예	244
종속성 주기	245
CPU 및 메모리 주소를 매핑하여 오류 발생 위치 확인	246
CPU 매핑	246
메모리 매핑	247
CPU 및 메모리 매핑 예	247
UUID(Universally Unique Identifier) 사용	248
가상 도메인 정보 명령 및 API	249

제2부 선택적 Oracle VM Server for SPARC 소프트웨어	251
13 Oracle VM Server for SPARC Physical-to-Virtual 변환 도구	253
Oracle VM Server for SPARC P2V 도구 개요	253
수집 단계	254
준비 단계	254
변환 단계	255
백엔드 장치	256
Oracle VM Server for SPARC P2V 도구 설치	257
필수 조건	257
제한 사항	257
▼ Oracle VM Server for SPARC P2V 도구를 설치하는 방법	258
ldmp2v 명령 사용	259
14 Oracle VM Server for SPARC Configuration Assistant(Oracle Solaris 10)	267
Configuration Assistant 사용(ldmconfig)	267
Configuration Assistant 설치	267
ldmconfig 기능	268
15 Oracle VM Server for SPARC Management Information Base 소프트웨어 사용	271
Oracle VM Server for SPARC Management Information Base 개요	272
소프트웨어 구성 요소	272
System Management Agent	273
Logical Domains Manager 및 Oracle VM Server for SPARC MIB	274
Oracle VM Server for SPARC MIB 객체 트리	274
Oracle VM Server for SPARC MIB 소프트웨어 설치 및 구성	275
Oracle VM Server for SPARC MIB 소프트웨어 설치 및 구성(작업 맵)	275
보안 관리	277
▼ 초기 snmpv3 사용자를 만드는 방법	277
도메인 모니터링	278
환경 변수 설정	279
Oracle VM Server for SPARC MIB 질의	279
Oracle VM Server for SPARC MIB 정보 검색	281
SNMP 트랩 사용	298

Oracle VM Server for SPARC MIB 모듈 트랩 사용	298
Oracle VM Server for SPARC MIB 트랩 설명	299
도메인 시작 및 중지	304
도메인 시작 및 중지	304
16 Logical Domains Manager 검색	309
Logical Domains Manager를 실행 중인 시스템 검색	309
멀티캐스트 통신	309
메시지 형식	309
▼ 서버넷에서 실행 중인 Logical Domains Manager를 검색하는 방법	310
17 Logical Domains Manager에서 XML 인터페이스 사용	313
XML 전송	313
XMPP 서버	314
로컬 연결	314
XML 프로토콜	314
요청 및 응답 메시지	315
이벤트 메시지	318
등록 및 등록 해제	318
<LDM_event> 메시지	319
이벤트 유형	320
Logical Domains Manager 작업	323
Logical Domains Manager 리소스 및 등록 정보	324
도메인 정보(ldom_info) 리소스	325
CPU(cpu) 리소스	326
MAU(mau) 리소스	327
메모리(memory) 리소스	327
가상 디스크 서버(vds) 리소스	328
가상 디스크 서버 볼륨(vds_volume) 리소스	328
디스크(disk) 리소스	329
가상 스위치(vsw) 리소스	329
네트워크(network) 리소스	331
가상 콘솔 집중기(vcc) 리소스	331
변수(var) 리소스	332
물리적 I/O 장치(physio_device) 리소스	333

SP 구성(spconfig) 리소스	333
DRM 정책 구성(policy) 리소스	334
가상 데이터 플레인 채널 서비스(vdpcs) 리소스	334
가상 데이터 플레인 채널 클라이언트(vdpcc) 리소스	335
콘솔(console) 리소스	336
도메인 마이그레이션	336
XML 스키마	337
 용어집	 339
 색인	 349

머리말

Oracle VM Server for SPARC 2.2 관리 설명서에서는 지원되는 서버, 블레이드 및 서버 모듈에서의 Oracle VM Server for SPARC 2.2 소프트웨어에 대한 개요, 보안 고려 사항, 설치, 구성, 수정 및 일반 작업 실행을 설명하는 세부 정보와 절차를 제공합니다. **Oracle VM Server for SPARC 2.2 릴리스 노트**의 “지원되는 플랫폼”를 참조하십시오.

본 설명서는 이러한 서버에서 UNIX 시스템 및 Oracle Solaris OS(Oracle Solaris 운영 체제)를 사용할 수 있는 실제적인 지식이 있는 시스템 관리자를 대상으로 작성되었습니다.

관련 문서

다음 표에서는 Oracle VM Server for SPARC 2.2 릴리스에 사용 가능한 설명서를 보여줍니다. 특별한 언급이 없는 한 이러한 문서는 HTML 형식과 PDF 형식으로 제공됩니다.

표 P-1 관련 문서

응용 프로그램	제목
Oracle VM Server for SPARC 2.2 소프트웨어	Oracle VM Server for SPARC 2.2 관리 설명서
	Oracle VM Server for SPARC 2.2 보안 설명서
	Oracle VM Server for SPARC 2.2 Reference Manual
	Oracle VM Server for SPARC 2.2 릴리스 노트
Oracle VM Server for SPARC 2.2 drd(1M) 및 vntsd(1M) 매뉴얼 페이지	Oracle Solaris OS 참조 설명서: <ul style="list-style-type: none">■ Oracle Solaris 10 Documentation■ Oracle Solaris 11 Documentation
	Oracle Solaris OS 설치 및 구성
Oracle Solaris OS: 설치 및 구성	Oracle Solaris OS 설치 및 구성 설명서: <ul style="list-style-type: none">■ Oracle Solaris 10 Documentation■ Oracle Solaris 11 Documentation

표 P-1 관련 문서 (계속)

응용 프로그램	제목
Oracle VM Server for SPARC 및 Oracle Solaris OS 보안	Oracle VM Server for SPARC 백서 및 Oracle Solaris OS 보안 설명서: <ul style="list-style-type: none">■ Secure Deployment of Oracle VM Server for SPARC (http://www.oracle.com/technetwork/articles/systems-hardware-architecture/secure-ovm-sparc-deployment-294062.pdf)■ Oracle Solaris 10 보안 지침■ Oracle Solaris 11 보안 지침

사용 중인 서버, 소프트웨어 또는 Oracle Solaris OS 관련 설명서는 <http://www.oracle.com/technetwork/indexes/documentation/index.html>에서 찾을 수 있습니다. 검색 상자를 이용하여 필요한 문서와 정보를 찾으십시오.

Oracle Support에 액세스

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

활자체 규약

다음 표는 이 책에서 사용되는 활자체 규약에 대해 설명합니다.

표 P-2 활자체 규약

활자체	의미	예
AaBbCc123	명령 및 파일, 디렉토리 이름; 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오. 모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오. machine_name% you have mail.
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	machine_name% su Password:
aabbcc123	새로 나오는 용어, 강조 표시할 용어입니다. 명령줄 변수를 실제 이름이나 값으로 바꾸십시오.	<code>rm filename</code> 명령을 사용하여 파일을 제거합니다.

표 P-2 활자체 규약 (계속)

활자체	의미	예
AaBbCc123	책 제목, 장, 절	사용자 설명서의 6장을 읽으십시오. 캐시는 로컬로 저장된 복사본입니다. 파일을 저장하면 안 됩니다 . 주: 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

명령 예의 셸 프롬프트

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

표 P-3 셸 프롬프트

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

제 1 부

Oracle VM Server for SPARC 2.2 소프트웨어

Oracle's SPARC T-Series 서버에 매우 효율적인 엔터프라이즈급 가상화 기능을 제공하는 Oracle VM Server for SPARC 2.2 소프트웨어를 소개합니다.

Oracle VM Server for SPARC 소프트웨어 개요

이 장에서는 Oracle VM Server for SPARC 소프트웨어의 개요를 제공합니다.

Oracle VM Server for SPARC 소프트웨어는 특정 Oracle Solaris OS 버전, 필요한 소프트웨어 패치 및 특정 버전의 시스템 펌웨어에 따라 다릅니다. 자세한 내용은 [Oracle VM Server for SPARC 2.2 릴리스 노트](#)의 “필수 및 권장 Oracle Solaris OS”를 참조하십시오.

Oracle VM Server for SPARC는 Oracle's SPARC T-Series 서버에 매우 효율적인 엔터프라이즈급 가상화 기능을 제공합니다. Oracle VM Server for SPARC 소프트웨어를 사용하여 단일 시스템에 논리적 도메인이라는 가상 서버를 최대 128개까지 만들 수 있습니다. 이러한 종류의 구성을 통해 SPARC T-Series 서버와 Oracle Solaris OS에서 제공한 대규모 스레드를 활용할 수 있습니다.

게스트 도메인에서 실행되는 Oracle Solaris OS 버전은 primary 도메인에서 실행되는 Oracle Solaris OS 버전과 독립적입니다. 따라서 primary 도메인에서 Oracle Solaris 10 OS를 실행할 경우 게스트 도메인에서 Oracle Solaris 11 OS를 계속 실행할 수 있습니다. 마찬가지로, primary 도메인에서 Oracle Solaris 11 OS를 실행할 경우 게스트 도메인에서 Oracle Solaris 10 OS를 계속 실행할 수 있습니다.

primary 도메인에서 Oracle Solaris 10 OS 또는 Oracle Solaris 11 OS 실행의 유일한 차이점은 각 OS의 기능 차이입니다.

이 장에서는 다음 주제를 다룹니다.

- 22 페이지 “하이퍼바이저 및 Logical Domains”
- 24 페이지 “Logical Domains Manager”
- 27 페이지 “Oracle VM Server for SPARC Physical-to-Virtual 변환 도구”
- 27 페이지 “Oracle VM Server for SPARC Configuration Assistant”
- 28 페이지 “Oracle VM Server for SPARC Management Information Base”

하이퍼바이저 및 Logical Domains

이 절에서는 논리적 도메인을 지원하는 SPARC 하이퍼바이저의 개요를 제공합니다.

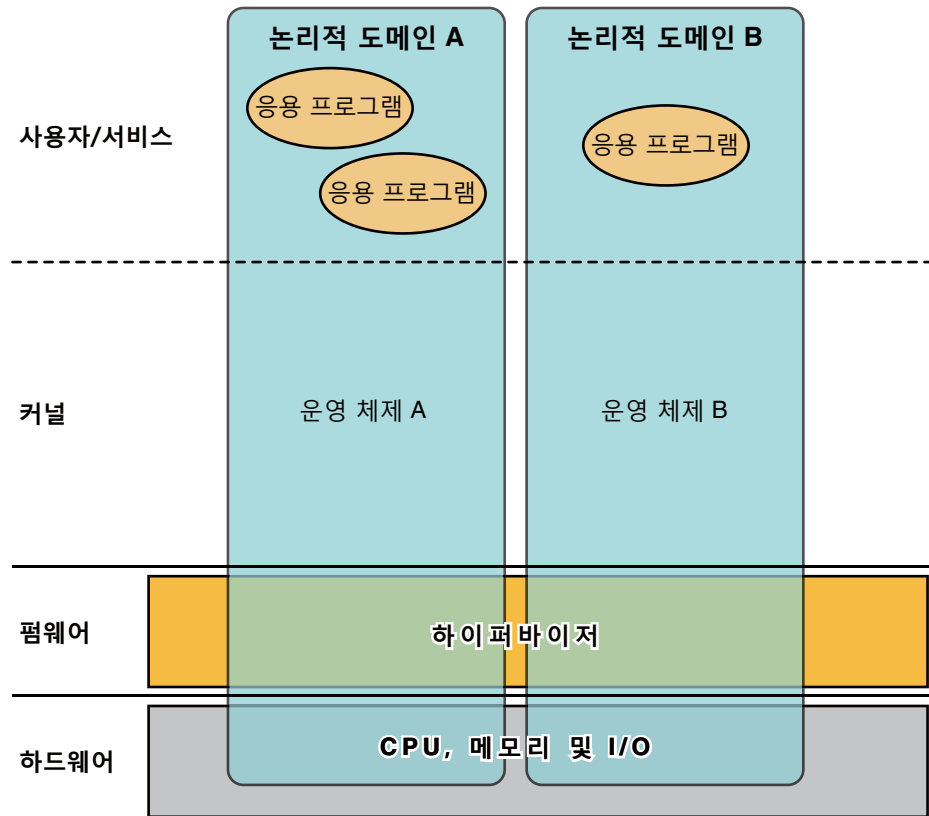
SPARC **하이퍼바이저**는 운영 체제가 작성될 수 있는 안정적인 가상화 컴퓨터 구조를 제공하는 작은 펌웨어 계층입니다. Oracle's Sun 서버는 하드웨어 기능을 통해 하이퍼바이저가 논리적 운영 체제의 작업을 통제할 수 있도록 지원합니다.

논리적 도메인은 별도의 논리적 리소스 그룹으로 구성된 가상 컴퓨터입니다. 논리적 도메인은 단일 컴퓨터 시스템 안에 자체 운영 체제와 신원을 가지고 있습니다. 각 논리적 도메인은 서버의 파워 사이클 없이 독립적으로 생성, 삭제, 재구성, 재부트할 수 있습니다. 여러 논리적 도메인에서 다양한 응용 프로그램 소프트웨어를 실행하고 성능 및 보안 목적에 맞게 독립적으로 유지할 수 있습니다.

각 논리적 도메인은 하이퍼바이저가 사용 가능하도록 설정한 서버 리소스만 관찰하고 상호 작용할 수 있습니다. **Logical Domains Manager**는 하이퍼바이저가 컨트롤 도메인을 통해 수행할 작업을 지정할 수 있습니다. 따라서 하이퍼바이저는 서버 리소스를 분할해서 여러 운영 체제 환경에 제한된 일부를 제공합니다. 이러한 분할 및 프로비전이 논리적 도메인을 만들기 위한 기본 방식입니다. 다음 다이어그램은 두 논리적 도메인을 지원하는 하이퍼바이저를 보여줍니다. 또한 Logical Domains 기능을 구성하는 다음 계층을 보여줍니다.

- 사용자/서비스 또는 응용 프로그램
- 커널 또는 운영 체제
- 펌웨어 또는 하이퍼바이저
- 하드웨어, CPU, 메모리, I/O 포함

그림 1-1 두 도메인을 지원하는 하이퍼바이저



특정 SPARC 하이퍼바이저가 지원하는 각 논리적 도메인의 개수와 기능은 서버에 종속적입니다. 하이퍼바이저는 서버의 전체 CPU, 메모리, I/O 리소스 중 일부를 주어진 논리적 도메인에 할당할 수 있습니다. 그러면 각각의 자체 논리적 도메인 내에서 여러 운영 체제를 동시에 지원할 수 있습니다. 임의의 세분성을 사용하여 별도의 논리적 도메인 간에 리소스를 재배열할 수 있습니다. 예를 들어, CPU 스레드의 세분성으로 논리적 도메인에 CPU를 지정할 수 있습니다.

각 논리적 도메인은 다음과 같은 자체 리소스를 사용하여 완전히 독립적인 컴퓨터로 관리할 수 있습니다.

- 커널, 패치 및 조정 매개변수
- 사용자 계정 및 관리자
- 디스크
- 네트워크 인터페이스, MAC 주소 및 IP 주소

각 논리적 도메인은 서버의 파워 사이클 없이 서로 독립적으로 중지, 시작, 재부트할 수 있습니다.

하이퍼바이저 소프트웨어는 논리적 도메인 사이에 구분을 유지할 책임이 있습니다. 또한 하이퍼바이저 소프트웨어는 논리적 도메인이 서로 통신할 수 있는 LDC(논리적 도메인 채널)를 제공합니다. LDC를 통해 도메인은 네트워크 또는 디스크 서비스를 서로 제공할 수 있습니다.

서비스 프로세서(SP)는 시스템 컨트롤러(SC)라고도 하며, 물리적 컴퓨터를 모니터링하고 실행하지만 논리적 도메인은 관리하지 않습니다. Logical Domains Manager가 논리적 도메인을 관리합니다.

Logical Domains Manager

Logical Domains Manager를 사용하여 논리적 도메인을 만들고 관리할 뿐만 아니라 논리적 도메인을 물리적 리소스에 매핑할 수 있습니다. 하나의 Logical Domains Manager만 서버에 실행할 수 있습니다.

도메인의 역할

모든 논리적 도메인은 동일하며, 사용자가 지정한 역할을 기반으로 서로 구별할 수 있습니다. 다음은 논리적 도메인이 수행할 수 있는 역할입니다.

- **컨트롤 도메인.** Logical Domains Manager는 이 도메인에서 실행되며, 다른 논리적 도메인을 만들고 관리하며 다른 도메인에 가상 리소스를 할당할 수 있습니다. 서버당 하나의 컨트롤 도메인만 가질 수 있습니다. 컨트롤 도메인은 Oracle VM Server for SPARC 소프트웨어를 설치할 때 처음 만든 도메인입니다. 컨트롤 도메인을 primary라고 합니다.
- **서비스 도메인.** 서비스 도메인은 가상 스위치, 가상 콘솔 집중기, 가상 디스크 서버와 같은 가상 장치 서비스를 다른 도메인에 제공합니다. 임의 도메인을 서비스 도메인으로 구성할 수 있습니다.
- **I/O 도메인.** I/O 도메인은 PCI EXPRESS(PCIe) 컨트롤러를 통해 네트워크 카드와 같은 물리적 I/O 장치에 직접 액세스를 제공합니다. I/O 도메인은 PCIe 루트 컴플렉스를 소유하거나, 직접 I/O(DIO) 기능을 사용하여 PCIe 슬롯 또는 온보드 PCIe 장치를 소유할 수 있습니다. 79 페이지 “PCIe 끝점 장치 지정”을 참조하십시오.
I/O 도메인은 서비스 도메인으로 사용될 경우 물리적 I/O 장치를 가상 장치 형태로 다른 도메인과 공유할 수 있습니다.
- **루트 도메인.** 루트 도메인에 PCIe 루트 컴플렉스가 지정됩니다. 이 도메인은 PCIe 패브릭을 소유하며 패브릭 오류 처리와 같은 모든 패브릭 관련 서비스를 제공합니다. 루트 도메인은 물리적 I/O 장치를 소유하고 직접 액세스를 제공하므로 I/O 도메인이기도 합니다.

플랫폼 구조에 따라 지정할 수 있는 루트 도메인 수가 다릅니다. 예를 들어, Sun SPARC Enterprise T5440 서버를 사용하는 경우 최대 4개의 루트 도메인을 지정할 수 있습니다.

- **게스트 도메인.** 게스트 도메인은 하나 이상의 서비스 도메인에서 제공한 가상 장치 서비스를 이용하는 비I/O 도메인입니다. 게스트 도메인에는 물리적 I/O 장치는 없고 가상 디스크, 가상 네트워크 인터페이스와 같은 가상 I/O 장치만 있습니다.

아직 Logical Domains로 구성되지 않은 기존 시스템에 Logical Domains Manager를 설치할 수 있습니다. 이 경우 현재 OS 인스턴스가 컨트롤 도메인이 됩니다. 또한 단 하나의 도메인, 즉 컨트롤 도메인으로 시스템이 구성됩니다. 컨트롤 도메인을 구성한 후에 전체 시스템을 가장 효율적으로 사용할 수 있도록 다른 도메인에서 응용 프로그램의 로드 균형을 조정할 수 있습니다. 도메인을 추가하고 해당 응용 프로그램을 컨트롤 도메인에서 새 도메인으로 이동하면 됩니다.

명령줄 인터페이스

Logical Domains Manager는 명령줄 인터페이스(CLI)를 사용하여 논리적 도메인을 만들고 구성합니다. CLI는 단일 명령 `ldm`에 여러 하위 명령이 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

Logical Domains Manager CLI를 사용하려면 Logical Domains Manager 데몬 `ldmd`가 실행 중이어야 합니다.

가상 입/출력

Logical Domains 환경에서 SPARC T-Series 시스템에 최대 128개의 도메인을 프로비전할 수 있습니다. 일부 SPARC T-Series 서버, 특히 단일 프로세서와 일부 듀얼 프로세서 시스템에는 제한된 수의 I/O 버스와 물리적 I/O 슬롯이 허용됩니다. 그 결과, 이러한 시스템의 모든 도메인에서 물리적 디스크 및 네트워크 장치에 배타적 액세스를 제공하지 못할 수 있습니다. 물리적 장치에 액세스를 제공하려면 PCIe 버스 또는 끝점 장치를 도메인에 지정할 수 있습니다. 이 해결법은 모든 도메인에 배타적 장치 액세스를 제공하기에는 부족합니다. [6 장, “I/O 도메인 설정”](#)을 참조하십시오. 직접 액세스할 수 있는 물리적 I/O 장치 수에 대한 이와 같은 제한은 가상화된 I/O 모델을 구현하여 해결할 수 있습니다.

물리적 I/O 액세스가 없는 논리적 도메인은 서비스 도메인과 통신하는 가상 I/O 장치로 구성됩니다. 서비스 도메인은 가상 장치 서비스를 실행하여 물리적 장치 또는 그 기능에 액세스할 수 있습니다. 이 클라이언트-서버 모델에서 가상 I/O 장치는 LDC(논리적 도메인 채널)라는 도메인간 통신 채널을 통해 서비스 상대방과 또는 서로 통신합니다. 가상화된 I/O 기능은 가상 네트워킹, 저장소, 콘솔에 지원됩니다.

가상 네트워크

Logical Domains는 가상 네트워크 장치 및 가상 네트워크 스위치 장치를 사용하여 가상 네트워킹을 구현합니다. 가상 네트워크(vnet) 장치는 이더넷 장치를 에뮬레이트하고 지점간 채널을 사용하여 시스템의 다른 vnet 장치와 통신합니다. 가상 스위치(vsw)

장치는 주로 모든 가상 네트워크의 송/수신 패킷의 멀티플렉서로 작동합니다. **vsw** 장치는 서비스 도메인에서 물리적 네트워크 어댑터와 직접 상호 작용하고 가상 네트워크 대신 패킷을 송/수신합니다. 또한 **vsw** 장치는 단순 계층-2 스위치로 작동하고 시스템에서 연결된 **vnet** 장치 간에 패킷을 전환합니다.

가상 저장소

가상 저장소 기반구조는 클라이언트-서버 모델을 사용하여 논리적 도메인에 직접 지정되지 않은 블록 레벨 저장소에 액세스할 수 있습니다. 이 모델은 다음 구성 요소를 사용합니다.

- 가상 디스크 클라이언트(**vdc**) - 블록 장치 인터페이스를 내보냅니다.
- 가상 디스크 서비스(**vds**) - 가상 디스크 클라이언트 대신 디스크 요청을 처리하고 서비스 도메인에 상주하는 백엔드 저장소에 제출합니다.

가상 디스크는 클라이언트 도메인에 일반 디스크로 나타나지만, 대부분의 디스크 작업이 가상 디스크 서비스로 전달되고 서비스 도메인에서 처리됩니다.

가상 콘솔

Logical Domains 환경에서 **primary** 도메인의 콘솔 I/O는 서비스 프로세서로 지정됩니다. 그 밖의 모든 도메인의 콘솔 I/O는 가상 콘솔 집중기(**vcc**)를 실행 중인 서비스 도메인으로 재지정됩니다. **vcc**를 실행하는 도메인은 일반적으로 **primary** 도메인입니다. 가상 콘솔 집중기 서비스는 모든 도메인의 콘솔 트래픽에 대한 집중기로 작동하고 가상 네트워크 터미널 서버 데몬(**vntsd**)과 상호 작용하여 UNIX 소켓을 통해 각 콘솔에 액세스할 수 있습니다.

리소스 구성

Oracle VM Server for SPARC 소프트웨어를 실행하는 시스템은 가상 CPU, 가상 I/O 장치, 암호화 장치, 메모리와 같은 리소스를 구성할 수 있습니다. 일부 리소스는 실행 중인 도메인에 동적으로 구성할 수 있는 반면, 나머지는 중지된 도메인에 구성해야 합니다. 컨트롤 도메인에서 리소스를 동적으로 구성할 수 없는 경우 먼저 지연된 재구성을 시작해야 합니다. 컨트롤 도메인이 재부트될 때까지 지연된 재구성이 구성 작업을 연기합니다. 자세한 내용은 [191 페이지 “리소스 재구성”](#)을 참조하십시오.

지속적 구성

ldm 명령을 사용하여 논리적 도메인의 현재 구성을 서비스 프로세서에 저장할 수 있습니다. 구성을 추가하고, 사용할 구성을 지정하고, 구성을 제거하고, 구성을 나열할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 또한 SP에서 부트할 구성을 지정할 수 있습니다. [242 페이지 “서비스 프로세서와 함께 Logical Domains 사용”](#)을 참조하십시오.

구성 관리에 대한 내용은 234 페이지 “[Logical Domains 구성 관리](#)”를 참조하십시오.

Oracle VM Server for SPARC Physical-to-Virtual 변환 도구

Oracle VM Server for SPARC P2V(Physical-to-Virtual) 변환 도구는 기존의 물리적 시스템을 CMT(칩 멀티스레딩) 시스템의 논리적 도메인에서 Oracle Solaris 10 OS를 실행하는 가상 시스템으로 자동으로 변환합니다. Oracle Solaris 10 OS 또는 Oracle Solaris 11 OS를 실행하는 컨트롤 도메인에서 `ldmp2v` 명령을 실행하여 다음 소스 시스템 중 하나를 논리적 도메인으로 변환할 수 있습니다.

- Solaris 8, Solaris 9, Oracle Solaris 10 OS를 실행하는 sun4u SPARC 기반 시스템
- Oracle Solaris 10 OS를 실행하지만 Oracle VM Server for SPARC 소프트웨어를 실행하지 않는 sun4v 시스템

주 - Oracle Solaris 11 물리적 시스템을 가상 시스템으로 변환하기 위해 P2V 도구를 사용할 수 없습니다.

도구 정보와 설치 방법은 13 장, “[Oracle VM Server for SPARC Physical-to-Virtual 변환 도구](#)”를 참조하십시오. `ldmp2v` 명령에 대한 내용은 `ldmp2v(1M)` 매뉴얼 페이지를 참조하십시오.

Oracle VM Server for SPARC Configuration Assistant

Oracle VM Server for SPARC Configuration Assistant는 기본 등록 정보를 설정하여 논리적 도메인의 구성 과정을 안내합니다. 이를 사용하여 Oracle VM Server for SPARC 소프트웨어가 설치되었지만 아직 구성되지 않은 시스템을 구성할 수 있습니다.

구성 데이터를 수집한 후 Configuration Assistant는 논리적 도메인으로 부트하기에 적합한 구성을 만듭니다. 또한 Configuration Assistant에서 선택한 기본값을 사용하여 사용 가능한 시스템 구성을 만들 수 있습니다.

주 - `ldmconfig` 명령은 Oracle Solaris 10 시스템에만 지원됩니다.

Configuration Assistant는 터미널 기반 도구입니다.

자세한 내용은 14 장, “[Oracle VM Server for SPARC Configuration Assistant\(Oracle Solaris 10\)](#)” 및 `ldmconfig(1M)` 매뉴얼 페이지를 참조하십시오.

Oracle VM Server for SPARC Management Information Base

Oracle VM Server for SPARC MIB(Management Information Base)를 사용하면 타사 시스템 관리 응용 프로그램이 SNMP(Simple Network Management Protocol)를 사용하여 도메인을 원격 모니터링하고 논리적 도메인을 시작/중지할 수 있습니다. 자세한 내용은 15 장, “Oracle VM Server for SPARC Management Information Base 소프트웨어 사용”을 참조하십시오.

주 - Oracle VM Server for SPARC MIB 소프트웨어는 Oracle Solaris 10 시스템에만 사용할 수 있습니다.

소프트웨어 설치 및 사용

이 장에서는 Oracle VM Server for SPARC 2.2 소프트웨어를 사용하는 데 필요한 여러 소프트웨어 구성 요소를 설치하거나 업그레이드하는 방법을 설명합니다. Oracle VM Server for SPARC 소프트웨어를 사용하려면 다음 구성 요소가 필요합니다.

- 지원되는 플랫폼. 지원되는 플랫폼 목록은 **Oracle VM Server for SPARC 2.2 릴리스 노트**의 “지원되는 플랫폼”를 참조하십시오.
- 적절한 운영 체제를 실행 중인 컨트롤 도메인. **Oracle VM Server for SPARC 2.2 릴리스 노트**의 “필수 소프트웨어 및 패치”에 권장된 대로 최소한 Oracle Solaris 11 OS와 적절한 SRU(Support Repository Update) 또는 Oracle Solaris 10 8/11 OS와 패치가 설치되어야 합니다. 35 페이지 “Oracle Solaris OS 업그레이드”를 참조하십시오.
- 최소한 시스템 펌웨어 버전 7.4.2(Sun UltraSPARC T2 또는 T2 Plus 플랫폼) 또는 버전 8.2.0(SPARC T3 플랫폼 및 SPARC T4 플랫폼). 31 페이지 “시스템 펌웨어 업그레이드”를 참조하십시오.
- Oracle VM Server for SPARC 2.2 소프트웨어가 컨트롤 도메인에 설치되고 사용으로 설정됨. 32 페이지 “Logical Domains Manager 설치”를 참조하십시오.
- (선택 사항) Oracle VM Server for SPARC MIB(Management Information Base) 소프트웨어 패키지. Oracle Solaris 10 OS에서만 사용 가능하며 15 장, “Oracle VM Server for SPARC Management Information Base 소프트웨어 사용”을 참조하십시오.

Logical Domains Manager를 설치/업그레이드하기 전에 Oracle Solaris OS와 시스템 펌웨어를 서버에 설치하거나 업그레이드해야 합니다. 시스템에서 Oracle VM Server for SPARC 소프트웨어를 이미 사용 중인 경우 35 페이지 “Oracle VM Server for SPARC를 이미 사용 중인 시스템 업그레이드”를 참조하십시오. 그렇지 않으면 30 페이지 “새 시스템에 Oracle VM Server for SPARC 소프트웨어 설치”를 참조하십시오.

이 장에서는 다음 주제를 다룹니다.

- 30 페이지 “새 시스템에 Oracle VM Server for SPARC 소프트웨어 설치”
- 35 페이지 “Oracle VM Server for SPARC를 이미 사용 중인 시스템 업그레이드”
- 40 페이지 “출하 시 기본 구성과 Logical Domains 사용 안함”

새 시스템에 Oracle VM Server for SPARC 소프트웨어 설치

Oracle VM Server for SPARC 소프트웨어를 지원하는 Oracle's Sun 플랫폼에 Oracle Solaris 10 OS 또는 Oracle Solaris 11 OS가 사전 설치됩니다. 처음에 플랫폼은 하나의 운영 체제만 호스트하는 단일 시스템으로 나타납니다. Oracle Solaris OS, 시스템 펌웨어, Logical Domains Manager가 설치된 후에 원래 시스템과 Oracle Solaris OS 인스턴스가 컨트로드메인이 됩니다. 이와 같은 플랫폼의 첫번째 도메인을 **primary**라고 하며, 해당 이름을 변경하거나 도메인을 삭제할 수 없습니다. 여기서부터 플랫폼을 재구성하여 다양한 Oracle Solaris OS 인스턴스를 호스트하는 여러 도메인을 지정할 수 있습니다.

주 - 게스트 도메인에서 실행되는 Oracle Solaris OS 소프트웨어 버전은 **primary** 도메인에서 실행되는 Oracle Solaris OS 버전과 **독립적**입니다. 따라서 **primary** 도메인에서 Oracle Solaris 10 OS를 실행할 경우 게스트 도메인에서 Oracle Solaris 11 OS를 계속 실행할 수 있습니다. 마찬가지로, **primary** 도메인에서 Oracle Solaris 11 OS를 실행할 경우 게스트 도메인에서 Oracle Solaris 10 OS를 계속 실행할 수 있습니다.

사용자 요구 사항과 Oracle Solaris 10 및 Oracle Solaris 11의 잠재적 기능 차이에 따라 **primary** 도메인에서 실행할 Oracle Solaris OS 버전을 결정하십시오. **Oracle Solaris 11 Release Notes** 및 **Transitioning From Oracle Solaris 10 to Oracle Solaris 11** 을 참조하십시오.

Oracle Solaris OS 업데이트

새 시스템에서 사용자 설치 정책에 맞게 출하시 설치된 OS를 재설치할 수 있습니다. **Oracle VM Server for SPARC 2.2 릴리스 노트**의 “필수 및 권장 Oracle Solaris OS”를 참조하십시오. 완전한 Oracle Solaris OS 설치 지침은 **Oracle Solaris 10 8/11 Information Library** (http://docs.oracle.com/cd/E23823_01/) 및 **Oracle Solaris 11 Information Library** (http://docs.oracle.com/cd/E23824_01/)를 참조하십시오. 시스템 요구 사항에 맞게 설치를 사용자 정의할 수 있습니다.

시스템에 Oracle Solaris OS가 이미 설치된 경우 Oracle VM Server for SPARC 2.2 소프트웨어와 연관된 OS 버전으로 업그레이드해야 합니다. **Oracle VM Server for SPARC 2.2 릴리스 노트**의 “필수 소프트웨어 및 패치”를 참조하십시오. 완전한 Oracle Solaris OS 업그레이드 지침은 **Oracle Solaris 10 8/11 Information Library** (http://docs.oracle.com/cd/E23823_01/) 및 **Oracle Solaris 11 Information Library** (http://docs.oracle.com/cd/E23824_01/)를 참조하십시오.

시스템 펌웨어 업그레이드

다음 작업은 ILOM(Integrated Lights Out Manager) 소프트웨어를 사용하여 시스템 펌웨어를 업그레이드하는 방법을 설명합니다. 다음 리소스를 참조하십시오.

- ILOM 소프트웨어를 사용하여 시스템 펌웨어를 업그레이드하는 방법은 <http://www.oracle.com/technetwork/documentation/sparc-tseries-servers-252697.html>의 “Update the Firmware” 및 **Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide**의 “Updating ILOM Firmware”를 참조하십시오.
- 플랫폼용 시스템 펌웨어는 <http://www.oracle.com/technetwork/systems/patches/firmware/index.html>에서 찾을 수 있습니다.
- 지원되는 서버의 필수 시스템 펌웨어 정보는 **Oracle VM Server for SPARC 2.2 릴리스 노트**의 “필수 및 권장 시스템 펌웨어 패치”를 참조하십시오.
- 컨트롤 도메인에서 시스템 펌웨어를 업그레이드하려면 해당 시스템 펌웨어 릴리스 노트를 참조하십시오.
- 지원되는 서버의 시스템 펌웨어를 설치하고 업그레이드하는 방법은 해당 서버의 관리 설명서나 제품 정보를 참조하십시오.
- ILOM 웹 인터페이스를 사용하여 시스템 펌웨어를 업그레이드할 수도 있습니다. **Sun Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide**의 “Updating ILOM Firmware”를 참조하십시오.

Logical Domains Manager 다운로드

Oracle Solaris 10 OS 및 Oracle Solaris 11 OS의 최신 패키지를 얻을 수 있습니다. Oracle VM Server for SPARC 소프트웨어는 Oracle Solaris 11 OS에 기본적으로 포함되어 있습니다.

- **Oracle Solaris 10 OS.** My Oracle Support에서 `OVM_Server_SPARC-2_2.zip` 패키지를 다운로드합니다. 31 페이지 “[Logical Domains Manager 소프트웨어를 다운로드하는 방법\(Oracle Solaris 10\)](#)”을 참조하십시오.
- **Oracle Solaris 11 OS.** Oracle Solaris 11 Support Repository에서 `ldomsmanager` 패키지를 얻습니다. 38 페이지 “[Oracle VM Server for SPARC 2.2 소프트웨어로 업그레이드하는 방법\(Oracle Solaris 11\)](#)”을 참조하십시오.

▼ Logical Domains Manager 소프트웨어를 다운로드하는 방법(Oracle Solaris 10)

- 1 zip 파일(`OVM_Server_SPARC-2_2.zip`)을 다운로드합니다.
<http://www.oracle.com/virtualization/index.html>에서 소프트웨어를 찾을 수 있습니다.

2 zip 파일의 압축을 풉니다.

```
$ unzip OVM_Server_SPARC-2_2.zip
```

파일의 구조와 내용물에 대한 세부 정보는 [Oracle VM Server for SPARC 2.2 릴리스 노트](#)의 “Oracle VM Server for SPARC 2.2 소프트웨어 위치”를 참조하십시오.

Logical Domains Manager 설치

다음은 Logical Domains Manager 소프트웨어의 설치 방법입니다.

- **Oracle Solaris 10만 해당.** 설치 스크립트를 사용하여 패키지 및 패치를 설치합니다. 이 방법은 Logical Domains Manager 소프트웨어를 자동으로 설치합니다. [32 페이지](#) “(Oracle Solaris 10) 자동으로 Logical Domains Manager 소프트웨어 설치”를 참조하십시오.
- **Oracle Solaris 10만 해당.** Oracle Solaris JumpStart 기능을 사용하여 네트워크 설치의 일부로 패키지를 설치합니다. JumpStart 서버 구성에 대한 내용은 [Oracle Solaris 10 8/11 설치 설명서: 사용자 정의 JumpStart 및 고급 설치](#)를 참조하십시오. 이 기능의 사용법에 대한 완전한 정보는 [JumpStart Technology: Effective Use in the Solaris Operating Environment](#)를 참조하십시오.
- **Oracle Solaris 11만 해당.** Oracle Solaris 11 자동 설치 프로그램 기능을 사용하여 네트워크 설치의 일부로 패키지를 설치합니다. [Oracle Solaris 11 시스템의 “Automated Installer를 어떻게 사용합니까?”](#) 및 [Oracle Solaris 10 JumpStart에서 Oracle Solaris 11 자동 설치 프로그램으로 전환](#)를 참조하십시오.
- 수동으로 패키지 설치. [33 페이지](#) “수동으로 Logical Domains Manager 소프트웨어 설치”를 참조하십시오.

주 - Oracle VM Server for SPARC 패키지를 설치한 후에 Oracle VM Server for SPARC MIB 소프트웨어 패키지를 수동으로 설치해야 합니다. 다른 패키지와 함께 자동으로 설치되지 않습니다. Oracle VM Server for SPARC MIB 설치 및 사용에 대한 자세한 내용은 [15 장, “Oracle VM Server for SPARC Management Information Base 소프트웨어 사용”](#)을 참조하십시오.

(Oracle Solaris 10) 자동으로 Logical Domains Manager 소프트웨어 설치

install-ldm 설치 스크립트를 사용하는 경우 스크립트 실행 방법을 지정할 수 있는 여러 선택이 있습니다. 각 선택은 아래의 절차에 설명됩니다.

- install-ldm 스크립트를 옵션 없이 사용하면 다음을 자동으로 수행합니다.
 - Oracle Solaris OS 릴리스가 Oracle Solaris 10 OS인지 검사합니다.
 - 패키지 하위 디렉토리 SUNWldm/ 및 SUNWldmp2v/가 존재하는지 확인합니다.
 - 필수 조건 Logical Domains 드라이버 패키지 SUNWldomr 및 SUNWldomu가 존재하는지 확인합니다.

- SUNWldm 및 SUNWldmp2v 패키지가 설치되지 않았는지 확인합니다.
- Oracle VM Server for SPARC 2.2 소프트웨어를 설치합니다.
- 모든 패키지가 설치되었는지 확인합니다.
- SST(SUNWjass)가 이미 설치된 경우 컨트롤 도메인에 Oracle Solaris OS를 강화하라는 메시지가 나타납니다.
- 설치를 수행하기 위해 Oracle VM Server for SPARC Configuration Assistant(ldmconfig)를 사용할지 여부를 결정합니다.
- install-ldm 스크립트를 -c 옵션과 함께 사용하면 소프트웨어가 설치된 후 Oracle VM Server for SPARC Configuration Assistant를 자동으로 실행합니다.
- install-ldm 스크립트를 -s 옵션과 함께 사용하면 Oracle VM Server for SPARC Configuration Assistant 실행을 건너뛸니다.
- install-ldm 스크립트와 아래 옵션을 SST 소프트웨어와 함께 사용하면 다음을 수행할 수 있습니다.
 - install-ldm -d. -secure.driver로 끝나는 드라이버 이외의 SST 드라이버를 지정할 수 있습니다. 이 옵션은 앞의 선택에 나열된 모든 기능을 자동으로 수행하고 사용자가 지정한 SST 드라이버(예: server-secure-myname.driver)를 사용하여 컨트롤 도메인에 Oracle Solaris OS를 강화합니다.
 - install-ldm -d none. SST를 사용하여 컨트롤 도메인에 실행 중인 Oracle Solaris OS를 강화하지 **않도록** 지정합니다. 이 옵션은 앞의 선택에 나열된 기능 중 강화를 제외한 모든 기능을 자동으로 수행합니다. SST 사용을 건너뛰는 것은 권장되지 않으며, 대체 프로세스를 사용하여 컨트롤 도메인을 강화하려는 경우에만 수행해야 합니다.
 - install-ldm -p. 사후 설치 작업으로 Logical Domains Manager 데몬(ldmd) 사용과 SST 실행만 수행하도록 지정합니다. 예를 들어, SUNWldm 및 SUNWjass 패키지가 서버에 사전 설치된 경우 이 옵션을 사용합니다.

수동으로 Logical Domains Manager 소프트웨어 설치

다음 절차는 Oracle Solaris 10 OS에 Oracle VM Server for SPARC 2.2 소프트웨어를 수동으로 설치하는 과정을 안내합니다.

Oracle Solaris 11 OS를 설치할 때 Oracle VM Server for SPARC 2.1 소프트웨어는 기본적으로 설치됩니다. Oracle VM Server for SPARC 2.2 소프트웨어를 설치하려면 38 페이지 “Oracle VM Server for SPARC 2.2 소프트웨어로 업그레이드하는 방법(Oracle Solaris 11)”을 참조하십시오.

▼ Oracle VM Server for SPARC 2.2 소프트웨어를 수동으로 설치하는 방법(Oracle Solaris 10)

시작하기 전에 Oracle VM Server for SPARC 2.2 소프트웨어(SUNWldm 및 SUNWldmp2v 패키지)를 다운로드합니다. 지침은 31 페이지 “Logical Domains Manager 소프트웨어를 다운로드하는 방법(Oracle Solaris 10)”을 참조하십시오.

1 (선택 사항) 필요한 경우 서비스 프로세서(SP)에 구성을 저장합니다.

이전 버전의 Oracle VM Server for SPARC 소프트웨어를 이미 실행 중인 경우에만 이 단계를 수행합니다.

```
primary# ldm add-config config-name
```

2 SUNWldm.v 및 SUNWldmp2v 패키지를 설치합니다.

```
# pkgadd -Gd . SUNWldm.v SUNWldmp2v
```

대화식 프롬프트의 모든 질문에 y(yes)로 대답합니다.

-G 옵션은 전역 영역에만 패키지를 설치합니다. -d 옵션은 SUNWldm.v 및 SUNWldmp2v 패키지를 포함하는 디렉토리의 경로를 지정합니다.

pkgadd 명령에 대한 자세한 내용은 pkgadd(1M) 매뉴얼 페이지를 참조하십시오.

3 SUNWldm 및 SUNWldmp2v 패키지가 설치되었는지 확인합니다.

다음은 개정(REV) 정보의 예입니다.

```
# pkginfo -l SUNWldm | grep VERSION
VERSION=2.2,REV=2012.02.01.10.20
```

pkginfo 명령에 대한 자세한 내용은 pkginfo(1) 매뉴얼 페이지를 참조하십시오.

Logical Domains Manager 데몬 사용

install-ldm 설치 스크립트는 Logical Domains Manager 데몬(ldmd)을 자동으로 사용으로 설정합니다. 또한 Oracle VM Server for SPARC 소프트웨어 패키지를 설치할 때 ldmd 데몬이 자동으로 사용으로 설정됩니다. 사용으로 설정된 경우 논리적 도메인을 생성, 수정, 제어할 수 있습니다.

▼ Logical Domains Manager 데몬을 사용으로 설정하는 방법

이 절차에 따라 ldmd 데몬이 사용 안함으로 설정된 경우 사용으로 설정합니다.

1 svcadm 명령을 사용하여 Logical Domains Manager 데몬 ldmd를 사용으로 설정합니다.

svcadm 명령에 대한 자세한 내용은 svcadm(1M) 매뉴얼 페이지를 참조하십시오.

```
# svcadm enable ldmd
```

2 ldm list 명령을 사용하여 Logical Domains Manager가 실행 중인지 확인합니다.

ldm list 명령은 현재 시스템에 정의된 모든 도메인을 나열해야 합니다. 특히 primary 도메인이 나열되고 active 상태여야 합니다. 다음 샘플 출력은 primary 도메인만 시스템에 정의되어 있음을 보여줍니다.

```
# /opt/SUNWldm/bin/ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active ---c-  SP    64     3264M   0.3%  19d 9m
```

Oracle VM Server for SPARC를 이미 사용 중인 시스템 업그레이드

이 절에서는 Oracle VM Server for SPARC 소프트웨어를 이미 사용 중인 시스템에서 Oracle Solaris OS, 펌웨어, Logical Domains Manager 구성 요소를 업그레이드하는 프로세스를 설명합니다.

시스템이 Oracle VM Server for SPARC 소프트웨어로 이미 구성된 경우 컨트롤 도메인과 기존 도메인을 업그레이드합니다. 이와 같이 업그레이드하면 해당 도메인에서 모든 Oracle VM Server for SPARC 2.2 기능을 사용할 수 있습니다.

주 - Oracle VM Server for SPARC 소프트웨어를 업그레이드하기 전에 다음 단계를 수행합니다.

- 필요한 시스템 펌웨어로 시스템을 업그레이드합니다.
Oracle VM Server for SPARC 2.2 릴리스 노트의 “Oracle VM Server for SPARC 2.2 기능을 사용으로 설정하기 위해 필요한 소프트웨어”를 참조하십시오.
 - 필요한 Oracle Solaris 10 OS 패치 또는 Oracle Solaris 11 OS SRU를 적용합니다.
Oracle VM Server for SPARC 2.2 릴리스 노트의 “필수 및 권장 Oracle Solaris OS”를 참조하십시오.
 - SP에 구성을 저장합니다.
-

Oracle Solaris OS 업그레이드

이 버전의 Oracle VM Server for SPARC 소프트웨어에 사용할 Oracle Solaris 10 또는 Oracle Solaris 11 OS와 다양한 도메인의 필수/권장 패치를 찾으려면 **Oracle VM Server for SPARC 2.2 릴리스 노트의 “필수 소프트웨어 및 패치”**를 참조하십시오. Oracle Solaris OS 업그레이드에 대한 완전한 지침은 Oracle Solaris 10 및 Oracle Solaris 11 설치 설명서를 참조하십시오.

컨트롤 도메인에 Oracle Solaris OS를 재설치할 때, 이 절에 설명된 대로 Logical Domains 자동 저장 구성 데이터 및 제약 조건 데이터베이스 파일을 저장하고 복원해야 합니다.

자동 저장 구성 디렉토리 저장 및 복원

컨트롤 도메인에 운영 체제를 재설치하기 전에 자동 저장 구성 디렉토리를 저장하고 복원할 수 있습니다. 컨트롤 도메인에 운영 체제를 재설치할 때마다

`/var/opt/SUNWldm/autosave-autosave-name` 디렉토리에 있는 Logical Domains 자동 저장 구성 데이터를 저장하고 복원해야 합니다.

`tar` 또는 `cpio` 명령을 사용하여 전체 디렉토리 내용을 저장하고 복원할 수 있습니다.

주- 각 자동 저장 디렉토리에는 관련 구성의 마지막 SP 구성 업데이트에 대한 시간 기록이 있습니다. 자동 저장 파일을 복원할 때 시간 기록이 동기화되지 않을 수 있습니다. 이 경우 복원된 자동 저장 구성이 이전 상태, 즉 [newer] 또는 up to date로 표시됩니다.

자동 저장 구성에 대한 자세한 내용은 234 페이지 “Logical Domains 구성 관리”를 참조하십시오.

▼ 자동 저장 디렉토리를 저장하고 복원하는 방법

이 절차는 자동 저장 디렉토리를 저장하고 복원하는 방법을 보여줍니다.

1 자동 저장 디렉토리를 저장합니다.

```
# cd /
# tar -cvpf autosave.tar var/opt/SUNWldm/autosave-*
```

2 (선택 사항) 깔끔한 복원 작업을 위해 기존 자동 저장 디렉토리를 제거합니다.

때때로 자동 저장 디렉토리에 잉여 파일이 포함될 수 있습니다. 아마 이전 구성에서 남은 파일로, SP에 다운로드된 구성을 손상시킬 수 있습니다. 이 경우, 이 예제에 표시된 대로 복원 작업 전에 자동 저장 디렉토리를 정리합니다.

```
# cd /
# rm -rf var/opt/SUNWldm/autosave-*
```

3 자동 저장 디렉토리를 복원합니다.

다음 명령은 /var/opt/SUNWldm 디렉토리에 파일 및 디렉토리를 복원합니다.

```
# cd /
# tar -xvpf autosave.tar
```

Logical Domains 제약조건 데이터베이스 파일 저장 및 복원

컨트롤 도메인에 운영 체제를 업그레이드할 때마다 /var/opt/SUNWldm/ldom-db.xml에 있는 Logical Domains 제약 조건 데이터베이스 파일을 저장하고 복원해야 합니다.

주- 또한 컨트롤 도메인의 파일 데이터를 파괴할 수 있는 다른 작업(예: 디스크 교체)을 수행할 때 /var/opt/SUNWldm/ldom-db.xml 파일을 저장하고 복원합니다.

Oracle Solaris 10 Live Upgrade 기능을 사용할 때 Logical Domains 제약 조건 데이터베이스 파일 보존

컨트롤 도메인에서 Oracle Solaris 10 Live Upgrade 기능을 사용하는 경우 다음 행을 /etc/lu/synclist 파일에 추가해 보십시오.

```
/var/opt/SUNWldm/ldom-db.xml OVERWRITE
```

이 행을 추가하면 활성 부트 환경에서 새 부트 환경으로 부트 환경을 전환할 때 데이터베이스가 자동으로 복사됩니다. `/etc/lu/synclist` 파일과 부트 환경 간의 파일 동기화에 대한 자세한 내용은 [Oracle Solaris 10 8/11 설치 설명서: Solaris Live Upgrade 및 업그레이드 계획의 “부트 환경 사이의 파일 동기화”](#)을 참조하십시오.

Logical Domains Manager 및 시스템 펌웨어 업그레이드

이 절에서는 Oracle VM Server for SPARC 2.2 소프트웨어로 업그레이드하는 방법을 보여줍니다.

먼저 Logical Domains Manager를 컨트롤 도메인으로 다운로드합니다. [31 페이지 “Logical Domains Manager 다운로드”](#)를 참조하십시오.

그런 다음 플랫폼에서 실행 중인 모든 도메인(컨트롤 도메인 제외)을 중지합니다.

▼ 컨트롤 도메인을 제외한, 플랫폼에서 실행 중인 모든 도메인을 중지하는 방법

시스템의 파워 사이클을 수행하거나 펌웨어를 업그레이드하려는 경우에만 이 작업을 수행합니다. Logical Domains Manager 소프트웨어만 업그레이드할 경우 이 작업을 수행할 필요가 없습니다.

- 1 각 도메인을 `ok` 프롬프트로 가져옵니다.
- 2 `-a` 옵션을 사용하여 모든 도메인을 중지합니다.
- 3 각 도메인에 대해 컨트롤 도메인에서 `unbind-domain` 하위 명령을 실행합니다.

```
primary# ldm unbind-domain ldom
```

Oracle VM Server for SPARC 2.2 소프트웨어로 업그레이드

이 절에서는 Oracle VM Server for SPARC 2.2 소프트웨어로 업그레이드하는 방법을 설명합니다.

▼ Oracle VM Server for SPARC 2.2 소프트웨어로 업그레이드하는 방법(Oracle Solaris 10)

- 1 시스템 펌웨어의 플래시 업그레이드를 수행합니다.
- 2 Logical Domains Manager 데몬(ldmd)을 사용 안함으로 설정합니다.

```
# svcadm disable ldmd
```
- 3 이전 SUNWldm 패키지를 제거합니다.

```
# pkgrm SUNWldm
```
- 4 새 SUNWldm 패키지를 추가합니다.

```
# pkgadd -Gd . SUNWldm.v
```

-d 옵션은 패키지가 현재 디렉토리에 있음을 나타냅니다.
- 5 `ldm list` 명령을 사용하여 Logical Domains Manager가 실행 중인지 확인합니다.

`ldm list` 명령은 현재 시스템에 정의된 모든 도메인을 나열해야 합니다. 특히 `primary` 도메인이 나열되고 `active` 상태여야 합니다. 다음 샘플 출력은 `primary` 도메인만 시스템에 정의되어 있음을 보여줍니다.

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	---c-	SP	32	3264M	0.3%	19d 9m

▼ Oracle VM Server for SPARC 2.2 소프트웨어로 업그레이드하는 방법(Oracle Solaris 11)

- 1 Logical Domains Manager 업그레이드를 위해 도메인을 준비합니다.

다음 단계를 수행하면 필요한 경우 Oracle VM Server for SPARC 2.1 소프트웨어를 실행하는 부트 환경(BE)으로 "롤백"할 수 있습니다.

 - a. SP에 구성을 저장합니다.

```
# ldm add-config config-name
```

다음 예제는 `ldoms-2.1-config`라는 구성을 저장합니다.

```
# ldm add-config ldoms-2.1-config
```
 - b. 기존 BE의 스냅샷을 만듭니다.

```
# beadm create snapshot-name
```

다음 예제는 `S10811@ldoms-2.1-backup`이라는 스냅샷을 만듭니다.

```
# beadm create S10811@ldoms-2.1-backup
```

c. 스냅샷을 기반으로 백업 BE를 만듭니다.

```
# beadm create -e snapshot-name BE-name
```

다음 예제는 S10811@ldoms-2.1-backup이라는 스냅샷에서 ldoms-2.1-backup이라는 새 BE를 만듭니다.

```
# beadm create -e S10811@ldoms-2.1-backup ldoms-2.1-backup
```

2 온라인 소프트웨어 저장소를 사용하도록 등록합니다.

[Certificate Generator Online Help \(https://pkg-register.oracle.com/help/#support\)](https://pkg-register.oracle.com/help/#support)를 참조하십시오.

3 최신 SRU 업데이트의 일부로 온라인 소프트웨어 저장소에서 Oracle VM Server for SPARC 2.2 버전의 ldomsmanager 패키지를 설치합니다.

```
# pkg update
```

```

Packages to install: 1
Packages to update: 89
Create boot environment: No
Create backup boot environment: No
Services to change: 3
```

PHASE	ACTIONS
Removal Phase	517/517
Install Phase	806/806
Update Phase	5325/5325

PHASE	ITEMS
Package State Update Phase	179/179
Package Cache Update Phase	89/89
Image State Update Phase	2/2

주 - Oracle Solaris 11 SRU에 대한 내용은 [Oracle Solaris 11 Support Repository Updates \(SRU\) Index \(https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1372094.1\)](https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1372094.1)를 참조하십시오.

4 패키지가 설치되었는지 확인합니다.

```
# pkg info ldomsmanager
```

```

Name: system/ldoms/ldomsmanager
Summary: Logical Domains Manager
Description: LDoms Manager - Virtualization for SPARC T-Series
Category: System/Virtualization
State: Installed
Publisher: solaris
Version: 2.2.0.0
Build Release: 5.11
Branch: 0.175.0.0.0.1.0
Packaging Date: Thu Mar 01 23:06:35 2011
Size: 2.34 MB
FMRI: pkg://solaris/system/ldoms/ldomsmanager@2.2.0.0,5.11-0.175.0.0.0.1.0:20120221T141945Z
```

- 5 **ldmd 서비스를 다시 시작합니다.**

```
# svcadm restart ldmd
```

- 6 **올바른 ldm 버전이 실행 중인지 확인합니다.**

```
# ldm -V
```

- 7 **SP에 구성을 저장합니다.**

```
# ldm add-config config-name
```

다음 예제는 ldms-2.2-config라는 구성을 저장합니다.

```
# ldm add-config ldms-2.2-config
```

출하 시 기본 구성과 Logical Domains 사용 안함

플랫폼이 하나의 운영 체제만 호스트하는 단일 시스템으로 나타나는 초기 구성을 출하 시 기본 구성이라고 합니다. 논리적 도메인을 사용 안함으로 설정하려는 경우 이 구성을 복원하려는 것입니다. 그러면 시스템이 다른 도메인에 지정된 모든 리소스(CPU, 메모리, I/O)에 대한 액세스를 되찾을 수 있습니다.

이 절에서는 모든 게스트 도메인을 제거하고, 모든 Logical Domains 구성을 제거하고, 구성을 출하 시 기본값으로 되돌리는 방법을 설명합니다.

▼ 모든 게스트 도메인을 제거하는 방법

- 1 **-a 옵션을 사용하여 모든 도메인을 중지합니다.**

```
primary# ldm stop-domain -a
```

- 2 **primary 도메인을 제외한 모든 도메인의 바인딩을 해제합니다.**

```
primary# ldm unbind-domain ldom
```

주-I/O 도메인이 컨트롤 도메인에 필요한 서비스를 제공하는 경우 바인딩을 해제할 수 없습니다. 이 경우, 이 단계를 건너뛰니다.

- 3 **primary 도메인을 제외한 모든 도메인을 삭제합니다.**

```
primary# ldm remove-domain -a
```

▼ 모든 논리적 도메인 구성을 제거하는 방법

- 1 **서비스 프로세서(SP)에 저장된 모든 논리적 도메인 구성을 나열합니다.**

```
primary# ldm list-config
```

- 2 **factory-default** 구성을 제외한, 이전에 SP에 저장된 모든 구성(*config-name*)을 제거합니다.

각 구성에 대해 다음 명령을 사용합니다.

```
primary# ldm rm-config config-name
```

이전에 SP에 저장된 모든 구성을 제거한 후 컨트롤 도메인(primary)을 재부트할 때 사용할 그 다음 도메인은 **factory-default**입니다.

▼ 출하시 기본 구성을 복원하는 방법

- 1 출하시 기본 구성을 선택합니다.

```
primary# ldm set-config factory-default
```

- 2 컨트롤 도메인을 중지합니다.

```
primary# shutdown -i1 -g0 -y
```

- 3 시스템의 파워 사이클을 수행하여 출하시 기본 구성을 로드합니다.

```
-> stop /SYS
```

```
-> start /SYS
```

▼ Logical Domains Manager를 사용 안함으로 설정하는 방법

- 컨트롤 도메인에서 Logical Domains Manager를 사용 안함으로 설정합니다.

```
primary# svcadm disable ldmd
```

주 - Logical Domains Manager를 사용 안함으로 설정하면 실행 중인 도메인이 중지되지는 않지만, 새 도메인을 만들거나 기존 도메인의 구성을 변경하거나 도메인의 상태를 모니터링하는 것이 불가능합니다.



주의 - Logical Domains Manager를 사용 안함으로 설정하면 오류 보고나 전원 관리와 같은 일부 서비스가 사용 안함으로 설정됩니다. 오류 보고의 경우 **factory-default** 구성 상태에서 컨트롤 도메인을 재부트하여 오류 보고를 복원할 수 있습니다. 그러나 전원 관리는 해당하지 않습니다. 게다가, 일부 시스템 관리나 모니터링 도구는 Logical Domains Manager에 의존합니다.

▼ Logical Domains Manager를 제거하는 방법

출하 시 기본 구성을 복원하고 Logical Domains Manager를 사용 안함으로 설정한 후 Logical Domains Manager 소프트웨어를 제거할 수 있습니다.

주 - 출하 시 기본 구성을 복원하기 전에 Logical Domains Manager를 제거할 경우, 다음 절차에 표시된 대로 서비스 프로세서에서 출하 시 기본 구성을 복원할 수 있습니다.

- Logical Domains Manager 소프트웨어를 제거합니다.
 - Oracle Solaris 10 SUNWldm 및 SUNWldmp2v 패키지를 제거합니다.
`primary# pkgrm SUNWldm SUNWldmp2v`
 - Oracle Solaris 11 ldomsmanager 패키지를 제거합니다.
`primary# pkg uninstall ldomsmanager`

▼ 서비스 프로세서에서 출하 시 기본 구성을 복원하는 방법

출하 시 기본 구성을 복원하기 전에 Logical Domains Manager를 제거할 경우, 서비스 프로세서에서 출하 시 기본 구성을 복원할 수 있습니다.

- 1 서비스 프로세서에서 출하 시 기본 구성을 복원합니다.
-> `set /HOST/bootmode config=factory-default`
- 2 시스템의 파워 사이클을 수행하여 출하 시 기본 구성을 로드합니다.
-> `reset /SYS`

Oracle VM Server for SPARC 보안

이 장에서는 Oracle VM Server for SPARC 시스템에서 사용으로 설정할 수 있는 몇 가지 보안 기능에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- 43 페이지 “RBAC를 사용하여 논리적 도메인의 관리 위임”
- 47 페이지 “RBAC를 사용하여 도메인 콘솔에 대한 액세스 제어”
- 53 페이지 “감사를 사용으로 설정한 후 사용”

RBAC를 사용하여 논리적 도메인의 관리 위임

Logical Domains Manager 패키지는 로컬 RBAC 구성에 두 가지 사전 정의된 RBAC(역할 기반 액세스 제어) 권한 프로파일을 추가합니다. 이러한 권한 프로파일을 사용하여 권한이 없는 사용자에게 다음과 같은 관리 권한을 위임할 수 있습니다.

- LDoms Management 프로파일은 사용자가 모든 ldm 하위 명령을 사용할 수 있도록 허용합니다.
- LDoms Review 프로파일은 사용자가 모든 목록 관련 ldm 하위 명령을 사용할 수 있도록 허용합니다.

사용자 또는 역할에 이러한 권한 프로파일을 직접 지정할 수 있습니다. 역할에 지정할 경우 해당 역할이 사용자에게 지정됩니다. 이러한 프로파일 중 하나를 사용자에게 직접 지정할 경우 ldm 명령을 사용하여 도메인을 관리하려면 pfexec 명령 또는 프로파일 셸(예: pfbash 또는 pfksh)을 사용해야 합니다. RBAC 구성을 기반으로 사용자 역할 또는 권한 프로파일을 사용할지 여부를 결정하십시오. [System Administration Guide: Security Services](#) 또는 [Oracle Solaris 관리: 보안 서비스의 제III부](#), “역할, 권한 프로파일 및 권한”를 참조하십시오.

다음 방법으로 사용자, 권한 부여, 권한 프로파일 및 역할을 구성할 수 있습니다.

- 파일을 사용하여 시스템에서 로컬로
- 이름 지정 서비스(예: LDAP)에서 중앙 집중식으로

Logical Domains Manager를 설치하면 필요한 권한 프로파일이 로컬 파일에 추가됩니다. 이름 지정 서비스에서 프로파일 및 역할을 구성하려면 [System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)를 참조하십시오. 이 장의 모든 예에서는 RBAC 구성에 로컬 파일이 사용된다고 가정합니다. Logical Domains Manager 패키지를 통해 전달된 권한 부여 및 실행 속성에 대한 개요는 [46 페이지 “Logical Domains Manager 프로파일 콘텐츠”](#)를 참조하십시오.

권한 프로파일 및 역할 사용



주의 - `usermod` 및 `rolemod` 명령을 사용하여 권한 부여, 권한 프로파일 또는 역할을 추가할 때는 신중해야 합니다.

- Oracle Solaris 10 OS의 경우 `usermod` 또는 `rolemod` 명령이 기존 값을 바꿉니다.
값을 바꾸지 않고 추가하려면 기존 값과 새 값 목록을 쉼표로 구분하여 지정하십시오.
- Oracle Solaris 11 OS의 경우 추가하는 각 권한 부여에 대해 플러스 기호(+)를 사용하여 값을 추가하십시오.
예를 들어, `usermod -A +auth username` 명령은 `rolemod` 명령과 유사하게 사용자 `username`에게 `auth` 권한 부여를 허가합니다.

사용자 권한 프로파일 관리

다음 절차에서는 로컬 파일을 사용하여 시스템에서 사용자 권한 프로파일을 관리하는 방법을 보여줍니다. 이름 지정 서비스에서 사용자 프로파일을 관리하려면 [System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)를 참조하십시오.

▼ 사용자에게 권한 프로파일을 지정하는 방법

LDoms Management 프로파일이 직접 지정된 사용자는 반드시 프로파일 셀을 호출하여 보안 속성이 사용된 `ldm` 명령을 실행해야 합니다. 자세한 내용은 [System Administration Guide: Security Services](#) 또는 [Oracle Solaris 관리: 보안 서비스의 제III부, “역할, 권한 프로파일 및 권한”](#)를 참조하십시오.

- 1 관리자, 수퍼유저 또는 동등한 역할의 사용자로 로그인합니다.

Oracle Solaris 10의 경우, [System Administration Guide: Security Services](#)의 “Configuring RBAC (Task Map)”를 참조하십시오. Oracle Solaris 11의 경우, [Oracle Solaris 관리: 보안 서비스의 제III부, “역할, 권한 프로파일 및 권한”](#)를 참조하십시오.

- 2 로컬 사용자 계정에 관리 프로파일을 지정합니다.

사용자 계정에 LDoms Review 프로파일 또는 LDoms Management 프로파일을 지정할 수 있습니다.

```
# usermod -P "profile-name" username
```

다음 명령은 사용자 sam에게 LDoms Management 프로파일을 지정합니다.

```
# usermod -P "LDoms Management" sam
```

사용자에게 역할 지정

다음 절차에서는 로컬 파일을 사용하여 역할을 만든 후 사용자에게 해당 역할을 지정하는 방법을 보여줍니다. 이름 지정 서비스에서 역할을 관리하려면 [System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)를 참조하십시오.

이 절차를 수행할 경우 특정 역할이 지정된 사용자만 해당 역할을 맡을 수 있다는 장점이 있습니다. 역할에 암호가 지정된 경우 역할을 맡을 때 암호가 필요합니다. 이러한 두 가지 보안 계층은 역할이 지정되지 않고 암호가 없는 사용자가 해당 역할을 맡을 수 없도록 합니다.

▼ 역할을 만들어 사용자에게 지정하는 방법

- 1 관리자, 수퍼유저 또는 동등한 역할의 사용자로 로그인합니다.

Oracle Solaris 10의 경우, [System Administration Guide: Security Services](#)의 “Configuring RBAC (Task Map)”를 참조하십시오. Oracle Solaris 11의 경우, [Oracle Solaris 관리: 보안 서비스의 제III부](#), “역할, 권한 프로파일 및 권한”를 참조하십시오.

- 2 역할을 만듭니다.

```
# roleadd -P "profile-name" role-name
```

- 3 역할에 암호를 지정합니다.

새 암호를 지정한 다음 확인하라는 메시지가 표시됩니다.

```
# passwd role-name
```

- 4 사용자에게 역할을 지정합니다.

```
# useradd -R role-name username
```

- 5 사용자에게 암호를 지정합니다.

새 암호를 지정한 다음 확인하라는 메시지가 표시됩니다.

```
# passwd username
```

- 6 해당 사용자로 로그인하고 필요한 경우 암호를 제공합니다.

```
# su username
```

- 7 사용자에게 지정된 역할에 대한 액세스 권한이 있는지 확인합니다.

```
$ id
uid=nn(username) gid=nn(group-name)
$ roles
role-name
```

8 역할을 맡고 필요한 경우 암호를 제공합니다.

```
$ su role-name
```

9 사용자가 역할을 맡았는지 확인합니다.

```
$ id
uid=nn(role-name) gid=nn(group-name)
```

예 3-1 역할을 만들어 사용자에게 지정

이 예에서는 `ldm_read` 역할을 만들어 사용자 `user_1`에게 지정하고 사용자 `user_1`로 로그인한 다음 `ldm_read` 역할을 맡는 방법을 보여줍니다.

```
# roleadd -P "LDoms Review" ldm_read
# passwd ldm_read
New Password: ldm_read-password
Re-enter new Password: ldm_read-password
passwd: password successfully changed for ldm_read
# useradd -R ldm_read user_1
# passwd user_1
New Password: user_1-password
Re-enter new Password: user_1-password
passwd: password successfully changed for user_1
# su user_1
Password: user_1-password
$ id
uid=95555(user_1) gid=10(staff)
$ roles
ldm_read
$ su ldm_read
Password: ldm_read-password
$ id
uid=99667(ldm_read) gid=14(sysadmin)
```

Logical Domains Manager 프로파일 콘텐츠

Logical Domains Manager 패키지는 로컬 `/etc/security/prof_attr` 파일에 다음과 같은 RBAC 프로파일을 추가합니다.

- LDoms Review:::Review LDoms configuration:auths=solaris.ldoms.read
- LDoms Management:::Manage LDoms domains:auths=solaris.ldoms.*

또한 Logical Domains Manager 패키지는 로컬 `/etc/security/exec_attr` 파일에 LDoms Management 프로파일과 연관된 다음과 같은 실행 속성을 추가합니다.

```
LDoms Management:suser:cmd:::/usr/sbin/ldm:prvs=file_dac_read,file_dac_search
```

다음 표에서는 명령 실행에 필요한 해당 사용자 권한 부여와 함께 `ldm` 하위 명령을 나열합니다.

표 3-1 ldm 하위 명령 및 사용자 권한 부여

ldm 하위 명령 ¹	사용자 권한 부여
add-*	solaris.ldoms.write
bind-domain	solaris.ldoms.write
list	solaris.ldoms.read
list-*	solaris.ldoms.read
panic-domain	solaris.ldoms.write
remove-*	solaris.ldoms.write
set-*	solaris.ldoms.write
start-domain	solaris.ldoms.write
stop-domain	solaris.ldoms.write
unbind-domain	solaris.ldoms.write

¹ 추가, 나열, 제거 또는 설정할 수 있는 모든 리소스를 나타냅니다.

RBAC를 사용하여 도메인 콘솔에 대한 액세스 제어

기본적으로 모든 사용자가 모든 도메인 콘솔에 액세스할 수 있습니다. 도메인 콘솔에 대한 액세스를 제어하려면 권한 부여 확인이 수행되도록 `vntsd` 데몬을 구성하십시오. `vntsd` 데몬은 `vntsd/authorization`이라는 SMF(서비스 관리 기능) 등록 정보를 제공합니다. 도메인 콘솔 또는 콘솔 그룹에 대해 사용자 및 역할의 권한 부여 확인이 사용으로 설정되도록 이 등록 정보를 구성할 수 있습니다. 권한 부여 확인을 사용으로 설정하려면 `svccfg` 명령을 사용하여 이 등록 정보의 값을 `true`로 설정하십시오. 이 옵션이 사용으로 설정된 상태에서 `vntsd`는 `localhost`에서만 연결을 수신하고 수락합니다. `vntsd/authorization`이 사용으로 설정된 상태에서는 `listen_addr` 등록 정보가 대체 IP 주소를 지정하는 경우에도 `vntsd`는 대체 IP 주소를 무시하고 계속 `localhost`에서만 수신합니다.



주의 - `localhost` 이외의 다른 호스트를 사용하려면 `vntsd` 서비스를 구성하지 마십시오.

`localhost` 이외의 다른 호스트를 지정할 경우 더 이상 컨트롤 도메인에서 게스트 도메인 콘솔로의 연결이 제한되지 않습니다. `telnet` 명령을 사용하여 게스트 도메인에 원격으로 연결할 경우 네트워크를 통해 일반 텍스트로 로그인 자격 증명이 전달됩니다.

기본적으로 모든 게스트 콘솔에 액세스할 수 있는 권한 부여는 로컬 `auth_attr` 데이터베이스에서 제공됩니다.

```
solaris.vntsd.consoles:::Access All LDoms Guest Consoles::
```

로컬 파일에서 사용자 또는 역할에 필요한 권한 부여를 지정하려면 `usermod` 명령을 사용하십시오. 이 명령은 필요한 권한 부여를 가진 사용자 또는 역할만 지정된 도메인 콘솔 또는 콘솔 그룹에 액세스할 수 있도록 허용합니다. 이름 지정 서비스에서 사용자 또는 역할에 권한 부여를 지정하려면 [System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)를 참조하십시오.

모든 도메인 콘솔에 대한 액세스를 제어할 수도 있고, 단일 도메인 콘솔에 대한 액세스를 제어할 수도 있습니다.

- 모든 도메인 콘솔에 대한 액세스를 제어하려면 48 페이지 “역할을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법” 및 50 페이지 “권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법”을 참조하십시오.
- 단일 도메인 콘솔에 대한 액세스를 제어하려면 51 페이지 “역할을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법” 및 53 페이지 “권한 프로파일을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법”을 참조하십시오.

▼ 역할을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법

- 1 콘솔 권한 부여 확인을 사용으로 설정하여 도메인 콘솔에 대한 액세스를 제한합니다.

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

- 2 모든도메인 콘솔에 대한 액세스를 허용하는 `solaris.vntsd.consoles` 권한 부여를 가진 역할을 만듭니다.

```
primary# roleadd -A solaris.vntsd.consoles role-name
primary# passwd all_cons
```

- 3 사용자에게 새 역할을 지정합니다.

```
primary# usermod -R role-name username
```

예 3-2 역할을 사용하여 모든 도메인 콘솔에 대한 액세스 제어

먼저 도메인 콘솔에 대한 액세스가 제한되도록 콘솔 권한 부여 확인을 사용으로 설정합니다.

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
primary# ldm ls
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	8	16G	0.2%	47m
ldg1	active	-n--v-	5000	2	1G	0.1%	17h 50m
ldg2	active	-t----	5001	4	2G	25%	11s

다음 예에서는 모든 도메인 콘솔에 대한 액세스를 허용하는 `solaris.vntsd.consoles` 권한 부여를 가진 `all_cons` 역할을 만드는 방법을 보여줍니다.

```
primary# roleadd -A solaris.vntsd.consoles all_cons
primary# passwd all_cons
New Password:
Re-enter new Password:
passwd: password successfully changed for all_cons
```

이 명령은 사용자 `sam`에게 `all_cons` 역할을 지정합니다.

```
primary# usermod -R all_cons sam
```

그러면 사용자 `sam`이 `all_cons` 역할을 맡고 모든 콘솔에 액세스할 수 있습니다. 예를 들어, 다음과 같습니다.

```
$ id
uid=700299(sam) gid=1(other)
-bash-3.2$ su all_cons
Password:
$ telnet 0 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

```
$ telnet 0 5001
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg2" in group "ldg2" ....
Press ~? for control options ..
```

이 예에서는 권한이 부여되지 않은 사용자 `dana`가 도메인 콘솔에 액세스하려고 시도할 때 발생하는 오류를 보여줍니다.

```
$ id
uid=702048(dana) gid=1(other)
$ telnet 0 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.
Connection to 0 closed by foreign host.
```

▼ 권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법

1 **solaris.vntsd.consoles** 권한 부여를 가진 권한 프로파일을 만듭니다.

- Oracle Solaris 10 OS의 경우 `/etc/security/prof_attr` 파일을 편집합니다.
다음 항목을 포함합니다.

```
LDoms Consoles::Access LDoms Consoles:auths=solaris.vntsd.consoles
```

- Oracle Solaris 11 OS의 경우 **profiles** 명령을 사용하여 새 프로파일을 만듭니다.

```
primary# profiles -p "LDoms Consoles" \
'set desc="Access LDoms Consoles"; set auths=solaris.vntsd.consoles'
```

2 사용자에게 권한 프로파일을 지정합니다.

- Oracle Solaris 10 OS의 경우 사용자에게 권한 프로파일을 지정합니다.

```
primary# usermod -P "All,Basic Solaris User,LDoms Consoles" username
```

LDoms Consoles 프로파일을 추가할 때 기존 프로파일을 지정하려는 경우 신중해야 합니다. 이전 명령은 사용자에게 이미 All 및 Basic Solaris User 프로파일이 있음을 보여줍니다.

- Oracle Solaris 11 OS의 경우 사용자에게 권한 프로파일을 지정합니다.

```
primary# usermod -P +"LDoms Consoles" username
```

3 해당 사용자로 도메인 콘솔에 연결합니다.

```
$ telnet 0 5000
```

예 3-3 권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스 제어

다음 예에서는 권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법을 보여줍니다.

- Oracle Solaris 10: `/etc/security/prof_attr` 파일에 다음 항목을 추가하여 **solaris.vntsd.consoles** 권한 부여를 가진 권한 프로파일을 만듭니다.

```
LDoms Consoles::Access LDoms Consoles:auths=solaris.vntsd.consoles
```

*username*에 권한 프로파일을 지정합니다.

```
primary# usermod -P "All,Basic Solaris User,LDoms Consoles" username
```

다음 명령은 사용자가 sam이며 All, Basic Solaris User 및 LDoms Consoles 권한 프로파일이 적용되는지 확인하는 방법을 보여줍니다. telnet 명령은 ldg1 도메인 콘솔에 액세스하는 방법을 보여줍니다.

```
$ id
uid=702048(sam) gid=1(other)
$ profiles
All
Basic Solaris User
LDoms Consoles
$ telnet 0 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

- **Oracle Solaris 11:** profiles 명령을 사용하여 /etc/security/prop_attr 파일의 solaris.vntsd.consoles 권한 부여를 가진 권한 프로파일을 만듭니다.

```
primary# profiles -p "LDoms Consoles" \
'set desc="Access LDoms Consoles"; set auths=solaris.vntsd.consoles'
```

사용자에게 권한 프로파일을 지정합니다.

```
primary# usermod -P +"LDoms Consoles" sam
```

다음 명령은 사용자가 sam이며 All, Basic Solaris User 및 LDoms Consoles 권한 프로파일이 적용되는지 확인하는 방법을 보여줍니다. telnet 명령은 ldg1 도메인 콘솔에 액세스하는 방법을 보여줍니다.

```
$ id
uid=702048(sam) gid=1(other)
$ profiles
All
Basic Solaris User
LDoms Consoles
$ telnet 0 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

▼ 역할을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법

- 1 /etc/security/auth_attr 파일에 단일 도메인에 대한 권한 부여를 추가합니다.

권한 부여 이름은 도메인 이름에서 파생되며 solaris.vntsd.console-domain-name 형식입니다.

```
solaris.vntsd.console-domain-name::Access domain-name Console::
```

2 해당 도메인 콘솔에 대해서만 액세스를 허용하는 새 권한 부여를 가진 역할을 만듭니다.

```
primary# roleadd -A solaris.vntsd.console-domain-name role-name
primary# passwd role-name
New Password:
Re-enter new Password:
passwd: password successfully changed for role-name
```

3 사용자에게 *role-name* 역할을 지정합니다.

```
primary# usermod -R role-name username
```

예 3-4 단일 도메인 콘솔에 액세스

이 예에서는 사용자 terry가 ldg1cons 역할을 맡고 ldg1 도메인 콘솔에 액세스하는 방법을 보여줍니다.

먼저 /etc/security/auth_attr 파일에 단일 도메인 ldg1에 대한 권한 부여를 추가합니다.

```
solaris.vntsd.console-ldg1::Access ldg1 Console::
```

그런 다음 해당 도메인 콘솔에 대해서만 액세스를 허용하는 새 권한 부여를 가진 역할을 만듭니다.

```
primary# roleadd -A solaris.vntsd.console-ldg1 ldg1cons
primary# passwd ldg1cons
New Password:
Re-enter new Password:
passwd: password successfully changed for ldg1cons
```

사용자 terry에게 ldg1cons 역할을 지정하고 ldg1cons 역할을 맡은 다음 도메인 콘솔에 액세스합니다.

```
primary# usermod -R ldg1cons terry
primary# su ldg1cons
Password:
$ id
uid=700303(ldg1cons) gid=1(other)
$ telnet 0 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
```

```
Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

다음에서는 사용자 terry가 ldg2 도메인 콘솔에 액세스할 수 없음을 보여줍니다.

```
$ telnet 0 5001
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
Connection to 0 closed by foreign host.
```

▼ 권한 프로파일을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법

- 1 `/etc/security/auth_attr` 파일에 단일 도메인에 대한 권한 부여를 추가합니다.

다음 항목 예에서는 단일 도메인 콘솔에 대한 권한 부여를 추가합니다.

```
solaris.vntsd.console-domain-name::Access domain-name Console::
```

- 2 특정 도메인 콘솔에 액세스하는 권한 부여를 가진 권한 프로파일을 만듭니다.

- Oracle Solaris 10 OS의 경우 `/etc/security/prof_attr` 파일을 편집합니다.

```
domain-name Console::Access domain-name
Console:auths=solaris.vntsd.console-domain-name
```

이 항목은 한 행이어야 합니다.

- Oracle Solaris 11 OS의 경우 `profiles` 명령을 사용하여 새 프로파일을 만듭니다.

```
primary# profiles -p "domain-name Console" \
'set desc="Access domain-name Console";
set auths=solaris.vntsd.console-domain-name'
```

- 3 사용자에게 권한 프로파일을 지정합니다.

다음 명령은 사용자에게 프로파일을 지정합니다.

- Oracle Solaris 10 OS의 경우 권한 프로파일을 지정합니다.

```
primary# usermod -P "All,Basic Solaris User,domain-name Console" username
```

All 및 Basic Solaris User 프로파일이 필요합니다.

- Oracle Solaris 11 OS의 경우 권한 프로파일을 지정합니다.

```
primary# usermod -P +"domain-name Console" username
```

감사를 사용으로 설정한 후 사용

Logical Domains Manager는 Oracle Solaris OS 감사 기능을 사용하여 컨트롤 도메인에서 발생한 작업 및 이벤트의 내역을 검사합니다. 내역은 수행된 작업, 수행 시기 및 수행자 및 영향을 받는 대상에 대한 로그에 보존됩니다.

사용 중인 시스템에서 실행되는 Oracle Solaris OS의 버전에 따라 다음과 같이 감사 기능을 사용 및 사용 안함으로 설정할 수 있습니다.

- **Oracle Solaris 10 OS.** bsmconv 및 bsmunconv 명령을 사용합니다. bsmconv(1M) 및 bsmunconv(1M) 매뉴얼 페이지와 [System Administration Guide: Security Services](#)의 제VII부, “Auditing in Oracle Solaris”를 참조하십시오.
- **Oracle Solaris 11 OS.** audit 명령을 사용합니다. audit(1M) 매뉴얼 페이지와 [Oracle Solaris 관리: 보안 서비스](#)의 제VII부, “Oracle Solaris에서 감사”를 참조하십시오.

▼ 감사를 사용으로 설정하는 방법

시스템에서 Oracle Solaris 감사 기능을 구성하고 사용으로 설정해야 합니다. Oracle Solaris OS 감사 기능을 사용하여 컨트롤 도메인에서 발생한 작업 및 이벤트의 내역을 검사할 수 있습니다. 내역은 수행된 작업, 수행 시기, 수행자 및 영향을 받는 대상에 대한 로그에 보존됩니다. 기본적으로 Oracle Solaris 11 감사 기능은 사용으로 설정되어 있지만 이 경우에도 몇 가지 구성 단계를 수행해야 합니다.

주- 가상화 소프트웨어(vs) 클래스에 대해서는 기존 프로세스가 감사되지 **않습니다**. 일반 사용자가 시스템에 로그인하기 전에 이 단계를 수행해야 합니다.

- 1 **/etc/security/audit_event 및 /etc/security/audit_class 파일에 사용자 정의를 추가합니다.**

이러한 사용자 정의는 Oracle Solaris 업그레이드 간에 보존되지만 새 Oracle Solaris 설치 후에는 다시 추가해야 합니다.

- a. **audit_event 파일에 다음 항목(제공되지 않은 경우)을 추가합니다.**

```
40700:AUE_ldoms:ldoms administration:vs
```

- b. **audit_class 파일에 다음 항목(제공되지 않은 경우)을 추가합니다.**

```
0x10000000:vs:virtualization software
```

- 2 **(Oracle Solaris 10) /etc/security/audit_control 파일에 vs 클래스를 추가합니다.**

다음 /etc/security/audit_control 단편 예에서는 vs 클래스를 지정한 방법을 보여줍니다.

```
dir:/var/audit
flags:lo,vs
minfree:20
naflags:lo,na
```

- 3 **(Oracle Solaris 10) 감사 기능을 사용으로 설정합니다.**

- a. **bsmconv 명령을 실행합니다.**

```
# /etc/security/bsmconv
```

b. 시스템을 재부트합니다.

4 (Oracle Solaris 11) vs 감사 클래스를 사전 선택합니다.

a. 이미 선택된 감사 클래스를 확인합니다.

이미 선택된 감사 클래스가 업데이트된 클래스 세트에 속하는지 확인합니다. 다음 예에서는 `lo` 클래스가 이미 선택되어 있음을 보여줍니다.

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
```

b. vs 감사 클래스를 추가합니다.

```
# auditconfig -setflags [class],vs
```

`class`는 쉼표로 구분된 0개 이상의 감사 클래스입니다. `/etc/security/audit_class` 파일에서 감사 클래스 목록을 확인할 수 있습니다. Oracle VM Server for SPARC 시스템에서 `vs` 클래스를 포함해야 합니다.

예를 들어, 다음 명령은 `lo` 클래스와 `vs` 클래스를 모두 선택합니다.

```
# auditconfig -setflags lo,vs
```

c. (선택 사항) 관리자 또는 구성자로 프로세스를 감사하려면 시스템에서 로그아웃합니다.

로그아웃하지 않으려면 [Oracle Solaris 관리: 보안 서비스의 “로그인한 사용자의 사전 선택 마스크를 업데이트하는 방법”](#)을 참조하십시오.

5 감사 소프트웨어가 실행 중인지 확인합니다.

```
# auditconfig -getcond
```

감사 소프트웨어가 실행 중이면 출력에 `audit condition = auditing`이 나타납니다.

▼ 감사를 사용 안함으로 설정하는 방법

● 감사 기능을 사용 안함으로 설정합니다.

■ Oracle Solaris 10 시스템에서 감사 기능을 사용 안함으로 설정합니다.

a. `bsmunconv` 명령을 실행합니다.

```
# /etc/security/bsmunconv
Are you sure you want to continue? [y/n] y
This script is used to disable the Basic Security Module (BSM).
Shall we continue the reversion to a non-BSM system now? [y/n] y
bsmunconv: INFO: removing c2audit:audit_load from /etc/system.
bsmunconv: INFO: stopping the cron daemon.
```

The Basic Security Module has been disabled.
Reboot this system now to come up without BSM.

b. 시스템을 재부트합니다.

- Oracle Solaris 11 시스템에서 감사 기능을 사용 안함으로 설정합니다.

a. `audit -t` 명령을 실행합니다.

```
# audit -t
```

b. 감사 소프트웨어가 더 이상 실행 중이 아닌지 확인합니다.

```
# auditconfig -getcond  
audit condition = noaudit
```

▼ 감사 레코드를 검토하는 방법

- 다음 방법 중 하나로 vs 감사 출력을 검토합니다.

- `auditreduce` 및 `praudit` 명령을 사용하여 감사 출력을 검토합니다.

```
# auditreduce -c vs | praudit  
# auditreduce -c vs -a 20060502000000 | praudit
```

- `praudit -x` 명령을 사용하여 XML 형식으로 감사 레코드를 인쇄합니다.

▼ 감사 로그를 교체하는 방법

- `audit -n` 명령을 사용하여 감사 로그를 교체합니다.

감사 로그를 교체하면 현재 감사 파일은 닫히고 현재 감사 디렉토리에서 새 파일이 열립니다.

서비스 및 컨트롤 도메인 설정

이 장에서는 기본 서비스 및 컨트롤 도메인을 설정하는 데 필요한 절차를 설명합니다.

또한 Oracle VM Server for SPARC Configuration Assistant를 사용하여 논리적 도메인과 서비스를 구성할 수 있습니다. 14 장, “[Oracle VM Server for SPARC Configuration Assistant\(Oracle Solaris 10\)](#)”를 참조하십시오.

이 장에서는 다음 주제를 다룹니다.

- 57 페이지 “출력 메시지”
- 58 페이지 “기본 서비스 만들기”
- 59 페이지 “컨트롤 도메인의 초기 구성”
- 61 페이지 “Logical Domains를 사용하도록 재부트”
- 61 페이지 “컨트롤/서비스 도메인과 다른 도메인 사이의 네트워킹 사용”
- 62 페이지 “가상 네트워크 터미널 서버 데몬 사용”

출력 메시지

Oracle VM Server for SPARC 2.0 릴리스부터 컨트롤 도메인에서 리소스를 동적으로 구성할 수 없는 경우 먼저 지연된 재구성을 시작하는 것이 좋습니다. 컨트롤 도메인이 재부트될 때까지 지연된 재구성이 구성 작업을 연기합니다.

primary 도메인에서 지연된 재구성을 시작할 때 다음 메시지가 수신됩니다.

```
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the
primary domain reboots, at which time the new configuration for the
primary domain also takes effect.
```

재부트할 때까지 primary 도메인에 매 후속 작업마다 다음 공지가 수신됩니다.

```
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
```

기본 서비스 만들기

컨트롤 도메인을 서비스 도메인으로 사용하고 다른 도메인에 대한 가상 장치를 만들려면 다음 가상 장치 서비스를 만들어야 합니다.

- vcc – 가상 콘솔 집중기 서비스
- vds – 가상 디스크 서버
- vsw – 가상 스위치 서비스

▼ 기본 서비스를 만드는 방법

- 1 가상 콘솔 집중기(vcc) 서비스를 만듭니다. 가상 네트워크 터미널 서버 데몬(vntsd)에서 모든 논리적 도메인 콘솔의 집중기로 사용할 수 있습니다.

예를 들어, 다음 명령은 포트 범위 5000 - 5100의 가상 콘솔 집중기 서비스(primary-vcc0)를 컨트롤 도메인(primary)에 추가합니다.

```
primary# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

- 2 가상 디스크 서버(vds)를 만듭니다. 가상 디스크를 논리적 도메인으로 가져올 수 있습니다.

예를 들어, 다음 명령은 가상 디스크 서버(primary-vds0)를 컨트롤 도메인(primary)에 추가합니다.

```
primary# ldm add-vds primary-vds0 primary
```

- 3 가상 스위치 서비스(vsw)를 만듭니다. 논리적 도메인에서 가상 네트워크(vnet) 장치 사이의 네트워킹을 사용으로 설정할 수 있습니다.

각 논리적 도메인이 가상 스위치를 통해 범위 밖에서 통신해야 하는 경우 GLDv3 호환 네트워크 어댑터를 가상 스위치에 지정합니다.

- Oracle Solaris 10에서는 네트워크 어댑터 드라이버의 가상 스위치 서비스를 컨트롤 도메인에 추가합니다.

```
primary# ldm add-vsw net-dev=net-driver vsw-service primary
```

예를 들어, 다음 명령은 네트워크 어댑터 드라이버 nxge0의 가상 스위치 서비스(primary-vsw0)를 컨트롤 도메인(primary)에 추가합니다.

```
primary# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

- Oracle Solaris 11에서는 네트워크 어댑터 드라이버 net0의 가상 스위치 서비스(primary-vsw0)를 컨트롤 도메인(primary)에 추가합니다.

```
primary# ldm add-vsw net-dev=net-driver vsw-service primary
```

예를 들어, 다음 명령은 네트워크 어댑터 드라이버 net0의 가상 스위치 서비스(primary-vsw0)를 컨트롤 도메인(primary)에 추가합니다.

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

- 다음은 Oracle Solaris 10 OS에만 적용되고 Oracle Solaris 11 시스템에 수행하면 안됩니다.

이 명령은 가상 스위치에 MAC 주소를 자동으로 할당합니다. `ldm add-vsw` 명령의 옵션으로 고유의 MAC 주소를 지정할 수 있습니다. 그러나 이 경우 지정된 MAC 주소가 기존의 MAC 주소와 충돌하지 않도록 해야 합니다.

추가 중인 가상 스위치가 기본 물리적 어댑터를 주 네트워크 인터페이스로 대체하는 경우 물리적 어댑터의 MAC 주소가 지정되어야 합니다. 그래야 DHCP(Dynamic Host Configuration Protocol) 서버가 동일한 IP 주소에 도메인을 지정합니다. [61 페이지 “컨트롤/서비스 도메인과 다른 도메인 사이의 네트워킹 사용”](#)을 참조하십시오.

```
primary# ldm add-vsw mac-addr=2:04:4f:fb:9f:0d net-dev=nxge0 primary-vsw0 primary
```

4 list-services 하위 명령을 사용하여 서비스가 만들어졌는지 확인합니다.

다음과 비슷한 출력이 나타나야 합니다.

```
primary# ldm list-services primary
VDS
  NAME          VOLUME          OPTIONS          DEVICE
  primary-vds0

VCC
  NAME          PORT-RANGE
  primary-vcc0  5000-5100

VSW
  NAME          MAC              NET-DEV          DEVICE          MODE
  primary-vsw0  02:04:4f:fb:9f:0d nxge0            switch@0        prog,promisc
```

컨트롤 도메인의 초기 구성

처음에 모든 시스템 리소스가 컨트롤 도메인에 할당됩니다. 다른 논리적 도메인을 만들려면 이러한 리소스의 일부를 해제해야 합니다.

컨트롤 도메인의 초기 구성을 수행할 때 메모리 동적 재구성(DR)을 사용하려고 시도하지 **마십시오**. 메모리 DR을 사용하여 재부트 없이 이 구성을 수행할 수 있더라도 이는 권장되지 **않습니다**. 메모리 DR 접근법은 시간이 오래 걸리므로(재부트보다 오래 걸림) 실패할 가능성이 높습니다. 대신, 메모리 구성을 변경하기 전에 `ldm start-reconf` 명령을 사용하여 컨트롤 도메인을 지연된 재구성 모드에 놓습니다. 그런 다음, 모든 구성 단계를 완료한 후 컨트롤 도메인을 재부트할 수 있습니다.

▼ 컨트롤 도메인을 설정하는 방법

주 - 이 절차에는 컨트롤 도메인에 설정할 리소스의 예가 포함됩니다. 이러한 숫자는 예시일 뿐이며, 사용된 값이 사용자의 컨트롤 도메인에 부적합할 수 있습니다.

1 컨트롤 도메인에 암호화 장치가 있는지 여부를 확인합니다.

UltraSPARC T2, UltraSPARC T2 Plus, SPARC T3 플랫폼만 암호화 장치를 사용할 수 있습니다.

```
primary# ldm list -o crypto primary
```

2 해당하는 경우, 컨트롤 도메인에 암호화 리소스를 지정합니다.

다음 예제는 1개의 암호화 리소스를 컨트롤 도메인 `primary`에 지정합니다. 나머지 암호화 리소스는 게스트 도메인에 사용하도록 남겨 둡니다.

```
primary# ldm set-mau 1 primary
```

3 컨트롤 도메인에 가상 CPU를 지정합니다.

예를 들어, 다음 명령은 8개의 가상 CPU를 컨트롤 도메인 `primary`에 지정합니다. 나머지 가상 CPU는 게스트 도메인에 사용하도록 남겨 둡니다.

```
primary# ldm set-vcpu 8 primary
```

4 컨트롤 도메인에서 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf primary
```

5 컨트롤 도메인에 메모리를 지정합니다.

예를 들어, 다음 명령은 4GB의 메모리를 컨트롤 도메인 `primary`에 지정합니다. 나머지 메모리는 게스트 도메인에 사용하도록 남겨 둡니다.

```
primary# ldm set-memory 4G primary
```

6 논리적 도메인 컴퓨터 구성을 서비스 프로세서(SP)에 추가합니다.

예를 들어, 다음 명령은 `initial`이라는 구성을 추가합니다.

```
primary# ldm add-config initial
```

7 다음 재부트 시 구성이 사용될 준비가 되었는지 확인합니다.

```
primary# ldm list-config
factory-default
initial [next poweron]
```

이 list 하위 명령은 `initial` 구성 세트가 파워 사이클 후 사용될 것임을 보여줍니다.

Logical Domains를 사용하도록 재부트

구성 변경 사항을 적용하고 다른 논리적 도메인에 사용할 리소스를 해제하려면 컨트롤 도메인을 재부트해야 합니다.

▼ 재부트하는 방법

- 컨트롤 도메인을 종료하고 재부트합니다.

```
primary# shutdown -y -g0 -i6
```

주 - 재부트나 파워 사이클은 새 구성을 인스턴스화합니다. 파워 사이클만 실제로 서비스 프로세서(SP)에 저장된 구성을 부트하고, 그 다음 list-config 출력에 반영됩니다.

컨트롤/서비스 도메인과 다른 도메인 사이의 네트워킹 사용



주의 - 이 절은 Oracle Solaris 10 시스템에 **만** 적용됩니다. Oracle Solaris 11 시스템에 vsw 인터페이스를 구성하지 **마십시오**.

기본적으로 컨트롤 도메인과 시스템의 다른 도메인 사이의 네트워킹은 사용 안함으로 설정됩니다. 사용으로 설정하려면 가상 스위치 장치를 네트워크 장치로 구성해야 합니다. 가상 스위치는 기본 물리적 장치(이 예제에서 `nxge0`)를 주 인터페이스로 대체하거나 도메인에 추가 네트워크 인터페이스로 구성할 수 있습니다.

해당하는 네트워크 백엔드 장치가 동일한 가상 LAN 또는 가상 네트워크에 구성된 경우, 게스트 도메인은 컨트롤 도메인 또는 서비스 도메인과 자동으로 통신할 수 있습니다.

주 - 다음 절차는 도메인에 네트워크 연결을 일시적으로 중단할 수 있으므로 컨트롤 도메인의 콘솔에서 수행하십시오.

▼ 가상 스위치를 주 인터페이스로 구성하는 방법

주-필요한 경우, 가상 스위치와 물리적 네트워크 장치를 구성할 수 있습니다. 이 경우 단계 2에서 가상 스위치를 만들고 물리적 장치를 삭제하지 마십시오(단계 3 생략). 그 다음 가상 스위치를 정적 IP 주소 또는 동적 IP 주소로 구성해야 합니다. DHCP 서버에서 동적 IP 주소를 얻을 수 있습니다. 이 사례의 추가 정보와 예제는 [150 페이지 “NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성”](#)을 참조하십시오.

- 1 모든 인터페이스에 대한 주소 지정 정보를 인쇄합니다.

```
primary# ifconfig -a
```

- 2 가상 스위치를 만듭니다.

```
primary# ifconfig vsw0 plumb
```

- 3 가상 스위치에 지정된 물리적 네트워크 장치(net-dev)를 삭제합니다.

```
primary# ifconfig nxge0 down unplumb
```

- 4 물리적 네트워크 장치(nxge0)의 등록 정보를 가상 스위치 장치(vsw0)로 마이그레이션하려면 다음 중 하나를 수행합니다.

- 네트워킹이 정적 IP 주소로 구성된 경우 가상 스위치에 대해 nxge0의 IP 주소와 넷마스크를 재사용합니다.

```
primary# ifconfig vsw0 IP-of-nxge0 netmask netmask-of-nxge0 broadcast + up
```

- 네트워킹이 DHCP로 구성된 경우 가상 스위치에 대해 DHCP를 사용으로 설정합니다.

```
primary# ifconfig vsw0 dhcp start
```

- 5 필요한 구성 파일을 수정하여 이 변경 사항을 영구 저장합니다.

```
primary# mv /etc/hostname.nxge0 /etc/hostname.vsw0
```

```
primary# mv /etc/dhcp.nxge0 /etc/dhcp.vsw0
```

가상 네트워크 터미널 서버 데몬 사용

각 논리적 도메인의 가상 콘솔에 액세스하려면 가상 네트워크 터미널 서버 데몬(vntsd)을 사용으로 설정해야 합니다. 이 도메인의 사용 방법은 vntsd(1M) 매뉴얼 페이지를 참조하십시오.

▼ 가상 네트워크 터미널 서버 데몬을 사용으로 설정하는 방법

주-vntsd를 사용으로 설정하기 전에 컨트롤 도메인에 기본 서비스 vconscon(vcc)이 만들어졌는지 확인합니다. 자세한 내용은 [58 페이지](#) “기본 서비스 만들기”를 참조하십시오.

- 1 **svcadm** 명령을 사용하여 가상 네트워크 터미널 서버 데몬 **vntsd**를 사용으로 설정합니다.

```
primary# svcadm enable vntsd
```

- 2 **svcs** 명령을 사용하여 **vntsd** 데몬이 사용으로 설정되었는지 확인합니다.

```
primary# svcs vntsd
STATE      STIME      FMRI
online     Oct_08     svc:/ldoms/vntsd:default
```


게스트 도메인 설정

이 장에서는 게스트 도메인을 설정하는 데 필요한 절차를 설명합니다.

또한 Oracle VM Server for SPARC Configuration Assistant를 사용하여 논리적 도메인과 서비스를 구성할 수 있습니다. 14 장, “[Oracle VM Server for SPARC Configuration Assistant\(Oracle Solaris 10\)](#)”를 참조하십시오.

이 장에서는 다음 주제를 다룹니다.

- 65 페이지 “게스트 도메인 만들기 및 시작”
- 68 페이지 “게스트 도메인에 [Oracle Solaris OS 설치](#)”

게스트 도메인 만들기 및 시작

게스트 도메인은 sun4v 플랫폼과 하이퍼바이저에서 제공한 가상 장치를 인식하는 운영 체제를 실행해야 합니다. 현재 시점에서 최소한 Oracle Solaris 10 11/06 OS를 실행해야 합니다. Oracle Solaris 10 8/11 OS를 실행하면 모든 Oracle VM Server for SPARC 2.2 기능이 제공됩니다. 특정 패치가 필요한 경우 [Oracle VM Server for SPARC 2.2 릴리스 노트](#)를 참조하십시오. 컨트롤 도메인에서 기본 서비스를 만들고 리소스를 재할당한 후에 게스트 도메인을 만들고 시작할 수 있습니다.

▼ 게스트 도메인을 만들고 시작하는 방법

- 1 논리적 도메인을 만듭니다.

예를 들어, 다음 명령은 ldg1이라는 게스트 도메인을 만듭니다.

```
primary# ldm add-domain ldg1
```

- 2 게스트 도메인에 CPU를 추가합니다.

예를 들어, 다음 명령은 8개의 가상 CPU를 게스트 도메인 ldg1에 추가합니다.

```
primary# ldm add-vcpu 8 ldg1
```

3 게스트 도메인에 메모리를 추가합니다.

예를 들어, 다음 명령은 2GB의 메모리를 게스트 도메인 `ldg1`에 추가합니다.

```
primary# ldm add-memory 2G ldg1
```

4 게스트 도메인에 가상 네트워크 장치를 추가합니다.

예를 들어, 다음 명령은 아래와 같은 가상 네트워크 장치를 게스트 도메인 `ldg1`에 추가합니다.

```
primary# ldm add-vnet vnet1 primary-vsw0 ldg1
```

여기서:

- `vnet1`은 이 가상 네트워크 장치 인스턴스에 지정된 논리적 도메인의 고유한 인터페이스 이름으로, 나중에 `set-vnet` 또는 `remove-vnet` 하위 명령에서 참조할 수 있습니다.
- `primary-vsw0`은 연결할 기존 네트워크 서비스(가상 스위치)의 이름입니다.

주 - 단계 5-6은 `primary` 도메인에 가상 디스크 서버 장치(`vdsdev`)를 추가하고 게스트 도메인에 가상 디스크(`vdisk`)를 추가하기 위한 간단한 지침입니다. ZFS 볼륨 및 파일 시스템을 가상 디스크로 사용하는 방법을 알아보려면 [118 페이지 “ZFS 볼륨을 단일 슬라이스 디스크로 내보내는 방법”](#) 및 [127 페이지 “가상 디스크에서 ZFS 사용”](#)을 참조하십시오.

5 가상 디스크 서버가 게스트 도메인에 가상 디스크로 내보낼 장치를 지정합니다.

물리적 디스크, 디스크 슬라이스, 볼륨 또는 파일을 블록 장치로 내보낼 수 있습니다. 다음 예제는 물리적 디스크 및 파일을 보여줍니다.

- **물리적 디스크 예제.** 첫번째 예제는 아래와 같은 물리적 디스크를 추가합니다.

```
primary# ldm add-vdsdev /dev/dsk/c2t1d0s2 vol1@primary-vds0
```

여기서:

- `/dev/dsk/c2t1d0s2`는 실제 물리적 장치의 경로 이름입니다. 장치를 추가할 때 경로 이름이 장치 이름과 쌍을 이루어야 합니다.
- `vol1`은 가상 디스크 서버에 추가하려는 장치에 지정할 고유 이름입니다. 볼륨 이름은 이 가상 디스크 서버 인스턴스에 고유해야 합니다. 가상 디스크 서버가 이 이름을 클라이언트로 내보내기 때문입니다. 장치를 추가할 때 볼륨 이름이 실제 장치의 경로 이름과 쌍을 이루어야 합니다.
- `primary-vds0`은 이 장치를 추가할 가상 디스크 서버의 이름입니다.
- **파일 예제.** 이 두번째 예제는 파일을 블록 장치로 내보내는 것입니다.

```
primary# ldm add-vdsdev backend vol1@primary-vds0
```

여기서:

- *backend*는 블록 장치로 내보낸 실제 파일의 경로 이름입니다. 장치를 추가할 때 백엔드가 장치 이름과 쌍을 이루어야 합니다.
- *vol1*은 가상 디스크 서버에 추가하려는 장치에 지정할 고유 이름입니다. 볼륨 이름은 이 가상 디스크 서버 인스턴스에 고유해야 합니다. 가상 디스크 서버가 이름을 클라이언트로 내보내기 때문입니다. 장치를 추가할 때 볼륨 이름이 실제 장치의 경로 이름과 쌍을 이루어야 합니다.
- *primary-vds0*은 이 장치를 추가할 가상 디스크 서버의 이름입니다.

6 게스트 도메인에 가상 디스크를 추가합니다.

다음 예제는 가상 디스크를 게스트 도메인 *ldg1*에 추가합니다.

```
primary# ldm add-vdisk vdisk1 vol1@primary-vds0 ldg1
```

여기서:

- *vdisk1*은 가상 디스크의 이름입니다.
- *vol1*은 연결할 기존 볼륨의 이름입니다.
- *primary-vds0*은 연결할 기존 가상 디스크 서버의 이름입니다.

주 - 가상 디스크는 여러 유형의 물리적 장치, 볼륨, 파일과 연관된 일반 블록 장치입니다. 가상 디스크는 SCSI 디스크와 동의어가 아니므로 디스크 레이블에서 대상 ID가 제외됩니다. 논리적 도메인의 가상 디스크는 *cNdNsN* 형식을 사용합니다. 여기서 *cN*은 가상 컨트롤러이고 *dN*은 가상 디스크 번호이고 *sN*은 슬라이스입니다.

7 게스트 도메인에 대해 *auto-boot?* 및 *boot-device* 변수를 설정합니다.

첫번째 예제 명령은 게스트 도메인 *ldg1*에 대해 *auto-boot?*를 *true*로 설정합니다.

```
primary# ldm set-var auto-boot\?=true ldg1
```

두번째 예제 명령은 게스트 도메인 *ldg1*에 대해 *boot-device*를 *vdisk*로 설정합니다.

```
primary# ldm set-var boot-device=vdisk1 ldg1
```

8 게스트 도메인 *ldg1*에 리소스를 바인딩하고 도메인 목록을 나열하여 바인딩되었는지 확인합니다.

```
primary# ldm bind-domain ldg1
```

```
primary# ldm list-domain ldg1
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg1	bound	-----	5000	8	2G		

9 게스트 도메인의 콘솔 포트를 찾으려면 앞의 *list-domain* 하위 명령의 출력을 살펴볼 수 있습니다.

머리글 CONS 아래를 보면 논리적 도메인 게스트 1(*ldg1*)의 콘솔 출력이 포트 5000에 바인딩된 것을 알 수 있습니다.

- 10 컨트롤 도메인에 로그인하고 로컬 호스트의 콘솔 포트에 직접 연결하여 다른 터미널에서 게스트 도메인의 콘솔에 연결합니다.

```
$ ssh hostname.domain-name
$ telnet localhost 5000
```

- 11 게스트 도메인 ldg1을 시작합니다.

```
primary# ldm start-domain ldg1
```

게스트 도메인에 Oracle Solaris OS 설치

이 절에서는 게스트 도메인에 Oracle Solaris OS를 설치할 수 있는 여러 가지 방법의 지침을 제공합니다.



주의 - Oracle Solaris OS 설치 중 가상 콘솔에서 연결을 끊지 마십시오.

Oracle Solaris 11 도메인의 경우 DefaultFixed 네트워크 구성 프로파일(NCP)을 사용합니다. 설치 중이나 설치 후에 이 프로파일을 사용으로 설정할 수 있습니다.

Oracle Solaris 11 설치 중에는 수동 네트워킹 구성을 선택합니다. Oracle Solaris 11 설치 후에는 `netadm list` 명령을 사용하여 DefaultFixed NCP가 사용으로 설정되었는지 확인합니다. [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 7 장](#), “프로파일에 데이터 링크 및 인터페이스 구성 명령 사용”을 참조하십시오.

▼ DVD에서 게스트 도메인에 Oracle Solaris OS를 설치하는 방법

- 1 Oracle Solaris 10 OS 또는 Oracle Solaris 11 OS DVD를 DVD 드라이브에 넣습니다.

- 2 primary 도메인에서 볼륨 관리 데몬 `vol`(1M)를 중지합니다.

```
primary# svcadm disable volfs
```

- 3 게스트 도메인(ldg1)을 중지하고 바인딩을 해제합니다.

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
```

- 4 DVD-ROM 매체가 있는 DVD를 보조 볼륨 및 가상 디스크로 추가합니다.

다음은 `c0t0d0s2`를 Oracle Solaris 매체가 상주하는 DVD 드라이브로, `dvd_vol@primary-vds0`을 보조 볼륨으로, `vdisk_cd_media`를 가상 디스크로 사용합니다.

```
primary# ldm add-vdsdev /dev/dsk/c0t0d0s2 dvd_vol@primary-vds0
primary# ldm add-vdisk vdisk_cd_media dvd_vol@primary-vds0 ldg1
```

5 DVD가 보조 볼륨 및 가상 디스크로 추가되었는지 확인합니다.

```
primary# ldm list-bindings
NAME          STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active   -n-cv    SP      4       4G        0.2%    22h 45m
...
VDS
  NAME          VOLUME          OPTIONS          DEVICE
  primary-vds0  voll            /dev/dsk/c2t1d0s2
  dvd_vol       /dev/dsk/c0t0d0s2
....
-----
NAME          STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg1          inactive -----          60      6G
...
...
DISK
  NAME          VOLUME          TOUT DEVICE  SERVER
  vdisk1        voll@primary-vds0
  vdisk_cd_media dvd_vol@primary-vds0
....
```

6 게스트 도메인(ldg1)을 바인딩하고 시작합니다.

```
primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
primary# telnet localhost 5000
Trying 027.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

7 클라이언트 OpenBoot PROM에 장치 별명을 표시합니다.

이 예제에서 vdisk_cd_media의 장치 별명은 Oracle Solaris DVD이고, vdisk1은 Oracle Solaris OS를 설치할 수 있는 가상 디스크입니다.

```
ok devalias
vdisk_cd_media /virtual-devices@100/channel-devices@200/disk@1
vdisk1        /virtual-devices@100/channel-devices@200/disk@0
vnet1         /virtual-devices@100/channel-devices@200/network@0
virtual-console /virtual-devices/console@1
name          aliases
```

8 게스트 도메인의 콘솔에서 vdisk_cd_media(disk@1)를 슬라이스 f에 부트합니다.

```
ok boot vdisk_cd_media:f
Boot device: /virtual-devices@100/channel-devices@200/disk@1:f File and args: -s
SunOS Release 5.10 Version Generic_139555-08 64-bit
Copyright (c), 1983-2010, Oracle and/or its affiliates. All rights reserved.
```

9 Oracle Solaris OS 설치 메뉴를 계속합니다.

▼ Oracle Solaris ISO 파일에서 게스트 도메인에 Oracle Solaris OS를 설치하는 방법

- 1 게스트 도메인(ldg1)을 중지하고 바인딩을 해제합니다.

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
```

- 2 Oracle Solaris ISO 파일을 보조 볼륨 및 가상 디스크로 추가합니다.

다음은 solarisdvd.iso를 Oracle Solaris ISO 파일로, iso_vol@primary-vds0을 보조 볼륨으로, vdisk_iso를 가상 디스크로 사용합니다.

```
primary# ldm add-vdsdev /export/solarisdvd.iso iso_vol@primary-vds0
primary# ldm add-vdisk vdisk_iso iso_vol@primary-vds0 ldg1
```

- 3 Oracle Solaris ISO 파일이 보조 볼륨 및 가상 디스크로 추가되었는지 확인합니다.

```
primary# ldm list-bindings
NAME          STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active   -n-cv    SP      4       4G        0.2%    22h 45m
...
VDS
  NAME          VOLUME          OPTIONS          DEVICE
  primary-vds0  voll            /dev/dsk/c2t1d0s2
  iso_vol       /export/solarisdvd.iso
....
-----
NAME          STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg1          inactive -----
...
DISK
  NAME          VOLUME          TOUT ID DEVICE  SERVER  MPGROUP
  vdisk1        voll@primary-vds0
  vdisk_iso     iso_vol@primary-vds0
....
```

- 4 게스트 도메인(ldg1)을 바인딩하고 시작합니다.

```
primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^J'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

- 5 클라이언트 OpenBoot PROM에 장치 별명을 표시합니다.

이 예제에서 vdisk_iso의 장치 별명은 Oracle Solaris ISO 이미지이고, vdisk_install은 디스크 공간입니다.

```
ok devalias
vdisk_iso      /virtual-devices@100/channel-devices@200/disk@1
```

```
vdisk1          /virtual-devices@100/channel-devices@200/disk@0
vnet1           /virtual-devices@100/channel-devices@200/network@0
virtual-console /virtual-devices/console@1
name            aliases
```

- 6 게스트 도메인의 콘솔에서 **vdisk_iso(disk@1)**를 슬라이스 **f**에 부트합니다.

```
ok boot vdisk_iso:f
Boot device: /virtual-devices@100/channel-devices@200/disk@1:f File and args: -s
SunOS Release 5.10 Version Generic_139555-08 64-bit
Copyright (c) 1983-2010, Oracle and/or its affiliates. All rights reserved.
```

- 7 Oracle Solaris OS 설치 메뉴를 계속합니다.

▼ Oracle Solaris 10 게스트 도메인에서 Oracle Solaris JumpStart 기능을 사용하는 방법

주 - Oracle Solaris JumpStart 기능은 Oracle Solaris 10 OS에만 사용할 수 있습니다. Oracle Solaris 11 OS의 자동 설치를 수행하려면 자동 설치 프로그램 기능을 사용하십시오. [Oracle Solaris 10 JumpStart에서 Oracle Solaris 11 자동 설치 프로그램으로 전환](#)를 참조하십시오.

이 절차는 게스트 도메인에서 Oracle Solaris JumpStart 기능을 사용하는 방법을 설명합니다. 이 절차는 일반적인 JumpStart 절차를 따르지만, 게스트 도메인을 위해 JumpStart 프로파일에 사용할 여러 디스크 장치 이름 형식을 설명합니다. [Oracle Solaris 10 8/11 설치 설명서: 사용자 정의 JumpStart 및 고급 설치](#)를 참조하십시오.

논리적 도메인의 가상 디스크 장치 이름은 물리적 디스크 장치 이름과 다릅니다. 가상 디스크 장치 이름에는 대상 ID(tN)를 넣지 않습니다.

일반적인 cNtNdNsN 형식 대신, 가상 디스크 장치 이름은 cNdNsN 형식을 사용합니다. 여기서 cN은 가상 컨트롤러이고 dN은 가상 디스크 번호이고 sN은 슬라이스 번호입니다.

- 이 변경 사항을 반영하도록 JumpStart 프로파일을 수정합니다.

가상 디스크는 전체 디스크 또는 단일 슬라이스 디스크로 나타낼 수 있습니다. Oracle Solaris OS는 다중 분할 영역을 지정하는 일반 JumpStart 프로파일을 사용하여 전체 디스크에 설치할 수 있습니다. 단일 슬라이스 디스크는 단일 분할 영역 s0이 디스크 전체를 사용합니다. 단일 디스크에 Oracle Solaris OS를 설치하려면 단일 분할 영역(/)이 디스크 전체를 사용하는 프로파일을 사용해야 합니다. 교체 분할 영역과 같은 다른 분할

영역은 정의할 수 없습니다. 전체 디스크 및 단일 슬라이스 디스크에 대한 자세한 내용은 [111 페이지 “가상 디스크 표시”](#)를 참조하십시오.

- UFS 루트 파일 시스템을 설치하기 위한 JumpStart 프로파일.

Oracle Solaris 10 8/11 설치 설명서: 사용자 정의 JumpStart 및 고급 설치를
참조하십시오.

일반 UFS 프로파일

```
filesys c1t1d0s0 free /
filesys c1t1d0s1 2048 swap
filesys c1t1d0s5 120 /spare1
filesys c1t1d0s6 120 /spare2
```

전체 디스크에 도메인을 설치하기 위한 실제 UFS 프로파일

```
filesys c0d0s0 free /
filesys c0d0s1 2048 swap
filesys c0d0s5 120 /spare1
filesys c0d0s6 120 /spare2
```

단일 슬라이스 디스크에 도메인을 설치하기 위한 실제 UFS 프로파일

```
filesys c0d0s0 free /
```

- ZFS 루트 파일 시스템을 설치하기 위한 JumpStart 프로파일.

Oracle Solaris 10 8/11 설치 설명서: 사용자 정의 JumpStart 및 고급 설치의 9 장,
“JumpStart로 ZFS 루트 풀 설치”를 참조하십시오.

일반 ZFS 프로파일

```
pool rpool auto 2G 2G c1t1d0s0
```

도메인을 설치하기 위한 실제 ZFS 프로파일

```
pool rpool auto 2G 2G c0d0s0
```

I/O 도메인 설정

이 장에서는 I/O 도메인과 Logical Domains 환경에서 I/O 도메인을 구성하는 방법에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- 73 페이지 “I/O 도메인 개요”
- 74 페이지 “PCIe 버스 지정”
- 79 페이지 “PCIe 끝점 장치 지정”
- 90 페이지 “PCIe SR-IOV 가상 기능 사용”

I/O 도메인 개요

I/O 도메인은 물리적 I/O 장치를 직접 소유하며 물리적 I/O 장치에 대한 직접 액세스 권한을 가집니다. 도메인에 PCIe(PCI EXPRESS) 버스 또는 PCIe 끝점 장치를 지정하여 I/O 도메인을 만들 수 있습니다. 도메인에 버스 또는 장치를 지정하려면 `ldm add-io` 명령을 사용하십시오.

다음과 같은 이유로 I/O 도메인을 구성할 수 있습니다.

- I/O 도메인은 물리적 I/O 장치에 대한 직접 액세스 권한을 가지므로 가상 I/O와 연관된 성능 오버헤드가 발생하지 않습니다. 따라서 I/O 도메인의 I/O 성능이 베어 메탈 시스템의 I/O 성능과 거의 동일합니다.
- I/O 도메인은 다른 게스트 도메인에 사용될 가상 I/O 서비스를 호스트할 수 있습니다.

I/O 도메인 구성에 대한 자세한 내용은 다음을 참조하십시오.

- 74 페이지 “PCIe 버스 지정”
- 79 페이지 “PCIe 끝점 장치 지정”

주 - PCIe 끝점 장치로 구성된 I/O 도메인은 마이그레이션할 수 없습니다. 기타 마이그레이션 제한 사항에 대한 자세한 내용은 9 장, “도메인 마이그레이션”을 참조하십시오.

일반적인 I/O 도메인 만들기 지침

I/O 도메인은 PCIe 버스, NIU(네트워크 인터페이스 장치), PCIe 끝점 장치, PCIe SR-IOV(단일 루트 I/O 가상화) 가상 기능 등 하나 이상의 I/O 장치에 대한 직접 액세스 권한을 가질 수 있습니다.

I/O 장치에 대한 이 유형의 직접 액세스 권한은 보다 넓은 I/O 대역폭을 사용하여 다음을 가능하게 합니다.

- I/O 도메인의 응용 프로그램에 서비스 제공
- 게스트 도메인에 가상 I/O 서비스 제공

다음과 같은 기본적인 지침을 통해 I/O 대역폭을 효율적으로 활용할 수 있습니다.

- CPU 코어 단위에서 CPU 리소스를 지정하십시오. I/O 도메인 내 I/O 장치의 유형 및 수에 따라 하나 이상의 CPU 코어를 지정하십시오.
예를 들어, 1Gbps 이더넷 장치의 경우 10Gbps 이더넷 장치에 비해 보다 적은 CPU 코어로도 대역폭을 완전히 활용할 수 있습니다.
- 메모리 요구 사항을 준수하십시오. 메모리 요구 사항은 도메인에 지정된 I/O 장치의 유형에 따라 다릅니다. I/O 장치당 최소 4GB가 권장됩니다. 지정한 I/O 장치 수가 많을수록 메모리를 보다 많이 할당해야 합니다.
- PCIe SR-IOV 기능을 사용할 때는 다른 I/O 장치에 적용되는 것과 동일한 지침에 따라 각 SR-IOV 가상 기능을 사용하십시오. 따라서 가상 기능에서 사용 가능한 대역폭을 완전히 활용하려면 하나 이상의 CPU 코어 및 메모리(GB)를 지정하십시오.

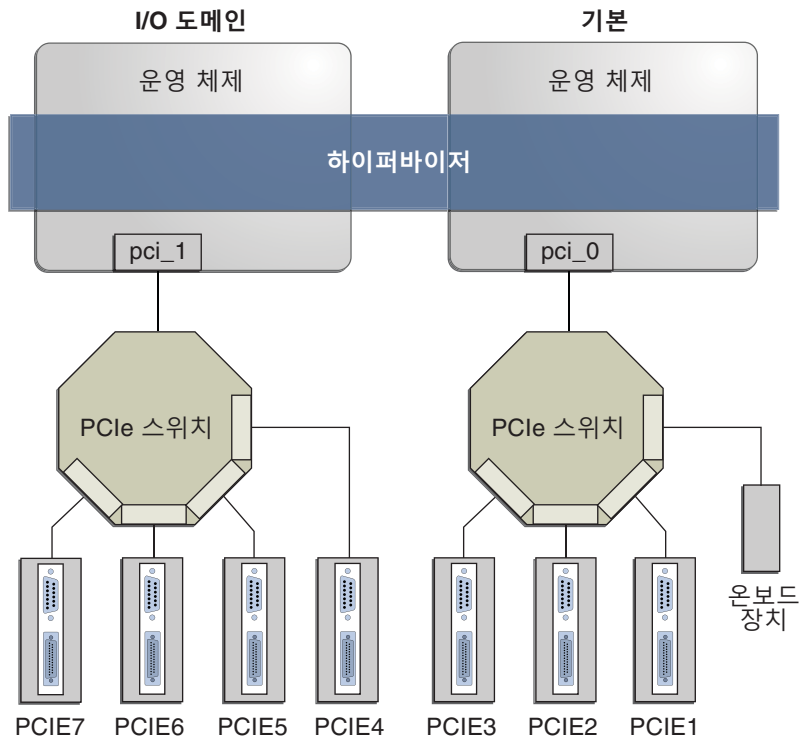
CPU 및 메모리 리소스가 충분하지 않은 많은 수의 가상 기능을 만들어 도메인에 지정하면 구성이 최적화되지 않을 수 있습니다.

PCIe 버스 지정

Oracle VM Server for SPARC 소프트웨어를 사용하여 도메인에 전체 PCIe 버스(루트 컴플렉스라고도 함)를 지정할 수 있습니다. 전체 PCIe 버스는 PCIe 버스 자체 및 관련된 모든 PCI 스위치와 장치로 구성됩니다. 서버에 있는 PCIe 버스는 pci@400(pci_0)과 같은 이름으로 식별됩니다. 전체 PCIe 버스로 구성된 I/O 도메인을 루트 도메인이라고도 합니다.

다음 다이어그램은 PCIe 버스가 2개(pci_0 및 pci_1)인 시스템을 보여줍니다. 각 버스는 다른 도메인에 지정되어 있습니다. 따라서 시스템은 2개의 I/O 도메인으로 구성되었습니다.

그림 6-1 I/O 도메인에 PCIe 버스 지정



PCIe 버스로 만들 수 있는 최대 I/O 도메인 수는 서버에서 사용 가능한 PCIe 버스 수에 따라 다릅니다. 예를 들어, Sun SPARC Enterprise T5440 서버를 사용 중인 경우 최대 4개의 I/O 도메인이 있을 수 있습니다.

주 - 일부 Sun UltraSPARC 서버에는 PCIe 버스가 하나만 있습니다. 이 경우 도메인에 PCIe 끝점(또는 직접 I/O 지정 가능) 장치를 지정하여 I/O 도메인을 만들 수 있습니다. 79 페이지 “PCIe 끝점 장치 지정”을 참조하십시오. 시스템에 NIU(네트워크 인터페이스 장치)가 있을 경우 도메인에 NIU를 지정하여 I/O 도메인을 만들 수도 있습니다.

I/O 도메인에 PCIe 버스를 지정하면 I/O 도메인이 해당 버스의 모든 장치를 소유합니다. 해당 버스의 PCIe 끝점 장치는 다른 도메인에 지정할 수 없습니다. primary 도메인에 지정된 PCIe 버스의 PCIe 끝점 장치만 다른 도메인에 지정할 수 있습니다.

서버가 처음에 Logical Domains 환경에서 구성되거나 factory-default 구성을 사용 중인 경우 primary 도메인은 모든 물리적 장치 리소스에 대한 액세스 권한을 가집니다. 즉, primary 도메인이 시스템에서 구성된 유일한 I/O 도메인이며 모든 PCIe 버스를 소유합니다.

▼ PCIe 버스를 지정하여 I/O 도메인을 만드는 방법

이 절차 예에서는 primary 도메인이 여러 버스를 소유한 초기 구성에서 새 I/O 도메인을 만드는 방법을 보여줍니다. 기본적으로 primary 도메인은 시스템에 있는 모든 버스를 소유합니다. 이 예는 Sun SPARC Enterprise T5440 서버에 해당합니다. 다른 서버에서도 이 절차를 사용할 수 있습니다. 다른 서버에 대한 지침은 이 예의 지침과 약간 다를 수 있지만 이 예를 통해 기본 원칙을 확인할 수 있습니다.

먼저 primary 도메인의 부트 디스크가 있는 버스를 보존해야 합니다. 그런 다음 primary 도메인에서 다른 버스를 제거하여 다른 도메인에 지정하십시오.



주의 - 지원되는 서버의 모든 내부 디스크를 단일 PCIe 버스에 연결할 수 있습니다. 도메인이 내부 디스크에서 부트되는 경우 도메인에서 해당 버스를 제거하지 마십시오. 또한 도메인이 사용하는 장치(예: 네트워크 포트)가 있는 버스를 제거하고 있지 않은지 확인하십시오. 잘못된 버스를 제거하면 도메인이 필요한 장치에 액세스하지 못할 수 있으며 도메인을 사용하지 못할 수 있습니다. 도메인이 사용하는 장치가 있는 버스를 제거하려면 다른 버스에서 장치를 사용하도록 해당 도메인을 재구성하십시오. 예를 들어, 다른 PCIe 슬롯에서 다른 내장 네트워크 포트 또는 PCIe 카드를 사용하도록 도메인을 재구성해야 할 수 있습니다.

이 예에서 primary 도메인은 ZFS 풀(rpool(c0t1d0s0)) 및 네트워크 인터페이스(nxge0)만 사용합니다. primary 도메인이 보다 많은 장치를 사용하는 경우 각 장치에 대해 2-4 단계를 반복하여 제거할 버스에 장치가 남아 있지 않도록 하십시오.

1 primary 도메인이 2개 이상의 PCIe 버스를 소유하는지 확인합니다.

```
primary# ldm list-io
```

NAME	TYPE	DOMAIN	STATUS
----	----	-----	-----
pci_0	BUS	primary	
pci_1	BUS	primary	
pci_2	BUS	primary	
pci_3	BUS	primary	
MB/PCIE0	PCIE	-	EMP
MB/PCIE1	PCIE	primary	OCC
MB/HBA	PCIE	primary	OCC
MB/PCIE4	PCIE	-	EMP
MB/PCIE5	PCIE	-	EMP
MB/XAUI1	PCIE	primary	OCC
MB/PCIE2	PCIE	primary	OCC
MB/PCIE3	PCIE	primary	OCC
MB/PCIE6	PCIE	primary	OCC
MB/PCIE7	PCIE	-	EMP

2 보존해야 할 부트 디스크의 장치 경로를 확인합니다.

- UFS 파일 시스템의 경우 **df /** 명령을 실행하여 부트 디스크의 장치 경로를 확인합니다.

```
primary# df /
/                               (/dev/dsk/c0t1d0s0 ): 1309384 blocks   457028 files
```

- ZFS 파일 시스템의 경우 먼저 **df /** 명령을 실행하여 풀 이름을 확인한 다음 **zpool status** 명령을 실행하여 부트 디스크의 장치 경로를 확인합니다.

```
primary# df /
/                               (rpool/ROOT/s10s_u8wos_08a):245176332 blocks 245176332 files
primary# zpool status rpool
zpool status rpool
  pool: rpool
  state: ONLINE
  scrub: none requested
  config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
c0t1d0s0	ONLINE	0	0	0

3 블록 장치가 연결된 물리적 장치를 확인합니다.

다음 예에서는 블록 장치 **c1t0d0s0**을 사용합니다.

```
primary# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx 1 root root          49 Oct  1 10:39 /dev/dsk/c0t1d0s0 ->
../../../../devices/pci@400/pci@0/pci@1/scsi@0/sd@1,0:a
```

이 예에서 **primary** 도메인의 부트 디스크에 대한 물리적 장치는 이전 **pci_0** 목록에 해당하는 버스 **pci@400**에 연결되어 있습니다. 따라서 다른 도메인에 **pci_0(pci@400)**을 지정할 수 없습니다.

4 시스템이 사용하는 네트워크 인터페이스를 확인합니다.

- Oracle Solaris 10. 다음 명령을 실행합니다.

```
primary# dladm show-dev
vsw0          link: up          speed: 1000 Mbps      duplex: full
nxge0         link: up          speed: 1000 Mbps      duplex: full
nxge1         link: unknown   speed: 0 Mbps         duplex: unknown
nxge2         link: unknown   speed: 0 Mbps         duplex: unknown
nxge3         link: unknown   speed: 0 Mbps         duplex: unknown
```

- Oracle Solaris 11. 다음 명령을 실행합니다.

```
primary# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX  DEVICE
net0          Ethernet      unknown  0      unknown vnet0
net1          Ethernet      up       1000   full    vsw0
net2          Ethernet      up       1000   full    nxge0
```

unknown 상태의 인터페이스는 구성되어 있지 않으므로 사용되지 않습니다. 이 예에서는 **nxge0** 인터페이스가 사용됩니다.

5 네트워크 인터페이스가 연결된 물리적 장치를 확인합니다.

다음 명령은 `nxge0` 네트워크 인터페이스를 사용합니다.

```
primary# ls -l /dev/nxge0
lrwxrwxrwx 1 root root          46 Oct 1 10:39 /dev/nxge0 ->
../devices/pci@500/pci@0/pci@c/network@0:nxge0
```

이 예에서 `primary` 도메인이 사용하는 네트워크 인터페이스에 대한 물리적 장치는 이전 `pci_1` 목록에 해당하는 버스 `pci@500`에 있습니다. 따라서 다른 2개의 버스 `pci_2(pci@600)` 및 `pci_3(pci@700)`은 `primary` 도메인이 사용하지 않으므로 다른 도메인에 지정해도 됩니다.

`primary` 도메인이 사용하는 네트워크 인터페이스가 다른 도메인에 지정할 버스에 있을 경우 다른 네트워크 인터페이스를 사용하도록 `primary` 도메인을 재구성해야 합니다.

6 부트 디스크 또는 네트워크 인터페이스를 포함하지 않는 버스를 `primary` 도메인에서 제거합니다.

이 예에서는 버스 `pci_2` 및 버스 `pci_3`이 `primary` 도메인에서 제거되고 있습니다. `primary` 도메인이 재구성 지연 모드로 전환되고 있음을 알리는 메시지가 `ldm` 명령에서 표시될 수 있습니다.

```
primary# ldm remove-io pci_2 primary
primary# ldm remove-io pci_3 primary
```

7 서비스 프로세서에 이 구성을 저장합니다.

이 예에서 구성은 `io-domain`입니다.

```
primary# ldm add-config io-domain
```

이 구성 `io-domain`은 재부트 후 사용할 다음 구성으로도 설정됩니다.

주 - 현재 UltraSPARC T2 및 UltraSPARC T2 Plus 시스템의 SP에서는 `factory-default` 구성을 포함하여 8개 이하의 구성을 저장할 수 있습니다. SPARC T3 및 SPARC T4 시스템에서는 구성을 저장하는 데 10MB의 공간을 사용할 수 있습니다. 저장되는 총 구성 수는 SP에 저장된 각 구성의 크기에 따라 다릅니다.

8 변경 사항이 적용되도록 `primary` 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

9 PCIe 버스를 추가할 도메인을 중지합니다.

다음 예에서는 `ldg1` 도메인을 중지합니다.

```
primary# ldm stop ldg1
```

10 직접 액세스 권한이 필요한 도메인에 사용 가능한 버스를 추가합니다.

사용 가능한 버스는 `pci_2`이며 도메인은 `ldg1`입니다.

```
primary# ldm add-io pci_2 ldg1
```

11 변경 사항이 적용되도록 도메인을 다시 시작합니다.

다음 명령은 ldg1 도메인을 다시 시작합니다.

```
primary# ldm start ldg1
```

12 primary 도메인에 여전히 올바른 버스가 지정되어 있으며 도메인 ldg1에 올바른 버스가 지정되었는지 확인합니다.

```
primary# ldm list-io
```

NAME	TYPE	DOMAIN	STATUS
-----	----	-----	-----
pci_0	BUS	primary	
pci_1	BUS	primary	
pci_2	BUS	ldg1	
pci_3	BUS		
MB/PCIE0	PCIE	-	EMP
MB/PCIE1	PCIE	primary	OCC
MB/HBA	PCIE	primary	OCC
MB/PCIE4	PCIE	-	EMP
MB/PCIE5	PCIE	-	EMP
MB/XAUI1	PCIE	primary	OCC
MB/PCIE2	PCIE	-	UNK
MB/PCIE3	PCIE	-	UNK
MB/PCIE6	PCIE	-	UNK
MB/PCIE7	PCIE	-	UNK

이 출력은 PCIe 버스 pci_0 및 pci_1과 이 아래의 장치도 도메인 primary에 지정되었으며 pci_2 및 관련 장치가 ldg1에 지정되었는지 확인합니다.

PCIe 끝점 장치 지정

Oracle VM Server for SPARC 2.0 릴리스 및 Oracle Solaris 10 9/10 OS부터 도메인에 개별 PCIe 끝점(또는 직접 I/O 지정 가능) 장치를 지정할 수 있습니다. 이와 같이 PCIe 끝점 장치를 사용하면 I/O 도메인에 대한 장치 지정 단위가 증가합니다. 이 기능은 DIO(직접 I/O) 기능을 통해 전달됩니다.

DIO 기능을 사용하여 시스템에 있는 PCIe 버스 수보다 많은 I/O 도메인을 만들 수 있습니다. 그러면 가능한 I/O 도메인 수가 PCIe 끝점 장치 수에 의해서만 제한됩니다.

PCIe 끝점 장치는 다음 중 하나일 수 있습니다.

- 슬롯의 PCIe 카드
- 플랫폼이 식별한 내장 PCIe 장치

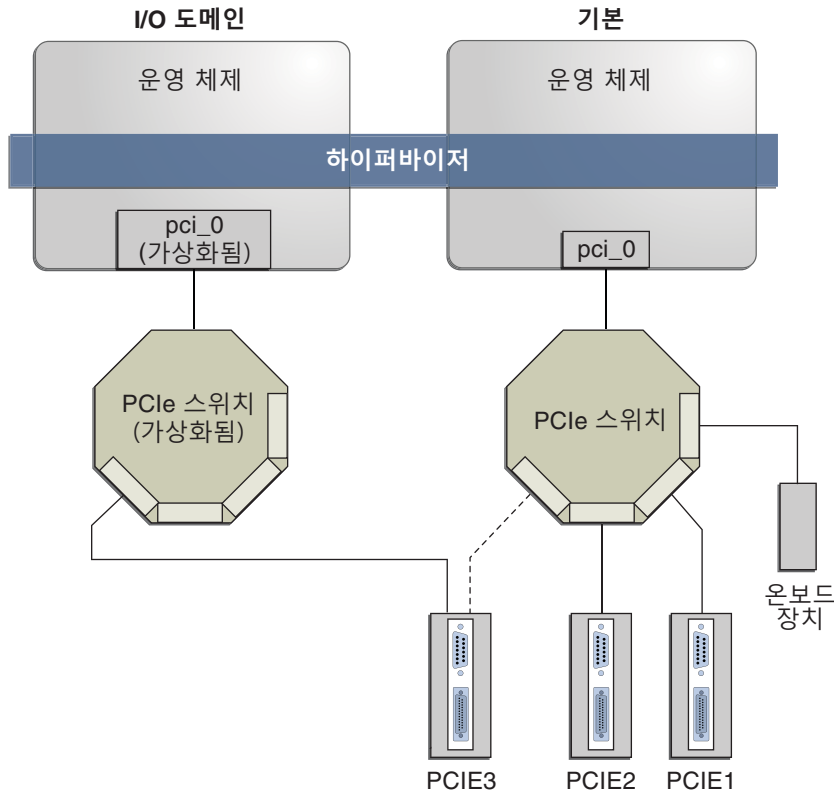
다음 다이어그램은 PCIe 끝점 장치 PCIE3이 I/O 도메인에 지정되었음을 보여줍니다. I/O 도메인의 버스 pci_0과 스위치는 모두 가상입니다. primary 도메인에서는 더 이상 PCIE3 끝점 장치에 액세스할 수 없습니다.

I/O 도메인에서 pci_0 블록과 스위치는 각각 가상 루트 컴플렉스와 가상 PCIe 스위치입니다. 이 블록과 스위치는 primary 도메인의 pci_0 블록 및 스위치와 매우 유사합니다. primary 도메인에서 슬롯 PCIE3의 장치는 원래 장치의 새도우 형식이며 SUNW,assigned로 식별됩니다.



주의 - `ldm rm-io` 명령을 사용하여 `primary` 도메인에서 해당 장치를 제거한 후에는 Oracle Solaris 핫 플러그 작업을 통해 PCIe 끝점 장치를 “제거”할 수 없습니다. PCIe 끝점 장치 교체 또는 제거에 대한 자세한 내용은 84 페이지 “PCIe 하드웨어 변경”을 참조하십시오.

그림 6-2 I/O 도메인에 PCIe 끝점 장치 지정



PCIe 끝점 장치를 나열하려면 `ldm list-io` 명령을 사용하십시오.

DIO 기능이 슬롯의 PCIe 카드를 I/O 도메인에 지정할 수 있도록 허용하기는 하지만 특정 PCIe 카드만 지원됩니다. [Oracle VM Server for SPARC 2.2 릴리스 노트의 “직접 I/O 하드웨어 및 소프트웨어 요구 사항”](#)을 참조하십시오.



주의 - 브릿지가 있는 PCIe 카드는 지원되지 않습니다. PCIe 기능 레벨 지정도 지원되지 않습니다. I/O 도메인에 지원되지 않는 PCIe 카드를 지정하면 예상하지 못한 동작이 발생할 수 있습니다.

DIO 기능에 대한 몇 가지 중요한 세부 정보는 다음과 같습니다.

- 이 기능은 모든 소프트웨어 요구 사항이 충족되는 경우에만 사용으로 설정됩니다. **Oracle VM Server for SPARC 2.2 릴리스 노트의 “직접 I/O 하드웨어 및 소프트웨어 요구 사항”**를 참조하십시오.
- primary 도메인에 지정된 PCIe 버스와 연결되어 있는 PCIe 끝점만 DIO 기능을 사용하는 다른 도메인에 지정할 수 있습니다.
- DIO를 사용하는 I/O 도메인은 primary 도메인이 실행 중인 경우에만 PCIe 끝점 장치에 대한 액세스 권한을 가집니다.
- primary 도메인을 재부트하면 PCIe 끝점 장치가 있는 I/O 도메인이 영향을 받습니다. **83 페이지 “primary 도메인 재부트”**를 참조하십시오. primary 도메인은 다음 권한도 가집니다.
 - PCIe 버스를 초기화하고 버스를 관리합니다.
 - I/O 도메인에 지정된 PCIe 끝점 장치로 트리거된 모든 버스 오류를 처리합니다. primary 도메인만 모든 PCIe 버스 관련 오류를 수신합니다.

직접 I/O 하드웨어 및 소프트웨어 요구 사항

DIO 기능을 성공적으로 사용하려면 적절한 소프트웨어를 실행 중이어야 하며 DIO 기능이 지원하는 PCIe 카드만 I/O 도메인에 지정해야 합니다. 하드웨어 및 소프트웨어 요구 사항은 **Oracle VM Server for SPARC 2.2 릴리스 노트의 “직접 I/O 하드웨어 및 소프트웨어 요구 사항”**를 참조하십시오.

주 - 플랫폼에서 지원되는 모든 PCIe 카드는 primary 도메인에서 지원됩니다. 지원되는 PCIe 카드 목록은 사용 중인 플랫폼에 대한 설명서를 참조하십시오. 단, 직접 I/O가 지원하는 PCIe 카드만 I/O 도메인에 지정할 수 있습니다.

현재 직접 I/O 기능 제한 사항

다음 제한 사항을 해결하는 방법은 **82 페이지 “PCIe 끝점 장치 구성 계획”**을 참조하십시오.

- PCIe 끝점 장치를 primary 도메인에 지정하거나 이 도메인에서 제거할 때 재구성 지연이 시작됩니다. 즉, primary 도메인이 재부트된 후에만 변경 사항이 적용됩니다.

primary 도메인을 재부트하면 직접 I/O가 영향을 받으므로 primary 도메인에 대한 직접 I/O 관련 변경 사항 적용을 최대화하고 primary 도메인 재부트를 최소화하려면 신중하게 직접 I/O 구성 변경을 계획해야 합니다.

- PCIe 끝점 장치를 다른 도메인에 지정하거나 다른 도메인에서 제거하는 작업은 해당 도메인이 중지되거나 비활성 상태인 경우에만 허용됩니다.

PCIe 끝점 장치 구성 계획

primary 도메인 작동 중지 시간이 발생하지 않도록 PCIe 끝점 장치를 지정하거나 제거할 때는 사전에 신중하게 계획하십시오. primary 도메인 재부트는 primary 도메인 자체에서 사용 가능한 서비스에만 영향을 끼치는 것이 아니라 PCIe 끝점 장치가 지정된 I/O 도메인에도 영향을 끼칩니다. 각 I/O 도메인에 대한 변경 사항이 다른 도메인에 영향을 끼치지는 않지만 사전 계획을 통해 해당 도메인이 제공하는 서비스에 끼치는 영향을 최소화할 수 있습니다.

장치를 처음 지정하거나 제거할 때 재구성 지연이 시작됩니다. 따라서 더 많은 장치를 계속 추가하거나 제거한 다음 primary 도메인을 한 번만 재부트하여 모든 변경 사항이 적용되도록 할 수 있습니다.

예는 85 페이지 “PCIe 끝점 장치를 지정하여 I/O 도메인을 만드는 방법”을 참조하십시오.

다음에서는 DIO 장치 구성을 계획하고 수행하기 위해 따라야 할 일반적인 단계에 대해 설명합니다.

1. 시스템 하드웨어 구성을 이해하고 기록합니다.

특히, 시스템에 있는 PCIe 카드의 부품 번호 및 기타 세부 정보에 대한 내용을 기록합니다.

`ldm list-io -l` 및 `prtdiag -v` 명령을 사용하여 정보를 얻고 나중에 사용할 수 있도록 저장합니다.

2. primary 도메인에 있어야 할 PCIe 끝점 장치를 확인합니다.

예를 들어, 다음에 대한 액세스를 제공하는 PCIe 끝점 장치를 확인합니다.

- 부트 디스크 장치
- 네트워크 장치
- primary 도메인이 서비스로 제공하는 기타 장치

3. I/O 도메인에서 사용할 수 있는 모든 PCIe 끝점 장치를 제거합니다.

이 단계를 수행하면 primary 도메인에서 I/O 도메인에 영향을 끼치는 후속 재부트 작업을 수행하지 않아도 됩니다.

`ldm rm-io` 명령을 사용하여 PCIe 끝점 장치를 제거합니다. `rm-io` 및 `add-io` 하위 명령에 장치를 지정할 때는 장치 경로보다 익명을 사용하는 것이 좋습니다.

주 - 처음 PCIe 끝점 장치를 제거할 때 재구성 지연이 시작될 수 있기는 하지만 장치를 계속 제거할 수 있습니다. 원하는 모든 장치를 제거한 후 **primary** 도메인을 한 번 재부트하면 모든 변경 사항이 적용됩니다.

4. SP(서비스 프로세서)에 이 구성을 저장합니다.

`ldm add-config` 명령을 사용합니다.

5. 3단계에서 제거한 PCIe 끝점 장치가 해제되도록 **primary** 도메인을 재부트합니다.

6. 제거한 PCIe 끝점 장치가 더 이상 **primary** 도메인에 지정되어 있지 않은지 확인합니다.

`ldm list-io -l` 명령을 사용하여 제거한 장치가 출력에서 **SUNW,assigned-device**로 표시되는지 확인합니다.

7. 물리적 장치에 대한 직접 액세스 권한을 제공하도록 게스트 도메인에 사용 가능한 PCIe 끝점 장치를 지정합니다.

이 지정이 완료되면 도메인 마이그레이션 기능을 통해 더 이상 다른 물리적 시스템에 게스트 도메인을 마이그레이션할 수 없습니다.

8. PCIe 끝점 장치를 게스트 도메인에 추가하거나 게스트 도메인에서 제거합니다.

`ldm add-io` 명령을 사용합니다.

재부트 작업을 줄이고 해당 도메인이 제공하는 서비스의 작동 중지 시간이 발생하지 않도록 하여 I/O 도메인에 대한 변경을 최소화합니다.

9. (선택 사항) PCIe 하드웨어를 변경합니다.

84 페이지 “PCIe 하드웨어 변경”을 참조하십시오.

primary 도메인 재부트

primary 도메인은 PCIe 버스의 소유자이며 버스 초기화 및 관리를 담당합니다. **primary** 도메인은 활성 상태여야 하며 DIO 기능을 지원하는 Oracle Solaris OS 버전을 실행 중이어야 합니다. **primary** 도메인을 종료, 정지 또는 재부트하면 PCIe 버스에 대한 액세스가 중단됩니다. PCIe 버스를 사용할 수 없는 경우 해당 버스의 PCIe 장치가 영향을 받아 이러한 장치를 사용하지 못할 수 있습니다.

PCIe 끝점 장치가 있는 I/O 도메인이 실행 중인 동안 **primary** 도메인이 재부트되면 해당 I/O 도메인의 동작을 예측할 수 없습니다. 예를 들어, 재부트 도중 또는 재부트 후 PCIe 끝점 장치가 있는 I/O 도메인에 패닉이 발생할 수 있습니다. **primary** 도메인을 재부트할 때 수동으로 각 도메인을 중지하고 시작해야 합니다.

이러한 문제를 해결하려면 다음 단계 중 하나를 수행하십시오.

- **primary** 도메인을 종료하기 전에 PCIe 끝점 장치가 지정된 시스템에서 수동으로 도메인을 종료합니다.

이 단계를 수행하면 **primary** 도메인을 종료, 정지 또는 재부트하기 전에 이러한 도메인이 완전히 종료됩니다.

PCIe 끝점 장치가 지정된 도메인을 모두 찾으려면 `ldm list-io` 명령을 실행합니다. 이 명령을 사용하면 시스템의 도메인에 지정된 PCIe 끝점 장치를 나열할 수 있습니다. 따라서 이 정보를 기반으로 계획을 세울 수 있습니다. 이 명령 출력에 대한 자세한 설명은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

발견된 각 도메인에 대해 `ldm stop` 명령을 실행하여 도메인을 중지합니다.

- **primary** 도메인과 PCIe 끝점 장치가 지정된 도메인 간의 도메인 종속성 관계를 구성합니다.

이 종속성 관계를 사용하면 어떤 이유로든 **primary** 도메인이 재부트될 때 PCIe 끝점 장치가 있는 도메인이 자동으로 다시 시작됩니다.

이 종속성 관계는 해당 도메인을 강제로 재설정하며 완전히 종료할 수 없습니다. 단, 종속성 관계는 수동으로 종료된 도메인에는 영향을 끼치지 않습니다.

```
# ldm set-domain failure-policy=reset primary
# ldm set-domain master=primary ldom
```

PCIe 하드웨어 변경

다음 단계는 PCIe 끝점 지정을 잘못 구성하지 않도록 하는 데 유용합니다. 특정 하드웨어 설치 및 제거에 대한 플랫폼 관련 정보는 사용 중인 플랫폼에 대한 설명서를 참조하십시오.

- 빈 슬롯에 PCIe 카드를 설치 중인 경우 필요한 작업이 없습니다. 이 PCIe 카드는 PCIe 버스를 소유한 도메인이 자동으로 소유합니다.

I/O 도메인에 새 PCIe 카드를 지정하려면 `ldm rm-io` 명령을 사용하여 먼저 **primary** 도메인에서 카드를 제거하십시오. 그런 다음 `ldm add-io` 명령을 사용하여 I/O 도메인에 카드를 지정하십시오.

- PCIe 카드가 시스템에서 제거되어 **primary** 도메인에 지정된 경우 필요한 작업이 없습니다.
- I/O 도메인에 지정된 PCIe 카드를 제거하려면 먼저 I/O 도메인에서 장치를 제거하십시오. 그런 다음 시스템에서 물리적으로 장치를 제거하기 전에 **primary** 도메인에 장치를 추가하십시오.

- I/O 도메인에 지정된 PCIe 카드를 교체하려면 DIO 기능이 새 카드를 지원하는지 확인하십시오.

지원하는 경우 현재 I/O 도메인에 자동으로 새 카드가 지정되므로 필요한 작업이 없습니다.

지원하지 않는 경우 `ldm rm-io` 명령을 사용하여 먼저 I/O 도메인에서 해당 PCIe 카드를 제거하십시오. 다음으로 `ldm add-io` 명령을 사용하여 `primary` 도메인에 해당 PCIe 카드를 재지정하십시오. 그런 다음 `primary` 도메인에 지정한 PCIe 카드를 물리적으로 다른 PCIe 카드로 교체하십시오. 이러한 단계를 수행하면 DIO 기능이 지원하지 않는 구성을 피할 수 있습니다.

▼ PCIe 끝점 장치를 지정하여 I/O 도메인을 만드는 방법

작동 중지 시간이 최소화되도록 사전에 모든 DIO 배포를 계획하십시오.



주의 - DIO 도메인에 SPARC T3-1 또는 SPARC T4-1 시스템의 `/SYS/MB/SASHBA1` 슬롯을 지정할 경우 `primary` 도메인이 내장 DVD 장치에 대한 액세스 권한을 손실합니다.

SPARC T3-1 및 SPARC T4-1 시스템에는 `/SYS/MB/SASHBA0` 및 `/SYS/MB/SASHBA1` 경로로 표시되는 내장 저장소용 DIO 슬롯이 2개 있습니다. 멀티헤드 내장 디스크를 호스트하는 것 외에 `/SYS/MB/SASHBA1` 슬롯은 내장 DVD 장치를 호스트합니다. 따라서 DIO 도메인에 `/SYS/MB/SASHBA1`을 지정할 경우 `primary` 도메인이 내장 DVD 장치에 대한 액세스 권한을 손실합니다.

SPARC T3-2 및 SPARC T4-2 시스템에는 모든 내장 디스크와 내장 DVD 장치를 호스트하는 단일 `SASHBA` 슬롯이 있습니다. 따라서 DIO 도메인에 `SASHBA`를 지정할 경우 내장 디스크 및 내장 DVD 장치가 DIO 도메인에 대여되어 `primary` 도메인에서 사용할 수 없게 됩니다.

PCIe 끝점 장치를 추가하여 I/O 도메인을 만드는 예는 [82 페이지 “PCIe 끝점 장치 구성 계획”](#)을 참조하십시오.

주 - 이 릴리스에서는 `DefaultFixed NCP`를 사용하여 Oracle Solaris 11 시스템에 데이터 링크와 네트워크 인터페이스를 구성하는 것이 가장 좋습니다.

Oracle Solaris 11 OS에 포함된 NCP는 다음과 같습니다.

- `DefaultFixed.dladm` 또는 `ipadm` 명령을 사용하여 네트워킹을 관리할 수 있습니다.
- `Automatic.netcfg` 또는 `netadm` 명령을 사용하여 네트워킹을 관리할 수 있습니다.

`netadm list` 명령을 사용하여 `DefaultFixed NCP`가 사용으로 설정되어 있는지 확인하십시오. [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 7 장, “프로파일에 데이터 링크 및 인터페이스 구성 명령 사용”](#)을 참조하십시오.

1 현재 시스템에 설치된 장치를 식별하고 아카이브합니다.

`ldm list-io -l` 명령의 출력은 현재 I/O 장치가 구성된 방법을 보여줍니다. `prtdiag -v` 명령을 사용하여 보다 자세한 정보를 얻을 수 있습니다.

주 - I/O 도메인에 장치가 지정된 후에는 I/O 도메인에서만 장치 ID를 확인할 수 있습니다.

```
# ldm list-io -l
```

NAME	TYPE	DOMAIN	STATUS
pci_0	BUS	primary	
[pci@400]			
pci_1	BUS	primary	
[pci@500]			
PCIE1	PCIE	-	EMP
[pci@400/pci@0/pci@c]			
PCIE2	PCIE	primary	OCC
[pci@400/pci@0/pci@9]			
network@0			
network@0,1			
network@0,2			
network@0,3			
PCIE3	PCIE	primary	OCC
[pci@400/pci@0/pci@d]			
SUNW,emlxs/fp/disk			
SUNW,emlxs@0,1/fp/disk			
SUNW,emlxs@0,1/fp@0,0			
MB/SASHBA	PCIE	primary	OCC
[pci@400/pci@0/pci@8]			
scsi@0/tape			
scsi@0/disk			
scsi@0/sd@0,0			
scsi@0/sd@1,0			
PCIE0	PCIE	-	EMP
[pci@500/pci@0/pci@9]			
PCIE4	PCIE	primary	OCC
[pci@500/pci@0/pci@d]			
network@0			
network@0,1			
PCIE5	PCIE	primary	OCC
[pci@500/pci@0/pci@c]			
SUNW,qlc@0/fp/disk			
SUNW,qlc@0/fp@0,0			
SUNW,qlc@0,1/fp/disk			
SUNW,qlc@0,1/fp@0,0			
MB/NET0	PCIE	primary	OCC
[pci@500/pci@0/pci@8]			
network@0			
network@0,1			
network@0,2			
network@0,3			

2 보존해야 할 부트 디스크의 장치 경로를 확인합니다.

- UFS 파일 시스템의 경우 **df /** 명령을 실행하여 부트 디스크의 장치 경로를 확인합니다.

```
primary# df /
/ (/dev/dsk/c0t1d0s0 ): 1309384 blocks 457028 files
```

- ZFS 파일 시스템의 경우 먼저 **df** / 명령을 실행하여 풀 이름을 확인한 다음 **zpool status** 명령을 실행하여 부트 디스크의 장치 경로를 확인합니다.

```
primary# df /
/
(rpool/ROOT/s10s_u8wos_08a):245176332 blocks 245176332 files
primary# zpool status rpool
zpool status rpool
pool: rpool
state: ONLINE
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
c0t1d0s0	ONLINE	0	0	0

3 블록 장치가 연결된 물리적 장치를 확인합니다.

다음 예에서는 블록 장치 **c0t1d0s0**을 사용합니다.

```
primary# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx 1 root root 49 Jul 20 22:17 /dev/dsk/c0t1d0s0 ->
../devices/pci@400/pci@0/pci@8/scsi@0/sd@0,0:a
```

이 예에서 **primary** 도메인의 부트 디스크에 대한 물리적 장치는 1단계의 MB/SASHBA 목록에 해당하는 PCIe 끝점 장치(**pci@400/pci@0/pci@8**)에 연결되어 있습니다. 이 장치를 제거하면 **primary** 도메인이 부트되지 않으므로 **primary** 도메인에서 이 장치를 제거하지 마십시오.

4 시스템이 사용하는 네트워크 인터페이스를 확인합니다.

- Oracle Solaris 10 OS. **ifconfig** 명령을 사용하여 **primary** 도메인이 사용하는 네트워크 인터페이스(이 예의 경우 **nxge0**)를 확인합니다.

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
nxge0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu 1500 index 2
    inet 10.6.212.149 netmask fffffe00 broadcast 10.6.213.255
    ether 0:21:28:4:27:cc
```

- Oracle Solaris 11 OS. **ipadm** 명령을 사용하여 **primary** 도메인이 사용하는 네트워크 인터페이스(이 예의 경우 **net0**)를 확인합니다.

```
# ipadm show-addr
ADDROBJ    TYPE      STATE     ADDR
lo0/v4     static    ok        127.0.0.1/8
net0/ipv4   static    ok        10.6.212.149/23
```

5 네트워크 인터페이스가 연결된 물리적 장치를 확인합니다.

다음 명령은 **nxge0** 네트워크 인터페이스를 사용합니다.

```
primary# ls -l /dev/nxge0
lrwxrwxrwx 1 root root 46 Jul 30 17:29 /dev/nxge0 ->
../devices/pci@500/pci@0/pci@8/network@0:nxge0
```

이 예에서 **primary** 도메인이 사용하는 네트워크 인터페이스에 대한 물리적 장치는 1단계의 **MB/NET0** 목록에 해당하는 PCIe 끝점 장치(**pci@500/pci@0/pci@8**)에 연결되어 있습니다. 따라서 **primary** 도메인에서 이 장치를 제거하지 **마십시오**. **primary** 도메인이 사용하지 않는 기타 모든 PCIe 장치는 다른 도메인에 지정해도 됩니다.

primary 도메인이 사용하는 네트워크 인터페이스가 다른 도메인에 지정할 버스에 있을 경우 다른 네트워크 인터페이스를 사용하도록 **primary** 도메인을 재구성해야 합니다.

6 I/O 도메인에서 사용할 수 있는 PCIe 끝점 장치를 제거합니다.

이 예에서는 **primary** 도메인이 사용하고 있지 않는 **PCIE2**, **PCIE3**, **PCIE4** 및 **PCIE5** 끝점 장치를 제거할 수 있습니다.

a. PCIe 끝점 장치를 제거합니다.



주의 - **primary** 도메인이 사용하거나 필요로 하는 장치는 제거하지 **마십시오**.

실수로 잘못된 장치를 제거할 경우 **ldm cancel-reconf primary** 명령을 사용하여 **primary** 도메인에서 재구성 지연을 취소하십시오.

여러 번 재부트하지 않아도 되도록 다중 장치를 한 번에 제거할 수 있습니다.

```
# ldm rm-io PCIE2 primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
# ldm rm-io PCIE3 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm rm-io PCIE4 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm rm-io PCIE5 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

b. SP(서비스 프로세서)에 새 구성을 저장합니다.

다음 명령은 **dio**라는 파일에 구성을 저장합니다.

```
# ldm add-config dio
```

c. PCIe 끝점 장치 제거가 반영되도록 시스템을 재부트합니다.

```
# reboot -- -r
```

- 7 **primary** 도메인에 로그인하고 PCIe 끝점 장치가 더 이상 도메인에 지정되어 있지 않은지 확인합니다.

```
# ldm list-io
```

NAME	TYPE	DOMAIN	STATUS
pci_0	BUS	primary	
pci_1	BUS	primary	
PCIE1	PCIE	-	EMP
PCIE2	PCIE		OCC
PCIE3	PCIE		OCC
MB/SASHBA	PCIE	primary	OCC
PCIE0	PCIE	-	EMP
PCIE4	PCIE		OCC
PCIE5	PCIE		OCC
MB/NET0	PCIE	primary	OCC

주 - ldm list-io -l 출력에는 제거된 PCIe 끝점 장치가 SUNW,assigned-device로 표시될 수 있습니다. primary 도메인에서 더 이상 실제 정보를 사용할 수 없지만 장치가 지정된 도메인에는 이 정보가 들어 있습니다.

- 8 도메인에 PCIe 끝점 장치를 지정합니다.

- a. ldg1 도메인에 PCIE2 장치를 추가합니다.

```
# ldm add-io PCIE2 ldg1
```

- b. 바인딩 후 ldg1 도메인을 시작합니다.

```
# ldm bind ldg1
# ldm start ldg1
LDom ldg1 started
```

- 9 ldg1 도메인에 로그인하고 장치를 사용할 수 있는지 확인합니다.

네트워크 장치를 사용할 수 있는지 확인한 다음 도메인에서 사용할 네트워크 장치를 구성합니다.

- Oracle Solaris 10. 다음 명령을 실행합니다.

```
# dladm show-dev
```

vnet0	link: up	speed: 0	Mbps	duplex: unknown
nxge0	link: unknown	speed: 0	Mbps	duplex: unknown
nxge1	link: unknown	speed: 0	Mbps	duplex: unknown
nxge2	link: unknown	speed: 0	Mbps	duplex: unknown
nxge3	link: unknown	speed: 0	Mbps	duplex: unknown

- Oracle Solaris 11. 다음 명령을 실행합니다.

```
# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	0	unknown	vnet0
net1	Ethernet	unknown	0	unknown	vsw0
net2	Ethernet	unknown	0	unknown	nxge0

PCIe SR-IOV 가상 기능 사용

Oracle VM Server for SPARC 2.2 릴리스부터 SPARC T3 및 SPARC T4 플랫폼에서 PCIe(Peripheral Component Interconnect Express) SR-IOV(단일 루트 I/O 가상화) 기능이 지원됩니다.

SR-IOV 개요

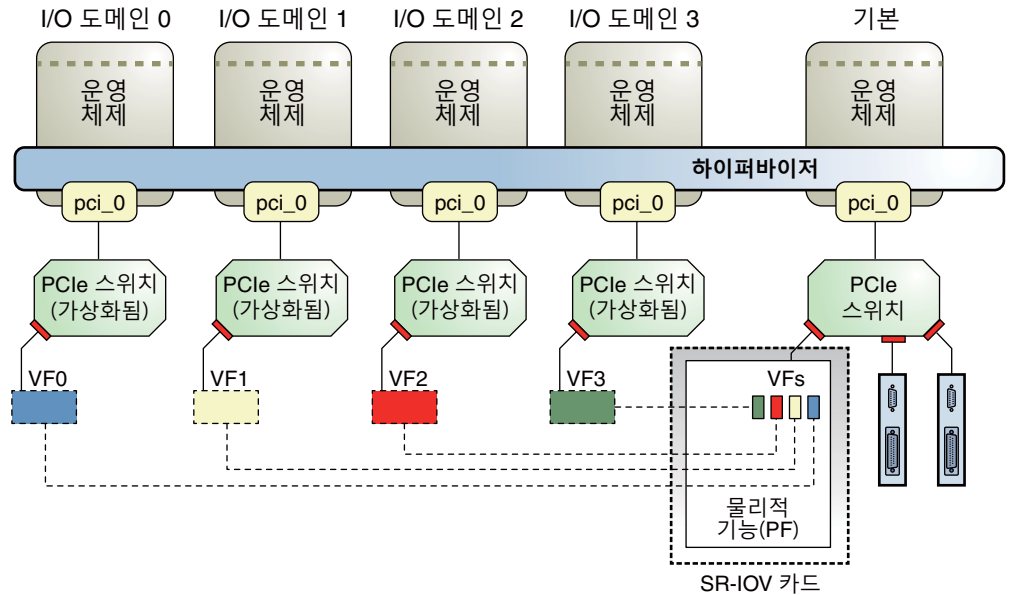
SR-IOV 구현은 PCI-SIG가 정의한 표준의 버전 1.1을 기반으로 합니다. SR-IOV 표준은 가상 시스템에서 효율적으로 PCIe 장치를 공유할 수 있도록 하며 고유 성능과 유사한 I/O 성능이 발휘되도록 하드웨어에서 구현됩니다. SR-IOV 사양은 만들어진 새 장치를 통해 가상 시스템을 I/O 장치에 직접 연결할 수 있다는 점에서 새로운 표준을 정의합니다.

물리적 기능이라는 단일 I/O 리소스는 여러 가상 시스템에서 공유될 수 있습니다. 공유된 장치는 전용 리소스를 제공하며 공유된 공통 리소스를 사용하기도 합니다. 따라서 각 가상 시스템은 고유한 리소스에 대한 액세스 권한을 가집니다. 이로 인해 적절한 하드웨어 및 OS 지원을 통해 SR-IOV가 사용으로 설정된 이더넷 포트 등의 PCIe 장치는 각각 고유의 PCIe 구성 공간을 가진 별도의 다중 물리적 장치로 표시될 수 있습니다.

SR-IOV에 대한 자세한 내용은 **PCI-SIG 웹 사이트** (<http://www.pcisig.com/>)를 참조하십시오.

다음 그림에서는 I/O 도메인에서 가상 기능과 물리적 기능 간의 관계를 보여줍니다.

그림 6-3 I/O 도메인에서 가상 기능과 물리적 기능 사용



SR-IOV의 기능 유형은 다음과 같습니다.

- **물리적 기능** - SR-IOV 사양에 정의된 SR-IOV 기능을 지원하는 PCI 기능입니다. 물리적 기능은 SR-IOV 기능 구조를 포함하며 SR-IOV 기능을 관리합니다. 물리 함수는 다른 PCIe 장치처럼 검색, 관리 및 조작할 수 있는 완전형 PCIe 함수입니다. 물리적 기능을 사용하여 PCIe 장치를 구성 및 제어할 수 있습니다.
- **가상 기능** - 물리적 기능과 연관된 PCI 기능입니다. 가상 기능은 하나 이상의 물리적 리소스를 물리적 기능 및 해당 물리적 기능과 연관된 가상 기능과 공유하는 경량형 PCIe 기능입니다. 물리적 기능과 달리 가상 기능은 고유의 동작만 구성할 수 있습니다.

각 SR-IOV 장치는 물리적 기능을 사용할 수 있으며 각 물리적 기능에는 최대 64,000개의 가상 기능이 연관될 수 있습니다. 이 수치는 SR-IOV 장치마다 다릅니다. 가상 기능은 물리적 기능에 의해 만들어집니다.

SR-IOV가 물리적 기능에서 사용으로 설정되면 물리적 기능의 버스, 장치 및 기능 번호가 각 가상 기능의 PCI 구성 공간에 액세스할 수 있습니다. 각 가상 기능에는 레지스터 세트 매핑에 사용되는 PCI 메모리 공간이 있습니다. 가상 기능 장치 드라이버는 기능이 사용으로 설정되도록 레지스터 세트에서 작동하며 가상 기능은 실제 PCI 장치로 표시됩니다. 만들어진 가상 기능은 I/O 도메인에 직접 지정할 수 있습니다. 이를 통해 가상 기능이 물리적 장치를 공유하고 CPU 및 하이퍼바이저 소프트웨어 오버헤드 없이 I/O를 수행할 수 있습니다.

SR-IOV 기능이 있는 장치를 사용할 경우 다음과 같은 이점이 제공됩니다.

- **성능 향상 및 대기 시간 감소** - 가상 시스템 환경에서 하드웨어에 직접 액세스합니다.
- **비용 절감** - 다음과 같이 자본 및 운영비가 절감됩니다.
 - 절전
 - 어댑터 수 감소
 - 케이블 연결 감소
 - 스위치 포트 감소

SR-IOV 하드웨어 및 소프트웨어 요구 사항

Oracle VM Server for SPARC 2.2 릴리스부터 SPARC T3 및 SPARC T4 플랫폼에서 PCIe SR-IOV 기능이 지원됩니다. 필요한 하드웨어, 소프트웨어 및 펌웨어 버전에 대한 자세한 내용은 [Oracle VM Server for SPARC 2.2 릴리스 노트의 “PCIe SR-IOV 하드웨어 및 소프트웨어 요구 사항”](#)을 참조하십시오.

현재 SR-IOV 기능 제한 사항

주 - 재부트를 최소화하려면 여러 작업을 동일한 재구성 지연 내에서 수행하십시오.

이 릴리스에서 SR-IOV 기능의 제한 사항은 다음과 같습니다.

- 가상 기능이 하나 이상 지정된 도메인에 대해 마이그레이션이 사용 안함으로 설정됩니다.
- 가상 기능을 만들고 삭제하면 재구성 지연이 시작됩니다.
- 활성 상태의 도메인에 가상 기능을 지정할 수 없습니다. **primary** 도메인에 가상 기능을 지정할 경우 재구성 지연이 시작됩니다.
- 물리적 기능에 대해 마지막으로 만들어진 가상 기능만 삭제할 수 있습니다. 따라서 3개의 가상 기능을 만들 경우 삭제할 수 있는 첫번째 가상 기능은 세번째 가상 기능입니다.
- 이더넷 SR-IOV 카드만 지원됩니다.
- **primary** 도메인에 설치된 SR-IOV 카드에 대해서만 SR-IOV 기능이 사용으로 설정됩니다. PCIe 버스 지정 또는 DIO(직접 I/O) 기능을 사용하여 도메인에 SR-IOV 카드를 지정할 경우 해당 카드에 대해 SR-IOV 기능이 사용으로 설정되지 않습니다.
- **pvid** 또는 **vid** 등록 정보를 설정하여 가상 기능의 VLAN 구성을 사용으로 설정할 수 있습니다. 이러한 가상 기능 등록 정보는 동시에 설정할 수 없습니다.

PCIe SR-IOV 가상 기능 사용 계획

가상 기능을 만들고 삭제하기 전에 사전 계획을 통해 구성에서 사용할 가상 기능을 확인하십시오. 가상 기능을 만들고 삭제할 때는 **primary** 도메인을 재부트해야 합니다. 해당 재부트는 PCIe 끝점 또는 SR-IOV 가상 기능이 구성된 모든 I/O 도메인에 부정적인 영향을 끼칩니다. 따라서 **primary** 도메인 재부트 횟수를 최소화해야 합니다. 현재 및 향후 구성 요구 사항이 충족되도록 다양한 SR-IOV 장치에서 필요한 가상 기능 수를 확인하십시오.

I/O 도메인에 대한 자세한 내용은 [74 페이지 “일반적인 I/O 도메인 만들기 지침”](#)을 참조하십시오.

SR-IOV 가상 기능 구성 및 지정을 계획하고 수행하려면 다음과 같은 일반 단계를 사용하십시오.

1. 시스템에서 사용 가능하며 요구 사항에 가장 적합한 PCIe SR-IOV 물리적 기능을 확인합니다.

다음 명령을 사용하여 필요한 정보를 식별합니다.

<code>ldm list-io</code>	사용 가능한 SR-IOV 물리적 기능 장치를 식별합니다.
<code>prtdiag -v</code>	사용 가능한 PCIe SR-IOV 카드 및 내장 장치를 식별합니다.
<code>ldm list-io -l pf-name</code>	지정된 물리적 기능에 대한 추가 정보(예: 장치에서 지원하는 최대 가상 기능 수)를 식별합니다.
<code>ldm list-io -d pf-name</code>	장치에서 지원하는 장치 관련 등록 정보를 식별합니다. 102 페이지 “고급 SR-IOV 항목” 을 참조하십시오.

2. 지정된 SR-IOV 물리적 기능에 필요한 수의 가상 기능을 만듭니다.

다음 명령을 사용하여 가상 기능을 만듭니다.

`primary# ldm create-vf pf-name`

`ldm create-vf` 명령을 사용하여 가상 기능의 장치 관련 및 네트워크 관련 등록 정보를 설정합니다. `unicast-slots`는 장치 관련 등록 정보입니다. `mac-addr`, `alt-mac-addr`s, `mtu`, `pvid` 및 `vid`는 네트워크 관련 등록 정보입니다.

`mac-addr`, `alt-mac-addr`s 및 `mtu` 네트워크 관련 등록 정보는 다음과 같이 변경할 수 있습니다.

- **primary 도메인에 가상 기능이 지정된 경우:** 등록 정보 변경 요청이 재구성 지연을 시작합니다.
- **활성 I/O 도메인에 가상 기능이 지정된 경우:** 소유 도메인이 비활성 또는 바인딩 상태일 때 변경을 수행해야 하므로 등록 정보 변경 요청이 거부됩니다.
- **비primary 도메인에 가상 기능이 지정되고 재구성 지연이 이미 적용된 경우:** 오류 메시지와 함께 등록 정보 변경 요청이 실패합니다.

pvid 및 vid 네트워크 관련 등록 정보는 제한 없이 변경할 수 있습니다.

가상 기능을 만들면 재구성 지연이 시작되므로 더 많은 가상 기능을 만들고 primary 도메인을 한 번만 재부트하여 변경 사항을 적용할 수 있습니다. 가상 기능을 만들 때마다 primary 도메인을 재부트할 필요가 없습니다.

특정 SR-IOV 물리적 기능이 여러 가상 기능을 지원하는 경우도 있습니다. 필요한 가상 기능만 만드십시오. 권장되는 최대 구성 수는 102 페이지 “고급 SR-IOV 항목”을 참조하십시오.

3. `ldm add-config` 명령을 사용하여 SP에 구성을 저장합니다.
4. 가상 기능이 만들어지도록 primary 도메인을 재부트합니다.
5. `ldm add-io` 명령을 사용하여 활성 도메인에 가상 기능을 지정하기 전에 활성 도메인을 중지해야 합니다. 모든 I/O 도메인 변경 사항을 함께 적용하여 I/O 도메인 작동 중지 시간을 최소화합니다. 이 방법을 사용하면 해당 구성 설정에 필요한 primary 도메인 재부트 횟수를 줄일 수 있습니다.
6. I/O 도메인을 부트하고 다른 네트워크 장치인 것처럼 가상 기능을 구성합니다.
가상 기능 제한 사항에 대한 자세한 내용은 102 페이지 “고급 SR-IOV 항목”을 참조하십시오.

가상 기능 만들기, 수정 및 삭제

이 절에서는 가상 기능을 만들고 수정 및 삭제하는 방법에 대해 설명합니다.

▼ 가상 기능을 만드는 방법

- 1 물리적 기능 장치를 식별합니다.

```
primary# ldm list-io
```

물리적 기능의 이름에는 PCIe SR-IOV 카드 또는 내장 장치에 대한 위치 정보가 포함되어 있습니다.

- 2 물리적 기능을 기반으로 가상 기능을 만듭니다.

```
primary# ldm create-vf [mac-addr=num] [alt-mac-addr=[auto|num1,[auto|num2,...]]]
[pvid=pvid] [vid=vid1,vid2,...] [mtu=size] [name=value...] pf-name
```

주 - 네트워크 장치에 대한 MAC 주소가 자동으로 할당됩니다.

경로 이름 또는 익명 이름을 사용하여 가상 기능을 지정할 수 있습니다. 하지만 익명 이름을 사용하는 것이 좋습니다.

예 6-1 가상 기능 만들기

다음 예에서는 물리적 기능 /SYS/MB/NET0/IOVNET.PF0에 대한 정보를 보여줍니다.

- 이 물리적 기능은 내장 NET0 네트워크 장치에서 온 것입니다.
- IOVNET 문자열은 물리적 기능이 네트워크 SR-IOV 장치임을 나타냅니다.

```
primary# ldm list-io
NAME                                     TYPE   DOMAIN   STATUS
----
pci_0                                  BUS    primary
niu_0                                  NIU    primary
/SYS/MB/RISER0/PCIE0                   PCIE   -        EMP
/SYS/MB/RISER1/PCIE1                   PCIE   -        EMP
/SYS/MB/RISER2/PCIE2                   PCIE   -        EMP
/SYS/MB/RISER0/PCIE3                   PCIE   -        EMP
/SYS/MB/RISER1/PCIE4                   PCIE   primary  OCC
/SYS/MB/RISER2/PCIE5                   PCIE   primary  OCC
/SYS/MB/SASHBA0                        PCIE   primary  OCC
/SYS/MB/SASHBA1                        PCIE   primary  OCC
/SYS/MB/NET0                           PCIE   primary  OCC
/SYS/MB/NET2                           PCIE   primary  OCC
/SYS/MB/RISER1/PCIE4/IOVNET.PF0        PF     -
/SYS/MB/RISER1/PCIE4/IOVNET.PF1        PF     -
/SYS/MB/RISER2/PCIE5/P0/P2/IOVNET.PF0  PF     -
/SYS/MB/RISER2/PCIE5/P0/P2/IOVNET.PF1  PF     -
/SYS/MB/RISER2/PCIE5/P0/P4/IOVNET.PF0  PF     -
/SYS/MB/RISER2/PCIE5/P0/P4/IOVNET.PF1  PF     -
/SYS/MB/NET0/IOVNET.PF0                PF     -
/SYS/MB/NET0/IOVNET.PF1                PF     -
/SYS/MB/NET2/IOVNET.PF0                PF     -
/SYS/MB/NET2/IOVNET.PF1                PF     -
```

다음 명령은 지정된 물리적 기능에 대한 세부 정보를 보여줍니다. maxvfs 값은 장치에서 지원하는 최대 가상 기능 수를 나타냅니다.

```
primary# ldm list-io -l /SYS/MB/NET0/IOVNET.PF0
NAME                                     TYPE   DOMAIN   STATUS
----
/SYS/MB/NET0/IOVNET.PF0                 PF     -
[pci@400/pci@2/pci@0/pci@6/network@0]
maxvfs = 7
```

다음 예에서는 가상 기능을 만드는 방법을 보여줍니다.

- 이 예에서는 선택적 등록 정보를 설정하지 않고 가상 기능을 만듭니다. 이 경우 네트워크 클래스 가상 기능에 대한 MAC 주소가 자동으로 할당됩니다.

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
```

```
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

- 이 예에서는 mac-addr 등록 정보를 00:14:2f:f9:14:c0으로, vid 등록 정보를 VLAN ID 2 및 3으로 설정한 상태에서 가상 기능을 만듭니다.

```
primary# ldm create-vf mac-addr=00:14:2f:f9:14:c0 vid=2,3 /SYS/MB/NET0/IOVNET.PF0
```

- 이 예에서는 2개의 대체 MAC 주소를 가진 가상 기능을 만듭니다. 하나의 MAC 주소는 자동으로 할당되며 다른 MAC 주소는 명시적으로 00:14:2f:f9:14:c2로 지정됩니다.

```
primary# ldm create-vf alt-mac-addr=auto,00:14:2f:f9:14:c2 /SYS/MB/NET0/IOVNET.PF0
```

▼ 가상 기능을 수정하는 방법

ldm set-io vf-name 명령은 등록 정보 값을 변경하거나 새 등록 정보를 설정하여 가상 기능의 현재 구성을 수정합니다. 이 명령은 네트워크 관련 등록 정보와 장치 관련 등록 정보를 모두 수정할 수 있습니다. 장치 관련 등록 정보에 대한 자세한 내용은 [102 페이지 “고급 SR-IOV 항목”](#)을 참조하십시오.

ldm set-io 명령을 사용하여 다음 등록 정보를 수정할 수 있습니다.

- mac-addr, alt-mac-addr 및 mtu

이러한 가상 기능 등록 정보를 변경하려면 먼저 가상 기능을 소유한 도메인을 중지합니다. 해당하는 가상 기능이 primary 도메인에 지정된 경우 재구성 지연이 실행되므로 변경 사항이 적용되도록 재부트해야 합니다.

- pvid 및 vid

도메인에 가상 기능이 지정되는 동안 이러한 등록 정보를 동적으로 변경할 수 있습니다. 이 작업을 수행하면 활성 가상 기능의 네트워크 트래픽이 변경될 수 있습니다. 즉, pvid 등록 정보를 설정하면 투명한 VLAN이 가능합니다. vid 등록 정보를 설정하여 VLAN ID를 지정하면 지정된 해당 VLAN에 대한 VLAN 트래픽이 허용됩니다.

- 장치 관련 등록 정보

ldm list-io -d pf-name 명령을 사용하여 유효한 장치 관련 등록 정보 목록을 확인합니다. 물리적 기능과 가상 기능 모두에 대해 이러한 등록 정보를 수정할 수 있습니다. 이러한 등록 정보를 변경하면 재구성 지연이 실행되므로 변경 사항이 적용되도록 primary 도메인을 재부트해야 합니다. 장치 관련 등록 정보에 대한 자세한 내용은 [102 페이지 “고급 SR-IOV 항목”](#)을 참조하십시오.

- 가상 기능을 수정합니다.

```
primary# ldm set-io name=value [name=value...] vf-name
```

예 6-2 가상 기능 수정

다음 예에서는 `ldm set-io` 명령을 사용하여 가상 기능에 대한 등록 정보를 설정하는 방법을 보여줍니다.

- 다음 예에서는 VLAN ID 2, 3 및 4에 속하도록 지정된 가상 기능 `/SYS/MB/NET0/IOVNET.PF0.VF0`을 수정합니다.

```
primary# ldm set-io vid=2,3,4 /SYS/MB/NET0/IOVNET.PF0.VF0
```

이 명령은 가상 기능에 대한 VLAN 연관을 동적으로 변경합니다. 이러한 VLAN을 사용하려면 적절한 Oracle Solaris OS 네트워킹 명령을 사용하여 I/O 도메인의 VLAN 인터페이스를 구성해야 합니다.

- 다음 예에서는 가상 기능이 투명하게 VLAN 2에 속하도록 `/SYS/MB/NET0/IOVNET.PF0.VF0` 가상 기능에 대한 `pvid` 등록 정보 값을 2로 설정합니다. 즉, 가상 기능에서 태그가 지정된 VLAN 트래픽을 확인하지 않습니다.

```
primary# ldm set-io pvid=2 /SYS/MB/NET0/IOVNET.PF0.VF0
```

- 다음 예에서는 자동으로 할당된 3개의 대체 MAC 주소를 가상 기능에 지정합니다. 대체 주소를 통해 가상 기능에 Oracle Solaris 11 VNIC(가상 네트워크 인터페이스 카드)를 만들 수 있습니다. VNIC를 사용하려면 도메인에서 Oracle Solaris 11 OS를 실행해야 합니다.

주 - 이 명령을 실행하기 전에 가상 기능을 소유한 도메인을 중지하십시오.

```
primary# ldm set-io alt-mac-addr=auto,auto,auto /SYS/MB/NET0/IOVNET.PF0.VF0
```

- 다음 예에서는 지정된 가상 기능에 대한 장치 관련 `unicast-slots` 등록 정보를 12로 설정합니다. 물리적 기능에 유효한 장치 관련 등록 정보를 찾으려면 `ldm list-io -d pf-name` 명령을 사용합니다.

```
primary# ldm set-io unicast-slots=12 /SYS/MB/NET0/IOVNET.PF0.VF0
```

Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary domain reboots, at which time the new configuration for the primary domain will also take effect.

▼ 가상 기능을 삭제하는 방법

가상 기능은 현재 도메인에 지정되지 않은 경우 삭제할 수 있습니다. 또한 마지막으로 만들어진 가상 기능만 삭제할 수 있습니다. 이로 인한 구성은 물리적 기능 드라이버를 통해 검증됩니다. 작업이 성공하면 재구성 지연이 시작되므로 변경 사항이 적용되도록 재부트해야 합니다.

- 가상 기능을 삭제합니다.

```
primary# ldm destroy-vf vf-name
```

예 6-3 가상 기능 삭제

다음 예에서는 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능을 삭제하는 방법을 보여줍니다.

```
primary# ldm destroy-vf /SYS/MB/NET0/IOVNET.PF0.VF0
```

Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary domain reboots, at which time the new configuration for the primary domain will also take effect.

I/O 도메인에서 가상 기능 추가 및 제거

▼ I/O 도메인에 가상 기능을 추가하는 방법

다음 명령은 논리적 도메인에 가상 기능을 추가합니다.

```
ldm add-io vf-name ldom
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 익명 이름을 사용하는 것이 좋습니다. *ldom*은 가상 기능을 추가할 도메인의 이름을 지정합니다. 지정된 게스트는 비활성 또는 바인딩 상태여야 합니다. *primary* 도메인을 지정할 경우 이 명령은 재구성 지연을 시작합니다.

● 가상 기능을 추가합니다.

```
primary# ldm add-io vf-name ldom
```

도메인의 가상 기능에 대한 장치 경로 이름은 `list-io -l` 출력에 표시된 경로입니다.

예 6-4 가상 기능 추가

다음 예에서는 *ldg1* 도메인에 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능을 추가하는 방법을 보여줍니다. 성공하려면 지정된 도메인이 비활성 또는 바인딩 상태여야 합니다. 도메인이 *primary* 도메인일 경우 재구성 지연이 시작됩니다.

```
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

명령이 성공하면 *ldg1* 도메인에 가상 기능이 추가됩니다. *ldg1*이 이미 바인딩된 경우(또는 나중에 바인딩되는 경우) 도메인이 시작될 수 있으며 게스트 OS가 I/O 작업에 대해 추가된 가상 기능을 사용할 수 있습니다.

▼ I/O 도메인에서 가상 기능을 제거하는 방법

다음 명령은 논리적 도메인에서 SR-IOV 가상 기능을 제거합니다.

```
ldm remove-io vf-name ldom
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 장치 이름을 사용하는 것이 좋습니다. *ldom*은 가상 기능을 제거할 도메인의 이름을 지정합니다. 지정된 게스트는 비활성 또는 바인딩 상태여야 합니다. *primary* 도메인을 지정할 경우 이 명령은 재구성 지연을 시작합니다.

주 - 도메인에서 가상 기능을 제거하기 전에 해당 도메인 부트에 반드시 필요한 가상 기능이 아닌지 확인하십시오.

● 가상 기능을 제거합니다.

```
primary# ldm rm-io vf-name ldom
```

예 6-5 가상 기능 제거

다음 예에서는 *ldg1* 도메인에서 */SYS/MB/NET0/IOVNET.PF0.VF0* 가상 기능을 제거하는 방법을 보여줍니다.

```
primary# ldm rm-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

명령이 성공하면 *ldg1* 도메인에서 가상 기능이 제거됩니다. *ldg1*이 다시 시작되면 지정된 가상 기능이 더 이상 해당 도메인에 나타나지 않습니다.

가상 기능이 있는 도메인이 *primary* 도메인일 경우 재구성 지연이 시작됩니다.

SR-IOV:primary 도메인 재부트

primary 도메인을 재부트할 때는 신중해야 합니다. 83 페이지 “*primary* 도메인 재부트”를 참조하십시오. I/O 도메인의 PCIe 슬롯과 마찬가지로 이 절에서 설명되는 사항은 I/O 도메인에 지정된 가상 기능과도 관련이 있습니다.

SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기

다음 절차에서는 PCIe SR-IOV 가상 기능이 포함된 I/O 도메인을 만드는 방법에 대해 설명합니다.

▼ SR-IOV 가상 기능을 지정하여 I/O 도메인을 만드는 방법

작동 중지 시간이 최소화되도록 사전 계획을 통해 primary 도메인 재부트 횟수를 최소화하십시오.

- 1 SR-IOV 기능을 사용하는 I/O 도메인과 공유할 SR-IOV 물리적 기능을 식별합니다.

```
primary# ldm list-io
```

- 2 공유할 물리적 기능에 대한 세부 정보를 얻습니다.

```
primary# ldm list-io -l pf-name
```

- 3 물리적 기능에 대한 가상 기능을 하나 이상 만듭니다.

```
primary# ldm create-vf pf-name
```

만들려는 각 가상 기능에 대해 이 명령을 실행할 수 있습니다. 이러한 명령을 일괄 처리로 수행할 경우 primary 도메인을 한 번만 재부트하면 됩니다.

- 4 PCIe 끝점 장치 또는 가상 기능이 지정된 I/O 도메인을 중지합니다.

```
primary# ldm stop ldom
```

- 5 primary 도메인을 재부트합니다.

```
primary# reboot
```

- 6 primary 도메인에서 사용 가능한 가상 기능 목록을 확인합니다.

```
primary# ldm list-io
```

- 7 도메인에 가상 기능을 지정합니다.

```
primary# ldm add-io vf-name ldom
```

- 8 바인딩 후 도메인을 시작합니다.

```
primary# ldm bind ldom
primary# ldm start ldom
```

- 9 I/O 도메인에서 가상 기능을 사용할 수 있는지 확인합니다.

다음 Oracle Solaris 11 명령은 가상 기능의 가용성을 보여줍니다.

```
guest# dladm show-phys
```

예 6-6 SR-IOV 가상 기능을 지정하여 I/O 도메인 만들기

다음 예에서는 물리적 기능 /SYS/MB/NET0/IOVNET.PF0에 대해 가상 기능 /SYS/MB/NET0/IOVNET.PF0.VF0을 만들고 ldg1 I/O 도메인에 가상 기능을 지정하는 방법을 보여줍니다.

다음 ldm list-io 출력에서는 /SYS/MB/NET0/IOVNET.PF0 물리적 기능을 사용할 수 있음을 보여줍니다.

```
primary# ldm list-io
NAME                                     TYPE   DOMAIN   STATUS
----
pci_0                                   BUS    primary
niu_0                                   NIU    primary
/SYS/MB/RISER0/PCIE0                   PCIE   -        EMP
/SYS/MB/RISER1/PCIE1                   PCIE   -        EMP
/SYS/MB/RISER2/PCIE2                   PCIE   -        EMP
/SYS/MB/RISER0/PCIE3                   PCIE   -        EMP
/SYS/MB/RISER1/PCIE4                   PCIE   primary  OCC
/SYS/MB/RISER2/PCIE5                   PCIE   primary  OCC
/SYS/MB/SASHBA0                        PCIE   primary  OCC
/SYS/MB/SASHBA1                        PCIE   primary  OCC
/SYS/MB/NET0                           PCIE   primary  OCC
/SYS/MB/NET2                           PCIE   primary  OCC
/SYS/MB/RISER1/PCIE4/IOVNET.PF0        PF     -
/SYS/MB/RISER1/PCIE4/IOVNET.PF1        PF     -
/SYS/MB/RISER2/PCIE5/P0/P2/IOVNET.PF0  PF     -
/SYS/MB/RISER2/PCIE5/P0/P2/IOVNET.PF1  PF     -
/SYS/MB/RISER2/PCIE5/P0/P4/IOVNET.PF0  PF     -
/SYS/MB/RISER2/PCIE5/P0/P4/IOVNET.PF1  PF     -
/SYS/MB/NET0/IOVNET.PF0                PF     -
/SYS/MB/NET0/IOVNET.PF1                PF     -
/SYS/MB/NET2/IOVNET.PF0                PF     -
/SYS/MB/NET2/IOVNET.PF1                PF     -
```

다음 명령은 /SYS/MB/NET0/IOVNET.PF0 물리적 기능에 대한 추가 세부 정보를 보여 주며 만들 수 있는 최대 가상 기능 수를 제공합니다.

```
primary# ldm list-io -l /SYS/MB/NET0/IOVNET.PF0
NAME                                     TYPE   DOMAIN   STATUS
----
/SYS/MB/NET0/IOVNET.PF0                 PF     -
[pci@400/pci@2/pci@0/pci@6/network@0]
maxvfs = 7
```

다음 명령은 /SYS/MB/NET0/IOVNET.PF0 물리적 기능에 대해 /SYS/MB/NET0/IOVNET.PF0.VF0이라는 가상 기능을 만듭니다.

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
```

```
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
Created new VF: /SYS/MB/NET0/IOVNET.PF0.VF0
```

ldg1 I/O 도메인에는 DIO 기능을 사용하여 만들어진 PCIe 끝점 장치가 있으므로 다음과 같이 ldg1 도메인을 중지하고 primary 도메인을 재부트해야 합니다.

```
primary# ldm stop ldg1
primary# reboot
```

다음 명령은 새 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능이 존재하는지 확인합니다.

```
primary# ldm list-io
NAME                                TYPE    DOMAIN    STATUS
----                                -
pci_0                              BUS     primary
niu_0                              NIU     primary
/SYS/MB/RISER0/PCIE0               PCIE    -         EMP
/SYS/MB/RISER1/PCIE1               PCIE    -         EMP
/SYS/MB/RISER2/PCIE2               PCIE    -         EMP
/SYS/MB/RISER0/PCIE3               PCIE    -         EMP
/SYS/MB/RISER1/PCIE4               PCIE    primary   OCC
/SYS/MB/RISER2/PCIE5               PCIE    primary   OCC
/SYS/MB/SASHBA0                    PCIE    primary   OCC
/SYS/MB/SASHBA1                    PCIE    primary   OCC
/SYS/MB/NET0                        PCIE    primary   OCC
/SYS/MB/NET2                        PCIE    primary   OCC
/SYS/MB/RISER1/PCIE4/IOVNET.PF0    PF      -
/SYS/MB/RISER1/PCIE4/IOVNET.PF1    PF      -
/SYS/MB/RISER2/PCIE5/P0/P2/IOVNET.PF0 PF      -
/SYS/MB/RISER2/PCIE5/P0/P2/IOVNET.PF1 PF      -
/SYS/MB/RISER2/PCIE5/P0/P4/IOVNET.PF0 PF      -
/SYS/MB/RISER2/PCIE5/P0/P4/IOVNET.PF1 PF      -
/SYS/MB/NET0/IOVNET.PF0            PF      -
/SYS/MB/NET0/IOVNET.PF1            PF      -
/SYS/MB/NET2/IOVNET.PF0            PF      -
/SYS/MB/NET2/IOVNET.PF1            PF      -
/SYS/MB/NET0/IOVNET.PF0.VF0        VF
```

다음 명령은 ldg1 도메인에 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능을 지정합니다.

```
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

다음 명령은 바인딩 후 ldg1 도메인을 다시 시작합니다.

```
primary# ldm bind ldg1
primary# ldm start ldg1
```

다음 명령은 가상 기능을 사용할 수 있는지 확인합니다.

```
guest# dladm show-phys
LINK          MEDIA          STATE    SPEED    DUPLEX    DEVICE
net0          Ethernet       up       0        unknown  vnet0
net1          Ethernet       up       1000     full     igbvf0
```

고급 SR-IOV 항목

이 절에서는 PCIe SR-IOV 가상 I/O 장치를 사용할 때 제기되는 여러 가지 고급 항목에 대해 설명합니다.

- **SR-IOV 가상 기능을 사용하여 I/O 도메인 부트.** SR-IOV 가상 기능은 다른 유형의 PCIe 장치와 유사한 기능(예: 가상 기능을 논리적 도메인 부트 장치로 사용할 수 있는 기능)을 제공합니다. 예를 들어, 네트워크 가상 기능을 사용하면 네트워크를 통해 부트하여 I/O 도메인에 Oracle Solaris OS를 설치할 수 있습니다.

주 - 가상 기능 장치에서 Oracle Solaris OS를 부트할 때 로드하려는 Oracle Solaris OS가 가상 기능 장치를 지원하는지 확인하십시오. 지원하는 경우 계획대로 나머지 설치를 계속할 수 있습니다.

- **지원되는 최대 I/O 도메인 및 가상 기능 수.** 특정 PCIe 버스의 PCIe 끝점 장치 및 SR-IOV 가상 기능은 최대 15개의 도메인에 지정할 수 있습니다. PCIe 리소스(예: 각 PCIe 버스에 대한 인터럽트 벡터)는 루트 도메인 및 I/O 도메인에서 배분됩니다. 이에 따라 특정 I/O 도메인에 지정할 수 있는 장치 수도 제한됩니다. 그러므로 동일한 I/O 도메인에 많은 수의 가상 기능을 지정하지 않아야 합니다. SR-IOV 관련 문제에 대한 설명은 [Oracle VM Server for SPARC 2.2 릴리스 노트](#)를 참조하십시오.

SR-IOV 장치 관련 등록 정보

SR-IOV 물리적 기능 장치 드라이버는 장치 관련 등록 정보를 내보낼 수 있습니다. 이러한 등록 정보를 사용하여 물리적 기능과 해당 가상 기능의 리소스 할당을 조정할 수 있습니다. 등록 정보에 대한 자세한 내용은 [igb\(7D\)](#) 및 [ixgbe\(7D\)](#) 매뉴얼 페이지와 같은 물리적 기능 드라이버에 대한 매뉴얼 페이지를 참조하십시오.

`ldm list-io -d` 명령은 지정된 물리적 기능 장치 드라이버가 내보낸 장치 관련 등록 정보를 보여줍니다. 각 등록 정보에는 이름, 간단한 설명, 기본값, 최대값 및 하나 이상의 다음 플래그가 있습니다.

- P 물리적 기능에 적용됩니다.
- V 가상 기능에 적용됩니다.
- R 읽기 전용 또는 단순한 정보 매개변수입니다.

```
primary# ldm list-io -d pf-name
```

물리적 기능 또는 가상 기능에 대해 읽기/쓰기 등록 정보를 설정하려면 `ldm create-vf` 또는 `ldm set-io` 명령을 사용하십시오. 장치 관련 등록 정보를 설정하면 재구성 지연이 시작됩니다.

다음 예에서는 내장 Intel 1Gbps SR-IOV 장치가 내보낸 장치 관련 등록 정보를 보여줍니다.

```
primary# ldm list-io -d /SYS/MB/NET0/IOVNET.PF0
Device-specific Parameters
-----
max-config-vfs
  Flags = PR
  Default = 7
  Descr = Max number of configurable VFs
max-vf-mtu
  Flags = VR
  Default = 9216
```

```

    Descr = Max MTU supported for a VF
max-vlans
    Flags = VR
    Default = 32
    Descr = Max number of VLAN filters supported
pvid-exclusive
    Flags = VR
    Default = 1
    Descr = Exclusive configuration of pvid required
unicast-slots
    Flags = PV
    Default = 0 Min = 0 Max = 24
    Descr = Number of unicast mac-address slots

```

다음 예에서는 unicast-slots 등록 정보를 8로 설정합니다.

```
primary# ldm create-vf unicast-slots=8 /SYS/MB/NET0/IOVNET.PF0
```

가상 기능에 대한 고급 네트워크 구성

- SR-IOV 가상 기능은 Logical Domains Manager가 지정한 MAC 주소만 사용할 수 있습니다. 다른 Oracle Solaris OS 네트워킹 명령을 사용하여 I/O 도메인에서 MAC 주소를 변경할 경우 명령이 실패하거나 제대로 작동하지 않을 수 있습니다.
- 이때 I/O 도메인에 있는 SR-IOV 네트워크 가상 기능의 링크 통합은 지원되지 않습니다. 링크 통합을 만들려고 하면 예상대로 작동하지 않을 수 있습니다.
- 가상 I/O 서비스를 만들어 I/O 도메인에 지정할 수 있습니다. 이러한 가상 I/O 서비스는 가상 기능이 만들어진 것과 동일한 물리적 기능에 만들 수 있습니다. 예를 들어, 내장 1Gbps 네트워크 장치(net0 또는 igb0)를 가상 스위치용 네트워크 백엔드 장치로 사용할 수 있으며 동일한 물리적 기능 장치에서 가상 기능을 만들 수도 있습니다.

SR-IOV 가상 기능에 VNIC 만들기

SR-IOV 가상 기능에 Oracle Solaris 11 VNIC를 만들 수 있습니다. 하지만 지원되는 VNIC 수는 가상 기능에 지정된 대체 MAC 주소 수(alt-mac-addr 등록 정보)로 제한됩니다. 따라서 가상 기능에서 VNIC를 사용할 때는 대체 MAC 주소 수를 충분히 지정해야 합니다. 대체 MAC 주소와 함께 alt-mac-addr 등록 정보를 설정하려면 ldm create-vf 또는 ldm set-io 명령을 사용하십시오.

다음 예에서는 SR-IOV 가상 기능에 4개의 VNIC를 생성하는 과정을 보여줍니다. 첫번째 명령은 가상 기능 장치에 대체 MAC 주소를 지정합니다. 이 명령에서는 자동 할당 방법을 사용하여 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능 장치에 4개의 대체 MAC 주소를 할당합니다.

```
primary# ldm set-io alt-mac-addr=auto,auto,auto,auto /SYS/MB/NET0/IOVNET.PF0.VF0
```

다음 명령은 I/O 도메인에서 Oracle Solaris 11 OS를 시작하고 부트합니다. 이 예에서 ldg1은 I/O 도메인입니다.

```
primary# ldm start ldg1
```

다음 명령에서는 게스트 도메인에서 Oracle Solaris 11 **dladm** 명령을 사용하여 4개의 VNIC를 만듭니다. 대체 MAC 주소를 사용하여 지정된 것보다 많은 수의 VNIC를 만들려고 시도하면 실패합니다.

```
guest# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	0	unknown	vnet0
net1	Ethernet	up	1000	full	igbvf0

```

guest# dladm create-vnic -l net1 vnic0
guest# dladm create-vnic -l net1 vnic1
guest# dladm create-vnic -l net1 vnic2
guest# dladm create-vnic -l net1 vnic3
guest# dladm show-link
```

LINK	CLASS	MTU	STATE	OVER
net0	phys	1500	up	--
net1	phys	1500	up	--
vnic0	vnic	1500	up	net1
vnic1	vnic	1500	up	net1
vnic2	vnic	1500	up	net1
vnic3	vnic	1500	up	net1

가상 디스크 사용

이 장에서는 Oracle VM Server for SPARC 소프트웨어에서 가상 디스크를 사용하는 방법에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- 107 페이지 “가상 디스크 소개”
- 108 페이지 “가상 디스크 관리”
- 110 페이지 “가상 디스크 식별자 및 장치 이름”
- 111 페이지 “가상 디스크 표시”
- 112 페이지 “가상 디스크 백엔드 옵션”
- 114 페이지 “가상 디스크 백엔드”
- 120 페이지 “가상 디스크 다중 경로 구성”
- 122 페이지 “CD, DVD 및 ISO 이미지”
- 126 페이지 “가상 디스크 시간 초과”
- 127 페이지 “가상 디스크 및 SCSI”
- 127 페이지 “가상 디스크 및 format 명령”
- 127 페이지 “가상 디스크에서 ZFS 사용”
- 131 페이지 “Logical Domains 환경에서 볼륨 관리자 사용”

가상 디스크 소개

가상 디스크에는 게스트 도메인에 나타나는 가상 디스크 자체와 데이터가 저장되고 가상 I/O가 끝나는 가상 디스크 백엔드의 두 구성 요소가 포함됩니다. 가상 디스크 백엔드는 가상 디스크 서버(vds) 드라이버로 서비스 도메인에서 내보냅니다. vds 드라이버는 LDC(논리적 도메인 채널)를 사용하여 하이퍼바이저를 통해 게스트 도메인의 가상 디스크 클라이언트(vdc) 드라이버와 통신합니다. 마지막으로 가상 디스크는 게스트 도메인에서 /dev/[r]dsk/cXdYsZ 장치로 나타납니다.

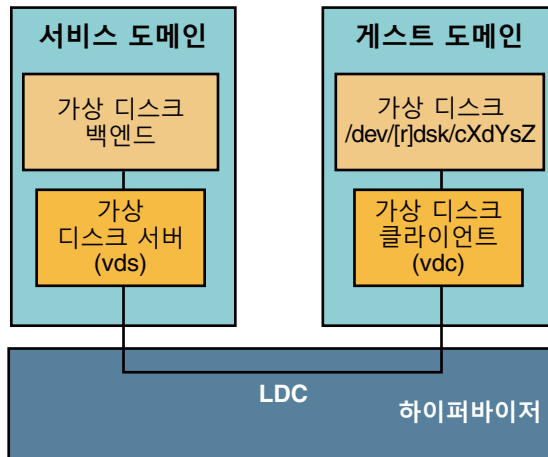
가상 디스크 백엔드는 물리적이거나 논리적일 수 있습니다. 물리적 장치에는 다음이 포함될 수 있습니다.

- 물리적 디스크 또는 디스크 논리 장치 번호(LUN)
- 물리적 디스크 슬라이스

논리적 장치는 다음 중 하나가 될 수 있습니다.

- 파일 시스템(예: ZFS 또는 UFS)의 파일
- 볼륨 관리자(예: ZFS, VxVM 또는 Solaris Volume Manager)의 논리적 볼륨
- 서비스 도메인에서 액세스 가능한 디스크 의사 장치

그림 7-1 Logical Domains의 가상 디스크



가상 디스크 관리

이 절에서는 게스트 도메인에 가상 디스크 추가, 가상 디스크와 시간 초과 옵션 변경 및 게스트 도메인에서 가상 디스크 제거에 대해 설명합니다. 가상 디스크 옵션에 대한 설명은 [112 페이지](#) “가상 디스크 백엔드 옵션”을 설명하십시오. 가상 디스크 시간 초과에 대한 설명은 [126 페이지](#) “가상 디스크 시간 초과”를 참조하십시오.

▼ 가상 디스크를 추가하는 방법

- 1 서비스도메인에서 가상 디스크 백엔드를 내보냅니다.

```
# ldm add-vdsdev [-fq] [options={ro,slice,excl}] [mpgroup=mpgroup] \
backend volume-name@service-name
```

2 게스트 도메인에 백엔드를 지정합니다.

```
# ldm add-vdisk [timeout=seconds] [id=disk-id] disk-name volume-name@service-name ldom
```

id 등록 정보를 설정하여 새 가상 디스크 장치의 ID를 지정할 수 있습니다. 기본적으로 ID 값은 자동으로 생성되므로 OS의 기존 장치 이름과 일치시켜야 하는 경우에만 이 등록 정보를 설정하십시오. [110 페이지 “가상 디스크 식별자 및 장치 이름”](#)을 참조하십시오.

주 - 백엔드는 게스트 도메인(ldom)이 바인드될 때 실제로 서비스 도메인에서 내보내지고 게스트 도메인에 지정됩니다.

▼ 가상 디스크 백엔드를 여러 번 내보내는 방법

가상 디스크 백엔드는 동일하거나 다른 가상 디스크 서버를 통해 여러 번 내보낼 수 있습니다. 그런 다음 각 가상 디스크 백엔드의 내보낸 인스턴스를 동일하거나 다른 게스트 도메인에 지정할 수 있습니다.

가상 디스크 백엔드를 여러 번 내보낼 경우 배타적(excl) 옵션을 사용하여 내보내면 안됩니다. excl 옵션을 지정하면 백엔드 내보내기가 한 번만 허용됩니다. 백엔드는 ro 옵션을 사용하여 읽기 전용 장치로 안전하게 여러 번 내보낼 수 있습니다.



주의 - 가상 디스크 백엔드를 여러 번 내보낼 경우 게스트 도메인에서 실행 중이고 해당 가상 디스크를 사용 중인 응용 프로그램이 데이터 일관성 보장을 위해 동시 쓰기 액세스를 조정하고 동기화해야 합니다.

다음 예에서는 동일한 가상 디스크 서비스를 통해 동일 가상 디스크를 서로 다른 두 게스트 도메인에 추가하는 방법을 설명합니다.

1 다음 명령을 사용하여 서비스 도메인에서 가상 디스크 백엔드를 두 번 내보냅니다.

```
# ldm add-vdsdev [options={ro,slice}] backend volume1@service-name
# ldm add-vdsdev -f [options={ro,slice}] backend volume2@service-name
```

두 번째 ldm add-vdsdev 명령에서는 -f 옵션을 사용하여 백엔드의 두 번째 내보내기를 수행하고 있습니다. 두 명령에 대해 동일한 백엔드 경로를 사용할 때 및 가상 디스크 서버가 동일한 서비스 도메인에 있을 때 이 옵션을 사용하십시오.

2 다음 명령을 사용하여 내보낸 백엔드를 각 게스트 도메인에 지정합니다.

disk-name은 ldom1 및 ldom2에 대해 서로 다를 수 있습니다.

```
# ldm add-vdisk [timeout=seconds] disk-name volume1@service-name ldom1
# ldm add-vdisk [timeout=seconds] disk-name volume2@service-name ldom2
```

▼ 가상 디스크 옵션을 변경하는 방법

가상 디스크 옵션에 대한 자세한 내용은 [112 페이지 “가상 디스크 백엔드 옵션”](#)을 참조하십시오.

- 서비스 도메인에서 백엔드를 내보낸 후 다음 명령을 사용하여 가상 디스크 옵션을 변경할 수 있습니다.

```
# ldm set-vdsdev options=[{ro,slice,excl}] volume-name@service-name
```

▼ 시간 초과 옵션을 변경하는 방법

가상 디스크 옵션에 대한 자세한 내용은 [112 페이지 “가상 디스크 백엔드 옵션”](#)을 참조하십시오.

- 게스트 도메인에 가상 디스크를 지정한 후 다음 명령을 사용하여 가상 디스크 시간 초과를 변경할 수 있습니다.

```
# ldm set-vdisk timeout=seconds disk-name ldom
```

▼ 가상 디스크를 제거하는 방법

- 1 다음 명령을 사용하여 게스트 도메인에서 가상 디스크를 제거합니다.

```
# ldm rm-vdisk disk-name ldom
```

- 2 다음 명령을 사용하여 서비스 도메인에서 해당 백엔드 내보내기를 중지합니다.

```
# ldm rm-vdsdev volume-name@service-name
```

가상 디스크 식별자 및 장치 이름

ldm add-vdisk 명령을 사용하여 가상 디스크를 도메인에 추가하는 경우 id 등록 정보를 설정하여 해당 장치 번호를 지정할 수 있습니다.

```
# ldm add-vdisk [id=disk-id] disk-name volume-name@service-name ldom
```

도메인의 각 가상 디스크는 도메인이 바인드될 때 지정되는 고유한 장치 번호를 가집니다. 가상 디스크가 명시적인 장치 번호와 함께 추가될 경우(id 등록 정보를 설정하여) 지정된 장치 번호가 사용됩니다. 그렇지 않은 경우 시스템에서 사용 가능한 가장 낮은 장치 번호를 자동으로 지정합니다. 이 경우 지정되는 장치 번호는 도메인에 가상 디스크가 추가된 방법에 따라 달라집니다. 결국 가상 디스크에 지정되는 장치 번호는 도메인이 바인드될 때 ldm list-bindings 명령의 출력에 표시됩니다.

가상 디스크가 있는 도메인에서 Oracle Solaris OS를 실행 중인 경우 각 가상 디스크는 도메인에서 c0dn 디스크 장치로 나타나며, 여기서 n은 가상 디스크의 장치 번호입니다.

다음 예에서 `ldg1` 도메인에는 `rootdisk` 및 `pdisk`의 두 가상 디스크가 있습니다. `rootdisk`의 장치 번호는 `0(disk@0)`이고 도메인에서 디스크 장치 `c0d0`으로 나타납니다. `pdisk`의 장치 번호는 `1(disk@1)`이고 도메인에서 디스크 장치 `c0d1`로 나타납니다.

```
primary# ldm list-bindings ldg1
...
DISK
  NAME          VOLUME          TOUT DEVICE  SERVER      MPGROUP
  rootdisk      dsk_nevada@primary-vds0    disk@0  primary
  pdisk         c3t40d1@primary-vds0    disk@1  primary
...
```



주의 - 장치 번호가 가상 디스크에 명시적으로 지정되지 않은 경우 도메인이 바인드 해제되고 나중에 재바인드될 때 해당 장치 번호를 변경할 수 있습니다. 이 경우 도메인에서 실행 중인 OS에서 지정한 장치 이름도 변경할 수 있으며 시스템의 기존 구성이 달라집니다. 예를 들어, 가상 디스크가 도메인의 구성에서 제거될 경우 이러한 상황이 발생할 수 있습니다.

가상 디스크 표시

백엔드를 가상 디스크로 내보낼 경우 게스트 도메인에서 전체 디스크나 단일 슬라이스 디스크로 나타낼 수 있습니다. 나타나는 방식은 백엔드의 유형 및 내보내는 데 사용된 옵션에 따라 달라집니다.

전체 디스크

백엔드를 전체 디스크로 도메인에 내보낼 경우 해당 도메인에서 8개의 슬라이스(`s0-s7`)를 가진 일반 디스크로 나타납니다. 이러한 디스크는 `format(1M)` 명령으로 볼 수 있습니다. 디스크의 분할 영역 테이블은 `fmthard` 또는 `format` 명령을 사용하여 변경할 수 있습니다.

전체 디스크는 OS 설치 소프트웨어에도 표시되며 OS를 설치할 디스크로 선택할 수 있습니다.

단일 슬라이스 디스크로만 내보낼 수 있는 물리적 디스크 슬라이스를 제외하고 모든 백엔드는 전체 디스크로 내보낼 수 있습니다.

단일 슬라이스 디스크

백엔드를 단일 슬라이스 디스크로 도메인에 내보낼 경우 해당 도메인에서 8개의 슬라이스(`s0` to `s7`)를 가진 일반 디스크로 나타냅니다. 하지만 첫번째 슬라이스(`s0`)만 사용할 수 있습니다. 이러한 디스크는 `format(1M)` 명령으로 볼 수 있지만, 디스크의 분할 영역 테이블은 변경할 수 없습니다.

단일 슬라이스 디스크는 OS 설치 소프트웨어에도 표시되며 OS를 설치할 디스크로 선택할 수 있습니다. 이 때 UNIX 파일 시스템(UFS)을 사용하여 OS를 설치할 경우 루트 분할 영역(/)만 정의해야 하며 이 분할 영역에서 모든 디스크 공간을 사용해야 합니다.

전체 디스크로만 내보낼 수 있는 물리적 디스크를 제외하고 모든 백엔드는 단일 슬라이스 디스크로 내보낼 수 있습니다.

주 - Oracle Solaris 10 10/08 OS 릴리스 이전에는 단일 슬라이스 디스크가 단일 분할 영역(s0)을 가진 디스크로 나타났습니다. 이러한 디스크는 `format` 명령을 사용하여 볼 수 없었습니다. 또한 디스크는 OS 설치 소프트웨어에 표시되지 않았으며 OS를 설치할 디스크 장치로 선택할 수 없었습니다.

가상 디스크 백엔드 옵션

가상 디스크 백엔드를 내보낼 때 서로 다른 옵션을 지정할 수 있습니다. 이러한 옵션은 `ldmadd-vdsdev` 명령의 `options=` 인수에 콤마로 구분된 목록으로 지정됩니다. 유효한 옵션은 `ro`, `slice` 및 `excl`입니다.

읽기 전용(ro) 옵션

읽기 전용(ro) 옵션은 백엔드를 읽기 전용 장치로 내보내도록 지정합니다. 이 경우 게스트 도메인에 지정되는 가상 디스크는 읽기 작업용으로만 액세스할 수 있으며, 가상 디스크에 대한 모든 쓰기 작업은 실패합니다.

배타적(excl) 옵션

배타적(excl) 옵션은 다른 도메인에 가상 디스크로 내보낼 때 서비스 도메인의 백엔드를 가상 디스크 서버에서 배타적으로 열도록 지정합니다. 백엔드가 배타적으로 열리면 서비스 도메인의 다른 응용 프로그램에서 액세스할 수 없게 됩니다. 그러면 서비스 도메인에서 실행 중인 응용 프로그램이 게스트 도메인에서도 사용되고 있는 백엔드를 실수로 사용하는 일이 없어집니다.

주 - 일부 드라이버는 `excl` 옵션을 준수하지 않으며 일부 가상 디스크 백엔드가 배타적으로 열리는 것을 허용하지 않습니다. `excl` 옵션은 물리적 디스크 및 슬라이스에서 작동하는 것으로 알려져 있지만, 이 옵션은 파일에서 작동하지 않습니다. 디스크 볼륨과 같은 의사 장치에서는 작동하거나 작동하지 않을 수 있습니다. 백엔드의 드라이버가 배타적 열기를 준수하지 않을 경우 백엔드 `excl` 옵션은 무시되고 백엔드가 배타적으로 열리지 않습니다.

excl 옵션은 서비스 도메인에서 실행 중인 응용 프로그램이 게스트 도메인으로 내보낸 백엔드에 액세스하지 못하도록 하므로 다음 상황에서는 excl 옵션을 설정하지 마십시오.

- 게스트 도메인이 실행 중일 때 format 또는 luxadm과 같은 명령을 사용하여 물리적 디스크를 관리하려는 경우 excl 옵션과 함께 이러한 디스크를 내보내지 마십시오.
- Solaris Volume Manager 볼륨(RAID 또는 미러링된 볼륨 등)을 내보낼 때 excl 옵션을 설정하지 마십시오. 그렇지 않으면 RAID 또는 미러링된 볼륨의 구성 요소가 실패할 경우 Solaris Volume Manager에서 일부 복구 작업을 시작하지 못할 수 있습니다. 자세한 내용은 132 페이지 “Solaris Volume Manager 위에서 가상 디스크 사용”을 참조하십시오.
- VxVM(Veritas Volume Manager)이 서비스 도메인에 설치되고 VxDMP(Veritas Dynamic Multipathing)가 물리적 디스크에 대해 사용으로 설정된 경우 물리적 디스크는 excl 옵션 없이(기본값 아님) 내보내야 합니다. 그렇지 않으면 가상 디스크 서버(vds)에서 물리적 디스크 장치를 열 수 없으므로 내보내기가 실패합니다. 자세한 내용은 133 페이지 “VxVM이 설치된 경우 가상 디스크 사용”을 참조하십시오.
- 동일 가상 디스크 서비스에서 동일한 가상 디스크 백엔드를 여러 번 내보내는 경우 자세한 내용은 109 페이지 “가상 디스크 백엔드를 여러 번 내보내는 방법”을 참조하십시오.

기본적으로 백엔드는 비배타적으로 열립니다. 이에 따라 백엔드는 서비스 도메인에서 실행 중인 응용 프로그램이 사용할 수 있고 다른 도메인에도 내보낼 수 있습니다. 이 옵션은 Oracle Solaris 10 5/08 OS 릴리스부터 시작되는 새로운 기능입니다. Oracle Solaris 10 5/08 OS 릴리스 이전에는 디스크 백엔드가 항상 배타적으로 열렸으며 백엔드를 비배타적으로 여는 것은 불가능했습니다.

슬라이스(slice) 옵션

백엔드는 해당 유형에 따라 일반적으로 전체 디스크 또는 단일 슬라이스 디스크로 내보내집니다. slice 옵션이 지정되면 백엔드는 강제로 단일 슬라이스 디스크로 내보내집니다.

이 옵션은 백엔드의 원시 콘텐츠를 내보내고자 할 때 유용합니다. 예를 들어, 이미 데이터를 저장한 ZFS 또는 Solaris Volume Manager 볼륨이 있고 게스트 도메인에서 이 데이터에 액세스하도록 하려는 경우 slice 옵션을 사용하여 ZFS 또는 Solaris Volume Manager 볼륨을 내보내야 합니다.

이 옵션에 대한 자세한 내용은 114 페이지 “가상 디스크 백엔드”를 참조하십시오.

가상 디스크 백엔드

가상 디스크 백엔드는 가상 디스크의 데이터가 저장되는 위치입니다. 백엔드는 디스크, 디스크 슬라이스, 파일 또는 볼륨(예: ZFS, Solaris Volume Manager 또는 VxVM)이 될 수 있습니다. 백엔드를 서비스 도메인에서 내보낼 때 `slice` 옵션이 설정되었는지 여부에 따라 백엔드는 게스트 도메인에서 전체 디스크 또는 단일 슬라이스 디스크로 나타납니다. 기본적으로 가상 디스크 백엔드는 읽기 가능-쓰기 가능 전체 디스크로 비배타적으로 내보냅니다.

물리적 디스크 또는 디스크 LUN

물리적 디스크 또는 디스크 LUN은 항상 전체 디스크로 내보냅니다. 이 경우 가상 디스크 드라이버(`vds` 및 `vdc`)가 가상 디스크에서 I/O를 전달하고 물리적 디스크 또는 디스크 LUN에 통로로 작동합니다.

`slice` 옵션을 설정하지 않고 해당 디스크의 슬라이스 2(`s2`)에 해당하는 장치를 내보내서 물리적 디스크 또는 디스크 LUN을 서비스 도메인에서 내보냅니다. `slice` 옵션을 사용하여 디스크의 슬라이스 2를 내보낼 경우 전체 디스크가 아닌 이 슬라이스만 내보내집니다.

▼ 물리적 디스크를 가상 디스크로 내보내는 방법

- 1 물리적 디스크를 가상 디스크로 내보냅니다.

예를 들어, 물리적 디스크 `c1t48d0`을 가상 디스크로 내보내려면 해당 디스크의 슬라이스 2(`c1t48d0s2`)를 내보내야 합니다.

```
primary# ldm add-vdsdev /dev/dsk/c1t48d0s2 c1t48d0@primary-vds0
```

- 2 게스트 도메인에 디스크를 지정합니다.

예를 들어, 디스크(`pdisk`)를 게스트 도메인 `ldg1`에 지정합니다.

```
primary# ldm add-vdisk pdisk c1t48d0@primary-vds0 ldg1
```

- 3 게스트 도메인이 시작되고 Oracle Solaris OS를 실행 중일 때 디스크에 액세스 가능하고 전체 디스크인지 확인합니다.

전체 디스크는 8개의 슬라이스가 있는 일반 디스크입니다.

예를 들어, 확인할 디스크는 `c0d1`입니다.

```
ldg1# ls -l /dev/dsk/c0d1s*
/dev/dsk/c0d1s0
/dev/dsk/c0d1s1
/dev/dsk/c0d1s2
/dev/dsk/c0d1s3
/dev/dsk/c0d1s4
/dev/dsk/c0d1s5
```

```
/dev/dsk/c0d1s6
/dev/dsk/c0d1s7
```

물리적 디스크 슬라이스

물리적 디스크 슬라이스는 항상 단일 슬라이스 디스크로 내보냅니다. 이 경우 가상 디스크 드라이버(vds 및 vdc)가 가상 디스크에서 I/O를 전달하고 물리적 디스크 슬라이스에 통로로 작동합니다.

물리적 디스크 슬라이스는 해당하는 슬라이스 장치를 내보내서 서비스 도메인에서 내보냅니다. 장치가 슬라이스 2와 다른 경우 slice 옵션 지정 여부에 상관 없이 자동으로 단일 슬라이스 디스크로 내보냅니다. 장치가 해당 디스크의 슬라이스 2인 경우 slice 옵션을 설정하여 슬라이스 2만 단일 슬라이스 디스크로 내보내야 합니다. 그렇지 않으면 모든 디스크가 전체 디스크로 내보내집니다.

▼ 물리적 디스크 슬라이스를 가상 디스크로 내보내는 방법

- 1 물리적 디스크의 슬라이스를 가상 디스크로 내보냅니다.

예를 들어, 물리적 디스크(c1t57d0)의 슬라이스 0을 가상 디스크로 내보내려면 해당 슬라이스에 해당하는 장치(c1t57d0s0)를 다음과 같이 내보내야 합니다.

```
primary# ldm add-vdsdev /dev/dsk/c1t57d0s0 c1t57d0s0@primary-vds0
```

슬라이스는 항상 단일 슬라이스 디스크로 내보내지므로 slice 옵션을 지정할 필요가 없습니다.

- 2 게스트 도메인에 디스크를 지정합니다.

예를 들어, 디스크(pslice)를 게스트 도메인 ldg1에 지정합니다.

```
primary# ldm add-vdisk pslice c1t57d0s0@primary-vds0 ldg1
```

- 3 게스트 도메인이 시작되고 Oracle Solaris OS를 실행 중일 때 디스크(예: c0d13)를 나열하고 해당 디스크에 액세스할 수 있는지 확인합니다.

```
ldg1# ls -l /dev/dsk/c0d13s*
/dev/dsk/c0d13s0
/dev/dsk/c0d13s1
/dev/dsk/c0d13s2
/dev/dsk/c0d13s3
/dev/dsk/c0d13s4
/dev/dsk/c0d13s5
/dev/dsk/c0d13s6
/dev/dsk/c0d13s7
```

8개의 장치가 있지만 디스크는 단일 슬라이스 디스크이므로 첫번째 슬라이스(s0)만 사용할 수 있습니다.

▼ 슬라이스 2를 내보내는 방법

- 슬라이스 2(예: 디스크 c1t57d0s2)를 내보내려면 slice 옵션을 지정해야 합니다. 그렇지 않으면 전체 디스크가 내보내집니다.

```
# ldm add-vdsdev options=slice /dev/dsk/c1t57d0s2 c1t57d0s2@primary-vds0
```

파일 및 볼륨

파일 또는 볼륨(예: ZFS 또는 Solaris Volume Manager)은 slice 옵션 설정 여부에 따라 전체 디스크 또는 단일 슬라이스 디스크로 내보내집니다.

전체 디스크로 내보낸 파일 또는 볼륨

slice 옵션을 설정하지 않을 경우 파일 또는 볼륨은 전체 디스크로 내보내집니다. 이 경우 가상 디스크 드라이버(vds 및 vdc)가 가상 디스크에서 I/O를 전달하고 가상 디스크의 분할 영역을 관리합니다. 파일 또는 볼륨은 결국 가상 디스크의 모든 슬라이스 데이터와 분할 영역 및 디스크 구조를 관리하는 데 사용되는 메타 데이터를 포함하는 디스크 이미지가 됩니다.

빈 파일 또는 볼륨을 전체 디스크로 내보낼 경우 게스트 도메인에서 포맷되지 않은 디스크(즉, 분할 영역이 없는 디스크)로 나타납니다. 이 경우 게스트 도메인에서 format 명령을 실행하여 사용 가능한 분할 영역을 정의하고 유효한 디스크 레이블을 작성해야 합니다. 디스크가 포맷되지 않은 동안에는 가상 디스크에 대한 모든 I/O가 실패합니다.

주 - Oracle Solaris 5/08 OS 릴리스 이전에는 빈 파일을 가상 디스크로 내보낼 경우 시스템에서 기본 디스크 레이블을 작성하고 기본 분할 영역을 만들었습니다. Oracle Solaris 5/08 OS 릴리스에서는 더 이상 이 작업이 수행되지 않으며 게스트 도메인에서 format(1M)을 실행하여 분할 영역을 만들어야 합니다.

▼ 파일을 전체 디스크로 내보내는 방법

- 1 서비스 도메인에서 가상 디스크로 사용할 파일(예: fdisk0)을 만듭니다.

```
service# mkfile 100m /ldoms/domain/test/fdisk0
```

파일의 크기로 가상 디스크의 크기가 정의됩니다. 이 예에서는 100MB의 가상 디스크를 얻기 위해 100MB의 빈 파일을 만듭니다.

- 2 제어 도메인에서 파일을 가상 디스크로 내보냅니다.

```
primary# ldm add-vdsdev /ldoms/domain/test/fdisk0 fdisk0@primary-vds0
```

이 예에서는 slice 옵션이 설정되지 않았으므로 파일이 전체 디스크로 내보내집니다.

3 제어 도메인에서 디스크를 게스트 도메인에 지정합니다.

예를 들어, 디스크(fdisk)를 게스트 도메인 ldg1에 지정합니다.

```
primary# ldm add-vdisk fdisk fdisk0@primary-vds0 ldg1
```

4 게스트 도메인이 시작되고 Oracle Solaris OS를 실행 중일 때 디스크에 액세스 가능하고 전체 디스크인지 확인합니다.

전체 디스크는 8개의 슬라이스가 있는 일반 디스크입니다.

다음 예에서는 c0d5 디스크를 나열하며, 액세스 가능하고 전체 디스크인지 확인하는 방법을 보여줍니다.

```
ldg1# ls -l /dev/dsk/c0d5s*
/dev/dsk/c0d5s0
/dev/dsk/c0d5s1
/dev/dsk/c0d5s2
/dev/dsk/c0d5s3
/dev/dsk/c0d5s4
/dev/dsk/c0d5s5
/dev/dsk/c0d5s6
/dev/dsk/c0d5s7
```

▼ ZFS 볼륨을 전체 디스크로 내보내는 방법

1 전체 디스크로 사용할 ZFS 볼륨을 만듭니다.

다음 예에서는 전체 디스크로 사용할 ZFS 볼륨 zdisk0을 만드는 방법을 보여줍니다.

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

볼륨의 크기로 가상 디스크의 크기가 정의됩니다. 이 예에서는 100MB의 가상 디스크를 얻기 위해 100MB의 볼륨을 만듭니다.

2 제어 도메인에서 해당 장치를 ZFS 볼륨으로 내보냅니다.

```
primary# ldm add-vdsdev /dev/zvol/dsk/ldoms/domain/test/zdisk0 \
zdisk0@primary-vds0
```

이 예에서는 slice 옵션이 설정되지 않았으므로 파일이 전체 디스크로 내보내집니다.

3 제어 도메인에서 볼륨을 게스트 도메인에 지정합니다.

다음 예에서는 zdisk0 볼륨을 게스트 도메인 ldg1에 지정하는 방법을 보여줍니다.

```
primary# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```

4 게스트 도메인이 시작되고 Oracle Solaris OS를 실행 중일 때 디스크에 액세스 가능하고 전체 디스크인지 확인합니다.

전체 디스크는 8개의 슬라이스가 있는 일반 디스크입니다.

다음 예에서는 c0d9 디스크를 나열하며, 액세스 가능하고 전체 디스크인지 확인하는 방법을 보여줍니다.

```
ldg1# ls -l /dev/dsk/c0d9s*
/dev/dsk/c0d9s0
```

```

/dev/dsk/c0d9s1
/dev/dsk/c0d9s2
/dev/dsk/c0d9s3
/dev/dsk/c0d9s4
/dev/dsk/c0d9s5
/dev/dsk/c0d9s6
/dev/dsk/c0d9s7

```

단일 슬라이스 디스크로 내보낸 파일 또는 볼륨

slice 옵션이 설정되면 파일 또는 볼륨이 단일 슬라이스 디스크로 내보내집니다. 이 경우 가상 디스크에는 직접 파일 또는 볼륨 백엔드로 매핑되는 하나의 분할 영역(s0)만 있습니다. 파일 또는 볼륨에는 분할 영역 정보나 디스크 구조와 같은 추가 데이터 없이 가상 디스크에 기록된 데이터만 포함됩니다.

파일 또는 볼륨을 단일 슬라이스 디스크로 내보낼 경우 시스템에서는 파일 또는 볼륨이 디스크 슬라이스로 나타나도록 가짜 디스크 분할을 시뮬레이트합니다. 디스크 분할이 시뮬레이트되므로 해당 디스크에 대한 분할 영역을 만들지 않습니다.

▼ ZFS 볼륨을 단일 슬라이스 디스크로 내보내는 방법

1 단일 슬라이스 디스크로 사용할 ZFS 볼륨을 만듭니다.

다음 예에서는 단일 슬라이스 디스크로 사용할 ZFS 볼륨 zdisk0을 만드는 방법을 보여줍니다.

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

볼륨의 크기로 가상 디스크의 크기가 정의됩니다. 이 예에서는 100MB의 가상 디스크를 얻기 위해 100MB의 볼륨을 만듭니다.

2 제어 도메인에서 해당 장치를 ZFS 볼륨으로 내보내고 slice 옵션을 설정하여 볼륨이 단일 슬라이스 디스크로 내보내지도록 합니다.

```
primary# ldm add-vdsdev options=slice /dev/zvol/dsk/ldoms/domain/test/zdisk0 \
zdisk0@primary-vds0
```

3 제어 도메인에서 볼륨을 게스트 도메인에 지정합니다.

다음 예에서는 zdisk0 볼륨을 게스트 도메인 ldg1에 지정하는 방법을 보여줍니다.

```
primary# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```

4 게스트 도메인이 시작되고 Oracle Solaris OS를 실행 중일 때 디스크(예: c0d9)를 나열한 후 해당 디스크에 액세스 가능하고 단일 슬라이스 디스크(s0)인지 확인합니다.

```
ldg1# ls -l /dev/dsk/c0d9s*
/dev/dsk/c0d9s0
/dev/dsk/c0d9s1
/dev/dsk/c0d9s2
/dev/dsk/c0d9s3
/dev/dsk/c0d9s4
/dev/dsk/c0d9s5
/dev/dsk/c0d9s6
/dev/dsk/c0d9s7

```

볼륨 내보내기 및 역호환성

Oracle Solaris 10 5/08 OS 릴리스 이전에는 `slice` 옵션이 존재하지 않았고 볼륨은 단일 슬라이스 디스크로 내보내졌습니다. 볼륨을 가상 디스크로 내보내는 구성이 있고 시스템을 Oracle Solaris 10 5/08 OS로 업그레이드할 경우 이제 볼륨이 단일 슬라이스 디스크 대신 전체 디스크로 내보내집니다. 이전 동작을 유지하고 볼륨을 단일 슬라이스 디스크로 내보내려면 다음 중 하나를 수행해야 합니다.

- Oracle VM Server for SPARC 2.2 소프트웨어에서 `ldm set -vdsdev` 명령을 사용하고 단일 슬라이스 디스크로 내보낼 모든 볼륨에 대해 `slice` 옵션을 설정합니다. 이 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- 서비스 도메인에서 `/etc/system` 파일에 다음 행을 추가합니다.

```
set vds:vd_volume_force_slice = 1
```

주 - 이 조정 가능 변수를 설정하면 모든 볼륨이 단일 슬라이스 디스크로 내보내지고 전체 디스크로는 내보낼 수 없게 됩니다.

백엔드를 내보내는 여러 유형 요약

백엔드	슬라이스 옵션 없음	슬라이스 옵션 설정
디스크(디스크 슬라이스 2)	전체 디스크 ¹	단일 슬라이스 디스크 ²
디스크 슬라이스(슬라이스 2 아님)	단일 슬라이스 디스크 ³	단일 슬라이스 디스크
파일	전체 디스크	단일 슬라이스 디스크
볼륨(ZFS, Solaris Volume Manager, VxVM 등)	전체 디스크	단일 슬라이스 디스크

¹ 전체 디스크로 내보냅니다.

² 슬라이스 2만 내보냅니다.

³ 슬라이스는 항상 단일 슬라이스 디스크로 내보냅니다.

파일 및 디스크 슬라이스를 가상 디스크로 내보내기 위한 지침

이 절에는 파일 및 디스크 슬라이스를 가상 디스크로 내보내기 위한 지침이 포함되어 있습니다.

루프백 파일(lofi) 드라이버 사용

루프백 파일(lofi) 드라이버를 사용하여 파일을 가상 디스크로 내보낼 수 있습니다. 하지만 이렇게 하면 추가 드라이버 층이 추가되고 가상 디스크의 성능이 저하될 수 있습니다. 대신 파일을 전체 디스크나 단일 슬라이스 디스크로 직접 내보낼 수 있습니다. [116 페이지](#) “파일 및 볼륨”을 참조하십시오.

직접 또는 간접적으로 디스크 슬라이스 내보내기

슬라이스를 직접 또는 간접적으로(예: Solaris Volume Manager 볼륨을 통해) 가상 디스크로 내보내려면 `prtvtoc` 명령을 사용하여 슬라이스가 물리적 디스크의 첫번째 블록(블록 0)에서 시작되지 않는지 확인하십시오.

물리적 디스크의 첫번째 블록에서 시작되는 디스크 슬라이스를 직접 또는 간접적으로 내보낼 경우 물리적 디스크의 분할 영역 테이블을 덮어쓰고 해당 디스크의 모든 분할 영역에 액세스하지 못하게 될 수 있습니다.

가상 디스크 다중 경로 구성

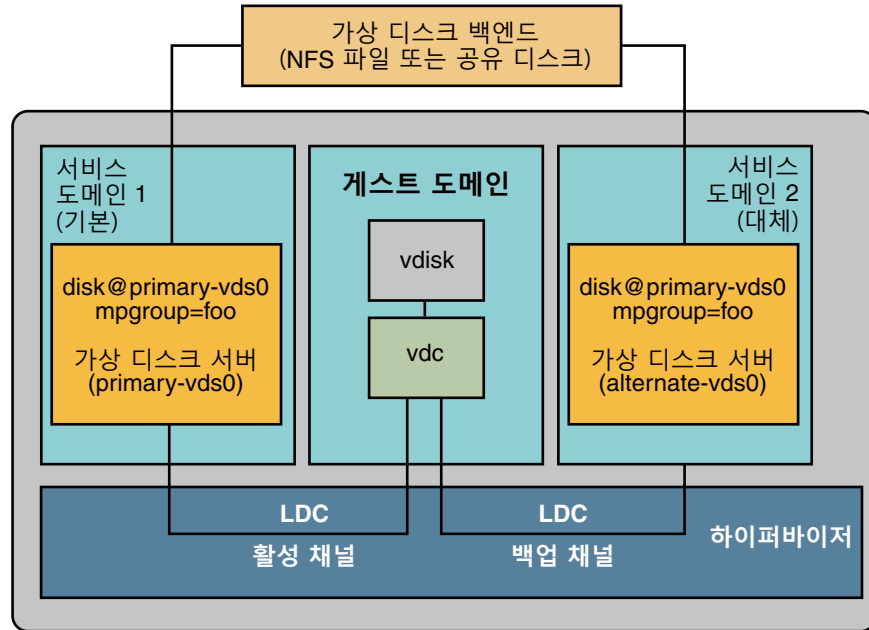
가상 디스크 다중 경로를 통해 여러 경로에서 백엔드 저장소에 액세스하도록 게스트 도메인의 가상 디스크를 구성할 수 있습니다. 경로는 동일 백엔드 저장소(디스크 LUN 등)에 대한 액세스를 제공하는 여러 서비스 도메인을 통해 연결됩니다. 이 기능을 사용하면 서비스 도메인 중 하나가 작동 중지되더라도 게스트 도메인의 가상 디스크에 액세스할 수 있습니다. 예를 들어, 네트워크 파일 시스템(NFS) 서버의 파일에 액세스하도록 가상 디스크 다중 경로 구성을 설정할 수 있습니다. 또는 이 구성을 사용하여 둘 이상의 서비스 도메인에 연결된 공유 저장소에서 LUN에 액세스할 수 있습니다. 따라서 게스트 도메인에서 가상 디스크에 액세스할 경우 가상 디스크 드라이버가 서비스 도메인 중 하나를 통과하여 백엔드 저장소에 액세스하게 됩니다. 가상 디스크 드라이버가 서비스 도메인에 연결할 수 없는 경우 가상 디스크는 다른 서비스 도메인을 통해 백엔드 저장소에 접근하려고 시도합니다.

주 - Oracle VM Server for SPARC 2.0 릴리스부터 가상 디스크 다중 경로 기능에서는 서비스 도메인이 백엔드 저장소에 액세스할 수 없을 때 감지할 수 있습니다. 이러한 경우 가상 디스크 드라이버는 다른 경로로 백엔드 저장소에 액세스를 시도합니다.

가상 디스크 다중 경로를 사용으로 설정하려면 가상 디스크 백엔드를 각 서비스 도메인에서 내보내고 가상 디스크를 동일 다중 경로 그룹(`mpgroup`)에 추가해야 합니다. `mpgroup`은 이름으로 식별되고 가상 디스크 백엔드를 내보낼 때 구성됩니다.

다음 그림은 121 페이지 “가상 디스크 다중 경로를 구성하는 방법” 절차에서 예로 사용된 가상 디스크 다중 경로 구성을 나타냅니다. 이 예에서 `foo` 이름의 다중 경로 그룹이 가상 디스크를 만드는 데 사용되며, 백엔드는 `primary` 및 `alternate`의 두 서비스 도메인에서 액세스할 수 있습니다.

그림 7-2 가상 디스크 다중 경로 구성



가상 디스크 다중 경로 및 가상 디스크 시간 초과

가상 디스크 다중 경로를 사용하면 현재 활성 경로로 백엔드에 액세스할 수 없는 경우 백엔드에 액세스하는 데 사용되는 경로가 자동으로 변경됩니다. 이 경로 변경은 가상 디스크 timeout 등록 정보의 값과 무관하게 발생합니다.

가상 디스크 timeout 등록 정보는 I/O를 처리하는 데 사용할 수 있는 서비스 도메인이 없을 때 I/O가 실패하는 시간을 지정합니다. 이 시간 초과는 가상 디스크 다중 경로를 사용하는 가상 디스크를 포함한 모든 가상 디스크에 적용됩니다.

이에 따라 가상 디스크 다중 경로가 구성되었을 때 가상 디스크 시간 초과를 설정하면 다중 경로가 제대로 작동하지 않을 수 있습니다. 특히, 시간 초과 값이 작을수록 더욱 그러합니다. 따라서 다중 경로 그룹의 일부인 가상 디스크에 대해서는 가상 디스크 시간 초과를 설정하지 않는 것이 좋습니다.

자세한 내용은 [126 페이지](#) “가상 디스크 시간 초과”를 참조하십시오.

▼ 가상 디스크 다중 경로를 구성하는 방법

- 1 primary 서비스도메인에서 가상 디스크 백엔드를 내보냅니다.

```
# ldm add-vdsdev mpgroup=foo backend-path1 volume@primary-vds0
```

여기서 *backend-path1*은 primary 도메인에서 가상 디스크 백엔드로 연결되는 경로입니다.

2 alternate 서비스도메인에서 동일한 가상 디스크 백엔드를 내보냅니다.

```
# ldm add-vdsdev mpgroup=foo backend-path2 volume@alternate-vds0
```

여기서 *backend-path2*는 alternate 도메인에서 가상 디스크 백엔드로 연결되는 경로입니다.

주 - *backend-path1* 및 *backend-path2*는 서로 다른 두 도메인(primary 및 alternate)에서 동일한 가상 디스크 백엔드에 대한 경로입니다. 이러한 경로는 primary 및 alternate 도메인의 구성에 따라 같거나 서로 다를 수 있습니다. *volume* 이름은 사용자 선택입니다. 두 명령에 대해 같거나 서로 다를 수 있습니다.

3 가상 디스크를 게스트 도메인으로 내보냅니다.

```
# ldm add-vdisk disk-name volume@primary-vds0 ldom
```

주 - 가상 디스크 백엔드는 여러 서비스 도메인을 통해 여러 번 내보낼 수 있지만, 하나의 가상 디스크만 게스트 도메인에 지정하고 서비스 도메인을 통해 가상 디스크 백엔드와 연결해야 합니다.

자세한 정보 가상 디스크 다중 경로 결과

다중 경로로 가상 디스크를 구성하고 게스트 도메인을 시작하면 가상 디스크에서 연결된 서비스 도메인(이 예에서는 primary 도메인)을 통해 백엔드에 액세스합니다. 이 서비스 도메인을 사용할 수 없는 경우 가상 디스크는 동일 다중 경로 그룹의 일부인 다른 서비스 도메인을 통해 백엔드에 액세스를 시도합니다.



주의 - 다중 경로 그룹(mpgroup)을 정의할 때 동일 mpgroup의 일부인 가상 디스크 백엔드가 실제로 동일한 가상 디스크 백엔드인지 확인하십시오. 서로 다른 백엔드를 동일 mpgroup에 추가할 경우 예상치 않은 동작이 발생할 수 있으며, 백엔드에 저장된 데이터가 손실되거나 손상될 수 있습니다.

CD, DVD 및 ISO 이미지

가상 디스크를 내보내는 방법과 동일하게 CD(compact disc) 또는 DVD(digital versatile disc)를 내보낼 수 있습니다. CD 또는 DVD를 게스트 도메인으로 내보내려면 CD 또는 DVD 장치의 슬라이스 2를 전체 디스크로 내보냅니다(즉, slice 옵션 없이).

주 - CD 또는 DVD 드라이브 자체를 내보낼 수는 없습니다. CD 또는 DVD 드라이브 안에 있는 CD 또는 DVD만 내보낼 수 있습니다. 따라서 CD 또는 DVD를 내보내려면 드라이브 안에 CD 또는 DVD가 있어야 합니다. 또한 CD 또는 DVD를 내보낼 수 있으려면 해당 CD 또는 DVD가 서비스 도메인에서 사용 중이 아니어야 합니다. 특히, 볼륨 관리 파일 시스템 `volfs` 서비스에서 CD 또는 DVD를 사용하지 않아야 합니다. `volfs`에서 사용 중인 장치를 제거하는 방법에 대한 자세한 내용은 [123 페이지 “CD 또는 DVD를 서비스 도메인에서 게스트 도메인으로 내보내는 방법”](#)을 참조하십시오.

파일이나 볼륨에 저장된 CD 또는 DVD의 ISO(International Organization for Standardization) 이미지가 있고 해당 파일이나 볼륨을 전체 디스크로 내보낼 경우 게스트 도메인에서는 CD 또는 DVD로 나타납니다.

CD, DVD 또는 ISO 이미지를 내보낼 경우 게스트 도메인에서는 자동으로 읽기 전용 장치로 나타납니다. 하지만 게스트 도메인에서 CD 제어 작업(즉, 게스트 도메인에서 CD를 시작, 중지 또는 꺼내기)을 수행할 수는 없습니다. 내보낸 CD, DVD 또는 ISO 이미지가 부트 가능한 경우 게스트 도메인을 해당 가상 디스크에서 부트할 수 있습니다.

예를 들어, Oracle Solaris OS 설치 DVD를 내보낼 경우 DVD에 해당하는 가상 디스크에서 게스트 도메인을 부트하고 해당 DVD에서 게스트 도메인을 설치할 수 있습니다. 이렇게 하려면 게스트 도메인에서 `ok` 프롬프트가 나타날 때 다음 명령을 사용해야 합니다.

```
ok boot /virtual-devices@100/channel-devices@200/disk@n:f
```

여기서 `n`은 내보낸 DVD를 나타내는 가상 디스크의 인덱스입니다.

주 - Oracle Solaris OS 설치 DVD를 내보내고 DVD에 해당하는 가상 디스크에서 게스트 도메인을 부트하여 게스트 도메인을 설치할 경우 설치 중에는 DVD를 변경할 수 없습니다. 따라서 다른 CD/DVD를 요구하는 설치 단계를 건너뛰거나 요청된 매체에 액세스하기 위한 대체 경로를 제공해야 할 수 있습니다.

▼ CD 또는 DVD를 서비스 도메인에서 게스트 도메인으로 내보내는 방법

- 1 서비스 도메인에서 볼륨 관리 데몬 `vol`가 실행 중이고 온라인 상태인지 확인합니다.

```
service# svcs volfs
STATE      STIME      FMRI
online     12:28:12   svc:/system/filesystem/volfs:default
```

- 2 다음 중 하나를 수행합니다.

- 볼륨 관리 데몬이 실행 중이 아니거나 온라인 상태가 아닌 경우 단계 3으로 이동합니다.

- 볼륨 관리 데몬이 실행 중이고 온라인 상태인 경우 단계 1의 예에 나온 대로 다음을 수행합니다.

a. `/etc/vold.conf` 파일을 편집하고 다음 단어로 시작되는 행을 주석 처리합니다.

```
use cdrom drive...
```

vold.conf(4) 매뉴얼 페이지를 참조하십시오.

b. CD 또는 DVD 드라이브에 CD 또는 DVD를 넣습니다.

c. 서비스도메인에서 볼륨 관리 파일 시스템을 다시 시작합니다.

```
service# svcadm refresh volfs
service# svcadm restart volfs
```

3 서비스도메인에서 CD-ROM 장치에 대한 디스크 경로를 찾습니다.

```
service# cdrw -l
Looking for CD devices...
  Node                               Connected Device                               Device type
-----+-----+-----+-----+-----+-----+-----+-----+-----+
/dev/rdsk/clt0d0s2 | MATSHITA CD-RW  CW-8124  DZ13 | CD Reader/Writer
```

4 CD 또는 DVD 디스크 장치를 전체 디스크로 내보냅니다.

```
primary# ldm add-vdsdev /dev/dsk/c1t0d0s2 cdrom@primary-vds0
```

5 내보낸 CD 또는 DVD를 게스트 도메인에 지정합니다.

다음은 내보낸 CD 또는 DVD를 도메인 ldg1에 지정하는 방법을 보여줍니다.

```
primary# ldm add-vdisk cdrom cdrom@primary-vds0 ldg1
```

자세한 정보 CD 또는 DVD 여러 번 내보내기

CD 또는 DVD를 여러 번 내보내고 서로 다른 게스트 도메인에 지정할 수 있습니다.

자세한 내용은 109 페이지 “가상 디스크 백엔드를 여러 번 내보내는 방법”을 참조하십시오.

▼ primary 도메인에서 ISO 이미지를 내보내서 게스트 도메인을 설치하는 방법

이 절차에서는 **primary** 도메인에서 ISO 이미지를 내보내고 해당 이미지를 사용하여 게스트 도메인을 설치하는 방법을 설명합니다. 이 절차에서는 **primary** 도메인과 게스트 도메인이 모두 구성되었다고 가정합니다.

예를 들어, 다음 `ldm list`는 `primary` 및 `ldom1` 도메인이 모두 구성되었음을 나타냅니다.

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
------	-------	-------	------	------	--------	------	--------

```
primary      active -n-cv SP      4      4G      0.3% 15m
ldom1        active -t--- 5000   4      1G      25%  8m
```

1 ISO 이미지를 내보낼 가상 디스크 서버 장치를 추가합니다.

이 예에서 ISO 이미지는 /export/images/sol-10-u8-ga-sparc-dvd.iso입니다.

```
# ldm add-vdsdev /export/images/sol-10-u8-ga-sparc-dvd.iso dvd-iso@primary-vds0
```

2 게스트 도메인을 중지합니다.

이 예에서 논리적 도메인은 ldom1입니다.

```
# ldm stop-domain ldom1
LDom ldom1 stopped
```

3 ISO 이미지에 대한 가상 디스크를 논리적 도메인에 추가합니다.

이 예에서 논리적 도메인은 ldom1입니다.

```
# ldm add-vdisk s10-dvd dvd-iso@primary-vds0 ldom1
```

4 게스트 도메인을 다시 시작합니다.

이 예에서 논리적 도메인은 ldom1입니다.

```
# ldm start-domain ldom1
LDom ldom1 started
# ldm list
NAME          STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active  -n-cv    SP      4      4G      0.4%   25m
ldom1         active  -t---    5000    4      1G      0.0%    0s
```

이 예에서 ldm list 명령은 ldom1 도메인이 방금 시작되었음을 보여줍니다.

5 게스트 도메인에 연결합니다.

```
# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
```

```
Connecting to console "ldom1" in group "ldom1" ....
Press ~? for control options ..
```

6 ISO 이미지가 가상 디스크로 존재하는지 확인합니다.

```
{0} ok show-disks
a) /virtual-devices@100/channel-devices@200/disk@1
b) /virtual-devices@100/channel-devices@200/disk@0
q) NO SELECTION
Enter Selection, q to quit: q
```

이 예에서 새로 추가된 장치는

/virtual-devices@100/channel-devices@200/disk@1입니다.

7 게스트 도메인을 부트하여 ISO 이미지에서 설치합니다.

이 예에서는 /virtual-devices@100/channel-devices@200/disk@1 디스크의 f 슬라이스에서 부트합니다.

```
{0} ok boot /virtual-devices@100/channel-devices@200/disk@1:f
```

가상 디스크 시간 초과

기본적으로 가상 디스크 백엔드에 대한 액세스를 제공하는 서비스 도메인이 작동 중지될 경우 게스트 도메인에서 해당 가상 디스크로의 모든 I/O가 차단됩니다. 서비스 도메인이 작동하고 가상 디스크 백엔드에 I/O 요청을 서비스하면 I/O는 자동으로 재개됩니다.

하지만 서비스 도메인이 오랫동안 작동 중지 상태일 경우 파일 시스템이나 응용 프로그램에서 I/O 작업이 차단되지 않고 실패하여 오류를 보고하려는 경우가 있습니다. 이제 각 가상 디스크에 대한 연결 시간 초과 기간을 설정하여 게스트 도메인의 가상 디스크 클라이언트와 서비스 도메인의 가상 디스크 서버간 연결을 설정하는 데 사용할 수 있습니다. 시간 초과 기간에 도달하면 서비스 도메인이 작동 중지되고 가상 디스크 클라이언트와 서버간 연결이 다시 설정되지 않는 동안 보류 중인 모든 I/O 및 새로운 I/O는 실패합니다.

이 시간 초과는 다음 중 하나를 수행하여 설정할 수 있습니다.

- `ldm add-vdisk` 명령 사용.

```
ldm add-vdisk timeout=seconds disk-name volume-name@service-name ldom
```

- `ldm set-vdisk` 명령 사용.

```
ldm set-vdisk timeout=seconds disk-name ldom
```

시간 초과는 초 단위로 지정합니다. 시간 초과가 0으로 설정되면 시간 초과는 사용 안함으로 설정되고 서비스 도메인이 작동 중지된 동안 I/O가 차단됩니다(기본 설정 및 동작).

또는 게스트 시스템에서 `/etc/system` 파일에 다음 행을 추가하여 시간 초과를 설정할 수도 있습니다.

```
set vdc:vdc_timeout=seconds
```

주 - 이 조정 가능 변수가 설정될 경우 `ldm CLI`를 사용하여 수행된 모든 시간 초과 설정을 덮어씁니다. 또한 조정 가능 변수는 게스트 도메인의 모든 가상 디스크에 대한 시간 초과를 설정합니다.

가상 디스크 및 SCSI

물리적 SCSI 디스크 또는 LUN을 전체 디스크로 내보낼 경우 해당하는 가상 디스크는 사용자 SCSI 명령 인터페이스 `uscsi` 및 다중 호스트 디스크 제어 작업 `mhd`를 지원합니다. 파일이나 볼륨을 백엔드로 가지는 가상 디스크와 같은 다른 가상 디스크는 이러한 인터페이스를 지원하지 않습니다.

결과적으로 SCSI 명령을 사용하는 응용 프로그램이나 제품 기능(예: Solaris Volume Manager `metaset` 또는 Oracle Solaris Cluster `shared devices`)은 물리적 SCSI 디스크를 백엔드로 가지는 가상 디스크가 있는 게스트 도메인에서만 사용할 수 있습니다.

주 - SCSI 작업은 사실상 가상 디스크 백엔드로 사용되는 물리적 SCSI 디스크 또는 LUN을 관리하는 서비스 도메인에 의해 실행됩니다. 특히, SCSI 예약은 서비스 도메인에 의해 수행됩니다. 따라서 서비스 도메인 및 게스트 도메인에서 실행 중인 응용 프로그램은 동일한 물리적 SCSI 디스크에 SCSI 명령을 실행하면 안 됩니다. 그렇지 않으면 예상치 않은 디스크 상태로 이어질 수 있습니다.

가상 디스크 및 format 명령

`format` 명령은 도메인에 존재하는 모든 가상 디스크를 인식합니다. 하지만 단일 슬라이스 디스크로 내보낸 가상 디스크의 경우 `format` 명령은 가상 디스크의 분할 영역 테이블을 변경할 수 없습니다. 가상 디스크와 이미 연결된 디스크 레이블과 유사한 레이블을 작성하려고 시도하지 않으면 `label`과 같은 명령이 실패합니다.

백엔드가 SCSI 디스크인 가상 디스크는 모든 `format(1M)` 하위 명령을 지원합니다. 백엔드가 SCSI 디스크가 아닌 가상 디스크는 일부 `format(1M)` 하위 명령(`repair` 및 `defect` 등)을 지원하지 않습니다. 이 경우 `format(1M)`의 동작은 IDE(Integrated Drive Electronics) 디스크의 동작과 유사합니다.

가상 디스크에서 ZFS 사용

이 절에서는 ZFS(Zettabyte File System)를 사용하여 게스트 도메인으로 내보낸 가상 디스크 백엔드를 저장하는 방법을 설명합니다. ZFS는 가상 디스크 백엔드를 만들고 관리할 수 있는 편리하면서 강력한 솔루션을 제공합니다. ZFS를 통해 다음이 가능합니다.

- ZFS 볼륨 또는 ZFS 파일에 디스크 이미지 저장
- 스냅샷을 사용하여 디스크 이미지 백업
- 복제본을 사용하여 디스크 이미지 복제 및 추가 도메인 프로비전

ZFS 사용에 대한 자세한 내용은 [Oracle Solaris ZFS 관리 설명서](#)를 참조하십시오.

다음 설명 및 예에서는 기본 도메인이 디스크 이미지가 저장되는 서비스 도메인이기도 합니다.

서비스 도메인에서 ZFS 풀 구성

디스크 이미지를 저장하려면 먼저 서비스 도메인에서 ZFS 저장소 풀을 만듭니다. 예를 들어, 이 명령은 primary 도메인에서 c1t50d0 디스크를 포함하는 ZFS 저장소 풀 ldmpool을 만듭니다.

```
primary# zpool create ldmpool c1t50d0
```

ZFS를 사용하여 디스크 이미지 저장

다음 명령은 게스트 도메인 ldg1에 대한 디스크 이미지를 만듭니다. 이 게스트 도메인에 대한 ZFS 파일 시스템이 만들어지고, 이 게스트 도메인의 모든 디스크 이미지가 해당 파일 시스템에 저장됩니다.

```
primary# zfs create ldmpool/ldg1
```

디스크 이미지는 ZFS 볼륨이나 ZFS 파일에 저장할 수 있습니다. zfs create -V 명령을 사용하면 크기에 상관 없이 ZFS 볼륨이 빠르게 만들어집니다. 반면 ZFS 파일은 mkfile 명령을 사용하여 만들어야 합니다. 특히, 만들 파일이 큰 경우(대개 디스크 이미지를 만드는 경우) 명령이 완료되려면 어느 정도 시간이 소요될 수 있습니다.

ZFS 볼륨과 ZFS 파일은 스냅샷 및 복제본 기능과 같은 ZFS 기능을 활용하지만, ZFS 볼륨은 의사 장치이고 ZFS 파일은 일반 파일입니다.

디스크 이미지가 OS가 설치되는 가상 디스크로 사용될 경우 디스크 이미지는 OS 설치 요구 사항을 충족할 만큼 커야 합니다. 이 크기는 OS의 버전 및 수행되는 설치 유형에 따라 달라집니다. Oracle Solaris OS를 설치하는 경우 20GB의 디스크 크기를 사용하면 Oracle Solaris OS 모든 버전의 모든 유형 설치가 가능합니다.

예: ZFS를 사용하여 디스크 이미지 저장

예는 다음과 같습니다.

1. ZFS 볼륨이나 파일에서 20GB 이미지를 만듭니다.
2. ZFS 볼륨이나 파일을 가상 디스크로 내보냅니다. ZFS 볼륨이나 파일을 내보내기 위한 구문은 동일하지만 백엔드에 대한 경로는 서로 다릅니다.
3. 내보낸 ZFS 볼륨이나 파일을 게스트 도메인에 지정합니다.

게스트 도메인이 만들어지면 ZFS 볼륨이나 파일은 Oracle Solaris OS를 설치할 수 있는 가상 디스크로 나타납니다.

▼ ZFS 볼륨을 사용하여 디스크 이미지를 만드는 방법

- 예를 들어, ZFS 볼륨에서 20GB 디스크 이미지를 만듭니다.

```
primary# zfs create -V 20gb ldmpool/ldg1/disk0
```

▼ ZFS 파일을 사용하여 디스크 이미지를 만드는 방법

- 예를 들어, ZFS 볼륨에서 20GB 디스크 이미지를 만듭니다.

```
primary# zfs create ldmpool/ldg1/disk0
primary# mkfile 20g /ldmpool/ldg1/disk0/file
```

▼ ZFS 볼륨을 내보내는 방법

- ZFS 볼륨을 가상 디스크로 내보냅니다.

```
primary# ldm add-vdsdev /dev/zvol/dsk/ldmpool/ldg1/disk0 ldg1_disk0@primary-vds0
```

▼ ZFS 파일을 내보내는 방법

- ZFS 파일을 가상 디스크로 내보냅니다.

```
primary# ldm add-vdsdev /ldmpool/ldg1/disk0/file ldg1_disk0@primary-vds0
```

▼ ZFS 볼륨이나 파일을 게스트 도메인에 지정하는 방법

- ZFS 볼륨이나 파일을 게스트 도메인(이 예에서는 ldg1)에 지정합니다.

```
primary# ldm add-vdisk disk0 ldg1_disk0@primary-vds0 ldg1
```

디스크 이미지의 스냅샷 만들기

디스크 이미지가 ZFS 볼륨이나 ZFS 파일에 저장된 경우 ZFS 스냅샷 명령을 사용하여 이 디스크 이미지의 스냅샷을 만들 수 있습니다.

디스크 이미지의 스냅샷을 만들기 전에 디스크가 현재 게스트 도메인에서 사용되고 있지 않은지 확인하여 디스크 이미지에 현재 저장된 데이터가 일관성을 유지하도록 합니다. 디스크가 게스트 도메인에서 사용되고 있지 않은지 확인하는 방법에는 여러 가지가 있습니다. 다음 중 하나를 수행할 수 있습니다.

- 게스트 도메인을 중지하고 바인드 해제합니다. 이 방법은 가장 안전한 방법이며, 게스트 도메인의 부트 디스크로 사용되는 디스크 이미지의 스냅샷을 만들려는 경우 사용할 수 있는 유일한 방법입니다.
- 또는 게스트 도메인에서 사용되는 스냅샷을 만들 디스크의 모든 슬라이스를 마운트 해제하고 게스트 도메인에서 사용되는 슬라이스가 없도록 할 수 있습니다.

이 예에서는 ZFS 레이아웃으로 인해 디스크 이미지가 ZFS 볼륨 또는 ZFS 파일에 저장되어 있는지 상관 없이 디스크 이미지의 스냅샷을 만들기 위한 명령이 동일합니다.

▼ 디스크 이미지의 스냅샷을 만드는 방법

- 예를 들어, `ldg1` 도메인에 대해 만들어진 디스크 이미지의 스냅샷을 만듭니다.

```
primary# zfs snapshot ldmpool/ldg1/disk0@version_1
```

복제본을 사용하여 새 도메인 프로비전

디스크 이미지의 스냅샷을 만들었으면 ZFS 복제본 명령을 사용하여 이 디스크 이미지를 복제할 수 있습니다. 그런 다음 복제된 이미지를 다른 도메인에 지정할 수 있습니다. 부트 디스크 이미지 복제는 전체 Oracle Solaris OS 설치 프로세스를 수행할 필요 없이 새로운 게스트 도메인에 대한 부트 디스크를 빠르게 만듭니다.

예를 들어, 만들어진 `disk0`이 `ldg1` 도메인의 부트 디스크인 경우 다음을 수행하여 해당 디스크를 복제하고 `ldg2` 도메인에 대한 부트 디스크를 만듭니다.

```
primary# zfs create ldmpool/ldg2
primary# zfs clone ldmpool/ldg1/disk0@version_1 ldmpool/ldg2/disk0
```

그런 다음 `ldmpool/ldg2/disk0`을 가상 디스크로 내보내고 새로운 `ldg2` 도메인에 지정할 수 있습니다. `ldg2` 도메인은 OS 설치 프로세스를 거칠 필요 없이 가상 디스크에서 직접 부트할 수 있습니다.

부트 디스크 이미지 복제

부트 디스크 이미지가 복제되면 새로운 이미지는 원본 부트 디스크와 완전히 동일하며, 이미지가 복제되기 전 부트 디스크에 저장된 모든 정보(호스트 이름, IP 주소, 마운트된 파일 시스템 테이블, 모든 시스템 구성 또는 조정 등)가 포함됩니다.

마운트된 파일 시스템 테이블은 원본 부트 디스크 이미지와 복제된 디스크 이미지에서 동일하므로 복제된 디스크 이미지를 원래 도메인에 있었던 같은 순서로 새로운 도메인에 지정해야 합니다. 예를 들어, 부트 디스크 이미지가 원래 도메인의 첫번째 디스크로 지정된 경우 복제된 디스크 이미지도 새로운 도메인의 첫번째 디스크로 지정해야 합니다. 그렇지 않으면 새로운 도메인이 부트되지 않습니다.

원래 도메인이 정적 IP 주소로 구성된 경우 복제된 이미지를 사용하는 새로운 도메인도 동일한 IP 주소로 시작됩니다. 이 경우 `sys-unconfig` 명령을 사용하여 새로운 도메인의 네트워크 구성을 변경해야 합니다. 이 문제를 피하려면 구성되지 않은 시스템의 디스크 이미지 스냅샷을 만들 수도 있습니다.

원래 도메인이 DHCP(Dynamic Host Configuration Protocol)로 구성된 경우 복제된 이미지를 사용하는 새로운 도메인도 DHCP를 사용합니다. 이 경우 새로운 도메인이 부트될 때 IP 주소 및 네트워크 구성을 자동으로 가져오므로 새로운 도메인의 네트워크 구성을 변경할 필요가 없습니다.

주 - 도메인의 호스트 ID는 부트 디스크에 저장되지 않지만, 도메인을 만들 때 Logical Domains Manager에 의해 지정됩니다. 따라서 디스크 이미지를 복제할 경우 새로운 도메인은 원래 도메인의 호스트 ID를 유지하지 않습니다.

▼ 구성되지 않은 시스템의 디스크 이미지 스냅샷을 만드는 방법

- 1 원래 도메인을 바인드하고 시작합니다.
- 2 `sys-unconfig` 명령을 실행합니다.
- 3 `sys-unconfig` 명령이 완료되면 도메인이 멈춥니다.
- 4 도메인을 중지하고 바인드 해제합니다. 재부트하지는 마십시오.

- 5 도메인 부트 디스크 이미지의 스냅샷을 만듭니다.

예를 들어, 다음과 같습니다.

```
primary# zfs snapshot ldmpool/ldg1/disk0@unconfigured
```

이 시점에 구성되지 않은 시스템의 부트 디스크 이미지 스냅샷을 얻게 됩니다.

- 6 이 이미지를 복제하여 새로운 도메인을 만듭니다. 그러면 처음 부트할 때 시스템의 구성을 물어봅니다.

Logical Domains 환경에서 볼륨 관리자 사용

이 절에서는 Logical Domains 환경에서 볼륨 관리자를 사용하는 방법에 대해 설명합니다.

볼륨 관리자 위에서 가상 디스크 사용

ZFS(Zettabyte File System), Solaris Volume Manager 또는 VxVM(Veritas Volume Manager) 볼륨은 서비스 도메인에서 게스트 도메인에 가상 디스크로 내보낼 수 있습니다. 볼륨은 단일 슬라이스 디스크(slice 옵션이 `ldm add -vdsdev` 명령과 함께 지정된 경우) 또는 전체 디스크로 내보낼 수 있습니다.

주 - 이 절의 나머지 부분에서는 Solaris Volume Manager 볼륨을 예로 사용합니다. 하지만 설명은 ZFS 및 VxVM 볼륨에도 해당됩니다.

다음 예에서는 볼륨을 단일 슬라이스 디스크로 내보내는 방법을 보여줍니다.

게스트 도메인의 가상 디스크(예: `/dev/dsk/c0d2s0`)는 연결된 볼륨(예: `/dev/md/dsk/d0`)에 직접 매핑되고, 게스트 도메인에서 가상 디스크에 저장되는 데이터는 추가 메타 데이터 없이 연결된 볼륨에 직접 저장됩니다. 따라서 게스트 도메인에서 가상 디스크에 저장된 데이터는 연결된 볼륨을 통해 서비스 도메인에서 직접 액세스할 수도 있습니다.

예

- Solaris Volume Manager 볼륨 `d0`을 primary 도메인에서 `domain1`로 내보낸 경우 `domain1`의 구성을 위해서는 일부 추가 단계가 필요합니다.

```
primary# metainit d0 3 1 c2t70d0s6 1 c2t80d0s6 1 c2t90d0s6
primary# ldm add-vdsdev options=slice /dev/md/dsk/d0 vol3@primary-vds0
primary# ldm add-vdisk vdisk3 vol3@primary-vds0 domain1
```

- `domain1`이 바인드되고 시작된 후 내보낸 볼륨은 `/dev/dsk/c0d2s0`(예)으로 나타나고 사용할 수 있습니다.

```
domain1# newfs /dev/rdisk/c0d2s0
domain1# mount /dev/dsk/c0d2s0 /mnt
domain1# echo test-domain1 > /mnt/file
```

- `domain1`이 중지되고 바인드 해제된 후 `domain1`에서 가상 디스크에 저장된 데이터는 Solaris Volume Manager 볼륨 `d0`을 통해 기본 도메인에서 직접 액세스할 수 있습니다.

```
primary# mount /dev/md/dsk/d0 /mnt
primary# cat /mnt/file
test-domain1
```

Solaris Volume Manager 위에서 가상 디스크 사용

RAID 또는 미리 Solaris Volume Manager 볼륨이 다른 도메인에서 가상 디스크로 사용되는 경우 배타적(`excl`) 옵션 설정 없이 내보내야 합니다. 그렇지 않으면 Solaris Volume Manager 볼륨 구성 요소 중 하나에서 오류가 발생할 경우 `metareplace` 명령 또는 핫 스왑어를 사용한 Solaris Volume Manager 볼륨 복구가 시작되지 않습니다. `metastat` 명령은 볼륨을 재동기화 중으로 인식하지만 재동기화가 진행되지 않습니다.

예를 들어, `/dev/md/dsk/d0`은 `excl` 옵션을 사용하여 다른 도메인에 가상 디스크로 내보낸 RAID Solaris Volume Manager 볼륨이고, `d0`은 몇 가지 핫 스왑어 장치로 구성되어 있습니다. `d0`의 구성 요소가 실패할 경우 Solaris Volume Manager는 실패한 구성 요소를 핫 스왑어로 교체하고 Solaris Volume Manager 볼륨을 재동기화합니다. 하지만 재동기화가 시작되지 않습니다. 볼륨이 재동기화 중으로 보고되지만 재동기화가 진행되지 않습니다.

```
# metastat d0
d0: RAID
    State: Resyncing
    Hot spare pool: hsp000
    Interlace: 32 blocks
    Size: 20097600 blocks (9.6 GB)
Original device:
    Size: 20100992 blocks (9.6 GB)
```

Device	Start Block	Dbase	State	Reloc
c2t2d0s1	330	No	Okay	Yes
c4t12d0s1	330	No	Okay	Yes
/dev/dsk/c10t600C0FF0000000000015153295A4B100d0s1	330	No	Resyncing	Yes

이러한 경우 Solaris Volume Manager 볼륨을 가상 디스크로 사용하는 도메인을 중지하고 바인드 해제하여 재동기화를 완료해야 합니다. 그런 다음 `metasync` 명령을 사용하여 Solaris Volume Manager 볼륨을 재동기화할 수 있습니다.

```
# metasync d0
```

VxVM이 설치된 경우 가상 디스크 사용

VxVM(Veritas Volume Manager)이 시스템에 설치되어 있고 DMP(Veritas Dynamic Multipathing)가 가상 디스크로 내보낼 물리적 디스크 또는 분할 영역에서 사용으로 설정된 경우 `excl` 옵션 설정 없이(기본값 아님) 해당 디스크 또는 분할 영역을 내보내야 합니다. 그렇지 않으면 해당 디스크를 사용하는 도메인을 바인드할 때 `/var/adm/messages`에 오류가 나타납니다.

```
vd_setup_vd(): ldi_open_by_name(/dev/dsk/c4t12d0s2) = errno 16
vds_add_vd(): Failed to add vdisk ID 0
```

`vxdisklist` 명령의 출력에서 다중 경로 정보를 확인하여 Veritas DMP가 사용으로 설정되어 있는지 확인할 수 있습니다. 예를 들면 다음과 같습니다.

```
# vxdisk list Disk_3
Device:      Disk_3
devicetag:   Disk_3
type:        auto
info:        format=none
flags:       online ready private autoconfig invalid
pubpaths:    block=/dev/vx/dmp/Disk_3s2 char=/dev/vx/rdmp/Disk_3s2
guid:        -
udid:        SEAGATE%5FST336753LSUN36G%5FDISKS%5F3032333948303144304E0000
site:        -
Multipathing information:
numpaths:    1
c4t12d0s2   state=enabled
```

또는 `excl` 옵션이 설정된 상태에서 가상 디스크로 내보낼 디스크 또는 슬라이스에서 Veritas DMP가 사용으로 설정된 경우 `vxddmpadm` 명령을 사용하여 DMP를 사용 안함으로 설정할 수 있습니다. 예를 들어, 다음과 같습니다.

```
# vxddmpadm -f disable path=/dev/dsk/c4t12d0s2
```

가상 디스크 위에서 볼륨 관리자 사용

이 절에서는 가상 디스크 위에서 볼륨 관리자를 사용하는 방법에 대해 설명합니다.

가상 디스크 위에서 ZFS 사용

모든 가상 디스크는 ZFS로 사용할 수 있습니다. ZFS 저장소 풀(pool)은 이 zpool의 일부인 저장소 장치를 인식(도메인에서 이러한 장치를 가상 장치 또는 실제 장치로 인식하는지 상관 없이)하는 모든 도메인에서 가져올 수 있습니다.

가상 디스크 위에서 Solaris Volume Manager 사용

모든 가상 디스크는 Solaris Volume Manager 로컬 디스크 세트에서 사용할 수 있습니다. 예를 들어, 가상 디스크는 로컬 디스크 세트의 Solaris Volume Manager 메타 장치 상태 데이터베이스 metadb를 저장하고 로컬 디스크 세트에서 Solaris Volume Manager 볼륨을 만드는 데 사용할 수 있습니다.

백엔드가 SCSI 디스크인 모든 가상 디스크는 Solaris Volume Manager 공유 디스크 세트 metaset에서 사용할 수 있습니다. 백엔드가 SCSI 디스크가 아닌 가상 디스크는 Solaris Volume Manager 공유 디스크 세트에 추가할 수 없습니다. 백엔드가 SCSI 디스크가 아닌 가상 디스크를 Solaris Volume Manager 공유 디스크 세트에 추가하려고 시도하면 다음과 같은 오류와 함께 실패합니다.

```
# metaset -s test -a c2d2
metaset: domain1: test: failed to reserve any drives
```

가상 디스크 위에서 VxVM 사용

게스트 도메인에서 VxVM 지원은 Symantec의 VxVM 설명서를 참조하십시오.

가상 네트워크 사용

이 장은 Oracle VM Server for SPARC 소프트웨어로 가상 네트워크를 사용하는 방법에 대해 설명하며, 다음과 같은 항목으로 구성됩니다.

- 136 페이지 “가상 네트워크에 대한 소개”
- 136 페이지 “Oracle Solaris 10 네트워킹 개요”
- 138 페이지 “Oracle Solaris 11 네트워킹 개요”
- 141 페이지 “가상 스위치”
- 142 페이지 “가상 네트워크 장치”
- 144 페이지 “가상 장치 식별자 및 네트워크 인터페이스 이름”
- 146 페이지 “자동 또는 수동 MAC 주소 지정”
- 149 페이지 “Logical Domains에서 네트워크 어댑터 사용”
- 150 페이지 “NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성”
- 154 페이지 “Logical Domains 환경에서 IPMP 구성”
- 162 페이지 “VLAN 태그 지정 사용”
- 165 페이지 “NIU 하이브리드 I/O 사용”
- 169 페이지 “가상 스위치에서 링크 통합 사용”
- 171 페이지 “정보 프레임 구성”
- 175 페이지 “Oracle Solaris 11 네트워킹 관련 기능 차이점”

Oracle Solaris OS 네트워킹은 Oracle Solaris 10 OS와 Oracle Solaris 11 OS 간에 상당히 다릅니다. 고려할 문제에 대한 자세한 내용은 136 페이지 “Oracle Solaris 10 네트워킹 개요”, 138 페이지 “Oracle Solaris 11 네트워킹 개요” 및 175 페이지 “Oracle Solaris 11 네트워킹 관련 기능 차이점”을 참조하십시오.

가상 네트워크에 대한 소개

가상 네트워크는 외부 물리적 네트워크를 사용하지 않고도 도메인이 서로 통신할 수 있도록 해줍니다. 또한 도메인이 동일한 물리적 네트워크 인터페이스를 사용하여 물리적 네트워크에 액세스하고 원격 시스템과 통신할 수 있도록 해줍니다. 가상 네트워크 장치를 연결할 수 있는 가상 스위치를 생성하면 가상 네트워크가 만들어집니다.

Oracle Solaris 네트워킹은 Oracle Solaris 10 OS와 Oracle Solaris 11 OS 간에 상당히 다릅니다. 다음 두 단원에서는 OS별 네트워킹에 대한 개요 정보를 다룹니다.

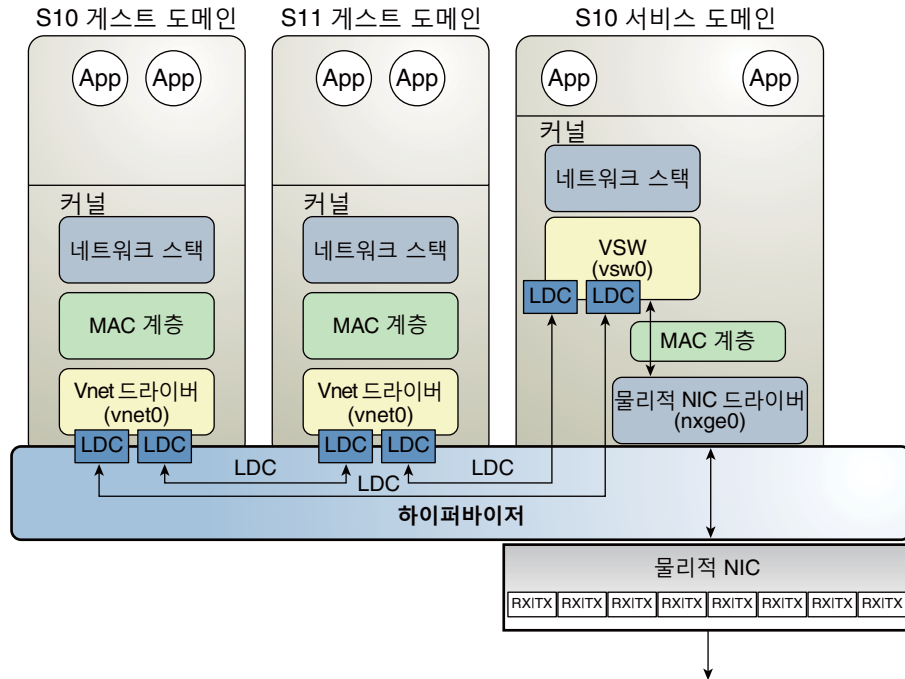
주 - Oracle Solaris 10 네트워킹은 도메인 또는 시스템에서와 동일하게 작동합니다. Oracle Solaris 11 네트워킹의 경우도 마찬가지입니다. Oracle Solaris OS 네트워킹에 대한 자세한 내용은 [Oracle Solaris 10 Documentation](#) 및 [Oracle Solaris 11 Documentation](#)을 참조하십시오.

Oracle Solaris 10 및 Oracle Solaris 11 네트워킹 간 기능 차이점은 [138 페이지 “Oracle Solaris 11 네트워킹 개요”](#)에 설명되어 있습니다.

Oracle Solaris 10 네트워킹 개요

다음 다이어그램은 Oracle Solaris 11 OS를 실행하는 게스트 도메인이 Oracle Solaris 10 서비스 도메인과 완전하게 호환됨을 보여줍니다. 단, Oracle Solaris 11 OS에서 기능이 추가되었거나 향상되었다는 점이 다릅니다.

그림 8-1 Oracle Solaris 10 OS에 대한 Oracle VM Server for SPARC 네트워크 개요



다음은 `nxge0`, `vsw0`, `vnet0` 등의 인터페이스 이름이 표시된 위 다이어그램에 대한 설명입니다. 이러한 인터페이스 이름은 Oracle Solaris 10 OS에만 적용됩니다.

- 서비스 도메인의 가상 스위치는 게스트 도메인이 서로 통신할 수 있도록 게스트 도메인에 연결됩니다.
- 가상 스위치는 또한 게스트 도메인이 물리적 네트워크와 통신할 수 있도록 물리적 네트워크 인터페이스 `nxge0`에도 연결됩니다.
- 가상 스위치 네트워크 인터페이스 `vsw0`은 두 게스트 도메인이 서비스 도메인과 통신할 수 있도록 서비스 도메인에 생성됩니다.
- 서비스 도메인의 가상 스위치 네트워크 인터페이스 `vsw0`은 Oracle Solaris 10 `ifconfig` 명령을 사용하여 구성할 수 있습니다.
- Oracle Solaris 10 게스트 도메인의 가상 네트워크 장치 `vnet0`은 `ifconfig` 명령을 사용하여 네트워크 인터페이스로 구성할 수 있습니다.
- Oracle Solaris 11 게스트 도메인의 가상 네트워크 장치 `vnet0`은 `net0`과 같은 일반 링크 이름으로 표시될 수 있습니다. 또한 `ipadm` 명령을 사용하여 네트워크 인터페이스로 구성할 수 있습니다.

가상 스위치는 정규 물리적 네트워크 스위치처럼 작동하며, 게스트 도메인, 서비스 도메인, 물리적 네트워크 등 연결되어 있는 여러 시스템 간에 네트워크 패킷을 전환합니다. `vsw` 드라이버는 가상 스위치를 네트워크 인터페이스로 구성할 수 있도록 해주는 네트워크 장치 기능을 제공합니다.

Oracle Solaris 11 네트워킹 개요

Oracle Solaris 11 OS에서는 [Oracle Solaris 11 Documentation](#)의 Oracle Solaris 11 네트워킹 설명서에 설명된 새로운 여러 네트워킹 기능이 도입되었습니다.

Oracle VM Server for SPARC 소프트웨어를 사용할 때 다음과 같은 Oracle Solaris 11 네트워킹 기능을 잘 알고 있는 것이 중요합니다.

- 모든 네트워크 구성은 `ipadm` 및 `dladm` 명령으로 수행됩니다.
- “기본 베니티 이름” 기능은 모든 물리적 네트워크 어댑터에 대해 일반 링크 이름(예: `net0`)을 생성합니다. 이 기능은 또한 OS에서 물리적 네트워크 어댑터처럼 인식되는 가상 스위치(`vsw`) 및 가상 네트워크 장치(`vnetn`)에 대해서도 일반 이름을 생성합니다. 물리적 네트워크 장치와 연관된 일반 링크 이름을 식별하려면 `dladm show-phys` 명령을 사용합니다.

기본적으로 Oracle Solaris 11에서 물리적 네트워크 장치 이름은 일반 “베니티” 이름을 사용합니다. Oracle Solaris 10에서 사용되던 장치 드라이버 이름(예: `nxge0`) 대신 일반 이름(예: `net0`)이 사용됩니다.

가상 스위치에 대한 백엔드 장치로 사용할 네트워크 장치를 확인하려면 `dladm show-phys` 출력에서 `vsw`를 검색합니다.

다음 명령은 드라이버 이름(예: `nxge0`) 대신 일반 이름(예: `net0`)을 지정하여 `primary` 도메인에 대한 가상 스위치를 만듭니다.

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

- Oracle Solaris 11 OS에서는 VNIC(가상 네트워크 인터페이스 카드)를 사용하여 내부 가상 네트워크를 만듭니다.

VNIC는 물리적 네트워크 장치에서 만들어 영역에 지정할 수 있는 물리적 네트워크 장치에 대한 가상 인스턴스화입니다.

주 - 가상 스위치(`vsw`) 및 가상 네트워크 장치(`vnetn`)에 VNIC를 만드는 것은 지원되지 않습니다. **Oracle VM Server for SPARC 2.2 Release Notes**에서 [Oracle VM Server for SPARC 2.2 릴리스 노트](#)의 “Oracle Solaris 11: 자동 네트워크 인터페이스로 구성된 영역을 시작하지 못할 수 있음”를 참조하십시오.

- Oracle VM Server for SPARC 소프트웨어를 구성할 경우 Oracle Solaris 11 DefaultFixed NCP(네트워크 구성 프로파일)를 사용합니다.

주 - 이 릴리스에서는 DefaultFixed NCP를 사용하여 Oracle Solaris 11 시스템에 데이터 링크와 네트워크 인터페이스를 구성하는 것이 가장 좋습니다.

Oracle Solaris 11 OS에 포함된 NCP는 다음과 같습니다.

- DefaultFixed.dladm 또는 ipadm 명령을 사용하여 네트워킹을 관리할 수 있습니다.
- Automatic.netcfg 또는 netadm 명령을 사용하여 네트워킹을 관리할 수 있습니다.

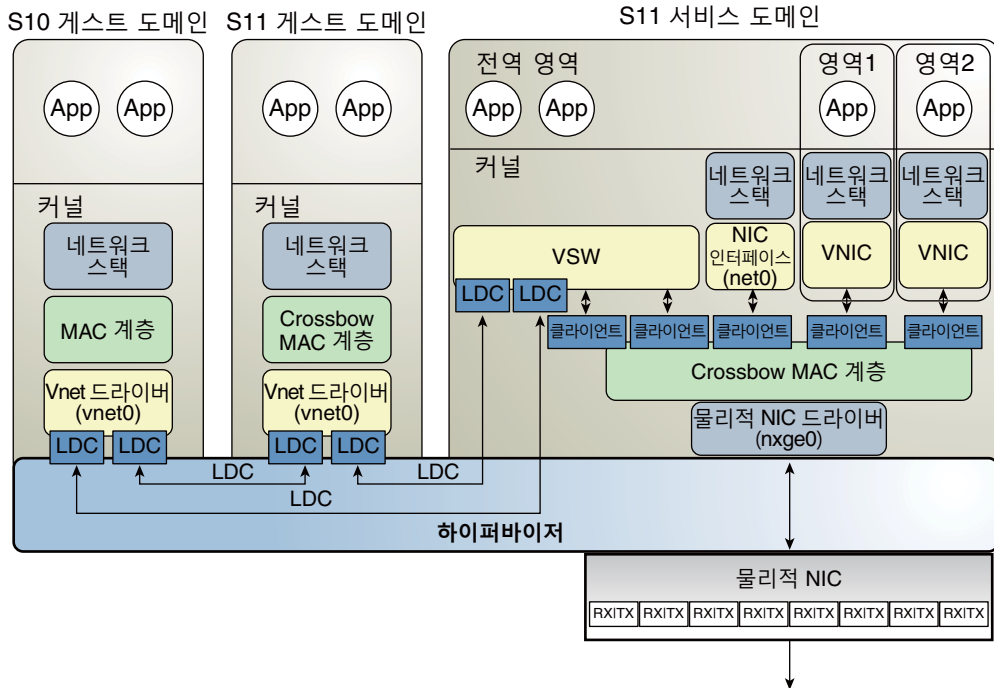
netadm list 명령을 사용하여 DefaultFixed NCP가 사용으로 설정되어 있는지 확인하십시오. **Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 7 장, “프로파일에 데이터 링크 및 인터페이스 구성 명령 사용”**을 참조하십시오.

Oracle Solaris 11 도메인의 경우, DefaultFixed NCP를 사용합니다. 설치 중이나 설치 후에 이 프로파일을 사용으로 설정할 수 있습니다. Oracle Solaris 11 설치 중 수동 네트워킹 구성을 선택합니다.

- 기본 네트워크 인터페이스를 가상 스위치(vsw) 인터페이스로 바꾸지 **마십시오**. 컨트롤 도메인이 기존의 기본 네트워크 인터페이스를 사용하여 동일한 가상 스위치에 가상 네트워크 장치가 연결되어 있는 게스트 도메인과 통신할 수 있습니다.
- 가상 스위치에 물리적 어댑터의 MAC 주소를 사용할 경우 기본 네트워크 인터페이스와 충돌하므로 가상 스위치에 물리적 어댑터의 MAC 주소를 사용하지 **마십시오**.

다음 다이어그램은 Oracle Solaris 10 OS를 실행하는 게스트 도메인이 Oracle Solaris 11 서비스 도메인과 완전하게 호환됨을 보여줍니다. 단, Oracle Solaris 11 OS에서 기능이 추가되었거나 향상되었다는 점이 다릅니다.

그림 8-2 Oracle Solaris 11 OS에 대한 Oracle VM Server for SPARC 네트워크 개요



다음은 Oracle Solaris 11에서 네트워크 장치 이름(예: `nxge0` 및 `vnet0`)을 일반 링크 이름(예: `netn`)으로 바꿀 수 있음을 보여주는 위 다이어그램에 대한 설명입니다.

- 서비스 도메인의 가상 스위치는 게스트 도메인에 연결됩니다. 이렇게 하면 게스트 도메인이 서로 통신할 수 있습니다.
- 가상 스위치는 또한 물리적 네트워크 장치 인터페이스 `nxge0`에도 연결됩니다. 이렇게 하면 게스트 도메인이 물리적 네트워크와 통신할 수 있습니다.

가상 스위치는 또한 게스트 도메인을 통해 서비스 도메인 네트워크 인터페이스 `net0` 및 `nxge0`와 동일한 물리적 네트워크 장치에 있는 **VNIC**와 통신할 수 있습니다. 따라서 Oracle Solaris 11 MAC 계층의 네트워크 기능이 향상되었으므로 Oracle Solaris 11 서비스 도메인에서 `vsw`를 네트워크 인터페이스로 구성할 필요가 없습니다.

- Oracle Solaris 10 게스트 도메인의 가상 네트워크 장치 `vnet0`은 `ifconfig` 명령을 사용하여 네트워크 인터페이스로 구성할 수 있습니다.
- Oracle Solaris 11 게스트 도메인의 가상 네트워크 장치 `vnet0`은 `net0`과 같은 일반 링크 이름으로 표시될 수 있습니다. 또한 `ipadm` 명령을 사용하여 네트워크 인터페이스로 구성할 수 있습니다.

가상 스위치는 정규 물리적 네트워크 스위치처럼 작동하며, 연결되어 있는 여러 시스템 간에 네트워크 패킷을 전환합니다. 시스템은 게스트 도메인, 서비스 도메인 또는 물리적 네트워크 중 하나일 수 있습니다.

가상 스위치

가상 스위치(vsw)는 서비스 도메인에서 실행되고 가상 스위치 드라이버에 의해 관리되는 구성 요소로, 일부 게스트 도메인에 연결되어 게스트 도메인 간 네트워크 통신이 가능하도록 해줍니다. 또한 가상 스위치가 물리적 네트워크 인터페이스와도 연관된 경우, 물리적 네트워크 인터페이스를 통해 게스트 도메인과 물리적 네트워크 간 네트워크 통신이 허용됩니다. 가상 스위치에는 또한 서비스 도메인이 가상 스위치에 연결된 다른 도메인과 통신할 수 있도록 해주는 네트워크 인터페이스 `vswm`이 있습니다. 가상 스위치는 정규 네트워크 인터페이스처럼 사용할 수 있으며, Oracle Solaris 10 `ifconfig` 명령 또는 Oracle Solaris 11 `ipadm` 명령으로 구성할 수 있습니다.

주 - Oracle Solaris 10 서비스 도메인에 가상 스위치를 추가하면 네트워크 인터페이스가 생성되지 않습니다. 따라서 기본적으로 서비스 도메인은 가상 스위치에 연결된 게스트 도메인과 통신할 수 없습니다. 게스트 도메인과 서비스 도메인 간 네트워크 통신이 가능하게 하려면 연관된 가상 스위치의 네트워크 인터페이스를 서비스 도메인에 생성하고 구성해야 합니다. 지침은 [61 페이지 “컨트롤/서비스 도메인과 다른 도메인 사이의 네트워킹 사용”](#)을 참조하십시오.

이러한 상황은 Oracle Solaris 10 OS의 **경우에만** 발생하며 Oracle Solaris 11 OS의 경우에는 발생하지 **않습니다**.

`ldm add-vsw`, `ldm set-vsw` 및 `ldm rm-vsw` 명령을 각각 사용하여 도메인에 가상 스위치를 추가하고, 가상 스위치 옵션을 설정하며, 가상 스위치를 제거할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 예에서는 물리적 네트워크 어댑터에 가상 스위치를 만드는 방법에 대해 설명합니다.

- **Oracle Solaris 10:** 다음 명령은 물리적 네트워크 어댑터에 `nxge0`이라는 가상 스위치를 만듭니다.

```
primary# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

가상 스위치를 네트워크 인터페이스로 구성하는 방법은 [61 페이지 “컨트롤/서비스 도메인과 다른 도메인 사이의 네트워킹 사용”](#)을 참조하십시오.

- **Oracle Solaris 11:** 다음 명령은 물리적 네트워크 어댑터에 `net0`이라는 가상 스위치를 만듭니다.

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

가상 네트워크 장치

가상 네트워크(vnet) 장치는 가상 스위치에 연결된 도메인에 정의된 가상 장치로, 가상 네트워크 드라이버에 의해 관리되며, LDC(논리적 도메인 채널)를 사용하여 하이퍼바이저를 통해 가상 네트워크에 연결됩니다.

가상 네트워크 장치는 `vnet n` 이라는 이름의 네트워크 인터페이스로 사용할 수 있으며, 정규 네트워크 인터페이스처럼 사용되어 Oracle Solaris 10 `ifconfig` 명령 또는 Oracle Solaris 11 `ipadm` 명령으로 구성할 수 있습니다.

주 - Oracle Solaris 11에서는 장치에 일반 이름이 지정되므로, `vnet n` 에는 `net0`과 같은 일반 이름이 사용됩니다.

`ldm add-vnet`, `ldm set-vnet` 및 `ldm rm-vnet` 명령을 각각 사용하여 도메인에 가상 네트워크 장치를 추가하고, 기존의 가상 네트워크 장치에 대한 옵션을 설정하며, 가상 네트워크 장치를 제거할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

[그림 8-1](#) 및 [그림 8-2](#)에서 각각 Oracle Solaris 10 및 Oracle Solaris 11에 대한 Oracle VM Server for SPARC 네트워크 정보를 참조하십시오.

Vnet 간 LDC 채널

Oracle VM Server for SPARC 2.1 릴리스까지는 Logical Domains Manager가 다음과 같은 방식으로 LDC 채널을 지정했습니다.

- 가상 네트워크 장치와 가상 스위치 장치 간에 LDC 채널을 지정합니다.
- 동일한 가상 스위치에 연결된 가상 네트워크 장치 쌍 간에 LDC 채널을 지정합니다(Vnet 간).

Vnet 간 LDC 채널은 뛰어난 게스트 간 통신 성능을 얻기 위해 가상 네트워크 장치가 직접 통신할 수 있도록 구성됩니다. 그러나 가상 스위치 장치에 있는 가상 네트워크 장치의 수가 늘어나면 Vnet 간 통신에 필요한 LDC 채널 수도 기하급수적으로 늘어납니다.

지정된 가상 스위치 장치에 연결된 모든 가상 네트워크 장치에 대해 Vnet 간 LDC 채널 할당을 사용 또는 사용 안함으로 설정할 수 있습니다. 이 할당을 사용 안함으로 설정하면 LDC 채널의 개수가 제한되어 소비가 줄어들 수 있습니다.

다음과 같은 경우에 이 할당을 사용 안함으로 설정하는 것이 좋습니다.

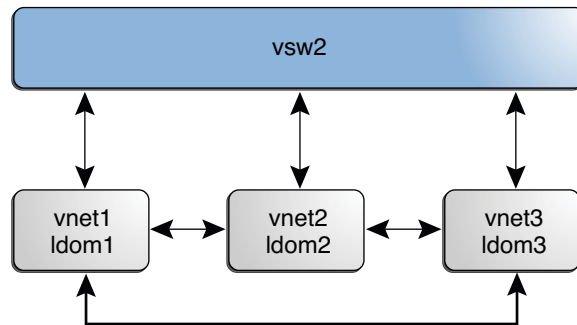
- 게스트 간 통신 성능이 중요하지 않은 경우
- 가상 스위치 장치에 많은 수의 가상 네트워크 장치가 필요한 경우

Vnet 간 채널을 지정하지 않으면 게스트 도메인에 가상 I/O 장치를 추가할 때 더 많은 LDC 채널을 사용할 수 있습니다.

주 - 게스트 간 성능이 시스템에서 가상 네트워크 장치 수를 늘리는 것보다 더 중요한 경우 Vnet 간 LDC 채널 할당을 사용 안함으로 설정하지 마십시오.

다음 그림은 가상 네트워크 장치를 세 개 포함하는 일반적인 가상 스위치를 보여줍니다. `inter-vnet-link` 등록 정보가 `on`으로 설정되어 있어 Vnet 간 LDC 채널이 할당되었음을 나타냅니다. `vnet1`과 `vnet2` 사이에서 이루어지는 게스트 간 통신은 가상 스위치를 거치지 않고 직접 수행됩니다.

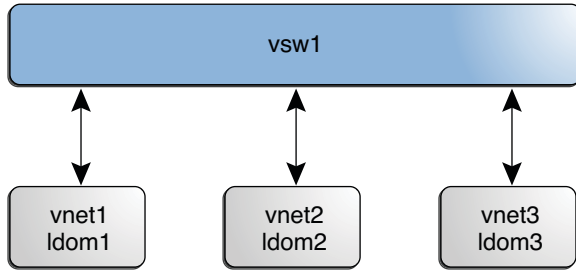
그림 8-3 Vnet 간 채널을 사용하는 가상 스위치 구성



다음 그림은 `inter-vnet-link` 등록 정보가 `off`로 설정된 동일한 가상 스위치 구성을 보여줍니다. 즉, Vnet 간 LDC 채널이 할당되지 않았습니다. `inter-vnet-link` 등록 정보가 `on`으로 설정된 경우보다 더 적은 수의 LDC 채널이 사용되었음을 확인할 수 있습니다. 이 구성에서는 `vnet1`과 `vnet2` 사이에서 이루어지는 게스트 간 통신이 `vsw1`을 통해 수행되어야 합니다.

주 - Vnet 간 LDC 채널 지정을 사용 안함으로 설정해도 게스트 간 통신이 수행됩니다. 대신, 모든 게스트 간 통신 트래픽이 한 게스트 도메인에서 다른 게스트 도메인으로 직접 이동하지 않고 가상 스위치를 거칩니다.

그림 8-4 Vnet 간 채널을 사용하지 않는 가상 스위치 구성



가상 장치 식별자 및 네트워크 인터페이스 이름

가상 스위치 또는 가상 네트워크 장치를 도메인에 추가하는 경우 id 등록 정보를 설정하여 장치 번호를 지정할 수 있습니다.

```
# ldm add-vsw [id=switch-id] vswitch-name ldom
# ldm add-vnet [id=network-id] if-name vswitch-name ldom
```

도메인의 가상 스위치 및 가상 네트워크 장치마다 고유한 장치 번호가 지정되는데, 이 번호는 도메인이 바인딩될 때 지정됩니다. id 등록 정보를 설정하여 명시적 장치 번호로 가상 스위치 또는 가상 네트워크 장치를 추가한 경우 지정된 장치 번호가 사용됩니다. 그렇지 않은 경우 시스템에서 사용 가능한 가장 낮은 장치 번호를 자동으로 지정합니다. 이 경우 가상 스위치 또는 가상 네트워크 장치가 시스템에 추가된 방식에 따라 지정된 장치 번호가 달라집니다. 가상 스위치 또는 가상 네트워크 장치에 지정된 장치 번호는 도메인이 바인딩될 때 ldm list-bindings 명령의 출력에 표시됩니다.

다음 예는 primary 도메인에 primary-vsw0라는 한 개의 가상 스위치가 있음을 보여줍니다. 이 가상 스위치의 장치 번호는 0(switch@0)입니다.

```
primary# ldm list-bindings primary
...
VSW
  NAME          MAC          NET-DEV DEVICE  DEFAULT-VLAN-ID PVID VID MTU MODE
  primary-vsw0  00:14:4f:fb:54:f2 nxge0  switch@0  1              1   5,6 1500
...
```

다음 예는 ldg1 도메인에 vnet 및 vnet1이라는 두 개의 가상 네트워크 장치가 있음을 보여줍니다. vnet 장치의 장치 번호는 0(network@0)이고, vnet1 장치의 장치 번호는 1(network@1)입니다.

```
primary# ldm list-bindings ldg1
...
NETWORK
  NAME  SERVICE          DEVICE  MAC          MODE  PVID VID MTU
  vnet  primary-vsw0@primary network@0 00:14:4f:fb:e0:4b hybrid 1    1500
  ...
  vnet1 primary-vsw0@primary network@1 00:14:4f:f8:e1:ea 1    1500
...
```

마찬가지로, 가상 네트워크 장치가 있는 도메인에서 Oracle Solaris OS가 실행 중인 경우, 가상 네트워크 장치는 네트워크 인터페이스 `vnet N` 을 사용합니다. 그러나 가상 네트워크 장치의 네트워크 인터페이스 번호 N 이 반드시 가상 네트워크 장치의 장치 번호 n 과 같을 필요는 없습니다.

주 - Oracle Solaris 11 시스템에서는 `netn` 형식의 일반 링크 이름이 `vsw n` 과 `vnet n` 에 지정됩니다. `dladm show-phys` 명령을 사용하면 어떤 `netn` 이름이 `vsw n` 및 `vnet n` 장치에 매핑되는지 확인할 수 있습니다.



주의 - Oracle Solaris OS에서는 네트워크 인터페이스의 이름과 장치 번호 기반의 가상 스위치 또는 가상 네트워크 간 매핑이 유지됩니다. 가상 스위치 또는 가상 네트워크 장치에 장치 번호가 명시적으로 지정되지 않은 경우, 도메인의 바인딩이 취소되었다가 다시 바인딩될 때 장치 번호가 변경될 수 있습니다. 이 경우, 도메인에서 실행되는 OS에 의해 지정된 네트워크 인터페이스 이름도 변경되고 시스템의 기존 구성이 손상됩니다. 예를 들어 이러한 상황은 가상 스위치 또는 가상 네트워크 인터페이스가 도메인 구성에서 제거된 경우에 발생할 수 있습니다.

`ldm list-*` 명령을 사용하여 가상 스위치 또는 가상 네트워크 장치에 해당하는 Oracle Solaris OS 네트워크 인터페이스 이름을 직접 확인할 수는 없습니다. 그러나 `ldm list -l` 명령의 결과와 Oracle Solaris OS의 `/devices` 아래에 있는 항목의 결과를 조합하여 이 정보를 확인할 수는 있습니다.

▼ Oracle Solaris OS 네트워크 인터페이스 이름을 찾는 방법

이 예제 절차에서는 게스트 도메인 `ldg1`에 `net-a` 및 `net-c`라는 두 개의 가상 네트워크 장치가 있습니다. `ldg1`에서 `net-c`에 해당하는 Oracle Solaris OS 네트워크 인터페이스 이름을 찾으려면 다음을 수행하십시오. 이 예는 또한 가상 네트워크 장치 대신 가상 스위치의 네트워크 인터페이스 이름을 찾을 경우 차이점도 보여줍니다.

- 1 `ldm` 명령을 사용하여 `net-c`에 대한 가상 네트워크 장치 번호를 찾습니다.

```
# ldm list -l ldg1
...
NETWORK
NAME      SERVICE          DEVICE          MAC
net-a     primary-vsw0@primary  network@0      00:14:4f:f8:91:4f
net-c     primary-vsw0@primary  network@2      00:14:4f:f8:dd:68
...
```

`net-c`에 대한 가상 네트워크 장치 번호는 2(`network@2`)입니다.

가상 스위치의 네트워크 인터페이스 이름을 확인하려면 가상 스위치 장치 번호 n (예: `switch@ n`)을 찾습니다.

- 2 **ldg1**에서 해당 네트워크 인터페이스를 찾으려면 **ldg1**에 로그인한 다음 **/devices**에서 이 장치 번호에 해당하는 항목을 찾습니다.

```
# uname -n
ldg1
# find /devices/virtual-devices@100 -type c -name network@2\*
/devices/virtual-devices@100/channel-devices@200/network@2:vnet1
```

네트워크 인터페이스 이름은 콜론 뒤에 오는 항목(**vnet1**)의 일부입니다.

가상 스위치의 네트워크 인터페이스 이름을 확인하려면 **-name** 옵션에 대한 인수를 **virtual-network-switch@n***로 바꿉니다. 그런 다음 이름이 **vswN**인 네트워크 인터페이스를 찾습니다.

- 3 1단계의 **net-c**에 대한 **ldm list -l** 출력에 표시된 것과 같이, **vnet1**의 MAC 주소가 **00:14:4f:f8:dd:68**인지 확인합니다.

- Oracle Solaris 10 OS. **ifconfig** 명령을 사용합니다.

```
# ifconfig vnet1
vnet1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 0.0.0.0 netmask 0
        ether 0:14:4f:f8:dd:68
```

- Oracle Solaris 11 OS. **dladm** 명령을 사용합니다.

먼저 **dladm show-phys** 명령을 사용하여 **vnet1**에 대해 지정할 인터페이스의 이름을 확인해야 합니다.

```
primary# dladm show-phys |grep vnet1
net2                Ethernet                up                0                unknown        vnet1
```

```
primary# dladm show-linkprop -p mac-address net2
LINK      PROPERTY      PERM VALUE      DEFAULT      POSSIBLE
net2      mac-address   rw    00:14:4f:f8:dd:68 00:14:4f:f8:dd:68 --
```

자동 또는 수동 MAC 주소 지정

사용하려는 논리적 도메인, 가상 스위치 및 가상 네트워크 수에 지정할 MAC(매체 액세스 제어) 주소가 충분해야 합니다. Logical Domains Manager가 자동으로 논리적 도메인, 가상 네트워크(**vnet**) 및 가상 스위치(**vsw**)에 MAC 주소를 지정하거나, 지정된 MAC 주소의 사용자 고유 풀에서 수동으로 MAC 주소를 지정할 수 있습니다. MAC 주소를 설정하는 **ldm** 하위 명령은 **add-domain**, **add-vsw**, **set-vsw**, **add-vnet** 및 **set-vnet**입니다. 이러한 하위 명령에 MAC 주소를 지정하지 않은 경우 Logical Domains Manager에서 자동으로 지정합니다.

Logical Domains Manager가 MAC 주소를 지정할 경우 이점은 논리적 도메인에만 사용할 수 있는 MAC 주소 블록을 활용할 수 있다는 점입니다. 또한 Logical Domains Manager는 같은 서버넷에 있는 다른 Logical Domains Manager 인스턴스와 MAC 주소가 충돌하는 것을 찾아내어 방지합니다. 따라서 MAC 주소 풀을 수동으로 관리할 필요가 없습니다.

MAC 주소는 논리적 도메인이 만들어지거나 네트워크 장치가 도메인에 구성되는 즉시 지정됩니다. 또한 장치 또는 논리적 도메인 자체가 제거될 때까지 지정된 주소가 유지됩니다.

Logical Domains에 지정된 MAC 주소의 범위

Logical Domains에는 다음과 같은 512K MAC 주소 블록이 지정됩니다.

00:14:4F:F8:00:00 ~ 00:14:4F:FF:FF:FF

이보다 낮은 256K 주소는 **자동 MAC 주소 할당**을 위해 Logical Domains Manager에서 사용하므로, 이 범위의 주소는 수동으로 요청할 수 없습니다.

00:14:4F:F8:00:00 - 00:14:4F:FB:FF:FF

이 범위의 위쪽 절반은 **수동 MAC 주소 할당**에 사용할 수 있습니다.

00:14:4F:FC:00:00 - 00:14:4F:FF:FF:FF

주 - Oracle Solaris 11에서 VNIC에 대한 MAC 할당은 이 범위 밖에 있는 주소를 사용합니다.

자동 지정 알고리즘

논리적 도메인이나 네트워크 장치를 만들 때 MAC 주소를 지정하지 않은 경우 Logical Domains Manager가 자동으로 해당 논리적 도메인이나 네트워크 장치에 MAC 주소를 할당합니다. 이 MAC 주소를 확인하기 위해 Logical Domains Manager에서는 반복적으로 주소를 선택한 다음 충돌 가능성이 있는지 확인합니다.

가능한 주소를 선택하기 전에 Logical Domains Manager에서는 먼저 최근에 해제되어 자동으로 지정된 주소가 이 목적을 위해 데이터베이스에 저장되어 있는지 확인합니다([148 페이지 “해제된 MAC 주소”](#) 참조). 그럴 경우 Logical Domains Manager는 데이터베이스에서 후보 주소를 선택합니다.

최근에 해제된 주소를 사용할 수 없는 경우 이 목적을 위해 설정된 256K 주소 범위에서 MAC 주소가 무작위로 선택됩니다. 중복된 MAC 주소를 후보로 선택할 가능성을 줄이기 위해 MAC 주소가 무작위로 선택됩니다.

그런 다음 중복된 MAC 주소가 실제로 지정되는 것을 방지하기 위해 선택된 주소는 다른 시스템에 있는 다른 Logical Domains Manager에 대해 검사됩니다. 사용된 알고리즘은 [148 페이지 “중복된 MAC 주소 감지”](#)에 설명되어 있습니다. 주소가 이미 지정된 경우 Logical Domains Manager가 반복적으로 다른 주소를 선택하여 충돌 여부를 다시 검사합니다. 이 작업은 아직 할당되지 않은 MAC 주소를 찾을 때까지 또는 30초 시간 제한이 경과할 때까지 계속됩니다. 이 시간 제한에 도달하면 장치 만들기가 실패하고 다음과 비슷한 오류 메시지가 표시됩니다.

Automatic MAC allocation failed. Please set the vnet MAC address manually.

중복된 MAC 주소 감지

동일한 MAC 주소가 다른 장치에 할당되지 않도록 하기 위해 Logical Domains Manager가 장치에 지정하려고 하는 주소를 비롯한 멀티캐스트 메시지를 컨트롤 도메인의 기본 네트워크 인터페이스를 통해 전송하는 방식으로 한 Logical Domains Manager가 다른 시스템에 있는 다른 Logical Domains Manager와 함께 검사를 수행합니다. MAC 주소를 지정하려고 시도하는 Logical Domains Manager는 1초간 응답을 다시 기다립니다. 다른 Logical Domains 지원 시스템에 있는 다른 장치에 이미 해당 MAC 주소가 지정된 경우, 해당 시스템의 Logical Domains Manager는 해당 MAC 주소가 포함된 응답을 다시 전송합니다. 요청하는 Logical Domains Manager가 응답을 받으면 선택한 MAC 주소가 이미 할당되었음을 알게 되고 다른 주소를 반복적으로 선택합니다.

기본적으로 이러한 멀티캐스트 메시지는 같은 서브넷에 있는 다른 관리자에게만 전송되며, 기본 TTL(활성 시간)은 1입니다. TTL은 SMF(서비스 관리 기능) 등록 정보 ldmd/hops를 사용하여 구성할 수 있습니다.

각 Logical Domains Manager는 다음과 같은 작업을 수행합니다.

- 멀티캐스트 메시지 수신
- 도메인에 지정된 MAC 주소 추적
- 중복된 주소 찾기
- 중복이 발생하지 않도록 응답

어떤 이유로 시스템에 있는 Logical Domains Manager가 종료된 경우, Logical Domains Manager가 작동 중지된 동안 중복된 MAC 주소가 발생할 수 있습니다.

자동 MAC 할당은 논리적 도메인 또는 네트워크 장치가 생성될 때 발생하며, 장치 또는 논리적 도메인이 제거될 때까지 유지됩니다.

주 - 논리적 도메인 또는 네트워크 장치가 생성될 때 중복된 MAC 주소를 감지하기 위한 검사가 수행되고 논리적 도메인이 시작됩니다.

해제된 MAC 주소

자동 MAC 주소와 연관된 논리적 도메인이나 장치가 제거되면 해당 MAC 주소는 나중에 해당 시스템에서 사용할 수 있도록 최근에 해제된 MAC 주소의 데이터베이스에 저장됩니다. 이러한 MAC 주소는 DHCP(동적 호스트 구성 프로토콜) 서버에서 IP(인터넷 프로토콜) 주소가 모두 소진되는 것을 방지하기 위해 저장됩니다. DHCP 서버가 IP 주소를 할당하는 경우 일정 기간(임대 기간) 동안 주소를 할당합니다. 임대 기간은 대개 몇 시간 또는 며칠 정도로 길게 구성됩니다. 자동으로 할당된 MAC 주소가 Logical Domains Manager에서 다시 사용되지 않고 네트워크 장치가 높은 비율로 생성되었다가 제거되는 경우, 할당된 MAC 주소 수가 일반적으로 구성된 DHCP 서버에서 너무 많아지게 됩니다.

Logical Domains Manager가 논리적 도메인 또는 네트워크 장치에 대한 MAC 주소를 자동으로 얻도록 요청 받은 경우, 먼저 해제된 MAC 주소 데이터베이스에서 이전에 지정된 적이 있는 다시 사용 가능한 MAC 주소가 있는지 확인합니다. 이 데이터베이스에서 사용 가능한 MAC 주소가 있는 경우 중복된 MAC 주소 감지 알고리즘이 실행됩니다. 이전에 해제된 이후로 MAC 주소가 다른 사용자에게 지정되지 않은 경우 다시 사용되며 데이터베이스에서 제거됩니다. 충돌이 감지될 경우 해당 주소가 데이터베이스에서 제거됩니다. 그런 다음 Logical Domains Manager는 데이터베이스에 있는 다음 주소를 시도하거나, 사용 가능한 주소가 없는 경우 새 MAC 주소를 무작위로 선택합니다.

Logical Domains에서 네트워크 어댑터 사용

Oracle Solaris 10 논리적 도메인 환경에서 서비스 도메인에서 실행되는 가상 스위치 서비스는 GLDv3 준수 네트워크 어댑터와 직접 상호 작용할 수 있습니다. GLDv3 비준수 네트워크 어댑터를 이러한 시스템에 사용할 수는 있지만, 가상 스위치가 이러한 시스템과 직접 연결될 수 없습니다. GLDv3 비준수 네트워크 어댑터 사용법에 대한 자세한 내용은 [150 페이지 “NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성”](#)을 참조하십시오.

주 - Oracle Solaris 11 환경의 경우 GLDv3 준수 문제가 해당되지 않습니다.

링크 통합 사용에 대한 자세한 내용은 [169 페이지 “가상 스위치에서 링크 통합 사용”](#)을 참조하십시오.

▼ GLDv3 준수 네트워크 어댑터인지 확인하는 방법(Oracle Solaris 10)

이 절차는 Oracle Solaris 10 도메인에만 적용됩니다.

- **네트워크 어댑터가 GLDv3 준수 네트워크 어댑터인지 확인합니다.**

다음 예에서는 bge0이 네트워크 장치 이름으로 사용됩니다.

```
# dladm show-link bge0
bge0          type: non-vlan   mtu: 1500      device: bge0
```

type: 필드의 값은 다음 중 하나입니다.

- GLDv3 준수 드라이버의 유형은 non-vlan 또는 vlan입니다.
- GLDv3 비준수 드라이버의 유형은 legacy입니다.

NAT 및 경로 지정을 위해 가상 스위치 및 서비스도메인 구성

Oracle Solaris 10 OS에서 가상 스위치(vsw)는 계층 2 스위치로서, 서비스도메인에서 네트워크 장치로 사용할 수도 있습니다. 가상 스위치가 여러 논리적 도메인의 가상 네트워크(vnet) 장치 간 스위치로만 사용되고 물리적 장치를 통해 외부 네트워크에 연결할 수 없도록 구성할 수 있습니다. 이 모드에서 vsw를 네트워크 장치로 만들고 서비스도메인에서 IP 경로 지정을 사용으로 설정하면 가상 네트워크가 서비스도메인을 라우터로 사용하여 외부와 통신할 수 있습니다. 이 작동 모드는 물리적 네트워크 어댑터가 GLDv3 비준수 어댑터일 경우 도메인에 대한 외부 연결을 제공하는 데 반드시 필요합니다.

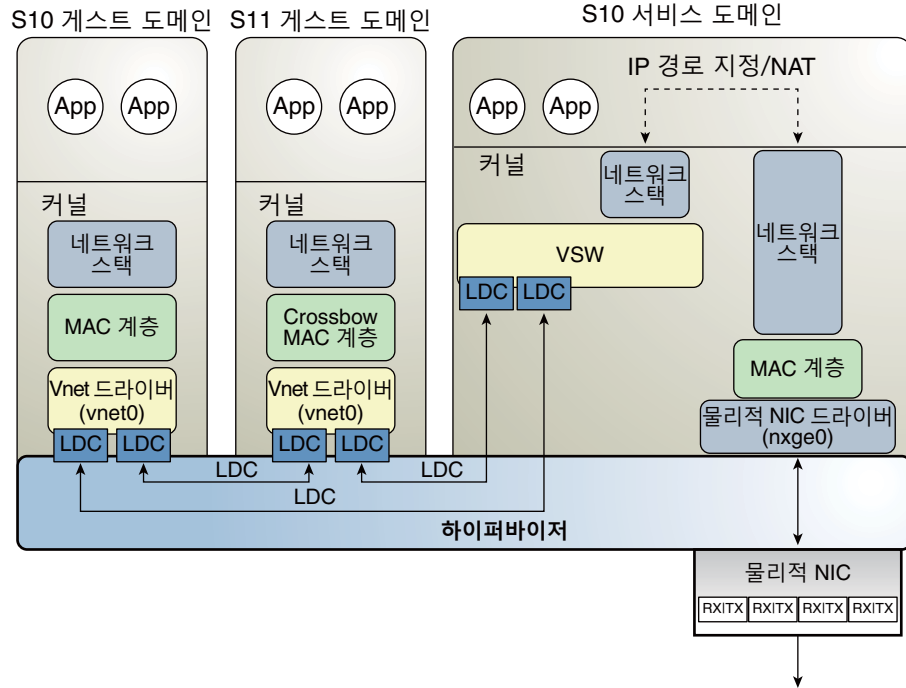
이 구성의 이점은 다음과 같습니다.

- 가상 스위치가 물리적 장치를 직접 사용할 필요가 없으며 기본 장치가 GLDv3 비준수 장치일 경우에도 외부 연결을 제공할 수 있습니다.
- Oracle Solaris OS의 IP 경로 지정 및 필터링 기능을 구성에 활용할 수 있습니다.

Oracle Solaris 10 시스템에서 NAT 구성

다음 다이어그램은 게스트 도메인에 대한 외부 연결을 제공하기 위해 가상 스위치를 사용하여 서비스도메인에서 NAT(Network Address Translation)를 구성하는 방법을 보여줍니다.

그림 8-5 가상 네트워크 경로 지정



▼ 도메인에 대한 외부 연결을 제공하기 위해 가상 스위치를 설정하는 방법(Oracle Solaris 10)

- 1 연관된 물리적 장치가 없는 가상 스위치를 만듭니다.

주소를 지정한 경우 가상 스위치의 MAC 주소가 고유한지 확인합니다.

```
primary# ldm add-vsw [mac-addr=xx:xx:xx:xx:xx:xx] primary-vsw0 primary
```

- 2 가상 스위치를 도메인에서 사용 중인 물리적 네트워크 장치 및 네트워크 장치로 만듭니다.

가상 스위치 만들기에 대한 자세한 내용은 62 페이지 “가상 스위치를 주 인터페이스로 구성하는 방법”을 참조하십시오.

- 3 필요한 경우 DHCP용 가상 스위치 장치를 구성합니다.

DHCP용 가상 스위치 장치 만들기에 대한 자세한 내용은 62 페이지 “가상 스위치를 주 인터페이스로 구성하는 방법”을 참조하십시오.

- 4 필요한 경우 /etc/dhcp.vsw 파일을 만듭니다.

- 5 서비스도메인에서 IP 경로 지정을 구성한 다음 모든 도메인에서 필요한 경로 지정 테이블을 설정합니다.

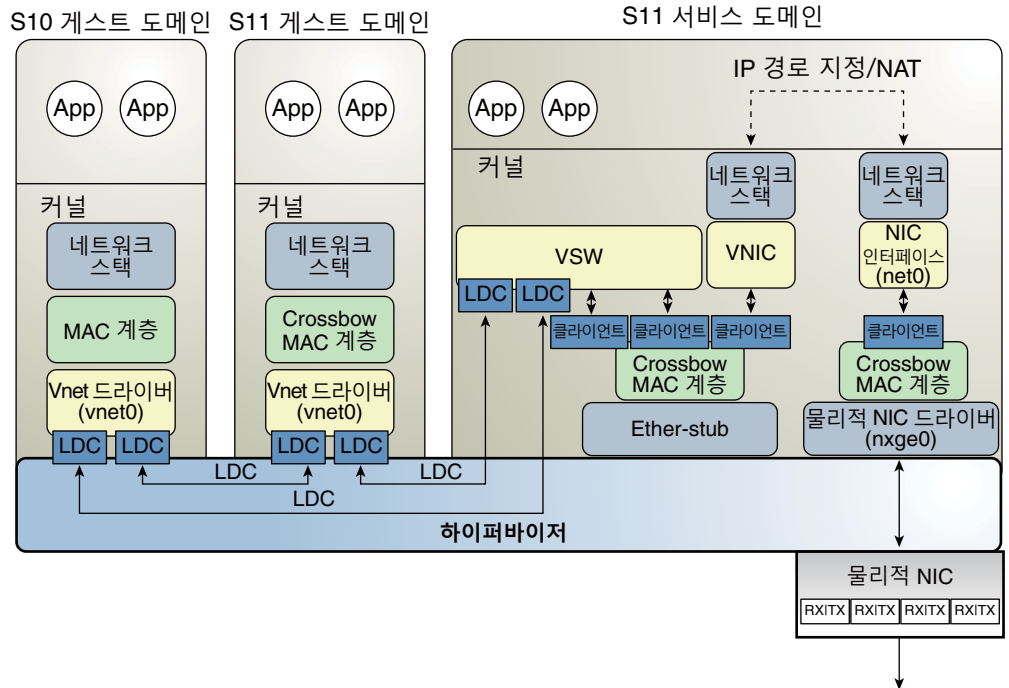
IP 경로 지정에 대한 자세한 내용은 [System Administration Guide: IP Services](#)의 “Packet Forwarding and Routing on IPv4 Networks”를 참조하십시오.

Oracle Solaris 11 시스템에서 NAT 구성

Oracle Solaris 11 네트워크 가상화 기능에는 의사 네트워크 장치인 `etherstub`가 포함되어 있습니다. 이 장치는 물리적 네트워크 장치와 비슷한 기능을 제공하지만, 클라이언트와의 개인 통신에 한합니다. 이 의사 장치는 가상 네트워크 간 개인 통신을 제공하는 가상 스위치에 대한 네트워크 백엔드 장치로 사용할 수 있습니다. `etherstub` 장치를 백엔드 장치로 사용하면 게스트 도메인이 같은 `etherstub` 장치에 있는 VNIC와도 통신할 수 있습니다. 이 방식으로 `etherstub`를 사용하면 게스트 도메인이 서비스 도메인의 영역과 통신할 수 있습니다. `etherstub` 장치를 만들려면 `dladm create-etherstub` 명령을 사용하십시오.

다음 다이어그램은 가상 스위치, `etherstub` 장치 및 VNIC를 사용하여 서비스 도메인에서 NAT(Network Address Translation)를 설정하는 방법을 보여줍니다.

그림 8-6 가상 네트워크 경로 지정



▼ 도메인에 대한 외부 연결을 제공하기 위해 가상 스위치를 설정하는 방법 (Oracle Solaris 11)

- 1 Oracle Solaris 11 etherstub 장치를 만듭니다.

```
primary# dladm create-etherstub stub0
```

- 2 stub0을 물리적 백엔드 장치로 사용하는 가상 스위치를 만듭니다.

```
primary# ldm add-vsw net-dev=stub0 primary-stub-vsw0 primary
```

- 3 stub0 장치에 VNIC를 만듭니다.

```
primary# dladm create-vnic -l stub0 vnic0
```

- 4 vnic0을 네트워크 인터페이스로 구성합니다.

```
primary# ipadm create-ip vnicstub0
primary# ipadm create-addr -T static -a 192.168.100.1/24 vnicstub0/v4static
```

- 5 IPv4 전달을 사용으로 설정하고 NAT 규칙을 만듭니다.

Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “IP 인터페이스 등록 정보 설정” 및 **System Administration Guide: IP Services**의 “Packet Forwarding and Routing on IPv4 Networks”를 참조하십시오.

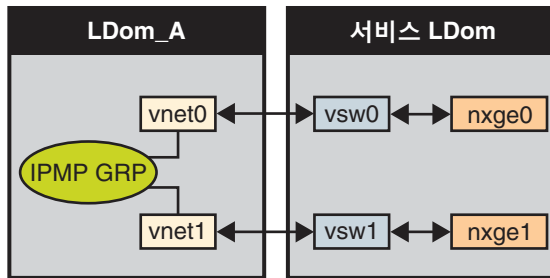
Logical Domains 환경에서 IPMP 구성

Oracle VM Server for SPARC 소프트웨어는 가상 네트워크 장치에 대해 링크 기반 IPMP(IP Network Multipathing)를 지원합니다. 가상 네트워크 장치에 대해 IPMP 그룹을 구성할 경우 링크 기반 감지를 사용할 그룹을 구성하십시오. 이전 버전의 Oracle VM Server for SPARC(Logical Domains) 소프트웨어를 사용할 경우 가상 네트워크 장치에 대해 프로브 기반 감지만 구성할 수 있습니다.

도메인에서 가상 네트워크 장치를 IPMP 그룹으로 구성

다음 다이어그램은 두 개의 가상 네트워크(vnet0 및 vnet1)가 서비스 도메인의 개별 가상 스위치 인스턴스(vsw0 및 vsw1)에 연결되어 있고, 서로 다른 두 개의 물리적 인터페이스를 사용함을 보여줍니다. 물리적 인터페이스는 Oracle Solaris 10의 경우 nxge0 및 nxge1이고, Oracle Solaris 11의 경우 net0 및 net1입니다. 이 다이어그램은 Oracle Solaris 10 물리적 인터페이스 이름을 보여줍니다. 서비스 도메인에서 물리적 링크 오류가 발생하면 해당 물리적 장치에 바인드된 가상 스위치 장치에서도 링크 오류가 발생합니다. 그러면 가상 스위치가 이 가상 스위치에 바인드된 해당 가상 네트워크 장치에 오류를 전파합니다. 가상 네트워크 장치는 이 링크 이벤트에 대한 알림을 LDom_A 게스트의 IP 계층에 전송합니다. 그러면 IPMP 그룹에 있는 다른 가상 네트워크 장치로 페일오버됩니다.

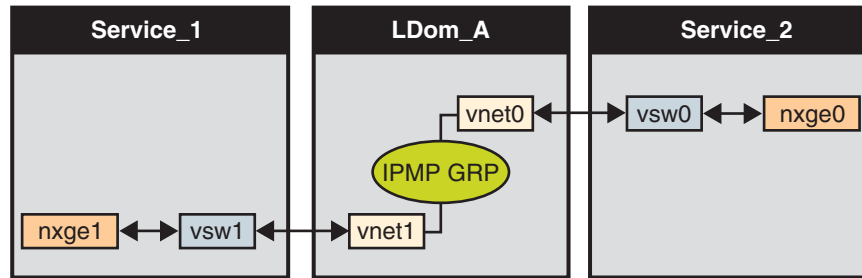
그림 8-7 개별 가상 스위치 인스턴스에 연결된 두 개의 가상 네트워크



주 - 그림 8-7은 Oracle Solaris 10 시스템의 구성을 보여줍니다. Oracle Solaris 11 시스템의 경우에는 nxge0 및 nxge1에 각각 net0 및 net1 등의 일반 이름을 사용하도록 인터페이스 이름만 변경됩니다.

각각의 가상 네트워크 장치(vnet0 및 vnet1)를 다른 서비스 도메인의 가상 스위치 인스턴스에 연결(다음 다이어그램에 표시됨)하면 논리적 도메인의 신뢰성이 보다 향상될 수 있습니다. 이 경우 LDom_A는 물리적 네트워크 오류 이외에 가상 네트워크 오류를 감지하고 서비스 도메인 충돌 및 종료 후 페일오버를 트리거합니다.

그림 8-8 다른 서비스 도메인에 연결된 개별 가상 네트워크 장치



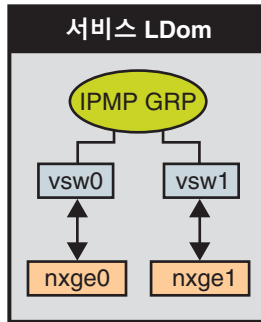
주 - 그림 8-8은 Oracle Solaris 10 시스템의 구성을 보여줍니다. Oracle Solaris 11 시스템의 경우에는 nxge0 및 nxge1에 각각 net0 및 net1 등의 일반 이름을 사용하도록 인터페이스 이름만 변경됩니다.

자세한 내용은 Oracle Solaris 10 [System Administration Guide: IP Services](#) 또는 Oracle Solaris 11 [Oracle Solaris 관리: IP 서비스](#)를 참조하십시오.

서비스 도메인에서 IPMP 구성 및 사용

가상 스위치 인터페이스를 그룹으로 구성하여 서비스 도메인에서 IPMP를 구성할 수 있습니다. 다음 다이어그램은 서로 다른 두 개의 물리적 장치에 바인드된 두 개의 가상 스위치 인스턴스(vsw0 및 vsw1)를 보여줍니다. 그런 다음 두 개의 가상 스위치 인터페이스를 만들어 IPMP 그룹으로 구성할 수 있습니다. 물리적 링크 오류가 발생하면 해당 물리적 장치에 바인드된 가상 스위치 장치에서도 링크 오류가 발생합니다. 그런 다음 가상 스위치 장치가 이 링크 이벤트에 대한 알림을 서비스 도메인의 IP 계층에 전송합니다. 그러면 IPMP 그룹에 있는 다른 가상 스위치 장치로 패일오버됩니다. 두 개의 물리적 인터페이스는 Oracle Solaris 10의 경우 nxge0 및 nxge1이고, Oracle Solaris 11의 경우 net0 및 net1입니다. 다음 다이어그램은 Oracle Solaris 10 물리적 인터페이스 이름을 보여줍니다.

그림 8-9 IPMP 그룹의 일부로 구성된 두 개의 가상 스위치 인터페이스



주 - 그림 8-9는 Oracle Solaris 10 시스템의 구성을 보여줍니다. Oracle Solaris 11 시스템의 경우에는 nxge0 및 nxge1에 각각 net0 및 net1 등의 일반 이름을 사용하도록 인터페이스 이름만 변경됩니다.

Logical Domains 가상 네트워킹에서 링크 기반 IPMP 사용

가상 네트워크 및 가상 스위치 장치는 네트워크 스택에 대한 링크 상태 업데이트를 지원합니다. 기본적으로 가상 네트워크 장치는 가상 링크(가상 스위치에 대한 LDC)의 상태를 보고합니다. 이 구성은 기본적으로 사용으로 설정되므로 구성 단계를 추가로 수행할 필요가 없습니다.

경우에 따라 물리적 네트워크 링크 상태 변경을 감지해야 할 수 있습니다. 예를 들어 가상 장치에 물리적 장치가 지정된 경우, 가상 네트워크 장치에서 가상 스위치 장치로의 링크가 작동하더라도 서비스 도메인에서 외부 네트워크로의 물리적 네트워크 링크는 작동하지 않을 수 있습니다. 이 경우 물리적 링크 상태를 확인하여 가상 네트워크 장치와 해당 스택에 보고해야 할 수 있습니다.

`linkprop=phys-state` 옵션은 가상 네트워크 장치와 가상 스위치 장치에 대해 물리적 링크 상태 추적을 구성하는 데 사용할 수 있습니다. 이 옵션을 사용으로 설정하면 가상 장치(가상 네트워크 또는 가상 스위치)가 도메인에 인터페이스로 생성된 경우 물리적 링크 상태에 따라 링크 상태를 보고합니다. 표준 Oracle Solaris 네트워크 관리 명령(예: `dladm` 및 `ifconfig`)을 사용하여 링크 상태를 확인할 수 있습니다. 또한 링크 상태는 `/var/adm/messages` 파일에도 기록됩니다.

Oracle Solaris 10의 경우, `dladm(1M)` 및 `ifconfig(1M)` 매뉴얼 페이지를 참조하십시오. Oracle Solaris 11의 경우, `dladm(1M)`, `ipadm(1M)` 및 `ipmpstat(1M)` 매뉴얼 페이지를 참조하십시오.

주 - Logical Domains 시스템에서는 현재 링크 상태 미인식 및 링크 상태 인식 vnet 및 vsw 드라이버를 모두 실행할 수 있습니다. 그러나 링크 기반 IPMP를 구성하려는 경우에는 링크 상태 인식 드라이버를 설치해야 합니다. 물리적 링크 상태 업데이트를 사용으로 설정하려면 vnet 및 vsw 드라이버를 Oracle Solaris 10 8/11 OS로 업그레이드하고 Logical Domains Manager 버전 1.3 이상을 실행하십시오.

▼ 물리적 링크 상태 업데이트를 구성하는 방법

이 절차는 가상 네트워크 장치에 대해 물리적 링크 상태 업데이트를 사용으로 설정하는 방법을 보여줍니다.

linkprop=phys-state 옵션을 ldm add-vsw 및 ldm set-vsw 명령에 지정하면 가상 스위치 장치에 대해서도 물리적 링크 상태 업데이트를 사용으로 설정할 수 있습니다.

주 - linkprop=phys-state 옵션은 가상 스위치 장치 자체가 인터페이스로 생성된 경우에만 사용해야 합니다. linkprop=phys-state를 지정했는데 물리적 링크가 작동하지 않을 경우, 가상 스위치에 대한 연결이 작동하더라도 가상 네트워크 장치가 링크 상태를 down으로 보고합니다. 이러한 상황은 Oracle Solaris OS에서 현재 두 개의 고유 링크 상태(예: virtual-link-state 및 physical-link-state)를 보고할 수 있는 인터페이스를 제공하지 않기 때문에 발생합니다.

1 관리자, 수퍼유저 또는 동등한 역할의 사용자로 로그인합니다.

Oracle Solaris 10의 경우, [System Administration Guide: Security Services](#)의 “Configuring RBAC (Task Map)”를 참조하십시오. Oracle Solaris 11의 경우, [Oracle Solaris 관리: 보안 서비스의 제III부](#), “역할, 권한 프로파일 및 권한”를 참조하십시오.

2 가상 장치에 대해 물리적 링크 상태 업데이트를 사용으로 설정합니다.

다음과 같은 방법으로 가상 네트워크 장치에 대해 물리적 링크 상태 업데이트를 사용으로 설정할 수 있습니다.

- ldm add-vnet 명령을 실행할 때 linkprop=phys-state를 지정하여 가상 네트워크 장치를 만듭니다.

linkprop=phys-state 옵션을 지정하면 물리적 링크 상태 업데이트를 확인하고 이력 스택에 보고하도록 가상 네트워크 장치가 구성됩니다.

주 - linkprop=phys-state를 지정했는데 물리적 링크가 작동하지 않을 경우, 가상 스위치에 대한 연결이 작동하더라도 가상 네트워크 장치가 링크 상태를 down으로 보고합니다. 이러한 상황은 Oracle Solaris OS에서 현재 두 개의 고유 링크 상태(예: virtual-link-state 및 physical-link-state)를 보고할 수 있는 인터페이스를 제공하지 않기 때문에 발생합니다.

```
# ldm add-vnet linkprop=phys-state if-name vswitch-name ldom
```

다음 예는 논리적 도메인 `ldom1`에서 `primary-vsw0`에 연결된 `vnet0`에 대한 물리적 링크 상태 업데이트를 사용으로 설정합니다.

```
# ldm add-vnet linkprop=phys-state vnet0 primary-vsw0 ldom1
```

- `ldm set-vnet` 명령을 실행할 때 `linkprop=phys-state`를 지정하여 기존 가상 네트워크 장치를 수정합니다.

```
# ldm set-vnet linkprop=phys-state if-name ldom
```

다음 예는 논리적 도메인 `ldom1`에서 `vnet0`에 대한 물리적 링크 상태 업데이트를 사용으로 설정합니다.

```
# ldm set-vnet linkprop=phys-state vnet0 ldom1
```

물리적 링크 상태 업데이트를 사용 안함으로 설정하려면 `ldm set-vnet` 명령을 실행할 때 `linkprop=`를 지정합니다.

다음 예는 논리적 도메인 `ldom1`에서 `vnet0`에 대한 물리적 링크 상태 업데이트를 사용 안함으로 설정합니다.

```
# ldm set-vnet linkprop= vnet0 ldom1
```

예 8-1 링크 기반 IPMP 구성

다음 예에서는 물리적 링크 상태 업데이트를 사용하거나 사용하지 않고 링크 기반 IPMP를 구성하는 방법에 대해 설명합니다.

- 다음 예는 도메인에 가상 네트워크 장치를 구성합니다. 각 가상 네트워크 장치는 링크 기반 IPMP를 사용하기 위해 서비스 도메인의 개별 가상 스위치 장치에 연결되어 있습니다.

주 - 이러한 가상 네트워크 장치에는 테스트 주소가 구성되지 않습니다. 또한 `ldm add-vnet` 명령을 사용하여 이러한 가상 네트워크 장치를 만들 때 추가 구성을 수행할 필요가 없습니다.

다음 명령은 도메인에 가상 네트워크 장치를 추가합니다. `linkprop=phys-state`가 지정되어 있지 않으므로 가상 스위치에 대한 링크에 대해서만 상태 변경을 모니터링합니다.

```
# ldm add-vnet vnet0 primary-vsw0 ldom1
# ldm add-vnet vnet1 primary-vsw1 ldom1
```

다음 명령은 게스트 도메인에서 가상 네트워크 장치를 구성하고 이 장치를 IPMP 그룹에 지정합니다. 링크 기반 오류 감지를 사용 중이므로 이러한 가상 네트워크 장치에는 테스트 주소가 구성되지 않습니다.

- **Oracle Solaris 10 OS.** `ifconfig` 명령을 사용합니다.

```
# ifconfig vnet0 plumb
# ifconfig vnet1 plumb
# ifconfig vnet0 192.168.1.1/24 up
```

```
# ifconfig vnet1 192.168.1.2/24 up
# ifconfig vnet0 group ipmp0
# ifconfig vnet1 group ipmp0
```

- Oracle Solaris 11 OS. ipadm 명령을 사용합니다.

net0 및 net1은 vnet0 및 vnet1 각각에 대한 Oracle Solaris 11 베니티 이름입니다.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i net0 -i net1 ipmp0
# ipadm create-addr -T static -a 192.168.1.1/24 ipmp0/v4addr1
# ipadm create-addr -T static -a 192.168.1.2/24 ipmp0/v4addr2
```

- 다음 예는 도메인에 가상 네트워크 장치를 구성합니다. 각 도메인은 링크 기반 IPMP를 사용하기 위해 서비스 도메인의 개별 가상 스위치 장치에 연결되어 있습니다. 가상 네트워크 장치도 물리적 링크 상태 업데이트를 확인하도록 구성되어 있습니다.

net0 및 net1은 vnet0 및 vnet1 각각에 대한 Oracle Solaris 11 베니티 이름입니다.

- Oracle Solaris 10 OS. 다음 명령을 사용하십시오.

```
# ldm add-vnet linkprop=phys-state vnet0 primary-vsw0 ldom1
# ldm add-vnet linkprop=phys-state vnet1 primary-vsw1 ldom1
```

- Oracle Solaris 11 OS. 다음 명령을 사용하십시오.

```
# ldm add-vnet linkprop=phys-state net0 primary-vsw0 ldom1
# ldm add-vnet linkprop=phys-state net1 primary-vsw1 ldom1
```

주 - 도메인을 성공적으로 바인드하려면 가상 스위치에 물리적 네트워크 장치가 지정되어 있어야 합니다. 도메인이 이미 바인드되었으며 가상 스위치에 물리적 네트워크 장치가 지정되지 않은 경우 ldm add-vnet 명령이 실패합니다.

다음 명령은 가상 네트워크 장치를 만들어 이 장치를 IPMP 그룹에 지정합니다.

- Oracle Solaris 10 OS. ifconfig 명령을 사용합니다.

```
# ifconfig vnet0 plumb
# ifconfig vnet1 plumb
# ifconfig vnet0 192.168.1.1/24 up
# ifconfig vnet1 192.168.1.2/24 up
# ifconfig vnet0 group ipmp0
# ifconfig vnet1 group ipmp0
```

- Oracle Solaris 11 OS. ipadm 명령을 사용합니다.

net0 및 net1은 vnet0 및 vnet1 각각에 대한 베니티 이름입니다.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i net0 -i net1 ipmp0
# ipadm create-addr -T static -a 192.168.1.1/24 ipmp0/v4addr1
# ipadm create-addr -T static -a 192.168.1.2/24 ipmp0/v4addr2
```

Logical Domains 1.3 이전 릴리스에서 IPMP 구성 및 사용

Logical Domains 1.3 이전 릴리스에서는 가상 스위치와 가상 네트워크 장치가 링크 오류 감지를 수행할 수 없습니다. 이러한 릴리스에서는 프로브 기반 IPMP를 사용하여 네트워크 오류 감지 및 복구를 설정할 수 있습니다.

게스트 도메인에서 IPMP 구성

게스트 도메인의 가상 네트워크 장치는 [그림 8-7](#) 및 [그림 8-8](#)에 표시된 것과 같이 IPMP 그룹으로 구성할 수 있습니다. 단, 가상 네트워크 장치에서 테스트 주소를 구성하여 프로브 기반 오류 감지를 사용한다는 점만 다릅니다. 프로브 기반 IPMP 구성에 대한 자세한 내용은 [System Administration Guide: IP Services](#)를 참조하십시오.

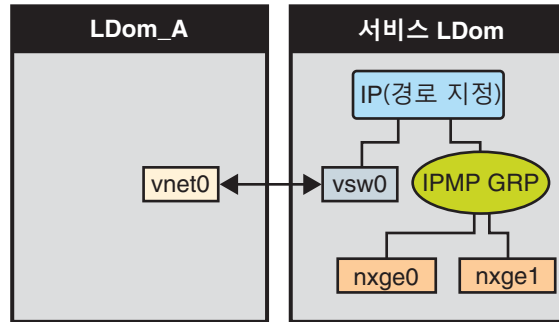
서비스 도메인에서 IPMP 구성

Logical Domains 1.3 이전 릴리스에서는 가상 스위치 장치가 물리적 링크 오류 감지를 수행할 수 없습니다. 이 경우 서비스 도메인의 물리적 인터페이스를 IPMP 그룹으로 구성하여 네트워크 오류 감지 및 복구를 설정할 수 있습니다. 이를 위해서는 서비스 도메인에서 가상 스위치를 구성하되, 가상 스위치에 물리적 네트워크 장치를 지정하지 마십시오. 즉, `ldm add-vswitch` 명령을 사용하여 가상 스위치를 만드는 경우 `net-dev(net-dev=)` 등록 정보에 대한 값을 지정하십시오. 서비스 도메인에서 가상 스위치 인터페이스를 만들고 서비스 도메인 자체가 IP 라우터로 사용되도록 구성하십시오. IP 경로 지정 설정에 대한 자세한 내용은 Oracle Solaris 10 [System Administration Guide: IP Services](#)를 참조하십시오.

일단 구성되면 가상 스위치는 물리적 장치를 통해 직접 패킷을 전송하는 대신 가상 네트워크에서 비롯되어 외부 시스템을 대상으로 하는 모든 패킷을 IP 계층에 전송합니다. 물리적 인터페이스 오류가 발생하면 IP 계층에서 오류를 감지하고 보조 인터페이스를 통해 자동으로 패킷 경로를 다시 지정합니다.

물리적 인터페이스는 직접 IPMP 그룹으로 구성되므로 그룹을 링크 기반 또는 프로브 기반 감지용으로 설정할 수 있습니다. 다음 다이어그램은 IPMP 그룹의 일부로 구성된 두 개의 네트워크 인터페이스(`nxge0` 및 `nxge1`)를 보여줍니다. 가상 스위치 인스턴스(`vsw0`)는 패킷을 IP 계층으로 전송하기 위한 네트워크 장치로 생성되었습니다.

그림 8-10 IPMP 그룹의 일부로 구성된 두 개의 네트워크 인터페이스



주 - 그림 8-10은 Oracle Solaris 10 시스템의 구성을 보여줍니다. Oracle Solaris 11 시스템의 경우에는 nxge0 및 nxge1에 각각 net0 및 net1 등의 일반 이름을 사용하도록 인터페이스 이름만 변경됩니다.

▼ 프로브 기반 IPMP에 대해 호스트 경로를 구성하는 방법

주 - 이 절차는 프로브 기반 IPMP만 지원하는 게스트 도메인 및 1.3 이전 릴리스에만 적용됩니다.

IPMP 인터페이스에 해당하는 네트워크의 라우터에 대해 구성된 명시적 경로가 없는 경우, IPMP 프로브 기반 감지가 예상대로 작동하려면 대상 시스템에 대해 명시적 호스트 경로를 하나 이상 구성해야 합니다. 그렇지 않을 경우 네트워크 오류를 감지하기 위한 프로브 감지가 실패할 수 있습니다.

● 호스트 경로를 구성합니다.

```
# route add -host destination-IP gateway-IP -static
```

예를 들어, 다음과 같습니다.

```
# route add -host 192.168.102.1 192.168.102.1 -static
```

자세한 내용은 [System Administration Guide: IP Services](#)의 “Configuring Target Systems”를 참조하십시오.

VLAN 태그 지정 사용

Oracle VM Server for SPARC 소프트웨어는 네트워크 기반구조에서 802.1Q VLAN 태깅을 지원합니다.

주 - Logical Domains 네트워킹 구성 요소의 이전 릴리스에서는 태그 지정된 VLAN이 지원되지 않습니다.

가상 스위치(vsw) 및 가상 네트워크(vnet) 장치는 VLAN(Virtual Local-Area Network) 식별자(ID)를 기준으로 이더넷 패킷 전환을 지원하며, 이더넷 프레임의 필요한 태그 지정 또는 태그 미지정을 처리합니다.

게스트 도메인에서 vnet 장치를 통해 VLAN 인터페이스를 여러 개 만들 수 있습니다. 가상 네트워크 장치를 통해 VLAN 인터페이스를 만들려면 Oracle Solaris 10 ifconfig 명령 또는 Oracle Solaris 11 dladm 및 ipadm 명령을 사용하십시오. 만드는 방법은 다른 물리적 네트워크 장치를 통해 VLAN 인터페이스를 구성할 때 사용한 방법과 같습니다. Logical Domains 환경의 경우에는 Logical Domains Manager CLI 명령을 사용하여 해당 VLAN에 vnet을 추가로 지정해야 합니다. Logical Domains Manager CLI 명령에 대한 자세한 내용은 [ldm\(1M\)](#)을 참조하십시오.

마찬가지로, 서비스 도메인에서 가상 스위치 장치를 통해 VLAN 인터페이스를 구성할 수 있습니다. VLAN ID 2 - 4094가 유효하며, VLAN ID 1는 default-vlan-id용으로 예약되었습니다.

게스트 도메인에서 vnet 장치를 만들 경우, ldm add-vnet 명령에 pvid= 및 vid= 인수를 사용하여 이 vnet에 대해 포트 VLAN ID 및 0개 이상의 VLAN ID를 지정하는 방식으로 필요한 VLAN에 해당 장치를 지정해야 합니다. 이렇게 하면 Logical Domains 네트워크에서 여러 개의 VLAN을 지원하고 네트워크에서 MAC 주소와 VLAN ID를 모두 사용하여 패킷을 전환하는 가상 스위치가 구성됩니다.

마찬가지로, vsw 장치 자체가 속해 있어야 하는 VLAN을 네트워크 인터페이스로 만들 경우 ldm add-vsw 명령에 pvid= 및 vid= 인수를 사용하여 vsw 장치에 VLAN을 구성해야 합니다.

ldm set-vnet 또는 ldm set-vsw 명령을 사용하여 장치가 속한 VLAN을 변경할 수 있습니다.

PVID(포트 VLAN ID)

PVID는 태그 미지정 모드에서 가상 네트워크 장치가 멤버로 속해 있어야 하는 VLAN을 나타냅니다. 이 경우 vsw 장치는 PVID로 지정된 VLAN을 통해 vnet 장치에 프레임의 필요한 태그 지정 또는 태그 미지정을 제공합니다. 태그가 지정되지 않은 가상 네트워크의 아웃바운드 프레임은 가상 스위치에 의해 PVID로 태그 지정됩니다. 이

PVID로 태그 지정된 인바운드 프레임은 vnet 장치로 전송되기 전에 가상 스위치에 의해 태그 지정이 취소됩니다. 따라서 vnet에 PVID를 지정하면 암시적으로 가상 스위치의 해당 가상 네트워크 포트가 PVID로 지정된 VLAN에 대해 태그 미지정으로 표시됨을 의미합니다. vnet 장치에는 PVID를 한 개만 지정할 수 있습니다.

해당 가상 네트워크 인터페이스가 VLAN ID 없이 해당 장치 인스턴스만 사용하여 구성된 경우 암시적으로 가상 네트워크의 PVID로 지정된 VLAN에 인터페이스가 지정됩니다.

예를 들어 다음 명령 중 하나를 사용하여 vnet 인스턴스 0을 만들고 vnet에 대한 pvid=인수를 10으로 지정한 경우, vnet0 인터페이스는 암시적으로 VLAN 10에 속하는 것으로 지정됩니다. 다음 명령은 Oracle Solaris 10에 해당하는 vnet0 인터페이스 이름을 표시합니다. Oracle Solaris 11의 경우 net0과 같은 일반 이름을 대신 사용하십시오.

- Oracle Solaris 10 OS. ifconfig 명령을 사용합니다.

```
# ifconfig vnet0 plumb
```

- Oracle Solaris 11 OS. ipadm 명령을 사용합니다.

```
# ipadm create-ip net0
```

VID(VLAN ID)

VID는 태그 지정 모드에서 가상 네트워크 장치 또는 가상 스위치가 멤버로 속해 있어야 하는 VLAN을 나타냅니다. 가상 네트워크 장치는 VID로 지정된 VLAN을 통해 태그 지정된 프레임을 보내고 받습니다. 가상 스위치는 지정된 VID로 태그 지정된 프레임을 가상 네트워크 장치와 외부 네트워크 간에 전달합니다.

▼ 가상 스위치 및 가상 네트워크 장치에 VLAN을 지정하는 방법

- 1 두 개의 VLAN에 가상 스위치(vsw)를 지정합니다.

예를 들어 VLAN 21은 태그 미지정으로 구성하고 VLAN 20은 태그 지정으로 구성합니다. 세 개의 VLAN에 가상 네트워크(vnet)를 지정합니다. VLAN 20은 태그 미지정으로 구성하고 VLAN 21 및 22는 태그 지정으로 구성합니다.

```
# ldm add-vsw net-dev=nxge0 pvid=21 vid=20 primary-vsw0 primary
# ldm add-vnet pvid=20 vid=21,22 vnet01 primary-vsw0 ldom1
```

- 2 VLAN 인터페이스를 만듭니다.

이 예는 이러한 장치의 인스턴스 번호가 도메인에서 0이고, VLAN이 해당 서브넷에 매핑되었다고 가정합니다.

VLAN 20 서브넷 192.168.1.0(넷마스크: 255.255.255.0)

VLAN 21 서브넷 192.168.2.0(넷마스크: 255.255.255.0)

VLAN 22 서브넷 192.168.3.0(넷마스크: 255.255.255.0)

a. 서비스(primary) 도메인에 VLAN 인터페이스를 만듭니다.

- Oracle Solaris 10 OS.ifconfig 명령을 사용합니다.

```
primary# ifconfig vsw0 plumb
primary# ifconfig vsw0 192.168.2.100 netmask 0xffffffff00 broadcast + up
primary# ifconfig vsw20000 plumb
primary# ifconfig vsw20000 192.168.1.100 netmask 0xffffffff00 broadcast + up
```

- Oracle Solaris 11 OS.dladm 및 ipadm 명령을 사용합니다.

```
primary# dladm create-vlan -l vsw0 -v20
primary# ipadm create-ip net0
primary# ipadm create-addr -T static -a 192.168.2.100/24 net0/ipv4
primary# ipadm create-ip net20000
primary# ipadm create-addr -T static -a 192.168.1.100/24 net20000/ipv4
```

b. 게스트(ldom1) 도메인에 VLAN 인터페이스를 만듭니다.

- Oracle Solaris 10 OS.ifconfig 명령을 사용합니다.

```
ldom1# ifconfig vnet0 plumb
ldom1# ifconfig vnet0 192.168.1.101 netmask 0xffffffff00 broadcast + up
ldom1# ifconfig vnet21000 plumb
ldom1# ifconfig vnet21000 192.168.2.101 netmask 0xffffffff00 broadcast + up
ldom1# ifconfig vnet22000 plumb
ldom1# ifconfig vnet22000 192.168.3.101 netmask 0xffffffff00 broadcast + up
```

Oracle Solaris 10 OS에서 VLAN 인터페이스를 구성하는 자세한 방법은 [System Administration Guide: IP Services](#)의 “Administering Virtual Local Area Networks”를 참조하십시오.

- Oracle Solaris 11 OS.dladm 및 ipadm 명령을 사용합니다.

```
ldom1# dladm create-vlan -l net0 -v21
ldom1# ipadm create-ip net0
ldom1# ipadm create-addr -T static -a 192.168.1.101/24 net0/ipv4
ldom1# ipadm create-ip net21000
ldom1# ipadm create-addr -T static -a 192.168.2.101/24 net21000/ipv4
ldom1# ipadm create-ip net22000
ldom1# ipadm create-addr -T static -a 192.168.3.101/24 net22000/ipv4
```

Oracle Solaris 11 OS에서 VLAN 인터페이스를 구성하는 자세한 방법은 [Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화의 “VLAN\(가상 LAN\) 관리”](#)을 참조하십시오.

▼ 설치 서버가 VLAN에 있을 때 게스트 도메인을 설치하는 방법

설치 서버가 VLAN에 있을 때 Oracle Solaris JumpStart 기능을 사용하여 네트워크를 통해 게스트 도메인을 설치하는 경우 주의하십시오. 이 기능은 오직 Oracle Solaris 10 시스템에서만 지원됩니다. 설치 서버와 연관된 VLAN ID를 가상 네트워크 장치의 PVID로 지정하고, 해당 네트워크 장치에 대해 태그 지정된 VLAN(vid)을 구성하지 마십시오. OBP에서 VLAN을 인식하지 못해 VLAN 태그가 지정된 네트워크 패킷을 처리할 수 없기 때문에 이와 같이 해야 합니다. 가상 스위치는 네트워크 설치 중 게스트 도메인으로 들어오거나 게스트 도메인에서 나가는 패킷의 태그 미지정 및 태그 지정을 처리합니다. 네트워크 설치가 완료되고 Oracle Solaris OS가 부트되면 해당 VLAN에서 태그가 지정되도록 가상 네트워크 장치를 구성할 수 있습니다. 그런 다음 태그 지정 모드에서 가상 네트워크 장치를 추가 VLAN에 추가할 수 있습니다.

Oracle Solaris JumpStart 기능을 사용하여 게스트 도메인을 설치하는 자세한 방법은 71 페이지 “Oracle Solaris 10 게스트 도메인에서 Oracle Solaris JumpStart 기능을 사용하는 방법”을 참조하십시오.

1 처음에 태그 미지정 모드에서 네트워크 장치를 구성합니다.

예를 들어 설치 서버가 VLAN 21에 있는 경우, 가상 네트워크를 처음에 다음과 같이 구성합니다.

```
primary# ldm add-vnet pvid=21 vnet01 primary-vsw0 ldom1
```

2 설치가 완료되고 Oracle Solaris OS가 부트되면 태그 지정 모드에서 가상 네트워크를 구성합니다.

```
primary# ldm set-vnet pvid= vid=21, 22, 23 vnet01 primary-vsw0 ldom1
```

NIU 하이브리드 I/O 사용

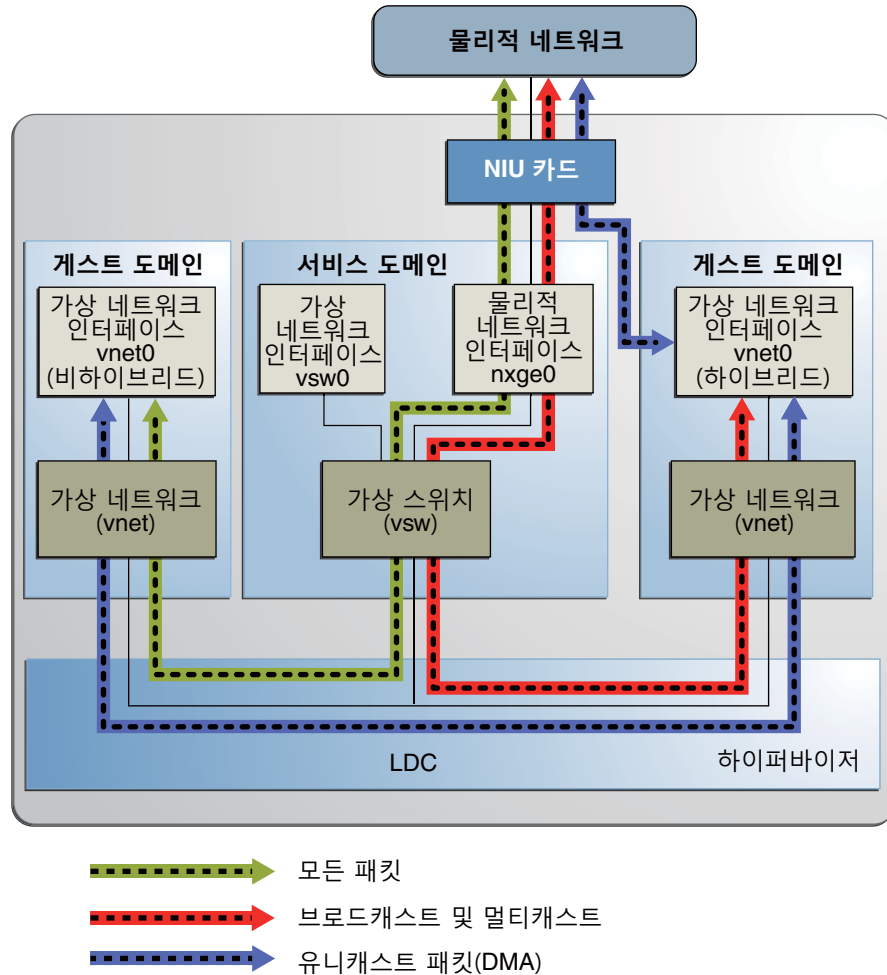
가상 I/O 프레임워크는 향상된 기능과 성능을 위해 하이브리드 I/O 모델을 구현합니다. 하이브리드 I/O 모델은 I/O 리소스가 가상 시스템에 융통적으로 배포될 수 있도록 직접 I/O와 가상화된 I/O를 결합합니다. 따라서 직접 I/O가 가상 시스템에 완전한 기능을 제공하지 않거나, 가상 시스템에서 직접 I/O를 지속적으로 사용할 수 없는 경우에 특히 유용합니다. 이러한 상황은 리소스 가용성 또는 가상 시스템 마이그레이션으로 인해 발생할 수 있습니다. 하이브리드 I/O 구조는 Sun UltraSPARC T2, SPARC T3 및 SPARC T4 플랫폼에서 NIU(Network Interface Unit)에 적합합니다. NIU는 칩에 통합된 네트워크 I/O 인터페이스입니다. 이 구조를 사용하면 DMA(Direct Memory Access) 리소스를 가상 네트워크 장치에 동적으로 지정할 수 있으므로, 도메인 내의 응용 프로그램에 일관된 성능이 제공됩니다.

NIU 하이브리드 I/O는 Sun UltraSPARC T2, SPARC T3 및 SPARC T4 플랫폼에서 사용할 수 있습니다. 이 기능은 선택적 하이브리드 모드를 통해 사용으로 설정할 수 있는데, 이

모드는 향상된 성능을 위해 DMA 하드웨어 리소스를 게스트 도메인의 **vnet** 장치에 임대하는 가상 네트워크(**vnet**) 장치에 제공됩니다. 하이브리드 모드에서는 게스트 도메인의 **vnet** 장치가 DMA 하드웨어 리소스를 사용하여 유니캐스트 트래픽을 외부 네트워크에서 게스트 도메인으로 직접 보내고 받을 수 있습니다. 같은 시스템의 다른 게스트 도메인에 대해 수행되는 브로드캐스트 또는 멀티캐스트 트래픽과 유니캐스트 트래픽은 계속 가상 I/O 통신 방식을 사용하여 계속 전송됩니다.

주 - UltraSPARC T2 Plus 플랫폼에서는 NIU 하이브리드 I/O를 사용할 수 없습니다.

그림 8-11 하이브리드 가상 네트워킹



주 - 그림 8-11은 Oracle Solaris 10 시스템의 구성을 보여줍니다. Oracle Solaris 11 시스템의 경우에는 nxge0에 net0과 같은 일반 이름을 사용하도록 인터페이스 이름만 변경됩니다.

하이브리드 모드는 NIU 네트워크 장치를 사용하도록 구성된 가상 스위치(vsw)와 연관된 vnet 장치에만 적용됩니다. 공유 가능한 DMA 하드웨어 리소스는 제한적이므로, 지정된 시간에 vsw당 최대 세 개의 vnet 장치에만 DMA 하드웨어 리소스를 지정할 수 있습니다. 넷 이상의 vnet 장치에서 하이브리드 모드가 사용으로 설정된 경우 선착순으로 지정이

수행됩니다. 한 시스템에는 두 개의 NIU 네트워크 장치가 있으므로, DMA 하드웨어 리소스가 지정된 두 개의 가상 스위치에는 총 6개의 vnet 장치가 있을 수 있습니다.

이 기능을 사용할 때 유의해야 할 사항은 다음과 같습니다.

- vnet 장치에 대한 하이브리드 모드 옵션은 제안 사항으로 취급됩니다. 즉, DMA 리소스는 사용 가능하고 장치가 이 리소스를 사용할 수 있는 경우에만 지정됩니다.
- Logical Domains Manager CLI 명령은 하이브리드 모드 옵션을 검증하지 않습니다. 즉, 하이브리드 모드는 원하는 vnet 또는 원하는 개수의 vnet 장치에 설정할 수 있습니다.
- 게스트 도메인과 서비스 도메인은 최소한 Oracle Solaris 10 10/08 OS를 실행해야 합니다.
- 지정된 시간에 vsw당 최대 세 개의 vnet 장치에만 DMA 하드웨어 리소스를 임대할 수 있습니다. 두 개의 NIU 네트워크 장치가 있으므로, DMA 하드웨어 리소스가 임대된 vnet 장치는 총 6개가 있을 수 있습니다.

주 - vsw당 세 개의 vnet 장치에 대해서만 하이브리드 모드를 설정해야 DMA 하드웨어 리소스가 지정될 수 있습니다.

- vnet 장치의 경우 기본적으로 하이브리드 모드가 사용 안함으로 설정되어 있습니다. ldm 명령을 사용하여 명시적으로 사용으로 설정해야 합니다. [169 페이지 “하이브리드 모드를 사용으로 설정하는 방법”](#) 및 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- 게스트 도메인이 활성 상태인 동안에는 하이브리드 모드를 동적으로 변경할 수 없습니다.
- DMA 하드웨어 리소스는 게스트 도메인에 만들어진 vnet 장치가 활성 상태일 경우에만 지정됩니다.
- NIU 10GB 이더넷 드라이버(nxge)가 NIU 카드에 사용됩니다. 다른 10GB 네트워크 카드에도 같은 드라이버가 사용됩니다. 그러나 NIU 하이브리드 I/O 기능은 NIU 네트워크 장치에만 사용할 수 있습니다.

▼ NIU 네트워크 장치로 가상 스위치를 구성하는 방법

1 NIU 네트워크 장치를 확인합니다.

다음 예는 UltraSPARC T2 서버의 출력을 보여줍니다.

```
# grep nxge /etc/path_to_inst
"/niu@80/network@0" 0 "nxge"
"/niu@80/network@1" 1 "nxge"
```

다음 예는 SPARC T3-1 또는 SPARC T4-1 서버의 출력을 보여줍니다.

```
# grep nxge /etc/path_to_inst
"/niu@480/network@0" 0 "nxge"
"/niu@480/network@1" 1 "nxge"
```

- 2 Oracle Solaris 11 OS만 해당: NIU 네트워크 장치에 해당하는 링크 이름(예: nxge0)을 지정합니다.

```
primary# dladm show-phys -L |grep nxge0
net2                nxge0                /SYS/MB
```

- 3 가상 스위치를 구성합니다.

- Oracle Solaris 10 OS. 다음 명령을 사용합니다.

```
# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

- Oracle Solaris 11 OS. 다음 명령을 사용합니다.

다음 예는 nxge0 대신 net0을 사용합니다.

```
# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

▼ 하이브리드 모드를 사용으로 설정하는 방법

- 예를 들어 vnet 장치를 생성하는 중 이 장치에 대해 하이브리드 모드를 사용으로 설정합니다.

```
# ldm add-vnet mode=hybrid vnet01 primary-vsw0 ldom01
```

▼ 하이브리드 모드를 사용 안함으로 설정하는 방법

- 예를 들어 vnet 장치를 생성하는 중 이 장치에 대해 하이브리드 모드를 사용 안함으로 설정합니다.

```
# ldm set-vnet mode= vnet01 ldom01
```

가상 스위치에서 링크 통합 사용

링크 통합을 사용하도록 가상 스위치를 구성할 수 있습니다. 링크 통합은 가상 스위치의 네트워크 장치로 사용되어 물리적 네트워크에 연결합니다. 이 구성을 사용하면 가상 스위치가 IEEE 802.3ad Link Aggregation Standard에서 제공하는 기능을 활용할 수 있습니다. 이러한 기능에는 대역폭 증가, 로드 균형 조정 및 페일오버가 포함됩니다. 링크 통합을 구성하는 자세한 방법은 [System Administration Guide: IP Services](#)를 참조하십시오.

링크 통합을 만든 후에는 가상 스위치에 지정할 수 있습니다. 이와 같이 지정하는 것은 물리적 네트워크 장치를 가상 스위치에 지정하는 것과 비슷합니다. `net-dev` 등록 정보를 설정하려면 `ldm add-vswitch` 또는 `ldm set-vswitch` 명령을 사용하십시오.

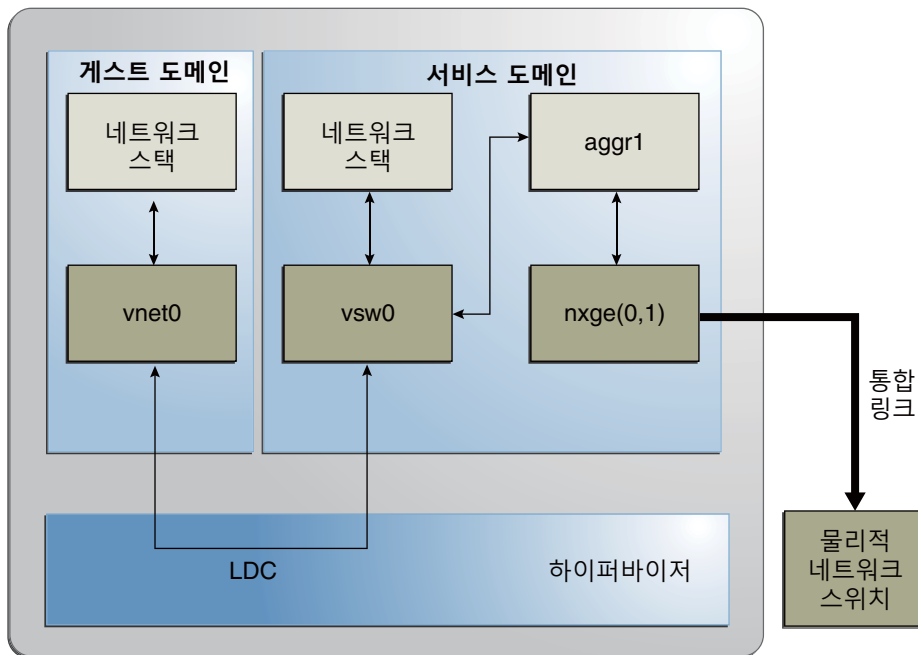
링크 통합이 가상 스위치에 지정되면 물리적 네트워크로 들어오거나 물리적 네트워크에서 나가는 트래픽은 통합을 통해 이동합니다. 필요한 균형 조정 또는 패일오버는 기본 통합 프레임워크를 통해 투명하게 처리됩니다. 링크 통합은 게스트 도메인에 있으며 통합을 사용하는 가상 스위치에 바인드된 가상 네트워크(vnet) 장치에 전혀 영향을 주지 않습니다.

주 - 가상 네트워크 장치(vnet 및 vsw)는 링크 통합으로 그룹화할 수 없습니다.

서비스 도메인에서 링크 통합을 사용하도록 구성된 가상 스위치를 만들어 사용할 수 있습니다. 62 페이지 “가상 스위치를 주 인터페이스로 구성하는 방법”을 참조하십시오.

다음 그림은 링크 통합을 사용하도록 구성된 가상 스위치 `aggr1`, 물리적 인터페이스 `nxge0` 및 `nxge1`을 보여줍니다.

그림 8-12 링크 통합을 사용하도록 가상 스위치 구성



주 - [그림 8-12](#)는 Oracle Solaris 10 시스템의 구성을 보여줍니다. Oracle Solaris 11 시스템의 경우에는 `nxge0` 및 `nxge1`에 각각 `net0` 및 `net1` 등의 일반 이름을 사용하도록 인터페이스 이름만 변경됩니다.

점보 프레임 구성

Logical Domains 가상 스위치(vsw) 및 가상 네트워크(vnet) 장치는 이제 페이로드 크기가 1500바이트를 초과하는 이더넷 프레임을 지원할 수 있습니다. 이 변경 사항으로 인해 이러한 드라이버의 네트워크 처리량을 늘릴 수 있습니다.

▼ 점보 프레임을 사용하도록 가상 네트워크 및 가상 스위치 장치를 구성하는 방법

점보 프레임을 사용으로 설정하려면 가상 스위치 장치의 MTU(최대 전송 단위)를 지정하십시오. 이 경우 가상 스위치 장치 및 가상 스위치 장치에 바인드된 모든 가상 네트워크 장치가 지정된 MTU 값을 사용합니다.

특정 경우 가상 네트워크 장치에 직접 MTU 값을 지정할 수도 있습니다. 가상 네트워크 장치에 필요한 MTU 값이 가상 스위치에서 지원하는 값보다 작을 경우 이와 같이 지정할 수 있습니다.

주 - Oracle Solaris 10 5/09 OS에서는 물리적 장치의 MTU가 가상 스위치의 MTU와 일치하도록 구성해야 합니다. 특정 드라이버 구성에 대한 자세한 내용은 Oracle Solaris 참조 설명서의 7D절에서 해당 드라이버와 일치하는 매뉴얼 페이지를 참조하십시오. 예를 들어 Oracle Solaris 10 `nxge` 드라이버에 대한 정보는 [nxge\(7D\)](#) 매뉴얼 페이지를 참조하십시오.

- 1 컨트롤 도메인에 로그인합니다.
- 2 관리자, 수퍼유저 또는 동등한 역할의 사용자로 로그인합니다.
Oracle Solaris 10의 경우, [System Administration Guide: Security Services](#)의 “Configuring RBAC (Task Map)”를 참조하십시오. Oracle Solaris 11의 경우, [Oracle Solaris 관리: 보안 서비스의 제III부](#), “역할, 권한 프로파일 및 권한”를 참조하십시오.
- 3 가상 네트워크에 사용할 MTU 값을 확인합니다.
MTU 값은 1500 - 16000바이트 범위에서 지정할 수 있습니다. 지정된 MTU는 가상 스위치에 지정된 물리적 네트워크 장치의 MTU와 일치해야 합니다.

4 가상 스위치 또는 가상 네트워크 장치의 MTU 값을 지정합니다.

다음 중 하나를 수행합니다.

- 해당 MTU를 `mtu` 등록 정보의 값으로 지정하여 서비스 도메인의 새 가상 스위치 장치에 점보 프레임을 사용으로 설정합니다.

```
# ldm add-vsw mtu=value vswitch-name ldom
```

이 명령은 가상 스위치를 구성하는 것 이외에도 이 가상 스위치에 바인드될 각 가상 네트워크 장치의 MTU 값을 업데이트합니다.

- 해당 MTU를 `mtu` 등록 정보의 값으로 지정하여 서비스 도메인의 기존 가상 스위치 장치에 점보 프레임을 사용으로 설정합니다.

```
# ldm set-vsw mtu=value vswitch-name
```

이 명령은 가상 스위치를 구성하는 것 이외에도 이 가상 스위치에 바인드될 각 가상 네트워크 장치의 MTU 값을 업데이트합니다.

드문 경우지만, `ldm add-vnet` 또는 `ldm set-vnet` 명령을 사용하여 가상 네트워크 장치의 MTU 값을 가상 스위치의 MTU 값과 다르게 지정해야 할 수 있습니다. 예를 들어 가상 네트워크 장치에 대해 VLAN을 구성하고 가장 큰 VLAN MTU가 가상 스위치의 MTU 값보다 작을 경우 가상 네트워크 장치의 MTU 값을 변경할 수 있습니다. 기본 MTU 값만 사용되는 도메인에는 점보 프레임을 지원하는 `vnet` 드라이버가 필요하지 않을 수 있습니다. 그러나 도메인에서 가상 네트워크 장치가 점보 프레임을 사용하는 가상 스위치에 바인드된 경우, `vnet` 드라이버가 점보 프레임을 지원하는지 확인하십시오.

`ldm set-vnet` 명령을 사용하여 가상 네트워크 장치에서 `mtu` 값을 지정하는 경우 가상 스위치 장치의 MTU 값을 이후에 업데이트하면 업데이트 사항이 해당 가상 네트워크 장치에 전파되지 않습니다. 가상 네트워크 장치를 다시 사용으로 설정하여 가상 스위치 장치에서 MTU 값을 가져오려면 다음 명령을 실행하십시오.

```
# ldm set-vnet mtu= vnet-name ldom
```

가상 네트워크 장치의 점보 프레임을 사용으로 설정하면 해당 가상 네트워크 장치에 지정된 하이브리드 IO 리소스에 대한 점보 프레임이 자동으로 사용으로 설정됩니다.

컨트롤 도메인에서 Logical Domains Manager는 `ldm set-vsw` 및 `ldm set-vnet` 명령으로 시작된 MTU 값을 지연된 재구성 작업으로 업데이트합니다. 컨트롤 도메인이 아닌 다른 도메인에서 MTU를 업데이트하려면 `ldm set-vsw` 또는 `ldm set-vnet` 명령을 실행하여 MTU 값을 수정하기 전에 도메인을 중지해야 합니다.

예 8-2 가상 스위치 및 가상 네트워크 장치에서 점보 프레임 구성

- 다음 예는 MTU 값 9000을 사용하는 가상 스위치 장치를 새로 추가하는 방법을 보여줍니다. MTU 값은 가상 스위치 장치에서 모든 클라이언트 가상 네트워크 장치로 전파됩니다.

먼저, `ldm add-vsw` 명령이 MTU 값으로 9000을 사용하는 가상 스위치 장치 `primary-vsw0`을 만듭니다. 네트워크 장치 `nxge0`의 인스턴스 0이 `net-dev` 등록 정보의 값으로 지정됩니다.

```
# ldm add-vsw net-dev=nxge0 mtu=9000 primary-vsw0 primary
```

다음으로, `ldm add-vnet` 명령이 클라이언트 가상 네트워크 장치를 이 가상 스위치 `primary-vsw0`에 추가합니다. 가상 네트워크 장치의 MTU는 암시적으로 가상 네트워크 장치가 바인드된 가상 스위치에서 지정됩니다. 그 결과 `ldm add-vnet` 명령에 `mtu` 등록 정보의 값을 지정할 필요가 없습니다.

```
# ldm add-vnet vnet01 primary-vsw0 ldom1
```

실행 중인 Oracle Solaris OS의 버전에 따라 다음을 수행하십시오.

- **Oracle Solaris 10 OS.** `ifconfig` 명령이 서비스 도메인 `primary`에 가상 스위치 인터페이스를 만듭니다. `ifconfig vsw0` 명령 출력은 `mtu` 등록 정보의 값이 9000임을 보여줍니다.

```
# ifconfig vsw0 plumb
# ifconfig vsw0 192.168.1.100/24 up
# ifconfig vsw0
vsw0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 5
inet 192.168.1.100 netmask ffffffff0 broadcast 192.168.1.255
ether 0:14:4f:fa:0:99
```

`ifconfig` 명령이 게스트 도메인 `ldom1`에 가상 네트워크 인터페이스를 만듭니다. `ifconfig vnet0` 명령 출력은 `mtu` 등록 정보의 값이 9000임을 보여줍니다.

```
# ifconfig vnet0 plumb
# ifconfig vnet0 192.168.1.101/24 up
# ifconfig vnet0
vnet0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 4
inet 192.168.1.101 netmask ffffffff0 broadcast 192.168.1.255
ether 0:14:4f:f9:c4:13
```

- **Oracle Solaris 11 OS.** `ipadm` 및 `dladm` 명령을 사용하여 기본 인터페이스의 `mtu` 등록 정보 값을 9000으로 변경합니다.

```
# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0    mtu      ipv4  rw   9000    --          9000    68-9000
```

`ipadm` 명령이 게스트 도메인 `ldom1`에 가상 네트워크 인터페이스를 만듭니다. `ipadm show-ifprop` 명령 출력은 `mtu` 등록 정보의 값이 9000임을 보여줍니다.

```
# ipadm create-ip net0
# ipadm create-addr -T static -a 192.168.1.101/24 net0/ipv4
# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0    mtu      ipv4  rw   9000    --          9000    68-9000
```

- 다음 예는 인터페이스의 MTU를 4000으로 변경하는 방법을 보여줍니다.

인터페이스의 MTU는 Logical Domains Manager에서 지정한 장치의 MTU보다 작은 값으로만 변경할 수 있습니다. 이 방법은 VLAN이 구성되어 있고 각 VLAN 인터페이스에 다른 MTU가 필요한 경우에 유용합니다.

- **Oracle Solaris 10 OS. ifconfig 명령을 사용합니다.**

```
# ifconfig vnet0 mtu 4000
# ifconfig vnet0
vnet0: flags=1201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS,FIXEDMTU>
mtu 4000 index 4
    inet 192.168.1.101 netmask ffffffff00 broadcast 192.168.1.255
    ether 0:14:4f:f9:c4:13
```

- **Oracle Solaris 11 OS. ipadm 명령을 사용합니다.**

```
# ipadm set-ifprop -p mtu=4000 net0
# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0    mtu          ipv4    rw    4000    --        9000    68-9000
```

vnet 및 vsw 드라이버의 이전(점보 미인식) 버전과의 호환성(Oracle Solaris 10)

주 - 이 단원은 Oracle Solaris 10 OS에만 적용됩니다.

점보 프레임을 지원하는 드라이버는 같은 시스템의 점보 프레임을 지원하지 않는 드라이버와 상호 운용될 수 있습니다. 가상 스위치를 만들 때 점보 프레임 지원이 사용으로 설정되지 않는 한 이 상호 운용성이 지원됩니다.

주 - 가상 스위치와 연관된 게스트 또는 서비스 도메인에서 점보 프레임을 지원하는 Logical Domains 드라이버를 사용하지 않는 경우 mtu 등록 정보를 설정하지 마십시오.

가상 스위치의 mtu 등록 정보를 기본값 1500에서 변경하여 점보 프레임을 사용으로 설정할 수 있습니다. 이 경우 이전 드라이버 버전은 mtu 설정을 무시하고 기본값을 계속 사용합니다. `ldm list` 출력에는 기본값이 아닌 사용자가 지정한 MTU 값이 표시됩니다. 기본 MTU보다 큰 프레임은 해당 장치로 전송되지 않고 새 드라이버에 의해 삭제됩니다. 이러한 상황으로 인해 이전 드라이버를 여전히 사용하는 게스트에서 일관되지 않는 네트워크 동작이 발생할 수 있습니다. 이러한 상황은 클라이언트 게스트 도메인과 서비스 도메인 모두에 적용됩니다.

따라서 점보 프레임이 사용으로 설정된 경우, Logical Domains 네트워크의 모든 가상 장치가 점보 프레임을 지원하는 새 드라이버로 업그레이드되었는지 확인하십시오. 점보 프레임을 구성하려면 Logical Domains 1.2 이상이 실행 중이어야 합니다.

Oracle Solaris 11 네트워킹 관련 기능 차이점

도메인에서 Oracle Solaris 10 OS가 실행되는 경우 일부 Oracle VM Server for SPARC 네트워킹 기능이 Oracle Solaris 11 OS에서와 다르게 작동합니다. 도메인에서 Oracle Solaris 11 OS가 실행되는 경우 Oracle VM Server for SPARC 가상 네트워크 장치 및 가상 스위치에 대한 기능 차이점은 다음과 같습니다.

- **vswn** 장치를 기본 네트워크 인터페이스로 구성하면 서비스 도메인이 게스트 도메인과 통신할 수 있음

이 구성은 오직 Oracle Solaris 10 OS를 실행하는 도메인에만 필요합니다. Oracle Solaris 11의 경우 가상 스위치가 Oracle Solaris 11 네트워크 스택을 사용하므로, 가상 네트워크 장치가 자동으로 백엔드 장치(예: `net0`)에 해당하는 네트워크 인터페이스와 통신할 수 있습니다.

- **Oracle Solaris 11 etherstub** 장치를 백엔드 장치로 사용하면 개인 가상 스위치가 생성됨

이 장치를 사용하면 게스트 도메인이 Oracle Solaris 11 서비스 도메인에 구성된 영역과 통신할 수 있습니다.

- **가상 스위치 및 가상 네트워크 장치에 일반 이름 사용**

Oracle Solaris 11 OS에서는 `vswn` 및 `vnetn` 장치에 일반 이름이 지정됩니다. 따라서 다른 `vsw` 또는 `vnet` 장치인 백엔드 장치로 가상 스위치를 만들지 마십시오. 일반 네트워크 장치 이름과 연관된 실제 물리적 장치를 확인하려면 `dladm show-phys` 명령을 사용하십시오.

- **가상 스위치 및 가상 네트워크 장치에 VNIC 사용**

`vswn` 및 `vnetn` 장치에는 VNIC를 사용할 수 없습니다. 따라서 `vswn` 및 `vnetn`에 VNIC를 만들려고 하는 시도가 실패합니다. **Oracle VM Server for SPARC 2.2 Release Notes**에서 **Oracle VM Server for SPARC 2.2 릴리스 노트의 “Oracle Solaris 11: 자동 네트워크 인터페이스로 구성된 영역을 시작하지 못할 수 있음”**를 참조하십시오.

도메인 마이그레이션

이 장에서는 호스트 시스템 간에 도메인을 마이그레이션하는 방법에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- 178 페이지 “도메인 마이그레이션 소개”
- 178 페이지 “마이그레이션 작업 개요”
- 179 페이지 “소프트웨어 호환성”
- 179 페이지 “마이그레이션 작업 보안”
- 180 페이지 “도메인 마이그레이션”
- 181 페이지 “활성 도메인 마이그레이션”
- 186 페이지 “바인딩된 도메인 또는 비활성 도메인 마이그레이션”
- 180 페이지 “Dry Run 수행”
- 186 페이지 “진행 중인 마이그레이션 모니터”
- 187 페이지 “진행 중인 마이그레이션 취소”
- 188 페이지 “실패한 마이그레이션 복구”
- 180 페이지 “비대화식 마이그레이션 수행”
- 188 페이지 “마이그레이션 예”

주 - 이 장에서 설명되는 마이그레이션 기능을 사용하려면 최신 버전의 Logical Domains Manager, 시스템 펌웨어 및 Oracle Solaris OS를 실행 중이어야 합니다. 이전 버전의 Oracle VM Server for SPARC를 사용하여 마이그레이션하는 방법은 **Oracle VM Server for SPARC 2.2 릴리스 노트** 및 관리 설명서의 관련 버전을 참조하십시오.

도메인 마이그레이션 소개

도메인 마이그레이션을 통해 호스트 시스템 간에 게스트 도메인을 마이그레이션할 수 있습니다. 마이그레이션이 시작되는 시스템을 **소스 시스템**이라고 합니다. 도메인이 마이그레이션되는 시스템을 **대상 시스템**이라고 합니다.

마이그레이션 작업이 진행되는 동안 소스 시스템의 **마이그레이션할 도메인**이 대상 시스템의 **마이그레이션 대상 도메인**으로 전송됩니다.

라이브 마이그레이션 기능은 활성 도메인을 계속 실행하면서 마이그레이션할 수 있도록 하여 성능을 향상시킵니다. 라이브 마이그레이션 외에 바인딩된 도메인 또는 비활성 도메인을 마이그레이션할 수 있는데, 이를 **콜드 마이그레이션**이라고 합니다.

도메인 마이그레이션을 사용하여 다음과 같은 작업을 수행할 수 있습니다.

- 시스템 간의 로드 균형 조정
- 게스트 도메인을 계속 실행하면서 하드웨어 유지 관리 수행

마이그레이션 작업 개요

소스 시스템의 Logical Domains Manager는 도메인 마이그레이션 요청을 수락하고 대상 시스템에서 실행되는 Logical Domains Manager에 대한 보안 네트워크 연결을 설정합니다. 마이그레이션은 이 연결이 설정된 후 발생합니다. 마이그레이션 작업은 다음 단계에 따라 수행됩니다.

1단계: 소스 시스템이 대상 시스템에서 실행되는 Logical Domains Manager에 대한 연결을 설정하면 소스 시스템 및 마이그레이션할 도메인에 대한 정보가 대상 시스템으로 전송됩니다. 이 정보는 마이그레이션 가능 여부를 확인하기 위해 일련의 검사를 수행하는 데 사용됩니다. 수행할 검사는 마이그레이션할 도메인의 상태를 기반으로 합니다. 예를 들어, 마이그레이션할 도메인이 활성 상태이면 해당 도메인이 바인딩 또는 비활성 상태인 경우와는 다른 일련의 검사가 수행됩니다.

2단계: 1단계의 모든 검사가 성공하면 소스 및 대상 시스템이 마이그레이션을 준비합니다. 대상 시스템에서는 마이그레이션할 도메인을 수신할 도메인이 만들어집니다. 마이그레이션할 도메인이 비활성 또는 바인딩 상태이면 마이그레이션 작업이 5단계를 진행합니다.

3단계: 마이그레이션할 도메인이 활성 상태이면 도메인의 런타임 상태 정보가 대상 시스템으로 전송됩니다. 마이그레이션할 도메인이 계속 실행되고 동시에 Logical Domains Manager는 OS가 이 도메인에 대해 적용 중인 수정 사항을 추적합니다. 이 정보는 소스 시스템의 하이퍼바이저에서 검색되어 대상 시스템의 하이퍼바이저에 설치됩니다.

4단계: 마이그레이션할 도메인이 일시 중지됩니다. 이 단계에서 수정된 나머지 상태 정보가 모두 대상 시스템으로 다시 복사됩니다. 따라서 도메인 중단이 거의 발생하지 않거나 아주 잠깐 동안 발생합니다. 중단 시간은 작업 부하에 따라 다릅니다.

5단계: 소스 시스템의 Logical Domains Manager에서 대상 시스템의 Logical Domains Manager로 전달이 발생합니다. 마이그레이션 대상 도메인의 실행이 재개될 때 전달이 발생(마이그레이션할 도메인이 활성 상태인 경우)하며 소스 시스템의 도메인은 삭제됩니다. 이때부터 마이그레이션 대상 도메인은 실행 중인 도메인의 유일한 버전입니다.

소프트웨어 호환성

마이그레이션을 수행하려면 소스 시스템과 대상 시스템이 다음과 같이 호환되는 소프트웨어를 실행 중이어야 합니다.

- Logical Domains Manager 버전 2.1 이상이 두 시스템에서 실행 중이어야 합니다.
- 라이브 마이그레이션이 지원되도록 소스 시스템과 대상 시스템에 호환되는 펌웨어 버전이 설치되어 있어야 합니다. [Oracle VM Server for SPARC 2.2 릴리스 노트의 “라이브 마이그레이션 요구 사항”](#)를 참조하십시오.

마이그레이션 작업 보안

Oracle VM Server for SPARC에서는 마이그레이션 작업을 위해 다음과 같은 보안 기능을 제공합니다.

- **인증.** 마이그레이션 작업은 두 시스템에서 실행되므로 사용자는 소스 시스템과 대상 시스템에서 모두 인증되어야 합니다. 특히 슈퍼유저 이외의 다른 사용자는 LDoms Management 권한 프로파일을 사용해야 합니다.

ldmmigrate-domain 명령은 대상 시스템에서의 인증을 위해 선택적으로 대체 사용자 이름을 지정할 수 있도록 허용합니다. 이 대체 사용자 이름을 지정하지 않을 경우 마이그레이션 명령을 실행 중인 사용자의 사용자 이름이 사용됩니다. [예 9-2](#)를 참조하십시오. 어떤 경우에는 -p 옵션을 사용하여 비대화식 마이그레이션을 시작하지 않는 한 대상 시스템에 대한 암호를 제공하라는 메시지가 표시됩니다. [180 페이지 “비대화식 마이그레이션 수행”](#)을 참조하십시오.

- **암호화.** Oracle VM Server for SPARC에서는 SSL을 통해 마이그레이션 트래픽을 암호화하여 중요한 데이터가 악용되지 않도록 보호하고 추가 하드웨어 및 전용 네트워크에 대한 요구 사항이 발생하지 않도록 합니다.

암호화 장치가 있는 플랫폼에서 소스 및 대상 시스템의 primary 도메인에 암호화 장치가 지정된 경우 마이그레이션 작업 속도가 빨라집니다. SSL 작업이 암호화 장치로 오프로드될 수 있기 때문입니다.

도메인 마이그레이션

`ldm migrate-domain` 명령을 사용하여 호스트 시스템 간에 도메인 마이그레이션을 시작할 수 있습니다.

활성 도메인을 계속 실행하면서 마이그레이션하는 방법은 [181 페이지 “활성 도메인 마이그레이션”](#)을 참조하십시오. 바인딩된 도메인 또는 비활성 도메인을 마이그레이션하는 방법은 [186 페이지 “바인딩된 도메인 또는 비활성 도메인 마이그레이션”](#)을 참조하십시오.

마이그레이션 옵션 및 피연산자에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

Dry Run 수행

`ldm migrate-domain` 명령에 `-n` 옵션을 제공하면 마이그레이션 검사가 수행되지만 도메인이 마이그레이션되지는 않습니다. 요구 사항이 충족되지 않으면 오류로 보고됩니다. Dry Run 결과를 통해 실제 마이그레이션을 시도하기 전에 구성 오류를 수정할 수 있습니다.

주 - 논리적 도메인의 동적 특성으로 인해 Dry Run은 성공해도 실제 마이그레이션은 실패하는 경우가 있으며 Dry Run은 실패해도 실제 마이그레이션은 성공하는 경우도 있습니다.

비대화식 마이그레이션 수행

`ldm migrate-domain -p filename` 명령을 사용하여 비대화식 마이그레이션 작업을 시작할 수 있습니다.

`-p` 옵션에 대해 인수로 지정하는 파일 이름은 다음 특성을 가져야 합니다.

- 파일의 첫번째 행에 암호가 포함되어야 합니다.
- 암호는 일반 텍스트여야 합니다.
- 암호는 256자를 초과하지 않아야 합니다.

암호 끝의 줄바꿈 문자와 첫번째 행 뒤에 오는 모든 행은 무시됩니다.

대상 시스템의 암호를 저장한 파일은 적절히 보안되어야 합니다. 이 방식으로 암호를 저장하려면 루트 소유자 또는 권한이 있는 사용자만 파일을 읽거나 쓸 수 있도록 파일 권한이 설정되었는지 확인하십시오(400 또는 600).

활성 도메인 마이그레이션

활성 도메인을 마이그레이션하려는 경우 마이그레이션할 도메인, 소스 시스템 및 대상 시스템에는 특정 요구 사항 및 제한 사항이 적용됩니다. 자세한 내용은 [Oracle VM Server for SPARC 2.2 릴리스 노트의 “도메인 마이그레이션 제한 사항”](#)을 참조하십시오.

참고 - 소스 시스템과 대상 시스템의 primary 도메인에 가상 CPU를 더 추가하여 전체 마이그레이션 시간을 단축시킬 수 있습니다. 각 primary 도메인에는 16개 이상의 CPU가 있는 것이 좋지만 요구 사항은 아닙니다.

마이그레이션 프로세스를 수행하는 동안 도메인 "시간 손실"이 발생합니다. 이러한 시간 손실 문제를 줄이려면 마이그레이션할 도메인을 NTP(Network Time Protocol) 서버와 같은 외부 시간 소스와 동기화하십시오. 도메인을 NTP 클라이언트로 구성하면 마이그레이션이 완료된 후 바로 도메인의 날짜 및 시간이 수정됩니다.

도메인을 Oracle Solaris 10 NTP 클라이언트로 구성하려면 [System Administration Guide: Network Services](#)의 “Managing Network Time Protocol (Tasks)”를 참조하십시오. 도메인을 Oracle Solaris 11 NTP 클라이언트로 구성하려면 [Oracle Solaris 관리: 네트워크 서비스의 “NTP\(Network Time Protocol\) 관리\(작업\)”](#)를 참조하십시오.

CPU에 대한 도메인 마이그레이션 요구 사항

마이그레이션을 수행할 때 CPU에 적용되는 요구 사항 및 제한 사항은 다음과 같습니다.

- 대상 시스템에는 마이그레이션할 도메인에서 사용 중인 가상 CPU 수가 수용되도록 사용 가능한 가상 CPU가 충분해야 합니다.
- Oracle Solaris 10 OS를 실행하는 시스템의 경우 소스 시스템과 대상 시스템의 프로세서 유형이 동일해야 합니다.
- Oracle Solaris 11 OS의 경우 cpu-arch 등록 정보를 설정하면 프로세서 유형이 다른 시스템 간에 마이그레이션을 수행할 수 있습니다. 지원되는 cpu-arch 등록 정보 값은 다음과 같습니다.
 - native는 CPU 유형이 동일한 플랫폼 간에만 게스트 도메인을 마이그레이션할 수 있도록 하는 CPU 특정 하드웨어 기능을 사용합니다. native가 기본값입니다.
 - generic은 CPU 유형에 관계없이 게스트 도메인을 마이그레이션할 수 있도록 하는 공통 CPU 하드웨어 기능을 사용합니다.

generic 값을 사용하면 native 값을 사용할 때 비해 성능이 저하될 수 있습니다. 성능 저하가 발생하는 것은 게스트 도메인이 특정 CPU의 고유 하드웨어 기능을 사용하지 않고 지원되는 모든 CPU 유형에서 사용 가능한 일반 CPU 기능만 사용하기 때문입니다. 이러한 기능을 사용하지 않음으로써 generic 값은 다른 기능을 지원하는 CPU를 사용하는 시스템 간에 도메인을 마이그레이션할 수 있는 유연성을 가능하게 합니다.

프로세서 유형을 확인하려면 다음과 같이 `psrinfo -pv` 명령을 사용하십시오.

```
# psrinfo -pv
The physical processor has 8 virtual processors (0-7)
SPARC-T4 (chipid 0, clock 2548 MHz)
```

- 마이그레이션할 도메인에서 Oracle Solaris 11 OS가 실행되는 경우 프로세서 주파수가 다르며 `STICK` 주파수 값을 사용하는 소스 시스템과 대상 시스템 간에 도메인을 마이그레이션할 수 있습니다. `cpu-arch` 등록 정보 값이 설정되지 않은 경우에도 이 유형의 마이그레이션이 가능합니다. 하지만 마이그레이션할 도메인에서 Oracle Solaris 10 OS가 실행되는 경우 프로세서 주파수 및 `STICK` 주파수 값이 반드시 일치해야 합니다.

`STICK` 주파수를 확인하려면 다음과 같이 `prtconf -pv` 명령을 사용하십시오.

```
# prtconf -pv | grep stick-frequency
stick-frequency: 05f4bc08
```

주-`STICK` 레지스터가 증분되는 주파수는 최대 속도의 CPU 주파수에서 파생됩니다. 하지만 두 시스템에서 CPU 주파수가 동일한 경우에도 정확한 `STICK` 레지스터 주파수는 약간 다를 수 있으므로 마이그레이션이 차단될 수 있습니다.

메모리에 대한 마이그레이션 요구 사항

도메인 마이그레이션이 수용되도록 대상 시스템의 사용 가능한 메모리가 충분해야 합니다. 또한 다음과 같이 마이그레이션 간에 유지되어야 할 등록 정보가 있습니다.

- 같은 크기의 메모리 블록을 동일한 개수로 만들 수 있어야 합니다.
- 메모리 블록의 물리적 주소가 일치할 필요는 없지만 실제 주소는 마이그레이션 간에 동일하게 유지되어야 합니다.

또한 대상 시스템에서 사용 가능한 메모리의 레이아웃이 마이그레이션할 도메인의 메모리 레이아웃에 맞아야 합니다. 그렇지 않으면 마이그레이션이 실패합니다. 특히 대상 시스템의 메모리는 여러 개의 작은 주소 범위로 단편화되었지만 마이그레이션할 도메인에는 하나의 큰 주소 범위가 필요한 경우 마이그레이션이 실패합니다. 다음 예에서 이 시나리오를 설명합니다. 대상 시스템의 2개 메모리 블록에서 사용 가능한 메모리가 2GB입니다.

```
# ldm list-devices memory
MEMORY
  PA          SIZE
  0x108000000 1G
  0x188000000 1G
```

마이그레이션할 도메인 `ldg-src`의 사용 가능한 메모리도 2GB이지만 마이그레이션할 도메인은 단일 메모리 블록에서 레이아웃되었습니다.

```
# ldm list -o memory ldg-src
NAME
ldg-src
```

```
MEMORY
  RA          PA          SIZE
  0x80000000  0x208000000      2G
```

이와 같은 메모리 레이아웃에서는 마이그레이션이 실패합니다.

```
# ldm migrate-domain ldg-src t5440-sys-2
Target Password:
Unable to bind 2G memory region at real address 0x80000000
Domain Migration of LDom ldg-src failed
```

주 - 마이그레이션 후에는 마이그레이션 대상 도메인이 재부트될 때까지 마이그레이션 대상 도메인에 대해 메모리 DR(동적 재구성)이 사용 안함으로 설정됩니다. 재부트가 완료되면 마이그레이션 대상 도메인에 대해 메모리 DR이 다시 사용으로 설정됩니다.

물리적 I/O 장치에 대한 마이그레이션 요구 사항

물리적 장치에 대한 직접 액세스 권한을 가진 도메인은 마이그레이션할 수 없습니다. 예를 들어, I/O 도메인은 마이그레이션할 수 없습니다. 하지만 물리적 장치와 연관된 가상 장치는 마이그레이션할 수 있습니다.

가상 I/O 장치에 대한 마이그레이션 요구 사항

마이그레이션할 도메인에서 사용되는 모든 가상 I/O 서비스를 대상 시스템에서 사용할 수 있어야 합니다. 즉, 다음 조건이 충족되어야 합니다.

- 마이그레이션할 도메인에서 사용되는 각 가상 디스크 백엔드가 대상 시스템에 정의되어 있어야 합니다. 정의한 가상 디스크 백엔드의 볼륨 및 서비스 이름이 소스 시스템의 볼륨 및 서비스 이름과 동일해야 합니다. 백엔드 경로는 소스 시스템과 대상 시스템에서 다를 수 있지만 반드시 동일한 백엔드를 가리켜야 합니다.



주의 - 소스 시스템과 대상 시스템의 가상 디스크 백엔드 경로가 동일한 저장소를 가리키지 않는 경우에도 마이그레이션이 성공합니다. 하지만 대상 시스템의 도메인 동작을 예측할 수 없으며 도메인을 사용하지 못할 수도 있습니다. 이 문제를 해결하려면 도메인을 중지하고 구성 문제를 수정한 다음 도메인을 다시 시작하십시오. 이러한 단계를 수행하지 않을 경우 도메인이 불일치 상태로 유지될 수 있습니다.

- 마이그레이션할 도메인의 각 가상 네트워크 장치와 대상 시스템의 가상 네트워크 스위치가 상응해야 합니다. 각 가상 네트워크 스위치의 이름은 소스 시스템에서 장치가 연결된 가상 네트워크 스위치의 이름과 동일해야 합니다.

예를 들어, 마이그레이션할 도메인의 `vnet0`이 `switch-y`라는 가상 스위치 서비스에 연결된 경우 대상 시스템의 도메인이 `switch-y`라는 가상 스위치 서비스를 제공해야 합니다.

주-마이그레이션 대상 도메인이 필요한 네트워크 리소스에 액세스할 수 있도록 대상 시스템의 물리적 네트워크가 올바르게 구성되어 있어야 합니다. 그렇지 않으면 마이그레이션이 완료된 후 도메인에서 일부 네트워크 서비스를 사용하지 못할 수 있습니다.

예를 들어, 도메인이 올바른 네트워크 서브넷에 액세스할 수 있는지 확인해야 할 수 있습니다. 또한 도메인이 대상 시스템에서 필요한 원격 시스템에 연결할 수 있도록 게이트웨이, 라우터 또는 방화벽이 제대로 구성되어 있는지 확인해야 할 수도 있습니다.

마이그레이션할 도메인에서 사용되며 자동으로 할당된 범위에 속한 MAC 주소를 대상 시스템에서 사용할 수 있어야 합니다.

- 가상 콘솔 집중기(vcc) 서비스가 대상 시스템에 존재해야 하며 하나 이상의 사용 가능한 포트를 가져야 합니다. 명시적 콘솔 제약 조건은 마이그레이션 중 무시됩니다. 마이그레이션 대상 도메인 이름을 콘솔 그룹으로 사용하고 컨트롤 도메인의 첫 번째 vcc 장치에서 사용 가능한 포트를 사용하여 마이그레이션 대상 도메인의 콘솔이 만들어집니다. 기본 그룹 이름이 충돌할 경우 마이그레이션이 실패합니다.

NIU 하이브리드 I/O에 대한 마이그레이션 요구 사항

NIU 하이브리드 I/O 리소스를 사용하는 도메인을 마이그레이션할 수 있습니다. NIU 하이브리드 I/O 리소스를 지정하는 제약 조건은 도메인에 대한 엄격한 요구 사항이 아닙니다. 사용 가능한 NIU 리소스가 없는 시스템으로 해당 도메인이 마이그레이션되는 경우 제약 조건은 보존되지만 이행되지 않습니다.

암호화 장치에 대한 마이그레이션 요구 사항

암호화 장치가 있는 플랫폼에서 바인딩된 암호화 장치가 있는 게스트 도메인을 마이그레이션할 수 있습니다. 단, 게스트 도메인이 암호화 장치 DR(동적 재구성)을 지원하는 운영 체제에서 실행되어야 합니다.

다음과 같은 Oracle Solaris OS 버전에서 암호화 장치 DR을 지원합니다.

- Solaris 10 10/09 OS 이상
- 패치 ID 142245-01이 설치된 Solaris 10 5/08 OS 이상

마이그레이션 시작 시 Logical Domains Manager는 마이그레이션할 도메인이 암호화 장치 DR을 지원하는지 여부를 확인합니다. 지원되는 경우 Logical Domains Manager는

도메인에서 암호화 장치를 제거하려고 합니다. 마이그레이션이 완료되면 마이그레이션 대상 도메인에 암호화 장치가 다시 추가됩니다.

주- 대상 시스템에서 암호화 장치에 대한 제약 조건이 충족되지 않는 경우에도 마이그레이션 작업이 차단되지 않습니다. 이 경우 마이그레이션 대상 도메인의 암호화 장치 수가 마이그레이션 작업 전의 수보다 줄어든 수 있습니다.

활성 도메인의 재구성 지연

소스 또는 대상 시스템에서의 활성 재구성 지연 작업은 마이그레이션이 시작되지 않도록 합니다. 마이그레이션이 진행되는 동안에는 재구성 지연 작업이 차단됩니다.

활성 도메인에 전원 관리 탄력적 정책이 적용되는 동안 마이그레이션

PM(전원 관리) 탄력적 정책이 적용되는 소스 또는 대상 시스템에 대해서는 도메인 마이그레이션이 지원되지 않습니다. 마이그레이션이 진행되는 동안 소스 또는 대상 시스템의 PM 정책이 성능 정책에서 탄력적 정책으로 전환되면 마이그레이션이 완료될 때까지 정책 전환이 지연됩니다. 소스 또는 대상 시스템에 탄력적 정책이 적용되는 동안 도메인 마이그레이션이 시도되면 마이그레이션 명령이 오류를 반환합니다.

다른 도메인에 대한 작업

시스템에서 마이그레이션이 진행되는 동안 마이그레이션 중인 도메인의 상태 또는 구성을 수정할 수 있는 작업은 차단됩니다. 도메인 자체에 대한 모든 작업과 시스템의 다른 도메인에 대한 작업(예: 바인딩 및 중지)도 차단됩니다.

OpenBoot PROM의 도메인 또는 커널 디버거에서 실행 중인 도메인 마이그레이션

도메인을 마이그레이션하려면 Logical Domains Manager와 마이그레이션할 도메인에서 실행 중인 OS 간에 조정을 수행해야 합니다. 마이그레이션할 도메인이 OpenBoot 또는 커널 디버거(kmdb)에서 실행 중인 경우 이 조정이 불가능합니다. 따라서 마이그레이션할 도메인의 CPU가 하나가 아닌 경우 마이그레이션 시도가 실패합니다. 마이그레이션할 도메인의 CPU가 하나인 경우 특정 요구 사항 및 제한 사항이 충족될 때 마이그레이션이 진행됩니다. [Oracle VM Server for SPARC 2.2 릴리스 노트](#)의 “도메인 마이그레이션 제한 사항”를 참조하십시오.

바인딩된 도메인 또는 비활성 도메인 마이그레이션

바인딩된 도메인 또는 비활성 도메인은 마이그레이션 시 실행 중이 아닌 상태이므로 해당 도메인에는 몇 가지 도메인 마이그레이션 제한 사항만 적용됩니다.

바인딩된 도메인을 마이그레이션하려면 대상 시스템이 마이그레이션할 도메인의 CPU, 메모리 및 I/O 제약조건을 충족할 수 있어야 합니다. 이러한 제약조건을 충족할 수 없을 경우 마이그레이션이 실패합니다.



주의 - 바인딩된 도메인을 마이그레이션할 때 대상 시스템과 런타임 상태 정보가 교환되지 않으므로 가상 디스크 백엔드 `options` 및 `mpgroup` 값이 확인되지 않습니다. 활성 도메인을 마이그레이션할 때는 이러한 값이 **확인됩니다**.

비활성 도메인 마이그레이션에는 해당 요구 사항이 적용되지 않습니다. 하지만 나중에 바인딩이 시도되거나 도메인 바인딩이 실패할 경우 대상 시스템이 마이그레이션 대상 도메인의 제약조건을 충족해야 합니다.

가상 I/O 장치에 대한 마이그레이션 요구 사항

비활성 도메인의 경우 VIO(가상 I/O) 제약조건 검사가 수행되지 않습니다. 따라서 VIO 서버가 존재하지 않아도 마이그레이션이 성공합니다. 도메인이 바인딩될 때는 비활성 도메인과 마찬가지로 VIO 서버가 존재해야 하며 사용 가능한 상태여야 합니다.

PCIe 끝점 장치에 대한 마이그레이션 요구 사항

PCIe 끝점 장치로 구성된 I/O 도메인에서는 도메인 마이그레이션을 수행할 수 없습니다.

DIO(직접 I/O) 기능에 대한 자세한 내용은 [79 페이지 “PCIe 끝점 장치 지정”](#)을 참조하십시오.

진행 중인 마이그레이션 모니터

마이그레이션이 진행 중인 경우 상태 출력에 마이그레이션 중인 도메인과 마이그레이션 대상 도메인이 다르게 표시됩니다. `ldm list` 명령의 출력은 도메인 마이그레이션 상태를 나타냅니다.

FLAGS 필드의 여섯번째 열에는 다음 값 중 하나가 표시됩니다.

- 마이그레이션 중인 도메인에는 이 도메인이 마이그레이션 소스임을 나타내는 s가 표시됩니다.
- 마이그레이션 대상 도메인에는 이 도메인이 마이그레이션 대상임을 나타내는 t가 표시됩니다.
- 사용자 개입이 필요한 오류가 발생하면 e가 표시됩니다.

다음 명령은 ldg-src 도메인이 마이그레이션 소스임을 보여줍니다.

```
# ldm list ldg-src
NAME      STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg-src    suspended  -n---s          1      1G         0.0%    2h 7m
```

다음 명령은 ldg-tgt 도메인이 마이그레이션 대상임을 보여줍니다.

```
# ldm list ldg-tgt
NAME      STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg-tgt    bound      -----t  5000     1      1G
```

긴 형식의 상태 출력은 마이그레이션에 대한 추가 정보를 보여줍니다. 소스 시스템에서 상태 출력은 작업 완료율과 대상 시스템 및 마이그레이션 대상 도메인의 이름을 보여줍니다. 마찬가지로 대상 시스템에서 상태 출력은 작업 완료율과 소스 시스템 및 마이그레이션 중인 도메인의 이름을 보여줍니다.

다음 명령은 ldg-src 도메인에 대한 마이그레이션 작업 진행률을 보여줍니다.

```
# ldm list -o status ldg-src
NAME
ldg-src

STATUS
  OPERATION    PROGRESS    TARGET
  migration    17%         t5440-sys-2
```

진행 중인 마이그레이션 취소

마이그레이션이 시작된 후 KILL 신호로 ldm 명령이 중단되면 마이그레이션 작업이 종료됩니다. 마이그레이션 작업이 종료되면 마이그레이션 대상 도메인이 삭제되고 마이그레이션할 도메인(활성 상태인 경우)이 재개됩니다. ldm 명령의 제어 셸이 손실되면 마이그레이션이 계속 백그라운드로 실행됩니다.

ldm cancel-operation 명령을 사용하여 외부적으로 마이그레이션 작업을 취소할 수도 있습니다. 이 명령이 진행 중인 마이그레이션을 종료하면 마이그레이션 중인 도메인이 활성 도메인으로 재개됩니다. ldm cancel-operation 명령은 소스 시스템에서 시작되어야 합니다. 지정된 시스템에서 마이그레이션 관련 명령은 해당 시스템에서 시작된 마이그레이션 작업에 영향을 끼칩니다. 대상 시스템은 마이그레이션 작업을 제어할 수 없습니다.

주- 마이그레이션이 시작된 후에는 ldm 프로세스를 일시 중단해도 작업이 일시 중지되지 않습니다. ldm 프로세스가 아니라 소스 시스템과 대상 시스템의 Logical Domains Manager 데몬(ldmd)이 마이그레이션에 영향을 끼치므로 작업이 일시 중지되지 않는 것입니다. ldm 프로세스는 ldmd로부터 마이그레이션이 완료되었음을 알리는 신호를 기다린 후 다시 시작됩니다.

실패한 마이그레이션 복구

다음과 같은 방식으로 네트워크 연결이 끊긴 경우 마이그레이션 작업이 종료됩니다.

- 마이그레이션 중인 도메인이 마이그레이션 대상 도메인으로 런타임 상태 정보를 모두 전송한 후
- 마이그레이션 대상 도메인이 도메인 재개를 인식하기 전

다음 단계를 수행하여 마이그레이션이 성공적으로 완료되었는지 여부를 확인해야 합니다.

1. 마이그레이션 대상 도메인이 성공적으로 작업을 재개했는지 여부를 확인합니다. 마이그레이션 대상 도메인은 다음과 같은 두 가지 상태 중 하나입니다.
 - 마이그레이션이 성공적으로 완료된 경우 마이그레이션 대상 도메인은 정상 상태입니다.
 - 마이그레이션이 실패한 경우 대상 시스템이 마이그레이션 대상 도메인을 정리하고 삭제합니다.
2. 마이그레이션 대상 도메인이 성공적으로 작업을 재개한 경우 오류 상태의 소스 시스템에서 도메인을 삭제해도 됩니다. 하지만 마이그레이션 대상 도메인이 없을 경우 소스 시스템의 도메인이 여전히 도메인의 마스터 버전이므로 해당 도메인을 복구해야 합니다. 이 도메인을 복구하려면 소스 시스템에서 `ldm cancel-operation` 명령을 실행하십시오. 이 명령은 오류 상태를 처리하고 도메인을 원래 상태로 복원합니다.

마이그레이션 예

예 9-1 게스트 도메인 마이그레이션

이 예에서는 ldg1 도메인을 t5440-sys-2라는 시스템으로 마이그레이션하는 방법을 보여줍니다.

```
# ldm migrate-domain ldg1 t5440-sys-2
Target Password:
```

대상 시스템 암호를 묻지 않고 이 마이그레이션을 수행하려면 다음 명령을 사용하십시오.

예 9-1 게스트 도메인 마이그레이션 (계속)

```
# ldm migrate-domain -p pfile ldg1 t5440-sys-2
```

-p 옵션은 파일 이름을 인수로 사용합니다. 지정된 파일에는 대상 시스템에 대한 수퍼유저 암호가 들어 있습니다. 이 예에서는 pfile에 대상 시스템 t5440-sys-2에 대한 암호가 들어 있습니다.

예 9-2 게스트도메인 마이그레이션 및 이름 바꾸기

이 예에서는 마이그레이션 작업의 일부로 도메인의 이름을 바꾸는 방법을 보여줍니다. 마이그레이션의 일부로 소스 시스템의 ldg-src 도메인 이름이 대상 시스템(t5440-sys-2)의 ldg-tgt로 바뀝니다. 또한 ldm-admin 사용자가 대상 시스템에서의 인증에 사용됩니다.

```
# ldm migrate ldg-src ldm-admin@t5440-sys-2:ldg-tgt
Target Password:
```

예 9-3 마이그레이션 실패 메시지

이 예에서는 대상 시스템이 최신 마이그레이션 기능을 지원하지 않을 경우 표시되는 오류 메시지를 보여줍니다.

```
# ldm migrate ldg1 dt212-346
Target Password:
The target machine is running an older version of the domain
manager that does not support the latest migration functionality.
```

Upgrading to the latest software will remove restrictions on a migrated domain that are in effect until it is rebooted. Consult the product documentation for a full description of these restrictions.

The target machine is running an older version of the domain manager that is not compatible with the version running on the source machine.

Domain Migration of LDom ldg1 failed

예 9-4 대상 시스템의 도메인에 대한 마이그레이션 상태 확인

이 예에서는 마이그레이션이 진행되는 동안 마이그레이션 대상 도메인의 상태를 확인하는 방법을 보여줍니다. 이 예에서 소스 시스템은 t5440-sys-1입니다.

```
# ldm list -o status ldg-tgt
NAME
ldg-tgt

STATUS
      OPERATION    PROGRESS    SOURCE
migration    55%        t5440-sys-1
```

예 9-5 소스 시스템의 도메인에 대해 구문 분석이 가능한 마이그레이션 상태 확인

이 예에서는 마이그레이션이 진행되는 동안 마이그레이션 중인 도메인에 대해 구문 분석이 가능한 상태를 확인하는 방법을 보여줍니다. 이 예에서 대상 시스템은 **t5440-sys-2**입니다.

```
# ldm list -o status -p ldg-src
VERSION 1.6
DOMAIN|name=ldg-src|
STATUS
|op=migration|progress=42|error=no|target=t5440-sys-2
```

리소스 관리

이 장에는 Oracle VM Server for SPARC 시스템에서 리소스 관리 수행에 대한 정보가 포함되어 있습니다.

이 장에서는 다음 주제를 다룹니다.

- 191 페이지 “리소스 재구성”
- 193 페이지 “리소스 할당”
- 193 페이지 “CPU 할당”
- 197 페이지 “SPARC CPU를 조정하여 SPARC T4 시스템에서 작업 로드 성능 최적화”
- 200 페이지 “하드 분할을 사용하여 시스템 구성”
- 206 페이지 “도메인에 물리적 리소스 지정”
- 209 페이지 “메모리 동적 재구성 사용”
- 216 페이지 “전원 관리 사용”
- 221 페이지 “동적 리소스 관리 사용”
- 224 페이지 “도메인 리소스 나열”

리소스 재구성

Oracle VM Server for SPARC 소프트웨어를 실행하는 시스템은 가상 CPU, 가상 I/O 장치, 암호화 장치, 메모리 등의 리소스를 구성할 수 있습니다. 일부 리소스는 실행 중인 도메인에 동적으로 구성할 수 있는 반면, 나머지는 중지된 도메인에 구성해야 합니다. 컨트롤 도메인에서 리소스를 동적으로 구성할 수 없는 경우 먼저 지연된 재구성을 시작해야 합니다. 컨트롤 도메인이 재부트될 때까지 지연된 재구성이 구성 작업을 연기합니다.

동적 재구성

동적 재구성(DR)은 운영 체제(OS) 실행 중에 리소스를 추가하거나 제거할 수 있도록 합니다. 특정 리소스 유형의 DR 수행 기능은 논리적 도메인에서 실행 중인 OS의 지원 여부에 따라 결정됩니다.

동적 재구성은 다음 리소스에 대해 지원됩니다.

- 가상 CPU – 모든 버전의 Oracle Solaris 10 OS에서 지원
- 가상 I/O 장치 – Oracle Solaris 10 10/08 OS 이상에서 지원
- 암호화 장치 – Oracle Solaris 10 8/11 OS 이상에서 지원
- 메모리 – Oracle VM Server for SPARC 2.0 릴리스부터 지원([209 페이지 “메모리 동적 재구성 사용”](#) 참조)
- 물리적 I/O devices – 지원되지 않음

DR 기능을 사용하려면 Logical Domains DR 도메인 `drd`가 변경할 도메인에서 실행되고 있어야 합니다. `drd(1M)` 매뉴얼 페이지를 참조하십시오.

지연된 재구성

즉시 발생하는 DR 작업과 달리 지연된 재구성은 다음 상황에서 적용됩니다.

- OS의 다음 재부트 이후
- 논리적 도메인 중지 및 시작 이후

지연된 재구성 작업은 제어 도메인으로 제한됩니다. 기타 모든 도메인의 경우 리소스를 동적으로 재구성할 수 없다면 도메인을 중지하여 구성을 수정해야 합니다.

지연된 재구성이 제어 도메인에서 진행 중일 때 제어 도메인에 대한 다른 재구성 요청은 재부트되거나 중지 및 시작될 때까지 연기됩니다.

`ldm cancel-reconf` 명령은 제어 도메인에서 지연된 재구성 작업을 취소합니다. 지연된 재구성 기능 사용 방법에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

주 – 다른 `ldm remove-*` 명령이 이미 가상 I/O 장치에 대해 지연된 재구성 작업을 실행한 경우 `ldm cancel-reconf` 명령을 사용할 수 없습니다. `ldm cancel-reconf` 명령은 이러한 상황에서 실패합니다.

지연된 재구성을 사용하여 제어 도메인의 리소스를 줄일 수 있습니다. 제어 도메인에서 많은 수의 CPU를 제거하려면 [Oracle VM Server for SPARC 2.2 릴리스 노트](#)의 “컨트롤 도메인에서 많은 수의 CPU 제거”를 참조하십시오. 제어 도메인에서 많은 양의 메모리를 제거하려면 [211 페이지 “제어 도메인의 메모리 줄이기”](#)를 참조하십시오.

리소스 할당

Oracle VM Server for SPARC 2.0 릴리스부터 리소스 할당 방식에서는 리소스 할당 제약 조건을 사용하여 바인드 시 도메인에 리소스를 지정합니다.

리소스 할당 제약 조건은 도메인에 리소스를 지정할 때 시스템이 **반드시** 충족해야 하는 피할 수 없는 사항입니다. 제약 조건이 충족되지 않을 경우 리소스 할당과 도메인 바인드가 모두 실패합니다.

CPU 할당

CPU 할당 방식에서는 CPU 리소스에 대해 다음 제약 조건을 사용합니다.

- **전체 코어 제약 조건.** 이 제약 조건은 CPU 코어가 가상 CPU가 아닌 도메인에 할당되도록 지정합니다. 도메인에 사용으로 설정된 최대 코어 제약 조건이 없는 한 전체 코어 제약 조건은 동적이며, 이것은 추가 또는 제거 작업을 `ldm set-core` 또는 `ldm set-vcpu` 명령을 사용하여 각각 수행할 수 있음을 의미합니다. 도메인은 비활성, 바인드된 또는 활성일 수 있습니다. 하지만 제약 조건 적용 요청을 충족하려면 사용 가능한 충분한 코어가 있어야 합니다. 가장 나쁜 사례로는 다른 도메인과 코어를 공유하는 도메인에서 전체 코어 제약 조건을 요청할 경우 요청을 충족하려면 해제 목록의 코어를 사용할 수 있어야 하는 경우입니다. 가장 좋은 사례로는 코어의 모든 가상 CPU가 이미 코어 경계에 있어서 CPU 리소스 변경 없이 제약 조건이 적용되는 경우입니다.
- **최대 코어 수(max-cores) 제약 조건.** 이 제약 조건은 바인드된 도메인 또는 활성 도메인에 지정할 수 있는 최대 코어 수를 지정합니다.

주-도메인이 중지 및 바인드 해제되거나 제어 도메인이 지연된 재구성 상태가 아니라면 `max-cores` 등록 정보는 변경할 수 **없습니다**. 따라서 전체 코어 제약 조건이 사용으로 설정되었을 때 지정된 값에서 최대 코어 수를 높이려면 먼저 도메인을 중지 및 바인드 해제해야 합니다.

▼ 전체 코어 제약 조건을 적용하는 방법

최대 코어 제약 조건을 설정하기 전에 제어 도메인에서 전체 코어 제약 조건을 사용으로 설정하는 것이 가장 좋습니다.

- 1 **primary** 도메인에서 전체 코어 제약 조건을 적용합니다.
`# ldm set-core 1 primary`
- 2 제어 도메인에서 전체 코어 제약 조건이 사용으로 설정되었는지 확인합니다.
`# ldm ls -o re primary`

max-cores는 unlimited로 설정되어 있습니다. 다른 도메인과 마찬가지로 제어 도메인은 최대 코어 제약 조건이 사용으로 설정될 때까지 하드 분할과 함께 사용할 수 없습니다.

3 primary 도메인에서 지연된 재구성을 시작합니다.

```
# ldm start-reconf primary
```

4 primary 도메인에서 최대 코어 제약 조건을 사용으로 설정합니다.

```
# ldm set-domain max-cores=number-of-CPU-cores primary
```

주 - 이러한 코어와 연결된 암호화 장치는 코어 추가의 영향을 받지 않습니다. 따라서 시스템에서는 연결된 암호화 장치를 도메인에 자동으로 추가하지 않습니다. 하지만 코어의 마지막 가상 CPU가 제거될 **경우에만** 암호화 장치가 자동으로 제거됩니다. 이 작업은 암호화 장치가 “고립화”되지 않도록 합니다.

최대 코어 제약 조건은 바인드되거나 활성인 도메인이 아닌 비활성 도메인에서만 사용 안함으로 설정할 수 있습니다. 제어 도메인에서 최대 코어 제약 조건을 사용 안함으로 설정하기 전에 먼저 지연된 재구성을 시작해야 합니다.

5 전체 코어 제약 조건이 사용으로 설정되었는지 확인합니다.

```
# ldm ls -o re primary
```

6 primary 도메인을 재부트합니다.

```
# reboot
```

재부트 시 하드 분할과 함께 제어 도메인을 사용할 수 있습니다.

예 10-1 전체 코어 제약 조건 적용

이 예에서는 primary 도메인에서 전체 코어 제약 조건을 적용하는 방법을 보여줍니다. 첫번째 명령은 제약 조건을 적용하고, 두번째 명령은 사용으로 설정되었는지 여부를 확인합니다.

```
# ldm set-core 1 primary
# ldm ls -o re primary
NAME
primary

CONSTRAINT
  cpu=whole-core
  max-cores=unlimited
  threading=max-throughput
```

다음 명령은 지연된 재구성을 시작하고 max-cores 등록 정보를 설정한 다음 제약 조건이 사용으로 설정되었는지 확인하여 최대 코어를 3코어로 제한합니다.

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
```

domain reboots, at which time the new configuration for the primary domain will also take effect.

```
# ldm set-domain max-cores=3 primary
```

```
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
# ldm ls -o re primary
```

```
NAME
```

```
primary
```

```
FLAGS
```

```
normal, delayed(modify), control, vio-service
```

```
CONSTRAINT
```

```
cpu=whole-core
```

```
max-cores=3
```

```
threading=max-throughput
```

재부트 시 하드 분할과 함께 제어 도메인을 사용할 수 있습니다.

다음 명령은 최대 코어 제약 조건을 제거하지만, `ldg1` 도메인에서 전체 코어 제약 조건은 그대로 둡니다.

```
# ldm set-domain max-cores=unlimited ldg1
```

`ldg1` 도메인에서 최대 코어 제약 조건과 전체 코어 제약 조건을 모두 제거하려면 코어 대신 가상 CPU를 다음과 같이 지정합니다.

```
# ldm set-vcpu 8 ldg1
```

전체 코어 제약 조건과 다른 도메인 기능 사이의 상호 작용

이 절에서는 전체 코어 제약 조건과 다음 기능 사이의 상호 작용을 설명합니다.

- 195 페이지 “CPU 동적 재구성”
- 196 페이지 “동적 리소스 관리”
- 196 페이지 “도메인 마이그레이션”
- 196 페이지 “전원 관리”

CPU 동적 재구성

전체 코어 제약 조건은 CPU 동적 재구성(DR)과 완벽하게 호환됩니다. 도메인이 전체 코어 제약 조건으로 정의된 경우 `ldm add-core`, `ldm set-core` 또는 `ldm remove-core` 명령을 사용하여 활성 도메인에서 코어 수를 변경할 수 있습니다.

하지만 바인드된 도메인이나 활성 도메인이 지연된 재구성 모드가 아닌 경우 코어 수는 최대 코어 수를 초과할 수 없습니다. 이 최대값은 최대 코어 제약 조건으로 설정되며, 전체 코어 제약 조건이 사용으로 설정되면 자동으로 사용으로 설정됩니다. 최대 코어 제약 조건을 충족하지 못하는 모든 CPU DR 작업은 실패합니다.

동적 리소스 관리

전체 코어 제약 조건은 동적 리소스 관리(DRM)와 호환되지 않습니다. 전체 코어 제약 조건을 사용하는 도메인에서 DRM 정책이 사용으로 설정된 경우 해당 정책은 자동으로 사용 안함으로 설정됩니다. 전체 코어 제약 조건은 계속 사용으로 설정됩니다.

전체 코어 제약 조건이 적용 중일 때 DRM 정책을 사용으로 설정할 수 없더라도 도메인에 대한 DRM 정책은 정의할 수 있습니다. 정책이 자동으로 사용 안함으로 설정되더라도 여전히 활성 상태로 유지됩니다. 도메인이 전체 코어 제약 조건 없이 다시 시작될 경우 정책은 자동으로 다시 사용으로 설정됩니다.

다음은 전체 코어 제약 조건과 DRM 사이에 예상되는 상호 작용입니다.

- 전체 코어 제약 조건이 도메인에서 설정된 경우 해당 도메인에서 DRM 정책을 사용으로 설정하려고 시도하면 경고 메시지가 표시됩니다.
- DRM 정책이 비활성 도메인에서 적용 중인 경우 해당 도메인에서 전체 코어 제약 조건을 사용으로 설정할 수 있습니다. 도메인이 활성 상태가 되고 정책이 사용으로 설정된 경우 시스템은 도메인에 대한 DRM 정책을 자동으로 사용 안함으로 설정합니다.
- DRM 정책이 활성 도메인 또는 바인드된 도메인에서 사용으로 설정된 경우 전체 코어 제약 조건을 사용으로 설정할 수 없습니다.

도메인 마이그레이션

CPU 전체 코어 구성은 도메인 마이그레이션과 호환되지 않습니다. 하지만 CPU 전체 코어로 구성된 도메인을 마이그레이션할 수는 있습니다. 이러한 마이그레이션 이후 대상 시스템에서 하드 분할은 적용되지 않습니다. 또한 전체 코어 구성 및 최대 CPU 코어 수는 대상 시스템에서 유지되지 않습니다.

전체 코어로 구성된 도메인을 마이그레이션하는 경우 마이그레이션이 완료된 후 하드 분할을 사용하려면 대상 도메인을 재구성해야 합니다. 또한 소스 및 대상 시스템 모두에서 도메인을 사용할 수 있는 사용권 계약이 있는지 확인해야 합니다.

전원 관리

전체 코어 제약 조건은 전원 관리(PM) 성능 및 탄력적 정책과 완벽하게 호환됩니다. 탄력적 정책이 적용 중인 경우 PM 부속 시스템은 전체 코어 제약 조건으로 구성된 도메인에서 CPU 코어를 추가하거나 제거할 수 있습니다. 이러한 경우 전체 코어 제약 조건은 계속 유지되며, 해당 제약 조건을 사용하는 도메인은 전체 코어로만 구성됩니다.

SPARC CPU를 조정하여 SPARCT4 시스템에서 작업 로드 성능 최적화

동적 CPU 스레드 제어를 사용하여 SPARC T4 시스템에서 작업 로드 성능을 최적화할 수 있습니다.

이러한 스레드 제어를 통해 코어당 활성화할 하드웨어 스레드 수를 지정할 수 있습니다. 기존 응용 프로그램은 재작성이나 재컴파일 없이 SPARC CPU에 대한 동적 스레드 성능 이점을 활용할 수 있습니다.

이 절에서는 CPU 스레드 제어를 사용하여 SPARC T4 시스템에서 CPU 성능을 최적화하는 방법을 설명합니다. CPU 성능은 최대 CPU 스레드 수를 사용하도록 CPU 코어를 조정하여 최대 처리량으로 최적화할 수 있습니다. 기본적으로 CPU는 최대 처리량으로 조정되어 있습니다. 또는 CPU 성능은 IPC(사이클당 명령 개수)를 최대화하도록 CPU 코어를 조정하여 CPU 중심 작업 로드로 최적화할 수 있습니다.

CPU 스레드 모드 및 작업 로드

SPARC T4 시스템에서는 CPU 스레드 모드를 지정하여 CPU 성능을 최적화할 수 있습니다. 스레드 모드는 시스템의 각 도메인에 대해 동적으로 및 독립적으로 설정할 수 있습니다. 스레드 모드 변경을 위해 재부트가 필요하지 않으며, 설정된 모드는 도메인 재부트 및 플랫폼 전원을 켜다가 켜 이후에도 계속 유지됩니다.

알맞은 CPU 스레드 모드를 선택하면 도메인에서 실행 중인 응용 프로그램 및 작업 로드의 성능을 향상시킬 수 있습니다. 다음과 같이 처리량을 최대화하거나 사이클당 명령 개수를 최대화하는 스레드 모드를 선택할 수 있습니다.

- **처리량 최대화(max-throughput).** 높은 처리량에서 가장 많은 이점을 누리는 작업 로드가 많은 소프트웨어를 실행하고 많은 I/O 작업을 수행합니다. 최대 처리량으로 최적화할 경우 CPU 코어에서 최대 하드웨어 스레드 수를 동시에 실행하게 됩니다. 이 모드는 혼합 응용 프로그램 작업 로드 및 다중 스레드 작업(예: 웹 서버, 데이터베이스 서버 및 파일 서버에서 수행되는 작업)이 많은 작업 로드에서 가장 좋습니다. 이 모드는 기본적으로 사용되며 이전 SPARC T 시리즈 플랫폼(예: SPARC T3 플랫폼)에서도 사용됩니다.
- **IPC 최대화(max-ipc).** 높은 IPC에서 가장 많은 이점을 누리는 작업 로드는 대개 CPU 중심의 단일 스레드 응용 프로그램(예: 집중적으로 산술 연산을 실행하는 시스템)입니다. 최대 IPC로 최적화할 경우 CPU 스레드에서 CPU 사이클당 더 많은 명령을 실행하게 됩니다. 이 최적화는 동일 CPU 코어에서 동시에 활성화되는 CPU 수를 줄임으로써 달성됩니다.

CPU 스레드 모드 선택

ldm add-domain 또는 ldm set-domain 명령을 사용하여 threading 등록 정보를 설정함으로써 도메인에 대한 CPU 스레드 모드를 선택합니다.

```
ldm add-domain [threading=max-throughput|max-ipc] ldom
```

```
ldm set-domain [threading=max-throughput|max-ipc] ldom
```

threading 등록 정보는 다음 값 중 하나를 지정하여 스레드 모드를 동적으로 변경하는 데 사용됩니다.

- **max-throughput.** 처리량을 최대화하는 스레드 모드를 선택하려면 이 값을 사용합니다. 이 모드는 도메인에 지정된 모든 스레드를 활성화합니다. 이 모드는 기본적으로 사용되며 모드를 지정하지 않을 경우(threading=)에도 선택됩니다.
- **max-ipc.** IPC(사이클당 명령 개수)를 최대화하는 스레드 모드를 선택하려면 이 값을 사용합니다. SPARC T4 플랫폼에서 이 모드를 사용할 경우 도메인에 지정된 각 CPU 코어에 대해 하나의 스레드만 활성화됩니다. 이 모드를 선택하려면 도메인이 전체 코어 제약 조건으로 구성되어 있어야 합니다.

전체 코어 제약 조건을 구성하려면 `ldm add-core` 또는 `ldm set-core` 명령을 사용합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

스레드 모드를 변경하면 CPU 스레드가 동적으로 활성화 또는 비활성화됩니다. 따라서 도메인에서 사용 가능한 가상 CPU 수도 동적으로 변경됩니다.

max-ipc 스레드 모드는 전체 코어 제약 조건을 활용하므로 전체 코어 제약 조건 요구 사항 및 제한 사항에 따라 다음을 수행해야 합니다.

- 도메인에 할당된 코어 수를 변경합니다.
- 전체 코어 제약 조건을 사용 또는 사용 안함으로 설정합니다.

따라서 실행 중인 도메인의 스레드 모드를 **max-ipc** 모드로 동적으로 변경하려면 전체 코어 제약 조건으로 도메인을 구성해야 합니다.

제한 사항에 대한 자세한 내용은 [199 페이지 “스레드 제어 제한 사항”](#)을 참조하십시오. `add-domain` 및 `set-domain` 하위 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

threading 등록 정보 값 보기

다음 명령을 사용하여 threading 등록 정보 값을 볼 수 있습니다.

- `ldm list -o resmgmt` 명령은 제약 조건을 표시합니다. 다음 출력 예는 threading 등록 정보가 **max-ipc**로 설정되어 있음을 보여줍니다.

```
# ldm list -o resmgmt ldg1
NAME
ldg1
CONSTRAINT
whole-core
max-cores=3
threading=max-ipc
```

- `ldm list -o cpu` 명령은 UTIL 열에서 값 0을 지정하여 비활성화된 가상 CPU를 표시합니다. 다음 **max-ipc** 예에서 굵은 텍스트는 CPU당 하나의 스레드만 활성화되었음을 보여줍니다.

```
# ldm list -o cpu ldg1
NAME
ldg1
VCPU
VID PID CID UTIL STRAND
0 8 1 0.3% 100%
1 9 1 0 100%
2 10 1 0 100%
3 11 1 0 100%
4 12 1 0 100%
5 13 1 0 100%
6 14 1 0 100%
7 15 1 0 100%
8 24 2 0.4% 100%
...
```

- `ldm list -l` 명령에는 지정된 도메인에 대한 모든 정보가 포함됩니다. 다음 예에서 굵은 텍스트는 `threading` 등록 정보가 `max-ipc`로 설정되어 있음을 보여줍니다.

```
# ldm list -l ldg1
...
VID PID CID UTIL STRAND
0 8 1 0.6% 100%
1 9 1 0 100%
2 10 1 0 100%
3 11 1 0 100%
4 12 1 0 100%
5 13 1 0 100%
6 14 1 0 100%
...
CONSTRAINT
whole-core
max-cores=3
threading=max-ipc
...
```

스레드 제어 제한 사항

스레드 제어 기능에는 다음 제한 사항이 있습니다.

- 전체 코어 제약 조건 제한 사항이 적용됩니다. [193 페이지 “CPU 할당”](#)을 참조하십시오.
- `threading` 등록 정보 값은 도메인 마이그레이션 시 유지되지 않습니다.
- `threading` 등록 정보는 전원 관리(PM)가 사용으로 설정된 경우 `max-ipc`로 설정할 수 없습니다.

PM이 실행될 경우 모든 도메인의 `threading` 등록 정보는 `max-throughput`으로 설정되어야 합니다.

하드 분할을 사용하여 시스템 구성

이 절에서는 Oracle VM Server for SPARC 소프트웨어에서 하드 분할을 설명하고 Oracle CPU 라이선스 요구 사항을 준수하도록 하드 분할을 사용하는 방법에 대해 설명합니다.

- **CPU 코어 및 CPU 스레드.** Oracle VM Server for SPARC 소프트웨어는 SPARC T 시리즈 프로세서를 사용하는 Oracle's SPARC T 시리즈 서버에서 실행됩니다. SPARC T 시리즈 프로세서에는 다중 CPU 코어가 있으며, 각 코어에는 다중 CPU 스레드가 포함되어 있습니다.
- **하드 분할 및 CPU 전체 코어.** Oracle VM Server for SPARC 2.0 릴리스부터 하드 분할이 CPU 전체 코어 구성을 사용하여 강제 적용됩니다. CPU 전체 코어 구성에는 개별 CPU 스레드 대신 CPU 전체 코어가 할당된 도메인이 있습니다. 기본적으로 도메인은 CPU 스레드를 사용하도록 구성됩니다.

전체 코어 구성으로 도메인을 바인드할 경우 시스템은 지정된 수의 CPU 코어 및 모든 해당 CPU 스레드를 도메인에 프로비전합니다. CPU 전체 코어 구성을 사용하면 바인드된 도메인이나 활성 도메인에 동적으로 지정할 수 있는 CPU 코어 수가 제한됩니다.

- **Oracle 하드 분할 라이선스.** Oracle 하드 분할 라이선스를 준수하려면 Oracle VM Server for SPARC 2.0 릴리스 이상을 사용해야 합니다. 또한 CPU 전체 코어를 다음과 같이 사용해야 합니다.
 - 도메인에서 Oracle 하드 분할 라이선스를 사용하는 응용 프로그램을 실행할 경우 해당 도메인은 CPU 전체 코어로 구성되어야 합니다.
 - 도메인에서 Oracle 하드 분할 라이선스를 사용하는 응용 프로그램을 실행하지 않을 경우 해당 도메인은 CPU 전체 코어로 구성하지 않아도 됩니다. 예를 들어, 제어 도메인에서 Oracle 응용 프로그램을 실행하지 않을 경우 해당 도메인은 CPU 전체 코어로 구성하지 않아도 됩니다.

도메인의 구성 확인

이 절의 작업에서는 도메인이 CPU 전체 코어로 구성되었는지 여부를 확인하는 방법과 도메인에 지정된 CPU 코어를 나열하는 방법에 대해 설명합니다.

▼ 도메인이 CPU 전체 코어로 구성되었는지 여부를 확인하는 방법

- 도메인이 CPU 전체 코어로 구성되었는지 여부를 확인합니다.

```
# ldm list -o resgmt domain
```

전체 코어 제약 조건이 출력에 나타나고 max-cores 등록 정보가 도메인에 대해 구성된 최대 CPU 코어 수를 지정하는지 확인합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예 10-2 도메인이 CPU 전체 코어로 구성되었는지 여부 확인

다음 명령은 ldg1 도메인이 CPU 전체 코어 및 최대 5코어로 구성되었음을 보여줍니다.

```
# ldm list -o resmgmt ldg1
NAME
ldg1

CONSTRAINT
  whole-core
  max-cores=5
```

▼ 도메인에 지정된 CPU 코어를 나열하는 방법

도메인이 바인드될 경우 CPU 코어가 도메인에 지정됩니다.

- 도메인에 지정된 CPU 코어를 나열합니다.

```
# ldm list -o core domain
```

예 10-3 도메인에 지정된 CPU 코어 나열

다음 명령은 ldg1 도메인에 지정된 코어를 보여줍니다.

```
# ldm list -o core ldg1
NAME
ldg1

CORE
CID    PCPUSET
1      (8, 9, 10, 11, 12, 13, 14, 15)
2      (16, 17, 18, 19, 20, 21, 22, 23)
```

CPU 전체 코어로 도메인 구성

이 절의 작업에서는 CPU 전체 코어로 새 도메인을 만드는 방법, CPU 전체 코어로 기존 도메인을 구성하는 방법 및 CPU 전체 코어로 primary 도메인을 구성하는 방법에 대해 설명합니다.

주 - 전체 코어를 지정하는 데 사용되는 ldm 하위 명령은 Oracle VM Server for SPARC 2.2 릴리스에서 변경되었습니다.

이 절의 작업 및 예에서는 Oracle VM Server for SPARC 2.2 릴리스에서 도입된 새로운 명령을 사용합니다.

Logical Domains Manager의 버전 2.0 또는 2.1을 사용하여 전체 코어를 도메인에 지정할 경우 ldm add-core, ldm set-core 및 ldm remove-core 명령 대신 각각 ldm add-vcpu -c, ldm set-vcpu -c 및 ldm remove-vcpu -c 명령을 사용하십시오.

도메인에서 CPU 전체 코어를 사용하도록 구성하려면 다음 명령을 사용합니다.

```
ldm set-core number-of-cpu-cores domain
```

이 명령은 도메인에 대한 최대 CPU 코어 수(CPU 최대값)도 지정합니다. `ldm(1M)` 매뉴얼 페이지를 참조하십시오.

Oracle VM Server for SPARC 2.2 릴리스부터 CPU 상한값 및 CPU 코어 할당은 별도의 명령으로 처리됩니다. 이러한 명령을 사용하면 CPU 코어 할당, 최대값 설정 또는 두 작업 모두 독립적으로 가능합니다. 설정된 CPU 상한값이 없더라도 할당 단위는 코어로 설정할 수 있습니다. 하지만 Oracle VM Server for SPARC 시스템에서 하드 분할을 구성하는 경우 이 모드에서 시스템 실행은 허용되지 **않습니다**.

- `add-core`, `set-core` 또는 `rm-core` 하위 명령을 사용하여 도메인에 지정된 수의 CPU 코어를 할당합니다.
- `create-domain` 또는 `set-domain` 하위 명령을 사용하여 `max-cores` 등록 정보 값을 지정함으로써 CPU 상한값을 설정합니다.

Oracle VM Server for SPARC 시스템에서 하드 분할을 구성하려는 경우에는 상한값을 **설정해야** 합니다.

▼ CPU 전체 코어로 새 도메인을 만드는 방법

1 도메인을 만듭니다.

```
# ldm create domain
```

2 도메인에 대한 CPU 전체 코어 수를 설정합니다.

```
# ldm set-core number-of-cpu-cores domain
```

이 명령은 도메인에 대한 최대 CPU 코어 수도 `number-of-cpu-cores`로 설정합니다.

3 도메인을 구성합니다.

이 구성 중에 `ldm add-core`, `ldm set-core` 또는 `ldm rm-core` 명령을 사용합니다.

4 바인딩 후 도메인을 시작합니다.

```
# ldm bind domain
# ldm start domain
```

예 10-4 2개의 CPU 전체 코어로 새 도메인 만들기

이 예에서는 2개의 CPU 전체 코어로 `ldg1` 도메인을 만듭니다. 첫번째 명령은 `ldg1` 도메인을 만듭니다. 두번째 명령은 2개의 CPU 전체 코어로 `ldg1` 도메인을 구성합니다. 두번째 명령은 `ldg1`에 대한 최대 CPU 코어 수도 2로 설정합니다.

이 시점에 202 페이지 “CPU 전체 코어로 새 도메인을 만드는 방법”의 단계 3에 설명된 제한 사항에 따라 도메인에서 추가 구성을 수행할 수 있습니다.

세번째 및 네번째 명령은 `ldg1` 도메인을 사용할 수 있도록 `ldg1` 도메인을 바인드하고 시작하는 방법을 보여줍니다.

```
# ldm create ldg1
# ldm set-core 2 ldg1
...
# ldm bind ldg1
# ldm start ldg1
```

▼ CPU 전체 코어로 기존 도메인을 구성하는 방법

도메인이 존재하고 CPU 스레드를 사용하도록 구성된 경우 CPU 전체 코어를 사용하도록 구성을 변경할 수 있습니다.

- 1 도메인을 중지하고 바인드 해제합니다.

```
# ldm stop domain
# ldm unbind domain
```

- 2 도메인에 대한 CPU 전체 코어 수를 설정합니다.

```
# ldm set-core number-of-cpu-cores domain
```

이 명령은 도메인에 대한 최대 CPU 코어 수도 *number-of-cpu-cores*로 설정합니다.

- 3 도메인을 재바인드하고 다시 시작합니다.

```
# ldm bind domain
# ldm start domain
```

예 10-5 4개의 CPU 전체 코어로 기존 도메인 구성

이 예는 기존 도메인 *ldg1*의 구성을 업데이트합니다. 첫번째 및 두번째 명령은 *ldg1* 도메인을 중지하고 바인드 해제합니다. 세번째 명령은 4개의 CPU 전체 코어로 *ldg1* 도메인을 구성합니다. 이 명령은 *ldg1*에 대한 최대 CPU 코어 수도 4로 설정합니다. 네번째 및 다섯번째 명령은 *ldg1* 도메인을 바인드하고 다시 시작합니다.

```
# ldm stop ldg1
# ldm unbind ldg1
# ldm set-core 4 ldg1
# ldm bind ldg1
# ldm start ldg1
```

▼ CPU 전체 코어로 기본 도메인을 구성하는 방법

primary 도메인이 존재하고 CPU 스레드를 사용하도록 구성된 경우 CPU 전체 코어를 사용하도록 구성을 변경할 수 있습니다.

- 1 *primary* 도메인을 지연된 재구성 모드로 둡니다.

```
# ldm start-reconf primary
```

- 2 *primary* 도메인에 대한 CPU 전체 코어 수를 설정합니다.

```
# ldm set-core number-of-cpu-cores primary
```

이 명령은 *primary* 도메인에 대한 최대 CPU 코어 수도 *number-of-cpu-cores*로 설정합니다.

3 primary 도메인을 재부트합니다.

시스템 구성에 따라 알맞은 절차를 사용하여 primary 도메인을 재부트합니다. [83 페이지](#) “primary 도메인 재부트”를 참조하십시오.

예 10-6 2개의 CPU 전체 코어로 primary 도메인 구성

이 예에서는 primary 도메인에서 CPU 전체 코어를 구성합니다. 첫번째 명령은 primary 도메인에서 지연된 재구성 모드를 시작합니다. 두번째 명령은 2개의 CPU 전체 코어로 primary 도메인을 구성합니다. 이 명령은 primary 도메인에 대한 최대 CPU 코어 수도 2로 설정합니다. 세번째 명령은 primary 도메인을 재부트합니다.

```
# ldm start-reconf primary
# ldm set-core 2 primary
# shutdown -i 5
```

다른 Oracle VM Server for SPARC 기능과의 상호 작용

CPU 동적 재구성

CPU 전체 코어로 구성된 도메인에서 CPU 동적 재구성을 사용할 수 있습니다. 하지만 개별 CPU 스레드가 아닌 전체 CPU 코어만 추가 또는 제거할 수 있습니다. 따라서 시스템의 하드 분할 상태는 CPU 동적 재구성 기능으로 유지됩니다. 또한 CPU 코어가 도메인에 동적으로 추가될 경우 최대값이 강제 적용됩니다. 따라서 최대 CPU 수를 초과하는 경우 CPU DR 명령을 실패합니다.

주 - 도메인이 중지 및 바인드 해제된 상태가 아니라면 `max-cores` 등록 정보는 변경할 수 없습니다. 따라서 전체 코어 제약 조건이 설정되었을 때 지정된 값에서 최대 코어 수를 높이려면 먼저 도메인을 중지 및 바인드 해제해야 합니다.

CPU 전체 코어를 바인드된 도메인이나 활성 도메인에 동적으로 추가, 설정 또는 제거하려면 다음 명령을 사용합니다.

```
ldm add-core number-of-cpu-cores domain
ldm set-core number-of-cpu-cores domain
ldm rm-core number-of-cpu-cores domain
```

주 - 도메인이 활성 상태가 아닌 경우 이러한 명령은 도메인에 대한 최대 CPU 코어 수도 조정합니다. 도메인이 바인드되었거나 활성 상태인 경우 이러한 명령은 도메인에 대한 최대 CPU 코어 수에 영향을 주지 않습니다.

예 10-7 2개의 CPU 전체 코어를 도메인에 동적으로 추가

이 예에서는 2개의 CPU 전체 코어를 `ldg1` 도메인에 동적으로 추가하는 방법을 보여줍니다. `ldg1` 도메인은 CPU 전체 코어로 구성된 활성 도메인입니다. 첫번째 명령은 `ldg1` 도메인이 활성 상태임을 보여줍니다. 두번째 명령은 `ldg1` 도메인이 CPU 전체 코어 및 최대 4개의 CPU 코어로 구성되었음을 보여줍니다. 세번째 및 네번째 명령은 2개의 CPU 전체 코어 추가 전후에 도메인에 지정된 CPU 코어를 보여줍니다. 네번째 명령은 2개의 CPU 전체 코어를 `ldg1` 도메인에 동적으로 추가합니다.

```
# ldm list ldg1
NAME      STATE  FLAGS  CONS  VCPU  MEMORY UTIL  UPTIME
ldg1      active -n---- 5000   16    2G     0.4%  5d 17h 49m

# ldm list -o resgmt ldg1
NAME
ldg1

CONSTRAINT
  whole-core
    max-cores=4

# ldm list -o core ldg1
NAME
ldg1

CORE
CID      PCPUSET
1        (8, 9, 10, 11, 12, 13, 14, 15)
2        (16, 17, 18, 19, 20, 21, 22, 23)

# ldm add-core 2 ldg1
# ldm list -o core ldg1
NAME
ldg1

CORE
CID      PCPUSET
1        (8, 9, 10, 11, 12, 13, 14, 15)
2        (16, 17, 18, 19, 20, 21, 22, 23)
3        (24, 25, 26, 27, 28, 29, 30, 31)
4        (32, 33, 34, 35, 36, 37, 38, 39)
```

CPU 동적 리소스 관리

동적 리소스 관리(DRM)를 사용하여 일부 도메인에서 CPU 리소스를 자동으로 관리할 수 있습니다. DRM이 사용되는 경우 CPU 전체 코어로 구성된 도메인에 DRM 정책이 적용되지 않습니다.

DRM 정책에는 CPU 전체 코어로 구성된 도메인이 포함될 수 있습니다. 하지만 이러한 정책이 활성화된 경우 해당 도메인에 대해 자동으로 사용 안함으로 설정됩니다. 도메인이 나중에 CPU 전체 코어 대신 CPU 스레드로 구성되지 않으면(및 구성될 때까지) 도메인은 계속 CPU 전체 코어로 구성됩니다. 도메인이 CPU 스레드를 사용하도록 구성되면 DRM 정책은 해당 도메인에 대해 자동으로 다시 사용으로 설정됩니다.

CPU 전원 관리

CPU 전체 코어로 구성된 도메인에 대해 성능 또는 탄력적 정책이 사용으로 설정된 CPU 전원 관리(PM)를 사용할 수 있습니다. PM을 사용하면 시스템의 하드 분할이 유지됩니다.

도메인 재부트 또는 재바인드

CPU 전체 코어로 구성된 도메인은 도메인이 다시 시작되거나 전체 시스템이 다시 시작될 경우에도 CPU 전체 코어로 계속 구성됩니다. 도메인은 바인드가 유지되는 기간 내내 동일한 물리적 CPU 코어를 사용합니다. 예를 들어, 도메인이 재부트될 경우 재부트 이전 및 이후 모두 동일한 물리적 CPU 코어를 사용합니다. 또는 도메인이 바인드된 동안 전체 시스템의 전원이 꺼질 경우 시스템의 전원이 켜지면 해당 도메인은 동일한 물리적 CPU 코어로 구성됩니다. 도메인을 바인드 해제한 다음 재바인드하거나 전체 시스템이 새로운 구성으로 다시 시작될 경우에는 도메인이 다른 물리적 CPU 코어를 사용할 수 있습니다.

도메인 마이그레이션 비호환성

CPU 전체 코어 구성은 도메인 마이그레이션과 호환되지 않습니다. 하지만 CPU 전체 코어로 구성된 도메인을 마이그레이션할 수는 있습니다. 이러한 마이그레이션 이후 대상 시스템에서 하드 분할은 적용되지 않습니다. 또한 전체 코어 구성 및 최대 CPU 코어 수는 마이그레이션 시 대상 시스템에서 유지되지 않습니다.

전체 코어로 구성된 도메인을 마이그레이션하는 경우 마이그레이션이 완료된 후 하드 분할을 사용하려면 대상 도메인을 재구성해야 합니다. 또한 소스 및 대상 시스템 모두에서 도메인을 사용할 수 있는 사용권 계약이 있는지 확인해야 합니다.

도메인에 물리적 리소스 지정

Logical Domains Manager는 도메인에 지정될 물리적 리소스를 자동으로 선택합니다. 또한 Oracle VM Server for SPARC 2.2 소프트웨어를 사용하면 도메인에 지정하거나 도메인에서 제거할 물리적 리소스를 명시적으로 선택할 수 있습니다. 이 기능은 제어 도메인에서 Oracle Solaris 11 OS를 실행할 경우만 사용할 수 있습니다.

명시적으로 지정하는 리소스를 **명명된 리소스**라고 합니다. 자동으로 지정되는 리소스를 **익명 리소스**라고 합니다.

제어 도메인 및 게스트 도메인에 물리적 리소스를 명시적으로 지정할 수 있습니다. 제어 도메인은 활성으로 유지되므로 물리적 리소스 지정 전에 제어 도메인은 선택적으로 지연된 재구성 상태일 수 있습니다. 또는 물리적 지정 시 지연된 재구성이 자동으로 트리거될 수 있습니다. [208 페이지 “제어 도메인에서 물리적 리소스 관리”](#)를 참조하십시오. 물리적 리소스 제한 사항에 대한 자세한 내용은 [208 페이지 “도메인에서 물리적 리소스 관리 제한 사항”](#)을 참조하십시오.

제어 도메인 및 게스트 도메인에 다음 물리적 리소스를 명시적으로 지정할 수 있습니다.

■ **물리적 CPU.** cid 등록 정보를 설정하여 물리적 코어 ID를 도메인에 지정합니다.

cid 등록 정보는 구성할 시스템 토폴로지에 대해 잘 알고 있는 관리자**만** 사용해야 합니다. 이 고급 구성 기능은 특정 할당 규칙을 적용하며 시스템의 전체 성능에 영향을 줄 수 있습니다.

다음 중 하나의 명령을 실행하여 이 등록 정보를 설정할 수 있습니다.

```
# ldm add-core cid=core-ID[,core-ID[,...]] ldom
# ldm set-core cid=core-ID[,core-ID[,...]] ldom
# ldm rm-core [-f] cid=core-ID[,core-ID[,...]] ldom
```

코어 ID를 cid 등록 정보의 값으로 지정할 경우 *core-ID*가 명시적으로 도메인에 지정되거나 도메인에서 제거됩니다.

■ **물리적 메모리.** mblock 등록 정보를 설정하여 연속된 물리적 메모리 영역 세트를 도메인에 지정합니다. 각 물리적 메모리 영역은 물리적 메모리 시작 주소 및 크기로 지정됩니다.

mblock 등록 정보는 구성할 시스템 토폴로지에 대해 잘 알고 있는 관리자**만** 사용해야 합니다. 이 고급 구성 기능은 특정 할당 규칙을 적용하며 시스템의 전체 성능에 영향을 줄 수 있습니다.

다음 중 하나의 명령을 실행하여 이 등록 정보를 설정할 수 있습니다.

```
# ldm add-mem mblock=PA-start:size[,PA-start:size[,...]] ldom
# ldm set-mem mblock=PA-start:size[,PA-start:size[,...]] ldom
# ldm rm-mem mblock=PA-start:size[,PA-start:size[,...]] ldom
```

메모리 블록을 도메인에 지정하거나 도메인에서 제거하려면 mblock 등록 정보를 설정합니다. 유효한 값에는 콜론(:)으로 구분된 물리적 메모리 시작 주소(*PA-start*) 및 메모리 블록 크기(*size*)가 포함됩니다.

주 - mblock 또는 cid 등록 정보를 설정할 때 실행 중인 도메인 간에는 동적 재구성(DR)을 사용하여 메모리나 코어 리소스를 이동할 수 **없습니다**. 도메인 간에 리소스를 이동하려면 도메인이 바인드된 상태이거나 바인드 해제된 상태인지 확인합니다. 제어 도메인에서 물리적 리소스 관리에 대한 자세한 내용은 [208 페이지 “제어 도메인에서 물리적 리소스 관리”](#)를 참조하십시오.

ldm list-constraints 명령을 사용하여 도메인에 대한 리소스 제약 조건을 볼 수 있습니다. physical-bindings 제약 조건은 도메인에 물리적으로 지정된 리소스 유형을 지정합니다. 도메인이 만들어질 때 physical-bindings 제약 조건은 물리적 리소스가 해당 도메인에 지정될 때까지 설정되지 않습니다. mblock 등록 정보를 설정하면 physical-bindings 제약 조건이 memory로 설정됩니다. 마찬가지로 cid 등록 정보를 설정하면 physical-bindings 제약 조건이 core로 설정됩니다. cid 및 mblock 등록 정보가 모두 설정될 경우 physical-bindings 제약 조건은 core, memory로 설정됩니다.

제어 도메인에서 리소스 유형에 대한 `physical-bindings` 제약 조건을 변경하려면 먼저 다음과 같이 리소스 수를 0으로 설정하여 해당 유형의 모든 리소스를 **제거**해야 합니다.

- `ldm set-core 0` 또는 `ldm set-mem 0` 명령을 사용하여 리소스 수를 0으로 설정합니다.
- 특정 리소스 유형에 대해 지정된 `physical-bindings` 제약 조건을 모두 제거합니다.
모든 명명된 코어 및 메모리 블록을 제거하려면 `ldm set-core cid=` 및 `ldm set-mem mblock=` 명령을 각각 실행합니다. 모든 익명 코어 및 메모리 블록을 제거하려면 `ldm set-core 0` 및 `ldm set-mem 0` 명령을 각각 실행합니다.
제어 도메인에는 할당된 CPU 및 메모리가 **있어야** 하므로 제어 도메인에서 `cid=` 또는 `mblock=`을 지정하면 오류가 반환됩니다.
- 도메인에서 각 리소스를 개별적으로 삭제합니다.

제어 도메인에서 물리적 리소스 관리

제어 도메인은 항상 활성 상태이므로 물리적 리소스 지정 전에 지연된 재구성 모드일 수 있습니다. 명시적으로 물리적 리소스를 지정할 경우 제어 도메인은 자동으로 지연된 재구성 모드가 되고 `physical-bindings` 제약 조건이 설정됩니다.

`physical-bindings=core`인 경우 `ldm set-core cid=core-ID primary` 명령 또는 `ldm set-vcpu CPU-count primary` 명령을 실행하면 다음 재부트 시 `physical-bindings` 제약 조건이 지워집니다. `physical-bindings` 제약 조건이 `core`로 설정되지 않은 경우 `ldm set-core cid=core-ID primary` 명령을 실행하면 다음 재부트 시 `physical-bindings=core`로 설정됩니다.

`physical-bindings=memory`인 경우 `ldm set-mem size primary` 명령을 실행하면 다음 재부트 시 `physical-bindings` 제약 조건이 지워집니다. `physical-bindings` 제약 조건이 `memory`로 설정되지 않은 경우 `ldm set-mem mblock=PA-start:size primary` 명령을 실행하면 다음 재부트 시 `physical-bindings` 제약 조건이 설정됩니다.

주 - 제어 도메인이 지연된 재구성 모드인 경우 제어 도메인에서 `ldm add-mem` 및 `ldm rm-mem` 명령을 사용하여 무제한 메모리 지정을 수행할 수 있습니다. 하지만 `ldm set-core` 명령을 사용할 경우 제어 도메인에 **하나의** 코어 지정만 수행할 수 있습니다.

도메인에서 물리적 리소스 관리 제한 사항

물리적 리소스 지정에 다음 제한 사항이 적용됩니다.

- 동일 도메인에서 물리적 메모리와 비물리적 메모리 바인딩 또는 물리적 코어와 비물리적 코어 바인딩은 불가능합니다.
- 동일 도메인에서 비물리적 메모리와 물리적 코어 바인딩 또는 비물리적 코어와 물리적 메모리 바인딩은 가능합니다.

- 도메인에 물리적 리소스를 추가할 경우 해당하는 리소스 유형은 물리적 바인딩으로 제약됩니다.
- 개별 CPU를 도메인에 추가하거나 도메인에서 제거하려고 시도하면 `physical-bindings=core`가 실패합니다.
- 바인드되지 않은 리소스의 경우 리소스의 할당 및 확인은 `ldm bind` 명령을 실행할 때만 가능합니다.
- 도메인에서 물리적 메모리를 제거할 경우 이전에 추가된 **동일한** 물리적 메모리 블록을 제거해야 합니다.
- 물리적 메모리 범위는 겹칠 수 **없습니다**.
- `ldm add-core` 또는 `ldm set-core` 명령을 사용하여 도메인에 물리적 리소스를 지정할 수 **없습니다**.
- `ldm add-mem` 또는 `ldm set-mem` 명령을 사용하여 여러 물리적 메모리 블록을 지정할 경우 주소와 크기가 즉시 확인됩니다.
- 일부 코어가 지정된 도메인은 해당 코어의 나머지 CPU가 해제되고 사용 가능한 경우 전체 코어를 사용할 수 있습니다.

메모리 동적 재구성 사용

Oracle VM Server for SPARC 2.0 릴리스에는 메모리 동적 재구성(DR)이 도입되었습니다. 이 기능은 용량 기반이며 임의 용량의 메모리를 활성 논리적 도메인에 추가하거나 제거할 수 있습니다.

다음은 메모리 DR 기능 사용을 위한 요구 사항 및 제한 사항입니다.

- 모든 도메인에서 메모리 DR 작업을 수행할 수 있습니다. 하지만 한 번에 하나의 메모리 DR 작업만 도메인에서 진행 가능합니다.
- 메모리 DR 기능은 한 번의 작업에 포함되는 메모리 주소 및 크기에 대해 256MB 정렬을 적용합니다. [211 페이지 “메모리 정렬”](#)을 참조하십시오.
- 해제 메모리 풀에서 정렬되지 않은 메모리는 메모리 DR 기능을 사용하여 도메인에 지정할 수 **없습니다**. [212 페이지 “정렬되지 않은 메모리 추가”](#)를 참조하십시오.

메모리 DR 작업을 사용하여 도메인의 메모리를 재구성할 수 없는 경우 메모리를 재구성하려면 먼저 도메인을 중지해야 합니다. 도메인이 제어 도메인인 경우 먼저 지연된 재구성을 시작해야 합니다.

메모리 추가

도메인이 활성 상태인 경우 `ldm add-memory` 명령을 사용하여 도메인에 메모리를 동적으로 추가할 수 있습니다. 지정된 메모리 크기가 도메인의 현재 메모리 크기보다 큰 경우 `ldm set-memory` 명령도 메모리를 동적으로 추가할 수 있습니다.

메모리 제거

도메인이 활성 상태인 경우 `ldm remove-memory` 명령을 사용하여 도메인에서 메모리를 동적으로 제거할 수 있습니다. 지정된 메모리 크기가 도메인의 현재 메모리 크기보다 작은 경우 `ldm set-memory` 명령도 메모리를 동적으로 제거할 수 있습니다.

메모리 제거는 오래 실행되는 작업이 될 수 있습니다. 작업 진행을 추적하거나 진행 중인 메모리 DR 요청을 취소할 수 있습니다.

메모리 DR 요청 진행 추적

지정된 도메인에 대해 `ldm list -l` 명령을 실행하여 `ldm remove-memory` 명령 진행을 추적할 수 있습니다.

메모리 DR 요청 취소

`ldm remove-memory` 명령을 중단(Ctrl-C 누름)시키거나 `ldm cancel-operation memdr` 명령을 실행하여 진행 중인 제거 요청을 취소할 수 있습니다. 메모리 제거 요청을 취소할 경우 제거 요청의 미결 부분(즉, 도메인에서 제거될 메모리 용량)만 영향을 받습니다.

일부 메모리 DR 요청

전체 요청을 처리할 수 있는 해제 메모리가 부족한 경우 메모리 추가 요청은 거부됩니다. 하지만 대상 도메인이 Logical Domains Manager에서 요청한 메모리를 모두 추가하지 못할 경우 메모리 추가 요청을 부분적으로 처리할 수 있습니다.

도메인에 전체 요청을 처리할 수 있는 메모리가 부족한 경우 메모리 제거 요청은 거부됩니다. 하지만 대상 도메인이 Logical Domains Manager에서 요청한 메모리를 모두 제거하지 못할 경우 메모리 제거 요청을 부분적으로 처리할 수 있습니다.

주 - 메모리는 도메인에서 제거된 후 다른 도메인에 추가되기 전에 지워집니다.

제어 도메인의 메모리 재구성

메모리 DR 기능을 사용하여 제어 도메인의 메모리를 재구성할 수 있습니다. 제어 도메인에서 메모리 DR 요청을 수행할 수 없는 경우 먼저 지연된 재구성을 시작해야 합니다.

메모리 DR 작업은 오래 실행될 수 있으므로 활성 도메인에서 많은 용량의 메모리 제거를 위해 메모리 DR을 사용하는 것은 적절하지 않을 수 있습니다. 특히, 시스템의 초기 구성 중에는 지연된 재구성을 사용하여 제어 도메인의 메모리를 줄여야 합니다.

제어 도메인의 메모리 줄이기

초기 공장 출하시 기본 구성에서 제어 도메인의 메모리를 줄이려면 메모리 DR 대신 지연된 재구성을 사용합니다. 이러한 경우 제어 도메인의 호스트 시스템의 모든 메모리를 소유하고 있습니다. 활성 도메인은 요청된 모든 메모리를 추가하거나 더 일반적으로는 포기하지 않으므로 메모리 DR 기능은 이 용도로 적합하지 않습니다. 한편 해당 도메인에서 실행 중인 OS는 요청을 처리하기 위해 최선을 다합니다. 또한 메모리 제거는 오래 실행되는 작업이 될 수 있습니다. 초기 제어 도메인의 메모리 줄이기와 마찬가지로 대량 메모리 작업이 수행되는 경우 이러한 문제는 더 커집니다.

이러한 이유로 다음 단계를 수행하여 지연된 재구성을 사용하십시오.

1. `ldm start-reconf primary` 명령을 사용하여 컨트롤 도메인을 지연된 재구성 모드로 설정합니다.
2. 필요에 따라 제어 도메인이 소유하고 있는 호스트 시스템의 리소스를 분할합니다.
3. 필요한 경우 `ldm cancel-reconf` 명령을 사용하여 단계 2의 작업을 실행 취소하고 다시 시작합니다.
4. 제어 도메인을 재부트하여 재구성 변경 사항을 적용합니다.

동적 및 지연된 재구성

지연된 재구성이 제어 도메인에서 보류 중인 경우 다른 도메인에 대한 메모리 재구성 요청은 거부됩니다. 지연된 재구성이 제어 도메인에서 보류 중이 아닌 경우 메모리 DR을 지원하지 않는 도메인에 대한 메모리 재구성 요청은 거부됩니다. 메모리 DR을 지원하지 않는 제어 도메인에서 메모리 재구성 요청은 지연된 재구성 요청으로 변환됩니다.

메모리 정렬

메모리 재구성 요청은 요청이 적용되는 도메인의 상태에 따라 정렬 요구 사항이 달라집니다.

활성 도메인에 대한 메모리 정렬

- **동적 추가 및 제거.** 메모리 블록의 주소와 크기는 동적 추가 및 동적 제거의 경우 256MB로 정렬됩니다. 최소 작업 크기는 256MB입니다.
정렬되지 않은 요청이나 바인드된 크기보다 큰 제거 요청은 거부됩니다.

메모리 할당을 조정하려면 다음 명령을 사용합니다.

- **ldm add-memory.** 이 명령과 함께 **--auto-adj** 옵션을 지정할 경우 추가되는 메모리 용량은 256MB로 정렬되며, 이 경우 도메인에 실제로 추가되는 메모리의 용량이 늘어날 수 있습니다.
- **ldm remove-memory.** 이 명령과 함께 **--auto-adj** 옵션을 지정할 경우 제거되는 메모리 용량은 256MB로 정렬되며, 이 경우 도메인에서 실제로 제거되는 메모리의 용량이 줄어 들 수 있습니다.
- **ldm set-memory.** 이 명령은 추가 또는 제거 작업으로 취급됩니다. **--auto-adj** 옵션을 지정할 경우 추가되거나 제거되는 메모리 용량은 앞에서 설명한 대로 256MB로 정렬됩니다. 이 정렬로 도메인의 실제 메모리 크기가 늘어날 수 있습니다.
- **지연된 재구성.** 메모리 블록의 주소와 크기는 4MB로 정렬됩니다. 정렬되지 않은 요청을 할 경우 요청이 반올림되어 4MB로 정렬됩니다.

바인드된 도메인에 대한 메모리 정렬

메모리 블록의 주소와 크기는 바인드된 도메인의 경우 4MB로 정렬됩니다. 정렬되지 않은 요청을 할 경우 요청이 반올림되어 4MB로 정렬됩니다. 이로 인해 도메인의 실제 메모리 크기는 예상한 것보다 늘어날 수 있습니다.

ldm add-memory, **ldm set-memory** 및 **ldm remove-memory** 명령의 경우 **--auto-adj** 옵션은 결과 메모리의 크기를 반올림하여 256MB로 정렬합니다. 이로 인해 실제 메모리는 예상한 것보다 늘어날 수 있습니다.

비활성 도메인에 대한 메모리 정렬

ldm add-memory, **ldm set-memory** 및 **ldm remove-memory** 명령의 경우 **--auto-adj** 옵션은 결과 메모리의 크기를 반올림하여 256MB로 정렬합니다. 비활성 도메인에 대한 정렬 요구 사항은 없습니다. [212 페이지 “바인드된 도메인에 대한 메모리 정렬”](#)에 설명된 제한 사항은 이러한 도메인이 바인드된 후 적용됩니다.

정렬되지 않은 메모리 추가

메모리 DR 기능은 활성 도메인에 동적으로 추가되거나 제거되는 메모리 주소 및 크기에 대해 256MB 메모리 정렬을 적용합니다. 이에 따라 활성 도메인에서 정렬되지 않은 메모리는 메모리 DR을 사용하여 제거할 수 없습니다.

또한 해제 메모리 풀에서 정렬되지 않은 메모리는 메모리 DR을 사용하여 활성 도메인에 추가할 수 없습니다.

모든 정렬된 메모리가 할당된 후 **ldm add-memory** 명령을 사용하여 나머지 정렬되지 않은 메모리를 바인드된 도메인이나 활성 도메인에 추가할 수 있습니다. 또한 이 명령을 사용하면 지연된 재구성 작업을 통해 나머지 정렬되지 않은 메모리를 제어 도메인에 추가할 수 있습니다.

다음 예는 2개의 나머지 128MB 메모리 블록을 **primary** 및 **ldom1** 도메인에 추가하는 방법을 보여줍니다. **ldom1** 도메인은 바인드된 상태입니다. 다음 명령은 2개의 나머지 메모리 블록을 추가합니다. 첫번째 명령은 제어 도메인에서 지연된 재구성 작업을 시작합니다. 두번째 명령은 128MB 메모리 블록 중 하나를 제어 도메인에 추가합니다. 다섯번째 명령은 다른 128MB 메모리 블록을 **ldom1** 도메인에 추가합니다.

```
# ldm start-reconf primary
```

Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the
primary domain reboots, at which time the new configuration for the
primary domain also takes effect.

```
# ldm add-memory 128M primary
```

Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-ndcv-	SP	8	2688M	0.1%	23d 8h 8m

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	8	2560M	0.5%	23d 8h 9m
ldom1	bound	-----	5000	1	524M		

```
# ldm add-mem 128M ldom1
```

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	8	2560M	0.1%	23d 8h 9m
ldom1	bound	-----	5000	1	652M		

메모리 DR 예

다음 예에서는 메모리 DR 작업을 수행하는 방법을 보여줍니다. 관련 CLI 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예 10-8 활성 도메인에서 메모리 DR 작업

이 예는 활성 도메인인 **ldom1**에 동적으로 메모리를 추가하고 제거하는 방법을 보여줍니다.

ldm list 출력은 Memory 필드의 각 도메인에 대한 메모리를 보여줍니다. 메모리를 256MB의 배수로 지정해야 하므로 첫번째 **ldm add-mem** 명령은 오류와 함께 종료됩니다. 다음 **ldm add-mem** 명령은 **--auto-adj** 옵션을 사용하므로 200M를 추가할 메모리 용량으로 지정하더라도 반올림되어 256MB가 됩니다.

메모리를 256MB의 배수로 지정해야 하므로 **ldm rm-mem** 명령은 오류와 함께 종료됩니다. 동일 명령에 **--auto-adj** 옵션을 추가하면 메모리 용량이 반내림되어 다음 256MB 경계가 되므로 메모리 제거를 성공합니다.

예 10-8 활성 도메인에서 메모리 DR 작업 (계속)

```
# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP       4       27392M    0.4%    1d 22h 53m
ldom1         active    -n----   5000     2       2G        0.4%    1d 1h 23m
ldom2         bound     ------ 5001     2       200M

# ldm add-mem 200M ldom1
The size of memory must be a multiple of 256MB.

# ldm add-mem --auto-adj 200M ldom1
Adjusting request size to 256M.
The ldom1 domain has been allocated 56M more memory
than requested because of memory alignment constraints.

# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP       4       27392M    5.0%     8m
ldom1         active    -n----   5000     2       2304M     0.5%     1m
ldom2         bound     ------ 5001     2       200M

# ldm rm-mem --auto-adj 300M ldom1
Adjusting requested size to 256M.
The ldom1 domain has been allocated 44M more memory
than requested because of memory alignment constraints.

# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP       4       27392M    0.3%     8m
ldom1         active    -n----   5000     2       2G        0.2%     2m
ldom2         bound     ------ 5001     2       200M
```

예 10-9 바인드된 도메인에서 메모리 DR 작업

이 예는 바인드된 도메인인 ldom2에 동적으로 메모리를 추가하고 제거하는 방법을 보여줍니다.

ldm list 출력은 Memory 필드의 각 도메인에 대한 메모리를 보여줍니다. 첫번째 ldm add-mem 명령은 100MB의 메모리를 ldom2 도메인에 추가합니다. 다음 ldm add-mem 명령은 --auto-adj 옵션을 지정하여 추가 112MB의 메모리가 ldom2에 동적으로 추가됩니다.

ldm rm-mem 명령은 ldom2 도메인에서 100MB를 동적으로 제거합니다. 동일 명령에 --auto-adj 옵션을 사용하여 300MB의 메모리를 제거할 경우 메모리 용량이 반내림되어 다음 256MB 경계가 됩니다.

```
# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP       4       27392M    0.4%    1d 22h 53m
ldom1         active    -n----   5000     2       2G        0.4%    1d 1h 23m
ldom2         bound     ------ 5001     2       200M

# ldm add-mem 100M ldom2

# ldm list
```

예 10-9 바인드된 도메인에서 메모리 DR 작업 (계속)

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	27392M	0.5%	1d 22h 54m
ldom1	active	-n----	5000	2	2G	0.2%	1d 1h 25m
ldom2	bound	-----	5001	2	300M		

```
# ldm add-mem --auto-adj 100M ldom2
```

Adjusting request size to 256M.

The ldom2 domain has been allocated 112M more memory than requested because of memory alignment constraints.

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	27392M	0.4%	1d 22h 55m
ldom1	active	-n----	5000	2	2G	0.5%	1d 1h 25m
ldom2	bound	-----	5001	2	512M		

```
# ldm rm-mem 100M ldom2
```

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	27392M	3.3%	1d 22h 55m
ldom1	active	-n----	5000	2	2G	0.2%	1d 1h 25m
ldom2	bound	-----	5001	2	412M		

```
# ldm rm-mem --auto-adj 300M ldom2
```

Adjusting request size to 256M.

The ldom2 domain has been allocated 144M more memory than requested because of memory alignment constraints.

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	27392M	0.5%	1d 22h 55m
ldom1	active	-n----	5000	2	2G	0.2%	1d 1h 26m
ldom2	bound	-----	5001	2	256M		

예 10-10 도메인 메모리 크기 설정

이 예는 ldm set-memory 명령을 사용하여 도메인에 메모리를 추가하고 제거하는 방법을 보여줍니다.

ldm list 출력은 Memory 필드의 각 도메인에 대한 메모리를 보여줍니다. 첫번째 ldm set-mem 명령은 primary 도메인의 크기를 3400MB로 설정하려고 시도합니다. 지정된 값이 256MB 경계에 있지 않다는 오류가 나타납니다. 동일 명령에 --auto-adj 옵션을 추가하면 일부 메모리를 자동으로 제거하고 256MB 경계에 있을 수 있습니다. 또한 이 명령은 도메인에서 해당 메모리를 사용 중이므로 요청된 메모리 중 일부를 제거할 수 없다는 경고를 표시합니다.

다음 ldm set-mem 명령은 바인드된 상태인 ldom2 도메인의 메모리 크기를 690MB로 설정합니다. 동일 명령에 --auto-adj 옵션을 사용할 경우 추가 78MB의 메모리가 ldom2에 동적으로 추가되어 256MB 경계에 있게 됩니다.

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	27392M	0.5%	1d 22h 55m

예 10-10 도메인 메모리 크기 설정 (계속)

```
ldom1          active   -n---- 5000    2    2G      0.2% 1d 1h 26m
ldom2          bound   ------ 5001    2    256M

# ldm set-mem 3400M primary
An ldm set-mem 3400M command would remove 23992MB, which is not a multiple
of 256MB. Instead, run ldm rm-mem 23808MB to ensure a 256MB alignment.

# ldm set-mem --auto-adj 3400M primary
Adjusting request size to 3.4G.
The primary domain has been allocated 184M more memory
than requested because of memory alignment constraints.
Only 9472M of memory could be removed from the primary domain
because the rest of the memory is in use.

# ldm set-mem 690M ldom2
# ldm list
NAME          STATE   FLAGS   CONS   VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv-  SP     4     17920M  0.5%  1d 22h 56m
ldom1         active -n---- 5000    2     2G     0.6%  1d 1h 27m
ldom2         bound  ------ 5001    2     690M

# ldm set-mem --auto-adj 690M ldom2
Adjusting request size to 256M.
The ldom2 domain has been allocated 78M more memory
than requested because of memory alignment constraints.

# ldm list
NAME          STATE   FLAGS   CONS   VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv-  SP     4     17920M  2.1%  1d 22h 57m
ldom1         active -n---- 5000    2     2G     0.2%  1d 1h 27m
ldom2         bound  ------ 5001    2     768M
```

전원 관리 사용

전원 관리(PM)를 사용하려면 먼저 Oracle ILOM(Integrated Lights Out Manager) 3.0 펌웨어에서 PM 정책을 설정해야 합니다. 이 절에서는 Oracle VM Server for SPARC 소프트웨어에서 PM을 사용하기 위해 필요한 정보를 요약합니다.

ILOM에 대한 자세한 내용은 다음을 참조하십시오.

- **Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide**의 “Monitoring Power Consumption”
- **Oracle Integrated Lights Out Manager (ILOM) 3.0 Feature Updates and Release Notes**

전원 정책은 특정 시점의 시스템 전원 사용을 제어합니다. 기본 플랫폼에서 PM 기능을 구현했다면 다음 전원 정책이 지원됩니다.

- **성능.** 시스템이 사용 가능한 모든 전원을 사용하도록 허용됩니다.
- **탄력적.** 시스템 전원이 현재 사용률 레벨에 맞추어 조정됩니다. 예를 들어, 사용률이 줄어들면 리소스의 전원 상태가 감소합니다.

다음은 PM 기능입니다.

- **CPU 코어 자동 사용 안함.** 해당 코어의 모든 스레드(스트랜드)가 사용 안함으로 설정되면 PM에서 자동으로 CPU 코어를 사용 안함으로 설정합니다.
- **CPU 클럭 사이클 건너뛰기.** Oracle VM Server for SPARC 2.0 릴리스부터 PM에서 CPU 클럭 사이클 건너뛰기를 자동으로 조정할 수 있습니다. 이 기능은 SPARC T3 및 SPARC T4 플랫폼에 대해서만 사용할 수 있습니다. 모든 도메인이 전원 사용률 임계값 내에 유지되도록 조정을 통해 건너뛰는 클럭 사이클의 수를 높이거나 낮출 수 있습니다. PM은 CPU 사용률을 기준으로 이러한 조정 여부를 결정합니다. 시스템에서 성능 정책을 사용하는 경우 건너뛰는 클럭 사이클의 수는 자동으로 없음으로 조정됩니다.
- **깊은 유휴 모드로 메모리 작동.** Oracle VM Server for SPARC 2.0 릴리스부터 SPARC T3 및 SPARC T4 플랫폼에 탄력적 정책이 적용 중인 경우 전원 절약을 위해 사용되지 않는 메모리가 더 깊은 유휴 모드로 작동하도록 자동으로 구성됩니다.
- **전원 제한.** SPARC T3 및 SPARC T4 플랫폼에서 **전원 제한**을 설정하여 시스템의 전원 소비를 제한할 수 있습니다. 전원 소비가 전원 제한보다 클 경우 PM 기술이 사용되어 전원을 줄입니다. ILOM 서비스 프로세서(SP)를 사용하여 전원 제한을 설정할 수 있습니다.

다음 문서를 참조하십시오.

- **Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide**
- **Oracle Integrated Lights Out Manager (ILOM) 3.0 Feature Updates and Release Notes**

ILOM 인터페이스를 사용하여 전원 제한, 유예 기간 및 위반 작업을 설정할 수 있습니다. 전원 제한이 유예 기간보다 오랫동안 초과될 경우 위반 작업이 수행됩니다.

현재 전원 소비가 전원 제한을 초과할 경우 전원 관리가 가능한 리소스의 전원 상태를 줄이기 위한 시도가 수행됩니다. 전원 소비가 전원 제한 아래로 낮아지면 이러한 리소스의 전원 상태는 다시 높아질 수 있습니다. 시스템에 탄력적 정책이 적용 중인 경우 리소스의 전원 상태 증가는 사용률 레벨로 결정됩니다.

시스템에 탄력적 정책이 적용 중인 경우 일부 도메인 구성 수정이 먼저 검증되어 전원 제한이 초과되지 않는지 확인합니다. 전원 제한이 초과될 경우 요청에 따라 리소스 중 일부만 수정되거나 추가될 수 있습니다. 나중에 전원 제한이 늘어나면 성공적으로 수정되지 않은 리소스를 추가할 수 있습니다.

도메인의 로드로 인해 리소스에서 더 많은 전원을 소비할 경우 전원 소비를 전원 제한 아래로 유지하는 리소스만 성공적으로 전원이 증가됩니다.

ILOM 3.0 펌웨어 CLI를 사용하여 전원 정책을 구성하는 방법에 대한 자세한 내용은 **Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide**의 “Monitoring Power Consumption”을 참조하십시오.

전원 관리되는 CPU 스레드 및 가상 CPU 나열

이 절에서는 전원 관리되는 CPU 스레드 및 가상 CPU를 나열하는 방법을 설명합니다.

▼ 전원 관리되는 CPU 스레드를 나열하는 방법

- 다음 명령 중 하나를 사용하여 전원 관리되는 스레드를 나열합니다.

- `list -l` 하위 명령을 사용합니다.

```
# ldm list -l primary
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-  UART     64    16G     1.0%  21h 33m

SOFTSTATE
Solaris running

UUID
b9288150-327f-44f7-8c64-d4d57b92e524

MAC
00:21:28:8f:8f:34

HOSTID
0x858f8f34

CONTROL
failure-policy=ignore

DEPENDENCY
master=

CORE
CID  CPUSSET
0    (0, 1, 2, 3, 4, 5, 6, 7)
1    (8, 9, 10, 11, 12, 13, 14, 15)
2    (16, 17, 18, 19, 20, 21, 22, 23)
3    (24, 25, 26, 27, 28, 29, 30, 31)
4    (32, 33, 34, 35, 36, 37, 38, 39)
5    (40, 41, 42, 43, 44, 45, 46, 47)
6    (48, 49, 50, 51, 52, 53, 54, 55)
7    (56, 57, 58, 59, 60, 61, 62, 63)

VCPU
VID  PID  CID  UTIL  STRAND
0    0    0    1.2%  100%
1    1    0    1.5%  100%
2    2    0    0.1%  100%
3    3    0    0.2%  100%
.
.
.
```

다음 출력은 CPU에 대한 UTIL 열에서 대시(---)를 표시하며, 이것은 스레드가 전원 관리됨을 의미합니다. 대시는 primary 도메인 이외의 도메인에 대해서만 표시됩니다.

```
# ldm list -l ldg1
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
ldg1          active     -n--v-   5000    64    16G     1.1%  20h 55m

SOFTSTATE
Solaris running

UUID
98d86371-24f6-4792-c631-eb14e81ad4a0

MAC
00:14:4f:f9:02:f2

HOSTID
0x84f902f2

CONTROL
failure-policy=ignore

DEPENDENCY
master=

CORE
CID    CPuset
8      (64, 65, 66, 67, 68, 69, 70, 71)
9      (72, 73, 74, 75, 76, 77, 78, 79)
10     (80, 81, 82, 83, 84, 85, 86, 87)
11     (88, 89, 90, 91, 92, 93, 94, 95)
12     (96, 97, 98, 99, 100, 101, 102, 103)
13     (104, 105, 106, 107, 108, 109, 110, 111)
14     (112, 113, 114, 115, 116, 117, 118, 119)
15     (120, 121, 122, 123, 124, 125, 126, 127)

VCPU
VID    PID    CID    UTIL  STRAND
0      64     8      0.8%  100%
1      65     8      2.0%  100%
2      66     8      ----  100%
3      67     8      ----  100%
4      68     8      ----  100%
.
.
.
```

- **list -l** 하위 명령에 구문 분석 가능 옵션(-p)을 사용합니다.

출력에서 util= 다음의 공백은 스레드(스트랜드)가 전원 관리됨을 의미합니다.

```
# ldm list -l -p

VCPU
|vid=0|pid=0|util=0.7%|strand=100
|vid=1|pid=1|util=|strand=100
|vid=2|pid=2|util=|strand=100
|vid=3|pid=3|util=|strand=100
|vid=4|pid=4|util=0.7%|strand=100
```

```
|vid=5|pid=5|util=|strand=100
|vid=6|pid=6|util=|strand=100
|vid=7|pid=7|util=|strand=100
```

▼ 전원 관리되는 CPU를 나열하는 방법

- 다음 명령 중 하나를 사용하여 전원 관리되는 CPU를 나열합니다.

a. `list-devices -a cpu` 명령을 사용합니다.

출력의 PM 열에서 yes는 CPU가 전원 관리됨을 의미하고 no는 CPU 전원이 켜졌음을 의미합니다. 100% 해제 CPU는 기본적으로 전원 관리되는 것으로 간주되므로 PM 열 아래에 대시(---)가 표시됩니다.

```
# ldm list-devices -a cpu
VCPU
  PID      %FREE      PM
  0         0      no
  1         0      yes
  2         0      yes
  3         0      yes
  4        100      ---
  5        100      ---
  6        100      ---
  7        100      ---
```

b. `list-devices -a cpu` 하위 명령에 구문 분석 가능 옵션(-p)을 사용합니다.

출력의 pm= 필드에서 yes는 CPU가 전원 관리됨을 의미하고 no는 CPU 전원이 켜졌음을 의미합니다. 100% 해제 CPU는 기본적으로 전원 관리되는 것으로 간주되므로 해당 필드에 공백이 나타납니다.

```
# ldm list-devices -a -p cpu
VERSION 1.6
VCPU
|pid=0|free=0|pm=no
|pid=1|free=0|pm=yes
|pid=2|free=0|pm=yes
|pid=3|free=0|pm=yes
|pid=4|free=0|pm=no
|pid=5|free=0|pm=yes
|pid=6|free=0|pm=yes
|pid=7|free=0|pm=yes
|pid=8|free=100|pm=
|pid=9|free=100|pm=
|pid=10|free=100|pm=
```

동적 리소스 관리 사용

정책을 사용하여 DR 작업을 자동으로 수행할 방법을 결정할 수 있습니다. 이 때 가상 CPU의 동적 리소스 관리를 제어하기 위한 정책만 만들 수 있습니다.



주의 - 다음 제한 사항은 CPU 동적 리소스 관리(DRM)에 영향을 줍니다.

- PM 탄력적 정책이 설정된 경우 DRM을 사용으로 설정할 수 없습니다.
- DRM이 사용으로 설정되어 있는 동안에는 성능 정책에서 탄력적 정책으로의 변경이 지연됩니다.
- 도메인 마이그레이션 작업을 수행하기 전에 CPU DRM을 사용 안함으로 설정했는지 확인합니다.
- DRM 정책은 whole-core 제약 조건으로 구성된 도메인에 적용되지 않습니다.

리소스 관리 정책은 어떤 조건에서 가상 CPU를 논리적 도메인에 자동으로 추가하고 제거할 수 있는지 지정합니다. 정책은 `ldm add-policy`, `ldm set-policy` 및 `ldm remove-policy` 명령을 사용하여 관리됩니다.

```
ldm add-policy [enable=yes|no] [priority=value] [attack=value] [decay=value]
               [elastic-margin=value] [sample-rate=value] [tod-begin=hh:mm[:ss]]
               [tod-end=hh:mm[:ss]] [util-lower=percent] [util-upper=percent] [vcpu-min=value]
               [vcpu-max=value] name=policy-name ldom...
ldm set-policy [enable=[yes|no]] [priority=[value]] [attack=[value]] [decay=[value]]
               [elastic-margin=[value]] [sample-rate=[value]] [tod-begin=[hh:mm:ss]]
               [tod-end=[hh:mm:ss]] [util-lower=[percent]] [util-upper=[percent]] [vcpu-min=[value]]
               [vcpu-max=[value]] name=policy-name ldom...
ldm remove-policy [name=]policy-name... ldom
```

이러한 명령 및 리소스 관리 정책 만들기에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

정책은 `tod-begin` 및 `tod-end` 등록 정보로 지정된 기간 동안 유효합니다. `tod-begin`으로 지정된 시간은 24시간 기간에서 `tod-end`로 지정된 시간보다 이전이어야 합니다. 기본적으로 `tod-begin` 및 `tod-end` 등록 정보에 대한 값은 각각 00:00:00 및 23:59:59입니다. 기본값이 사용될 경우 정책은 항상 유효합니다.

정책에서는 `priority` 등록 정보의 값을 사용하여 동적 리소스 관리(DRM) 정책에 대한 우선 순위를 지정합니다. 우선 순위 값은 단일 도메인에서 DRM 정책 사이의 관계 및 단일 시스템에서 DRM이 사용으로 설정된 도메인 사이의 관계를 결정하는 데 사용됩니다. 숫자 값이 낮을수록 더 높은(좋은) 우선 순위를 나타냅니다. 유효한 값은 1-9999입니다. 기본값은 99입니다.

priority 등록 정보의 동작은 다음과 같이 해제 CPU 리소스 풀의 가용성에 따라 달라집니다.

- **해제 CPU 리소스를 풀에서 사용할 수 있음.** 이 경우 priority 등록 정보는 단일 도메인에 대해 둘 이상의 겹치는 정책이 정의된 경우 적용할 DRM 정책을 결정합니다.
- **해제 CPU 리소스를 풀에서 사용할 수 없음.** 이 경우 priority 등록 정보는 동일 시스템의 낮은 우선 순위 도메인에서 높은 우선 순위 도메인으로 리소스를 동적으로 이동할 수 있는지 여부를 지정합니다. 도메인의 우선 순위는 해당 도메인에 대해 적용된 DRM 정책으로 지정되는 우선 순위입니다.

예를 들어, 높은 우선 순위의 도메인은 낮은 우선 순위의 DRM 정책을 가진 다른 도메인에서 CPU 리소스를 가져올 수 있습니다. 이 리소스 획득 기능은 DRM 정책이 사용으로 설정된 도메인에 **만** 해당합니다. priority 값이 같은 도메인은 이 기능의 영향을 받지 않습니다. 따라서 모든 정책에 대해 기본 우선 순위가 사용될 경우 도메인은 낮은 우선 순위의 도메인에서 리소스를 가져올 수 없습니다. 이 기능을 활용하려면 priority 등록 정보 값이 같지 않도록 조정하십시오.

예를 들어, ldg1 및 ldg2 도메인에는 모두 적용 중인 DRM 정책이 있습니다. ldg1 도메인에 대한 priority 등록 정보는 ldg2 도메인의 priority 등록 정보 값(2)보다 우선하는 1입니다. 다음 상황에서 ldg1 도메인은 ldg2 도메인에서 CPU 리소스를 동적으로 제거하고 자신에게 지정할 수 있습니다.

- ldg1 도메인에 다른 CPU 리소스가 필요합니다.
- 해제 CPU 리소스 풀이 모두 소진되었습니다.

정책에서는 util-high 및 util-low 등록 정보 값을 사용하여 CPU 사용률에 대한 상위 및 하위 임계값을 지정합니다. 사용률이 util-high의 값을 초과할 경우 숫자가 vcpu-min과 vcpu-max 값 사이가 될 때까지 가상 CPU가 도메인에 추가됩니다. 사용률이 util-low 값 아래로 떨어지면 숫자가 vcpu-min과 vcpu-max 값 사이가 될 때까지 가상 CPU가 도메인에서 제거됩니다. vcpu-min에 도달하면 더 이상 가상 CPU를 동적으로 제거할 수 없습니다. vcpu-max에 도달하면 더 이상 가상 CPU를 동적으로 추가할 수 없습니다.

예 10-11 리소스 관리 정책 추가

예를 들어, 여러 주에 걸쳐 시스템의 일반적인 사용률을 관찰한 후 리소스 사용을 최적화하기 위한 정책을 설정할 수 있습니다. 가장 높은 사용 시간은 매일 오전 9:00 - 오후 6:00(태평양 표준시)이고, 낮은 사용 시간은 매일 오후 6:00 - 오전 9:00(태평양 표준시)입니다.

이 시스템 사용률 관찰 결과에 따라 전체적인 시스템 사용률을 기준으로 다음과 같이 상위 및 하위 정책을 만들기로 결정합니다.

- **상위:** 매일 오전 9:00 - 오후 6:00(태평양 표준시)
- **하위:** 매일 오후 6:00 - 오전 9:00(태평양 표준시)

다음 ldm add-policy 명령은 ldom1 도메인에서 높은 사용률 기간 동안 사용될 high-usage 정책을 만듭니다.

예 10-11 리소스 관리 정책 추가 (계속)

다음 high-usage 정책은 다음을 수행합니다.

- `tod-begin` 및 `tod-end` 등록 정보를 설정하여 시작 및 종료 시간이 각각 오전 9:00 및 오후 6:00가 되도록 지정합니다.
- `util-lower` 및 `util-upper` 등록 정보를 설정하여 정책 분석을 수행할 하한 및 상한이 각각 25% 및 75%가 되도록 지정합니다.
- `vcpu-min` 및 `vcpu-max` 등록 정보를 설정하여 최소 및 최대 가상 CPU 수가 각각 2 및 16이 되도록 지정합니다.
- `attack` 등록 정보를 설정하여 한 리소스 제어 사이클 중 추가되는 최대 가상 CPU 수가 1이 되도록 지정합니다.
- `decay` 등록 정보를 설정하여 한 리소스 제어 사이클 중 제거되는 최대 가상 CPU 수가 1이 되도록 지정합니다.
- `priority` 등록 정보를 설정하여 이 정책의 우선 순위가 1이 되도록 지정합니다. 우선 순위 1은 다른 정책을 적용할 수 있더라도 이 정책이 강제 적용됨을 의미합니다.
- `name` 등록 정보를 설정하여 정책 파일의 이름이 `high-usage`가 되도록 지정합니다.
- 지정되지 않은 등록 정보(`enable` 및 `sample-rate` 등)에 대해 기본값을 사용합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
# ldm add-policy tod-begin=09:00 tod-end=18:00 util-lower=25 util-upper=75 \
vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1 name=high-usage ldm1
```

다음 `ldm add-policy` 명령은 `ldm1` 도메인에서 낮은 사용률 기간 동안 사용될 `med-usage` 정책을 만듭니다.

다음 `med-usage` 정책은 다음을 수행합니다.

- `tod-begin` 및 `tod-end` 등록 정보를 설정하여 시작 및 종료 시간이 각각 오후 6:00 및 오전 9:00가 되도록 지정합니다.
- `util-lower` 및 `util-upper` 등록 정보를 설정하여 정책 분석을 수행할 하한 및 상한이 각각 10% 및 50%가 되도록 지정합니다.
- `vcpu-min` 및 `vcpu-max` 등록 정보를 설정하여 최소 및 최대 가상 CPU 수가 각각 2 및 16이 되도록 지정합니다.
- `attack` 등록 정보를 설정하여 한 리소스 제어 사이클 중 추가되는 최대 가상 CPU 수가 1이 되도록 지정합니다.
- `decay` 등록 정보를 설정하여 한 리소스 제어 사이클 중 제거되는 최대 가상 CPU 수가 1이 되도록 지정합니다.
- `priority` 등록 정보를 설정하여 이 정책의 우선 순위가 1이 되도록 지정합니다. 우선 순위 1은 다른 정책을 적용할 수 있더라도 이 정책이 강제 적용됨을 의미합니다.
- `name` 등록 정보를 설정하여 정책 파일의 이름이 `high-usage`가 되도록 지정합니다.

예 10-11 리소스 관리 정책 추가 (계속)

- 지정되지 않은 등록 정보(enable 및 sample-rate 등)에 대해 기본값을 사용합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
# ldm add-policy tod-begin=18:00 tod-end=09:00 util-lower=10 util-upper=50 \
  vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1 name=med-usage ldom1
```

도메인 리소스 나열

이 절에서는 ldm 하위 명령에 대한 구문 사용법을 표시하고, 플래그 및 사용률 통계와 같은 몇 가지 출력 용어를 정의하며, 실제로 출력으로 보게 되는 것과 유사한 예를 제공합니다.

시스템에서 읽을 수 있는 출력

ldm list 명령 출력을 사용하는 스크립트를 만들 경우 **항상** -p 옵션을 사용하여 시스템에서 읽을 수 있는 형식의 출력을 생성하십시오. 자세한 내용은 [226 페이지](#) “구문 분석 가능하고 시스템에서 읽을 수 있는 목록을 생성하는 방법(-p)”을 참조하십시오.

▼ ldm 명령에 대한 구문 사용법을 표시하는 방법

- 모든 ldm 하위 명령에 대한 구문 사용법을 표시합니다.

```
# ldm --help
```

ldm 하위 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

플래그 정의

도메인에 대한 출력(ldm list)에 다음 플래그가 표시될 수 있습니다. 명령에 대해 긴 구문 분석 가능 옵션(-l -p)을 사용할 경우 플래그 철자가 나타납니다(예:

flags=normal,control,vio-service). 그렇지 않을 경우 문자 약어를 보게 됩니다(예: -n-cv-). list 플래그 값의 위치에 따라 달라집니다. 다음은 왼쪽에서 오른쪽까지 6개의 각 열에 나타날 수 있는 값입니다.

1열

- s 시작 또는 중지
- - 자리 표시자

2열

- n 보통
- t 전환

3열

- d 지연된 재구성
- r 메모리 동적 재구성(DR)
- - 자리 표시자

4열

- c 제어 도메인
- - 자리 표시자

5열

- v 가상 I/O 서비스 도메인
- - 자리 표시자

6열

- s 마이그레이션에서 소스 도메인
- t 마이그레이션에서 대상 도메인
- e 마이그레이션 중 발생한 오류
- - 자리 표시자

사용률 통계 정의

가상 CPU당 사용률 통계(UTIL)가 `ldm list` 명령의 `긴(-l)` 옵션에 나타납니다. 통계는 가상 CPU가 게스트 운영 체제 대신 실행하는 데 걸린 시간의 백분율입니다. 가상 CPU는 하이퍼바이저에 양보한 경우를 제외하고 게스트 운영 체제 대신 실행하는 것으로 간주됩니다. 게스트 운영 체제가 가상 CPU를 하이퍼바이저에 양보하지 않을 경우 게스트 운영 체제에서 CPU 사용률은 항상 **100%**로 표시됩니다.

논리적 도메인에 대해 보고되는 사용률 통계는 도메인의 가상 CPU에 대한 가상 CPU 사용률의 평균입니다. UTIL 열의 대시(...)는 스레드가 전원 관리됨을 의미합니다.

다양한 목록 보기

▼ 소프트웨어 버전을 표시하는 방법(-V)

- 설치된 현재 소프트웨어 버전을 봅니다.

```
# ldm -V
```

▼ 짧은 목록을 생성하는 방법

- 모든도메인에 대해 짧은 목록을 생성합니다.

```
# ldm list
```

▼ 긴 목록을 생성하는 방법(-l)

- 모든도메인에 대해 긴 목록을 생성합니다.

```
# ldm list -l
```

▼ 확장된 목록을 생성하는 방법(-e)

- 모든도메인의 확장된 목록을 생성합니다.

```
# ldm list -e
```

▼ 구문 분석 가능하고 시스템에서 읽을 수 있는 목록을 생성하는 방법(-p)

- 모든도메인의 구문 분석 가능하고 시스템에서 읽을 수 있는 목록을 생성합니다.

```
# ldm list -p
```

▼ 긴 목록의 하위 세트를 생성하는 방법(-o *format*)

- 다음 *format* 옵션 중 하나 이상을 입력하여 출력을 리소스의 하위 세트로 생성합니다. 둘 이상의 형식을 지정할 경우 공백 없이 콤마로 항목을 구분하십시오.

```
# ldm list -o resource[,resource...] ldom
```

- console - 출력에 가상 콘솔(vcons) 및 가상 콘솔 집중기(vcc) 서비스가 포함됩니다.
- core - 출력에 전체 코어가 할당된 도메인에 대한 정보가 포함됩니다.
- cpu - 출력에 가상 CPU(vcpu), 물리적 CPU(pcpu) 및 코어 ID에 대한 정보가 포함됩니다.
- crypto - 암호화 장치 출력에 모듈식 산술 장치(mau) 및 CWQ(Control Word Queue)와 같은 기타 지원되는 암호화 장치가 포함됩니다.
- disk - 출력에 가상 디스크(vdisk) 및 가상 디스크 서버(vds)가 포함됩니다.
- domain - 출력에 변수(var), 호스트 ID(hostid), 도메인 상태, 플래그, UUID 및 소프트웨어 상태가 포함됩니다.
- memory - 출력에 memory가 포함됩니다.
- network - 출력에 매체 액세스 제어(mac) 주소, 가상 네트워크 스위치(vsw) 및 가상 네트워크(vnet) 장치가 포함됩니다.

- **physio** - 물리적 입/출력에 주변 장치 구성 요소 상호 연결(pci) 및 네트워크 인터페이스 장치(niu)가 포함됩니다.
- **resmgmt** - 출력에 동적 리소스 관리(DRM) 정책 정보가 포함되고, 현재 실행 중인 정책이 나타나며, 전체 코어 구성과 관련된 제약 조건이 나열됩니다.
- **serial** - 출력에 가상 논리적 도메인 채널(vldc) 서비스, 가상 논리적 도메인 채널 클라이언트(vldcc), 가상 데이터 플레인 채널 클라이언트(vdpcc), 가상 데이터 플레인 채널 서비스(vdpcs)가 포함됩니다.
- **stats** - 출력에 리소스 관리 정책과 관련된 통계가 포함됩니다.
- **status** - 출력에 진행 중인 도메인 마이그레이션에 대한 상태가 포함됩니다.

다음 예에서는 지정할 수 있는 출력의 다양한 하위 세트를 보여줍니다.

- 제어 도메인에 대한 CPU 정보 나열


```
# ldm list -o cpu primary
```
- 게스트 도메인에 대한 도메인 정보 나열


```
# ldm list -o domain ldm2
```
- 게스트 도메인에 대한 메모리 및 네트워크 정보 나열


```
# ldm list -o network,memory ldm1
```
- 게스트 도메인에 대한 DRM 정책 정보 나열


```
# ldm list -o resmgmt,stats ldm1
```

▼ 변수를 나열하는 방법

- 도메인에 대한 변수 및 해당 값을 표시합니다.

```
# ldm list-variable variable-name ldom
```

예를 들어, 다음 명령은 ldg1 도메인에서 boot-device 변수에 대한 값을 표시합니다.

```
# ldm list-variable boot-device ldg1
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

▼ 바인딩을 나열하는 방법

- 도메인에 바인드된 리소스를 나열합니다.

```
# ldm list-bindings ldom
```

▼ 구성을 나열하는 방법

- SP에 저장된 논리적 도메인 구성을 나열합니다.

예 10-12 구성 목록

`ldm list-config` 명령은 서비스 프로세서에 저장된 논리적 도메인 구성을 나열합니다. `-r` 옵션과 함께 사용하면 이 명령은 제어 도메인에 자동 저장 파일이 존재하는 구성을 나열합니다.

구성에 대한 자세한 내용은 [234 페이지 “Logical Domains 구성 관리”](#)를 참조하십시오. 더 많은 예는 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
# ldm list-config
factory-default
3guests
foo [next poweron]
primary
reconfig-primary
```

자세한 정보 레이블의 의미

구성 이름 오른쪽에 있는 레이블은 다음을 의미합니다.

- [current] - 마지막으로 부트된 구성(현재 실행 중인 구성과 일치하는 경우만 즉, 재구성을 시작할 때까지)입니다. 재구성 이후 주석은 [next poweron]으로 변경됩니다.
- [next poweron] - 다음 전원 사이클에서 사용될 구성입니다.

▼ 장치를 나열하는 방법

- 바인드 및 바인드 해제된 모든 서버 리소스를 나열합니다.

```
# ldm list-devices -a
```

▼ 사용 가능한 메모리를 나열하는 방법

- 할당할 수 있는 메모리 용량을 나열합니다.

```
# ldm list-devices mem
MEMORY
  PA                SIZE
  0x14e000000       2848M
```

▼ 서비스를 나열하는 방법

- 사용 가능한 서비스를 나열합니다.

```
# ldm list-services
```

제약 조건 나열

Logical Domains Manager에서 제약 조건은 특정 도메인에 지정하려는 하나 이상의 리소스입니다. 사용 가능한 리소스에 따라, 도메인에 추가하도록 요청한 리소스를 모두 받거나 아무 것도 얻지 못합니다. `list-constraints` 하위 명령은 도메인에 지정되도록 요청한 리소스를 나열합니다.

▼ 한 도메인에 대한 제약 조건을 나열하는 방법

- 한 도메인에 대한 제약 조건을 나열합니다.

```
# ldm list-constraints ldom
```

▼ XML 형식으로 제약 조건을 나열하는 방법

- 특정 도메인에 대해 XML 형식으로 제약 조건을 나열합니다.

```
# ldm list-constraints -x ldom
```

▼ 시스템에서 읽을 수 있는 형식으로 제약 조건을 나열하는 방법

- 구문 분석 가능한 형식으로 모든 도메인에 대한 제약 조건을 나열합니다.

```
# ldm list-constraints -p
```


도메인 구성 관리

이 장에서는 도메인 구성 관리에 대한 정보를 제공합니다.

이 장에서는 다음 주제를 다룹니다.

- 231 페이지 “나중에 재작성을 위해 도메인 구성 저장”
- 234 페이지 “Logical Domains 구성 관리”

나중에 재작성을 위해 도메인 구성 저장

기본 프로세스는 각 도메인에 대한 리소스 제약 조건 정보를 XML 파일에 저장하는 것입니다. 그런 다음 하드웨어 고장 후 Logical Domains Manager에 재실행하면 원하는 구성을 재작성할 수 있습니다.

232 페이지 “XML 파일에서 도메인 구성을 복원하는 방법(`ldm add-domain`)”은 게스트 도메인에 작동하지만 컨트롤(primary) 도메인에는 작동하지 않습니다. primary 도메인의 제약 조건은 XML 파일에 저장할 수 있지만 파일을 다시 `ldm add-domain -i` 명령으로 공급할 수 없습니다. 그러나 `ldm init-system` 명령과 XML 파일의 리소스 제약 조건을 사용하여 primary 도메인을 재구성할 수 있습니다. 또한 `ldm init-system` 명령을 사용하여 XML 파일에 설명된 다른 도메인을 재구성할 수 있습니다. 그러나 이러한 도메인은 구성이 완료될 때 비활성으로 남습니다.

명명된 물리적 리소스를 제외하고, 다음 방법은 실제 바인딩을 보존하지 않습니다. 그러나 해당 바인딩을 만드는 데 사용된 제약 조건은 보존합니다. 따라서 다음 절차를 완료한 후 도메인은 동일한 가상 리소스를 보유하지만 반드시 동일한 물리적 리소스에 바인딩되는 것은 아닙니다. 명명된 물리적 리소스는 관리자가 지정한 대로 바인딩됩니다.

▼ 도메인 구성을 저장하는 방법

이 절차는 단일 도메인 또는 시스템의 모든 도메인에 대한 도메인 구성을 저장하는 방법을 보여줍니다.

● 하나 이상의 도메인에 대한 도메인 구성을 저장합니다.

- 단일 도메인에 대한 구성을 저장하려면 도메인의 제약조건을 포함하는 XML 파일을 만듭니다.

```
# ldm list-constraints -x ldom >ldom.xml
```

다음 예제는 ldg1 도메인의 제약 조건을 포함하는 XML 파일 ldg1.xml을 만드는 방법을 보여줍니다.

```
# ldm list-constraints -x ldg1 >ldg1.xml
```

- 시스템의 모든 도메인에 대한 구성을 저장하려면 모든 도메인의 제약조건을 포함하는 XML 파일을 만듭니다.

```
# ldm list-constraints -x >file.xml
```

다음 예제는 시스템의 모든 도메인에 대한 제약 조건을 포함하는 XML 파일 config.xml을 만드는 방법을 보여줍니다.

```
# ldm list-constraints -x >config.xml
```

▼ XML 파일에서 도메인 구성을 복원하는 방법(ldm add-domain)

이 절차 대신, ldm init-system 명령을 사용하여 XML 파일에서 도메인 구성을 복원할 수 있습니다. [233 페이지](#) “XML 파일에서 도메인 구성을 복원하는 방법(ldm init-system)”을 참조하십시오.

1 입력으로 만든 XML 파일을 사용하여 도메인을 만듭니다.

```
# ldm add-domain -i ldom.xml
```

2 도메인을 바인딩합니다.

```
# ldm bind-domain [-fq] ldom
```

-f 옵션은 잘못된 백엔드 장치가 감지된 경우에도 강제로 도메인을 바인딩합니다. -q 옵션은 명령이 더 빠르게 실행되도록 백엔드 장치의 검증을 사용 안함으로 설정합니다.

3 도메인을 시작합니다.

```
# ldm start-domain ldom
```

예 11-1 XML 파일에서 단일 도메인 복원

다음 예제는 단일 도메인을 복원하는 방법을 보여줍니다. 먼저, XML 파일에서 `ldg1` 도메인을 복원합니다. 그 다음, 복원한 `ldg1` 도메인을 마인팅하고 다시 시작합니다.

```
# ldm add-domain -i ldg1.xml
# ldm bind ldg1
# ldm start ldg1
```

▼ XML 파일에서 도메인 구성을 복원하는 방법(ldm init-system)

이 절차는 `ldm init-system` 명령을 XML 파일과 함께 사용하여 이전에 저장된 구성을 다시 만드는 방법을 설명합니다. XML 파일은 하나 이상의 도메인 구성을 설명합니다. `ldm ls-constraints -x` 명령을 실행하여 XML 파일을 만들 수 있습니다. `ldm init-system` 명령은 `factory-default` 구성에서 실행되어야 하지만 XML 파일에서 어떤 구성도 복원할 수 있습니다. `primary` 도메인은 파일에 지정된 대로 재구성되고, XML 파일에 구성이 포함된 비-`primary` 도메인은 재구성되지만 비활성으로 남습니다.

이 절차 대신, `ldm add-domain` 명령을 사용하여 XML 파일에서 단일 도메인 구성을 복원할 수 있습니다. [232 페이지 “XML 파일에서 도메인 구성을 복원하는 방법\(ldm add-domain\)”](#)을 참조하십시오.

1 primary 도메인에 로그인합니다.

2 시스템이 factory-default 구성인지 확인합니다.

```
primary# ldm list-config | grep "factory-default"
factory-default [current]
```

시스템이 `factory-default` 구성이 아닌 경우 [41 페이지 “출하시 기본 구성을 복원하는 방법”](#)을 참조하십시오.

3 관리자, 수퍼유저 또는 동등한 역할의 사용자로 로그인합니다.

Oracle Solaris 10의 경우, [System Administration Guide: Security Services](#)의 “Configuring RBAC (Task Map)”를 참조하십시오. Oracle Solaris 11의 경우, [Oracle Solaris 관리: 보안 서비스의 제III부, “역할, 권한 프로파일 및 권한”](#)를 참조하십시오.

4 XML 파일에서 도메인 구성을 복원합니다.

```
# ldm init-system [-frs] -i filename.xml
```

구성을 적용하려면 `primary` 도메인을 반드시 재부트해야 합니다. `-r` 옵션은 구성 후 `primary` 도메인을 재부트합니다. `-r` 옵션을 지정하지 않으면 반드시 수동으로 재부트를 수행해야 합니다.

`-s` 옵션은 가상 서비스 구성(`vds`, `vcc`, `vsw`)만 복원하고 재부트 없이 수행할 수 있습니다.

-f 옵션은 factory-default 구성 검사를 건너뛰고 시스템에 구성된 사항에 관계없이 작업을 계속합니다. -f 옵션은 주의해서 사용하십시오. `ldm init-system` 명령은 시스템이 factory-default 구성이라고 가정하므로 XML 파일에 지정된 변경 사항이 직접 적용됩니다. 시스템이 factory-default 이외의 구성일 때 -f 옵션을 사용하면 XML 파일에 지정된 대로 시스템이 구성되지 않습니다. XML 파일과 초기 구성의 변경 사항 조합에 따라 하나 이상의 변경 사항이 시스템에 적용되지 못할 수 있습니다.

예 11-2 XML 구성 파일에서 도메인 복원

다음 예제는 `ldm init-system` 명령을 사용하여 factory-default 구성에서 primary 도메인 및 시스템의 모든 구성을 복원하는 방법을 보여줍니다.

- **primary 도메인 복원.** -r 옵션을 사용하여 구성이 완료된 후 primary 도메인을 재부트합니다. primary.xml 파일은 이전에 저장한 XML 도메인 구성을 포함합니다.

```
primary# ldm init-system -r -i primary.xml
```

- **시스템의 모든 도메인 복원.** 시스템의 도메인을 config.xml XML 파일의 구성으로 복원합니다. config.xml 파일은 이전에 저장한 XML 도메인 구성을 포함합니다. primary 도메인은 `ldm init-system` 명령에 의해 자동으로 다시 시작됩니다. 다른 도메인은 복원되지만 바인딩 후 다시 시작되지 않습니다.

```
# ldm init-system -r -i config.xml
```

시스템을 재부트한 후 다음 명령이 ldg1 및 ldg2 도메인을 바인딩하고 다시 시작합니다.

```
# ldm bind ldg1
# ldm start ldg1
# ldm bind ldg2
# ldm start ldg2
```

Logical Domains 구성 관리

Logical Domains 구성은 단일 시스템 내의 모든 도메인과 해당 리소스 지정으로 완전히 설명됩니다. 나중에 사용할 수 있도록 서비스 프로세서(SP)에 구성을 저장할 수 있습니다.

시스템 전원을 켤 때 SP가 선택한 구성을 부트합니다. 구성을 부트하면 시스템이 구성에 지정된 것과 동일한 도메인 세트를 실행하고 동일한 가상화 및 분할 리소스 지정을 사용합니다. 기본 구성은 가장 최근에 저장된 구성입니다.

Logical Domains 구성을 변경할 때마다 현재 구성의 복사본이 컨트롤 도메인에 자동으로 저장됩니다.

다음 경우에도 자동 저장 작업이 즉시 발생합니다.

- 새 구성이 SP에 명시적으로 저장되지 않은 경우
- 영향을 받는 도메인이 재부트될 때까지 실제 구성이 변경되지 않은 경우

이 자동 저장 작업은 SP에 저장된 구성이 손실된 경우 구성을 복구할 수 있습니다. 또한 이 작업은 시스템을 파워 사이클할 때 현재 구성이 SP에 명시적으로 저장되지 않은 경우 구성을 복구할 수 있습니다. 이러한 상황에서 Logical Domains Manager는 해당 구성이 다음 부트용으로 표시된 구성보다 최신인 경우 다시 시작 시 구성을 복원할 수 있습니다.

주 - 전원 관리, FMA, ASR, PRI 업데이트 이벤트로 자동 저장 파일이 업데이트되지는 않습니다.

자동 저장 파일을 새 구성이나 기존 구성으로 자동/수동으로 복원할 수 있습니다. 기본적으로 자동 저장 구성이 실행 중인 구성보다 최신인 경우 Logical Domains 로그에 메시지가 기록됩니다. 따라서 `ldm add-spconfig -r` 명령을 사용하여 기존 구성을 수동으로 업데이트하거나 자동 저장 데이터를 기반으로 새 구성을 만들어야 합니다.

주 - 지연된 재구성이 보류 중인 경우 구성 변경 사항이 즉시 자동 저장됩니다. 그 결과, `ldm list-config -r` 명령을 실행하면 자동 저장 구성이 현재 구성보다 최신인 것으로 나타납니다.

`ldm *-spconfig` 명령을 사용하여 구성을 관리하고 자동 저장 파일을 수동으로 복구하는 방법은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

부트할 구성을 선택하는 방법은 [242 페이지 “서비스 프로세서와 함께 Logical Domains 사용”](#)을 참조하십시오.

▼ 자동 복구 정책을 수정하는 방법

자동 복구 정책은 컨트롤 도메인에 자동으로 저장된 구성이 실행 중인 구성보다 최신인 경우 구성의 복구 처리 방법을 지정합니다. `ldmd` SMF 서비스의 `autorecovery_policy` 등록 정보를 설정하여 자동 복구 정책을 지정합니다. `autorecovery_policy` 등록 정보는 다음 값을 사용할 수 있습니다.

- `autorecovery_policy=1` - 자동 저장 구성이 실행 중인 구성보다 최신일 때 경고 메시지를 기록합니다. 이러한 메시지는 `ldmd` SMF 로그 파일에 기록됩니다. 사용자가 구성 복구를 수동으로 수행해야 합니다. 이것이 기본 정책입니다.
- `autorecovery_policy=2` - 자동 저장 구성이 실행 중인 구성보다 최신일 때 통지 메시지를 표시합니다. 이 통지 메시지는 Logical Domains Manager를 다시 시작한 후 처음 `ldm` 명령을 실행할 때 `ldm` 명령의 출력에 인쇄됩니다. 사용자가 구성 복구를 수동으로 수행해야 합니다.
- `autorecovery_policy=3` - 자동 저장 구성이 실행 중인 구성보다 최신일 때 구성을 자동으로 업데이트합니다. 이렇게 하면 다음 파워 사이클 중 사용할 SP 구성이 겹쳐쓰입니다. 이 구성은 컨트롤 도메인에 저장된 최신 구성으로 업데이트됩니다. 이 작업은 현재 실행 중인 구성에 영향을 주지 않습니다. 다음 파워 사이클 중 사용할

구성에만 영향을 줍니다. 또한 최신 구성이 SP에 저장되었으며 다음에 시스템을 파워 사이클할 때 부트될 것이라는 메시지가 기록됩니다. 이러한 메시지는 ldmd SMF 로그 파일에 기록됩니다.

1 컨트롤 도메인에 로그인합니다.

2 관리자, 수퍼유저 또는 동등한 역할의 사용자로 로그인합니다.

Oracle Solaris 10의 경우, **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오. Oracle Solaris 11의 경우, **Oracle Solaris 관리: 보안 서비스의 제III부**, “역할, 권한 프로파일 및 권한”를 참조하십시오.

3 autorecovery_policy 등록 정보 값을 확인합니다.

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy
```

4 ldmd 서비스를 중지합니다.

```
# svcadm disable ldmd
```

5 autorecovery_policy 등록 정보 값을 변경합니다.

```
# svccfg -s ldmd setprop ldmd/autorecovery_policy=value
```

예를 들어, 자동 복구를 수행할 정책을 설정하려면 등록 정보 값을 3으로 설정합니다.

```
# svccfg -s ldmd setprop ldmd/autorecovery_policy=3
```

6 ldmd 서비스를 새로 고치고 다시 시작합니다.

```
# svcadm refresh ldmd
# svcadm enable ldmd
```

예 11-3 자동 복구 정책을 로그에서 자동 복구로 수정

다음 예제는 autorecovery_policy 등록 정보의 현재 값을 확인하고 새 값으로 변경하는 방법을 보여줍니다. 이 등록 정보의 원래 값은 1이며, 자동 저장 변경 사항이 기록됨을 의미합니다. svcadm 명령을 사용하여 ldmd 서비스를 중지 후 다시 시작하고, svccfg 명령을 사용하여 등록 정보 값을 확인하고 설정할 수 있습니다.

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy
ldmd/autorecovery_policy integer 1
# svcadm disable ldmd
# svccfg -s ldmd setprop ldmd/autorecovery_policy=3
# svcadm refresh ldmd
# svcadm enable ldmd
```

기타 관리 작업 수행

이 장에서는 이전 장에서 설명되지 않은 Oracle VM Server for SPARC 소프트웨어 사용에 대한 정보 및 작업을 제공합니다.

이 장에서는 다음 주제를 다룹니다.

- 237 페이지 “CLI에서 이름 입력”
- 238 페이지 “네트워크를 통해 게스트 콘솔에 연결”
- 239 페이지 “콘솔 그룹 사용”
- 239 페이지 “부하가 높은 도메인을 중지할 때 시간 초과가 발생할 수 있음”
- 240 페이지 “Oracle VM Server for SPARC와 함께 Oracle Solaris OS 작동”
- 242 페이지 “서비스 프로세서와 함께 Logical Domains 사용”
- 242 페이지 “도메인 종속성 구성”
- 246 페이지 “CPU 및 메모리 주소를 매핑하여 오류 발생 위치 확인”
- 248 페이지 “UUID(Universally Unique Identifier) 사용”
- 249 페이지 “가상 도메인 정보 명령 및 API”

CLI에서 이름 입력

다음 절에서는 Logical Domains Manager CLI에서 이름을 입력하는 것과 관련된 제한 사항에 대해 설명합니다.

파일 이름(*file*) 및 변수 이름(*var-name*)

- 첫번째 문자는 문자, 숫자 또는 슬래시(/)여야 합니다.
- 후속 문자는 문자, 숫자 또는 문장 부호여야 합니다.

가상 디스크 서버 *backend* 및 가상 스위치 장치 이름

이 이름에는 문자, 숫자 또는 문장 부호가 들어 있어야 합니다.

구성 이름(config-name)

SP(서비스 프로세서)에서 저장된 구성에 지정하는 논리적 도메인 구성 이름(config-name)은 64자를 초과하지 않아야 합니다.

기타 모든 이름

논리적 도메인 이름(*ldom*), 서비스 이름(*vswitch-name*, *service-name*, *vdpcs-service-name* 및 *vcc-name*), 가상 네트워크 이름(*if-name*), 가상 디스크 이름(*disk-name*) 등 기타 모든 이름은 다음 형식이어야 합니다.

- 첫번째 문자는 문자 또는 숫자여야 합니다.
- 후속 문자는 문자, 숫자 또는 `-_#.:;~()` 문자여야 합니다.

네트워크를 통해 게스트 콘솔에 연결

`listen_addr` 등록 정보가 `vntsd(1M)` SMF 매니페스트에서 컨트롤 도메인의 IP 주소로 설정된 경우 네트워크를 통해 게스트 콘솔에 연결할 수 있습니다. 예를 들어, 다음과 같습니다.

```
$ telnet host-name 5001
```

주 - 콘솔에 대한 네트워크 액세스를 사용으로 설정하면 보안이 영향을 받을 수 있습니다. 모든 사용자가 콘솔에 연결할 수 있으므로 기본적으로 사용 안함으로 설정되어 있습니다.

서비스 관리 기능 매니페스트는 서비스를 기술하는 XML 파일입니다. SMF 매니페스트 만들기에 대한 자세한 내용은 [Oracle Solaris 10 System Administrator Documentation](http://download.oracle.com/docs/cd/E18752_01/index.html) (http://download.oracle.com/docs/cd/E18752_01/index.html)을 참조하십시오.

주 - 콘솔을 통해 게스트 도메인에서 영어 이외의 OS에 액세스하려면 콘솔 터미널이 OS에 필요한 로케일이어야 합니다.

콘솔 그룹 사용

가상 네트워크 터미널 서버 데몬 `vntsd`를 사용하면 단일 TCP 포트를 통해 다중 도메인 콘솔에 대한 액세스를 제공할 수 있습니다. 도메인을 만들 때 Logical Domains Manager는 해당 도메인의 콘솔에 대해 새 기본 그룹을 만들어 각 콘솔에 고유한 TCP 포트를 지정합니다. 그러면 콘솔 자체와 반대로 콘솔 그룹에 TCP 포트가 지정됩니다. `set-vcons` 하위 명령을 사용하여 기존 그룹에 콘솔을 바인딩할 수 있습니다.

▼ 하나의 그룹에 다중 콘솔을 결합하는 방법

- 1 하나의 그룹에 다중 도메인에 대한 콘솔을 바인딩합니다.

다음 예에서는 동일한 콘솔 그룹(`group1`)에 3개의 다른 도메인(`ldg1`, `ldg2` 및 `ldg3`)에 대한 콘솔을 바인딩하는 방법을 보여줍니다.

```
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg1
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg2
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg3
```

- 2 연관된 TCP 포트(이 예의 경우 포트 5000의 `localhost`)에 연결합니다.

```
# telnet localhost 5000
primary-vnts-group1: h, l, c{id}, n{name}, q:
도메인 콘솔 중 하나를 선택하라는 메시지가 표시됩니다.
```

- 3 `l(list)`을 선택하여 그룹 내 도메인을 나열합니다.

```
primary-vnts-group1: h, l, c{id}, n{name}, q: l
DOMAIN ID          DOMAIN NAME          DOMAIN STATE
0                   ldg1                 online
1                   ldg2                 online
2                   ldg3                 online
```

주 - 다른 그룹 또는 `vcc` 인스턴스에 콘솔을 재지정하려면 도메인의 바인딩을 해제해야 합니다. 즉, 도메인이 비활성 상태여야 합니다. SMF를 구성 및 사용하여 `vntsd`를 관리하고 콘솔 그룹을 사용하는 것과 관련된 자세한 내용은 Oracle Solaris 10 OS `vntsd(1M)` 매뉴얼 페이지를 참조하십시오.

부하가 높은 도메인을 중지할 때 시간 초과가 발생할 수 있음

도메인 종료가 완료되기 전에 `ldm stop-domain` 명령 시간 초과가 발생할 수 있습니다. 이 문제가 발생하면 Logical Domains Manager가 다음과 유사한 오류를 반환합니다.

```
LDom ldg8 stop notification failed
```

하지만 이 경우에도 도메인 종료 요청은 계속 처리될 수 있습니다. 도메인 상태를 확인하려면 `ldm list-domain` 명령을 사용하십시오. 예를 들어, 다음과 같습니다.

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg8          active s---- 5000   22    3328M  0.3% 1d 14h 31m
```

위 목록에서는 도메인이 활성 상태임을 보여 주지만 `s` 플래그를 통해 도메인 중지가 진행 중임을 알 수 있습니다. 이는 일시적인 상태입니다.

다음 예에서는 도메인이 이제 중지되었음을 보여줍니다.

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg8          bound  ----- 5000   22    3328M
```

Oracle VM Server for SPARC와 함께 Oracle Solaris OS 작동

이 절에서는 Logical Domains Manager가 만든 구성이 인스턴스화된 후 Oracle Solaris OS를 사용할 때 변경되는 동작에 대해 설명합니다.

Oracle Solaris OS가 시작된 후 OpenBoot 펌웨어를 사용할 수 없음

Oracle Solaris OS가 시작된 후에는 OpenBoot 펌웨어를 사용할 수 없습니다. 메모리에서 제거되었기 때문입니다.

Oracle Solaris OS에서 `ok` 프롬프트에 도달하려면 도메인을 정지해야 합니다. Oracle Solaris OS `halt` 명령을 사용하여 도메인을 정지할 수 있습니다.

서버 전원 켜다 켜기

Oracle VM Server for SPARC 소프트웨어를 실행하는 시스템에서 유지 관리를 수행할 때마다 서버 전원을 켜다 켜야 하므로 먼저 SP에 현재 논리적 도메인 구성을 저장해야 합니다.

▼ SP에 현재 도메인 구성을 저장하는 방법

- 다음 명령을 사용합니다.

```
# ldm add-config config-name
```

전원 관리 도메인에서 활성 CPU에 대해 psradm(1M) 명령을 사용하지 않아야 함

전원 관리 도메인에서 psradm 명령을 사용하여 활성 CPU의 작동 상태를 변경하려고 하지 마십시오.

Oracle Solaris OS 중단 결과

이 절에서 설명되는 동작은 다음 작업을 수행할 때 발생합니다.

1. 입력 장치가 keyboard로 설정된 경우 L1-A 키 시퀀스를 누릅니다.
2. 가상 콘솔이 telnet 프롬프트에 있을 때 send break 명령을 입력합니다.

이러한 유형의 중단이 발생하면 다음 프롬프트가 표시됩니다.

```
c)ontinue, s)ync, r)eset, h)alt?
```

이러한 유형의 중단이 발생한 후 시스템에서 수행할 작업을 나타내는 문자를 입력합니다.

컨트롤 도메인 정지 또는 재부트 결과

다음 표에서는 컨트롤(primary) 도메인 정지 또는 재부트로 인한 예상 동작을 보여줍니다.

표 12-1 컨트롤(primary) 도메인 정지 또는 재부트로 인한 예상 동작

명령	기타도메인 구성 여부	동작
halt	구성되지 않음	호스트 전원을 끄고 SP에서 다시 전원이 켜질 때까지 꺼진 상태로 유지합니다.
	구성됨	변수가 auto-boot?=true인 경우 소프트웨어 재설정 후 부트합니다. 변수가 auto-boot?=false인 경우 소프트웨어 재설정 후 ok 프롬프트에서 정지합니다.
reboot	구성되지 않음	호스트를 재부트하고 전원을 끄지 않습니다.
	구성됨	호스트를 재부트하고 전원을 끄지 않습니다.
shutdown -i 5	구성되지 않음	호스트 전원을 끄고 SP에서 다시 전원이 켜질 때까지 꺼진 상태로 유지합니다.
	구성됨	소프트웨어 재설정 후 재부트합니다.

루트 도메인 역할이 있는 컨트롤 도메인의 재부트 결과에 대한 자세한 내용은 [83 페이지](#) “primary 도메인 재부트”를 참조하십시오.

서비스 프로세서와 함께 Logical Domains 사용

이 절에서는 Logical Domains Manager와 함께 ILOM(Integrated Lights Out Manager) SP(서비스 프로세서)를 사용할 때 알아야 할 정보에 대해 설명합니다. ILOM 소프트웨어 사용에 대한 자세한 내용은 사용 중인 특정 플랫폼에 대한 문서(예:

<http://www.oracle.com/>

[technetwork/documentation/sparc-tseries-servers-252697.html](http://www.oracle.com/technetwork/documentation/sparc-tseries-servers-252697.html))를 참조하십시오.

기존 ILOM 명령에 추가 옵션을 사용할 수 있습니다.

```
-> set /HOST/bootmode config=config-name
```

config=config-name 옵션을 통해 다음 번 전원 켜기에 대한 구성을 factory-default 출하시 구성을 비롯한 다른 구성으로 설정할 수 있습니다.

호스트 전원을 켜거나 끌 때 명령을 호출할 수 있습니다. 그러면 다음 번 호스트 재설정 또는 전원 켜기에 적용됩니다.

▼ 도메인 구성을 기본 또는 다른 구성으로 재설정하는 방법

- 이 명령을 실행하여 다음 번 전원 켜기에 대한 논리적 도메인 구성을 기본 출하시 구성으로 재설정합니다.

```
-> set /HOST/bootmode config=factory-default
```

ldm add-config 명령을 사용하여 Logical Domains Manager로 만들어진 후 SP(서비스 프로세서)에 저장된 다른 구성을 선택할 수도 있습니다. Logical Domains Manager ldm add-config 명령에서 지정한 이름을 사용하여 ILOM bootmode 명령을 통해 해당 구성을 선택할 수 있습니다. 예를 들어, 이름이 ldm-config1인 구성을 저장했다고 가정합니다.

```
-> set /HOST/bootmode config=ldm-config1
```

그러면 새 구성이 로드되도록 시스템 전원을 껐다 켜야 합니다.

ldm add-config 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

도메인 종속성 구성

Logical Domains Manager를 사용하여 도메인 간의 종속성 관계를 설정할 수 있습니다. 종속된 하나 이상의 도메인을 가진 도메인을 **마스터 도메인**이라고 합니다. 다른 도메인에 종속된 도메인을 **슬레이브 도메인**이라고 합니다.

master 등록 정보를 설정하여 슬레이브 도메인마다 최대 4개의 마스터 도메인을 지정할 수 있습니다. 예를 들어, pine 슬레이브 도메인은 다음과 같이 쉼표로 구분된 목록에서 4개의 마스터 도메인을 지정합니다.

```
# ldm add-domain master=apple,lemon,orange,peach pine
```

각 마스터 도메인은 마스터 도메인 실패 시 슬레이브 도메인에 발생하는 동작을 지정할 수 있습니다. 예를 들어, 마스터 도메인이 실패할 경우 슬레이브 도메인에 패닉이 발생하도록 할 수 있습니다. 슬레이브 도메인에 2개 이상의 마스터 도메인이 있을 경우 실패하는 첫번째 마스터 도메인에 따라 모든 슬레이브 도메인에 대해 정의된 실패 정책이 트리거됩니다.

주-2개 이상의 마스터 도메인이 동시에 실패할 경우 영향을 받는 모든 슬레이브 도메인에 대해 지정된 실패 정책 중 하나만 적용됩니다. 예를 들어, 실패한 마스터 도메인의 실패 정책이 stop 및 panic일 경우 모든 슬레이브 도메인이 중지되거나 패닉이 발생합니다.

마스터 도메인의 실패 정책을 제어하려면 `failure-policy` 등록 정보를 다음 값 중 하나로 설정하십시오.

- ignore로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인이 무시됩니다.
- panic으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인에 패닉이 발생합니다.
- reset으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인이 재설정됩니다.
- stop으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인이 중지됩니다.

이 예에서 마스터 도메인은 다음과 같이 실패 정책을 지정합니다.

```
# ldm set-domain failure-policy=ignore apple
# ldm set-domain failure-policy=panic lemon
# ldm set-domain failure-policy=reset orange
# ldm set-domain failure-policy=stop peach
```

이 방식을 사용하여 도메인 간의 명시적 종속성을 만들 수 있습니다. 예를 들어, 게스트 도메인은 암시적으로 서비스 도메인에 종속되어 가상 장치를 제공합니다. 게스트 도메인이 종속된 서비스 도메인이 작동 및 실행 중이 아닐 경우 게스트 도메인의 I/O가 차단됩니다. 게스트 도메인을 서비스 도메인의 슬레이브로 정의하여 서비스 도메인의 작동이 중지될 때 게스트 도메인에 발생하는 동작을 지정할 수 있습니다. 종속성이 설정되지 않은 경우 게스트 도메인은 서비스 도메인이 다시 작동할 때까지 기다립니다.

주-Logical Domains Manager에서는 종속성 주기를 만드는 도메인 관계를 만들 수 없습니다. 자세한 내용은 245 페이지 “종속성 주기”를 참조하십시오.

도메인 종속성 XML 예는 예 17-6을 참조하십시오.

도메인 종속성 예

다음 예에서는 도메인 종속성을 구성하는 방법을 보여줍니다.

- 첫번째 명령은 **twizzle**이라는 마스터 도메인을 만듭니다. 이 명령은 **failure-policy=reset**을 사용하여 **twizzle** 도메인이 실패할 경우 슬레이브 도메인이 재설정되도록 지정합니다. 두번째 명령은 **primary**라는 마스터 도메인을 수정합니다. 이 명령은 **failure-policy=panic**을 사용하여 **primary** 도메인이 실패할 경우 슬레이브 도메인에 패닉이 발생하도록 지정합니다. 세번째 명령은 2개의 마스터 도메인(**twizzle** 및 **primary**)에 종속된 **chocktaw**라는 슬레이브 도메인을 만듭니다. 슬레이브 도메인은 **master=twizzle,primary**를 사용하여 마스터 도메인을 지정합니다. **twizzle** 또는 **primary** 도메인이 실패할 경우 **chocktaw** 도메인이 재설정되거나 패닉이 발생합니다. 실패하는 첫번째 마스터 도메인에 따라 슬레이브 도메인의 동작이 결정됩니다.

```
# ldm add-domain failure-policy=reset twizzle
# ldm set-domain failure-policy=panic primary
# ldm add-domain master=twizzle,primary chocktaw
```

- 이 예에서는 **ldm set-domain** 명령을 통해 **orange** 도메인을 수정하여 **primary**를 마스터 도메인으로 지정하는 방법을 보여줍니다. 두번째 명령은 **ldm set-domain** 명령을 사용하여 **orange** 및 **primary**를 **tangerine** 도메인의 마스터 도메인으로 지정합니다. 세번째 명령은 이러한 모든 도메인에 대한 정보를 나열합니다.

```
# ldm set-domain master=primary orange
# ldm set-domain master=orange,primary tangerine
# ldm list -o domain
NAME          STATE      FLAGS    UTIL
primary       active    -n-cv-   0.2%
```

SOFTSTATE
Solaris running

HOSTID
0x83d8b31c

CONTROL
failure-policy=ignore

DEPENDENCY
master=

NAME STATE FLAGS UTIL
orange bound -----

HOSTID
0x84fb28ef

CONTROL
failure-policy=stop

DEPENDENCY
master=primary

```
-----
NAME          STATE      FLAGS    UTIL
tangerine     bound     -----
```

```
HOSTID
0x84f948e9
```

```
CONTROL
failure-policy=ignore
```

```
DEPENDENCY
master=orange,primary
```

- 다음에서는 구문 분석이 가능한 출력이 포함된 목록 예를 보여줍니다.

```
# ldm list -o domain -p
```

종속성 주기

Logical Domains Manager에서는 종속성 주기를 만드는 도메인 관계를 만들 수 없습니다. 종속성 주기는 슬레이브 도메인이 자체적으로 종속되거나 마스터 도메인이 슬레이브 도메인 중 하나에 종속되도록 하는 2개 이상의 도메인 간 관계입니다.

Logical Domains Manager는 종속성을 추가하기 전에 종속성 주기가 존재하는지 여부를 확인합니다. Logical Domains Manager는 슬레이브 도메인부터 시작하여 경로 끝에 도달할 때까지 마스터 배열에 따라 지정된 모든 경로를 검색합니다. 해당 경로에서 발견된 종속성 주기는 오류로 보고됩니다.

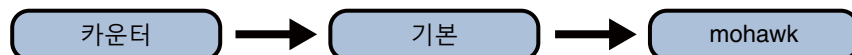
다음 예에서는 종속성 주기가 만들어질 수 있는 방법을 보여줍니다. 첫번째 명령은 마스터 도메인을 primary로 지정하는 mohawk라는 슬레이브 도메인을 만듭니다. 따라서 mohawk는 다음 종속성 체인에서 primary에 종속됩니다.

그림 12-1 단일 도메인 종속성



두번째 명령은 마스터 도메인을 counter로 지정하는 primary라는 슬레이브 도메인을 만듭니다. 따라서 mohawk는 다음 종속성 체인에서 counter에 종속되는 primary에 종속됩니다.

그림 12-2 다중 도메인 종속성



세번째 명령은 다음과 같은 종속성 주기가 생성되도록 counter 도메인과 mohawk 도메인 간에 종속성을 만들려고 시도합니다.

그림 12-3 도메인 종속성 주기



ldm set-domain 명령은 다음 오류 메시지와 함께 실패합니다.

```
# ldm add-domain master=primary mohawk
# ldm set-domain master=counter primary
# ldm set-domain master=mohawk counter
Dependency cycle detected: LDom "counter" indicates "primary" as its master
```

CPU 및 메모리 주소를 매핑하여 오류 발생 위치 확인

이 절에서는 Oracle Solaris FMA(Fault Management Architecture)가 보고하는 정보와 결함이 있는 것으로 표시된 논리적 도메인 리소스를 상관시키는 방법에 대해 설명합니다.

FMA는 물리적 CPU 번호와 관련하여 CPU 오류를 보고하고 물리적 메모리 주소와 관련하여 메모리 오류를 보고합니다.

논리적 도메인에서 오류가 발생한 위치와 도메인 내 해당 가상 CPU 번호 또는 실제 메모리 주소를 확인하려면 매핑을 수행해야 합니다.

CPU 매핑

지정된 물리적 CPU 번호에 해당하는 도메인 및 도메인 내 가상 CPU 번호는 다음 절차로 확인할 수 있습니다.

▼ CPU 번호를 확인하는 방법

- 1 모든도메인에 대해 구문 분석이 가능한 긴 목록을 생성합니다.
primary# ldm list -l -p
- 2 목록의 vCPU 섹션에서 pid 필드가 물리적 CPU 번호인 항목을 찾습니다.
 - 해당 항목이 발견되면 항목이 나열된 도메인에 CPU가 있으며 항목의 vid 필드에 도메인 내 가상 CPU 번호가 지정된 것입니다.

- 해당 항목이 발견되지 않으면 CPU가 도메인에 없는 것입니다.

메모리 매핑

지정된 PA(물리적 메모리 주소)에 해당하는 도메인 및 도메인 내 실제 메모리 주소는 다음과 같이 확인할 수 있습니다.

▼ 실제 메모리 주소를 확인하는 방법

- 1 모든 도메인에 대해 구문 분석이 가능한 긴 목록을 생성합니다.

```
primary# ldm list -l -p
```

- 2 목록의 MEMORY 섹션에서 PA가 포함 범위 pa 에서 $(pa + size - 1)$ 사이에 속한 행, 즉 $pa \leq PA \leq (pa + size - 1)$ 인 행을 찾습니다.

여기서 pa 및 $size$ 는 행의 해당 필드에 있는 값을 나타냅니다.

- 항목이 발견되면 항목이 나열된 도메인에 PA가 있으며 $ra + (PA - pa)$ 로 도메인 내 해당 실제 주소가 지정된 것입니다.
- 해당 항목이 발견되지 않으면 PA가 도메인에 없는 것입니다.

CPU 및 메모리 매핑 예

예 12-1과 같이 논리적 도메인 구성을 사용하며 물리적 CPU 번호 5에 해당하는 도메인 및 가상 CPU와 물리적 주소 0x7e816000에 해당하는 도메인 및 실제 주소를 확인하려고 한다고 가정합니다.

목록의 VCPU 항목에서 pid 필드가 5인 항목을 통해 논리적 도메인 ldg1에서 다음 항목을 찾을 수 있습니다.

```
|vid=1|pid=5|util=29|strand=100
```

따라서 물리적 CPU 번호 5는 도메인 ldg1에 있으며 도메인 내 가상 CPU 번호는 1임을 알 수 있습니다.

목록의 MEMORY 항목을 통해 도메인 ldg2에서 다음 항목을 찾을 수 있습니다.

```
ra=0x80000000|pa=0x78000000|size=1073741824
```

여기서는 $0x78000000 \leq 0x7e816000 \leq (0x78000000 + 1073741824 - 1)$, 즉 $pa \leq PA \leq (pa + size - 1)$ 입니다. 따라서 PA는 도메인 ldg2에 있으며 해당 실제 주소는 $0x80000000 + (0x7e816000 - 0x78000000) = 0xe816000$ 임을 알 수 있습니다.

예 12-1 구문 분석이 가능한 긴 Logical Domains 구성 목록

```
primary# ldm list -l -p
VERSION 1.6
DOMAIN|name=primary|state=active|flags=normal,control,vio-service|cons=SP|ncpu=4|mem=1073741824|util=0.6|
uptime=64801|softstate=Solaris running
VCPU
|vid=0|pid=0|util=0.9|strand=100
|vid=1|pid=1|util=0.5|strand=100
|vid=2|pid=2|util=0.6|strand=100
|vid=3|pid=3|util=0.6|strand=100
MEMORY
|ra=0x80000000|pa=0x80000000|size=1073741824
IO
|dev=pci@780|alias=bus_a
|dev=pci@7c0|alias=bus_b
...
DOMAIN|name=ldg1|state=active|flags=normal|cons=5000|ncpu=2|mem=805306368|util=29|uptime=903|
softstate=Solaris running
VCPU
|vid=0|pid=4|util=29|strand=100
|vid=1|pid=5|util=29|strand=100
MEMORY
|ra=0x80000000|pa=0x48000000|size=805306368
...
DOMAIN|name=ldg2|state=active|flags=normal|cons=5001|ncpu=3|mem=1073741824|util=35|uptime=775|
softstate=Solaris running
VCPU
|vid=0|pid=6|util=35|strand=100
|vid=1|pid=7|util=34|strand=100
|vid=2|pid=8|util=35|strand=100
MEMORY
|ra=0x80000000|pa=0x78000000|size=1073741824
...
```

UUID(Universally Unique Identifier) 사용

Oracle VM Server for SPARC 2.0 릴리스부터 각 도메인에는 UUID(Universally Unique Identifier)가 지정됩니다. UUID는 도메인이 만들어질 때 지정됩니다. 레거시 도메인의 경우 UUID는 ldmd 데몬이 초기화될 때 지정됩니다.

주 - ldm migrate-domain -f 명령을 사용하여 이전 버전의 Logical Domains Manager가 실행되는 대상 시스템으로 도메인을 마이그레이션할 경우 UUID가 손실됩니다. 이전 버전의 Logical Domains Manager가 실행되는 소스 시스템에서 도메인을 마이그레이션할 때 마이그레이션의 일부로 도메인에 새 UUID가 지정됩니다. 그렇지 않은 경우 UUID가 마이그레이션됩니다.

ldm list -l, ldm list-bindings 또는 ldm list -o domain 명령을 실행하여 도메인의 UUID를 확인할 수 있습니다. 다음 예에서는 ldg1 도메인의 UUID를 보여줍니다.

```
primary# ldm create ldg1
primary# ldm ls -l ldg1
```

```

NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
ldg1          inactive  -----
UUID
6c908858-12ef-e520-9eb3-f1cd3dbc3a59

primary# ldm ls -l -p ldg1
VERSION 1.6
DOMAIN|name=ldg1|state=inactive|flags=|cons=|ncpu=|mem=|util=|uptime=
UUID|uuid=6c908858-12ef-e520-9eb3-f1cd3dbc3a59

```

가상 도메인 정보 명령 및 API

virtinfo 명령을 사용하여 실행 중인 가상 도메인에 대한 정보를 수집할 수 있습니다. 가상 도메인 정보 API를 사용하여 가상 도메인 관련 정보를 수집할 프로그램을 만들 수도 있습니다.

다음 목록에서는 명령 또는 API를 사용하여 가상 도메인에 대해 수집할 수 있는 몇 가지 정보를 보여줍니다.

- 도메인 유형(구현, 컨트롤, 게스트, I/O, 서비스, 루트)
- Virtual Domain Manager로 확인된 도메인 이름
- 도메인의 UUID(Universally Unique Identifier)
- 도메인의 컨트롤 도메인에 대한 네트워크 노드 이름
- 도메인이 실행 중인 새시 일련 번호

virtinfo 명령에 대한 자세한 내용은 virtinfo(1M) 매뉴얼 페이지를 참조하십시오. API에 대한 자세한 내용은 libv12n(3LIB) 및 v12n(3EXT) 매뉴얼 페이지를 참조하십시오.

제 2 부

선택적 Oracle VM Server for SPARC 소프트웨어

Oracle VM Server for SPARC 2.2 소프트웨어와 함께 사용할 수 있는 선택적 소프트웨어 및 기능을 소개합니다.

Oracle VM Server for SPARC Physical-to-Virtual 변환 도구

이 장에서는 다음 주제를 다룹니다.

- 253 페이지 “Oracle VM Server for SPARC P2V 도구 개요”
- 256 페이지 “백엔드 장치”
- 257 페이지 “Oracle VM Server for SPARC P2V 도구 설치”
- 259 페이지 “ldmp2v 명령 사용”

Oracle VM Server for SPARC P2V 도구 개요

Oracle VM Server for SPARC P2V(Physical-to-Virtual) 변환 도구는 기존의 물리적 시스템을 CMT(칩 멀티스레딩) 시스템의 논리적 도메인에서 Oracle Solaris 10 OS를 실행하는 가상 시스템으로 자동으로 변환합니다. Oracle Solaris 10 OS 또는 Oracle Solaris 11 OS를 실행하는 컨트롤 도메인에서 `ldmp2v` 명령을 실행하여 다음 소스 시스템 중 하나를 논리적 도메인으로 변환할 수 있습니다.

- Solaris 8, Solaris 9, Oracle Solaris 10 OS를 실행하는 sun4u SPARC 기반 시스템
- Oracle Solaris 10 OS를 실행하지만 논리적 도메인에서 실행되지 않는 sun4v 시스템

주 - Oracle Solaris 11 물리적 시스템을 가상 시스템으로 변환하기 위해 P2V 도구를 사용할 수 없습니다.

다음 단계에 따라 물리적 시스템에서 가상 시스템으로의 변환이 수행됩니다.

- **수집 단계.** 물리적 소스 시스템에서 실행됩니다. **collect** 단계에서는 소스 시스템에 대해 수집된 구성 정보를 기반으로 소스 시스템의 파일 시스템 이미지가 만들어집니다.
- **준비 단계.** 대상 시스템의 컨트롤 도메인에서 실행됩니다. **prepare** 단계에서는 **collect** 단계에서 수집된 구성 정보를 기반으로 대상 시스템에 논리적 도메인이 만들어집니다. 파일 시스템 이미지가 하나 이상의 가상 디스크로 복원됩니다. P2V 도구를 사용하여 일반 파일 또는 ZFS 볼륨에 가상 디스크를 만들 수 있습니다. 물리적 디스크 또는 LUN이나 만들어 놓은 **Volume Manager** 볼륨에도 가상 디스크를 만들 수 있습니다. 논리적 도메인으로 실행되도록 사용으로 설정하면 이미지가 수정됩니다.
- **변환 단계.** 대상 시스템의 컨트롤 도메인에서 실행됩니다. **convert** 단계에서는 만들어진 논리적 도메인이 표준 Oracle Solaris 업그레이드 프로세스를 사용하여 Oracle Solaris 10 OS를 실행하는 논리적 도메인으로 변환됩니다.

P2V 도구에 대한 자세한 내용은 **ldmp2v(1M)** 매뉴얼 페이지를 참조하십시오.

다음 절에서는 단계에 따라 물리적 시스템에서 가상 시스템으로의 변환이 수행되는 방법에 대해 설명합니다.

수집 단계

수집 단계는 변환할 시스템에서 실행됩니다. 일관성 있는 파일 시스템 이미지를 만들려면 시스템의 작동이 최소화되고 모든 응용 프로그램이 중지되었는지 확인하십시오. **ldmp2v** 명령은 논리적 도메인으로 이동될 파일 시스템이 마운트되도록 마운트된 모든 UFS 파일 시스템의 백업을 만듭니다. 마운트된 파일 시스템 중 이동하지 않으려는 파일 시스템(예: SAN 저장소의 파일 시스템 또는 다른 방법으로 이동될 파일 시스템)을 제외할 수 있습니다. 해당 파일 시스템을 제외하려면 **-x** 옵션을 사용하십시오. **-x** 옵션으로 제외된 파일 시스템은 게스트 도메인에 다시 만들어지지 않습니다. **-o** 옵션을 사용하여 파일 및 디렉토리를 제외할 수 있습니다.

소스 시스템에서는 변경이 필요하지 않습니다. 단, **ldmp2v** 스크립트가 컨트롤 도메인에 설치되어 있어야 합니다. **flarcreate** 유틸리티가 소스 시스템에 있는지 확인하십시오.

준비 단계

준비 단계에서는 수집 단계 중 수집된 데이터를 사용하여 소스 시스템과 유사한 논리적 도메인을 만듭니다.

다음 방법 중 하나로 `ldmp2v prepare` 명령을 사용할 수 있습니다.

- **자동 모드.** 이 모드는 자동으로 가상 디스크를 만들고 파일 시스템 데이터를 복원합니다.
 - 논리적 도메인 및 소스 시스템에서와 동일한 크기로 필요한 가상 디스크를 만듭니다.
 - 디스크를 분할하고 파일 시스템을 복원합니다.
`/, /usr` 및 `/var` 파일 시스템의 크기 합계가 10GB 미만이면 보다 큰 디스크 공간이 필요한 Oracle Solaris 10 OS 요구 사항에 맞게 해당 파일 시스템의 크기가 자동으로 조정됩니다. `-x no-auto-adjust-fs` 옵션을 사용하여 자동 크기 조정을 사용 안함으로 설정하거나 `-m` 옵션을 사용하여 파일 시스템 크기를 수동으로 조정할 수 있습니다.
 - 물리적 하드웨어에 대한 모든 참조가 논리적 도메인에 적절한 버전으로 바뀌도록 논리적 도메인의 OS 이미지를 수정합니다. 그러면 일반적인 Oracle Solaris 업그레이드 프로세스를 사용하여 시스템을 Oracle Solaris 10 OS로 업그레이드할 수 있습니다. 수정 시 새 디스크 이름이 고려되도록 `/etc/vfstab` 파일이 업데이트됩니다. 이 프로세스가 수행되는 동안 Oracle Solaris Volume Manager 또는 VxVM(Veritas Volume Manager)이 캡슐화한 부트 디스크의 캡슐화가 자동으로 해제됩니다. 디스크의 캡슐화가 해제되면 디스크가 일반 디스크 슬라이스로 변환됩니다. 소스 시스템에 VxVM이 설치된 경우 P2V 프로세스는 만들어진 게스트 도메인에서 VxVM을 사용 안함으로 설정합니다.
- **비자동 모드.** 수동으로 가상 디스크를 만들고 파일 시스템 데이터를 복원해야 합니다. 이 모드에서는 디스크의 크기 및 수, 분할, 파일 시스템 레이아웃을 변경할 수 있습니다. 이 모드의 준비 단계는 파일 시스템에서 논리적 도메인 만들기 및 OS 이미지 수정 단계만 실행합니다.
- **정리 모드.** 논리적 도메인 및 `ldmp2v`로 만들어진 모든 기본 백엔드 장치를 제거합니다.

변환 단계

변환 단계에서 논리적 도메인은 Oracle Solaris 업그레이드 프로세스를 사용하여 Oracle Solaris 10 OS로 업그레이드합니다. 업그레이드 작업에서는 기존 패키지를 모두 제거하고 Oracle Solaris 10 sun4v 패키지를 설치하면서 sun4u에서 sun4v로의 변환을 자동으로 수행합니다. `convert` 단계는 Oracle Solaris DVD ISO 이미지 또는 네트워크 설치 이미지를 사용할 수 있습니다. Oracle Solaris 10 시스템에서는 Oracle Solaris JumpStart 기능을 사용하여 완전히 자동화된 업그레이드 작업을 수행할 수도 있습니다.

백엔드 장치

파일(file), ZFS 볼륨(zvol), 물리적 디스크 또는 LUN(disk), Volume Manager 볼륨(disk) 등 다양한 백엔드 유형에 게스트 도메인에 대한 가상 디스크를 만들 수 있습니다. 다음 방법 중 하나로 file 또는 zvol을 백엔드 유형으로 지정할 경우 ldmp2v 명령이 자동으로 적절한 크기의 파일 또는 ZFS 볼륨을 만듭니다.

- -b 옵션 사용
- /etc/ldmp2v.conf 파일에서 BACKEND_TYPE 매개변수의 값 지정

disk 백엔드 유형을 통해 물리적 디스크, LUN 또는 Volume Manager 볼륨(Oracle Solaris Volume Manager 및 VxVM(Veritas Volume Manager))을 가상 디스크용 백엔드 장치로 사용할 수 있습니다. prepare 단계를 시작하기 전에 적절한 크기로 디스크 또는 볼륨을 만들어야 합니다. 물리적 디스크나 LUN의 경우 디스크 블록 또는 문자 장치의 슬라이스 2로 백엔드 장치를 지정하십시오(예: /dev/dsk/c0t3d0s2). Volume Manager 볼륨의 경우 볼륨에 대한 블록 또는 문자 장치를 지정하십시오(예: Oracle Solaris Volume Manager의 경우 /dev/md/dsk/d100 또는 VxVM의 경우 /dev/vx/dsk/ldomdg/vol1).

-B backend:volume:vdisk 옵션으로 볼륨 및 가상 디스크 이름을 지정하지 않을 경우 게스트에 대해 만드는 볼륨 및 가상 디스크에 기본 이름이 지정됩니다.

- backend는 사용할 백엔드의 이름을 지정합니다. disk 백엔드 유형에 대해 backend를 지정해야 합니다. backend는 file 및 zvol 백엔드 유형에 대해 선택 사항이며 ldmp2v가 만든 파일 또는 ZFS 볼륨에 대해 기본 이름 이외의 이름을 설정하는 데 사용할 수 있습니다. 기본 이름은 \$BACKEND_PREFIX/guest-name/diskN입니다.
- volume은 모든 백엔드 유형에 대해 선택 사항이며 게스트 도메인에 대해 만들 가상 디스크 서버 볼륨의 이름을 지정합니다. 지정되지 않은 경우 volume은 guest-name-volN입니다.
- vdisk는 모든 백엔드 유형에 대해 선택 사항이며 게스트 도메인의 볼륨에 대한 이름을 지정합니다. 지정되지 않은 경우 vdisk는 diskN입니다.

주- 변환 프로세스를 수행하는 동안 컨트롤 도메인의 이름이 고유하도록 임시로 가상 디스크에 guest-name-diskN이라는 이름이 지정됩니다.

backend, volume 또는 vdisk에 대해 공백 값을 지정하려면 콜론 구분자만 포함하십시오. 예를 들어, -B::vdisk001을 지정하면 가상 디스크의 이름이 vdisk001로 설정되며 백엔드 및 볼륨에 대해 기본 이름이 사용됩니다. vdisk를 지정하지 않을 경우 후행 콜론 구분자를 생략할 수 있습니다. 예를 들어, -B/ldoms/ldom1/vol001:vol001은 백엔드 파일 이름을 /ldoms/ldom1/vol001로 지정하며 볼륨 이름을 vol001로 지정합니다. 기본 가상 디스크 이름은 disk0입니다.

Oracle VM Server for SPARC P2V 도구 설치

대상 시스템의 컨트롤 도메인에서 **만** Oracle VM Server for SPARC P2V 도구 패키지를 설치하고 구성해야 합니다. 소스 시스템에서는 패키지를 설치할 필요가 없습니다. 대신, `/usr/sbin/ldmp2v` 스크립트를 대상 시스템에서 소스 시스템으로 복사하면 됩니다.

주 - Oracle Solaris 10 시스템의 경우 `ldmp2v`는 `SUNWldmp2v` 패키지에서 설치되며 Oracle Solaris 11 시스템의 경우 `ldmp2v`는 기본적으로 `ldomsmanager` 패키지에서 설치됩니다.

필수 조건

Oracle VM Server for SPARC P2V 도구를 실행하려면 다음 조건이 충족되어야 합니다.

- 소스 시스템에 다음과 같은 플래시 아카이브 패치가 설치되어 있어야 합니다.
 - **Solaris 8 OS의 경우:** 패치 ID 109318-34 이상
 - **Solaris 9 OS의 경우:** 패치 ID 113434-06 이상
- 다음 OS의 대상 시스템에서 Logical Domains 1.1 이상이 실행되어야 합니다.
 - Solaris 10 10/08 OS
 - 적절한 Logical Domains 1.1 패치가 설치된 Solaris 10 5/08 OS
- 게스트 도메인에서 Solaris 10 5/08 OS 이상이 실행되어야 합니다.
- 소스 시스템에서 Solaris 8 OS 이상이 실행되어야 합니다.

이러한 필요 조건 외에 소스 시스템과 대상 시스템에서 NFS 파일 시스템이 공유되도록 구성해야 합니다. `root`로 이 파일 시스템에 쓸 수 있어야 합니다. 단, 공유 파일 시스템을 사용할 수 없는 경우 소스 시스템과 대상 시스템에서 소스 시스템의 파일 시스템 덤프를 보관할 수 있을 만큼 큰 로컬 파일 시스템을 사용하십시오.

제한 사항

Oracle VM Server for SPARC P2V 도구에는 다음 제한 사항이 적용됩니다.

- UFS 파일 시스템만 지원됩니다.
- 소스 시스템에서 일반 디스크(`/dev/dsk/c0t0d0s0`), Oracle Solaris Volume Manager 메타 장치(`/dev/md/dsk/dNNN`) 및 VxVM이 캡슐화한 부트 디스크만 지원됩니다.
- P2V 프로세스를 수행하는 동안 각 게스트 도메인에서는 가상 스위치와 가상 디스크 서버가 하나만 사용될 수 있습니다. P2V 변환이 완료되면 도메인에 가상 스위치와 가상 디스크 서버를 더 추가할 수 있습니다.

- VxVM 볼륨에 대한 지원은 캡슐화된 부트 디스크의 `rootvol`, `swapvol`, `usr`, `var`, `opt` 및 `home` 볼륨으로 제한됩니다. 이러한 볼륨에 대한 원래 슬라이스는 계속 부트 디스크에 있어야 합니다. P2V 도구는 Oracle Solaris 10 OS에서 Veritas Volume Manager 5.x를 지원합니다. 하지만 P2V 도구를 사용하여 VxVM을 사용하는 Solaris 8 및 Solaris 9 운영 체제를 변환할 수 있습니다.
- `ldmp2v collect` 작업을 실행하기 전에 `zoneadm detach` 명령을 사용하여 영역을 분리한 경우 영역이 있는 Oracle Solaris 10 시스템을 변환할 수 있습니다. P2V 변환이 완료되면 `zoneadm attach` 명령을 사용하여 게스트 도메인에 만들어진 영역을 다시 연결할 수 있습니다. 게스트 도메인에서 이러한 단계를 수행하는 것과 관련된 자세한 내용은 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 “비전역 영역을 다른 시스템으로 마이그레이션”를 참조하십시오.

주 - P2V 도구는 영역 구성(예: 영역 경로 또는 네트워크 인터페이스)을 업데이트하지 않습니다. 도구가 이동되거나 영역 경로에 대한 저장소가 구성되지 않습니다. 게스트 도메인에서 수동으로 영역 구성을 업데이트하고 영역 경로를 이동해야 합니다.

Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리의 “비전역 영역을 다른 시스템으로 마이그레이션”를 참조하십시오.

▼ Oracle VM Server for SPARC P2V 도구를 설치하는 방법

이 절차에서는 `SUNWldmp2v` 패키지를 사용하여 Oracle Solaris 10 시스템에서 `ldmp2v` 명령을 설치하는 방법에 대해 설명합니다.

Oracle Solaris 11 시스템에 `ldmp2v` 명령을 설치하려는 경우 `ldomsmanager` 패키지를 설치할 때 기본적으로 이 명령이 설치됩니다.

- 1 Oracle VM Server for SPARC 다운로드 페이지(<http://www.oracle.com/virtualization/index.html>)로 이동합니다.
- 2 P2V 소프트웨어 패키지 `SUNWldmp2v`를 다운로드합니다.
`SUNWldmp2v` 패키지는 Oracle VM Server for SPARC zip 파일에 포함되어 있습니다.
- 3 관리자, 수퍼유저 또는 동등한 역할의 사용자로 로그인합니다.
 Oracle Solaris 10의 경우, **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오. Oracle Solaris 11의 경우, **Oracle Solaris 관리: 보안 서비스의 제III부, “역할, 권한 프로파일 및 권한”**를 참조하십시오.
- 4 `pkgadd` 명령을 사용하여 `SUNWldmp2v` 패키지를 설치합니다.

```
# pkgadd -d . SUNWldmp2v
```
- 5 `/etc/ldmp2v.conf` 파일을 만들고 다음과 같은 기본 등록 정보를 구성합니다.

- VDS – 가상 디스크 서비스의 이름(예: VDS="primary-vds0")입니다.
 - VSW – 가상 스위치의 이름(예: VSW="primary-vsw0")입니다.
 - VCC – 가상 콘솔 집중기의 이름(예: VCC="primary-vcc0")입니다.
 - BACKEND_TYPE – 백엔드 유형(zvol, file 또는 disk)입니다.
 - BACKEND_SPARSE – 백엔드 장치를 스파스 볼륨 또는 파일로 만들지(BACKEND_SPARSE="yes") 아니면 비스파스 볼륨 또는 파일로 만들지(BACKEND_SPARSE="no") 여부입니다.
 - BACKEND_PREFIX – 가상 디스크 백엔드 장치를 만들 위치입니다.
BACKEND_TYPE="zvol"인 경우 BACKEND_PREFIX 값을 ZFS 데이터 세트 이름으로 지정하십시오. BACKEND_TYPE="files"인 경우 BACKEND_PREFIX 값이 /에 대한 상대 디렉토리의 경로 이름으로 해석됩니다.
예를 들어, BACKEND_PREFIX="tank/ldoms"를 사용하면 tank/ldoms/domain-name 데이터 세트에 ZVOL이 만들어지고 /tank/ldoms/domain-name 하위 디렉토리에 파일이 만들어집니다.
BACKEND_PREFIX 등록 정보는 disk 백엔드에 적용할 수 없습니다.
 - BOOT_TIMEOUT – Oracle Solaris OS 부트에 대한 시간 초과(초)입니다.
- 자세한 내용은 다운로드할 수 있는 번들에 포함된 ldmp2v.conf.sample 구성 파일을 참조하십시오.

ldmp2v 명령 사용

이 절에서는 세 단계에 대한 예를 제공합니다.

예 13-1 수집 단계 예

다음 예에서는 ldmp2v collect 명령을 사용하는 방법을 보여줍니다.

- **NFS 마운트된 파일 시스템 공유.** 다음 예에서는 소스 시스템과 대상 시스템이 NFS 마운트된 파일 시스템을 공유하는 경우 collect 단계를 수행하는 가장 간단한 방법을 보여줍니다.
수퍼유저로 필요한 모든 UFS 파일 시스템이 마운트되어 있는지 확인하십시오.

```
volumia# df -k
Filesystem          kbytes    used  avail capacity  Mounted on
/dev/dsk/clt1d0s0    16516485  463289 15888032     3%      /
/proc                0          0      0      0%     /proc
fd                   0          0      0      0%     /dev/fd
mnttab               0          0      0      0%     /etc/mnttab
/dev/dsk/clt1d0s3    8258597   4304  8171708     1%     /var
swap                 4487448    16  4487432     1%     /var/run
swap                 4487448    16  4487432     1%     /tmp
/dev/dsk/clt0d0s0    1016122    9   955146     1%     /u01
vandikhout:/u1/home/dana
6230996752 1051158977 5179837775    17%    /home/dana
```

예 13-1 수집 단계 예 (계속)

다음에서는 소스 시스템과 대상 시스템이 NFS 마운트된 파일 시스템을 공유하는 경우 수집 도구를 실행하는 방법을 보여줍니다.

```
volumia# ldmp2v collect -d home/dana/volumia
Collecting system configuration ...
Archiving file systems ...
Determining which filesystems will be included in the archive...
Creating the archive...
895080 blocks
Archive creation complete.
```

- **NFS 마운트된 파일 시스템 공유 안함.** 소스 시스템과 대상 시스템이 NFS 마운트된 파일 시스템을 공유하지 않는 경우 로컬 저장소에 파일 시스템 이미지를 기록하여 나중에 컨트롤 도메인에 복사할 수 있습니다. 플래시 유틸리티가 만든 아카이브를 자동으로 제외합니다.

```
volumia# ldmp2v collect -d /var/tmp/volumia
Collecting system configuration ...
Archiving file systems ...
Determining which filesystems will be included in the archive...
Creating the archive...
895080 blocks
Archive creation complete.
```

플래시 아카이브 및 manifest 파일을 /var/tmp/volumia 디렉토리에서 대상 시스템으로 복사하십시오.

참고 - ldmp2v가 cpio 명령 오류를 표시하는 경우도 있습니다. 대부분의 경우 이러한 오류는 File size of etc/mnttab has increased by 435와 같은 메시지를 생성합니다. 로그 파일 또는 시스템 상태를 반영하는 파일에 속한 메시지는 무시할 수 있습니다. 모든 오류 메시지를 철저히 검토해야 합니다.

- **파일 시스템 백업 단계 건너뛰기.** 타사 백업 도구(예: NetBackup)를 통해 이미 시스템 백업을 사용할 수 있는 경우 none 아카이브 방법을 사용하여 파일 시스템 백업 단계를 건너뛸 수 있습니다. 이 옵션을 사용할 경우 시스템 구성 매니페스트만 만들어집니다.

```
volumia# ldmp2v collect -d /home/dana/p2v/volumia -a none
Collecting system configuration ...
The following file system(s) must be archived manually: / /u01 /var
```

소스 시스템과 대상 시스템이 -d로 지정된 디렉토리를 공유하지 않을 경우 해당 디렉토리의 콘텐츠를 컨트롤 도메인에 복사해야 합니다. 디렉토리 콘텐츠는 준비 단계 전에 컨트롤 도메인에 복사해야 합니다.

예 13-2 준비 단계 예

다음 예에서는 `ldmp2v prepare` 명령을 사용하는 방법을 보여줍니다.

- 다음 예에서는 물리적 시스템의 MAC 주소를 유지하면서 `/etc/ldmp2v.conf`에서 구성된 기본값을 사용하여 `volumia`라는 논리적 도메인을 만듭니다.

```
# ldmp2v prepare -d /home/dana/p2v/volumia -o keep-mac volumia
Creating vdisks ...
Creating file systems ...
Populating file systems ...
Modifying guest domain OS image ...
Removing SVM configuration ...
Unmounting guest file systems ...
Creating domain volumia ...
Attaching vdisks to domain volumia ...
```

- 다음 명령은 `volumia` 논리적 도메인에 대한 정보를 보여줍니다.

```
# ldm list -l volumia
NAME                STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
volumia             inactive  -----  2        4G

NETWORK
NAME      SERVICE          DEVICE    MAC              MODE    PVID VID
vnet0     primary-vsw0      00:03:ba:1d:7a:5a  1

DISK
NAME      DEVICE  TOUT  MPGROUP    VOLUME              SERVER
disk0     disk0   0      0          volumia-vol0@primary-vds0
disk1     disk1   0      0          volumia-vol1@primary-vds0
```

- 다음에서는 `-C` 옵션을 사용하여 도메인 및 해당 백엔드 장치를 완전히 제거할 수 있음을 보여줍니다.

```
# ldmp2v prepare -C volumia
Cleaning up domain volumia ...
Removing vdisk disk0 ...
Removing vdisk disk1 ...
Removing domain volumia ...
Removing volume volumia-vol0@primary-vds0 ...
Removing ZFS volume tank/ldoms/volumia/disk0 ...
Removing volume volumia-vol1@primary-vds0 ...
Removing ZFS volume tank/ldoms/volumia/disk1 ...
```

- 다음에서는 `-m` 옵션으로 마운트 지점 및 새 크기를 지정하여 P2V 중 파일 시스템 하나 이상의 크기를 조정할 수 있음을 보여줍니다.

```
# ldmp2v prepare -d /home/dana/p2v/volumia -m /:8g volumia
Resizing file systems ...
Creating vdisks ...
Creating file systems ...
Populating file systems ...
Modifying guest domain OS image ...
Removing SVM configuration ...
Modifying file systems on SVM devices ...
Unmounting guest file systems ...
Creating domain volumia ...
Attaching vdisks to domain volumia ...
```

예 13-3 변환 단계 예

다음 예에서는 ldmp2v convert 명령을 사용하는 방법을 보여줍니다.

- **네트워크 설치 서버 사용.** ldmp2v convert 명령은 지정된 가상 네트워크 인터페이스를 사용하여 네트워크를 통해 도메인을 부트합니다. 설치 서버에서 setup_install_server 및 add_install_client 스크립트를 실행해야 합니다.

Oracle Solaris 10 시스템에서는 Oracle Solaris JumpStart 기능을 사용하여 완전히 자동화된 변환 작업을 수행할 수 있습니다. 이 기능을 사용하려면 JumpStart 서버에서 클라이언트에 적절한 sysidcfg 및 프로파일 파일을 만들고 구성해야 합니다. 프로파일은 다음 행으로 구성되어야 합니다.

```
install_type      upgrade
root_device      c0d0s0
```

sysidcfg 파일은 업그레이드 작업에만 사용되므로 다음과 같은 구성으로도 충분합니다.

```
name_service=NONE
root_password=uQkoXlMLCsZhI
system_locale=C
timeserver=localhost
timezone=Europe/Amsterdam
terminal=vt100
security_policy=NONE
nfs4_domain=dynamic
auto_reg=disable
network_interface=PRIMARY {netmask=255.255.255.192
                             default_route=none protocol_ipv6=no}
```

JumpStart 사용에 대한 자세한 내용은 [Oracle Solaris 10 8/11 설치 설명서: 사용자 정의 JumpStart 및 고급 설치](#)를 참조하십시오.

주 - 예제 sysidcfg 파일에는 Oracle Solaris 10 9/10 릴리스에서 소개된 auto_reg 키워드가 들어 있습니다. 이 키워드는 Oracle Solaris 10 9/10 릴리스 이상을 실행 중인 경우에 **만** 필요합니다.

```
# ldmp2v convert -j -n vnet0 -d /p2v/volumia volumia
LDom volumia started
Waiting for Solaris to come up ...
Using Custom JumpStart
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "volumia" in group "volumia" ....
Press ~? for control options ..
SunOS Release 5.10 Version Generic_137137-09 64-bit
Copyright (c) 1983-2010, Oracle and/or its affiliates. All rights reserved.
Configuring devices.
Using RPC Bootparams for network configuration information.
Attempting to configure interface vnet0...
Configured interface vnet0
```

예 13-3 변환 단계 예 (계속)

```

Reading ZFS config: done.
Setting up Java. Please wait...
Serial console, reverting to text install
Beginning system identification...
Searching for configuration file(s)...
Using sysid configuration file
  129.159.206.54:/opt/SUNWjet/Clients/volumia/sysidcfg
Search complete.
Discovering additional network configuration...
Completing system identification...
Starting remote procedure call (RPC) services: done.
System identification complete.
Starting Solaris installation program...
Searching for JumpStart directory...
Using rules.ok from 129.159.206.54:/opt/SUNWjet.
Checking rules.ok file...
Using begin script: Clients/volumia/begin
Using profile: Clients/volumia/profile
Using finish script: Clients/volumia/finish
Executing JumpStart preinstall phase...
Executing begin script "Clients/volumia/begin"...
Begin script Clients/volumia/begin execution completed.
Searching for SolStart directory...
Checking rules.ok file...
Using begin script: install_begin
Using finish script: patch_finish
Executing SolStart preinstall phase...
Executing begin script "install_begin"...
Begin script install_begin execution completed.
WARNING: Backup media not specified. A backup media (backup_media)
  keyword must be specified if an upgrade with disk space reallocation
  is required

Processing profile

Loading local environment and services

Generating upgrade actions
Checking file system space: 100% completed
Space check complete.

Building upgrade script

Preparing system for Solaris upgrade

Upgrading Solaris: 10% completed
[...]
```

- **ISO 이미지 사용.** ldmp2v convert 명령은 Oracle Solaris DVD ISO 이미지를 논리적 도메인에 연결하고 여기서 부트합니다. 업그레이드하려면 sysid 프롬프트에 모두 응답하고 Upgrade를 선택하십시오.

예 13-3 변환 단계 예 (계속)



주의 - 게스트 도메인을 변환하기 전에 안전 검사가 수행됩니다. 이 검사는 네트워크에서 활성 IP 주소가 중복되지 않도록 원래 시스템의 IP 주소가 활성 상태가 아닌지 확인합니다. `-x skip-ping-test` 옵션을 사용하여 이 안전 검사를 건너뛸 수 있습니다. 이 검사를 건너뛰면 변환 프로세스 진행 속도가 빨라집니다. 이 옵션은 원래 호스트가 활성 상태가 아닌 경우처럼 중복되는 IP 주소가 없는 것이 확실한 경우에 **만** 사용하십시오.

주 - `sysid` 질문에 대한 응답은 업그레이드 프로세스를 수행하는 동안에 **만** 사용됩니다. 이 데이터는 디스크의 기존 OS 이미지에 적용되지 않습니다. 가장 빠르고 간단하게 변환을 실행하는 방법은 `Non-networked`를 선택하는 것입니다. 지정하는 `root` 암호는 소스 시스템의 `root` 암호와 일치하지 않아도 됩니다. 시스템의 원래 ID는 업그레이드 간에 보존되며 사후 업그레이드 재부트 후에도 적용됩니다. 업그레이드 수행에 필요한 시간은 원래 시스템에 설치된 Oracle Solaris Cluster에 따라 다릅니다.

```
# ldmp2v convert -i /tank/iso/s10s_u5.iso -d /home/dana/p2v/volumia volumia
Testing original system status ...
LDom volumia started
Waiting for Solaris to come up ...
```

```
      Select 'Upgrade' (F2) when prompted for the installation type.
      Disconnect from the console after the Upgrade has finished.
```

```
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
```

```
Connecting to console "volumia" in group "volumia" ....
Press ~? for control options ..
Configuring devices.
Using RPC Bootparams for network configuration information.
Attempting to configure interface vnet0...
Extracting windowing system. Please wait...
Beginning system identification...
Searching for configuration file(s)...
Search complete.
Discovering additional network configuration...
Configured interface vnet0
Setting up Java. Please wait...
```

```
Select a Language
```

- 0. English
- 1. French
- 2. German
- 3. Italian
- 4. Japanese

예 13-3 변환 단계 예 (계속)

- 5. Korean
- 6. Simplified Chinese
- 7. Spanish
- 8. Swedish
- 9. Traditional Chinese

Please make a choice (0 - 9), or press h or ? for help:

[...]

- Solaris Interactive Installation -----

This system is upgradable, so there are two ways to install the Solaris software.

The Upgrade option updates the Solaris software to the new release, saving as many modifications to the previous version of Solaris software as possible. Back up the system before using the Upgrade option.

The Initial option overwrites the system disks with the new version of Solaris software. This option allows you to preserve any existing file systems. Back up any modifications made to the previous version of Solaris software before starting the Initial option.

After you select an option and complete the tasks that follow, a summary of your actions will be displayed.

 F2_Upgrade F3_Go Back F4_Initial F5_Exit F6_Help

Oracle VM Server for SPARC Configuration Assistant(Oracle Solaris 10)

Oracle VM Server for SPARC Configuration Assistant(`ldmconfig` 명령)는 기본 등록 정보를 설정하여 논리적 도메인의 구성 과정을 안내합니다. CMT(칩 멀티스레딩) 기반 시스템에서 실행됩니다.

구성 데이터를 수집한 후 Configuration Assistant는 논리적 도메인으로 부트하기에 적합한 구성을 만듭니다. 또한 Configuration Assistant에서 선택한 기본값을 사용하여 사용 가능한 시스템 구성을 만들 수 있습니다.

주 - `ldmconfig` 명령은 Oracle Solaris 10 시스템에만 지원됩니다.

이 장과 더불어 [ldmconfig\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

Configuration Assistant 사용(`ldmconfig`)

`ldmconfig` 명령은 사용자 인터페이스 화면에 해당하는 일련의 작업을 통해 작동합니다. 최종 결과는 논리적 도메인에 배치할 수 있는 구성을 만드는 것입니다.

다음 절은 `ldmconfig` 명령과 Configuration Assistant 도구의 일부 기능을 설치하는 방법을 설명합니다.

Configuration Assistant 설치

Configuration Assistant는 `SUNWldm` 패키지의 일부로 제공됩니다.

`SUNWldm` 패키지를 설치한 후 `/usr/sbin` 디렉토리에서 `ldmconfig` 명령을 찾을 수 있습니다. 또한 레거시 목적으로 `/opt/SUNWldm/bin` 디렉토리에 명령이 설치됩니다.

필수 조건

Configuration Assistant를 설치하고 실행하기 전에 다음 조건을 충족하는지 확인합니다.

- 대상 시스템은 최소한 Logical Domains 1.2 소프트웨어를 실행 중이어야 합니다.
- 터미널 창은 최소한 80자 너비에 24행 길이어야 합니다.

제한 사항 및 알려진 문제

Configuration Assistant에는 다음 제한 사항이 있습니다.

- ldmconfig 사용 중 터미널 크기를 조정하면 왜곡된 출력이 나타날 수 있습니다.
- UFS 디스크 파일만 가상 디스크로 지원됩니다.
- 기존의 논리적 도메인 구성이 없는 시스템에만 작동합니다.
- 가상 콘솔 집중기 포트가 5000 - 5100 범위입니다.
- 게스트 도메인, 서비스, 장치에 사용되는 기본 이름은 변경할 수 없습니다.

ldmconfig 기능

ldmconfig 명령은 사용자 인터페이스 화면에 해당하는 일련의 작업을 통해 작동합니다. 최종 단계에 도달할 때까지 이러한 화면을 뒤로(이전) 및 앞으로(다음) 이동할 수 있습니다. 최종 단계에 구성이 생성됩니다. 언제든지 Configuration Assistant를 종료하거나 기본값을 사용하도록 구성을 재설정할 수 있습니다. 최종 화면에서 논리적 도메인에 구성을 배치할 수 있습니다.

먼저 Configuration Assistant가 자동으로 시스템을 검사하여 최적의 사용법에 준하여 가장 적합한 기본 등록 정보 값을 결정한 다음, 배치를 제어하는 데 필요한 등록 정보를 표시합니다. 이것은 완전한 목록이 아닙니다. 추가로 구성을 사용자 정의하려면 다른 등록 정보를 설정할 수 있습니다.

ldmconfig 도구 사용에 대한 내용은 [ldmconfig\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 등록 정보를 조정할 수 있습니다.

- **게스트 도메인 수.** 만들려는 응용 프로그램에 대한 게스트 도메인 수를 지정합니다. 최소값은 1개 게스트 도메인입니다. 최대값은 VCPU 리소스의 가용성에 따라 결정됩니다. 예를 들어, 64-스레드 CMT 시스템에서 각 단일 스레드마다 게스트 도메인을 최대 60개까지 만들 수 있습니다. 4개 스레드는 컨트롤 도메인용으로 예약되어 있습니다. 최적의 사용법이 선택된 경우 게스트 도메인당 최소 VCPU 리소스 수는 단일 코어입니다. 따라서 최적의 사용법이 선택된 8-코어(코어당 8-스레드) 시스템에서 각 1개 코어마다 게스트 도메인을 최대 7개까지 만들 수 있습니다. 또한 1개 코어는 컨트롤 도메인에 지정됩니다.

Configuration Assistant는 이 시스템에 구성할 수 있는 최대 도메인 수를 표시합니다.

Configuration Assistant는 도메인을 만들기 위해 다음 작업을 수행합니다.

■ **모든 도메인에 대해**

- 가상 터미널 서비스를 5000 - 5100 범위의 포트에 만듭니다.
- 가상 디스크 서비스를 만듭니다.
- 추천된 네트워크 어댑터에 가상 네트워크 스위치를 만듭니다.
- 가상 터미널 서버 데몬을 사용으로 설정합니다.

■ **각 도메인에 대해**

- 논리적 도메인을 만듭니다.
- 도메인에 지정된 VCPU를 구성합니다.
- 도메인에 지정된 메모리를 구성합니다.
- 가상 디스크로 사용할 UFS 디스크 파일을 만듭니다.
- 디스크 파일에 대한 가상 디스크 서버 장치(vdsdev)를 만듭니다.
- 디스크 파일을 도메인의 가상 디스크 vdisk0으로 지정합니다.
- 추천된 네트워크 어댑터의 가상 스위치에 첨부된 가상 네트워크 어댑터를 추가합니다.
- OBP 등록 정보 auto-boot?=true를 설정합니다.
- OBP 등록 정보 boot-device=vdisk0을 설정합니다.
- 도메인을 바인딩합니다.
- 도메인을 시작합니다.

- **기본 네트워크.** 새 도메인이 가상 네트워킹에 사용할 네트워크 어댑터를 지정합니다. 어댑터는 시스템에 존재해야 합니다. Configuration Assistant는 현재 시스템에서 기본 어댑터로 사용 중인 어댑터와 활성 링크 상태의 어댑터(케이블로 연결된 어댑터)를 강조 표시합니다.

- **가상 디스크 크기.** 새 도메인마다 가상 디스크를 만듭니다. 이러한 가상 디스크는 로컬 파일 시스템에 위치한 디스크 파일을 기반으로 만듭니다. 이 등록 정보는 각 가상 디스크의 크기(GB)를 제어합니다. Oracle Solaris 10 OS를 포함하는 데 필요한 대략적 크기에 준하여 최소 크기는 8GB이고 최대 크기는 100GB입니다.

Configuration Assistant가 모든 도메인의 디스크 파일을 포함할 충분한 공간을 가진 파일 시스템을 찾을 수 없는 경우 오류 화면이 표시됩니다. 이 경우 응용 프로그램을 다시 실행하기 전에 다음을 수행해야 합니다.

- 가상 디스크의 크기를 줄입니다.
- 도메인 수를 줄입니다.
- 더 높은 용량의 파일 시스템을 추가합니다.
- **가상 디스크 디렉토리.** 새 도메인의 가상 디스크로 만들어질 파일을 저장할 충분한 용량이 있는 파일 시스템을 지정합니다. 디렉토리는 선택된 도메인 수와 가상 디스크의 크기를 기반으로 합니다. 이러한 등록 정보 값이 변경될 때 언제든지 값을 다시 계산하고 대상 디렉토리를 선택해야 합니다. Configuration Assistant는 충분한

공간을 가진 파일 시스템 목록을 제시합니다. 파일 시스템 이름을 지정한 후 Configuration Assistant는 /ldoms/disks라는 파일 시스템에 디스크 이미지를 생성할 디렉토리를 만듭니다.

- **최적의 사용법.** 등록 정보 값에 최적의 사용법을 사용할지 여부를 지정합니다.
 - **yes** 값인 경우 Configuration Assistant는 여러 구성 등록 정보 값에 최적의 사용법을 사용합니다. 시스템 도메인을 포함하여 도메인당 최소한 1개 코어를 강제 지정합니다. 그 결과, 최대 게스트 도메인 수는 시스템에 제공된 전체 코어 수에서 시스템 도메인용 1개 코어를 뺀 값으로 제한됩니다. 예를 들어, 각각 8개 코어가 있는 2-소켓 SPARC Enterprise T5140의 경우 최대 게스트 도메인 수는 15이고 여기에 시스템 도메인이 추가됩니다.
 - **no** 값인 경우 Configuration Assistant는 최소한 1개 스레드가 있는 도메인을 생성하지만, 시스템 도메인에 대해 적어도 4개 스레드를 유지합니다.

그 다음, Configuration Assistant는 만들어질 배치 구성을 다음 정보로 요약합니다.

- 도메인 수
- 각 게스트 도메인에 지정된 CPU
- 각 게스트 도메인에 지정된 메모리
- 가상 디스크의 크기 및 위치
- 게스트 도메인의 가상 네트워크 서비스에 사용할 네트워크 어댑터
- 시스템에서 서비스에 사용할 CPU 및 메모리 양
- 유효한 Oracle Solaris OS DVD가 식별된 경우 공유 가상 CD-ROM 장치를 만들어서 게스트 도메인에 Oracle Solaris OS가 설치되도록 허용합니다.

마지막으로, Configuration Assistant는 지정된 Logical Domains 배치를 만들도록 시스템을 구성합니다. 또한 시스템을 구성하기 위해 취할 조치를 설명하고 실행할 명령을 표시합니다. 이 정보는 시스템을 구성하는데 필요한 ldm 명령의 사용법을 배우는 데 도움이 됩니다.



주의 - 이 구성 단계와 상호 작용하거나 이 프로세스를 중단하지 **마십시오**. 그렇지 않으면 부분적으로 구성된 시스템이 나타날 수 있습니다.

명령이 성공적으로 완료된 후에 변경 사항을 적용하려면 시스템을 재부트합니다.

Oracle VM Server for SPARC Management Information Base 소프트웨어 사용

Oracle VM Server for SPARC MIB(Management Information Base)를 사용하면 타사 시스템 관리 응용 프로그램이 SNMP(Simple Network Management Protocol)를 사용하여 도메인을 원격 모니터링하고 논리적 도메인을 시작/중지할 수 있습니다.

컨트롤 도메인에 Oracle VM Server for SPARC MIB 소프트웨어의 인스턴스를 하나만 실행할 수 있습니다. 컨트롤 도메인은 최소한 Solaris 10 11/06 OS 및 Oracle VM Server for SPARC 2.2 소프트웨어를 실행해야 합니다.

주 - Oracle VM Server for SPARC MIB 소프트웨어는 Oracle Solaris 10 시스템에만 사용할 수 있습니다.

이 장에서는 다음 주제를 다룹니다.

- 272 페이지 “Oracle VM Server for SPARC Management Information Base 개요”
- 275 페이지 “Oracle VM Server for SPARC MIB 소프트웨어 설치 및 구성”
- 277 페이지 “보안 관리”
- 278 페이지 “도메인 모니터링”
- 298 페이지 “SNMP 트랩 사용”
- 304 페이지 “도메인 시작 및 중지”

Oracle VM Server for SPARC MIB를 성공적으로 사용하려면 다음 소프트웨어 제품 및 기능 사용법을 이해해야 합니다.

- Oracle Solaris OS
- Oracle VM Server for SPARC 소프트웨어
- SNMP(Simple Network Management Protocol)
- SNMP MIB(Management Information Base)
- SMA(System Management Agent)
- SNMP 버전 1(SNMPv1), SNMP 버전 2(SNMPv2c), SNMP 버전 3(SNMPv3) 프로토콜
- SMI(Structure of Management Information) 버전 1 및 버전 2
- MIB(Management Information Base) 구조

- ASN.1(Abstract Syntax Notation)

Oracle VM Server for SPARC Management Information Base 개요

이 절에서는 다음 주제를 다룹니다.

- 272 페이지 “소프트웨어 구성 요소”
- 273 페이지 “System Management Agent”
- 274 페이지 “Logical Domains Manager 및 Oracle VM Server for SPARC MIB”
- 274 페이지 “Oracle VM Server for SPARC MIB 객체 트리”

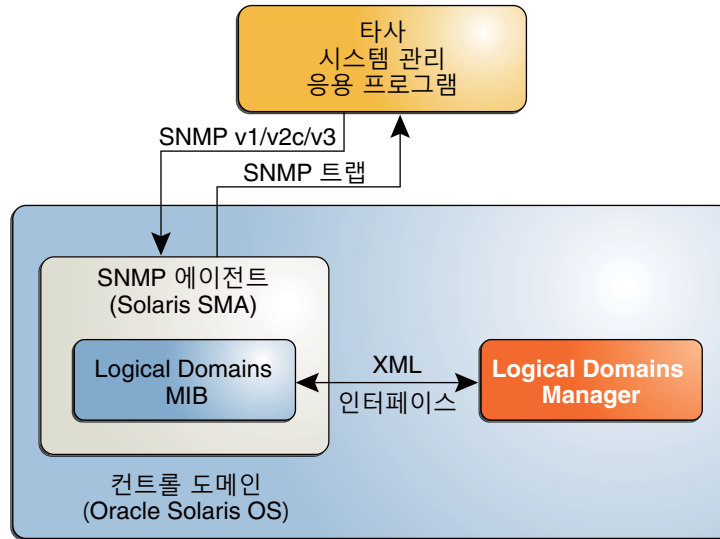
소프트웨어 구성 요소

Oracle VM Server for SPARC MIB 패키지 `SUNWldmib.v`는 다음 소프트웨어 구성 요소를 포함합니다.

- `SUN-LDOM-MIB.mib`는 텍스트 파일 형식의 SNMP MIB입니다. 이 파일은 Oracle VM Server for SPARC MIB의 객체를 정의합니다.
- `ldomMIB.so`는 공유 라이브러리 형식의 System Management Agent 확장 모듈입니다. 이 모듈을 사용하여 SMA는 Oracle VM Server for SPARC MIB에 지정된 정보 요청에 응답하고 트랩을 생성할 수 있습니다.

다음 그림은 Oracle VM Server for SPARC MIB, SMA, Logical Domains Manager 및 타사 시스템 관리 응용 프로그램 사이의 상호 작용을 보여줍니다. 이 그림에 표시된 상호 작용은 273 페이지 “System Management Agent” 및 274 페이지 “Logical Domains Manager 및 Oracle VM Server for SPARC MIB”에 설명됩니다.

그림 15-1 Oracle VM Server for SPARC MIB와 SMA, Logical Domains Manager 및 타사 시스템 관리 응용 프로그램의 상호 작용



System Management Agent

Solaris SNMP 에이전트(SMA)는 다음 기능을 수행합니다.

- 타사 시스템 관리 응용 프로그램에서 요청을 수신하여 Oracle VM Server for SPARC MIB에서 제공한 데이터를 얻거나 설정합니다. 에이전트는 표준 SNMP 포트 161에서 수신합니다.
- SNMP 통지용 표준 포트 162를 사용하여 구성된 시스템 관리 응용 프로그램으로 트랩을 발행합니다.

Oracle Solaris OS 기본 SMA가 컨트롤 도메인에 Oracle VM Server for SPARC MIB를 내보냅니다.

SMA는 SNMP 버전 v1, v2c, v3의 `get`, `set`, `trap` 함수를 지원합니다. 대부분의 Oracle VM Server for SPARC MIB 객체는 모니터링 목적상 읽기 전용입니다. 그러나 도메인을 시작/중지하려면 `ldomTable` 테이블의 `ldomAdminState` 등록 정보에 값을 써야 합니다. 표 15-1을 참조하십시오.

Logical Domains Manager 및 Oracle VM Server for SPARC MIB

도메인은 게스트 운영 체제에 대한 일련의 가상 리소스로 구성된 컨테이너입니다. Logical Domains Manager는 도메인 생성, 구성, 관리를 위한 명령줄 인터페이스(CLI)를 제공합니다. Logical Domains Manager 및 Oracle VM Server for SPARC MIB는 다음 가상 리소스를 지원합니다.

- CPU
- 메모리
- 디스크, 네트워크, 콘솔 I/O
- 암호화 장치

XML 기반 컨트롤 인터페이스 구문 분석

Logical Domains Manager는 XML 기반 컨트롤 인터페이스를 Oracle VM Server for SPARC MIB로 내보냅니다. Oracle VM Server for SPARC MIB는 XML 인터페이스 구문을 분석하고 MIB를 채웁니다. Oracle VM Server for SPARC MIB는 컨트롤 도메인에만 지원됩니다.

SNMP 트랩 제공

Oracle VM Server for SPARC MIB는 업데이트나 상태 변경이 있는지 정기적으로 Logical Domains Manager를 폴링하고, 시스템 관리 응용 프로그램으로 SNMP 트랩을 발행합니다.

결함 및 복구 정보 제공

Oracle VM Server for SPARC MIB가 필요한 리소스를 더 이상 할당할 수 없는 경우 MIB는 SNMP 에이전트를 통해 시스템 관리 응용 프로그램으로 일반 오류를 반환합니다. SNMP 트랩-전달 방식은 오류를 확인하지 않습니다. 특정 상태나 체크포인트는 Oracle VM Server for SPARC MIB에서 구현되지 않습니다. SMA와 Oracle VM Server for SPARC MIB는 init 프로세스와 SMF(서비스 관리 기능)를 통해 시작되고 모니터링됩니다. SMA가 실패하고 종료되면 SMF는 자동으로 프로세스를 다시 시작하고 새 프로세스가 동적으로 Oracle VM Server for SPARC MIB 모듈을 다시 시작합니다.

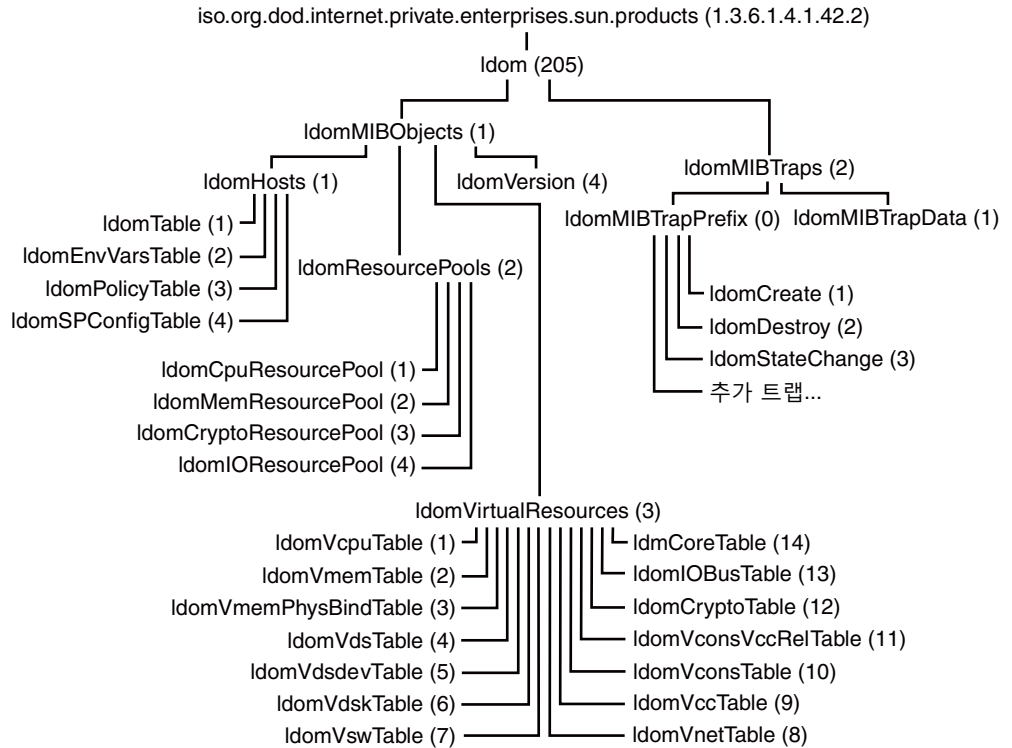
Oracle VM Server for SPARC MIB 객체 트리

SNMP 관리 객체는 트리형 계층으로 구성됩니다. 객체 식별자(OID)는 트리의 노드에 기반한 정수 시리즈를 점으로 구분하여 지정합니다. 각 관리 객체에는 숫자 OID와 연관된 텍스트 이름이 있습니다. 객체 트리의 이 부분에서 Oracle VM Server for SPARC MIB는 ldom (205) 분기로 등록되어 있습니다.

```
iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).sun(42).products(2)
```

다음 그림은 Oracle VM Server for SPARC MIB 아래의 주 하위 트리를 보여줍니다.

그림 15-2 Oracle VM Server for SPARC MIB 트리



Oracle VM Server for SPARC MIB 소프트웨어 설치 및 구성

이 절에서는 Oracle VM Server for SPARC MIB 소프트웨어의 설치 및 구성을 다룹니다. SNMP 관리에 대한 내용은 `snmpd.conf(4)` 매뉴얼 페이지를 참조하십시오.

Oracle VM Server for SPARC MIB 소프트웨어 설치 및 구성(작업 맵)

다음 표는 Oracle VM Server for SPARC MIB 소프트웨어의 설치 및 구성에 사용할 수 있는 작업을 가리킵니다.

작업	설명	수행 방법
primary 도메인에 Oracle VM Server for SPARC MIB 소프트웨어 패키지를 설치합니다.	pkgadd 명령을 사용하여 SUNWldmib.v 패키지를 설치합니다.	276 페이지 “Oracle VM Server for SPARC MIB 소프트웨어 패키지를 설치하는 방법”
Oracle VM Server for SPARC MIB 모듈을 SMA로 로드합니다.	ldomMIB.so 모듈을 Oracle VM Server for SPARC MIB를 질의할 Oracle Solaris OS SMA로 로드합니다.	276 페이지 “Oracle VM Server for SPARC MIB 모듈을 SMA로 로드하는 방법”
primary 도메인에서 Oracle VM Server for SPARC MIB 소프트웨어 패키지를 제거합니다.	pkgrm 명령을 사용하여 SUNWldmib.v 패키지를 제거합니다.	277 페이지 “Oracle VM Server for SPARC MIB 소프트웨어 패키지를 제거하는 방법”

▼ Oracle VM Server for SPARC MIB 소프트웨어 패키지를 설치하는 방법

이 절차는 Oracle VM Server for SPARC 2.2 소프트웨어의 일부로 포함된 Oracle VM Server for SPARC MIB 소프트웨어 패키지 SUNWldmib.v를 설치하는 방법을 설명합니다.

이 패키지를 설치한 후 Oracle VM Server for SPARC MIB 모듈을 동적으로 로드하도록 시스템을 구성할 수 있습니다. [276 페이지 “Oracle VM Server for SPARC MIB 모듈을 SMA로 로드하는 방법”](#)을 참조하십시오.

시작하기 전에 Oracle VM Server for SPARC 2.2 소프트웨어를 다운로드하고 설치합니다. [2 장, “소프트웨어 설치 및 사용”](#)을 참조하십시오.

● SUNWldmib.v 패키지를 primary 도메인에 추가합니다.

```
# pkgadd -d . SUNWldmib.v
```

이 명령은 다음 파일을 설치합니다.

- /opt/SUNWldmib/lib/mibs/SUN-LDOM-MIB.mib
- /opt/SUNWldmib/lib/ldomMIB.so

▼ Oracle VM Server for SPARC MIB 모듈을 SMA로 로드하는 방법

Oracle VM Server for SPARC MIB 모듈 ldomMIB.so가 Oracle VM Server for SPARC MIB를 질의할 Oracle Solaris OS SMA로 로드되어야 합니다. Oracle VM Server for SPARC MIB 모듈은 SNMP 에이전트 안에서 동적으로 로드되므로 에이전트 이진을 재컴파일하고 재링크할 필요가 없습니다.

이 절차는 Oracle VM Server for SPARC MIB 모듈을 동적으로 로드하도록 시스템을 구성하는 방법을 설명합니다. SMA를 다시 시작하지 않고 모듈을 동적으로 로드하기 위한 지침은 [Solaris System Management Agent Developer's Guide](#)에 제공됩니다. SMA에 대한 자세한 내용은 [Solaris System Management Administration Guide](#)를 참조하십시오.

1 SMA SNMP 구성 파일을 업데이트합니다.

다음 행을 /etc/sma/snmp/snmpd.conf 구성 파일에 추가합니다.

```
dlmod ldomMIB /opt/SUNWldmib/lib/ldomMIB.so
```

2 SMA를 다시 시작합니다.

```
# svcadm restart svc:/application/management/sma:default
```

▼ Oracle VM Server for SPARC MIB 소프트웨어 패키지를 제거하는 방법

이 절차는 Oracle VM Server for SPARC MIB 소프트웨어 패키지 SUNWldmib.v를 제거하고 SMA에서 Oracle VM Server for SPARC MIB 모듈을 언로드하는 방법을 설명합니다.

1 SMA를 중지합니다.

```
# svcadm disable svc:/application/management/sma:default
```

2 primary 도메인에서 Oracle VM Server for SPARC MIB 소프트웨어 패키지를 제거합니다.

```
# pkgrm SUNWldmib
```

3 SMA SNMP 구성 파일을 업데이트합니다.

설치 중 /etc/sma/snmp/snmpd.conf 파일에 추가한 행을 제거합니다.

```
dlmod ldomMIB /opt/SUNWldmib/lib/ldomMIB.so
```

4 SMA를 다시 시작합니다.

```
# svcadm restart svc:/application/management/sma:default
```

보안 관리

이 절에서는 SMA(System Management Agent)에 보안 액세스할 수 있는 SNMP(Simple Network Management Protocol) 버전 3(v3) 사용자를 새로 만드는 방법을 설명합니다. SNMP 버전 1(v1) 및 버전 2(v2c)의 경우, 액세스 제어 방식은 SNMP 서버와 클라이언트 사이의 관계를 정의하는 커뮤니티 문자열입니다. 이 문자열은 사용자의 시스템 액세스를 제어하는 암호와 마찬가지로 클라이언트의 서버 액세스를 제어합니다. **Solaris System Management Agent Administration Guide**를 참조하십시오.

주 - snmpv3 사용자를 만들면 SNMP를 통해 SMA와 Oracle VM Server for SPARC MIB가 사용할 수 있습니다. 이 유형의 사용자는 Logical Domains Manager에 대해 Oracle Solaris의 RBAC(역할 기반 액세스 제어) 기능을 사용하여 구성한 사용자와 전혀 상호 작용하거나 충돌하지 않습니다.

▼ 초기 snmpv3 사용자를 만드는 방법

이 절차는 초기 snmpv3 사용자를 만드는 방법을 설명합니다.

이 초기 사용자를 복제하여 추가 사용자를 만들 수 있습니다. 복제된 후속 사용자는 초기 사용자의 인증 및 보안 유형을 상속합니다. 이러한 유형을 나중에 변경할 수 있습니다.

초기 사용자를 복제할 때 새 사용자에 대한 비밀번호 데이터를 설정합니다. 초기 사용자와 구성할 후속 사용자에 대한 암호를 알아야 합니다. 초기 사용자로부터 한 번에 하나의 사용자만 복제할 수 있습니다. Oracle Solaris OS 버전은 **Solaris System Management Agent Administration Guide**의 “To Create Additional SNMPv3 Users with Security”를 참조하십시오.

1 SMA를 중지합니다.

```
# svcadm disable -t svc:/application/management/sma:default
```

2 초기 사용자를 만듭니다.

```
# /usr/sfw/bin/net-snmp-config --create-snmpv3-user -a my-password initial-user
```

이 명령은 사용자 *initial-user*를 선택한 암호 *my-password*로 만들고 */etc/sma/snmp/snmpd.conf* 파일에 항목을 추가합니다. 이 항목은 초기 사용자에게 에이전트 읽기-쓰기 액세스를 부여합니다.

주 - 암호는 최소한 8자를 포함해야 합니다.

3 SMA를 시작합니다.

```
# svcadm enable svc:/application/management/sma:default
```

4 초기 사용자가 만들어졌는지 확인합니다.

```
# snmpget -v 3 -u initial-user -l authNoPriv -a MD5 -A my-password localhost sysUpTime.0
```

도메인 모니터링

이 절에서는 Oracle VM Server for SPARC MIB를 질의하여 논리적 도메인을 모니터링하는 방법을 설명합니다. 또한 이 절에서는 다양한 유형의 MIB 출력을 설명합니다.

이 절에서는 다음 주제를 다룹니다.

- 279 페이지 “환경 변수 설정”
- 279 페이지 “Oracle VM Server for SPARC MIB 질의”
- 281 페이지 “Oracle VM Server for SPARC MIB 정보 검색”

환경 변수 설정

▼ 환경 변수를 설정하는 방법

Oracle VM Server for SPARC MIB를 질의하기 전에 사용할 셸에 대한 환경 변수를 설정해야 합니다. 이 절차는 C 셸, Bourne 셸, Korn 셸에 대해 이러한 변수를 설정하는 방법을 설명합니다.

● PATH, MIBDIRS, MIBS 환경 변수를 설정합니다.

■ C 셸 사용자:

```
% setenv PATH /usr/sfw/bin:$PATH
% setenv MIBDIRS /opt/SUNWldmib/lib/mibs:/etc/sma/snmp/mibs
% setenv MIBS +SUN-LDOM-MIB
```

■ Bourne 및 Korn 셸 사용자:

```
$ PATH=/usr/sfw/bin:$PATH; export PATH
$ MIBDIRS=/opt/SUNWldmib/lib/mibs:/etc/sma/snmp/mibs; export MIBDIRS
$ MIBS=+SUN-LDOM-MIB; export MIBS
```

Oracle VM Server for SPARC MIB 질의

▼ Oracle VM Server for SPARC MIB 객체를 검색하는 방법

시스템에 많은 수의 도메인이 있는 경우 SNMP 요청에 응답하기 전에 SNMP 에이전트가 시간 초과될 수 있습니다. 시간 초과 값을 늘리려면 -t 옵션을 사용하여 시간 초과 값을 길게 지정합니다. 예를 들어, 다음 snmpwalk 명령은 시간 초과 값을 20초로 설정합니다.

```
# snmpwalk -t 20 -v1 -c public localhost SUN-LDOM-MIB::ldomTable
```

또한 -t 옵션을 사용하여 snmpget 및 snmptable 명령에 대한 시간 초과 값을 지정할 수 있습니다.

● 하나 이상의 MIB 객체를 검색합니다.

■ 단일 MIB 객체를 검색합니다.

```
# snmpget -v version -c community-string host MIB-object
```

■ MIB 객체의 배열을 검색합니다.

snmpwalk 또는 snmptable 명령을 사용합니다.

```
# snmpwalk -v version -c community-string host MIB-object
# snmptable -v version -c community-string host MIB-object
```

예 15-1 단일 Oracle VM Server for SPARC MIB 객체 검색(snmppget)

다음 snmpget 명령은 ldomVersionMajor 객체의 값을 질의합니다. localhost 호스트에 대해 snmpv1(-v1) 및 커뮤니티 문자열(-c public)을 지정합니다.

```
# snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomVersionMajor.0
SUN-LDOM-MIB::ldomVersionMajor.0 = INTEGER: 1
```

예 15-2 ldomTable에서 객체 값 검색(snmppwalk)

다음 예제는 snmppwalk 명령을 사용하여 ldomTable에서 객체 값을 검색하는 방법을 보여줍니다.

- 다음 snmppwalk -v1 명령은 ldomTable 테이블의 모든 객체에 대한 값을 반환합니다.

```
# snmppwalk -v1 -c public localhost SUN-LDOM-MIB::ldomTable
SUN-LDOM-MIB::ldomName.1 = STRING: primary
SUN-LDOM-MIB::ldomName.2 = STRING: LdomMibTest_1
SUN-LDOM-MIB::ldomAdminState.1 = INTEGER: 0
SUN-LDOM-MIB::ldomAdminState.2 = INTEGER: 0
SUN-LDOM-MIB::ldomOperState.1 = INTEGER: active(1)
SUN-LDOM-MIB::ldomOperState.2 = INTEGER: bound(6)
SUN-LDOM-MIB::ldomNumVCpu.1 = INTEGER: 32
SUN-LDOM-MIB::ldomNumVCpu.2 = INTEGER: 2
SUN-LDOM-MIB::ldomMemSize.1 = INTEGER: 3968
SUN-LDOM-MIB::ldomMemSize.2 = INTEGER: 256
SUN-LDOM-MIB::ldomMemUnit.1 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomMemUnit.2 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomNumCrypto.1 = INTEGER: 8
SUN-LDOM-MIB::ldomNumCrypto.2 = INTEGER: 0
SUN-LDOM-MIB::ldomNumIOBus.1 = INTEGER: 2
SUN-LDOM-MIB::ldomNumIOBus.2 = INTEGER: 0
SUN-LDOM-MIB::ldomUUID.1 = STRING: c2c3d93b-a3f9-60f6-a45e-f35d55c05fb6
SUN-LDOM-MIB::ldomUUID.2 = STRING: af0b05f0-d262-e633-af32-a6c4e81fb81c
SUN-LDOM-MIB::ldomMacAddress.1 = STRING: 00:14:4f:86:63:2a
SUN-LDOM-MIB::ldomMacAddress.2 = STRING: 00:14:4f:fa:78:b9
SUN-LDOM-MIB::ldomHostID.1 = STRING: 0x8486632a
SUN-LDOM-MIB::ldomHostID.2 = STRING: 0x84fa78b9
SUN-LDOM-MIB::ldomFailurePolicy.1 = STRING: ignore
SUN-LDOM-MIB::ldomFailurePolicy.2 = STRING: ignore
SUN-LDOM-MIB::ldomMaster.1 = STRING:
SUN-LDOM-MIB::ldomMaster.2 = STRING:
```

- 다음 snmppwalk 명령은 snmpv2c 및 snmpv3를 사용하여 ldomTable의 내용을 검색합니다.

```
# snmppwalk -v2c -c public localhost SUN-LDOM-MIB::ldomTable
# snmppwalk -v 3 -u test -l authNoPriv -a MD5 -A testpassword localhost \
SUN-LDOM-MIB::ldomTable
```

예 15-3 ldomTable에서 객체 값을 표 형식으로 검색(snmptable)

다음 예제는 snmptable 명령을 사용하여 ldomTable에서 객체 값을 표 형식으로 검색하는 방법을 보여줍니다.

- 다음 snmptable -v1 명령은 ldomTable의 내용을 표 형식으로 보여줍니다.

```
# snmptable -v1 -c public localhost SUN-LDOM-MIB::ldomTable
```

- 다음 snmptable 명령은 snmpv2c를 사용하여 ldomTable의 내용을 표 형식으로 보여줍니다.

v2c 또는 v3 snmptable 명령의 경우 데이터를 검색하기 위해 -CB 옵션에 GETBULK가 아닌 GETNEXT 요청만 지정할 수 있습니다.

```
# snmptable -v2c -CB -c public localhost SUN-LDOM-MIB::ldomTable
```

Oracle VM Server for SPARC MIB 정보 검색

이 절에서는 Oracle VM Server for SPARC MIB에서 테이블 및 스칼라 객체 형식으로 검색할 수 있는 정보를 설명합니다.

도메인 테이블(ldomTable)

ldomTable은 시스템의 각 도메인을 나타내는 데 사용됩니다. 가상 CPU, 메모리, 암호화 장치, I/O 버스에 대한 리소스 제약 조건이 포함됩니다. 또한 UUID(Universally Unique Identifier), MAC 주소, 호스트 ID, 장애 정책, 마스터 도메인과 같은 다른 도메인 정보도 포함됩니다.

표 15-1 도메인 테이블(ldomTable)

이름	데이터 유형	액세스	설명
ldomIndex	정수	액세스할 수 없음	이 테이블의 인덱스로 사용되는 정수 없음
ldomName	표시 문자열	읽기 전용	도메인의 이름
ldomAdminState	정수	읽기/쓰기	활성 관리를 위해 도메인을 시작/중지합니다. <ul style="list-style-type: none"> ■ 값 1 - 도메인 시작 ■ 값 2 - 도메인 중지
ldomOperState	정수	읽기 전용	도메인의 현재 상태로, 다음 값 중 하나일 수 있습니다. <ul style="list-style-type: none"> ■ 1 - 활성 상태 ■ 2 - 중지 중 상태 ■ 3 - 비활성 상태 ■ 4 - 바인딩 중 상태 ■ 5 - 바인딩 해제 중 상태 ■ 6 - 바인딩됨 상태 ■ 7 - 시작 중 상태
ldomNumVCPU	정수	읽기 전용	사용된 가상 CPU 수. 도메인이 비활성 상태인 경우 이 값은 요청된 가상 CPU 수입니다.

표 15-1 도메인 테이블(ldomTable) (계속)

이름	데이터 유형	액세스	설명
ldomMemSize	정수	읽기 전용	사용된 가상 메모리 양. 도메인이 비활성 상태인 경우 이 값은 요청된 메모리 크기입니다.
ldomMemUnit	정수	읽기 전용	다음 메모리 단위 중 하나입니다. <ul style="list-style-type: none"> ■ 1 - KB ■ 2 - MB ■ 3 - GB ■ 4 - 바이트 지정되지 않은 경우 단위 값은 바이트입니다.
ldomNumCrypto	정수	읽기 전용	사용된 암호화 장치 수. 도메인이 비활성 상태인 경우 이 값은 요청된 암호화 장치 수입니다.
ldomNumIOBus	정수	읽기 전용	사용된 물리적 I/O 장치 수
ldomUUID	표시 문자열	읽기 전용	도메인의 UUID
ldomMacAddress	표시 문자열	읽기 전용	도메인의 MAC 주소
ldomHostID	표시 문자열	읽기 전용	도메인의 호스트 ID
ldomFailurePolicy	표시 문자열	읽기 전용	마스터 도메인의 장애 정책으로 ignore, panic, reset, stop 중 하나일 수 있습니다.
ldomMaster	표시 문자열	읽기 전용	슬레이브 도메인에 대한 최대 4개 마스터 도메인의 이름

환경 변수 테이블(ldomEnvVarsTable)

ldomEnvVarsTable은 모든 도메인이 사용하는 OpenBoot PROM 환경 변수를 설명합니다.

표 15-2 환경 변수 테이블(ldomEnvVarsTable)

이름	데이터 유형	액세스	설명
ldomEnvVarsLdomIndex	정수	읽기 전용	OpenBoot PROM 환경 변수를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomEnvVarsIndex	정수	읽기 전용	이 테이블에서 OpenBoot PROM 환경 변수를 인덱싱하는 데 사용되는 정수
ldomEnvVarsName	표시 문자열	읽기 전용	OpenBoot PROM 변수의 이름
ldomEnvVarsValue	표시 문자열	읽기 전용	OpenBoot PROM 변수의 값

도메인 정책 테이블(ldomPolicyTable)

ldomPolicyTable은 모든 도메인에 적용되는 DRM(동적 리소스 관리) 정책을 설명합니다.

표 15-3 도메인 정책 테이블(ldomPolicyTable)

이름	데이터 유형	액세스	설명
ldomPolicyLdomIndex	정수	읽기 전용	DRM 정책을 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomPolicyIndex	정수	액세스할 수 없음	이 테이블에서 DRM 정책을 인덱싱하는 데 사용되는 정수
ldomPolicyName	표시 문자열	읽기 전용	정책 이름
ldomPolicyStatus	표시 문자열	읽기 전용	정책 상태
ldomPolicyPriority	정수	읽기 전용	정책이 겹칠 때 선택할 DRM 정책을 결정하는 데 사용되는 우선 순위
ldomPolicyVcpuMin	정수	읽기 전용	도메인의 최소 가상 CPU 수
ldomPolicyVcpuMax	정수	읽기 전용	도메인의 최대 가상 CPU 수. unlimited 값은 최대 정수 값 2147483647을 사용합니다.
ldomPolicyUtilLower	정수	읽기 전용	정책 분석이 트리거되는 하한 활용률 레벨
ldomPolicyUtilUpper	정수	읽기 전용	정책 분석이 트리거되는 상한 활용률 레벨
ldomPolicyTodBegin	표시 문자열	읽기 전용	정책의 유효 시작 시간. 형식 <i>hh:mm:ss</i>
ldomPolicyTodEnd	표시 문자열	읽기 전용	정책의 유효 중지 시간. 형식 <i>hh:mm:ss</i>
ldomPolicySampleRate	정수	읽기 전용	리소스 순환 시간(초)
ldomPolicyElasticMargin	정수	읽기 전용	낮은 가상 CPU 수에서 변동을 피하기 위해 util-lower 등록 정보(ldomPolicyUtilLower)와 사용 가능한 가상 CPU 수 사이의 버퍼 양
ldomPolicyAttack	정수	읽기 전용	한 번의 리소스-컨트롤 주기 동안 추가할 최대 리소스 양. unlimited 값은 최대 정수 값 2147483647을 사용합니다.
ldomPolicyDecay	정수	읽기 전용	한 번의 리소스-컨트롤 주기 동안 제거할 최대 리소스 양

서비스 프로세서 구성 테이블(ldomSPConfigTable)

ldomSPConfigTable은 모든 도메인에 대한 서비스 프로세서(SP) 구성을 설명합니다.

표 15-4 서비스 프로세서 구성 테이블(ldomSPConfigTable)

이름	데이터 유형	액세스	설명
ldomSPConfigIndex	정수	액세스할 수 없음	이 테이블에서 SP 구성을 인덱싱하는데 사용되는 정수
ldomSPConfigName	표시 문자열	읽기 전용	SP 구성 이름
ldomSPConfigStatus	표시 문자열	읽기 전용	SP 구성 상태

도메인 리소스 풀 및 스칼라 변수

다음 리소스를 도메인에 지정할 수 있습니다.

- 가상 CPU(vcpu)
- 메모리(mem)
- 암호화 장치(mau)
- 가상 스위치(vsw)
- 가상 네트워크(vnet)
- 가상 디스크 서버(vds)
- 가상 디스크 서버 장치(vdsdev)
- 가상 디스크(vdisk)
- 가상 콘솔 집중기(vcc)
- 가상 콘솔(vcons)
- 물리적 I/O 장치(io)

다음 스칼라 MIB 변수는 리소스 풀과 해당 등록 정보를 나타내는 데 사용됩니다.

표 15-5 CPU 리소스 풀에 대한 스칼라 변수

이름	데이터 유형	액세스	설명
ldomCpuRpCapacity	정수	읽기 전용	ldomCpuRpCapacityUnit 단위로 리소스 풀에 허용된 최대 예약
ldomCpuRpReserved	정수	읽기 전용	현재 리소스 풀에서 예약된 CPU의 누적된 프로세서 클럭 속도(MHz)
ldomCpuRpCapacityUnit 및 ldomCpuRpReservedUnit	정수	읽기 전용	다음 CPU 할당 단위 중 하나입니다. <ul style="list-style-type: none"> ■ 1 - MHz ■ 2 - GHz 기본값은 MHz입니다.

표 15-6 메모리 리소스 풀에 대한 스칼라 변수

이름	데이터 유형	액세스	설명
ldomMemRpCapacity	정수	읽기 전용	MemRpCapacityUnit 단위로 리소스 풀에 허용된 최대 예약
ldomMemRpReserved	정수	읽기 전용	현재 리소스 풀에서 예약된 메모리 양(MemRpReservedUnit 단위)
ldomMemRpCapacityUnit 및 ldomMemRpReservedUnit	정수	읽기 전용	다음 메모리 할당 단위 중 하나입니다. <ul style="list-style-type: none"> ■ 1 - KB ■ 2 - MB ■ 3 - GB ■ 4 - 바이트 지정되지 않은 경우 단위 값은 바이트입니다.

표 15-7 암호화 리소스 풀에 대한 스칼라 변수

이름	데이터 유형	액세스	설명
ldomCryptoRpCapacity	정수	읽기 전용	리소스 풀에 허용된 최대 예약
ldomCryptoRpReserved	정수	읽기 전용	현재 리소스 풀에서 예약된 암호화 장치 수

표 15-8 I/O 버스 리소스 풀에 대한 스칼라 변수

이름	데이터 유형	액세스	설명
ldomIOBusRpCapacity	정수	읽기 전용	풀에 허용된 최대 예약
ldomIOBusRpReserved	정수	읽기 전용	현재 리소스 풀에서 예약된 I/O 버스 수

가상 CPU 테이블(ldomVcpuTable)

ldomVcpuTable은 모든 도메인이 사용하는 가상 CPU를 설명합니다.

표 15-9 가상 CPU 테이블(ldomVcpuTable)

이름	데이터 유형	액세스	설명
ldomVcpuLdomIndex	정수	읽기 전용	가상 CPU를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomVcpuIndex	정수	액세스할 수 없음	이 테이블에서 가상 CPU를 인덱싱하는 데 사용되는 정수

표 15-9 가상 CPU 테이블(ldomVcpuTable) (계속)

이름	데이터 유형	액세스	설명
ldomVcpuDeviceID	표시 문자열	읽기 전용	가상 CPU의 식별자(VID)
ldomVcpuOperationalStatus	정수	읽기 전용	다음 CPU 상태 중 하나입니다. 1=알 수 없음 2=기타 3=정상 4=저하됨 5=스트레스 6=실패 예상 7=오류 8=복구할 수 없는 오류 9=시작 중 10=중지 중 11=중지됨 12=서비스 중 13=연결 안됨 14=통신 손실 15=중단됨 16=유 휴 17=지원 엔티티 오류 18=완료됨 19=전원 모드 Logical Domains Manager는 CPU 상태를 제공하지 않으므로 기본값은 1(알 수 없음)입니다.
ldomVcpuPhysBind	표시 문자열	읽기 전용	물리적 바인딩(PID). 이 가상 CPU에 지정된 하드웨어 스레드(스트랜드)의 식별자를 포함합니다. 이 식별자는 코어 및 칩을 고유하게 식별합니다.

표 15-9 가상 CPU 테이블(ldomVcpuTable) (계속)

이름	데이터 유형	액세스	설명
ldomVcpuPhysBindUsage	정수	읽기 전용	전체 스레드 용량 중 이 가상 CPU에서 사용된 양(MHz)을 나타냅니다. 예를 들어, 스레드가 최대 1GHz에서 실행할 수 있다고 가정합니다. 해당 용량의 절반만 이 가상 CPU에 할당된 경우(스레드의 50%) 등록 정보 값은 500입니다.
ldomVcpuCoreID	표시 문자열	읽기 전용	코어의 식별자(코어 ID)
ldomVcpuUtilPercent	표시 문자열	읽기 전용	가상 CPU의 활용률(백분율)을 나타냅니다.

가상 메모리 테이블

도메인의 메모리 공간을 **실 메모리**, 즉 **가상 메모리**라고 합니다. 하이퍼바이저에서 감지된 호스트 플랫폼 메모리 공간을 **물리적 메모리**라고 합니다. 하이퍼바이저는 물리적 메모리 블록을 매핑하여 도메인에서 사용되는 실 메모리 블록을 구성합니다.

다음 예제는 요청된 메모리 크기를 단일 대형 메모리 블록에 지정하는 대신, 두 메모리 블록으로 분할할 수 있음을 보여줍니다. 도메인이 521MB의 실 메모리를 요청한다고 가정합니다. 메모리는 {*physical-address, real-address, size*} 형식을 사용하여 호스트 시스템의 256MB 블록 두 개에 물리적 메모리로 지정할 수 있습니다.

```
{0x1000000, 0x1000000, 256}, {0x2000000, 0x2000000, 256}
```

도메인은 최대 64개의 물리적 메모리 세그먼트를 게스트 도메인에 지정할 수 있습니다. 따라서 표시 문자열 대신 보조 테이블을 사용하여 각 메모리 세그먼트를 보유합니다. 표시 문자열은 255자 제한이 있습니다.

가상 메모리 테이블(ldomVmemTable)

ldomVmemTable은 도메인이 사용하는 가상 메모리의 등록 정보를 설명합니다.

표 15-10 가상 메모리 테이블(ldomVmemTable)

이름	데이터 유형	액세스	설명
ldomVmemLdomIndex	정수	읽기 전용	가상 메모리를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomVmemIndex	정수	액세스할 수 없음	이 테이블에서 가상 메모리를 인덱싱하는 데 사용되는 정수
ldomVmemNumberOfBlocks	정수	읽기 전용	가상 메모리의 블록 수

가상 메모리 물리적 바인딩 테이블(ldomVmemPhysBindTable)

ldomVmemPhysBindTable은 모든 도메인에 대한 물리적 메모리 세그먼트를 포함하는 보조 테이블입니다.

표 15-11 가상 메모리 물리적 바인딩 테이블(ldomVmemPhysBindTable)

이름	데이터 유형	액세스	설명
ldomVmemPhysBindLdomIndex	정수	읽기 전용	물리적 메모리 세그먼트를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomVmemPhysBind	표시 문자열	읽기 전용	이 가상 메모리 블록에 매핑된 물리적 메모리 목록으로, 다음 형식을 사용합니다. { <i>physical-address</i> , <i>real-address</i> , <i>size</i> }

가상 디스크 테이블

가상 디스크 서비스(vds)와 매핑된 물리적 장치(vdsdev)는 Oracle VM Server for SPARC 기술에 가상 디스크 기능을 제공합니다. 가상 디스크 서비스는 많은 로컬 볼륨(물리적 디스크 또는 파일 시스템)을 내보냅니다. 가상 디스크 서비스가 지정된 경우 다음이 포함됩니다.

- 지원 장치(vdsdev)의 완전한 /dev 경로
- 서비스에 추가하려는 장치의 고유 이름(볼륨 이름)

하나 이상의 디스크, 디스크 슬라이스, 파일 시스템을 단일 디스크 서비스에 바인딩할 수 있습니다. 각 디스크에는 고유 이름과 볼륨 이름이 있습니다. 볼륨 이름은 디스크를 서비스에 바인딩할 때 사용됩니다. Logical Domains Manager는 가상 디스크 서비스와 해당 논리적 볼륨으로부터 가상 디스크 클라이언트(vdisk)를 만듭니다.

가상 디스크 서비스 테이블(ldomVdsTable)

ldomVdsTable은 모든 도메인에 대한 가상 디스크 서비스를 설명합니다.

표 15-12 가상 디스크 서비스 테이블(ldomVdsTable)

이름	데이터 유형	액세스	설명
ldomVdsLdomIndex	정수	읽기 전용	가상 디스크 서비스를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomVdsIndex	정수	액세스할 수 없음	이 테이블에서 가상 디스크 서비스를 인덱싱하는 데 사용되는 정수

표 15-12 가상 디스크 서비스 테이블(ldomVdsTable) (계속)

이름	데이터 유형	액세스	설명
ldomVdsServiceName	표시 문자열	읽기 전용	가상 디스크 서비스의 서비스 이름. 등록 정보 값은 <code>ldm add-vds</code> 명령으로 지정된 <i>service-name</i> 입니다.
ldomVdsNumofAvailVolume	정수	읽기 전용	이 가상 디스크 서비스에서 내보낸 논리적 볼륨 수
ldomVdsNumofUsedVolume	정수	읽기 전용	이 가상 디스크 서비스에 사용된(바인딩됨) 논리적 볼륨 수

가상 디스크 서비스 장치 테이블(ldomVdsdevTable)

ldomVdsdevTable은 모든 가상 디스크 서비스가 사용하는 가상 디스크 서비스 장치를 설명합니다.

표 15-13 가상 디스크 서비스 장치 테이블(ldomVdsdevTable)

이름	데이터 유형	액세스	설명
ldomVdsdevVdsIndex	정수	읽기 전용	가상 디스크 장치를 포함하는 가상 디스크 서비스를 나타내는 ldomVdsTable의 인덱스로 사용되는 정수
ldomVdsdevIndex	정수	액세스할 수 없음	이 테이블에서 가상 디스크 서비스 장치를 인덱싱하는 데 사용되는 정수
ldomVdsdevVolumeName	표시 문자열	읽기 전용	가상 디스크 서비스 장치의 볼륨 이름. 이 등록 정보는 가상 디스크 서비스에 추가하려는 장치의 고유 이름을 지정합니다. 이 장치를 추가할 목적으로 가상 디스크 서비스가 이 이름을 클라이언트로 내보냅니다. 등록 정보 값은 <code>ldm add-vdsdev</code> 명령으로 지정된 <i>volume-name</i> 입니다.
ldomVdsdevDevPath	표시 문자열	읽기 전용	물리적 디스크 장치의 경로 이름. 등록 정보 값은 <code>ldm add-vdsdev</code> 명령으로 지정된 <i>backend</i> 입니다.
ldomVdsdevOptions	표시 문자열	읽기 전용	디스크 장치에 대한 하나 이상의 옵션으로 <i>ro</i> , <i>slice</i> , <i>excl</i> 입니다.
ldomVdsdevMPGroup	표시 문자열	읽기 전용	디스크 장치의 다중 경로 그룹 이름

가상 디스크 테이블(ldomVdiskTable)

ldomVdiskTable은 모든 도메인에 대한 가상 디스크를 설명합니다.

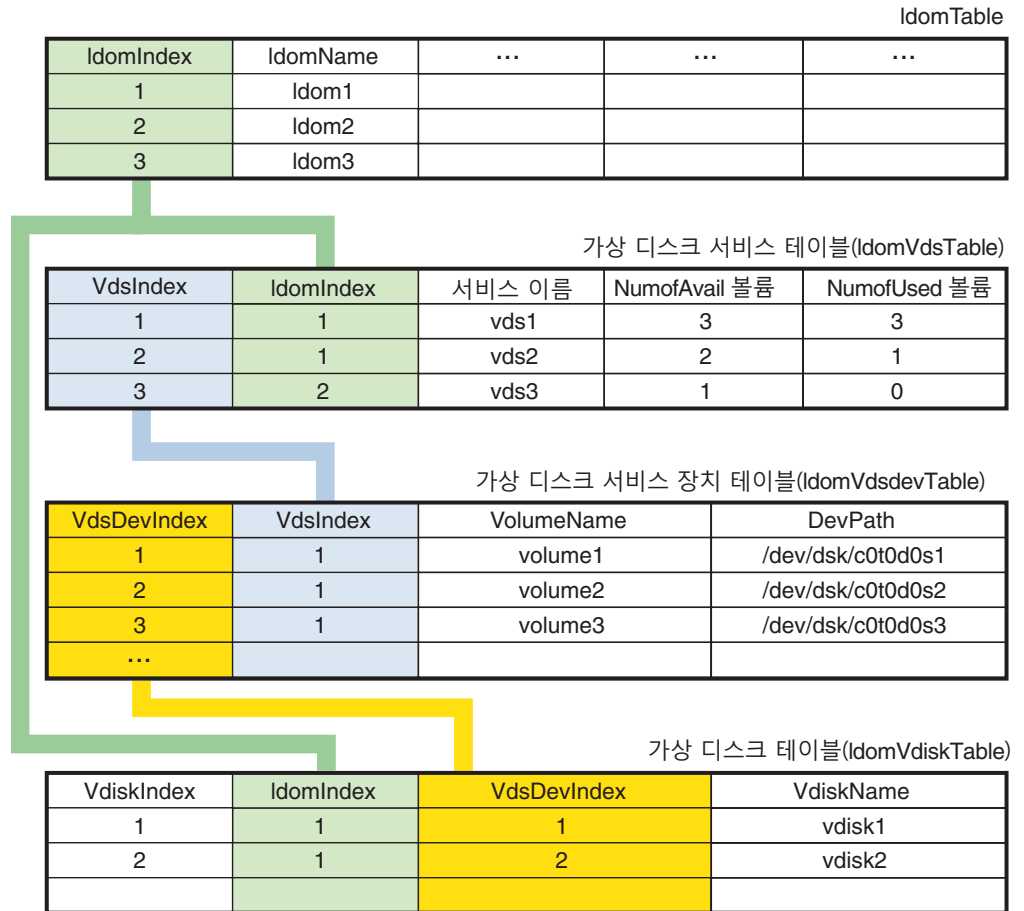
표 15-14 가상 디스크 테이블(ldomVdiskTable)

이름	데이터 유형	액세스	설명
ldomVdiskLdomIndex	정수	읽기 전용	가상 디스크 장치를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomVdiskVdsDevIndex	정수	읽기 전용	가상 디스크 서비스 장치를 나타내는 ldomVdsdevTable의 인덱스로 사용되는 정수
ldomVdiskIndex	정수	액세스할 수 없음	이 테이블에서 가상 디스크를 인덱싱하는 데 사용되는 정수
ldomVdiskName	표시 문자열	읽기 전용	가상 디스크의 이름. 등록 정보 값은 ldm add-vdisk 명령으로 지정된 <i>disk-name</i> 입니다.
ldomVdiskTimeout	정수	읽기 전용	가상 디스크 클라이언트와 가상 디스크 서버 사이의 연결을 설정하기 위한 시간 초과(초)
ldomVdiskID	표시 문자열	읽기 전용	가상 디스크의 식별자

다음 그림은 가상 디스크 테이블과 도메인 테이블 간의 관계를 정의하기 위한 인덱스 사용법을 보여줍니다. 인덱스는 다음과 같이 사용됩니다.

- ldomVdsTable 및 ldomVdiskTable의 ldomIndex가 ldomTable을 가리킵니다.
- ldomVdsdevTable의 VdsIndex가 ldomVdsTable을 가리킵니다.
- ldomVdiskTable의 VdsDevIndex가 ldomVdsdevTable을 가리킵니다.

그림 15-3 가상 디스크 테이블과 도메인 테이블 간의 관계



가상 네트워크 테이블

Oracle VM Server for SPARC 가상 네트워크 지원 덕분에 게스트 도메인은 물리적 이더넷 장치를 통해 외부 호스트와 또는 서로 통신할 수 있습니다. 가상 네트워크는 다음 주요 구성 요소를 포함합니다.

- 가상 스위치(vsw)
- 가상 네트워크 장치(vnet)

서비스 도메인에 가상 스위치를 만든 후 물리적 네트워크 장치를 가상 스위치에 바인딩할 수 있습니다. 그 후에, 통신을 위해 가상 스위치 서비스를 사용하는 도메인에 대해 가상 네트워크 장치를 만들 수 있습니다. 가상 스위치 서비스는 동일한 가상 스위치에 연결하여 다른 도메인과 통신합니다. 물리적 장치가 가상 스위치에 바인딩된 경우 가상 스위치 서비스는 외부 호스트와 통신합니다.

가상 스위치 서비스 테이블(ldomVswTable)

ldomVswTable은 모든 도메인에 대한 가상 스위치 서비스를 설명합니다.

표 15-15 가상 스위치 서비스 테이블(ldomVswTable)

이름	데이터 유형	액세스	설명
ldomVswLdomIndex	정수	읽기 전용	가상 스위치 서비스를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomVswIndex	정수	액세스할 수 없음	이 테이블에서 가상 스위치 장치를 인덱싱하는 데 사용되는 정수
ldomVswServiceName	표시 문자열	읽기 전용	가상 스위치 서비스 이름
ldomVswMacAddress	표시 문자열	읽기 전용	가상 스위치에서 사용된 MAC 주소
ldomVswPhysDevPath	표시 문자열	읽기 전용	가상 네트워크 스위치의 물리적 장치 경로. 물리적 장치가 가상 스위치에 바인딩되지 않은 경우 등록 정보 값이 없습니다.
ldomVswMode	표시 문자열	읽기 전용	클러스터 노드를 실행하려면 값이 mode=sc입니다.
ldomVswDefaultVlanID	표시 문자열	읽기 전용	가상 스위치의 기본 VLAN ID
ldomVswPortVlanID	표시 문자열	읽기 전용	가상 스위치의 포트 VLAN ID
ldomVswVlanID	표시 문자열	읽기 전용	가상 스위치의 VLAN ID
ldomVswLinkprop	표시 문자열	읽기 전용	물리적 네트워크 장치에 기반한 링크 상태를 보고하려면 값이 linkprop=phys-state입니다.
ldomVswMtu	정수	읽기 전용	가상 스위치 장치에 대한 MTU(최대 전송 단위)
ldomVswID	표시 문자열	읽기 전용	가상 스위치 장치의 식별자
ldomVswInterVnetLink	표시 문자열	읽기 전용	vnet 간 통신을 위한 LDC 채널의 지정 상태. 값은 on 또는 off입니다.

가상 네트워크 장치 테이블(ldomVnetTable)

ldomVnetTable은 모든 도메인에 대한 가상 네트워크 장치를 설명합니다.

표 15-16 가상 네트워크 장치 테이블(ldomVnetTable)

이름	데이터 유형	액세스	설명
ldomVnetLdomIndex	정수	읽기 전용	가상 네트워크 장치를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomVnetVswIndex	정수	읽기 전용	가상 스위치 서비스 테이블을 인덱싱하는 데 사용되는 정수
ldomVnetIndex	정수	액세스할 수 없음	이 테이블에서 가상 네트워크 장치를 인덱싱하는 데 사용되는 정수
ldomVnetDevName	표시 문자열	읽기 전용	가상 네트워크 장치 이름. 등록 정보 값은 <code>ldm add-vnet</code> 명령으로 지정된 <code>net-dev</code> 등록 정보입니다.
ldomVnetDevMacAddress	표시 문자열	읽기 전용	이 네트워크 장치의 MAC 주소. 등록 정보 값은 <code>ldm add-vnet</code> 명령으로 지정된 <code>mac-addr</code> 등록 정보입니다.
ldomVnetMode	표시 문자열	읽기 전용	가상 네트워크 장치에 NIU 하이브리드 I/O를 사용하려면 값이 <code>mode=hybrid</code> 입니다.
ldomVnetPortVlanID	표시 문자열	읽기 전용	가상 네트워크 장치의 포트 VLAN ID
ldomVnetVlanID	표시 문자열	읽기 전용	가상 네트워크 장치의 VLAN ID
ldomVnetLinkprop	표시 문자열	읽기 전용	물리적 네트워크 장치에 기반한 링크 상태를 보고하려면 값이 <code>linkprop=phys-state</code> 입니다.
ldomVnetMtu	정수	읽기 전용	가상 네트워크 장치의 MTU
ldomVnetID	표시 문자열	읽기 전용	가상 네트워크 장치의 식별자

가상 콘솔 테이블

Oracle VM Server for SPARC 서비스 도메인은 가상 네트워크 터미널 서비스(vNTS)를 제공합니다. vNTS는 포트 번호 범위 내에서 가상 콘솔 집중기(vcc)라는 가상 콘솔 서비스를 제공합니다. 각 가상 콘솔 집중기에는 여러 콘솔 그룹(vcons)이 있고, 각 그룹에 포트 번호가 지정됩니다. 각 그룹은 여러 도메인을 포함할 수 있습니다.

가상 콘솔 집중기 테이블(ldomVccTable)

ldomVccTable은 모든 도메인에 대한 가상 콘솔 집중기를 설명합니다.

표 15-17 가상 콘솔 집중기 테이블(ldomVccTable)

이름	데이터 유형	액세스	설명
ldomVccLdomIndex	정수	읽기 전용	가상 콘솔 서비스를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomVccIndex	정수	액세스할 수 없음	이 테이블에서 가상 콘솔 집중기를 인덱싱하는 데 사용되는 정수
ldomVccName	표시 문자열	읽기 전용	가상 콘솔 집중기 이름. 등록 정보 값은 ldm add-vcc 명령으로 지정된 vcc-name입니다.
ldomVccPortRangeLow	정수	읽기 전용	가상 콘솔 집중기에서 사용될 TCP 포트 범위의 하한 번호. 등록 정보 값은 ldm add-vcc 명령으로 지정된 port-range의 x부분입니다.
ldomVccPortRangeHigh	정수	읽기 전용	가상 콘솔 집중기에서 사용될 TCP 포트 범위의 상한 번호. 등록 정보 값은 ldm add-vcc 명령으로 지정된 port-range의 y부분입니다.

가상 콘솔 그룹 테이블(ldomVconsTable)

ldomVconsTable은 모든 가상 콘솔 서비스에 대한 가상 콘솔 그룹을 설명합니다.

표 15-18 가상 콘솔 그룹 테이블(ldomVconsTable)

이름	데이터 유형	액세스	설명
ldomVconsIndex	정수	액세스할 수 없음	이 테이블에서 가상 그룹을 인덱싱하는 데 사용되는 정수
ldomVconsGroupName	표시 문자열	읽기 전용	가상 콘솔을 첨부할 그룹 이름. 등록 정보 값은 ldm set-vcons 명령으로 지정된 group입니다.
ldomVconsPortNumber	정수	읽기 전용	이 그룹에 지정된 포트 번호. 등록 정보 값은 ldm set-vcons 명령으로 지정된 port입니다.

가상 콘솔 관계 테이블(ldomVconsVccRelTable)

ldomVconsVccRelTable은 도메인, 가상 콘솔 집중기, 콘솔 그룹 사이의 테이블간 관계를 보여주는 인덱스 값을 포함합니다.

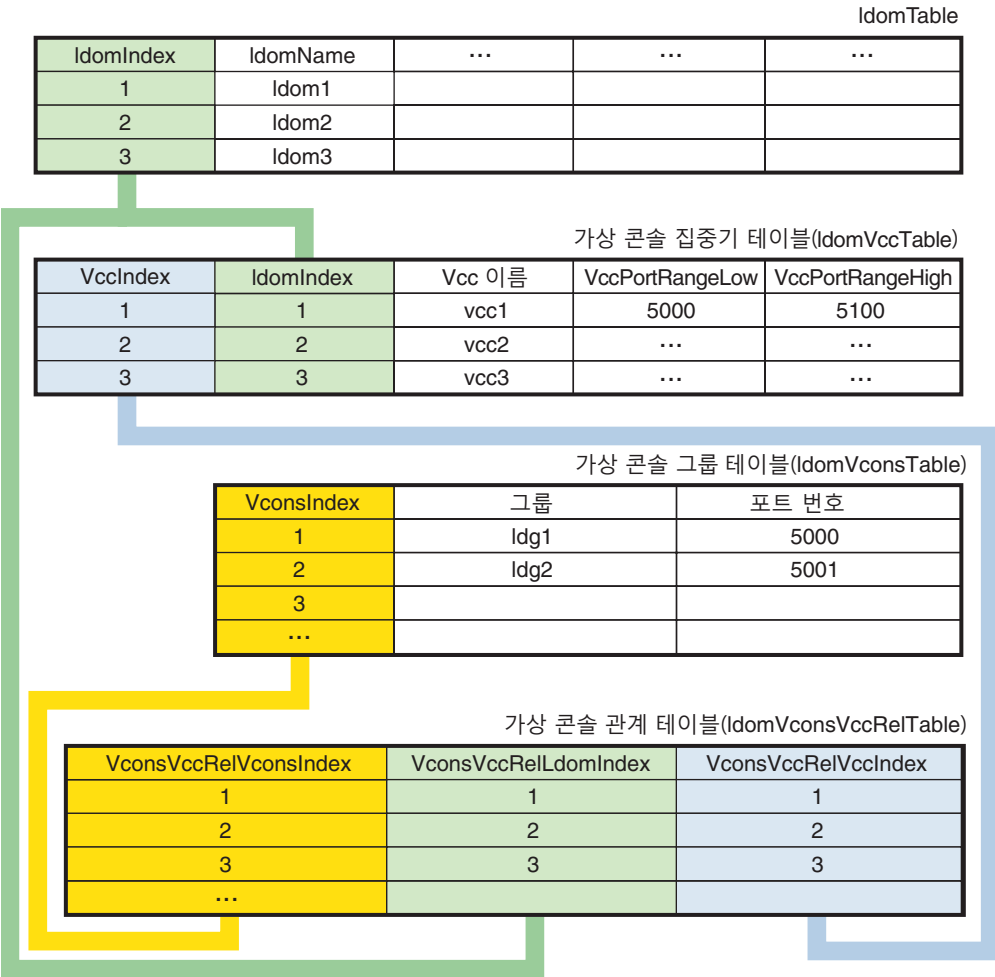
표 15-19 가상 콘솔 관계 테이블(ldomVconsVccRelTable)

이름	데이터 유형	액세스	설명
ldomVconsVccRelVconsIndex	정수	읽기 전용	ldomVconsTable의 ldomVconsIndex 값
ldomVconsVccRelLdomIndex	정수	읽기 전용	ldomTable의 ldomIndex 값
ldomVconsVccRelVccIndex	정수	읽기 전용	ldomVccTable의 ldomVccIndex 값

다음 그림은 가상 콘솔 테이블과 도메인 테이블 간의 관계를 정의하기 위한 인덱스 사용법을 보여줍니다. 인덱스는 다음과 같이 사용됩니다.

- ldomVccTable 및 ldomVconsVccRelTable의 ldomIndex가 ldomTable을 가리킵니다.
- ldomVconsVccRelTable의 VccIndex가 ldomVccTable을 가리킵니다.
- ldomVconsVccRelTable의 VconsIndex가 ldomVconsTable을 가리킵니다.

그림 15-4 가상 콘솔 테이블과 도메인 테이블 간의 관계



암호화 장치 테이블(IdomCryptoTable)

IdomCryptoTable은 모든 도메인이 사용하는 암호화 장치를 설명합니다. 암호화 장치는 때때로 MAU(모듈식 산술 장치)라고도 합니다.

표 15-20 암호화 장치 테이블(IdomCryptoTable)

이름	데이터 유형	액세스	설명
IdomCryptoIdomIndex	정수	읽기 전용	암호화 장치를 포함하는 도메인을 나타내는 IdomTable의 인덱스로 사용되는 정수

표 15-20 암호화 장치 테이블(ldomCryptoTable) (계속)

이름	데이터 유형	액세스	설명
ldomCryptoIndex	정수	액세스할 수 없음	이 테이블에서 암호화 장치를 인덱싱하는 데 사용되는 정수
ldomCryptoCpuSet	표시 문자열	읽기 전용	MAU-unit cpuset에 매핑된 CPU 목록. 예를 들어, {0, 1, 2, 3}입니다.

I/O 버스 테이블(ldomIOBusTable)

ldomIOBusTable은 모든 도메인이 사용하는 물리적 I/O 장치 및 PCI 버스를 설명합니다.

표 15-21 I/O 버스 테이블(ldomIOBusTable)

이름	데이터 유형	액세스	설명
ldomIOBusLdomIndex	정수	읽기 전용	I/O 버스를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomIOBusIndex	정수	액세스할 수 없음	이 테이블에서 I/O 버스를 인덱싱하는 데 사용되는 정수
ldomIOBusName	표시 문자열	읽기 전용	물리적 I/O 장치 이름
ldomIOBusPath	표시 문자열	읽기 전용	물리적 I/O 장치 경로
ldomIOBusOptions	표시 문자열	읽기 전용	물리적 I/O 장치 옵션

코어 테이블(ldomCoreTable)

ldomCoreTable은 모든 도메인에 대한 core-id 및 cpuset와 같은 코어 정보를 설명합니다.

표 15-22 코어 테이블(ldomCoreTable)

이름	데이터 유형	액세스	설명
ldomCoreLdomIndex	정수	읽기 전용	코어를 포함하는 도메인을 나타내는 ldomTable의 인덱스로 사용되는 정수
ldomCoreIndex	정수	액세스할 수 없음	이 테이블에서 코어를 인덱싱하는 데 사용되는 정수
ldomCoreID	표시 문자열	읽기 전용	코어의 식별자(코어 ID)
ldomCoreCpuSet	표시 문자열	읽기 전용	코어 cpuset에 매핑된 CPU 목록

Logical Domains 버전 정보에 대한 스칼라 변수

Logical Domains Manager 프로토콜은 주 번호와 부 번호로 구성된 Logical Domains 버전을 지원합니다. Oracle VM Server for SPARC MIB에는 Logical Domains 버전 정보를 설명하는 스칼라 변수가 있습니다.

표 15-23 Logical Domains 버전 정보에 대한 스칼라 변수

이름	데이터 유형	액세스	설명
ldomVersionMajor	정수	읽기 전용	주 버전 번호
ldomVersionMinor	정수	읽기 전용	부 버전 번호

ldomVersionMajor 및 ldomVersionMinor의 값은 `ldm list -p` 명령으로 표시된 버전에 해당합니다. 예를 들어, 다음과 같습니다.

```
$ ldm ls -p
VERSION 1.6
...

$ snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomVersionMajor.0
SUN-LDOM-MIB::ldomVersionMajor.0 = INTEGER: 1

$ snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomVersionMinor.0
SUN-LDOM-MIB::ldomVersionMinor.0 = INTEGER: 5
```

SNMP 트랩 사용

이 절에서는 트랩을 송/수신하도록 시스템을 설정하는 방법을 설명합니다. 또한 논리적 도메인의 변경 통지를 수신하는 데 사용할 수 있는 트랩과 기타 사용 가능한 트랩을 설명합니다.

Oracle VM Server for SPARC MIB 모듈 트랩 사용

트랩 전송 및 수신

▼ 트랩을 전송하는 방법

- 트랩을 구성합니다.

/etc/sma/snmp/snmpd.conf 파일을 편집하여 트랩, 정보 버전, 대상을 정의하는 지시어를 추가합니다.

```
trapcommunity string --> define community string to be used when sending traps
trapsink host[community [port]] --> to send v1 traps
trap2sink host[community [port]] --> to send v2c traps
informsink host[community [port]] --> to send informs
```

자세한 내용은 `snmpd.conf(4)` 매뉴얼 페이지를 참조하십시오.

예 15-4 SNMP v1 및 v2c 트랩 전송

이 예제는 v1 및 v2c 트랩을 동일한 호스트에서 실행되는 SNMP 트랩 데몬으로 전송합니다. 다음 지시어로 `/etc/sma/snmp/snmpd.conf` 파일이 업데이트됩니다.

```
trapcommunity public
trapsink localhost
trap2sink localhost
```

▼ 트랩을 수신하는 방법

● SNMP 트랩 데몬 유틸리티를 시작합니다.

출력 형식 옵션에 대한 내용은 `snmptrapd(1M)` 매뉴얼 페이지를 참조하십시오.

`snmptrapd` 유틸리티는 SNMP TRAP 메시지를 수신하고 기록하는 SNMP 응용 프로그램입니다. 예를 들어, 다음 `snmptrapd` 명령은 이름 `ldg2`로(`ldomName = ldg2`) 새 도메인이 만들어졌음을(`ldomTrapDesc = Ldom Created`) 보여줍니다.

```
# /usr/sfw/sbin/snmptrapd -P -F \
"TRAP from %B on %m/%L/%y at %h:%j:%k Enterprise=%N Type=%w SubType=%q \nwith Varbinds: %v \nSecurity info:%P\n\
localhost:162
TRAP from localhost on 5/18/2007 at 16:30:10 Enterprise=. Type=0 SubType=0
with Varbinds: DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (47105)
0:07:51.05 SNMPv2-MIB::snmpTrapOID.0 = OID: SUN-LDOM-MIB::ldomCreate
SUN-LDOM-MIB::ldomIndexNotif = INTEGER: 3 SUN-LDOM-MIB::ldomName = STRING: ldg2
SUN-LDOM-MIB::ldomTrapDesc = STRING: Ldom Created
Security info:TRAP2, SNMP v2c, community public
```

Oracle VM Server for SPARC MIB 트랩 설명

이 절에서는 사용할 수 있는 Oracle VM Server for SPARC MIB 트랩을 설명합니다.

도메인 만들기(LdomCreate)

이 트랩은 도메인이 만들어질 때 알림을 보냅니다.

표 15-24 도메인 만들기 트랩(LdomCreate)

이름	데이터 유형	설명
<code>ldomIndexNotif</code>	정수	<code>ldomTable</code> 의 인덱스
<code>ldomName</code>	표시 문자열	도메인의 이름
<code>ldomTrapDesc</code>	표시 문자열	트랩의 설명

도메인 삭제(IdomDestroy)

이 트랩은 도메인이 삭제될 때 알림을 보냅니다.

표 15-25 도메인 삭제 트랩(IdomDestroy)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	표시 문자열	도메인의 이름
ldomTrapDesc	표시 문자열	트랩의 설명

도메인 상태 변경(IdomStateChange)

이 트랩은 도메인 작동 상태 변경을 알립니다.

표 15-26 도메인 상태 변경 트랩(IdomStateChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	표시 문자열	도메인의 이름
ldomOperState	정수	도메인의 새 상태
ldomStatePrev	정수	도메인의 이전 상태
ldomTrapDesc	표시 문자열	트랩의 설명

가상 CPU 변경(IdomVCpuChange)

이 트랩은 도메인의 가상 CPU 수가 변경될 때 알림을 보냅니다.

표 15-27 도메인 가상 CPU 변경 트랩(IdomVCpuChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	표시 문자열	가상 CPU를 포함하는 도메인의 이름
ldomNumVCPU	정수	도메인의 새 가상 CPU 수
ldomNumVCPUPrev	정수	도메인의 이전 가상 CPU 수
ldomTrapDesc	표시 문자열	트랩의 설명

가상 메모리 변경(IdomVMemChange)

이 트랩은 도메인의 가상 메모리 양이 변경될 때 알림을 보냅니다.

표 15-28 도메인 가상 메모리 변경 트랩(IdomVMemChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	표시 문자열	가상 메모리를 포함하는 도메인의 이름
ldomMemSize	정수	도메인의 가상 메모리 양
ldomMemSizePrev	정수	도메인의 이전 가상 메모리 양
ldomMemUnit	정수	가상 메모리의 메모리 단위로, 다음 중 하나입니다. <ul style="list-style-type: none"> ■ 1 - KB ■ 2 - MB ■ 3 - GB ■ 4 - 바이트 지정되지 않은 경우 단위 값은 바이트입니다.
ldomMemUnitPrev	정수	이전 가상 메모리의 메모리 단위로, 다음 중 하나입니다. <ul style="list-style-type: none"> ■ 1 - KB ■ 2 - MB ■ 3 - GB ■ 4 - 바이트 지정되지 않은 경우 단위 값은 바이트입니다.
ldomTrapDesc	표시 문자열	트랩의 설명

가상 디스크 서비스 변경(IdomVdsChange)

이 트랩은 도메인의 가상 디스크 서비스가 변경될 때 알림을 보냅니다.

표 15-29 도메인 가상 디스크 서비스 변경 트랩(IdomVdsChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	표시 문자열	가상 디스크 서비스를 포함하는 도메인의 이름
ldomVdsServiceName	표시 문자열	변경된 가상 디스크 서비스의 이름

표 15-29 도메인 가상 디스크 서비스 변경 트랩(ldomVdsChange) (계속)

이름	데이터 유형	설명
ldomChangeFlag	정수	다음과 같이 가상 디스크 서비스에 발생한 변경 사항 중 하나를 나타냅니다. <ul style="list-style-type: none"> ■ 1 - 추가됨 ■ 2 - 수정됨 ■ 3 - 제거됨
ldomTrapDesc	포시 문자열	트랩의 설명

가상 디스크 변경(ldomVdiskChange)

이 트랩은 도메인의 가상 디스크가 변경될 때 알림을 보냅니다.

표 15-30 가상 디스크 변경 트랩(ldomVdiskChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	포시 문자열	가상 디스크 장치를 포함하는 도메인의 이름
ldomVdiskName	포시 문자열	변경된 가상 디스크 장치의 이름
ldomChangeFlag	정수	다음과 같이 가상 디스크 서비스에 발생한 변경 사항 중 하나를 나타냅니다. <ul style="list-style-type: none"> ■ 1 - 추가됨 ■ 2 - 수정됨 ■ 3 - 제거됨
ldomTrapDesc	포시 문자열	트랩의 설명

가상 스위치 변경(ldomVswChange)

이 트랩은 도메인의 가상 스위치가 변경될 때 알림을 보냅니다.

표 15-31 가상 스위치 변경 트랩(ldomVswChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	포시 문자열	가상 스위치 서비스를 포함하는 도메인의 이름
ldomVswServiceName	포시 문자열	변경된 가상 스위치 서비스의 이름

표 15-31 가상 스위치 변경 트랩(ldomVswChange) (계속)

이름	데이터 유형	설명
ldomChangeFlag	정수	다음과 같이 가상 스위치 서비스에 발생한 변경 사항 중 하나를 나타냅니다. <ul style="list-style-type: none"> ■ 1 - 추가됨 ■ 2 - 수정됨 ■ 3 - 제거됨
ldomTrapDesc	표시 문자열	트랩의 설명

가상 네트워크 변경(ldomVnetChange)

이 트랩은 도메인의 가상 네트워크가 변경될 때 알림을 보냅니다.

표 15-32 가상 네트워크 변경 트랩(ldomVnetChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	표시 문자열	가상 네트워크 장치를 포함하는 도메인의 이름
ldomVnetDevName	표시 문자열	도메인의 가상 네트워크 장치 이름
ldomChangeFlag	정수	다음과 같이 가상 디스크 서비스에 발생한 변경 사항 중 하나를 나타냅니다. <ul style="list-style-type: none"> ■ 1 - 추가됨 ■ 2 - 수정됨 ■ 3 - 제거됨
ldomTrapDesc	표시 문자열	트랩의 설명

가상 콘솔 집중기 변경(ldomVccChange)

이 트랩은 도메인의 가상 콘솔 집중기가 변경될 때 알림을 보냅니다.

표 15-33 가상 콘솔 집중기 변경 트랩(ldomVccChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	표시 문자열	가상 콘솔 집중기를 포함하는 도메인의 이름
ldomVccName	표시 문자열	변경된 가상 콘솔 집중기 서비스의 이름

표 15-33 가상 콘솔 집중기 변경 트랩(ldomVccChange) (계속)		
이름	데이터 유형	설명
ldomChangeFlag	정수	다음과 같이 가상 콘솔 집중기에 발생한 변경 사항 중 하나를 나타냅니다. <ul style="list-style-type: none"> ■ 1 - 추가됨 ■ 2 - 수정됨 ■ 3 - 제거됨
ldomTrapDesc	표시 문자열	트랩의 설명

가상 콘솔 그룹 변경(ldomVconsChange)

이 트랩은 도메인의 가상 콘솔 그룹이 변경될 때 알림을 보냅니다.

표 15-34 가상 콘솔 그룹 변경 트랩(ldomVconsChange)

이름	데이터 유형	설명
ldomIndexNotif	정수	ldomTable의 인덱스
ldomName	표시 문자열	가상 콘솔 그룹을 포함하는 도메인의 이름
ldomVconsGroupName	표시 문자열	변경된 가상 콘솔 그룹의 이름
ldomChangeFlag	정수	다음과 같이 가상 콘솔 그룹에 발생한 변경 사항 중 하나를 나타냅니다. <ul style="list-style-type: none"> ■ 1 - 추가됨 ■ 2 - 수정됨 ■ 3 - 제거됨
ldomTrapDesc	표시 문자열	트랩의 설명

도메인 시작 및 중지

이 절에서는 도메인 시작/중지에 사용할 활성화 관리 작업을 설명합니다. 도메인 테이블 ldomTable의 ldomAdminState 등록 정보 값을 설정하여 이러한 활성화 관리 작업을 제어할 수 있습니다. [표 15-1](#)을 참조하십시오.

도메인 시작 및 중지

▼ 도메인을 시작하는 방법

이 절차는 기존의 바인딩된 도메인을 시작하는 방법을 설명합니다. 지정된 도메인 이름을 가진 도메인이 존재하지 않거나 아직 바인딩되지 않은 경우 이 작업을 실패합니다.

1 *domain-name* 도메인이 존재하고 바인딩되었는지 확인합니다.

```
# ldm list domain-name
```

2 *ldmTable*의 *domain-name*을 식별합니다.

```
# snmpwalk -v1 -c public localhost SUN-LDOM-MIB::ldmTable
SUN-LDOM-MIB::ldmName.1 = STRING: primary
SUN-LDOM-MIB::ldmName.2 = STRING: LdomMibTest_1
SUN-LDOM-MIB::ldmAdminState.1 = INTEGER: 0
SUN-LDOM-MIB::ldmAdminState.2 = INTEGER: 0
SUN-LDOM-MIB::ldmOperState.1 = INTEGER: active(1)
SUN-LDOM-MIB::ldmOperState.2 = INTEGER: bound(6)
SUN-LDOM-MIB::ldmNumVCpu.1 = INTEGER: 32
SUN-LDOM-MIB::ldmNumVCpu.2 = INTEGER: 2
SUN-LDOM-MIB::ldmMemSize.1 = INTEGER: 3968
SUN-LDOM-MIB::ldmMemSize.2 = INTEGER: 256
SUN-LDOM-MIB::ldmMemUnit.1 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldmMemUnit.2 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldmNumCrypto.1 = INTEGER: 8
SUN-LDOM-MIB::ldmNumCrypto.2 = INTEGER: 0
SUN-LDOM-MIB::ldmNumIOBus.1 = INTEGER: 2
SUN-LDOM-MIB::ldmNumIOBus.2 = INTEGER: 0
SUN-LDOM-MIB::ldmUUID.1 = STRING: c2c3d93b-a3f9-60f6-a45e-f35d55c05fb6
SUN-LDOM-MIB::ldmUUID.2 = STRING: af0b05f0-d262-e633-af32-a6c4e81fb81c
SUN-LDOM-MIB::ldmMacAddress.1 = STRING: 00:14:4f:86:63:2a
SUN-LDOM-MIB::ldmMacAddress.2 = STRING: 00:14:4f:fa:78:b9
SUN-LDOM-MIB::ldmHostID.1 = STRING: 0x8486632a
SUN-LDOM-MIB::ldmHostID.2 = STRING: 0x84fa78b9
SUN-LDOM-MIB::ldmFailurePolicy.1 = STRING: ignore
SUN-LDOM-MIB::ldmFailurePolicy.2 = STRING: ignore
SUN-LDOM-MIB::ldmMaster.1 = STRING:
SUN-LDOM-MIB::ldmMaster.2 = STRING:
```

3 *domain-name* 도메인을 시작합니다.

snmpset 명령에서 *ldmAdminState* 등록 정보에 1 값을 설정하여 도메인을 시작합니다.
*n*은 시작할 도메인을 지정합니다.

```
# snmpset -v version -c community-string hostname \
SUN-LDOM-MIB::ldmTable.1.ldmAdminState.n = 1
```

4 *domain-name* 도메인이 활성화인지 확인합니다.

■ *ldm list* 명령을 사용합니다.

```
# ldm list domain-name
```

■ *snmpget* 명령을 사용합니다.

```
# snmpget -v version -c community-string hostname SUN-LDOM-MIB::ldmOperState.n
```

예 15-5 게스트 도메인 시작

이 예제는 *ldmAdminState* 등록 정보를 1로 설정하기 전에 *LdomMibTest_1* 도메인이 존재하고 바인딩되었는지 확인합니다. 마지막으로, *ldm list LdomMibTest_1* 명령은 *LdomMibTest_1* 도메인이 활성화인지 확인합니다.

```
# ldm list LdomMibTest_1
# snmpset -v1 -c private localhost SUN-LDOM-MIB::ldomTable.1.ldomAdminState.2 = 1
# ldm list LdomMibTest_1
```

LdomMibTest_1 도메인의 상태를 검색하기 위해 `ldm list` 명령을 사용하는 대신, `snmpget` 명령을 사용할 수 있습니다.

```
# snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomOperState.2
```

도메인을 시작하기 위해 `snmpset`를 사용할 때 도메인이 비활성인 경우 먼저 도메인이 바인딩된 후 시작됩니다.

▼ 도메인을 중지하는 방법

이 절차는 시작된 도메인을 중지하는 방법을 설명합니다. 도메인에서 호스트하는 모든 운영 체제 인스턴스가 중지됩니다.

1 ldomTable의 domain-name을 식별합니다.

```
# snmpwalk -v1 -c public localhost SUN-LDOM-MIB::ldomTable
SUN-LDOM-MIB::ldomName.1 = STRING: primary
SUN-LDOM-MIB::ldomName.2 = STRING: LdomMibTest_1
SUN-LDOM-MIB::ldomAdminState.1 = INTEGER: 0
SUN-LDOM-MIB::ldomAdminState.2 = INTEGER: 0
SUN-LDOM-MIB::ldomOperState.1 = INTEGER: active(1)
SUN-LDOM-MIB::ldomOperState.2 = INTEGER: bound(6)
SUN-LDOM-MIB::ldomNumVCpu.1 = INTEGER: 32
SUN-LDOM-MIB::ldomNumVCpu.2 = INTEGER: 2
SUN-LDOM-MIB::ldomMemSize.1 = INTEGER: 3968
SUN-LDOM-MIB::ldomMemSize.2 = INTEGER: 256
SUN-LDOM-MIB::ldomMemUnit.1 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomMemUnit.2 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomNumCrypto.1 = INTEGER: 8
SUN-LDOM-MIB::ldomNumCrypto.2 = INTEGER: 0
SUN-LDOM-MIB::ldomNumIOBus.1 = INTEGER: 2
SUN-LDOM-MIB::ldomNumIOBus.2 = INTEGER: 0
SUN-LDOM-MIB::ldomUUID.1 = STRING: c2c3d93b-a3f9-60f6-a45e-f35d55c05fb6
SUN-LDOM-MIB::ldomUUID.2 = STRING: af0b05f0-d262-e633-af32-a6c4e81fb81c
SUN-LDOM-MIB::ldomMacAddress.1 = STRING: 00:14:4f:86:63:2a
SUN-LDOM-MIB::ldomMacAddress.2 = STRING: 00:14:4f:fa:78:b9
SUN-LDOM-MIB::ldomHostID.1 = STRING: 0x8486632a
SUN-LDOM-MIB::ldomHostID.2 = STRING: 0x84fa78b9
SUN-LDOM-MIB::ldomFailurePolicy.1 = STRING: ignore
SUN-LDOM-MIB::ldomFailurePolicy.2 = STRING: ignore
SUN-LDOM-MIB::ldomMaster.1 = STRING:
SUN-LDOM-MIB::ldomMaster.2 = STRING:
```

2 domain-name 도메인을 중지합니다.

`snmpset` 명령에서 `ldomAdminState` 등록 정보에 2 값을 설정하여 도메인을 중지합니다. `n`은 중지할 도메인을 지정합니다.

```
# snmpset -v version -c community-string hostname \
SUN-LDOM-MIB::ldomTable.1.ldomAdminState.n = 2
```

3 *domain-name* 도메인이 바인딩되었는지 확인합니다.

- `ldm list` 명령을 사용합니다.
`# ldm list domain-name`
- `snmpget` 명령을 사용합니다.
`# snmpget -v version -c community-string hostname SUN-LDOM-MIB::ldomOperState.n`

예 15-6 게스트 도메인 중지

이 예제는 `ldomAdminState` 등록 정보를 2로 설정하여 게스트 도메인을 중지하고 `ldm list LdomMibTest_1` 명령을 사용하여 `LdomMibTest_1` 도메인이 바인딩되었는지 확인합니다.

```
# snmpset -v1 -c private localhost SUN-LDOM-MIB::ldomTable.1.ldomAdminState.2 = 2
# ldm list LdomMibTest_1
```


Logical Domains Manager 검색

멀티캐스트 메시지를 사용하여 서브넷에서 Logical Domains Manager를 검색할 수 있습니다. `ldmd` 데몬은 특정 멀티캐스트 패킷을 네트워크에서 수신할 수 있습니다. 해당 멀티캐스트 메시지가 특정 유형인 경우 `ldmd`이 호출자에게 회신합니다. 이렇게 하면 Oracle VM Server for SPARC를 실행 중인 시스템에서 `ldmd`를 검색할 수 있습니다.

이 장에서는 서브넷의 시스템에 실행 중인 Logical Domains Manager 검색에 대한 정보를 제공합니다.

Logical Domains Manager를 실행 중인 시스템 검색

멀티캐스트 통신

이 검색 방식은 `ldmd` 데몬에서 사용된 것과 동일한 멀티캐스트 네트워크를 사용하여 MAC 주소를 자동으로 지정할 때 충돌을 감지합니다. 멀티캐스트 소켓을 구성하려면 다음 정보를 제공해야 합니다.

```
#define MAC_MULTI_PORT      64535
#define MAC_MULTI_GROUP     "239.129.9.27"
```

기본적으로, 멀티캐스트 패킷만 컴퓨터가 연결된 서브넷에 전송할 수 있습니다. `ldmd` 데몬에 `ldmd/hops` SMF 등록 정보를 설정하여 동작을 변경할 수 있습니다.

메시지 형식

검색 메시지는 다른 메시지와 혼동되지 않도록 명확히 표시해야 합니다. 다음 멀티캐스트 메시지 형식은 검색 수신 프로세스에서 검색 메시지를 구별할 수 있습니다.

```
#include <netdb.h> /* Used for MAXHOSTNAMELEN definition */
#define MAC_MULTI_MAGIC_NO 92792004
#define MAC_MULTI_VERSION 1
```

```
enum {
    SEND_MSG = 0,
    RESPONSE_MSG,
    LDMD_DISC_SEND,
    LDMD_DISC_RESP,
};

typedef struct {
    uint32_t    version_no;
    uint32_t    magic_no;
    uint32_t    msg_type;
    uint32_t    resv;
    union {
        mac_lookup_t    Mac_lookup;
        ldmd_discovery_t    Ldmd_discovery;
    } payload;
#define    lookup        payload.Mac_lookup
#define    discovery    payload.Ldmd_discovery
} multicast_msg_t;

#define    LDMD_VERSION_LEN    32

typedef struct {
    uint64_t    mac_addr;
    char        source_ip[INET_ADDRSTRLEN];
} mac_lookup_t;

typedef struct {
    char        ldmd_version[LDMD_VERSION_LEN];
    char        hostname[MAXHOSTNAMELEN];
    struct in_addr    ip_address;
    int        port_no;
} ldmd_discovery_t;
```

▼ 서브넷에서 실행 중인 Logical Domains Manager를 검색하는 방법

1 멀티캐스트 소켓을 엽니다.

309 페이지 “멀티캐스트 통신”에 지정된 포트 및 그룹 정보를 사용해야 합니다.

2 소켓을 통해 `multicast_msg_t` 메시지를 전송합니다.

다음은 메시지에 포함해야 합니다.

- `version_no`의 유효한 값 - `MAC_MULTI_VERSION`에 정의된 대로 1입니다.
- `magic_no`의 유효한 값 - `MAC_MULTI_MAGIC_NO`에 정의된 대로 92792004입니다.
- `msg_type`을 `LDMD_DISC_SEND`로 설정

3 Logical Domains Manager의 응답을 멀티캐스트 소켓에서 수신합니다.

다음에 포함된 `multicast_msg_t` 메시지로 응답해야 합니다.

- version_no의 유효한 값
- magic_no의 유효한 값
- msg_type을 LDMD_DISC_RESP로 설정
- ldmd_discovery_t 구조로 구성된 페이로드. 다음 정보를 포함합니다.
 - ldmd_version - 시스템에 실행 중인 Logical Domains Manager의 버전
 - hostname - 시스템의 호스트 이름
 - ip_address - 시스템의 IP 주소
 - port_no - 통신을 위해 Logical Domains Manager에서 사용 중인 포트 번호. XMPP 포트 6482여야 합니다.

Logical Domains Manager에서 응답을 수신할 때 auto-allocation MAC collision-detection 패킷이 폐기되었는지 확인합니다.

Logical Domains Manager에서 XML 인터페이스 사용

이 장에서는 외부 사용자 프로그램이 Oracle VM Server for SPARC 소프트웨어와 연결하는 데 사용할 수 있는 XML(확장성 마크업 언어) 통신 방식에 대해 설명합니다. 다음 기본 항목을 다룹니다.

- 313 페이지 “XML 전송”
- 314 페이지 “XML 프로토콜”
- 318 페이지 “이벤트 메시지”
- 323 페이지 “Logical Domains Manager 작업”
- 324 페이지 “Logical Domains Manager 리소스 및 등록 정보”
- 337 페이지 “XML 스키마”

XML 전송

외부 프로그램은 확장성 메시징 및 프레즌스 프로토콜(XMPP – RFC 3920)을 사용하여 Logical Domains Manager와 통신할 수 있습니다. XMPP는 로컬 연결과 원격 연결 모두에 대해 지원되며, 기본적으로 설정되어 있습니다. 원격 연결을 해제하려면 `ldmd/xmpp_enabled` SMF 등록 정보를 `false`로 설정하고 Logical Domains Manager를 다시 시작하십시오.

```
# svccfg -s ldom/ldmd setprop ldmd/xmpp_enabled=false
# svcadm refresh ldmd
# svcadm restart ldmd
```

주 - XMPP 서버를 사용 안함으로 설정하면 도메인 마이그레이션 및 메모리 동적 재구성도 수행되지 않습니다.

XMPP 서버

Logical Domains Manager는 사용 가능한 여러 XMPP 클라이언트 응용 프로그램과 라이브러리와 통신할 수 있는 XMPP 서버를 구현합니다. Logical Domains Manager에서 사용하는 보안 방식은 다음과 같습니다.

- 클라이언트와 자신과의 통신 채널에 보안을 설정하는 TLS(전송 계층 보안)
- 인증용 SASL(Simple Authentication and Security Layer). PLAIN은 유일하게 지원되는 SASL 방식입니다. 모니터링 또는 관리 작업을 허용하기 전에 사용자를 인증할 수 있도록 사용자 이름과 암호를 제공해야 합니다.

로컬 연결

Logical Domains Manager는 사용자 클라이언트가 자신과 동일한 도메인에서 실행 중인지 여부를 감지하여, 그럴 경우 해당 클라이언트와 최소한의 XMPP 핸드셰이크를 수행합니다. 특히 TLS를 통해 보안 채널을 설정한 후 SASL 인증 단계를 건너뛸 수 있습니다. 인증 및 권한 부여는 클라이언트 인터페이스를 구현하는 프로세스의 자격 증명을 기준으로 수행됩니다.

클라이언트는 전체 XMPP 클라이언트를 구현하거나 단순히 libxml2 SAX(Simple API for XML) 구문 분석기와 같은 스트리밍 XML 구문 분석기를 실행할 수 있습니다. 어떤 방식을 사용하든지 클라이언트는 TLS 협상 지점에 대해 XMPP 핸드셰이크를 처리해야 합니다. 필요한 순서는 XMPP 사양을 참조하십시오.

XML 프로토콜

통신 초기화가 완료되면 Logical Domains에서 정의한 XML 메시지가 전송됩니다. 다음 두 가지 유형의 XML 메시지가 있습니다.

- 요청 및 응답 메시지는 <LDM_interface> 태그를 사용합니다. 이 유형의 XML 메시지는 CLI(명령줄 인터페이스)를 사용하여 명령을 실행하는 것과 비슷하게, 명령을 전달하고 Logical Domains Manager에서 다시 결과를 가져오는 데 사용됩니다. 이 태그는 이벤트 등록 및 등록 해제에도 사용됩니다.
- 이벤트 메시지는 <LDM_event> 태그를 사용합니다. 이 유형의 XML 메시지는 Logical Domains Manager에서 게시한 이벤트를 비동기적으로 보고하는 데 사용됩니다.

요청 및 응답 메시지

Logical Domains에 대한 XML 인터페이스는 두 가지 형식입니다.

- 한 형식은 명령을 Logical Domains Manager로 전송하는 데 사용됩니다.
- 다른 형식은 Logical Domains Manager가 수신 메시지의 상태 및 해당 메시지 내에서 요청된 작업에 대해 응답하는 데 사용됩니다.

이 두 형식은 여러 가지 공통 XML 구조를 공유하지만, 두 형식의 차이점을 보다 잘 이해할 수 있도록 여기서는 개별적으로 설명합니다.

요청 메시지

가장 기본적인 레벨의 Logical Domains Manager로 들어오는 XML 요청에는 단일 객체에 대해 작동하는 단일 명령에 대한 설명이 포함됩니다. 보다 복잡한 요청은 여러 개의 명령과 명령당 여러 개의 객체를 처리할 수 있습니다. 기본 XML 명령의 구조는 다음과 같습니다.

예 17-1 단일 객체에 대해 작동하는 단일 명령의 형식

```
<LDM_interface version="1.3">
  <cmd>
    <action>Place command here</action>
    <option>Place options for certain commands here</option>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content> -->
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">Property Value</gprop:GenericProperty>
            </Item>
          </Section>
          <!-- Note: More Sections sections can be placed here -->
        </Content>
      </Envelope>
    </data>
    <!-- Note: More Data sections can be placed here -->
  </cmd>
  <!-- Note: More Commands sections can be placed here -->
</LDM_interface>
```

<LDM_interface> 태그

Logical Domains Manager로 전송되는 모든 명령은 <LDM_interface> 태그로 시작해야 합니다. Logical Domains Manager로 전송되는 모든 문서는 <LDM_interface> 태그를 한 개만 포함해야 합니다. <LDM_interface> 태그에는 예 17-1에 표시된 것과 같이 버전 속성이 포함되어야 합니다.

<cmd> 태그

<LDM_interface> 태그 내에서 문서는 <cmd> 태그를 적어도 한 개 포함해야 합니다. 각 <cmd> 섹션에는 <action> 태그가 한 개만 있어야 합니다. <action> 태그는 실행할 명령을 설명하는 데 사용됩니다. 각 <cmd> 태그는 명령이 작동할 객체를 설명하는 <data> 태그를 적어도 한 개 포함해야 합니다.

<cmd> 태그는 또한 일부 명령과 연관된 옵션과 플래그에 사용되는 <option> 태그도 포함할 수 있습니다. 옵션을 사용하는 명령은 다음과 같습니다.

- remove-domain 명령은 -a 옵션을 사용할 수 있습니다.
- stop-domain 명령은 -f 옵션을 사용할 수 있습니다.
- cancel-operation 명령은 migration 또는 reconf 옵션을 사용할 수 있습니다.
- add-spconfig 명령은 -r autosave-name 옵션을 사용할 수 있습니다.
- remove-spconfig 명령은 -r 옵션을 사용할 수 있습니다.
- list-spconfig 명령은 -r [autosave-name] 옵션을 사용할 수 있습니다.

<data> 태그

각 <data> 섹션은 지정된 명령과 관련된 객체에 대한 설명을 포함할 수 있습니다. 데이터 섹션의 형식은 OVF(Open Virtualization Format) 드래프트 사양의 XML 스키마 부분을 토대로 합니다. 해당 스키마는 <References> 태그(Logical Domains에서 사용하지 않음)와 <Content> 및 <Section> 섹션을 포함하는 <Envelope> 섹션을 정의합니다.

Logical Domains의 경우, <Content> 섹션은 특정 도메인을 식별하고 설명하는 데 사용됩니다. <Content> 노드의 id= 속성에 있는 도메인 이름으로 도메인을 식별합니다. <Content> 섹션 내에는 특정 명령에 필요한 도메인의 리소스를 설명하는 하나 이상의 <Section> 섹션이 있습니다.

도메인 이름만 식별해야 하는 경우에는 <Section> 태그를 사용할 필요가 없습니다. 반대로, 명령에 도메인 식별자가 필요하지 않은 경우 명령에 필요한 리소스를 설명하는 <Section> 섹션을 <Content> 섹션 외부에 제공할 필요는 없지만, <Envelope> 섹션 내에는 제공해야 합니다.

객체 정보를 추론할 수 있는 경우 <data> 섹션은 <Envelope> 태그를 포함할 필요가 없습니다. 이러한 상황은 작업에 해당하는 모든 객체를 모니터링하는 요청, 이벤트 등록 및 등록 해제 요청에 주로 적용됩니다.

OVF 사양의 스키마를 사용하여 모든 유형의 객체를 적절하게 정의할 수 있으려면 다음 두 가지의 OVF 유형을 추가로 정의해야 합니다.

- <gprop:GenericProperty> 태그
- <Binding> 태그

<gprop:GenericProperty> 태그는 OVF 사양에 정의가 없는 객체 등록 정보를 처리하기 위해 정의되었습니다. 등록 정보 이름은 노드의 key= 속성에 정의되며 등록 정보 값은 노드 콘텐츠입니다. <binding> 태그는 다른 리소스에 바인드된 리소스를 정의하기 위해 list-bindings 하위 명령 출력에 사용됩니다.

응답 메시지

발신 XML 응답은 포함된 명령과 객체의 관점에서 수신 요청의 구조와 일치하지만, 요청에 대한 전체 <Response> 섹션과 지정된 각 객체와 명령에 대한 <Response> 섹션이 추가됩니다. <Response> 섹션은 예 17-2에 설명된 것과 같이 상태 및 메시지 정보를 제공합니다. 기본 XML 요청에 대한 응답의 구조는 다음과 같습니다.

예 17-2 단일 객체에 대해 작동하는 단일 명령에 대한 응답의 형식

```
<LDM_interface version="1.3">
  <cmd>
    <action>Place command here</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content> -->
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>
                LDom Resource Type
              </rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">
                Property Value
              </gprop:GenericProperty>
            </Item>
          </Section>
          <!-- Note: More <Section> sections can be placed here -->
        </Content>
      </Envelope>
      <response>
        <status>success or failure</status>
        <resp_msg>Reason for failure</resp_msg>
      </response>
    </data>
    <!-- Note: More Data sections can be placed here -->
    <response>
      <status>success or failure</status>
      <resp_msg>Reason for failure</resp_msg>
    </response>
  </cmd>
  <!-- Note: More Command sections can be placed here -->
  <response>
    <status>success or failure</status>
    <resp_msg>Reason for failure</resp_msg>
  </response>
</LDM_interface>
```

전체 응답

<response> 섹션은 <LDM_interface> 섹션의 직속 하위로, 전체 요청의 전반적인 성공 또는 실패를 나타냅니다. 수신 XML 문서의 형식이 잘못된 경우가 아니라면 <response> 섹션에는 <status> 태그만 포함됩니다. 이 응답 상태가 성공을 나타낸다면 모든 객체에 대한 모든 명령이 성공한 것입니다. 이 응답 상태가 실패이고 <resp_msg> 태그가 없는

경우 원래 요청에 포함된 명령 중 하나가 실패한 것입니다. <resp_msg> 태그는 XML 문서 자체와 관련된 문제를 설명하는 데만 사용됩니다.

명령 응답

<cmd> 섹션 아래에 있는 <response> 섹션은 사용자에게 특정 명령의 성공이나 실패에 대해 알립니다. <status> 태그는 명령의 성공 또는 실패 여부를 표시합니다. 전체 응답의 경우와 마찬가지로, 명령이 실패하면 요청의 <cmd> 섹션 콘텐츠의 형식이 잘못된 경우 <response> 섹션에는 <resp_msg> 태그만 포함됩니다. 그렇지 않은 경우 실패 상태는 명령이 실행된 객체 중 하나에서 오류가 발생했음을 의미합니다.

객체 응답

끝으로, <cmd> 섹션에 있는 각 <data> 섹션에도 <response> 섹션이 있습니다. 이 섹션에는 특정 객체에 대해 실행된 명령이 성공 또는 실패했는지 여부가 표시됩니다. 응답 상태가 SUCCESS일 경우, <response> 섹션에 <resp_msg> 태그가 없습니다. 상태가 FAILURE일 경우, 객체에 대해 명령을 실행 중일 때 발생한 오류에 따라 <response> 필드에 <resp_msg> 태그가 하나 이상 있습니다. 객체 오류는 명령 실행 시 발견된 문제 또는 형식이 잘못되었거나 알 수 없는 객체로 인해 발생할 수 있습니다.

<data> 섹션은 <response> 섹션 이외에 다른 정보를 포함할 수 있습니다. 이 정보는 오류가 발생한 객체를 설명하는 수신 <data> 필드와 같은 형식으로 되어 있습니다. [316 페이지 “<data> 태그”](#)를 참조하십시오. 이 추가 정보는 특히 다음과 같은 경우에 유용합니다.

- 명령이 특정 <data> 섹션에 대해 실패하지만 추가 <data> 섹션에 대해 성공하는 경우
- 빈 <data> 섹션이 명령에 전달되어 어떤 도메인에서는 실패하고 다른 도메인에서는 성공하는 경우

이벤트 메시지

폴링 대신 특정 상태가 변경될 경우 이벤트 알림을 수신하도록 가입할 수 있습니다. 개별적으로 또는 일괄적으로 가입할 수 있는 이벤트 유형에는 세 가지가 있습니다. 자세한 내용은 [320 페이지 “이벤트 유형”](#)을 참조하십시오.

등록 및 등록 해제

<LDM_interface> 메시지를 사용하여 이벤트에 대해 등록할 수 있습니다. [315 페이지 “<LDM_interface> 태그”](#)를 참조하십시오. action 태그는 등록 또는 등록 해제할 이벤트의 유형을 자세히 설명하며 <data> 섹션은 비어 있습니다.

예 17-3 이벤트 등록 요청 메시지 예

```
<LDM_interface version="1.3">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
  </cmd>
</LDM_interface>
```

Logical Domains Manager는 등록 또는 등록 해제가 성공적인지 여부를 나타내는 <LDM_interface> 응답 메시지로 응답합니다.

예 17-4 이벤트 등록 응답 메시지 예

```
<LDM_interface version="1.3">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
      <response>
        <status>success</status>
      </response>
    </data>
    <response>
      <status>success</status>
    </response>
  </cmd>
  <response>
    <status>success</status>
  </response>
</LDM_interface>
```

각 이벤트 유형에 대한 작업 문자열은 이벤트 하위 섹션에 나열됩니다.

<LDM_event> 메시지

이벤트 메시지는 수신 <LDM_interface> 메시지와 같은 형식입니다. 단, 메시지의 시작 태그가 <LDM_event>라는 점만 다릅니다. 메시지의 action 태그는 이벤트 트리거를 위해 수행된 작업입니다. 메시지의 데이터 섹션은 이벤트와 연관된 객체에 대해 설명하는데, 세부 사항은 발생한 이벤트의 유형에 따라 다릅니다.

예 17-5 <LDM_event> 알림 예

```
<LDM_event version='1.1'>
  <cmd>
    <action>Event command here</action>
    <data version='3.0'>
      <Envelope>
        <References/>
        <Content xsi:type='ovf:VirtualSystem_Type' ovf:id='ldg1'/>
        <Section xsi:type="ovf:ResourceAllocationSection_type">
          <Item>
            <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
```

예 17-5 <LDM_event> 알림 예 (계속)

```

        <gprop:GenericProperty
          key="Property name">Property Value</gprop:GenericProperty>
        </Item>
      </Section>
    </Envelope>
  </data>
</cmd>
</LDM_event>

```

이벤트 유형

가입할 수 있는 이벤트 유형은 다음과 같습니다.

- 도메인 이벤트
- 하드웨어 이벤트
- 진행률 이벤트
- 리소스 이벤트

모든 이벤트는 `ldm` 하위 명령에 해당합니다.

도메인 이벤트

도메인 이벤트는 도메인에 대해 직접 수행할 수 있는 작업에 대해 설명합니다. 다음은 <LDM_event> 메시지의 <action> 태그에 지정할 수 있는 도메인 이벤트 목록입니다.

- add-domain
- bind-domain
- domain-reset
- migrate-domain
- panic-domain
- remove-domain
- start-domain
- stop-domain
- unbind-domain

이러한 이벤트는 항상 OVF 데이터 섹션에 이벤트가 발생한 도메인을 설명하는 <Content> 태그만 포함합니다. 도메인 이벤트에 대해 등록하려면 <action> 태그를 **reg-domain-events**로 설정하여 <LDM_interface> 메시지를 전송하십시오. 이 이벤트에 대한 등록을 해제하려면 action 태그가 **unreg-domain-events**로 설정된 <LDM_interface> 메시지가 필요합니다.

하드웨어 이벤트

하드웨어 이벤트는 물리적 시스템 하드웨어 변경과 관련됩니다. Oracle VM Server for SPARC 소프트웨어의 경우, 사용자가 SP(서비스 프로세서) 구성을 변경할 때 SP에 대해 수행되는 변경이 유일하게 가능한 하드웨어 변경입니다. 현재 이 유형의 이벤트는 세 가지입니다.

- add-spconfig
- set-spconfig
- remove-spconfig

하드웨어 이벤트는 항상 OVF 데이터 섹션에 이벤트가 발생한 SP 구성을 설명하는 <Section> 태그만 포함합니다. 이 이벤트에 대해 등록하려면 <action> 태그를 **reg-hardware-events**로 설정하여 <LDM_interface> 메시지를 전송하십시오. 이 이벤트에 대한 등록을 해제하려면 <action> 태그가 **unreg-hardware-events**로 설정된 <LDM_interface> 메시지가 필요합니다.

진행률 이벤트

진행률 이벤트는 도메인 마이그레이션과 같이 장기 실행 중인 명령에 대해 발행됩니다. 이 이벤트는 명령 수명 기간 동안 진행된 정도를 보고합니다. 이때 migration-process 이벤트만 보고됩니다.

진행률 이벤트는 항상 OVF 데이터 섹션에 이벤트의 영향을 받는 SP 구성을 설명하는 <Section> 태그만 포함합니다. 이 이벤트에 대해 등록하려면 <action> 태그를 **reg-hardware-events**로 설정하여 <LDM_interface> 메시지를 전송하십시오. 이 이벤트에 대한 등록을 해제하려면 <action> 태그가 **unreg-hardware-events**로 설정된 <LDM_interface> 메시지가 필요합니다.

진행률 이벤트의 <data> 섹션은 영향을 받는 도메인을 설명하는 <content> 섹션으로 구성됩니다. 이 <content> 섹션은 **ldom_info** <Section> 태그를 사용하여 진행률을 업데이트합니다. **ldom_info** 섹션에 표시되는 일반 등록 정보는 다음과 같습니다.

- --progress - 명령에 의해 수행된 진행률입니다.
- --status - 명령 상태를 나타내며, ongoing, failed, done 중 하나일 수 있습니다.
- --source - 진행률을 보고하는 시스템입니다.

리소스 이벤트

리소스 이벤트는 도메인에서 리소스가 추가, 제거 또는 변경될 경우에 발생합니다. 이 이벤트 중 일부에 대한 데이터 섹션에는 OVF 데이터 섹션에 서비스 이름을 제공하는 <Section> 태그와 함께 <Content> 태그가 포함되어 있습니다.

다음은 <LDM_event> 메시지의 <action> 태그에 지정할 수 있는 이벤트 목록입니다.

- add-vdiskserverdevice
- remove-vdiskserverdevice
- set-vdiskserverdevice

- remove-vdiskserver
- set-vconscon
- remove-vconscon
- set-vswitch
- remove-vswitch
- remove-vdpcs

다음 리소스 이벤트는 항상 OVF 데이터 섹션에 이벤트가 발생한 도메인을 설명하는 <Content> 태그만 포함합니다.

- add-vcpu
- add-crypto
- add-memory
- add-io
- add-variable
- add-vconscon
- add-vdisk
- add-vdiskserver
- add-vnet
- add-vswitch
- add-vdpcs
- add-vdpsc
- set-vcpu
- set-crypto
- set-memory
- set-variable
- set-vnet
- set-vconsole
- set-vdisk
- remove-vcpu
- remove-crypto
- remove-memory
- remove-io
- remove-variable
- remove-vdisk
- remove-vnet
- remove-vdpsc

리소스 이벤트에 대해 등록하려면 <action> 태그를 **reg-resource-events**로 설정하여 <LDM_interface> 메시지를 전송하십시오. 이 이벤트에 대한 등록을 해제하려면 <action> 태그가 **unreg-resource-events**로 설정된 <LDM_interface> 메시지가 필요합니다.

모든 이벤트

각 이벤트에 대해 개별적으로 등록할 필요 없이 세 가지 유형의 이벤트 전체를 수신하도록 등록할 수도 있습니다. 세 가지 유형의 이벤트에 대해 동시에 등록하려면

<action> 태그를 **reg-all-events**로 설정하여 <LDM_interface> 메시지를 전송하십시오. 이 이벤트에 대한 등록을 해제하려면 <action> 태그가 **unreg-all-events**로 설정된 <LDM_interface> 메시지가 필요합니다.

Logical Domains Manager 작업

<action> 태그에 지정된 명령(*.*-events 명령 제외)은 ldm 명령줄 인터페이스에 지정된 명령에 해당합니다. ldm 하위 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

주 - XML 인터페이스는 Logical Domains Manager CLI에서 지원하는 동사 또는 명령 별명을 지원하지 않습니다.

<action> 태그에 지원되는 문자열은 다음과 같습니다.

- add-domain
- add-io
- add-mau
- add-memory
- add-spconfig
- add-variable
- add-vconscon
- add-vcpu
- add-vdisk
- add-vdiskserver
- add-vdiskserverdevice
- add-vdpc
- add-vdpcs
- add-vnet
- add-vswitch
- bind-domain
- cancel-operation
- list-bindings
- list-constraints
- list-devices
- list-domain
- list-services
- list-spconfig
- list-variable
- migrate-domain
- reg-all-events
- reg-domain-events

- reg-hardware-events
- reg-resource-events
- remove-domain
- remove-io
- remove-mau
- remove-memory
- remove-reconf
- remove-spconfig
- remove-variable
- remove-vconscon
- remove-vcpu
- remove-vdisk
- remove-vdiskserver
- remove-vdiskserverdevice
- remove-udpcc
- remove-udpccs
- remove-vnet
- remove-vswitch
- set-domain
- set-mau
- set-memory
- set-spconfig
- set-variable
- set-vconscon
- set-vconsole
- set-vcpu
- set-vnet
- set-vswitch
- start-domain
- stop-domain
- unbind-domain
- unreg-all-events
- unreg-domain-events
- unreg-hardware-events
- unreg-resource-events

Logical Domains Manager 리소스 및 등록 정보

다음은 Logical Domains Manager 리소스 및 이러한 리소스 각각에 대해 정의된 등록 정보입니다. 리소스와 등록 정보는 XML 예제에서 **굵게** 표시됩니다. 이 예는 바인딩 출력이 아닌 리소스를 보여줍니다. 제약 조건 출력은 Logical Domains Manager 작업에 대한 입력을 만드는데 사용할 수 있습니다. 이에 대한 예외는 도메인 마이그레이션 출력입니다. [336 페이지 “도메인 마이그레이션”](#)을 참조하십시오. 각 리소스는 <Section> OVF 섹션에 정의되며 <rasd:OtherResourceType> 태그로 지정됩니다.

도메인 정보(ldom_info) 리소스

예 17-6 ldom_info XML 출력 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="primary">
    <Section xsi:type="ovf:ResourceAllocationSection_type">
      <Item>
        <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
        <uuid>c2c3d93b-a3f9-60f6-a45e-f35d55c05fb6</uuid>
        <rasd:Address>00:03:ba:d8:ba:f6</rasd:Address>
        <gprop:GenericPropertykey="hostid">83d8baf6</gprop:GenericProperty>
        <gprop:GenericProperty key="master">plum</gprop:GenericProperty>
        <gprop:GenericProperty key="failure-policy">reset</gprop:GenericProperty>
        <gprop:GenericProperty key="extended-mapin-space">on</gprop:GenericProperty>
        <gprop:GenericProperty key="progress">45%</gprop:GenericProperty>
        <gprop:GenericProperty key="status">ongoing</gprop:GenericProperty>
        <gprop:GenericProperty key="source">dt90-319</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

ldom_info 리소스는 항상 <Content> 섹션 내에 있습니다. ldom_info 리소스 내에 있는 다음 등록 정보는 선택적 등록 정보입니다.

- <uuid> 태그 - 도메인의 UUID를 지정합니다.
- <rasd:Address> 태그 - 도메인에 지정할 MAC 주소를 지정합니다.
- <gprop:GenericPropertykey="extended-mapin-space"> 태그 - 도메인에 대해 extended mapin space를 사용(on) 또는 사용 안함(off)으로 설정할지 여부를 지정합니다. 기본값은 off입니다.
- <gprop:GenericPropertykey="failure-policy"> 태그 - 마스터 도메인에서 오류가 발생할 경우 슬레이브 도메인의 작동 방식을 지정합니다. 기본값은 ignore입니다. 유효한 등록 정보 값은 다음과 같습니다.
 - ignore - 마스터 도메인의 오류를 무시합니다(슬레이브 도메인이 영향을 받지 않음).
 - panic으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인에 패닉이 발생합니다.
 - reset으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인이 재설정됩니다.
 - stop으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인이 중지됩니다.
- <gprop:GenericPropertykey="hostid"> 태그 - 도메인에 지정할 호스트 ID를 지정합니다.
- <gprop:GenericPropertykey="master"> 태그 - 마스터 도메인 이름을 쉼표로 구분하여 네 개까지 지정합니다.

- `<gprop:GenericPropertykey="progress">` 태그 - 명령에 의해 수행된 진행률을 지정합니다.
- `<gprop:GenericPropertykey="source">` 태그 - 명령의 진행률에 대해 보고할 시스템을 지정합니다.
- `<gprop:GenericPropertykey="status">` 태그 - 명령의 상태(done, failed, ongoing)를 지정합니다.

CPU(cpu) 리소스

`add-vcpu`, `set-vcpu` 및 `remove-vcpu` XML 요청 작업은 `<gpropGenericPropertykey="wcore">` 태그의 값을 다음과 같이 설정하는 것과 같습니다.

- `-c` 옵션을 사용할 경우, `wcore` 등록 정보를 지정된 전체 코어 수로 설정합니다.
- `-c` 옵션을 사용하지 **않을** 경우, `wcore` 등록 정보를 `0`으로 설정합니다.

`cpu` 리소스에 대한 할당 단위 등록 정보 `<rasd:AllocationUnits>`는 항상 코어 수가 아닌 가상 CPU 수를 지정합니다.

예 17-7 `cpu` XML 예

다음 예는 `ldm add-vcpu -c 1 ldg1` 명령에 해당하는 XML 요청을 보여줍니다.

```
<?xml version="1.0"?>
<LDM_interface version="1.3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./schemas/combined-v3.xsd"
xmlns:ovf="./schemas/envelope"
xmlns:rasd="./schemas/CIM_ResourceAllocationSettingData"
xmlns:vssd="./schemas/CIM_VirtualSystemSettingData"
xmlns:gprop="./schemas/GenericProperty"
xmlns:bind="./schemas/Binding">
  <cmd>
    <action>add-vcpu</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1">
          <Section xsi:type="ovf:VirtualHardwareSection_Type">
            <Item>
              <rasd:OtherResourceType>cpu</rasd:OtherResourceType>
              <rasd:AllocationUnits>8</rasd:AllocationUnits>
              <gprop:GenericProperty key="wcore">1</gprop:GenericProperty>
            </Item>
          </Section>
        </Content>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

`cpu` 리소스는 항상 `<Content>` 섹션 내에 있습니다.

MAU(mau) 리소스

주 - mau 리소스는 지원되는 서버에서 지원되는 암호화 단위입니다. 현재 지원되는 두 개의 암호화 단위는 MAU(Modular Arithmetic Unit) 및 CWQ(Control Word Queue)입니다.

예 17-8 mau XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>mau</rasd:OtherResourceType>
        <rasd:AllocationUnits>1</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>
```

mau 리소스는 항상 <Content> 섹션 내에 있습니다. 유일한 등록 정보는 MAU 또는 다른 암호화 단위의 개수를 나타내는 <rasd:AllocationUnits> 태그입니다.

메모리(memory) 리소스

예 17-9 memory XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>memory</rasd:OtherResourceType>
        <rasd:AllocationUnits>4G</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>
```

memory 리소스는 항상 <Content> 섹션 내에 있습니다. 유일한 등록 정보는 메모리의 양을 나타내는 <rasd:AllocationUnits> 태그입니다.

가상 디스크 서버(vds) 리소스

예 17-10 vds XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vds</rasd:OtherResourceType>
        <gprop:GenericProperty
          key="service_name">vdstmp</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

가상 디스크 서버(vds) 리소스는 도메인 설명의 일부로 <Content> 섹션에 표시되거나, <Envelope> 섹션에 자체적으로 표시될 수 있습니다. 유일한 등록 정보는 설명할 vds 리소스의 이름을 포함하는 service_name 키가 포함된 <gprop:GenericProperty> 태그입니다.

가상 디스크 서버 볼륨(vds_volume) 리소스

예 17-11 vds_volume XML 예

```
<Envelope>
  <References/>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>vds_volume</rasd:OtherResourceType>
      <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
      <gprop:GenericProperty key="service_name">primary-vds0</gprop:GenericProperty>
      <gprop:GenericProperty key="block_dev">
        opt/SUNWldm/domain_disks/testdisk1</gprop:GenericProperty>
      <gprop:GenericProperty key="vol_opts">ro</gprop:GenericProperty>
      <gprop:GenericProperty key="mpgroup">mpgroup-name</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>
```

vds_volume 리소스는 도메인 설명의 일부로 <Content> 섹션에 표시되거나, <Envelope> 섹션에 자체적으로 표시될 수 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있어야 합니다.

- vol_name – 볼륨 이름입니다.
- service_name – 이 볼륨이 바인드될 가상 디스크 서버의 이름입니다.
- block_dev – 이 볼륨과 연관될 파일 또는 장치 이름입니다.

선택적으로 `vds_volume` 리소스는 다음과 같은 등록 정보를 포함할 수도 있습니다.

- `vol_opts` - `{ro,slice,excl}`과 같이 한 문자열 내에 쉼표로 구분된 하나 이상의 옵션입니다.
- `mpgroup` - 다중 경로(페일오버) 그룹의 이름입니다.

디스크(disk) 리소스

예 17-12 disk XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>disk</rasd:OtherResourceType>
        <gprop:GenericProperty key="vdisk_name">vdisk0</gprop:GenericProperty>
        <gprop:GenericProperty
          key="service_name">primary-vds0</gprop:GenericProperty>
        <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
        <gprop:GenericProperty key="timeout">60</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

disk 리소스는 항상 `<Content>` 섹션 내에 있습니다. 다음 키를 포함하는 `<gprop:GenericProperty>` 태그가 있어야 합니다.

- `vdisk_name` - 가상 디스크의 이름입니다.
- `service_name` - 이 가상 디스크가 바인드될 가상 디스크 서버의 이름입니다.
- `vol_name` - 이 가상 디스크와 연관될 가상 디스크 서비스 장치입니다.

선택적으로 `disk` 리소스는 `timeout` 등록 정보를 포함할 수도 있습니다. 이 등록 정보는 가상 디스크 클라이언트(`vdc`)와 가상 디스크 서버(`vds`) 간에 연결을 설정하기 위한 시간 초과 값(초)입니다. 가상 디스크(`vdisk`) 경로가 여러 개 있는 경우 `vdc`는 다른 `vds`에 연결을 시도할 수 있으며, 시간 초과는 지정된 시간 내에 `vds`에 연결되도록 해줍니다.

가상 스위치(vsw) 리소스

예 17-13 vsw XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg2">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vsw</rasd:OtherResourceType>
```

예 17-13 vsw XML 예 (계속)

```

<rasd:Address>00:14:4f:fb:ec:00</rasd:Address>
<gprop:GenericProperty key="service_name">test-vsw1</gprop:GenericProperty>
<gprop:GenericProperty key="inter_vnet_link">on</gprop:GenericProperty>
<gprop:GenericProperty key="default-vlan-id">1</gprop:GenericProperty>
<gprop:GenericProperty key="pvid">1</gprop:GenericProperty>
<gprop:GenericProperty key="mtu">1500</gprop:GenericProperty>
<gprop:GenericProperty key="dev_path">switch@0</gprop:GenericProperty>
<gprop:GenericProperty key="id">0</gprop:GenericProperty>
</Item>
</Section>
</Content>
</Envelope>

```

vsw 리소스는 도메인 설명의 일부로 <Content> 섹션에 표시되거나, <Envelope> 섹션에 자체적으로 표시될 수 있습니다. service_name(가상 스위치에 지정될 이름) 키를 포함하는 <gprop:GenericProperty> 태그가 **반드시** 있어야 합니다.

선택적으로 vsw 리소스는 다음과 같은 등록 정보를 포함할 수도 있습니다.

- <rasd:Address> – 가상 스위치에 MAC 주소를 지정합니다.
- default-vlan-id – 태그 지정 모드에서 가상 네트워크 장치 또는 가상 스위치가 멤버로 속해 있어야 하는 기본 VLAN(Virtual Local Area Network) 을 지정합니다. 첫번째 VLAN ID(vid1)는 default-vlan-id용으로 예약되었습니다.
- dev_path – 이 가상 스위치와 연관될 네트워크 장치의 경로입니다.
- id – 새 가상 스위치 장치의 ID를 지정합니다. 기본적으로 ID 값은 자동으로 생성되므로, OS의 기존 장치 이름과 일치해야 하는 경우 이 등록 정보를 설정하십시오.
- inter_vnet_link – Vnet 간 통신을 위해 LDC 채널을 지정할지 여부를 지정합니다. 기본값은 on입니다.
- linkprop – 가상 장치가 물리적 링크 상태 업데이트를 가져올지 여부를 지정합니다. 값이 phys-state일 경우, 가상 장치가 물리적 링크 상태 업데이트를 가져옵니다. 값이 비어 있을 경우, 가상 장치가 물리적 링크 상태 업데이트를 가져오지 않습니다. 기본적으로 가상 장치는 물리적 링크 상태 업데이트를 가져오지 않습니다.
- mode – Oracle Solaris Cluster 하트비트 지원의 경우 sc입니다.
- pvid – 포트 VLAN(Virtual Local Area Network) ID(식별자)는 태그 미지정 모드에서 가상 네트워크가 멤버로 속해 있어야 하는 VLAN을 나타냅니다.
- mtu – 가상 스위치나 가상 스위치에 바인드된 가상 네트워크 장치 또는 둘 다의 MTU(최대 전송 단위)를 지정합니다. 유효한 값의 범위는 1500-16000입니다. 잘못된 값이 지정된 경우 ldm 명령이 오류를 발행합니다.
- vid – VLAN(Virtual Local Area Network) ID(식별자)는 태그 지정 모드에서 가상 네트워크와 가상 스위치가 멤버로 속해 있어야 하는 VLAN을 나타냅니다.

네트워크(network) 리소스

예 17-14 network XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>network</rasd:OtherResourceType>
        <gprop:GenericProperty key="linkprop">phys-state</gprop:GenericProperty>
        <gprop:GenericProperty key="vnet_name">ldg1-vnet0</gprop:GenericProperty>
        <gprop:GenericProperty
          key="service_name">primary-vsw0</gprop:GenericProperty>
        <rasd:Address>00:14:4f:fc:00:01</rasd:Address>
      </Item>
    </Section>
  </Content>
</Envelope>
```

network 리소스는 항상 <Content> 섹션 내에 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있어야 합니다.

- linkprop – 가상 장치가 물리적 링크 상태 업데이트를 가져올지 여부를 지정합니다. 값이 phys-state일 경우, 가상 장치가 물리적 링크 상태 업데이트를 가져옵니다. 값이 비어 있을 경우, 가상 장치가 물리적 링크 상태 업데이트를 가져오지 않습니다. 기본적으로 가상 장치는 물리적 링크 상태 업데이트를 가져오지 않습니다.
- vnet_name – 가상 네트워크(vnet)의 이름입니다.
- service_name – 이 가상 네트워크가 바인드될 가상 스위치(vswitch)의 이름입니다.

선택적으로 network 리소스는 다음과 같은 등록 정보를 포함할 수도 있습니다.

- <rasd:Address> – 가상 스위치에 MAC 주소를 지정합니다.
- pvid – 포트 VLAN(Virtual Local Area Network) ID(식별자)는 태그 미지정 모드에서 가상 네트워크가 멤버로 속해 있어야 하는 VLAN을 나타냅니다.
- vid – VLAN(Virtual Local Area Network) ID(식별자)는 태그 지정 모드에서 가상 네트워크와 가상 스위치가 멤버로 속해 있어야 하는 VLAN을 나타냅니다.
- mode – 해당 가상 네트워크에 하이브리드 I/O를 사용하려면 hybrid로 설정합니다.

가상 콘솔 집중기(vcc) 리소스

예 17-15 vcc XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
```

예 17-15 vcc XML 예 (계속)

```

<Item>
  <rasd:OtherResourceType>vcc</rasd:OtherResourceType>
  <gprop:GenericProperty key="service_name">vcc1</gprop:GenericProperty>
  <gprop:GenericProperty key="min_port">6000</gprop:GenericProperty>
  <gprop:GenericProperty key="max_port">6100</gprop:GenericProperty>
</Item>
</Section>
</Content>
</Envelope>

```

vcc 리소스는 도메인 설명의 일부로 <Content> 섹션에 표시되거나, <Envelope> 섹션에 자체적으로 표시될 수 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- service_name – 가상 콘솔 집중기 서비스에 지정할 이름입니다.
- min_port – 이 vcc와 연관될 최소 포트 번호입니다.
- max_port – 이 vcc와 연관될 최대 포트 번호입니다.

변수(var) 리소스

예 17-16 var XML 예

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>var</rasd:OtherResourceType>
        <gprop:GenericProperty key="name">test_var</gprop:GenericProperty>
        <gprop:GenericProperty key="value">test1</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

var 리소스는 항상 <Content> 섹션 내에 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- name – 변수의 이름입니다.
- value – 변수 값입니다.

물리적 I/O 장치(physio_device) 리소스

예 17-17 physio_device XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
        <gprop:GenericProperty key="name">pci@780</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

physio_device 리소스는 항상 <Content> 섹션 내에 있습니다. 유일한 등록 정보는 name 키 등록 정보 값(설명할 I/O 장치의 이름)을 포함하는 <gprop:GenericProperty> 태그입니다.

SP 구성(spconfig) 리소스

예 17-18 spconfig XML 예

```
<Envelope>
  <Section xsi:type="ovf:ResourceAllocationSection_type">
    <Item>
      <rasd:OtherResourceType>spconfig</rasd:OtherResourceType>
      <gprop:GenericProperty
        key="spconfig_name">primary</gprop:GenericProperty>
      <gprop:GenericProperty
        key="spconfig_status">current</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>
```

SP(서비스 프로세서) 구성(spconfig) 리소스는 항상 <Envelope> 섹션에 자체적으로 표시됩니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- spconfig_name – SP에 저장될 구성의 이름입니다.
- spconfig_status – 특정 SP 구성의 현재 상태입니다. 이 등록 정보는 ldm list-spconfig 명령의 출력에 사용됩니다.

DRM 정책 구성(policy) 리소스

예 17-19 policyXML 예

```
<Envelope>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>policy</rasd:OtherResourceType>
      <gprop:GenericProperty key="policy_name">test-policy</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_enable">on</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_priority">1</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_vcpu_min">12</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_vcpu_max">13</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_util_lower">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_util_upper">9</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_tod_begin">07:08:09</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_tod_end">09:08:07</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_sample_rate">1</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_elastic_margin">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_attack">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_decay">9</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>
```

DRM 정책(policy) 리소스는 <Envelope> 섹션에 표시되며 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- policy_name – DRM 정책의 이름입니다.
- policy_enable – DRM 정책을 사용 또는 사용 안함으로 설정할지 여부를 지정합니다.
- policy_priority – DRM 정책의 우선 순위입니다.
- policy_vcpu_min – 도메인의 최소 가상 CPU 리소스 수입니다.
- policy_vcpu_max – 도메인의 최대 가상 CPU 리소스 수입니다.
- policy_util_lower – 정책 분석이 트리거되는 하한 사용량 레벨입니다.
- policy_util_upper – 정책 분석이 트리거되는 상한 사용량 레벨입니다.
- policy_tod_begin – DRM 정책의 유효 시작 시간입니다.
- policy_tod_end – DRM 정책의 유효 중지 시간입니다.
- policy_sample_rate – 초 단위의 주기 시간인 샘플 비율입니다.
- policy_elastic_margin – CPU 사용량의 상한 및 하한 간 버퍼 양입니다.
- policy_attack – 한 리소스 제어 주기 중에 추가될 최대 리소스 양입니다.
- policy_decay – 한 리소스 제어 주기 중에 제거될 최대 리소스 양입니다.

가상 데이터 플레인 채널 서비스(vdpcs) 리소스

예 17-20 vdpcsXML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
```

예 17-20 vdpccs XML 예 (계속)

```

<Section xsi:type="ovf:VirtualHardwareSection_Type">
  <Item>
    <rasd:OtherResourceType>vdpccs</rasd:OtherResourceType>
    <gprop:GenericProperty key="service_name">ldg1-vdpccs</gprop:GenericProperty>
  </Item>
</Section>
</Content>
</Envelope>

```

이 리소스는 Netra DPS 환경에만 사용됩니다. vdpccs 리소스는 도메인 설명의 일부로 <Content> 섹션에 표시되거나, <Envelope> 섹션에 자체적으로 표시될 수 있습니다. 유일한 등록 정보는 설명할 가상 데이터 플레인 채널 서비스(vdpccs) 리소스의 이름인 service_name 키 등록 정보 값을 포함하는 <gprop:GenericProperty> 태그입니다.

가상 데이터 플레인 채널 클라이언트(vdpcc) 리소스

예 17-21 vdpcc XML 예

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vdpcc</rasd:OtherResourceType>
        <gprop:GenericProperty key="vdpcc_name">vdpcc</gprop:GenericProperty>
        <gprop:GenericProperty
          key="service_name">ldg1-vdpccs</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

이 리소스는 Netra DPS 환경에만 사용됩니다. 가상 데이터 플레인 채널 클라이언트 리소스는 항상 <Content> 섹션 내에 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- vdpcc_name – 가상 데이터 플레인 채널 클라이언트(vdpcc)의 이름입니다.
- service_name – 이 vdpcc가 바인드될 가상 데이터 플레인 채널 서비스 vdpccs의 이름입니다.

콘솔(console) 리소스

예 17-22 console XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>console</rasd:OtherResourceType>
        <gprop:GenericProperty key="port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="service_name">vcc2</gprop:GenericProperty>
        <gprop:GenericProperty key="group">group-name</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

console 리소스는 항상 <Content> 섹션 내에 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- port – 이 가상 콘솔(console)을 변경할 포트입니다.
- service_name – 이 console을 바인드할 가상 콘솔 집중기(vcc) 서비스입니다.
- group – 이 console을 바인드할 그룹의 이름입니다.

도메인 마이그레이션

이 예는 migrate-domain 하위 명령의 <data> 섹션에 포함된 내용을 보여줍니다.

예 17-23 migrate-domain <data> 섹션의 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
    <Section xsi:type="ovf:ResourceAllocationSection_Type">
      <Item>
        <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
        <gprop:GenericProperty key="target">target-host</gprop:GenericProperty>
        <gprop:GenericProperty key="username">user-name</gprop:GenericProperty>
        <gprop:GenericProperty key="password">password</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

여기서:

- 첫번째 <Content> 노드(<ldom_info> 섹션 없음)는 마이그레이션할 소스 도메인입니다.
- 두번째 <Content> 노드(<ldom_info> 섹션 있음)는 마이그레이션할 대상 도메인입니다. 소스 및 대상 도메인 이름은 같아도 됩니다.
- 대상 도메인의 <ldom_info> 섹션은 마이그레이션할 시스템 및 해당 시스템으로 마이그레이션하는 데 필요한 세부 정보를 설명합니다.
 - target-host - 마이그레이션할 대상 시스템입니다.
 - user-name - 대상 시스템의 로그인 사용자 이름입니다. SASL 64비트로 인코딩해야 합니다.
 - password - 대상 시스템에 로그인하는 데 사용할 암호입니다. SASL 64비트로 인코딩해야 합니다.

주 - Logical Domains Manager는 대상 사용자 이름과 암호를 디코딩하는 데 `sasl_decode64()`를 사용하고, 이러한 값을 인코딩하는 데 `sasl_encode64()`를 사용합니다. SASL 64 인코딩은 base64 인코딩에 해당합니다.

XML 스키마

/opt/SUNWldm/bin/schemas 디렉토리에 있는 각 XML 스키마 파일 이름은 다음과 같습니다. 이러한 스키마는 Logical Domains Manager에 사용됩니다.

- cim-common.xsd - cim-common.xsd 스키마
- cim-rasd.xsd - cim-rasd.xsd 스키마
- cim-vssd.xsd - cim-vssd.xsd 스키마
- cli-list-constraint-v3.xsd - cli-list-constraint-v3.xsd 스키마
- combined-v3.xsd - LDM_interface XML 스키마
- event-v3.xsd - LDM_Event XML 스키마
- ldmd-binding.xsd - Binding_Type XML 스키마
- ldmd-property.xsd - GenericProperty XML 스키마
- ovf-core.xsd - ovf-core.xsd 스키마
- ovf-envelope.xsd - ovf-envelope.xsd 스키마
- ovf-section.xsd - ovf-section.xsd 스키마
- ovf-strings.xsd - ovf-strings.xsd 스키마
- ovfenv-core.xsd - ovfenv-core.xsd 스키마
- ovfenv-section.xsd - ovfenv-section.xsd 스키마

용어 집

이 목록은 Oracle VM Server for SPARC 설명서의 용어, 약어, 머리글자어를 정의합니다.

A

API	Application programming interface, 응용 프로그래밍 인터페이스
ASN	Abstract Syntax Notation, 추상 구문 표기법
auditreduce	감사 추적 파일에서 감사 레코드 병합 및 선택(auditreduce(1M) 매뉴얼 페이지 참조)
감사	Oracle Solaris OS 감사를 사용하여 보안 변경의 원인을 파악합니다.
권한 부여	Oracle Solaris OS RBAC를 사용하여 권한 부여를 설정합니다.

B

bge	Broadcom BCM57xx 장치의 Broadcom 기가비트 이더넷 드라이버
BSM	Basic Security module, 기본 보안 모듈
bsmconv	BSM 사용(bsmconv(1M) 매뉴얼 페이지 참조)
bsmunconv	BSM 사용 안함(bsmunconv(1M) 매뉴얼 페이지 참조)

C

CD	Compact disc, 콤팩트 디스크
CLI	Command-line interface, 명령줄 인터페이스
CMT	Chip multithreading, 칩 멀티스레딩
준수	시스템 구성이 미리 정의된 보안 프로파일을 준수하는지 확인합니다.

구성	서비스 프로세서에 저장된 논리적 도메인 구성의 이름입니다.
제약 조건	Logical Domains Manager에서 제약 조건은 특정 도메인에 지정하려는 하나 이상의 리소스입니다. 사용 가능한 리소스에 따라, 도메인에 추가하도록 요청한 리소스를 모두 받거나 아무 것도 얻지 못합니다.
컨트롤 도메인	Logical Domains Manager를 사용하여 다른 논리적 도메인 및 서비스를 만들고 관리하는 특권 도메인입니다.
CPU	Central processing unit, 중앙 처리 장치
CWQ	Control Word Queue; Oracle's Sun UltraSPARC T2 기반 플랫폼의 암호화 장치

D

DHCP	Dynamic Host Configuration Protocol, 동적 호스트 구성 프로토콜
DIO	Direct I/O, 직접 입/출력
DMA	직접 메모리 액세스는 CPU 관여 없이 메모리와 장치(예: 네트워크 카드) 간에 직접 데이터를 전송하는 기능입니다.
DMP	Dynamic Multipathing(Veritas), 동적 다중 경로 제어
도메인	논리적 도메인 을 참조하십시오.
Logical Domains Manager	논리적 도메인을 만들고 관리하거나 도메인에 리소스를 할당하기 위한 CLI입니다.
DPS	Data plane software, 데이터 평면 소프트웨어
DR	Dynamic reconfiguration, 동적 재구성
drd	Oracle Solaris 10 OS의 Logical Domains Manager용 동적 재구성 데몬(drd(1M) 매뉴얼 페이지 참조)
DRM	Dynamic resource management, 동적 리소스 관리
DS	도메인 서비스 모듈(Oracle Solaris 10 OS)
DVD	Digital versatile disc, 디지털 다기능 디스크

E

EFI	Extensible firmware interface, 확장성 펌웨어 인터페이스
ETM	인코딩 테이블 관리 모듈(Oracle Solaris 10 OS)

F

FC_AL	Fiber Channel Arbitrated Loop, 광섬유 채널 중재 루프
FMA	Fault Management Architecture, 결함 관리 구조
fmd	Oracle Solaris 10 OS 결함 관리자 데몬(fmd(1M) 매뉴얼 페이지 참조)
format	디스크 분할 및 유지 관리 유틸리티(format(1M) 매뉴얼 페이지 참조)
fmthard	하드 디스크에 레이블 채우기(fmthard(1M) 매뉴얼 페이지 참조)
FTP	File Transfer Protocol, 파일 전송 프로토콜

G

Gb	Gigabit, 기가비트
게스트 도메인	I/O 및 서비스 도메인에서 서비스를 사용하고 컨트롤 도메인에 의해 관리됩니다.
GLDv3	Generic LAN Driver 버전 3

H

강화	보안을 향상하도록 Oracle Solaris OS 구성을 수정합니다.
HDD	Hard disk drive, 하드 디스크 드라이브
하이퍼바이저	운영 체제와 하드웨어 계층 사이에 위치한 펌웨어 계층입니다.

I

I/O 도메인	물리적 I/O 장치에 직접 소유권과 직접 액세스를 보유한 도메인으로, 이러한 장치를 가상 장치 형태로 다른 논리적 도메인과 공유합니다.
IB	Infiniband
IDE	Integrated Drive Electronics
IDR	Interim Diagnostics Release
ILOM	Integrated Lights Out Manager
I/O	내부 디스크 및 PCIe 컨트롤러와 같은 입/출력 장치와 여기에 연결된 어댑터 및 장치입니다.

ioctl	입/출력 제어 호출
IP	Internet Protocol, 인터넷 프로토콜
IPMP	Internet Protocol Network Multipathing, 인터넷 프로토콜 네트워크 다중 경로 제어
ISO	International Organization for Standardization, 국제 표준화 기구

K

kaio	커널 비동기 입/출력
KB	Kilobyte, 킬로바이트
KU	Kernel update, 커널 업데이트

L

LAN	Local-area network, 근거리 통신망
LDAP	Lightweight Directory Access Protocol, 경량 디렉토리 액세스 프로토콜
LDC	Logical domain channel, 논리적 도메인 채널
ldm	Logical Domains Manager 유틸리티(ldm(1M) 매뉴얼 페이지 참조)
ldmd	Logical Domains Manager 데몬
lofi	루프백 파일
논리적 도메인	별도의 논리적 리소스 그룹으로 구성된 가상 컴퓨터로, 단일 컴퓨터 시스템 안에 자체 운영 체제와 신원을 가지고 있습니다. 도메인 이라고도 합니다.
LUN	Logical unit number, 논리적 장치 번호

M

MAC	매체 액세스 제어 주소로, Logical Domains가 자동으로 지정하거나 사용자가 수동으로 지정할 수 있습니다.
MAU	Modular Arithmetic Unit, 모듈식 산술 장치
MB	Megabyte, 메가바이트

MD	서버 데이터베이스의 시스템 설명
mem, memory	메모리 단위 - 기본 크기인 바이트를 사용하거나 기가바이트(G), 킬로바이트(K), 메가바이트(M)를 지정합니다. 게스트 도메인에 할당할 수 있는 서버의 가상화된 메모리입니다.
metadb	Solaris Volume Manager 메타 장치 상태 데이터베이스의 복제본 만들기 및 삭제(metadb(1M) 매뉴얼 페이지 참조)
metaset	디스크 세트 구성(metaset(1M) 매뉴얼 페이지 참조)
mhd	다중 호스트 디스크 제어 작업(mhd(7i) 매뉴얼 페이지 참조)
MIB	Management Information Base
최소화	필요한 최소 수의 코어 Oracle Solaris OS 패키지를 설치합니다.
MMF	Multimode fiber, 다중 모드 광섬유
MMU	Memory management unit, 메모리 관리 장치
mpgroup	가상 디스크 장애 조치용 다중 경로 그룹 이름
mtu	Maximum transmission unit, 최대 전송 단위

N

NAT	Network Address Translation, 네트워크 주소 변환
ndpsldcc	Netra DPS 논리적 도메인 채널 클라이언트. vdpccl 을 참조하십시오.
ndpsldcs	Netra DPS 논리적 도메인 채널 서비스. vdpcs 를 참조하십시오.
NFS	Network file system, 네트워크 파일 시스템
NIS	Network Information Services, 네트워크 정보 서비스
NIU	네트워크 인터페이스 장치(Oracle's Sun SPARC Enterprise T5120 및 T5220 서버)
NTS	Network terminal server, 네트워크 터미널 서버
NVRAM	Non-volatile random-access memory, 비휘발성 랜덤 액세스 메모리
nxge	NIU 10Gb 이더넷 어댑터용 드라이버

O

OID	객체 식별자로, MIB에서 각 객체를 고유하게 식별하는 번호 시퀀스입니다.
------------	---

OS	Operating system, 운영 체제
OVF	Open Virtualization Format, 개방형 가상화 형식

P

P2V	Logical Domains Physical-to-Virtual 변환 도구
PA	Physical address, 물리적 주소
PCI	Peripheral component interconnect 버스
PCIe	PCI EXPRESS 버스
PCI-X	PCI Extended 버스
pcpu	물리적 CPU
물리 함수	SR-IOV 사양에 정의된 대로 SR-IOV 기능을 지원하는 PCI 함수입니다. 물리 함수는 SR-IOV 기능 구조를 포함하며 SR-IOV 기능을 관리하는 데 사용됩니다. 물리 함수는 다른 PCIe 장치처럼 검색, 관리 및 조작할 수 있는 완전형 PCIe 함수입니다. 물리 함수에는 전체 구성 리소스가 있으며, PCIe 장치를 구성하고 제어하는 데 사용할 수 있습니다.
physio	물리적 입/출력
PICL	Platform Information and Control Library, 플랫폼 정보 및 컨트롤 라이브러리
picld	PICL 데몬(picld(1M) 매뉴얼 페이지 참조)
PM	가상 CPU 및 메모리의 전원 관리
praudit	감사 추적 파일의 내용 인쇄(praudit(1M) 매뉴얼 페이지 참조)
PRI	Priority, 우선 순위
PROM	Programmable read-only memory, 프로그램 가능 읽기 전용 메모리

R

RA	Real address, 실 주소
RAID	Redundant Array of Inexpensive Disks
RBAC	Role-Based Access Control, 역할 기반 액세스 제어
RPC	Remote Procedure Call, 원격 프로시저 호출

S

SASL	Simple Authentication and Security Layer, 단순 인증 및 보안 계층
SAX	Simple API for XML(XML용 단순 API) 구문 분석기로, XML 문서를 순회합니다. SAX 구문 분석기는 이벤트 기반이며 대부분 데이터 스트리밍에 사용됩니다.
시스템 컨트롤러(SC)	서비스 프로세서를 참조하십시오.
SCSI	Small Computer System Interface, 소형 컴퓨터 시스템 인터페이스
서비스 도메인	다른 논리적 도메인에 가상 스위치, 가상 콘솔 커넥터, 가상 디스크 서버와 같은 장치를 제공하는 논리적 도메인입니다.
SMA	System Management Agent
SMF	Service Management Facility, 서비스 관리 기능
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol, 단순 네트워크 관리 프로토콜
서비스 프로세서(SP)	SP는 시스템 컨트롤러(SC)라고도 하며, 물리적 컴퓨터를 모니터링하고 실행합니다.
SR-IOV	Single root I/O virtualization, 단일 루트 입/출력 가상화
SSH	Secure Shell, 보안 셸
ssh	보안 셸 명령(ssh(1) 매뉴얼 페이지 참조)
sshd	보안 셸 데몬(sshd(1M) 매뉴얼 페이지 참조)
SunVTS	Sun Validation Test Suite, Sun 검증 테스트 모음
svcadm	서비스 인스턴스 조작(svcadm(1M) 매뉴얼 페이지 참조)

T

TCP	Transmission Control Protocol, 전송 제어 프로토콜
TLS	Transport Layer Security, 전송 계층 보안

U

UDP	User Datagram Protocol, 사용자 다이어그램 프로토콜
------------	--

UFS	UNIX File System, UNIX 파일 시스템
유니캐스트	단일 발신자와 단일 수신자 사이에 발생하는 네트워크 통신입니다.
USB	Universal Serial Bus, 범용 직렬 버스
uscsi	사용자 SCSI 명령 인터페이스(uscsi(7I) 매뉴얼 페이지 참조)
UTP	Unshielded twisted pair, 비차폐 연선

V

var	변수
VBSC	Virtual blade system controller, 가상 블레이드 시스템 컨트롤러
vcc, vconscn	특정 포트 범위의 가상 콘솔 집중기 서비스를 게스트 도메인에 지정합니다.
vcons, vconsole	시스템 레벨 메시지에 액세스하기 위한 가상 콘솔입니다. 특정 포트의 컨트롤 도메인에서 vconscn 서비스에 연결합니다.
vcpu	가상 중앙 처리 장치. 서버의 각 코어는 가상 CPU로 표현됩니다. 예를 들어, Oracle의 8-코어 Sun Fire T2000 Server는 논리적 도메인 사이에 할당할 수 있는 32개 가상 CPU가 있습니다.
vdc	가상 디스크 클라이언트
vdisk	가상 디스크는 여러 유형의 물리적 장치, 볼륨, 파일과 연관된 일반 블록 장치입니다.
vdppc	Netra DPS 환경의 가상 데이터 평면 채널 클라이언트
vdpcs	Netra DPS 환경의 가상 데이터 평면 채널 서비스
vds, vdiskserver	가상 디스크 서버에서 가상 디스크를 논리적 도메인으로 가져올 수 있습니다.
vdsdev, vdiskserverdevice	가상 디스크 서버가 가상 디스크 서버 장치를 내보냅니다. 장치는 전체 디스크, 디스크의 슬라이스, 파일 또는 디스크 볼륨일 수 있습니다.
가상 함수	물리 함수와 연관된 PCI 함수입니다. 가상 함수는 하나 이상의 물리적 리소스를 물리 함수와 공유하거나, 동일한 물리 함수로 연관된 다른 가상 함수와 공유하는 경량 PCIe 함수입니다. 가상 함수는 자체 동작을 위한 구성 리소스만 가질 수 있습니다.
VNIC	가상 네트워크 인터페이스 카드로, 물리적 네트워크 장치를 가상 인스턴스화한 것이며 물리적 네트워크 장치에서 만들어 영역에 지정할 수 있습니다.
VLAN	Virtual local area network, 가상 근거리 통신망
vldc	가상 논리적 도메인 채널 서비스
vldcc	가상 논리적 도메인 채널 클라이언트

vnet	가상 네트워크 장치는 가상 이더넷 장치를 구현하고 가상 네트워크 스위치(vswitch)를 사용하여 시스템의 다른 vnet 장치와 통신합니다.
vNTS	Virtual network terminal service, 가상 네트워크 터미널 서비스
vntsd	Oracle Solaris 10 OS의 Logical Domains 콘솔용 가상 네트워크 터미널 서버 데몬(vntsd(1M) 매뉴얼 페이지 참조)
volfs	볼륨 관리 파일 시스템(volfs(7FS) 매뉴얼 페이지 참조)
vsw, vswitch	가상 네트워크 장치를 외부 네트워크에 연결하고 장치 사이에 패킷을 전환하는 가상 네트워크 스위치입니다.
VTOC	Volume table of contents, 볼륨 목차
VxDMP	Veritas Dynamic Multipathing, 동적 다중 경로 제어
VxVM	Veritas Volume Manager, Veritas 볼륨 관리자

W

WAN	Wide-area network, 원거리 통신망
------------	----------------------------

X

XFP	eXtreme Fast Path
XML	Extensible Markup Language, 확장성 마크업 언어
XMPP	Extensible Messaging and Presence Protocol, 확장성 메시징 및 프레즌스 프로토콜

Z

ZFS	제타바이트 파일 시스템(Oracle Solaris 10 OS)
zpool	ZFS 저장소 풀(zpool(1M) 매뉴얼 페이지 참조)
ZVOL	ZFS Volume Emulation Driver, ZFS 볼륨 에뮬레이션 드라이버

색인

C

cancel-reconf 하위 명령, 192
CLI, “명령줄 인터페이스”참조
CPU 코어 사용 안함, 217
CPU 클럭 사이클 건너뛰기, 217

D

daemons, drd, 192
DIO(직접 I/O), 계획, 82
DR, “동적 재구성”참조
dynamic reconfiguration daemon (drd), 192

I

I/O 도메인, 73-74, 74-79, 79-89, 90-105
 PCIe(PCI EXPRESS) 버스, 73-74
 PCIe SR-IOV 가상 기능 사용, 90-105
 마이그레이션 제한 사항, 74
 만들기, 76
 지정
 PCIe 버스, 74-79
 끝점 장치, 79-89
I/O 도메인 만들기, 전체 PCIe 버스, 76

L

LDC, “논리적 도메인 채널”참조
LDC(논리적 도메인 채널), 24

ldm 하위 명령

 cancel-reconf, 192
 사용자 권한 부여, 46
ldm(1M) 매뉴얼 페이지, 25
ldm(1M) 명령, 25
ldmconfig(1M) 명령, 27, 267, 268
ldmd, Logical Domains Manager 데몬, 25
ldmp2v(1M) 명령, 254
ldomCoreTable 테이블, 297-298
ldomCreate 트랩, 299-300
ldomCryptoTable 테이블, 296-297
ldomDestroy 트랩, 300
ldomEnvVarsTable 테이블, 282-283
ldomIOBusTable 테이블, 297
ldomPolicyTable 테이블, 283-284
ldomSPConfigTable 테이블, 284
ldomStateChange 트랩, 300
ldomTable 테이블, 281-282
ldomVccChange 트랩, 303-304
ldomVccTable 테이블, 293-294
ldomVconsChange 트랩, 304
ldomVconsTable 테이블, 294
ldomVconsVccRelTable 테이블, 294-295
ldomVCpuChange 트랩, 300
ldomVcpuTable 테이블, 285-287
ldomVdiskChange 트랩, 302
ldomVdiskTable 테이블, 289-290
ldomVdsChange 트랩, 301-302
ldomVdsdevTable 테이블, 289
ldomVdsTable 테이블, 288-289
ldomVMemChange 트랩, 300-301
ldomVmemPhysBindTable 테이블, 288

ldomVmemTable 테이블, 287-288
 ldomVnetChange 트랩, 303
 ldomVnetTable 테이블, 292-293
 ldomVswChange 트랩, 302-303
 ldomVswTable 테이블, 292
 Logical Domains Manager, 22, 24
 Oracle VM Server for SPARC MIB, 274
 XML 스키마 사용, 313
 검색 방식, 309
 데몬(ldmd), 25

M

MIB, 271
 MIB(Management Information Base), 271

O

Oracle VM Server for SPARC MIB
 Logical Domains Manager, 274
 객체 트리, 274-275
 소프트웨어 구성 요소, 272
 Oracle VM Server for SPARC MIB 객체,
 검색, 279-281
 Oracle VM Server for SPARC MIB 모듈, SMA에서
 로드, 275-277
 Oracle VM Server for SPARC MIB 소프트웨어
 구성, 275-277
 설치, 275-277
 제거, 275-277
 Oracle VM Server for SPARC MIB 테이블
 CPU 리소스 풀에 대한 스칼라 변수, 284
 I/O 버스 리소스 풀에 대한 스칼라 변수, 285
 I/O 버스 테이블(ldomIOBusTable), 297
 Logical Domains 버전 정보에 대한 스칼라
 변수, 298
 가상 CPU 테이블(ldomVcpuTable), 285-287
 가상 네트워크 장치
 테이블(ldomVnetTable), 292-293
 가상 디스크 서비스 장치
 테이블(ldomVdsdevTable), 289
 가상 디스크 서비스
 테이블(ldomVdsTable), 288-289

Oracle VM Server for SPARC MIB 테이블 (계속)
 가상 디스크 테이블(ldomVdiskTable), 289-290
 가상 메모리 물리적 바인딩
 테이블(ldomVmemPhysBindTable), 288
 가상 메모리 테이블(ldomVmemTable), 287-288
 가상 스위치 서비스 장치
 테이블(ldomVswTable), 292
 가상 콘솔 관계
 테이블(ldomVconsVccRelTable), 294-295
 가상 콘솔 그룹 테이블(ldomVconsTable), 294
 가상 콘솔 집중기
 테이블(ldomVccTable), 293-294
 도메인 정책 테이블(ldomPolicyTable), 283-284
 도메인 테이블(ldomTable), 281-282
 서비스 프로세서 구성
 테이블(ldomSPConfigTable), 284
 암호화 리소스 풀에 대한 스칼라 변수, 285
 암호화 장치 테이블(ldomCryptoTable), 296-297
 코어 테이블(ldomCoreTable), 297-298
 환경 변수 테이블(ldomEnvVarsTable), 282-283
 Oracle VM Server for SPARC MIB 트랩, 299-304
 가상 CPU 변경(ldomVCpuChange), 300
 가상 네트워크 변경(ldomVnetChange), 303
 가상 디스크 변경(ldomVdiskChange), 302
 가상 디스크 서비스
 변경(ldomVdsChange), 301-302
 가상 메모리 변경(ldomVMemChange), 300-301
 가상 스위치 변경(ldomVswChange), 302-303
 가상 콘솔 그룹 변경(ldomVconsChange), 304
 가상 콘솔 집중기 변경(ldomVccChange), 303-304
 도메인 만들기(ldomCreate), 299-300
 도메인 삭제(ldomDestroy), 300
 도메인 상태 변경(ldomStateChange), 300
 수신, 299
 전송, 298-299

P

PCIe(PCI EXPRESS) 버스, 73-74
 primary 도메인, 24
 재부트, 83-84
 primary 도메인 재부트, 83-84

S

snmpv3 사용자, 만들기, 277-278
 SPARC T-Series 서버, 25
 SUNWldm 패키지, 25
 System Management Agent, 273

V

vNTS, 293-295

X

XML 기반 컨트롤 인터페이스, 구문 분석, 274
 XML 스키마, Logical Domains Manager에서
 사용, 313

가

가상 네트워크 터미널 서버 데몬(vntsd), 26
 가상 디스크 다중 경로, 120
 가상 장치
 I/O, 25
 가상 네트워크(vnet), 25
 가상 디스크 서비스(vds), 26
 가상 디스크 클라이언트(vdc), 26
 가상 스위치(vsw), 25
 가상 콘솔 집중기(vcc), 26
 가상 컴퓨터, 24

검**검색**

Oracle VM Server for SPARC MIB 객체, 279-281
 Oracle VM Server for SPARC MIB 정보, 281-298

계

게스트 도메인, 25

결

결함 및 복구 정보, 제공, 274

계**계획**

DIO(직접 I/O), 82

구

구문 분석, XML 기반 컨트롤 인터페이스, 274
 구성
 Oracle VM Server for SPARC MIB
 소프트웨어, 275-277
 부트용 선택, 26
 점보 프레임, 171-174

권

권한 부여, ldm 하위 명령, 46

논

논리적 도메인
 역할, 24
 정의, 22

다

다중 경로, 가상 디스크, 120

데**데몬**

ldmd, 25
 vntsd, 26

도

도메인

- 서비스, 25
- 시작, 304–306
- 유형, 24, 25
- 중지, 306–307

도메인 마이그레이션, 비대화식, 188

동

동적 재구성(DR), 191

로

로드, SMA에서 Oracle VM Server for SPARC MIB
모듈, 275–277

루

루트 도메인, 24

리

리소스

- “가상 장치”참조
- 정의, 23

링

링크 기반 IPMP, 사용, 156–159

마

마이그레이션, 비대화식, 188
마이그레이션 제한 사항, I/O 도메인, 74

만

만들기, snmpv3 사용자, 277–278

메

메모리 동적 재구성(DR), 209
메모리 전원 관리(PM), 217

명

명령

- ldm(1M), 25
 - ldmconfig(1M), 27, 267, 268
 - ldmp2v(1M), 254
- 명령줄 인터페이스, 25

물

물리적 장치, 24, 25
물리적 컴퓨터, 24

비

비대화식 도메인 마이그레이션, 188

사

사용

I/O 도메인의 PCIe SR-IOV 가상 기능, 90–105
링크 기반 IPMP, 156–159

서

서비스 도메인, 24, 25
서비스 프로세서(SP), 물리적 컴퓨터 모니터링 및
실행, 24

설

설정

전원 제한, 217

환경 변수, 279

설치, Oracle VM Server for SPARC MIB

소프트웨어, 275-277

수

수신, Oracle VM Server for SPARC MIB 트랩, 299

시

시스템 컨트롤러, “서비스 프로세서(SP)”참조

시작, 도메인, 304-306

역

역할, 논리적 도메인, 24

전

전송, Oracle VM Server for SPARC MIB

트랩, 298-299

전원 관리(PM), 217

전원 제한, 217

점

점보 프레임, 구성, 171-174

제

제거, Oracle VM Server for SPARC MIB

소프트웨어, 275-277

제공

결함 및 복구 정보, 274

트랩, 274

중

중지, 도메인, 306-307

지

지연된 재구성, 192

지정

I/O 도메인에 PCIe 버스, 74-79

I/O 도메인에 끝점 장치, 79-89

컨

컨트롤 도메인, 24

테

테이블, “Oracle VM Server for SPARC MIB

테이블”참조

트

트랩

“Oracle VM Server for SPARC MIB 트랩”참조

제공, 274

패

패키지, SUNWldm, 25

플

플랫폼, SPARC T-Series 서버, 25

하

하이퍼바이저, 정의, 22

환

환경 변수, 설정, 279