



Solaris 공통 데스크탑 환경: 고급 사용자 및 시스템 관리자 안내서

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

부품 번호: 816-4021-10
2002년5월

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 모든 권리는 저작권자의 소유입니다.

이 제품이나 문서는 저작권의 보호를 받으며, 사용, 복사, 배포 및 발제를 제한하는 라이선스에 따라 배포됩니다. 이 제품이나 문서의 어떤 부분도 Sun 및 그 라이선스 부여자의 사전 서면 승인 없이 어떤 형태로든 어떤 수단을 통해서든 복제할 수 없습니다. 글꼴 기술 등의 협력업체 소프트웨어는 저작권의 보호를 받으며 Sun 공급업체가 해당 라이선스를 부여하고 있습니다.

제품 일부는 캘리포니아 대학이 라이선스를 부여한 버클리 BSD 시스템에서 파생한 것일 수 있습니다. UNIX는 미국 및 기타 국가에서 등록된 상표로서 X/Open Company, Ltd를 통해 독점적으로 라이선스가 부여됩니다.

Sun, Sun Microsystems, Sun 로고, docs.sun.com, AnswerBook, AnswerBook2, 및 Solaris는 미국 및 기타 국가에서 Sun Microsystems, Inc.의 상표, 등록 상표 또는 서비스 상표입니다. SPARC 상표는 모두 해당 라이선스에 따라 사용하며 미국 및 기타 국가에서 사용하고 있는 SPARC International, Inc.의 상표 또는 등록상표입니다. DtComBox와 DtSpinBox 위지트의 코드와 설명서는 Interleaf, Inc.가 제공한 것입니다. Copyright 1993, Interleaf, Inc.

OPEN LOOK과 Sun™ 그래픽 사용자 인터페이스는 Sun Microsystems, Inc.가 사용자 및 라이선스 취득자용으로 개발했습니다. Sun은 컴퓨터 업계에 위한 비주얼 또는 그래픽 사용자 인터페이스의 개념을 연구 개발한 Xerox사의 선구적인 노력을 높이 인정하고 있습니다. Sun은 Xerox 그래픽 사용자 인터페이스에 대한 비독점적인 라이선스를 Xerox사로부터 취득 보유하고 있으며, 이 라이선스는 또한 OPEN LOOK GUI를 구현하고 기타 Sun 라이선스 계약을 준수하는 Sun의 라이선스 취득자에게도 적용됩니다.

권리 제한: 미국 정부에 의한 사용, 복제 또는 공개는 FAR 52.227-14(g)(2)(87. 6) 및 FAR 52.227-19(87. 6) 또는 DFAR 252.227-7015(b)(95. 6) 및 DFAR 227.7202-3(a)의 제한 사항을 따릅니다.

설명서는 “있는 그대로” 제공되며, 시장성의 묵시적 보증, 특정 용도에의 적합성, 권리 불침해 등의 조건, 주장 및 보증 사실 등은 모두 명시적이든 묵시적이든 부인됩니다. 다만, 상기의 부인이 법률적으로 무효라는 판결을 받는 경우에는 예외입니다.

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la d'écompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. Le code et la documentation pour les produits DtComBox et DtSpinBox ont été fournis par Interleaf, Inc. Copyright 1993, Interleaf, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licences de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



020403@3688



목차

머리말 15

1 로그인 관리자 구성	19
로그인 서버 시작	20
지역 표시장치 및 네트워크 표시장치 관리	20
로그인 서버 프로세스 ID 찾기	21
로그인 화면을 지역 표시장치에 표시	22
지역 표시장치 없이 로그인 서버 실행	22
지역 표시장치에서 명령행 로그인 액세스	23
문자 표시 콘솔 허용	23
네트워크 표시장치에 로그인 화면 표시	24
로그인 서버에 대한 액세스 제어	25
오류 점검	26
로그인 서버 중지	27
로그인 화면	27
로그인 화면 모양 변경	28
▼ 로고를 변경하는 방법	29
▼ 환영 메시지 변경 방법	29
▼ 글꼴 변경 방법	29
▼ 각 언어 표시를 위한 대체 텍스트 제공 방법	30
표시 장치별 로그인 화면 동작 변경	31
X 서버 액세스 변경	31
로그인 화면이 표시되기 전에 명령어 수행	33
Failsafe 세션 시작	33
사용자 세션이 끝난 후	34
로그인 서버 환경	34

사용자 또는 시스템 경로 변경	35
▼ 사용자 또는 시스템 쉘 변경 방법	35
▼ 시간대 변경 방법	35
로그인 관리자 관리	36
로그인 관리자 파일	36
2 세션 관리자 구성	39
세션이란?	39
초기 세션	40
현재 세션	40
홈 세션	40
특정 표시장치 세션	40
세션 시작	40
세션 관리자가 시작될 때	41
.dtprofile 스크립트를 소스로 사용하기	41
Xsession.d 스크립트를 소스로 사용하기	42
환영 메시지 표시	42
데스크탑 검색 경로 설정	43
사용 가능한 응용 프로그램 모으기	43
선택적으로 .profile 또는 .login 스크립트를 소스로 사용하기	44
ToolTalk 메시지 표시 데몬 시작	44
세션 관리자 클라이언트 시작	45
세션 자원 로드	45
색상 서버 시작	46
작업 공간 관리자 시작	46
세션 응용 프로그램 시작	47
추가 세션 시작 사용자 정의	47
▼ 환경 변수 설정 방법	47
▼ 자원 설정 방법	48
▼ 특정 표시장치 자원 설정 방법	48
▼ 초기 세션에 대한 응용 프로그램 변경 방법	49
▼ 특정 표시장치 세션 설정 방법	49
세션 시작 및 로그아웃시 추가 명령어 실행	50
▼ 세션 시작시 추가 명령어 실행 방법	50
▼ 로그아웃시 추가 명령어 실행 방법	50
▼ 백업으로부터 세션 복구 방법	50
▼ 세션 시작 문제 조사 방법	51

	세션 관리자 파일 및 디렉토리	51
3	문제 해결 로그인 및 세션 시작 문제	53
	로그인 시작 파일	53
	오류 로그 위치	54
	사용자 시작 파일	54
	Solaris CDE 시작 예	55
4	응용 프로그램의 추가 및 관리	57
	응용 프로그램 관리자의 구조	57
	응용 프로그램 관리자의 디렉토리 위치	58
	응용 프로그램 관리자가 응용 프로그램을 찾고 모으는 방법	58
	응용 프로그램 모으기의 우선순위 규칙	59
	기본 데스크탑이 제공하는 응용 프로그램 그룹	59
	응용 프로그램 그룹을 모으는 방법의 예	60
	응용 프로그램 관리자에 응용 프로그램 추가	61
	응용 프로그램 관리자에 응용 프로그램을 추가하는 방법	61
	▼ 응용 프로그램 관리자에 데스크탑 스마트 응용 프로그램을 추가하는 방법	62
	▼ 기존 응용 프로그램 또는 데스크탑 스마트가 아닌 응용 프로그램을 등록하는 방법	62
	▼ 기존 응용 프로그램 그룹에 응용 프로그램 아이콘을 추가하는 방법	63
	일반 응용 프로그램 그룹 작성 및 관리	63
	▼ 시스템 전체 일반 응용 프로그램 그룹을 작성하는 방법	63
	▼ 개인 일반 응용 프로그램 그룹을 작성하는 방법	64
	▼ 내장 응용 프로그램 그룹을 사용자 정의하는 방법	64
	응용 프로그램 위치를 찾는데 사용되는 검색 경로 수정	64
	기본 검색 경로	64
	응용 프로그램 검색 경로에 응용 프로그램 서버 추가	65
	일반 응용 프로그램 관리자 관리	66
	▼ 응용 프로그램을 제거하는 방법	66
	▼ 세션 중에 응용 프로그램 관리자를 갱신하는 방법	66
	문서 편집기 및 터미널 에뮬레이터 변경	67
	▼ 기본 문서 편집기 또는 터미널 에뮬레이터를 변경하는 방법	67
5	응용 프로그램 등록	71
	응용 프로그램 등록 개요	72
	응용 프로그램 등록으로 제공되는 기능	72

응용 프로그램 등록의 목적	74
응용 프로그램 등록을 위한 일반적 단계	75
1단계: 글꼴 및 색상 자원 수정	75
2단계: 데스크탑 응용 프로그램 루트 작성	77
3단계: 등록 패키지 디렉토리 작성	77
4단계: 응용 프로그램의 활동 및 데이터 유형 작성	79
5단계: 등록 패키지에 도움말 파일 넣기	82
6단계: 응용 프로그램 아이콘 작성	83
7단계: 응용 프로그램 그룹 작성	84
8단계: dtappintegrate를 사용한 응용 프로그램 등록	89
등록 패키지 작성 예	92
"BestTextEditor"에 관한 필수 정보	92
"BestTextEditor"를 등록하는 단계	93
6 기타 구성	99
Solaris CDE 디렉토리 구조	99
/usr/dt	99
/etc/dt	100
/var/dt	100
\$HomeDirectory	100
키 환경설정 파일	101
Xconfig	101
Xservers	101
로그인 서버 시작	102
설치된 CDE를 다른 워크스테이션이나 네트워크 서버 설치 위치로부터 마운트하기	104
▼ 설치된 CDE 마운트 방법	104
▼ 마운트된 CDE 디렉토리를 마운트 해제하는 방법	104
데스크탑이 복수 화면을 사용하도록 구성	105
▼ 데스크탑이 복수 화면에서 시작되도록 하는 방법	105
네트워크화된 데스크탑	106
XTerminals 사용	108
로그인 로케일 및 글꼴 경로	109
워크스테이션을 XTerminals로 사용	109
▼ 호스트 CDE 로그인 선택을 위해 선택기를 사용하는 방법	110
▼ 특정 호스트 CDE 로그인을 사용하는 방법	110
▼ 첫번째 사용가능한 호스트 로그인을 사용하는 방법	110
특수 CDE 구성	110

	우편 인쇄 사용자 정의	111
	캘린더를 새 데이터 형식으로 전환	112
	네트워크에서 AnswerBook 패키지 추가	112
	CDE 데스크탑 외부에서 CDE 환경 설정하기	113
	데스크탑 환경 파일	113
	Apple Macintosh 응용 프로그램 환경에서 플로피 및 CD 매체 사용	114
7	네트워크에서 데스크탑 환경설정	117
	데스크탑 네트워킹 개요	117
	네트워크화된 데스크탑 서비스 유형	118
	전형적인 네트워크 환경	118
	기타 네트워킹 환경	121
	요약—서버 유형	122
	데스크탑 네트워킹 환경설정의 일반적 단계	123
	데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정	123
	사용자에게 로그인 계정 제공하기	123
	분산 파일 시스템 액세스의 환경설정	124
	원격 프린터에 대한 액세스 구성	125
	전자우편의 구성	125
	X 인증 구성	126
	데스크탑 클라이언트 및 서버의 환경설정	126
	로그인 및 세션 서비스 구성	126
	기타 응용 프로그램 관련 서비스 구성	127
	응용 프로그램 서비스의 관리	131
	검색 경로 환경 변수	131
	응용 프로그램 서버 및 클라이언트의 구성	131
	데이터베이스, 아이콘, 도움말 서비스의 구성	133
	특수 네트워크 응용 프로그램 환경설정	134
8	데스크탑에서 인쇄 구성 및 관리	139
	프린터 추가 및 삭제	139
	▼ 데스크탑에 프린터 추가 방법	139
	▼ 데스크탑에서 프린터 삭제 방법	140
	작업 갱신 간격 수정	140
	프린터 아이콘 이미지	141
	아이콘 파일 이름 및 크기	141
	▼ 프린터의 아이콘, 인쇄 레이블 또는 설명을 전체적으로 변경하는 방법	141

기본 프린터 구성	142
▼ 기본 인쇄를 위한 대상 변경 방법	142
인쇄 개념	143

9 데스크탑 검색 경로 145

데스크탑 검색 경로 및 환경 변수	146
검색 경로 값 설정	146
▼ 검색 경로의 현재 값을 보는 방법(출력 변수)	146
▼ 검색 경로를 개별적으로 수정하는 방법	147
▼ 검색 경로를 시스템 전체에서 수정하는 방법	147
응용 프로그램 검색 경로	148
기본 응용 프로그램 검색 경로	148
응용 프로그램 검색 경로 환경 변수	148
응용 프로그램 검색 경로 입력 변수에 대한 구문	148
응용 프로그램 검색 경로 값을 조합하는 방법	149
시스템 전체 지역 위치의 우선순위 변경	149
응용 프로그램 검색 경로가 데이터베이스, 아이콘 및 도움말 검색 경로에 미치는 영향	150
데이터베이스(활동/데이터 유형) 검색 경로	151
기본 데이터베이스 검색 경로	151
응용 프로그램 검색 경로가 데이터베이스 검색 경로에 미치는 영향	151
데이터베이스 검색 경로 환경 변수	151
데이터베이스 검색 경로 입력 변수에 대한 구문	152
데이터베이스 검색 경로를 조합하는 방법	152
아이콘 검색 경로	153
기본 아이콘 검색 경로	153
응용 프로그램 검색 경로가 아이콘 검색 경로에 미치는 영향	153
아이콘 검색 경로 환경 변수	153
아이콘 검색 경로 입력 변수에 대한 구문	154
아이콘 검색 경로를 조합하는 방법	154
도움말 검색 경로	154
기본 도움말 검색 경로	154
응용 프로그램 검색 경로가 도움말 검색 경로에 미치는 영향	155
도움말 검색 경로 환경 변수	155
도움말 검색 경로 입력 변수에 대한 구문	155
도움말 검색 경로를 조합하는 방법	155
지역화된 검색 경로	156

10	활동 및 데이터 유형 소개	157
	활동 소개	157
	활동이 응용 프로그램에 대한 아이콘을 작성하는 방법	159
	활동이 데이터 파일을 인수로 사용하는 방법	162
	활동의 다른 용도	163
	데이터 유형 소개	163
	데이터 유형이란?	163
	데이터 유형이 데이터 파일을 활동에 연결하는 방법	164
	데이터 유형에 대한 데스크탑 인쇄 작성	167
11	활동 만들기를 사용한 활동 및 데이터 유형 작성	169
	활동 만들기의 기능	169
	활동 만들기의 제한	170
	활동 제한	170
	데이터 유형 제한	171
	활동 만들기로 응용 프로그램에 대한 활동 및 데이터 유형 작성	171
	▼ 응용 프로그램의 활동을 작성하는 방법	171
	▼ 응용 프로그램에 대해 하나 이상의 데이터 유형을 작성하는 방법	174
	아이콘 세트 찾기 대화 상자를 사용하여 아이콘 지정	180
12	수동으로 활동 작성	183
	활동을 수동으로 작성해야 하는 이유	183
	COMMAND 활동	184
	MAP 활동	184
	TT_MSG(ToolTalk 메시지) 활동	184
	수동으로 활동 작성: 일반적 단계	185
	활동의 환경설정 파일	185
	▼ 활동을 수동으로 작성하는 방법	185
	COMMAND 활동 작성의 예	186
	MAP 활동 작성 예	187
	▼ 활동/데이터 유형 데이터베이스를 재로드하는 방법	188
	활동에 대한 활동 파일(아이콘) 작성	188
	활동이 사용하는 아이콘 이미지의 지정	189
	▼ 기존 활동 정의를 수정하는 방법	190
	활동 정의 우선순위	191
	COMMAND 활동에 대한 실행 문자열 구축	193
	실행 문자열의 일반적 특징	193

인수를 사용하지 않는 활동 작성	194
삭제된 파일을 허용하는 활동 작성	194
파일 인수에 대한 프롬프트를 표시하는 활동 작성	195
놓은 파일을 허용하거나 프롬프트를 표시하는 활동 작성	195
비파일 인수에 대해 프롬프트를 표시하는 활동 작성	196
파일 인수를 문자열로 해석하기	196
활동에 셸 기능 제공	196
복수 파일 인수에 대한 COMMAND 활동 작성	197
COMMAND 활동을 위한 윈도우 지원 및 터미널 애플레이터	199
활동을 위한 윈도우 지원의 지정	199
터미널 애플레이터에 명령행 옵션 지정	200
다른 기본 터미널 애플레이터 지정	200
특정 인수에 대한 활동 제한	201
지정된 데이터 유형에 대한 활동 제한	201
인수 수에 기초한 활동 제한	201
▼ 서로 다른 두 번 누르기 및 놓기 동작을 제공하는 방법	202
인수 모드에 기초한 활동 제한	203
원격 시스템에서 응용 프로그램을 실행하는 활동 작성	203
원격 응용 프로그램을 실행하는 활동 작성	203
활동 및 데이터 유형 정의에서 변수 사용	204
활동에서 문자열 변수 사용	204
활동 및 데이터 유형에서 환경 변수 사용	205
명령행에서 활동 호출	205
dtaction 구문	205
다른 활동을 실행하는 활동 작성	206
다른 사용자로서 실행하는 활동 작성	206
지역화된 활동 작성	207
지역화된 활동의 위치	207
▼ 기존 활동을 지역화하는 방법	207
ToolTalk 응용 프로그램에 대한 활동 작성	208
addressing 및 disposition 필드	208
지원되지 않는 메시지	208

13 수동으로 데이터 유형 작성 211

데이터 유형을 수동으로 작성해야 하는 이유	211
데이터 유형 정의의 구성요소: 기준 및 속성	212
수동으로 데이터 유형 작성: 일반적 단계	212

	데이터 유형에 대한 환경설정 파일	212
	▼ 데이터 유형 정의 작성 방법	213
	개인 활동 및 데이터 유형 작성의 예	214
	데이터 유형의 데이터 속성 정의	215
	데이터 유형에 사용되는 아이콘 이미지의 지정	215
	활동과 데이터 유형 연결	216
	데이터 유형에 기초한 파일 숨기기	217
	파일을 조작할 때의 동작 지정	217
	데이터 유형에 대한 데이터 기준 정의	218
	이름 기반 데이터 유형	218
	위치 기반 데이터 유형	219
	이름 및 위치에 기초한 데이터 유형	220
	파일 모드를 유형 지정 기준으로 사용	220
	내용 기반 데이터 유형 지정	222
	▼ 다양한 독립적 기준으로 데이터 유형을 작성하는 방법	223
	지역화된 데이터 유형 작성	223
	지역화된 데이터 유형의 위치	223
	▼ 데이터 유형을 지역화하는 방법	224
14	데스크탑 아이콘 작성	225
	아이콘 이미지 파일	225
	아이콘 파일 형식	226
	아이콘 파일 이름	226
	아이콘 크기 규약	226
	아이콘 검색 경로	227
	네트워크를 통한 아이콘 액세스	227
	아이콘 연결	228
	아이콘 파일 지정	228
	▼ 아이콘을 활동 또는 데이터 유형에 연결하는 방법	228
	▼ 아이콘을 프론트 패널 제어도구에 표시하는 방법	229
	▼ 아이콘을 응용 프로그램 윈도우에 연결하는 방법	229
	▼ 파일 관리자를 아이콘 브라우저로 사용하는 방법	230
	아이콘 디자인 권장사항	231
	색상 사용	231
15	고급 프론트 패널의 사용자 정의	233
	프론트 패널 환경설정 파일	233

기본 프론트 패널 환경설정 파일	234
프론트 패널 환경설정 파일의 검색 경로	234
프론트 패널이 조합되는 방법: 우선순위 규칙	235
동적으로 작성되는 프론트 패널 파일	235
사용자 인터페이스 사용자 정의 관리	236
▼ 개인 사용자 정의를 금지하는 방법	236
▼ 삭제된 제어도구 또는 서브패널을 복구하는 방법	236
프론트 패널 정의 구성	237
프론트 패널 구성요소	237
프론트 패널 정의의 일반 구문	237
주 패널 수정	240
▼ 주 패널에 제어도구를 추가하는 방법	240
▼ 제어도구 삭제 방법	241
▼ 제어도구 수정 방법	242
▼ 제어도구 위치 교환 방법	242
▼ 프론트 패널 제어도구 교체 방법	243
제어도구가 사용하는 아이콘 지정	243
서브패널의 작성 및 수정	244
▼ 새로운 시스템 전체 서브패널 작성 방법	245
내장 서브패널 사용자 정의	245
▼ 서브패널의 자동단기 동작 변경 방법	247
프론트 패널 제어도구 정의	247
프론트 패널 제어도구 정의	248
제어도구 유형	248
▼ 새로운 제어도구 작성 방법	248
작업 공간 스위치의 사용자 정의	254
▼ 작업 공간의 기본 수 변경 방법	254
▼ 스위치 행 수 변경 방법	255
▼ 작업 공간 스위치에서 제어도구를 변경하거나 추가하는 방법	255
일반 프론트 패널 환경설정	256
일반적 단계	256
▼ 기본 프론트 패널 위치 변경 방법	256
▼ 기본 패널의 제어도구에 레이블을 표시하는 방법	257
▼ 제어도구의 누르기 동작 변경 방법	257
▼ 완전히 새로운 프론트 패널 작성 방법	257
3행을 갖는 개인 프론트 패널을 작성하는 예	257

16	작업 공간 관리자의 사용자 정의	259
	작업 공간 관리자 환경설정 파일	260
	▼ 개인 환경설정 파일의 작성 또는 수정 방법	260
	▼ 시스템 전체 환경설정 파일 작성 방법	261
	▼ 다른 파일을 포함하는 방법	261
	▼ 작업 공간 관리자를 재시작하는 방법	262
	작업 공간 사용자 정의	262
	▼ 시스템 전체의 작업 공간 수를 변경하는 방법	262
	▼ 시스템 전체의 작업 공간 이름을 제공하는 방법	263
	▼ 추가 배경 작성 방법	263
	▼ 그래픽 이미지로 배경을 대체하는 방법	263
	작업 공간 관리자 메뉴	264
	작업 공간 관리자 메뉴 구문	264
	▼ 작업 공간 메뉴에 새 메뉴 항목을 추가하는 방법	265
	▼ 작업 공간 메뉴를 수정하는 방법	266
	▼ 새 작업 공간 (루트) 메뉴 작성 방법	267
	▼ 새 윈도우 메뉴 작성 방법	267
	버튼 바인딩 사용자 정의	268
	버튼 바인딩 구문	268
	▼ 버튼 바인딩을 추가하는 방법	269
	▼ 새 버튼 바인딩 세트 작성 방법	269
	키 바인딩의 사용자 정의	270
	기본 데스크탑 키 바인딩	270
	키 바인딩 구문	270
	▼ 사용자 키 바인딩 세트 작성 방법	271
	기본 동작과 사용자 정의 동작 사이의 전환	272
17	응용 프로그램 자원, 글꼴, 색상의 관리	273
	응용 프로그램 자원 설정	273
	▼ 시스템 전체 자원 설정 방법	274
	▼ 개인 자원 설정 방법	274
	데스크탑이 자원을 로드하는 방법	274
	프로세스 관리자 자원	274
	UNIX 바인딩 정의	275
	▼ EMACS 스타일 변환을 지정하는 방법	275
	▼ EMACS 스타일 변환을 수정하는 방법	275
	UNIXBindings 파일이 제공하는 UNIX 바인딩	275

글꼴 관리	279
데스크탑 글꼴 자원 설정	279
▼ 사용가능한 글꼴을 나열하는 방법	280
▼ 명령행에서 글꼴을 지정하는 방법	281
X 논리적 글꼴 설명(XLFD)	281
사용자 글꼴 그룹 파일 시스템 저장	283
시스템 관리자 글꼴 그룹 작성	283
색상 관리	283
색상 팔레트	283
색상 세트	284
스타일 관리자를 사용하여 색상 제어	287
스타일 관리자가 사용하는 색상 수	287
응용 프로그램 윈도우의 음영 농도 설정	290

18	지역화된 데스크탑 세션 구성	291
	LANG 환경 변수 관리	291
	복수 사용자를 위한 언어 설정	292
	단일 세션에 대한 언어 설정	293
	한 사용자를 위한 언어 설정	293
	LANG 환경 변수 및 세션 구성	293
	다른 NLS 환경 변수 설정	293
	글꼴 찾기	294
	app-defaults 자원 파일 지역화하기	294
	활동 및 데이터 유형 지역화하기	295
	아이콘 및 비트맵 지역화하기	295
	배경 이름 지역화하기	295
	팔레트 이름 지역화하기	296
	도움말 볼륨 지역화하기	297
	메시지 카탈로그 지역화하기	297
	지역화된 데스크탑 응용 프로그램을 원격으로 실행하기	297
	사용자 키보드 맵 재설정	298

A dtconfig(1) Man Page 299

색인	301
----	-----

머리말

이 설명서는 Solaris 공통 데스크탑 환경(CDE)의 모양과 동작을 사용자 정의하는 고급 작업들에 대해 설명합니다. 이 설명서는 다음과 같은 장으로 구성되어 있습니다.

- 시스템 초기화, 로그인 및 세션 개시의 사용자 정의
- 응용 프로그램을 추가하고 응용 프로그램 및 데이터의 인터페이스 표현 제공하기
- 네트워크를 통한 데스크탑 프로세스, 응용 프로그램 및 데이터의 구성
- 윈도우 관리, 인쇄, 색상 및 글꼴 등과 같은 데스크탑 서비스의 사용자 정의

주 - 이 설명서에서 "IA"는 Intel 32-비트 프로세서 구조를 의미하며, 여기에는 Pentium, Pentium Pro, Pentium II, Pentium II Xeon, Celeron, Pentium III, Pentium III Xeon 프로세서, AMD 및 Cyrix가 제조한 호환 마이크로 프로세서 칩 등이 포함됩니다.

이 책의 사용자

이 책의 사용자는 다음과 같습니다.

- 시스템 관리자 이 책의 여러 작업들을 수행하려면 루트 사용 권한이 있어야 합니다.
- 데스크탑 사용자 인터페이스로는 불가능한 사용자 정의를 수행하려는 고급 사용자. 데스크탑은 여러 환경설정 파일에 사용자별 위치를 제공합니다.

이 책을 읽기 전에

사용자는 다음과 같은 책 내용을 잘 알고 있어야 합니다.

- Solaris 공통 데스크탑 환경: 사용 설명서
 - Solaris 공통 데스크탑 환경: 사용자 변환 설명서
-

이 책의 구성

이 책은 다음과 같은 여러 장으로 구성되어 있습니다.

제 1 장에서는 데스크탑 로그인 관리자의 모양과 동작을 구성하는 방법을 설명합니다.

제 2 장에서는 데스크탑이 세션을 저장하고 불러오는 방법과 세션 시작을 사용자 정의하는 방법을 설명합니다.

제 3 장에서는 Solaris CDE 시작 파일, Solaris CDE 시작시 발생할 수 있는 문제를 설명하고 시작 문제의 해결책을 제시합니다.

제 4 장에서는 응용 프로그램 관리자가 응용 프로그램을 모으는 방법과 응용 프로그램을 추가하는 방법을 설명합니다.

제 5 장에서는 응용 프로그램용 등록 패키지를 작성하는 방법을 설명합니다.

제 6 장에서는 사용자 정의 로그인 구성, 복수 화면 설정, 네트워크화된 데스크탑과 X 터미널, 사용자 도트 파일 수정, 우편 인쇄 사용자 정의, 데스크탑 구성 및 오류 로그 유형 등과 같은 구성에 관한 고급 주제들을 설명합니다.

제 7 장에서 데스크탑 환경설정에서는 네트워크를 통해 데스크탑 서비스, 응용 프로그램 및 데이터를 분산하는 방법을 설명합니다.

제 8 장에서 인쇄 구성 및 관리에서는 데스크탑 프린터를 추가하고 제거하는 방법, 기본 프린터를 지정하는 방법 등을 설명합니다.

제 9 장에서는 데스크탑이 네트워크에서 응용 프로그램, 도움말 파일, 아이콘 및 기타 데스크탑 데이터를 찾는 방법을 설명합니다.

제 10 장에서는 활동 및 데이터 유형의 개념을 소개하고, 이들이 응용 프로그램 사용자 인터페이스를 제공하기 위해 어떻게 사용되는지를 설명합니다.

제 11 장에서는 활동 작성 응용 프로그램을 사용하여 활동 및 데이터 유형을 작성하는 방법을 설명합니다.

제 12 장에서는 데이터베이스 환경설정 파일을 편집하여 활동 정의를 작성하는 방법을 설명합니다.

제 13 장에서는 데이터베이스 환경설정 파일을 편집하여 데이터 유형 정의를 작성하는 방법을 설명합니다.

제 14 장에서는 아이콘 편집기를 사용하는 방법과 데스크탑 아이콘의 검색 경로, 크기 및 이름 지정 규약을 설명합니다.

제 15 장에서는 새로운 시스템 전체 제어도구와 서브패널을 작성하는 방법과 기타 패널을 사용자 정의하는 방법을 설명합니다.

제 16 장에서는 윈도우의 사용자 정의, 마우스 버튼 바인딩, 키보드 바인딩 및 작업 공간 관리자 메뉴에 대해 설명합니다.

제 17 장에서는 응용 프로그램 자원을 설정하는 방법과 데스크탑이 글꼴과 색상을 사용하는 방법에 대해 설명합니다.

제 18 장에서는 국제 세션을 수행하는 시스템에 대한 시스템 관리 작업을 설명합니다.

부록 A는 dtconfig(1) 매뉴얼 페이지의 사본입니다.

Sun 문서의 주문 방법

인터넷 전문서점인 Fatbrain.com이 Sun Microsystems, Inc.의 제품 설명서를 선정하고 있습니다.

책 목록과 주문 방법에 대해서는

<http://www1.fatbrain.com/documentation/sun>에서 Sun Documentation Center를 방문하십시오.

Sun 문서 온라인 액세스

docs.sun.comSM 웹사이트를 통해 Sun 기술 설명서 온라인에 액세스할 수 있습니다.

docs.sun.com 아카이브를 찾거나 특정 책 제목 또는 주제를 검색할 수 있습니다. URL은 <http://docs.sun.com>입니다.

활자체 규약

다음 표에는 이 책에서 사용하는 활자체 변경사항을 설명합니다.

표 P-1 활자체 규약

서체 또는 기호	의미	예
AaBbCc123	컴퓨터 화면상의 출력. 명령어, 파일 및 디렉토리 이름	.login 파일을 편집하십시오. ls -a를 사용하여 모든 파일을 나열하십시오. machine_name% 메일이 왔습니다.
AaBbCc123	컴퓨터의 화면상 출력과 대비되는 입력사항	machine_name% su Password:
AaBbCc123	명령줄에서의 대체기호. 실제 이름이나 값으로 대체하십시오.	파일을 삭제하려면 rm filename 을 입력하십시오.
AaBbCc123	책 제목, 새 단어, 용어, 강조할 단어 등.	사용자 설명서 의 6장을 참조하십시오. 이것은 <i>class</i> 옵션입니다. 이 작업을 수행하려면 사용자가 <i>root</i> 여야 합니다.

명령어 쉘 프롬프트의 예

다음 표는 C 쉘, Bourne 쉘 및 Korn 쉘의 기본 시스템 프롬프트와 슈퍼유저 프롬프트를 나타냅니다.

표 P-2 쉘 프롬프트

셸	프롬프트
C 쉘 프롬프트	machine_name%
C 쉘 슈퍼유저 프롬프트	machine_name#
Bourne 쉘 및 Korn 쉘 프롬프트	\$
Bourne 쉘 및 Korn 쉘 슈퍼유저 프롬프트	#

로그인 관리자 구성

로그인 관리자는 로그인 화면을 표시하고, 사용자를 인증하고, 사용자 세션을 시작하도록 할 책임이 있는 서버입니다. 그래픽 방식의 로그인은 비트맵 표시장치에서 기존의 문자 모드 로그인을 대체하는 편리한 방식입니다. 로그인 서버에 의해 관리되는 표시장치는 로그인 서버에 직접 연결되거나 또는 네트워크상의 X 터미널이나 워크스테이션에 연결됩니다.

주 - 로그인 서버를 시작, 중지, 또는 사용자 정의하려면 루트 사용자여야 합니다.

- “로그인 서버 시작” 20 페이지
- “지역 표시장치 및 네트워크 표시장치 관리” 20 페이지
- “오류 점검” 26 페이지
- “로그인 서버 중지” 27 페이지
- “로그인 화면 모양 변경” 28 페이지
- “로그인 관리자 관리” 36 페이지
- “로그인 관리자 파일” 36 페이지

로그인 서버가 수행하는 일은 다음과 같습니다.

- 로그인 화면을 비트맵 표시장치에 표시할 수 있습니다.
- 직접 연결된 문자 콘솔 표시장치를 허용합니다.
- 사용자가 네트워크상의 다른 로그인 서버로부터 로그인 화면을 표시할 수 있는 선택 키 화면을 표시할 수 있습니다.
- 로그인 서버에 대한 액세스를 통제할 수 있습니다.
- 기존 문자 모드 로그인에 대한 액세스를 제공합니다.

로그인 관리자가 관리하는 표시장치는 로그인 관리자 서버나 네트워크상의 X 터미널 또는 워크스테이션에 직접 연결할 수 있습니다. 지역 표시장치의 경우, 로그인 서버가 자동으로 X 서버를 시작한 다음, 로그인 화면을 표시합니다. X 터미널과 같은 네트워크 표시장치의 경우, 로그인 서버는 X 표시장치 관리자 프로토콜(XDMCP) 1.0을 지원하며, 표시장치는 이 프로토콜을 사용하여 로그인 서버가 표시장치에 로그인 화면을 표시하도록 요청할 수 있습니다.

로그인 서버 시작

로그인 서버는 일반적으로 시스템이 부트될 때 시작됩니다. 로그인 서버는 또한 명령행에서 시작할 수도 있습니다.

- 로그인 서버가 시스템이 부트될 때 시작하도록 설정하려면 다음을 입력하십시오.
`/usr/dt/bin/dtconfig -e`
그러면 시스템을 재부트할 때 로그인 서버가 자동으로 시작됩니다.

데스크탑 구성 유틸리티, `dtconfig`에 대한 자세한 내용은 부록 A를 참조하십시오. 부록 A에는 `dtconfig.1` 매뉴얼 페이지의 사본이 수록되어 있습니다.

- 명령행에서 로그인 서버를 시작하려면 다음을 입력하십시오.
`/usr/dt/bin/dtlogin -daemon; exit`

주 - 임시 환경설정 시험을 위해 명령행에서 로그인 서버를 시작할 수는 있지만, 로그인 서버는 시스템을 부트할 때 정상적으로 시작해야 합니다.

지역 표시장치 및 네트워크 표시장치 관리

그림 1-1은 로그인 서버 구성의 한 예입니다.

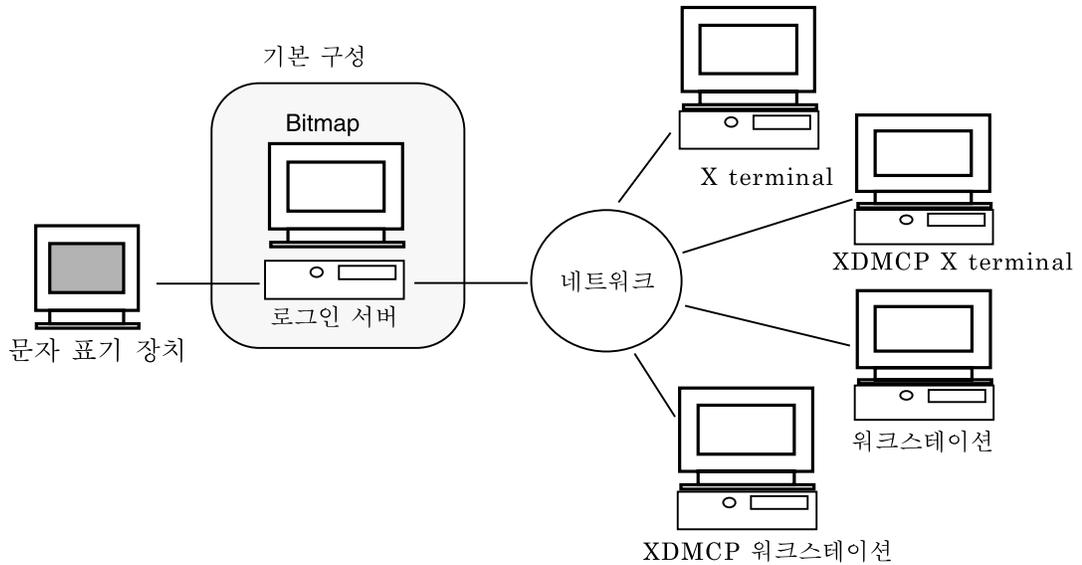


그림 1-1 로그인 서버 구성의 예

로그인 서버 프로세스 ID 찾기

기본적으로 로그인 서버는 프로세스 ID를 `/var/dt/Xpid`에 저장합니다.

이를 변경하려면 `Xconfig` 파일에서 `Dtlogin.pidFile` 자원을 설정합니다. 변경된 경우에는 지정된 디렉토리가 로그인 서버 시작시에 존재해야 합니다.

`Xconfig`를 수정하려면, `Xconfig`를 `/usr/dt/config`에서 `/etc/dt/config`로 복사하십시오. `/etc/dt/config/Xconfig`를 수정한 후에 다음을 입력하여 로그인 서버에 `Xconfig`를 다시 읽도록 알려주십시오.

```
/usr/dt/bin/dtconfig -reset
```

이렇게 하면 명령어 `kill -HUP login_server_process_ID`가 수행됩니다.

예를 들어, 로그인 서버 프로세스 ID를 `/var/myserver/Dtpid`에 저장하려면, `Xconfig` 파일에 다음을 설정하십시오.

```
Dtlogin.pidFile: /var/myserver/Dtpid
```

로그인 서버를 다시 시작할 때 로그인 서버가 프로세스 ID를 `/var/myserver/Dtpid`에 저장합니다. 로그인 서버가 시작할 때 `/var/myserver` 디렉토리가 존재해야 합니다.

로그인 화면을 지역 표시장치에 표시

시작시, 로그인 서버는 `Xservers` 파일을 점검하여 X 서버를 시작해야 하는지를 결정하고, 로그인 화면이 지역 또는 네트워크 표시장치에 표시되어야 하는지 여부와 그 표시 방법을 결정합니다.

`Xservers`를 수정하려면, `Xservers`를 `/usr/dt/config`에서 `/etc/dt/config`로 복사하십시오. `/etc/dt/config/Xservers`를 수정한 후에 다음을 입력하여 로그인 서버에 `Xservers`를 다시 읽도록 알려주십시오.

```
/usr/dt/bin/dtconfig -reset
```

이렇게 하면 명령어 `kill -HUP login_server_process_ID`가 수행됩니다.

`Xservers` 행의 형식은 다음과 같습니다.

```
display_name  
display_class  
display_type  
X_server_command
```

여기서

display_name—X 서버에 연결할 때 사용할 연결 이름을 로그인 서버에 알려줍니다(다음 예에서는 :0). 별표(*) 값은 호스트명:0으로 확장됩니다. 지정된 숫자는 *X_server_command* 연결 번호에서 지정된 번호와 일치해야 합니다.

display_class—이 표시장치에 특정한 자원을 식별합니다(다음 예에서는 Local).

display_type—표시장치가 지역 또는 네트워크 표시 장치인지와 로그인 화면에서 명령행 로그인 옵션을 관리하는 방법을 로그인 서버에 알려줍니다(다음 예에서는 local @console).

X_server_command—로그인 서버가 X 서버를 시작하기 위해 사용할 명령행, 연결 번호 및 기타 옵션을 식별합니다(다음 예에서는 /usr/bin/X11/X:0). 지정된 연결 번호는 *display_name*에서 지정된 숫자와 일치해야 합니다.

기본 `Xservers` 행은 다음과 유사합니다.

```
:0 Local local@console /usr/bin/X11/X :0
```

지역 표시장치 없이 로그인 서버 실행

로그인 서버 시스템에 비트맵 표시장치가 없는 경우, #(파운드 부호)를 사용하여 지역 표시장치에 대한 `Xservers` 행을 주석으로 만들어 로그인 서버를 지역 표시장치 없이 실행하십시오. 예를 들면 다음과 같습니다.

```
# :0 Local local@console /usr/bin/X11/X :0
```

로그인 서버가 시작될 때, 네트워크 표시장치의 요청을 기다리며 백그라운드에서 실행됩니다.

지역 표시장치에서 명령행 로그인 액세스

사용자가 로그인 화면에서 명령행 로그인을 선택한 경우, 로그인 서버는 일시적으로 X 서버를 종료하여 비트맵 표시 터미널 장치에서 실행되는 이전의 명령행 로그인에 액세스할 수 있게 합니다. 사용자가 로그인했다가 로그아웃한 후, 또는 지정된 시간이 종료한 후에 로그인 서버는 X 서버를 다시 시작하게 합니다.

주 - 명령행 로그인 옵션은 네트워크 표시장치에서는 사용할 수 없습니다.

*display_type*은 명령행 로그인의 동작을 제어합니다. *display_type*의 형식은 다음과 같습니다.

- `local@display_terminal_device`
- `local`
- `foreign`

`local@display_terminal_device`가 지정되면, 로그인 서버는 X 서버와 `/dev/display_terminal_device`가 동일한 물리적 장치에 있으며, 명령행 로그인(일반적으로 `getty`)이 이 장치에서 실행되고 있다고 가정합니다. 사용자가 명령행 로그인을 선택하면, X 서버가 종료되어 `/dev/display_terminal_device`에서 실행되는 실행중인 명령행 로그인(`getty`)에 액세스할 수 있게 합니다.

표시장치에서 명령행 로그인 옵션을 사용하지 않으려면, *display_terminal_device*를 `none`로 지정하십시오. 기본 *display_terminal_device*는 `console`입니다. `local`이 지정되면 *display_terminal_device*는 기본적으로 `console`이 됩니다. `foreign`이 지정되면, 명령행 로그인을 사용할 수 없습니다.

주 - 로그인 서버가 명령행에서 시작될 때는 명령행 로그인 옵션을 지역 표시장치에서 사용할 수 없게 됩니다.

문자 표시 콘솔 허용

로그인 서버 시스템이 콘솔 역할을 하는 문자 표시장치에 직접 연결된 경우, 비트맵 표시 로그인 화면에서 명령행 로그인을 사용 불가로 설정하기 위해 *display_terminal_device*를 `none`로 설정할 수도 있습니다.

그렇지 않으면, 명령행 로그인(`getty`)이 문자 표시 콘솔과 비트맵 표시장치 모두에서 실행 중인 경우 비트맵 표시장치에서 *display_terminal_device*를 명령행 로그인(`getty`) 장치로 변경할 수 있습니다.

예를 들어, 비트맵 표시장치 명령행 로그인(`getty`)이 장치 `/dev/tty01`에 있으면, *display_type*을 `local@tty01`로 변경하십시오.

네트워크 표시장치에 로그인 화면 표시

로그인 서버는 로그인 화면을 특정 표시장치에 표시하려는 네트워크 표시장치의 요청을 승인할 수 있습니다. 네트워크 표시장치는 일반적으로 X 터미널이지만 워크스테이션이 될 수도 있습니다.

네트워크 표시장치의 요청을 관리하기 위해 로그인 서버는 X 표시장치 관리자 프로토콜 (XDMCP) 1.0을 지원합니다. 이 프로토콜을 통해 로그인 서버는 네트워크 표시장치의 요청을 협상하여 승인하거나 거부할 수 있습니다. 대부분의 X 터미널에는 XDMCP가 내장되어 있습니다.

XDMCP 네트워크 표시장치의 직접 요청

X 터미널이 XDMCP 직접(조회 모드)을 사용하도록 구성한 경우, X 터미널에 로그인 서버 호스트의 호스트명을 알려주어야 합니다. X 터미널은 부트되면 자동으로 로그인 서버에 연결하고 로그인 서버는 로그인 화면을 X 터미널에 표시합니다. X 터미널을 XDMCP 직접 모드로 구성하는 방법에 관한 내용은 사용자의 X 터미널 설명서를 참조하십시오.

대부분의 X 서버는 `-query` 옵션도 지원합니다. 이 모드에서, 사용자의 X 서버는 X 터미널인 것처럼 동작하여, 로그인 서버 호스트를 직접 연결하고 로그인 화면을 X 서버에 표시할 것을 요청합니다. 예를 들어, X 서버를 워크스테이션 `bridget`의 비트맵 표시장치에서 시작하면 로그인 서버 `anita`가 X 서버에 로그인 화면을 표시할 것입니다.

```
X -query anita
```

XDMCP 네트워크 표시장치의 간접 요청

X 터미널이 XDMCP 간접 모드를 사용하도록 구성한 경우, X 터미널에 로그인 서버 호스트의 호스트명을 알려주어야 합니다. X 터미널은 부트되면 로그인 서버에 연결하고, 로그인 서버는 선택기 화면을 통해 네트워크상의 다른 로그인 서버 호스트의 목록을 표시합니다. 이 목록에서 사용자가 호스트를 선택하면 호스트는 로그인 화면을 사용자의 X 터미널에 표시할 것입니다. X 터미널을 XDMCP 간접 모드로 구성하는 방법에 관한 내용은 사용자의 X 터미널 설명서를 참조하십시오.

대부분의 X 서버는 직접 모드와 함께 `-indirect` 옵션을 지원하므로 X 서버는 XDMCP 간접 모드로 로그인 서버를 연결할 수 있습니다.

XDMCP를 지원하지 않는 네트워크 표시장치 관리

이전 X 터미널은 XDMCP를 지원하지 않을 수도 있습니다. 로그인 서버가 이런 유형의 X terminal에 로그인 화면을 표시하도록 하려면, X 터미널 이름을 `Xservers` 파일에 나열하십시오.

표시장치는 네트워크상에 있으므로 *display_name*은 이름의 일부에 호스트명을 포함합니다. *display class*는 특정 클래스의 X 터미널에 고유한 자원을 지정하는데 사용할 수 있습니다. (사용자의 X 터미널 설명서에 X 터미널의 표시장치 클래스가 표시되어 있습니다.) 이기종인 *display_type*은 로그인 서버에게 직접 시작하지 말고 기존의 X 서버에 연결하도록 알려줍니다. 이 경우, *X_server_command*는 지정되지 않습니다.

예

Xservers 파일의 다음 행들은 로그인 서버가 XDMCP를 지원하지 않는 X 터미널인 ruby와 wolfie에 로그인 화면을 표시하도록 지시합니다.

```
ruby.blackdog.com:0 AcmeXsta foreign wolfie:0 PandaCo foreign
```

로그인 서버에 대한 액세스 제어

기본적으로, 로그인 서버 호스트에 액세스할 수 있는 네트워크상의 모든 호스트는 로그인 화면이 표시되도록 요청할 수 있습니다. Xaccess 파일을 수정하여 로그인 서버에 대한 액세스를 제한할 수 있습니다.

Xaccess를 수정하려면, Xaccess를 /usr/dt/config에서 /etc/dt/config로 복사하십시오. /etc/dt/config/Xservers를 수정한 후에 다음을 입력하여 로그인 서버에 Xservers를 다시 읽도록 알려주십시오.

```
/usr/dt/bin/dtconfig -reset
```

이렇게 하면 명령어 `kill -HUP login server process ID`가 수행됩니다.

XDMCP 직접

호스트가 XDMCP 직접을 통해 로그인 서버에 연결하려고 시도하는 경우, 호스트명을 Xaccess 항목과 비교하여 호스트가 로그인 서버에 액세스할 수 있는지 결정됩니다. 각 Xaccess 항목은 와일드카드 *(별표)와?(물음표)를 포함하는 호스트명입니다. *(별표)는 모든 문자에 해당하며?(물음표)는 단일 문자에 해당합니다.!(느낌표)가 항목 앞에 있으면 액세스를 허용하지 않으며, 없는 경우는 액세스를 허용합니다.

예를 들어, say Xaccess에는 다음과 같은 세 항목이 포함됩니다.

```
amazon.waterloo.com *.dept5.waterloo.com !*
```

첫번째 항목은 호스트 amazon.waterloo.com에서 로그인 서버로 액세스할 수 있으며, 두번째 항목은 전체 도메인 이름이 dept5.waterloo.com으로 끝나는 모든 호스트의 액세스를 허용하며, 마지막 항목은 그 외의 어떤 호스트의 액세스도 허용하지 않습니다.

XDMCP 간접

호스트가 XDMCP 간접을 통해 로그인 서버에 연결하려고 시도하는 경우, 호스트가 로그인 서버에 액세스할 수 있는지 결정하기 위해 호스트명을 `xaccess` 항목과 비교합니다. 각 `xaccess` 항목은 CHOOSER 문자열로 표시된 것을 제외하고는 XDMCP 직접 항목과 유사하며 와일드카드를 포함합니다. 예를 들면 다음과 같습니다.

```
amazon.waterloo.com CHOOSER BROADCAST
*.dept5.waterloo.com CHOOSER BROADCAST
!* CHOOSER BROADCAST
```

역시, 첫번째 항목은 호스트 `amazon.waterloo.com`로부터 로그인 서버로 액세스할 수 있으며, 두번째 항목은 전체 도메인 이름이 `dept5.waterloo.com`으로 끝나는 호스트의 액세스를 허용하며, 마지막 항목은 어떤 호스트의 액세스도 허용하지 않습니다.

다음 중 하나가 CHOOSER 다음에 올 수 있습니다.

- BROADCAST
- 호스트 이름 목록

BROADCAST는 로그인 서버에게 로그인 서버 하위 네트워크에 사용 가능한 로그인 서버 호스트 목록을 작성하도록 브로드캐스트하도록 지시합니다. 호스트명 목록은 로그인 서버에게 이 목록을 사용 가능한 로그인 호스트 목록으로 사용하도록 알려줍니다. 예를 들어,

```
amazon.waterloo.com CHOOSER shoal.waterloo.com alum.waterloo.com
*.dept5.waterloo.com CHOOSER BROADCAST
!* CHOOSER BROADCAST
```

`amazon.waterloo.com`이 XDMCP 간접을 통해 연결되면, `shoal`과 `alum`을 포함한 목록이 표시될 것입니다. `alice.dept5.waterloo.com`이 연결되면, 로그인 서버 하위 네트워크상에 모든 사용 가능한 로그인 서버 호스트 목록과 함께 표시됩니다. 다른 XDMCP 간접 요청은 거부됩니다.

호스트명 목록을 지정하는 또 다른 방법은 호스트명 목록을 포함하는 하나 이상의 매크로를 정의하는 것입니다. 예를 들면 다음과 같습니다.

```
%list1 shoal.waterloo.com alum.waterloo.com
amazon.waterloo.com CHOOSER %list1
```

오류 점검

기본적으로, 로그인 서버는 오류를 `/var/dt/Xerrors` 파일에 기록합니다. 이를 변경하려면 `Xconfig` 파일에 `Dtlogin.errorLogFile` 자원을 설정하면 됩니다. 로그인 서버가 시작될 때 지정된 디렉토리가 존재해야 합니다.

예를 들어, 로그인 서버가 오류를 `/var/mylogs/Dterrors` 파일에 저장하려면, `Xconfig` 파일에 다음을 설정하십시오.

Dtlogin.errorLogFile: /var/mylogs/Derrors

로그인 서버를 다시 시작할 때 로그인 서버가 오류를 /var/mylogs/Derrors 파일에 저장합니다. /var/mylogs 디렉토리는 로그인 서버가 시작할 때 존재해야 합니다.

로그인 서버 중지

- 시스템 부트시 로그인 서버가 시작되지 않도록 하려면, 다음을 입력하십시오.

```
/usr/dt/bin/dtconfig -d
```

이렇게 하면 다음에 재부트할 때 시스템이 로그인 서버를 시작하지 않게 됩니다.

- 프로세스 ID를 강제 종료하여 로그인 서버를 중지하려면 다음을 입력하십시오.

```
/usr/dt/bin/dtconfig -kill
```

이렇게 하면 명령어 `kill login_server_process_ID`가 실행됩니다.

주 - 로그인 서버 프로세스를 강제 종료하면 로그인 서버가 관리하는 모든 사용자 세션이 종료됩니다.

프로세스 ID를 강제 종료하여 로그인 서버를 중지할 수도 있습니다. 로그인 서버 프로세스 ID는 /var/dt/Xpid에 저장되거나, Dtlogin.pidFile 자원이 xconfig에 지정된 파일에 저장됩니다.

로그인 서버를 강제 종료할 때 데스크탑에 로그인하면 데스크탑 세션이 즉시 종료됩니다.

로그인 화면

로그인 서버가 표시하는 로그인 화면은 이전의 문자 모드 로그인 화면의 효과적인 대안으로서 문자 모드 로그인에서 제공한 기능 이상을 제공합니다.

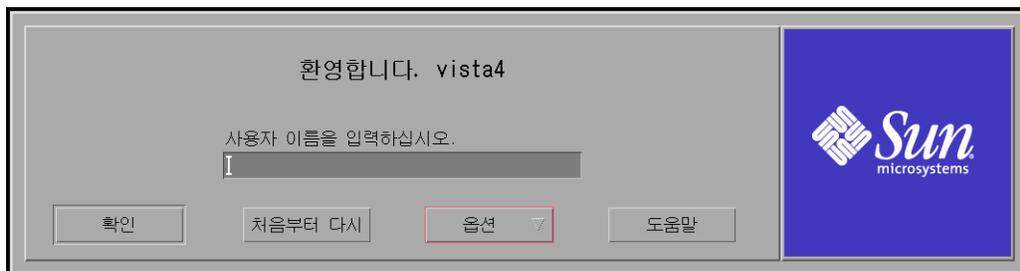


그림 1-2 데스크탑 로그인 화면

문자 모드 로그인과 마찬가지로 사용자는 사용자 이름을 입력한 후에 암호를 입력합니다. 인증을 받으면 로그인 서버가 사용자에게 대한 데스크탑 세션을 시작합니다. 사용자가 데스크탑 세션을 종료하면, 로그인 서버는 새 로그인 화면을 표시하며 프로세스가 다시 시작됩니다.

로그인 화면을 사용자 정의하는 방법

- 로그인 화면 모양을 변경하십시오.
- X 서버 권한을 구성하십시오.
- 기본 언어를 변경하십시오.
- 로그인 화면을 표시하기 전에 명령어를 수행하십시오.
- 로그인 화면 언어 메뉴의 내용을 변경하십시오.
- 사용자의 세션을 시작하는 명령어를 지정하십시오.
- 사용자의 데스크탑 세션을 시작하기 전에 명령어를 수행하십시오.
- 사용자의 세션이 종료한 후에 명령어를 수행하십시오.

이러한 작업은 각각 모든 표시장치에서 또는 개별적인 표시장치에서 수행할 수 있습니다.

로그인 화면 모양 변경

로그인 화면 모양을 사용자 정의하려면, 로고, 그래픽, 환영 메시지, 글꼴 등을 변경하면 됩니다.

Xresources를 수정하려면, Xresources를 `/usr/dt/config/language`에서 `/etc/dt/config/language`로 복사하십시오. 다음 번에 로그인 화면을 표시할 때, 모든 변경 사항이 로그인 화면에 반영됩니다. 로그인 화면을 다시 표시하게 하려면 로그인 화면 옵션 메뉴에서 로그인 화면 재설정을 선택하십시오.

Xresources 파일의 자원 사양에 따라 결정되는 로그인 화면 속성은 다음과 같습니다.

`Dtlogin*logo*bitmapFile`—로고 이미지로 표시할 비트맵 또는 픽스맵 파일

Dtlogin*greeting*persLabelString—개인별 환영 메시지
 Dtlogin*greeting*labelString—환영 메시지
 Dtlogin*greeting*fontList 환영 메시지의 글꼴
 Dtlogin*labelFont 누름 버튼 및 레이블의 글꼴
 Dtlogin*textFont 도움말 및 오류 메시지의 글꼴
 Dtlogin*language*languageName 로케일 이름 *language*의 대체 텍스트

▼ 로고를 변경하는 방법

- Xresources에서 Dtlogin*logo*bitmapFile 자원을 설정하십시오.

로고는 색상 픽스맵 또는 비트맵 파일일 수 있습니다.

다음 예에서는 Mylogo 비트맵을 로고로 사용합니다.

```
Dtlogin*logo*bitmapFile: /usr/local/lib/X11/dt/bitmaps/Mylogo.bm
```

▼ 환영 메시지 변경 방법

기본적으로, 로그인 서버는 환영합니다! 호스트: **호스트명**이라는 메시지를 로그인 화면에 표시합니다. 이 메시지를 변경하려면

- Xresources에 Dtlogin*greeting*labelString 자원을 설정하십시오.

labelString 자원 값에는 로그인 서버 호스트명으로 대체할 수 있는% *LocalHost*%와, X 서버 표시장치 이름으로 대체할 수 있는% *DisplayName*%이 포함될 수 있습니다.

다음 예에서는 환영 메시지를 Here's *host name*!으로 변경하고 있습니다.

```
Dtlogin*greeting*labelString: Here's %LocalHost%!
```

일단 사용자명이 입력되면, 로그인 서버는 기본적으로 *username* 환영합니다! 메시지를 표시합니다. Xresources에Dtlogin*greeting*persLabelString 자원을 설정하여 이 메시지를 변경할 수 있습니다. persLabelString 값에는 % *s*%가 포함될 수 있으며 이는 **사용자명**으로 대체될 것입니다.

다음 예는 개인별 환영 메시지를 안녕하십니까 **사용자명**으로 변경합니다.

```
Dtlogin*greeting*persLabelString: 안녕하십니까 %s
```

▼ 글꼴 변경 방법

Xresources에 다음 글꼴 자원 중 하나를 설정하여 로그인 화면에 사용된 글꼴을 변경할 수 있습니다.

Dtlogin*greeting*fontList—환영 메시지의 글꼴

Dtlogin*labelFont—누름 버튼 및 레이블의 글꼴

Dtlogin*textFont—도움말 및 오류 메시지의 글꼴

사용가능한 글꼴을 나열하려면 다음을 입력하십시오.

```
xlsfonts [-options] [-fn  
pattern]
```

다음 예에서는 환영 메시지에 큰 글꼴을 사용하고 있습니다(사용자가 지정하는 값은 모두 한 행에 입력해야 합니다).

```
Dtlogin*greeting*fontList: -dt-interface \  
system-medium-r-normal-xxl*-*-**-*-*-*-*-*;
```

▼ 각 언어 표시를 위한 대체 텍스트 제공 방법

로케일 이름의 기본 표시 대신 로그인 화면 언어 메뉴에 로케일별 텍스트를 표시하려면 Xresources에서 Dtlogin**language***languageName* 자원 이름 자원을 수정하십시오.

```
Dtlogin*En_US*languageName: American
```

이제 로케일 이름 En_US 대신에 American이 표시됩니다.

로그인 화면 동작 변경

로그인 화면의 동작을 사용자 정의하려면 Xconfig 파일에서 지정된 자원을 수정합니다.

Xconfig를 수정하려면, Xconfig를 /usr/dt/config에서 /etc/dt/config로 복사하십시오. /etc/dt/config/Xconfig를 수정한 후에 다음을 입력하여 로그인 서버에 Xconfig를 다시 읽도록 알려주십시오.

```
/usr/dt/bin/dtconfig -reset
```

이렇게 하면 명령어 kill -HUPlogin server process ID가 수행됩니다.)

Xconfig 파일에 지정된 자원은 다음과 같습니다.

Dtlogin*authorize—Xaccess 파일 사양

Dtlogin*environment—X 서버 환경

Dtlogin*language—기본 언어

Dtlogin*languageList—로그인 화면 언어 메뉴의 언어 목록

Dtlogin*resources—Xresources 사양

Dtlogin*setup—Xsetup 파일 사양
 Dtlogin*startup—Xstartup 파일 사양
 Dtlogin*session—Xsession 파일 사양
 Dtlogin*failsafeClient—Xfailsafe 스크립트 사양
 Dtlogin*reset—Xreset 스크립트 사양
 Dtlogin*userPath—Xsession 및 Xfailsafe의 경로
 Dtlogin*systemPath—Xsetup, Xstartup 및 Xfailsafe의 경로
 Dtlogin*systemShell—Xsetup, Xstartup 및 Xfailsafe의 경로
 Dtlogin.timeZone—모든 스크립트에 대한 TZ

표시 장치별 로그인 화면 동작 변경

다음 예에서, Xconfig 자원을 변경하면 모든 표시장치의 로그인 화면이 변경됩니다. * (별표)와 함께 나열된 자원은 표시장치별로 지정될 수 있습니다. 이렇게 하면 특정 표시 장치에 대한 사용자 정의 로그인 화면 동작을 지정할 수 있습니다. 특정 표시장치에 대한 자원을 지정하려면 자원이 Dtlogin**displayName***resource*로 지정되어야 합니다. 예를 들어, 사용자 기준 액세스 제어가 expo:0 표시장치에 대해서는 작동되지 않게 하고 다른 표시장치에 대해서는 작동되게 하려면, 다음과 같이 지정하십시오.

```
Dtlogin*expo_0*authorize: False
```

주 - 표시장치 이름에서 :(콜론) 또는 .(마침표)와 같은 특수 문자는 _(밑줄)로 교체됩니다.

X 서버 액세스 변경

기본적으로, 로그인 서버는 각 사용자마다 X 서버 액세스 제어를 허용하며, *HomeDirectory/.Xauthority* 파일에 저장되어 보호되고 있는 허가 데이터에 근거하고 있습니다. 이 파일을 읽을 수 있는 사용자만이 X 서버에 연결하도록 허용됩니다. 일반적으로, 이 방법이 X 서버 액세스 제어가 선호하는 방법입니다.

사용자 기준 액세스 제어의 대안은 호스트 기준 액세스 제어입니다. 이 방법을 사용하면, 호스트가 X 서버에 액세스할 권한이 있는 경우, 해당 호스트의 모든 사용자는 X 서버에 연결할 수 있습니다. 호스트 기준 제어를 사용해야 하는 이유는 다음과 같습니다.

- 이전의 R2 및 R3 X 클라이언트는 사용자 기준 액세스 제어를 사용하여 X 서버에 연결할 수 없기 때문입니다.

- 보안이 되지 않은 네트워크상에서는 외부인이 네트워크의 X 클라이언트와 X 서버 사이에 전달된 허가 데이터를 가로챌 수 있습니다.

XconfigDtlogin*authorize 자원이 로그인 서버에게 사용자 기준 X 서버 액세스 제어를 사용하도록 알려줍니다. 호스트 기준 액세스 제어를 사용하려면 허가 자원 값을 False로 변경하십시오. 예를 들면 다음과 같습니다.

```
Dtlogin*authorize: False
```

▼ X 서버 환경 변경 방법

X 서버가 로그인 서버에 의해 시작될 때 X 서버에 하나 이상의 환경 변수 및 값을 제공하려면, Xconfig에서 Dtlogin*environment 자원을 사용하여 이를 지정할 수 있습니다. 예를 들어,

```
Dtlogin*environment: VAR1=foo VAR2=bar
```

는 변수 VAR1 및 VAR2를 지역 X 서버 프로세스가 사용할 수 있도록 만듭니다. 이러한 변수는 Xsession 및 Xfailsafe 스크립트로 내보낼 수도 있습니다.

▼ 기본 언어 변경 방법

사용자가 로그인 화면에서 데스크탑으로 로그인한 경우, 사용자 세션은 옵션 메뉴의 언어 하위 메뉴에서 선택된 로케에서 실행됩니다. 사용자가 언어를 선택하지 않은 경우, 로그인 서버 기본 언어가 사용됩니다. Xconfig에서 Dtlogin*language 자원을 설정하여 기본 언어 값을 제어할 수 있습니다. 예를 들면 다음과 같습니다.

```
Dtlogin*language: Ja_JP
```

시스템 설명서를 보고 시스템에 설치된 언어를 확인하십시오.

▼ 로그인 화면 언어 메뉴 내용 변경 방법

기본적으로 로그인 서버는 시스템에 설치된 모든 로케 목록이 들어있는 로그인 화면 언어 메뉴를 작성합니다. 사용자가 로그인 화면 언어 목록에서 로케를 선택하면 로그인 서버는 로그인 화면을 선택된 로케에서 다시 표시합니다. 사용자가 다음에 로그인하면, 로그인 서버는 해당 로케로 사용자에게 대한 데스크탑 세션을 시작합니다.

다음과 같이 Xconfig에서 Dtlogin*languageList 자원을 수정하여 사용자 자신의 언어 목록을 지정할 수 있습니다.

```
Dtlogin*languageList: En_US De_DE
```

로그인 서버는 이제 로그인 화면 언어 메뉴에 En_US와 De_DE만 표시합니다.

로그인 화면이 표시되기 전에 명령어 수행

X 서버가 시작된 후 로그인 화면을 표시하기 전에 로그인 서버는 `Xsetup` 스크립트를 실행합니다. `Xsetup`은 루트 권한으로 실행되며 로그인 화면이 표시되기 전에 실행해야 하는 명령어를 수행합니다.

`Xsetup`을 수정하려면, `Xsetup`을 `/usr/dt/config`에서 `/etc/dt/config`로 복사하십시오. 다음 번에 로그인 화면이 표시될 때 수정된 `Xsetup`이 실행됩니다.

사용자 세션을 시작하기 전에 명령어 수행

사용자가 사용자명과 암호를 입력하여 인증을 받은 후 사용자 세션을 시작하기 전에 로그인 서버는 `Xstartup` 스크립트를 실행합니다. `Xstartup`은 루트 권한으로 실행되며 사용자 세션이 시작되기 전에 루트로 실행해야 하는 명령어를 수행합니다.

`Xstartup`을 수정하려면, `Xstartup`을 `/usr/dt/config`에서 `/etc/dt/config`로 복사하십시오. 다음 번에 사용자가 로그인하면 수정된 `Xstartup`이 실행됩니다.

데스크탑 세션 시작

기본적으로, 로그인 서버는 `Xsession` 스크립트를 실행하여 사용자 세션을 시작합니다. `Xsession`은 사용자 권한으로 실행되며 데스크탑을 시작하는데 필요한 명령어를 수행합니다.

주 - `Xsession` 스크립트를 직접 갱신하지 마십시오.

사용자 데스크탑 세션 시작을 사용자 정의하는 방법에 관한 내용은 제 2 장을 참조하십시오.

Failsafe 세션 시작

사용자가 로그인 화면 옵션 메뉴의 세션 하위 메뉴에서 `Failsafe` 세션을 선택하면, 로그인 서버는 `Xfailsafe` 스크립트를 실행합니다. `Xfailsafe`는 사용자 권한으로 실행되며 일반적으로 터미널 윈도우 및 선택적 윈도우 관리자인 최소 윈도우 환경을 시작하는데 필요한 명령어를 수행합니다.

`Xfailsafe`를 수정하려면, `Xfailsafe`를 `/usr/dt/config`에서 `/etc/dt/config`로 복사하십시오. 다음 번에 사용자가 로그인하면 수정된 `Xfailsafe`가 실행됩니다.

사용자 세션이 끝난 후

사용자가 데스크탑 또는 failsafe 세션을 종료하면, 로그인 서버는 Xreset 스크립트를 실행합니다. Xreset은 루트 권한으로 실행되며 사용자 세션이 끝난 후에 루트로 실행해야 하는 명령어를 수행합니다.

Xreset을 수정하려면, Xreset을 /usr/dt/config에서 /etc/dt/config로 복사하십시오. 다음 번에 사용자가 로그인하면 수정된 Xreset이 실행됩니다.

로그인 서버 환경

로그인 서버는 Xsetup, Xstartup, Xsession, Xfailsafe 및 Xreset 스크립트로 내보낼 환경을 제공합니다. 이 환경은 표 1-1에 설명되어 있습니다. 추가 변수도 로그인 서버가 내보낼 수 있습니다.

표 1-1 로그인 서버 환경

환경 변수	Xsetup	Xstartup	Xsession	Xreset	설명
LANG	X	X	X	X	기본 또는 선택된 언어
XAUTHORITY	X	X	X	X	대체 X 권한 파일(선택사항)
PATH	X	X	X	X	Dtlogin*userPath 자원 (Xsession, Xfailsafe) 또는 Dtlogin*systemPath 자원 (Xsetup, Xstartup, Xreset)의 값
DISPLAY	X	X	X	X	X 서버 연결 번호
SHELL	X	X	X	X	/etc/passwd (Xsession, Xfailsafe) 또는 Dtlogin*systemShell 자원 (Xsetup, Xstartup, Xreset)에 지정된 셸
TZ	X	X	X	X	Dtlogin.timeZone 자원의 값 또는 시스템에서 결정한 시간대
USER		X	X	X	사용자 이름
HOME		X	X	X	/etc/passwd에 지정된 홈 디렉토리
LOGNAME		X	X	X	사용자 이름

사용자 또는 시스템 경로 변경

로그인 서버는 Xsession과 Xfailsafe 스크립트를 실행할 때 PATH 환경 변수를 설정합니다. 사용자는 이 스트립트에 대체 경로를 제공할 수 있습니다.

▼ 사용자 경로 변경 방법

- Xconfig에서 Dtlogin*userPath 자원을 설정하십시오. 예를 들면 다음과 같습니다.

```
Dtlogin*userPath: /usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11
```

▼ 시스템 경로 변경 방법

- Xconfig에서 Dtlogin*systemPath 자원을 설정하십시오. 예를 들면 다음과 같습니다.

```
Dtlogin*systemPath: /usr/bin/X11:/etc:/bin:/usr/bin:/usr/ucb
```

▼ 사용자 또는 시스템 셸 변경 방법

로그인 서버는 Xsetup, Xstartup 및 Xfailsafe 스크립트를 실행할 때 SHELL 환경 변수를 설정합니다. 기본값은 /bin/sh입니다. 이 스크립트에 대체 셸을 제공하려면 Xconfig에서 Dtlogin*systemShell 자원을 설정합니다. 예를 들면 다음과 같습니다.

```
Dtlogin*systemShell: /bin/ksh
```

▼ 시간대 변경 방법

로그인 서버는 Xsetup, Xstartup, Xsession, Xfailsafe 및 Xreset 스크립트를 실행할 때 TZ 환경 변수를 설정합니다. 기본값은 시스템에서 얻게 되므로 일반적으로 이 동작을 변경할 필요가 없습니다. 이 스크립트에 대체 시간대를 제공하려면 Xconfig에서 Dtlogin.timeZone 자원을 설정하십시오. 예를 들면 다음과 같습니다.

```
Dtlogin.timeZone: CST6CDT
```

로그인 관리자 관리

로그인 서버가 시작되면, 하나의 dtlogin 프로세스가 시작됩니다. dtlogin 프로세스는 Xconfig 파일을 읽어 초기 로그인 서버 구성을 결정하고 다른 로그인 서버 환경설정 파일의 위치를 찾습니다. 그런 다음 로그인 서버는 Xservers 파일을 읽어 명시적으로 관리해야 할 표시장치가 있는지를 확인하고, 또한 Xaccess 파일을 읽어 로그인 서버로의 액세스를 제어합니다.

로그인 서버가 Xservers 파일에서 지역 표시장치를 관리할 필요가 있다는 것을 발견하면, Xservers 파일에 지시된 대로 X 서버를 시작한 다음에 해당 표시장치에 로그인 화면을 표시합니다.

로그인 서버가 네트워크 표시장치를 관리할 필요가 있다는 것을 Xservers 파일에서 발견하면, X 서버가 이미 지정된 표시장치 이름으로 실행된다고 가정하고 그 표시장치에 로그인 화면을 표시합니다.

그런 다음 로그인 서버는 네트워크로부터 XDMCP 요청을 기다립니다.

관리되는 각 표시장치에 대해 로그인 서버는 먼저 해당 표시장치에 대해 새로운 dtlogin 프로세스를 작성합니다. 이는 로그인 서버가 n 개의 표시장치를 관리하고 있는 경우, $n+1$ 개의 dtlogin 프로세스가 생기게 된다는 것을 의미합니다. 로그인 서버는 Xsetup 스크립트를 실행하고, Xresources 파일을 로드한 후, 로그인 화면을 표시하기 위해 dtgreet를 실행합니다. 일단 사용자가 사용자명과 암호를 입력하고 승인을 받으면, 로그인 서버는 Xstartup 스크립트를 실행한 후, Xsession 또는 Xfailsafe 스크립트를 실행합니다. 사용자가 세션을 종료하면, 로그인 서버는 Xreset 스크립트를 실행합니다.

로그인 서버가 XDMCP 간접 요청을 받으면, dtchooser를 실행하여 표시장치에 로그인 서버 호스트 목록을 표시합니다. 사용자가 목록에서 호스트를 선택하면 해당 호스트의 로그인 서버가 표시장치를 관리합니다.

Xaccess, Xconfig, Xfailsafe, Xreset, *language*/Xresources, Xservers, Xsetup 및 Xstartup 환경설정 파일의 경우, 로그인 서버는 기본적으로 먼저 /etc/dt/config에서 찾고, 그 다음에 /usr/dt/config에서 찾아 가장 먼저 발견된 파일을 사용합니다.

로그인 관리자 파일

로그인 관리자 파일의 기본 위치는 다음과 같습니다.

/usr/dt/bin/dtlogin—로그인 서버와 표시장치 관리자

/usr/dt/bin/dtgreet—표시장치에 대한 로그인 화면을 표시합니다.

`/usr/dt/bin/dtchooser`—표시장치에 대한 선택기 화면을 표시합니다.
`/usr/dt/bin/Xsession`—데스크탑 세션을 시작합니다.
`/usr/dt/config/Xfailsafe`—`failsafe` 세션을 시작합니다.
`/usr/dt/config/Xconfig`—로그인 서버 환경설정 파일
`/usr/dt/config/Xservers`—로그인 서버 표시장치 설명 파일
`/usr/dt/config/Xaccess`—로그인 서버 액세스 설명 파일
`/usr/dt/config/language/Xresources`—표시장치 레이아웃 자원
`/usr/dt/config/Xsetup`—표시장치 설치 파일
`/usr/dt/config/Xstartup`—세션 전 시작 파일
`/usr/dt/config/Xreset`—세션 후 재설정 파일
`/var/dt/Xpid`—로그인 서버의 프로세스 ID
`/var/dt/Xerrors`—로그인 서버의 오류 로그 파일

세션 관리자 구성

세션 관리자는 데스크탑을 시작하고 실행 중인 응용 프로그램, 색상, 글꼴, 마우스 동작, 오디오 볼륨 및 키보드 누름을 자동으로 저장하고 복원하는 일을 담당합니다.

- “세션이란?” 39 페이지
- “세션 시작” 40 페이지
- “세션 관리자가 시작될 때” 41 페이지
- “추가 세션 시작 사용자 정의” 47 페이지
- “세션 관리자 파일 및 디렉토리” 51 페이지

세션 관리자를 사용하여 수행할 수 있는 작업은 다음과 같습니다.

- 모든 데스크탑 사용자를 위한 초기 세션의 사용자 정의
- 모든 데스크탑 사용자를 위한 환경 및 자원의 사용자 정의
- 세션 시작 메시지의 변경
- 세션 시작 도구 및 데몬의 매개변수 변경
- 모든 사용자에게 대한 데스크탑 색상 사용의 사용자 정의

세션이란?

세션은 사용자의 데스크탑에 있는 응용 프로그램, 설정값 및 자원의 모음입니다. 세션 관리자는 세션 관리자가 사용자 세션을 저장하고 복구할 수 있게 하는 규약과 프로토콜의 집합입니다. 사용자는 시스템에 로그인할 수 있으며, 사용자가 로그오프했을 때 나타났던 것과 같이 실행 중인 응용 프로그램, 설정값 및 자원이 표시될 수 있습니다. 사용자가 맨 처음 데스크탑에 로그인할 때는 기본 초기 세션이 로드됩니다. 그 후에는 세션 관리자가 현재 세션 및 홈 세션의 개념을 지원합니다.

초기 세션

사용자가 맨 처음 데스크탑에 로그인하면, 세션 관리자는 시스템 기본값을 사용하여 사용자의 초기 세션을 작성합니다. 기본적으로 파일 관리자와 데스크탑 소개, 도움말 볼륨이 시작됩니다.

현재 세션

사용자의 실행 세션은 저장된 홈 세션이나, 저장된 현재 세션 또는 시스템 기본 초기 세션 중 어느 곳으로부터 로그인해서 복구되었든 항상 현재 세션으로 간주됩니다. 사용자의 스타일 관리자 시작 설정값을 기준으로 사용자가 세션을 종료할 때 세션 관리자는 자동으로 현재 세션을 저장합니다. 다음 번에 사용자가 데스크탑에 로그인하면 세션 관리자는 이전에 저장된 현재 세션을 다시 시작하는데, 이는 데스크탑이 사용자가 마지막 로그아웃했을 때와 같은 상태로 복구됨을 의미합니다.

홈 세션

사용자가 로그아웃했을 때의 상태와 무관하게 사용자는 로그인할 때마다 같은 상태로 데스크탑을 복원할 수 있습니다. 사용자는 현재 세션의 상태를 저장할 수 있으며 스타일 관리자 시작 설정값을 사용하여 사용자가 로그인할 때마다 세션 관리자가 그 세션을 시작하도록 할 수 있습니다.

특정 표시장치 세션

특정 표시장치를 위한 특정 세션을 실행하기 위해, 사용자는 특정 표시장치 세션을 작성할 수 있습니다. 그러기 위해서 사용자는 `HomeDirectory/.dt/sessions` 디렉토리를 `HomeDirectory/.dt/display`로 복사할 수 있는데, 여기서 `display`는 수식되지 않은 실제 호스트 이름입니다(예를 들어, `pablo:0`는 유효하며, `pablo.gato.com:0` 또는 `unix:0`는 유효하지 않습니다). 사용자가 표시장치 `pablo:0`에 로그인하면, 세션 관리자는 해당 특정 표시장치 세션을 시작합니다.

세션 시작

세션 관리자는 `/usr/dt/bin/Xsession`을 통해 시작됩니다. 사용자가 로그인 관리자를 이용하여 로그인하면, `Xsession`이 기본적으로 시작됩니다.

선택적으로, 사용자는 종전의 문자 모드(`getty`) 로그인을 사용하여 로그인할 수 있으며 X 서버를 시작하는 `xinit`와 같은 툴을 사용하여 수동으로 세션 관리자를 시작할 수 있습니다. 예를 들면 다음과 같습니다. `xinit /usr/dt/bin/Xsession`.

세션 관리자가 시작될 때

세션 관리자가 시작될 때 다음 단계에 따라 사용자 세션을 시작합니다.

1. `HomeDirectory/.dtprofile` 스크립트를 소스로 사용합니다.
2. `Xsession.d` 스크립트를 소스로 사용합니다.
3. 환영 메시지를 표시합니다.
4. 데스크탑 검색 경로를 설정합니다.
5. 사용 가능한 응용 프로그램을 수집합니다.
6. 선택적으로 `HomeDirectory/.profile` 또는 `HomeDirectory/.login`을 소스로 사용합니다.
7. ToolTalk[®]을 시작합니다. 메시지 표시 데몬
8. 세션 자원을 로드합니다.
9. 색상 서버를 시작합니다.
10. 작업 공간 관리자를 시작합니다.
11. 세션 응용 프로그램을 시작합니다.

다음 절부터는 위에 나열된 단계들을 설명합니다.

.dtprofile 스크립트를 소스로 사용하기

세션 시작시, `Xsession` 스크립트는 사용자의 `HomeDirectory/.dtprofile` 스크립트를 소스로 사용합니다. `HomeDirectory/.dtprofile` 스크립트는 `/bin/sh` 또는 `/bin/ksh` 스크립트로서, 사용자 자신의 세션에 대한 환경 변수를 설정할 수 있게 합니다. 환경 변수 설정에 대한 자세한 내용은 “추가 세션 시작 사용자 정의” 47 페이지를 참조하십시오.

`HomeDirectory/.dtprofile` 스크립트가 없을 경우, 예를 들어 사용자가 데스크탑에 처음 로그인한 경우, `Xsession`은 데스크탑 기본값 `sys.dtprofile`을 `HomeDirectory/.dtprofile`로 복사합니다.

데스크탑 기본값은 `/usr/dt/config/sys.dtprofile`입니다. `sys.dtprofile` 스크립트를 사용자 정의하려면, `sys.dtprofile`을 `/usr/dt/config`에서 `/etc/dt/config`로 복사하고 새 파일을 편집하십시오.

Xsession.d 스크립트를 소스로 사용하기

HomeDirectory/.dtprofile 스크립트를 소스로 사용한 후, *Xsession* 스크립트는 *Xsession.d* 스크립트를 소스로 사용합니다. 이 스크립트들은 추가 환경 변수를 설정하고, 사용자 세션에 대한 선택적 데몬을 시작하는데 사용됩니다. 기본 *Xsession.d* 스크립트는 다음과 같습니다.

0010.dtpaths—사용자 정의할 수 있는 데스크탑 검색 경로를 제공합니다.

0020.dtimms—선택적 입력 방법 서버를 시작합니다.

0030.dttmpdir—사용자별 세션별 임시 디렉토리를 작성합니다.

0040.xmbind—*\$XMBINDDIR*를 데스크탑 기본값으로 설정합니다.

이 외에도 *Xsession.d*에는 특정 구입업체의 스크립트가 있을 수 있습니다.

*Xsession*는 먼저 */etc/dt/config/Xsession.d* 디렉토리의 모든 파일을 소스로 사용한 후 */usr/dt/config/Xsession.d* 디렉토리의 파일을 소스로 사용합니다.

데스크탑 기본값 *Xsession.d* 스크립트는 */usr/dt/config/Xsession.d* 디렉토리에 있습니다. *Xsession.d* 스크립트를 사용자 정의하려면, 스크립트를 */usr/dt/config/Xsession.d*에서 */etc/dt/config/Xsession.d*로 복사하고 새 파일을 편집하십시오. 이 작업을 수행하려면 실행 권한이 있어야 합니다.

또한 *Xsession*이 자동으로 사용자 자신의 스크립트를 소스로 사용하도록 하려면, 이 스크립트를 */etc/dt/config/Xsession.d*에 복사하십시오.

주 - *Xsession.d* 스크립트를 수정하거나 작성하려면 사용자가 발행한 포그라운드 명령어의 지속 시간이 짧아야 합니다. 이는 명령어에 소요되는 시간이 세션 시작 시간에 직접적으로 영향을 주기 때문입니다. 포그라운드 명령어를 종료하지 않은 경우는 세션 시작이 멈출 것입니다. *Xsession.d* 스크립트에서 실행되는 명령어 중 세션이 지속되는 동안 계속 실행되기를 원하는 명령어는 백그라운드에서 실행되어야 합니다.

환영 메시지 표시

*HomeDirectory/.dtprofile*와 *Xsession.d* 스크립트를 소스로 사용한 후, *Xsession*은 화면 전체에 환영 메시지를 표시합니다. 표시되는 환영 메시지를 사용자 정의하거나, 메시지 전체를 표시되지 않도록 할 수 있습니다. *dthello* 클라이언트는 메시지를 표시하는데 사용됩니다.

메시지 텍스트를 바꾸려면 *dtstart_hello[0]* 변수를 수정하여 *dthello* 옵션을 변경하십시오.

dtstart_hello[0]를 변경하려면, 새 값을 설정하는 /etc/dt/config/Xsession.d 스크립트를 작성하십시오. 모든 사용자를 위해 그 날의 메시지를 표시하려면, 실행 가능한 sh 또는 ksh 스크립트 (예:/etc/dt/config/Xsession.d/myvars)를 작성하고, dtstart_hello[0]를 다음과 같이 설정하십시오.

```
dtstart_hello[0]="/usr/dt/bin/dthello -file /etc/motd &"
```

마찬가지로 *HomeDirectory/.dtprofile*에서 dtstart_hello[0]를 설정하여 세션의 환영 메시지를 변경할 수 있습니다.

환영 메시지가 표시되지 않도록 하려면 dtstart_hello[0]=" "으로 설정하십시오.

dthello에 대한 자세한 내용은 dthello 매뉴얼 페이지를 참조하십시오.

데스크탑 검색 경로 설정

데스크탑 검색 경로는 로그인시 dtsearchpath에 의해 작성됩니다. dtsearchpath에 의해 사용되는 환경 변수에는 두 가지 종류가 있습니다.

입력 변수—시스템 관리자 또는 최종 사용자에게 의해 값이 설정되는 시스템 전체 변수 및 개인 환경 변수

출력 변수—dtsearchpath에 의해 작성되고 값을 할당받는 변수 각 변수의 값은 데스크탑 세션에 대한 검색 경로입니다.

dtsearchpath의 명령행 옵션을 변경하려면, dtstart_searchpath 변수를 수정하십시오. 모든 사용자에게 대해 dtstart_searchpath 변수를 변경하려면, 실행 가능한 sh 또는 ksh 스크립트(예:/etc/dt/config/Xsession.d/myvars)를 작성하고 dtstart_searchpath를 다음과 같이 설정하십시오.

```
dtstart_searchpath="/usr/dt/bin/dtsearchpath"
```

마찬가지로 사용자는 *HomeDirectory/.dtprofile*에서 dtstart_searchpath를 설정하여 자신의 세션에 대해서만 dtsearchpath 옵션을 변경할 수 있습니다.

dtsearchpath에 대한 자세한 내용은 제 9 장 “데스크탑 검색 경로를 참조하십시오. dtsearchpath 옵션에 대한 자세한 내용은 dtsearchpath 매뉴얼 페이지를 참조하십시오.

사용 가능한 응용 프로그램 모으기

데스크탑 검색 경로를 설정한 후의 다음 단계는 dtappgather를 사용하여 사용 가능한 응용 프로그램을 모으는 것입니다. dtappgather의 명령행 옵션을 변경하려면, dtstart_appgather 변수를 수정하십시오. 모든 사용자에게 대해 dtstart_appgather 변수를 변경하려면, 실행 가능한 sh 또는 ksh 스크립트 (예:/etc/dt/config/Xsession.d/myvars)를 작성하고 dtstart_appgather를 다음과 같이 설정하십시오.

```
dtstart_apppgather="/usr/dt/bin/dtappgather &"
```

마찬가지로 사용자는 *HomeDirectory/.dtprofile*에서 `dtstart_apppgather`를 설정하여 자신의 세션에 대해서만 `dtappgather` 옵션을 변경할 수 있습니다.

`dtappgather` 옵션에 대한 자세한 내용은 `dtappgather(4)` 매뉴얼 페이지를 참조하십시오.

선택적으로 .profile 또는 .login 스크립트를 소스로 사용하기

`Xsession`은 사용자가 종전에 사용하던 *HomeDirectory/.profile* 또는 *HomeDirectory/.login* 스크립트를 소스로 사용할 수 있습니다. 기본적으로 이 기능은 사용불가능합니다. `Xsession`에 `.profile` 또는 `.login` 스크립트를 소스로 사용하도록 알려주려면, `DTSOURCEPROFILE`를 `true`로 설정하십시오.

모든 사용자에게 `DTSOURCEPROFILE`를 변경하려면, 새 값을 설정하는 `/etc/dt/config/Xsession.d` 스크립트를 작성하십시오. 모든 사용자에게 `DTSOURCEPROFILE`를 `true`로 설정하려면, `sh` 또는 `ksh` 스크립트 (`/etc/dt/config/Xsession.d/myvars`)를 작성하고 다음과 같이 `DTSOURCEPROFILE`를 설정하십시오.

```
DTSOURCEPROFILE=true
```

마찬가지로 사용자는 *HomeDirectory/.dtprofile*에서 `DTSOURCEPROFILE`를 `true`로 설정하여 자신의 세션만을 위한 `DTSOURCEPROFILE`를 변경할 수 있습니다.

ToolTalk 메시지 표시 데몬 시작

`ToolTalk` 메시지 표시 데몬, `ttsession`은 독립적인 응용 프로그램이 상대 응용 프로그램에 대한 직접적인 정보없이도 서로 통신할 수 있게 합니다. 응용 프로그램은 서로 통신하기 위해 `ToolTalk` 메시지를 작성하여 전송합니다. `ttsession`은 메시지를 전달하기 위해 네트워크상에서 통신합니다.

`ttsession`의 명령행 옵션을 바꾸려면, `dtstart_ttsession` 변수를 수정하십시오. 모든 사용자에게 `dtstart_ttsession` 변수를 변경하려면, 실행 가능한 `sh` 또는 `ksh` 스크립트(예: `/etc/dt/config/Xsession.d/myvars`)를 작성하고 다음과 같이 `dtstart_ttsession`을 설정하십시오.

```
dtstart_ttsession="/usr/dt/bin/ttsession -s"
```

사용자는 마찬가지로 *HomeDirectory/.dtprofile*에서 `dtstart_ttsession`을 설정하여 자신의 세션에 대한 `ttsession` 옵션을 변경할 수 있습니다.

`ttsession` 옵션에 대한 자세한 내용은 `ttsession` 매뉴얼 페이지를 참조하십시오. `ttsession`에 대한 자세한 내용은 *Common Desktop Environment: ToolTalk Messaging Overview*를 참조하십시오.

세션 관리자 클라이언트 시작

이 시점에서, Xsession은 세션 시작 프로세스를 계속하는 `/usr/dt/bin/dtsession`을 시작합니다.

세션 자원 로드

세션 관리자는 X 서버 RESOURCE MANAGER 특성을 이용하여 데스크탑 자원을 모든 응용 프로그램에서 사용할 수 있게 합니다. 세션 관리자는 다음과 같은 방식을 통해 RESOURCE_MANAGER를 로드합니다.

- 시스템 기본 자원을 로드합니다.
- 시스템 관리자가 지정하는 시스템 전체 자원을 병합합니다.
- 사용자가 지정하는 자원을 병합합니다.

데스크탑 기본 자원은 `/usr/dt/config/language/sys.resources`에서 찾을 수 있습니다. RESOURCE_MANAGER 특성을 통해 각 사용자의 세션이 이러한 자원을 사용할 수 있도록 할 수 있습니다. 이 파일은 다음 데스크탑 설치 때 겹쳐 쓰게 되므로 편집하면 안 됩니다.

사용자는 `/etc/dt/config/language/sys.resources`를 작성하여 시스템 기본 자원을 늘릴 수 있습니다. 이 파일에서는 모든 데스크탑 사용자를 위해 기본 자원을 대체하거나 추가 자원을 지정할 수 있습니다. 이 파일은 시작시 데스크탑 기본 자원으로 병합되므로, 갱신되거나 새로운 자원 사양만이 이 파일에 포함될 수 있습니다. RESOURCE_MANAGER 특성을 통해 이 파일에 지정된 자원을 각 사용자 세션에서 사용 가능하게 할 수 있습니다. 이 파일에 지정된 자원은 데스크탑 기본 자원 파일에 지정된 자원보다 우선합니다.

사용자는 자신의 `HomeDirectory/.Xdefaults` 파일을 사용하여 데스크탑 기본 자원 및 시스템 전체 자원을 늘릴 수 있습니다. RESOURCE_MANAGER 특성을 통해 이 파일에서 지정된 자원을 해당 사용자 세션에서 사용 가능하게 할 수 있습니다. 이 파일에 지정된 자원은 데스크탑 기본 또는 시스템 관리자 자원 파일에 지정된 자원보다 우선합니다.

주 - X Toolkit Intrinsics 유틸리티는 응용 프로그램에 대한 자원을 RESOURCE_MANAGER 또는 `HomeDirectory/.Xdefaults`로부터 (양쪽 모두에서는 안됨) 로드하도록 지정합니다. 일반적으로 이는 사용자의 `HomeDirectory/.Xdefaults` 파일이 무시됨을 의미합니다. 그러나, 세션 관리자는 위에서 설명한 것처럼 세션 시작시 `HomeDirectory/.Xdefaults`를 RESOURCE_MANAGER에 병합하여 수용합니다. 사용자가 `HomeDirectory/.Xdefaults`를 변경하면, 사용자가 자원 재로드 활동을 호출할 때까지 새 응용 프로그램에서 변경 내용을 볼 수 없습니다. 자원 재로드 활동은 세션 관리자에게 기본, 시스템 전체 및 특정 사용자 자원과 함께 RESOURCE_MANAGER를 재로드할 것을 지시합니다. 이렇게 하면 시스템 전체 및 개인 자원 파일에 대한 변경 내용을 응용 프로그램이 사용할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- “응용 프로그램 자원 설정” 273 페이지.
- dtresourcesfile(4) 매뉴얼 페이지를 참조하십시오.

색상 서버 시작

세션 관리자는 데스크탑을 위한 색상 서버 역할을 하며 이를 구성하는데 사용되는 다음과 같은 dtsession 자원 집합을 제공합니다.

`foregroundColor`—픽셀이 전경 색상에 할당되는지 여부를 제어합니다.

`dynamicColor`—읽기 전용 색상이 할당되는지 여부를 지정합니다.

`shadowPixmap`—상단 음영이나 하단 음영에 색상이 할당되는지 여부를 지정합니다.

`colorUse`—색상 할당을 제한합니다.

`writeXrdbColors`—*background와 *foreground 자원이 자원 데이터베이스에 있는지 여부를 지정합니다.

`/etc/dt/config/language/sys.resources`를 작성하고 색상 서버를 이 파일에 지정함으로써 모든 사용자에게 대해 색상 서버 자원을 설정할 수 있습니다.

마찬가지로 사용자는 색상 서버 자원을 `HomeDirectory/.xdefaults`에 지정하여 자신의 세션에 대해 색상 서버를 설정할 수 있습니다.

색상 서버 자원 설정에 대한 자세한 내용은 “색상 관리” 283 페이지를 참조하십시오.

작업 공간 관리자 시작

세션 관리자는 작업 공간 관리자를 시작할 책임이 있습니다. 기본적으로 `/usr/dt/bin/dtwm`이 시작됩니다. `wmStartupCommand` 자원과 함께 대체 윈도우 관리자가 지정될 수 있습니다.

`/etc/dt/config/language/sys.resources`를 작성하고 윈도우 관리자에 대한 전체 경로 이름과 옵션을 이 파일에 `Dtsession*wmStartupCommand` 자원과 함께 지정함으로써 모든 사용자에게 대해 대체 윈도우 관리자를 지정할 수 있습니다.

마찬가지로 사용자는 `HomeDirectory/.xdefaults`에 `Dtsession*wmStartupCommand` 자원을 지정함으로써 자신의 세션에 대해 대체 윈도우 관리자를 지정할 수 있습니다.

윈도우 관리자에 대한 자세한 내용은 제 16 장을 참조하십시오.

세션 응용 프로그램 시작

세션 시작시, 세션 관리자는 세션의 일부로 저장된 모든 응용 프로그램을 다시 시작합니다. 사용자의 초기 세션의 일부로 복원되어야 하는 응용 프로그램의 시스템 기본 집합은 `/usr/dt/config/language/sys.session`에서 찾을 수 있습니다. 이 파일은 다음 데스크탑 설치시 겹쳐 쓰여지므로 편집하면 안됩니다.

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

시스템 관리자는 `/usr/dt/config/language/sys.session`을 `/etc/dt/config/language/sys.session`에 복사하고 이 파일을 수정함으로써 사용자 초기 세션의 일부로 시작된 응용 프로그램 집합을 교체할 수 있습니다. 자원 파일과는 달리 이 파일은 데스크탑 기본 파일로 완전히 교체하는 파일로 사용되므로, 사용자는 시스템 기본 파일의 사본을 만들어 필요한 수정을 하면 됩니다.

추가 세션 시작 사용자 정의

이 절에서 다루는 내용은 다음과 같습니다.

- 환경 변수 설정
- 자원 설정
- 특정 표시장치 세션 사용
- 로그인시 스크립트 실행
- 백업 세션 복구

▼ 환경 변수 설정 방법

- 시스템 전체 환경 변수를 설정하려면, 변수를 설정하고 내보내는 `/etc/dt/config/Xsession.d` 디렉토리에서 파일을 작성하십시오.

예를 들어, 사용자가 다음을 포함하는 실행 가능한 ksh 스크립트, `/etc/dt/config/Xsession.d/myvars`를 작성하는 경우,

```
export MYVARIABLE="value"
```

`MYVARIABLE`가 다음 로그인시에 각 사용자의 환경에 설정될 것입니다.

- 개인 환경 변수를 설정하려면, `HomeDirectory` `/.dtprofile`에서 변수를 설정하십시오.

예를 들어,

```
export MYVARIABLE="value"
```

변수 `MYVARIABLE`을 다음 로그인시 각 사용자 환경에 설정합니다.

주 - 세션 관리자는 .profile 또는 .login 파일을 자동으로 읽지 않습니다. 그러나 이러한 파일을 사용하기 위해 구성될 수도 있습니다. "선택적으로 .profile 또는 .login 스크립트를 소스로 사용하기" 44 페이지를 참조하십시오.

▼ 자원 설정 방법

- 시스템 전체 자원을 설정하려면, /etc/dt/config/language/sys.resources 파일에 자원을 추가하십시오(해당 파일을 작성해야 하는 경우도 있습니다).

주 - .dtprofile은 /bin/sh 또는 /bin/ksh 구문만을 지원합니다.

예를 들어, /etc/dt/config/C/sys.resources에서 다음과 같이 지정하는 경우,

```
AnApplication*resource: value
```

다음 번 로그인시 자원 AnApplication*resource이 각 사용자의 RESOURCE_MANAGER 특성에 설정됩니다.

- 개인 자원을 설정하려면, 파일 HomeDirectory/.Xdefaults에 자원을 추가하십시오.

▼ 특정 표시장치 자원 설정 방법

모든 사용자를 위해 특정 표시장치 자원을 시스템에서 설정할 수 있습니다. 또한 사용자는 자신의 세션에만 한정된 특정 표시장치 자원도 설정할 수 있습니다. 이를 통해 사용자가 데스크탑에 로그인할 때 사용하는 표시장치에 따라 자원을 지정할 수가 있습니다.

- 시스템의 모든 데스크탑 사용자를 위해 특정 표시장치 자원을 설정하려면, 특정 표시장치 자원을 지정하는 파일 /etc/dt/config/language/sys.resources를 작성하십시오.
- 개인의 특정 표시장치 자원을 설정하려면, HomeDirectory/.Xdefaults에 이 자원을 지정하십시오.

이 자원들을 cpp 조건문으로 묶어 구분합니다. DISPLAY_displayname 매크로가 \$DISPLAY 변수 값에 따라 정의됩니다. 이 작업은 다음과 같은 방식으로 수행됩니다. 모든 .(마침표) 및 :(콜론) 문자를 _(밑줄)로 바꾸고, 모든 화면 사양을 제거한 후, 마지막으로 결과 앞에 DISPLAY_를 붙입니다.

예를 들어, :0의 \$DISPLAY는 DISPLAY_0이 되며,blanco.gato.com:0.0의 \$DISPLAY는 DISPLAY_blanco_gato_com_0이 됩니다. 결과값은 세션 자원 파일에서 cpp 테스트의 일부로 사용될 수 있습니다. 예를 들어, /etc/dt/config/C/sys.resources에서 다음과 같이 지정하는 경우,

```
Myapp*resource: value
```

```

#ifdef DISPLAY_blanco_gato_com_0
Myapp*resource: specialvalue1
#endif

#ifdef DISPLAY_pablo_gato_com_0
Myapp*resource: specialvalue2
#endif

```

자원 MyApp*resource이 RESOURCE MANAGER에서 사용자가 표시장치 blanco.gato.com:0에서 로그인하는 경우에는 specialvalue1로, 사용자가 pablo.gato.com:0에서 로그인하는 경우에는 specialvalue2로, 사용자가 다른 표시장치에서 로그인하는 경우에는 값으로 설정됩니다.

▼ 초기 세션에 대한 응용 프로그램 변경 방법

사용자는 사용자 초기 세션의 일부로 시작되는 대체 응용 프로그램을 지정할 수 있습니다.

1. /usr/dt/config/language/sys.session을 /etc/dt/config/language/sys.session으로 복사하십시오.
2. 새 sys.session 파일을 수정하십시오.
sys.session에 있는 각 항목은 다음과 같이 나타납니다.

```
dtsmcmd -cmd command_and_options
```

사용자 초기 세션의 일부로 추가 응용 프로그램을 시작하려면, 새 sys.session 항목을 전체 경로명으로 지정하십시오. 예를 들어, /usr/bin/X11/xclock을 사용자 초기 세션의 일부로 시작하려면 xclock 항목을 /etc/dt/config/C/sys.session에 추가하십시오.

```

#
# Start up xclock...
#
dtsmcmd -cmd "/usr/bin/X11/xclock -digital"

```

▼ 특정 표시장치 세션 설정 방법

사용자는 세션을 특정 표시장치에 맞도록 조정하기 위해 특정 표시장치 세션을 설정할 수 있습니다.

- HomeDirectory/.dt/sessions 디렉토리를 HomeDirectory/.dt/display로 복사하십시오. 여기서 display는 수식되지 않은 실제 호스트명입니다(pablo:0은 유효하며, pablo.gato.com:0 또는 unix:0은 유효하지 않습니다).

예를 들어, 표시장치 pablo.gato.com:0에 대한 특정 표시장치 세션을 작성하려면

```
cp -r HomeDirectory/.dt/sessions HomeDirectory/.dt/pablo:0
```

다음 번에 사용자가 표시장치 pablo.gato.com:0에서 로그인하면 세션 관리자가 해당 특정 표시장치 세션을 시작합니다.

세션 시작 및 로그아웃시 추가 명령어 실행

사용자는 자신의 데스크탑 세션에 로그인할 때 추가 명령어가 시작되도록 지정할 수 있습니다. 이는 세션 관리자에 의해 저장되지 않는 X 설정값을 설정하는데 유용합니다. 예를 들어, 사용자는 `xsetroot`를 사용하여 루트(작업 공간) 포인터를 사용자 정의할 수 있습니다. 다른 용도로는 세션 관리자에 의해 저장되거나 복구될 수 없는 응용 프로그램을 시작할 수도 있습니다. 응용 프로그램이 세션이 복원될 때 다시 시작되지 않으면, 사용자는 이 방법을 사용하여 클라이언트를 시작할 수 있습니다.

▼ 세션 시작시 추가 명령어 실행 방법

- 명령어를 포함하는 파일 `HomeDirectory/.dt/sessions/sessionetc`을 작성하십시오.

일반적으로 이 파일은 스크립트이며 실행 권한이 있어야 합니다. `sessionetc`에서 시작된 프로세스는 백그라운드에서 실행되어야 합니다.

주 - 세션 관리자에 의해 자동으로 복구된 클라이언트를 시작하려면 `sessionetc`를 사용하지 마십시오. 그렇게 하면 응용 프로그램의 여러 사본이 시작됩니다. 윈도우가 다른 윈도우의 위에 쌓일 수 있으므로 사용자는 사본을 즉시 보지 못할 수도 있습니다.

▼ 로그아웃시 추가 명령어 실행 방법

`sessionetc`의 동반 파일은 `sessionexit`입니다. `sessionexit`를 사용하여 세션 종료시 세션 관리자에 의해 처리되지 않는 작업을 수행하십시오.

- 명령어를 포함하는 파일 `HomeDirectory/.dt/sessions/sessionexit`를 작성하십시오.

`sessionetc`처럼, 이 파일은 일반적으로 실행 권한이 있는 스크립트입니다.

▼ 백업으로부터 세션 복구 방법

세션 관리자가 세션을 저장할 때, 세션 정보는 `HomeDirectory/.dt/sessions` 디렉토리에 저장되고, 특정 표시장치 세션을 사용할 경우에는 `HomeDirectory/.dt/display` 디렉토리에 저장됩니다. 이 디렉토리에서 세션 관리자는 이름이 `current` 또는 `home`인 하위 디렉토리를 작성하여 각각의 현재 세션 또는 홈 세션을 위한 정보를 저장합니다. 세션 정보가 저장되기 전에 세션 관리자는 이전 세션의 백업을 세션 이름으로 만들고 이를 `current.old` 또는 `home.old`에 저장합니다.

1. 로그인 화면에서 명령행 로그인 또는 비상 안전 세션을 사용하여 로그인하십시오.

2. 백업 세션 디렉토리를 사용중인 이름으로 복사하십시오. 예를 들어, 백업 홈 세션을 복구하려면

```
cp -r HomeDirectory/.dt/sessions/home.old HomeDirectory/.dt/sessions/home
```

특정 표시장치 세션을 동일한 방식으로 복구할 수 있습니다.

▼ 세션 시작 문제 조사 방법

- 파일 *HomeDirectory/.dt/startlog*를 점검하십시오.

세션 관리자는 각 사용자의 세션 시작 진행 과정을 이 파일에 기록합니다.

세션 관리자 파일 및 디렉토리

- */usr/dt/bin/Xsession*
- */usr/dt/config/Xsession.d/**
- */usr/dt/bin/dtsession*
- */usr/dt/bin/dtsession_res*
- *HomeDirectory/.dt/sessions/current*
- *HomeDirectory/.dt/sessions/home*
- *HomeDirectory/.dt/display/current*
- *HomeDirectory/.dt/display/home*

문제 해결 로그인 및 세션 시작 문제

이 장에서는 Solaris CDE 시작 파일, Solaris CDE 시작시 발생할 수 있는 문제를 설명하고 시작 문제의 해결책을 제시합니다.

- “로그인 시작 파일” 53 페이지
- “오류 로그 위치” 54 페이지
- “사용자 시작 파일” 54 페이지
- “Solaris CDE 시작 예” 55 페이지

로그인 시작 파일

Solaris CDE 로그인 관리자가 사용자를 인증하면, 데스크탑을 시작하기 위해 다음 스크립트를 호출합니다.

```
/usr/dt/bin/Xsession
```

Xsession이 호출하는 최초의 특정 사용자 파일은 *HomeDirectory/.dtprofile*입니다.

새로운 사용자가 맨 처음 Solaris CDE에 로그인할 때, .dtprofile 파일이 사용자의 홈 디렉토리로 복사됩니다. 기본적으로 이 파일은 아무것도 수행하지 않습니다. 그러나 이 파일에는 이 파일을 편집할 수 있는 방법에 대한 주석이 많이 포함되어 있습니다. 사용자는 이 파일을 편집하여 특정 사용자 환경 변수를 추가할 수 있습니다.

주 - .dtprofile 을 호출하는 Xsession 스크립트처럼, 이 파일은 ksh 구문을 사용합니다.

유용한 편집 방법의 하나는 .dtprofile 파일에서 마지막 행의 주석 표시를 다음과 같이 없애는 것입니다.

DTSOURCEPROFILE=true

이 행은 사용자의 *HomeDirectory/.login*(*csch* 사용자의 경우) 또는 *HomeDirectory/.profile*(다른 셸 사용자의 경우)이 시작 프로세스의 일부로서 사용되도록 합니다.

오류 로그 위치

Solaris CDE 로그인 화면의 옵션 메뉴에서 비상 안전 세션을 선택하여, 오류 로그를 보고 가능한 사용자 도트 파일 문제를 수정하려면 일반 Xsession 시작을 건너 뛰십시오. 표 3-1은 오류 로그와 그 위치를 나타냅니다.

표 3-1 오류 로그 위치

위치	오류 로그
<i>/var/dt/Xerrors</i>	사용자 로그인 이전의 Solaris CDE 로그인 윈도우 시스템 오류.
<i>HomeDirectory/.dt/startlog</i>	Xsession, <i>.dtprofile</i> , <i>.login</i> 또는 <i>.profile</i> 동안의 Solaris CDE 시작 오류
<i>HomeDirectory/.dt/errorlog</i>	Xsession 시작 후의 Solaris CDE 오류
<i>HomeDirectory/.dt/sessionlog</i>	세션 관리자와 윈도우 관리자 오류를 위한 세션 로그 디렉토리

어떤 오류는 시스템 콘솔 윈도우에도 표시됩니다. 시스템 콘솔 윈도우가 작동하지 않는 경우, 폴백 콘솔 로그 파일 이름은 시간/표시장치 코드가 붙은 *wscn*이 됩니다. 예를 들면 다음과 같습니다.

```
/usr/tmp/wscnAAAa004EE:0.0
```

사용자 시작 파일

.login(*csch* 사용자의 경우) 또는 *.profile*(*sh* 또는 *ksh* 사용자의 경우) 파일을 소스로 사용하려면, 다음과 같이 읽을 *.dtprofile* 파일에서 마지막 행의 주석을 없애십시오.

```
DTSOURCEPROFILE=true
```

대부분의 경우 이것이 필요한 전부입니다. 그러나 어떤 경우에는 `.login` 또는 `.profile` 파일을 수정해야 하며 이는 이 파일에 Solaris CDE 로그인 관리자에서 수행할 수 없는 명령어가 포함되어 있기 때문입니다. 명령어 중 하나라도 문제가 있는 경우, 일반적으로 이는 `stty`, `tset` 또는 “wait for input” 명령어와 같은 터미널 키보드 입력 소스를 기다리는 파일과 관련이 있습니다.



주의 - `.dtprofile`이 쉘과의 충돌 문제가 있는 명령어를 갖고 있는 `.login` 또는 `.profile` 파일을 소스로 사용하도록 설정되어 있으면, 데스크탑 시작은 실패하게 됩니다. 그러면 데스크탑이 나타나지 않습니다. 그 대신 Solaris CDE 로그인 화면이 다시 표시될 것입니다. `.login` 또는 `.profile`의 시작 오류는 일반적으로 `HomeDirectory/.dt/startlog`에 기록됩니다. `failsafe` 로그인 세션 또는 명령행 로그인을 사용하여 `.login` 또는 `.profile`의 문제 명령어를 디버그하십시오.

문제점과 가능한 해결책에 대한 자세한 내용은 `.dtprofile` 파일을 검토하십시오. 일반적으로 문제 명령어는 터미널 정보 및 제어와 관련되어 있습니다.

Solaris CDE 시작 예

이 절에서는 다음과 같은 사용자 시작 파일을 편집하는 방법을 보여줍니다.

- `.login` (csh 사용자의 경우)
- `.profile` (sh 또는 ksh 사용자의 경우)
- `.Xdefaults`

Solaris CDE 시작 프로세스는 `.login` 또는 `.profile` 스크립트에서 이름이 DT인 쉘 변수를 정의하여 Solaris CDE 시작 프로세스 동안 점검이 수행되도록 합니다. 이렇게 하면 `tty`와 `stty` 같은 터미널 관련 명령어가 실행되는 것을 막을 수 있습니다. 다음 스크립트 예를 참조하십시오.

`.login` (C shell)

```
if ( ! ${?DT} ) then
    stty erase `^h`
endif
```

`.profile` (sh or ksh)

```
if [ ! "$DT" ]; then
    stty erase `^h`
fi
```

DT 주변의 다른 구문은 DT가 정의되지 않았을 때 C 셸이 경고를 내보내지 않도록 합니다. 사용자가 종전의 텍스트 방식 콘솔 로그인 프롬프트에서 로그인한 경우 DT는 정의되지 않습니다. 도트 파일 설정에 대한 자세한 내용은 *HomeDirectory/.dtprofile* 파일을 참조하십시오.

주 - 사용자는 Solaris CDE tty 설정값을 *.xdefaults*에 지정합니다.

사용자의 *.xdefaults* 파일은 Solaris CDE 시작시 특정 사용자 자원에 대해 소스로 사용됩니다. 예를 들어, *ttyModes*는 사용자가 선호하는 tty 설정값이 *dtterm* 및 *xterm* 과 같은 터미널 에뮬레이션 윈도우에서 사용되도록 합니다. 다음 행은 *.xdefaults* 파일의 전형적인 *ttyModes* 설정값을 보여줍니다.

```
*ttyModes: erase ^H intr ^C kill ^U start ^Q stop ^S susp ^Z`
```

주 - 선호하는 Solaris CDE 자원은 기본값에 따라 다릅니다.

데스크탑 윈도우 관리자의 아이콘 위치는 기본 설정값의 한 예입니다. 이 경우 *.xdefaults* 파일의 다음 행이 기본 아이콘 위치 설정값을 보여줍니다.

```
Dtwm*iconPlacement: right top
```

응용 프로그램의 추가 및 관리

응용 프로그램 관리자는 사용자가 사용할 수 있는 응용 프로그램의 데스크탑 보관소입니다.

- “응용 프로그램 관리자의 구조” 57 페이지
- “응용 프로그램 관리자에 응용 프로그램 추가” 61 페이지
- “일반 응용 프로그램 그룹 작성 및 관리” 63 페이지
- “응용 프로그램 위치를 찾는데 사용되는 검색 경로 수정” 64 페이지
- “일반 응용 프로그램 관리자 관리” 66 페이지
- “문서 편집기 및 터미널 애플레이터 변경” 67 페이지

응용 프로그램 관리자의 구조

응용 프로그램 관리자의 최상위 레벨에는 일반적으로 디렉토리가 있습니다. 이러한 각 디렉토리와 내용을 **응용 프로그램 그룹**이라고 합니다.



응용 프로그램 그룹

그림 4-1 응용 프로그램 관리자의 응용 프로그램 그룹

응용 프로그램 그룹 및 내용은 지역적으로 또는 네트워크를 통한 여러 위치에서 모아집니다.

응용 프로그램 관리자의 디렉토리 위치

파일 시스템에서 응용 프로그램 관리자는 `/var/dt/appconfig/appmanager/login-hostname-display` 디렉토리입니다. 디렉토리는 사용자가 로그인할 때마다 동적으로 작성됩니다.

예를 들어, 사용자 ronv가 표시장치 wxyz:0으로부터 로그인한 경우, 응용 프로그램 관리자 디렉토리 `/var/dt/appconfig/appmanager/ronv-wxyz-0`이 작성됩니다.

응용 프로그램 관리자가 응용 프로그램을 찾고 모으는 방법

지역 및 원격 응용 프로그램 그룹을 모으면 응용 프로그램 관리자가 구축됩니다. 응용 프로그램 그룹은 응용 프로그램 검색 경로에 위치한 디렉토리로부터 모아집니다.

기본 응용 프로그램 검색 경로는 표 4-1과 같이 구성됩니다.

표 4-1 기본 응용 프로그램 검색 경로 위치

범위	위치
내장	<code>/usr/dt/appconfig/appmanager/language</code>
시스템 전체	<code>/etc/dt/appconfig/appmanager/language</code>
개인	홈 디렉토리/ <code>.dt/appmanager</code>

응용 프로그램의 최상위 레벨을 작성하기 위해 응용 프로그램 관리자 디렉토리 `/var/dt/appconfig/appmanager/login-hostname-display`에 대한 응용 프로그램 검색 경로에 있는 그룹(디렉토리)의 응용 프로그램 그룹에서 로그인할 때 링크가 작성됩니다. 모으기 작업은 데스크탑 유틸리티 `dtappgather`에 의해 수행되며, 이 유틸리티는 사용자가 성공적으로 로그인하면 로그인 관리자가 자동으로 실행합니다.

예를 들어, 데스크탑은 다음과 같은 내장 응용 프로그램 그룹을 제공합니다.

```
/usr/dt/appconfig/appmanager/language/Desktop_Tools
```

로그인 시 기호 링크가 다음의 위치에 작성됩니다.

```
/var/dt/appconfig/appmanager/login-hostname-display/Desktop_Tools
```

응용 프로그램 검색 경로에 원격 디렉토리가 포함될 수 있습니다. 이 검색 경로는 네트워크에 위치한 시스템으로부터 응용 프로그램 그룹을 모으는 방법을 제공합니다. 자세한 내용은 51 페이지의 “응용 프로그램 검색 경로에 응용 프로그램 서버 추가” 65 페이지를 참조하십시오.

응용 프로그램 모으기의 우선순위 규칙

검색 경로에 중복이 존재하는 경우, 개인 응용 프로그램 그룹이 시스템 전체 그룹에 우선하며, 시스템 전체 그룹이 내장 그룹에 우선합니다. 예를 들어, `/usr/dt/appconfig/appmanager/C/Desktop_Tools` 및 `/etc/dt/appconfig/appmanager/C/Desktop_Tools`가 모두 존재할 경우, `/etc` 아래에 있는 응용 프로그램 그룹이 사용됩니다.

기본 데스크탑이 제공하는 응용 프로그램 그룹

사용자 정의되지 않은 데스크탑은 네 개의 응용 프로그램 그룹을 제공합니다.

- 데스크탑_응용프로그램 (Desktop_Apps)
- 데스크탑_도구 (Desktop_Tools)
- 데스크탑_제어기 (Desktop_Controls)
- 정보 (Information)
- 시스템_관리 (System_Admin)

응용 프로그램 그룹을 모으는 방법의 예

그림 4-2는 다양한 응용 프로그램 그룹이 포함된 응용 프로그램 관리자 윈도우를 보여줍니다. 표 4-2는 응용 프로그램 그룹의 소스 디렉토리를 보여줍니다.



그림 4-2 전형적인 응용 프로그램 관리자 윈도우

표 4-2 의 응용 프로그램 그룹 소스

이름	모은 디렉토리
CAD_App	/net/ApServA/etc/dt/appconfig/appmanager/C/CAD_App
DrawingApp	/etc/dt/appconfig/appmanager/C/DrawingApp
Desktop_Apps	/usr/dt/appconfig/appmanager/C/Desktop_Apps
데스크탑_도구	/usr/dt/appconfig/appmanager/C/Desktop_Tools
정보	/usr/dt/appconfig/appmanager/C/Information
OpenWindows	/usr/dt/appconfig/appmanager/C/Information
시스템_관리	/etc/dt/appconfig/appmanager/C/System_Admin
MySpreadSheet	/users/anna/.dt/appmanager/MySpreadSheet
Media_Tools	/etc/dt/appconfig/appmanager/C/Media_Tools

정보나 시스템_관리 응용 프로그램 그룹을 사용자 정의한 경우에는 /etc/dt/appconfig/appmanager/C에서 모아야 합니다.

이름이 ApServA인 시스템이 응용 프로그램 검색 경로에 추가되었기 때문에 CAD_App 그룹을 모읍니다(“응용 프로그램 검색 경로에 응용 프로그램 서버 추가” 65 페이지 참조). MySpreadSheet는 개인 응용 프로그램 그룹이며 사용자 anna만이 사용할 수 있습니다.

응용 프로그램 관리자에 응용 프로그램 추가

응용 프로그램이 응용 프로그램 관리자에 추가되면 응용 프로그램 그룹내에 해당 응용 프로그램을 시작하는 아이콘이 생깁니다.

많은 응용 프로그램이 응용 프로그램 그룹을 제공합니다. 응용 프로그램 그룹은 응용 프로그램 관리자의 최상위 레벨에 있는 디렉토리로서, 응용 프로그램 아이콘과 응용 프로그램과 관련된 다른 파일이 포함되어 있습니다.

일부 응용 프로그램에는 자체 응용 프로그램 그룹이 없을 수도 있습니다. 그 대신, 응용 프로그램을 시작하는 아이콘이 일반 응용 프로그램 그룹에 위치합니다. 예를 들어, 이름이 "Games"인 빈 응용 프로그램 그룹을 작성하여 시스템에 설치된 모든 게임의 보관소로 사용할 수 있습니다.

응용 프로그램 관리자에 응용 프로그램을 추가하는 방법

응용 프로그램을 응용 프로그램 관리자에 추가하는 방법은 두 가지가 있습니다.

- 응용 프로그램을 등록하는 방법
- 응용 프로그램을 등록하지 않고 응용 프로그램 아이콘을 추가하는 방법

응용 프로그램 등록

응용 프로그램 등록은 전체 응용 프로그램 통합을 제공합니다.

등록된 응용 프로그램의 특성은 다음과 같습니다.

- 자체의 응용 프로그램 그룹을 갖고 있습니다.
- 단일 위치에 모아진 자체 데스크탑 환경설정 파일을 갖고 있습니다. 이 데스크탑 환경설정 파일 그룹을 **등록 패키지**라고 합니다.
- 등록된 도움말 볼륨이 있을 수 있습니다.

응용 프로그램을 등록하는 방법은 두 가지가 있습니다.

- 데스크탑 스마트 응용 프로그램을 설치하면 자동으로 등록됩니다. "응용 프로그램 관리자에 데스크탑 스마트 응용 프로그램을 추가하는 방법" 62 페이지를 참조하십시오.
- 기존 응용 프로그램은 등록 패키지를 작성함으로써 등록할 수 있습니다. "기존 응용 프로그램 또는 데스크탑 스마트가 아닌 응용 프로그램을 등록하는 방법" 62 페이지를 참조하십시오.

등록 패키지를 사용하면 데스크탑에서 응용 프로그램을 관리하기가 쉽습니다. 등록 패키지는 데스크탑 환경설정 파일에 사용되는 위치가 아닌 파일 시스템의 위치에 작성됩니다.

등록 패키지를 사용하지 않고 응용 프로그램 추가하기

응용 프로그램 관리자에 응용 프로그램을 시작하는 아이콘만을 포함하도록 하려는 경우, 이 방법이 선호되는 응용 프로그램 추가 방법입니다.

등록 패키지를 사용하지 않고 추가된 응용 프로그램의 특성은 다음과 같습니다.

- 자체 응용 프로그램 그룹을 가질 수도 있지만 보통 해당 아이콘이 기존 응용 프로그램 그룹에 위치합니다.
- 데스크탑 환경설정 파일이 데스크탑 검색 경로에 위치합니다.

“기존 응용 프로그램 그룹에 응용 프로그램 아이콘을 추가하는 방법” 63 페이지를 참조하십시오.

▼ 응용 프로그램 관리자에 데스크탑 스마트 응용 프로그램을 추가하는 방법

데스크탑 스마트 응용 프로그램은 응용 프로그램이 설치될 때 자동으로 응용 프로그램 관리자에 등록되는 응용 프로그램입니다. 응용 프로그램의 파일 세트에는 데스크탑이 필요로 하는 등록 패키지가 포함되어 있습니다.

1. 응용 프로그램과 함께 제공된 지시사항에 따라 응용 프로그램을 설치하십시오.
2. 설치가 끝나면, 데스크탑_제어기의 추가 폴더에 있는 응용프로그램 재로드를 두 번 누르십시오.
3. 설치가 완료되었는지 확인하십시오.
 - a. 응용 프로그램 관리자를 열고 새 응용 프로그램 그룹이 생겼는지 점검하십시오.
 - b. 응용 프로그램을 열려면, 응용 프로그램 그룹을 열고 응용 프로그램 아이콘을 두 번 누르십시오.

▼ 기존 응용 프로그램 또는 데스크탑 스마트가 아닌 응용 프로그램을 등록하는 방법

이 방법은 응용 프로그램을 데스크탑으로 완전히 통합할 때 선호하는 방법입니다.

데스크탑은 등록 패키지 파일과 데스크탑의 검색 경로 간의 링크를 작성하는 도구인 dtappintegrate를 제공합니다.

데스크탑 등록은 제 5 장에서 설명합니다.

▼ 기존 응용 프로그램 그룹에 응용 프로그램 아이콘을 추가하는 방법

이 절차는 기존 응용 프로그램 그룹에 응용 프로그램 아이콘을 추가하는 방법을 설명합니다.

예를 들어, 데스크탑은 이름이 시스템_관리인 응용 프로그램 그룹을 제공하며 이 그룹은 시스템 관리와 관련된 다양한 응용 프로그램과 스크립트를 위해 예약된 것입니다. 자주 실행하는 스크립트가 있는 경우, 이 스크립트를 시스템_관리 응용 프로그램 그룹에서 아이콘을 두 번 눌러 실행할 수도 있습니다.

1. **활동 만들기 (활동 생성)을 사용하여 응용 프로그램에 대한 활동 정의를 작성하십시오.**
자세한 내용은 제 11 장을 참조하십시오.
2. **응용 프로그램 그룹 디렉토리에 활동 이름과 동일한 이름으로 된 실행 파일을 만드십시오.** 파일의 내용은 상관이 없습니다.

예를 들어, 시스템 관리 도구를 실행시키는 “Cleanup”이라는 활동이 작성된 경우, 다음과 같은 실행 파일이 작성됩니다.

```
/etc/dt/appconfig/appmanager/language/System_Admin/Cleanup
```

일반 응용 프로그램 그룹 작성 및 관리

일반 응용 프로그램은 하나의 특정 응용 프로그램 제품과 연관되지 않은 응용 프로그램 그룹(디렉토리)입니다. 예를 들어, 내장 데스크탑_도구 응용 프로그램 그룹은 단일 제품과 관련이 있지만 그 일부는 아닌 여러 개의 응용 프로그램 아이콘을 포함하는 일반 그룹입니다.

사용자는 추가로 일반 응용 프로그램 그룹을 작성할 수 있습니다. 예를 들어, 시스템에서 사용 가능한 다양한 게임을 함께 그룹화하기 위해 게임이라는 그룹을 작성할 수 있습니다.

일반 응용 프로그램 그룹의 범위는 시스템 전체 또는 개인이 될 수 있습니다.

▼ 시스템 전체 일반 응용 프로그램 그룹을 작성하는 방법

1. 루트로 로그인하십시오.
2. /etc/dt/appconfig/appmanager/language에 디렉토리를 만드십시오.

디렉토리 이름이 응용 프로그램 그룹의 이름이 됩니다.

3. 데스크탑_제어기의 추가 폴더에 있는 응용프로그램 재로드를 두 번 누르십시오.

▼ 개인 일반 응용 프로그램 그룹을 작성하는 방법

1. *HomeDirectory/.dt/appmanager*에 디렉토리를 만드십시오.
디렉토리 이름이 응용 프로그램 그룹의 이름이 됩니다.
2. 데스크탑_제어기의 추가 폴더에 있는 응용프로그램 재로드를 두 번 누르십시오.

▼ 내장 응용 프로그램 그룹을 사용자 정의하는 방법

1. 루트로 로그인하십시오.
2. 응용 프로그램 그룹이 */usr/dt/appconfig/appmanager/language*에 위치한 경우, 응용 프로그램 그룹을 */etc/dt/appconfig/appmanager/language*로 복사하십시오.
예를 들어, 다음 명령어는 데스크탑_도구 응용 프로그램 그룹을 복사합니다.

```
cp -r /usr/dt/appconfig/appmanager/C/Desktop_Tools \  
/etc/dt/appconfig/appmanager/C
```

응용 프로그램 그룹의 새로운 사본은 내장 버전보다 우선합니다.

3. 응용 프로그램 그룹의 사본을 수정하십시오. 예를 들어, 새 활동 파일(활동과 이름이 동일한 실행 파일)을 추가할 수 있습니다.
4. 변경 내용을 보려면, 로그아웃 후 다시 로그인하십시오.

응용 프로그램 위치를 찾는데 사용되는 검색 경로 수정

응용 프로그램 검색 경로를 수정하는 가장 큰 이유는 응용 프로그램 서버를 추가하기 위해서입니다. 응용 프로그램 서버를 검색 경로에 추가할 때 응용 프로그램 관리자는 서버의 모든 시스템 전체 응용 프로그램 그룹을 모읍니다.

응용 프로그램 검색 경로에 대한 자세한 내용은 “응용 프로그램 검색 경로” 148 페이지를 참조하십시오.

기본 검색 경로

기본 응용 프로그램 검색 경로에는 표 4-3에 표시된 디렉토리가 포함됩니다.

표 4-3 기본 응용 프로그램 검색 경로 디렉토리

범위	검색 경로 디렉토리
개인	<i>HomeDirectory/.dt/appmanager/</i>
시스템 전체	<i>/etc/dt/appconfig/appmanager/language</i>
내장	<i>/usr/dt/appconfig/appmanager/language</i>

응용 프로그램 검색 경로에 응용 프로그램 서버 추가

응용 프로그램 검색 경로를 수정하는 것 외에 응용 프로그램 서버와 통신하기 위해 추가 구성 작업을 수행해야 할 수도 있습니다. “응용 프로그램 서비스의 관리” 131 페이지를 참조하십시오.

▼ 시스템 전체 응용 프로그램 검색 경로를 설정하는 방법

1. 루트로 로그인하십시오.
2. 파일 `/etc/dt/config/Xsession.d/0010.dtpaths`가 없으면, `/usr/dt/config/Xsession.d/0010.dtpaths`를 복사하여 만드십시오.
3. 편집하려면 `/etc/dt/Xsession.d/0010.paths`를 여십시오. **DTSPSYSAPPHOSTS** 변수를 설정하여 내보내는 다음과 같은 명령행을 추가하거나 편집하십시오.

```
export DTSPSYSAPPHOSTS=hostname: [,hostname]
```

예를 들어, 다음과 같은 명령행은 응용 프로그램 검색 경로에 시스템 ApServA를 추가합니다.

```
export DTSPSYSAPPHOSTS=ApServA:
```

4. 시스템의 모든 사용자에게 로그아웃한 후 다시 로그인하여 변경 내용을 적용해야 한다고 알려 주십시오.

▼ 개인 응용 프로그램 검색 경로를 설정하는 방법

1. `HomeDirectory/.dtprofile`을 열고 편집하십시오.
2. **DTSPUSERAPPHOSTS** 변수를 설정한 후 내보내는 다음과 같은 행을 추가하거나 편집하십시오.

```
export DTSPSYSAPPHOSTS=hostname: [,hostname]
```

예를 들어, 다음과 같은 명령행은 응용 프로그램 검색 경로에 시스템 ApServB와 시스템 ApServC를 추가합니다.

```
export DTSPSYSAPPHOSTS=ApServB: ,ApServC
```

3. 로그아웃한 후 다시 로그인하십시오.

일반 응용 프로그램 관리자 관리

일반 응용 프로그램 관리자 관리 업무는 다음과 같습니다.

- 응용 프로그램의 제거
- 세션 중에 응용 프로그램의 데이터베이스 다시 읽기

▼ 응용 프로그램을 제거하는 방법

응용 프로그램이 dtappintegrate 도구를 사용하여 등록된 경우, dtappintegrate 를 사용하여 처리를 반대로 진행할 수 있습니다. 응용 프로그램이 등록 취소되면, 응용 프로그램 그룹이 응용 프로그램 관리자에서 제거되며 응용 프로그램의 활동, 데이터 유형, 아이콘 및 도움말을 더 이상 사용할 수 없습니다.

1. 루트로 로그인하십시오.
2. 다음 명령어를 실행하십시오.

```
dtappintegrate -s app_root -u
```

▼ 세션 중에 응용 프로그램 관리자를 갱신하는 방법

응용 프로그램을 추가하고 그 변경 내용이 즉시 반영되도록 하려면 응용 프로그램 관리자를 재구축해야 합니다.

- 데스크탑_제어의 추가 폴더에 있는 응용프로그램 재로드를 두 번 누르십시오.

응용프로그램 재로드는 응용 프로그램이 응용 프로그램 서버에 추가되었을 때 응용 프로그램 관리자를 갱신하는데 유용합니다. 그러나 응용 프로그램 재로드는 응용 프로그램 서버에서 제거된 응용 프로그램이나 다른 위치로 이동한 응용 프로그램을 찾지는 못합니다. 이 변경 내용은 사용자가 로그아웃한 후 다시 로그인했을 때 반영됩니다.

문서 편집기 및 터미널 에플레이터 변경

문서 편집기와 터미널 에플레이터 응용 프로그램은 프론트 패널의 제어 도구를 선택하거나 응용 프로그램 관리자의 아이콘을 두 번 누르면 시작됩니다.

이 응용 프로그램들은 기타 데스크탑 활동에 의해서도 시작됩니다.

- 문서 편집기 응용 프로그램은 사용자가 파일 관리자에서 텍스트 파일을 선택하고 선택 메뉴에서 열기를 선택하면 열립니다. 기본 문서 편집기는 `dtpad`입니다.
- 터미널 에플레이터는 사용자가 파일 관리자의 파일 메뉴에서 터미널 열기를 선택하거나 활동이 터미널 에플레이터 윈도우를 열면 실행됩니다. 기본 터미널 에플레이터는 `dtterm`입니다.

사용자는 이 상황에서 데스크탑이 다른 문서 편집기나 터미널 에플레이터 응용 프로그램을 사용하도록 구성할 수 있습니다.

▼ 기본 문서 편집기 또는 터미널 에플레이터를 변경하는 방법

1. 변경 내용이 시스템 전체에 걸친 경우에는 루트로 로그인하십시오.
2. 새로운 문서 편집기 또는 터미널 에플레이터 응용 프로그램에 대한 활동을 작성하십시오.
 - 활동 만들기 응용 프로그램을 사용할 수 있습니다. 그림 4-3은 TextPad라는 응용 프로그램에 대한 활동명을 입력한 활동 만들기 윈도우입니다. 자세한 내용은 제 11 장을 참조하십시오.

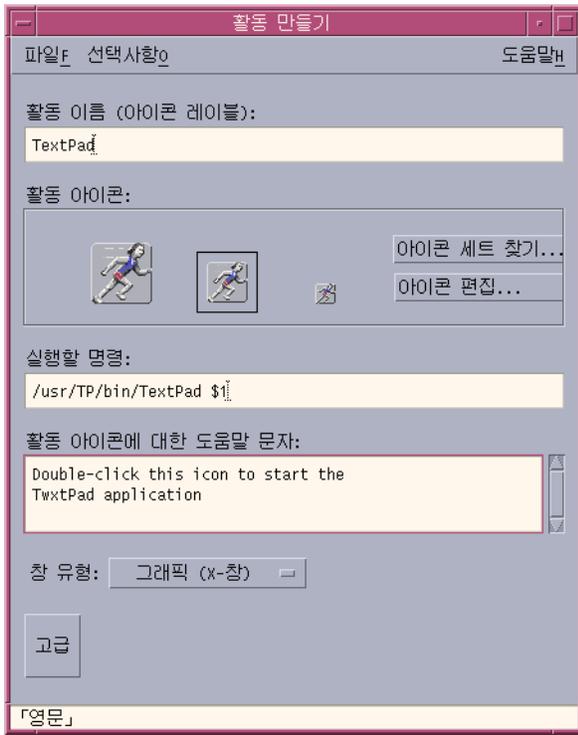


그림 4-3 활동 만들기 윈도우

- 또는 수동으로 활동 정의를 작성할 수 있으며, 그 예는 다음과 같습니다.

```

ACTION TextPad
{
    LABELTextPad
    TYPE                COMMAND
    WINDOW_TYPE         NO_STDIO
    EXEC_STRING         /usr/TP/bin/TextPad %(File)Arg_1%
    DESCRIPTION         TextPad 응용 프로그램을 시작하려면 \
                        이 아이콘을 두 번 누르십시오.
}

```

수동으로 활동 정의를 작성하는 방법에 대한 내용은 제 12 장을 참조하십시오.

3. 새로운 활동을 포함하는 환경설정 파일을 해당 디렉토리에 놓으십시오.

- 시스템 전체: `/etc/dt/appconfig/types/language`
- 개인: `HomeDirectory/.dt/types`

4. 환경설정 파일이 없는 경우에는

`/usr/dt/appconfig/types/language/user-prefs.dt`를 다음 디렉토리로 복사하여 해당 `user-prefs.dt` 파일을 작성하십시오.

- 시스템 전체: `/etc/dt/appconfig/types/language` 디렉토리

■ 개인: *HomeDirectory/.dt/types* 디렉토리

5. 시스템 전체 또는 개인 *user-prefs.dt* 파일에서 문서 편집기 또는 터미널 활동을 편집하십시오. **MAP_ACTION** 행을 수정하여 활동을 새 활동으로 매핑하십시오.

예를 들어, 명령행

```
MAP_ACTION Dtpad
```

를

```
MAP_ACTION TxtPd
```

6. 로 변경하십시오. *user-prefs.dt* 파일을 저장하십시오.
7. 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 눌러, 활동 데이터베이스를 재로드하십시오.

응용 프로그램 등록

이 장에서는 응용 프로그램의 등록 패키지를 작성하는 방법과 데스크탑에 응용 프로그램을 등록하는 방법에 대해 설명합니다.

- “응용 프로그램 등록 개요” 72 페이지
- “응용 프로그램 등록을 위한 일반적 단계” 75 페이지
- “1단계: 글꼴 및 색상 자원 수정” 75 페이지
- “2단계: 데스크탑 응용 프로그램 루트 작성” 77 페이지
- “3단계: 등록 패키지 디렉토리 작성” 77 페이지
- “4단계: 응용 프로그램의 활동 및 데이터 유형 작성” 79 페이지
- “5단계: 등록 패키지에 도움말 파일 넣기” 82 페이지
- “6단계: 응용 프로그램 아이콘 작성” 83 페이지
- “7단계: 응용 프로그램 그룹 작성” 84 페이지
- “8단계: dtappintegrate를 사용한 응용 프로그램 등록” 89 페이지
- “등록 패키지 작성 예” 92 페이지

응용 프로그램 전체를 데스크탑에 등록하면 다음과 같은 항목을 얻게 됩니다.

- 응용 프로그램 관리자의 최상위 레벨에 있는 자신의 응용 프로그램 그룹
- 응용 프로그램을 시작하는 활동 활동은 응용 프로그램 그룹의 아이콘으로 나타납니다.
- 데이터 파일의 데이터 유형(선택항목)

응용 프로그램을 등록하더라도 해당 프로그램을 운영하는 데에는 아무런 영향을 주지 않습니다.

- 응용 프로그램 등록 작업은 해당 프로그램의 실행 파일을 수정하는 것이 아닙니다. 따라서, 기존 응용 프로그램을 시스템에 등록할 수 있습니다.
- 응용 프로그램으로 전달된 파일(예: 실행 파일 및 app-defaults 등)을 다른 위치로 이동할 필요는 없습니다.
- 실행은 간단하게 취소할 수 있습니다. 응용 프로그램을 등록하는데 사용되는 dtappintegrate 도구는 프로세스를 취소하기 위한 명령행 옵션을 제공합니다.

다음과 같은 사용자는 등록 패키지를 작성할 수도 있습니다.

- 기존 응용 프로그램을 데스크탑에 등록하고자 하는 시스템 관리자
- 데스크탑 스마트 응용 프로그램의 설치 패키지를 작성하고자 하는 소프트웨어 프로그래머

응용 프로그램 등록 개요

이 절에서 다루는 내용은 다음과 같습니다.

- 응용 프로그램 등록의 목적
- 응용 프로그램의 등록으로 해당 프로그램에 제공되는 기능

주 - 기존 응용 프로그램 등록 방법에 관한 자세한 예는 “등록 패키지 작성 예” 92 페이지를 참조하십시오.

응용 프로그램 등록으로 제공되는 기능

응용 프로그램 등록은 다음과 같은 작업을 사용자에게 그래픽 방식으로 제공합니다.

- 응용 프로그램을 찾습니다.
설치시, 사용자 응용 프로그램은 응용 프로그램 관리자에 “등록”되고 자신의 응용 프로그램 그룹을 갖게 됩니다.

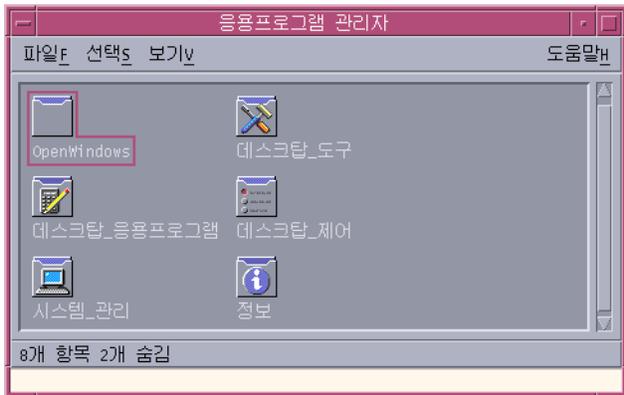


그림 5-1 응용 프로그램 관리자 최상위 레벨에 있는 응용 프로그램 그룹

- 응용 프로그램을 시작합니다.
사용자의 응용 프로그램에 대한 응용 프로그램 그룹에는 사용자가 두 번 눌러서 응용 프로그램을 시작할 수 있는 아이콘이 포함되어 있습니다.

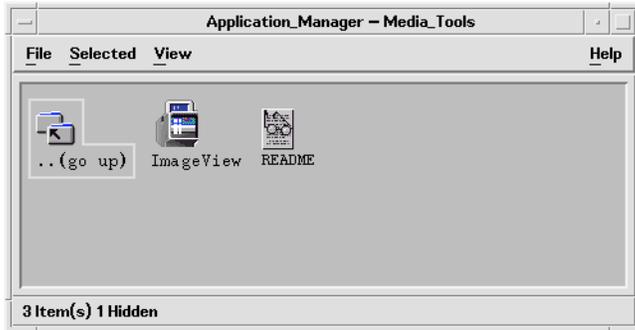


그림 5-2 응용 프로그램을 시작하는 아이콘이 있는 응용 프로그램 그룹

- 데이터 파일을 확인하고 조작합니다. 응용 프로그램의 데이터 파일은 파일 관리자 내에 고유한 아이콘을 가지고 있습니다.
데이터 파일 아이콘을 사용하는 용도는 다음과 같습니다.
 - 응용 프로그램 시작(개시)
 - 데이터 파일 인쇄

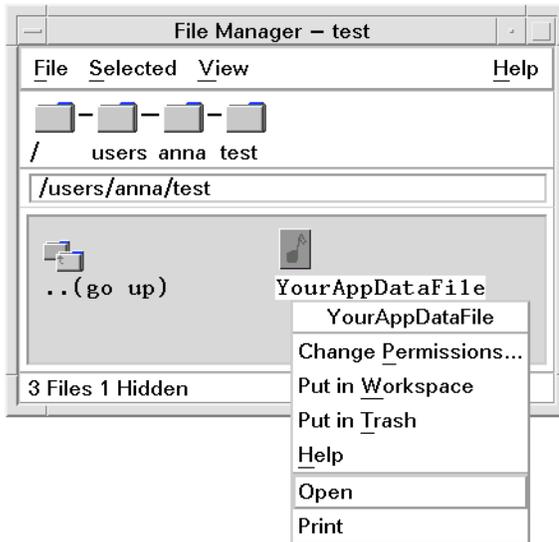


그림 5-3 "열기"와 "인쇄"가 있는 데이터 파일 팝업 메뉴

- 우편발송, 압축, 보기 또는 (오디오) 데이터 재생과 같은 기타 작업을 수행합니다.

응용 프로그램 등록의 목적

등록된 데스크탑 응용 프로그램에는 응용 프로그램의 사용자 인터페이스를 제공하기 위해 데스크탑이 사용하는 다음과 같은 특정한 환경설정 파일들이 있습니다.

- 활동 및 데이터 유형 정의 파일
- 아이콘 이미지(픽스맵 또는 비트맵) 파일
- 응용 프로그램 그룹을 작성하는 디렉토리 및 파일
- 데스크탑 도움말 파일 및 프론트 패널 정의 파일(선택항목)

데스크탑이 이러한 파일을 인식하고 사용하려면 파일이 데스크탑의 검색 경로가 지정하는 특정 디렉토리에 있어야 합니다.

응용 프로그램의 환경설정 파일이 여러 디렉토리에 흩어져 있는 경우, 응용 프로그램을 관리하기가 어려울 수 있습니다. 따라서, 데스크탑은 응용 프로그램이 자신의 데스크탑 환경설정 파일을 하나의 디렉토리에 모아두는 것을 허용합니다. 이렇게 파일을 그룹화하는 것을 **등록 패키지**라고 합니다.

응용 프로그램이 데스크탑 스마트인 경우, 설치 패키지의 일부로 등록 패키지를 제공합니다. 사용자가 환경설정 파일을 직접 만드는 시스템 관리자인 경우 등록 패키지를 직접 만들 수 있습니다.

등록 패키지의 환경설정 파일은 적절한 검색 경로 디렉토리에 있지 않으므로 데스크탑에서 사용할 수 없습니다. 이 파일을 적절한 위치에 갖다 놓는 프로세스를 응용 프로그램 **등록** 또는 **통합**이라고 합니다.

데스크탑은 파일의 기호 링크 표현을 적절한 검색 경로 디렉토리에 작성함으로써 등록을 수행하는 도구인 dtappintegrate를 제공합니다.

대부분의 데스크탑 스마트 응용 프로그램은 설치 프로세스중에 자동으로 dtappintegrate를 실행합니다. 사용자가 기존 응용 프로그램을 통합하는 시스템 관리자인 경우 등록 패키지를 작성한 후에 직접 이를 실행할 수 있습니다.

일단 응용 프로그램이 시스템 데스크탑에 등록되면 시스템상의 모든 사용자가 응용 프로그램을 사용할 수 있습니다. 시스템이 데스크탑 응용 프로그램 서버로 구성되면, 네트워크 전반의 다른 시스템에서도 응용 프로그램을 사용할 수 있습니다.

dtappintegrate 도구에는 링크를 끊어서 프로세스를 취소하는 명령행 옵션이 있습니다. 이 옵션을 사용하여 응용 프로그램을 응용 프로그램 관리자에서 쉽게 제거하여 다른 응용 프로그램 서버로 이동하거나 갱신할 수 있습니다.

응용 프로그램 등록을 위한 일반적 단계

주 - 이러한 단계를 사용하여 응용 프로그램 패키지를 작성하는 자세한 예는 “등록 패키지 작성 예” 92 페이지를 참조하십시오.

1. 글꼴 및 색상을 설정하는 모든 응용 프로그램 자원을 수정하십시오. 그렇지 않으면, 데스크탑의 동적 글꼴 및 색상이 제대로 작동되지 않을 것입니다.
“1단계: 글꼴 및 색상 자원 수정” 75 페이지를 참조하십시오.
2. 응용 프로그램 루트 위치를 만드십시오.
“2단계: 데스크탑 응용 프로그램 루트 작성” 77 페이지를 참조하십시오.
3. 응용 프로그램 루트 밑에 디렉토리 구조를 만드십시오.
“3단계: 등록 패키지 디렉토리 작성” 77 페이지를 참조하십시오.
4. 응용 프로그램의 활동 및 데이터 유형을 만드십시오.
“4단계: 응용 프로그램의 활동 및 데이터 유형 작성” 79 페이지를 참조하십시오.
5. 도움말 파일을 적합한 디렉토리에 놓으십시오.
“5단계: 등록 패키지에 도움말 파일 넣기” 82 페이지를 참조하십시오.
6. 응용 프로그램의 아이콘을 만드십시오.
“6단계: 응용 프로그램 아이콘 작성” 83 페이지를 참조하십시오.
7. 응용 프로그램에 대한 응용 프로그램 그룹을 만드십시오.
“7단계: 응용 프로그램 그룹 작성” 84 페이지를 참조하십시오.
8. dtappintegrate를 사용하여 응용 프로그램을 등록하십시오.
“8단계: dtappintegrate를 사용한 응용 프로그램 등록” 89 페이지를 참조하십시오.

1단계: 글꼴 및 색상 자원 수정

주 - 응용 프로그램용 자원을 수정하는 예는 “등록 패키지 작성 예” 92 페이지에서 1 단계를 참조하십시오.

데스크탑은 인터페이스 글꼴 및 윈도우 색상을 설정하고 조작하는 메커니즘을 제공합니다. 응용 프로그램이 이 메커니즘을 적절히 사용하도록 응용 프로그램의 `app-defaults` 파일을 수정해야 할 수도 있습니다.

글꼴 자원 수정

주 - 이 절은 OSF/Motif 1.2™(또는 그 이후 버전)를 사용하여 만들어진 응용 프로그램에 적용됩니다. 스타일 관리자는 이전 버전의 OSF/Motif를 사용하여 기록된 응용 프로그램용 인터페이스 글꼴을 설정할 수 없습니다.

응용 프로그램이 특정 응용 프로그램 인터페이스 글꼴을 지정하지 않은 경우, 데스크탑 스타일 관리자는 OSF/Motif 1.2(또는 그 이후 버전)를 사용하여 작성된 응용 프로그램용 인터페이스 글꼴을 설정합니다.

스타일 관리자가 제공하는 두 가지 글꼴은 다음과 같습니다.

시스템 글꼴—레이블, 메뉴 및 버튼과 같은 시스템 영역에서 사용

사용자 글꼴—텍스트 필드와 같은 편집 가능한 영역에서 사용

각 글꼴은 7가지 크기로 제공되며 글꼴 대화상자에 1~7 레이블로 표시되어 있습니다. 스타일 관리자 글꼴은 /usr/dt/app-defaults/language/Dtstyle에 설정된 스타일 관리자 자원을 통해 시스템의 실제 글꼴과 연결됩니다.

응용 프로그램이 스타일 관리자 글꼴을 사용하도록 하려면 지정된 글꼴과 인터페이스하는 응용 프로그램 자원을 제거해야 합니다. 데스크탑이 자동으로 응용 프로그램 자원을 적절하게 설정합니다.

FontList—시스템 글꼴로 설정됨

XmText*FontList—사용자 글꼴로 설정됨

XmTextField*FontList—사용자 글꼴로 설정됨

색상 자원 수정

스타일 관리자는 응용 프로그램 색상을 동적으로 변경할 수 있습니다. 응용 프로그램은 OSF/Motif 1.1 또는 1.2 클라이언트여야 합니다. 그외의 툴킷으로 작성된 클라이언트는 색상을 동적으로 변경할 수 없습니다. 색상 변경은 클라이언트가 다시 시작할 때 반영됩니다.

데스크탑에 의해 제공된 동적 색상을 사용하는 가장 쉬운 방법은 배경색 및 전경색에 대한 모든 응용 프로그램 색상 자원을 제거하는 것입니다.

2단계: 데스크탑 응용 프로그램 루트 작성

주 - 응용 프로그램에 대한 데스크탑 응용 프로그램 루트 디렉토리의 작성 예는 “등록 패키지 작성 예” 92 페이지에서 2단계를 참조하십시오.

응용 프로그램에 대한 등록 패키지 파일은 응용 프로그램 루트 또는 `app_root` 디렉토리 아래에 그룹화됩니다. 데스크탑 환경설정 파일에 사용되는 `app_root` 디렉토리는 응용 프로그램 설치 `app_root` 또는 그외의 다른 위치와 동일한 디렉토리가 될 수 있습니다.

예를 들어, 응용 프로그램이 `/usr/BTE` 디렉토리 아래에 설치되어 있다고 가정합니다. 데스크탑 환경설정 파일의 경우는 이 디렉토리가 `app_root`로 사용될 수 있습니다. 그러나, 기존의 비데스크탑 스마트 응용 프로그램을 통합하는 경우에는 다른 데스크탑 `app_root` 디렉토리를 만드십시오. 이렇게 하면 사용자가 만든 환경설정 파일이 응용 프로그램을 갱신할 때 겹쳐 쓰여지는 것을 막을 수 있습니다.

예를 들어, 시스템 관리자가 디렉토리 `/etc/desktop_approots/BTE`를 데스크탑 `app_root` 디렉토리로 만들고자 할 수 있습니다.

3단계: 등록 패키지 디렉토리 작성

주 - 응용 프로그램에 대한 등록 패키지 디렉토리의 작성 예는 “등록 패키지 작성 예” 92 페이지에서 3단계를 참조하십시오.

등록 패키지는 응용 프로그램에 그래픽 인터페이스를 제공하기 위해 데스크탑이 사용하는 환경설정 파일의 그룹입니다.

등록 패키지 내용

데스크탑 환경설정 파일에는 다음 내용이 들어 있습니다.

- 활동 및 데이터 유형 정의 파일
- 아이콘 이미지 파일
- 응용 프로그램 그룹 디렉토리 및 내용
- 도움말 파일 및 프론트 패널 환경설정 파일(선택항목)

등록 패키지는 응용 프로그램 루트나 `app_root`라고 부르는 최상위 레벨 디렉토리 아래에 모입니다.

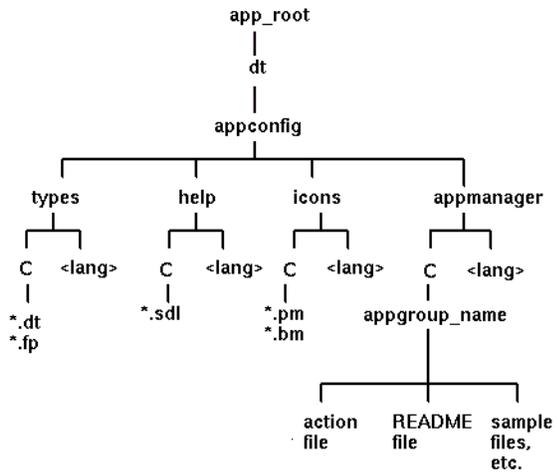


그림 5-4 응용 프로그램 루트 디렉토리 아래의 등록 패키지

`app_root/dt/appconfig` 디렉토리 아래의 주요 구성 필드 범주가 표 5-1에 나와있습니다.

표 5-1 주요 구성 필드 범주

하위 디렉토리	내용
types	활동 및 데이터 유형 정의 파일
help	데스크탑 도움말 파일
icons	응용 프로그램 활동 및 데이터 유형이 사용하는 비트맵 및 픽스맵 파일
appmanager	응용 프로그램 그룹을 작성하는 디렉토리 및 그 내용

각 주요 범주는 언어별 파일을 위한 하위 디렉토리를 가지고 있습니다. 기본 언어 파일은 C 디렉토리에 있습니다.

▼ 등록 패키지 작성 방법

- 다음과 같은 디렉토리를 작성하십시오. 사용자가 언어마다 다른 환경설정 파일을 제공하고 있는 경우에는 각 언어에 대해 별도의 디렉토리를 만드십시오. 사용자가 하나의 언어만 제공하고 있는 경우, 파일을 C 디렉토리에 두십시오.
 - `app_root/dt/appconfig/types/language`
 - `app_root/dt/appconfig/help/language`
 - `app_root/dt/appconfig/icons/language`

- `app_root/dt/appconfig/appmanager/language/appgroup_name`, 여기서 `appgroup_name`은 응용 프로그램 그룹의 이름입니다.

예를 들어, 그림 5-5는 `appgroup_name`이 “Media_Tools”인 그룹을 포함하는 응용 프로그램 관리자를 보여주고 있습니다.

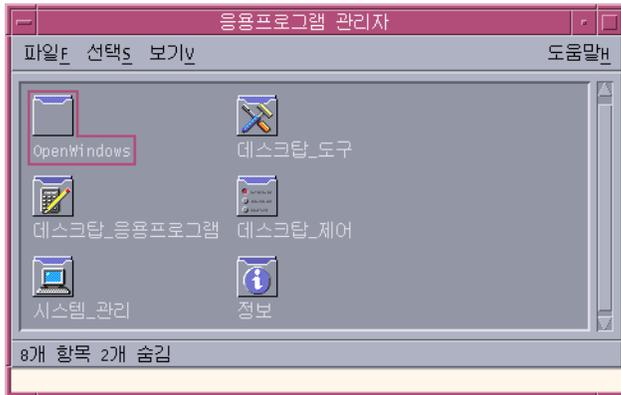


그림 5-5 응용 프로그램 관리자의 최상위 레벨에 있는 응용 프로그램 그룹

`dtappintegrate` 도구는 `types`, `help`, `icons` 및 `appmanager` 디렉토리에 있는 데스크탑 환경설정 파일에서만 동작합니다. 응용 프로그램의 이진 실행 파일, `app-defaults` 및 메시지 카탈로그 파일은 개별적으로 관리됩니다.

4단계: 응용 프로그램의 활동 및 데이터 유형 작성

주 - 응용 프로그램의 활동 및 데이터 유형을 작성하는 예는 “등록 패키지 작성 예” 92 페이지에서 4단계를 참조하십시오.

활동 및 데이터 유형은 응용 프로그램용 사용자 인터페이스를 제공합니다.

- 활동은 응용 프로그램을 시작하는 명령어에 대한 사용자 인터페이스를 제공합니다.
- 데이터 유형은 응용 프로그램 데이터 파일에 대해 사용자 정의된 모양과 동작을 제공합니다.

응용 프로그램이 요구하는 활동 및 데이터 유형

일반적인 응용 프로그램은 다음과 같은 활동 및 데이터 유형 정의를 요구합니다.

- 응용 프로그램을 여는 활동
- 사용자 응용 프로그램의 데이터 파일의 데이터 유형 데이터 유형을 작성하는 경우, 다음과 같은 활동도 작성할 수 있습니다.

- 응용 프로그램 데이터 파일의 열기 활동
- 응용 프로그램 데이터 파일의 인쇄 활동
- 응용 프로그램의 데이터 유형 “응용 프로그램 그룹이 고유한 아이콘을 사용하도록 구성하기” 85 페이지를 참조하십시오.
데스크탑에서 활동 및 데이터 유형을 사용하는 방법의 소개를 보려면 제 10 장을 참조하십시오.

활동 및 데이터 유형 정의 환경설정 파일의 위치

활동 및 데이터 유형은 환경설정 파일에서 정의합니다. 활동 및 데이터 유형 정의를 포함하는 파일 이름 지정에 대한 유일한 요구사항은 `.dt` 접미사를 가져야 한다는 것입니다. 규약에 따라, 사용자는 파일 `action_name.dt` 또는 `application_name.dt` 라는 파일 이름을 지정할 수 있습니다.

활동 및 데이터 유형을 포함하는 파일을 디렉토리 `app_root` `/dt/appconfig/types/language`의 응용 프로그램 루트 아래에 두십시오. 기본 `language`는 C입니다.

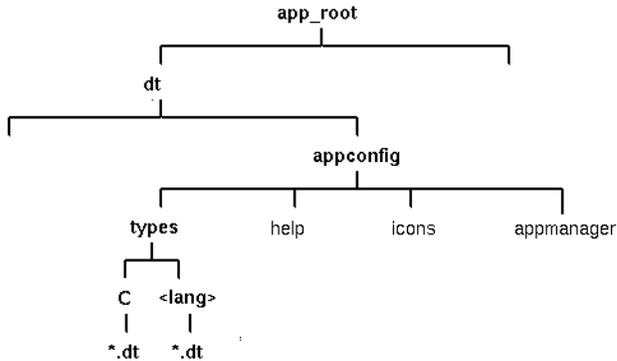


그림 5-6 활동 및 데이터 유형 정의 파일

활동 및 데이터 유형의 작성 방법

응용 프로그램의 활동 및 데이터 유형 정의를 다음 두 가지 방법 중 하나로 작성할 수 있습니다.

- 활동 만들기 (활동 작성) 도구를 사용하십시오.
활동 만들기는 입력해야 할 텍스트 필드와 함께 사용하기 쉬운 인터페이스를 제공합니다. 그러나 도구에는 일정한 제한이 있습니다.
- 수동으로 정의를 작성하십시오.
그러기 위해서는 정의를 작성하기 위한 구문을 알아야 하지만 대신 모든 기능에 액세스할 수 있습니다.

▼ 활동 만들기를 사용하여 활동 및 데이터 유형을 작성하는 방법

이 절차에서는 응용 프로그램의 활동 및 데이터 유형을 작성하기 위해 활동 만들기 유틸리티를 사용합니다.

활동 만들기에 관한 자세한 내용은 온라인 도움말을 사용하거나 제 11 장을 참조하십시오.

1. 데스크탑_응용프로그램 그룹을 열고 활동 만들기를 두 번 누르십시오.
2. 활동 만들기를 사용하여 응용 프로그램과 데이터 유형에 대한 활동 및 데이터 유형 정의를 작성하십시오.
활동 만들기로 작성된 환경설정 파일은 `HomeDirectory/.dt/type/action_name.dt`에 기록될 것입니다. 활동 파일(활동과 같은 이름으로 된 실행 파일)은 사용자의 홈 디렉토리에 위치합니다.
3. 사용자의 홈 디렉토리에서 작성된 활동 파일을 사용하여 활동을 테스트하십시오.
4. 활동 정의 파일 `HomeDirectory/.dt/type/action_name.dt`를 `app_root/dt/appconfig/types/language` 디렉토리에 복사하십시오.
5. 응용 프로그램 그룹 디렉토리를 작성한 후에는("7단계: 응용 프로그램 그룹 작성" 84 페이지 참조) 활동 파일 `HomeDirectory/action_name`을 `app_root/dt/appconfig/appmanager/language/appgroup_name` 디렉토리에 복사하십시오.

▼ 활동 및 데이터 유형을 수동으로 작성하는 방법

- 응용 프로그램에 대한 활동 및 데이터 유형 정의를 포함하는 환경설정 파일을 만드십시오.

활동 및 데이터 유형 정의 파일은 이름 지정 규약 이름.dt를 따라야 합니다.

사용자는 사용자의 모든 활동 및 데이터 유형 정의를 한 파일내에 둘 수도 있고, 여러 파일에 분산시킬 수도 있습니다. 각 파일에는 시스템 관리자가 사용자 응용 프로그램에 쉽게 접속할 수 있는 파일 이름을 사용하십시오.

활동 및 데이터 유형의 이름은 한 단어여야 합니다(공백이 포함되지 않아야 함). 밑줄은 사용할 수 없습니다. 규약에 따라 활동 또는 데이터 유형 이름의 첫 글자는 대문자여야 합니다. 이미 존재하는 활동 이름 또는 파일 이름을 사용하지 마십시오. 고급 사용자와 시스템 관리자가 사용자 응용 프로그램에 쉽게 접속할 수 있는 이름을 사용하십시오.

응용 프로그램 아이콘 레이블을 활동 이름과 다르게 하려면, 활동 정의에 LABEL 필드를 포함시키십시오.

활동 및 데이터 유형에 관한 자세한 내용은 다음을 참조하십시오.

- 제 10 장

- 제 11 장
- 제 12 장
- 제 13 장

5단계: 등록 패키지에 도움말 파일 넣기

주 - 도움말 파일을 등록 패키지에 추가하는 예는 “등록 패키지 작성 예” 92 페이지에서 5단계를 참조하십시오.

응용 프로그램이 데스크탑 도움말 볼륨(데스크탑 도움말 개발자 키트를 사용하여 작성한 도움말 볼륨)을 포함하는 경우, 도움말 볼륨 마스터 파일 (*.sdl)은 디렉토리 *app_root/appconfig/help/ language*에 두어야 합니다.

도움말 파일이 사용하는 그래픽은 일반적으로 *graphics* 하위 디렉토리에 위치합니다. 그래픽은 도움말 볼륨이 작성될 때 위치했던 마스터 도움말 볼륨(*.sdl) 파일과 같은 디렉토리에 위치해야 합니다.

응용 프로그램이 도움말 볼륨을 제공하지 않는 경우, 도움말 개발자 키트가 있다면 도움말을 만들 수 있습니다.

도움말 볼륨을 통합하는 데는 두 가지 레벨이 있습니다.

- 전체 통합

데스크탑 도움말이 전체적으로 통합되면 응용 프로그램(예: 항목 도움말 및 도움말 메뉴)에서 도움말 볼륨으로 액세스할 수 있습니다. 전체 통합은 응용 프로그램의 실행 파일을 수정하는 것도 포함됩니다.

- 부분 통합

데스크탑 도움말이 부분적으로 통합되면 도움말 관리자의 최상위 레벨에서 사용할 수 있습니다. 그러나 응용 프로그램 윈도우에서는 도움말 볼륨에 액세스할 수 없습니다. 응용 프로그램 그룹에서 도움말에 액세스하려면 활동을 제공하면 됩니다. 다음 활동 예는 도움말 마스터 파일 *MyApp.sdl*에 위치한 도움말 볼륨을 표시하고 있습니다.

```

ACTION OpenMyAppHelp
{
    LABEL           MyAppHelp
    ARG_COUNT       0
    TYPE            COMMAND
    WINDOW_TYPE     NO_STDIO
    EXEC_STRING     /usr/dt/bin/dthelpview -helpVolume MyApp
    DESCRIPTION     Displays help for the MyApp application.
}

```

6단계: 응용 프로그램 아이콘 작성

주 - 응용 프로그램의 아이콘 파일을 작성하는 예는 “등록 패키지 작성 예” 92 페이지에서 6단계를 참조하십시오.

데스크탑은 활동, 데이터 유형, 및 응용 프로그램 그룹에 대한 기본 아이콘을 제공합니다. 그러나 사용자는 응용 프로그램에 대한 고유한 아이콘을 작성할 수 있습니다.

아이콘은 `app_root/dt/appconfig/icons/language` 디렉토리에 위치하고 있습니다.

데스크탑에 필요한 아이콘

응용 프로그램은 데스크탑에서 다음과 같은 아이콘 이미지를 사용합니다.

- **활동 아이콘.** 응용 프로그램 (활동)을 시작하기 위해 두 번 누르는 아이콘입니다. 응용 프로그램을 시작하는 활동의 ICON 필드에서 참조합니다.
크기는 소형, 중형, 대형 세 가지가 있습니다.
- **데이터 유형 아이콘.** 파일 관리자에서 응용 프로그램의 데이터 파일을 표시할 때 사용됩니다. 데이터 유형 정의의 ICON 필드에서 참조합니다.
응용 프로그램이 여러 데이터 유형을 지원하는 경우, 각 데이터 유형에 서로 다른 아이콘을 제공해야 합니다.
크기는 소형과 중형 두 가지가 있습니다.
- **응용 프로그램 그룹 아이콘.** 응용 프로그램 관리자의 최상위 레벨에 있는 디렉토리를 표시하는 아이콘입니다. 응용 프로그램 그룹에 대한 데이터 유형 정의의 ICON 필드에서 참조합니다. (72 페이지의 “7단계: 응용 프로그램 그룹 작성” 84 페이지 참조.)
크기는 소형과 중형 두 가지가 있습니다.
컬러(8비트 이상)와 흑백(8비트 미만) 표시장치를 지원하기 위해 각 아이콘에 픽스맵 및 비트맵 버전을 모두 제공해야 하는 경우도 있습니다.

표 5-2 아이콘 파일의 이름 지정 규약

크기	픽셀 크기	비트맵 이름	픽스맵 이름
소형	16 X 16	<i>basename.t.bm</i>	<i>basename.t.pm</i>
중형	32 X 32	<i>basename.m.bm</i>	<i>basename.m.pm</i>
대형	48 X 48	<i>basename.l.bm</i>	<i>basename.l.pm</i>

비트맵 파일을 제공하지 않으면 데스크탑이 픽스맵 파일의 색상 사양을 검은색과 흰색으로 대응시킵니다. 그러나, 이러한 대응은 사용자가 원하는 모양을 만들지 못할 수도 있습니다.

아이콘에 관한 자세한 내용은 “아이콘 이미지 파일” 225 페이지를 참조하십시오.

7단계: 응용 프로그램 그룹 작성

주 - 응용 프로그램 그룹을 작성하는 예는 “등록 패키지 작성 예” 92 페이지에서 7단계를 참조하십시오.

응용 프로그램의 활동 및 데이터 유형 정의를 작성했으면, 사용자가 실제로 보게 되는 내용(응용 프로그램 그룹 및 내용)을 작성하는데 대한 책임이 있는 환경설정 파일을 작성해야 합니다.

응용 프로그램 그룹은 응용 프로그램 관리자의 최상위 레벨에 있는 디렉토리입니다. (그림 5-1 참조).

응용 프로그램 그룹은 아래와 같이 3단계로 작성합니다.

- 등록 패키지에 응용 프로그램 그룹 디렉토리를 작성하십시오.
- 선택사항: 응용 프로그램 그룹이 고유한 아이콘을 사용하도록 구성하십시오. 여기에는 응용 프로그램 그룹 디렉토리에 대해 데이터 유형 정의를 작성하는 것이 포함됩니다.
- 응용 프로그램 그룹의 내용을 작성하십시오.

응용 프로그램 그룹 디렉토리의 작성

응용 프로그램 그룹을 작성하려면, 그림 5-7과 같이 `appmanager` 아래의 등록 패키지에 디렉토리를 만드십시오.

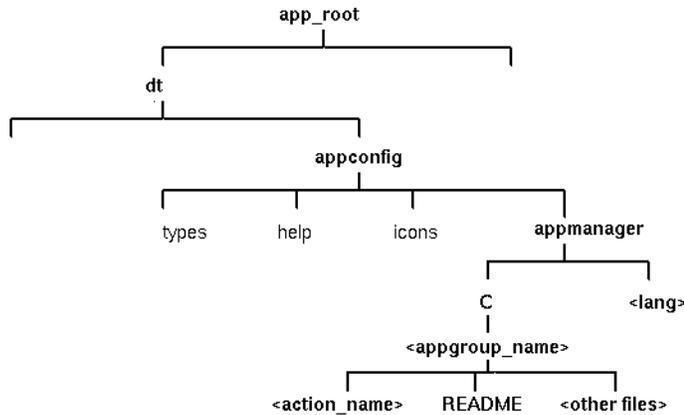


그림 5-7 appmanager 디렉토리

응용 프로그램 그룹 이름

에서 <appgroup_name>은 응용 프로그램 그룹의 이름입니다.

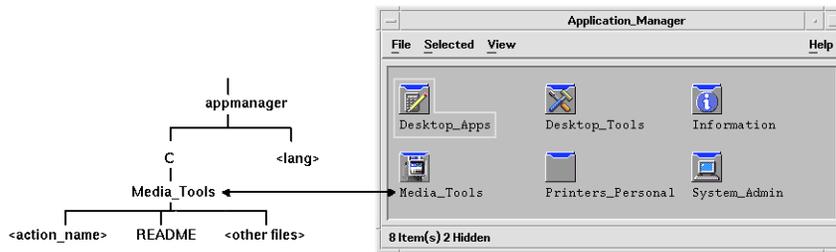


그림 5-8 응용 프로그램 그룹 이름(<appgroup_name>)

이름은 허용 가능한 파일(디렉토리) 이름이면 무엇이든 가능합니다. 응용 프로그램을 설명하는 이름을 사용하십시오.

응용 프로그램 그룹이 고유한 아이콘을 사용하도록 구성하기

데스크탑은 기본 응용 프로그램 그룹 아이콘을 제공합니다. 그러나 사용자는 사용자 정의 아이콘을 만들고자 할 것입니다.

응용 프로그램 그룹에 고유 아이콘을 제공하려면 다음 항목을 작성해야 합니다.

- 응용 프로그램 관리자의 최상위 레벨에 나타나는 디렉토리의 데이터 유형.
- 데이터 유형에 대한 열기 및 인쇄 활동

예를 들어, 사용자가 이름이 Media_Tools인 응용 프로그램 그룹을 만든다고 가정합시다. 파일 `app_root/dt/appconfig/types/language/name.dt`에 위치하고 있는 다음과 같은 데이터 유형 정의는 고유한 아이콘을 응용 프로그램 그룹 아이콘에 할당합니다.

```
DATA_ATTRIBUTES Media_ToolsAppgroup
{
  ACTIONS      OpenInPlace,OpenNewView
  ICON         MediaTools
  DESCRIPTION   Double-click to open the Media_Tools \
application group
}

DATA_CRITERIA      Media_ToolsAppgroupCriteria1
{
  DATA_ATTRIBUTES_NAME Media_ToolsAppgroup
  MODE                 d
  PATH_PATTERN         */appmanager/*/Media_Tools
}
```

정의의 `attributes` 부분에서는 사용할 아이콘을 지정합니다. `criteria` 부분은 `appmanager` 디렉토리의 하위 디렉토리인 `Media_Tools` 디렉토리로 데이터 유형이 제한되도록 지정합니다.

그림 5-9는 응용 프로그램 그룹 이름과 데이터 유형 정의 사이의 관계를 나타내고 있습니다. 데이터 유형 정의에서 `PATH_PATTERN` 필드는 고유한 아이콘을 응용 프로그램 그룹에 연결합니다.

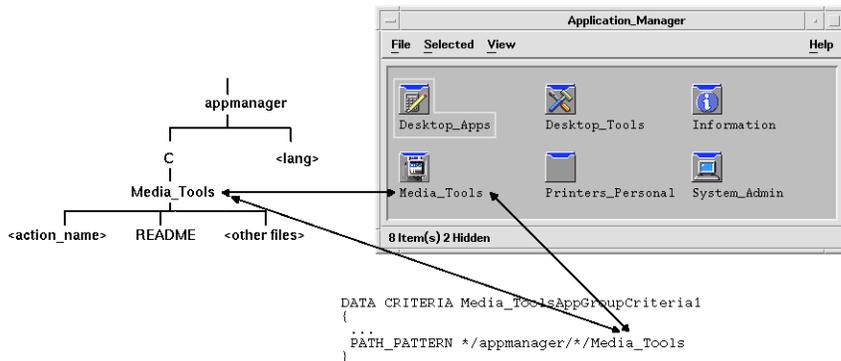


그림 5-9 응용 프로그램이 고유 아이콘을 얻는 방법

사용자는 또한 응용 프로그램 그룹 데이터 유형에 대한 열기 및 인쇄 활동을 작성해야 합니다.

```

ACTION Open
{
    ARG_TYPE           Media_ToolsAppGroup
    TYPE               MAP
    MAP_ACTION         OpenAppGroup
}

ACTION Print
{
    ARG_TYPE           Media_ToolsAppGroup
    TYPE               MAP
    MAP_ACTION         PrintAppGroup
}

```

OpenAppGroup 및 PrintAppGroup 활동은 /usr/dt/appconfig/types/language/dtappman.dt에서 정의하는 내장 활동입니다.

응용 프로그램 그룹 내용 작성

응용 프로그램 그룹에서 가장 중요한 항목은 응용 프로그램을 시작하는 아이콘(활동 아이콘)입니다. 응용 프로그램 그룹이 응용 프로그램 모음을 포함하는 경우, 일반적으로 각 응용 프로그램에 대한 아이콘이 있습니다.

하나 이상의 아이콘 이외에도 응용 프로그램 그룹에는 다음과 같은 것들이 있을 수 있습니다.

- 하나 이상의 README 파일
- 하나 이상의 샘플 데이터 파일
- 템플릿
- 사용자가 두 번 눌러서 도움말 정보를 볼 수 있는 아이콘
- 매뉴얼 페이지
- 특수 프론트 패널 제어 도구

응용 프로그램에는 하위 디렉토리가 있을 수 있습니다.

활동 파일 작성(응용 프로그램 아이콘)

응용 프로그램 그룹에는 응용 프로그램을 시작하는 아이콘이 포함되어야 합니다. 그룹이 응용 프로그램 모음을 제공하는 경우에는 각각에 대한 아이콘이 있어야 합니다. 이 아이콘들은 기본 활동을 나타내므로 **응용 프로그램 아이콘** 또는 **활동 아이콘**이라고 합니다.

활동 아이콘은 활동을 실행할 때 다음과 같이 동일한 이름의 실행 파일을 작성하여 작성할 수 있습니다.

```
app_root/dt/appconfig/appmanager/appgroup_name/action_name
```

파일은 기본 활동의 시각적 표현을 작성하는 목적을 가지고 있기 때문에 **활동 파일**이라고 부릅니다.

예를 들어, 사용자가 BestTextEditor 응용 프로그램을 실행하는 BestTextEditor라는 활동을 작성했을 경우, 이름이 BestTextEditor인 실행 파일을 만들어야 합니다. 파일 관리자와 응용 프로그램 관리자에서는 활동 파일이 활동 정의에 지정된 아이콘 이미지를 사용할 것입니다.

그림 5-10은 응용 프로그램 관리자 윈도우의 활동 정의, 활동 파일 및 실제 항목 사이의 관계를 나타내고 있습니다.

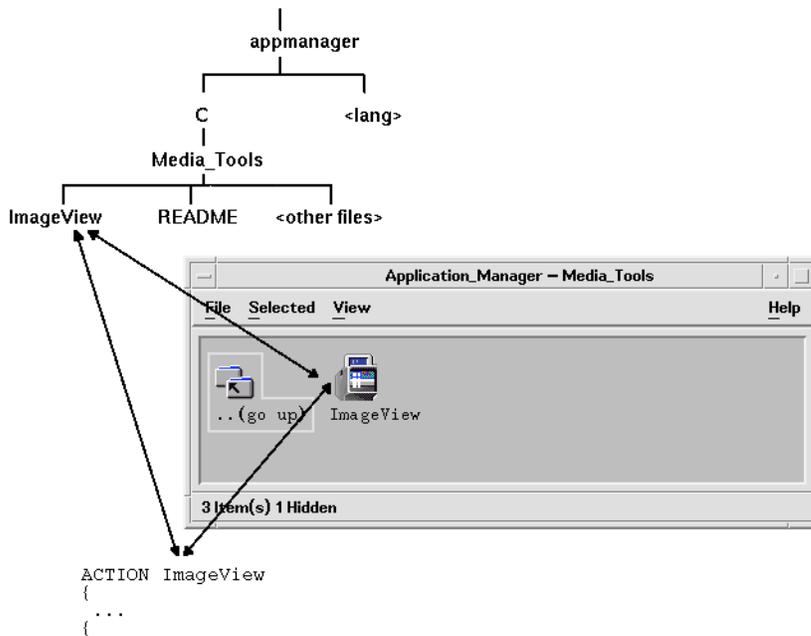


그림 5-10 응용 프로그램 아이콘은 응용 프로그램 그룹에 있는 파일입니다.

Read Me 파일

데스크탑은 README 데이터 유형을 제공하므로 이를 응용 프로그램 README 파일에 사용할 수 있습니다. 다음 이름 지정 규약 중 하나를 사용하십시오.

- README
- readme
- README.*
- Read.*.Me
- read.*.me
- READ.*.ME

특수 프론트 패널 제어 도구 작성

대부분의 경우 사용자가 프론트 패널 제어 도구 정의를 제공할 필요는 없습니다. 활동 아이콘을 서브패널의 아이콘 설치 위에 놓으면 응용 프로그램을 프론트 패널에 추가할 수 있습니다.

제어 도구가 파일을 감시하여 해당 파일이 변경되면 모양이 변하는 경우와 같이, 사용자가 활동 아이콘과 다르게 동작하는 제어 도구를 설치할 수 있게 하려는 경우, 응용 프로그램의 제어 도구 정의가 포함된 프론트 패널 환경설정 파일을 작성하고자 할 수도 있습니다.

프론트 패널 환경설정 파일은 `app_root/dt/appconfig/types/language` 디렉토리에 위치합니다. 이름 지정 규약은 `name.fp`입니다.

제어 도구가 포함된 환경설정 파일이 제공된 경우, 사용자는 *.fp 파일을 서브패널의 아이콘 설치위에 놓음으로써 서브패널에 제어 도구를 추가할 수 있습니다.

예를 들어, 다음 정의가 응용 프로그램 그룹의 프론트 패널 환경설정 파일에 있을 수 있습니다. 사용자가 이 파일을 서브패널의 아이콘 설치 제어 도구 위에 놓으면, 제어 도구는 BestTextEditor 응용 프로그램의 단일 인스턴스를 실행하는 서브패널에 작성됩니다. BestTextEditor가 이미 실행중이면, 윈도우는 현재 작업 공간의 윈도우 스택 맨위로 이동합니다.

```
CONTROL BestTextEditorControl
{
    TYPE                icon
    ICON                BTEFPanel
    PUSH_RECALL        True
    CLIENT_NAME        BTEd
    PUSH_ACTION        BTEditor
    DROP_ACTION        BTEditor
    HELP_STRING        BestTextEditor 응용 프로그램을 시작합니다.
}
```

프론트 패널 환경설정 파일 작성에 관한 자세한 내용은 다음을 참조하십시오.

- 제 15 장
- dtffile(4) 매뉴얼 페이지

8단계: dtappintegrate를 사용한 응용 프로그램 등록

주 - 응용 프로그램 그룹을 작성하는 예는 “등록 패키지 작성 예” 92 페이지에서 8단계를 참조하십시오.

등록 패키지를 응용 프로그램 루트 아래에 작성했으면, 실제 응용 프로그램을 등록할 준비가 된 것입니다.

응용 프로그램을 등록하면 등록 패키지와 데스크탑 검색 경로에 있는 디렉토리들 사이에 링크가 생깁니다("dtappintegrate가 응용 프로그램을 통합하는 방법" 91 페이지 참조).

▼ dtappintegrate로 응용 프로그램을 등록하는 방법

응용 프로그램이 데스크탑 스마트인 경우에는 일반적으로 dtappintegrate이 설치 프로세스의 마지막 단계로 자동 실행됩니다. 자동으로 실행되지 않는 경우나 비데스크탑 스마트 응용 프로그램을 통합하기 위해 환경설정 파일을 작성한 경우, dtappintegrate를 수동으로 실행할 수 있습니다.

1. 루트로 로그인하십시오.
2. 다음 명령어를 실행하십시오.

```
/usr/dt/bin/dtappintegrate -s app_root
```

여기서 *app_root*는 데스크탑 응용 프로그램 루트 디렉토리입니다. 자세한 내용은 dtappintegrate(1) 매뉴얼 페이지를 참조하십시오.

3. 데스크탑_제어의 추가 폴더에 있는 응용프로그램 재로드를 두 번 누르십시오.
4. 응용 프로그램이 적절하게 등록되었는지 확인하십시오.
 - a. 응용 프로그램 관리자의 최상위 레벨을 표시하십시오. 새 응용 프로그램 그룹이 응용 프로그램 관리자에 나타나야 합니다.
 - b. 응용 프로그램 그룹을 열고 활동 아이콘을 두 번 누르십시오.

dtappintegrate의 구문 및 옵션

```
dtappintegrate -s app_root [-t target_path] [-l language] [-u]
```

-s <i>app_root</i>	필수 매개변수. 해당 응용 프로그램이 아래에 설치되어 있는 응용 프로그램 루트를 지정합니다.
-t <i>target_path</i>	선택적 매개변수. 시스템 위치 /etc/dt/appconfig가 기본값입니다. 데스크탑 환경설정 파일이 링크되는 위치를 지정합니다. 응용 프로그램 검색 경로상의 위치를 사용해야 합니다.
-l <i>language</i>	선택적 매개변수. 모든 언어가 기본값입니다. 통합할 언어별 데스크탑 환경설정 파일을 지정합니다.
-u	선택적 매개변수. 응용 프로그램을 분리하여 통합시 설정된 모든 링크를 제거합니다.

dtappintegrate가 응용 프로그램을 통합하는 방법

dtappintegrate의 기능은 설치된 파일과 데스크탑이 환경설정 파일을 찾는 위치 사이의 링크를 설정하는 것입니다.

활동 및 데이터 유형

dtappintegrate는 등록 패키지의 활동 및 데이터 유형 정의 파일에서 활동 데이터베이스 도움말 검색 경로상의 시스템 전체 디렉토리로 기호 링크를 작성합니다. 이 작업은 다음 위치에서

```
app_root/dt/appconfig/types/language/*.dt
```

다음 위치로 링크를 작성하여 수행됩니다.

```
/etc/dt/appconfig/types/language/*.dt
```

도움말 정보 파일

dtappintegrate은 등록 패키지의 도움말 파일에서 도움말 검색 경로상의 시스템 전체 디렉토리로 기호 링크를 작성합니다. 이 작업은 다음 위치에서

```
app_root/dt/appconfig/types/language/help_filesdl
```

다음 위치로 링크를 작성하여 수행됩니다.

```
/etc/dt/appconfig/help/language/help_file.sdl
```

아이콘 파일

dtappintegrate은 등록 패키지의 아이콘 파일에서 아이콘 검색 경로상의 시스템 전체 디렉토리로 기호 링크를 작성합니다. 이 작업은 다음 위치에서

```
app_root/dt/appconfig/icons/language/icon_files
```

다음 위치로 링크를 작성하여 수행됩니다.

```
/etc/dt/appconfig/icons/language/icons_files
```

응용 프로그램 그룹

응용 프로그램에 대한 응용 프로그램 그룹을 응용 프로그램 관리자의 최상위 레벨에 놓기 위해, dtappintegrate은 등록 패키지의 응용 프로그램 그룹 디렉토리와 응용 프로그램 검색 경로상의 시스템 전체 위치 사이에 링크를 작성합니다. 이 작업은 다음 디렉토리에서

```
app_root/dt/appconfig/appmanager/language/appgroup_name
```

다음 디렉토리로 링크를 작성하여 수행됩니다.

등록 패키지 작성 예

다음은 BestTextEditor라는 기존의 비데스크탑 스마트 응용 프로그램의 등록 패키지를 작성하는 단계입니다.

"BestTextEditor"에 관한 필수 정보

예에서는 BestTextEditor 응용 프로그램에 대해 다음과 같은 사항을 가정하고 있습니다.

- 응용 프로그램이 디렉토리 /usr/BTE에 설치되었다고 가정합니다.
- 사용자 세션 언어는 기본값인 C입니다.
- BestTextEditor를 시작하는 명령행은 다음과 같습니다.

```
BTEd {filename}
```

여기서 *filename*은 새 윈도우에서 열리는 데이터 파일의 이름입니다. BestTextEditor는 자신의 윈도우를 작성합니다—즉, 터미널 에뮬레이터 윈도우내에서는 BestTextEditor가 실행되지 않습니다.

- BestTextEditor는 두 가지 유형의 데이터 파일을 작성하여 사용합니다.
 - 설명서 파일. 이름 지정 규약 *.bte를 사용합니다. BestTextEditor는 .bte 데이터 파일을 인쇄하기 위한 명령행을 제공합니다. 이 명령어의 구문은 다음과 같습니다.

```
BTEPrint [-d destination] [-s] filename
```

여기서

-d *destination*—대상 프린터를 지정합니다.

-s—무응답 인쇄를 지정합니다. 응용 프로그램의 인쇄 대화 상자가 표시되지 않습니다.

파일 이름 —인쇄할 파일을 지정합니다.

- 템플릿 파일. 이름 지정 규약은 *.tpl을 사용합니다. 템플릿 파일은 인쇄할 수 없습니다.
- BestTextEditor의 경우, 기존의 비데스크탑 app-defaults 파일은 인터페이스 글꼴 및 전경색과 배경색에 대한 자원을 포함합니다.
- BestTextEditor의 온라인 도움말 볼륨은 데스크탑 도움말 개발자 키트를 사용하여 작성되었습니다. 온라인 도움말 볼륨을 구축할 때 다음 소스 파일을 사용했고

```
.../BTEHelp.htg
.../graphics/BTE1.xwd
.../graphics/BTE2.xwd
```

파일 .../BTEHelp.sdl을 작성했습니다.

"BestTextEditor"를 등록하는 단계

다음 단계별 절차에 따라 BestTextEditor를 등록하십시오.

1. 글꼴 및 색상 자원을 수정하십시오.

BestTextEditor의 app-defaults 파일에서 다음을 설정하는 자원을 제거하십시오.

- 텍스트 글꼴
- 전경색 및 배경색

2. 응용 프로그램 루트를 작성하십시오.

다음 디렉토리를 작성하십시오.

```
/desktop_approots/BTE
```

기존 응용 프로그램을 통합하려면 응용 프로그램 루트 디렉토리를 응용 프로그램의 설치 위치 이외의 다른 곳에 만들어야 합니다. 그렇지 않으면 응용 프로그램을 갱신할 때 사용자가 작성하는 환경설정 파일이 제거될 수도 있습니다.

3. 등록 패키지 디렉토리를 만드십시오.

다음 디렉토리를 작성하십시오.

```
/desktop_approots/BTE/dt/appconfig/types/C
/desktop_approots/BTE/dt/appconfig/help/C
/desktop_approots/BTE/dt/appconfig/icons/C
/desktop_approots/BTE/dt/appconfig/appmanager/C/BestTextEditor
```

4. 응용 프로그램의 활동 및 데이터 유형을 작성하십시오.

a. 활동 및 데이터 유형 정의에 대한 환경설정 파일을 작성하십시오.

```
/desktop_approots/BTE/dt/appconfig/types/C/BTE.dt
```

b. BestTextEditor를 실행하는 활동 정의를 작성하십시오.

```
ACTION BTEditor
{
    WINDOW_TYPE          NO_STDIO
    ICON                  BTERun
    DESCRIPTION          Double-click this icon or drop \
                        a BTE data file on it to run \
                        BestTextEditor.
    EXEC_STRING          /usr/BTE/BTEd %Arg_1%
}
```

c. *.bte 파일에 대한 데이터 유형을 작성하십시오.

```

DATA_ATTRIBUTES BTEDataFile
{
    DESCRIPTION          BestTextEditor data file.
    ICON                 BTEData
    ACTIONS              Open,Print
}

```

```

DATA_CRITERIA BTEDataFileCriteria1
{
    DATA_ATTRIBUTES_NAME BTEDataFile
    NAME_PATTERN          *.bte
    MODE                  f
}

```

d. *.tpl 파일에 대한 데이터 유형을 작성하십시오.

```

DATA_ATTRIBUTES BTETemplateFile
{
    DESCRIPTION          BestTextEditor template file.
    ICON                 BTETempl
    ACTIONS              Open
}

```

```

DATA_CRITERIAL BTETemplateFileCriteria1
{
    DATA_ATTRIBUTES_NAME BTETemplateFile
    NAME_PATTERN          *.tpl
    MODE                  f
}

```

e. *.bte 파일에 대한 열기 활동을 작성하십시오.

```

ACTION Open
{
    ARG_TYPE             BTEDataFile
    TYPE                 MAP
    MAP_ACTION           BTEditor
}

```

f. *.bte 파일에 대한 인쇄 활동을 작성하십시오.

다음은 데이터 파일을 인쇄하는 간단한 인쇄 활동입니다. 이 활동에는 LPDEST 환경 변수 값이 필요하며 -s 인쇄 옵션은 무시합니다. (LPDEST가 설정되어 있지 않으면 활동이 실패할 것입니다.)

```

ACTION Print
{
    ARG_TYPE             BTEDataFile
    TYPE                 MAP
    MAP_ACTION           BTEPrintData
}

ACTION BTEPrintData
{
    WINDOW_TYPE          NO_STDIO
    EXEC_STRING           BTEPrint -d $LPDEST %Arg_1%
}

```

```
}
```

다음은 BTEPrintData 활동의 다른 버전과 그에 수반되는 스크립트입니다. 이 둘은 함께 LPDEST가 설정되어 있지 않거나 무응답 인쇄가 요청된 상황을 다룹니다.

```
ACTION BTEPrintData
{
    WINDOW_TYPE          NO_STUDIO
    EXEC_STRING           /usr/BTE/bin/BTEenvprint \
                        %(File)Arg_1%
}

```

/usr/BTE/bin/BTEenvprint 스크립트 내용은 다음과 같습니다.

```
# BTEenvprint
#!/bin/sh
DEST=""
SILENT=""
if [ $LPDEST ] ; then
    DEST="-d $LPDEST"
fi
BTEPrint $DEST SILENT $1

```

g. *.tpl 파일에 대한 열기 활동을 작성하십시오.

```
ACTION Open
{
    ARG_TYPE          BTETemplateFile
    TYPE              MAP
    MAP_ACTION        BTEditor
}

```

h. *.bte 파일에 대한 인쇄 활동을 작성하십시오.

```
ACTION Print
{
    ARG_TYPES         BTETemplateFile
    TYPE              MAP
    MAP_ACTION        NoPrint
}

```

NoPrint는 내장된 활동으로서 사용자에게 파일을 인쇄할 수 없음을 알려주는 대화 상자를 표시합니다.

5. 도움말 파일을 등록 패키지에 두십시오.

a. 도움말 파일을 다음 위치에 두십시오.

```
/desktop_approots/BTE/dt/appconfig/help/C/BTEhelp.sdl
/desktop_approots/BTE/dt/appconfig/help/C/graphics/BTE1.xwd
/desktop_approots/BTE/dt/appconfig/help/C/graphics/BTE2.xwd

```

b. 다음 파일을 작성하십시오.

```
/desktop_approots/BTE/dt/appconfig/types/C/BTEhelp.dt.
파일에 다음과 같은 활동 정의를 두십시오.
```

```

ACTION BTEHelp
{
    WINDOW_TYPE          NO_STDIO
    EXEC_STRING          /usr/dt/bin/dthelpview -helpVolume \
                        BTEHelp.sdl
    DESCRIPTION          Opens the BestTextEditor help volume.
}

```

6. 응용 프로그램에 대한 아이콘을 작성하십시오.

아이콘 편집기를 사용하여 아이콘을 만드십시오. 표 5-3은 사용할 크기에 대한 지침을 나타내고 있습니다.

표 5-3 아이콘 크기 지침

이름	크기
<i>basename.t.pm</i>	16 x 16
<i>basename.m.pm</i>	32 x 32
<i>basename.l.pm</i>	48 x 48

다음 아이콘 파일을 디렉토리에 작성하십시오.

`/desktop_approots/BTE/dt/appconfig/icons/C:`

- 응용 프로그램을 실행하는 활동을 표시하는 아이콘: `BTERun.t.pm`, `BTERun.m.pm`, `BTERun.l.pm`
- *.bte 파일을 표시하는 아이콘: `BTEData.t.pm`, `BTEData.m.pm`,
- *.tpl 파일을 표시하는 아이콘: `BTETempl.t.pm`, `BTETempl.m.pm`
- (7단계에서 사용하는) 응용 프로그램 그룹을 표시하는 아이콘: `BTEApp.t.pm`, `BTEApp.m.pm`

7. 응용 프로그램 그룹을 작성하십시오.

a. 디렉토리를 아직 만들지 않았으면 다음과 같이 만드십시오.

`/desktop_approots/BTE/dt/appconfig/appmanager/C/BestTextEditor`

- b. 이 단계는 선택적입니다. 여기서는 응용 프로그램 그룹에 대해 데이터 유형 및 관련 활동을 작성함으로써 응용 프로그램 그룹 아이콘에 대한 고유한 아이콘을 제공합니다. 이 단계를 생략하면 응용 프로그램 그룹은 기본 아이콘을 사용할 것입니다.

다음 데이터 유형 및 활동 정의를 `/desktop_approots/BTE/dt/appconfig/types/C/BTE.dt` 파일에 추가하십시오. 데이터 유형은 `BestTextEditor` 응용 프로그램 그룹이 사용할 아이콘을 지정합니다. 이 활동들은 내장 응용 프로그램 그룹과 동일한 열기 및 인쇄 동작을 제공합니다.

```

DATA_ATTRIBUTES BestTextEditorAppGroup
{
    ACTIONS          OpenInPlace, OpenNewView
    ICON             BTEApp
}

```

```

}
DATA_CRITERIA BestTextEditorAppGroupCriteria1
{
    DATA_ATTRIBUTES_NAME      BestTextEditorAppGroup
    MODE                       d
    PATH_PATTERN                */appmanager/*/BestTextEditor
}
ACTION Open
{
    ARG_TYPE                   BestTextEditorAppGroup
    TYPE                       MAP
    MAP_ACTION                 OpenAppGroup
}
ACTION Print
{
    ARG_TYPE                   BestTextEditorAppGroup
    TYPE                       MAP
    MAP_ACTION                 PrintAppGroup
}

```

- c. 응용 프로그램 그룹에 응용 프로그램을 시작할 아이콘을 만드십시오. 이를 위해서 다음 파일을 작성하고

```

        /desktop_approots/BTE/dt/appconfig/appmanager/C \
        /BestTextEditor/BTEditor

```

파일을 실행할 수 있도록 만드십시오.

- d. 응용 프로그램 그룹에 도움말 볼륨을 여는 활동 파일을 작성하십시오. 이를 위해서 다음 파일을 작성하고

```

        /desktop_approots/BTE/dt/appconfig/appmanager/C \
        /BestTextEditor/BTEditor

```

파일을 실행할 수 있도록 만드십시오.

- e. "read me" 파일, 샘플 데이터 및 템플릿 파일들을 응용 프로그램 그룹에 두십시오.

8. 응용 프로그램을 등록하십시오.

터미널 에뮬레이터 윈도우에서

- a. 루트로 로그인하십시오.

- b. 다음 명령어를 실행하십시오.

```

/usr/dt/bin/dtappintegrate -s /desktop_approots/BTE

```

- c. 데스크탑_제어기의 추가 폴더에 있는 응용프로그램 재로드를 두 번 누르십시오.

기타 구성

이 장에서는 설치 및 시스템 관리 주제에 대해 설명합니다.

- “Solaris CDE 디렉토리 구조” 99 페이지
- “키 환경설정 파일” 101 페이지
- “로그인 서버 시작” 102 페이지
- “설치된 CDE를 다른 워크스테이션이나 네트워크 서버 설치 위치로부터 마운트하기” 104 페이지
- “데스크탑이 복수 화면을 사용하도록 구성” 105 페이지
- “네트워크화된 데스크탑” 106 페이지
- “XTerminals 사용” 108 페이지
- “로그인 로컬 및 글꼴 경로” 109 페이지
- “워크스테이션을 XTerminals로 사용” 109 페이지
- “특수 CDE 구성” 110 페이지

Solaris CDE 디렉토리 구조

이 절에서는 사용자 데스크탑 환경에 포함된 기본 디렉토리에 대해 설명합니다.

`/usr/dt`

이 디렉토리는 Solaris CDE 설치 위치입니다. 또한 원격 파일 서버의 마운트 지점일 수도 있습니다. 표 6-1은 `/usr/dt` 하위 디렉토리를 나타냅니다.

표 6-1 /usr/dt 하위 디렉토리

하위 디렉토리	설명
/bin	SCDE 응용 프로그램 및 유틸리티
/lib	SCDE 실행 시간 공유 라이브러리
/config	기본 시스템 환경설정 파일
/man	[선택사항] 매뉴얼 페이지
/app-defaults	기본 응용 프로그램 자원
/appconfig	기본 응용 프로그램 아이콘, 유형 및 활동
/examples	[선택사항] CDE 코드/프로그램 예
/include	[선택사항] 개발자 포함 파일
/palettes	색상 팔레트
/share	CDE AnswerBook 설명서 및 기본 배경

/etc/dt

이 디렉토리에는 사용자 정의된 특정 작업 공간 환경설정 파일이 있습니다. 이 파일을 이용하여 다음과 같은 방법으로 환경을 사용자 정의할 수 있습니다.

- X 서버 환경설정 옵션을 설정하십시오.
- 복수 화면을 사용하십시오.
- 워크스테이션의 활동 파일, 데이터 유형, 아이콘 및 글꼴을 사용자 정의하십시오.

/var/dt

이 디렉토리는 로그인 관리자와 응용 프로그램 관리자 같은 Solaris CDE 응용 프로그램의 임시 파일을 저장하는데 사용됩니다.

\$HomeDirectory

이 디렉토리에는 사용자 데스크탑 설정과 관련된 특정 사용자 파일이 있습니다. 이러한 파일에는 응용 프로그램, 색상 체계, 작업 공간 메뉴 및 프론트 패널 수정, 오류 로그 등이 포함됩니다.

키 환경설정 파일

데스크탑 환경의 사용자 정의 대부분이 여러 파일과 관련되어 있습니다. 다음 파일은 두 가지 예입니다.

- /usr/dt/config/Xconfig
- /usr/dt/config/Xservers

Xconfig

Xconfig는 dtlogin이 사용하는 마스터 환경설정 파일입니다. 이 파일로 로그인 자원을 설정하고 dtlogin이 요구하는 추가 파일의 위치를 지정합니다. 다음 예는 기본 Xconfig 파일에 포함된 몇 개의 행을 나타내고 있습니다.

```
Dtlogin.errorLogFile:      /var/dt/Xerrors
Dtlogin.servers:           /usr/dt/config/Xservers
Dtlogin*session:          /usr/dt/bin/Xsession
```

이 파일에 대한 자세한 내용은 부록 A를 참조하십시오. 파일 자체에 있는 설명 주석을 참조할 수도 있습니다.

Xservers

dtlogin은 XDM(X 표시장치 관리자)과 마찬가지로, Xservers 파일을 사용하여 지역 Xserver의 시작 방법을 지정합니다. 기본적으로 이 파일의 마지막 행은 다음과 같습니다.

```
:0 Local local@console /usr/openwin/bin/Xsun :0
```

- :0은 Xserver 표시장치가 <localhost:0>임을 의미합니다.
- local은 지역적으로 새로운 X11 서버를 시작할 것을 나타냅니다.

주 - 실행 중인 X11 서버에 접속하려면 local을 foreign으로 바꾸십시오.

- console은 명령행 로그인 제어 문자가 /dev/console로 향함을 의미합니다.
- /usr/openwin/bin/Xsun은 X11 서버에 대한 경로입니다.

주 - Solaris CDE와 OpenWindows™ 환경은 동일한 X11 서버를 실행합니다.

이 파일에 대한 자세한 정보는 dtlogin(1) 매뉴얼 페이지를 참조하십시오. 파일 자체에 있는 설명 주석을 참조할 수도 있습니다.

로그인 서버 시작

로그인 서버는 일반적으로 시스템이 부트될 때 자동으로 시작됩니다. 명령행에서 로그인 서버를 시작할 수도 있지만 먼저 루트 사용자로 로그인해야 합니다.

시스템이 부트될 때 로그인 서버가 시작되도록 설정하려면 다음과 같이 하십시오.

- 다음 명령어를 입력하고 Return을 누르십시오.

```
# /usr/dt/bin/dtconfig -e
```

이렇게 하면 S99dtlogin 파일이 사용자의 /etc/rc2.d 디렉토리에 추가됩니다. 로그인 서버는 재부트하면 자동으로 시작됩니다.

시스템을 부트할 때 로그인 서버가 자동으로 시작되지 않도록 하려면 다음과 같이 하십시오.

- 다음 명령어를 입력하고 Return을 누르십시오.

```
# /usr/dt/bin/dtconfig -d
```

명령행에서 로그인 서버를 시작하려면,

- 다음 명령행을 입력하고 Return을 누르십시오.

```
# /usr/dt/bin/dtlogin -daemon; exit
```

주 - 임시 환경설정 시험을 위해 명령행에서 로그인 서버를 시작할 수는 있지만, 로그인 서버는 시스템을 부트할 때 정상적으로 시작해야 합니다.

로그인 서버, Xserver 및 전체 Solaris CDE 데스크탑을 강제 종료하려면 다음과 같이 하십시오.

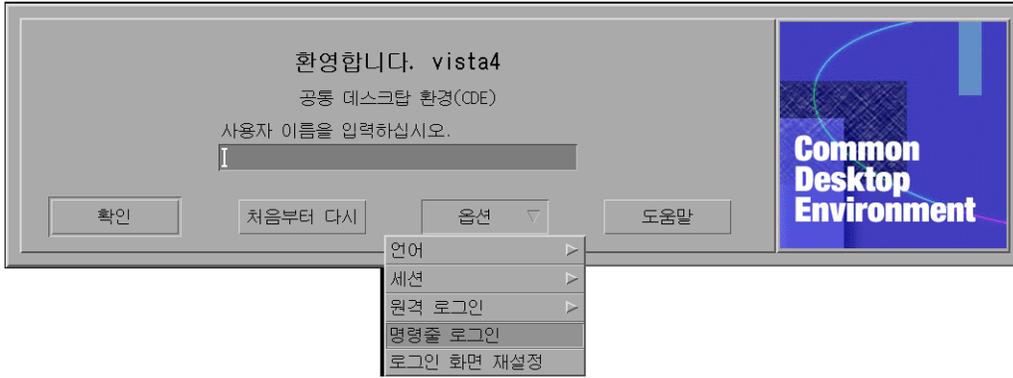
- 다음 명령어를 입력하고 Return을 누르십시오.

```
# /usr/dt/bin/dtconfig -kill
```

데스크탑 구성 유틸리티, dtconfig에 대한 자세한 정보는 부록 A를 참조하십시오. dtconfig.1 매뉴얼 페이지의 사본이 수록되어 있습니다.

명령행 로그인 옵션을 사용하여 Solaris CDE 로그인 화면을 종료하려면 다음과 같이 하십시오.

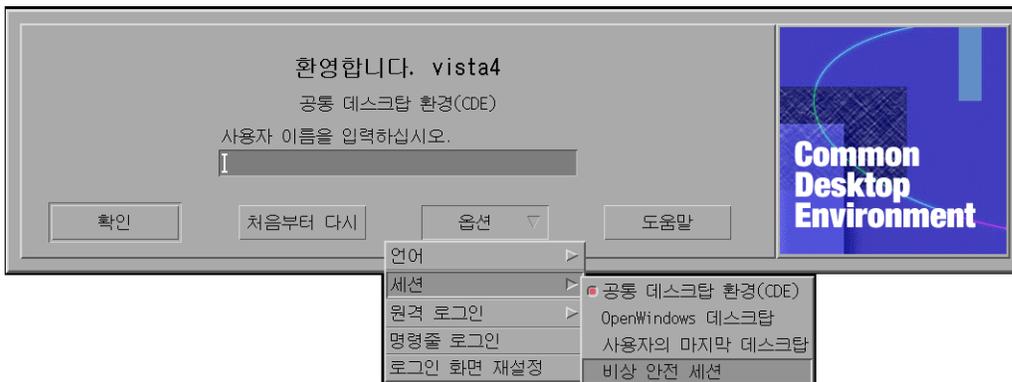
- Solaris CDE 로그인 화면의 옵션 메뉴에서 명령행 로그인을 선택하십시오. 화면이 없거나 Return 키를 눌러 로그인 프롬프트를 표시하십시오.



주 - 데스크탑 로그인 데몬은 사용자가 명령행 로그인 사용을 마치면 데스크탑 로그인 화면을 자동으로 시작할 것입니다.

단일 터미널 애플리케이션 윈도우를 시작하는 방법

- Solaris CDE 로그인 화면의 옵션 메뉴 중 세션 하위 메뉴에서 비상 안전 세션을 선택하십시오.



Xserver가 실행되도록 하려면, 세션 하위 메뉴에서 비상 안전 세션을 선택하십시오. 그러면 단일 xterm 윈도우가 시작됩니다. 비상 안전세션 옵션은 명령행 로그인 옵션을 사용할 수 없는 경우에도 언제나 사용가능합니다.

설치된 CDE를 다른 워크스테이션이나 네트워크 서버 설치 위치로부터 마운트하기

이 절차는 사용자의 지역 디스크 용량을 사용하지 않으므로, 사용자 워크스테이션에 필요한 디스크 용량이 없는 경우에 설치된 CDE를 다른 워크스테이션이나 네트워크 서버 설치 위치로부터 마운트할 수 있습니다.

주 - /usr/dt 디렉토리 구조가 Solaris 릴리스와 클라이언트 워크스테이션에서는 다르므로(SPARC 디렉토리 구조는 Intel 디렉토리 구조와 다름), 클라이언트 워크스테이션은 적합한 NFS 서버 /usr/dt 이미지를 마운트해야 합니다. 예를 들어, Solaris 2.5를 사용하는 SPARC 시스템은 CDE가 설치된 /usr/dt가 있는 다른 Solaris 2.5 시스템에서 /usr/dt를 마운트해야 합니다.

▼ 설치된 CDE 마운트 방법

1. /usr/dt 디렉토리를 이미 설치된 워크스테이션이나 네트워크 서버에서 사용자 워크스테이션의 /usr/dt 디렉토리로 마운트하십시오.
2. /usr/dt/bin/dtconfig -inetd 를 입력하십시오.
3. 다음을 입력하여 Solaris 데스크탑 로그인을 사용가능하게 하십시오.

```
/usr/dt/bin/dtconfig -e
```

4. 워크스테이션을 재부트하십시오.

데스크탑 구성 유틸리티, dtconfig에 대한 자세한 정보는 부록 A를 참조하십시오. dtconfig.1 매뉴얼 페이지의 사본이 수록되어 있습니다.

▼ 마운트된 CDE 디렉토리를 마운트 해제하는 방법

1. 다음을 입력하여 Solaris 데스크탑 로그인을 사용불가능하게 하십시오.

```
/usr/dt/bin/dtconfig -d
```

2. /usr/dt/bin/dtconfig -inetd.ow 를 입력하십시오.

3. /usr/dt를 마운트 해제하십시오.

4. 워크스테이션을 재부트하십시오.

데스크탑 구성 유틸리티, dtconfig에 대한 자세한 정보는 부록 A를 참조하십시오. dtconfig.1 매뉴얼 페이지의 사본이 수록되어 있습니다.

데스크탑이 복수 화면을 사용하도록 구성

표준 로그인은 데스크탑을 하나의 화면에서 시작합니다. Xconfig 파일을 편집하면 데스크탑이 복수 화면에서 시작하도록 할 수 있습니다. 이 파일을 변경하려면 루트 사용자로 로그인해야 합니다.

주 - 로그인 구성 정보를 보다 빨리 편집하려면 비상 안전 세션 옵션을 사용하여 전체 데스크탑이 아닌 단일 xterm 윈도우를 실행하십시오.

▼ 데스크탑이 복수 화면에서 시작되도록 하는 방법

1. 다음 명령어를 입력하고 **Return**을 눌러 Xserver 파일 사본을 작성하십시오.

```
# cp /usr/dt/config/Xservers /etc/dt/config/Xservers
```

주 - /etc/dt/config/Xservers overrides /usr/dt/config/Xservers.

2. /etc/dt/config/Xservers 파일을 편집하여 두 개의 프레임 버퍼(화면)를 설치하십시오.

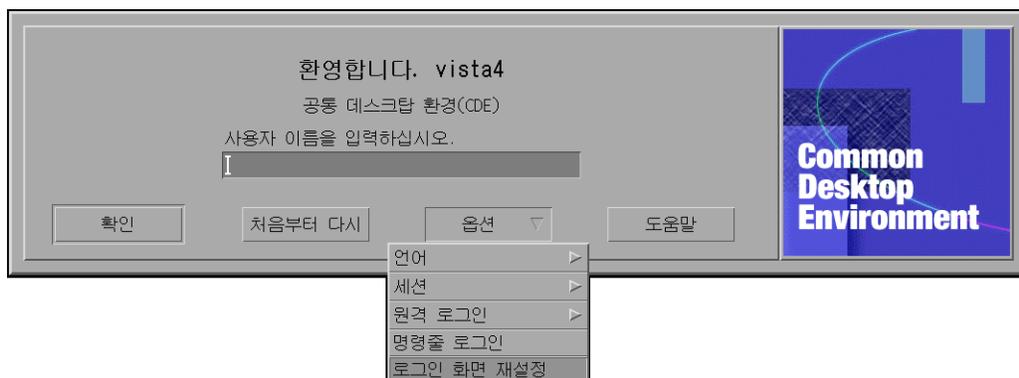
- a. 다음 행을 찾으십시오.

```
:0 Local local_uid@console root /usr/openwin/bin/Xsun :0
```

- b. 이 행의 끝에 두 -dev 옵션을 추가하십시오. 이 예에서는 이 옵션의 이름이 /dev/fb0와 /dev/fb1이라고 가정합니다. 공백을 한 칸 띄우고 다음을 추가하십시오.

```
-dev /dev/fb0 -dev /dev/fb1
```

3. 옵션 메뉴에서 로그인 화면 재설정을 선택하십시오.



주 - Solaris CDE를 복수 화면에서 실행할 때, 프론트 패널은 각 화면에 표시됩니다. 이 프론트 패널들은 상호 독립적이며 별도의 환경설정 파일을 갖고 있습니다.

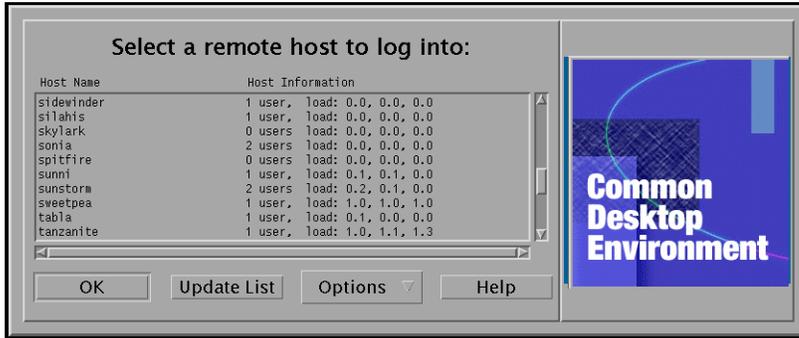
네트워크화된 데스크탑

Solaris CDE 로그인 관리자는 네트워크를 인식하고 있습니다. 기본적으로 로그인 화면은 Solaris CDE 선택기의 조화에 응답합니다.

원격 호스트 목록을 표시하려면

- Solaris CDE 로그인 화면의 원격 로그인 메뉴 중 목록에서 호스트 선택을 선택하십시오.

또한, 선택기를 로그인 화면과 무관하게 실행하려면 “워크스테이션을 XTerminals로 사용” 109 페이지를 참조하십시오. 다음 화면은 사용 가능한 서버의 일반적인 선택기 목록을 나타내고 있습니다.



목록에서 유틸 서버를 선택하고 선택기를 통해 이 서버의 자원을 사용하여 Solaris CDE 에 로그인할 수 있습니다.

Solaris CDE 로그인 관리자를 실행하는 모든 네트워크화된 워크스테이션은 워크스테이션을 Solaris CDE 데스크탑 서버로 사용하는 Xterminal 사용자를 비롯하여, 복수 원격 사용자와 지역 데스크탑 사용자 모두를 지원할 수 있습니다. (그림 6-1을 참조하십시오).

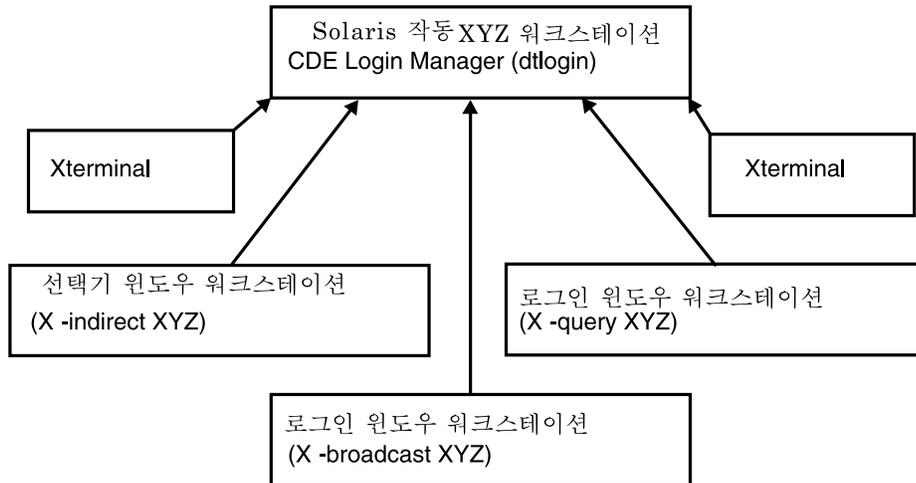


그림 6-1 네트워크화된 데스크탑

주 - 그림 6-1에는 혼합된 하드웨어 환경을 포함할 수 있습니다.

XTerminals 사용

XDM 프로토콜을 지원하는 Xterminal은 Solaris CDE에 로그인하기 위해 선택기를 사용할 수 있습니다. 선택기 실행에 관한 내용은 “워크스테이션을 XTerminals로 사용” 109 페이지를 참조하십시오. SPARC Xterminal™ 소프트웨어 버전 2.0(또는 그 이후 버전)은 Solaris CDE와 함께 잘 작동합니다. X-Terminal 제어 도구 윈도우에서 특성을 누르고 범주를 세션으로 선택하여 그림 6-2와 같은 설정 화면을 표시하십시오.

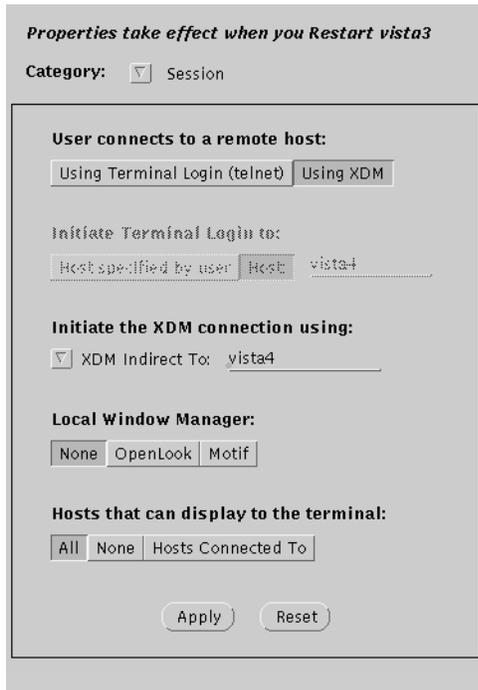
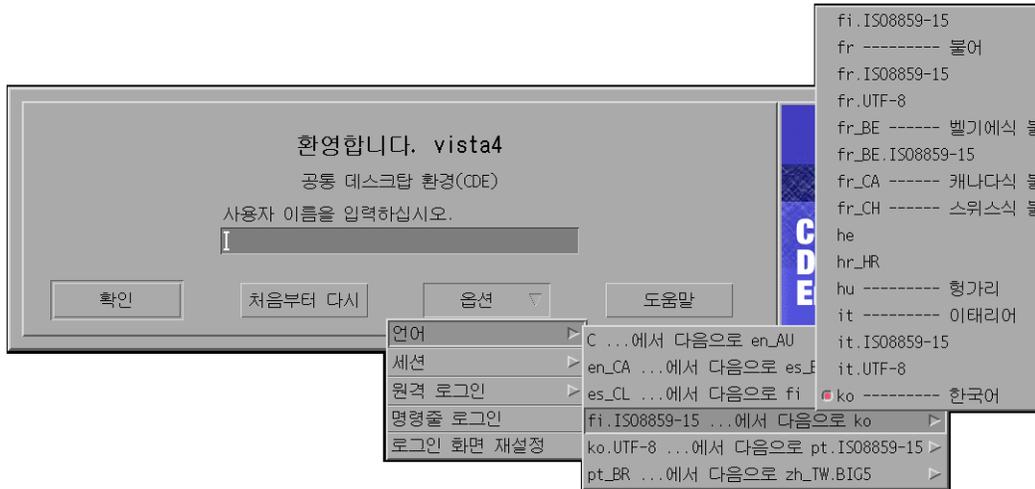


그림 6-2 세션 설치 화면

지역 윈도우 관리자는 없음으로 선택하고, Solaris CDE를 실행하는 원격 시스템에 대한 XDM 연결은 XDM 간접을 선택하십시오.

로그인 로케일 및 글꼴 경로

사용자는 Solaris CDE에 로그인할 때 원하는 언어를 선택합니다. 다음 Solaris CDE 로그인 화면에서는 옵션 메뉴에 있는 언어 선택항목을 보여주고 있습니다.



워크스테이션에서 글꼴 경로는 원하는 언어와 시스템의 기본 "C"로케일 모두와 연관된 글꼴(및 별명)을 포함하도록 자동으로 설정됩니다. Xterminals의 경우, 이 글꼴은 X11 글꼴 서버가 자동으로 제공합니다.

워크스테이션을 XTerminals로 사용

이전의 워크스테이션 또는 16메가바이트 이하의 메모리를 가진 워크스테이션이 있는 경우, 이 워크스테이션에 Solaris 2.4 이상의 소프트웨어(또는 Xserver 점보 패치가 있는 Solaris 2.3 소프트웨어)를 로드하면 이를 Xterminals로 사용할 수 있습니다.

주 - 아래의 절차를 따를 때 워크스테이션의 화면 글꼴이 정상적으로 표시되지 않으면, Solaris CDE는 호스트 워크스테이션에 설치되지 않을 수도 있습니다. pkgadd 유틸리티를 사용하여 Solaris CDE 1.0.x 버전들 릴리스에서 별도로 SUNWdftf () Solaris CDE 글꼴 패키지를 추가하십시오.

▼ 호스트 CDE 로그인 선택을 위해 선택기를 사용하는 방법

1. 실행중인 모든 윈도우 시스템을 종료하십시오.
2. 선택기를 시작하십시오. 콘솔 명령행에서 (csh)을 사용하여 다음 명령어를 입력하십시오.

```
setenv OPENWINHOME /usr/openwin  
  
/usr/openwin/bin/X -indirect CDE_login_host
```

Xserver가 호스트로부터 선택기 윈도우를 시작하고 표시합니다.

3. 선택기 윈도우에서 원하는 로그인 호스트를 선택하십시오.

▼ 특정 호스트 CDE 로그인을 사용하는 방법

- 다음 명령어를 입력하고 Return을 누르십시오.

```
/usr/openwin/bin/X -once -query CDE_login_host
```

Xserver가 호스트로부터 로그인 화면을 시작하고 표시합니다. -once 옵션은 로그인/로그아웃 세션이 한 번 끝나면 서버를 종료할 것입니다. -once 옵션을 추가하지 않으면, Solaris CDE 로그아웃 후에 로그인 화면이 다시 나타납니다.

▼ 첫번째 사용가능한 호스트 로그인을 사용하는 방법

- 다음 명령어를 입력하고 Return을 누르십시오.

```
/usr/openwin/bin/X -broadcast
```

이렇게 하면 Xserver가 시작됩니다. 이것은 지역 서브네트에서 XDM(X 표시장치 관리자) 로그인 서비스 요청을 브로드캐스트합니다. 서브네트상에 있는 시스템들이 Solaris CDE 로그인 관리자(또는 다른 XDM 기반 로그인 윈도우)를 실행하고 있는 중이라면, 응답하는 최초의 호스트가 해당 로그인 윈도우를 사용자의 데스크탑에 위치시킵니다.

특수 CDE 구성

이 절은 특수 구성에 대해 설명합니다.

우편 인쇄 사용자 정의

CDE 우편 시스템은 DTMAIL_FILE 데이터 유형에 정의된 인쇄 활동을 사용하여 메시지를 인쇄합니다. OpenWindows 우편 도구가 지정하는 인쇄 스크립트를 사용하지 않습니다. 인쇄 동작을 변경하려면 이 인쇄 활동을 수정해야 합니다.

▼ 인쇄 활동을 수정하는 방법

1. 원하는 편집기를 사용하여 다음 파일을 작성하십시오.

```
HomeDirectory/.dt/types/dtmail.dt
```

2. 다음 명령행을 이 파일에 입력하십시오.

```
#
# Override default Print action for mailboxes
#
ACTION Print
{
LABEL          Print
ARG_TYPE       DTMAIL_FILE
TYPE           COMMAND
WINDOW_TYPE    NO_STDIO
EXEC_STRING    sh -c ' \
                dtmailpr -p -f %(File)Arg_1% | mp -m -1 | \
                dtlp -u %(File)Arg_1%;'
}
```

주 - dtmailpr는 인쇄 필터로서, 첨부 파일을 떼어내고 일련의 텍스트만 남게 합니다. dtlp는 lp에 대한 표준 CDE 인터페이스입니다. %(File)Arg_1%은 인쇄중인 파일입니다.

3. 사용자가 원하는 인쇄 명령어를 포함하도록 EXEC_STRING을 수정하십시오.
4. 우편 시스템을 다시 시작하십시오.

캘린더를 새 데이터 형식으로 전환

버전 4 확장 가능 데이터 형식은 CDE 캘린더가 지원하는 새로운 형식입니다. OpenWindows 캘린더 관리자는 이 형식을 읽을 수 없습니다. OpenWindows와 CDE 플랫폼 사이에서 앞뒤로 전환할 필요가 있는 경우에는 사용자의 캘린더를 버전 4 데이터 형식으로 변환해서는 안 됩니다. 그러나 캘린더의 불필요한 부분을 없애기 위해 `sdtcm_convert` 스크립트를 사용할 수는 있습니다.

주 - 캘린더의 도움말 메뉴에서 캘린더 정보를 선택하면 사용자 캘린더의 데이터 버전을 알 수 있습니다.

그렇지 않으면, `sdtcm_convert` 스크립트를 사용하여 캘린더를 버전 4의 데이터 형식으로 변환하십시오.

이 파일에 대한 자세한 정보는 `sdtcm_convert(1)` 매뉴얼 페이지를 참조하십시오.

주 - 최종 사용자 CDE 패키지만을 설치한 경우, 매뉴얼 페이지는 설치되지 않습니다. 매뉴얼 페이지를 보려면, `pkgadd` 유틸리티를 사용하여 설치해야 합니다.

네트워크에서 AnswerBook 패키지 추가

Solaris CDE를 설치할 때, 설치 메뉴는 AnswerBook CDE 패키지를 설치할 수 있는 옵션을 제공합니다. YES를 선택하면, AnswerBook 패키지가 `/usr/dt/share/answerbooks/language` 디렉토리(`language`는 특정 AnswerBook 번역본의 지역명임)에 설치됩니다.

AnswerBook 패키지를 설치하려면 109메가바이트의 디스크 공간이 필요하므로 사용자가 설치하기를 원하지 않을 수도 있습니다. 그러나, 다음 두 옵션을 사용할 수 있습니다.

- AnswerBook 패키지를 다음에 마운트하거나 `/net/. . .`로부터 링크할 수 있습니다. `/usr/dt/share/answerbooks/language`
- 사용자는 `AB_CARDCATALOG` 환경 변수를 `HomeDirectory/.dtprofile`에 추가할 수 있습니다. 예를 들어, 내보낸 버전이 네트워크상에 있는 경우에는 다음 행을 `HomeDirectory/.dtprofile`에 추가하십시오.

```
export AB_CARDCATALOG=/net/hostname/usr/dt/share/answerbooks \
/language/ab_cardcatalog
```

CDE 데스크탑 외부에서 CDE 환경 설정하기

Solaris CDE 응용 프로그램이 사용하는 환경 변수는 CDE 데스크탑 외부에서 설정할 수 있습니다. 예를 들어, 사용자는 원격 워크스테이션에 로그인하고 CDE 응용 프로그램을 사용자 워크스테이션으로 가져와 표시하기를 원할 수도 있습니다. CDE 유틸리티 `dtsearchpath`를 사용하여 다양한 CDE 셸 환경 변수를 설정하십시오.

Bourne 셸과 Korn 셸의 경우에는 다음 명령어를 입력하십시오.

```
eval `/usr/dt/bin/dtsearchpath`
```

C 셸의 경우에는 다음 명령어를 입력하십시오.

```
eval `/usr/dt/bin/dtsearchpath -c`
```

이 명령어는 `DISPLAY`를 다시 지역 워크스테이션으로 설정한 후에 CDE 응용 프로그램을 원격으로 실행함으로써 실행할 수 있습니다. 결과는 지역 워크스테이션에 표시될 것입니다.

주 - 이 예에서 Bourne 셸 및 Korn 셸 `dtsearchpath` 구문은 CDE를 실행하는 모든 플랫폼에서 사용할 수 있는 CDE 샘플 구현의 일부입니다. 그러나 C 셸(-c) 옵션은 Sun 플랫폼에서만 사용할 수 있습니다.

데스크탑 환경 파일

CDE 데스크탑은 OpenWindows DeskSet™ 응용 프로그램의 실행 시간 환경을 제공합니다. 이 기능을 위해 별도의 CDE 설정을 수행할 필요는 없습니다. 몇 가지 환경 설정은 다음 파일에서 제공됩니다.

```
/usr/dt/config/Xsession.d/0015.sun.env
```

특정 워크스테이션의 환경 설정값을 늘리거나 줄여야 할 경우에 이 파일을 원래의 위치에서 편집하거나 먼저 이 파일을 다음 파일 위치에 복사할 수 있습니다.

```
/etc/dt/config/Xsession.d/0015.sun.env
```

값을 늘리는 한 가지 예로는 OpenWindows 데스크탑 응용 프로그램이 사용하는 이전 OpenWindows 가상 키보드를 시작하는 것을 들 수 있습니다. CDE 데스크탑의 대부분의 응용 프로그램(및 대부분의 Sun 사용자)은 이 가상 키보드 유틸리티 프로그램을 사용하지 않으므로, 이 프로그램은 전반적인 데스크탑 시작 성능을 향상시키기 위해 기본 시작 순서 밖에 남아 있었습니다.

선택적 OpenWindows 가상 키보드(vkdb) 시작에 대한 추가 정보는 `0015.sun.env` 파일의 주석을 참조하십시오.

Apple Macintosh 응용 프로그램 환경에서 플로피 및 CD 매체 사용

Apple Macintosh 응용 프로그램 환경(MAE) 버전 1.0을 설치하고 OpenWindows 파일 관리자 응용 프로그램을 통해 분리형 매체(디스켓 또는 CD-ROM)를 사용하려면, /etc/rmmount.conf 파일을 편집해야 합니다. 이렇게 하면 MAE가 파일 관리자 분리형 매체와 제대로 작동할 수 있습니다.

▼ /etc/rmmount.conf 파일 변경 방법

1. su를 입력하여 루트 사용자가 된 후, 암호를 입력하십시오.
2. /etc 디렉토리로 가십시오. 다음 명령어를 입력하고 Return을 누르십시오.

```
# cd /etc
```

3. /etc/rmmount.conf 파일을 선호하는 편집기에서 여십시오.
4. #Actions 아래의 명령어를 목록의 끝으로 가져 가십시오.

```
action floppy action_macfs.so
예를 들어,
# more rmmount.conf
# @(#)rmmount.conf 1.2      92/09/23 SMI
#
# Removable Media Mounter configuration file.
#
# File system identification
ident hsfs ident_hsfs.so cdrom
ident ufs ident_ufs.so cdrom floppy
ident pcfs ident_pcfs.so floppy
ident macfs ident_macfs.so floppy
# Actions
action cdrom action_filemgr.so
action floppy action_filemgr.so
action floppy action_macfs.so
```

5. 저장하고 종료하십시오.

MAE가 설치되면, MAE는 사용자를 위해 macfs action 행을 rmmount.conf 파일에 둡니다. MAE가 제거되면, 이 행도 제거됩니다

이러한 변경을 통해 분리형 매체를 OpenWindows 또는 CDE 파일 관리자 응용 프로그램에서 사용할 수 있으며, MAE의 Macintosh 플로피를 계속 사용할 수 있습니다. 그러나 이 변경으로 인해 MAE는 포맷되지 않았거나, 읽을 수 없는 플로피, 또는 DOS용 플로피를 인식할 수 없게 합니다. 이러한 디스크는 이 설정 절차를 사용하여 OpenWindows 파일 관리자 응용 프로그램으로 가게 됩니다.

MAE가 읽을 수 없거나, 포맷되지 않은 플로피나 DOS 플로피에 액세스하도록 하려면, `action floppy action action_macfs.so` 행이 `action floppy action _filemgr.so` 행 앞에 와야 합니다.

네트워크에서 데스크탑 환경설정

데스크탑은 고도로 네트워크화된 환경에서도 잘 작동하도록 설계되어 있습니다.

- “데스크탑 네트워킹 개요” 117 페이지
- “데스크탑 네트워킹 환경설정의 일반적 단계” 123 페이지
- “데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지
- “데스크탑 클라이언트 및 서버의 환경설정” 126 페이지
- “응용 프로그램 서비스의 관리” 131 페이지

데스크탑 구조를 통해 시스템 관리자가 전체 네트워크에서 분산할 수 있는 컴퓨팅 자원은 다음과 같습니다.

- 응용 프로그램
- 응용 프로그램용 데이터 파일
- 데스크탑 세션 서비스(로그인 관리자와 파일 관리자와 같은 데스크탑 응용 프로그램)
- 도움말 서비스(도움말 데이터 파일은 중앙 도움말 서버에 둘 수 있습니다)

데스크탑 네트워킹 개요

운영 체제는 분산 파일 시스템과 원격 실행과 같은 다양한 네트워킹 서비스를 제공합니다. X 서버는 원격 모니터 액세스, 보안 서비스와 같은 네트워킹 기능을 추가로 제공합니다.

데스크탑에서는 이러한 네트워킹 기능 위에 사용자 인터페이스를 두고 있습니다. 이 인터페이스와 기본 구조는 다음과 같은 네트워크화된 시스템을 만드는데 목표를 두고 있습니다.

- 간편한 사용. 사용자는 응용 프로그램과 데이터가 네트워크 어디에 위치하는지를 걱정하지 않고도 응용 프로그램을 실행하고 데이터 파일에 액세스할 수 있습니다.

- 간편한 관리. 데스크탑은 시스템에 원격 데이터 및 응용 프로그램을 보다 쉽게 찾을 수 있도록 하는 응용 프로그램 통합 도구 및 네트워크 검색 경로를 제공합니다. 또한 데스크탑의 파일 이름 매핑 프로세스를 통해 수많은 서버가 있는 복잡한 네트워크도 보다 쉽게 관리할 수 있습니다.
- 유연성. 데스크탑의 관리 기능은 특정 공통 네트워크 환경에 맞도록 설계된 것이기는 하지만 데스크탑은 다양한 사용자 정의 네트워크 환경설정도 수용할 수 있습니다.

네트워크화된 데스크탑 서비스 유형

네트워킹을 통해 사용자는 다른 시스템에 분산된 다음과 같은 다양한 컴퓨팅 서비스에 액세스할 수 있습니다.

- 데스크탑 세션 및 그 응용 프로그램— 예를 들어, 작업공간 관리자와 파일 관리자
- 기타 응용 프로그램
- 데이터 파일

네트워킹 용어에서는 하나 이상의 다른 시스템에 컴퓨팅 서비스를 제공하는 시스템을 **서버**라고 합니다. 서버로부터 서비스를 받는 시스템은 해당 서버의 **클라이언트**라고 합니다.

복잡한 네트워크에서는 시스템이 네트워크 전체의 많은 시스템에 있는 서비스를 사용할 수 있습니다. 또한 한 시스템이 특정 서버 유형(예: 세션 서버)으로서의 기능을 하면서 동시에 클라이언트(예: 응용 프로그램 서버의 클라이언트)로서의 기능을 수행할 수도 있습니다.

전형적인 네트워크 환경

데스크탑에서의 네트워크는 보통 다음과 같은 주요 구성요소를 조합하여 구성할 수 있습니다.

모니터—X 서버가 실행되는 곳입니다.

로그인/세션 서버—데스크탑 응용 프로그램 (로그인 관리자, 작업공간 관리자 등)이 실행되는 곳입니다.

응용 프로그램 서버—기타 응용 프로그램이 실행되는 곳입니다.

파일 서버—응용 프로그램이 사용하는 데이터가 위치하는 곳입니다.

가장 일반적인 네트워크 환경설정 중의 하나는 시스템이 응용 프로그램 서버에 액세스하는 것입니다. 그림 7-1은 응용 프로그램 서버를 사용하는 워크스테이션을 나타냅니다. X 서버와 데스크탑 세션이 워크스테이션에서 실행되고 있습니다.

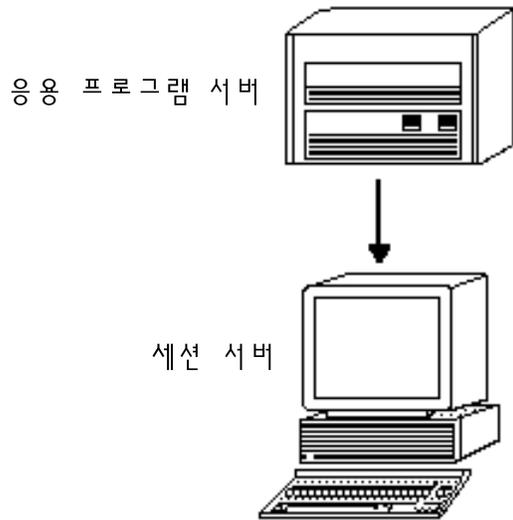


그림 7-1 응용 프로그램 서버는 데스크탑 세션에 서비스를 제공합니다.

또한 네트워크는 보통 파일 서버를 사용하여 대량의 데이터를 저장합니다. 이 데이터는 응용 프로그램 서버에서 실행되는 응용 프로그램이 사용하거나 데스크탑 응용 프로그램이 사용할 수 있습니다(예를 들어, 파일 관리자가 데이터 파일을 파일 관리자 윈도우에 표시하려면 해당 데이터 파일에 액세스해야 합니다).

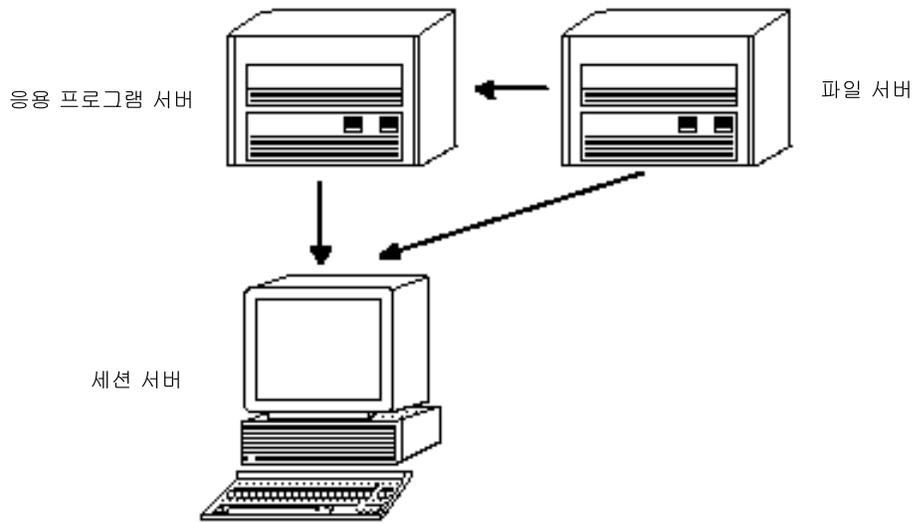


그림 7-2 파일 서버는 응용 프로그램 서버와 세션 서버에 데이터를 제공합니다.

X 터미널은 X 서버를 실행하고 다른 시스템으로부터 데스크탑 세션 서비스를 얻습니다.

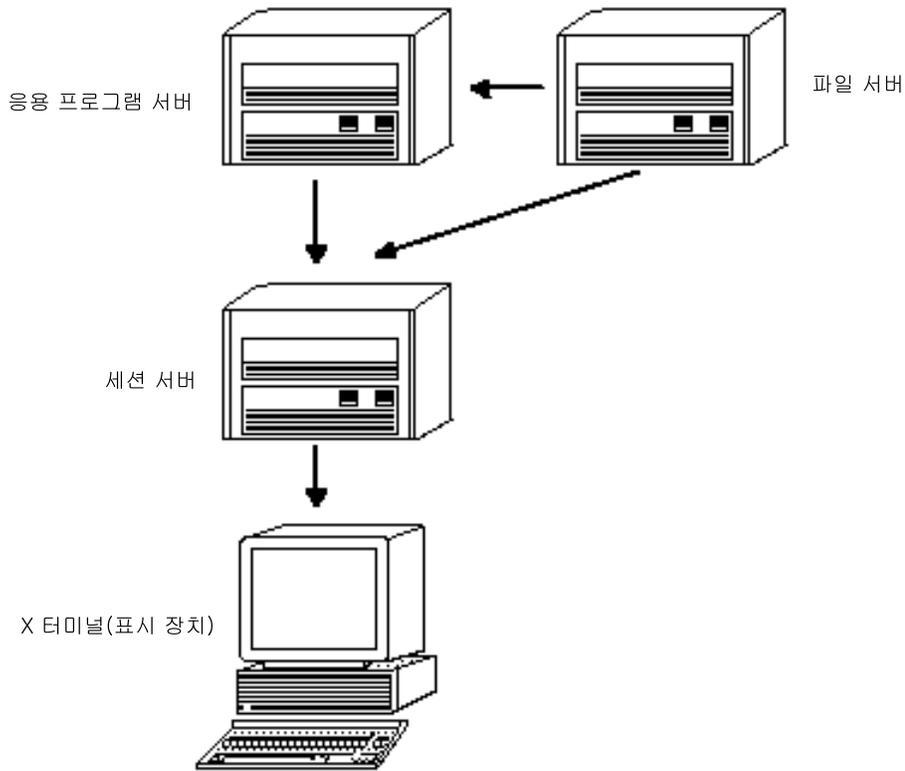


그림 7-3 X 터미널은 세션 서버로부터 세션 서비스를 얻습니다.

기타 네트워킹 환경

데스크탑은 유연성을 갖추고 있어 보다 복잡한 네트워크 환경설정도 지원할 수 있습니다. 이것은 파일 서버 이외에 응용 프로그램 서버가 사용할 수 있는 다양한 서비스를 구성할 수 있다는 것을 의미합니다.

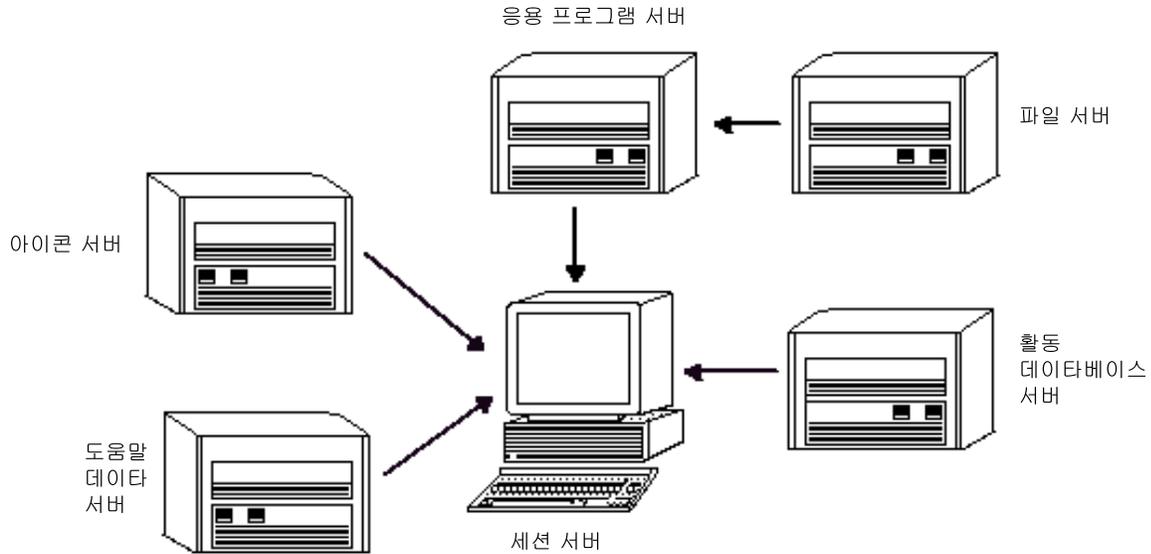


그림 7-4 데스크탑 응용 프로그램 서버에 필요한 서비스를 분산할 수 있습니다.

요약—서버 유형

모니터—X 서버를 실행하는 시스템입니다.

로그인 및 세션 서버—데스크탑 세션(로그인 관리자, 세션 관리자, 윈도우 관리자, 파일 관리자 등)을 실행하는 시스템입니다.

응용 프로그램 서버—응용 프로그램이 실행되는 시스템입니다. 또한 실행 호스트라고 부르기도 합니다.

파일 서버—응용 프로그램용 데이터 파일이 저장되는 시스템입니다.

도움말 서버—도움말 데이터 파일이 저장되는 시스템입니다.

(활동) 데이터베이스 서버—활동과 데이터 유형 정의가 수록된 파일이 저장되는 시스템입니다.

아이콘 서버—아이콘 파일이 저장되는 시스템입니다.

이외에도 네트워크에는 암호 서버, 우편 서버, 비디오 서버 등이 포함될 수 있습니다.

데스크탑 네트워킹 환경설정의 일반적 단계

데스크탑 네트워킹 환경설정은 일반적으로 다음과 같은 3단계로 진행됩니다.

1. 기본 운영 체제 네트워크 서비스를 구성합니다.

데스크탑이 의존하는 운영 체제에서 제공하는 네트워킹 서비스를 말합니다. “데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지를 참조하십시오.

2. 데스크탑 네트워킹 소프트웨어 및 서비스를 설치하고 구성합니다.

설치하는 클라이언트 또는 서버 시스템의 유형에 관계없이 데스크탑에 필요한 서비스를 말합니다. “데스크탑 클라이언트 및 서버의 환경설정” 126 페이지를 참조하십시오.

3. 특정 유형의 서버 또는 클라이언트를 구성합니다.

예를 들어, 응용 프로그램 서버는 파일 서버와 다른 단계를 거쳐 구성해야 합니다. “응용 프로그램 서비스의 관리” 131 페이지를 참조하십시오.

데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정

데스크탑에는 다음과 같은 기본 네트워킹 환경설정이 필요합니다.

- 사용자는 세션 서버와 데스크탑 서비스를 세션 서버에 제공하는 각 시스템에 로그인 계정을 가지고 있어야 합니다. 사용자는 모든 클라이언트 및 서버 시스템에서 사용자 ID와 그룹 ID가 동일해야 합니다.
- 시스템은 세션 및 기타 응용 프로그램이 사용하는 데이터가 있는 원격 파일 시스템에 대한 액세스 권한이 있어야 합니다.
- 원격 프린터에 액세스할 수 있도록 lp 인쇄 스폰서를 구성해야 합니다.
- 전자우편 서비스를 위한 sendmail을 구성해야 합니다.
- X 인증을 설치해야 합니다.

사용자에게 로그인 계정 제공하기

이 절에서는 데스크탑 네트워킹에 필요한 로그인 계정 요구사항을 설명합니다.

로그인 계정 제공

사용자는 다음과 같은 시스템에 로그인 계정을 가지고 있어야 합니다.

- 응용 프로그램 서버, 파일 서버, 네트워크화된 프린터를 제공하는 시스템 등을 포함하여 서비스를 데스크탑에 제공하는 모든 시스템.
- 사용자가 액세스하는 모든 세션 서버. 세션 서버는 보통 X 터미널과 함께 사용됩니다.

동일한 사용자 및 그룹 ID 제공

UNIX 사용자는 로그인 이름과 숫자로 된 사용자 ID(UID)로 식별됩니다. 데스크탑 네트워크에서 사용자는 모든 클라이언트 및 서버 시스템에서 동일한 로그인 이름과 UID를 가지고 있어야 합니다.

UNIX 사용자는 또한 하나 이상의 로그인 그룹으로 할당됩니다. 각 그룹에는 그룹 이름과 숫자로 된 그룹 ID(GID)가 있습니다. 데스크탑 네트워크에서 모든 시스템은 동일한 그룹 이름과 그룹 ID를 사용해야 합니다.

자세한 내용은 `id(1)` 또는 `id(1M)` 매뉴얼 페이지를 참조하십시오.

분산 파일 시스템 액세스의 환경설정

데스크탑은 NFS를 사용하여 시스템 간에 파일을 공유합니다. 공유 파일이 있는 네트워크상의 모든 파일 시스템을 식별하여 공유 파일이 모두 적절한 시스템에 정확하게 마운트되어 있는지 확인해야 합니다.

일반적으로 사용자는 다음과 같은 원격 파일 액세스를 제공해야 합니다.

- 사용자의 홈 디렉토리는 모든 데스크탑 클라이언트 및 서버 시스템에서 공유해야 합니다. 이 작업이 필요한 이유는 다음과 같습니다.
 - 원격 시스템의 응용 프로그램이 액세스해야 하는 데이터 파일이 홈 디렉토리에 있기 때문입니다. 예를 들어, 데이터 파일을 사용하는 응용 프로그램은 보통 홈 디렉토리를 기본 데이터 파일 위치로 사용합니다.
 - 홈 디렉토리가 기본 `dtspcd` 인증 디렉토리이기 때문입니다. `dtspcd`에 대한 자세한 내용은 “서브 프로세스 제어 데몬 구성” 129 페이지를 참조하십시오.
- 사용자가 홈 디렉토리에 없는 데이터 파일에 액세스해야 하는 경우, 데이터 파일에서 운영되는 모든 데스크탑 클라이언트 및 서버 시스템이 이러한 데이터 파일을 공유해야 합니다.
- 모든 데스크탑 클라이언트 및 서버 시스템이 데스크탑 설치 및 구성 디렉토리 (`/usr/dt`와 `/etc/dt`)를 공유함으로써 사용자의 모든 응용 프로그램이 동일한 데스크탑 환경설정 파일에 액세스할 수 있어야 합니다.

네트워크화된 홈 디렉토리의 제공

데스크탑 네트워크는 사용자가 네트워크상의 모든 클라이언트 및 서버 시스템 간에 공유되는 단일 홈 디렉토리를 가지고 있을 때 가장 효율적으로 작동합니다.

네트워크화된 홈 디렉토리를 통해 사용자는 개인 사용자 정의 및 환경설정 값을 상실하지 않으면서 네트워크상의 다른 시스템을 사용할 수 있습니다. 이는 이전 세션을 복원하는데 필요한 개인 사용자 정의와 정보가 홈 디렉토리의 하위 디렉토리에 저장되기 때문입니다.

또한 공통 홈 디렉토리는 다음의 사항에도 필요합니다.

- 기본 X 인증 메커니즘 “X 인증 구성” 126 페이지를 참조하십시오.
- 데스크탑 서브 프로세스 제어 데몬. 이 데몬은 원격 응용 프로그램 시작에 포함되며 사용자의 홈 디렉토리에 쓸 수 있어야 합니다.

파일 이름 일관성

네트워크는 사용자가 동일한 이름을 사용하여 모든 시스템의 데이터 파일에 액세스할 수 있도록 구성해야 합니다. 이를 **파일 이름 일관성**의 제공이라고 하며 일반적으로 해당 기호 링크를 작성함으로써 달성할 수 있습니다. 예를 들어, 실제 디렉토리 마운트 위치에 대한 기호 링크를 작성함으로써 각 사용자의 홈 디렉토리를 `/users/login_name`으로 사용할 수 있도록 모든 시스템을 구성할 수 있습니다.

원격 프린터에 대한 액세스 구성

데스크탑은 lp 인쇄 스플러를 사용하여 지역 또는 원격 프린터에 액세스합니다. lp 스플러의 구성에 관한 내용은 `lpadmin(1M)` 매뉴얼 페이지를 참조하십시오.

데스크탑 그래픽 인터페이스를 사용하여 인쇄를 시도하기 전에 lp 명령어를 사용하여 모든 프린터로 정확하게 인쇄할 수 있는지 시험해야 합니다.

반드시 동일한 프린터 장치 이름을 사용하십시오. 예를 들어, 특정 프린터가 직접 연결된 시스템에 `Postscript1`으로 알려져 있는 경우에 해당 프린터에 원격 액세스하는 다른 모든 시스템도 `Postscript1`이라는 이름을 사용해야 합니다.

전자우편의 구성

데스크탑 우편 시스템은 `sendmail`을 사용하여 시스템에 우편을 배달합니다. 전자우편 연결 구성 방법에 관한 자세한 내용은 `sendmail(1M)` 매뉴얼 페이지를 참조하십시오.

데스크탑으로부터 우편을 송수신하기 전에 `mailx` 명령어를 사용하여 우편을 정확하게 송수신할 수 있는지 시험해야 합니다.

X 인증 구성

데스크탑은 기본 X 메커니즘을 사용하여 원격 응용 프로그램(X 클라이언트)이 지역 표시장치에 액세스하도록 허용합니다. 이를 구성하는 가장 손쉬운 방법은 각 사용자에게 네트워크화된 홈 디렉토리를 제공하는 것입니다. 이렇게 함으로써 다음과 같은 요구사항을 만족시킬 수 있습니다.

- 사용자는 *HomeDirectory/.Xauthority* 파일에 대한 읽기 및 쓰기 사용권한이 있어야 합니다.
- 응용 프로그램 서버의 *.Xauthority* 파일에 응용 프로그램을 실행할 표시장치를 위한 “매직 쿠키”가 있어야 합니다.

자세한 내용은 X(1) 또는 *xauth(1M)* 매뉴얼 페이지를 참조하십시오.

데스크탑 클라이언트 및 서버의 환경설정

이 절에서는 데스크탑에 고유한 네트워크 환경설정 요구사항을 다룹니다. —즉, 이러한 기능은 기본 운영 체제가 아닌 데스크탑에서 제공됩니다.

이 절은 다음과 같은 두 부분으로 나누어 집니다.

- 로그인 및 세션 서비스 구성
- 응용 프로그램 및 데이터에 필요한 서비스 구성 응용 프로그램, 데이터베이스, 아이콘, 파일, 도움말 서버 및 클라이언트 등이 포함됩니다.

로그인 및 세션 서비스 구성

로그인 및 세션 서버는 데스크탑 서비스(로그인 관리자, 세션 관리자, 파일 관리자, 윈도우 관리자 등)를 표시장치 및 X 서버에 제공하는 시스템입니다.

일반적으로 세션 서버는 X 터미널에 서비스를 제공합니다. 그러나 세션 서비스가 X 터미널 및 워크스테이션이 모두 액세스하는 하나 이상의 서버에 집중되도록 네트워크를 구성할 수도 있습니다.

로그인 관리자는 로그인 서비스를 기타 표시장치에 제공하는 일을 맡는 데스크탑 구성 요소입니다. 사용자가 로그인하면 세션 관리자가 사용자를 위해 시작됩니다.

로그인 및 세션 서버와 X 터미널의 구성에 관한 내용은 “네트워크 표시장치에 로그인 화면 표시” 24 페이지를 참조하십시오.

기타 응용 프로그램 관련 서비스 구성

이 절에서는 데스크탑에 공통적으로 적용되는 네트워킹 요구사항을 다룹니다.

- 응용 프로그램 서버
- 데이터베이스 서버
- 아이콘 서버
- 도움말 서버

▼ 데스크탑 클라이언트 및 서버를 구성하는 방법

1. 데스크탑에 필요한 운영 체제 네트워크 환경설정 값을 제공하십시오.

“데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지를 참조하십시오.

2. 데스크탑이나 최소 파일 세트를 설치하십시오.

다음 파일을 설치해야 합니다.

- 전체 공통 데스크탑 환경 런타임 파일 세트
- 또는 CDE-MIN 및 CDE-TT 파일 세트

주 - 설치 및 파일 세트는 구입업체에 따라 다를 수 있습니다.

3. ToolTalk 파일이름 데이터베이스 서버 데몬 `rpc.ttdserver`를 위한 시스템을 구성하십시오.

이 작업은 데스크탑을 설치할 때 자동으로 진행됩니다. 자세한 내용은 “ToolTalk 데이터베이스 서버의 구성” 130 페이지를 참조하십시오.

4. 서브 프로세스 제어 데몬(`dtspcd`)을 설치하고 구성하십시오.

이 작업은 데스크탑을 설치할 때 자동으로 진행됩니다. 자세한 내용은 “서브 프로세스 제어 데몬 구성” 129 페이지를 참조하십시오.

5. 필요한 모든 원격 데이터를 마운트하십시오.

데이터가 해당 데이터를 사용하는 응용 프로그램이 실행되고 있는 시스템 이외의 시스템에 위치하는 경우 “원격” 데이터로 간주됩니다.

예를 들어,

- 응용 프로그램이 파일 서버에 있는 데이터를 사용하는 경우에는 이러한 파일을 마운트해야 합니다.
- 파일 관리자 아이콘이 아이콘 서버에 위치하는 경우에는 세션 서버가 이러한 파일을 마운트해야 합니다.
- 네트워크가 도움말 서버를 데스크탑 도움말 파일에 사용하는 경우에는 세션 서버와 모든 응용 프로그램 서버가 도움말 데이터를 마운트해야 합니다.

마운트 지점에 관한 자세한 내용은 다음 절 “원격 파일 시스템을 위한 마운트 지점 구성” 128 페이지를 참조하십시오.

원격 파일 시스템을 위한 마운트 지점 구성

데스크탑이 파일 이름을 한 시스템에서 다른 시스템으로 보내는 경우에는 이러한 파일 이름을 대상 시스템이 이해할 수 있는 이름으로 변환하거나 **매핑**해야 합니다. 매핑은 파일이 다른 시스템의 다른 위치에 마운트되어 있을 수도 있고 따라서 다른 이름을 사용하여 액세스해야 하기 때문에 필요합니다. 예를 들어, sysA의 /projects/big 파일은 sysB의 /net/sysA/projects/big으로 액세스할 수 있습니다.

파일 이름 매핑의 요구사항

이와 같은 파일 이름 매핑 작업을 정확하게 수행하려면 다음 사항 중 하나에 해당해야 합니다.

- **mount** 명령어를 사용하여 파일 시스템을 정적으로 마운트합니다. 이러한 유형의 정적 마운트는 대개 /etc/checklist, /etc/mnttab 또는 /etc/filesystems와 같은 파일에 구성됩니다.
파일 이름 매핑을 시스템 간에 정확하게 수행하려면 파일 시스템 마운트가 동일한 호스트 이름을 사용해야 합니다. 호스트가 여러 이름(예: 별명)으로 알려진 경우(또는 호스트가 다른 이름으로 알려진 LAN 주소를 하나 이상 가지고 있는 경우), 모든 마운트에 대해 동일한 이름 및 이름 형식을 사용해야 합니다.
- **또는** 자동 마운터를 사용하여 파일 시스템을 기본 /net 마운트 지점에 마운트합니다.
- **또는** 자동 마운터를 사용하여 파일 시스템을 기본 /net 이외의 다른 위치에 마운트하고 DTMOUNTPOINT 환경 변수가 마운트 지점을 표시하도록 설정되어 있습니다. 다음 절 “DTMOUNTPOINT의 값 설정” 128 페이지를 참조하십시오.
자동 마운터에 관한 자세한 내용은 automount(1M) 매뉴얼 페이지를 참조하십시오.

▼ DTMOUNTPOINT의 값 설정

다음 조건을 모두 만족시키는 경우에는 DTMOUNTPOINT 환경 변수를 설정해야 합니다.

- 자동 마운터는 파일 시스템을 마운트하는데 사용됩니다.
- **그리고** 원격 파일 시스템을 /net 이외의 위치에 마운트합니다.
다음과 같이 프로세스에 대한 DTMOUNTPOINT를 설정해야 합니다.
- 작업공간 관리자(dtwm)와 파일 관리자(dtfile)와 같이 사용자가 로그인할 때 자동으로 시작되는 사용자의 데스크탑 프로세스.
- inetd와 같은 메커니즘에서 시작되는 rpc.ttdbserver와 dtspcd와 같은 시스템 프로세스
- 지역 또는 원격 시스템의 데스크탑이 시작하는 응용 프로그램
- 사용자가 셸 명령행에서 시작하는 응용 프로그램
이러한 프로세스에 대한 DTMOUNTPOINT를 설정하려면

1. /etc/inetd.conf 파일을 편집하십시오.

2. dtspcd 항목을 찾아서 다음을 추가하십시오.

```
-mount_point mount_point
```

3. rpc.ttdbserver 항목을 찾아서 다음을 추가하십시오.

```
-m mount_point
```

예를 들어, 마운트 지점이 /nfs인 상태에서 자동 마운터를 사용하는 경우에 /etc/inetd.conf의 항목은

```
dtspc stream tcp nowait root /usr/dt/bin/dtspcd \  
/usr/dt/bin/dtspcd -mount_point /nfs  
rpc stream tcp wait root /usr/dt/bin/rpc.ttdbserver \  
100083 1 rpc.ttdbserver -m /nfs
```

4. 시스템에서 /etc/inetd.conf를 다시 읽는 절차를 수행하십시오. 자세한 내용은 **inetd(1M) 매뉴얼 페이지**를 참조하십시오.

5. 사용자 로그인이 값을 상속받을 수 있도록 **DTMOUNTPOINT**를 설정하십시오.

이 작업은 /etc/dt/config/Xsession.d에 변수를 설정함으로써 수행할 수 있습니다. 환경 변수의 설정에 관한 자세한 내용은 “환경 변수 설정 방법” 47 페이지를 참조하십시오.

서브 프로세스 제어 데몬 구성

데스크탑 서브 프로세스 제어(SPC) 서비스는 클라이언트/서버 명령어 실행을 제공합니다.

데스크탑 서브 프로세스 제어 데몬(dtspcd)은 데스크탑이 원격 응용 프로그램을 시작하는데 사용됩니다. 원격 클라이언트로부터 명령을 실행하도록 요청을 받는 것은 inet 데몬입니다. inet 데몬을 구성하는 방법에 대한 자세한 내용은 **inetd.conf(1M) 매뉴얼 페이지**를 참조하십시오.

데스크탑 활동 호출 라이브러리는 SPC 서비스를 사용하여 원격 활동을 호출합니다.

▼ dtspcd를 구성하는 방법

- dtspc가 /etc/services와 /inetd.conf에 적절하게 등록되어 있는지 확인하십시오.

dtspcd(1M) 매뉴얼 페이지를 참조하십시오.

SPC 보안

서브 프로세스 제어 서비스에 대한 인증은 파일 시스템 인증에 기초하여 수행됩니다. dtspcd는 모든 SPC 클라이언트 시스템에서 마운트하는 **인증 디렉토리**에 대한 액세스 권한이 있어야 합니다.

기본적으로 dtspcd 인증 디렉토리는 사용자의 홈 디렉토리입니다. 그러나 /etc/inetd.conf 디렉토리에서 --auth_dir를 설정함으로써 다른 위치를 사용하는 dtspcd를 구성할 수 있습니다. 자세한 내용은 dtspcd(1M) 매뉴얼 페이지를 참조하십시오.

SPC 인증은 파일 시스템 인증에 기초하여 수행되기 때문에 SPC 서비스는 분산 파일 시스템이 갖는 보안성만 가질 수 있습니다. 사용자가 분산 파일 시스템을 신뢰하고 있지 않은 네트워크에서 데스크탑을 사용하고 있는 경우에는 dtspcd를 사용불가능하게 만들 수 있습니다. dtspcd를 사용불가능하게 하려면 /etc/services에서 dtspc 항목을 주석으로 만드십시오.

원격 실행을 위한 환경 변수 구성

데스크탑이 원격 시스템의 응용 프로그램을 시작하는 활동을 사용하면 사용자의 환경 변수는 해당 원격 시스템에 복사되어 응용 프로그램의 환경에 놓이게 됩니다.

기본적으로 일부 환경 변수는 원격 시스템에 복사되기 전에 변경됩니다. 환경 변수가 응용 프로그램의 환경에 놓이기 전에 활동 호출 구성요소와 데스크탑의 서브 프로세스 제어 서비스를 모두 구성하여 추가 환경 변수 처리를 수행할 수 있습니다.

기본 환경설정 및 수정 방법에 관한 자세한 내용은 dtactionfile(4)과 dtspcdenv(4) 매뉴얼 페이지를 참조하십시오.

ToolTalk 데이터베이스 서버의 구성

ToolTalk의 한 구성요소는 ToolTalker 데이터베이스 서버 /usr/dt/bin/rpc.ttdbserver입니다.

ToolTalk 데이터베이스는 ToolTalk 메시지 서비스가 파일 이름 매핑에 사용합니다. 데스크탑이 설치되고 더 이상 추가 구성이 필요하지 않으면 보통 /etc/inetd.conf에 등록됩니다.

ToolTalk 데이터베이스 서버 및 구성 옵션에 관한 자세한 내용은 rpc.ttdbserver(1M) 매뉴얼 페이지를 참조하십시오.

ToolTalk 메시지 서버의 구성

ToolTalk 메시지 서버는 ttssession입니다. 기본적으로 어떠한 환경설정도 필요하지 않으며 로그인 중에 Xsession 스크립트가 시작합니다.

ToolTalk 메시지 서버 및 구성 옵션에 관한 자세한 내용은 ttssession 매뉴얼 페이지를 참조하십시오.

캘린더 데몬의 구성

캘린더 응용 프로그램의 한 구성요소는 캘린더 데몬 rpc.cmsd입니다. 데스크탑이 설치되고 더 이상 추가 구성이 필요하지 않으면 보통 /etc/inetd.conf에 등록됩니다.

캘린더 데몬 및 구성 옵션에 관한 자세한 내용은 rpc.cmsd(1) 매뉴얼 페이지를 참조하십시오.

응용 프로그램 서비스의 관리

이 절에서는 다음 사항에 대한 구체적인 환경설정 요구사항을 다룹니다.

- 응용 프로그램 서버 및 클라이언트
- 특정 서비스를 제공하는 데스크탑 서버—데이터베이스 서버, 아이콘 서버, 도움말 서버
또한 네트워크화된 응용 프로그램의 두 가지 특수 환경설정에 대한 네트워킹 요구사항을 다룹니다.
- 원격 실행 호스트
- 여러 파일 시스템 마운트에서 실행되는 응용 프로그램

검색 경로 환경 변수

데스크탑은 일련의 환경 변수를 사용하여 활동 및 데이터 유형 데이터베이스, 도움말 파일, 아이콘 파일과 같은 응용 프로그램 데스크탑 환경설정 파일을 찾는 데 사용하는 검색 경로를 지정합니다.

검색 경로 환경 변수의 사용 방법에 관한 자세한 내용은 “데스크탑 검색 경로 및 환경 변수” 146 페이지 또는 dtenwar(5) 매뉴얼 페이지를 참조하십시오.

응용 프로그램 서버 및 클라이언트의 구성

표준 응용 프로그램 서버의 환경설정에서는 응용 프로그램 서버에 다음과 같은 모든 이진 파일 및 환경설정 파일이 포함됩니다.

- 응용 프로그램 실행 파일
- app-defaults, 메시지 카탈로그, 해당 응용 프로그램에 대한 공유 라이브러리와 같은 표준 응용 프로그램 환경설정 파일
- 데스크탑 환경설정 파일
 - 활동 및 데이터 유형 정의 파일
 - 아이콘 이미지 파일
 - 데스크탑 도움말 데이터 파일

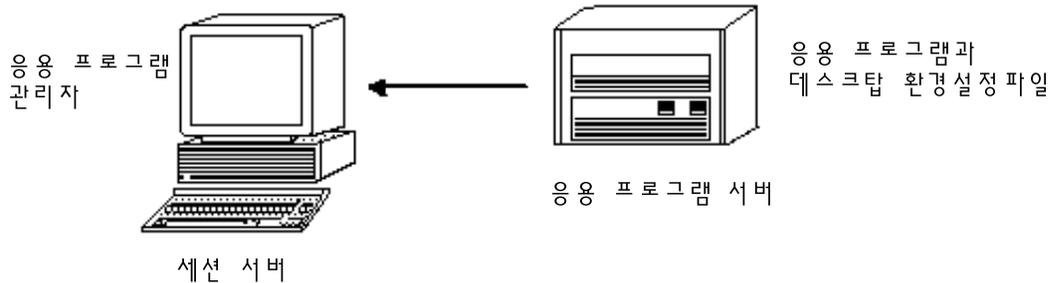


그림 7-5 표준 응용 프로그램 서버 환경설정

▼ 응용 프로그램 서버의 구성 방법

1. 데스크탑에 필요한 운영 체제 네트워크 환경설정 값을 제공하십시오.
“데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지를 참조하십시오.
2. 서버에 필요한 일반 데스크탑 환경설정 값을 제공하십시오.
“데스크탑 클라이언트 및 서버를 구성하는 방법” 127 페이지를 참조하십시오.
3. 응용 프로그램을 설치하십시오.
4. 응용 프로그램이 자동으로 등록되지 않는 경우에는 사용자가 등록 절차를 수행해야 합니다.
제 5 장을 참조하십시오.

▼ 응용 프로그램 서버의 클라이언트 구성 방법

1. 데스크탑에 필요한 운영 체제 네트워크 환경설정 값을 제공하십시오.
“데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지를 참조하십시오.
2. 클라이언트에 필요한 일반 데스크탑 환경설정 값을 제공하십시오.
“데스크탑 클라이언트 및 서버를 구성하는 방법” 127 페이지를 참조하십시오.
3. 시스템 전체에서 또는 개별적으로 응용 프로그램 서버를 응용 프로그램 검색 경로에 추가하십시오.
시스템 전체— /etc/dt/config/Xsession.d/0010.dtpaths에서 DTSPSYSAPPHOSTS 변수를 설정하고 보내십시오.
개별적 —HomeDirectory/.dtprofile에서 DTSPUSERAPPHOSTS 변수를 설정하고 보내십시오.

예를 들어, /etc/dt/config/Xsession.d/0010.dtpaths에 다음과 같은 명령행을 입력하면 호스트 이름이 SysAAA와 SysBBB인 시스템을 응용 프로그램 검색 경로에 추가합니다.

```
export DTSPSYSAPPHOSTS=SysAAA:,SysBBB:
```

응용 프로그램 검색 경로 설정에 관한 자세한 내용은

- “응용 프로그램 검색 경로” 148 페이지
- “검색 경로 값 설정” 146 페이지

데이터베이스, 아이콘, 도움말 서비스의 구성

응용 프로그램과 관련된 활동 및 데이터 유형 정의, 아이콘, 도움말 데이터 파일 등은 일반적으로 해당 응용 프로그램과 동일한 시스템에 설치됩니다.

예를 들어, 도움말 데이터 파일의 전형적인 구성 방법을 생각해 봅시다.

- 파일 관리자의 도움말 파일은 보통 세션 서버에 위치합니다. 도움말 검색 경로는 자동으로 세션 서버에서 적절한 위치를 검색하기 때문에 데스크탑은 이러한 파일을 찾을 수 있습니다.
- 다른 응용 프로그램에 대한 도움말 파일은 보통 해당 응용 프로그램과 동일한 응용 프로그램 서버에 위치합니다. 응용 프로그램의 검색 경로를 수정하면 도움말 검색 경로도 자동으로 수정되기 때문에 세션 서버는 이러한 파일을 찾을 수 있습니다.

데이터베이스(활동 및 데이터 유형), 도움말, 아이콘 데이터 등을 네트워크상의 다른 위치에 둘 수도 있습니다. 예를 들어, 네트워크가 다중 세션 서버를 사용하는 경우에 데스크탑 응용 프로그램(파일 관리자, 스타일 관리자 등)의 모든 도움말 데이터 파일을 저장하는 도움말 서버를 구성할 수도 있습니다. 이렇게 하면 도움말 파일을 각 세션 서버에 복제할 필요가 없기 때문에 디스크 공간을 보존할 수 있습니다.

▼ 데이터베이스, 도움말, 아이콘 서버의 작성 방법

1. 데스크탑에 필요한 운영 체제 네트워크 환경설정 값을 제공하십시오.

“데스크탑을 위한 기본 운영 체제 네트워크의 환경설정” 123 페이지를 참조하십시오.

2. 클라이언트에 필요한 일반 데스크탑 환경설정 값을 제공하십시오.

“데스크탑 클라이언트 및 서버를 구성하는 방법” 127 페이지를 참조하십시오.

3. 데이터베이스, 도움말, 아이콘 파일을 설치하십시오.

파일은 시스템의 어느 곳에나 둘 수 있습니다. 그러나 이러한 파일은 시스템이 응용 프로그램 서버에 지정된 경우에 자동으로 검색되는 디렉토리이기 때문에 다음과 같은 위치를 사용하는 것이 보다 쉬울 수 있습니다.

- 데이터베이스 파일: /etc/dt/appconfig/types/language
- 도움말 파일: /etc/dt/appconfig/help/language
- 아이콘 파일: /etc/dt/appconfig/iconslanguage

데이터베이스 서버를 설치하는 경우에는 활동 명령어(EXEC_STRING)를 실행할 곳을 지정하기 위해 활동을 작성해야 합니다. “원격 실행 호스트의 지정” 134 페이지를 참조하십시오.

▼ 데이터베이스, 아이콘, 도움말 서버를 찾기 위해 세션 서버를 구성하는 방법

1. 데스크탑에 필요한 운영 체제 네트워크 환경설정 값을 제공하십시오.

“데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지를 참조하십시오.

2. 클라이언트에 필요한 일반 데스크탑 환경설정 값을 제공하십시오.

“데스크탑 클라이언트 및 서버를 구성하는 방법” 127 페이지를 참조하십시오.

3. 데이터베이스, 아이콘, 도움말 서버를 해당 검색 경로에 추가하십시오.

- 데이터 파일을 “데이터베이스, 도움말 아이콘 서버의 작성 방법”의 단계 3의 “데이터베이스, 도움말, 아이콘 서버의 작성 방법” 133 페이지에서 지정한 위치에 둔 경우에는 응용 프로그램 검색 경로를 수정할 수 있습니다.

- 데이터 파일을 다른 위치에 둔 경우에는 지정된 검색 경로를 수정해야 합니다.

예를 들어, 도움말 파일을 시스템 SysCCC의 /etc/dt/help 디렉토리에 둔 경우 다음과 같은 명령행을 /etc/dt/config/Xsession.d/0010.dtpaths에 추가합니다.

```
export DTSPYSHELP=/net/SysCCC/etc/dt/help
```

검색 경로 설정에 관한 자세한 내용은 다음 내용을 참조하십시오.

- “데이터베이스(활동/데이터 유형) 검색 경로” 151 페이지
- “아이콘 검색 경로” 153 페이지
- “도움말 검색 경로” 154 페이지
- “검색 경로 값 설정” 146 페이지

특수 네트워크 응용 프로그램 환경설정

이 절에서는 응용 프로그램을 다음과 같이 실행하기 위해 시스템을 구성하는 방법을 설명합니다.

- 활동을 포함하는 시스템이 이외의 위치—원격 실행 호스트에서 실행
- 파일 시스템 마운트에 대해 지역적으로 실행

원격 실행 호스트의 지정

전형적인 응용 프로그램 서버 구성에서 활동 정의는 실행 가능한 응용 프로그램과 동일한 시스템에 위치합니다. 그러나 다른 시스템에서 명령어를 실행하도록 활동을 작성할 수 있습니다. 이 구성에서 응용 프로그램을 포함하는 시스템을 **실행 호스트**라고 합니다.

활동 정의는 세션 서버 또는 활동 및 데이터 유형 서비스를 **데이터베이스 서버** 또는 **데이터베이스 호스트**라고 하는 세션 서버에 제공하는 시스템에 위치할 수 있습니다.

활동 정의는 EXEC_HOST 필드를 사용하여 활동 명령어(EXEC_STRING)를 실행할 곳을 지정합니다. 예를 들어, 다음과 같은 활동 정의는 xload 클라이언트가 호스트 이름이 SysDDD인 시스템에서 실행되도록 지정합니다.

```
ACTION XloadSysDDD
{
  TYPE                COMMAND
  EXEC_HOST           SysDDD
  EXEC_STRING         /usr/bin/X11/xload -label SysDDD
}
```

EXEC_HOST 필드가 둘 이상의 호스트 이름을 지정하는 경우, 데스크탑은 해당 활동을 실행할 수 있는 호스트를 발견할 때까지 각 호스트에서 EXEC_STRING을 실행하도록 시도합니다. 예를 들어, 다음과 같은 EXEC_HOST 필드는 활동이 먼저 SysDDD에서 EXEC_STRING을 실행하려고 시도하고 이것이 실패하는 경우에 SysEEE를 시도해야 한다고 지정합니다.

```
EXEC_HOST                SysDDD, SysEEE
```

EXEC_HOST 필드가 특정 활동에 대해 설정되지 않은 경우에는 기본값 %DatabaseHost%를 갖게 됩니다. %DatabaseHost%의 값은 데이터베이스 검색 경로에서 얻습니다.

예를 들어, 다음 명령행을 /etc/dt/config/Xsession.d/0010.dtpaths에 추가하여 데이터베이스 검색 경로를 수정했다고 가정합니다.

```
DTSPSPSYSDATABASEHOSTS=SysAAA: , /net/SysBBB/etc/dt/appconfig/types/C
```

호스트 수식 구문—SysAAA를 사용하여 SysAAA가 지정됩니다. 이 검색 경로 요소를 사용하여 발견한 활동 정의는 데이터베이스 호스트를 SysAAA로 설정합니다. 그러나 구문에 호스트 수식자가 포함되지 않았기 때문에 검색 경로의 /net/SysBBB... 부분을 사용하여 발견한 활동은 데이터베이스 호스트를 지역 시스템으로 설정합니다.

▼ 원격 실행 호스트를 구성하는 방법

1. **데스크탑에 필요한 운영 체제 네트워크 환경설정 값을 제공하십시오.**
“데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지를 참조하십시오.
2. **서버에 필요한 일반 데스크탑 환경설정 값을 제공하십시오.**
“데스크탑 클라이언트 및 서버를 구성하는 방법” 127 페이지를 참조하십시오.
3. **지역 실행을 위해 응용 프로그램을 적절하게 설치하고 구성했는지 확인하십시오.**

▼ 활동 정의가 포함된 시스템을 구성하는 방법

1. **데스크탑에 필요한 운영 체제 네트워크 환경설정 값을 제공하십시오.**
“데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지를 참조하십시오.

2. 서버에 필요한 일반 데스크탑 환경설정 값을 제공하십시오.
“데스크탑 클라이언트 및 서버를 구성하는 방법” 127 페이지를 참조하십시오.
3. 활동 정의 및 응용 프로그램 그룹을 작성하고 설치하십시오.
“원격 시스템에서 응용 프로그램을 실행하는 활동 작성” 203 페이지와 “일반 응용 프로그램 그룹 작성 및 관리” 63 페이지를 참조하십시오.

▼ 세션 서버를 구성하는 방법

1. 데스크탑에 필요한 운영 체제 네트워크 환경설정 값을 제공하십시오.
“데스크탑을 위한 기본 운영 체제 네트워킹의 환경설정” 123 페이지를 참조하십시오.
2. 클라이언트에 필요한 일반 데스크탑 환경설정 값을 제공하십시오.
“데스크탑 클라이언트 및 서버를 구성하는 방법” 127 페이지를 참조하십시오.
3. 데이터베이스 호스트를 포함하도록 활동 검색 경로를 수정하십시오.
“데이터베이스(활동/데이터 유형) 검색 경로” 151 페이지를 참조하십시오.
4. 실행 호스트를 포함하도록 응용 프로그램 검색 경로를 수정하십시오.
“응용 프로그램 검색 경로” 148 페이지를 참조하십시오.

지역적으로 응용 프로그램 실행하기

표준 응용 프로그램 서버 구성은 응용 프로그램을 응용 프로그램 서버에서 실행합니다. 응용 프로그램을 원격 시스템에 설치한 경우에도 지역적으로 세션 서버에서 실행하는 것이 바람직할 때도 있습니다.

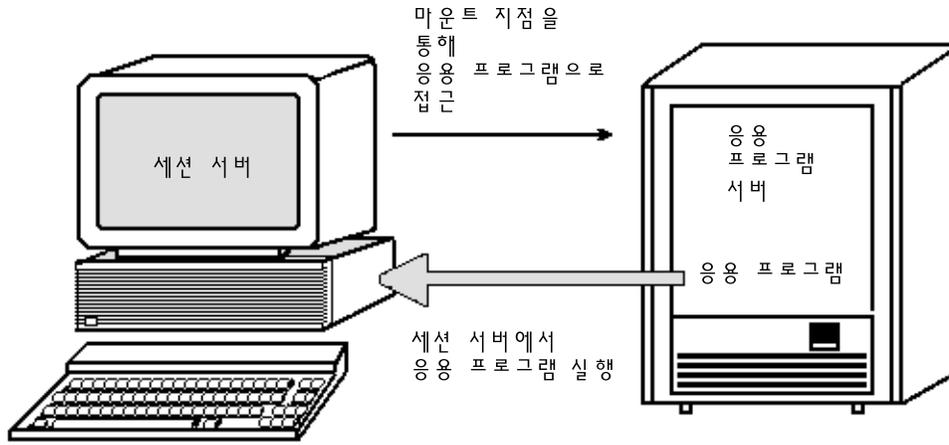


그림 7-6 여러 마운트 지점에서 실행

▼ 응용 프로그램 서버를 구성하는 방법

특별한 환경설정이 필요하지 않습니다.

▼ 세션 서버를 구성하는 방법

- 응용 프로그램 검색 경로를 수정하십시오. 응용 프로그램에 대한 지역 절대 경로를 사용하십시오.

예를 들어, 다음과 같은 변수 정의를 사용하여 sysAAA에 등록된 응용 프로그램을 찾을 수 있습니다.

```
DTSPSYSAPPHOSTS=/net/SysAAA/etc/dt/appconfig/appmanager/C
```

세션 서버가 app-defaults, 메시지 카탈로그, 공유 라이브러리와 같은 응용 프로그램 환경설정 파일에 액세스할 수 있어야 합니다.

데스크탑에서 인쇄 구성 및 관리

데스크탑 사용자가 파일을 인쇄하는 방법에는 여러 가지가 있습니다. 그것은 데스크탑에서 인쇄하는 것과 응용 프로그램에서 인쇄하는 것입니다.

데스크탑에서 인쇄하는 방법은 다음과 같습니다.

- 파일 관리자에서 파일을 선택하고 선택 메뉴 또는 아이콘 팝업 메뉴에서 인쇄를 선택하는 방법.
- 파일을 파일 관리자로부터 프론트 패널 기본 프린터 제어 또는 개인 프린터 서브 패널의 기본 프린터 제어로 끄는 방법.
- 파일을 파일 관리자에서 인쇄 관리자 기본 윈도우의 프린터로 끄는 방법.
응용 프로그램에서 인쇄하려면 보통 해당 응용 프로그램 윈도우 안에 있는 메뉴 또는 기타 제어도구에서 액세스할 수 있는 인쇄 명령어를 사용하십시오.

- “프린터 추가 및 삭제” 139 페이지
- “프린터 아이콘 이미지” 141 페이지
- “기본 프린터 구성” 142 페이지
- “인쇄 개념” 143 페이지

프린터 추가 및 삭제

이 절에서는 데스크탑에서 프린터를 추가하고 삭제하는 절차를 설명합니다.

▼ 데스크탑에 프린터 추가 방법

1. 프린터를 시스템의 구성에 추가하십시오.
사용자 운영 체제의 시스템 관리 설명서에 있는 지시사항을 따르십시오.
2. 다음 명령어를 실행하십시오.

```
env LANG=language /usr/dt/bin/dtprintinfo -populate
```

- 인쇄 관리자를 다시 시작하거나 응용 프로그램 관리자의 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오. 프린터가 나타나는지 확인하십시오.
- 사용자에게 우편을 보내 사용자 역시 인쇄 관리자를 다시 시작하거나 활동 재로드를 실행해야 함을 알려주십시오.

시스템 프린터 구성 목록을 호출할 때마다 인쇄 관리자가 이 목록을 읽습니다. 새 프린터를 발견하면 새 데스크탑 프린터 활동 및 해당 프린터의 아이콘을 자동으로 작성합니다. 프린터가 데스크탑에 나타나도록 하는 것 외에 별도로 수행할 작업은 없습니다.

▼ 데스크탑에서 프린터 삭제 방법

- 프린터를 시스템 구성에서 제거하십시오.
사용자 운영 체제의 시스템 관리 설명서에 있는 지시사항을 따르십시오.
- 인쇄 관리자를 다시 시작하거나 응용 프로그램 관리자의 데스크탑 제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오. 프린터가 없어졌는지 확인하십시오.
- 사용자에게 우편을 보내 사용자 역시 인쇄 관리자를 다시 시작하거나 활동 재로드를 실행해야 함을 알려주십시오.

시스템 프린터 구성 목록을 호출할 때마다 인쇄 관리자가 이 목록을 읽습니다. 프린터가 목록에서 없어진 것을 발견하면, 해당 프린터의 활동 및 아이콘을 인쇄 관리자와 파일 관리자에서 자동으로 제거합니다. 프린터를 데스크탑에서 삭제하는 것 외에 별도로 수행할 작업은 없습니다.

주 - 인쇄 관리자는 프린터를 프론트 패널에서 제거할 수 없습니다. 따라서 구성에서 프린터를 제거할 때마다 시스템의 모든 사용자에게 삭제된 프린터의 아이콘을 프론트 패널에서 제거할 것임을 알려주는 우편을 보내야 합니다.

작업 갱신 간격 수정

인쇄 관리자에 표시되는 정보가 갱신되는 주기를 변경하려면, 작업 갱신 간격을 수정하십시오. 기본적으로 인쇄 관리자는 인쇄 작업에 대한 정보를 보기 위해 30초마다 프린터를 조회합니다. 옵션 설정 대화 상자(보기 메뉴에서 옵션 설정을 선택하면 표시됨)의 간격 갱신 슬라이더를 사용하여 인쇄 관리자가 프린터를 조회하는 주기를 변경할 수 있습니다.

프린터 아이콘 이미지

프린터를 추가하면, 프린터에 기본 프린터 아이콘이 자동으로 할당됩니다. 다른 아이콘을 사용하고 싶으면, 아이콘 파일을 `/etc/dt/appconfig/icons/language`에 두거나, 아이콘 검색 경로에 있는 다른 디렉토리에 두십시오. 그러면 사용자가 이 아이콘을 선택하여 프린터의 기본 아이콘을 교체할 수 있습니다.

아이콘의 전체 세트(대형, 중형, 소형)를 작성해야 합니다. 그렇지 않으면 인쇄 관리자의 아이콘 선택기에 나타나지 않을 것입니다.

아이콘 검색 경로에 대한 자세한 정보는 “아이콘 검색 경로” 153 페이지를 참조하십시오.

아이콘 파일 이름 및 크기

아이콘 파일 이름 지정 요구사항은 다음과 같습니다.

base_name.size.type

여기서

size—l (대형), m (중형), t (최소형). 아이콘 크기에 대한 자세한 내용은 “아이콘 크기 요약” 226 페이지를 참조하십시오.

type—pm(색상 픽스맵), bm(비트맵)

예를 들어, 색상 프린터의 경우, 중형 및 소형 픽스맵 아이콘의 아이콘 파일 이름은 `ColorPrinter.m.pm` 및 `ColorPrinter.t.pm`입니다.

아이콘 작성에 관한 자세한 내용은 제 14 장을 참조하십시오.

▼ 프린터의 아이콘, 인쇄 레이블 또는 설명을 전체적으로 변경하는 방법

사용자가 인쇄 관리자를 사용하여 프린터를 수정하기 전에, 프린터를 추가하는 즉시 전역 프린터의 특성을 변경해야 합니다. 일단 사용자가 인쇄 관리자를 사용하여 프린터 특성을 수정하면, 변경된 내용을 볼 수 없게 됩니다.

아이콘, 프린터 레이블 또는 설명 등에 관한 원하는 정보에 기초하여 파일 `/etc/dt/appconfig/types/language/printer_queue_name.dt`를 편집하십시오.

1. ICON 필드에서 *basename*를 새 아이콘 기본 이름으로 갱신하십시오.
2. LABEL 필드에서 *labelname*을 프린터의 새 레이블로 갱신하십시오.

3. DESCRIPTION 필드에서 텍스트를 갱신하십시오.

이 필드는 프린터 위치, 프린터 유형 및 프린터 연결을 입력하기에 좋은 장소입니다. 둘 이상의 행을 추가하려면, 행 끝에 \을 입력하십시오. 예를 들어,

```
DESCRIPTION      This is a PostScript Printer in Building 1 \
                  Room 123. Call 555-5555 for problems.
```

기본 프린터 구성

사용자는 다음과 같은 방식으로 기본 프린터에 액세스할 수 있습니다.

- 객체를 프론트 패널 기본 프린터 제어 도구에 끌어 놓습니다.
- 파일 관리자에서 객체를 선택하고 선택 메뉴 또는 아이콘 팝업 메뉴에서 인쇄를 선택합니다.
- 기본 프린터를 사용하는 응용 프로그램에서 인쇄합니다.

▼ 기본 인쇄를 위한 대상 변경 방법

모든 사용자에게 대하여 기본 프린터를 변경하려면

1. 파일 `/etc/dt/config/Xsession.d/0010.dtpaths`을 여십시오.
`/etc/dt/config/Xsession.d/0010.dtpaths`가 없으면,
`/usr/dt/config/Xsession.d/0010.dtpaths`에서 복사하십시오.
2. `LPDEST=printer` 행에서, `printer`를 기본 인쇄를 위한 새 대상으로 변경하십시오.
이 행이 없으면, `LPDEST=printer` 행을 추가하고, `printer`에는 기본 프린터로 설정하려는 프린터의 이름을 쓰십시오.
3. 사용자는 로그아웃한 후 다시 로그인해야 합니다.

단일 사용자에게 대해 기본 프린터를 변경하려면 해당 사용자가 다음과 같이 해야 합니다.

- 개인 프린터 서브패널에서 다른 프린터를 프론트 패널로 복사합니다.

다른 프린터를 기본 프린터로 지정하려면

1. 홈 폴더로 가서 `.dtprofile` 파일을 여십시오.
2. `LPDEST` 환경 변수의 값을 설정하는 행을 추가하거나 편집하십시오.

```
LPDEST=printer_device; export LPDEST
csh를 사용하고 있는 경우, 구문은 다음과 같습니다.
```

```
setenv LPDEST printer_device
```

예를 들어, 다음 명령행은 기본 프린터를 장치 이름이 laser3d인 프린터로 변경합니다.

```
LPDEST=laser3d; export LPDEST
csh를 사용하는 경우, 구문은 다음과 같습니다.

setenv LPDEST laser3d
```

인쇄 개념

파일을 프린터 제어도구에 놓음으로써 인쇄 요청이 개시되면, 시스템은 다음과 같이 작업을 진행합니다.

1. 시스템은 놓은 객체의 정의에 맞는 데이터 유형 데이터베이스를 검색합니다.
2. 데이터 유형(인쇄 활동에서 ARG_TYPE 필드를 사용하여 지정됨)에 대한 고유한 인쇄 활동이 있는 경우에는 이 인쇄 활동이 사용되며, 그렇지 않은 경우에는 기본 인쇄 활동(dt1p)이 사용됩니다. 예를 들어, 파일이 PostScript®인 경우, 시스템은 PostScript 파일에 대한 인쇄 활동을 사용합니다. (이 활동은 /usr/dt/appconfig/types/language/dt.dt에 정의되어 있습니다). 이 데이터 유형에 대해 활동 작성 도구를 사용했으면, 입력한 인쇄 명령어는 이 데이터 유형을 사용하여 파일을 인쇄하는데 사용될 고유한 인쇄 활동입니다.
3. 파일이 일반 UNIX lp 인쇄 서브 시스템을 사용하는 프린터로 전달됩니다.

데스크탑 검색 경로

데스크탑은 검색 경로를 사용하여 응용 프로그램과 이에 연관된 데스크탑 파일을 찾습니다.

- “데스크탑 검색 경로 및 환경 변수” 146 페이지
- “검색 경로 값 설정” 146 페이지
- “응용 프로그램 검색 경로” 148 페이지
- “데이터베이스(활동/데이터 유형) 검색 경로” 151 페이지
- “아이콘 검색 경로” 153 페이지
- “도움말 검색 경로” 154 페이지
- “지역화된 검색 경로” 156 페이지

데스크탑은 표 9-1에 나열된 네 가지 검색 경로를 제공합니다.

표 9-1 데스크탑 검색 경로

검색 경로	설명
응용 프로그램	응용 프로그램을 찾는데 사용됩니다. 응용 프로그램 관리자는 응용 프로그램 검색 경로를 사용하여 사용자가 로그인할 때 최상위 레벨을 동적으로 위치시킵니다.
데이터베이스	활동 및 데이터 유형 정의 파일(*.dt 파일)과 프론트 패널 파일(*.fp 파일)의 위치를 추가로 지정하는데 사용됩니다.
아이콘	아이콘의 위치를 추가로 지정하는데 사용됩니다.
도움말 데이터	데스크탑 도움말 데이터의 위치를 추가로 지정하는데 사용됩니다.

검색 경로에는 지역 및 원격 디렉토리 모두가 포함될 수 있습니다. 따라서, 검색 경로는 데스크탑의 네트워크 구조에서 중요한 역할을 합니다. 예를 들어, 응용 프로그램 서버는 응용 프로그램 검색 경로에 나열되어 있으므로 시스템은 응용 프로그램을 응용 프로그램 서버에서 찾습니다.

검색 경로에 원격 위치가 포함될 경우, 해당 위치에 대한 원격 파일 액세스를 구성해야 합니다. 자세한 내용은 “분산 파일 시스템 액세스의 환경설정” 124 페이지를 참조하십시오.

데스크탑 검색 경로 및 환경 변수

데스크탑 검색 경로는 로그인시 데스크탑 유틸리티 dtsearchpath에 의해 작성됩니다. dtsearchpath 유틸리티는 환경 변수와 내장 위치를 조합하여 검색 경로를 작성합니다.

dtsearchpath가 읽는 환경 변수를 **입력 변수**라고 합니다. 이러한 변수는 시스템 관리자 또는 최종 사용자가 설정하는 변수입니다. 입력 변수는 이름 지정 규약 DTSP*를 사용합니다.

로그인시 dtsearchpath가 실행되면, dtsearchpath는 이러한 변수에 할당된 값을 조합하고, 내장 위치를 추가하여 **출력 변수**의 값을 작성합니다. 각 검색 경로에는 하나의 출력 변수가 있습니다.

표 9-2 데스크탑 검색 경로 환경 변수

검색 경로	출력 환경 변수	시스템 전체 입력 변수	개인 입력 변수
응용 프로그램	DTAPPSEARCHPATH	DTSPSYSAPPHOSTS	DTSPUSERAPPHOSTS
데이터베이스 ¹	DTDATABASESEARCHPATH	DTSPSYSDATABASEHOSTS	DTSPUSERDATABASEHOSTS
아이콘	XMICONSEARCHPATH, XMICONBMSEARCHPATH	DTSPSYSICON	DTSPUSERICON
도움말 데이터	DTHELPSEARCHPATH	DTSPSYSHELP	DTSPUSERHELP

¹활동, 데이터 유형, 프론트 패널 정의

구성요소는 출력 변수의 값을 사용합니다. 예를 들어, 응용 프로그램 관리자는 응용 프로그램 검색 경로(DTAPPSEARCHPATH)의 값을 사용하여 응용 프로그램 그룹을 찾습니다.

검색 경로 값 설정

검색 경로는 시스템 전체에서 또는 개별적으로 수정할 수 있습니다. 검색 경로를 수정하려면 시스템 전체 또는 개인 입력 변수에 대한 값을 설정하면 됩니다. 모든 수정 내용이 내장 검색 경로 위치에 추가됩니다.

▼ 검색 경로의 현재 값을 보는 방법(출력 변수)

- dtsearchpath 명령어를 사용하여 검색 경로의 현재 값을 표시하십시오.
 - 현재 (로그인) 사용자의 값을 얻으려면 다음 명령어를 입력하십시오.

```
dtsearchpath -v
```

- 다른 사용자의 값을 얻으려면 다음 명령어를 입력하십시오.

```
dtsearchpath -user
```

검색 경로 값에는 다음 변수가 포함되어야 합니다.

%H—DTHELPPSEARCHPATH에서 사용. 도움말 볼륨 이름.

%B—XMICONSEARCHPATH에서 사용. 아이콘 파일의 기본 이름

%M—XMICONSEARCHPATH에서 사용되는 아이콘 파일(.1, .m, .s, .t)의 크기.

%L—LANG 환경 변수의 값.

▼ 검색 경로를 개별적으로 수정하는 방법

1. *HomeDirectory* / *.dtprofile*을 열어서 편집하십시오.
2. 개인 입력 변수를 정의하고 내보내는 행을 추가하거나 편집하십시오.
예를 들어, 다음과 같은 명령행은 사용자의 개인 응용 프로그램 검색 경로에 위치를 추가합니다.

```
export DTSPUSERAPPHOSTS+=/projects1/editors
```

3. 변경내용을 반영하려면 로그아웃한 후 다시 로그인하십시오.

▼ 검색 경로를 시스템 전체에서 수정하는 방법

1. 루트로 로그인하십시오.
2. */etc/dt/config/Xsession.d/0010.dtpaths* 파일이 없으면, */usr/dt/config/Xsession.d/0010.dtpaths*를 복사하여 만드십시오.
3. */etc/dt/config/Xsession.d/0010.paths*를 열고 편집하십시오. 시스템 전체 입력 변수를 정의하고 내보내는 행을 추가하거나 편집하십시오.
예를 들어, 다음과 같은 명령행은 시스템 전체 도움말 검색 경로에 위치를 추가합니다.

```
export DTSPSYSHELP=/applications/helpdata
```

4. 시스템의 모든 사용자에게 로그아웃한 후 다시 로그인하여 변경 내용을 반영해야 한다고 알려 주십시오.

응용 프로그램 검색 경로

응용 프로그램 검색 경로는 데스크탑이 네트워크를 통해 응용 프로그램 서버와 지역 시스템에서 응용 프로그램을 찾기 위해 사용하는 일차적인 검색 경로입니다.

응용 프로그램 검색 경로에 위치가 추가되면, 다른 검색 경로(데이터베이스, 아이콘 및 도움말)가 해당 데이터에 해당되는 위치를 반영하기 위해 자동으로 갱신됩니다. 따라서, 응용 프로그램 검색 경로는 응용 프로그램 및 데스크탑 환경설정 파일을 비교적 간단하게 관리할 수 있게 합니다. “응용 프로그램 검색 경로가 데이터베이스, 아이콘 및 도움말 검색 경로에 미치는 영향” 150 페이지를 참조하십시오.

기본 응용 프로그램 검색 경로

기본 응용 프로그램 검색 경로에는 개인, 시스템 전체 및 내장 위치가 포함됩니다. 기본 *language*는 C입니다.

개인 위치—*HomeDirectory/.dt/appmanager*

시스템 전체 위치—*/etc/dt/appconfig/appmanagers/language*

내장 위치—*/usr/dt/appconfig/appmanager/language*

응용 프로그램 검색 경로 환경 변수

응용 프로그램 검색 경로는 내장 위치와 다음과 같은 입력 변수로부터 만들어집니다.

DTSPSYSAPPHOSTS—시스템 전체 응용 프로그램 검색 경로 입력 변수

DTSPUSERAPPHOSTS—개인 응용 프로그램 검색 경로 입력 변수

조합된 검색 경로는 출력 변수 DTAPPSEARCHPATH로 지정됩니다.

응용 프로그램 검색 경로 입력 변수에 대한 구문

변수 DTSPSYSAPPHOSTS와 DTSPUSERAPPHOSTS의 구문은 다음과 같습니다.

VARIABLE=location [, location . . .]

여기서 *location*이 가질 수 있는 구문은 다음과 같습니다.

/path—지역 (세션 서버) 시스템의 디렉토리를 지정합니다. 이 구문을 사용하여 지역 디렉토리를 추가하십시오.

hostname:—시스템 *hostname*의 시스템 전체 디렉토리 `/etc/dt/appconfig/appmanager/language`를 지정합니다. 이 구문을 사용하여 응용 프로그램 서버를 추가하십시오.

hostname:/*path*—원격 시스템 *hostname*의 디렉토리를 지정합니다.

localhost:—지역 시스템 전체 위치. 이 키워드는 지역 시스템 전체 위치의 우선순위를 변경하는데 사용됩니다. “시스템 전체 지역 위치의 우선순위 변경” 149 페이지를 참조하십시오.

응용 프로그램 검색 경로 값을 조합하는 방법

응용 프로그램 검색 경로(DTAPPSEARCHPATH)의 값은 우선순위에 따라 나열된 다음 위치를 조합하여 만들어집니다.

- DTSPUSERAPPHOSTS 변수를 사용하여 지정된 위치
- 기본 개인 위치: `HomeDirectory/.dt/appmanager`
- 기본 위치: `/etc/dt/appconfig/appmanager/language`
- DTSPSYSAPPHOSTS 변수를 사용하여 지정된 위치
- `/usr/dt/appconfig/appmanager/language`
구문

hostname:

은 시스템 *hostname*의 `/etc/dt/appconfig/appmanager` 디렉토리를 지정하기 위해 확장됩니다.

시스템 전체 지역 위치의 우선순위 변경

기본적으로 지역적인 시스템 전체 위치 (`/etc/dt/appconfig/appmanager/language`)는 원격 위치보다 우선순위가 높습니다. 따라서, 지역 응용 프로그램 그룹은 동일한 이름의 원격 그룹보다 우선순위가 높습니다. 예를 들어, 지역 및 원격 시스템 모두에 프린터 응용 프로그램 그룹(`/etc/dt/appconfig/appmanager/language/Printers`)이 있는 경우, 지역 그룹이 사용됩니다.

응용 프로그램 검색 경로 입력 변수는 지역 시스템 전체 응용 프로그램 그룹의 우선순위를 지정하기 위한 구문을 제공합니다.

localhost:

예를 들어, 시스템이 응용 프로그램 서버 `SysA`, `SysB` 및 `SysC`에 액세스해야 하고, `SysB`에 있는 시스템 전체 응용 프로그램 그룹이 동일한 이름으로 된 지역 그룹보다 우선순위가 높아야 한다고 가정합니다.

다음과 같은 DTSPSYSAPPHOSTS 값이 이러한 동작을 만듭니다.

DTSPSYSAPPHOSTS=SysB:,localhost:,SysA:,SysC:

응용 프로그램 검색 경로가 데이터베이스, 아이콘 및 도움말 검색 경로에 미치는 영향

응용 프로그램 검색 경로에 위치가 추가되면 해당 위치가 데이터베이스, 아이콘 및 도움말 검색 경로에 자동으로 추가됩니다. 따라서 응용 프로그램 검색 경로 입력 변수만 설정하면 응용 프로그램 서버를 검색 경로에 추가할 수 있습니다.

예를 들어, DTSPSYSAPPHOSTS를 다음과 같이 설정한 경우,

```
export DTSPSYSAPPHOSTS=servera:
```

검색 경로는 표 9-3와 같은 영향을 받습니다.

표 9-3 영향을 받은 검색 경로

검색 경로	검색 경로에 추가되는 디렉토리
응용 프로그램	servera:/etc/dt/appconfig/appmanager/language
데이터베이스	servera:/etc/dt/appconfig/types/language
아이콘	servera:/etc/dt/appconfig/icons/language
도움말	servera:/etc/dt/appconfig/help/language

또한, DTSPSYSAPPHOSTS를 다음과 같이 설정한 경우,

```
export DTSPSYSAPPHOSTS=/projects1/apps
```

검색 경로는 표 9-4와 같은 영향을 받습니다.

표 9-4 영향을 받은 검색 경로

검색 경로	검색 경로에 추가되는 디렉토리
응용 프로그램	/projects1/apps/appmanager/language
데이터베이스	/projects1/apps/types/language
아이콘	/projects1/apps/icons/language
도움말	/projects1/apps/help/language

데이터베이스(활동/데이터 유형) 검색 경로

데이터베이스 검색 경로는 데스크탑이 다음 사항이 포함된 지정된 파일 위치를 검색하도록 지시합니다.

- 활동 및 데이터 유형 정의(*.dt 파일)
- 프론트 패널 정의(*.fp 파일).

데이터베이스 서버를 작성하거나 데이터베이스 파일의 지역 위치를 추가할 때, 데이터베이스 검색 경로를 수정해야 할 경우가 있습니다.

기본 데이터베이스 검색 경로

기본 데이터베이스 검색 경로에는 기본, 시스템 전체 및 내장 위치가 포함됩니다. 기본 *language*는 C입니다.

개인 위치—*HomeDirectory/.dt/types*

시스템 전체 위치—*/etc/dt/appconfig/types/language*

내장 위치—*/usr/dt/appconfig/types/language*

응용 프로그램 검색 경로가 데이터베이스 검색 경로에 미치는 영향

응용 프로그램 검색 경로에 위치가 추가되면 해당 데이터베이스 하위 디렉토리가 데이터베이스 검색 경로에 자동으로 추가됩니다(“응용 프로그램 검색 경로가 데이터베이스, 아이콘 및 도움말 검색 경로에 미치는 영향” 150 페이지를 참조하십시오.)

예를 들어, 응용 프로그램 서버 *hosta:*가 응용 프로그램 검색 경로에 추가된 경우, 디렉토리 *hosta:/etc/dt/appconfig/types/language*가 데이터베이스 검색 경로에 자동으로 추가됩니다.

데이터베이스 검색 경로 환경 변수

데이터베이스 검색 경로는 내장 위치와 다음과 같은 입력 변수로부터 만들어집니다.

DTSPSYSDATABASEHOSTS—시스템 전체 데이터베이스 검색 경로 입력 변수

DTSPUSERDATABASEHOSTS—개인 데이터베이스 검색 경로 입력 변수

응용 프로그램 검색 경로 밖에 위치를 지정하려면 이 입력 변수를 사용하십시오.

조합된 데이터베이스 검색 경로는 출력 변수 DTDATABASESEARCHPATH로 지정됩니다.

데이터베이스 검색 경로 입력 변수에 대한 구문

변수 DTSPSYSDATABASEHOSTS 및 DTSPUSERDATABASEHOSTS에 대한 구문은 다음과 같습니다.

```
VARIABLE=location [,location...]
```

여기서 *location*이 가질 수 있는 구문은 다음과 같습니다.

/path—지역 (세션 서버) 시스템의 디렉토리를 지정합니다. 이 구문을 사용하여 지역 디렉토리를 추가하십시오.

hostname:—시스템 *hostname*의 시스템 전체 디렉토리 */etc/dt/appconfig/types/language*를 지정합니다.

hostname:/path—원격 시스템 *hostname*의 디렉토리를 지정합니다.

데이터베이스 검색 경로를 조합하는 방법

데이터베이스 검색 경로(DTDATABASESEARCHPATH)는 우선순위에 따라 나열된 다음 위치를 조합하여 만들어집니다.

- DTSPUSERDATABASEHOSTS 변수를 사용하여 지정된 위치
- DTSPUSERAPPHOSTS 변수에서 파생된 위치
- 기본 개인 위치: *HomeDirectory/.dt/types*
- 기본 위치: */etc/dt/appconfig/types/language*
- DTSPSYSDATABASEHOSTS 변수를 사용하여 지정된 위치
- DTSPSYSAPPHOSTS 변수에서 파생된 위치
- */usr/dt/appconfig/types/language*

구문

hostname:

은 시스템 *hostname*의 */etc/dt/appconfig/types* 디렉토리를 지정하기 위해 확장됩니다.

아이콘 검색 경로

아이콘 검색 경로는 데스크탑에 사용된 비트맵과 픽스맵 이미지가 있는 파일에 지정된 위치를 데스크탑이 검색하도록 지시합니다.

기본 아이콘 검색 경로

기본 아이콘 검색 경로에는 개인, 시스템 전체 및 내장 위치가 포함됩니다. 기본 *language* 는 C입니다.

개인 위치—*HomeDirectory/.dt/icons*

시스템 전체 위치—*/etc/dt/appconfig/icons/language*

내장 위치—*/usr/dt/appconfig/icons/language*

응용 프로그램 검색 경로가 아이콘 검색 경로에 미치는 영향

응용 프로그램 검색 경로에 위치가 추가되면 해당 아이콘 하위 디렉토리가 아이콘 검색 경로에 자동으로 추가됩니다(“응용 프로그램 검색 경로가 데이터베이스, 아이콘 및 도움말 검색 경로에 미치는 영향” 150 페이지를 참조하십시오.)

예를 들어, 응용 프로그램 서버 *hosta:*가 응용 프로그램 검색 경로에 추가되면, 디렉토리 *hosta:/etc/dt/appconfig/icons/language*가 아이콘 검색 경로에 자동으로 추가됩니다.

아이콘 검색 경로 환경 변수

데이터베이스 검색 경로는 내장 위치와 다음과 같은 입력 변수로부터 만들어집니다.

DTSPSYSICON—시스템 전체 아이콘 검색 경로 입력 변수

DTSPUSEICON—개인 응용 프로그램 검색 경로 입력 변수

응용 프로그램 검색 경로 밖에 위치를 지정하려면 이 입력 변수를 사용하십시오.

조합된 데이터베이스 검색 경로는 다음과 같은 두 출력 변수로 지정됩니다.

XMICONSEARCHPATH—컬러 표시장치에서 사용

XMICONBMSEARCHPATH—흑백 표시장치에서 사용

아이콘 검색 경로 입력 변수에 대한 구문

변수 DTSPSYSICON 및 DTSPUSERICON에 대한 구문은 다음과 같습니다.

```
VARIABLE=location [,location...]
```

여기서 *location*이 가질 수 있는 구문은 다음과 같습니다.

/path—지역 (세션 서버) 시스템의 디렉토리를 지정합니다. 이 구문을 사용하여 지역 디렉토리를 추가하십시오.

다른 시스템에 위치를 지정하려면 네트워크 파일 이름(예를 들어, */nfs/servera/projects/icons*)을 지정하십시오.

아이콘 검색 경로를 조합하는 방법

아이콘 검색 경로(XMICONSEARCHPATH 및 XMICONBMSEARCHPATH)는 우선순위에 따라 나열된 다음 위치를 조합하여 만들어집니다.

- DTSPUSERICON 변수를 사용하여 지정된 위치
- DTSPUSERAPPHOSTS 변수에서 파생된 위치
- 기본 개인 위치: *HomeDirectory/.dt/icons*
- 기본 위치: */etc/dt/appconfig/icons/language*
- DTSPSYSICON 변수를 사용하여 지정된 위치
- DTSPSYSAPPHOSTS 변수에서 파생된 위치
- */usr/dt/appconfig/icons/language*

컬러와 흑백 검색 경로는 픽스맵과 비트맵에 부여하는 우선순위만 다릅니다. XMICONSEARCHPATH 변수는 비트맵 앞에 픽스맵을 나열하며, XMICONBMSEARCHPATH는 픽스맵 앞에 비트맵을 나열합니다.

도움말 검색 경로

도움말 검색 경로는 데스크탑이 시스템에 등록될 도움말 정보가 있는 파일의 지정된 위치를 검색하도록 지시합니다.

기본 도움말 검색 경로

기본 도움말 검색 경로에는 개인, 시스템 전체 및 내장 위치가 포함됩니다. 기본 *language*는 C입니다.

개인 위치—*HomeDirectory/.dt/help*

시스템 전체 위치—/etc/dt/appconfig/help/language

내장 위치—/usr/dt/appconfig/help/language

응용 프로그램 검색 경로가 도움말 검색 경로에 미치는 영향

응용 프로그램 검색 경로에 위치가 추가되면 해당 도움말 하위 디렉토리가 도움말 검색 경로에 자동으로 추가됩니다(“응용 프로그램 검색 경로가 데이터베이스, 아이콘 및 도움말 검색 경로에 미치는 영향” 150 페이지를 참조하십시오.)

예를 들어, 응용 프로그램 서버 `hosta:`가 응용 프로그램 검색 경로에 추가되면, 디렉토리 `hosta:/etc/dt/appconfig/help/language`가 도움말 검색 경로에 자동으로 추가됩니다.

도움말 검색 경로 환경 변수

도움말 검색 경로는 내장 위치와 다음과 같은 입력 변수로부터 만들어집니다.

`DTSPSYSHELP`—시스템 전체 도움말 검색 경로 입력 변수

`DTSPUSERHELP`—개인 도움말 검색 경로 입력 변수

응용 프로그램 검색 경로 밖에 위치를 지정하려면 이 입력 변수를 사용하십시오.

조합된 데이터베이스 검색 경로는 출력 변수 `DTHELPPSEARCHPATH`로 지정됩니다.

도움말 검색 경로 입력 변수에 대한 구문

변수 `DTSPSYSHELP` 및 `DTSPUSERHELP`에 대한 구문은 다음과 같습니다.

```
VARIABLE=location [,location...]
```

여기서 `location`이 가질 수 있는 구문은 다음과 같습니다.

`/path`—지역 (세션 서버) 시스템의 디렉토리를 지정합니다. 이 구문을 사용하여 지역 디렉토리를 추가하십시오.

다른 시스템에 위치를 지정하려면 네트워크 파일 이름(예를 들어, `/nfs/servera/projects/help`)를 사용하십시오.

도움말 검색 경로를 조합하는 방법

도움말 검색 경로(`DTHELPPSEARCHPATH`)의 값은 우선순위에 따라 나열된 다음 위치를 조합하여 만들어집니다.

- DTSPUSERHELP 변수를 사용하여 지정된 위치
- DTSPUSERAPPHOSTS 변수에서 파생된 위치
- 기본 개인 위치: *HomeDirectory/.dt/help*
- 기본 위치: */etc/dt/appconfig/help/language*
- DTSPSYSHELP 변수를 사용하여 지정된 위치
- DTSPSYSAPPHOSTS 변수에서 파생된 위치
- */usr/dt/appconfig/help/language*

지역화된 검색 경로

출력 변수에는 기본(C) 위치 및 지역화된 위치에 대한 항목이 포함됩니다.

예를 들어, 기본 응용 프로그램 검색 경로는 다음과 같습니다.

```
HomeDirectory/.dt/appmanager
/etc/dt/appconfig/appmanager/language
/etc/dt/appconfig/appmanager/C/
usr/dt/appconfig/appmanager/language
/usr/dt/appconfig/appmanager/C
```

여기서 *language*는 LANG 환경 변수의 값입니다.

각 범위(시스템 전체 및 내장)에서 특정 언어 위치는 기본 위치보다 우선순위가 높습니다.

활동 및 데이터 유형 소개

활동과 데이터 유형은 응용 프로그램을 데스크탑에 통합하는 강력한 구성요소입니다. 이들은 응용 프로그램을 시작하고 데이터 파일을 조작하기 위한 사용자 인터페이스를 작성하는 방식을 제공합니다.

- “활동 소개” 157 페이지
- “데이터 유형 소개” 163 페이지

이 장에서는 활동 및 데이터 유형의 개념을 소개합니다. 설명하는 내용은 다음과 같습니다.

- 응용 프로그램에 대한 활동 및 데이터 유형을 작성하려는 이유
- 활동 및 데이터 유형을 서로 연관시키는 방법
- 활동 및 데이터 유형을 데스크탑 인쇄와 연관시키는 방법

활동 및 데이터 유형을 작성하는 절차와 규칙이 이 매뉴얼의 세 장에서 설명됩니다.

- 제 11 장을 사용하여 활동 및 데이터 유형을 작성하는 방법을 설명합니다.
활동 작성을 사용하면 대부분의 응용 프로그램에 대한 활동 및 데이터 유형의 정의 구문 규칙을 배우지 않고도 이를 작성할 수 있습니다.
- 제 12 장과 제 13 장에서는 환경설정 파일을 작성하고 편집하여 활동 및 데이터 유형을 수동으로 작성하는 방법을 설명합니다.
활동 작성에서 지원되지 않는 고급 기능을 사용하고자 할 경우에는 활동 및 데이터 유형을 수동으로 작성해야 합니다.

활동 소개

활동은 응용 프로그램 실행 및 데이터 파일 열기 같은 데스크탑 작업을 자동화하도록 작성된 명령어입니다. 활동은 응용 프로그램 매크로나 프로그램 함수와 매우 유사하게 실행됩니다. 각 활동에는 활동을 실행하는데 사용되는 이름이 있습니다.

일단 활동이 정의되면 이것을 데스크탑 사용자 인터페이스를 조정하는데 사용할 수 있으므로 작업이 보다 쉬워집니다. 데스크탑은 아이콘, 프론트 패널 제어 도구 및 메뉴 항목 같은 사용자 인터페이스 구성요소를 활동에 연결할 수 있는 기능을 제공합니다.

예를 들어, 응용 프로그램 관리자의 Desktop_Tools 응용 프로그램 그룹에는 다양한 유틸리티를 시작하는 아이콘이 포함됩니다.



그림 10-1 Desktop_Tools 응용 프로그램 그룹의 활동 아이콘

각 아이콘을 두 번 누르면 활동이 실행됩니다. 예를 들어, 다음은 사용자가 레이블이 Xwd인 표시장치 아이콘을 두 번 누를 때 실행되는 활동 정의의 일부입니다. 활동은 환경설정 파일 `/usr/dt/appconfig/types/language/xclients.dt`에서 다음과 같이 정의됩니다.

```

ACTION Xwd
{
    LABEL                Xwd Display
    TYPE                 COMMAND
    EXEC_STRING          /usr/bin/X11/xwd -noclick -in \
                        %(File)Arg_1"Xwd File To Display:"%
    ...
}

```

활동의 EXEC_STRING에 있는 명령어는 사용자가 아이콘을 두 번 누르면 실행됩니다.

프론트 패널 역시 활동을 사용합니다. 예를 들어, 다음은 개인 응용 프로그램 서브패널에서 레이블이 Terminal인 제어 도구 정의의 일부입니다. 제어 도구는 환경설정 파일 `/usr/dt/appconfig/types/language/dtwwm.fp`에 정의되어 있습니다.

```

CONTROL Term
{
    ICON                 Fpterminal
    LABEL               Terminal
    PUSH_ACTION         Dtterminal
    ...
}

```

PUSH ACTION 필드는 사용자가 제어도구를 누를 때 실행되는 활동을 지정합니다. 이 경우에 활동의 이름은 Dtterm입니다.

또한 활동은 일반적으로 메뉴에서 사용할 수 있습니다. 파일 관리자의 선택 메뉴에는 일반적으로 데이터 파일에 대한 활동이 있습니다. 예를 들어, XWD 파일(이름이 .xwd 또는 .wd로 끝나는 파일)에는 Xwud 활동을 실행하여 화면 이미지를 표시하는 열기 활동이 있습니다.

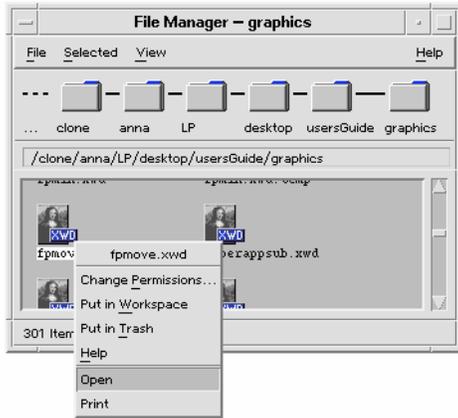


그림 10-2 데이터 유형이 XWD인 파일의 열기 활동

선택 메뉴에 있는 활동은 XWD 파일에 대한 데이터 유형 정의에 지정되어 있습니다. 정의는 환경설정 파일 `/usr/dt/appconfig/types/language/xclients.dt`에 위치하고 있습니다.

```
DATA_ATTRIBUTES XWD
{
    ACTIONS                Open,Print
    ICON                   Dtxwd
    ...
}
```

XWD 데이터 유형 및 관련된 열기 및 인쇄 활동은 “데이터 유형이 데이터 파일을 활동에 연결하는 방법” 164 페이지에 설명되어 있습니다.

활동이 응용 프로그램에 대한 아이콘을 작성하는 방법

Desktop_Tools 응용 프로그램 그룹에서 Xwd 표시장치 아이콘이 있다고 가정합니다. 이 아이콘을 두 번 누르면 X 클라이언트 `xwud`가 실행됩니다. 그러나 이 아이콘은 `xwud`의 실제 실행 파일 `/usr/bin/X11/xwud`를 직접적으로 표시하지는 않습니다.

레이블이 Xwd인 표시장치 아이콘이 응용 프로그램 그룹에 나타나며, 이는 해당 디렉토리에 Xwd라는 파일이 있기 때문입니다(그림 10-3 참조). 이 파일은 동일한 이름 —Xwd로 된 기본 활동을 나타냅니다. 활동 정의에서 활동 이름은 ACTION 키워드 다음에 오는 이름입니다.

```

ACTION Xwd
{
    LABEL                Xwd Display
    TYPE                 COMMAND
    WINDOW_TYPE         NO_STDIO
    EXEC_STRING          /usr/bin/X11/xwd -noclick -in \
                        %(File)Arg_1"Xwd File To Display:"%
    DESCRIPTION         The Xwd Display (Xwd) XwdDisplay action \
                        displays an xwd file that was created using the \
                        Xwd Capture (Xwd) action. It uses \
                        the xwd command.
}

```

이 파일은 활동을 나타내므로, **활동 파일**이라고 합니다. 파일은 활동과 동일한 이름으로 된 실행 파일인 경우에 활동 파일이 됩니다. 응용 프로그램 관리자(또는 파일 관리자)에 있는 아이콘을 **활동 아이콘**이라고 하고, 이 아이콘을 두 번 누르면 응용 프로그램이 실행되므로 **응용 프로그램 아이콘**이라고도 합니다.



활동 파일을 나타내는 활동 아이콘

LABEL

그림 10-3 활동 파일을 나타내는 응용 프로그램(활동) 아이콘

응용 프로그램 관리자가 실행 파일을 발견하면, 파일 이름과 일치하는 활동이 있는지 알아 보기 위해 활동 데이터베이스를 검색합니다. 일치되는 활동이 있으면 응용 프로그램 관리자는 이 파일이 활동 파일이라는 것을 알게 됩니다.

활동 파일의 내용은 상관이 없습니다. 활동 파일에는 일반적으로 자신의 데스크탑 기능을 설명하는 주석이 들어있습니다.

주 - 활동 파일과 활동 정의 파일은 같지 않습니다. 활동 파일은 활동과 동일한 이름을 갖는 파일입니다. 응용 프로그램 아이콘을 파일 관리자 또는 응용 프로그램 관리자에 작성하는데 사용합니다. 활동 정의 파일은 활동 정의가 들어있는 `name.dt`라는 파일입니다.

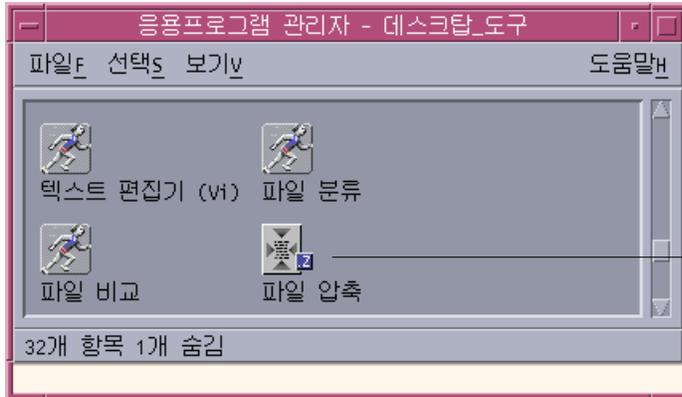
일단 데스크탑이 파일을 활동 파일이라고 결정하면, 기본 활동 정의를 사용하여 해당 활동 파일의 모양과 동작을 정의합니다.

- EXEC_STRING 필드는 응용 프로그램 아이콘의 동작을 지정합니다. Xwd 표시장치 아이콘의 경우, EXEC_STRING은 활동 아이콘이 특정 명령 행 인수와 함께 `xwud X` 클라이언트를 실행하도록 지정합니다.
- LABEL 필드는 응용 프로그램 아이콘에 대한 레이블을 지정합니다.
- DESCRIPTION 필드는 사용자가 항목 도움말을 요청할 때 표시되는 텍스트를 설명합니다.
- Xwud 응용 프로그램 아이콘은 활동 정의에 다른 이미지를 지정하기 위한 ICON 필드가 있으므로, 활동에 대해 기본 아이콘 이미지를 사용합니다.

이와는 반대로 레이블이 파일 압축인 아이콘은 기본 활동 정의에 ICON 필드를 포함하므로 다른 아이콘 이미지를 사용합니다.

예를 들어,

```
ACTION Compress
{
    LABEL                Compress File
    ICON                 Dtcmprs
    ...
}
```



ICON 필드에 의해 지정된 아이콘 이미지

그림 10-4 활동 정의에서 ICON 필드가 지정하는 아이콘 이미지

Xwud 활동 정의에는 실행될 명령어(EXEC_STRING)가 포함되어 있으므로 Xwud 활동을 **명령어 활동**이라고 합니다. 활동 정의의 TYPE 필드는 활동 유형을 정의합니다.

처음에는 Xwd 표시장치 아이콘이 Desktop_Tools 응용 프로그램 그룹에 나타납니다. 그러나 쓰기가 허용된 디렉토리에는 활동 아이콘의 추가 사본을 작성할 수 있습니다. Xwud 활동 정의가 데이터베이스의 일부이면, Xwud라고 작성된 실행 파일은 활동을 나타내는 활동 파일이 되며, 파일 관리자나 응용 프로그램 관리자의 아이콘이 활동을 실행하는데 사용될 수 있습니다.

활동이 데이터 파일을 인수로 사용하는 방법

명령어의 **인수**는 명령어가 실행되는 대상으로서 주로 파일을 말합니다. 활동은 파일 인수를 허용하도록 작성할 수 있습니다.

예를 들어, Xwud 활동의 EXEC_STRING은 파일 인수가 필수임을 지정합니다.

```
EXEC_STRING          /usr/bin/X11/xwud -noclick -in \  
                    %(File)Arg_1"Xwd File To Display:"%
```

용어 Arg는 **인수**를 의미합니다. 구문 Arg_1은 첫번째 인수를 의미하며 (File)은 활동이 이 인수를 파일로 취급함을 의미합니다.

사용자가 파일 인수를 제공하는 가장 쉬운 방법은 데이터 파일을 응용 프로그램 아이콘에 끌어다 놓는 것입니다. 데스크탑은 끌어 놓은 파일의 경로를 결정하여 명령행에서 % 기호 사이에 있는 문장(%(File)Arg_1"Xwd File To Display:"%) 대신 파일 경로를 대체합니다. 따라서 실행될 명령어는 다음과 같습니다.

```
/usr/bin/X11/xwud -noclick -in file_path
```

사용자가 응용 프로그램 아이콘을 두 번 누르면, 데스크탑은 EXEC_STRING으로부터 파일 인수가 필수인지를 판단하고, 사용자가 파일 이름 또는 경로를 입력할 수 있도록 프롬프트를 표시하는 대화 상자를 표시합니다. Xwd 활동의 경우에 프롬프트는 다음과 같습니다.

Xwd File To Display:

사용자가 제공하는 파일 이름 또는 경로가 파일 인수로 사용됩니다.

활동의 다른 용도

활동은 응용 프로그램을 시작하는 것 이외에도 데스크탑 전체에 걸쳐서 다음 항목의 기능을 작성하는 데에도 사용됩니다.

- **프론트 패널**

프론트 패널 제어도구의 정의에는 사용자가 제어도구를 누르거나 파일을 놓았을 때 실행되는 활동을 지정하는 필드가 포함됩니다. 자세한 내용은 “프론트 패널 제어 도구 정의” 247 페이지를 참조하십시오.

- **메뉴**

윈도우 및 작업 공간 메뉴 정의 구문을 통해 메뉴 항목이 실행할 활동을 지정할 수 있습니다. 자세한 내용은 “작업 공간 관리자 메뉴” 264 페이지와 dtwmrc(4) 매뉴얼 페이지를 참조하십시오.

- **응용 프로그램 간의 통신**

응용 프로그램은 ToolTalk 메시지(TT_MSG)라는 특별한 유형의 활동을 사용하여 정보를 주고 받도록 설계할 수 있습니다. TT_MSG 활동은 데스크탑에 대한 개발자 환경 설명서에 설명되어 있습니다.

데이터 유형 소개

사용자가 새로운 데이터 파일을 작성하면, 파일 관리자에 있는 파일 아이콘의 모양과 동작은 사용자가 작성한 데이터 파일의 유형에 따라 달라집니다. 파일과 디렉토리에 대한 사용자 정의 모양과 동작을 작성하는 이러한 기능은 데스크탑의 데이터 유형 메커니즘에 의해 제공됩니다.

데이터 유형이란?

데이터 유형은 데스크탑 데이터베이스에 정의된 구성입니다. 예를 들어, 다음은 XWD 데이터 유형에 대한 정의입니다. 정의는 환경설정 파일 /usr/dt/appconfig/types/language/xclients.dt에 있습니다.

```

DATA_ATTRIBUTES XWD
{
    ACTIONS          Open,Print
    ICON             Dtxwd
    NAME_TEMPLATE    %s.xwd
    MIME_TYPE        application/octet-stream
    SUNV3_TYPE       xwd-file
    DESCRIPTION      This file contains a graphics image in the XWD \
                    format. These files are typically created by \
                    taking snapshots of windows using the XwdCapture \
                    action. Its data type is named XWD. XWD files \
                    have names ending with '.xwd' or '.wd'.
}

DATA_CRITERIA XWD1
{
    DATA_ATTRIBUTES_NAME    XWD
    MODE                     f
    NAME_PATTERN              *.xwd
}

DATA_CRITERIA XWD2
{
    DATA_ATTRIBUTES_NAME    XWD
    MODE                     f
    NAME_PATTERN              *.wd
}

```

모든 데이터 유형은 다음과 같은 두 부분으로 구성되어 있습니다.

DATA_ATTRIBUTES—데이터 유형의 모양 및 동작을 설명합니다.

DATA_CRITERIA—파일이 해당 데이터 유형에 속하도록 범주화하는 (이름 지정 또는 내용) 규칙을 지정합니다.

DATA_ATTRIBUTES_NAME 필드는 기준을 속성에 연결합니다.

DATA_ATTRIBUTE에는 여러 개의 **DATA_CRITERIA**가 있을 수 있습니다. 예를 들어, XWD 데이터 유형은 서로 다른 두 가지 이름 지정 기준 (**NAME_PATTERN**)—.xwd 또는 .wd로 끝나는 이름을 지정하는 두 가지 기준이 있습니다.

데이터 유형이 데이터 파일을 활동에 연결하는 방법

XWD 데이터 유형에 대해 생각해 봅시다. 사용자는 파일에 두 개의 파일 이름 접미사(확장자), .xwd 또는 .wd 중 하나를 사용하여 XWD 유형 파일을 작성합니다. 데스크탑은 해당 유형으로 파일을 지정하기 위해 파일 이름을 기준으로 사용합니다.

XWD 데이터 유형은 해당 데이터 유형의 각 파일에 다음 항목을 제공합니다.

- 사용자가 데이터 유형을 인식하는데 도움을 주는 고유한 아이콘 이미지.
- 데이터 유형을 설명하는 항목 도움말

- 열기 및 인쇄 활동을 포함한 파일 관리자에 있는 사용자 정의된 선택 메뉴. XWD 파일에 대한 열기 활동은 Xwud 활동을 실행합니다.

선택 메뉴에서 활동 실행

파일 관리자의 선택 메뉴는 파일 또는 디렉토리가 선택된 경우에만 활성화됩니다. 선택 메뉴 하단의 명령어는 데이터 유형에 따라 달라집니다. 예를 들어, XWD 파일이 선택되면 선택 메뉴에는 열기와 인쇄 항목이 포함됩니다.

데이터 유형 정의의 ACTIONS 필드는 명령어가 데이터 유형의 선택 메뉴 하단에 추가되도록 지정합니다.

```
DATA_ATTRIBUTES XWD { ACTIONS Open,Print ... }
```

선택 메뉴의 내용은 데이터 유형에 따라 달라집니다. 그러나 서로 다른 여러 데이터 유형이 열기 활동을 제공합니다. 즉, 파일 관리자에서 특정 데이터 유형의 파일을 선택하고 선택 메뉴를 표시하면 열기 명령어를 볼 수 있습니다.

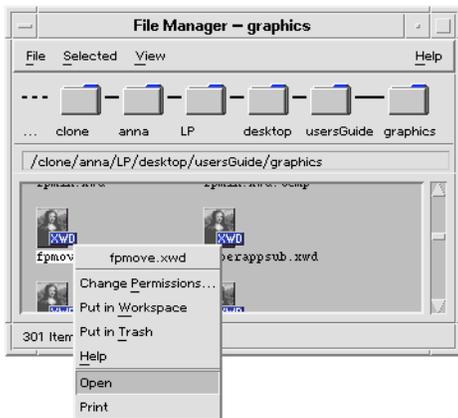


그림 10-5 XWD 파일의 선택 메뉴

열기 활동은 일반적으로 데이터 파일과 관련된 응용 프로그램을 실행합니다. 예를 들어, XWD 파일을 열면 Xwud 활동이 실행되며, 이 활동은 xwud X 클라이언트를 실행하고 화면 이미지를 표시합니다. 즉, XWD 데이터 유형에서 열기 활동은 Xwud 활동과 동의어입니다. 마찬가지로 데이터 유형 TEXTFILE의 파일을 열면 문서 편집기가 실행되고, BM(비트맵) 또는 PM(픽스맵) 파일을 열면 아이콘 편집기가 실행됩니다.

활동 정의의 다음 두 가지 기능을 사용하여 서로 다른 작업을 수행하는 다양한 열기 활동을 작성할 수 있습니다.

- 활동 매핑

활동 매핑을 통해 명령어를 직접 실행하지 않고 다른 활동을 실행하는 활동을 작성할 수 있습니다. 예를 들어, Xwud 활동에 매핑(실행)되는 열기 활동을 작성할 수 있습니다.

■ 활동에 대한 데이터 유형 제한사항

활동 정의에는 활동을 특정 데이터 유형으로 제한하는 ARG_TYPE 필드가 포함될 수 있습니다. 예를 들어, Xwud 활동에 매핑되는 열기 활동이 XWD 데이터 유형 파일에만 적용되도록 지정할 수 있습니다.

다음은 열기 활동을 XWD 데이터 유형의 Xwud 활동에 매핑하는 활동 정의입니다. 이 정의는 데이터베이스 환경설정 파일 /usr/dt/appconfig/types/C/xclients.dt에 있습니다.

```
ACTION Open
{
    LABEL                Open
    ARG_TYPE              XWD
    TYPE                  MAP
    MAP_ACTION            Xwud
}
```

TYPE 필드는 이 활동이 매핑 활동임을 지정합니다. MAP_ACTION 필드는 이 활동이 Xwud 활동을 실행하도록 지정합니다. ARG_TYPE 필드는 이 활동이 데이터 유형이 XWD인 파일에만 적용되도록 지정합니다.

열기 활동의 이전 정의와 데이터베이스 파일 /usr/dt/appconfig/types/C/dt.dt에 있는 다음 정의를 비교하십시오.

```
ACTION Open
{
    LABEL                Open
    ARG_TYPE              BM
    TYPE                  MAP
    MAP_ACTION            Dticon
}
```

이 정의는 데이터 유형(ARG_TYPE)이 BM(비트맵)인 파일에 적용됩니다. 정의는 열기 활동을 아이콘 편집기를 실행하는 Dticon 활동에 매핑합니다.

데이터 유형의 두 번 누르기 동작 정의

데이터 유형의 두 번 누르기 동작은 ACTIONS 필드의 첫번째 항목에 의해 정의됩니다. 예를 들어, XWD 데이터 유형에 대한 두 번 누르기 동작은 열기 활동을 실행한 다음 Xwud 활동을 실행합니다.

데이터 파일을 활동 아이콘에 놓기

사용자가 데이터 파일을 활동 아이콘에 놓으면 시스템은 이 데이터 파일을 활동에 대한 인수로 사용하여 활동을 실행합니다("활동이 데이터 파일을 인수로 사용하는 방법" 162 페이지를 참조하십시오).

예를 들어, XWD 파일을 Xwd 표시장치 아이콘에 놓으면, Xwud 활동이 데이터 파일 인수를 사용하여 실행됩니다. 이 작업은 해당 데이터 파일과 함께 xwud X 클라이언트를 실행합니다.

데이터 유형에 대한 데스크탑 인쇄 작성

데스크탑 인쇄를 통해 다음과 같은 방식으로 데이터 파일을 인쇄할 수 있습니다.

- 가능한 경우, 파일 관리자의 선택 메뉴에서 인쇄 명령어 사용.
- 데이터 파일을 데스크탑 프린터 놓기 영역(프론트 패널 프린터 제어 도구 또는 인쇄 관리자의 프린터 아이콘)에 놓기.

데스크탑 인쇄 외에도, 여러 응용 프로그램이 응용 프로그램내에서 인쇄할 수 있는 방법을 제공합니다.

데스크탑 인쇄는 인쇄라는 활동을 사용합니다. 열기와 마찬가지로 인쇄는 서로 다른 많은 데이터 유형에 사용되는 활동 이름입니다. 따라서, 인쇄 활동은 활동 매핑과 ARG_TYPE 필드를 사용하여 각 데이터 유형에 대한 인쇄를 사용자 정의합니다.

예를 들어, 다음은 XWD 데이터 유형에 대한 인쇄 활동입니다. 정의는 /usr/dt/appconfig/types/language/xclients.dt에 있습니다.

```
ACTION Print
{
    LABEL                Print
    ARG_TYPE              XWD
    TYPE                 MAP
    AP_ACTION            NoPrint
}
```

XWD 파일에 대해 특정한 인쇄 활동은 NoPrint 활동에 매핑됩니다. NoPrint는 /usr/dt/appconfig/types/language/dt.dt에 정의된 특별한 활동입니다. NoPrint 활동은 사용자에게 이 데이터 유형은 인쇄할 수 없음을 알려주는 대화 상자를 표시합니다.

XWD 인쇄 활동과 PCL 파일에 대한 다음 인쇄 활동을 비교하십시오.

```
ACTION Print
{
    LABEL                Print
    ARG_TYPE              PCL
    TYPE                 MAP
    MAP_ACTION           PrintRaw
}
```

환경설정 파일 /usr/dt/appconfig/types/language/print.dt에 정의된 PrintRaw 활동에는 PCL 파일 인쇄를 위한 명령행이 들어 있습니다.

```
ACTION PrintRaw
{
    TYPE                 COMMAND
```

```
WINDOW_TYPE          NO_STDIO
EXEC_STRING           /usr/dt/bin/dt1p -w %(File)Arg_1%
}
```

활동 만들기를 사용한 활동 및 데이터 유형 작성

활동 만들기는 다음 항목을 작성하는 도구입니다.

- 응용 프로그램을 시작하는 활동
- 응용 프로그램의 데이터 파일에 대한 하나 이상의 데이터 유형
- 응용 프로그램의 데이터 파일을 열고 인쇄하는 활동

활동 만들기는 또한 응용 프로그램을 실행하기 위한 간단한 활동과 셸 스크립트를 작성하는 데에도 유용하게 사용할 수 있습니다.

- “활동 만들기의 기능” 169 페이지
- “활동 만들기의 제한” 170 페이지
- “활동 만들기로 응용 프로그램에 대한 활동 및 데이터 유형 작성” 171 페이지

관련 정보는 dtcreate(1X) 매뉴얼 페이지를 참조하십시오.

활동 만들기의 기능

활동 만들기에는 활동 및 관련 데이터 유형을 작성하기 위한 일련의 대화 상자와 기본 원도우가 포함됩니다.

활동 만들기는 다음과 같은 작업을 수행합니다.

- 명령어를 실행하는 활동 정의를 작성합니다.
- 파일 `HomeDirectory/.dt/types/action_name.dt`를 작성합니다. 이 파일은 응용 프로그램에 대해 작성된 활동 및 데이터 유형 정의를 저장합니다.
- **활동 파일**을 사용자의 홈 디렉토리에 작성합니다. 활동 파일은 활동과 동일한 이름을 갖는 실행 파일입니다.

파일 관리자에서 활동을 파일로 나타내는 것을 두 번 누르면 응용 프로그램이 시작되기 때문에 **응용 프로그램 아이콘**이라고 합니다.

선택적으로, 활동을 작성할 때 가져다 놓을 수 있는 데이터 유형을 지정하여 활동 아이콘에 놓기 영역을 만들 수 있습니다.

- 응용 프로그램 데이터 파일에 대해 하나 이상의 데이터 유형을 작성합니다(선택사항).
- 각 데이터 유형에 대해 열기 활동을 작성합니다.
- 각 데이터 유형에 대해 인쇄 활동을 작성합니다(선택사항).
- 활동 및 데이터 유형의 데이터베이스를 재로드합니다. 이것은 활동 및 데이터 유형을 즉시 반영합니다.

활동 만들기의 제한

활동 만들기는 응용 프로그램을 실행하기 위한 활동 및 데이터 유형을 작성하도록 설계되어 있습니다. 그러나 활동 및 데이터 유형은 매우 유연하며 수동으로 정의를 작성했을 경우에만 액세스할 수 있는 추가 기능을 포함합니다.

자세한 내용은 다음 장을 참조하십시오.

- 제 12 장
- 제 13 장

활동 제한

다음 조건 중 하나라도 참인 경우에는 응용 프로그램에 대한 활동을 작성하기 위해 활동 만들기를 사용할 수 없습니다.

- 명령행에 비파일 인수(매개변수)가 필요한 경우.
예를 들어, 활동 만들기를 사용하여 다음 명령어에 대한 활동을 작성할 수 없습니다.

```
lp -ddevice filename
```

여기서 사용자는 명령어를 실행할 때마다 *device*를 제공해야 합니다.

- 응용 프로그램 아이콘의 레이블이 활동 이름의 레이블과 달라야 하는 경우.
예를 들어, 기존 활동의 지역 언어 버전을 제공하기 위해 활동 만들기를 사용할 수 없습니다.
- 활동이 활동 데이터베이스의 고급 기능을 필요로 하는 경우.
이러한 고급 기능의 예로는 다음 작업을 수행하는 활동을 들 수 있습니다.
 - 활동 정의에서 원격인 시스템에서 명령어를 시작합니다.
 - 다른 활동을 호출하는 경우.
 - 다른 사용자(예를 들어 슈퍼유저)로 실행해야 하는 경우.

- "매핑" 기능을 폭넓게 사용하는 경우.
- 활동에 제공된 파일 인수의 수에 따라 동작이 달라지는 경우.

데이터 유형 제한

다음 조건 중 하나라도 참인 경우에는 응용 프로그램에 대한 데이터 유형을 작성하기 위해 활동 만들기를 사용할 수 없습니다.

- 데이터 유형이 열기 및 인쇄 외에 관련된 추가 활동을 가져야 하는 경우.
- 데이터 유형에 대한 열기 활동이 활동의 명령어가 아닌 경우.
예를 들어, 응용 프로그램의 응용 프로그램 그룹을 표시하는 디렉토리에 대해 고유한 아이콘을 제공하는 데이터 유형을 작성하기 위해 활동 만들기를 사용할 수 없습니다.

활동 만들기로 응용 프로그램에 대한 활동 및 데이터 유형 작성

활동 만들기를 실행하기 전에 응용 프로그램에 대해 알아야 할 사항들이 있습니다.

- 응용 프로그램을 시작하기 위한 명령행
명령행에 필수 파일 인수나 선택적 파일 인수가 포함되는지 또는 아무런 파일 인수도 포함되지 않는지 알아야 합니다.
응용 프로그램이 비파일 인수를 요구할 경우에는 활동을 작성하기 위해 활동 만들기를 사용할 수 없습니다.
- 응용 프로그램이 허용하는 데이터 파일의 유형
일부 응용 프로그램은 한 가지 데이터 유형만을 허용합니다. 복수의 데이터 유형을 허용할 수 있는 응용 프로그램(예를 들어, ASCII 편집기 또는 그래픽 편집기)도 있습니다.
- 응용 프로그램이 데이터 파일을 식별하는 방법
이 방법은 이름 지정 규약(예를 들어, .doc로 끝나는 파일 이름)이 될 수도 있으며, 파일 내용에 따라 달라질 수 있습니다. 응용 프로그램이 파일 이름 규약을 사용하지 않는 경우, 활동 아이콘에 대해 하나를 설정할 수 있습니다.
- 선택사항: 파일을 인쇄하는 명령행

▼ 응용 프로그램의 활동을 작성하는 방법

1. 데스크탑_응용프로그램에 있는 활동 만들기 (활동 생성)을 두 번 누르십시오.



활동 작성 두 번 누르기

그림 11-1 응용프로그램 관리자에 있는 활동 만들기 (활동 생성) 아이콘

주 활동 만들기 윈도우입니다.

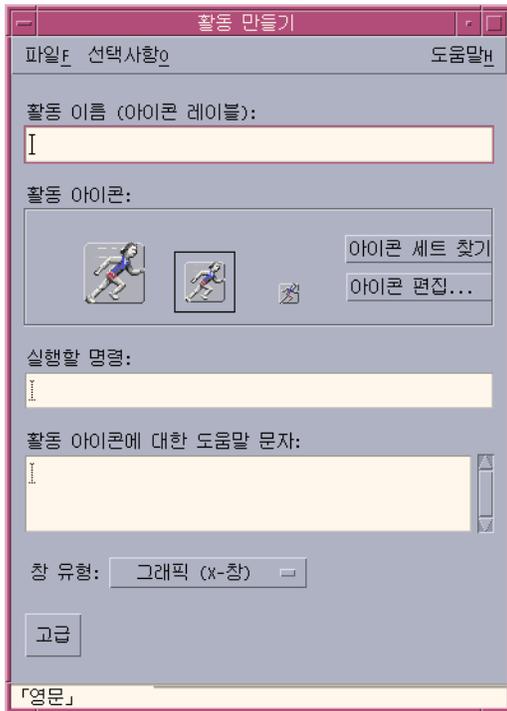


그림 11-2 활동 만들기 기본 윈도우

2. 활동 아이콘의 레이블이 될 이름을 활동 이름 텍스트 필드에 입력하십시오.
3. 활동 아이콘 제어도구를 사용하여 응용 프로그램에 대한 아이콘을 지정하십시오. 처음에는 기본 아이콘이 표시됩니다.
 - 기존의 아이콘 중 다른 것을 선택하려면 아이콘 세트 찾기를 눌러 세트 찾기 대화 상자를 표시하십시오. “아이콘 세트 찾기 대화 상자를 사용하여 아이콘 지정” 180 페이지를 참조하십시오.
 - 새 아이콘을 작성하려면, 아이콘 편집을 선택하여 아이콘 편집기를 실행하십시오.
4. 실행할 명령어 텍스트 필드에 명령어를 입력하여 응용 프로그램을 시작하십시오. 파일 인수에 대해서는 구문 $\$n$ 을 사용하십시오. 그 예는 다음과 같습니다.

```
emacs
  bitmap $1
  diff $1 $2
  lp -oraw $1
```

명령행에 파일 인수($\$n$)가 포함되어 있으면 활동 아이콘이 파일에 대한 놓기 영역이 됩니다.

셸의 사용을 명시적으로 지정하지 않으면 명령행은 셸에 전달되지 않습니다. 예를 들어, 다음 명령행은 셸 처리를 사용합니다.

```
/bin/sh -c 'ps | lp'  
/bin/sh -c 'spell $1 | more'
```

5. 활동 아이콘에 대한 항목별 도움말 텍스트를 활동 아이콘에 대한 도움말 텍스트 텍스트 필드에 입력하십시오.

텍스트는 텍스트 필드에서 자동적으로 줄이 바뀝니다. 그러나 이러한 줄 바꿈이 온라인에서는 보존되지 않을 것입니다. 줄이 바뀌도록 지정하려면 \n을 사용하십시오.

6. 창 유형 옵션 메뉴에서 활동에 필요한 윈도우 지원사항을 선택하십시오.

그래픽(X 윈도우)—응용 프로그램이 자체 윈도우를 작성합니다.

터미널(자동 닫기)—응용 프로그램이 사용자가 해당 프로그램을 종료할 때 자동으로 닫히는 터미널 에뮬레이터 윈도우에서 실행됩니다.

터미널(수동 닫기)—응용 프로그램이 사용자가 명시적으로 닫을 때까지 열린 상태로 있는 터미널 에뮬레이터 윈도우에서 실행됩니다.

출력 없음—응용 프로그램이 표시장치에 출력하지 않습니다.

7. 다음과 같이 진행하십시오.

- 응용 프로그램이 데이터 파일을 가지고 있고, 응용 프로그램에 대해 하나 이상의 데이터 유형을 작성하려면, 다음 절 “응용 프로그램에 대해 하나 이상의 데이터 유형을 작성하는 방법” 174 페이지를 참조하십시오.
- 데이터 유형을 작성할 필요가 없으면, 파일 메뉴의 저장을 선택하여 활동을 저장하십시오. 그런 다음, 홈 디렉토리의 활동 아이콘을 두 번 눌러 새 활동을 검사하십시오.

▼ 응용 프로그램에 대해 하나 이상의 데이터 유형을 작성하는 방법

1. 앞 절, “응용 프로그램의 활동을 작성하는 방법” 171 페이지에 설명된 절차에 따라 응용 프로그램의 활동을 정의하십시오.
2. 활동 만들기 윈도우에서 고급 버튼을 눌러 윈도우를 확장하십시오.

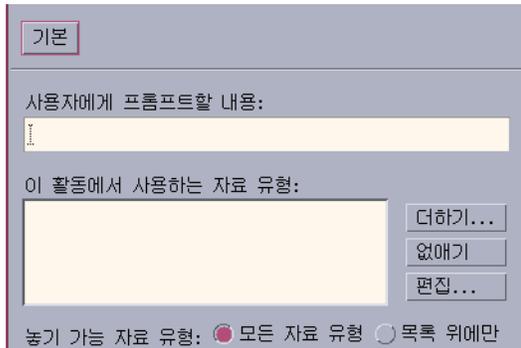


그림 11-3 활동 만들기 기본 윈도우에 있는 고급 기능

3. 아이콘을 두 번 눌렀을 때 응용 프로그램 아이콘이 파일 인수에 대한 프롬프트를 표시하도록 하려면, "활동이 열릴 때, 다음 내용 질문" 텍스트 필드에 프롬프트 텍스트를 입력하십시오.

이 텍스트 필드에 다음 지침을 적용하십시오.

- 응용 프로그램 명령행에 필수 파일 인수가 있으면 이 필드를 사용해야 합니다.
- 명령행이 파일 인수를 포함하지 않으면 이 필드를 공백으로 두어야 합니다.
- 응용 프로그램 명령행의 파일 인수가 선택적인 경우 선택할 수 있습니다. 프롬프트 텍스트를 제공하면, 활동 아이콘을 두 번 눌렀을 때 파일에 대한 프롬프트가 표시됩니다. 프롬프트 텍스트를 제공하지 않으면, 활동은 파일 인수가 빈 문자열인 상태로 실행됩니다.

4. 활동이 인수로서 허용할 파일 유형을 지정하십시오.

- 활동이 모든 자료 유형을 허용할 수 있는 경우에는 모든 자료 유형을 선택하십시오.
- 활동이 응용 프로그램에 대해 작성된 데이터 유형만을 허용하는 경우 목록 위에만 선택하십시오.

처음에는 이 활동을 사용하는 데이터 유형 목록이 비어 있습니다. 응용 프로그램에 대한 데이터 유형을 작성하면 목록에 추가됩니다.

5. 이 활동을 사용하는 데이터 유형 목록 상자 옆의 추가를 눌러 자료 유형 추가 대화 상자를 표시하십시오.

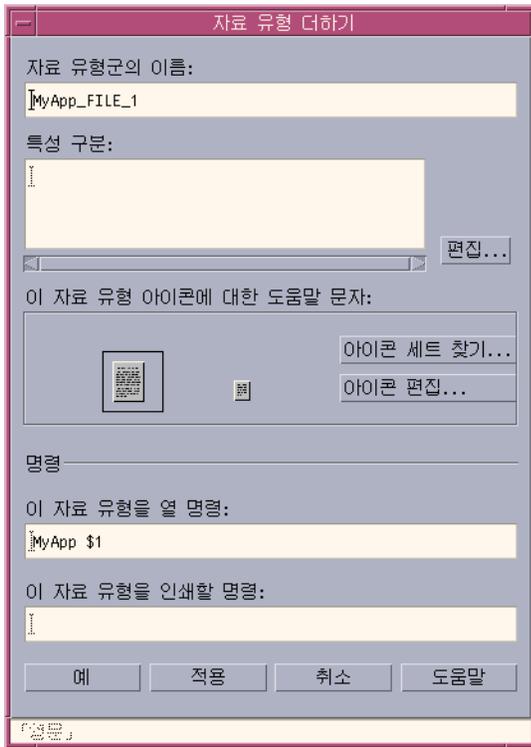


그림 11-4 활동 만들기의 자료 유형 추가 대화 상자

6. **선택사항:** 기본 데이터 유형 이름을 사용하지 않으려면 데이터 유형에 대한 새 이름을 자료 유형군 이름 텍스트 필드에 입력하십시오.

이름에는 공백이 포함될 수 없습니다. 데이터 유형 이름을 응용 프로그램 사용자는 볼 수 없습니다. 이 이름은 데이터 유형 정의를 식별하기 위해 활동/데이터 유형 데이터베이스에서 사용됩니다.

7. 특성 구분 상자 옆의 편집 버튼을 눌러 특성 구분 대화 상자를 표시하십시오.

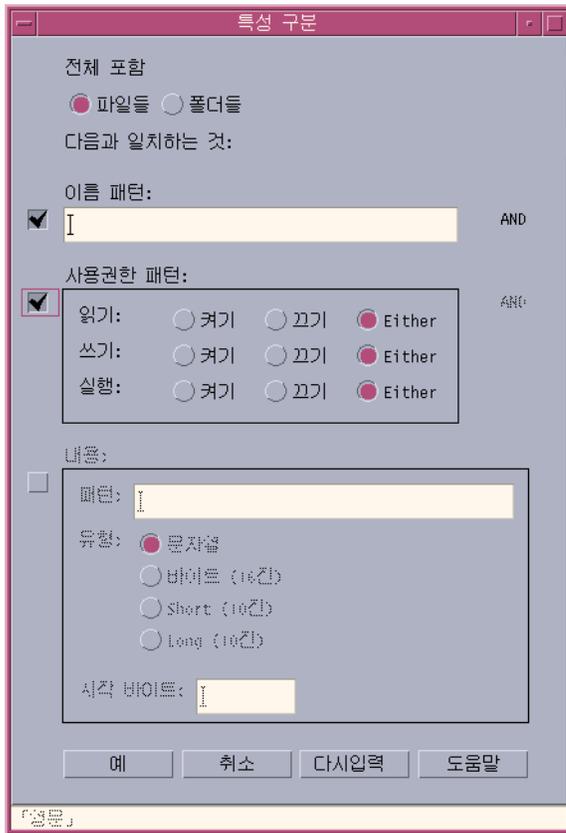


그림 11-5 활동 만들기의 특성 식별 대화 상자

데이터 유형의 특성은 데이터 유형을 다른 데이터 유형과 구별하기 위해 사용하는 기준입니다. 다음 기준 중에서 하나 이상을 선택할 수 있습니다.

파일 또는 폴더—데이터 유형이 파일 또는 폴더에만 적용됩니다.

이름 패턴—파일 이름에 기초한 데이터 유형입니다.

사용 권한 패턴—읽기, 쓰기, 실행 권한

내용—파일의 지정된 부분의 내용

8. 데이터 유형이 파일을 나타내는지 폴더를 나타내는지 선택하십시오.

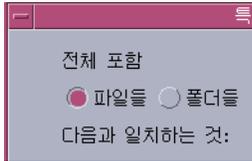


그림 11-6 데이터 유형에 대한 파일 또는 디렉토리 특성 지정

9. 이름별로 데이터 유형을 지정하는 경우, 이름 패턴 확인란을 선택하고 텍스트 필드를 채워주세요.



그림 11-7 데이터 유형에 대한 파일 이름 특성 지정

- *와 ?를 와일드카드로 사용할 수 있습니다.
- *—일련의 문자에 해당합니다.
- ?—단일 문자에 해당합니다.

10. 데이터 유형 지정이 사용 권한에 따라 다른 경우에는 사용 권한 패턴 확인란을 선택한 다음 해당 데이터 유형에 대한 사용 권한을 선택하십시오.



그림 11-8 데이터 유형에 대한 사용 권한 특성 지정

- On—파일이 지정된 사용 권한을 가져야 합니다.
- Off—파일이 지정된 사용 권한을 갖지 않아야 합니다.
- Either—어떤 사용 권한을 지정하든 상관 없습니다.

11. 데이터 유형 지정이 내용에 따라 다른 경우, 내용 확인란을 선택한 다음 찾을 패턴과 내용 유형을 입력하십시오. 선택적으로 검색을 시작할 바이트 위치를 입력할 수 있습니다.

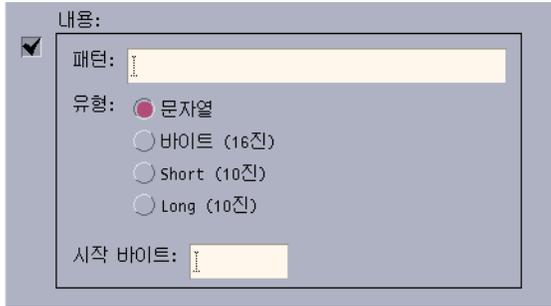


그림 11-9 데이터 유형에 대한 내용 특성 지정

주 - 내용 기반 데이터 유형을 지정하면 시스템 성능에 영향을 줄 수 있습니다.

12. 확인을 눌러 특성 구분 대화 상자를 닫으십시오.

다음 코드를 사용하여 특성 구분 필드에 특성이 표시됩니다.

d—디렉토리

r—파일이 읽기 권한을 가지고 있습니다.

w—파일이 쓰기 권한을 가지고 있습니다.

x—파일이 실행 권한을 가지고 있습니다.

!—논리 연산자 NOT

&—논리 연산자 AND

13. 데이터 파일의 도움말 텍스트를 도움말 텍스트 필드에 입력하십시오.

14. 데이터 유형 아이콘 제어도구를 사용하여 데이터 파일에 대한 아이콘을 지정하십시오. 처음에는 기본 아이콘이 표시됩니다.

- 기존의 아이콘 중 다른 것을 선택하려면 아이콘 세트 찾기를 눌러 세트 찾기 대화 상자를 표시하십시오. “아이콘 세트 찾기 대화 상자를 사용하여 아이콘 지정” 180 페이지를 참조하십시오.

- 새 아이콘을 작성하려면, 아이콘 편집을 눌러 아이콘 편집기를 실행하십시오.

15. 이 데이터 유형을 열기 위한 명령어 텍스트 필드의 명령어를 확인하십시오. 이 명령어는 사용자가 데이터 파일을 두 번 누르면 실행될 명령어입니다.

16. 선택사항: 응용 프로그램이 데이터 파일을 명령행에서 인쇄하기 위한 인쇄 명령어를 제공하는 경우, 이 명령어를 파일 인수에 대해 구문 \$n을 사용하여 이 자료 유형을 인쇄할 명령 텍스트 필드에 입력하십시오.

17. 다음 작업 중 하나를 수행하여 데이터 유형 정의를 저장하십시오.

- 데이터 유형을 저장하고 자료 유형 추가 대화 상자를 닫으려면 확인을 누르십시오.

- 자료 유형 추가 대화 상자를 닫지 않은 상태에서 데이터 유형을 저장하려면 적용을 누르십시오. 그러면 활동에 대해 다른 데이터 유형 정의를 바로 진행할 수 있습니다.

아이콘 세트 찾기 대화 상자를 사용하여 아이콘 지정

아이콘 세트 찾기 대화 상자는 활동 만들기 기본 윈도우 또는 자료 유형 추가 윈도우의 아이콘 세트 찾기를 누르면 표시됩니다. 대화 상자를 사용하여 활동 또는 데이터 유형에 사용될 아이콘을 지정하십시오.

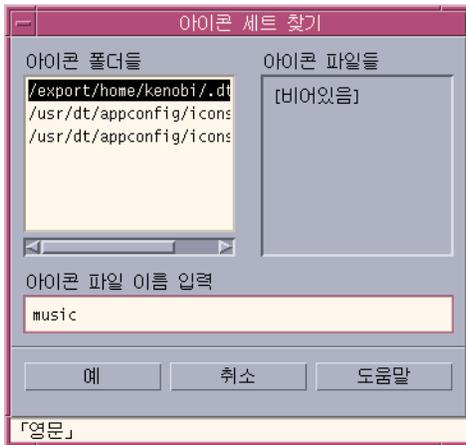


그림 11-10 아이콘 세트 찾기 대화 상자

아이콘 세트 찾기 대화 상자를 통해 다음 위치에 있는 아이콘 이미지 파일 세트를 지정할 수 있습니다.

- 아이콘 검색 경로상의 디렉토리. 아이콘 폴더들 목록에는 아이콘 검색 경로상의 모든 디렉토리가 포함됩니다.
- dtappintegrate를 사용하여 데스크탑과 통합될 등록 패키지. 이러한 아이콘은 아직 아이콘 검색 경로의 디렉토리에 없지만, dtappintegrate에 의해 해당 디렉토리에 위치하게 됩니다.

주 - 활동 만들기를 사용하여 작성된 활동 및 데이터 유형 정의는 아이콘 파일에 대한 기본 이름(파일 이름에서 크기 및 유형에 대한 파일 이름 접미사를 뺀 이름)을 기록합니다. 활동 만들기에 의해 작성된 활동 및 데이터 유형은 결과적으로 아이콘 검색 경로의 디렉토리에 있어야 합니다.

▼ 아이콘 검색 경로상에 위치하는 아이콘 세트를 지정하는 방법

1. 아이콘 세트 찾기 대화 상자의 아이콘 폴더들 목록에서 아이콘을 포함하는 폴더 경로를 두 번 누르십시오.
아이콘 파일 목록에 해당 폴더에 있는 모든 아이콘 파일이 표시됩니다.
2. 아이콘 파일 목록에서 사용하려는 아이콘을 누르십시오.
그러면 아이콘 파일의 기본 이름이 아이콘 파일 이름 입력 텍스트 필드에 놓이게 됩니다.
3. 확인을 누르십시오.

▼ 등록 패키지의 아이콘을 지정하는 방법

사용자가 등록 패키지를 작성하는 시스템 관리자 또는 프로그래머인 경우, 아이콘 이미지 파일은 처음에 다음과 같은 등록 패키지의 디렉토리에 위치합니다.

```
app_root/dt/appconfig/icons/language
```

dtappintegrate로 등록한 후, 아이콘 파일은 아이콘 검색 경로상에 있는 */etc/dt/appconfig/icons/language*로 복사됩니다.

다음 절차에 따라 등록 패키지의 일부인 아이콘을 지정하십시오.

1. 아이콘 세트 찾기 대화 상자의 아이콘 파일 이름 입력 텍스트 필드에 아이콘 파일의 기본 이름을 입력하십시오.
2. 확인을 누르십시오.
활동 만들기는 아이콘을 아이콘 검색 경로의 디렉토리에서 찾을 수 없다고 알리는 대화 상자를 표시합니다.
3. 표시되는 정보 대화 상자에서 이름 확인을 선택하십시오.

수동으로 활동 작성

활동은 다음과 같은 두 가지 방법으로 작성할 수 있습니다.

- 활동 만들기 데스크탑 응용 프로그램을 사용하는 방법
- 활동 정의를 수동으로 작성하는 방법

활동을 수동으로 작성하려면 데이터베이스 파일을 편집해야 합니다. 이 장에서는 활동 정의를 수동으로 작성하는 방법을 설명합니다.

- “활동을 수동으로 작성해야 하는 이유” 183 페이지
- “수동으로 활동 작성: 일반적 단계” 185 페이지
- “COMMAND 활동에 대한 실행 문자열 구축” 193 페이지
- “COMMAND 활동을 위한 윈도우 지원 및 터미널 애플레이터” 199 페이지
- “특정 인수에 대한 활동 제한” 201 페이지
- “원격 시스템에서 응용 프로그램을 실행하는 활동 작성” 203 페이지
- “활동 및 데이터 유형 정의에서 변수 사용” 204 페이지
- “명령행에서 활동 호출” 205 페이지
- “지역화된 활동 작성” 207 페이지
- “ToolTalk 응용 프로그램에 대한 활동 작성” 208 페이지
- 활동의 소개를 보려면 제 10 장을 참조하십시오.
- 활동 만들기에 관한 자세한 내용은 제 11 장을 참조하십시오.
- 활동 정의에 대한 참조 정보는 dtactionfile(4) 매뉴얼 페이지를 참조하십시오.

활동을 수동으로 작성해야 하는 이유

활동에는 세 가지 기본 유형이 있습니다.

- COMMAND
- MAP
- TT_MSG

활동 만들기 도구는 특정 유형의 COMMAND 및 MAP 활동을 작성하도록 설계되어 있습니다. 모든 TT_MSG 활동은 수동으로 작성해야 합니다.

자세한 내용은 “활동 만들기의 제한” 170 페이지를 참조하십시오.

COMMAND 활동

command 활동은 응용 프로그램이나 유틸리티를 시작하는 명령어를 실행하고, 셸 스크립트를 실행하거나 운영 체제 명령어를 실행합니다. 활동의 정의에는 실행될 명령어 (EXEC_STRING)가 포함됩니다.

활동 만들기 도구는 가장 일반적인 유형의 명령어 활동을 작성하는데 사용할 수 있습니다. 그러나 활동을 수동으로 작성해야 하는 경우도 있습니다. 예를 들어, 활동이 다음 사항을 지정하는 경우에는 COMMAND 활동을 수동으로 작성해야 합니다.

- 각 인수마다 프롬프트가 다른 여러 파일 인수.
- 활동 호출—다른 활동을 호출하는 활동의 기능
- 인수 개수 의존 동작—파일 인수의 수가 달라지면 동작이 달라지는 활동을 작성하는 기능
- 원격 실행 호스트—응용 프로그램을 활동 정의가 있는 시스템이 아닌 다른 시스템에서 실행하는 기능
- 사용자 변경—다른 사용자로서 활동을 실행하는 기능(예: 루트 암호에 대한 프롬프트를 표시하고 루트로 실행).

MAP 활동

map 활동은 명령어나 ToolTalk 메시지를 직접 지정하는 활동이 아닌 다른 활동으로 "매핑"되는 활동입니다.

매핑은 활동에 대해 다른 이름을 지정하는 기능을 제공합니다. 예를 들어, IconEditor라는 내장 명령어 활동은 아이콘 편집기를 시작합니다. 데이터베이스에는 비트맵 및 픽스맵 파일의 정의가 (ARG_TYPE 필드에 의해) 제한된, IconEditor 활동에 대응되는 열기 활동도 포함되어 있습니다. 이것은 사용자가 파일 관리자에서 비트맵 또는 픽스맵을 선택한 후 선택 메뉴에서 열기를 선택하여 아이콘 편집기를 시작할 수 있게 합니다.

활동 만들기는 열기 및 인쇄 활동에 제한된 매핑을 제공합니다.그 외의 다른 매핑 활동은 수동으로 작성해야 합니다.

TT_MSG(ToolTalk 메시지) 활동

TT_MSG 활동은 ToolTalk 메시지를 전송합니다. 모든 TT_MSG 활동은 수동으로 작성해야 합니다.

수동으로 활동 작성: 일반적 단계

이 절에서는 활동 정의에 대한 환경설정 파일을 작성하는 방법을 설명합니다.

활동의 환경설정 파일

활동 정의를 포함하는 환경설정 파일은 다음과 같은 요구사항을 만족해야 합니다.

- 파일은 명명규약에 따른 *name*.dt를 사용해야 합니다.
- 파일은 데이터베이스 (활동 및 데이터 유형) 검색 경로에 위치해야 합니다. 기본 검색 경로는 다음과 같습니다.

개인 활동—*HomeDirectory/.dt/types*

시스템 전체 활동—*/etc/dt/appconfig/types/language*

내장 활동—*/usr/dt/appconfig/types/language*. 사용자는 이 디렉토리를 사용할 수 없습니다.

활동 및 데이터 유형 검색 경로에 대한 내용은 “검색 경로 값 설정” 146 페이지를 참조하십시오.

▼ 활동을 수동으로 작성하는 방법

1. 기존 데이터베이스 파일을 열거나 새로 만드십시오.
이전 절 “활동의 환경설정 파일” 185 페이지를 참조하십시오.
2. 다음 구문을 사용하여 활동 정의를 작성하십시오.

```
ACTION action_name
{
    TYPE                action_type
    action_field      ...
}
```

설명:

action_name—활동을 실행하기 위해 사용된 이름.

action_type—COMMAND(기본), MAP 또는 TT_MSG.

action_field—이 유형의 활동에 대한 필수 또는 선택적 필드 중 하나. 모든 필드는 키워드와 값으로 구성됩니다.

여러 활동 필드가 이 장에서 설명됩니다. 자세한 내용은 *dtactionfile(4)* 매뉴얼 페이지를 참조하십시오.

3. 파일을 저장하십시오.

4. 활동 아이콘이 고유한 이미지를 갖게 하려면, 활동에 대한 아이콘을 작성하십시오. 아이콘의 기본 위치는 다음과 같습니다.
 - 개인 아이콘 `HomeDirectory/.dt/icons`
 - 시스템 전체 아이콘 `/etc/dt/appconfig/icons/language`. 기본 `language`는 C입니다.

자세한 내용은 “활동이 사용하는 아이콘 이미지의 지정” 189 페이지를 참조하십시오.
5. 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오.
6. 활동에 대한 활동 파일을 작성하십시오. 활동 파일이 해당 활동을 표시하는 아이콘을 파일 관리자 또는 응용 프로그램 관리자에 작성합니다. (활동이 응용 프로그램을 시작하도록 작성된 경우, 이 아이콘을 응용 프로그램 아이콘이라고 합니다.)

활동 파일을 작성하려면, `action_name`와 같은 이름으로 된 실행 파일을 작성하십시오. 이 파일은 쓰기 권한이 있는 임의의 디렉토리에 둘 수 있습니다. 원하는 수 만큼 활동 파일을 작성할 수 있습니다.

COMMAND 활동 작성의 예

다음 단계에서는 원격 시스템 AppServerA에서 팩스 응용 프로그램을 시작하는 개인 활동이 작성됩니다. 팩스 응용 프로그램을 시작하는 명령어는 다음과 같습니다.

```
/usr/fax/bin/faxcompose [파일이름]
```

1. `HomeDirectory/.dt/types/Fax.dt` 파일을 작성하십시오.
2. 파일에 다음과 같은 활동 정의를 두십시오.

```
ACTION FaxComposer
{
    TYPE                COMMAND
    ICON                fax
    WINDOW_TYPE        NO_STDIO
    EXEC_STRING        /usr/fax/bin/faxcompose -c %Arg_1%
    EXEC_HOST          AppServerA
    DESCRIPTION        팩스 작성기를 실행합니다.
}
```

`WINDOW_TYPE` 및 `EXEC_STRING` 필드는 활동의 동작에 대해 설명합니다.

`WINDOW_TYPE—NO_STDIO` 키워드는 활동을 터미널 에뮬레이터 윈도우에서 실행할 필요가 없는 활동을 지정합니다.

“활동을 위한 윈도우 지원의 지정” 199 페이지를 참조하십시오.

`EXEC_STRING—구문 %Arg_1%`는 놓은 파일을 허용합니다. 활동 아이콘을 두 번 누르면 활동이 빈 팩스 작성기 윈도우를 엽니다.

“COMMAND 활동에 대한 실행 문자열 구축” 193 페이지를 참조하십시오.

3. 파일을 저장하십시오.

4. 아이콘 편집기를 사용하여 다음과 같은 아이콘 이미지 파일을 *HomeDirectory/.dt/icons* 디렉토리에 작성하십시오.
 - fax.m.pm 크기 32 x 32 픽셀
 - fax.t.pm 크기 16 x 16 픽셀
5. 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오.
6. FaxComposer라는 실행 파일을 쓰기 권한이 있는 디렉토리(예를 들어, 홈 디렉토리)에 작성하십시오.

MAP 활동 작성 예

팩스로 보내는 대부분의 파일은 문서 편집기로 작성하였으며 데이터 유형이 TEXTFILE (이름이 *.txt인 파일)이라고 가정합니다.

다음 단계는 "팩스" 메뉴 항목을 데이터 유형의 선택 메뉴에 추가합니다.

1. 앞 예에서 작성된 파일 *HomeDirectory/.dt/types/Fax.dt*를 여십시오.
2. 이 맵 활동 정의를 다음 파일에 추가하십시오.

```

ACTION Fax
{
  ARG_TYPE          TEXTFILE
  TYPE              MAP
  MAP_ACTION        FaxComposer
}

```

3. 파일을 저장하십시오.
4. TEXTFILE에 대한 데이터 속성 정의를 */usr/dt/appconfig/types/language/dtpad.dt*에서 새 파일 *HomeDirectory/.dt/types/textfile.dt*로 복사하십시오. 팩스 활동을 ACTIONS 필드에 추가하십시오.

```

DATA_ATTRIBUTES TEXTFILE
{
  ACTIONS          Open, Print, Fax
  ICON             Dtpenpd
  ...
}

```

5. 파일을 저장하십시오.
6. 응용 프로그램 관리자를 열고 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오.

▼ 활동/데이터 유형 데이터베이스를 재로드하는 방법

새 활동 정의나 편집된 활동 정의가 반영되게 하려면, 데스크탑이 데이터베이스를 다시 읽어야 합니다.

- 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오.
- 또는 다음과 같은 명령어를 실행하십시오.

```
dtaction ReloadActions
```

ReloadActions는 아이콘의 레이블이 "활동 재로드"인 활동의 이름입니다.

또한 사용자가 다음과 같은 작업을 수행할 때에도 활동 데이터베이스를 다시 읽게 됩니다.

- 로그인할 때
- 작업 공간 관리자를 재시작할 때
- 파일 메뉴에서 저장을 선택하여 활동을 활동 만들기 윈도우에 저장할 때

활동에 대한 활동 파일(아이콘) 작성

활동 파일은 파일 관리자나 응용 프로그램 관리자에서 활동을 시각적으로 표현하기 위해 작성된 파일입니다.



그림 12-1 응용 프로그램 관리자의 활동 파일(활동 아이콘)

활동 파일의 아이콘은 활동을 나타내므로 **활동 아이콘**이라고 부르기도 합니다. 기본 활동이 응용 프로그램을 시작하면, 활동 파일 아이콘은 **응용 프로그램 아이콘**이라고 합니다.

활동 아이콘을 두 번 누르면 활동이 실행됩니다. 활동 아이콘은 놓기 영역이 되기도 합니다.

▼ 활동 파일(활동 아이콘)을 작성하는 방법

- **action_name**과 같은 이름으로 된 실행 파일을 작성하십시오. 파일의 내용은 상관이 없습니다.

예를 들어, 활동 정의가 다음과 같으면,

```
ACTION MyFavoriteApp
{
EXEC_STRING          Mfa -file %Arg_1%
DESCRIPTION          MyFavoriteApp를 실행합니다.
ICON                 Mfapp
}
```

활동 파일이 MyFavoriteApp라는 실행 파일이 됩니다. 파일 관리자 및 응용 프로그램 관리자에서는 MyFavoriteApp 파일이 Mfapp.size.type 아이콘 이미지를 사용하게 됩니다. MyFavoriteApp의 아이콘을 두 번 누르면 활동의 실행 문자열이 실행되고 아이콘의 항목 도움말이 DESCRIPTION 필드("MyFavoriteApp를 실행합니다.")의 내용이 됩니다.

활동 레이블

활동 정의에 LABEL 필드가 포함되면, 파일 관리자 및 응용 프로그램 관리자에서 파일 이름(action_name)이 아닌 이 필드의 내용으로 활동 파일에 레이블이 붙습니다. 예를 들어, 활동 정의에 다음과 같은 필드가 포함되면,

```
ACTION MyFavoriteApp
{
LABEL                Favorite Application
...
}
```

활동 아이콘에 "선호하는 응용 프로그램"이라는 레이블이 붙습니다.

활동이 사용하는 아이콘 이미지의 지정

ICON 필드를 사용하여 활동을 위해 작성된 활동 아이콘에 대해 파일 관리자 및 응용 프로그램 관리자에 사용된 아이콘을 지정하십시오.

아이콘을 지정하지 않으면 시스템은 기본 활동 이미지 파일 /usr/dt/appconfig/icons/language/Dtactn.*을 사용합니다.



그림 12-2 기본 활동 아이콘 이미지

기본 활동 아이콘은 다음 자원을 사용하여 변경할 수 있습니다.

*actionIcon: `icon_file_name`

여기서 `icon_file_name`은 기본 이름 또는 절대 경로입니다.

가능한 ICON 필드의 값은 다음과 같습니다.

■ 기본 파일 이름

기본 파일 이름은 아이콘 이미지를 포함하는 파일 이름에서 크기(m과 t)와 이미지 유형(bm과 pm)에 대한 파일 이름 접미사를 뺀 이름입니다. 예를 들어, 파일의 이름이 `GameIcon.m.pm` 및 `GameIcon.t.pm`이면, `GameIcon`을 사용하십시오.

기본 파일 이름을 사용하는 경우에는 아이콘 파일이 다음과 같은 아이콘 경로의 디렉토리에 위치해야 합니다.

- 개인 아이콘: `HomeDirectory/.dt/icons`
- 시스템 전체 아이콘: `/etc/dt/appconfig/icons/language`

■ 전체 파일 이름을 포함하는 아이콘 파일의 절대 경로

절대 경로는 아이콘 파일이 아이콘 검색 경로에 위치하지 않을 경우에만 사용해야 합니다. 예를 들어, 아이콘 파일 `GameIcon.m.pm`이 해당 아이콘 검색 경로에 없는 디렉토리 `/doc/projects`에 있는 경우, ICON 필드의 값은 `/doc/projects/GameIcon.m.pm`이 됩니다.

표 12-1에는 사용자가 작성해야 하는 아이콘의 크기와 해당 파일 이름이 나열되어 있습니다.

표 12-1 활동 아이콘의 이름 및 크기

픽셀 크기	비트맵 이름	픽스맵 이름
48 x 48	<code>name.l.bm</code>	<code>name.l.pm</code>
32 x 32	<code>name.m.bm</code>	<code>name.m.pm</code>
16 x 16	<code>name.t.bm</code>	<code>name.t.pm</code>

▼ 기존 활동 정의를 수정하는 방법

내장 활동을 포함하여 시스템에서 사용 가능한 모든 활동을 수정할 수 있습니다.

주 - 내장 활동 데이터베이스를 수정할 때는 주의를 사용하십시오. 내장 활동은 데스크탑 응용 프로그램에서 잘 수행되도록 설계되었습니다.

1. 수정하려는 활동 정의의 위치를 찾으십시오.

활동 정의의 기본 위치는 다음과 같습니다.

- 내장 활동: `/usr/dt/appconfig/types/language`
- 시스템 전체 활동: `/etc/dt/appconfig/types/language`
- 개인 활동: `HomeDirectory/.dt/types`

시스템에 다른 위치가 포함되어 있을 수도 있습니다. 시스템이 활동을 위해 사용하는 위치의 목록을 보려면, 다음 명령어를 입력하십시오.

```
dtsearchpath -v
```

시스템은 `DTDATABASESEARCHPATH` 아래에 나열된 디렉토리를 사용합니다.

2. 필요한 경우, 다음 디렉토리에서 활동 정의 텍스트를 새 파일 또는 기존 파일에 복사하십시오.

- 시스템 전체 활동: `/etc/dt/appconfig/types/language`
- 개인 활동: `HomeDirectory/.dt/types`

`/usr/dt/appconfig/types/language` 디렉토리에서 파일을 편집하면 안되기 때문에 내장 활동을 복사해야 합니다.

3. 활동 정의를 편집하십시오.

4. 편집이 끝났으면 파일을 저장하십시오.

5. 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오.

활동 정의 우선순위

사용자가 활동을 호출하면, 시스템은 일치하는 활동 이름을 찾기 위해 데이터베이스를 검색합니다. 해당 이름으로 된 활동이 둘 이상 존재하면, 시스템은 사용할 활동을 결정하기 위해 우선순위 규칙을 사용합니다.

- 다른 우선순위 규칙이 적용되지 않으면, 우선순위는 정의의 위치를 근거로 합니다. 다음 활동은 위에서 아래로 나열된 순서대로 우선순위를 갖습니다.
 - 개인 활동 (`HomeDirectory/.dt/types`)
 - 시스템 전체 지역 활동(`/etc/dt/appconfig/types/ language`)
 - 시스템 전체 원격 활동 (`hostname: /etc/dt/appconfig/types/ language`). 검색된 원격 호스트는 응용 프로그램 검색 경로에 나열되어 있는 것입니다.
 - 내장 활동 (`/usr/dt/appconfig/types/ language`)
- 주어진 디렉토리에서 *.dt 파일은 알파벳순으로 읽혀집니다.

- ARG_CLASS, ARG_TYPE, ARG_MODE, 또는 ARG_COUNT에 의해 제한된 활동은 제한되지 않은 활동보다 우선순위가 높습니다. (이 네 필드의 기본값은 *입니다.)
둘 이상의 제한이 적용되는 경우에는 다음과 같이 나열된 순서대로 우선순위를 갖습니다.
 - ARG_CLASS
 - ARG_TYPE
 - ARG_MODE
 - ARG_COUNT
ARG_COUNT가 둘 이상 제한되는 경우에는 다음과 같이 나열된 순서대로 우선순위를 갖습니다.
 - 특정 정수값 n
 - $<n$
 - $>n$
 - *

예를 들어, 다음과 같은 활동 정의를 가정합니다.

```

ACTION EditGraphics
# EditGraphics-1
{
    ARG_TYPE          XWD
    ...
}

ACTION EditGraphics
# EditGraphics-2
{
    ARG_COUNT        0
    ...
}

ACTION EditGraphics
# EditGraphics-3
{
    ARG_TYPE          *
    ...
}

```

EditGraphics 활동 아이콘을 두 번 누르면 제공된 인수가 없고 ARG_COUNT 0이 우선순위를 갖기 때문에 EditGraphics-2를 시작합니다. XWD 유형 파일 인수가 제공되면, EditGraphics-1이 XWD ARG_TYPE을 지정했기 때문에 EditGraphics-1이 사용됩니다. 다른 파일 인수에 대해서는 EditGraphics-3이 사용됩니다.

COMMAND 활동에 대한 실행 문자열 구축

COMMAND 활동에 대한 최소 요구사항은 두 개의 필드—ACTION 및 EXEC_STRING입니다.

```
ACTION action_name
{
EXEC_STRING execution_string
}
```

실행 문자열은 COMMAND 활동 정의에서 가장 중요한 부분입니다. 실행 문자열은 터미널 윈도우에서 실행되는 명령행과 유사한 구문을 사용하지만 이외에도 파일과 문자열 인수를 취급하기 위한 구문이 더 있습니다.

실행 문자열의 일반적 특징

실행 문자열은 다음과 같이 구성할 수 있습니다.

- 파일 및 비파일 인수
- 셸 구문
- 실행 파일의 절대 경로 또는 이름

활동 인수

인수는 명령어 또는 응용 프로그램을 제대로 실행하는데 필요한 정보입니다. 예를 들어, 문서 편집기에서 파일을 열기 위해 사용하는 다음과 같은 명령행을 생각해 봅시다.

```
dtpad filename
```

이 명령어에서 *filename*은 dtpad 명령어의 파일 인수입니다.

활동은 응용 프로그램이나 명령어와 마찬가지로 인수를 가질 수 있습니다. COMMAND 활동이 사용할 수 있는 데이터 유형은 다음과 같습니다.

- 파일
- 문자열 데이터

실행 문자열에서 셸 사용

실행 문자열은 셸을 통하기보다는 직접 실행됩니다. 그러나 실행 문자열에서 명시적으로 셸을 호출할 수 있습니다.

예를 들어,

```
EXEC_STRING      \
                 /bin/sh -c \
                 'tar -tvf% (File)Arg_1% 2>&1 | \${PAGER:-more;} \
                 echo "\\n*** 윈도우에서 닫기를 선택하여 닫으십시오 ***"'
```

실행 파일의 이름 또는 절대 경로

응용 프로그램이 PATH 변수에 나열된 디렉토리에 있는 경우, 간단한 실행 파일 이름을 사용할 수 있습니다. 응용 프로그램이 다른 곳에 있으면, 실행 파일의 절대 경로를 사용해야 합니다.

인수를 사용하지 않는 활동 작성

명령행에서 응용 프로그램을 시작할 때 사용한 구문과 동일한 구문을 EXEC_STRING에 대해 사용하십시오.

예

- 이 실행 문자열은 X 클라이언트 `xcutsel`을 시작하는 활동의 일부입니다.

```
EXEC_STRING xcutsel
```

- 이 실행 문자열은 클라이언트 `xclock`을 디지털 시계로 시작합니다. 명령행은 명령행 옵션을 포함하지만 인수는 필요하지 않습니다.

```
EXEC_STRING xclock -digital
```

삭제된 파일을 허용하는 활동 작성

파일 인수에 다음 구문을 사용하십시오.

```
%Arg_n%
```

또는

```
%(파일)Arg_n%
```

(파일)은 Arg_n에 제공된 인수가 (기본적으로) 파일이라고 가정하기 때문에 선택적입니다. (% (문자열)Arg_n% 구문의 사용에 대해서는 “파일 인수를 문자열로 해석하기” 196 페이지를 참조하십시오.)

이 구문을 사용하면 사용자는 데이터 파일 객체를 활동 아이콘에 놓고 해당 파일 인수와 함께 활동을 시작할 수 있습니다. 이 구문은 n번째 인수를 명령행으로 대체합니다. 파일은 지역 파일 또는 원격 파일일 수 있습니다.

예

- 이 실행 문자열은 놓은 파일을 --load 매개변수로 사용하여 wc -w를 실행합니다.

```
EXEC_STRING wc -w %Arg_1%
```

- 이 예는 디렉토리 인수로만 작동하는 활동에 대한 정의 부분을 나타내고 있습니다. 디렉토리를 활동 아이콘에 놓으면, 활동은 읽기 쓰기 권한이 있는 디렉토리의 모든 파일 목록을 표시합니다.

```
ACTION List_Writable_Files
{
  ARG_TYPE FOLDER
  EXEC_STRING /bin/sh -c 's -l %Arg_1% | grep rw-'
  ...
}
```

파일 인수에 대한 프롬프트를 표시하는 활동 작성

파일 인수에 다음의 구문을 사용하십시오.

```
%(File) "prompt"%
```

이 구문은 사용자가 활동 아이콘을 두 번 눌렀을 때 파일 이름에 대한 프롬프트를 표시하는 활동을 작성합니다.

예를 들어, 다음과 같은 실행 문자열은 wc -w 명령어의 파일 인수에 대한 프롬프트를 표시하는 대화 상자를 표시합니다.

```
EXEC_STRING wc -w %(File) "Count words in file:"%
```

놓은 파일을 허용하거나 프롬프트를 표시하는 활동 작성

파일 인수에 다음의 구문을 사용하십시오.

```
%Arg_n"prompt"%
```

또는

```
%(File)Arg_n"prompt"%
```

이 구문은 다음과 같은 작업을 수행하는 활동을 작성합니다.

- 놓은 파일을 파일 인수로 허용합니다.
- 사용자가 활동 아이콘을 두 번 눌렀을 때 파일 이름에 대한 프롬프트를 표시하는 대화 상자를 표시합니다.

예를 들어, 다음 실행 문자열은 놓은 파일에 대해 lp-oraw를 수행합니다. 아이콘을 두 번 눌러 활동이 시작되면, 파일 이름에 대해 프롬프트를 표시하는 대화 상자가 나타납니다.

```
EXEC_STRING lp -oraw%Arg_1"File to print:"%
```

비파일 인수에 대해 프롬프트를 표시하는 활동 작성

비파일 매개변수에 다음 구문을 사용하십시오.

```
% "prompt" %
```

또는

```
%(문자열) "prompt" %
```

(문자열)은 인용된 텍스트가 기본적으로 문자열 데이터로 해석되기 때문에 선택적입니다. 이 구문은 비파일 데이터에 대한 프롬프트를 표시하는 대화 상자를 표시합니다. 파일 이름에 대한 프롬프트를 표시할 때는 이 구문을 사용하지 마십시오.

예를 들어, 다음 실행 문자열은 xwd 명령어를 실행하고 각 픽셀에 추가될 값에 대한 프롬프트를 표시합니다.

```
EXEC_STRING xwd -add %"Add value:"% -out %Arg_1"Filename:"%
```

파일 인수를 문자열로 해석하기

인수에 다음 구문을 사용하십시오.

```
%(문자열)Arg_n%
```

예를 들어, 이 실행 문자열은 lp -tbanner filename 명령어를 사용하여 파일 이름이 포함된 표제와 함께 파일을 인쇄합니다.

```
EXEC_STRING lp -t%(String)Arg_1%% (File)Arg_1"File to print:"%
```

활동에 셸 기능 제공

실행 문자열에 셸을 지정하십시오.

```
/bin/sh -c 'command '  
/bin/ksh -c 'command '  
/bin/csh -c 'command'
```

예

- 이 실행 문자열은 셸 파이프를 사용하는 활동을 나타냅니다.

```
EXEC_STRING /bin/sh -c 'ps | lp'
```

- 다음 문자열은 셸 처리를 요구하고 파일 인수를 허용하는 더 복잡한 실행 문자열입니다.

```
EXEC_STRING /bin/sh -c 'tbl %Arg_1"Man Page:%" | troff -man'
```

- 다음 실행 문자열은 압축된 파일을 인수로 사용하도록 요구합니다. 활동은 파일의 압축을 풀고 lp-oraw를 사용하여 파일을 인쇄합니다.

```
EXEC_STRING /bin/sh -c 'cat %Arg_1 "File to print:%" | \
uncompress | lp -oraw'
```

- 다음 실행 문자열은 셸 스크립트를 시작합니다.

```
EXEC_STRING /usr/local/bin/StartGnuClient
```

복수 파일 인수에 대한 COMMAND 활동 작성

활동에서 복수 파일 인수를 처리하는 방식은 세 가지가 있습니다.

- 활동을 각 인수에 한 번씩 반복하여 실행할 수 있습니다. EXEC_STRING이 단일 파일 인수를 포함하고, 활동 아이콘에 복수 파일을 놓음으로써 복수 파일 인수를 제공하는 경우, 활동은 각 파일 인수에 대해 별도로 실행됩니다.

예를 들어, 복수 파일 인수가 다음 활동 정의에 제공되는 경우,

```
ACTION DisplayScreenImage
{
EXEC_STRING          xwud -in %Arg_1%
...
}
```

DisplayScreenImage 활동이 반복하여 실행됩니다.

- 활동은 두 개 이상의 교환할 수 없는 파일 인수를 사용할 수 있습니다. 예를 들어,

```
xsetroot -cursor cursorfile maskfile
```

은 특정 순서로 두 개의 서로 다른 파일을 필요로 합니다.

- 활동은 각 파일 인수에 대해 동일한 명령어를 순차적으로 수행할 수 있습니다. 예를 들어,

```
pr file [file ...]
```

은 하나 또는 여러 개의 파일을 한 인쇄 작업에서 인쇄합니다.

교환할 수 없는 인수에 대한 활동 작성

다음과 같은 구문 규약 중 하나를 사용하십시오.

- 파일 이름에 대한 프롬프트를 표시하는 활동을 원하는 경우에는 각 파일 인수에 다음 구문을 사용하십시오.

```
%(파일) "prompt"%
```

각 인수에 서로 다른 *prompt* 문자열을 사용하십시오.

예를 들어, 이 실행 문자열은 두 파일에 대한 프롬프트를 표시합니다.

```
EXEC_STRING xsetroot -cursor %(File)"Cursor bitmap:%" \ %(File)"Mask bitmap:%"
```

- 놓은 파일을 허용하려면

각 파일 인수에 다른 값의 *n*을 사용하여 다음 구문을 사용하십시오.

```
    %Arg_n%  
예를 들어,
```

```
EXEC_STRING diff%Arg_1%%Arg_2%
```

교환할 수 있는 파일 인수를 사용한 활동 작성

다음 구문 규약 중 하나를 사용하십시오.

- 놓은 파일을 허용하고 *command file 1 file 2 ...*의 형태로 명령어를 수행하려면 파일 인수에 다음 구문을 사용하십시오.

```
%Args%
```

- 놓은 파일을 여러 개 허용하거나 두 번 눌렀을 때 단일 파일에 대한 프롬프트를 표시하려면 파일 인수에 다음 구문을 사용하십시오.

```
% Arg_1"prompt"% %Args%
```

활동은 *command file 1 file 2...*의 형태로 명령어를 발행하게 됩니다.

예

- 다음 실행 문자열은 복수의 파일 인수를 갖는

```
pr file 1 file 2
```

를 실행하는 활동을 작성합니다.

```
EXEC_STRING pr %Args%
```

- 다음 실행 문자열은 활동을 두 번 눌렀을 때(파일 인수가 없음) 프롬프트를 표시하는 것을 제외하고 앞의 예와 유사한 활동을 작성합니다.

```
EXEC_STRING pr %Arg_1"File(s) to print:"% %Args%
```

복수의 놓은 파일에 대한 활동 작성

복수의 놓은 파일 인수를 허용하고 다음 형태로 명령행을 실행하려면,

```
commandfile 1 file2 ...
```

다음 구문을 사용하십시오.

```
%Args%
```

예

- 다음 실행 문자열은 복수 파일에 대해 이름이 Checkout인 스크립트를 실행합니다.

```
EXEC_STRING          /usr/local/bin/Checkout \  
                    %Arg_1"Check out what file?"%%Args%
```

- 다음 실행 문자열은 복수의 파일을 갖는 lp -oraw를 실행합니다.

```
EXEC_STRING          lp -oraw %Arg_1"File to print:"% %Args%
```

COMMAND 활동을 위한 윈도우 지원 및 터미널 에뮬레이터

COMMAND 활동이 데스크탑에서 윈도우를 지원하는 방법은 여러 가지가 있습니다.

- 응용 프로그램이 자체 윈도우를 갖고 있는 경우, 추가 윈도우를 지원하지 않도록 활동을 작성할 수 있습니다. 이 옵션은 활동이 직접적인 사용자 입력을 요구하지 않고 출력이 없는 명령어를 실행할 경우에도 사용됩니다.
- 응용 프로그램을 터미널 에뮬레이터 윈도우에서 실행해야 하는 경우, 윈도우를 열고 나서 응용 프로그램을 실행하도록 활동을 작성할 수 있습니다. 터미널 옵션에는 여러 가지가 있습니다.

활동을 위한 윈도우 지원의 지정

표 12-2에 나열된 활동에 필요한 윈도우 지원 유형을 지정하려면 WINDOW_TYPE 필드를 사용하십시오.

표 12-2 제공되는 WINDOW_TYPE 필드 및 윈도우 지원

WINDOW_TYPE	제공되는 윈도우 지원
NO_STDIO	없음. 응용 프로그램에 자체 윈도우가 있는 경우, 또는 명령어로 인한 가시적 출력이 없는 경우에는 NO_STDIO를 사용하십시오.
PERM_TERMINAL	영구적인 터미널 에뮬레이터 윈도우. 활동은 사용자가 명시적으로 닫을 때까지 열려 있는 터미널 윈도우를 엽니다. 사용자는 윈도우에 데이터를 입력할 수 있습니다. 입력을 받아 출력을 작성한 후 종료되는 명령어(예를 들어, <code>lsdirectory</code>)와 함께 사용하십시오.
TERMINAL	임시 터미널 에뮬레이터 윈도우. 활동은 명령이 끝나면 닫히는 터미널 윈도우를 엽니다. 전체 화면 명령어(예: <code>vi</code>)와 함께 사용하십시오.

터미널 에플레이터에 명령행 옵션 지정

활동 정의에서 TERM_OPTS 필드를 사용하여 터미널 에플레이터에 대한 명령행 옵션을 지정하십시오.

예를 들어, 다음 활동은 실행 호스트에 대한 프롬프트를 표시합니다.

```
ACTION OpenTermOnSystemUserChooses
{
WINDOW_TYPE          PERM_TERMINAL
EXEC_HOST             %(String) "Remote terminal on: "%
TERM_OPTS             -title %(String) "Window title: "%
EXEC_STRING           $$SHELL
}
```

다른 기본 터미널 에플레이터 지정

활동이 사용하는 기본 터미널 에플레이터는 dtterm입니다. 이것을 다른 터미널 에플레이터로 변경할 수 있습니다. 활동이 사용할 터미널 에플레이터를 명시적으로 지정하지 않은 경우에는 기본 터미널 에플레이터가 사용됩니다.

활동이 사용하는 터미널 에플레이터는 다음과 같은 명령행 옵션을 가지고 있어야 합니다.

- - title *window_title*
- -e *command*

다음과 같은 두개의 자원이 활동이 사용하는 기본 터미널 에플레이터를 결정합니다.

- localTerminal 자원은 지역 응용 프로그램이 사용하는 터미널 에플레이터를 지정합니다.

```
*localTerminal:          terminal
```

예를 들어,

```
*localTerminal:          xterm
```

- remoteTerminal 자원은 원격 응용 프로그램이 사용하는 터미널 에플레이터를 지정합니다.

```
*remoteTerminal:        host:terminal [,host:terminal...]
```

예를 들어,

```
*remoteTerminal:        sysibm1:/usr/bin/xterm, syshp2:/usr/bin/yterm
```

특정 인수에 대한 활동 제한

특정 유형의 인수에 대한 활동의 제한은 활동을 세밀하게 구분합니다. 예를 들어, PostScript 파일에 대한 뷰어를 호출하는 활동은 PostScript 파일 인수에 대해서만 제한해야 합니다. 이 제한에 따라 PostScript 파일이 아닌 파일이 지정되면 활동은 오류 대화 상자를 표시합니다.

활동을 제한하는 기준은 다음과 같습니다.

- 파일 인수의 데이터 유형
- 파일 인수의 수—예를 들어, 인수 없음과 하나 이상의 인수. 이 제한은 활동 아이콘에 다른 놓기 동작 및 두 번 누르기 동작을 제공합니다.
- 인수의 읽기/쓰기 모드

지정된 데이터 유형에 대한 활동 제한

ARG_TYPE 필드를 사용하여 활동이 유효한 데이터 유형을 지정하십시오. 데이터 속성 이름을 사용하십시오.

각 항목을 쉼표로 구분하여 데이터 유형 목록을 입력할 수 있습니다.

예를 들어, 다음 활동 정의는 GIF 데이터 유형이 작성되었다고 가정합니다.

```
ACTION Open_Gif
{
    TYPE          COMMAND
    LABEL         "Display Gif"
    WINDOW_TYPE  NO_STUDIO
    ARG_TYPE      Gif
    ICON          xgif
    DESCRIPTION   Displays gif files
    EXEC_STRING   xgif
}
```

인수 수에 기초한 활동 제한

ARG_COUNT 필드를 사용하여 활동이 허용하는 인수의 수를 지정하십시오. 유효한 값은 다음과 같습니다.

* (기본값)—모든 수의 인수. 다른 값은 *보다 우선순위가 높습니다.

n —0을 포함한 음수가 아닌 모든 정수.

$>n$ — n 개를 초과하는 인수

<n—n개 미만의 인수.

ARG_COUNT를 사용하는 한 방법은 사용자가 아이콘을 두 번 눌렀는지, 또는 파일을 아이콘에 놓았는지에 따라 다른 활동 아이콘 동작을 제공하는 것입니다. 다음 절 “서로 다른 두 번 누르기 및 놓기 동작을 제공하는 방법”을 참조하십시오.

▼ 서로 다른 두 번 누르기 및 놓기 동작을 제공하는 방법

이 절차를 사용하여 놓은 파일은 허용하지만 활동 아이콘을 두 번 눌렀을 때 파일에 대한 프롬프트를 표시하지 않는 활동을 작성할 수 있습니다.

1. 두 번 누르기 기능에 대한 활동 정의를 작성하십시오.

ARG_COUNT 필드를 사용하여 0 인수를 지정하십시오. 놓은 인수를 허용하지 않는 EXEC_STRING에 대한 구문을 사용하십시오.

2. 놓기 기능에 대해 두 번째 활동 정의를 작성하십시오.

ARG_COUNT 필드를 사용하여 >0 인수를 지정하십시오. 놓은 파일을 허용하는 EXEC_STRING에 대한 구문을 사용하십시오.

예를 들어, `vedit`라는 편집기를 시작하는데 사용할 수 있는 다음과 같은 두 명령행을 가 적합시다.

- 파일 인수 없이 편집기 시작하기

```
vedit
```

- 읽기 전용 문서로 여는 파일 인수와 함께 편집기 시작하기

```
vedit -R filename
```

다음 두 활동은 `Vedit`라는 활동에 대한 놓기 및 두 번 누르기 기능을 작성합니다. 데이터베이스에서 일치하는 항목을 찾는 동안에는 ARG_COUNT 0이 놓기 기능 정의에 내포된 ARG_COUNT*보다 특정하기 때문에 첫 번째 활동이 우선권을 갖습니다.

```
# Double-click functionality
ACTION Vedit
{
    TYPE          COMMAND
    ARG_COUNT     0
    WINDOW_TYPE  PERM_TERMINAL
    EXEC_STRING   vedit
}
# Drop functionality
ACTION Vedit
{
    TYPE          COMMAND
    WINDOW_TYPE  PERM_TERMINAL
    EXEC_STRING   vedit -R %Arg_1%
}
}
```

인수 모드에 기초한 활동 제한

인수의 읽기/쓰기 모드를 지정하려면 ARG_MODE 필드를 사용하십시오. 유효한 값은 다음과 같습니다.

*(기본값)— 모든 모드

!w—쓰기 금지

w—쓰기 가능

원격 시스템에서 응용 프로그램을 실행하는 활동 작성

활동 및 원격 실행과 관련하여 자주 사용되는 두 용어가 있습니다.

데이터베이스 호스트—활동 정의가 포함된 시스템

실행 호스트—실행 파일이 실행되는 시스템

대부분의 경우, 활동 및 그 응용 프로그램은 동일한 시스템에 위치합니다. 활동에 대한 기본 활동 호스트가 데이터베이스 호스트이므로 특수 구문은 필요하지 않습니다.

그러나, 실행 호스트가 데이터베이스 호스트와 다른 경우, 활동 정의는 실행 문자열을 실행해야 하는 위치를 지정해야 합니다.

활동 및 응용 프로그램을 다른 시스템에 위치시킬 수 있는 기능은 데스크탑의 클라이언트/서버 구조의 일부입니다. 네트워크 응용 프로그램에 대한 자세한 설명은 “응용 프로그램 서비스의 관리” 131 페이지를 참조하십시오.

원격 응용 프로그램을 실행하는 활동 작성

활동 정의에서 EXEC_HOST 필드를 사용하여 응용 프로그램의 위치를 지정하십시오.

EXEC_HOST의 유효한 값은 다음과 같습니다.

%DatabaseHost%—활동이 정의된 호스트

%LocalHost%—활동이 호출되는 호스트(세션 서버).

%DisplayHost%—X 서버를 실행하는 호스트(X 터미널에는 허용되지 않음).

%SessionHost%—로그인 관리자 제어가 실행되는 호스트.

hostname—이름이 지정된 호스트. 항상 하나의 특정 호스트에서 활동을 호출하는 환경에서는 이 값을 사용하십시오.

%"prompt" %—활동이 호출될 때마다 사용자에게 호스트 이름에 대한 프롬프트를 표시합니다.

기본값은 *%DatabaseHost%*, *%LocalHost%*입니다. 따라서, *EXEC_HOST* 필드가 생략되면, 활동은 먼저 활동 정의를 포함하는 호스트에서 명령어를 실행하려고 시도합니다. 이 시도에 실패하면, 활동은 명령어를 세션 서버에서 실행하려고 시도합니다.

예

- 다음 필드는 호스트 *ddsyd*를 지정합니다.

```
EXEC_HOST ddsyd
```

- 다음 필드는 호스트 이름에 대한 프롬프트를 표시합니다.

```
EXEC_HOST %"Host containing application:"%
```

- 다음 필드는 활동이 활동 정의가 포함된 호스트에서 응용 프로그램을 실행할 것을 지정합니다. 이 시도에 실패하면, 활동은 응용 프로그램을 호스트 *ddsyd*에서 실행하려고 시도합니다.

```
EXEC_HOST %DatabaseHost%, ddsyd
```

활동 및 데이터 유형 정의에서 변수 사용

문자열 변수 및 환경 변수는 활동 및 데이터 유형 정의 파일에 포함될 수 있습니다.

활동에서 문자열 변수 사용

문자열 변수 정의는 정의 위치에서 파일 끝까지 유효합니다. 데이터베이스에 대한 전역 문자열 변수는 없습니다.

문자열 변수와 환경 변수의 이름이 같으면 문자열 변수가 우선권을 갖습니다.

▼ 문자열 변수를 정의하는 방법

- 다음 구문을 사용하십시오.

```
set variable_name =value
```

변수 이름은 알파벳, 숫자, 밑줄(_)을 포함할 수 있습니다. 각 변수 정의는 다른 행에 위치해야 합니다.

예

```
set Remote_Application_Server=sysapp
set Remote_File_Server=sysdata
```

▼ 문자열 변수를 참조하는 방법

- 다음 구문을 사용하십시오.

```
${[variable_name]}
```

예

```
EXEC-HOST          $Remote_Application_Server
CWD                /net/${Remote_File_Server}/doc/project
```

활동 및 데이터 유형에서 환경 변수 사용

- 다음 구문을 사용하여 환경 변수를 참조하십시오.

```
${[variable]}.
```

변수는 데이터베이스가 로드될 때 확장됩니다(값으로 바뀜). 문자열 변수 및 환경 변수의 이름이 같으면 문자열 변수가 우선권을 갖습니다.

예를 들어, 다음 실행 문자열은 로그인 이름을 포함하는 표제와 함께 파일을 인쇄합니다.

```
EXEC-STRING lp -t${LOGNAME} %(File)Arg_1%
```

명령행에서 활동 호출

데스크탑은 명령행에서 활동을 실행하기 위해 dtaction 명령어를 제공합니다. dtaction을 실행하여 다음에서 활동을 실행할 수 있습니다.

- 스크립트
- 기타 활동
- 터미널 애플레이터 명령행

dtaction 구문

```
dtaction [-user user_name] [-execHost hostname] \  
action_name [argument [argument] ...]
```

-user *user_name*—다른 사용자로서 활동을 실행하는 기능을 제공합니다. *dtaction*이 *user_name* 이외의 사용자에게 의해 호출되면, 암호에 대한 프롬프트가 표시됩니다.

-execHost *hostname*—COMMAND 활동인 경우에만 사용됩니다. 명령어가 실행될 호스트를 지정합니다.

argument—활동에 대한 인수. 일반적으로 파일 인수.

dtaction 클라이언트에는 추가 명령행 옵션도 있습니다. 자세한 내용은 *dtaction(1)* 매뉴얼 페이지를 참조하십시오.

다른 활동을 실행하는 활동 작성

*dtaction*을 활동의 EXEC_STRING에서 사용하십시오.

예를 들어, 다음 활동은 Spell이라는 내장 활동(이 활동은 응용 프로그램 관리자에서 "Check Spelling"이라는 레이블이 붙어 있음)을 사용합니다. 새로운 활동은 문서 편집기 및 Spell 활동을 실행하여 별도의 터미널 애플레이터 윈도우에 철자법 오류를 표시합니다.

```
ACTION EditAndSpell
{
  WINDOW_TYPE      NO_STDIO
  EXEC_STRING      /bin/sh -c 'dtaction Spell
  %Arg_1"File: "%; dtpad %Arg_1%'
}
```

다른 사용자로서 실행하는 활동 작성

EXEC_STRING에서 다음 구문을 사용하십시오.

```
EXEC_STRING dtaction -user user_name action_name [file_argument]
```

새로운 사용자(*user_name*)는 다음 메커니즘 중 하나를 통해 시스템에 대한 표시 액세스 권한을 가져야 합니다.

- 로그인 사용자의 .Xauthority 파일에 대한 읽기 권한
- 또는 xhost 사용 권한

예를 들어, 다음 두 활동은 루트가 되어 *app-defaults* 파일을 편집할 수 있는 기능을 제공합니다.

```
ACTION AppDefaults
{
  WINDOW_TYPE      NO_STDIO
  EXEC_STRING      /usr/dt/bin/dtaction -user root \
  EditAppDefaults %Arg_1"File: "%
}
ACTION EditAppDefaults
```

```

{
WINDOW_TYPE      TERMINAL
EXEC_STRING      /bin/sh -c 'chmod +w %Arg_1%; \
                 vi %Arg_1%; chmod -w %Arg_1%'
}

```

지역화된 활동 작성

데이터 유형의 검색 경로에는 언어별 위치가 포함됩니다. 데스크탑은 데이터 유형 정의를 검색하는 위치를 결정하기 위해 LANG 값을 사용합니다.

지역화된 활동의 위치

지역화된 활동 정의는 활동 검색 경로에서 언어별 디렉토리에 있어야 합니다.

기본 검색 경로는 다음과 같습니다.

- 개인 활동: *HomeDirectory/.dt/types*
- 시스템 전체 활동: */etc/dt/appconfig/types/language*
- 내장 활동: */usr/dt/appconfig/types/language*

▼ 기존 활동을 지역화하는 방법

1. 적절한 언어별 디렉토리(예: */etc/dt/appconfig/types/japanese*)에 파일을 작성하십시오.
2. 활동 정의를 언어별 환경설정 파일에 복사하십시오.
예를 들어, 활동 정의를

```

app_root/dt/appconfig/types/C/file.dt

```

에서 다음 위치로 복사할 수 있습니다.

```

app_root/dt/appconfig/types/japanese/newfile.dt

```
3. LABEL 필드를 추가하거나 기존 LABEL 필드를 수정하십시오.
레이블 *string*
응용 프로그램 관리자 및 파일 관리자는 레이블 문자열을 사용하여 활동의 아이콘을 식별합니다.
4. 활동 정의에서 다음 필드를 지역화하십시오.
 - 지역화된 아이콘: ICON

- 지역화된 항목 도움말: DESCRIPTION
- 지역화된 프롬프트: EXEC_STRING의 인용 텍스트

ToolTalk 응용 프로그램에 대한 활동 작성

주 - 다음 내용은 ToolTalk 메시지를 지원하는 응용 프로그램에만 적용됩니다.

활동 유형 TT_MSG를 사용하여 ToolTalk 메시지를 전송하는 활동을 작성하십시오.

```
ACTION action_name
{
  TYPE          TT_MSG
  ...
}
```

addressing 및 disposition 필드

- ToolTalk addressing 필드는 항상 TT_PROCEDURE로 설정됩니다.
- ToolTalk disposition 필드의 기본값은 정적 메시지 패턴의 지정값입니다.

지원되지 않는 메시지

다음 메시지는 TT_MSG 유형 활동에서 지원되지 않습니다.

- ToolTalk 객체 지향 메시지
- 메시지의 문맥 인수

TT_MSG 활동에 대한 키워드

표 12-3은 TT_MSG 활동의 키워드와 사용법을 나타냅니다.

표 12-3 TT_MSG 활동 키워드 및 사용법

키워드	사용
TT_CLASS	ToolTalk 클래스 메시지 필드의 값을 정의합니다.

표 12-3 TT_MSG 활동 키워드 및 사용법 (계속)

키워드	사용
TT_SCOPE	ToolTalk 범위 메시지 필드의 값을 정의합니다.
TT_OPERATION	ToolTalk 운영 메시지 필드의 값을 정의합니다.
TT_FILE	ToolTalk 파일 메시지 필드의 값을 정의합니다.
TT_ARGn_MODE	<i>n</i> 번째 인수에 대한 ToolTalk 모드 속성의 값을 정의합니다.
TT_ARGn_VTYPE	<i>n</i> 번째 인수에 대한 ToolTalk vtype 속성의 값을 정의합니다.
TT_ARGn_VALUE	<i>n</i> 번째 메시지 인수의 값을 정의합니다.

수동으로 데이터 유형 작성

데이터 유형 정의를 수동으로 작성하는 방법은 다음 두 가지가 있습니다.

- 활동 작성 도구의 사용. 활동 작성 사용은 제 11 장을 참조하십시오.
- 수동으로 데이터 유형 정의 작성

수동으로 데이터 유형 정의를 작성하려면 사용자가 데이터베이스를 편집해야 합니다.

이 장에서는 데이터 유형 정의를 수동으로 작성하는 방법에 대해 설명합니다.

- “데이터 유형을 수동으로 작성해야 하는 이유” 211 페이지
- “데이터 유형 정의의 구성요소: 기준 및 속성” 212 페이지
- “수동으로 데이터 유형 작성: 일반적 단계” 212 페이지
- “개인 활동 및 데이터 유형 작성의 예” 214 페이지
- “데이터 유형에 대한 데이터 기준 정의” 218 페이지
- 데이터 유형의 소개를 보려면 제 10 장을 참조하십시오.
- 데이터 유형 정의에 대한 참조 정보는 dtddsfile(4) 매뉴얼 페이지를 참조하십시오.

데이터 유형을 수동으로 작성해야 하는 이유

데이터 유형을 수동으로 작성하면 데이터 유형 정의 구문에 구축된 모든 기능을 사용할 수 있습니다.

다음과 같은 데이터 유형 기능을 사용하려면 데이터 유형을 수동으로 작성해야 합니다.

- 위치(경로) 기준 데이터 유형 지정
- 열기 및 인쇄 이외의 데이터 유형과 관련된 활동을 지정하는 기능

- 동일한 데이터 유형에 대한 복수의 이름, 패턴 또는 내용 기준—예를 들어, *.abc 또는 *.def 파일을 기준으로 하는 데이터 유형
- 링크 기준 데이터 유형 지정

데이터 유형 정의의 구성요소: 기준 및 속성

데이터 유형 정의는 다음과 같은 두 가지 별도의 데이터베이스 정의로 구성됩니다.

- DATA_ATTRIBUTES 정의
DATA_ATTRIBUTES 정의는 데이터 유형의 이름과 그 유형 파일의 모양 및 동작을 설명합니다.
- DATA_CRITERIA 정의
DATA_CRITERIA 정의는 유형 지정 기준을 설명합니다. 각 기준 정의는 해당 기준이 적용되는 DATA_ATTRIBUTES 정의를 지정합니다.

각 DATA_ATTRIBUTES 정의에는 하나 이상의 DATA_CRITERIA 정의가 있어야 합니다. DATA_ATTRIBUTES 정의는 그와 연관된 DATA_CRITERIA를 여러 개 가질 수 있습니다.

예를 들어, 포스트스크립트 파일에 대해 파일 관리자에서 포스트스크립트가 표시되는 모양과 동작하는 방법을 설명하는 속성 정의를 작성할 수 있습니다. 그런 다음 포스트스크립트 데이터 유형에 두 개의 개별 기준을 정의할 수 있는데, 하나는 파일 이름에 근거한 정의이고, 다른 하나는 파일 내용에 근거한 정의입니다.

자세한 내용은 “데이터 유형에 대한 데이터 기준 정의” 218 페이지를 참조하십시오.

수동으로 데이터 유형 작성: 일반적 단계

이 절에서는 데이터 유형 환경설정 파일을 작성하는 방법을 설명합니다.

데이터 유형에 대한 환경설정 파일

데이터 유형을 포함하는 환경설정 파일은 다음과 같은 요구사항을 만족해야 합니다.

- 파일은 이름 지정 규약 *name* .dt를 사용해야 합니다.

- 파일은 데이터베이스 검색 경로에 위치해야 합니다. 기본 검색 경로는 다음과 같습니다.
 개인 데이터 유형—*HomeDirectory/.dt/types*
 시스템 전체 데이터 유형—*/etc/dt/appconfig/types/language*
 내장 데이터 유형—*/usr/dt/appconfig/types/language*. 사용자는 이 디렉토리를 사용할 수 없습니다.
 데이터베이스 검색 경로를 수정하는데 대한 내용은 “검색 경로 값 설정” 146 페이지를 참조하십시오.

▼ 데이터 유형 정의 작성 방법

1. 기존의 데이터베이스 파일을 열거나 새로운 데이터베이스 파일을 작성하십시오.
 자세한 내용은 이전 절 “데이터 유형에 대한 환경설정 파일” 212 페이지를 참조하십시오.
2. 다음 구문을 사용하여 데이터 유형의 데이터 속성을 정의하십시오.

```
DATA_ATTRIBUTES data_type_name
{
    ICON                               image_name
    DESCRIPTION                         string
                                       attribute_field
                                       attribute_field
                                       ...
}
```

여기서
data_type_name—이 데이터 유형에 주어진 고유한 이름입니다.
image_name—아이콘 파일의 파일 이름 또는 경로. 파일의 기본 이름을 사용합니다. 예를 들어, 아이콘 파일 *myimage.m.pm*과 *myimage.t.pm*에는 *myimage*를 사용합니다.
attribute_field—데이터 유형의 모양과 동작을 정의하는 필드.
string—문자열. 내용은 데이터 유형에 대한 항목 도움말입니다.
 “개인 활동 및 데이터 유형 작성의 예” 214 페이지를 참조하십시오.

3. 다음 구문을 사용하여 데이터 유형의 데이터 기준을 정의하십시오.

```
DATA_CRITERIA criteria_name
{
    DATA_ATTRIBUTES_NAME             data_type_name
                                       criteria_field
                                       criteria_field
                                       ...
}
```

여기서
criteria_name—이 기준 정의의 고유한 이름입니다.
data_type_name—*DATA_ATTRIBUTES* 정의에서 사용하는 이름입니다.

criteria_field—이 데이터 유형에 파일을 지정하는 기준을 정의하는데 사용하는 필드입니다.

“데이터 유형에 대한 데이터 기준 정의” 218 페이지를 참조하십시오.

4. 데이터베이스 파일을 저장하십시오.
5. 데이터 유형에 대한 아이콘을 작성하십시오.
자세한 내용은 “데이터 유형에 사용되는 아이콘 이미지의 지정” 215 페이지를 참조하십시오.
6. 필요에 따라, 속성 정의의 ACTIONS 필드에 나열된 활동을 작성하십시오.
7. 데스크탑_제어의 추가 폴더에 있는 활동 재로드를 두 번 눌러 데이터베이스를 재로드하십시오.

개인 활동 및 데이터 유형 작성의 예

시스템에 *xgif*라는 GIF 그림을 표시하는 응용 프로그램이 있다고 가정합니다. 일반적으로 다음 명령어를 실행하여 프로그램을 실행합니다.

```
xgif filename
```

GIF 그림은 다음과 같은 몇 가지 방식으로 표시할 수 있습니다.

- GIF 데이터 파일을 두 번 누르는 방법
- 데이터 파일을 선택하고 선택 메뉴에서 해당 응용 프로그램을 선택하는 방법

1. 새 파일 *HomeDirectory / .dt/types/GifViewer.dt*를 열어서 편집하십시오.
2. 데이터 유형 정의를 입력하십시오.

```
DATA_ATTRIBUTES Gif
{
  DESCRIPTION      Gif image file.
  ICON              GifIcon
  ACTIONS           View
}
DATA_CRITERIA Gif_Criteria
{
  DATA_ATTRIBUTES_NAME      Gif
  NAME_PATTERN                *.gif
}
```

3. *GifViewer* 활동의 활동 정의를 입력하십시오.

```
ACTION      GifViewer
{
  EXEC_STRING  xgif %(File)Arg_1"Gif file to view:"
  WINDOW_TYPE NO_STDIO
  DESCRIPTION  파일을 두 번 누르거나 놓아 \
              Gif 뷰어를 시작하십시오.
```

```
}
```

이 정의에는 ICON 필드가 포함되지 않기 때문에 활동은 시스템의 기존 아이콘을 사용하게 됩니다.

4. 다음과 같은 매핑 활동을 입력하여 GifViewer 활동을 데이터 유형 정의에 나열된 보기 활동에 연결하십시오. 이 보기 활동을 Gif 유형 파일로 제한하려면 ARG_TYPE 필드를 사용하십시오.

```
ACTION View
{
  ARG_TYPE      Gif
  TYPE          MAP
  MAP_ACTION    GifViewer
}
```

5. 파일을 저장하십시오.
6. 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 눌러 데이터베이스를 재로드하십시오.

데이터 유형의 데이터 속성 정의

DATA_ATTRIBUTES 정의는 데이터 유형의 모양과 동작을 정의합니다. 데이터 유형의 이름을 지정하고 다음 사항을 지정할 수 있는 기능을 제공합니다.

- 파일 관리자 아이콘(ICON 필드)
- 선택 메뉴의 두 번 누르기 동작과 내용(ACTIONS 필드)
- 데이터 유형의 항목별 도움말(DESCRIPTION 필드)

데이터 유형에 사용되는 아이콘 이미지의 지정

파일 관리자에서 사용되는 아이콘을 지정하려면 ICON 필드를 사용하십시오. 사용자가 아이콘 이미지를 지정하지 않으면, 파일 관리자는 레이블만을 표시합니다.

가능한 ICON 필드의 값은 다음과 같습니다.

- 기본 파일 이름.
기본 파일 이름은 아이콘 이미지를 포함하는 파일 이름에서 크기(m과 t) 및 이미지 유형(bm과 pm)에서 파일 이름 접미사를 뺀 이름입니다. 예를 들어, 파일 이름이 GameIcon.m.pm과 GameIcon.t.pm인 경우, GameIcon을 사용합니다.
기본 파일 이름을 사용하는 경우에는 아이콘 파일이 다음과 같은 아이콘 검색 경로의 디렉토리에 위치해야 합니다.
- 개인 아이콘: `HomeDirectory/.dt/icons`

- 시스템 전체 아이콘: `/etc/dt/appconfig/icons/language`
- 전체 파일 이름을 포함하는 아이콘 파일의 절대 경로.
절대 경로는 아이콘 파일이 아이콘 검색 경로에 없는 경우에만 사용해야 합니다. 예를 들어, 아이콘 파일 `GameIcon.m.pm`이 아이콘 검색 경로에 없는 디렉토리 `/doc/projects`에 있는 경우, ICON 필드의 값은 `/doc/projects/GameIcon.m.pm`이 됩니다.
표 13-1에는 사용자가 작성해야 하는 아이콘의 크기와 해당 파일 이름이 나열되어 있습니다.

표 13-1 데이터 유형 아이콘의 이름 및 크기

픽셀 크기	비트맵 이름	픽스맵 이름
32 x 32	<code>name.m.bm</code>	<code>name .m.pm</code>
16 x 16	<code>name.t.bm</code>	<code>name .t.pm</code>

활동과 데이터 유형 연결

데이터 유형을 활동과 연결하는 데에는 다음과 같은 두 가지 방법이 있습니다.

- `DATA_ATTRIBUTES` 정의의 `ACTIONS` 필드는 파일 관리자에서 선택 메뉴에 나타나는 활동들을 나열합니다. 목록의 첫 번째 활동이 기본(두 번 누르기) 활동입니다.
- 활동은 활동 정의의 `ARG_TYPE` 필드를 사용하여 지정된 데이터 유형으로 제한될 수 있습니다.

예를 들어, 다음과 같은 데이터 유형 정의는 시스템 관리자가 작성하고 이름 지정 규약 `*.rm`을 사용하는 특별한 "readme" 파일에 대한 데이터 유형을 만듭니다.

```
DATA_ATTRIBUTES SysReadmeFile
{
    ICON                      SysReadMe
    ACTIONS                   Open, Respond
}
DATA_CRITERIA SysReadmeFileCriteria
{
    NAME_PATTERN              *.rm
    DATA_ATTRIBUTES_NAME    SysReadmFile
}
```

파일에 대한 특별한 응답 활동이 아래에 정의되어 있습니다. 이 활동은 문서 편집기에서 쓰기 가능한 파일의 사본을 엽니다. 파일을 저장하고 문서 편집기를 종료하면, 파일이 시스템 관리자에게 우편으로 전송됩니다(우편 주소는 `sysadmin@utd`입니다).

```
ACTION Respond
{
    ARG_TYPE                  SysReadmeFile
    EXEC_STRING               /bin/sh -c 'cp %Arg_1% $HOME/readme.temp;\
```

```

                                chmod +w $HOME/readme.temp;           \
                                dtpad $HOME/readme.temp;           \
                                cat $HOME/readme.temp |            \
                                /usr/bin/mailx sysadmin@utd;       \
                                rm $HOME/readme.temp'              \
WINDOW_TYPE                     NO_STDIO
}

```

데이터 유형에 기초한 파일 숨기기

파일이 보이지 않는 데이터 유형일 경우, 해당 파일은 파일 관리자에 나타나지 않습니다.

이러한 유형의 객체가 숨겨지도록 지정하려면 `DATA_ATTRIBUTES` 정의에서 `PROPERTIES` 필드를 다음과 같이 사용하십시오.

```
PROPERTIES          invisible
```

파일을 조작할 때의 동작 지정

표 13-2는 응용 프로그램 프로그래머가 주로 사용하는 `DATA_ATTRIBUTES` 필드를 나타냅니다. 이러한 필드는 사용자가 다양한 데스크탑 활동을 수행할 때 파일을 동작시키는 방법을 지정합니다.

자세한 내용은 개발자 환경 설명서의 일부인 *Common Desktop Environment Programmer's Guide*를 참조하십시오.

표 13-2 `DATA_ATTRIBUTES` 필드 및 설명

필드	설명
<code>MOVE_TO_ACTION</code>	디렉토리와 같은 보관소를 위한 필드. 파일이 이 데이터 유형의 보관소로 이동될 때 수행할 활동을 지정합니다.
<code>COPY_TO_ACTION</code>	디렉토리와 같은 보관소를 위한 필드. 파일이 이 데이터 유형의 보관소로 복사될 때 수행할 활동을 지정합니다.
<code>LINK_TO_ACTION</code>	파일이 이 데이터 유형의 파일에 링크될 때 수행할 활동을 지정합니다.
<code>IS_TEXT</code>	이 데이터 유형의 파일이 텍스트 상자에 표시될 수 있는 텍스트를 포함하도록 지정합니다.
<code>MEDIA</code>	해당 ToolTalk 매체 유형을 지정합니다.
<code>MIME_TYPE</code>	해당 MIME 유형을 지정합니다.
<code>X400_TYPE</code>	해당 X400 유형을 지정합니다.

데이터 유형에 대한 데이터 기준 정의

DATA_CRITERIA 정의는 객체 유형을 파일이나 디렉토리에 지정하는데 사용되는 기준을 정의합니다.

객체 유형 지정에 대해 표 13-3에 있는 기준을 사용할 수 있습니다.

표 13-3 DATA_CRITERIA 기준 및 설명

기준	설명
파일 이름	파일 이름은 지정된 패턴과 일치해야 합니다. NAME_PATTERN 필드를 사용하십시오.
파일 위치	경로는 지정된 패턴과 일치해야 합니다. PATH_PATTERN 필드를 사용하십시오.
파일 내용	파일 내용 중 지정된 부분이 지정된 데이터와 일치해야 합니다. CONTENT 필드를 사용하십시오.
파일 모드	파일이 지정된 권한(읽기, 쓰기, 실행 및 디렉토리)을 갖고 있어야 합니다. MODE 필드를 사용하십시오.
기호 링크	객체가 링크되는 파일에 따라 유형이 지정됩니다.

한 데이터 유형에 대해 두 개 이상의 기준을 사용할 수 있습니다. 그러나 한 데이터 유형에 NAME_PATTERN과 PATH_PATTERN 기준을 같이 사용할 수는 없습니다.

이름 기반 데이터 유형

이름 지정 요구사항을 지정하려면 NAME_PATTERN 필드를 사용하십시오. 필드 값은 다음과 같은 와일드카드를 포함할 수 있습니다.

?—단일 문자에 해당합니다.

*—모든 문자에 해당합니다(빈 문자열 포함).

[cc...]—대괄호로 묶은 모든 문자(c)에 해당합니다.

[c-c]—c에서 c까지의 모든 문자에 해당합니다.

예

- 다음과 같은 데이터 유형 정의는 파일 이름에 기초한 데이터 유형을 작성합니다. 파일 이름은 QS로 시작하여 .doc로 끝나야 합니다.

```

DATA_ATTRIBUTES QS_Doc
{
  DESCRIPTION   이 파일에는 QS 프로젝트에 대한 문서가 \
                포함되어 있습니다.
  ICON          Word_Doc
  ACTIONS       Open
}
DATA_CRITERIA QS_Doc_Criteria
{
  NAME_PATTERN   QS*.doc
  DATA_ATTRIBUTES_NAME QS_Doc
}

```

- 다음과 같은 정의는 Demo_n이라는 디렉토리에 대한 데이터 유형을 작성하며, n은 0-9 사이의 수입니다.

```

DATA_ATTRIBUTES Demo_directory
{
  DESCRIPTION   디렉토리입니다. 디렉토리를 열려면 두 번 누르십시오.
  ICON          Demo
  ACTIONS       OpenInPlace, OpenNewView
}
DATA_CRITERIA Demo_directory_criteria
{
  NAME_PATTERN   Demo_[0-9]
  MODE           d
  DATA_ATTRIBUTES_NAME Demo_directory
}

```

위치 기반 데이터 유형

경로를 지정하려면 PATH_PATTERN 필드를 사용하십시오. NAME_PATTERN에서와 같이 와일드카드 문자를 사용할 수 있습니다.

예를 들어, 다음과 같은 데이터 유형은 경로에 기초한 기준을 사용합니다.

```

DATA_ATTRIBUTES Project_Graphics
{
  DESCRIPTION   QS 프로젝트에 대한 그래픽 파일입니다. 그래픽을 보려면 \
                아이콘을 두 번 누르십시오.
  ICON          QSgraphics
}
DATA_CRITERIA Project_Graphics_Criteria
{
  DATA_ATTRIBUTES_NAME Project_Graphics
  PATH_PATTERN     */projects/QS/graphics/*
}

```

이름 및 위치에 기초한 데이터 유형

파일 이름과 파일 위치 모두에 기초한 데이터 유형을 작성하려면 `PATH_PATTERN` 값에 파일 이름을 포함시키십시오. 한 기준 정의에 `NAME_PATTERN`과 `PATH_PATTERN`을 같이 사용할 수 없습니다.

예

- 아래 정의된 `QS_Source_Files` 데이터 유형은 이름이 `appn.c`인 모든 파일에 적용됩니다. 여기서 `n`은 1에서 9 사이의 값이며 파일은 `*/projects/QS`의 하위 디렉토리에 있습니다.

```
DATA_ATTRIBUTES QS_Source_Files
{
    ...
}
DATA_CRITERIA QS_Source_Files_Criteria
{
    PATH_PATTERN          */projects/QS/*/app[1-9].c
    DATA_ATTRIBUTES_NAME QS_Source_Files
}
```

- 다음 데이터 유형은 `chnn.xxx`라는 디렉토리 `/doc/project1`에 있는 모든 파일에 적용되며, `n`은 0에서 9 사이의 수이고, `xxx`는 세 자로 된 파일 이름 접미사입니다.

```
DATA_ATTRIBUTES ChapterFiles
{
    DESCRIPTION          프로젝트 문서에 대한 장 파일입니다.
    ICON                 chapter
    ACTIONS              Edit,Print
}
DATA_CRITERIA Chapter_Criteria
{
    PATH_PATTERN          /doc/project1/ch[0-9][0-9].???
    DATA_ATTRIBUTES_NAME ChapterFiles
}
```

파일 모드를 유형 지정 기준으로 사용

필수 권한을 지정하려면 `MODE` 필드를 사용하십시오.

모드 기준은 일반적으로 이름, 위치 또는 내용에 기초한 유형 지정을 조합하여 사용됩니다. 이러한 기준을 사용하면 데이터 유형을 파일이나 디렉토리로 제한하거나 필수인 읽기, 쓰기 및 실행 권한을 지정할 수 있습니다.

`MODE` 필드에는 논리 연산자(표 13-4)와 문자(표 13-5)가 포함될 수 있습니다.

표 13-4 MODE 필드 논리 연산자 및 설명

연산자	설명
!	논리 연산자 NOT
&	논리 연산자 AND
	논리 연산자 OR

표 13-5 MODE 필드 논리 연산자 및 설명

문자	설명
f	데이터 유형이 파일에만 적용됩니다.
d	데이터 유형이 디렉토리에만 적용됩니다.
r	파일을 모든 사용자가 읽을 수 있습니다.
w	파일을 모든 사용자가 쓸 수 있습니다.
x	파일을 모든 사용자가 실행할 수 있습니다.
l	파일이 링크입니다.

특정 모드에 대한 기본값은 해당 모드에 구애받지 않는 것입니다.

예

- 다음 모드 필드는 데이터 유형을 설명한 내용으로 제한합니다.

f&!w—읽기 전용 파일

!w—읽기 전용 파일 및 디렉토리

f&x—실행 파일

f&r&x—쓰기와 실행이 모두 가능한 파일

x|!w—실행 가능하거나 읽기 전용인 파일

- 다음 데이터 유형은 이름 지정 규약 *.doc에 따라 파일 이름을 정한 읽기 전용이며 실행 불가능한 파일의 데이터 유형을 작성합니다.

```
DATA_ATTRIBUTES ReadOnlyDocument
{
    ICON                read_only
    DESCRIPTION         이 문서는 쓰기 가능하지 않습니다. 두 번 누르면
    \                   파일의 읽기 전용 사본으로 편집기를 실행합니다.
    ACTIONS             View
}
DATA_CRITERIA ReadOnlyDocument_Criteria
{
```

```

NAME_PATTERN          *.doc
MODE                  !d&!x&!w
DATA_ATTRIBUTES_NAME  ReadOnlyDocument
}

```

내용 기반 데이터 유형 지정

파일 내용에 기초한 데이터 유형을 지정하려면 CONTENT 필드를 사용하십시오. 내용 기반 데이터 유형 지정은 이름 또는 위치 기반 데이터 유형 지정과 조합하여 사용할 수 있습니다.

이 유형 지정은 파일의 문자열 또는 숫자를 기준으로 할 수 있습니다. 파일의 첫 바이트는 번호 0으로 지정됩니다.

- 파일의 문자열 내용에는 다음 구문을 사용하십시오.

```
CONTENT starting_byte string string
```

- 파일의 숫자 내용에는 다음 구문을 사용하십시오.

```
CONTENT starting_byte byte number
CONTENT starting_byte short number
CONTENT starting_byte long number
```

- 디렉토리의 내용에는 다음 구문을 사용하십시오.

```
CONTENT 0 filename "file_name"
```

8진(0로 시작) 및 16진(0X로 시작) 숫자에 대해 표준 C 표기법을 사용하십시오.

주 - 내용 기반 데이터 유형 지정을 사용하면 시스템 성능이 느려집니다. 가능하면 이름이나 위치 기반 유형 지정을 사용하십시오.

예를 들어, 다음과 같은 데이터 유형 Writable_Wingz는 파일의 맨 앞에 문자열 WNGZ이 있고 쓰기 권한을 갖는 모든 파일에 적용됩니다.

```

DATA_ATTRIBUTES Writable_Wingz
{
    ...
}
DATA_CRITERIA Writable_Wingz_Criteria
{
    CONTENT          0 string WNGZ
    MODE             w&!d
    DATA_ATTRIBUTES_NAME  Writable_Wingz
}

```

▼ 다양한 독립적 기준으로 데이터 유형을 작성하는 방법

다양한 독립적 기준을 사용하여 데이터 유형을 작성할 수 있습니다. 즉, 파일이 기준 중 하나(또는 두 가지 모두)를 충족하는 경우 데이터 유형에 지정됩니다.

1. 데이터 유형에 대한 `DATA_ATTRIBUTES` 정의를 작성하십시오.

2. 각 기준에 대한 `DATA_CRITERIA` 정의를 작성하십시오.

각 기준을 동일한 `DATA_ATTRIBUTES` 정의로 연결하려면 `DATA_ATTRIBUTES_NAME` 을 사용하십시오.

예를 들어, 다음 정의는 Mif 데이터 유형을 작성합니다. 유형 지정은 이름 또는 내용을 기준으로 합니다.

```
DATA_ATTRIBUTES Mif
{
    ICON                      Frame
    ACTION_LIST               Open,Print
}

DATA_CRITERIA Mif_Name_Criteria
{
    DATA_ATTRIBUTES_NAME     Mif
    NAME_PATTERN               *.mif
}

DATA_CRITERIA Mif_Content_Criteria
{
    DATA_ATTRIBUTES_NAME     Mif
    CONTENT                    1 string MIFFile
}
```

지역화된 데이터 유형 작성

데이터 유형의 검색 경로에는 언어별 위치가 포함됩니다. 데스크탑은 `LANG`의 값을 사용하여 데이터 유형 정의를 찾기 위해 검색되는 위치를 결정합니다.

지역화된 데이터 유형의 위치

지역화된 활동 정의는 활동 검색 경로의 언어별 디렉토리에 있어야 합니다.

기본 검색 경로는 다음과 같습니다.

- 개인 활동: `HomeDirectory/.dt/types`

- 시스템 전체 활동: `/etc/dt/appconfig/types/language`
- 내장 활동: `/usr/dt/appconfig/types/language`

▼ 데이터 유형을 지역화하는 방법

1. 적절한 언어별 디렉토리(예: `/etc/dt/appconfig/types/japanese`)에 파일을 작성하십시오.
2. 데이터 유형 정의를 언어별 환경설정 파일로 복사하십시오.
3. 데이터 유형 정의에서 하나 이상의 필드를 지역화하십시오.

데스크탑 아이콘 작성

데스크탑 아이콘은 다음과 같은 내용과 연관되어 있습니다.

- 파일 관리자 및 응용 프로그램 관리자의 활동 파일 및 데이터 유형
- 프론트 패널 제어 도구
- 최소화된 응용 프로그램 윈도우
- 팔레트와 도구모음과 같은 응용 프로그램이 사용하는 그래픽
- 작업 공간 배경
- “아이콘 이미지 파일” 225 페이지
- “아이콘 연결” 228 페이지
- “아이콘 디자인 권장사항” 231 페이지

주 - 개발 환경 설명서에 데스크탑 아이콘에 대한 추가 정보가 있습니다. *Common Desktop Environment: Style Guide and Certification Checklist*의 4장, “시작 디자인”을 참조하십시오.

아이콘 이미지 파일

데스크탑이 아이콘 이미지를 사용하려면 아이콘 이미지 파일이 다음 조건을 만족해야 합니다.

- 적절한 형식이어야 합니다.
- 적절한 파일 이름 지정 규약을 사용해야 합니다.
- 데스크탑 크기 규약 사용해야 합니다.
- 아이콘 검색 경로의 디렉토리에 위치해야 합니다.
- 적절한 구문을 사용하여 데스크탑 구성에서 호출되어야 합니다. 예를 들어, 프론트 패널에 대해 새로운 제어 도구를 작성하는 경우, 프론트 패널 정의에 있는 ICON 패널을 사용하여 제어 도구에 사용할 아이콘 이미지를 지정하십시오.

아이콘 파일 형식

컬러 표시장치의 경우에는 일반적으로 접미사 .pm을 갖는 X 픽스맵(XPM) 형식 아이콘 파일을 사용하십시오. 그 외의 경우에는 일반적으로 .bm 접미사를 갖는 X 비트맵(XBM) 형식 파일을 사용하십시오. 픽스맵 파일에 투명성이 사용되면 .bm 파일이 작성될 때 마스크 파일(_m.bm)이 작성됩니다. 데스크탑이 이러한 파일을 찾는 방법에 대한 자세한 내용은 “아이콘 검색 경로” 153 페이지를 참조하십시오.

아이콘 파일 이름

각 아이콘과 배경 이미지는 개별 파일로서 저장됩니다. 일반적으로 아이콘은 해당 파일의 기본 이름과 함께 지정됩니다. 예를 들어, 실제로 파일이 다음과 같은 이름으로 저장될 때 아이콘은 이름 우편과 함께 참조될 수 있습니다.

```
/usr/dt/appconfig/icons/language/mail.1.pm
```

접미사 추가에 대한 파일 이름 지정 규약은 크기와 유형별로 아이콘을 그룹화하는 것을 도와줍니다. 데스크탑 구성요소의 아이콘 이름은 일반적으로 다음과 같은 형식으로 되어 있습니다.

basename.size.format

또는

basename.format

여기서

basename—이미지를 참조하는데 사용하는 이미지 기본 이름

size—크기를 나타내는 문자. l(대형) m(중형) s(소형) t(최소형)

format—파일 형식. pm(픽스맵) bm(비트맵)

아이콘 크기 규약

표 14-1은 데스크탑 아이콘에 권장되는 픽셀 크기를 나타냅니다.

표 14-1 아이콘 크기 및 파일 이름

아이콘 크기	비트맵 이름	픽스맵 이름
16 x 16(최소형)	<i>name.t.bm</i>	<i>name.t.pm</i>
24 x 24(소형)	<i>name.s.bm</i>	<i>name.s.pm</i>

표 14-1 아이콘 크기 및 파일 이름 (계속)

아이콘 크기	비트맵 이름	픽스맵 이름
32 x 32(중형)	<i>name.m.bm</i>	<i>name.m.pm</i>
48 x 48(대형)	<i>name.l.bm</i>	<i>name.l.pm</i>

표 14-2는 데스크탑 구성요소에서 사용하는 아이콘 크기를 나타냅니다. 사용된 아이콘의 크기가 표시장치의 해상도에 따라 달라지는 경우도 있습니다.

표 14-2 데스크탑 구성요소 및 아이콘 크기

데스크탑 구성요소	고해상도	중해상도	저해상도
파일 관리자 및 응용 프로그램 관리자 (이름 및 아이콘별 보기)	중형	중형	중형
파일 관리자 및 응용 프로그램 관리자 (이름 및 작은 아이콘별 보기)	최소형	최소형	최소형
주 프론트 패널 제어도구	대형	대형	중형
프론트 패널 서브 패널	중형	중형	최소형
프론트 패널 스위치 제어도구	소형	소형	최소형
최소화된 윈도우	대형	대형	중형

예를 들어, 데이터 유형에 **mail** 이라는 아이콘을 지정하고 표시장치가 컬러이며 파일 관리자 우선순위를 작은 아이콘으로 설정한 경우에 사용되는 아이콘 이미지는 **mail.t.pm**입니다.

아이콘 검색 경로

데스크탑은 디렉토리 목록에 있는 파일을 검색하여 아이콘 파일이나 이미지를 찾습니다. **아이콘 검색 경로**라고 하는 이 디렉토리 목록은 여러 가지 환경 변수 값에 의해 결정됩니다. 아이콘 검색 경로를 작성하는데 사용하는 변수와 그 변수를 조합하는 방법은 “아이콘 검색 경로” 153 페이지에서 설명합니다.

기본 검색 경로는 다음과 같습니다.

- 내장 아이콘: `/usr/dt/appconfig/icons/language`
- 시스템 전체 아이콘: `/etc/dt/appconfig/icons/language`
- 개인 아이콘: `HomeDirectory/.dt/icons`

네트워크를 통한 아이콘 액세스

데스크탑은 원격 시스템의 아이콘을 액세스할 수 있습니다. 아이콘 서버의 작성에 관한 내용은 “데이터베이스, 아이콘, 도움말 서비스의 구성” 133 페이지를 참조하십시오.

아이콘 연결

보다 빠른 객체 인식을 위해 아이콘을 다음 항목과 연결할 수 있습니다.

- 활동 및 데이터 유형
- 프론트 패널 및 서브 패널의 제어 도구
- 최소화된 응용 프로그램 윈도우

아이콘 파일 지정

프론트 패널이나 서브 패널에서 활동, 데이터 유형에 사용되는 아이콘에 대해서는 아이콘의 기본 이름만을 지정합니다(접미사는 지정하지 않음). 알맞은 접미사가 자동으로 사용자의 표시장치 해상도, 색상 지원 및 파일 관리자 보기 옵션(예, 작은 아이콘 등)에 추가됩니다.

검색 경로를 겹쳐쓰려면 아이콘에 대한 전체 경로 및 이름을 제공하십시오.

▼ 아이콘을 활동 또는 데이터 유형에 연결하는 방법

1. ICON 필드를 사용하여 아이콘을 지정하십시오.

아이콘 파일에 대한 적절한 이름 지정 규약을 따르는 경우에는 아이콘의 기본 이름만을 지정하십시오. 표시장치의 해상도와 색상 지원에 기초한 올바른 아이콘이 표시될 것입니다.

2. 다음 아이콘 크기를 작성하십시오:

- 활동: 대형, 중형, 최소형
- 데이터 유형: 중형, 최소형

활동 정의의 예

다음 예는 Island Paint 그리기 도구를 시작하기 위한 활동 정의입니다. 아이콘 Ipaint.l과 Ipaint.s가 이 활동과 연결됩니다.

```
ACTION    IslandPaintOpenDoc
{
    WINDOW_TYPE          NO-STDIO
    ICON                  Ipaint
    EXEC_STRING           /usr/bin/IslandPaint %Arg_1"File to open:"%
```

컬러 아이콘을 사용하고 있는 경우, 데스크탑은 실제 아이콘 파일을 검색할 때 .pm을 먼저 추가합니다. 컬러 아이콘을 사용하지 않는 경우(또는 .pm과 일치하지 않는 경우), 데스크탑은 .bm을 추가합니다.

데이터 유형 정의의 예

다음과 같은 데이터 유형 정의는 아이콘 `comprsd.l`과 `comprsd.s`를 압축 파일에 연결합니다.

```
DATA_ATTRIBUTES COMPRESSED
{
    ICON                comprsd
    ACTIONS              Uncompress
    DESCRIPTION          COMPRESSED 파일은 공간을 줄이기 위해 \
                        'compress' 명령어를 사용하여 압축되었습니다.
}
```

▼ 아이콘을 프론트 패널 제어도구에 표시하는 방법

1. ICON 필드를 사용하여 아이콘을 지정하십시오.

제어가 파일을 감시하는 경우(MONITOR_TYPE이 우편이나 파일로 지정된 경우), ALTERNATE_ICON 필드를 사용하여 변경이 감지될 때 사용되는 아이콘을 지정하십시오.

사용자는 버튼과 놓기 영역 제어에 대한 애니메이션도 제공할 수 있습니다.

2. 다음 아이콘 크기를 작성하십시오.

- 프론트 패널 및 서브 패널: 대형, 중형, 최소형
- 작업 공간 스위치: 소형

예

파일 `report`가 `/doc/ftp/pub/` 디렉토리에 놓일 때 다음과 같은 제어도구가 모양을 변경합니다. 파일이 해당 디렉토리에 없을 경우에는 `NoReport.pm` 아이콘이 표시되고, 파일이 있을 경우에는 `Report.pm`이 표시됩니다.

```
CONTROL MonitorReport
{
    CONTAINER_NAME      container_name
    TYPE                 ICON
    MONITOR_TYPE         file
    FILE_NAME            /doc/ftp/pub/report
    ICON                 NoReport
    ALTERNATE_ICON      Report
}
```

▼ 아이콘을 응용 프로그램 윈도우에 연결하는 방법

1. 작업 공간 관리자에 대한 `iconImage` 자원을 다음과 같이 설정하십시오.

```
Dtwm*clientname*iconImage: icon_file_name
```

*clientname*에 적절한 값을 결정하려면 응용 프로그램 관리자를 열고 데스크탑_제어기의 추가 폴더에 있는 윈도우 정보를 두 번 누르십시오. 윈도우를 선택하면 윈도우의 특성이 나열됩니다. WM_CLASS 특성이 윈도우의 클래스 이름을 따옴표로 묶어 표시합니다. 자원 설정에 대한 자세한 내용은 “응용 프로그램 자원 설정” 273 페이지를 참조하십시오.

2. 작업 공간 메뉴에서 작업 공간 관리자 재시작을 선택하십시오.

작업 공간 관리자가 아이콘을 인식했는지 확인하려면 수정하려는 아이콘의 윈도우 크기를 최소화하십시오.

주 - 일부 응용 프로그램에서는 기본 윈도우 아이콘을 겹쳐쓰는 것을 허용하지 않습니다.

▼ 파일 관리자를 아이콘 브라우저로 사용하는 방법

1. 파일 `/usr/dt/examples/language/IconBrowse.dt`를 `HomeDirectory/.dt/types/Iconbrowse.dt` 디렉토리로 복사하십시오.

2. 응용 프로그램 관리자를 열고, 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오.

아이콘(.bm과 .pm 파일)이 있는 디렉토리로 변경하면, 각 아이콘 이름 옆에 아이콘이 표시됩니다. 예를 들어, `/usr/dt/appconfig/icons/language` 디렉토리로 변경하면, 많은 데스크탑 아이콘이 표시됩니다.

주 - 메모리 용량이 작은 시스템에서 아이콘 검색을 활성화하면 파일 관리자가 디렉토리를 표시하는 속도가 느려집니다. 256 x 256을 초과하는 이미지는 기본 구성에서 표시되지 않습니다.

아이콘 검색을 사용불가능하게 하는 방법

1. `IconBrowse.dt` 파일의 개인 사본을 제거하십시오.

2. 응용 프로그램 관리자를 열고, 데스크탑_제어기의 추가 폴더에 있는 활동 재로드를 두 번 누르십시오.

아이콘 디자인 권장사항

관련 아이콘에 대한 공통 주제를 사용하십시오. 예를 들어, 응용 프로그램 아이콘을 설계하는 경우, 응용 프로그램의 아이콘과 데이터 파일의 아이콘을 유사하게 만드십시오.

설계하는 색상 아이콘에 두 가지 색상 버전이 허용되게 하십시오. 아이콘이 단색이나 회색조 화면에 표시되는 경우(또는 충분한 색상을 사용할 수 없는 경우), 아이콘은 자동으로 두 가지 색상 형태로 표시됩니다.

시스템 색상 사용을 보존하려면 데스크탑에서 제공하는 색상으로 아이콘의 색상을 제한하십시오. (아이콘 편집기를 사용하여 작성된 아이콘은 데스크탑 색상만을 사용합니다.)

데스크탑 구성요소에서 사용하는 크기에 관한 내용은 표 14-1을 참조하십시오.

색상 사용

데스크탑 아이콘은 다음과 같은 22 색상의 팔레트를 사용합니다.

- 8가지 정적 회색
- 8가지 정적 색상: 빨간색, 파란색, 녹색, 하늘색, 자홍색, 노란색, 검은색, 흰색
- 6가지 동적 색상: 포그라운드, 백그라운드, 상단 음영, 하단 음영, 선택, 투명

이 팔레트는 다른 응용 프로그램에 필요한 색상 자원을 겹쳐쓰지 않으면서 읽기 쉬운 멋진 아이콘을 작성합니다. 데스크탑에 제공된 대부분의 아이콘은 색상으로 액센트를 준 회색을 사용합니다.

투명한 색상은 아이콘 뒤에 있는 색상이 비치기 때문에 사각형이 아닌 환상적인 아이콘을 만드는데 유용합니다.

고급 프론트 패널의 사용자 정의

사용자는 서브패널의 팝업 메뉴와 아이콘 설치 제어도구를 사용하여 프론트 패널을 사용자 정의할 수 있습니다.

이 장에서는 환경설정 파일을 작성하고 편집하여 프론트 패널을 사용자 정의하는 방법을 설명합니다.

- “프론트 패널 환경설정 파일” 233 페이지
- “사용자 인터페이스 사용자 정의 관리” 236 페이지
- “프론트 패널 정의 구성” 237 페이지
- “주 패널 수정” 240 페이지
- “서브패널의 작성 및 수정” 244 페이지
- “프론트 패널 제어도구 정의” 247 페이지
- “작업 공간 스위치의 사용자 정의” 254 페이지
- “일반 프론트 패널 환경설정” 256 페이지
- 프론트 패널 제어도구 및 구성에 대한 참조 정보는 dtfpfile(4X) 매뉴얼 페이지를 참조하십시오.
- 작업 공간 관리자에 대한 참조 정보는 dtwm(1)과 dtwmrc(4) 매뉴얼 페이지를 참조하십시오.

프론트 패널 환경설정 파일

프론트 패널은 환경설정 파일 데이터베이스에 정의됩니다.

환경설정 파일은 프론트 패널을 사용자 정의하는 방법을 제공합니다. 다음과 같은 유형의 수정은 환경설정 파일의 편집을 통해서만 가능합니다.

- 새로운 제어도구 위치 추가.
- 클라이언트 윈도우와 같은 특별한 제어도구 유형 추가.

- 일정한 기본 동작 변경—예를 들어, 프론트 패널 제어도구가 한 번 누르기에 응답할지 또는 두 번 누르기에 응답할지 여부.

패널 구성의 유연성을 최대화하기 위해, 이러한 파일은 개인 파일이나 시스템 전체 파일이 될 수 있고, 또는 다른 시스템에 있을 수도 있습니다.

프론트 패널은 작업 공간 관리자가 작성하고 관리합니다.

기본 프론트 패널 환경설정 파일

기본 프론트 패널은 프론트 패널 환경설정 파일인 `/usr/dt/appconfig/types/language/dtwm.fp`에서 정의됩니다.

이 파일은 변경할 수 없습니다.

프론트 패널 환경설정 파일의 검색 경로

프론트 패널 정의는 지역적으로 위치하거나 원격 시스템에 있는 많은 파일에 분산될 수 있습니다.

프론트 패널을 정의하기 위해 사용되는 파일은 다음과 같은 요구사항을 만족해야 합니다.

- 파일 이름은 `mail.fp`와 같이 `.fp`로 끝나야 합니다.
- 파일이 활동 데이터베이스 검색 경로상에 있어야 합니다.

기본 활동 데이터베이스 검색 경로에는 다음과 같은 디렉토리가 포함되며 나열된 순서대로 검색됩니다.

개인 사용자 정의—`HomeDirectory/.dt/types`

시스템 전체 사용자 정의—`/etc/dt/appconfig/types/language`

내장 패널 및 제어도구—`/usr/dt/appconfig/types/language`

추가 디렉토리 `HomeDirectory/.dt/types/fp_dynamic`은 사용자 인터페이스에서 작업하는 개인 사용자 정의에 사용됩니다. 이 디렉토리를 수동 사용자 정의에는 사용하지 마십시오.

활동 데이터베이스 검색 경로에 네트워킹을 위한 시스템 구성을 위해 추가된 디렉토리들이 추가로 포함될 수 있습니다. 특히, 응용 프로그램 서버를 액세스하기 위해 시스템을 구성하는 경우, 추가 원격 위치가 추가됩니다. 자세한 내용은 “데이터베이스(활동/데이터 유형) 검색 경로” 151 페이지를 참조하십시오.

프론트 패널이 조합되는 방법: 우선순위 규칙

프론트 패널은 활동 데이터베이스 검색 경로상에 위치한 모든 환경설정 파일로부터 조합됩니다.

정의의 구성요소 간에 충돌이 있을 경우, 우선순위 규칙이 사용될 정의를 결정합니다. 두 구성요소는 다음과 같은 경우 서로 충돌합니다.

- 두 구성요소의 제어 이름, CONTAINER_NAME 및 CONTAINER_TYPE가 동일한 경우.
- 같은 위치를 두고 경쟁할 경우(이름은 다르지만 동일한 CONTAINER_NAME, CONTAINER_TYPE 및 POSITION_HINTS를 갖는 경우).

프론트 패널은 다음과 같은 우선순위 규칙을 사용합니다.

- 구성요소의 제어도구명, 보관소명, 유형이 모두 같은 경우, 먼저 읽은 구성요소가 사용됩니다.

예를 들어, 시스템 전체 제어도구와 내장 제어도구 두 가지 모두 다음 필드를 갖고 있지만 그 외의 것은 서로 다른 경우,

```
CONTROL TextEditor
{
    CONTAINER_TYPE          BOX
    CONTAINER_NAME          Top
    ...
}
```

시스템 전체 제어 도구가 우선권을 갖습니다.

- 두 구성요소가 같은 위치를 두고 경쟁하는 경우, 읽은 순서대로 놓입니다.

예를 들어, 주 패널(CONTAINER_TYPE 상자와 CONTAINER_NAME 상단)에 대해 새로운 개인 제어도구를 만들고, 거기에 POSITION_HINTS 5를 지정하면, 개인 제어도구가 내장 제어도구와 위치 번호가 더 높은 다른 제어도구의 위치를 오른쪽으로 한 자리 밀어냅니다.

주 - 제어도구의 새로운 시스템 전체 또는 개인 버전을 작성하여 해당 제어도구를 수정하는 경우, 새로운 제어도구 정의에 동일한 제어도구명 CONTAINER_NAME과 CONTAINER_TYPE을 지정해야 합니다. 그렇지 않으면 새로운 제어도구가 기존의 제어도구에 추가되어 나타납니다.

동적으로 작성되는 프론트 패널 파일

아이콘 설치 제어도구나 팝업 메뉴를 사용하여 프론트 패널을 사용자 정의할 때, 파일은 디렉토리 *HomeDirectory/.dt/types/fp_dynamic*에 기록됩니다.

프론트 패널은 각 세션에 대해 사용자 정의된 프론트 패널의 상태를 저장 및 복원하기 위해 사용되는 추가 파일 *HomeDirectory/.dt/sessions/dtwmfp.session*을 작성합니다.

사용자 인터페이스 사용자 정의 관리

사용자는 프론트 패널의 팝업 메뉴와 아이콘 설치 제어도구를 사용하여 프론트 패널을 광범위하게 사용자 정의할 수 있습니다.

이 절에서는 다음과 같은 내용을 설명합니다.

- 개인 사용자 정의를 금지하는 방법 예를 들어, 임의의 사용자가 제어도구를 삭제하지 못하도록 만들 수 있습니다.
- 개인 사용자 정의의 취소. 예를 들어, 실수로 삭제한 제어도구를 복원하도록 요청할 수 있습니다.

▼ 개인 사용자 정의를 금지하는 방법

1. 제어도구가 내장 제어도구인 경우, 제어도구 정의를 `/usr/dt/appconfig/types/language/dtwm.fp`에서 `/etc/dt/appconfig/types/language/name.fp`로 복사하십시오.
2. 다음과 같은 명령행을 제어도구 정의에 추가하십시오.

```
LOCKED    True
```

▼ 삭제된 제어도구 또는 서브패널을 복구하는 방법

데스크탑_제어기의 추가 폴더에 있는 프론트 패널 복구 활동은 사용자 인터페이스를 사용하여 수행한 모든 프론트 패널 사용자 정의를 제거합니다. 이 활동을 사용하여 프론트 패널의 팝업 메뉴를 사용하여 수행한 모든 개인 사용자 정의를 제거할 수 있습니다.

다음 절차를 사용하여 개별 제어도구를 복구하십시오.

- `HomeDirectory/.dt/types/fp_dynamic` 디렉토리에서 제어도구를 삭제할 때 작성된 파일을 제거하십시오. 제어도구는 삭제된 원래 제어도구와 같은 이름을 갖게 될 것입니다.

예를 들어, 아이콘 편집기 제어도구를 삭제한 경우, `fp_dynamic` 디렉토리에 있는 파일에는 다음 내용이 포함될 것입니다.

```
CONTROL IconEditor
{
    ...
    DELETE    True
}
```

사용자가 서브패널을 삭제하면, 해당 서브패널과 서브패널의 각 제어도구에 대해 별도의 동적 파일이 작성됩니다.

프론트 패널 정의 구성

프론트 패널은 구성요소의 정의를 조합하여 작성됩니다. 각 구성요소에는 프론트 패널에서 요소가 놓이는 위치, 외적인 모양 및 동작을 정의하는 구문이 필요합니다.

프론트 패널 구성요소

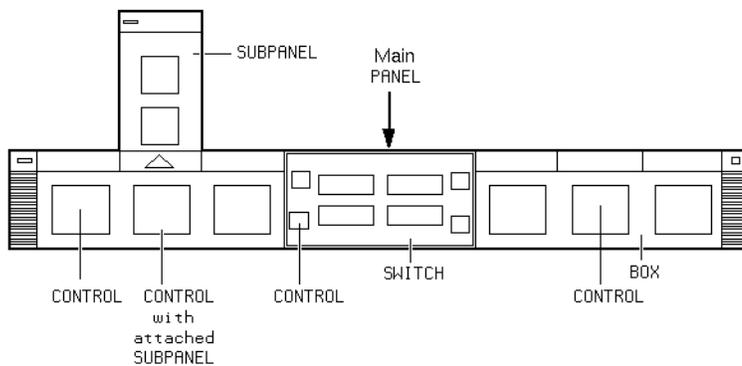


그림 15-1 프론트 패널 구성요소

프론트 패널은 바깥쪽에서 안쪽으로 조합됩니다.

- 패널은 전체 프론트 패널의 최상위 보관소 또는 상위 패널입니다.
- PANEL은 하나 이상의 BOX에 대한 보관소입니다.
- BOX는 하나 이상의 CONTROL에 대한 보관소입니다.
보관소에는 두 가지 특수 유형이 있습니다.
- SUBPANEL은 특정 제어도구와 연관이 있습니다(제어도구는 서브패널의 보관소임). 서브패널은 연관된 제어도구로부터 “슬라이드 업”됩니다.
- SWITCH에는 작업 공간 변경을 위한 버튼과 추가 제어도구가 포함되어 있습니다.

프론트 패널 정의의 일반 구문

프론트 패널의 각 구성요소는 다음과 같은 구문을 사용하여 개별적으로 정의됩니다.

```
COMPONENT name
{
    KEYWORD      value
```

```

        KEYWORD          value
        ...
    }

```

필수적인 키워드도 있고 선택적인 키워드도 있습니다. 자세한 내용은 dtfpfile(4X) 매뉴얼 페이지를 참조하십시오.

PANEL 정의

PANEL은 최상위 구성요소입니다. 정의에는 다음과 같은 내용이 포함됩니다.

- 프론트 패널 이름
- 전체 프론트 패널의 일반적인 모양 및 동작을 설명하는 필드

```

PANEL front_panel_name
{
    KEYWORD          value
    KEYWORD          value
    ...
}

```

*front_panel_name*은 프론트 패널의 고유 이름입니다. 기본 이름은 "FrontPanel"입니다.

BOX 정의

BOX 정의에는 다음과 같은 내용이 포함됩니다.

- BOX 이름
- 상자가 있는 PANEL(CONTAINER_NAME)
- PANEL에서 상자의 위치(POSITION_HINTS)
- 전체 상자에 적용되는 모양과 동작을 설명하는 필드

```

BOX box_name
{
    CONTAINER_NAME          front_panel_name
    POSITION_HINTS          position
    KEYWORD          value
    KEYWORD          value
    ...
}

```

CONTROL 정의

CONTROL 정의에는 다음과 같은 내용이 포함됩니다.

- CONTROL 이름
- 제어도구가 상자, 서브패널 또는 스위치에 있는지 여부(CONTAINER_TYPE)

- 제어도구가 있는 상자, 서브패널 또는 스위치(CONTAINER_NAME)
- BOX에서 CONTROL의 위치(POSITION_HINTS)
- 제어도구의 모양과 동작을 설명하는 필드

```
CONTROL control_name
{
    CONTAINER_TYPE           BOX or SUBPANEL or SWITCH
    CONTAINER_NAME           box_name or subpanel_name or switch_name
    TYPE                      control_type
    POSITION_HINTS            position
    KEYWORD                   value
    KEYWORD                   value
    ...
}
```

SUBPANEL 정의

SUBPANEL 정의에는 다음과 같은 내용이 포함됩니다.

- SUBPANEL 이름
- 서브패널이 연결된 제어도구의 이름(CONTAINER_TYPE)
- 서브패널 고유의 모양과 동작을 설명하는 필드

```
SUBPANELsubpanel_name
{
    CONTAINER_NAME           control_name
    KEYWORD                   value
    KEYWORD                   value
    ...
}
```

SWITCH 정의

SWITCH 정의에는 다음과 같은 내용이 포함됩니다.

- SWITCH 이름
- SWITCH가 있는 BOX(CONTAINER_NAME)
- BOX 내의 SWITCH 위치(POSITION_HINTS)
- SWITCH의 모양과 동작을 설명하는 필드

```
SWITCH switch_name
{
    CONTAINER_NAME           box_name
    POSITION_HINTS            position
    KEYWORD                   value
    KEYWORD                   value
    ...
}
```

주 패널 수정

주 패널은 서브패널을 제외한 프론트 패널 윈도우입니다.

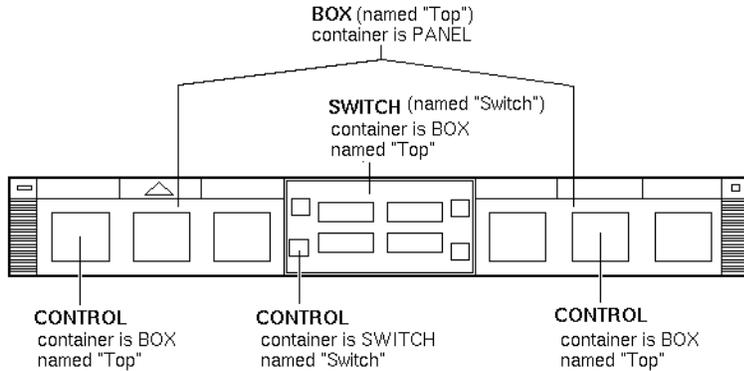


그림 15-2 주 패널 보관소

수정할 수 있는 사항은 다음과 같습니다.

- 제어도구의 추가 또는 제거
- 제어도구 위치 교환

▼ 주 패널에 제어도구를 추가하는 방법

1. 프론트 패널 환경설정 파일을 작성하십시오.

- 시스템 전체: `/etc/dt/appconfig/types/language/*.fp`
- 개인: `HomeDirectory/.dt/types/*.fp`

2. 파일에서 제어도구를 정의하십시오.

`CONTAINER_NAME`와 `CONTAINER_TYPE` 필드를 사용하여 제어도구의 보관소를 지정하십시오.

```
CONTAINER_NAME Top
CONTAINER_TYPE BOX
```

`POSITION_HINTS`를 사용하여 제어도구를 왼쪽에서 오른쪽으로 배치하십시오. 사용자 정의가 내장 제어도구보다 우선권을 가지므로, 새로운 정의가 기존의 제어도구의 위치를 오른쪽으로 한 자리 "밀어냅니다".

3. 환경설정 파일을 저장하십시오.

4. **프론트 패널 제어도구에 대한 아이콘을 작성하십시오.**

“제어도구가 사용하는 아이콘 지정” 243 페이지를 참조하십시오.

5. **작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.**

예를 들어, 파일 `/etc/dt/appconfig/types/language/audio.fp`에 있는 다음과 같은 제어도구 정의는 시계와 캘린더 제어도구 사이에 오디오 응용 프로그램을 삽입합니다.

```
CONTROL AudioApplication
{
    TYPE                icon
    CONTAINER_NAME      Top
    CONTAINER_TYPE      BOX
    ICON                AudioApp
    POSITION_HINTS       2
    PUSH_ACTION         StartAudioApplication
    PUSH_RECALL         true
}
```

▼ 제어도구 삭제 방법

1. **프론트 패널 환경설정 파일을 작성하십시오.**

- 시스템 전체: `/etc/dt/appconfig/types/language/name.fp`
- 개인: `HomeDirectory/.dt/types/name.fp`

2. **삭제하려는 제어도구의 정의를 새 파일에 복사하십시오.**

내장 제어도구의 경우, 제어도구 정의는 `/usr/dt/appconfig/types/language/dtwm.fp`에 있습니다.

사용자가 전체 정의를 복사할 필요는 없습니다. 그러나 사용자가 복사하는 부분에는 `CONTAINER_NAME`과 `CONTAINER_TYPE` 필드가 있어야 합니다.

3. **DELETE 필드를 정의에 추가하십시오.**

```
DELETE                True
```

4. **환경설정 파일을 저장하십시오.**

5. **작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.**

예를 들어, 파일 `/etc/dt/appconfig/types/language/TrashCan.fp`에 있는 다음과 같은 제어도구 정의는 프론트 패널에서 휴지통 제어 도구를 제거합니다.

```
CONTROL Trash
{
    CONTAINER_NAME      Top
    CONTAINER_TYPE      BOX
    DELETE              True
}
```

▼ 제어 도구 수정 방법

아이콘 이미지를 변경하는 것과 같이 제어 도구 정의를 수정하려는 경우에는 다음 절차를 사용하십시오.

1. 전체 제어 도구 정의를 `/usr/dt/appconfig/types/language/dtwm.fp`에서 다음의 위치로 복사하십시오.
 - 시스템 전체: `/etc/dt/appconfig/types/language/name.fp`
 - 개인: `HomeDirectory/.dt/types/name.fp`.
2. 변경하려는 필드를 편집하십시오. 다른 필드를 추가할 수도 있습니다.
3. 파일을 저장하십시오.
4. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

▼ 제어 도구 위치 교환 방법

1. 위치를 변경하려는 제어 도구의 정의를 `/usr/dt/appconfig/types/language/dtwm.fp`에서 다음의 위치로 복사하십시오.
 - 시스템 전체: `/etc/dt/appconfig/types/language/name.fp`
 - 개인: `HomeDirectory/.dt/types/name.fp`.
이동될 각 제어 도구의 전체 제어 도구 정의를 복사해야 합니다.
2. 제어 도구 정의에서 `POSITION_HINTS` 필드의 값을 교환하십시오.
3. 파일을 저장하십시오.
4. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.
예를 들어, 파일 `/etc/dt/appconfig/types/C/MailHelp.fp`에 있는 다음과 같은 정의는 우편 관리자와 도움말 관리자 제어 도구의 위치를 교환하고, 이 제어 도구의 개인적인 변경이 불가능하도록 잠급니다.

```
CONTROL Mail
{
    POSITION_HINTS                12
    LOCKED                       True
    ...the rest of the control definition
}

CONTROL Help
{
    POSITION_HINTS                5
    LOCKED                       True
    ...the rest of the control definition
}
```

▼ 프론트 패널 제어도구 교체 방법

- 다음 사항이 동일한 다른 제어도구 정의를 작성하십시오.

- *control_name*

- CONTAINER_NAME 값

예를 들어, 다음과 같은 두 개의 제어 도구는 서로 다른 두 환경설정 파일에 정의됩니다. 이 제어도구는 동일한 제어도구 이름과 보관소 이름을 가지므로 같은 제어도구로 간주됩니다.

- /etc/dt/appconfig/types/C/SysControls.fp에서의 정의:

```
Control ImportantApplication
{
    CONTAINER_NAME           Top
    CONTAINER_TYPE           BOX
    POSITION_HINTS            2
    ...
}
```

- *HomeDirectory/.dt/types/MyControls.fp*에서의 정의:

```
Control ImportantApplication
{
    CONTAINER_NAME           Top
    CONTAINER_TYPE           BOX
    POSITION_HINTS            6
    ...
}
```

개인 제어도구가 우선권을 가지므로 이 제어 도구가 위치 6에 놓입니다.

제어도구가 사용하는 아이콘 지정

제어도구 정의의 ICON 필드는 제어도구에 사용되는 아이콘 이미지를 정의합니다.

가능한 ICON 필드의 값은 다음과 같습니다.

- 기본 파일 이름

기본 파일 이름은 아이콘 이미지를 포함하는 파일 이름에서 크기(m과 t)와 이미지 유형(bm과 pm)에 대한 파일 이름 접미사를 뺀 이름입니다. 예를 들어, 파일 이름이 MyGame.l.pm 과 MyGame.m.pm인 경우, MyGame이 사용됩니다.

기본 파일 이름을 사용하는 경우에는 아이콘 파일이 다음과 같은 아이콘 검색 경로의 디렉토리에 위치해야 합니다.

- 개인 아이콘: *HomeDirectory/.dt/icons*

- 시스템 전체 아이콘: */etc/dt/appconfig/icons/language*

- 전체 파일 이름을 포함하는 아이콘 파일의 절대 경로.

절대 경로는 아이콘 파일이 아이콘 검색 경로에 없을 경우에만 사용해야 합니다.

필요한 크기의 아이콘은 다음과 같이 제어도구의 위치에 따라 달라집니다.

위치 크기

주 패널-48x48 픽셀(*name.l.pm* or *name.l.bm*)

서브패널-24x24 픽셀(*name.s.pm* or *name.s.bm*)

아이콘 파일을 다음과 같은 위치에 놓으십시오.

- 개인 아이콘: *HomeDirectory/.dt/icons*
- 시스템 전체 아이콘: */etc/dt/appconfig/icons/language*

서브패널의 작성 및 수정

서브패널은 프론트 패널의 팝업 메뉴를 사용하여 작성하고 수정할 수 있습니다.

이 절에서는 프론트 패널 환경설정 파일을 수정해야 하는 시스템 전체 사용자 정의를 제공하는 방법을 설명합니다.

서브패널은 기본 패널의 제어도구에 “연결”됩니다.

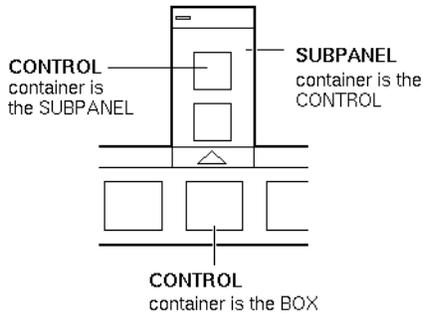


그림 15-3 서브패널의 보관소는 이 패널이 연결되어 있는 제어도구입니다.

연결은 서브패널 정의에서 수행됩니다. *CONTAINER_NAME* 필드가 서브패널이 연결되는 제어도구를 지정합니다.

```
CONTROL control_name
{
    ...
}

SUBPANEL subpanel_name
{
    CONTAINER_NAME                control_name
}
```

```
    ...  
}
```

▼ 새로운 시스템 전체 서브패널 작성 방법

1. 서브패널을 연결하려는 기본 패널의 제어도구 *control_name*을 찾으십시오.
제어도구가 내장 제어도구 중 하나인 경우, 제어도구의 정의는 `/usr/dt/appconfig/types/language/dtwm.fp`에 있습니다.
2. 새 파일 `/etc/dt/appconfig/types/language/*.fp`를 작성하십시오.
3. 서브패널을 정의하십시오.

```
SUBPANELsubpanel_name  
{  
    CONTAINER_NAME          control_name  
    TITLE                   value KEYWORD      value  
    ...  
}
```

4. 새로운 환경설정 파일을 저장하십시오.
5. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

내장 서브패널 사용자 정의

내장 서브패널의 일반 특성(제목 등)과 내용을 수정할 수 있습니다.

▼ 내장 서브패널의 일반 특성을 수정하는 방법

1. 새로운 프론트 패널 환경설정 파일을 작성하십시오.
 - 시스템 전체: `/etc/dt/appconfig/types/language/name.fp`
 - 개인: `HomeDirectory/.dt/types/name.fp`.
2. 전체 기본 서브패널 정의를 `/usr/dt/appconfig/types/language/dtwm.fp`에서 새 파일로 복사하십시오.

```
SUBPANEL subpanel_name  
{  
    ...  
}
```

3. 서브패널 정의를 수정하십시오.
4. 새로운 환경설정 파일을 저장하십시오.

5. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

예를 들어, 파일 /users/janice/.dt/types/PerApps.fp에 있는 다음과 같은 정의는 응용프로그램 서브패널의 이름을 변경합니다.

```
SUBPANEL PersAppsSubpanel
{
    CONTAINER_NAME      TextEditor
    TITLE                Janice's Applications
}
```

▼ 시스템 전체 제어도구를 내장 서브패널에 추가하는 방법

1. 프론트 패널 환경설정 파일 /etc/dt/appconfig/types/ languageName .fp를 작성하십시오.

2. 파일에 시스템 전체 제어도구를 정의하십시오.

다음과 같이 CONTAINER_NAME과 CONTAINER_TYPE 필드를 사용하여 제어도구의 보 관소를 지정하십시오.

```
CONTROL control_name
{
    CONTAINER_NAME      subpanel_name
    CONTAINER_TYPE      SUBPANEL
    ...
}
```

“프론트 패널 제어도구 정의” 247 페이지를 참조하십시오.

3. 환경설정 파일을 저장하십시오.

4. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

예를 들어, 파일 /etc/dt/appconfig/types/language/DigitalClock.fp에 정의된 다음과 같은 제어 도구는 디지털 시계(데스크탑_도구에 있음)를 모든 사용자를 위한 응용프로그램 서브패널에 추가합니다.

```
CONTROL DigitalClockControl
{
    TYPE                icon
    CONTAINER_NAME      PerAppsSubpanel
    CONTAINER_TYPE      SUBPANEL
    ICON                Dtdgclk
    PUSH_ACTION          DigitalClock
    PUSH_RECALL         True
}
```

▼ 제어도구를 내장 서브패널에서 제거하는 방법

- 주 패널 제어도구 삭제를 위한 절차와 동일한 방법을 사용하십시오. “제어도구 삭제 방법” 241 페이지를 참조하십시오.

▼ 아이콘 설치 제어 도구 삭제 방법

- 다음 필드를 서브패널 정의에 추가하십시오.

```
CONTROL_INSTALL      False
```

▼ 서브패널의 자동닫기 동작 변경 방법

사용자가 서브패널을 원래 위치에서 옮기지 않는 한, 제어도구를 선택하면 서브패널이 닫히는 것이 서브패널의 기본 동작입니다.

사용자가 명시적으로 닫을 때까지 서브패널을 열어 두도록 프론트 패널을 구성할 수 있습니다.

1. 다음 위치에 새로운 프론트 패널 환경설정 파일을 작성하십시오.

- 시스템 전체: `/etc/dt/appconfig/types/language/*.fp`
- 개인: `HomeDirectory/.dt/types/*.fp`

2. 기본 패널 정의를 다음과 같이 `/usr/dt/appconfig/types/language/dtwm.fp`에서 새 파일로 복사하십시오.

```
PANEL FrontPanel
{
    ...
}
```

3. 다음 필드를 패널 정의에 추가하십시오.

```
SUBPANEL_UNPOST      False
```

4. 새로운 환경설정 파일을 저장하십시오.

5. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

프론트 패널 제어 도구 정의

아이콘 설치 제어 도구에 아이콘을 끌어 놓음으로써 개인 제어 도구를 작성할 수 있습니다.

이 방법은 사용자 정의 작업을 쉽게 해주며, 이러한 제어 도구가 제공하는 기능은 프론트 패널 제어 도구가 수행할 수 있는 기능의 부분 집합입니다. 예를 들어, 아이콘 설치 제어 도구를 사용하여 작성한 제어 도구는 다음 작업을 수행할 수 없습니다.

- 애니메이션 제공
- 클라이언트 윈도우의 표시

- 이벤트가 발생할 때(예를 들어, 새로운 우편 수신) 모양 변경
이 절에서는 프론트 패널 제어도구를 수동으로 작성하는 방법을 설명합니다.
프론트 패널 제어도구의 구문에 대한 참조 정보는 dtffile(4X) 매뉴얼 페이지를 참조하십시오.

프론트 패널 제어도구 정의

프론트 패널 제어도구 정의의 구조는 다음과 같습니다.

```
CONTROL control_name
{
    TYPE                control_type
    CONTAINER_NAME      value
    CONTAINER_TYPE      value
    모양과 동작을 정의하는 다른 필드
}
```

제어도구 유형

제어도구 정의의 TYPE 필드는 제어도구의 기본적인 동작을 지정합니다.

제어도구 유형	제어도구 동작
icon(기본값).	사용자가 제어도구를 누르거나 제어도구에 파일을 끌어 놓을 때 제어도구가 지정된 활동을 실행합니다.
blank	제어도구의 간격 조정에 사용되는 위치 보유자.
busy	작업 중임을 표시하는 등. 활동을 호출하면 제어도구가 깜박입니다 (이미지 전환).
client	프론트 패널의 클라이언트 윈도우.
clock	시계.
date	현재 날짜를 표시합니다.
file	파일을 표시합니다. 제어도구를 선택하면 파일에 대한 기본 활동이 실행됩니다.

▼ 새로운 제어도구 작성 방법

이 절에서는 제어도구를 정의하기 위한 일반적인 절차와 다양한 유형의 제어도구를 작성하는 방법을 설명합니다.

1. 제어도구가 PUSH_ACTION 및/또는 DROP_ACTION을 가질 경우, 해당 활동 정의를 작성합니다. 이러한 활동이 바로 사용자가 제어도구를 누르거나 제어도구로 파일을 끌어 놓을 때 실행되는 활동입니다.

2. 제어도구에 대한 아이콘 이미지 파일을 작성하십시오.

아이콘 크기, 이름, 위치 등에 관한 내용은 “아이콘 이미지 파일” 225 페이지를 참조하십시오.

3. 새로운 프론트 패널 환경설정 파일을 다음 위치에 작성하십시오.

- 시스템 전체: /etc/dt/appconfig/types/language/*.fp
- 개인: *HomeDirectory*/.dt/types/*.fp

4. 파일에서 제어도구를 정의하십시오.

5. 파일을 저장하십시오.

6. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

누렸을 때 활동을 실행하는 제어도구 작성

다음 필드를 사용하여 제어도구의 동작을 정의하십시오.

- TYPE: icon으로 설정
- PUSH_ACTION: 실행할 활동의 이름을 지정합니다.
예를 들어, 응용프로그램 서브패널에 놓일 다음과 같은 제어도구는 사용자가 가지고 있는 게임을 실행합니다.

```
CONTROL Ball
{
    TYPE            icon
    CONTAINER_NAME  PerAppsSubpanel
    CONTAINER_TYPE  SUBPANEL
    ICON            ball
    PUSH_ACTION     RunBallGame
    HELP_STRING     "게임을 시작하려면 이 제어도구를 선택하십시오."
```

다음 제어도구는 스위치의 왼쪽 상단에 놓입니다. 이 제어도구는 CutDisp라는 활동을 실행합니다.

```
CONTROL StartCutDisp
{
    TYPE            icon
    CONTAINER_NAME  Switch
    CONTAINER_TYPE  SWITCH
    POSITION_HINTS   first
    ICON            cutdisp
    HELP_STRING     "cutdisp를 실행하려면 이 제어도구를 선택하십시오."
    PUSH_ACTION     CutDisp
}
```

파일을 여는 제어도구 작성

다음 필드를 사용하여 제어도구의 동작을 정의하십시오.

- TYPE: file로 설정
- FILE_NAME: 열려는 파일의 경로를 지정합니다.
- PUSH_ACTION: Open으로 설정
파일의 데이터 유형에 열기 활동이 정의되어 있어야 합니다.

예를 들어, 다음 제어도구는 기본 패널의 오른쪽 끝에 놓이게 됩니다. 이 제어도구는 데이터 파일 /users/ellen/PhoneList.txt로 문서 편집기를 시작합니다.

*.txt 파일에 대한 열기 활동은 기본 활동 데이터베이스의 일부입니다.

```
CONTROL EditPhoneList
{
    TYPE                file
    FILE_NAME           /users/ellen/PhoneList.txt
    CONTAINER_NAME      Top
    CONTAINER_TYPE      BOX
    POSITION_HINTS       last
    ICON                PhoneBook
    HELP_STRING         "Ellen의 전화번호 목록을 표시하려면 이 제어도구를 누르십시
오."
    PUSH_ACTION         Open
}
```

놓기 영역으로 작동할 제어도구 작성

놓기 영역 DROP_ACTION 필드를 사용하여 사용자가 파일을 제어도구로 끌어다 놓을 때 실행되는 활동을 지정하십시오. 활동이 파일 인수를 허용할 수 있어야 합니다.

일반적으로 제어도구 정의에는 PUSH_ACTION과 DROP_ACTION 필드 모두가 포함됩니다. 누름과 놓기 활동에 대해 동일한 활동을 사용할 수 있습니다.

예를 들어, 응용프로그램 서브패널에 위치한 다음과 같은 제어도구는 파일 인수를 사용하는 X 클라이언트 xwud를 실행합니다.

```
CONTROL Run_xwud
{
    CONTAINER_NAME      PerAppsSubpanel
    CONTAINER_TYPE      SUBPANEL
    POSITION_HINTS       2
    ICON                XwudImage
    PUSH_ACTION         RunXwud
    DROP_ACTION         RunXwud
}
```

파일을 감시하는 제어도구 작성

다음 필드를 사용하여 제어도구의 동작을 정의하십시오.

- **TYPE:** 다음 값 중에서 하나를 지정하십시오.
`icon`—제어도구에 대한 `PUSH_ACTION` 및 `DROP_ACTION`을 지정하려면 이 유형을 사용합니다.
`file`—제어 도구를 선택했을 때 파일 관리자에서 파일 아이콘을 두 번 누른 것과 같은 동작을 수행하도록 하려면 이 유형을 사용합니다.
- **ICON**과 **ALTERNATE_ICON:** 감시된 파일의 변경된 상태와 변경되지 않은 상태를 표시하기 위해 사용되는 이미지를 설명합니다.
- **MONITOR_TYPE:** 이미지 변경시키는 조건을 설명합니다. 다음 값 중 하나를 사용합니다.
`mail`—정보가 파일에 추가될 때 제어도구의 모양이 변경됩니다.
`file`—지정된 파일이 비어 있지 않으면 제어도구의 모양이 변경됩니다.
- **FILE_NAME:** 감시할 파일을 지정합니다.
예를 들어, 다음 제어도구는 익명의 `ftp`를 사용하여 사용자의 시스템으로 전송될 `meetings`라는 파일이 있는지를 검색합니다. 이 제어도구는 응용프로그램 서브패널의 상단에 놓입니다.

```
CONTROL MonitorCalendar
{
    TYPE                file
    CONTAINER_NAME      PersonalApps
    CONTAINER_TYPE      SUBPANEL
    POSITION_HINTS       first
    FILE_NAME            /users/ftp/meetings
    MONITOR_TYPE        file
    ICON                 meetingsno
    ALTERNATE_ICON      meetingsyes
}
```

단일 인스턴스(토글) 제어도구 작성

단일 인스턴스 제어도구는 `PUSH_ACTION`이 시작한 프로세스가 이미 실행중인지 여부를 점검합니다. 프로세스가 실행되고 있지 않으면 `PUSH_ACTION`이 수행됩니다. 프로세스가 이미 실행되고 있으면, 윈도우가 현재 작업 공간에서 윈도우 스택의 맨위로 이동합니다.

다음 필드를 사용하여 제어도구의 동작을 정의하십시오.

- **PUSH_RECALL:** True로 설정.
- **CLIENT_NAME:** 제어도구에 대한 클라이언트 이름을 지정합니다.

CLIENT_NAME의 값이 응용 프로그램의 최상위 윈도우에 있는 WM_CLASS 특성의 첫번째 문자열(res_name)과 일치해야 합니다. 자세한 내용은 xprop(1) 매뉴얼 페이지를 참조하십시오.

- PUSH_ACTION: 사용자가 제어도구를 눌렀을 때 실행되는 활동을 설명합니다. 예를 들어, 다음 제어도구는 활동명이 MyEditor인 응용 프로그램에 대한 단일 인스턴스를 실행합니다.

```
CONTROL MyEditor
{
    TYPE                icon
    CONTAINER_NAME      Top
    CONTAINER_TYPE      BOX
    POSITION_HINTS       15
    PUSH_RECALL         True
    CLIENT_NAME         BestEditor
    PUSH_ACTION         StartMyEditor
    ICON                MyEd
}
```

▼ 클라이언트 윈도우 제어도구 작성 방법

클라이언트 윈도우 제어도구는 프론트 패널에 삽입된 응용 프로그램 윈도우입니다. 예를 들어, xload 클라이언트 윈도우 제어도구를 작성하여 프론트 패널에 시스템 로드 측정기를 놓을 수 있습니다.

1. 제어도구를 정의하십시오.

다음 필드를 사용하여 제어도구의 동작을 정의하십시오.

- TYPE: client로 설정.
- CLIENT_NAME: 시작할 클라이언트를 지정합니다. CLIENT_NAME의 값이 응용 프로그램의 최상위 윈도우에 있는 WM_CLASS 특성의 첫번째 문자열(res_name)과 일치해야 합니다. 자세한 내용은 xprop(1) 매뉴얼 페이지를 참조하십시오.
- CLIENT_GEOMETRY: 클라이언트 프론트 패널 윈도우에 필요한 크기를 픽셀 단위로 지정합니다.

xwininfo(1) 매뉴얼 페이지에 윈도우의 크기(픽셀 단위)를 찾는 방법이 설명되어 있습니다.

2. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

3. 터미널 터미널 에뮬레이터 명령행에서 클라이언트를 시작하십시오.

예를 들어, 다음 제어도구는 30 x 20 픽셀 로드 측정기를 표시합니다..

```
CONTROL LoadMeter
{
    TYPE                client
```

```

CONTAINER_NAME          Top
CONTAINER_TYPE          BOX
CLIENT_NAME             xload
CLIENT_GEOMETRY         30x20
}

```

클라이언트가 세션 사이에 저장되고 복구되지 않을 경우, 사용자가 눌렀을 때 클라이언트를 시작하는 제어도구를 구성할 수도 있습니다. 예를 들어, 다음과 같은 행을 정의에 추가하여 xload 를 시작하는 LoadMeter 제어도구를 구성할 수 있습니다.

```

PUSH_ACTION             StartXload
그리고 다음과 같은 활동을 작성합니다.

ACTION StartXload
{
    WINDOW_TYPE          NO_STDIO
    EXEC_STRING           /usr/contrib/bin/X11/xload
}

```

▼ 제어도구 애니메이션 방법

제어도구를 누르거나 제어도구에 객체를 끌어 놓을 때 사용될 일련의 애니메이션을 추가할 수 있습니다.

일련의 애니메이션을 가지려면 제어도구가 다음의 조건을 만족해야 합니다.

- icon 유형이어야 합니다.
- PUSH_ACTION 또는 DROP_ACTION을 가지고 있어야 합니다.

1. ANIMATION 구성요소를 사용하여 일련의 애니메이션을 지정하십시오.

```

ANIMATION animation_name
{
    ANIMATION             icon1 [milisecond_delay]
    ANIMATION             icon2 [milisecond_delay]
    ...
}

```

여기서 *icon1*, *icon2* 등은 아이콘의 이름이고, *milisecond_delay*는 애니메이션 아이콘 사이의 시간 지연을 밀리초 단위로 나타낸 값입니다. 기본 시간 지연은 200 밀리초입니다.

2. PUSH_ANIMATION 및/또는 DROP_ANIMATION 필드를 제어도구 정의에 추가하십시오. 값은 ANIMATION의 이름입니다.

예를 들어, 다음 행은 BestEditor 응용 프로그램을 시작하는 제어도구를 애니메이션합니다. 아이콘 사이의 시간 지연은 300 밀리초입니다. 이 예에서는 사용자가 아이콘 파일 *frame1*, *frame2* 등을 작성했다고 가정합니다.

```

CONTROL BestEditor
{
    ...
    PUSH_ANIMATION BestEdAnimation
    ...
}

```

```

}
ANIMATION BestEdAnimation
{
    frame1                300
    frame2
    ...
}

```

프론트 패널 제어도구에 항목 도움말 제공

제어도구의 도움말을 제공하는 방식은 두 가지가 있습니다.

- 제어도구 정의에서 도움말 문자열 제공.

사용자가 제어도구에 대한 항목 도움말을 호출할 때 도움말 문자열이 도움말 표시기에 나타납니다. 도움말 문자열에는 형식(표제 등) 또는 링크가 포함될 수 없습니다.

도움말 문자열을 제공하려면 제어도구 정의에 다음과 같이 도움말 문자열을 지정하십시오.

```
HELP_STRING                help_string
```

- 등록된 도움말 볼륨에 도움말 항목 지정.

도움말 항목은 도움말 시스템의 모든 기능을 사용하여 작성한 정보입니다. 도움말 항목을 작성하려면 데스크탑 도움말 개발자 키트를 사용해야 합니다.

도움말 항목을 제공하려면 제어도구 정의에 도움말 볼륨과 항목 ID를 지정하십시오.

```
HELP_VOLUME help_volume_name
HELP_TOPIC                topic_id
```

작업 공간 스위치의 사용자 정의

작업 공간 스위치는 여러 방법으로 사용자 정의할 수 있습니다.

- 작업 공간 수 변경
- 스위치 레이아웃 변경
- 스위치 제어도구 변경

▼ 작업 공간의 기본 수 변경 방법

- 다음과 같이 작업 공간 관리자 자원을 수정하십시오.

```
Dtwm*workspaceCount:                n
```

자세한 내용은 “시스템 전체의 작업 공간 수를 변경하는 방법” 262 페이지를 참조하십시오.

▼ 스위치 행 수 변경 방법

- SWITCH 정의에서 NUMBER_OF_ROWS 필드를 수정하십시오.

예를 들어, 다음과 같은 정의는 3행의 스위치를 정의합니다.

```
SWITCH Switch
{
    CONTAINER_NAME          box_name
    NUMBER_OF_ROWS          3
    ...
}
```

▼ 작업 공간 스위치에서 제어도구를 변경하거나 추가하는 방법

1. 제어도구 정의를 사용하여 프론트 패널 환경설정 파일을 작성하십시오.

- 다음과 같이 제어도구가 스위치 내에 위치하도록 지정하십시오.

```
CONTAINER_NAME    Switch
CONTAINER_TYPE    SWITCH
```

- 스위치에서 위치를 지정하십시오.

```
POSITION_HINTS    n
```

여기서 n 은 정수입니다. 위치의 번호는 왼쪽에서 오른쪽으로 위에서 아래로 연속하여 지정됩니다. (기본 2행 스위치의 위치는 1에서 4까지입니다.)

2. 제어도구에 대한 아이콘을 작성하십시오. 권장하는 크기는 16 x 16 픽셀입니다.

예를 들어, 다음의 제어도구는 터미널 제어도구를 스위치에 넣습니다.

```
CONTROL SwitchTerminal
{
    TYPE                icon
    CONTAINER_NAME      Switch
    CONTAINER_TYPE      SWITCH
    POSITION_HINTS       3
    ICON                FpTerm
    LABEL               Terminal
    PUSH_ACTION         Dtterm
    HELP_TOPIC          FPOnItemTerm
    HELP_VOLUME         FPanel
}
```

제어도구는 내장 아이콘과 응용프로그램 서브패널의 터미널 제어도구가 사용하는 도움말 항목을 사용합니다.

일반 프론트 패널 환경설정

프론트 패널의 PANEL 구문을 통해 다음과 같은 사항을 수행할 수 있습니다.

- 프론트 패널의 위치를 변경할 수 있습니다.
- 윈도우 장식을 변경할 수 있습니다.
- 제어도구의 일반적인 모양과 동작을 설정할 수 있습니다.

기본 PANEL 설명은 `/usr/dt/appconfig/types/language/dtwm.fp`에 있습니다.

추가 정보는 `dtfile(4X)` 매뉴얼 페이지를 참조하십시오.

일반적 단계

1. `/etc/dt/appconfig/types/language` 또는 `HomeDirectory/.dt/types`에 새로운 프론트 패널 환경설정 파일을 작성하십시오.
2. 기본 PANEL 설명을 `/usr/dt/appconfig/types/language/dtwm.fp`에서 새 파일로 복사하십시오.
3. PANEL 설명을 편집하십시오.
새로 만든 PANEL 설명이 기본 설명보다 우선권을 갖습니다.

▼ 기본 프론트 패널 위치 변경 방법

- PANEL 정의에 `PANEL_GEOMETRY` 필드를 사용하여 위치를 지정하십시오.

예를 들어, 다음과 같은 패널은 오른쪽 상단에 놓입니다.

```
PANEL SpecialFrontPanel
{
    PANEL_GEOMETRY          -1+1
    ...
}
```

▼ 기본 패널의 제어도구에 레이블을 표시하는 방법

1. 다음과 같은 명령행을 PANEL 정의에 추가하십시오.

```
DISPLAY_CONTROL_LABELS True
```

2. LABEL 필드를 각 제어도구에 추가하십시오.

LABEL이 지정되지 않으면 control_name이 사용됩니다.

▼ 제어도구의 누르기 동작 변경 방법

- PANEL 정의에 있는 CONTROL_BEHAVIOR 필드를 사용하여 제어도구의 PUSH_ACTION을 실행하는 방법을 지정하십시오. 이 필드에는 다음과 같은 값을 사용할 수 있습니다.
single_click—제어도구를 한 번 눌러 PUSH_ACTION을 실행합니다.
double_click—제어도구를 두 번 눌러 PUSH_ACTION을 실행합니다.

▼ 완전히 새로운 프론트 패널 작성 방법

새 프론트 패널의 작성은 광범위한 변경을 수행하고자 할 때 선택할 수 있는 좋은 방법입니다.

내장 프론트 패널 구성요소가 충돌하는 것을 피하기 위해 완전히 새로운 프론트 패널은 PANEL 및 기타 보관소에 새로운 이름을 사용해야 합니다.

1. 새 프론트 패널에 대한 PANEL 구성요소를 작성하십시오. 다음과 같이 고유한 이름을 지정하십시오.

```
PANEL front_panel_name  
{  
  ...  
}
```

2. 새로운 보관소 이름을 사용하여 새로운 상자와 제어도구를 작성하십시오.

기존의 구성요소를 사용하려면 해당 구성요소의 정의를 복사한 후에 CONTAINER_NAME 값을 변경해야 합니다.

3. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

3행을 갖는 개인 프론트 패널을 작성하는 예

다음 예는 기본 프론트 패널을 변경하여 제어도구가 3행으로 구성되도록 합니다.

1. `/usr/dt/appconfig/types/language/dtwm.fp`를 `HomeDirectory/.dt/types/MyFrontPanel.fp`로 복사하십시오. 파일 쓰기 권한을 부여하십시오.

이 파일이 새로운 프론트 패널을 제공하기 위해 편집할 파일입니다.

2. 프론트 패널 이름을 변경하십시오.

```
PANEL NewFrontPanel
```

3. Top이라는 상자의 이름을 변경하고 보관소 이름을 편집하십시오.

```
BOX NewFrontPanelTop
{
    CONTAINER_NAME          NewFrontPanel
    POSITION_HINTS           first
    ...
}
```

4. 중간 행 및 하단 행에 대한 상자 정의를 추가하십시오.

```
BOX NewFrontPanelMiddle
{
    CONTAINER_NAME          NewFrontPanel
    POSITION_HINTS           second
}

BOX NewFrontPanelBottom
{
    CONTAINER_NAME          NewFrontPanel
    POSITION_HINTS           second
}
```

5. 다음 제어도구의 CONTAINER_NAME을 NewFrontPanelTop으로 변경하십시오.

- 시계
- 날짜
- 홈
- 문서 편집기
- 우편

6. 다음 제어도구의 CONTAINER_NAME을 NewFrontPanelBottom으로 변경하십시오.

- 프린터
- 스타일
- 응용 프로그램
- 도움말
- 휴지통

7. 스위치의 CONTAINER_NAME을 NewFrontPanelMiddle로 변경하십시오.

8. 작업 공간 메뉴의 윈도우에서 작업 공간 관리자 재시작을 선택하십시오.

작업 공간 관리자의 사용자 정의

이 장에서는 데스크탑 작업 공간 관리자를 사용자 정의하는 방법을 설명합니다.

- “작업 공간 관리자 환경설정 파일” 260 페이지
- “작업 공간 사용자 정의” 262 페이지
- “작업 공간 관리자 메뉴” 264 페이지
- “버튼 바인딩 사용자 정의” 268 페이지
- “키 바인딩의 사용자 정의” 270 페이지
- “기본 동작과 사용자 정의 동작 사이의 전환” 272 페이지

작업 공간 관리자는 데스크탑에서 제공하는 윈도우 관리자입니다. 다른 윈도우 관리자와 마찬가지로 작업 공간 관리자는 다음 사항을 관리합니다.

- 윈도우 프레임 구성요소
- 스택에 넣는 순서 및 초점 동작을 비롯한 윈도우 동작
- 키 바인딩 및 버튼 바인딩
- 최소화된 윈도우의 모양
- 작업 공간 및 윈도우 메뉴

작업 공간 관리자는 또한 다음과 같은 데스크탑 구성요소를 제어합니다.

- **작업 공간.** 작업 공간 관리자는 작업 공간의 수를 제어하고 각 작업 공간에서 열려있는 윈도우의 트랙을 보존합니다.
- **작업 공간 배경.** 작업 공간 배경은 스타일 관리자를 사용하여 변경합니다. 그러나 배경 관리는 작업 공간 관리자의 기능입니다.
- **프론트 패널.** 프론트 패널이 자체 환경설정 파일을 사용하기는 하지만, 작업 공간 관리자가 이 패널을 작성하여 관리합니다.

스타일 관리자를 사용하여 여러 프론트 패널을 변경할 수 있습니다. 스타일 관리자가 자주 사용되는 변경을 빠르게 수행할 수 있으므로 사용자의 노력이 거의 필요하지 않습니다. 그밖의 자원은 수동으로 설정해야 합니다.

작업 공간 관리자는 dtwm입니다. 이는 Motif 윈도우 관리자를 근거로 합니다.

- 작업 공간 관리자에 대한 참조 정보는 dtwm(1) 및 dtwmrc(4) 매뉴얼 페이지를 참조하십시오.

- 작업 공간 관리자 자원 설정에 대한 정보는 “응용 프로그램 자원 설정” 273 페이지를 참조하십시오.
- 프론트 패널 환경설정 파일에 대한 정보는 제 15 장을 참조하십시오.

자원 설정에 대한 자세한 정보는 “응용 프로그램 자원 설정” 273 페이지를 참조하십시오.

작업 공간 관리자 환경설정 파일

작업 공간 관리자는 환경설정 파일에서 윈도우 메뉴, 작업 공간 메뉴, 버튼 바인딩 및 키 바인딩에 대한 정보를 얻습니다.

다음과 같은 파일 중에서 하나를 사용합니다.

- 개인 파일: `HomeDirectory/.dt/dtwmrc`
- 시스템 사용자 정의 파일: `/etc/dt/config/language/sys.dtwmrc`
- 내장 파일: `/usr/dt/config/language/sys.dtwmrc`

작업 공간 관리자는 위에 나열된 순서로 환경설정 파일을 검색하여 맨 처음으로 찾는 파일을 사용합니다.

사용자가 둘 이상의 세션 언어를 사용하는 경우, `HomeDirectory/.dt/dtwmrc`보다 우선되는 개인, 언어별 환경 설정 파일인 `HomeDirectory/.dt/language/dtwmrc`를 작성할 수 있습니다.

▼ 개인 환경설정 파일의 작성 또는 수정 방법

개인 작업 공간 관리자 환경설정 파일은 `HomeDirectory/.dt/dtwmrc`입니다. 이 파일이 있으면 이 파일이 사용됩니다.

1. 데스크탑_제어기의 추가 폴더에 있는 `Dtwmrc` 편집을 두 번 누르십시오.

`dtwmrc` 파일이 이미 있는 경우에는 파일을 편집기에 로드할 수 있습니다. 파일이 없을 경우에는 `sys.dtwmrc`이 `HomeDirectory/.dt/dtwmrc`로 복사된 후에 편집기로 로드됩니다.

2. 파일을 편집하십시오.

3. 편집기를 종료하십시오.

파일은 원래 소스에 관계없이 사용자의 개인 `dtwmrc`로 저장됩니다.

▼ 시스템 전체 환경설정 파일 작성 방법

시스템 전체 작업 공간 관리자 환경설정 파일은 `/etc/dt/config/language/sys.dtwmrc`입니다.

- `/usr/dt/config/language/sys.dtwmrc`를 `/etc/dt/config/language/sys.dtwmrc`로 복사하십시오.

주 - `HomeDirectory/.dt/dtwmrc`가 있으면 이 파일이 사용되지 않습니다.

▼ 다른 파일을 포함하는 방법

- 다음 구문을 사용하십시오.

```
include
{
    path
    path
    ...
}
```

예를 들어, 다음 행은 파일 `/users/ellen/mymenu`에서 나온 것입니다.

```
include
{
    /users/ellen/mymenu
}
```

Include 문은 전체 환경설정 파일을 복사하지 않고 추가 기능을 제공하는데 유용하게 사용할 수 있습니다. 예를 들어, 전체 환경설정 파일을 관리할 필요없이 새로운 키 바인딩을 작성할 수 있습니다. 사용자는 다음과 같은 내용을 가진 파일 `HomeDirectory/.dt/dtwmrc`를 작성할 수 있습니다.

```
include
{
    /etc/dt/config/C/sys.dtwmrc
}
Keys DtKeyBindings
{
    Alt<Key>F5 root f.menu Applications
}
Menu Applications
{
    "GraphicsApp" f.exec "/usr/bin/GraphicsApp/GApp"
    ...
}
```

▼ 작업 공간 관리자를 재시작하는 방법

환경설정 파일에 수행한 변경내용을 반영하려면 작업 공간 관리자를 다시 시작해야 합니다.

- 작업공간 메뉴의 윈도우에서 작업공간 관리자 재시작을 선택하십시오(포인터가 배경에 있을 때 마우스 버튼 3을 누르십시오).

작업 공간 사용자 정의

작업 공간의 이름이나 공간의 번호 변경과 같은 작업 공간에 대한 대부분의 사용자 정의는 데스크탑 인터페이스를 사용하여 수행할 수 있습니다. 그러나 작업 공간 관리자는 시스템 전체 기본값 설정을 위한 자원을 제공합니다.

▼ 시스템 전체의 작업 공간 수를 변경하는 방법

기본 데스크탑 환경설정에서는 4개의 작업 공간을 제공합니다. 작업 공간 스위치와 관련된 있는 팝업 메뉴를 사용하여 작업 공간을 추가하거나 삭제할 수 있습니다.

`/usr/dt/app-defaults/C/Dtwm` 파일은 다음과 같은 기본 작업 공간 수로 설정된 `workspaceCount` 자원을 갖습니다.

```
Dtwm*0*workspaceCount: 4
Dtwm*workspaceCount: 1
```

화면 0에는 복수 개의 작업 공간이 지정되고, 다른 화면에는 하나의 작업 공간이 지정됩니다.

사용자는 `/etc/dt/config/C/sys.resources` 파일을 작성하여(또는 이미 존재하는 경우에는 수정하여), 작업 공간의 새로운 모든 사용자를 위해 기본 작업 공간의 수를 변경할 수 있습니다.

- 기본 표시장치에 대한 시스템 전체 기본값을 설정하려면 `0*workspaceCount` 자원을 사용하십시오.

```
Dtwm*0*workspaceCount: number
```

예를 들어, 다음과 같은 자원은 기본 표시장치의 시스템 전체 작업 공간의 수를 6으로 지정합니다.

```
Dtwm*0*workspaceCount: 6
```

작업 공간 관리자 자원 설정에 대한 정보는 “응용 프로그램 자원 설정” 273 페이지를 참조하십시오.

예를 들어, 다음 자원은 작업 공간의 수를 6으로 설정합니다.

Dtwm*workspaceCount: 6

▼ 시스템 전체의 작업 공간 이름을 제공하는 방법

내부적으로 작업 공간은 번호 지정 규약인 `wsn`에 의해 번호가 지정되는데, 이 때 n 은 0, 1, 2 등으로 진행됩니다. 예를 들어, 4개의 기본 작업 공간은 내부적으로 `ws0`에서 `ws3`으로 번호가 매겨집니다.

- 제목 자원을 사용하여 지정된 작업 공간의 이름을 변경하십시오.

```
Dtwm*wsn: name
```

자원 설정에 대한 자세한 정보는 “응용 프로그램 자원 설정” 273 페이지를 참조하십시오.

예를 들어, 다음과 같은 자원은 4개의 기본 작업 공간을 지정된 이름으로 설정합니다.

```
Dtwm*ws0*title: Anna
Dtwm*ws1*title: Don
Dtwm*ws2*title: Julia
Dtwm*ws3*title: Patti
```

▼ 추가 배경 작성 방법

1. 배경 이미지를 작성하십시오. 이미지는 비트맵 파일 또는 픽스맵 파일입니다.
2. 아이콘은 다음 위치 중 하나에 두십시오. (디렉토리를 작성해야 할 수도 있습니다.)

- 시스템 전체 배경: `/etc/dt/backdrops`
- 개인 배경: `HomeDirectory/.dt/backdrops`

3. 작업공간 메뉴에서 작업공간 관리자 재시작을 선택하십시오.

시스템 전체 및 개인 배경이 `/usr/dt/backdrops`의 내장 배경에 추가됩니다.

같은 이름으로 개인 배경이나 시스템 전체 배경을 작성하여 기존의 내장된 배경을 교체할 수 있습니다.

▼ 그래픽 이미지로 배경을 대체하는 방법

배경이 표시장치의 루트 윈도우 위에 놓입니다. 스타일 관리자-배경 대화 상자가 배경이 투명한 배경없음 설정을 제공합니다.

모든 작업 공간 배경 뒤에는 단 하나의 루트 윈도우가 있습니다. 따라서 루트 윈도우에 위치한 그래픽 이미지가 모든 작업 공간에서 유지됩니다. 배경으로 루트 윈도우를 덮는 작업 공간을 지정할 수 있습니다. 그러나 배경없음으로 설정되었을 때 볼 수 있는 이미지는 모든 작업 공간에서 동일합니다.

1. 그래픽 이미지를 작성하십시오.

이미지는 반드시 루트 윈도우에서 이미지 표시를 위해 존재하는 도구에 맞는 형식이어야 합니다. 예를 들어, `xsetroot`를 사용하려면 비트맵 파일을 만들어야 합니다.

2. 비트맵 파일이 아직 없으면, 실행 파일 `HomeDirectory/.dt/sessions/sessionetc`를 작성하십시오.

사용자가 로그인할 때마다 `sessionetc` 파일이 수행됩니다.

3. `sessionetc` 파일에 이미지를 표시하기 위한 명령어를 넣으십시오.

예를 들어, 다음과 같은 명령행은 지정된 비트맵과 루트 윈도우를 바둑판식으로 배열합니다.

```
xsetroot -bitmap /users/ellen/.dt/icons/root.bm
```

작업 공간 관리자 메뉴

작업 공간 관리자에는 3가지 기본 메뉴가 있습니다.

작업 공간 메뉴—루트 메뉴라고도 합니다. 포인터가 배경에 있을 때 마우스 버튼 3을 누르면 표시됩니다. 이 메뉴는 버튼 바인딩을 통해 마우스 버튼과 연결됩니다.

윈도우 메뉴—포인터가 윈도우 메뉴 버튼(윈도우 프레임의 왼쪽 상단 모서리)에 있을 때, 마우스 버튼 1이나 3을 누르면 표시되는 메뉴. 이 메뉴는 윈도우 메뉴 자원을 통해 버튼과 연결됩니다.

프론트 패널 메뉴—포인터가 프론트 패널의 윈도우 메뉴 버튼 위에 있을 때 마우스 버튼 1이나 3을 누르면 표시되는 메뉴.

작업 공간 관리자 메뉴 구문

작업 공간 관리자 메뉴의 구문은 다음과 같습니다.

```
Menu MenuName
{
    selection1 [mnemonic] [accelerator] function [argument]
    selection2 [mnemonic] [accelerator] function [argument]
    ...
}
```

여기서

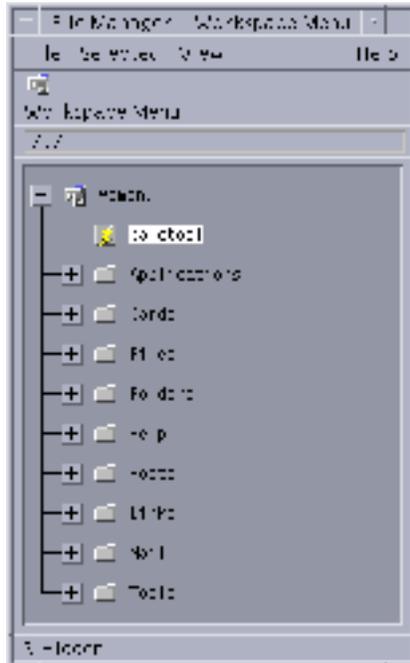
`selection`—메뉴에 표시되는 텍스트 또는 비트맵. 텍스트에 빈칸이 들어 있는 경우, 그 텍스트를 따옴표로 묶으십시오. 비트맵의 경우에는 구문 `@/path`를 사용하십시오.

파일이 작업 공간 메뉴의 첫번째 위치에 추가됩니다. 이 파일의 기본 아이콘(존재할 경우)과 파일 이름이 메뉴 항목에 대한 아이콘과 텍스트로서 사용됩니다.

주 - 작업 공간 메뉴에서 메뉴 항목의 위치를 변경하려면 "작업 공간 메뉴를 수정하는 방법" 266 페이지를 참조하십시오.

▼ 작업 공간 메뉴를 수정하는 방법

1. **프론트 패널의 도구 서브패널에서 작업 공간 메뉴 사용자 정의 제어 도구를 누르십시오.**
작업 공간 메뉴에 대한 구성 폴더를 보여주는 파일 관리자가 표시될 것입니다. 이 폴더에 있는 파일은 각 메뉴 항목을 나타내고, 하위 폴더들은 각 하위 메뉴를 나타냄에 유의하십시오. 작업 공간 메뉴 폴더의 내용을 재배열하여 작업 공간 메뉴를 다시 배열하십시오.



2. 이 하위 폴더 아래의 새 위치로 이동하려는 메뉴 항목을 끌어다 놓으십시오.
트리 뷰 모드에서 파일 관리자를 사용하는 방법에 대해서는 *Solaris* 공통 데스크탑 환경: **사용 설명서**를 참조하십시오.
3. 원하지 않는 메뉴 항목의 파일을 **프론트 패널의 휴지통**으로 끌어다 놓으면 메뉴 항목이 제거됩니다.

- 이름을 바꾸려는 메뉴 항목의 파일 이름이나 폴더명을 편집하여 메뉴 항목의 이름을 변경하십시오.
- 사용자가 수행한 변경 내용에 만족하면 파일 관리자의 파일 메뉴에서 작업 공간 메뉴 갱신을 선택한 후, 파일 관리자를 종료하십시오.
사용자가 작업 공간 메뉴 폴더에서 수행한 변경 내용이 작업 공간 메뉴에 반영됩니다.

▼ 새 작업 공간 (루트) 메뉴 작성 방법

- 해당 파일을 열어서 편집하십시오.

- 개인 `HomeDirectory/.dt/dtwmrc`
- 시스템 전체 `/etc/dt/config/language/sys.dtwmrc`

이러한 파일 작성에 관한 내용은 “작업 공간 관리자 환경설정 파일” 260 페이지를 참조하십시오.

- 새 메뉴를 작성하십시오.

```
Menu menu_name
{
    ...
}
```

“작업 공간 관리자 메뉴 구문” 264 페이지를 참조하십시오.

- 버튼 바인딩을 작성하거나 편집하여 새 메뉴를 표시하십시오.

새 메뉴가 기존의 메뉴를 교체하는 경우에는 작업 공간 메뉴를 표시하는 버튼 바인딩을 편집하십시오.

```
<Btn3Down> root f.menu menu_name
```

새 메뉴가 추가 메뉴인 경우, 새 마우스 버튼 바인딩을 작성하십시오. 예를 들어, 다음과 같은 버튼 바인딩은 배경 위에서 이동 마우스 버튼 3을 누르면 메뉴를 표시합니다.

```
Shift<Btn3Down> root f.menu menu_name
```

- 작업 공간 메뉴에서 작업 공간 관리자 재시작을 선택하십시오.

▼ 새 윈도우 메뉴 작성 방법

주 - 윈도우 메뉴는 작업 공간 관리자에 내장되어 있으며, 일반적으로 사용자 정의되지 않습니다. 응용 프로그램 간의 윈도우 동작이 일치하도록 하려면 윈도우 메뉴를 너무 많이 수정하지 않도록 해야 합니다.

- 해당 파일을 열어서 편집하십시오.

- 개인 *HomeDirectory/.dt/dtwmrc*
- 시스템 전체 */etc/dt/config/language/sys.dtwmrc*
이러한 파일 작성에 관한 내용은 “작업 공간 관리자 환경설정 파일” 260 페이지를 참조하십시오.

2. 새 메뉴를 작성하십시오.

```
Menu menu_name
{
    ...
}
```

3. windowMenu 자원을 사용하여 새 메뉴를 지정하십시오.

```
Dtwm>windowMenu: menu_name
```

4. 작업 공간 메뉴에서 작업 공간 관리자 재시작을 선택하십시오.

버튼 바인딩 사용자 정의

버튼 바인딩은 마우스 버튼 조작과 관련이 있고, 가능한 키보드 조작 보조키는 윈도우 관리자 기능과 관련이 있습니다. 버튼 바인딩은 모든 작업 공간에 적용됩니다.

데스크탑 기본 버튼 바인딩은 *DtButtonBindings*라는 버튼 바인딩 집합의 작업 공간 관리자 환경설정 파일에 정의됩니다.

```
Buttons DtButtonBindings
{
    ...
}
```

버튼 바인딩 구문

버튼 바인딩 구문은 다음과 같습니다.

```
Buttons ButtonBindingSetName
{
    [modifier] <button_nameMouse_action> context function [argument]
    [modifier] <button_nameMouse_action> context function [argument]
```

여기서

button_name—Btn1—왼쪽 마우스 버튼 Btn2—가운데 버튼(3 버튼 마우스) 또는 양쪽 버튼(2 버튼 마우스) Btn3—오른쪽 버튼 Btn4—3버튼 마우스에서 버튼 1과 2 모두 Btn5—3버튼 마우스에서 버튼 2와 3 모두

modifier—Ctrl, Shift, Alt, Lock

mouse_action—Down—마우스 버튼을 누르고 있는 동작, Up—마우스 버튼 누름 동작 해제 Click—마우스 버튼 누름 및 해제 동작, Click2—마우스 버튼 두 번 누르기, Drag—마우스 버튼을 누른 상태로 끌기

context—바인딩을 유효하게 하기 위한 포인터의 위치를 지정합니다. 필요에 따라 여러 개의 문맥을 "|" 문자로 분리하십시오.

root—작업 공간 윈도우. *window*—클라이언트 윈도우 또는 윈도우 프레임. *frame*—내용을 제외한 윈도우 프레임. *icon*(아이콘)—아이콘. *title*—제목 표시줄. *app*—클라이언트 윈도우(프레임 제외).

function—윈도우 관리자 함수 중 하나. 유효한 함수 목록을 보려면 dtwmrc(4) 매뉴얼 페이지를 참조하십시오.

인수—요구되는 윈도우 관리자 함수 인수. 자세한 정보는 dtwmrc(4) 매뉴얼 페이지를 참조하십시오.

예를 들어, 아래 행은 포인터가 작업 공간 윈도우(클라이언트 윈도우가 아님) 내에 있는 동안 마우스 버튼 3을 누를 때, DtRootMenu에 설명된 메뉴가 표시되도록 합니다.

```
<Btn3Down>      root      f.menu      DtRootMenu
```

주 - 버튼 바인딩 구문에 대한 모든 정보는 dtwmrc(4) 매뉴얼 페이지를 참조하십시오.

▼ 버튼 바인딩을 추가하는 방법

1. 해당 파일을 열어서 편집하십시오.

- 개인 *HomeDirectory*/.dt/dtwmrc
- 시스템 전체 /etc/dt/config/language/sys.dtwmrc

이러한 파일 작성에 관한 내용은 “작업 공간 관리자 환경설정 파일” 260 페이지를 참조하십시오.

2. 버튼 바인딩을 DtButtonBindings 정의에 추가하십시오.

같은 버튼을 누름과 누름 동작의 다른 기능으로 바인드하지 않아야 하고, 또한 둘 이상의 기능을 같은 버튼 및 문맥에 바인드하지 않아야 합니다.

3. 작업 공간 메뉴에서 작업 공간 관리자 재시작을 선택하십시오.

▼ 새 버튼 바인딩 세트 작성 방법

1. 해당 파일을 열어서 편집하십시오.

- 개인 *HomeDirectory*/.dt/dtwmrc

- 시스템 전체 `/etc/dt/config/language/sys.dtwmrc`
이러한 파일 작성에 관한 내용은 “작업 공간 관리자 환경설정 파일” 260 페이지를 참조하십시오.

2. 새로운 버튼 바인딩 세트를 작성하십시오. “버튼 바인딩 구문” 268 페이지를 참조하십시오.

3. 버튼 바인딩 자원을 새 이름으로 설정하십시오.

```
Dtwm*buttonBindings: ButtonBindingsSetName
```

4. 작업 공간 메뉴에서 작업 공간 관리자 재시작을 선택하십시오.

주 - 새 버튼 바인딩이 기존의 버튼 바인딩을 교체할 것입니다. 보존하려는 버튼 바인딩을 `DtButtonBindings`로부터 복사하십시오.

키 바인딩의 사용자 정의

키 바인딩이라고도 하는 키보드 바인딩은 작업 공간 관리자 기능들을 키 조합과 연결시킵니다. 키 바인딩은 모든 작업 공간에 적용됩니다.

주 - 공통 키 조합을 키보드 바인딩으로서 사용할 때는 주의해야 합니다. 예를 들어, `Shift-A`는 대개 현재 윈도우에 문자 "A"를 넣습니다. `Shift-A`를 한 기능에 바인드할 경우, 이러한 일반 용도가 없어지게 됩니다.

기본 데스크탑 키 바인딩

데스크탑 기본 키 바인딩은 `DtKeyBindings`라는 키 바인딩 세트의 작업 공간 관리자 환경설정 파일에 정의되어 있습니다.

```
Keys DtKeyBindings
{
    ...
}
```

키 바인딩 구문

키 바인딩의 구문은 다음과 같습니다.

```

Keys KeyBindingSetName
{
  [Modifiers]<Key>key_name context function [argument]
  [Modifiers]<Key>key_name context function [argument]
  ...}

```

where:

Modifiers—Ctrl, Shift, Alt, Lock. 여러 개의 보조키가 허용됩니다. 보조키 사이에 빈 칸을 넣어 구분하십시오.

key_name—기능이 대응되는 키. 문자나 숫자 키의 경우, *key_name* 이름이 보통 키 위에 인쇄되어 있습니다. a 키의 이름은 "a"이고, "2" 키의 이름은 "2"입니다. "Tab" 키의 이름은 "Tab"입니다. 마찬가지로, "F3" 키의 이름은 "F3"입니다.

그밖의 키들의 이름은 영문자로 표시됩니다. 예를 들어, plus는 "+" 키의 이름입니다. 시스템 종속 디렉토리에 위치한 파일 `keysymdef.h`에 키 이름에 대한 자세한 정보가 들어 있습니다.

문맥—이 활동이 활성화되기 위해 키보드 포커스를 가져야 하는 요소. 바인딩이 둘 이상의 문맥에 바인딩되는 경우, 함께 병합할 수 있습니다. 여러 개의 문맥은 "|" 문자로 구분하십시오.

root—작업 공간 배경; **window**—클라이언트 윈도우; **icon**—아이콘

function—윈도우 관리자 함수. 유효한 함수 목록을 보려면 `dtwmrc(4)` 매뉴얼 페이지를 참조하십시오.

인수—요구되는 윈도우 관리자 함수 인수. 자세한 정보는 `dtwmrc(4)` 매뉴얼 페이지를 참조하십시오.

예를 들어, 다음과 같은 키 바인딩을 통해 Alt+F6을 눌러 응용 프로그램의 다음 임시 윈도우로 키보드 포커스를 전환할 수 있습니다.

```
Alt<Key>F6      window      f.next_key      transient
```

주 - 버튼 바인딩 구문에 대한 모든 정보는 `dtwmrc(4)` 매뉴얼 페이지를 참조하십시오.

▼ 사용자 키 바인딩 세트 작성 방법

1. 해당 파일을 열어서 편집하십시오.

- 개인 `HomeDirectory/.dt/dtwmrc`
- 시스템 전체 `/etc/dt/config/language/sys.dtwmrc`

이러한 파일 작성에 관한 내용은 "작업 공간 관리자 환경설정 파일" 260 페이지를 참조하십시오.

2. 고유한 *KeyBindingSetName*로 설정된 새로운 키 바인딩을 작성하십시오. 참조로 데스크탑 기본 키 바인딩 세트인 *DtKeyBindings*을 사용하십시오.
3. 키 바인딩 자원을 새로운 세트명으로 설정하십시오.
`Dtwm*keyBindings: KeyBindingSetName`
4. 작업 공간 메뉴에서 작업 공간 관리자 재시작을 선택하십시오.

주 - 새로운 키 바인딩이 기존의 키 바인딩을 교체할 것입니다. 보존하려는 키 바인딩을 *DtKeyBindings*로부터 새로 작성한 세트로 복사하십시오.

기본 동작과 사용자 정의 동작 사이의 전환

Motif 기본 동작과 CDE 데스크탑 윈도우 동작 사이를 전환하려면 다음을 수행하십시오.

1. **Alt+Shift+Ctrl+!**를 누르십시오.
2. 대화 상자에서 확인을 누르십시오.

기본 동작으로 전환하면 프론트 패널과 사용자 키 및 버튼 바인딩이 제거됩니다.

응용 프로그램 자원, 글꼴, 색상의 관리

스타일 관리자를 사용하거나 추가 글꼴 및 색상 자원을 사용자 정의하여 표시장치에 맞는 색상과 글꼴 범위를 선택할 수 있습니다. 이 장에서는 글꼴 및 색상 자원을 사용자 정의하는 방법을 설명합니다.

이 장에서는 또한 데스크탑 문서 편집기(dtpad)와 우편 시스템(dtmail) 등의 DtEditor 위지트 응용 프로그램을 위한 스타일 변환을 지정하는 방법과 이러한 변환과 충돌하는 DtEditor 위지트 응용 프로그램 메뉴 가속키에 대한 대체키를 지정하는 방법도 설명합니다.

- “응용 프로그램 자원 설정” 273 페이지
- “UNIX 바인딩 정의” 275 페이지
- “글꼴 관리” 279 페이지
- “색상 관리” 283 페이지
- “응용 프로그램 윈도우의 음영 농도 설정” 290 페이지

응용 프로그램 자원 설정

자원은 응용 프로그램에서 모양과 동작에 대한 특정한 양상을 설정하는데 사용됩니다. 예를 들어, 스타일 관리자(dtstyle)는 시스템이 색상 팔레트에 관한 정보가 있는 파일을 어디에서 찾을지 지정할 수 있는 자원을 제공합니다.

```
dtstyle*paletteDirectories: /usr/dt/palettes/C HomeDirectory/.dt/palettes
```

데스크탑 응용 프로그램에 대한 App-default 파일은 /usr/dt/app-defaults/language 디렉토리에 위치하고 있습니다.

▼ 시스템 전체 자원 설정 방법

- 자원을 `/etc/dt/config/language/sys.resources` 파일에 추가하십시오. (파일을 작성해야 할 수도 있습니다.)

예를 들어, `/etc/dt/config/C/sys.resources`에서 다음과 같이 지정하는 경우,

```
AnApplication*resource: value
```

다음 번 로그인시 자원 `AnApplication*resource`이 각 사용자의 `RESOURCE` `_MANAGER` 특성에 설정됩니다.

주 - 스타일 관리자에 대한 자세한 내용은 `dtstyle` 매뉴얼 페이지를 참조하십시오. 우편 시스템 자원에 대한 자세한 내용은 `dtmail` 매뉴얼 페이지를 참조하십시오.

▼ 개인 자원 설정 방법

1. 자원을 파일 `HomeDirectory/.Xdefaults`에 추가하십시오.
2. 데스크탑_제어기의 추가 폴더에 있는 자원 재로드를 두 번 누르십시오.

데스크탑이 자원을 로드하는 방법

세션 관리자가 세션 시작할 때 자원이 로드됩니다. 세션 관리자가 자원을 `RESOURCE` `_MANAGER`에 로드하는 방법에 대해서는 “세션 자원 로드” 45 페이지를 참조하십시오.

프로세스 관리자 자원

다음과 같은 프로세스 관리자 자원을 사용할 수 있습니다.

- `sampleNowTR`
- `postPopupMenuTR`
- `selectNextProcessTR`
- `selectPrevProcessTR`
- `selectFirstProcessTR`
- `selectLastProcessTR`
- `killSelectedProcessTR`

UNIX 바인딩 정의

기본적으로 UNIX 바인딩은 사용가능하지 않습니다.

▼ EMACS 스타일 변환을 지정하는 방법

다음 절차에 따라 다음 사항을 지정할 수 있습니다.

- 데스크탑 문서 편집기(`dtpad`)와 우편 시스템(`dtmail`) 등의 `DtEditor` 위지트 응용 프로그램을 위한 EMACS 스타일 변환
- 이러한 변환과 충돌하는 `DtEditor` 위지트 응용 프로그램 메뉴 가속키에 대한 대체 키

1. 다음 명령행을 `HomeDirectory/.Xdefaults` 파일에 추가하십시오.

```
#include "/usr/dt/app-defaults/language/UNIXbindings"
```

여기서 `language`는 `LANG` 환경 변수의 값입니다.

2. 세션을 다시 시작하십시오.

▼ EMACS 스타일 변환을 수정하는 방법

1. 파일 `/usr/dt/app-defaults/language/UNIXbindings`의 내용을 `HomeDirectory/.Xdefaults`에 삽입합니다.
2. `.Xdefaults` 파일에 있는 바인딩을 편집합니다.
3. 작업이 완료되면 세션을 다시 시작하십시오.

UNIXBindings 파일이 제공하는 UNIX 바인딩

`/usr/dt/app-defaults/language/UNIXbindings` 파일은 다음 표에 설명된 바인딩을 제공합니다.

주 - UNIX 바인딩이 활성화되면, `Delete` 키는 앞의 문자를 삭제하고, `Shift-Delete` 키는 다음 문자를 삭제합니다.

표 17-1에는 UNIX 바인딩과 충돌하는 메뉴 가속키와 가속키 텍스트에 대한 `dtpad` 대체키가 나열되어 있습니다.

표 17-1 dtpad 대체키

메뉴 가속키와 가속키 텍스트	대체키
Dtpad*fileMenu.print.acceleratorText:	
Dtpad*fileMenu.print.accelerator:	
Dtpad*editMenu.undo.acceleratorText:	Ctrl+_
Dtpad*editMenu.undo.accelerator:	Ctrl<Key>_
Dtpad*editMenu.paste.acceleratorText:	Shift+Insert
Dtpad*editMenu.paste.accelerator:	Shift<Key>osfInsert
Dtpad*editMenu.findChange.acceleratorText:	Ctrl+S
Dtpad*editMenu.findChange.accelerator:	Ctrl<Key>s

표 17-2는 UNIX 바인딩과 충돌하는 메뉴 가속키와 가속키 텍스트에 대한 dtmail 구성 윈도우 대체키가 나열되어 있습니다.

표 17-2 dtmail 구성 윈도우 대체키

메뉴 가속키와 가속키 텍스트	대체키
Dtmail*ComposeDialog*menubar*Edit.Undo.acceleratorText:	Ctrl+_
Dtmail*ComposeDialog*menubar*Edit.Undo.accelerator:	Ctrl<Key>_
Dtmail*ComposeDialog*menubar*Edit.Paste.acceleratorText:	Shift+Insert
Dtmail*ComposeDialog*menubar*Edit.Paste.accelerator:	Shift<Key>osfInsert
Dtmail*ComposeDialog*menubar*Edit.Find/Change.acceleratorText:	Ctrl+S
Dtmail*ComposeDialog*menubar*Edit.Find/Change.accelerator:	Ctrl<Key>s

다음 변환은 (GNU 스타일) EMACS 제어와 메타키 바인딩에 몇 가지 추가 바인딩을 제 공합니다. 가능할 경우, 변환에서 Shift 키를 일반 바인딩과 조합하여 사용함으로써 조작 방향을 바꿀 수도 있습니다. 예를 들어, Ctrl+Shift+F는 커서를 한 문자 앞으로 옮기는 반면, Ctrl+F는 일반적으로 커서를 한 문자 뒤로 옮깁니다.

추가 바인딩은 다음과 같습니다.

Ctrl+comma—backward-word

Ctrl+Shift+comma—forward-word

Ctrl+period—forward-word

Ctrl+Shift+period—backward-word

Ctrl+Return—end-of-file

Ctrl+Shift+Return—beginning-of-file

GNU EMACS는 delete-next-character()가 아닌 delete-previous-character()를 삭제 키에 연결합니다. Meta+F는 일반적으로 파일 메뉴에 대한 대표키이므로 forward-word()로의 바인딩은 무시됩니다. 다음 단어에 대해 다른 바인딩 중 하나를 사용하십시오(예: Ctrl+마침표).

표 17-3에는 DtEditor.text 변환이 나열되어 있습니다.

표 17-3 DtEditor.text 변환

수정키	키	활동 루틴
c ~s	<키>a:	beginning-of-line()\n\
c s	<키>a:	end-of-line()\n\
c ~s	<키>b:	backward-character()\n\
c s	<키>b:	forward-character()\n\
c ~s	<키>b:	backward-character()\n\
c s	<키>b:	backward-word()\n\
m ~s	<키>b:	backward-word()\n\
m s	<키>b:	forward-word()\n\
c ~s	<키>d:	delete-next-character()\n\
c s	<키>d:	delete-previous-character()\n\
m ~s	<키>d:	kill-next-word()\n\
m s	<키>d:	kill-previous-word()\n\
c ~s	<키>e:	end-of-line()\n\
c s	<키>e:	beginning-of-line()\n\
c ~s	<키>f:	forward-character()\n\
c s	<키>f:	backward-character()\n\
m ~s	<키>f:	forward-word()\n\
m s	<키>f:	backward-word()\n\
c	<키>j:	newline-and-indent()\n\
c ~s	<키>k:	kill-to-end-of-line()\n\
c s	<키>k:	kill-to-start-of-line()\n\
c	<키>l:	redraw-display()\n\

표 17-3 DtEditor.text 변환 (계속)

수정키	키	활동 루틴
c	<키>m:	newline()\n\
c s	<키>n:	process-up()\n\
c ~s	<키>n:	process-down()\n\
c	<키>o:	newline-and-backup()\n\
c ~s	<키>p:	process-up()\n\
c s	<키>p:	process-down()\n\
c ~s	<키>u:	kill-to-start-of-line()\n\
c s	<키>u:	kill-to-end-of-line()\n\
c ~s	<키>v:	next-page()\n\
c s	<키>v:	previous-page()\n\
m ~s	<키>v:	previous-page()\n\
m s	<키>v:	next-page()\n\
c	<키>w:	kill-selection()\n\
c ~s	<키>y:	unkill()\n\
m	<키>]:	forward-paragraph()\n\
m	<키>[:	backward-paragraph()\n\
c ~s	<키>comma:	backward-paragraph()\n\
c s	<키>comma:	forward-paragraph()\n\
m	<키>\\<:	beginning-of-file()\n\
c ~s	<키>period:	forward-word()\n\
c s	<키>period:	backward-word()\n\
m	<키>\\>:	end-of-file()\n\
c ~s	<키>Return:	end-of-file()\n\
c s	<키>Return:	beginning-of-file()\n\
~c ~s ~m ~a	<키>osfDelete:	delete-previous-character()\n\
~c s ~m ~a	<키>osfDelete:	delete-next-character()

글꼴 관리

스타일 관리자 대화 상자를 사용하여 모든 응용 프로그램에 원하는 글꼴 그룹과 크기를 선택할 수 있습니다. 또한 명령행에서 글꼴을 지정하거나 자원을 사용하여 다음 사항을 지정할 수 있습니다.

- 개별 응용 프로그램에 대한 글꼴 자원을 설정할 수 있습니다.
- 글꼴 대화 상자가 사용하게 될 다른 글꼴을 지정할 수 있습니다.

글꼴은 텍스트 문자를 인쇄하거나 표시할 유형 스타일입니다. 데스크탑에는 다양한 스타일과 크기의 글꼴이 들어 있습니다.

비트맵 글꼴은 도트의 매트릭스로 만들어집니다. (기본적으로 스타일 관리자는 비트맵 글꼴만을 구성합니다.) 글꼴은 전체가 하나의 파일에 들어 있습니다. 완전한 범위의 크기와 기울기, 굵기를 가지려면 여러 개의 파일이 필요합니다.

글꼴은 자원의 값과 명령어에 대한 매개변수로서 지정됩니다. XLFD(X 논리적 글꼴 설명) 이름은 원하는 글꼴을 요청하는 방법입니다. 시스템은 주어진 설명과 가장 잘 맞는 글꼴을 찾습니다.

데스크탑 글꼴 자원 설정

스타일 관리자-글꼴 대화 상자에서 텍스트 입력과 레이블 등을 위한 글꼴(최대 7개의 크기)을 선택할 수 있습니다. 또한 글꼴 그룹을 추가하거나 삭제할 수도 있습니다.

글꼴 대화 상자에서 설정되는 자원

글꼴을 선택하면 다음과 같은 자원이 RESOURCE_MANAGER 특성에 기록됩니다.

- SystemFont는 메뉴 표시줄, 메뉴 표시 영역, 누름 버튼, 토글 버튼 및 레이블과 같은 시스템 영역에 사용됩니다. SystemFont는 다음 자원을 설정합니다.
 - *FontList—데스크탑 클라이언트 및 OSF/Motif 툴 키트를 사용하여 작성된 다른 클라이언트의 시스템 영역에 표시됩니다.
- UserFont는 윈도우에 입력되는 텍스트에 사용됩니다. UserFont는 다음 자원을 설정합니다.
 - *Font—X 응용 프로그램의 이전 버전을 지원합니다.
 - *FontSet—1차 설정
 - *XmText*FontList—텍스트 입력란에 표시
 - *XmTextField*FontList—텍스트 입력란에 표시

글꼴 대화 상자에서 사용하는 자원

글꼴 대화 상자의 각 선택에 사용되는 글꼴은 `/usr/dt/app-defaults/Dtstyle` 자원 파일에 지정됩니다. 최대 7개의 크기를 지정할 수 있습니다.

`NumFonts`—글꼴 대화 상자에 있는 글꼴 크기의 수

`SystemFont [1-7]`—`SystemFont`를 위한 글꼴 대화 상자 선택에 특정 글꼴을 지정하는 최대 7개의 자원

`UserFont [1-7]`—`UserFont`를 위한 글꼴 대화 상자 선택에 특정 글꼴을 지정하는 최대 7개의 자원

주 - 이러한 자원의 글꼴은 다양한 표시장치에서 읽을 수 있도록 기본 글꼴을 선택했습니다. 응용 프로그램에 특정 글꼴이 필요하다면, 이러한 데스크탑 글꼴을 변경하기보다는 응용 프로그램 글꼴 자원으로 글꼴을 설정하십시오.

응용 프로그램 글꼴에 대한 자세한 내용은 `DtStdAppFontNames(5)` 및 `DtStdInterfaceFontNames(5)` 매뉴얼 페이지를 참조하십시오.

▼ 사용가능한 글꼴을 나열하는 방법

1. 다음과 같이 입력하십시오.

```
xlsfonts [-options] [-fn pattern]
```

시스템에서 사용가능한 XLFD 이름과 글꼴 별명의 목록이 표시됩니다. 비트맵 글꼴은 14개의 XLFD 필드 전체에 값을 표시합니다. 크기 조절이 가능한 서체는 `PixelSize`, `PointSize`, `ResolutionX` 및 `ResolutionY` 위치에 0을 표시합니다.

2. 특정 글꼴을 확인하려면 `xlsfonts`의 패턴 대조 기능을 사용하십시오. 대조하지 않는 패턴 부분은 와일드카드를 사용하여 대체하십시오.

3. `xlsfonts`가 `dt`로 시작하는 글꼴을 표시하지 않으면, 사용자의 글꼴 경로에 데스크탑 글꼴이 포함되지 않은 것입니다. 다음 명령어를 입력하여 데스크탑 글꼴을 사용 가능한 글꼴에 포함시키십시오.

```
xset +fp directory name
```

여기서 `directory name`은 데스크탑 글꼴이 포함되어 있는 디렉토리입니다. 세션 시작으로 설정되는 기본 위치는 `/usr/dt/config/xfonts/ language`입니다.

추가 정보

- `xset` 및 `xlsfonts` 매뉴얼 페이지에 사용가능한 옵션이 나열되어 있습니다.
- X 윈도우 시스템 사용에서는 글꼴 별명과 `xset` 클라이언트에 대해 설명합니다.

▼ 명령행에서 글꼴을 지정하는 방법

- `-xrm` 명령행 옵션을 사용하여 특정 클라이언트를 위한 글꼴 자원을 지정하십시오. 예를 들어,

```
application name -xrm
"*bitstream-charter-medium-r-normal-8-88-75-75-p-45-iso8859-1"
```

X 논리적 글꼴 설명(XLFD)

대시로 구분된 14개의 다른 특성을 나열하면 글꼴이 지정됩니다. 이것을 X 논리적 글꼴 설명(XLFD)이라고 합니다. 목록에 있는 특성을 * 와일드카드로 바꾸고, 특성에 있는 하나의 문자를 ? 와일드카드로 바꿀 수도 있습니다. 표 17-4에는 글꼴 특성 문자열 사양이 나열되어 있습니다.

특성 문자열 사양의 양식은 다음과 같습니다.

```
"-Foundry-FamilyName-WeightName- Slant-SetwidthName-AddStyleName-PixelSize-
PointSize- ResolutionX-ResolutionY-Spacing- AverageWidth-CharSetRegistry-CharSetCoding
"
```

표 17-4 글꼴 특성 문자열 지정

특성 문자열	정의
<i>Foundry</i>	글꼴 설계자를 식별하는 문자열
<i>FamilyName</i>	글꼴의 상표명을 식별하는 문자열
<i>WeightName</i>	굵기와 같이 글꼴의 상대적 무게를 주는 문자열
<i>Slant</i>	기울어짐의 방향을 설명하는 코드 R (Roman-기울어짐 없음) I (Italic-오른쪽으로 기울어짐) O (Oblique-오른쪽으로 기울어짐) RI (Reverse Italic-왼쪽으로 기울어짐) RO (Reverse Oblique-왼쪽으로 기울어짐)
<i>SetwidthName</i>	압축 또는 확장과 같이 폭을 설명하는 문자열
<i>AddStyleName</i>	글꼴을 고유하게 식별하는데 필요한 추가 정보를 제공하는 문자열
<i>PixelSize</i>	em-square의 크기를 픽셀 단위로 나타내는 정수
<i>PointSize</i>	em-square의 크기를 소수점으로 표시하는 정수
<i>ResolutionX</i>	수평 해상도를 픽셀 단위로 표시하는 정수

표 17-4 글꼴 특성 문자열 지정 (계속)

특성 문자열	정의
<i>ResolutionY</i>	수직 해상도를 픽셀 단위로 표시하는 정수
<i>Spacing</i>	단위 사이의 공백을 지정하는 코드 M (하나의 공백-고정 피치) P (비례적 공백-가변 피치) C (문자 셀)
<i>AverageWidth</i>	평균 폭을 1/10 픽셀 단위로 표시하는 정수
<i>CharSetRegistry</i>	글꼴 코드화를 등록한 등록 권한을 식별하는 문자열
<i>CharSetEncoding</i>	지정된 레지스트리에 설정된 문자를 식별하는 문자열

예

다음 XLFD 이름은 ISO8859-1 표준 코드화를 지원하는 비트스트림으로 만들어진 `charter`라는 글꼴을 설명하고 있습니다.

```
-bitstream-charter-medium-r-normal--8-80-75-75-p-45-iso8859-1
```

이것은 특정한 기울어짐이 없는 중간 굵기의 정상적인 폭입니다. 글꼴은 8 픽셀 또는 8.0 포인트의 `em-square` 상자에 맞는 비례적 형태를 나타냅니다. 수평 및 수직 해상도는 모두 75 픽셀입니다. 문자의 평균 폭은 45 1/10 픽셀 또는 4.5 픽셀입니다.

이 문자열의 일부를 와일드카드로 바꿀 수 있습니다. 시스템은 첫 번째 글꼴을 사용하여 사용자가 지정한 부분과 일치하는 것을 찾습니다.

8 픽셀 `charter` 글꼴만이 필요한 경우, 다음과 같이 사용할 수 있습니다.

```
*-charter-*- *- *- *- *-8-*
```

선택된 글꼴 그룹 속성 보기

스타일 관리자-글꼴 대화 상자에서 속성 버튼을 선택하여 다음과 같은 글꼴 그룹 속성을 볼 수 있습니다.

- 글꼴 그룹
- 크기
- 별명
- 별명 XLFD
- 별명 위치
- 글꼴
- 글꼴 XLFD

사용자 글꼴 그룹 파일 시스템 저장

사용자가 글꼴을 추가하면 다음 위치에 저장됩니다.

`HomeDirectory/.dt/sdtfonts/host/locale/typeface-nnnnnnn`

`host`는 지역 워크스테이션의 호스트 이름입니다.

`locale`은 "C" 또는 "ja"와 같은 사용자의 현재 로캘입니다.

`typeface-nnnnnnn`은 선택된 글꼴에서 파생된 이름 및 고유하게 생성된 숫자입니다.

이 글꼴 그룹 디렉토리에는 다음과 같은 세 파일이 들어 있습니다.

- `fonts.alias`
- `fonts.dir`
- `sdtfonts.group`

`fonts.alias`와 `fonts.dir` 파일은 Xerver의 글꼴 경로에 추가하는데 적합한 일반적인 X11 글꼴 파일입니다. `sdtfonts.group` 파일에는 사용자 지정 글꼴 그룹명이 들어 있습니다.

시스템 관리자 글꼴 그룹 작성

워크스테이션의 다른 사용자들이 글꼴 그룹을 액세스할 수 있도록 시스템 관리자가 글꼴 그룹을 `/etc/dt/sdtfonts/locale` 디렉토리나 `/usr/openwin/lib/X11/stdfonts/locale` 디렉토리에 복사할 수 있습니다. 세션 관리자는 먼저 `HomeDirectory/.dt/stdfonts/host/locale`을 보고, 다음에 `/etc/dt/sdtfonts/locale`을 보고, 마지막으로 `/usr/openwin/lib/X11/stdfonts/locale`을 봅니다.

색상 관리

이 절에서는 다음 내용을 설명합니다.

- 스타일 관리자가 표시장치 색상을 설정하는 방법
- 스타일 관리자가 데스크탑 색상 사용법을 제어하기 위해 사용하는 자원

색상 팔레트

팔레트는 색상 세트 그룹으로 구성됩니다. 현재 팔레트의 색상 세트는 스타일 관리자 색상 대화 상자에 나와 있습니다.

각 팔레트에 대해 파일이 존재합니다. paletteDirectories 자원은 팔레트 파일을 포함하는 파일을 지정합니다. 기본적으로, 이 자원에는 다음과 같은 항목이 포함됩니다.

- 내장 팔레트 /usr/dt/palettes
- 시스템 전체 팔레트 /etc/dt/palettes
- 개인 팔레트 HomeDirectory/.dt/palettes

색상 세트

현재 팔레트에 있는 각 색상 세트는 스타일 관리자 대화 상자에 있는 색상 버튼으로 나타납니다. 각 색상은 색상 세트 ID(1-8 사이의 수)로 식별됩니다.

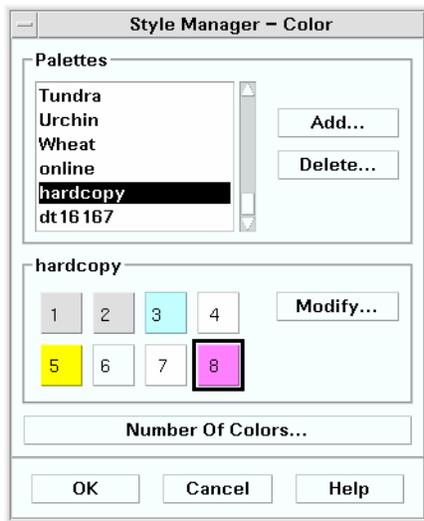


그림 17-1 HIGH_COLOR에 대한 색상 ID 값

각 색상 세트는 최대 5개의 색상으로 구성됩니다. 각 색상 버튼은 색상 세트의 배경 색상을 표시합니다. 각 색상 세트에 있는 5개의 색상은 다음과 같은 표시장치 구성요소 자원을 나타냅니다.

전경색—응용 프로그램 윈도우나 윈도우 프레임의 전경색. 이 색상은 언제나 검은색 아니면 흰색입니다. 일반적으로 윈도우와 제목에 있는 텍스트에 사용됩니다.

배경색—응용 프로그램 또는 윈도우 프레임의 배경색.

topShadowColor—응용 프로그램 제어 도구(누름 버튼 등)와 윈도우 프레임의 상단 및 왼쪽 경사면의 색상.

bottomShadowColor—응용 프로그램 제어 도구 및 윈도우 프레임의 하단 및 오른쪽 경사면의 색상.

`selectColor`—활성 토글 및 버튼과 같은 특정 제어도구의 활성 상태를 나타내는 색상.

각 팔레트에서 사용하는 색상 세트의 수는 `colorUse` 자원이 결정하며, 이 자원은 스타일 관리자-색상의 사용할 색상 수 대화 상자를 사용하여 설정할 수 있습니다.

색상 값 지정

스타일 관리자는 팔레트 파일에 대한 색상 정보를 기록할 때 RGB 값을 사용합니다. RGB 숫자에 대한 구문은 다음과 같습니다.

```
# RedGreenBlue
```

*Red*와 *Green*, *Blue*는 16진수로서, 1-4 자릿수이며, 각 색상이 사용된 양을 나타냅니다. 각 색상의 자릿수는 같아야 합니다. 따라서, 유효한 색상 값은 3, 6, 9, 12자리의 16진수로 구성됩니다.

예를 들어, 흰색은 다음 방법 중 하나로 지정할 수 있습니다.

```
#fff  
#ffffff  
#ffffffff  
#ffffffffffff
```

사용자가 직접 색상 자원을 설정하는 경우, 색상 이름을 사용하거나 RGB 값을 사용할 수 있습니다. 파일 `/usr/lib/X11/rgb.txt`는 이름이 지정된 모든 색상을 나열합니다.

색상 세트가 자원에 매핑되는 방법

데스크탑은 자원을 통해 다양한 표시장치 요소에 색상 세트를 매핑하고, 표 17-5과 같이 지정합니다.

표 17-5 자원에 매핑된 색상 세트

자원	표시장치 요소
<code>activeColorSetId</code>	활성 윈도우 프레임 색상
<code>inactiveColorSetId</code>	비활성 윈도우 프레임 색상
<code>textColorSetId</code>	텍스트 입력란
<code>primaryColorSetId</code>	응용 프로그램의 주 배경 영역
<code>secondaryColorSetId</code>	응용 프로그램의 메뉴 표시줄, 메뉴, 대화상자

이러한 자원은 색상 세트 ID를 자신의 값으로 가집니다. 색상 세트 ID로 표시장치 요소의 색상을 지정하면, 스타일 관리자에서 새로운 팔레트가 선택될 때, 동적으로 요소를 새로운 색상 체계로 변경할 수 있습니다.

개별 응용 프로그램에 이러한 자원을 사용할 수 있습니다. 예를 들어, 다음 행은 사용자가 모든 dtterm 윈도우의 1차 색상으로 색상 세트 8을 사용하여 이러한 윈도우를 시각적으로 그룹화하는 방법을 나타냅니다.

```
dtterm*primaryColorSetId: 8
```

기본 색상 세트 지정

표시장치 요소에 사용되는 색상 세트 ID는 스타일 관리자의 색상 수 설정에 따라 달라집니다.

표 17-6에는 고급 색상의 색상 세트 ID(8개의 색상 세트)가 나열되어 있습니다—스타일 관리자의 데스크탑의 자세한 색상 설정.

표 17-6 고급 색상

색상 세트 ID	표시장치 요소
1	활성 윈도우 프레임 색상
2	비활성 윈도우 프레임 색상
3	미사용(기본값)
4	텍스트 입력란
5	응용 프로그램의 주요 배경 영역
6	응용 프로그램의 메뉴 표시줄, 메뉴 및 대화 상자
7	기본적으로 미사용
8	프론트 패널 배경

표 17-7에는 중간 색상의 색상 세트 ID(4개의 색상 세트)가 나열되어 있습니다—스타일 관리자의 응용 프로그램의 자세한 색상 설정.

표 17-7 중간 색상

색상 세트 ID	표시장치 요소
1	활성 윈도우 프레임 색상
2	비활성 윈도우 프레임 색상
3	응용 프로그램 및 프론트 패널 배경 색상
4	텍스트 입력란

표 17-8에는 저급 색상의 색상 세트 ID(2개의 색상 세트)가 나열되어 있습니다—스타일 관리자의 응용 프로그램의 대부분의 색상 설정.

표 17-8 저급 색상

색상 세트 ID	표시장치 요소
1	활성 윈도우 프레임, 작업 공간 선택 버튼
2	다른 모든 표시장치 요소

스타일 관리자를 사용하여 색상 제어

스타일 관리자를 통해 데스크탑 응용 프로그램과 기타 협동 응용 프로그램용 색상을 동적으로 변경할 수 있습니다. 스타일 관리자가 설정한 전경 및 배경 색상 세트는 비협동 응용 프로그램에서도 사용가능합니다.

스타일 관리자의 색상 변경에 응답하는 클라이언트의 경우, 데스크탑 Motif 라이브러리를 사용하고 있어야 합니다. 다른 도구 키트로 작성한 클라이언트는 스타일 관리자 변경에 대한 응답으로 색상을 동적으로 변경할 수 없습니다. 이러한 클라이언트에 대한 색상 변경사항은 클라이언트를 다시 시작할 때까지 적용되지 않습니다.

해당 클라이언트에 적용되는 다른 특정 색상 자원이 없어야 합니다. 여기에는 사용자 지정 자원, aadefaults 및 응용 프로그램 내장 자원이 포함됩니다.

클라이언트는 primaryColorSetId와 secondaryColorSetId 자원을 지정하여 데스크탑 팔레트에서 특정 색상을 사용할 수 있습니다.

스타일 관리자가 사용하는 색상 수

스타일 관리자가 사용하는 색상의 수는 다음과 같은 자원의 값에 따라 달라집니다.

colorUse—데스크탑이 사용하는 색상의 수를 구성합니다.

shadowPixmaps—데스크탑에게 두 음영 색상을 픽스맵으로 대체하라고 지시합니다.

foregroundColor—전경 색상이 동적으로 변경되는지의 여부를 지정합니다.

dynamicColor—사용자가 팔레트를 전환할 때 응용 프로그램이 색상을 변경하는지의 여부를 제어합니다.

표 17-9에는 데스크탑이 할당하는 최대 색상 수가 나열되어 있습니다.

표 17-9 데스크탑 색상 수

표시장치	색상 최대 수	파생된 수
B_W	2	검은색과 흰색

표 17-9 데스크탑 색상 수 (계속)

표시장치	색상 최대 수	파생된 수
LOW_COLOR	12	2개의 색상 세트 x 5개의 색상 + 검은색과 흰색
MEDIUM_COLOR	22	4개의 색상 세트 x 5개의 색상 + 검은색과 흰색
HIGH_COLOR	42	8개의 색상 세트 x 5개의 색상 + 검은색과 흰색

최대 색상 수를 결정하는 방법

1. 팔레트의 색상 세트 수에 각 색상 세트의 색상 수를 곱하십시오.
2. 2(검은색과 흰색의 경우)를 더합니다.

그러나, 다음과 같은 구성에서는 팔레트에 10개의 색상만을 갖습니다. 즉, 4개의 색상 세트에 각 세트에 2개의 색상(배경색과 selectColor)을 곱한 후에 검은색과 흰색을 더합니다.

```
*colorUse:    MEDIUM_COLOR
*shadowPmaps: True
*foregroundColor: White
```

주 - 다중 색상 아이콘은 14개의 추가 색상을 사용합니다.

colorUse 자원

colorUse 자원의 기본값은 MEDIUM_COLOR입니다. 이 자원의 값은 팔레트에서 사용되는 색상 세트의 수에 영향을 줍니다. 다른 자원들은 음영을 만들기 위해 사용되는 색상에 영향을 줍니다. colorUse 자원의 값도 다중 색상 아이콘의 사용에 영향을 줍니다.

값	설명
B_W	“흑백”스타일 관리자 설정 1개에서 3개의 색상 플레인을 갖는 표시 장치 색상 세트의 수: 2 최대 색상 수: 2 기본 색상 수: 2 다중 색상 아이콘 없음
LOW_COLOR	“응용프로그램의 대부분의 색상”스타일 관리자 설정 4개에서 5개의 색상 플레인을 갖는 표시장치 색상 세트의 수: 2 최대 색상 수: 12 기본 색상 수: 12 다중 색상 아이콘 없음
MEDIUM_COLOR	“응용프로그램의 자세한 색상”스타일 관리자 설정 6개의 색상 플레인을 갖는 표시장치 색상 세트의 수: 4 최대 색상 수: 22 기본 색상 수: 22 다중 색상 아이콘

값	설명
HIGH_COLOR	“데스크탑의 자세한 색상”스타일 관리자 설정 7개 이상의 색상 플레인을 갖는 표시장치 색상 세트의 수: 8 최대 색상 수: 42 기본 색상 수: 42 다중 색상 아이콘
기본값	데스크탑은 표시장치에 대한 현재 값을 선택합니다. (고급 색상 표시장치용 데스크탑에서 사용하는 색상 수를 줄이기 위해 기본 colorUse 자원은 MEDIUM_COLOR로 설정되었습니다).

shadowPixmaps 자원

shadowPixmaps 자원은 데스크탑에 두 가지 음영 색상을 픽스맵으로 바꾸도록 지시합니다. 이러한 픽스맵은 배경 색상을 검은색이나 흰색과 혼합하여 상단 또는 하단 음영 값을 시뮬레이션합니다. 음영색에 색상 셀이 할당되지 않아도 되기 때문에 필요한 색상의 수가 2씩 감소하게 됩니다.

값	설명
참	데스크탑이 음영 색상 대신에 사용할 topShadowPixmap과 bottomShadowPixmap을 작성합니다.
거짓	팔레트의 topShadowColor와 bottomShadowColor를 사용합니다.

shadowPixmaps의 기본값은 사용자가 소유한 colorUse 자원과 표시장치에 대한 하드웨어 지원에 따라 달라집니다.

foregroundColor 자원

foregroundColor 자원은 전경색이 팔레트에 구성되는 방법을 지정합니다.

설정	결과
흰색	전경색이 흰색으로 설정됩니다.
검은색	전경색이 검은색으로 설정됩니다.
동적(기본값)	전경색이 배경색의 값에 따라 검은색 또는 흰색으로 동적으로 설정됩니다. 예를 들어, 노란색 배경의 흰색 글자는 읽기 어렵기 때문에 시스템은 검은색을 선택합니다.

foregroundColor가 검은색이나 흰색으로 설정된 경우, 색상 세트에 있는 색상의 수는 1씩 감소하고, 배경 색상이 변경되어도 전경색은 변경되지 않습니다.

foregroundColor의 기본값은 colorUse의 값이 B_W인 경우를 제외하고 동적입니다.

dynamicColor 자원

dynamicColor 자원은 응용 프로그램의 색상이 동적으로 변하는지 여부, 즉 사용자가 팔레트를 전환할 때 클라이언트가 색상을 변경하는지 여부를 제어합니다.

값	설명
참	새 팔레트가 선택되면 클라이언트가 색상을 동적으로 변경합니다. 이것이 기본값입니다.
거짓	클라이언트가 색상을 동적으로 변경하지 않습니다. 새 팔레트가 선택되면, 세션이 재시작될 때 클라이언트는 새로운 색상을 사용합니다.

dynamicColor 자원의 값이 참일 경우, 색상을 동적으로 변경할 수 없는 클라이언트(비 Motif 응용 프로그램)가 색상을 동적으로 변경할 수 있는 클라이언트의 색상 맵에 다른 셀을 할당합니다. 그러나 사용자는 같은 색상을 보게 됩니다.

주 - 모든 클라이언트가 같은 색상 셀을 공유하므로, dynamicColor를 거짓으로 설정하면 사용자의 데스크탑에서 사용하는 색상의 수가 감소합니다.

응용 프로그램 윈도우의 음영 농도 설정

데스크탑이 버튼 음영과 포커스 강조 표시와 같은 응용 프로그램의 구성요소에 대해 1 픽셀의 기본 음영 농도를 정의합니다. Motif 1.2 응용 프로그램은 이 자원 값을 사용하지만 다른 응용 프로그램이 이 자원 값을 확보하지 않을 수 있기 때문에 표시장치에 다르게 나타날 것입니다.

비 Motif 1.2 응용 프로그램에 대한 음영 농도를 1 픽셀로 설정하는 방법

1. 루트로 로그인하십시오.
2. `/etc/dt/config/language/sys.resources` 파일을 작성하십시오.
3. `/etc/dt/config/language/sys.resources`에 응용 프로그램별 자원을 다음과 같이 지정하십시오.

```
application_class_name*XmCascadeButton*shadowThickness: 1
```

시스템 기본 자원을 대체하는 방법과 모든 데스크탑 사용자를 위한 추가 자원을 지정하는 방법은 “세션 자원 로드” 45 페이지를 참조하십시오.

지역화된 데스크탑 세션 구성

지역화된 데스크탑 세션을 구성하려면 다음과 같은 작업을 수행해야 합니다.

- LANG 환경 변수와 기타 자국어 지원(NLS) 환경 변수를 설정합니다.
- 언어별 메시지 카탈로그와 자원 파일에 액세스합니다.
- 국제화된 시스템을 통해 원격으로 응용 프로그램을 실행합니다.
- “LANG 환경 변수 관리” 291 페이지
- “글꼴 찾기” 294 페이지
- “app-defaults 자원 파일 지역화하기” 294 페이지
- “활동 및 데이터 유형 지역화하기” 295 페이지
- “아이콘 및 비트맵 지역화하기” 295 페이지
- “도움말 볼륨 지역화하기” 297 페이지
- “메시지 카탈로그 지역화하기” 297 페이지
- “지역화된 데스크탑 응용 프로그램을 원격으로 실행하기” 297 페이지
- “사용자 키보드 맵 재설정” 298 페이지

LANG 환경 변수 관리

운영 체제의 언어 감지 루틴을 사용하려면 데스크탑에 대한 LANG 환경 변수를 설정해야 합니다. 데스크탑에서는 다음 언어가 지원됩니다.

- 서유럽의 라틴어 계열 언어.
- 일본어
- 정체 한자
- 간체 한자
- 한국어

주 - 이외에도 데스크탑 구입업체에 따라 다른 언어가 지원될 수도 있습니다.

사용자는 LANG을 운영 체제에서 지원하는 임의의 값으로 설정할 수 있습니다. 로그인 화면의 옵션 메뉴가 지원되는 언어와 지역 목록을 표시합니다.

데스크탑에 대한 LANG의 설정은 다음과 같이 4가지 방식으로 할 수 있습니다.

- Xconfig 파일에서 자원을 편집합니다.
- 로그인 화면에서 옵션 메뉴를 사용합니다.
- 실행 가능한 sh 또는 ksh Xsession.d 스크립트를 작성합니다(Xsession.d 스크립트 사용에 관한 자세한 내용은 “Xsession.d 스크립트를 소스로 사용하기” 42 페이지를 참조하십시오).
- 사용자의 .dtprofile 파일을 편집합니다.

LANG이 설정되면, 데스크탑이 다음과 같은 언어별 파일을 사용하여 지역화 인터페이스를 결정합니다.

색상—/usr/dt/palettes/desc.language

배경—/usr/dt/backdrops/desc.language

복수 사용자를 위한 언어 설정

Xconfig 파일을 사용하여 언어를 설정하는 경우, 로그인 세션이 지역화되고 LANG이 모든 사용자를 위해 설정됩니다. 이는 복수 표시장치 시스템에서 모든 표시장치에 대해 LANG을 변경할 수 있는 유일한 방법입니다. (Xconfig를 수정하려면, /usr/dt/config/Xconfig를 /etc/dt/config/Xconfig로 복사하십시오.)

언어는 다음 행을 /etc/dt/config/Xconfig 에 넣으면 설정됩니다.

```
dtlogin.host_display.language: language
```

예를 들어, 다음 행은 LANG을 표시장치 my_host:0의 Swedish_locale으로 설정합니다.

```
dtlogin.my_host_0.language: Swedish_locale
```

dtlogin 클라이언트가 언어에 적합한 메시지 카탈로그를 읽고 지역화된 로그인 화면으로 가져옵니다. 그러면 dtlogin 클라이언트가 /etc/dt/config/Xresources 자원 파일의 다음 자원을 사용하여 로케일 목록을 결정합니다.

- dtlogin*language
- dtlogin*languageList
- dtlogin*languageName

선택한 언어에 적합한 NLSPATH 환경 변수를 설정하기 위해 Xconfig 파일이 필요할 수도 있습니다. 이러한 경우가 아니거나 사용자 자신이 NLSPATH를 설정하려면 “NLSPATH 환경 변수” 294 페이지를 참조하십시오.

단일 세션에 대한 언어 설정

단일 세션을 위한 언어를 설정하려면 로그인 화면이 옵션 메뉴를 사용하십시오. 로그인 화면이 지역화되고 사용자에게 대해 LANG이 설정됩니다. LANG이 세션을 마감할 때 자신의 기본값(dtlogin에 설정됨)을 반환합니다.

한 사용자를 위한 언어 설정

사용자가 *HomeDirectory/.dtprofile* 파일에서 로그인용의 LANG 설정값을 대체할 수 있습니다. 그러면 로그인 화면이 지역화되지 않고, LANG이 사용자에게 대해 설정되지 않습니다.

- sh 또는 ksh를 사용하는 경우

```
LANG=language  
export LANG
```

- csh를 사용하는 경우

```
setenv LANG language
```

LANG 환경 변수 및 세션 구성

LANG 환경 변수가 사용자의 세션 환경설정 파일을 위해 검색되는 디렉토리명을 변경합니다.

지역화된 세션 환경설정 파일은 다음과 같습니다.

- /usr/dt/config/language/Xresources (로그인 관리자 자원 파일)
- /usr/dt/config/language/sys.font (세션 관리자 자원 파일)
- /usr/dt/config/language/sys.resources (세션 관리자 자원 파일)
- /usr/dt/config/language/sys.session (세션 관리자 자원 파일)
- /usr/dt/config/language/sys.dtwmrc (윈도우 관리자 자원 파일)
- /usr/dt/appconfig/types/language/dtwm.fp (윈도우 관리자 프론트 패널)

다른 NLS 환경 변수 설정

LANG 이외에도, LC_CTYPE과 LC_ALL 같은 다른 NLS 환경 변수도 있습니다. 이러한 변수는 dtlogin 언어 자원과 로그인 화면 옵션 메뉴 두 가지 모두의 영향을 받지 않습니다. 이러한 변수는 다음 파일에서 설정해야 합니다.

- 시스템 전체 변수: /etc/dt/config/Xsession.d
- 개인 변수: *HomeDirectory/.dtprofile*

NLSPATH 환경 변수

NLSPATH 환경 변수는 응용 프로그램이 메시지 카탈로그를 검색하는 디렉토리 경로를 결정합니다. LANG과 NLSPATH 모두 자체 메시지 카탈로그를 사용하도록 설정해야 합니다. 지역화된 메시지의 위치는 “메시지 카탈로그 지역화하기” 297 페이지를 참조하십시오. 대부분의 데스크탑 클라이언트는 시작시 NLSPATH에 대한 경로를 앞에 둡니다.

글꼴 찾기

데스크탑과 함께 포함된 글꼴은 /usr/lib/X11/fonts 디렉토리에 있습니다. 각 디렉토리에는 디렉토리 파일 fonts.dir과 별명 파일 fonts.alias가 들어 있습니다. fonts.dir 및 fonts.alias 파일을 작성하는 방법에 대해서는 mkfontdir 매뉴얼 페이지를 참조하십시오.

서버에서 사용할 수 있는 모든 글꼴을 나열하려면 xlsfonts 명령어를 사용하십시오. 서버에 대한 글꼴의 추가 또는 삭제하려면 xset 명령어를 사용하십시오.

app-defaults 자원 파일 지역화하기

데스크탑 클라이언트용 app-defaults 파일의 기본 위치는 /usr/dt/app-defaults/language입니다. 예를 들어, LANG이 Swedish_locale에 설정되어 있으면, 응용 프로그램은 /usr/dt/app-defaults/Swedish_locale\에 있는 app-defaults파일을 찾습니다. LANG이 설정되어 있지 않을 경우, language는 무시되고, 응용 프로그램은 /usr/app-defaults/C에 있는 자체 app-defaults 파일을 찾습니다.

app-defaults의 위치는 XFILESEARCHPATH 환경 변수를 사용하여 변경됩니다. 예를 들어, app-defaults를 /users로 이동하려면 XFILESEARCHPATH를 /usr/app-defaults/language/classname으로 설정하십시오.

HomeDirectory / .dtprofile에 XFILESEARCHPATH를 설정하면, 값은 모든 데스크탑과 X 클라이언트에 적용됩니다. 사용자가 클라이언트를 XFILESEARCHPATH에서 지정한 디렉토리에 링크하거나 복사하지 않는 한, 어느 클라이언트도 자체 자원 파일을 찾지 않습니다.

활동 및 데이터 유형 지역화하기

주 - `/usr/dt/appconfig` 디렉토리에 있는 파일을 사용자 정의하려면 사용자 정의하기 전에 먼저 파일을 `/etc/dt/appconfig` 디렉토리로 복사하십시오.

활동 및 데이터 유형 정의 파일의 검색 경로에는 다음에 위치하는 언어별 디렉토리가 포함됩니다.

- 개인: `HomeDirectory/dt/types`
- 시스템 전체: `/etc/dt/appconfig/types/language`
- 내장: `/usr/dt/appconfig/types/language`

응용 프로그램 관리자 환경설정 파일의 검색 경로는 다음과 같습니다.

- 개인: `HomeDirectory/dt/appmanager`
- 시스템 전체: `/etc/dt/appconfig/appmanager/language`
- 내장: `/usr/dt/appconfig/appmanager/language`
이 디렉토리의 파일 및 디렉토리 이름은 지역화되어 있습니다.

아이콘 및 비트맵 지역화하기

아이콘을 지역화하려면 아이콘을 아이콘 편집기로 편집한 다음에 저장하십시오.

`/etc/dt/appconfig/icons/language`

아이콘을 다른 디렉토리에 저장하는 경우에는 아이콘을 저장한 디렉토리를 포함할 수 있도록 `XMICONSEARCHPATH` 환경 변수를 설정하십시오. `XMICONBMSEARCHPATH` 환경 변수는 아이콘 검색에 사용되는 경로를 제어합니다.

배경 이름 지역화하기

배경의 지역화는 설명 파일 (`desc.language`와 `desc.backdrops`)을 사용하여 수행합니다. 배경 파일에 대해 고유한 지역화된 디렉토리(예: `/usr/dt/backdrops/language` 등)은 존재하지 않습니다. 모든 로케일은 같은 배경 파일 세트를 사용하지만, 배경의 번역된 이름이 들어있는 자체 `desc.language` 파일을 소유하고 있습니다.

설명 파일에는 번역된 배경 이름에 대한 자원 사양이 들어 있습니다. 예를 들면 다음과 같습니다.

```
Backdrops*Corduroy.desc:           Velours
Backdrops*DarkPaper.desc:         PapierKraft
Backdrops*Foreground.desc:       AvantPlan
```

desc. *language* 파일은 스타일 관리자에서 배경을 표시하기 위해 로컬 *language*에 대한 배경 설명을 검색하는데 사용됩니다. 설명 사양이 있는 경우, 해당 사양이 스타일 관리자 배경 목록에 표시됩니다. 사양이 없을 경우에는 배경 파일 이름이 사용됩니다.

사용자는 *HomeDirectory/.dt/backdrops/desc.backdrops* 파일에서 자신의 배경 설명을 추가할 수 있습니다. 이 파일은 로컬에 상관없이 사용자가 추가하는 모든 배경의 배경 설명을 검색하는데 사용됩니다.

설명 파일의 검색 경로는 다음과 같습니다.

- 개인: *HomeDirectory/.dt/backdrops/desc.backdrops*
- 시스템 전체: */etc/dt/backdrops/desc.language*
- 내장: */usr/dt/backdrops/desc.language*

팔레트 이름 지역화하기

팔레트의 지역화는 설명 파일(desc. *language*와 desc. *palettes*) 사용을 통해 수행됩니다. 특정한 지역화된 디렉토리(예: */usr/dt/palettes/language*)는 존재하지 않습니다. 모든 로컬은 같은 팔레트 파일 세트를 사용하지만 팔레트의 번역된 이름이 들어 있는 자체 desc. *palettes* 파일을 소유하고 있습니다.

설명 파일에는 번역된 팔레트 이름에 대한 자원 사양이 들어 있습니다. 예를 들면 다음과 같습니다.

```
Palettes*Cardamon.desc:           Cardamone
Palettes*Cinnamon.desc:          Cannelle
Palettes*Clove.desc:              Brun
```

desc. *language* 파일은 스타일 관리자에서 팔레트를 표시하기 위해 로컬 *language*에 대한 팔레트 설명을 검색하는데 사용됩니다. 설명 사양이 있는 경우, 스타일 관리자 팔레트 목록에 설명 사양이 표시됩니다. 설명 사양이 없을 경우, 팔레트 파일 이름이 사용됩니다.

사용자는 *HomeDirectory/.dt/palettes/desc.palettes* 파일에 자신의 팔레트 설명을 추가할 수 있습니다. 이 파일은 로컬 여부에 관계없이, 사용자가 추가한 모든 팔레트에 대한 팔레트 설명을 검색하는데 사용됩니다.

설명 파일의 검색 경로는 다음과 같습니다.

- 개인: *HomeDirectory/.dt/palettes/desc.palettes*
- 시스템 전체: */etc/dt/palettes/desc.language*

- 내장: `/usr/dt/palettes/desc.language`

도움말 볼륨 지역화하기

도움말 볼륨을 지역화한 경우, 다음 디렉토리 중 하나에 저장해야 합니다. 처음으로 발견되는 도움말 볼륨이 사용됩니다. 디렉토리는 다음과 같은 순서에 따라 검색됩니다.

- 개인: `HomeDirectory/.dt/help`
- 시스템 전체: `/etc/dt/appconfig/help/language`
- 내장: `/usr/dt/appconfig/help/language`

메시지 카탈로그 지역화하기

메시지 카탈로그를 지역화한 경우, 다음 디렉토리에 저장하십시오.

`/usr/dt/lib/nls/msg/language.`

이 디렉토리에는 `*.cat` 파일이 들어 있습니다.

지역화된 데스크탑 응용 프로그램을 원격으로 실행하기

사용자는 유사하게 지역화된 데스크탑이 설치된, 원격 실행 호스트상의 지역화된 데스크탑 응용 프로그램을 호출할 수 있습니다. 응용 프로그램을 시작할 때, 응용 프로그램을 호출하는 호스트에 있는 NLS 관련 환경 변수의 값이 원격 호스트로 전달됩니다. 그러나 환경 변수에는 호스트 정보가 포함되어 있지 않습니다.

사용자 키보드 맵 재설정

예기치 않은 문자와 동작이 있는 경우, 또는 문자가 표시되지 않거나 문자를 입력할 수 없는 경우에 키보드 맵을 재설정 또는 설치하거나, 또는 사용자의 입력 방법을 변경해야 합니다.

입력 방법은 `LC_CTYPE`, `LANG`, 또는 `LC_ALL` 환경 변수에 의해 결정되거나 `-lang` 옵션에 지정한 언어에 의해 결정됩니다.

예를 들어, 사용자가 로켈이 C인 터미널을 POSIX 셸에서 열고자 하는 경우 다음과 같이 입력합니다.

```
LANG=C dtterm
```

이 새 터미널은 C 입력 방법 및 글꼴을 포함하여 C 로켈을 사용합니다. 특정 언어 키보드를 사용하고 있는 경우, 입력 방법이 입력에 확장 문자를 허용하지 않을 수도 있습니다. 특정 언어 키보드로 C 로켈을 사용하고 있을 때는 터미널을 호출하기 전에 `LC_CTYPE` (또는 `LANG`나 `LC_ALL`) 환경 변수를 적절한 값으로 설정해야 합니다.

예를 들어, 독일어 키보드로 C 로켈을 사용하려면 다음과 같이 입력하십시오.

```
LANG=C LC_CTYPE=DeDEdtterm
```

X 서버를 재설정했고 키맵이 초기화되었으면, `xmodmap` 명령어를 사용하여 서버에서 적절한 키보드 맵을 재설정할 수 있습니다.

dtconfig(1) Man Page

NAME

dtconfig - desktop configuration utility

SYNOPSIS

dtconfig [-d |-e |-kill |-reset |-p |-inetd |-inetd.ow]

DESCRIPTION

Desktop configuration utility. Integrates CDE with the operating system of the underlying platform. System root login privilege is required to use dtconfig.

OPTIONS

-d Disables desktop auto-start feature. At end of boot cycle, platform's native text based login mechanism will be used.

-e Enable's desktop auto-start feature. Desktop login

-kill

Kill desktop (window based) login process and any user sessions associated with it. Return control to system's native text based console.

-reset

Tell desktop (window based) login process to reread

its

configuration file to incorporate any changes.

-p Printer actions for any printer known to platform will

be created if such print actions do not already exist in the platform's actions database. This option is executed automatically at boot time if desktop auto-start has been enabled.

-inetd

Adds /usr/dt/bin daemons to the /etc/inetd.conf file. Specific CDE background daemon setup includes rpc.ttdbserverd (ToolTalk), rpc.cmsd (Calendar Manager), and dtspcd (subprocess control). This

-inetd

option is called automatically by Solaris CDE package installs. This -inetd option is also useful for CDE daemon setup outside of normal Solaris CDE install, including system setup where /usr/dt has simply been mounted from some remote fileserver exporting the /usr/dt directory.

-inetd.ow

Switches the ToolTalk and Calendar Manager daemons (rpc.ttdbserverd & rpc.cmsd) start lines in /etc/inetd.conf back to the older /usr/openwin/bin area. This option is called automatically by Solaris CDE package remove scripts when needed. It is also useful outside of normal Solaris package remove operations when /usr/dt is about to be manually removed or unmounted.

RETURN VALUES

0 Successful completion
>0 Error condition

FILES

/usr/dt/bin/dtconfig location of dtconfig utility

SEE ALSO

dtlogin (1), dtprintinfo (1)

색인

번호와 기호

* 와일드카드 문자, 218
? 와일드카드 문자, 218
%B, 147
%DatabaseHost%, 203
%DisplayHost%, 203
%H, 147
%L, 147
%LocalHost%, 203
%M, 147
%SessionHost%, 203
?Xconfig 파일, 언어 설정, 292

A

actionIcon 자원, 190
ACTIONS 필드, 216
activeColorSetId 자원, 285
ALTERNATE_ICON 필드, 251
ANIMATION 정의, 253
AnswerBook 설명서, 네트워크에서 추가, 112
app-defaults
 데스크탑 응용 프로그램, 273
 언어별, 294
app_root
 참조 응용 프로그램 루트 디렉토리
 /appconfig, 100
 /app-defaults, 100
Apple Macintosh 응용 프로그램 환경, 114
Arg_1 구문, 162
ARG_CLASS 필드, 192
ARG_COUNT 필드, 192, 201, 202

ARG_MODE 필드, 192
Arg_n 구문, 194
ARG_TYPE 필드, 192, 201, 216
 인쇄, 143

B

/bin, 100
blank 유형 제어도구, 248
.bm 파일 이름 확장자, 226
bottomShadowColor 자원, 284
BOX 정의, 237
 구문, 238
BROADCAST, XDMCP 간접에서 사용, 26
-broadcast 플래그, 110

C

C 로케일, 109
CDE-MIN 파일, 127
CDE-TT 파일, 127
CHOOSER 문자열, 26
CLIENT_GEOMETRY 필드, 252
CLIENT_NAME 필드, 251, 252
colorUse 자원, 46, 287, 288
COMMAND 활동, 184
 실행 문자열, 193
 예, 186
 필수 필드, 193
/config, 100
CONTAINER_NAME 필드, 238, 239, 243

CONTAINER_TYPE 필드, 238
 CONTENT 필드, 218, 222
 CONTROL_BEHAVIOR 필드, 257
 CONTROL 정의, 구문, 238
 COPY_TO_ACTION 필드, 217
 cpp 문, 48
 current.old 디렉토리, 50

D

DATA_ATTRIBUTES
 구문, 213
 정의, 212, 215
 DATA_CRITERIA
 DATA_ATTRIBUTES와 쌍을 이룸, 212
 구문, 213
 복수, 223
 정의, 212, 218
 DataBaseHost 키워드, 203
 DELETE 필드, 236
 DESCRIPTION 필드, 189, 215
 /dev/console, 101
 DISPLAY_displayname 매크로, 48
 DISPLAY 변수, 로그인 관리자가 설정, 34
 DisplayHost 키워드, 203
 DROP_ACTION 필드, 250
 DROP_ANIMATION 필드, 253
 dt 파일, 185
 dtaction
 구문, 205
 사용자를 변경하기 위해 사용, 206
 dtappgather, 43, 59
 dtappintegrate, 89
 구문, 90
 기능, 91
 응용 프로그램 제거, 66
 DTAPPSEARCHPATH 변수
 정의, 146
 조합, 148, 149
 DtButtonBindings, 268
 dtchooser 파일, 37
 dtconfig(1) man page, 299
 dtconfig 명령어, 27
 dtCreate
 참조 활동
 DTDATABASESEARCHPATH 변수
 사용, 191

DTDATABASESEARCHPATH 변수 (계속)
 정의, 146
 조합, 152
 DtEditor, 스타일 변환, 275
 Dterrors 파일, 27
 dtgreet 파일, 36
 DTHELPSEARCHPATH 변수
 정의, 146
 조합, 155
 dtlogin
 참조 로그인 관리자
 Dtlogin*language resource, 32
 dtlp, 111
 dtmailpr, 111
 DTMOUNTPOINT 변수
 사용자가 상속, 129
 사용하는 프로세스, 128
 설정, 128
 필요로 하는 프로세스, 128
 Dtpid 파일, 21
 .dtprofile, 53, 55
 .dtprofile 파일
 작성, 41
 LANG 설정, 293
 환경 변수 설정, 47
 소스로 사용하기, 41
 구문, 48
 DtRootMenu, 266
 dtsearchpath, 43, 146, 191
 dtsmcmd 명령어, 49
 DTSOURCEPROFILE 변수, 44
 dtspcd, 127, 129
 인증 디렉토리, 124, 130
 환경설정, 129
 DTSPSYSAPPHOSTS 변수
 구문, 148
 수정, 65
 정의, 146
 DTSPSYSDATABASEHOSTS 변수
 구문, 152
 정의, 146, 151
 DTSPSYSHELP 변수, 155
 구문, 155
 정의, 146
 DTSPSYSICON 변수
 구문, 154
 정의, 146

DTSPUSERAPPHOSTS 변수
 구문, 148
 수정, 65
 정의, 146

DTSPUSERDATABASEHOSTS 변수
 구문, 152
 정의, 146, 151

DTSPUSERHELP 변수
 구문, 155
 정의, 146

DTSPUSERICON 변수
 구문, 154
 정의, 146

dtstart_appgather 변수, 43
 dtstart_searchpath 변수, 43
 dtstart_ttsession 변수, 44
 dtterm, 56
 dtwm,
 참조 작업 공간 관리자
 dtwm.fp 파일, 234
 dtwmfp.session 파일, 235
 dtwmrc 파일, 260
 편집, 260
 Dtwmrc 편집 활동, 260
 dynamicColor 자원, 46, 287, 290

E

EMACS 변환, 275
 EMACS 스타일 변환, 275
 /etc/dt, 100
 /etc/rmmount.conf, 114
 /examples, 100

EXEC_HOST
 참조 실행 호스트

EXEC_HOST 필드, 203
 기본값, 135, 204
 다중 값, 135
 데이터베이스 검색 경로에 의한 영향, 135

EXEC_STRING, 111
 EXEC_STRING 필드
 참조 실행 문자열

F

fail-safe 세션, 33

FILE_NAME 필드, 250, 251
 FontSet 자원, 279
 foregroundColor 자원, 46, 289
 fp_dynamic 디렉토리, 234

G

getty, 23, 40
 GID, 124

H

HELP_STRING 필드, 254
 HELP_TOPIC 필드, 254
 HIGH_COLOR, 288
 home.old 디렉토리, 50
 HOME 변수, 34

I

ICON 필드
 데이터 유형에 대한, 215
 유효한 값, 215
 프론트 패널, 243
 허용되는 값, 190

inactiveFrameColorId 자원, 285
 /include, 100
 Include 문, 작업 공간 관리자 파일에서, 261
 -indirect 옵션, 24
 inetd.conf, 129
 IS_TEXT 필드, 217

L

LABEL 활동 필드, 189
 LANG 변수, 291
 .dtprofile 내, 293
 데이터 유형에 대한 영향, 223
 로그인 관리자가 설정, 34
 /lib, 100
 LINK_TO_ACTION 필드, 217
 localTerminal 자원, 200
 LOCKED 필드, 236
 .login, 54

.login 파일, 41
login 계정, 123
.login 파일
 로그인 관리자가 읽지 않음, 48
 소스로 사용하기, 44
LOGNAME 변수, 34
LOW_COLOR, 288
lp
 명령어, 125
 인쇄 스플러, 125
LPDEST 변수, 142

M

mailx, 125
/man, 100
MAP 활동, 165
 예, 187
 정의, 184
MEDIA 필드, 217
MEDIUM_COLOR, 288
MIME_TYPE_MEDIA 필드, 217
mkfontdir 명령어, 파일 컴파일, 294
MODE 필드, 218
 구문, 220
MODE 필드의 AND 연산자, 221
MODE 필드의 NOT 연산자, 221
MODE 필드의 OR 연산자, 221
MONITOR_TYPE 필드, 251
Motif 윈도우 관리자
 참조 작업 공간 관리자
MOVE_TO_ACTION 필드, 217

N

NAME_PATTERN 필드, 218
NFS, 124
NLS 원격 실행, 297
NLS 환경 변수, 293
NO_STDIO 윈도우 지원, 199
NoPrint 활동, 167
NUMBER_OF_ROWS 필드, 255

O

-once 옵션, 110
OpenWindows, 환경, 101

P

/palettes, 100
PANEL_GEOMETRY 필드, 256
PANEL 정의, 구문, 238
PATH_PATTERN 필드, 218
 구문, 219
PATH 변수, 194
 로그인 관리자가 설정, 34
PERM_TERMINAL 윈도우 지원, 199
.pm 파일 이름 확장자, 226
POSITION_HINTS 필드, 242
primaryColorSetId 자원, 285, 287
.profile, 55
.profile 파일, 41
 로그인 관리자가 읽지 않음, 48
 소스로 사용하기, 44
PUSH_ACTION 필드, 249
PUSH_ANIMATION 필드, 253

Q

-query 옵션, 24

R

readme 파일, 88
RESOURCE_MANAGER 특성, 45, 48
rgb.txt 파일, 285
RGB 값, 285
RGB 색상 값, 285
rpc.cmsd, 130
rpc.ttdbserver, 127, 128

S

.sdl 파일, 82
sdtcm_convert script, 112
secondaryColorSetId 자원, 285, 287
selectColor 자원, 285

sendmail, 125
 sessionetc 파일, 50
 sessionexit 파일, 50
 SessionHost 키워드, 203
 shadowPixmap 자원, 46, 289
 /share, 100
 SHELL 변수, 로그인 관리자가 설정, 34
 Solaris CDE, 데스크탑 강제 종료, 102
 SPC, 129
 보안, 129
 startlog 파일, 51
 stty, 55
 SUBPANEL 정의, 237
 SWITCH 정의, 239
 sys.dtpfile 파일, 41
 sys.dtwmrc 파일, 260, 261
 sys.resources 파일, 45, 48, 274
 sys.session 파일, 47, 49
 systemPath 자원, 35

T

TERMINAL 윈도우 지원, 199
 textColorSetId 자원, 285
 ToolTalk
 데이터베이스 서버
 참조 rpc.ttdbserver
 메시지 표시 데몬, 41, 44
 응용 프로그램, 활동, 208
 활동
 참조 TT_MSG 활동
 ToolTalk 메시지 서버
 참조 ttsession
 topShadowColor 자원, 284
 tset, 55
 TT_MSG 활동
 작성, 208
 키워드, 208
 ttsession, 130
 시작, 44
 tty 설정값, 56
 ttyModes, 56
 TYPE 필드, 248
 TZ 변수, 34, 35

U

UID, 124
 UNIX 키 바인딩, 273, 275
 UNIXbindings 파일, 275
 user-prefs.dt 파일, 68
 USER 변수, 34
 userPath 자원, 35
 /usr/dt, 99

V

 /var/dt, 100

W

WINDOW_TYPE 필드, 199
 WM_CLASS 특성, 230
 wmStartupCommand 자원, 46
 workspaceCount 자원, 262
 writeXrdbColors 자원, 46
 wscon 콘솔 로그 파일, 54

X

X400_TYPE 필드, 217
 X11 서버, 101
 X 논리적 글꼴 설명, 279
 X 서버
 액세스 변경, 31
 환경 변경, 32
 X 인증, 126
 X 자원
 참조 자원
 X 터미널, 107, 109, 126
 CHOOSER 문자열, 26
 Xaccess 목록, 25
 XDMCP 간접, 24, 26
 XDMCP 직접, 24, 25
 XDMCP를 지원하지 않는 표시장치, 24
 로그인 서버 구성의 예, 21
 사용, 108
 세션 서비스 얻기, 120
 워크스테이션으로, 109
 X 터미널 사용, 108
 Xaccess 파일, 25

XAUTHORITY 변수, 로그인 관리자가 설정,
 34
 Xconfig, 101, 105
 Xconfig 파일
 수정, 21
 자원 설정, 30
 Xdefaults, 55, 56
 Xdefaults 파일, 45, 274
 XDM, 101, 108, 110
 프로토콜, 108
 XDMCP, 19
 간접 액세스, 26
 간접 요청, 24, 36
 정의, 24
 조회 모드, 24
 직접 액세스, 25
 직접 요청, 24
 Xerrors 파일, 26
 Xfailsafe 파일, 33, 35, 36
 xlsfonts 명령어
 서버에서 글꼴 나열, 294
 설치, 294
 XMICONBMSEARCHPATH 변수
 사용, 153
 정의, 146
 조합, 154
 XMICONSEARCHPATH 변수
 사용, 153
 정의, 146
 조합, 154
 XmText*FontList 자원, 279
 Xpid 파일, 21
 Xreset 파일, 34
 Xresources 파일, 28, 29
 Xserver, 110
 강제 종료, 102
 Xservers 파일, 101
 구문, 22
 기본, 22
 서버 시작, 22
 지역 표시장치 관리, 36
 Xsession
 스크립트, 53
 Xsession.d 디렉토리, 41, 47
 사용자 정의, 42
 스크립트, 42
 Xsession 파일, 41
 PATH 설정, 35

Xsession 파일 (계속)
 로그인 서버에 의해 실행, 33
 세션 관리자 시작, 40
 시스템 전체 사용자 정의, 42
 Xsetup 파일, 33
 Xstartup 파일, 33
 xterm, 56
 시작, 103
 XUSERFILESEARCHPATH 변수, 294

개

개인 데이터 유형 및 활동, 작성, 214
 개인 응용 프로그램 그룹, 64

검

검색 경로
 데스크탑에 의해 정의, 145
 도움말, 154
 설정, 146
 세션 관리자가 설정, 43
 아이콘, 227
 응용 프로그램, 58, 148
 입력 변수, 146
 지역화, 156
 출력 변수, 146
 프론트 패널 정의, 234
 현재값, 146
 환경 변수, 146
 활동, 185

경

경로
 사용자, 35
 시스템, 35
 경로 기반 데이터 유형, 219

국

국제화
 app-defaults, 294
 LANG 변수, 291

국제화 (계속)
NLS 환경 변수, 293
글꼴, 294
문제 해결, 298
언어 설정, 292

그
그룹 ID, 124

글
글꼴
1차 디렉토리, 294
mkfontdir 명령어를 사용하여 찾기, 294
X 논리적 글꼴 설명, 279, 281
xlsfonts 명령어, 294
관리, 279
디렉토리 파일을 사용하여 찾기, 294
별명 파일을 사용하여 찾기, 294
비트맵, 279
사용가능한 목록, 280
스타일 관리자에 있는 수, 280
시스템, 스타일 관리자에서, 280
자원, 등록하기 위해 수정, 76
자원, 설정, 279
특성 문자열 사양, 281
글꼴 경로, 109
글꼴 대화 상자, 279
글꼴 자원, 279

기
기본 파일 이름, 190, 215
기호 링크
데이터 유형 기준, 218
등록중에 작성, 91
파일 이름 일관성, 125

날
날짜 유형 제어 도구, 248

내
내용 기반 데이터 유형, 222

네
네트워크화된 데스크탑, 106
네트워킹, 117
X 인증, 126
개요, 117
기본 환경설정, 123
서비스 유형, 118
여러 마운트에서 응용 프로그램 실행, 136
일반적 환경설정 단계, 123
전자우편, 125
클라이언트 및 서버의 환경설정, 126
필요한 파일, 127

놓
놓기 영역
프론트 패널 제어 도구, 250
활동 아이콘, 194
놓은 파일, 허용하는 활동, 194

다
다중 표시장치, 로그인 관리자, 31

데
데스크탑, 네트워크화, 106
데스크탑 검색 경로, 41
데스크탑_도구 응용 프로그램 그룹, 수정, 64
데스크탑 스마트 응용 프로그램, 62
데스크탑 시작
문제, 53
복수 화면, 105
데스크탑 시작 문제점 해결, 53
데스크탑 환경 파일, 113
데이터 유형
dtappintegrate를 사용하여 통합, 91
검색 경로
참조 데이터베이스 검색 경로
경로 기반, 219

- 데이터 유형 (계속)
 - 기준, 218
 - 내용 기반, 222
 - 논기 동작, 166
 - 도움말, 215
 - 두 번 누르기 동작, 166
 - 등록에 필요, 79
 - 모드 기준, 220
 - 범주화 기준, 218
 - 사용 권한 패턴, 178
 - 서버, 133
 - 소개, 157, 163
 - 속성, 215
 - 수동 작성을 위한 요구사항, 171, 211
 - 수동으로 작성, 211
 - 숨겨진, 217
 - 실행 가능, 221
 - 아이콘, 83, 180, 215
 - 아이콘 연결 항목, 228
 - 여러 기준, 223
 - 예, 214
 - 위치 기반, 219
 - 이름 기반, 177, 218
 - 인쇄, 167
 - 읽기 전용, 221
 - 작성 목적, 73
 - 재로드, 188
 - 정의, 213
 - 정의에 있는 변수, 204
 - 지역화, 223
 - 차별화, 218
 - 환경설정 파일, 212
 - 활동 만들기로 작성, 174
 - 활동 만들기의 제한, 171
 - 활동 작성으로 작성, 163
 - 활동 재로드, 188
 - 활동 제한 기준, 201
 - 활동과 연결, 216
 - 활동과의 관계, 164
 - 데이터 유형 이름 텍스트 필드, 176
 - 데이터 유형으로 파일 숨기기, 217
 - 데이터 유형의 와일드카드 문자, 218
 - 데이터베이스
 - 재로드, 188
 - 활동 재로드, 188
 - 데이터베이스 검색 경로, 151, 185
 - 구문, 152
 - 기본, 151
 - 데이터베이스 검색 경로 (계속)
 - 응용 프로그램 검색 경로와 관련, 150, 151
 - 조합, 152
 - 환경 변수, 151
 - 데이터베이스 서버, 122, 135
 - 구성, 133
 - 작성, 133
 - 클라이언트, 134
 - 환경설정, 127
 - 데이터베이스 호스트, 135, 203
- 도**
- 도움말, 82
 - 데이터 유형에 대한, 215
 - 부분 통합, 82
 - 전체 통합, 82
 - 프론트 패널, 254
 - 프린터 아이콘, 142
 - 활동 만들기를 사용하여 지정, 174
 - 활동 파일, 189
 - 도움말 개발자 키트, 82
 - 도움말 검색 경로, 145, 154
 - 구문, 155
 - 기본, 154
 - 응용 프로그램 검색 경로와 관련, 150, 155
 - 조합, 155
 - 환경 변수, 155
 - 도움말 볼륨
 - 등록 패키지에 위치, 82
 - 마스터 도움말 파일, 82
 - 지역화, 297
 - 통합 레벨, 82
 - 도움말 서버, 122
 - 구성, 133
 - 작성, 133
 - 클라이언트, 134
 - 환경설정, 127
 - 도움말 파일
 - dtappintegrate를 사용하여 통합, 91
 - 등록 패키지에 있는, 82
- 등**
- 등록, 87
 - dtappintegrate, 89

등록 (계속)

- 개요, 72
 - 글꼴 수정, 76
 - 도움말 파일, 82
 - 목적, 74
 - 색상 수정, 76
 - 아이콘 요구사항, 83
 - 예, 92, 97
 - 응용 프로그램 그룹, 84
 - 응용 프로그램 루트 디렉토리, 77
 - 일반적 단계, 75
 - 자원 수정, 75
 - 정의, 61, 74
 - 제공되는 기능, 72
 - 필요한 데이터 유형, 79
 - 필요한 활동, 79
- ## 등록 패키지, 61, 87
- readme 파일, 88
 - 디렉토리, 77
 - 목적, 72
 - 응용 프로그램 그룹 내용, 87
 - 응용 프로그램 아이콘, 87
 - 작성 예, 92
 - 정의, 74
 - 프론트 패널 제어 도구, 89

디

- 디렉토리, 데이터 유형 기준, 221

레이

- 레이블
- 프론트 패널 제어 도구, 257
- 활동, 189, 207

로

- 로그인 관리자, 19
- 관리, 36
- 명령어 수행, 33
- 사용자 정의, 19
- 오류, 26
- 자원, 28, 30
- 정의, 19

로그인 관리자 (계속)

- 환경설정 파일, 36
- 로그인 로컬, 109
- 로그인 서버
- 강제 종료, 102
- 개요, 19
- 구성, 21
- 로그인 화면 표시, 19
- 명령행 로그인, 23
- 명령행에서 시작, 20, 102
- 문자 표시 콘솔, 23
- 문제 해결, 26
- 비트맵 표시장치 없음, 22
- 사용 불가능하게 하기, 27
- 사용자 경로, 35
- 사용자 인증, 19
- 세션 시작, 19
- 시간대 변경, 35
- 시스템 부트시 시작, 102
- 시스템 부트시 작동 불가, 102
- 시스템 쉘, 35
- 시작, 20
- 액세스 제어, 25
- 중지, 27
- 지역 표시장치 없이 실행, 22
- 표시장치 연결, 19
- 프로세스 ID, 21
- 프로세스 ID 강제 종료, 27
- 환경, 34
- 환경설정, 126
- 로그인 시작 파일, 53
- 로그인 오류 로그, 54
- 로그인 인증, 53
- 로그인 화면
- X 서버 액세스, 31
- X 서버 환경, 32
- 글꼴, 29
- 기본 언어 변경, 32
- 네트워크 표시장치에 표시, 24
- 동작 변경, 30
- 모양 변경, 28
- 비상 안전 세션 옵션, 103
- 사용자 정의, 28
- 언어 메뉴 내용 변경, 32
- 인사, 29
- 자원, 29
- 종료, 102
- 지역화, 30

로그인 화면 (계속)
 표시장치 종속 동작, 31
 환영 메시지 변경, 29
로컬, C, 109

루

루트 메뉴
 참조 작업 공간 메뉴
루트 윈도우, 263

링

링크, 데이터 유형 기준, 221

마

마우스 바인딩
 참조 버튼 바인딩
마운트, 설치된 CDE, 104
마운트, 응용 프로그램 실행, 136
마운트된 CDE 디렉토리의 마운트 해제, 104

메

메뉴
 작업 공간 관리자, 264
 활동 사용, 159
메시지 카탈로그, 294

명

명령행 로그인, 23

문

문서 편집기, 변경, 67
문자 표시 콘솔, 23
문자열 활동 인수, 196
문제점, 데스크탑 시작, 53

배

배경, 260
 그래픽 이미지 사용, 263
 추가, 263
 파일 위치, 263
배경색 자원, 284
배경없음 설정, 263

버

버튼 바인딩, 268
 구문, 268
 새 세트 작성, 269
 추가, 269
버튼 바인딩 자원, 270

변

변수
 환경, 환경 변수 참조, 47
 활동 정의에서, 204

복

복수 화면, 105

비

비상 안전 세션 옵션, 54
 로그인 화면, 103
비트맵, 226
 검색 경로, 227
 이름 지정 규약, 226
 파일을 찾는 방법, 226, 227
비트맵화된 표시장치, 로그인 서버 실행, 22

사

사용 권한 패턴, 활동 만들기로 지정, 177
사용자, 활동을 위해 변경, 206
사용자 ID, 124

새

새로운, 작업 공간 메뉴 항목, 265

색

색상

shadowPixmaps 자원으로 음영 작성, 289

값, 285

관리, 283

기본, 286

비활성 윈도우 프레임, 285

사용하는 수, 287

색상 세트, 284

아이콘에서 사용, 231

응용 프로그램 윈도우, 285

자원, 284

전경색 지정, 289

제어, dynamicColor 자원으로, 290

제어, 스타일 관리자 사용, 287

텍스트 입력란, 285

팔레트, 283

할당된 최대 수, 287

활성 윈도우 프레임, 285

색상 서버, 41

시작, 46

자원, 46

색상 세트, 284

기본, 286

요소 표시를 위한 매핑, 285

색상 자원, 등록하기 위해 수정, 76

색상 팔레트, 283

서

서버

login, 126

데이터 유형, 133

데이터베이스, 127

도움말, 122, 127

세션, 126

서버, 118

아이콘, 122, 127

유형, 122

응용 프로그램, 118, 127

정의, 118

파일, 118

서버 (계속)

환경설정, 126

활동, 133

서브 프로세스 제어 데몬

참조 dtspcd

서브 프로세스 제어 서비스

참조 SPC

서브패널

구문, 239

기본 패널과 연결, 244

내장 패널 사용자 정의, 245

보관소, 239

삭제 복구, 236

새로운, 245

수정, 244

시스템 전체 사용자 정의, 245

자동닫기 동작 변경, 247

작성, 244

정의, 239

선

선택 메뉴, 159, 165

선택기, 106, 107, 108

설

설치, 디렉토리 위치, 99

세

세션

failsafe, 33

기본, 39, 40

로그아웃시 명령어 실행, 50

마지막에 스크립트 실행, 34

백업, 50

복구, 50

시작, 40

시작시 명령어 실행, 50

오류 기록, 51

자원, 41

정의, 39

첫번째, 49

초기, 40

세션 (계속)
 특정 표시장치, 40, 49
 현재, 39, 40
 홈, 39, 40
 세션 관리자, 40
 검색 경로 설정, 43
 디렉토리, 51
 로그아웃시 명령어 실행, 50
 문제 해결, 51
 세션 백업, 50
 소개, 39
 시스템 전체 사용자 정의, 42
 시작, 40
 오류 로그, 51, 54
 응용 프로그램 모으기, 43
 응용 프로그램 시작, 47
 응용 프로그램 시작 사용자 정의, 47
 자원 로드, 45
 작업 공간 관리자 시작, 46
 추가 명령어 실행, 50
 클라이언트, 45
 파일, 51
 환영 메시지, 41
 세션 디렉토리, 49
 세션 서버
 참조 로그인 서버

셸

셸
 .profile 또는 .login를 소스로 사용하기, 44
 환경 변수, .profile, .login, .dtprofile도 참조하
 십시오., 48
 개인 사용자 정의, 41
 시스템 전체 사용자 정의, 41
 실행 문자열의 구문, 193
 활동에, 196
 활동에서 사용, 196

스

스위치
 참조 작업 공간 스위치
 스타일 관리자
 글꼴 통합, 76
 색상 지정에 사용, 285

스타일 관리자 (계속)
 색상 통합, 76

시

시간대, 변경, 35
 시간대 자원, 35
 시계 유형 제어 도구, 248
 시작 오류 로그, 54

실

실행 문자열, 193
 놓은 파일, 194
 복수 파일 인수, 197
 쉘 구문, 193
 실행 파일 지정, 194
 인수 없음, 194
 일반적 특징, 193
 절대 경로, 194
 파일에 대한 프롬프트, 195, 196
 실행 파일, 데이터 유형 기준, 221
 실행 호스트
 EXEC_HOST 필드에 의해 지정, 203
 구성, 135
 지정, 134
 활동 작성, 203

아

아이콘
 dtappintegrate를 사용하여 통합, 91
 검색 경로
 참조 아이콘 검색 경로
 기본 파일 이름, 215
 데이터 유형, 83, 180, 215
 등록에 필요, 83
 디자인 권장사항, 231
 색상 사용, 231
 서버
 참조 아이콘 서버
 세트 찾기 대화 상자, 180
 연결, 228
 영어 이외의 언어, 295, 296
 응용 프로그램 그룹, 83, 85

아이콘 (계속)

- 응용 프로그램 시작, 83
- 응용 프로그램 윈도우에 연결, 229
- 응용 프로그램 표시, 87
- 응용 프로그램에 대한, 159
- 이름 지정 규약, 226
- 지역화, 295, 296
- 크기 규약, 226
- 파일 관리자를 사용하여 검색, 230
- 파일 관리자를 아이콘 브라우저로 사용, 230
- 파일 이름, 226
- 파일 형식, 226
- 파일을 찾는 방법, 226
- 프론트 패널, 229, 243
- 프린터 이미지, 141
- 활동, 180, 189
- 활동 및 데이터 유형과 연결, 228
- 활동 아이콘, 188
- 아이콘 검색 경로, 153
 - 구문, 154
 - 기본, 153
 - 응용 프로그램 검색 경로와 관련, 150, 153
 - 조합, 154
 - 환경 변수, 153
- 아이콘 서버, 122
 - 구성, 133
 - 작성, 133
 - 클라이언트, 134
 - 환경설정, 127
- 아이콘 설치 제어 도구, 삭제, 247
- 아이콘 세트 찾기 대화 상자, 180
- 아이콘 위치, 56
- 아이콘 유형 제어 도구, 251
- 아이콘 크기, 226

언

- 언어, Xconfig 파일을 사용한 설정, 292
- 언어 메뉴, 사용자 정의, 32
- 언어 메뉴 항목, 109

열

- 열기 활동, 165

오

- 오류 로그
 - login, 54
 - Xsession, 54
 - 세션 관리자, 54
 - 시작, 54
 - 위치, 54

음

- 음선 메뉴, 언어, 109

우

- 우선순위
 - 프론트 패널 조합, 235
 - 활동 데이터베이스 어셈블리, 191
- 우편 유형 제어 도구, 251
- 우편 인쇄, 사용자 정의, 111
- 우편 인쇄 사용자 정의, 111

위

- 위크스테이션, X 터미널로, 109

원

- 원격 실행
 - 응용 프로그램 서버의 구성, 131
 - 응용 프로그램에서 원격인 활동, 134
 - 자국어 지원, 297
 - 활동에 의해, 203
- 원격 파일의 마운트 지점, 128

위

- 위치, 아이콘, 56
- 위치 기반 데이터 유형, 219

원

- 윈도우 관리자, 260
 - 변경, 46
- 윈도우 메뉴
 - 구문, 264
 - 새로운, 267
 - 정의, 264
- 윈도우 메뉴 자원, 264

음

- 음영 농도, 윈도우, 290

응

- 응용 프로그램
 - app_root 디렉토리, 77
 - 검색 경로, 58
 - 기존 응용 프로그램 그룹에 추가, 63
 - 데스크탑 스마트, 62
 - 데이터 유형에 대한 목적, 73
 - 등록
 - 참조 등록
 - 등록, 정의, 61
 - 등록 취소, 66
 - 등록의 기능, 72
 - 등록하지 않고 추가, 62
 - 로그인시 시작, 41, 47
 - 루트 디렉토리, 77
 - 모으기, 60
 - 세션 관리자가 모음, 43
 - 아이콘 작성, 159
 - 여러 마운트에서 지역적으로 실행, 136
 - 요구된 데이터 유형, 79
 - 요구된 활동, 79
 - 응용 프로그램 관리자로 모으기, 58
 - 응용 프로그램 관리자에 추가, 61
 - 재로드, 66
 - 제거, 66
 - 추가 방법, 61
- 응용 프로그램 검색 경로
 - 개인, 65
 - 구문, 148
 - 기본, 64, 148
 - 수정, 65
 - 수정 이유, 64

- 응용 프로그램 검색 경로 (계속)
 - 시스템 전체, 65
 - 우선순위 변경, 149
 - 응용 프로그램을 모으는데 사용, 58
 - 정의, 148
 - 조합, 149
 - 지역화, 156
 - 환경 변수, 148
- 응용 프로그램 관리자
 - 갱신, 66
 - 기호 링크, 59
 - 설명, 57
 - 우선순위 규칙, 59
 - 응용 프로그램 모으기, 43, 58
 - 응용 프로그램 추가, 61
 - 응용 프로그램 통합, 72
 - 일반 관리, 66
 - 파일 시스템 위치, 58
- 응용 프로그램 그룹
 - dtappintegrate를 사용하여 통합, 91
 - readme 파일, 88
 - 개인, 64
 - 관리, 63
 - 기본, 59
 - 내용, 87
 - 데이터 유형, 86
 - 등록 패키지에 작성, 84
 - 명명, 64
 - 모으기, 58
 - 사용자 정의, 64
 - 시스템 전체, 63
 - 아이콘, 83, 84, 85
 - 우선순위, 59
 - 이름, 85
 - 작성 예, 96
 - 정의, 57
 - 활동, 86
- 응용 프로그램 등록
 - 참조 등록
- 응용 프로그램 루트 디렉토리, 77
- 응용 프로그램 서버
 - 관리, 131
 - 구성, 132
 - 응용 프로그램 사용 가능성, 74
 - 정의, 118
 - 추가, 64
 - 클라이언트, 132
 - 클라이언트 구성, 132

- 응용 프로그램 서버 (계속)
 - 표준 환경설정, 132
 - 환경설정, 127
- 응용 프로그램 아이콘, 186
 - 놓은 파일, 162
 - 데스크탑에 필요, 83
 - 두 번 누르기, 162
 - 작성, 160, 188
 - 활동 만들기 사용, 169
- 응용 프로그램 윈도우, 아이콘 연결, 229
- 응용 프로그램 자원
 - 참조** 자원
- 응용 프로그램 재로드 활동, 66
- 응용 프로그램 패키지
 - 참조** 등록 패키지

이

- 이종 표시장치 유형, 23
- 이름 기반 데이터 유형, 218
- 이미지
 - 참조** 아이콘

인

- 인쇄
 - 개념, 143
 - 관리, 139
 - 기본 대상, 142
 - 다른 데이터 유형, 143
 - 데이터 유형을 위해 구성, 167
 - 시험, 125
- 인쇄 관리자, 140
 - 작업 갱신 간격, 140
- 인쇄 스플러, lp, 125
- 인쇄 작업 갱신 간격, 140
- 인수
 - 교환할 수 없는, 활동에 대한, 197
 - 비파일 인수, 196
 - 프롬프트, 195
 - 활동, 162
 - 활동에 대한 수, 201
 - 활동에 대한 제한, 201
 - 활동에 대해 교환할 수 있는, 198
 - 활동에 대해 복수, 197
 - 활동을 위한, 193

- 인증, X, 126
- 인증, 로그인, 53
- 인증 디렉토리, 124, 130

읽

- 읽기 전용 데이터 유형 기준, 221

입

- 입력 방법, 국제화, 298
- 입력 변수, 146

자

- 자국어 지원, 국제화, 291
- 자동 마운터, 128
- 자원
 - app-defaults, 273
 - colorUse, 288
 - foregroundColor, 289
 - shadowPixmaps, 289
 - 개인, 274
 - 글꼴, 279
 - 기본 데스크탑, 45
 - 로드, 45
 - 설정, 48, 273
 - 세션, 41
 - 시스템 전체, 274
 - 언어별, 294
 - 윈도우의 음영 농도, 290
 - 특정 표시장치, 48
- 자원 재로드 활동, 45

작

- 작업 공간
 - 기본 수 변경, 254
 - 배경
 - 참조** 배경
 - 사용자 정의, 262
 - 수, 262
 - 이름, 263

- 작업 공간 관리자
 - Motif로 변경, 272
 - 개인 사용자 정의, 260
 - 기능, 265
 - 다른 파일 포함, 261
 - 메뉴, 264
 - 버튼 바인딩, 268
 - 시스템 전체 사용자 정의, 261
 - 시작, 41, 46
 - 재시작, 262
 - 정의, 260
 - 프론트 패널 관리, 260
 - 환경설정 파일, 260
- 작업 공간 메뉴
 - 구문, 264
 - 메뉴 항목 추가, 265
 - 수정, 266
 - 작성, 267
 - 정의, 264
- 작업 공간 스위치
 - 사용자 정의, 254
 - 작업 공간 수, 254
 - 정의, 237
 - 정의 구문, 239
 - 제어 도구 추가, 255
 - 행 수, 255

전

- 전경색 자원, 284
- 전자우편, 구성, 125

제

- 제목 자원, 263
- 제어 도구
 - 감시, 251
 - 교체, 243
 - 놓기 영역, 250
 - 다른 제어 도구와 교환, 242
 - 단일 인스턴스, 251
 - 레이블 표시, 257
 - 모양, 243
 - 복구, 236
 - 수정, 242
 - 아이콘, 243

- 제어 도구 (계속)
 - 애니메이션, 253
 - 유형, 248
 - 작성, 248
 - 작업 공간 스위치에서, 255
 - 잠금, 236
 - 정의, 247
 - 주 패널에 추가, 240
 - 클라이언트, 252
 - 토글, 251
 - 파일을 여는, 250
 - 프론트 패널에서 제거, 241
 - 한 번 누르기 와 두 번 누르기, 257
 - 항목 도움말, 254
- 제어 도구에 레이블 표시, 257
- 제어 도구의 감시 유형, 251

지

- 지역 표시 장치 유형, 23
- 지역화, 296
 - 데이터 유형, 223
 - 로그인 화면, 30
 - 메시지 카탈로그, 297
 - 아이콘, 295, 296
 - 팔레트 이름, 296
 - 활동, 207
 - 활동 레이블, 207

출

- 출력 변수, 146

캘

- 캘린더 데몬, 130

클

- 클라이언트
 - 서버, 환경설정, 126
 - 정의, 118
 - 프론트 패널 윈도우, 252

클라이언트 서버 환경설정
참조 네트워크
클라이언트 유형 제어 도구, 248

키

키 바인딩 자원, 272
키 윈도우
구문, 270
기본, 270
새 세트 작성, 271
키 환경설정 파일, 101
키보드 맵, 재설정, 298

터

터미널 에뮬레이션, 56
터미널 에뮬레이터
변경, 67
활동 자동 닫기 옵션, 174
활동에 대한 명령행 옵션, 200
활동에 대해 기본, 200
활동을 위한, 199

특

특성 구분
대화 상자, 176
필드, 179
특정 표시장치 세션, 49
특정 표시장치 자원, 48

파

파일
네트워크에 필요, 127
데이터 유형에 기초한 숨기기, 217
마운트, 124
마운트 지점, 128
분산 액세스, 124
원격 데이터, 127
원격 액세스, 124
이름 일관성, 125
파일, 데이터 유형 기준, 221

파일, 로그인 시작, 53
파일 공유, 124
파일 관리자, 아이콘 브라우저로 사용, 230
파일 관리자를 사용하여 아이콘 검색, 230
파일 서버, 118, 119
파일 유형
참조 데이터 유형
파일 유형 제어 도구, 248, 251
파일 이름, 아이콘, 226
파일 이름 매핑, 128
파일 이름 일관성, 125
파일 인수
활동 만들기와 함께 지정, 173
활동에 사용, 162
파일 프롬프트, 활동 만들기로 지정, 175
파일 이름 데이터베이스 서버, 127

팔

팔레트, 283
이름 지역화, 296

패

패널 정의, 237

표

표시장치 색상, 할당된 최대 수, 287

프

프로토콜, XDM, 108
프론트 패널, 257
개인 사용자 정의 제어, 236
검색 경로, 234
구문, 237
구성요소, 237
놓기 영역 제어 도구, 250
도움말, 254
동적 사용자 정의, 234, 235
등록 패키지에 있는 제어 도구, 89
메뉴, 264
사용자 정의, 233

프론트 패널 (계속)
 새로운, 257
 수정, 240
 아이콘 표시, 229
 애니메이션, 253
 작업 공간 관리자가 관리, 260
 정의 구성, 237
 제어도구
참조 제어도구
 조합 우선순위, 235
 클라이언트, 252
 파일 이름 지정 규약, 234
 행 추가, 257
 화면상 위치, 256
 환경설정 파일, 233
 활동 사용, 158
프론트 패널 복구 활동, 236
프론트 패널의 애니메이션, 253
프린터
 기본, 142
 레이블, 141
 삭제, 140
 아이콘 이미지, 141
 원격 액세스, 125
 작업 갱신 간격, 140
 장치 이름, 125
 추가, 139
 항목 도움말, 142

플
플래그, -broadcast, 110

픽
픽스맵
 검색 경로, 227
 이름 지정 규약, 226
 파일을 찾는 방법, 226, 227

허
 허가 자원, 32

현
현재 세션, 39, 40

홈
홈 디렉토리, 100
 공유, 124
 네트워크화, 125
홈 세션, 39, 40

환
환경 변수
 .dtprofile 내, 41
 .login 또는 .profile를 소스로 사용하기, 44
 개인, 47
 검색 경로, 146
 기본, 41
 내보내기, 47
 데이터베이스 검색 경로, 151
 도움말 검색 경로, 155
 로그인 관리자, 35
 비트맵 검색 경로, 153
 설정, 47
 시스템 전체, 47
 아이콘 검색 경로, 153
 원격 실행, 130
 응용 프로그램 검색 경로, 148
 픽스맵 검색 경로, 153
 활동 정의, 205
환경 파일, 데스크탑, 113
환경설정 파일, 101
 데이터 유형, 212
 등록 패키지에 있는, 74
 로그인 관리자, 36
 세션 관리자, 51
 위치, 100
 윈도우 관리자, 260
 작업 공간 관리자, 260
프론트 패널, 233
 활동, 185
환영 메시지
 기본, 29
 변경, 29
 사용자 정의, 42
 표시, 41, 42

활

활동

- COMMAND, 184
- dtappintegrate를 사용하여 통합, 91
- MAP, 184
- TT_MSG, 184
- 검색 경로
- 참조 데이터베이스 검색 경로
- 교환할 수 없는 인수, 197
- 교환할 수 있는 인수, 198
- 기본 아이콘, 189
- 기본 터미널, 200
- 높은 파일 또는 프롬프트를 허용, 195
- 높은 파일 허용, 162, 194
- 다른 사용자로서 실행, 206
- 다른 활동 실행, 205, 206
- 데이터 없음, 194
- 데이터 유형과 연결, 216
- 데이터 유형과의 관계, 164
- 데이터 유형에 의해 제한, 166, 201
- 등록에 필요, 79
- 레이블, 189, 207
- 매개변수
- 참조 인수
- 매핑, 165
- 메뉴에서 사용, 159
- 문자열 변수, 204
- 복수의 높은 파일 허용, 198
- 비파일 인수, 170, 196
- 서로 다른 두 번 누르기 및 놓기 기능, 202
- 서버, 133
- 소개, 157
- 수동으로 작성, 183, 185
- 수동으로 작성하는 이유, 183
- 수정, 190
- 웹 제공, 196
- 실행 문자열, 193
- 아이콘, 180, 189
- 아이콘 연결 항목, 228
- 아이콘 표시, 188
- 예, 186, 187
- 용도, 163
- 우선순위 규칙, 191
- 원격 응용 프로그램 실행, 134, 203
- 윈도우 지원, 174, 199
- 응용 프로그램에 대한 아이콘 작성, 159
- 이름, 169, 189
- 인수, 193

활동 (계속)

- 인수, 비파일, 196
- 인수 개수 제한, 201
- 인수 없음, 194
- 인수 제한, 201
- 재로드, 188
- 정의에 있는 변수, 204
- 지역화, 207
- 터미널 옵션, 200
- 터미널 지원, 200
- 파일 인수, 162
- 파일 표시,
- 참조 활동 파일
- 파일에 대한 프롬프트, 195
- 편집, 190
- 표시장치 출력 없음, 174
- 프론트 패널에서 사용, 158
- 환경 변수, 205
- 환경설정 파일, 185
- 활동, 183, 185
- 활동 만들기의 제한, 170

활동 검색 경로

- 참조 데이터베이스 검색 경로

활동 만들기, 169

- 기능, 169
- 기본 윈도우, 172
- 데이터 유형 이름, 176
- 데이터 유형 작성, 174
- 사용, 171
- 소개, 169
- 시작, 171
- 아이콘 지정, 180
- 작성된 환경설정 파일, 169
- 제한, 170
- 파일 인수 지정, 173
- 파일 프롬프트, 175
- 활동 명령어 구문, 173
- 활동 이름, 173
- 활동 만들기에 있는 활동 이름 필드, 173
- 활동 만들기의 데이터 유형 목록, 175

활동 서버

- 참조 데이터베이스 서버

활동 아이콘, 160, 188

- 데스크탑에 필요, 83
- 작성, 188

활동 아이콘 제어 도구, 활동 만들기, 173

- 활동 작성, 데이터 유형 작성, 163
- 활동 재로드 활동, 188

활동 정의 파일, 활동 만들기에 의해 작성, 169
활동 정의에 있는 문자열 변수, 204
활동 파일, 169, 188
 내용, 160
 작성, 88, 188
 정의, 160
활동 편집, 190
활동에 대한 명령행, 193
활동에 대한 윈도우 지원사항, 174
활동에 대한 프롬프트, 195