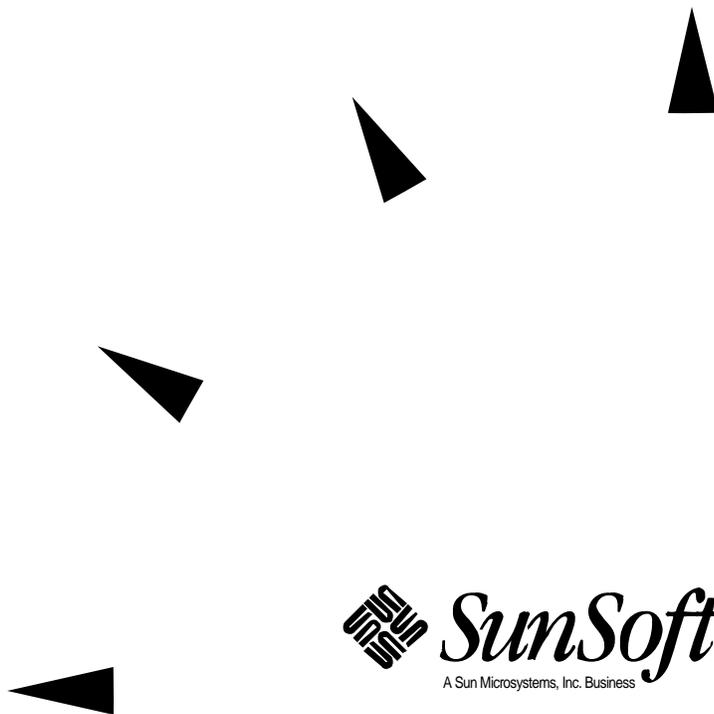


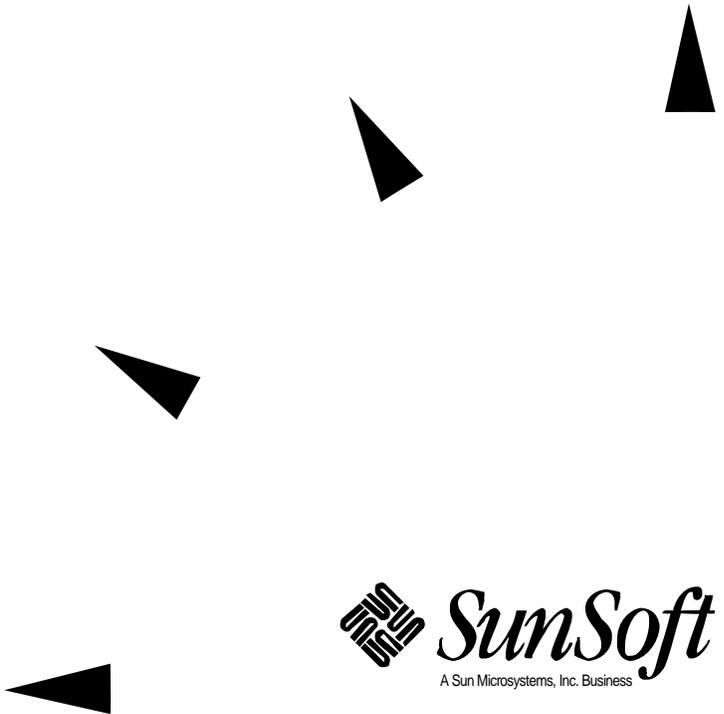
Common Desktop Environment: Benutzerhandbuch für Fortgeschrittene und Systemverwalter

2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

N. parte: 802-4174
Revisione A, dicembre 1995



2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.



 *SunSoft*
A Sun Microsystems, Inc. Business

Einige der von Herstellern und Händlern verwendeten Bezeichnungen zur Kennzeichnung ihrer Produkte sind als Warenzeichen eingetragen. An den Stellen in diesem Buch, wo Bezeichnungen erscheinen, und Addison-Wesley diese als Warenzeichen bekannt waren, wurden die Bezeichnungen hervorgehoben.

Dieses Handbuch wird "wie gesehen" ohne jegliche ausdrückliche oder stillschweigende Garantie einschließlich - jedoch nicht hierauf beschränkt - der Gewährleistung für handelsübliche Qualität, Eignung für einen bestimmten Zweck und Nichtverletzung rechtlicher Bestimmungen geliefert. Dieser Gewährleistungsausschluß gilt nur für die Länder, in denen er zulässig ist.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Änderung des Textes bleibt vorbehalten.

Der Code und die Dokumentation zu DtComboBox und DtSpinBox Widgets wurde von Interleaf, Inc. Copyright (c) 1993 Interleaf, Inc. zur Verfügung gestellt.

Copyright (c) 1993, 1994, 1995 Hewlett-Packard
Copyright (c) 1993, 1994, 1995 International Business Machines Corp.
Copyright (c) 1993, 1994, 1995 Novell, Inc.
Copyright (c) 1993, 1994, 1995 Sun Microsystems, Inc.

Alle Rechte vorbehalten. Dieses Produkt und zugehörige Dokumentationen sind durch das Urheberrecht geschützt. Es wird durch Lizenzvereinbarungen, die seine Verwendung, die Vervielfältigung, und Decompilierung einschränken, vertrieben. Kein Teil dieser Veröffentlichung darf für irgendwelche Zwecke vervielfältigt, in Datenverarbeitungsanlagen gespeichert, oder übertragen werden, unabhängig davon, auf welche Art und Weise oder mit welchen Mitteln, elektronisch, mechanisch, durch Aufzeichnung, durch Fotokopieren oder sonstige Mittel, dies geschieht.



Please
Recycle

Inhaltsverzeichnis

Vorwort.....	xviii
1. Anmeldemanager konfigurieren	1
Anmelde-Server starten.....	2
Lokale Bildschirme und Bildschirme im Netz verwalten.....	3
Prozeß-ID des Anmelde-Servers feststellen	3
Anmeldefenster an einem lokalen Bildschirm anzeigen. . .	4
Anmelde-Server ohne lokalen Bildschirm ausführen.....	5
Auf Anmeldung über Befehlszeile an einem lokalen Bildschirm zugreifen.....	6
Konsole mit zeichenorientierter Anzeige verwenden.....	7
Anmeldefenster an einem Bildschirm im Netz anzeigen .	7
Zugriff auf Anmelde-Server steuern	9
Fehlerprüfung	11
Anmelde-Server stoppen	11
Anmeldefenster	12
Erscheinungsbild des Anmeldefensters ändern	13

Verhalten des Anmeldefensters an einzelnen Bildschirmen ändern	16
Zugriffsrechte auf den X-Server ändern	16
Befehle vor der Anzeige des Anmeldefensters eingeben . .	18
Einfache Motif-Session starten	19
Nach dem Ende der Benutzer-Session	20
Umgebung des Anmelde-Servers	20
Pfad für Benutzer oder System ändern	21
Anmeldemanager verwalten	22
Dateien des Anmeldemanagers	23
2. Sessionmanager konfigurieren	25
Was ist eine Session?	25
Startsession	26
Aktuelle Session	26
Home-Session	26
Bildschirmspezifische Sessions	27
Session starten	27
Start einer Session	27
Prozedur <code>.dtprofile</code> als Ausgangsobjekt verwenden	28
Prozeduren <code>Xsession.d</code> als Ausgangsobjekte verwenden . .	28
Begrüßungsnachricht anzeigen	29
Desktop-Suchpfade einrichten	30
Verfügbare Anwendungen erfassen	31
Mögliche Verwendung der Prozeduren <code>.profile</code> oder <code>.login</code> als Ausgangsobjekte	31

Nachrichtendämon ToolTalk starten	32
Client des Sessionmanagers starten	32
Session-Ressourcen starten.	32
Farb-Server starten	34
Arbeitsbereichssteuerung starten	35
Session-Anwendungen starten	35
Zusätzliche Anpassungen beim Start der Session	36
Zusätzliche Befehle beim Starten einer Session und der Abmeldung ausführen	39
Dateien und Verzeichnisse des Sessionmanager.	41
3. Anwendungen hinzufügen und verwalten	43
Struktur des Anwendungsmanagers.	44
Verzeichnisposition des Anwendungsmanagers.	44
Wie der Anwendungsmanager Anwendungen sucht und zusammenstellt	45
Vorrangsregeln beim Zusammenstellen von Anwendungen	46
Gruppen von Anwendungen, die mit dem Standard- Desktop zur Verfügung gestellt werden	46
Beispiel für die Zusammenstellung von Gruppen von Anwendungen.	47
Anwendungen zum Anwendungsmanager hinzufügen	48
Verschiedene Möglichkeiten, Anwendungen zum Anwendungsmanager hinzuzufügen.	48
Allgemeine Gruppen von Anwendungen erstellen und verwalten	51
Suchpfad für Anwendungen ändern.	53

Standardsuchpfad	53
Anwendungs-Server dem Anwendungssuchpfad hinzufügen	53
Allgemeine Anwendungsmanagerverwaltung	54
Texteditor und den Terminal-Emulator ändern	55
4. Anwendungen registrieren	59
Übersicht über die Anwendungsregistrierung	60
Funktionen, die die Anwendungsregistrierung zur Verfügung stellt	61
Zweck der Anwendungsregistrierung	62
Allgemeine Schritte bei der Registrierung einer Anwendung .	64
Schritt 1: Schriftarten- und Farbressourcen ändern	65
Schriftartenressourcen ändern	65
Farbressourcen ändern	66
Schritt 2: Desktop-Anwendungsstammverzeichnis erstellen .	66
Schritt 3: Registrierungs paketverzeichnisse erstellen	67
Schritt 4: Aktionen und Datentypen für die Anwendung erstellen	70
Aktionen und Datentypen, die für eine Anwendung erforderlich sind	70
Position der Konfigurationsdateien mit den Definitionen für Aktionen und Datentyp	70
Möglichkeiten, Aktionen und Datentypen zu erstellen ...	71
Schritt 5: Hilfedateien dem Registrierungs paket hinzufügen .	73
Schritt 6: Symbole für die Anwendung erstellen	74
Für das Desktop erforderliche Symbole	74

Schritt 7: Die Gruppe von Anwendungen erstellen.	76
Das Verzeichnis für die Gruppe von Anwendungen erstellen	77
Name der Gruppe von Anwendungen.	77
Eindeutige Symbole für Gruppen von Anwendungen definieren	78
Inhalt der Gruppe von Anwendungen erstellen	80
Schritt 8: Anwendungen mit dtappintegrate registrieren.	83
Wie dtappintegrate Anwendungen integriert	84
Beispiel für die Erstellung eines Registrierungspakets.	86
5. Desktop in einem Netzwerk konfigurieren.	95
Übersicht über das Desktop im Netzwerk	96
Typen von Desktop Services im Netzwerk.	96
Typische Netzwerksituationen.	97
Weitere Netzwerksituationen.	99
Zusammenfassung—Servertypen	99
Allgemeine Schritte für die Konfiguration eines Desktops im Netzwerk	100
Netzwerk des Basisbetriebssystems für das Desktop konfigurieren	100
Anmeldeabrechnungen für Benutzer einrichten	101
Zugriff auf verteilte Dateisysteme konfigurieren	102
Zugriff auf ferne Drucker konfigurieren.	103
Elektronische Post konfigurieren.	104
X-Berechtigung konfigurieren	104
Clients und Server für das Desktop konfigurieren	104

Anmelde- und Session-Services konfigurieren	105
Weitere anwendungsbezogene Services konfigurieren.	105
Anwendungs-Services verwalten	111
Suchpfadumgebungsvariablen	112
Anwendungs-Server und dessen Clients konfigurieren.	112
Datenbank-, Symbol- und Hilfe-Services	114
Spezielle Konfigurationen für Anwendungen im Netzwerk	116
6. Verwaltung und Drucken über das Desktop konfigurieren.	121
Drucker hinzufügen und löschen	122
Intervall, in dem Jobs aktualisiert werden, ändern	123
Druckersymbolbilder.	123
Symboldateinamen und -größen	124
Standarddrucker konfigurieren	125
Druckerkonzepte	126
7. Desktop-Suchpfade	127
Desktop-Suchpfade und ihre Umgebungsvariablen	128
Wert eines Suchpfads setzen.	129
Anwendungssuchpfad	131
Standardanwendungssuchpfad	131
Umgebungsvariablen des Anwendungssuchpfads.	131
Syntax für die Eingabevariablen des Anwendungssuchpfads	132
Wie der Wert des Anwendungssuchpfads assembliert wird	132
Rangfolge der systemweiten lokalen Positionen ändern	133

Wie der Anwendungssuchpfad sich auf die Datenbank-, Symbol- und Hilfesuchpfade auswirkt	133
Datenbanksuchpfad (Aktionen/Datentypen)	134
Standarddatenbanksuchpfad	134
Wie der Anwendungssuchpfad sich auf den Datenbanksuchpfad auswirkt	135
Umgebungsvariablen des Datenbanksuchpfads	135
Syntax für die Eingabevariablen des Datenbanksuchpfads	135
Wie der Datenbanksuchpfad assembliert wird	136
Symbolsuchpfad	136
Standardsymbolsuchpfad	136
Wie der Anwendungssuchpfad sich auf den Symbolsuchpfad auswirkt	137
Umgebungsvariablen des Symbolsuchpfads	137
Syntax für die Eingabevariablen des Symbolsuchpfads . . .	137
Wie der Symbolsuchpfad assembliert wird	138
Hilfesuchpfad	138
Standardhilfesuchpfad	138
Wie der Anwendungssuchpfad sich auf den Hilfesuchpfad auswirkt	139
Umgebungsvariablen des Hilfesuchpfads	139
Syntax für die Eingabevariablen des Hilfesuchpfads	139
Wie der Hilfesuchpfad assembliert wird	140
Positionen der Suchpfade	140
8. Einführung in Aktionen und Datentypen	141

Einführung in Aktionen	142
Wie Aktionen Symbole für Anwendungen erstellen.	146
Wie Aktionen Datendateien als Argumente verwenden ..	149
Zusätzliche Verwendung für Aktionen.	150
Einführung in Datentypen	150
Was ist ein Datentyp?.....	150
Wie Datentypen Datendateien mit Aktionen verbinden ..	152
Drucken über das Desktop für einen Datentyp einrichten.	155
9. Aktionen und Datentypen mit 'Aktion erstellen' erstellen.	157
Funktionsweise von 'Aktion erstellen'	157
Einschränkungen von 'Aktion erstellen'.....	158
Einschränkungen bei Aktionen	159
Einschränkungen bei Datentypen	159
Aktionen und Datentypen für eine Anwendung mit 'Aktion erstellen' erstellen	160
Symbol über das Dialogfenster 'Gruppe suchen' angeben.	171
10. Aktionen manuell erstellen	173
Gründe, eine Aktion manuell zu erstellen	174
Befehlsaktionen	174
Zuordnungsaktionen.....	175
ToolTalk-Meldungsaktionen.....	175
Aktionen manuell erstellen: Allgemeine Schritte	175
Konfigurationsdateien für Aktionen.....	176
Beispiel für das Erstellen einer Befehlsaktion	177

Beispiel für das Erstellen einer Zuordnungsaktion.	178
Aktionsdateien (Symbole) für eine Aktion erstellen.	180
Das von der Aktion verwendete Symbolbild angeben	181
Vorrang bei Aktionsdefinitionen	184
Ausführungszeile für eine Befehlsaktion erstellen	185
Allgemeine Merkmale von Ausführungszeilen.	186
Aktionen ohne Argumente erstellen.	187
Aktionen erstellen, die übergebene Dateien akzeptieren . .	187
Aktionen erstellen, die Dateiangaben anfordern.	188
Aktionen erstellen, die übergebene Dateien akzeptieren oder anfordern	188
Aktion erstellen, die Parameter anfordern.	189
Dateiangaben als Zeichenfolge interpretieren	189
Shell-Funktionen in einer Aktion zur Verfügung stellen . .	190
Befehlsaktionen mit mehreren Dateiangaben erstellen	190
Fensterunterstützung und Terminal-Emulatoren für Befehlsaktionen	193
Fensterunterstützung für die Aktion angeben.	193
Befehlszeilenoptionen für den Terminal-Emulator angeben	194
Andere Standard-Terminal-Emulatoren angeben.	194
Aktionen auf bestimmte Argumente einschränken.	195
Aktionen auf einen angegebenen Datentyp einschränken .	195
Aktionen auf der Grundlage der Argumentanzahl einschränken	196
Aktionen auf der Grundlage des Argumentmodus einschränken	197

Aktionen erstellen, die Anwendungen auf fernen Systemen ausführen	198
Aktionen erstellen, die Anwendungen an einem anderen Rechner ausführen.	198
Variablen in Definitionen für Aktionen und Datentypen verwenden	199
Zeichenfolgevariablen in einer Aktion verwenden.	199
Umgebungsvariablen in Aktionen und Datentypen verwenden	200
Aktionen über eine Befehlszeile aufrufen.	200
Syntax von dtaction.	201
Aktionen erstellen, die andere Aktionen ausführen	201
Aktionen erstellen, die als andere Benutzer ausgeführt werden	201
Sprachenabhängige Aktionen erstellen.	202
Positionen für sprachenabhängige Aktionen.	202
Aktionen für ToolTalk-Anwendungen erstellen	203
11. Datentypen manuell erstellen	205
Gründe, einen Datentyp manuell zu erstellen	206
Komponenten einer Datentypdefinition: Kriterien und Attribute	206
Datentyp manuell erstellen: Allgemeine Schritte	207
Konfigurationsdateien für Datentypen	207
Beispiel für die Erstellung einer persönlichen Aktion und eines Datentyps	209
Datenattribute eines Datentyps definieren.	210

Symbolbild angeben, das für einen Datentyp verwendet wird	211
Datentypen Aktionen zuordnen	211
Dateien auf der Grundlage des Datentyps verdecken	212
Verhalten der Datei bei Bearbeitung angeben	213
Datenkriterien für einen Datentyp definieren	213
Datentypenzuordnung auf der Grundlage des Dateinamens	214
Datentypenzuordnung auf der Grundlage der Position	215
Datentypenzuordnung auf der Grundlage des Dateinamens und der Position	216
Dateimodi als Kriterium der Datentypzuordnung verwenden	217
Datentypzuordnung auf der Grundlage des Inhalts	218
Sprachenabhängige Datentypen erstellen	220
Positionen für sprachenabhängige Datentypen	220
12. Symbole für das Desktop erstellen	223
Symbolbilddateien	224
Symboldateiformate	224
Symboldateinamen	224
Symbolgrößenkonventionen	225
Symbolsuchpfad	226
Auf Symbole über das Netzwerk zugreifen	226
Symbolzuordnungen	226
Symboldateien angeben	227
Empfehlungen zur Erstellung von Symbolen	230

Farbpalette	231
13. 0Erweiterte Anpassung des Bedienfelds	233
Bedienfeldkonfigurationsdateien	234
Standardkonfigurationsdatei des Bedienfelds	234
Suchpfad für Bedienfeldkonfigurationsdateien	234
Wie das Bedienfeld assembliert wird: Vorrangregeln	235
Dynamisch erstellte Bedienfelddateien	236
Benutzerschnittstellenanpassungen verwalten	236
Organisation der Bedienfelddefinition	238
Bedienfeldkomponenten	238
Allgemeine Syntax der Bedienfelddefinition	239
Hauptanzeige ändern	242
Symbol angeben, das von einem Bedienelement verwendet werden soll	246
Bedientafeln erstellen und ändern	247
Integrierte Bedientafeln anpassen	248
Bedienelemente im Bedienfeld definieren	251
Definitionen für ein Bedienelement im Bedienfeld	252
Bedienelementtypen	252
Arbeitsbereichsumschalter anpassen	260
Allgemeine Bedienfeldkonfiguration	261
Allgemeine Schritte	262
Beispiel für die Erstellung eines persönlichen Bedienfelds mit drei Zeilen	263
14. Arbeitsbereichsmanager anpassen	265

Konfigurationsdateien des Arbeitsbereichsmanagers.	266
Arbeitsbereiche anpassen	269
Arbeitsbereichsmanagermenüs	271
Syntax des Arbeitsbereichsmanagermenüs	271
Maustastenbelegungen anpassen.	274
Syntax der Maustastenbelegung	275
Tastaturbelegungen anpassen	277
Standardtastenbelegungen des Desktop.	277
Syntax der Tastenbelegung.	277
Wechseln zwischen Standardverhalten und angepasstem Verhalten	279
15. Anwendungsressourcen, Schriftarten und Farben verwalten	281
Anwendungsressourcen einstellen	282
Wie der Desktop Ressourcen lädt	282
UNIX-Belegungen definieren.	283
UNIX-Belegungen über die UNIX-Belegungsdatei.	283
Schriftarten verwalten.	287
Schriftartressourcen des Desktop einstellen.	288
X Logical Font Description (XLFD)	290
Farben verwalten	292
Farbpaletten	293
Farbskalen.	293
Farbe mit dem Umgebungsmanager steuern.	296
Anzahl der vom Umgebungsmanager benutzten Farben. .	297

Schattenstärken für Anwendungsfenster einstellen	301
16. Länderspezifische Desktop-Sessions konfigurieren	303
LANG-Umgebungsvariable verwalten	304
Sprache für mehrere Benutzer einstellen	305
Sprache für eine Session einstellen	305
Sprache für einen Benutzer einstellen	306
LANG-Umgebungsvariable und Session-Konfiguration . .	306
Andere NLS-Umgebungsvariablen einstellen	306
Schriftarten finden	307
Ressourcendateien für Standardwerte von Anwendungen eingrenzen	307
Aktions- und Datenarten eingrenzen	308
Symbole und Bitmaps eingrenzen	308
Namen von Hintergrundmustern eingrenzen	308
Palettennamen eingrenzen	309
Hilfebände eingrenzen	310
Nachrichtenkataloge eingrenzen	310
Fernausführung von länderspezifischen Desktopanwendungen	310
Tastaturbelegung zurücksetzen	311

Vorwort

Dieses Handbuch behandelt die fortgeschrittene Anpassung des Erscheinungsbilds und des Verhaltens der allgemeinen Desktop-Umgebung. Es enthält Kapitel über:

- die Anpassung der Systeminitialisierung und der Anmeldung und Session-Start
- das Hinzufügen von Anwendungen das Bereitstellen von Schnittstellendarstellungen für Anwendungen und ihre Daten
- das Konfigurieren von Desktop-Prozessen, Anwendungen und Daten über das Netzwerk
- die Anpassung von Desktopservices, wie z.B. Fensterverwaltung, Drucken, Farben und Schriftarten

Wer sollte dieses Buch benutzen?

- Systemadministratoren. Viele der Aufgaben in diesem Buch benötigen Zugriffsberechtigung als Benutzer 'root'.
- Fortgeschrittene Benutzer, die Anpassungen durchführen wollen, die nicht mit der Benutzerschnittstelle des Desktop durchgeführt werden können. Das Desktop verfügt über benutzerspezifische Positionen für viele Konfigurationsdateien.

Wie dieses Buch aufgebaut ist

Dieses Handbuch enthält folgende Kapitel:

Kapitel 1, “Anmeldemanager konfigurieren” beschreibt, wie das Erscheinungsbild und das Verhalten des Desktop-Anmeldemanagers konfiguriert wird.

Kapitel 2, “Sessionmanager konfigurieren” beschreibt, wie das Desktop Sessions speichert und abrufen und wie der Session-Start angepasst wird.

Kapitel 3, “Anwendungen hinzufügen und verwalten” beschreibt, wie der Anwendungsmanager Anwendungen zusammenstellt und erklärt, wie Anwendungen hinzugefügt werden.

Kapitel 4, “Anwendungen registrieren” beschreibt, wie ein Registrierungs paket für eine Anwendung erstellt wird.

Kapitel 5, “Desktop in einem Netzwerk konfigurieren” beschreibt, wie Desktop-Services, Anwendungen und Daten über ein Netzwerk verteilt werden.

Kapitel 6, “Verwaltung und Drucken über das Desktop konfigurieren” beschreibt, wie Desktop-Drucker hinzugefügt und entfernt werden und wie der Standarddrucker angegeben wird.

Kapitel 7, “Desktop-Suchpfade” beschreibt, wie das Desktop Anwendungen, Hilfedateien, Symbole und andere Desktopdaten über das Netzwerk findet.

Kapitel 8, “Einführung in Aktionen und Datentypen” gibt eine Einführung in die Konzepte von Aktionen und Datentypen und erklärt, wie sie als Benutzerschnittstelle für Anwendungen benutzt werden.

Kapitel 9, “Aktionen und Datentypen mit ‘Aktion erstellen’ erstellen” beschreibt, wie die Anwendung ‘Aktion erstellen’ benutzt wird, um Aktionen und Datentypen zu erstellen.

Kapitel 10, “Aktionen manuell erstellen” beschreibt, wie Aktionsdefinitionen durch das Bearbeiten einer Datenbankkonfigurationsdatei erstellt werden.

Kapitel 11, “Datentypen manuell erstellen” beschreibt, wie Datentypdefinitionen durch das Bearbeiten einer Datenbankkonfigurationsdatei erstellt werden.

Kapitel 12, “Symbole für das Desktop erstellen” beschreibt, wie der Symboleditor verwendet wird und die Namenskonventionen, Größen und Suchpfade für Desktopsymbole.

Kapitel 13, “Erweiterte Anpassung des Bedienfelds” beschreibt, wie neue systemweite Bedienelemente und Bedientafeln erstellt werden und andere Anpassungen von Bedientafeln.

Kapitel 14, “Arbeitsbereichsmanager anpassen” beschreibt, wie Fenster, Maustastenbelegungen, Tastaturbelegungen und Arbeitsbereichsmanagermenüs angepasst werden.

Kapitel 15, “Anwendungsressourcen, Schriftarten und Farben verwalten” beschreibt, wie Anwendungsressourcen eingestellt werden und wie das Desktop Schriftarten und Farben verwendet.

Kapitel 16, “Länderspezifische Desktop-Sessions konfigurieren” beschreibt Systemverwaltungsaufgaben für Systeme, die internationale Sessions ausführen.

Bedeutung verschiedener Schriftarten und Symbole

Die folgende Tabelle beschreibt die verschiedenen Schriftarten und Symbole, die in diesem Buch verwendet werden.

Tabelle P-1 Typografische Konventionen

Schriftart oder Symbol	Bedeutung	Beispiel
AaBbCc123	Die Namen von Befehlen, Dateien, Verzeichnissen und Schlüsselwörtern.	Die Datei <code>.dtprofile</code> bearbeiten. <code>ls -a</code> verwenden, um alle Dateien aufzulisten.
<i>AaBbCc123</i>	Parameter oder Variable, durch realen Wert ersetzen.	Um eine Datei zu löschen, <code>rm</code> <i>Dateiname</i> eingeben.
AaBbCc123	Buchtitel, neue Wörter oder Begriffe oder Wörter hervorheben.	Kapitel 6 im <i>Benutzerhandbuch</i> lesen. Diese Optionen werden <i>Klasse</i> genannt. Der Benutzer <i>muß</i> root sein, um dies zu tun.

Anmeldemanager konfigurieren



Beim Anmeldemanager handelt es sich um einen Server , über den die Anzeige eines Anmeldefensters, die Identifikation von Benutzern, sowie der Start einer Benutzersitzung gesteuert wird. Die Anmeldung am grafischen Bildschirm bietet eine attraktive Alternative zur herkömmlichen Anmeldung im Zeichenmodus an Bitmap-Bildschirmen. Bildschirme, die über den Anmelde-Server verwaltet werden, können direkt an den Anmelde-Server oder an eine Xstation oder Datenstation im Netz angeschlossen werden.

Hinweis – Der Anmelde-Server kann nur durch den Benutzer ' root ' gestartet, gestoppt oder angepaßt werden.

<i>Anmelde-Server starten</i>	<i>2</i>
<i>Lokale Bildschirme und Bildschirme im Netz verwalten</i>	<i>3</i>
<i>Fehlerprüfung</i>	<i>11</i>
<i>Anmelde-Server stoppen</i>	<i>11</i>
<i>Erscheinungsbild des Anmeldefensters ändern</i>	<i>13</i>
<i>Anmeldemanager verwalten</i>	<i>22</i>
<i>Dateien des Anmeldemanagers</i>	<i>23</i>

Der Anmelde-Server:

- kann ein Anmeldefenster auf Bitmap-Bildschirmen sofort oder auf Aufforderung auf lokalen Bitmap-Bildschirmen oder Bitmap-Bildschirmen im Netz anzeigen
- ermöglicht den direkten Anschluß von Zeichenkonsolenanzeigen
- kann ein Auswahlfenster anzeigen, über das Benutzer Anmeldefenster anderer Anmelde-Server im Netz anzeigen können
- gestattet den gesteuerten Zugriff auf den Anmelde-Server
- ermöglicht den Zugriff auf die herkömmliche Anmeldung im Zeichenmodus

Bildschirme, die über den Anmeldemanager verwaltet werden, können direkt an den Anmeldemanager-Server oder an eine Xstation oder Datenstation im Netz angeschlossen sein. Bei lokalen Bildschirmen startet der Anmelde-Server automatisch einen X-Server und zeigt ein Anmeldefenster an. Bei Bildschirmen im Netz, wie z. B. Xstations, unterstützt der Anmelde-Server X Display Manager Protocol (XDMCP) 1.0, welches Bildschirmen ermöglicht, die Anzeige eines Anmeldefensters beim Anmelde-Server anzufordern.

Anmelde-Server starten

Normalerweise wird der Anmelde-Server beim Umladen des Systems gestartet. Der Anmelde-Server kann aber auch über die Befehlszeile gestartet werden.

- Soll der Anmelde-Server beim Umladen des Systems gestartet werden, folgendes eingeben:

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

Damit wird der Anmelde-Server beim Umladen des Systems automatisch gestartet.

- Um den Anmelde-Server über die Befehlszeile zu starten, folgendes eingeben:

```
/usr/dt/bin/dtlogin -daemon
```

Hinweis – Obwohl der Anmelde-Server auch über die Befehlszeile gestartet werden kann, damit temporäre Konfigurationen getestet werden können, sollte der Anmelde-Server normalerweise beim Umladen des Systems gestartet werden.

Lokale Bildschirme und Bildschirme im Netz verwalten

Abbildung 1-1 zeigt eine mögliche Anmelde-Serverkonfiguration.

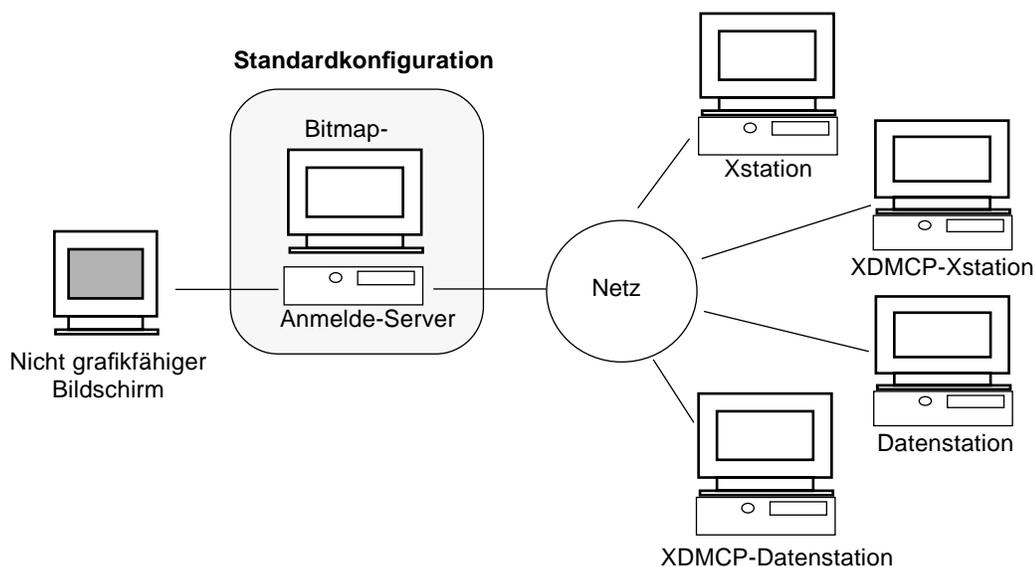


Abbildung 1-1 Mögliche Anmelde-Serverkonfiguration

Prozeß-ID des Anmelde-Servers feststellen

Standardmäßig wird die Prozeß-ID des Anmelde-Servers in der Datei `/var/dt/Xpidt` gespeichert.

Um eine Änderung vorzunehmen, kann die Ressource `Dtlogin.pidFile` in der Datei `Xconfig` gesetzt werden. Wird eine Änderung vorgenommen, muß das angegebene Verzeichnis beim Start des Anmelde-Servers bereits vorhanden sein.

Um eine Änderung in der Datei `Xconfig` vorzunehmen, wird `Xconfig` aus `/usr/dt/config` in `/etc/dt/config` kopiert. Nach der Änderung von `/etc/dt/config/Xconfig` wird der Anmelde-Server angewiesen, die Datei `Xconfig` erneut zu lesen, indem folgendes eingegeben wird:

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

Damit wird der Befehl `kill -HUP` *Prozeß-ID des Anmelde-Servers* abgesetzt.

Um, beispielsweise, die Prozeß-ID des Anmelde-Servers in `/var/mysservers/Dtpid` zu speichern, muß in der Datei `Xconfig` folgende Angabe vorhanden sein:

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

Wird der Anmelde-Server erneut gestartet, wird die Prozeß-ID des Anmelde-Servers in `/var/mysservers/Dtpid` gespeichert. Das Verzeichnis `/var/mysservers` muß beim Start des Anmelde-Servers bereits vorhanden sein.

Anmeldefenster an einem lokalen Bildschirm anzeigen

Beim Systemstart überprüft der Anmelde-Server die Datei `Xservers`, um festzustellen, ob ein X-Server gestartet werden muß, und ob und wie Anmeldefenster am lokalen Bildschirm oder an Bildschirmen im Netz angezeigt werden sollen.

Um eine Änderung in der Datei `Xservers` vorzunehmen, wird `Xservers` aus `/usr/dt/config` in `/etc/dt/config` kopiert. Nach der Änderung von `/etc/dt/config/Xservers` wird der Anmelde-Server angewiesen, die Datei `Xservers` erneut zu lesen, indem folgendes eingegeben wird:

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

Damit wird der Befehl `kill -HUP` *Prozeß-ID des Anmelde-Servers* abgesetzt.

Eine Zeile in der Datei `Xservers` hat folgendes Format:

```
anzeige_name anzeige_klasse anzeige_typ X_server_befehl
```

wobei die einzelnen Elemente folgende Bedeutung haben:

<i>anzeige_name</i>	Teilt dem Anmelde-Server den für die Verbindung zum X-Server zu verwendenden Verbindungsnamen mit (im folgenden Beispiel: :0). Ein Wert * (Stern) wird erweitert zu <i>host name:0</i> . Die angegebene Nummer muß der in <i>X_server_befehl</i> Verbindungsnummer angegebenen Nummer entsprechen.
<i>anzeige_klasse</i>	Gibt für diesen Bildschirm spezifische Ressourcen an (im folgenden Beispiel: Local).
<i>anzeige_typ</i>	Teilt dem Anmelde-Server mit, ob es sich bei dem Bildschirm um einen lokalen Bildschirm oder einen Bildschirm im Netz handelt, und wie die Option für die Anmeldung über Befehlszeile im Anmeldefenster verwaltet wird (im folgenden Beispiel: local@console).
<i>X_server_befehl</i>	Gibt die Befehlszeile, Verbindungsnummer und weitere Optionen an, die der Anmelde-Server benötigt, um den X-Server zu starten (im folgenden Beispiel: /usr/bin/X11/X :0). Die angegebene Verbindungsnummer muß der in <i>anzeige_name</i> angegebenen Nummer entsprechen.

Eine Zeile in der Datei `Xservers` sieht standardmäßig ungefähr folgendermaßen aus:

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

Anmelde-Server ohne lokalen Bildschirm ausführen

Verfügt das Anmelde-Server-System über keinen Bitmap-Bildschirm, wird der Anmelde-Server ohne lokalen Bildschirm ausgeführt, indem die Zeile für den lokalen Bildschirm in der Datei `Xservers` mit Hilfe eines # (Nummernzeichen) auf Kommentar gesetzt wird. Beispiel:

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

Nach dem Start läuft der Anmelde-Server im Hintergrund und wartet auf Anforderungen von Bildschirmen im Netz.

Auf Anmeldung über Befehlszeile an einem lokalen Bildschirm zugreifen

Wählt der Benutzer im Anmeldefenster 'Anmeldung über Befehlszeile' aus, beendet der Anmelde-Server den X-Server temporär, und gestattet den Zugriff auf die herkömmliche Anmeldung über Befehlszeile, die auf einer Datenstationseinheit mit Bitmap-Bildschirm ausgeführt wird. Nachdem der Benutzer sich an- und wieder abgemeldet hat, bzw. nach der Überschreitung eines angegebenen Zeitlimits, startet der Anmelde-Server den X-Server erneut.

Hinweis – Auf die Option 'Anmeldung über Befehlszeile' kann bei Bildschirmen im Netz nicht zugegriffen werden.

Die Angabe *anzeige_typ* steuert das Verhalten der Anmeldung über Befehlszeile. *anzeige_typ* hat folgendes Format:

- `local@anzeige_terminal_gerät`
- `local`
- `foreign`

Wird `local@anzeige_terminal_gerät` angegeben, nimmt der Anmelde-Server an, daß der X-Server und `/dev/anzeige_terminal_gerät` sich auf derselben physischen Einheit befinden, und daß eine Anmeldung über Befehlszeile (normalerweise `getty`) auf der Einheit ausgeführt wird. Wählt der Benutzer 'Anmeldung über Befehlszeile' aus, wird der X-Server beendet und es kann auf die Anmeldung über Befehlszeile (`getty`) zugegriffen werden, die auf `/dev/anzeige_terminal_gerät` ausgeführt wird.

Um die Option 'Anmeldung über Befehlszeile' an einem Bildschirm zu inaktivieren, muß für *anzeige_terminal_gerät* `none` angegeben werden. Standardmäßig wird bei *anzeige_terminal_gerät* `console` angegeben. Wird `local` angegeben, wird für *anzeige_terminal_gerät* automatisch der Standardwert `console` festgelegt. Wird `foreign` angegeben, wird die Option 'Anmeldung über Befehlszeile' inaktiviert.

Hinweis – Wird der Anmelde-Server über die Befehlszeile gestartet, wird die Option 'Anmeldung über Befehlszeile' am lokalen Bildschirm inaktiviert.

Konsole mit zeichenorientierter Anzeige verwenden

Verfügt das Anmelde-Server-System über einen direkt angeschlossenen nicht grafikfähigen Bildschirm, der als Konsole dient, kann `anzeige_terminal_gerät` auf `none` gesetzt werden, um die Anmeldung über Befehlszeile im Anmeldefenster des Bitmap-Bildschirms zu inaktivieren.

Alternativ kann, wenn die Anmeldung über Befehlszeile (`getty`) sowohl auf der Konsole mit zeichenorientierter Anzeige als auch auf dem Bitmap-Bildschirm ausgeführt wird, `anzeige_terminal_gerät` auf die Einheit mit Anmeldung über Befehlszeile (`getty`) für Bitmap-Bildschirm gesetzt werden.

Befindet sich, beispielsweise, die 'Anmeldung über Befehlszeile' (`getty`) für Bitmap-Bildschirm auf der Einheit `/dev/tty01`, kann `anzeige_typ` auf `local@tty01` festgelegt werden.

Anmeldefenster an einem Bildschirm im Netz anzeigen

Der Anmelde-Server kann Anforderungen von Bildschirmen im Netz entgegennehmen, ein Anmeldefenster an diesem bestimmten Bildschirm anzuzeigen. Bei dem Bildschirm im Netz handelt es sich normalerweise um eine Xstation, es kann aber auch eine Datenstation sein.

Um Anforderungen von Bildschirmen im Netz zu verwalten, unterstützt der Anmelde-Server das X Display Manager Protocol (XDMCP) 1.0. Dieses Protokoll ermöglicht dem Anmelde-Server Anforderungen von Bildschirmen im Netz auszuhandeln, zu akzeptieren oder zurückzuweisen. Bei den meisten Xstations ist XDMCP integriert.

XDMCP-Direktanforderung von Bildschirmen im Netz

Wird eine Xstation so konfiguriert, daß der XDMCP-Direktmodus (Abfragemodus) verwendet wird, wird der Xstation der Host-Name des Anmelde-Server-Hosts mitgeteilt. Beim URLaden der Xstation geht automatisch eine Anforderung an den Anmelde-Server und dieser zeigt ein Anmeldefenster an der Xstation an. Informationen zur Konfiguration einer Xstation für XDMCP-Direktmodus können der Dokumentation zur Xstation entnommen werden.

Die meisten X-Server unterstützen auch die Option `-query`. In diesem Modus verhält sich der X-Server als ob er eine Xstation sei und sendet eine Aufforderung zur Anzeige eines Anmeldefensters auf dem X-Server direkt an

den Anmelde-Server. Wird, beispielsweise, der X-Server auf einem Bitmap-Bildschirm auf der Datenstation `bridget` gestartet, wird der Anmelde-Server `anita` ein Anmeldefenster auf dem X-Server anzeigen:

```
X -query anita
```

Indirekte XDMCP-Anforderungen von Bildschirmen im Netz

Wird eine Xstation so konfiguriert, daß der indirekte XDMCP-Modus verwendet wird, wird der Xstation der Host-Name des Anmelde-Server-Hosts mitgeteilt. Beim Umladen der Xstation geht eine Anforderung an den Anmelde-Server, der daraufhin ein Auswahlfenster mit weiteren Anmelde-Server-Hosts im Netz anzeigt. Aus dieser Liste kann der Benutzer einen Host auswählen, der dann ein Anmeldefenster auf der Xstation des Benutzers anzeigt. Informationen zur Konfiguration einer Xstation für den indirekten XDMCP-Modus können der Dokumentation zur Xstation entnommen werden.

Wie beim Direktmodus unterstützten die meisten X-Server die Option `-indirect`, durch die ein X-Server im indirekten XDMCP-Modus Verbindung zum Anmelde-Server aufnimmt.

Bildschirme im Netz ohne XDMCP verwalten

Ältere Xstations unterstützen möglicherweise XDMCP nicht. Damit der Anmelde-Server ein Anmeldefenster auf diesen Xstations anzeigen kann, muß der Name der Xstation in der Datei `Xservers` enthalten sein.

Beispiel

Folgende Zeilen in der Datei `Xservers` geben dem Anmelde-Server an, daß Anmeldefenster an zwei Xstations ohne XDMCP, `ruby` und `wolfie` angezeigt werden sollen:

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

Da sich der Bildschirm im Netz befindet, enthält `anzeige_name` den Host-Namen als Teil des Namens. Bei `display_klasse` können Ressourcen angegeben werden, die für eine bestimmte Klasse von Xstations spezifisch sind. (Zu welcher Bildschirmklasse eine Xstation gehört, kann der zugehörigen Dokumentation entnommen werden.) Wird bei `anzeige_typ` `foreign`

angegeben, wird der Anmelde-Server angewiesen, eine Verbindung zu einem bereits vorhandenen X-Server herzustellen, statt einen eigenen X-Server zu starten. In diesem Fall erfolgt für `X_server_befehl` keine Angabe.

Zugriff auf Anmelde-Server steuern

Standardmäßig kann jeder Host im Netz, der Zugriff auf den Anmelde-Server-Host hat, die Anzeige eines Anmeldefenster anfordern. Der Zugriff auf den Anmelde-Server kann durch Änderung der Datei `Xaccess` eingeschränkt werden.

Um eine Änderung in der Datei `Xaccess` vorzunehmen, wird `Xaccess` aus `/usr/dt/config` in `/etc/dt/config` kopiert. Nach der Änderung von `/etc/dt/config/Xaccess` wird der Anmelde-Server angewiesen, die Datei `Xaccess` erneut zu lesen, indem folgendes eingegeben wird:

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

Damit wird der Befehl `kill -HUP Prozeß-ID des Anmelde-Servers` abgesetzt.

XDMCP-Direktmodus

Versucht ein Host über den XDMCP-Direktmodus eine Verbindung zum Anmelde-Server herzustellen, wird der Host-Name mit den Einträgen in der Datei `Xaccess` verglichen, um festzustellen, ob dem Host der Zugriff auf den Anmelde-Server gestattet ist. Bei den Einträgen in der Datei `Xaccess` handelt es sich um Namen von Hosts, denen der Zugriff auf den Anmelde-Server gestattet ist. Diese Host-Namen können auch die Platzhalterzeichen `*` (Stern) und `?` (Fragezeichen) enthalten. Ein `*` (Stern) entspricht keinem oder mehr Zeichen, ein `?` (Fragezeichen) entspricht einem beliebigen Zeichen. Ein `!` (Ausrufezeichen) vor einem Eintrag gibt an, daß dem entsprechenden Host der Zugriff nicht gestattet ist.

Beispiel einer Datei `Xaccess` mit drei Einträgen:

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

Der erste Eintrag gestattet dem Host `amazon.waterloo.com` den Zugriff auf den Anmelde-Server, der zweite Eintrag gestattet jedem Host den Zugriff, dessen vollständiger Domänenname auf `dept5.waterloo.com` endet. Der letzte Eintrag gibt an, daß jedem anderen Host der Zugriff nicht gestattet ist.

Indirekter XDMCP-Modus

Versucht ein Host über den indirekten XDMCP-Modus eine Verbindung zum Anmelde-Server herzustellen, wird der Host-Name mit den Einträgen in der Datei `xaccess` verglichen, um festzustellen, ob dem Host der Zugriff auf den Anmelde-Server gestattet ist. Die Einträge in der Datei `xaccess` entsprechen den Einträgen im XDMCP-Direktmodus einschließlich der Platzhalterzeichen, werden aber mit einer Zeichenfolge `CHOOSE` markiert. Beispiel:

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

Auch hier gestattet der erste Eintrag dem Host `amazon.waterloo.com` den Zugriff auf den Anmelde-Server, der zweite Eintrag gestattet jedem Host den Zugriff, dessen vollständiger Domänenname auf `dept5.waterloo.com` endet. Der letzte Eintrag gibt an, daß jedem anderen Host der Zugriff nicht gestattet ist.

Nach der Zeichenfolge `CHOOSE` kann eine der folgenden Angaben stehen:

- `BROADCAST`
- *liste der hostnamen*

`BROADCAST` bedeutet, daß der Anmelde-Server eine Anforderung zur Generierung einer Liste verfügbarer Anmelde-Server-Hosts an das Teilnetz des Anmelde-Server sendet. Wird eine Liste mit Host-Namen angegeben, wird diese Liste als Liste verfügbarer Anmelde-Hosts verwendet. Beispiel:

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

Stellt `amazon.waterloo.com` eine Verbindung über den indirekten XDMCP-Modus her, wird eine Liste angezeigt, die aus `shoal` und `alum` besteht. Stellt `alice.dept5.waterloo.com` eine Verbindung her, wird eine Liste aller verfügbarer Anmelde-Server-Hosts im Teilnetz des Anmelde-Servers angezeigt. Anforderungen anderer Anmelde-Server im indirekten XDMCP-Modus werden nicht akzeptiert.

Eine weitere Möglichkeit, eine Liste mit Host-Namen anzugeben, besteht darin, ein oder mehrere Makros zu definieren, die die Liste der Host-Namen enthalten. Beispiel:

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

Fehlerprüfung

Standardmäßig protokolliert der Anmelde-Server Fehler in der Datei `/var/dt/Xerrors`. Um eine Änderung vorzunehmen, kann die Ressource `Dtlogin.errorLogFile` in der Datei `Xconfig` festgelegt werden. Das angegebene Verzeichnis muß beim Start des Anmelde-Servers bereits vorhanden sein.

Damit, beispielsweise, der Anmelde-Server Fehler in der Datei `/var/mylogs/Derrors` protokolliert, muß die Datei `Xconfig` folgende Angabe enthalten:

```
Dtlogin.errorLogFile: /var/mylogs/Derrors
```

Wird der Anmelde-Server erneut gestartet, protokolliert der Anmelde-Server Fehler in der Datei `/var/mylogs/Derrors`. Das Verzeichnis `/var/mylogs` muß beim Start des Anmelde-Servers bereits vorhanden sein.

Anmelde-Server stoppen

- Soll der automatische Start des Anmelde-Servers beim Umladen des Systems inaktiviert werden, muß folgendes eingegeben werden:

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

Damit wird der Anmelde-Server beim nächsten Umladen des Systems nicht gestartet.

- Soll der Anmelde-Server durch Löschen der Prozeß-ID gestoppt werden, muß folgendes eingegeben werden:

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

Damit wird der Befehl `kill` *Prozeß-ID des Anmelde-Servers* abgesetzt.

Hinweis – Wird die Prozeß-ID des Anmelde-Servers gelöscht, werden alle vom Anmelde-Server verwalteten Benutzer-Sessions beendet.

Der Anmelde-Server kann auch durch Löschen der Prozeß-ID gestoppt werden. Die Prozeß-ID des Anmelde-Servers ist in `/var/dt/Xpid` oder in der in `Xconfig` durch die Ressource `Dtlogin.pidFile` angegebenen Datei gespeichert.

Ist der Benutzer zum Zeitpunkt, an dem er den Anmelde-Server löscht, auf dem Desktop angemeldet, wird die Desktop-Session sofort beendet.

Anmeldefenster

Das vom Anmelde-Server angezeigte Anmeldefenster stellt eine attraktive Alternative zu herkömmlichen Anmeldefenstern im Zeichenmodus dar und bietet dem Benutzer ein Leistungsspektrum, das über die Möglichkeiten einer Anmeldung im Zeichenmodus hinausgeht.

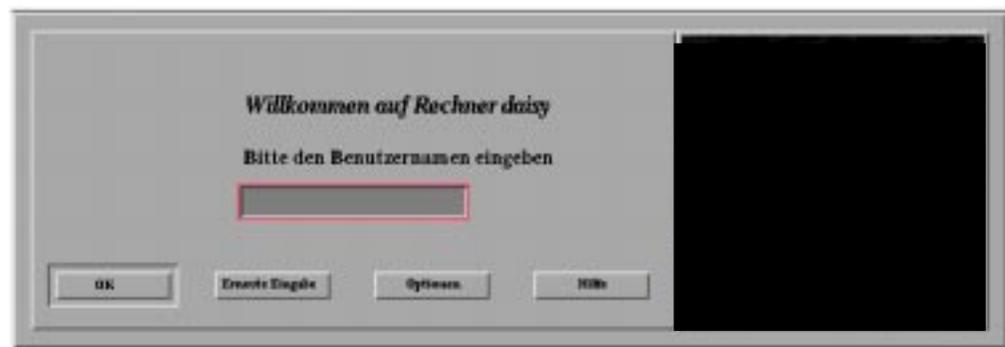


Abbildung 1-2 Desktop-Anmeldefenster

Wie bei der Anmeldung im Zeichenmodus gibt der Benutzer einen Benutzernamen und anschließend ein Kennwort ein. Wird der Benutzer identifiziert, startet der Anmelde-Server eine Desktop-Session für den Benutzer. Verläßt der Benutzer die Desktop-Session, zeigt der Anmelde-Server ein neues Anmeldefenster an und der Prozeß beginnt von neuem.

Das Anmeldefenster kann angepaßt werden, indem

- das Erscheinungsbild des Anmeldefensters geändert wird
- X-Server-Berechtigung konfiguriert werden
- die Standardsprache geändert wird
- vor der Anzeige des Anmeldefensters Befehle eingegeben werden
- der Inhalt des Sprachenmenüs des Anmeldefensters geändert wird
- der Befehl zum Start der Benutzersession angegeben wird
- vor dem Start der Desktop-Session des Benutzers Befehle eingegeben werden
- nach Beendigung der Benutzer-Session Befehle eingegeben werden

Dies kann jeweils für alle Bildschirme oder aber für jeden einzelnen Bildschirm erfolgen.

Erscheinungsbild des Anmeldefensters ändern

Das Erscheinungsbild des Anmeldefensters kann angepaßt werden, indem das Logo oder die Grafik, die Begrüßungsnachrichten und die Schriftarten geändert werden.

Um eine Änderung in der Datei `Xresources` vorzunehmen, wird `Xresources` aus `/usr/dt/config/sprache` in `/etc/dt/config/sprache` kopiert. Beim nächsten Aufruf des Anmeldefensters werden alle Änderungen wirksam. Um die Neuanzeige eines Anmeldefensters zu erzwingen, im Menü 'Optionen' des Anmeldefensters 'Anmeldefenster zurücksetzen' auswählen.

Zu den Attributen des Anmeldefensters, die durch Ressourcenspezifikationen in der Datei `Xresources` festgelegt werden können, gehören:

<code>Dtlogin*logo*bitmapFile</code>	Bitmap- oder Pixmap-Datei für die Anzeige als Logoabbild
<code>Dtlogin*greeting*labelString</code>	Begrüßungsnachricht
<code>Dtlogin*greeting*persLabelString</code>	Angepaßte Begrüßungsnachricht
<code>Dtlogin*greeting*fontList</code>	Schriftart für Begrüßungsnachrichten
<code>Dtlogin*labelFont</code>	Schriftart für Tasten und Bezeichnungen
<code>Dtlogin*textFont</code>	Schriftart für Hilfe und Fehlermeldungen
<code>Dtlogin*<i>sprache</i>*languageName</code>	Alternativtext für länderspezifische Angaben für <i>sprache</i>

▼ Logo ändern

- ♦ Die Ressource `Dtlogin*logo*bitmapFile` in der Datei `Xresources` festlegen.

Das Logo kann als farbige Pixmap- oder als Bitmap-Datei vorliegen.

Im folgenden Beispiel wird die Bitmap-Datei `Mylogo` als Logo verwendet:

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

▼ Begrüßungsnachricht ändern

Standardmäßig zeigt der Anmelde-Server die Nachricht "Willkommen auf Rechner *hostname*" im Anmeldefenster an. Diese Nachricht kann folgendermaßen geändert werden:

- ◆ Die Ressource `Dtlogin*greeting*labelString` in `Xresources` setzen.

Der Wert der Ressource `labelString` kann die Variable `%Lokaler Host%`, die durch den Host-Namen des Anmelde-Servers ersetzt wird, und die Variable `%Anzeige Name%`, die durch den Namen des X-Servers ersetzt wird, an dessen Bildschirm die Nachricht angezeigt wird, enthalten.

Im folgenden Beispiel wird die Begrüßungsnachricht in `Hier ist hostname!` geändert:

```
Dtlogin*greeting*labelString: Hier ist %LocalHost%
```

Nach Eingabe des Benutzernamens zeigt der Anmelde-Server standardmäßig die Nachricht `Willkommen: benutzername` an. Diese Nachricht kann geändert werden, indem die Ressource `Dtlogin*greeting*persLabelString` in `Xresources` festgelegt wird. Der Wert von `persLabelString` kann die Variable `%s` enthalten, die durch `benutzername` ersetzt wird.

Im folgenden Beispiel wird die angepaßte Begrüßungsnachricht in `Hallo benutzername` geändert.

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

▼ Schriftarten ändern

Die im Anmeldefenster verwendeten Schriftarten können geändert werden, indem eine der folgenden Schriftartressourcen in `Xresources` geändert werden:

Um die verfügbaren Schriftarten aufzulisten, folgendes eingeben:

```
xlsfonts [-optionen] [-fn muster]
```

```
Dtlogin*greeting*fontList Schriftart für Begrüßungsnachrichts
```

```
Dtlogin*labelFont Schriftart für Tasten und Bezeichnungen
```

Dtlogin*textFont Schriftart für Hilfe und Fehlermeldungen

Im folgenden Beispiel wird eine große Schriftart für die Begrüßungsnachricht angegeben (die Angabe darf eine Zeile nicht überschreiten):

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

▼ *Alternativtext für jede Sprache anzeigen lassen*

Damit im Menü 'Sprache' des Anmeldefenster statt der standardmäßigen länderspezifischen Angabe ein Alternativtext angezeigt wird, muß die Ressource

Dtlogin*sprache *languageName in der Datei Xresources geändert werden:

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

Statt der länderspezifischen Angabe En_US wird nun der Text American angezeigt.

Verhalten des Anmeldefensters ändern

Um das Verhalten des Anmeldefensters zu ändern, können Ressourcen in der Datei Xconfig geändert werden.

Um eine Änderung in der Datei Xconfig vorzunehmen, wird Xconfig aus /usr/dt/config in /etc/dt/config kopiert. Nach der Änderung von /etc/dt/config/Xconfig wird der Anmelde-Server angewiesen, die Datei Xconfig erneut zu lesen, indem folgendes eingegeben wird:

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

Damit wird der Befehl kill -HUP (*Prozeß-ID des Anmelde-Servers*) abgesetzt.

Zu den Ressourcen, die in der Datei Xconfig spezifiziert sind, gehören:

Dtlogin*authorize	Spezifikation der Datei Xaccess
Dtlogin*environment	X-Server-Umgebung
Dtlogin*language	Standardsprache
Dtlogin*languageList	Liste der Sprachen für das Menü 'Sprachen' im Anmeldefenster
Dtlogin*resources	Spezifikation der Datei Xresources
Dtlogin*setup	Spezifikation der Datei Xsetup

<code>Dtlogin*startup</code>	Spezifikation der Datei <code>Xstartup</code>
<code>Dtlogin*session</code>	Spezifikation der Datei <code>Xsession</code>
<code>Dtlogin*failsafeClient</code>	Spezifikation der Prozedur <code>Xfailsafe</code>
<code>Dtlogin*reset</code>	Spezifikation der Prozedur <code>Xreset</code>
<code>Dtlogin*userPath</code>	PATH für <code>Xsession</code> und <code>Xfailsafe</code>
<code>Dtlogin*systemPath</code>	PATH für <code>Xsetup</code> , <code>Xstartup</code> und <code>Xfailsafe</code>
<code>Dtlogin*systemShell</code>	SHELL für <code>Xsetup</code> , <code>Xstartup</code> und <code>Xfailsafe</code>
<code>Dtlogin.timeZone</code>	TZ für alle Prozeduren

Verhalten des Anmeldefensters an einzelnen Bildschirmen ändern

In den obigen Beispielen wurde durch die Änderung einer Ressource in der Datei `Xconfig` das Verhalten des Anmeldefensters an allen Bildschirmen geändert. Ressourcen, die mit einem * (Stern) gekennzeichnet sind, können auch für einzelne Bildschirme geändert werden. Damit kann der Benutzer das Verhalten einzelner Bildschirme seinen Wünschen anpassen. Um eine Ressource für einen bestimmten Bildschirm zu setzen, wird die Ressource folgendermaßen angegeben: `Dtlogin*anzeigeName*ressource`. Soll beispielsweise die benutzerabhängige Zugriffssteuerung für Bildschirm `expo:0` abgeschaltet werden, aber an anderen Bildschirmen erhalten bleiben, wird folgendes eingegeben:

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

Hinweis – Sämtliche Sonderzeichen im Namen des Bildschirms, wie z. B. : (Doppelpunkt) oder . (Punkt), werden durch _ (Unterstreichungszeichen) ersetzt.

Zugriffsrechte auf den X-Server ändern

Standardmäßig gestattet der Anmelde-Server die Zugriffssteuerung auf den X-Server jeweils einzelnen Benutzern auf der Grundlage von Berechtigungsdaten, die in der Datei *Home-Verzeichnis*/`.Xauthority` gespeichert und geschützt

sind. Nur Benutzern, die diese Datei lesen können, ist es gestattet, eine Verbindung zum X-Server herzustellen. Dies ist die bevorzugte Methode der Zugriffssteuerung auf den X-Server.

Eine Alternative zur benutzerabhängigen Zugriffssteuerung bietet die Host-gestützte Zugriffssteuerung. Bei dieser Methode erhält, wenn ein Host Zugriff auf einen X-Server hat, jeder Benutzer an diesem Host Zugriff auf den X-Server. Mögliche Gründe für die Verwendung der Host-gestützten Zugriffssteuerung sind:

- Ältere R2 und R3 X-Clients können nicht über die benutzerabhängige Zugriffssteuerung eine Verbindung zu einem X-Server herstellen.
- In nicht gesicherten Netzen könnte ein nicht berechtigter Benutzer die Berechtigungsdaten, die zwischen dem X-Client und dem X-Server im Netz weitergereicht werden, abfangen.

Die Ressource `Xconfig Dtlogin*authorize` weist den Anmelde-Server an, die benutzerabhängige Zugriffssteuerung zu verwenden. Soll die host-gestützte Zugriffssteuerung verwendet werden, muß der Wert der Ressource `authorize` auf `False` festgelegt werden. Beispiel:

```
Dtlogin*authorize: False
```

▼ *X-Server-Umgebung ändern*

Soll der X-Server mit einer oder mehreren Umgebungsvariablen und -werten ausgestattet werden, wenn er über den Anmelde-Server gestartet wird, können diese mit der Ressource `Dtlogin*environment` in `Xconfig` angegeben werden. Beispiel:

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

Hier werden die Variablen `VAR1` und `VAR2` dem lokalen X-Server-Prozeß zur Verfügung gestellt. Diese Variablen werden auch in die Prozeduren `Xsession` und `Xfailsafe` exportiert.

▼ *Standardsprache ändern*

Wenn sich der Benutzer über das Anmeldefenster am Desktop anmeldet, wird die Benutzer-Session unter der länderspezifischen Angabe ausgeführt, die der Benutzer im Untermenü 'Sprachen' des Menüs 'Optionen' ausgewählt hat. Wählt der Benutzer keine Sprache aus, verwendet der Anmelde-Server die

Standardsprache. Der Benutzer kann festlegen, welche Sprache als Standardsprache verwendet werden soll, indem er die Ressource `Dtlogin*language` in `Xconfig` setzt. Beispiel :

```
Dtlogin*language: Ja_JP
```

Welche Sprachen auf einem System installiert sind, kann der zugehörigen Dokumentation entnommen werden.

▼ *Inhalt des Menüs 'Sprachen' des Anmeldefensters ändern*

Standardmäßig erstellt der Anmelde-Server im Anmeldefenster das Menü 'Sprachen' mit einer Liste aller länderspezifischen Angaben, die auf dem System installiert sind. Wählt der Benutzer in dieser Liste eine länderspezifische Angabe aus, zeigt der Anmelde-Server das Anmeldefenster in der Sprache der ausgewählten länderspezifischen Angabe erneut an. Meldet sich der Benutzer anschließend im Anmeldefenster an, startet der Anmelde-Server eine Desktop-Session in der Sprache der ausgewählten länderspezifischen Angabe für den Benutzer.

Der Benutzer kann eine eigene Liste mit Sprachen erstellen, indem er die Ressource `Dtlogin*languageList` in `Xconfig` ändert:

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

Der Anmelde-Server zeigt nun nur `En_US` und `De_DE` im Menü 'Sprachen' des Anmeldefensters an.

Befehle vor der Anzeige des Anmeldefensters eingeben

Nachdem der X-Server gestartet wurde, aber bevor das Anmeldefenster angezeigt wird, führt der Anmelde-Server die Prozedur `Xsetup` aus. `Xsetup` wird mit Root-Berechtigung ausgeführt und setzt Befehle ab, die ausgeführt werden müssen, bevor das Anmeldefenster angezeigt wird.

Um eine Änderung in der Datei `Xsetup` vorzunehmen, wird `Xsetup` aus `/usr/dt/config` in `/etc/dt/config` kopiert. Beim nächsten Aufruf des Anmeldefensters wird die geänderte Datei `Xsetup` ausgeführt.

Befehle eingeben, bevor die Benutzer-Session gestartet wird

Nachdem der Benutzer den Benutzernamen und das Kennwort eingegeben hat und diese identifiziert worden sind, aber bevor die Benutzer-Session gestartet worden ist, führt der Anmelde-Server die Prozedur `Xstartup` aus. `Xstartup` wird mit Root-Berechtigung ausgeführt und setzt Befehle ab, die als Benutzer 'root' ausgeführt werden müssen, bevor die Benutzer-Session gestartet wird.

Um eine Änderung in der Datei `Xstartup` vorzunehmen, wird `Xstartup` aus `/usr/dt/config` in `/etc/dt/config` kopiert. Bei der nächsten Anmeldung des Benutzers wird die geänderte Datei `Xstartup` ausgeführt.

Desktop-Session starten

Standardmäßig startet der Anmelde-Server die Benutzer-Session, indem er die Prozedur `Xsession` ausführt. `Xsession` wird mit der Berechtigung des Benutzers ausgeführt und setzt Befehle ab, die für den Start des Desktops benötigt werden.

Hinweis – Die Prozedur `Xsession` darf nicht direkt aktualisiert werden.

Informationen über die Anpassung des Starts einer Desktop-Session für den Benutzer können Kapitel 2, "Sessionmanager konfigurieren" entnommen werden.

Einfache Motif-Session starten

Wählt der Benutzer im Untermenü 'Sessions' des Menüs 'Optionen' des Anmeldefensters 'einfache Motif-Session' aus, führt der Anmelde-Server die Prozedur `Xfailsafe` aus. `Xfailsafe` wird mit der Berechtigung des Benutzers ausgeführt und setzt Befehle ab, die eine Minimalumgebung mit Fenstertechnik, normalerweise ein Datenstationsfenster und eine wahlfreie Fenstersteuerung, startet.

Um eine Änderung in der Datei `Xfailsafe` vorzunehmen, wird `Xfailsafe` aus `/usr/dt/config` in `/etc/dt/config` kopiert. Bei der nächsten Anmeldung des Benutzers wird die geänderte Datei `Xfailsafe` ausgeführt.

Nach dem Ende der Benutzer-Session

Nachdem der Benutzer die Desktop- oder einfache Motif-Session verlassen hat, führt der Anmelde-Server die Prozedur `Xreset` aus. `Xreset` wird mit Root-Berechtigung ausgeführt und setzt Befehle ab, die als Benutzer 'root' ausgeführt werden müssen, nachdem die Benutzer-Session beendet wurde.

Um eine Änderung in der Datei `Xreset` vorzunehmen, wird `Xreset` aus `/usr/dt/config` in `/etc/dt/config` kopiert. Bei der nächsten Anmeldung des Benutzers wird die geänderte Datei `Xreset` ausgeführt.

Umgebung des Anmelde-Servers

Der Anmelde-Server stellt eine Umgebung zur Verfügung, die in die Prozeduren `Xsetup`, `Xstartup`, `Xsession`, `Xfailsafe` und `Xreset` exportiert wird. Diese Umgebung wird in Tabelle 1-1 beschrieben. Weitere Variablen können ebenfalls durch den Anmelde-Server exportiert werden.

Tabelle 1-1 Umgebungen des Anmelde-Servers

Umgebungsvariable	Xsetup	Xstartup	Xsession	Xreset	Beschreibung
LANG	X	X	X	X	Standard- oder ausgewählte Sprache
XAUTHORITY	X	X	X	X	X-Berechtigungsdatei ändern (Option)
PATH	X	X	X	X	Wert der Ressource <code>Dtlogin*userPath</code> (<code>Xsession</code> , <code>Xfailsafe</code>) oder der Ressource <code>Dtlogin*systemPath</code> (<code>Xsetup</code> , <code>Xstartup</code> , <code>Xreset</code>)
DISPLAY	X	X	X	X	Verbindungsnummer des X-Servers
SHELL	X	X	X	X	In <code>/etc/passwd</code> (<code>Xsession</code> , <code>Xfailsafe</code>) angegebene Shell oder Ressource <code>Dtlogin*systemShell</code> (<code>Xsetup</code> , <code>Xstartup</code> , <code>Xreset</code>)
TZ	X	X	X	X	Wert der Ressource <code>Dtlogin.timeZone</code> oder vom System festgelegte Zeitzone

Tabelle 1-1 Umgebungen des Anmelde-Servers (Fortsetzung)

Umgebungsvariable	Xsetup	Xstartup	Xsession	Xreset	Beschreibung
USER	X	X	X		Benutzername
HOME		X	X	X	In <code>/etc/passwd</code> angegebenes Benutzerverzeichnis
LOGNAME	X	X	X		Benutzername

Pfad für Benutzer oder System ändern

Der Anmelde-Server legt die Umgebungsvariable PATH fest, wenn die Prozeduren Xsession und Xfailsafe ausgeführt werden. Der Benutzer kann einen Alternativpfad zu diesen Prozeduren festlegen:

▼ Benutzerpfad ändern

♦ Die Ressource `Dtlogin*userPath` in `Xconfig` festlegen. Beispiel: :

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

▼ Systempfad ändern

♦ Die Ressource `Dtlogin*systemPath` in `Xconfig` festlegen. Beispiel:

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

▼ System-Shell ändern

Der Anmelde-Server legt die Umgebungsvariable SHELL fest, wenn die Prozeduren Xsetup, Xstartup und Xfailsafe ausgeführt werden. Standardmäßig wird `/bin/sh` festgelegt. Soll alternativ zu diesen Prozeduren eine andere Shell zur Verfügung gestellt werden, die Ressource `Dtlogin*systemShell` in `Xconfig` festlegen. Beispiel:

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

▼ Zeitzone ändern

Der Anmelde-Server legt die Umgebungsvariable TZ fest , wenn die Prozeduren Xsetup, Xstartup, Xsession, Xfailsafe und Xreset ausgeführt werden. Der Standardwert wird dem System entnommen, so daß eine Änderung normalerweise nicht notwendig ist. Soll eine andere Zeitzone zur Verfügung gestellt werden, die Ressource Dtlogin.timeZone in Xconfig festlegen. Beispiel :

```
Dtlogin.timeZone: CST6CDT
```

Anmeldemanager verwalten

Wird der Anmelde-Server gestartet, wird ein Prozeß dtlogin gestartet. Der Prozeß dtlogin liest die Datei Xconfig, um die Anfangskonfiguration des Anmelde-Servers und weitere Konfigurationsdateien des Anmelde-Servers festzustellen. Dann liest der Anmelde-Server die Datei Xservers , um festzustellen, ob Bildschirme vorhanden sind, die explizit verwaltet werden müssen und die Datei Xaccess , um den Zugriff auf den Anmelde-Server zu steuern.

Stellt der Anmelde-Server in der Datei Xservers fest, daß ein lokaler Bildschirm zu verwalten ist, startet er, gemäß den Instruktionen in der Datei Xservers , einen X-Server und zeigt an diesem Bildschirm ein Anmeldefenster an.

Stellt der Anmelde-Server in der Datei Xservers fest, daß ein Bildschirm im Netz zu verwalten ist, wird er davon ausgehen, daß ein X-Server mit dem angegebenen Bildschirmnamen bereits läuft und ein Anmeldefenster an diesem Bildschirm anzeigen.

Anschließend wird der Anmelde-Server auf XDMCP-Anforderungen aus dem Netz warten.

Für jeden Bildschirm, den er verwaltet, erstellt der Anmelde-Server zunächst einen neuen Prozeß dtlogin für diesen Bildschirm. Dies bedeutet, daß, wenn der Anmelde-Server n Bildschirme verwaltet, n+1 Prozesse dtlogin vorhanden sind. Der Anmelde-Server führt die Prozedur Xsetup aus, lädt die Datei Xresources , und führt dann dtgreet aus, um das Anmeldefenster anzuzeigen. Nachdem der Benutzer einen Benutzernamen und ein Kennwort eingegeben hat und identifiziert wurde, führt der Anmelde-Server die

Prozedur `Xstartup` aus, sowie anschließend die Prozedur `Xsession` bzw. `Xfailsafe`. Hat der Benutzer die Session beendet, führt der Anmelde-Server die Prozedur `Xreset` aus.

Erhält der Anmelde-Server eine indirekte XDMCP-Anforderung, wird `dtchooser` ausgeführt und eine Liste mit Anmelde-Server-Hosts auf dem Bildschirm angezeigt. Wählt der Benutzer einen Host aus dieser Liste, verwaltet der Anmelde-Server an diesem Host den Bildschirm.

Die Konfigurationsdateien `Xaccess`, `Xconfig`, `Xfailsafe`, `Xreset`, *sprache*/`Xresources`, `Xservers`, `Xsetup` und `Xstartup` sucht der Anmelde-Server standardmäßig zuerst in `/etc/dt/config` und dann in `/usr/dt/config`. Er verwendet die Datei, die er zuerst findet.

Dateien des Anmeldemanagers

Die Dateien des Anmeldemanagers befinden sich standardmäßig in:

<code>/usr/dt/bin/dtlogin</code>	Anmelde-Server und Anzeigemanager
<code>/usr/dt/bin/dtgreet</code>	Zeigt ein Anmeldefenster an einem Bildschirm an
<code>/usr/dt/bin/dtchooser</code>	Zeigt ein Auswahlfenster an einem Bildschirm an
<code>/usr/dt/bin/Xsession</code>	Startet eine Desktop-Session
<code>/usr/dt/config/Xfailsafe</code>	Startet eine einfache Motif-Session
<code>/usr/dt/config/Xconfig</code>	Konfigurationsdatei des Anmelde-Servers
<code>/usr/dt/config/Xservers</code>	Anzeigebeschreibungsdatei des Anmelde-Servers
<code>/usr/dt/config/Xaccess</code>	Zugriffsbeschreibungsdatei des Anmelde-Servers
<code>/usr/dt/config/<i>sprache</i> /Xresources</code>	Anzeigeentwurfsressourcen
<code>/usr/dt/config/Xsetup</code>	Anzeige Konfigurationsdatei
<code>/usr/dt/config/Xstartup</code>	Startdatei vor der Session
<code>/usr/dt/config/Xreset</code>	Datei für Rücksetzung nach der Session



`/var/dt/Xpid`

Prozeß-ID des Anmelde-Servers

`/var/dt/Xerrors`

Fehlerprotokolldatei des Anmelde-Servers

Sessionmanager konfigurieren

2

Über den Sessionmanager werden das Desktop gestartet und aktive Anwendungen, Farben, Schriftarten, Mausverhalten, Lautstärke und Tastaturklicken automatisch gesichert und wiederhergestellt.

Über den Sessionmanager kann der Benutzer

- die Startsession für alle Desktop-Benutzer anpassen
- die Umgebung und Ressourcen für alle Desktop-Benutzer anpassen
- die Session-Startnachricht ändern
- die Parameter für Session-Starthilfsprogramme und -dämonen ändern
- die Farbverwendung am Desktop für alle Benutzer ändern

<i>Was ist eine Session?</i>	<i>25</i>
<i>Session starten</i>	<i>27</i>
<i>Start einer Session</i>	<i>27</i>
<i>Zusätzliche Anpassungen beim Start der Session</i>	<i>36</i>
<i>Dateien und Verzeichnisse des Sessionmanager</i>	<i>41</i>

Was ist eine Session?

Bei einer Session handelt es sich um eine Datensammlung mit Anwendungen, Einstellungen und Ressourcen auf dem Desktop eines Benutzers. Unter "Sessionmanager" versteht man eine Gruppe von Übereinkünften und

Protokollen auf dem Desktop eines Benutzers, mit deren Hilfe die Session eines Benutzers gesichert und wiederhergestellt werden kann. Ein Benutzer kann sich am System anmelden und findet dieselben aktiven Anwendungen, Einstellungen und Ressourcen vor, wie bei der letzten Abmeldung. Meldet sich ein Benutzer zum ersten Mal am Desktop an, wird eine Standard-Startsession geladen. Danach unterscheidet der Sessionmanager zwischen aktueller und Home-Session.

Startsession

Meldet sich ein Benutzer zum ersten Mal am Desktop an, generiert der Sessionmanager die Startsession des Benutzers unter Verwendung von Systemstandardwerten. Standardmäßig werden der Dateimanager und 'Einführung in das Desktop', ein Hilfeband, gestartet.

Aktuelle Session

Unter der aktuellen Session versteht man die jeweils aktive Session des Benutzers, unabhängig davon, ob es sich um eine bei der Anmeldung wiederhergestellte gesicherte Home-Session, eine gesicherte aktuelle Session, oder die Standard-Startsession des Systems handelt. Auf der Grundlage der Einstellungen für den Start des Umgebungsmanagers des Benutzers sichert der Sessionmanager die aktuelle Session automatisch, wenn der Benutzer die Session verläßt. Bei der nächsten Anmeldung am Desktop startet der Sessionmanager die zuvor gesicherte aktuelle Session neu, d. h. das Desktop wird in demselben Status wiederhergestellt, wie bei der letzten Abmeldung des Benutzers.

Home-Session

Das Desktop kann aber auch bei jeder Anmeldung in demselben Status wiederhergestellt werden, unabhängig davon, in welchem Status es sich befand, als der Benutzer sich zuletzt abmeldete. Der Benutzer kann den Status der aktuellen Session speichern, und dann über die Einstellungen für den Start der Umgebungsparameter diese Session über den Sessionmanager bei jeder Anmeldung starten lassen.

Bildschirmspezifische Sessions

Um eine spezifische Session für einen spezifischen Bildschirm auszuführen, kann der Benutzer eine bildschirmspezifische Session erstellen. Hierzu kann der Benutzer das Verzeichnis *Home-Verzeichnis*/`.dt/sessions` in *Home-Verzeichnis*/`.dt/anzeige` kopieren, wobei es sich bei *anzeige* um den realen, nicht qualifizierten Host -Namen handelt (zum Beispiel ist `pablo:0` gültig, `pablo.gato.com:0` oder `unix:0` sind nicht gültig). Meldet sich der Benutzer am Bildschirm `pablo:0` an, startet der Sessionmanager diese bildschirmspezifische Session.

Session starten

Der Sessionmanager wird über `/usr/dt/bin/Xsession` gestartet. Meldet sich der Benutzer über den Anmeldemanager an, wird `Xsession` standardmäßig gestartet.

Der Benutzer kann auch die herkömmliche Anmeldung im Zeichenmodus (`getty`) verwenden und den Sessionmanager manuell mit Hilfsprogrammen, die einen X-Server starten, wie z. B. `xinit`, starten. Beispiel: `xinit /usr/dt/bin/Xsession`.

Start einer Session

Beim Start des Sessionmanagers werden folgende Schritte durchgeführt, um die Session eines Benutzers zu starten:

1. Die Prozedur *Home-Verzeichnis*/`.dtprofile` wird als Ausgangsobjekt verwendet.
2. Die Prozeduren `xsession.d` werden als Ausgangsobjekte verwendet.
3. Eine Begrüßungsnachricht wird angezeigt.
4. Desktop-Suchpfade werden eingerichtet.
5. Verfügbare Anwendungen werden erfaßt.
6. *Home-Verzeichnis*/`.profile` oder *Home-Verzeichnis*/`.login` werden möglicherweise zusätzlich als Ausgangsobjekte verwendet.
7. Der Nachrichtendämon ToolTalk[®] wird gestartet.

8. Die Session-Ressourcen werden geladen.
9. Der Farb-Server wird gestartet.
10. Der Arbeitsbereichsmanager wird gestartet.
11. Die Session-Anwendungen werden gestartet.

Die oben aufgeführten Schritte werden in den folgenden Abschnitten näher beschrieben.

Prozedur `.dtprofile` als Ausgangsobjekt verwenden

Beim Start einer Session verwendet die Prozedur `Xsession` die Prozedur des Benutzers *Home-Verzeichnis*/`.dtprofile` als Ausgangsobjekt. Bei der Prozedur *Home-Verzeichnis*/`.dtprofile` handelt es sich um eine Prozedur aus `/bin/sh` oder `/bin/ksh`, die Benutzern ermöglicht, Umgebungsvariablen für ihre Sessions einzurichten. Weitere Informationen zum Einrichten von Umgebungsvariablen können "Zusätzliche Anpassungen beim Start der Session" auf Seite 36 entnommen werden.

Ist die Prozedur *Home-Verzeichnis*/`.dtprofile` nicht vorhanden, wie das beispielsweise der Fall ist, wenn sich der Benutzer zum ersten Mal am Desktop angemeldet, kopiert `Xsession` die Standardprozedur für das Desktop `sys.dtprofile` in *Home-Verzeichnis*/`.dtprofile`.

Die Standardprozedur für das Desktop ist `/usr/dt/config/sys.dtprofile`. Um die Prozedur `sys.dtprofile` anzupassen, kopieren Sie `sys.dtprofile` aus `/usr/dt/config` in `/etc/dt/config` und bearbeiten die neue Datei.

Prozeduren `Xsession.d` als Ausgangsobjekte verwenden

Nach der Prozedur *Home-Verzeichnis*/`.dtprofile` verwendet die Prozedur `Xsession` die Prozeduren `Xsession.d` als Ausgangsobjekte. Diese Prozeduren werden zum Festlegen zusätzlicher Umgebungsvariablen verwendet und starten wahlweise Dämonen für die Session eines Benutzers. Die Standardprozeduren `Xsession.d` sind:

- | | |
|---------------------------|---|
| <code>0010.dtpaths</code> | dokumentiert anpaßbare Desktop-Suchpfade |
| <code>0020.dtims</code> | startet einen optionalen Eingabemethoden-Server |

0030.dttmpdir	erstellt ein temporäres Verzeichnis pro Benutzer pro Session
0040.xmbind	definiert für \$XMBINDDIR den Standardwert für das Desktop

Daneben können zusätzliche lieferantenspezifische Prozeduren in `Xsession.d` vorhanden sein.

`Xsession` verwendet zunächst alle Dateien im Verzeichnis `/etc/dt/config/Xsession.d` und anschließend die Dateien im Verzeichnis `/usr/dt/config/Xsession.d` als Ausgangsobjekte.

Die Standardprozeduren für das Desktop `Xsession.d` befinden sich im Verzeichnis `/usr/dt/config/Xsession.d`. Um eine Prozedur aus `Xsession.d` anzupassen, die Prozedur aus `/usr/dt/config/Xsession.d` in `/etc/dt/config/Xsession.d` kopieren und die neue Datei bearbeiten. Für diese Aufgabe muß Ausführungsberechtigung vorliegen.

Damit `Xsession` eine vom Benutzer erstellte Prozedur automatisch als Ausgangsobjekt verwendet, muß diese in `/etc/dt/config/Xsession.d` kopiert werden.

Hinweis – Ändert oder erstellt der Benutzer eine Prozedur für `Xsession.d`, sollte er sicherstellen, daß Vordergrundbefehle, die er absetzt, nicht zu lange dauern, da die Dauer dieser Befehle sich direkt auf die Startzeit der Session auswirkt. Ist kein Vordergrundbefehl vorhanden, wird der Start der Session blockiert. Befehle, die in einer Prozedur `Xsession.d` ausgeführt werden und die während der gesamten Dauer der Session ausgeführt werden sollen, sollten im Hintergrund ausgeführt werden.

Begrüßungsnachricht anzeigen

Nachdem *Home-Verzeichnis* `.dtprofile` und die Prozeduren `Xsession.d` als Ausgangsobjekte verwendet wurden, zeigt `Xsession` eine Begrüßungsnachricht über den gesamten Bildschirm an. Diese Begrüßungsnachricht kann angepaßt oder vollständig abgeschaltet werden. Client `dthello` zeigt die Nachricht an.

Um den Nachrichtentext zu ändern, die Optionen für `dthello` durch Änderung der Variablen `dtstart_hello[0]` ändern.

Um `dtstart_hello[0]` zu ändern, muß eine Prozedur `/etc/dt/config/Xsession.d` erstellt werden, die den neuen Wert setzt. Um die Nachricht des Tages für alle Benutzer anzuzeigen, muß eine ausführbare Prozedur `sh` oder `ksh` erstellt werden, z. B. `/etc/dt/config/Xsession.d/myvars`, und `dtstart_hello[0]` wie folgt festgelegt werden:

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

Auf dieselbe Weise können Benutzer die Begrüßungsnachricht für ihre eigenen Sessions ändern, indem sie `dtstart_hello[0]` in *Home-Verzeichnis*/`.dtprofile` festlegen.

Um die Begrüßungsnachricht abzuschalten, muß `dtstart_hello[0]=" "` festgelegt werden.

Weitere Informationen zu `dthello` können der Hilfeseite `dthello` entnommen werden.

Desktop-Suchpfade einrichten

Die Desktop-Suchpfade werden bei der Anmeldung durch `dtsearchpath` eingerichtet. `dtsearchpath` verwendet zwei Kategorien von Umgebungsvariablen:

Eingabevariablen	Systemumgebungsvariablen und Umgebungsvariablen für einzelne Benutzer, deren Werte vom Systemadministrator oder vom Endbenutzer gesetzt werden.
Ausgabevariablen	Variablen, die von <code>dtsearchpath</code> erstellt werden und denen von <code>dtsearchpath</code> Werte zugeordnet werden. Der Wert jeder Variablen ist der Suchpfad für die Desktop-Session.

Um die Befehlszeilenoptionen von `dtsearchpath` zu ändern, die Variable `dtstart_searchpath` ändern. Um die Variable `dtstart_searchpath` für alle Benutzer zu ändern, eine ausführbare Prozedur `sh` oder `ksh` (z. B. `/etc/dt/config/Xsession.d/myvars`) erstellen und `dtstart_searchpath` wie folgt festlegen:

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

Auf dieselbe Weise können Benutzer die Optionen für `dtsearchpath` für ihre eigenen Sessions ändern, indem sie `dtstart_searchpath` in *Home-Verzeichnis*/`.dtprofile` festlegen.

Weitere Informationen zu `dtsearchpath` können Kapitel 7, "Desktop-Suchpfade." entnommen werden. Weitere Informationen zu den Optionen für `dtsearchpath` können der Hilfeseite `dtsearchpath` entnommen werden.

Verfügbare Anwendungen erfassen

Nachdem die Desktop-Suchpfade eingerichtet sind, werden unter Verwendung von `dtappgather` verfügbare Anwendungen erfaßt. Um die Befehlszeilenooptionen von `dtappgather` zu ändern, muß die Variable `dtstart_appgather` geändert werden. Um die Variable `dtstart_appgather` für alle Benutzer zu ändern, eine ausführbare Prozedur `sh` oder `ksh` (z. B. `/etc/dt/config/Xsession.d/myvars`) erstellen und `dtstart_appgather` wie folgt festlegen:

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

Auf dieselbe Weise können Benutzer die Optionen für `dtappgather` für ihre eigenen Sessions ändern, indem sie `dtstart_appgather` in *Home-Verzeichnis*/`.dtprofile` festlegen.

Weitere Informationen zu den Optionen für `dtappgather` können der Hilfeseite `dtappgather (4)` entnommen werden.

Mögliche Verwendung der Prozeduren `.profile` oder `.login` als Ausgangsobjekte

`Xsession` ist in der Lage, die herkömmlichen Prozeduren *Home-Verzeichnis*/`.profile` oder *Home-Verzeichnis*/`.login` eines Benutzers als Ausgangsobjekte zu verwenden. Diese Möglichkeit ist standardmäßig inaktiviert. Damit `Xsession` die Prozeduren `.profile` oder `.login` als Ausgangsobjekte verwendet, muß `DTSOURCEPROFILE` auf `true` festgelegt werden.

Um `DTSOURCEPROFILE` für alle Benutzer zu ändern, muß eine Prozedur `/etc/dt/config/Xsession.d` erstellt werden, die den neuen Wert festlegt. Um `DTSOURCEPROFILE` für alle Benutzer auf `true` festzulegen, eine

ausführbare Prozedur `sh` oder `ksh`, z. B. `/etc/dt/config/Xsession.d/myvars` erstellen und `DTSOURCEPROFILE` wie folgt festlegen:

```
DTSOURCEPROFILE=true
```

Auf dieselbe Weise können Benutzer `DTSOURCEPROFILE` für ihre eigenen Sessions ändern, indem sie `DTSOURCEPROFILE` in *Home-Verzeichnis*/`.dtprofile` auf `true` festlegen.

Nachrichtendämon ToolTalk starten

Der Nachrichtendämon ToolTalk, `ttsession`, ermöglicht unabhängigen Anwendungen die Kommunikation untereinander, ohne direkt miteinander verbunden zu sein. Hierzu erstellen und senden Anwendungen ToolTalk-Nachrichten. `ttsession` kommuniziert im Netz und übermittelt Nachrichten.

Um die Befehlszeilenooptionen für `ttsession` zu ändern, muß die Variable `dtstart_ttsession` geändert werden. Um die Variable `dtstart_ttsession` für alle Benutzer zu ändern, eine ausführbare Prozedur `sh` oder `ksh` (z. B. `/etc/dt/config/Xsession.d/myvars`) erstellen und `dtstart_ttsession` wie folgt festlegen:

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

Auf dieselbe Weise können Benutzer die Optionen für `ttsession` für ihre eigenen Sessions ändern, indem sie `dtstart_ttsession` in *Home-Verzeichnis*/`.dtprofile` festlegen.

Weitere Informationen zu Optionen für `ttsession` können der Hilfeseite `ttsession` entnommen werden. Weitere Informationen zu `ttsession` können *Getting Started Using ToolTalk Messaging* entnommen werden.

Client des Sessionmanagers starten

An dieser Stelle startet `Xsession` `/usr/dt/bin/dtsession`, der den Session-Startprozeß fortführt.

Session-Ressourcen starten

Der Sessionmanager verwendet das X-Server-Merkmal `RESOURCE_MANAGER`, um Desktop-Ressourcen für alle Anwendungen zur Verfügung zu stellen. Der Sessionmanager lädt `RESOURCE_MANAGER`, indem sie:

- die Systemstandardressourcen lädt
- vom Systemadministrator für das System definierte Ressourcen hinzufügt
- benutzerdefinierte Ressourcen hinzufügt

Die Standardressourcen für das Desktop befinden sich in `/usr/dt/config/sprache/sys.resources`. Diese Ressourcen werden durch das Merkmal `RESOURCE_MANAGER` der Session jedes Benutzers verfügbar gemacht. Diese Datei sollte nicht bearbeitet werden, da sie bei anschließenden Desktop-Installationen überschrieben wird.

Die Anzahl der Systemstandardressourcen kann erhöht werden, indem `/etc/dt/config/sprache/sys.resources` erstellt wird. In dieser Datei können Standardressourcen außer Kraft gesetzt oder zusätzliche Ressourcen für alle Desktop-Benutzer erstellt werden. Da diese Datei während des Session-Starts mit den Standardressourcen für das Desktop gemischt wird, sollten nur neue oder aktualisierte Ressourcenspezifikationen in diese Datei gestellt werden. Die in dieser Datei angegebenen Ressourcen werden durch das Merkmal `RESOURCE_MANAGER` der Session jedes Benutzers verfügbar gemacht. Ressourcen, die in dieser Datei spezifiziert sind, haben Vorrang vor den Standardressourcen für das Desktop.

Benutzer können die Anzahl der Standardressourcen für das Desktop und der für das System definierten Ressourcen unter Verwendung ihrer Datei `Home-Verzeichnis/.xdefaults` erhöhen. Die in dieser Datei angegebenen Ressourcen werden durch das Merkmal `RESOURCE_MANAGER` der Session dieses Benutzers verfügbar gemacht. Ressourcen, die in dieser Datei spezifiziert sind, haben Vorrang vor den Standardressourcen für das Desktop oder Ressourcendateien des Systemadministrators.

Hinweis – Im Dienstprogramm X Toolkit Intrinsic wird angegeben, daß Ressourcen für eine Anwendung entweder aus `RESOURCE_MANAGER` oder aus `Home-Verzeichnis/.xdefaults` geladen werden, aber nicht aus beiden. Normalerweise würde dies bedeuten, daß die Datei `Home-Verzeichnis/.xdefaults` des Benutzers ignoriert würde. Aber da der Sessionmanager die Datei `Home-Verzeichnis/.xdefaults`, wie oben beschrieben, beim Sessionstart mit `RESOURCE_MANAGER` mischt, wird diese Datei dennoch berücksichtigt. Ändert ein Benutzer `Home-Verzeichnis/.xdefaults`, werden die Änderungen bei neuen Anwendungen erst sichtbar, wenn der Benutzer die Aktion 'Ressourcen erneut laden' aufruft. Die Aktion 'Ressourcen erneut laden' weist den Sessionmanager an, `RESOURCE_MANAGER` mit den Standardressourcen, den für das System

definierten Ressourcen und den benutzerdefinierten Ressourcen erneut zu laden. Dies bewirkt Änderungen in den für das System definierten und den benutzerdefinierten Ressourcendateien, die den Anwendungen zur Verfügung stehen.

Weitere Informationen hierzu siehe:

- “Anwendungsressourcen einstellen” auf Seite 282
- Hilfeseite `dtresourcesfile(4)`

Farb-Server starten

Der Sessionmanager dient als Farb-Server für das Desktop und stellt folgende Ressourcen in `dtsession` für dessen Konfiguration zur Verfügung.

<code>foregroundColor</code>	Steuert, ob ein Pixel der Vordergrundfarbe zugeordnet ist
<code>dynamicColor</code>	Gibt an, ob schreibgeschützte Farben zugeordnet sind
<code>shadowPixmaps</code>	Gibt an, ob Farben für Schatten über oberem oder unterem Bereich zugeordnet sind
<code>colorUse</code>	Begrenzt die Farbzurordnung
<code>writeXrdbColors</code>	Gibt an, ob die *Hintergrund- und *Vordergrundressourcen in die Ressourcendatenbank gestellt werden

Farb-Server-Ressourcen können für alle Benutzer festgelegt werden, indem `/etc/dt/config/sprache/sys.resources` erstellt wird und die Farb-Server-Ressourcen in dieser Datei definiert werden.

Auf dieselbe Weise können Benutzer Farb-Server-Ressourcen für ihre eigenen Sessions festlegen, indem sie Farb-Server-Ressourcen in `Home-Verzeichnis/.Xdefaults` definieren.

Weitere Informationen zum festlegen von Farb-Server-Ressourcen können dem Abschnitt “Farben verwalten” auf Seite 292 entnommen werden.

Arbeitsbereichssteuerung starten

Der Sessionmanager ist verantwortlich für den Start des Arbeitsbereichsmanagers. Standardmäßig wird `/usr/dt/bin/dtwm` gestartet. Mit der Ressource `wmStartupCommand` kann eine andere Fenstersteuerung angegeben werden.

Um eine andere Fenstersteuerung für alle Benutzer anzugeben, muß `/etc/dt/config/sprache/sys.resources` erstellt werden und der vollständige Pfadname sowie die Optionen für die Fenstersteuerung mit der Ressource `Dtsession*wmStartupCommand` in dieser Datei angegeben werden.

Auf dieselbe Weise können Benutzer eine andere Fenstersteuerung für ihre eigenen Sessions definieren, indem sie die Ressource `Dtsession*wmStartupCommand` in *Home-Verzeichnis*/`.Xdefaults` angeben.

Weitere Informationen zur Fenstersteuerung können Kapitel 14, "Arbeitsbereichsmanager anpassen" entnommen werden.

Session-Anwendungen starten

Beim Starten einer Session werden durch den Sessionmanager Anwendungen, die als Teil der Session gesichert wurden, erneut gestartet. Die Standardanwendungsgruppe des Systems, die als Teil der Startsession des Benutzers wiederhergestellt wird, befindet sich in `/usr/dt/config/sprache/sys.session`. Diese Datei sollte nicht bearbeitet werden, da sie bei anschließenden Desktop-Installationen überschrieben wird.

Weitere Informationen hierzu können der Hilfeseite `the dtsessionfile(4)` entnommen werden.

Ein Systemadministrator kann die Gruppe von Anwendungen, die als Teil der Startsession des Benutzers gestartet werden, ersetzen, indem er `/usr/dt/config/sprache/sys.session` in `/etc/dt/config/sprache/sys.session` kopiert und die letztgenannte Datei ändert. Im Gegensatz zu den Ressourcendateien ersetzt diese Datei die Desktop-Standarddatei vollständig, so daß eine Kopie der Systemstandarddatei erstellt werden kann und alle notwendigen Änderungen darin vorgenommen werden können.

Zusätzliche Anpassungen beim Start der Session

Hierzu gehören:

- das festlegen von Umgebungsvariablen
- das festlegen von Ressourcen
- die Verwendung bildschirmabhängiger Sessions
- die Ausführung von Prozeduren bei der Anmeldung
- die Wiederherstellung einer Sicherungs-Session

▼ Umgebungsvariablen festlegen

- ◆ Um für das System definierte Umgebungsvariablen festzulegen, wird eine Datei im Verzeichnis `/etc/dt/config/Xsession.d` erstellt, die die Variable festlegt und exportiert.

Wird beispielsweise eine ausführbare Prozedur `sh` oder `ksh`,
`/etc/dt/config/Xsession.d/myvars`,

mit der Angabe

```
export MYVARIABLE="wert"
```

erstellt, wird die Variable `MYVARIABLE` bei der nächsten Anmeldung in der Umgebung jedes Benutzers festgelegt.

- ◆ Um für einzelne Benutzer definierte Umgebungsvariablen zu setzen, wird die Variable in *Home-Verzeichnis*/`.dtprofile` festgelegt.

Beispiel:

```
export MYVARIABLE="wert"
```

wird die Variable `MYVARIABLE` bei der nächsten Anmeldung in der Umgebung des entsprechenden Benutzers festgelegt.

Hinweis – Der Sessionmanager liest die Dateien `.profile` oder `.login` nicht automatisch. Sie kann jedoch entsprechend konfiguriert werden. Informationen hierzu können dem Abschnitt “Mögliche Verwendung der Prozeduren `.profile` oder `.login` als Ausgangsobjekte” auf Seite 31 entnommen werden.

▼ Ressourcen festlegen

- ◆ Um für das System definierte Ressourcen festzulegen, werden diese Ressourcen der Datei `/etc/dt/config/sprache/sys.resources` hinzugefügt. (Möglicherweise muß diese Datei erstellt werden.)

Hinweis - `.dtprofile` unterstützt nur die Syntax von `/bin/sh` oder `/bin/ksh`.

Wird beispielsweise in `/etc/dt/config/C/sys.resources` angegeben:

```
AnApplication*resource: wert
```

wird die Ressource `AnApplication*resource` bei der nächsten Anmeldung in dem Merkmal `RESOURCE_MANAGER` jedes Benutzers festgelegt werden.

- ◆ Um für einzelne Benutzer definierte Ressourcen festzulegen, werden diese Ressourcen der Datei `Home-Verzeichnis/.Xdefaults` hinzugefügt.

▼ Bildschirmspezifische Ressourcen festlegen

Es können bildschirmspezifische Ressourcen für alle Desktop-Benutzer am System festgelegt werden. Benutzer können aber auch bildschirmspezifische Ressourcen für ihre eigenen Sessions festlegen. Dies ermöglicht, Ressourcen abhängig vom Bildschirm, an dem sich der Benutzer am Desktop anmeldet, festzulegen.

- ◆ Um bildschirmspezifische Ressourcen für alle Desktop-Benutzer am System festzulegen, muß die Datei `/etc/dt/config/sprache/sys.resources` erstellt werden, die die bildschirmspezifischen Ressourcen angibt.
- ◆ Um bildschirmspezifische Ressourcen für einzelne Benutzer festzulegen, die Ressource in `Home-Verzeichnis/.Xdefaults` festlegen.

Diese Ressourcen werden begrenzt, indem sie in bedingte Anweisungen des Übertragungsprotokollprogramms (`cpp`) eingeschlossen werden. Ein Makro `DISPLAY_anzeigenname` wird abhängig vom Wert der Variablen `$DISPLAY` definiert. Dies geschieht, indem die Satzzeichen `.` (Punkt) und `:` (Doppelpunkt) in `_` (Unterstreichungszeichen) umgesetzt, sämtliche Anzeigespezifikationen übergeben und den Ergebnissen schließlich `DISPLAY_` vorangesetzt werden.

So würde aus einem `$DISPLAY` von `:0` ein `DISPLAY_0` und ein `$DISPLAY` von `blanco.gato.com:0.0` wäre `DISPLAY_blanco_gato_com_0`. Der resultierende Wert kann als Teil eines `cpp`-Tests in einer Session-Ressourcendatei verwendet werden. Wird beispielsweise in `/etc/dt/config/C/sys.resources` folgendes angegeben:

```
Myapp*resource: value
#ifdef DISPLAY_blanco_gato_com_0
Myapp*resource: specialvalue1
#endif
#ifdef DISPLAY_pablo_gato_com_0
Myapp*resource: specialvalue2
#endif
```

wird die Ressource `MyApp*resource` in `RESOURCE_MANAGER` auf `specialvalue1` festgelegt, wenn sich der Benutzer am Bildschirm `blanco.gato.com:0` anmeldet, auf `specialvalue2`, wenn sich der Benutzer an `pablo.gato.com:0` anmeldet, und auf `value`, wenn sich der Benutzer an einem anderen Bildschirm anmeldet.

▼ Anwendungen für die Startsession ändern

Für die Startsession des Benutzers können andere, spezielle Anwendungen angegeben werden.

1. Hierzu `/usr/dt/config/sprache/sys.session` in `/etc/dt/config/sprache/sys.session` kopieren.
2. Die neue Datei `sys.session` ändern.
Jeder Eintrag in `sys.session` hat folgendes Format:
3. `dtsmcmd -cmd befehl_und_optionen`

Um eine zusätzliche Anwendung als Teil der Startsession eines Benutzers zu starten, einen neuen Eintrag in `sys.session` mit einem vollständigen Pfadnamen angeben. Soll, beispielsweise, `/usr/bin/X11/xclock` als Teil der Startsession eines Benutzers gestartet werden, einen Eintrag `xclock` in `/etc/dt/config/C/sys.resources` hinzufügen:

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

▼ Bildschirmspezifische Session einrichten

Ein Benutzer kann eine bildschirmspezifische Session einrichten, um eine Session optimal an einen bestimmten Bildschirm anzupassen.

- ◆ Hierzu das Verzeichnis *Home-Verzeichnis/.dt/sessions* in *Home-Verzeichnis/.dt/anzeige* kopieren, wobei es sich bei *anzeige* um den realen, nicht qualifizierten Host-Namen handelt (zum Beispiel ist `pablo:0` gültig, `pablo.gato.com:0` oder `unix:0` sind nicht gültig).

Beispiel für die Erstellung einer bildschirmspezifischen Session für Bildschirm `pablo.gato.com:0`:

```
cp -r Home-Verzeichnis/.dt/sessions Home-Verzeichnis/.dt/pablo:0
```

Bei der nächsten Anmeldung des Benutzers am Bildschirm `pablo.gato.com:0` startet der Sessionmanager diese bildschirmspezifische Session.

Zusätzliche Befehle beim Starten einer Session und der Abmeldung ausführen

Benutzer können festlegen, daß bei der Anmeldung von Desktop-Sessions zusätzliche Befehle gestartet werden. Dies ist nützlich bei der Einrichtung von X-Einstellungen, die über den Sessionmanager nicht gesichert werden. Beispielsweise kann der Benutzer mit `xsetroot` den Arbeitsbereichszeiger anpassen. Es können Anwendungen gestartet werden, die über den Sessionmanager nicht gesichert und wiederhergestellt werden können. Wird eine Anwendung bei der Wiederherstellung einer Session nicht erneut gestartet, kann der Benutzer über diese zusätzlichen Befehle den Client starten.

▼ Zusätzliche Befehle beim Starten einer Session ausführen

- ◆ Die Datei *Home-Verzeichnis/.dt/sessionetc*, die die gewünschten Befehlen enthält, erstellen.

Im allgemeinen handelt es sich bei dieser Datei um eine Prozedur, für die Ausführungsberechtigung vorliegen muß. Prozesse, die in `sessionetc` gestartet wurden, sollten im Hintergrund ausgeführt werden.

Hinweis – `sessionetc` sollte nicht verwendet werden, um Clients zu starten, die über den Sessionmanager automatisch wiederhergestellt werden. Es könnten auf diese Weise mehrere Kopien der Anwendung gleichzeitig gestartet werden, die der Benutzer möglicherweise nicht sofort sieht, da die Fenster übereinander gestapelt sind.

▼ Zusätzliche Befehle bei der Abmeldung ausführen

Eine Zusatzdatei zu `sessionetc` ist `sessionexit`. Mit `sessionexit` werden Arbeitsgänge am Ende einer Session durchgeführt, die vom Sessionmanager nicht übernommen werden.

- ◆ Die Datei *Home-Verzeichnis*/`.dt/sessionexit` erstellen.

Wie bei `sessionetc`, handelt es sich auch bei dieser Datei um eine Prozedur, für die Ausführungsberechtigung vorliegen muß.

▼ Session aus einer Sicherungskopie wiederherstellen

Wird eine Session über den Sessionmanager gesichert, werden die Session-Informationen im Verzeichnis *Home-Verzeichnis*/`.dt/sessions` oder, bei bildschirmspezifischen Sessions, im Verzeichnis *Home-Verzeichnis*/`.dt/anzeige` gespeichert. In diesen Verzeichnissen erstellt der Sessionmanager ein Unterverzeichnis mit dem Namen `current` oder `home`, um Informationen für die aktuelle bzw. die Home-Session zu speichern. Bevor die Session-Informationen gespeichert werden, erstellt der Sessionmanager eine Sicherungskopie der vorherigen Session mit diesem Namen und speichert sie in `current.old` bzw. `home.old`.

1. Um eine Session aus einer Sicherungskopie wiederherzustellen, muß sich der Benutzer über die Failsafe Session oder die Befehlszeile am Anmeldefenster anmelden.
2. Anschließend muß er das Verzeichnis mit der gesicherten Session in das Verzeichnis mit dem aktiven Namen kopieren. Um, beispielsweise, die gesicherte Home-Session wiederherzustellen, folgendes eingeben:

```
cp -r Home-Verzeichnis/.dt/sessions/home.old Home-Verzeichnis/.dt/sessions/home
```

Bildschirmspezifische Sessions können auf dieselbe Weise wiederhergestellt werden.

▼ Probleme beim Starten einer Session ermitteln

- ◆ Die Datei *Home-Verzeichnis*/`.dt/startlog` überprüfen.

Der Sessionmanager protokolliert den Start jeder Session eines Benutzers in dieser Datei.

Dateien und Verzeichnisse des Sessionmanager

- `/usr/dt/bin/Xsession`
- `/usr/dt/config/Xsession.d/*`
- `/usr/dt/bin/dtsession`
- `/usr/dt/bin/dtsession_res`
- *Home-Verzeichnis*/`.dt/sessions/current`
- *Home-Verzeichnis*/`.dt/sessions/home`
- *Home-Verzeichnis*/`.dt/anzeige/current`
- *Home-Verzeichnis*/`.dt/anzeige/home`

Anwendungen hinzufügen und verwalten

3 

Beim Anwendungsmanager handelt es sich um den Desktop-Container für Anwendungen, die dem Benutzer zur Verfügung stehen.

<i>Struktur des Anwendungsmanagers</i>	<i>44</i>
<i>Anwendungen zum Anwendungsmanager hinzufügen</i>	<i>48</i>
<i>Allgemeine Gruppen von Anwendungen erstellen und verwalten</i>	<i>51</i>
<i>Suchpfad für Anwendungen ändern</i>	<i>53</i>
<i>Allgemeine Anwendungsmanagerverwaltung</i>	<i>54</i>
<i>Texteditor und den Terminal-Emulator ändern</i>	<i>55</i>

Struktur des Anwendungsmanagers

Die oberste Ebene des Anwendungsmanagers enthält im allgemeinen Verzeichnisse. Jedes dieser Verzeichnisse und dessen Inhalt wird als *Gruppe von Anwendungen* bezeichnet.

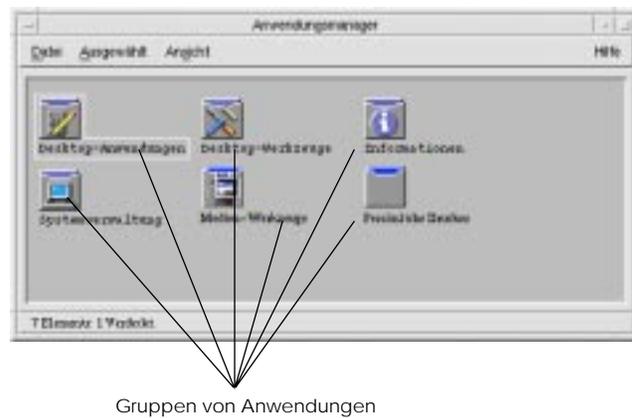


Abbildung 3-1 Gruppen von Anwendungen im Anwendungsmanager

Die Gruppen von Anwendungen und deren Inhalt werden aus vielen verschiedenen Positionen, sowohl lokal als auch im Netz, zusammengestellt.

Verzeichnisposition des Anwendungsmanagers

Im Dateisystem handelt es sich beim Anwendungsmanager um das Verzeichnis `/var/dt/appconfig/appmanager/anmeldung-hostname-anzeige`. Dieses Verzeichnis wird bei der Anmeldung eines Benutzers dynamisch erstellt.

Meldet sich beispielsweise der Benutzer `ronv` am Bildschirm `wxyz:0` an, wird für den Anwendungsmanager das Verzeichnis `/var/dt/appconfig/appmanager/ronv-wxyz-0` erstellt.

Wie der Anwendungsmanager Anwendungen sucht und zusammenstellt

Der Anwendungsmanager wird aufgebaut, indem lokale und ferne Gruppen von Anwendungen zusammengestellt werden. Die Gruppen von Anwendungen werden aus Verzeichnissen zusammengestellt, die auf dem Anwendungssuchpfad liegen.

Der Standardanwendungssuchpfad besteht aus folgenden Positionen:

Bereich	Position
integriert	<code>/usr/dt/appconfig/appmanager/<i>sprache</i></code>
systemweit	<code>/etc/dt/appconfig/appmanager/<i>sprache</i></code>
persönlich	<code>Home-Verzeichnis/.dt/appmanager</code>

Um die oberste Ebene des Anwendungsmanagers zu erstellen, werden zum Zeitpunkt der Anmeldung Verbindungen von den Gruppen von Anwendungen (Verzeichnisse), die sich in Verzeichnissen im Anwendungssuchpfad befinden, zum Anwendungsmanagerverzeichnis `/var/dt/appconfig/appmanager/anmeldung-hostname-anzeige` hergestellt. Dies erfolgt durch das Desktop-Dienstprogramm `dtappgather`, das vom Anmeldemanager automatisch ausgeführt wird, nachdem sich der Benutzer erfolgreich angemeldet hat.

Beispielsweise stellt das Desktop folgende integrierte Gruppe von Anwendungen zur Verfügung:

```
/usr/dt/appconfig/appmanager/sprache/Desktop_Werkzeuge
```

Zum Zeitpunkt der Anmeldung wird eine symbolische Verbindung hergestellt zu:

```
/var/dt/appconfig/appmanager/anmeldung-hostname-anzeige/Desktop_Werkzeuge
```

Der Anwendungssuchpfad kann ferne Verzeichnisse einschließen. Dadurch können Gruppen von Anwendungen aus Systemen im gesamten Netz zusammengestellt werden. Weitere Informationen könnendem Abschnitt "Anwendungs-Server dem Anwendungssuchpfad hinzufügen" auf Seite 53. entnommen werden.

Vorrangsregeln beim Zusammenstellen von Anwendungen

Sind Anwendungen im Suchpfad mehrfach vorhanden, haben für den einzelnen Benutzer definierte Gruppen von Anwendungen Vorrang vor für das System definierten Gruppen, und für das System definierte Gruppen haben Vorrang vor integrierten Gruppen. Sind, beispielsweise, sowohl `/usr/dt/appconfig/appmanager/C/Desktop_Werkzeuge` als auch `/etc/dt/appconfig/appmanager/C/Desktop_Werkzeuge` vorhanden, wird die Gruppe von Anwendungen unter `/etc` verwendet.

Gruppen von Anwendungen, die mit dem Standard- Desktop zur Verfügung gestellt werden

Das nicht angepaßte Desktop stellt vier Gruppen von Anwendungen zur Verfügung:

- Desktop_Anwendungen
- Desktop_Werkzeuge
- Informationen
- Systemadministrator

Beispiel für die Zusammenstellung von Gruppen von Anwendungen

Abbildung 3-2 zeigt ein Anwendungsmanagerfenster mit verschiedenen Gruppen von Anwendungen. Tabelle 3-1 zeigt die Verzeichnisse, aus denen die Gruppen von Anwendungen zusammengestellt werden.



Abbildung 3-2 Ein typisches Anwendungsmanagerfenster

Tabelle 3-1 Quelle der Gruppen von Anwendungen in Abbildung 3-2

Name	Verzeichnis
CAD_Anwendungen	/net/ApServA/etc/dt/appconfig/appmanager/C/CAD_App
Zeichen_Anwendungen	/etc/dt/appconfig/appmanager/C/DrawingApp
Desktop_Anwendungena	/usr/dt/appconfig/appmanager/C/Desktop_Apps
Desktop_Werkzeuge	/usr/dt/appconfig/appmanager/C/Desktop_Tools
Informationen	/usr/dt/appconfig/appmanager/C/Information
Systemadministrator	/etc/dt/appconfig/appmanager/C/System_Admin
Tabellenkalkulation	/users/anna/.dt/appmanager/MySpreadSheet
Medien_Werkzeuge	/etc/dt/appconfig/appmanager/C/Media_Tools

Wurden die Gruppen von Anwendungen Informationen oder Systemadministrator angepaßt, werden sie stattdessen aus /etc/dt/appconfig/appmanager/C erstellt.

Die Gruppe CAD_Anwendungen wird bei der Zusammenstellung erfaßt, weil ein System mit dem Namen `ApServA` zum Anwendungssuchpfad hinzugefügt wurde (Abschnitt “Anwendungs-Server dem Anwendungssuchpfad hinzufügen” auf Seite 53). Bei der Tabellenkalkulation handelt es sich um eine für den einzelnen Benutzer definierte Gruppe von Anwendungen, die nur dem Benutzer `anna` zur Verfügung steht.

Anwendungen zum Anwendungsmanager hinzufügen

Wurde eine Anwendung zum Anwendungsmanager hinzugefügt, erscheint ein Symbol in einer Gruppe von Anwendungen, über das diese Anwendung gestartet wird.

Viele Anwendungen stellen eine Gruppe von Anwendungen zur Verfügung. Bei der Gruppe von Anwendungen handelt es sich um ein Verzeichnis auf der obersten Ebene des Anwendungsmanagers, das das Anwendungssymbol sowie weitere der Anwendung zugehörige Dateien enthält.

Einige Anwendungen verfügen möglicherweise nicht über eine eigene Gruppe von Anwendungen. Statt dessen befindet sich das Symbol, mit dem diese Anwendung gestartet wird, in einer allgemeinen Gruppe von Anwendungen. Beispielsweise kann der Benutzer eine leere Gruppe von Anwendungen mit dem Namen “Spiele” erstellen, die als Container für alle Spiele, die am System installiert werden, verwendet werden kann.

Verschiedene Möglichkeiten, Anwendungen zum Anwendungsmanager hinzuzufügen

Es gibt zwei verschiedene Möglichkeiten, eine Anwendung dem Anwendungsmanager hinzuzufügen:

- Registrierung der Anwendung
- Hinzufügen eines Anwendungssymbols ohne Registrierung der Anwendung

Registrierung der Anwendung

Die Registrierung der Anwendung stellt die vollständige Integration der Anwendung zur Verfügung.

Eine registrierte Anwendung:

- hat ihre eigene Gruppen von Anwendungen
- ist eine Anwendung, bei der die Desktop-Konfigurationsdateien an einer Position zusammengestellt sind. Diese Gruppe von Desktop-Konfigurationsdateien wird *Registrierungspaket* genannt.
- verfügt möglicherweise über einen registrierten Hilfeband

Es gibt zwei Möglichkeiten, eine Anwendung zu registrieren:

- Wird eine desktop-fähige Anwendung installiert, wird automatisch eine Registrierung durchgeführt. Weitere Informationen können dem Abschnitt “Desktop-fähige Anwendungen zum Anwendungsmanager hinzufügen” auf Seite 50 entnommen werden.
- Eine bereits vorhandene Anwendung kann durch Erstellen eines Registrierungspakets registriert werden. Weitere Informationen können dem Abschnitt “Eine bereits vorhandene oder eine nicht Desktop-fähige Anwendung registrieren” auf Seite 50 entnommen werden.

Die Verwendung eines Registrierungspakets vereinfacht die Verwaltung einer Anwendung auf dem Desktop. Das Registrierungspaket darf nicht an derselben Stelle im Dateisystem erstellt werden, an der sich die Desktop-Konfigurationdateien befinden.

Anwendungen ohne ein Registrierungspaket hinzufügen

Soll der Anwendungsmanager nur ein Symbol zum Starten der Anwendung enthalten, wird diese Art und Weise, eine Anwendung hinzuzufügen, bevorzugt.

Eine Anwendung, die ohne ein Registrierungspaket hinzugefügt wird,

- hat möglicherweise eine eigene Gruppe von Anwendungen, aber normalerweise wird das Symbol in eine bereits vorhandene Gruppe von Anwendungen gestellt
- ist eine Anwendung, deren Desktop-Konfigurationsdateien sich direkt in den Suchpfaden des Desktops befinden.

Weitere Informationen können dem Abschnitt “Anwendungssymbole einer bereits vorhandenen Gruppe von Anwendungen hinzufügen” auf Seite 50 entnommen werden.

▼ Desktop-fähige Anwendungen zum Anwendungsmanager hinzufügen

Bei einer Desktop-fähigen Anwendung handelt es sich um eine Anwendung, die bei der Installation automatisch im Anwendungsmanager registriert wird. Die Dateigruppen der Anwendung enthalten auch das für das Desktop erforderliche Registrierungspaket.

1. Die Anwendung mit den mitgelieferten Instruktionen installieren.
2. Nachdem die Installation vollständig ausgeführt ist, 'Anwendungen erneut laden' in der Gruppe von Anwendungen 'Desktop-Werkzeuge' doppelt anklicken.
3. Überprüfen, ob die Installation vollständig ausgeführt ist.
 - a. Den Anwendungsmanager öffnen und überprüfen, ob die neue Gruppe von Anwendungen vorhanden ist.
 - b. Um die Anwendung zu öffnen, die Gruppe von Anwendungen öffnen und das Symbol der Anwendung doppelt anklicken.

▼ Eine bereits vorhandene oder eine nicht Desktop-fähige Anwendung registrieren

Hierbei handelt es sich um die bevorzugte Art und Weise, eine Anwendung auf dem Desktop vollständig zu integrieren.

Das Desktop stellt ein Werkzeug, `dtappintegrate`, zur Verfügung, das Verbindungen zwischen den Registrierungsdateien und den Verzeichnissen im Suchpfad des Desktops herstellt.

Die Registrierung auf dem Desktop wird in Kapitel 4, "Anwendungen registrieren" näher erläutert.

▼ Anwendungssymbole einer bereits vorhandenen Gruppe von Anwendungen hinzufügen

In diesem Abschnitt wird erläutert, wie ein Anwendungssymbol zu einer bereits vorhandenen Gruppe von Anwendungen hinzugefügt wird.

Beispielsweise stellt das Desktop eine Gruppe von Anwendungen mit dem Namen 'Systemadministrator' zur Verfügung, die für verschiedene Anwendungen und Prozeduren der Systemverwaltung reserviert ist. Bei einer

Prozedur, die von Benutzern häufig ausgeführt wird, besteht möglicherweise der Wunsch, diese Prozedur durch Doppelklicken auf einem Symbol in der Gruppe von Anwendungen 'Systemadministrator' auszuführen.

1. Mit 'Aktion erstellen' kann eine Aktionsdefinition für die Anwendung erstellt werden.

Weitere Informationen zu Aktionen erstellen können Kapitel 9, "Aktionen und Datentypen mit 'Aktion erstellen' erstellen" entnommen werden.

2. Eine ausführbare Datei mit demselben Namen wie dem Aktionsnamen in dem Verzeichnis für die Gruppe von Anwendungen erstellen. Der Inhalt der Datei ist irrelevant.

Wurde, beispielsweise, eine Aktion mit dem Namen "Bereinigung" erstellt, die ein Systemverwaltungswerkzeug ausführt, muß die ausführbare Datei `/etc/dt/appconfig/appmanager/sprache/System_Admin/Bereinigung` erstellt werden.

Allgemeine Gruppen von Anwendungen erstellen und verwalten

Bei einer allgemeinen Anwendung handelt es sich um eine Gruppe von Anwendungen (Verzeichnis), die nicht einem bestimmten Anwendungsprodukt zugeordnet ist. Beispielsweise handelt es sich bei der integrierten Gruppe von Anwendungen 'Desktop_Werkzeuge' um eine allgemeine Gruppe, die Symbole für eine große Anzahl von Anwendungen enthält, zwischen denen zwar Beziehungen bestehen, die aber nicht Bestandteil eines einzigen Produkts sind.

Es können zusätzliche allgemeine Gruppen von Anwendungen erstellt werden. Beispielsweise könnte der Benutzer eine Gruppe mit dem Namen Spiele erstellen, in die er alle am System verfügbare Spiele stellt.

Eine allgemeine Gruppe von Anwendungen kann sowohl für das System als auch für den einzelnen Benutzer definiert werden.

▼ Eine für das System definierte allgemeine Gruppe von Anwendungen erstellen

1. Als Benutzer 'root' anmelden.
2. Ein Verzeichnis in `/etc/dt/appconfig/appmanager/sprache` erstellen.

Der Name des Verzeichnisses wird zum Namen der Gruppe von Anwendungen.

3. Auf 'Anwendungen erneut laden' in der Gruppe von Anwendungen 'Desktop-Werkzeuge' doppelklicken.

▼ Eine für den einzelnen Benutzer definierte allgemeine Gruppe von Anwendungen erstellen

1. Ein Verzeichnis in *Home-Verzeichnis*/.dt/appmanager erstellen.

Der Name des Verzeichnisses wird zum Namen der Gruppe von Anwendungen.

2. Auf 'Anwendungen erneut laden' in der Gruppe von Anwendungen 'Desktop_Werkzeuge' doppelklicken.

▼ Eine integrierte Gruppe von Anwendungen anpassen

1. Als Benutzer 'root' anmelden.

2. Befindet sich die Gruppe von Anwendungen in /usr/dt/appconfig/appmanager/*sprache*, die Gruppe von Anwendungen in /etc/dt/appconfig/appmanager/*sprache* kopieren.

Der folgende Befehl kopiert, beispielsweise, die Gruppe von Anwendungen 'Desktop_Werkzeuge':

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

Die neue Kopie der Gruppe von Anwendungen hat Vorrang vor der integrierten Version.

3. Die Kopie der Gruppe von Anwendungen ändern. So können, beispielsweise, neue Aktionsdateien (ausführbare Dateien mit demselben Namen wie Aktionen) hinzugefügt werden.
4. Damit die Änderungen sichtbar werden, muß der Benutzer sich ab- und wieder anmelden.

Suchpfad für Anwendungen ändern

Der wichtigste Grund für die Änderung des Anwendungssuchpfads ist das Hinzufügen eines Anwendungs-Servers. Wird ein Anwendungs-Server zum Suchpfad hinzugefügt, stellt der Anwendungsmanager alle für das System definierte Gruppen von Anwendungen des Servers zusammen.

Weitere Informationen zum Anwendungssuchpfad können dem Abschnitt “Anwendungssuchpfad” auf Seite 131 entnommen werden.

Standardsuchpfad

Der Standardanwendungssuchpfad enthält die folgenden Verzeichnisse:

Bereich	Suchpfadverzeichnis
persönlich	<i>Home-Verzeichnis</i> /.dt/appmanager
systemweit	/etc/dt/appconfig/appmanager/ <i>sprache</i>
integriert	/usr/dt/appconfig/appmanager/ <i>sprache</i>

Anwendungs-Server dem Anwendungssuchpfad hinzufügen

Zusätzlich zu der Änderung des Anwendungssuchpfads ist es möglicherweise nötig, zusätzliche Konfiguration-Aufgaben auszuführen, um die Kommunikation mit dem Anwendungs-Server zu ermöglichen. Weitere Informationen können dem Abschnitt “Anwendungs-Services verwalten” auf Seite 111 entnommen werden.

▼ *Einen für das System definierten Anwendungssuchpfad setzen*

1. Als Benutzer ‘root’ anmelden.
2. Ist die Datei /etc/dt/config/Xsession.d/0010.dtpaths nicht vorhanden, wird sie durch Kopieren von /usr/dt/config/Xsession.d/0010.dtpaths erstellt.
3. /etc/dt/Xsession.d/0010.paths zum Bearbeiten öffnen. Eine Zeile mit der Variablen DTSPSYSAPPHOSTS bearbeiten oder hinzufügen:
`DTSPSYSAPPHOSTS=hostname: [,hostname]`

Die folgende Zeile fügt, beispielsweise, das System `ApServA` zum Anwendungssuchpfad hinzu:

```
DTSPSYSAPPHOSTS=ApServA:
```

4. Alle Benutzer am System müssen darüber informiert werden, daß sie sich ab- und wieder anmelden müssen, damit die Änderung sichtbar wird.

▼ **Einen für den einzelnen Benutzer definierten Anwendungssuchpfad setzen**

1. *Home-Verzeichnis* / `.dtprofile` zum Bearbeiten öffnen.
2. Eine Zeile mit der Variablen `DTSPUSERAPPHOSTS` bearbeiten oder hinzufügen:

```
DTSPUSERAPPHOSTS=hostname: [ ,hostname ]
```

Die folgende Zeile fügt, beispielsweise, die Systeme `ApServB` und `ApServC` zum Anwendungssuchpfad hinzu:

```
DTSPUSERAPPHOSTS=ApServB: ,ApServC:
```

3. Ab- und wieder anmelden.

Allgemeine Anwendungsmanagerverwaltung

Zur allgemeinen Anwendungsmanagerverwaltung gehören:

- das Entfernen einer Anwendung
- das erneute Lesen der Datenbank für Anwendungen während einer Session

▼ **Anwendung entfernen**

Wurde eine Anwendung mit dem Werkzeug `dtappintegrate` registriert, kann dieser Prozeß mit `dtappintegrate` auch wieder rückgängig gemacht werden. Ist eine Anwendung nicht registriert, wird ihre Gruppe von Anwendungen aus dem Anwendungsmanager entfernt und ihre Aktionen, Datentypen, Symbole und Hilfe sind nicht länger verfügbar.

1. Als Benutzer 'root' anmelden.
2. Folgenden Befehl ausführen:

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

▼ Anwendungsmanager während einer Session aktualisieren

Werden Anwendungen zum Anwendungsmanager hinzugefügt und sollen diese Änderungen sofort sichtbar werden, muß der Anwendungsmanager erneut generiert werden.

- ◆ Die Gruppe von Anwendungen 'Desktop_Werkzeuge' öffnen und auf 'Anwendungen erneut laden' doppelt klicken.

Die Aktion 'Anwendungen erneut laden' kann zum Aktualisieren des Anwendungsmanagers verwendet werden, wenn Anwendungen zu einem Anwendungs-Server hinzugefügt werden. Aber 'Anwendungen erneut laden' stellt nicht fest, daß Anwendungen vom Anwendungs-Server entfernt wurden oder von einer Position zu einer anderen verschoben wurden. Diese Änderungen werden erst sichtbar, wenn der Benutzer sich ab- und wieder anmeldet.

Texteditor und den Terminal-Emulator ändern

Sowohl die Anwendung 'Texteditor' als auch die Anwendung 'Terminal-Emulator' können durch Auswahl eines Bedienelements im Bedienfeld oder durch Doppelklicken eines Symbols im Anwendungsmanager gestartet werden.

Diese Anwendungen können auch durch andere Desktop-Aktivitäten gestartet werden.

- Die Anwendung 'Texteditor' wird geöffnet wenn der Benutzer eine Textdatei im Dateimanager auswählt und im Menü 'Ausgewählt Öffnen' auswählt. Der Standard-Texteditor ist `dtpad`.
- Ein Terminal-Emulator wird ausgeführt, wenn ein Benutzer im Menü 'Datei' des Dateimanagers 'Terminal-Fenster Öffnen' auswählt, oder wenn eine Aktion ein Terminal-Emulator-Fenster öffnet. Der Standard-Terminal-Emulator ist `dtterm`.

Das Desktop kann so konfiguriert werden, daß in diesen Situationen ein anderer Texteditor oder ein anderer Terminal-Emulator verwendet wird.

▼ Den standardmäßigen Texteditor oder Terminal-Emulator ändern

1. Soll die Änderung für das gesamte System gelten, als Benutzer 'root' anmelden.
2. Eine Aktion für den neuen Texteditor oder Terminal-Emulator erstellen.
 - Hierfür kann die Anwendung 'Aktion erstellen' verwendet werden. Abbildung 3-3 zeigt ein Fenster 'Aktion erstellen', in dem eine Anwendung mit dem Namen TextPad angegeben ist. Weitere Informationen zu 'Aktion erstellen' können Kapitel 9, "Aktionen und Datentypen mit 'Aktion erstellen' erstellen" entnommen werden.



Abbildung 3-3 Fenster Aktion erstellen

- **Oder aber**, der Benutzer erstellt die Aktionsdefinition manuell, zum Beispiel:

```
ACTION TextPad
{
    LABEL           TextPad
    TYPE            COMMAND
    WINDOW_TYPE     NO_STDIO
}
```

```
EXEC_STRING    /usr/TP/bin/TextPad %(File)Arg_1%
DESCRIPTION    Dieses Symbol doppelklicken, um \ die
Anwendung TextPad zu starten.
}
```

Weitere Informationen zum manuellen Erstellen von Aktionsdefinitionen können Kapitel 10, "Aktionen manuell erstellen" entnommen werden.

3. Die Konfigurationsdatei mit der neuen Aktion in das entsprechende Verzeichnis stellen:
 - systemweit `/etc/dt/appconfig/types/sprache`
 - persönlich: *Home-Verzeichnis*/`.dt/types`
4. Ist die Datei zutreffende Datei `user-prefs.dt` nicht bereits vorhanden, wird sie durch Kopieren von `/usr/dt/appconfig/types/sprache/user-prefs.dt` in folgende Verzeichnisse erstellt:
 - systemweit: Verzeichnis `/etc/dt/appconfig/types/sprache`
 - persönlich: Verzeichnis *Home-Verzeichnis*/`.dt/types`
5. Die Texteditor- oder Terminal-Aktion in der systemweiten oder persönlichen Datei `user-prefs.dt` bearbeiten. Die Zeile `MAP_ACTION` so ändern, daß die Aktion an die neue Aktion adressiert wird.

Beispielsweise kann die Zeile

```
MAP_ACTION Dtpad
```

wie folgt geändert werden:

```
MAP_ACTION TxtPd
```

6. Die Datei `user-prefs.dt` sichern.
7. Auf 'Aktionen erneut laden' in der Gruppe von Anwendungen 'Desktop_Werkzeuge' doppelklicken, um die Datenbank für Aktionen erneut zu laden.

Anwendungen registrieren

4

In diesem Kapitel wird beschrieben, wie ein Registrierungspaket für eine Anwendung erstellt wird, und wie die Anwendung am Desktop registriert wird.

<i>Übersicht über die Anwendungsregistrierung</i>	<i>60</i>
<i>Allgemeine Schritte bei der Registrierung einer Anwendung</i>	<i>64</i>
<i>Schritt 1: Schriftarten- und Farbbressourcen ändern</i>	<i>65</i>
<i>Schritt 2: Desktop-Anwendungsstammverzeichnis erstellen</i>	<i>66</i>
<i>Schritt 3: Registrierungspaketverzeichnisse erstellen</i>	<i>67</i>
<i>Schritt 4: Aktionen und Datentypen für die Anwendung erstellen</i>	<i>70</i>
<i>Schritt 5: Hilfedateien dem Registrierungspaket hinzufügen</i>	<i>73</i>
<i>Schritt 6: Symbole für die Anwendung erstellen</i>	<i>74</i>
<i>Schritt 7: Die Gruppe von Anwendungen erstellen</i>	<i>76</i>
<i>Schritt 8: Anwendungen mit dtappintegrate registrieren</i>	<i>83</i>
<i>Beispiel für die Erstellung eines Registrierungspakets</i>	<i>86</i>

Eine Anwendung, die am Desktop vollständig registriert ist, verfügt über:

- eine eigene Gruppe von Anwendungen auf der obersten Ebene des Anwendungsmanagers,
- eine Aktion, die die Anwendung startet. Die Aktion wird durch ein Symbol in der Gruppe von Anwendungen dargestellt,
- sowie, wahlweise, über Datentypen für ihre Datendateien.

Bei der Anwendungsregistrierung handelt es sich um eine Operation, die die Ausführung der Anwendung nicht beeinflusst:

- Die ausführbare Datei der Anwendung wird nicht geändert. Daher können auch bereits vorhandene Anwendungen am System registriert werden.
- Es ist nicht erforderlich, daß eine der gelieferten Dateien der Anwendung (z. B. die ausführbare Datei `app-defaults`) an eine andere Position im Dateisystem verschoben wird.
- Die Anwendungsregistrierung kann leicht rückgängig gemacht werden. Das Werkzeug `dtappintegrate`, mit dem Anwendungen registriert werden, stellt eine Befehlszeilenoption zur Verfügung, mit der der Prozeß rückgängig gemacht werden kann.

Ein Registrierungspaket wird erstellt durch:

- einen Systemadministrator, der eine bereits vorhandene Anwendung auf dem Desktop registrieren wil.
- einen Softwareprogrammierer, der ein Installationspaket für eine Desktop-fähige Anwendung erstellen will.

Übersicht über die Anwendungsregistrierung

Dieser Abschnitt erläutert:

- den Zweck der Anwendungsregistrierung
- die Funktionen, die die Anwendungsregistrierung einer Anwendung zur Verfügung stellt

Hinweis – Ein ausführliches Beispiel für die Registrierung einer bereits vorhandenen Anwendung kann dem Abschnitt “Beispiel für die Erstellung eines Registrierungspakets” auf Seite 86 entnommen werden.

Funktionen, die die Anwendungsregistrierung zur Verfügung stellt

Die Anwendungsregistrierung stellt Benutzern eine grafische Möglichkeit zur Verfügung, um:

- eine Anwendung zu suchen.

Nach der Installation ist die Anwendung im Anwendungsmanager “registriert” und verfügt über eine eigene Gruppe von Anwendungen.

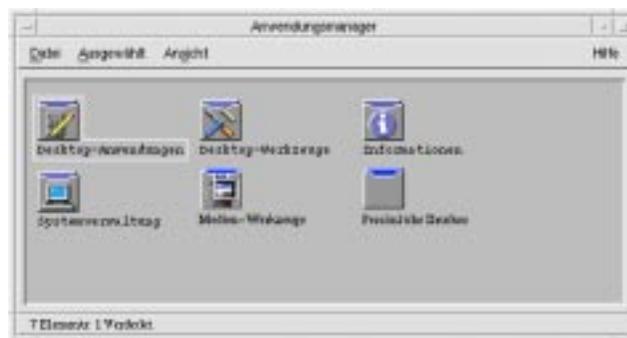


Abbildung 4-1 Gruppen von Anwendungen auf der obersten Ebene des Anwendungsmanagers

- eine Anwendung zu starten.

Die Gruppe von Anwendungen für die Anwendung enthält ein Symbol, das der Benutzer doppelklicken kann, um die Anwendung zu starten.



Abbildung 4-2 Eine Gruppe von Anwendungen mit einem Symbol zum Starten der Anwendung

- Datendateien zu erkennen und zu bearbeiten. Die Datendateien der Anwendung verfügen über ein eindeutiges Symbol im Dateimanager.

Der Benutzer kann Datendateisymbole verwenden, um

- die Anwendung zu starten (öffnen)
- Datendateien zu drucken



Abbildung 4-3 Menü einer Datendatei mit 'Öffnen' und 'Drucken'

- andere Operationen, wie beispielsweise das elektronische Versenden von Post, Komprimieren, Anzeigen oder Abspielen von (Audio-) Daten durchzuführen.

Zweck der Anwendungsregistrierung

Eine registrierte Desktop-Anwendung verfügt über bestimmte Konfigurationsdateien, die das Desktop verwendet, um die Benutzerschnittstelle der Anwendung zur Verfügung zu stellen:

- Definitionsdateien für Aktionen und Datentypen
- Symbolbilddateien (Pixmap oder Bitmap)
- ein Verzeichnis und Dateien, die die Gruppe von Anwendungen erstellen
- wahlweise Desktop-Hilfdateien und Bedienfelddefinitionsdateien

Damit diese Dateien vom Desktop erkannt und verwendet werden können, müssen sie in bestimmten Verzeichnissen stehen, die durch die Suchpfade des Desktops angegeben werden.

Die Verwaltung einer Anwendung wird erschwert, wenn deren Konfigurationsdateien auf viele verschiedene Verzeichnisse verteilt sind. Daher gestattet das Desktop einer Anwendung, alle ihre Desktop-Konfigurationsdateien in einem Verzeichnis zusammengestellt aufzubewahren. Diese Zusammenstellung von Dateien wird als *Registrierungspaket* bezeichnet.

Ist die Anwendung Desktop-fähig, stellt sie ein Registrierungspaket als Bestandteil des Installationspakets zur Verfügung. Erstellt der Systemadministrator die Konfigurationsdateien selbst, kann er das Registrierungspaket selbst erstellen.

Die Konfigurationsdateien in dem Registrierungspaket sind für das Desktop nicht verfügbar, weil sie sich nicht in den entsprechenden Verzeichnissen des Suchpfads befinden. Der Prozeß, bei dem diese Dateien an die zutreffende Position gestellt werden, wird als *Registrierung* oder *Integrierung* der Anwendung bezeichnet.

Das Desktop stellt ein Werkzeug, `dtappintegrate`, zur Verfügung, das die Registrierung durchführt, indem es symbolisch verbundene Darstellungen der Dateien in den entsprechen Verzeichnissen des Suchpfads erstellt.

Viele Desktop-fähige Anwendungen führen `dtappintegrate` während des Installationsprozesses automatisch aus. Integriert ein Systemadministrator eine bereits vorhandene Anwendung, kann er `dtappintegrate` selbst ausführen, nachdem er ein Registrierungspaket erstellt hat.

Sobald eine Anwendung auf dem Desktop eines Systems registriert ist, steht die Anwendung allen Benutzern am System zur Verfügung. Ist das System als ein Desktop-Anwendungs-Server konfiguriert, steht die Anwendung außerdem allen anderen Systemen im Netzwerk zur Verfügung.

Das Werkzeug `dtappintegrate` verfügt über eine Befehlszeilenoption, die den Prozeß rückgängig macht, indem sie die Verbindungen abbricht. Es ist daher kein Problem, die Anwendung aus dem Anwendungsmanager zu entfernen, um sie einem anderen Anwendungs-Server hinzuzufügen oder sie zu aktualisieren.

Allgemeine Schritte bei der Registrierung einer Anwendung

Hinweis – Ein ausführliches Beispiel, bei dem anhand dieser Schritte ein Anwendungspaket erstellt wird, kann dem Abschnitt “Beispiel für die Erstellung eines Registrierungspakets” auf Seite 86 entnommen werden.

1. Sämtliche Anwendungsressourcen, die Schriftarten und Farben setzen, ändern. Geschieht dies nicht, funktionieren die dynamischen Schriftarten und Farben des Desktops nicht richtig.

Weitere Informationen können dem Abschnitt “Schritt 1: Schriftarten- und Farbressourcen ändern” auf Seite 65 entnommen werden.

2. Ein Anwendungsstammverzeichnis erstellen.

Weitere Informationen können dem Abschnitt “Schritt 2: Desktop-Anwendungsstammverzeichnis erstellen” auf Seite 66 entnommen werden.

3. Die Verzeichnisstruktur unterhalb des Anwendungsstammverzeichnisses erstellen.

Weitere Informationen können dem Abschnitt “Schritt 3: Registrierungspaketverzeichnisse erstellen” auf Seite 67 entnommen werden.

4. Die Aktionen und Datentypen für die Anwendung erstellen.

Weitere Informationen können dem Abschnitt “Schritt 4: Aktionen und Datentypen für die Anwendung erstellen” auf Seite 70 entnommen werden.

5. Die Hilfedateien in ein entsprechendes Verzeichnis stellen.

Weitere Informationen können dem Abschnitt “Schritt 5: Hilfedateien dem Registrierungspaket hinzufügen” auf Seite 73 entnommen werden.

6. Die Symbole für die Anwendung erstellen.

Weitere Informationen können dem Abschnitt “Schritt 6: Symbole für die Anwendung erstellen” auf Seite 74 entnommen werden.

7. Die Gruppe von Anwendungen für die Anwendung erstellen.

Weitere Informationen können dem Abschnitt “Schritt 7: Die Gruppe von Anwendungen erstellen” auf Seite 76 entnommen werden.

8. Die Anwendung mit `dtappintegrate` registrieren.

Weitere Informationen können dem Abschnitt “Schritt 8: Anwendungen mit `dtappintegrate` registrieren” auf Seite 83 entnommen werden.

Schritt 1: Schriftarten- und Farbressourcen ändern

Hinweis – Ein Beispiel für das Ändern von Ressourcen für eine Anwendung kann Schritt 1 des Abschnitts “Beispiel für die Erstellung eines Registrierungs pakets” auf Seite 87 entnommen werden.

Das Desktop stellt Mechanismen für das Setzen und Bearbeiten von Schnittstellenschriftarten und Fensterfarben zur Verfügung. Damit eine Anwendung diese Mechanismen ohne Probleme verwenden kann, muß möglicherweise die Datei `app-defaults` der Anwendung geändert werden.

Schriftartenressourcen ändern

Hinweis – Dieser Abschnitt bezieht sich auf Anwendungen, die mit OSF/Motif 1.2™ (oder neueren Versionen) erstellt wurden. Der Umgebungsmanager kann keine Schnittstellenschriftarten für Anwendungen setzen, die mit älteren Versionen von OSF/Motif geschrieben wurden.

Der Desktop-Umgebungsmanager setzt Schnittstellenschriftarten für Anwendungen, die mit OSF/Motif 1.2 (oder neueren Versionen) erstellt wurden, sofern für die Anwendung nicht anwendungsspezifische Schnittstellenschriftarten angegeben sind.

Der Umgebungsmanager stellt zwei Schriftarten zur Verfügung:

Systemschriftart wird vom System für Bezeichnungen, Menüs und Tasten verwendet.

Benutzerschriftart wird für bearbeitbare Bereiche, wie z. B. Textfelder, verwendet.

Jede Schriftart steht in sieben Größen, die im Dialogfenster Schriftarten mit 1 bis 7 bezeichnet sind, zur Verfügung. Die Schriftarten des Umgebungsmanagers werden durch die Umgebungsmanagerressourcen, die

in
`/usr/dt/app-defaults/sprache/Dtstyle` gesetzt werden, mit den tatsächlichen Schriftarten am System verbunden.

Soll die Anwendung 'Schriftarten' des Umgebungsmanagers verwenden, müssen Anwendungsressourcen, die Schnittstellenschriftarten angeben, entfernt werden. Das Desktop legt die Ressourcen der Anwendung automatisch entsprechend fest:

<code>FontList</code>	auf Systemschriftart festgelegt
<code>XmText*FontList</code>	auf Benutzerschriftart festgelegt
<code>XmTextField*FontList</code>	auf Benutzerschriftart festgelegt

Farbressourcen ändern

Der Umgebungsmanager kann Anwendungsfarben dynamisch ändern. Die Anwendung muß ein Client unter OSF/Motif 1.1 oder 1.2 sein. Clients, die mit einem anderen Hilfsprogramm geschrieben wurden, können ihre Farben nicht dynamisch ändern. In diesem Fall muß der Client erneut gestartet werden.

Der einfachste Weg, um die dynamischen Farben, die das Desktop zur Verfügung stellt, zu verwenden, ist, alle Anwendungsfarbressourcen für Vordergrund- und Hintergrundfarbe zu entfernen.

Schritt 2: Desktop-Anwendungsstammverzeichnis erstellen

Hinweis – Ein Beispiel für die Erstellung des Desktop-Anwendungsstammverzeichnisses für eine Anwendung kann Schritt 2 des Abschnitts "Beispiel für die Erstellung eines Registrierungspakets" auf Seite 87 entnommen werden.

Die Registrierungspaketdateien für die Anwendung werden in ein Verzeichnis gestellt, das Anwendungsstammverzeichnis, oder *app_root* genannt wird. Das Verzeichnis *app_root*, das für die Desktop-Konfigurationsdateien verwendet wird, kann dasselbe Verzeichnis *app_root* sein, das für die Installation der Anwendung verwendet wurde, es kann aber auch an einer anderen Position sein.

Angenommen, eine Anwendung ist unter einem Verzeichnis `/usr/BTE` installiert. Dasselbe Verzeichnis könnte auch als `app_root` für die Desktop-Konfigurationsdateien verwendet werden. Dennoch empfiehlt es sich, bei der Integrierung einer bereits vorhandenen nicht Desktop-fähigen Anwendung, ein anderes Desktop-Verzeichnis `app_root` zu erstellen. Dadurch wird verhindert, daß die erstellten Konfigurationsdateien bei einer möglichen Aktualisierung der Anwendung überschrieben werden.

So könnte ein Systemadministrator, beispielsweise, ein Verzeichnis `/etc/desktop_approots/BTE` als das Desktop-Verzeichnis `app_root` erstellen.

Schritt 3: Registrierungs paketverzeichnisse erstellen

Hinweis – Ein Beispiel für die Erstellung der Registrierungs paketverzeichnisse für eine Anwendung kann Schritt 3 des Abschnitts “Beispiel für die Erstellung eines Registrierungs pakets” auf Seite 88 entnommen werden.

Bei dem Registrierungs paket handelt es sich um die Gruppe der Desktop-Konfigurationsdateien, die das Desktop verwendet, um eine grafische Schnittstelle für die Anwendung zur Verfügung zu stellen.

Inhalt des Registrierungs pakets

Die Desktop-Konfigurationsdateien enthalten:

- Definitionsdateien für Aktionen und Datentypen
- Symbolbilddateien
- ein Verzeichnis für eine Gruppe von Anwendungen und dessen Inhalt
- wahlweise: Hilfedatendateien und eine Bedienfeldkonfigurationsdatei

Das Registrierungspaket ist unter einem Verzeichnis der obersten Ebene, dem Anwendungsstammverzeichnis oder *app_root*, zusammengestellt.

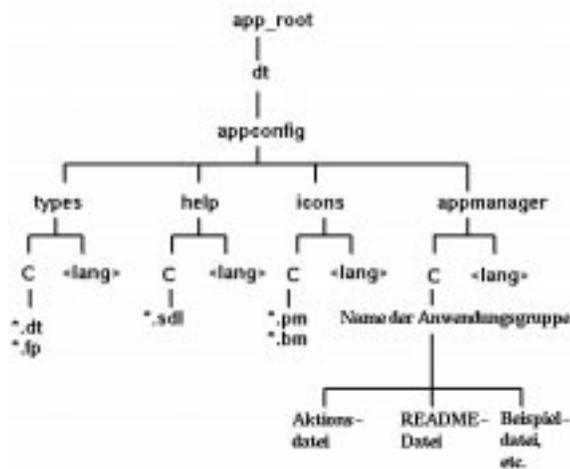


Abbildung 4-4 Ein Registrierungspaket unterhalb eines Anwendungsstammverzeichnisses

Die wichtigsten Kategorien von Konfigurationsfeldern unter dem Verzeichnis *app_root/dt/appconfig* sind:

Unterverzeichnis Inhalt

- types Definitionen für Aktionen und Datentypen
- help Desktop-Hilfedateien
- icons Bitmap- und Pixmap-Abbilddateien, die von den Aktionen und Datentypen der Anwendung verwendet werden
- appmanager das Verzeichnis mit Inhalt, das die Gruppe von Anwendungen erstellt

jede dieser Kategorien verfügt über Unterverzeichnisse für sprachabhängige Dateien. Standardsprachdateien stehen im Verzeichnis *C*.

Registrierungspaket erstellen

- ♦ Folgende Verzeichnisse erstellen. Werden sprachabhängige Konfigurationsdateien verwendet, muß ein eigenes Verzeichnis für jede Sprache erstellt werden. Wird nur eine Sprache verwendet, werden die Dateien in das Verzeichnis C gestellt.
 - *app_root*/dt/appconfig/types/*sprache*
 - *app_root*/dt/appconfig/help/*sprache*
 - *app_root*/dt/appconfig/icons/*sprache*
 - *app_root*/dt/appconfig/appmanager/*sprache*/*anwendungsgruppe_name*, wobei *anwendungsgruppe_name* der Name der Gruppe von Anwendungen ist.

Beispielsweise zeigt Abbildung 4-5 den Anwendungsmanager mit einer Gruppe, für die *anwendungsgruppe_name* "Medien_Werkzeuge" ist.

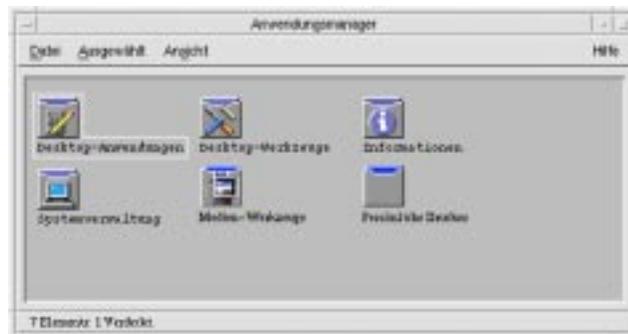


Abbildung 4-5 Gruppe von Anwendungen auf der obersten Ebene des Anwendungsmanagers

Das Werkzeug `dtappintegrate` kann nur für die Desktop-Konfigurationsdateien in den Verzeichnissen `types`, `help`, `icons` und `appmanager` angewendet werden. Die binäre ausführbare Datei `app-defaults` der Anwendung und Nachrichten katalogdateien werden separat verwaltet.

Schritt 4: Aktionen und Datentypen für die Anwendung erstellen

Hinweis – Ein Beispiel für die Erstellung der Aktionen und Datentypen für eine Anwendung kann Schritt 4 des Abschnitts “Beispiel für die Erstellung eines Registrierungs Pakets” auf Seite 88 entnommen werden.

Aktionen und Datentypen stellen eine Benutzerschnittstelle für die Anwendung zur Verfügung.

- Aktionen stellen eine Benutzerschnittstelle zur Verfügung, von der aus der Befehl die Anwendung startet.
- Datentypen stellen ein angepaßtes Erscheinungsbild und angepaßtes Verhalten für die Datendateien der Anwendung zur Verfügung.

Aktionen und Datentypen, die für eine Anwendung erforderlich sind

Bei typischen Anwendungen sind folgende Definitionen für Aktionen und Datentypen erforderlich:

- Eine Aktion, die die Anwendung öffnet.
- Ein Datentyp für die Datendateien der Anwendung. Wird ein Datentyp erstellt, müssen folgende Aktionen ebenfalls erstellt werden:
 - eine Aktion ‘Öffnen’ für die Datendateien der Anwendung
 - eine Aktion ‘Drucken’ für die Datendateien der Anwendung
- Ein Datentyp für die Gruppe von Anwendungen. (Weitere Informationen können dem Abschnitt “Eindeutige Symbole für Gruppen von Anwendungen definieren” auf Seite 78 entnommen werden.)

Eine Einführung zur Verwendung von Aktionen und Datentypen auf dem Desktop enthält Kapitel 8, “Einführung in Aktionen und Datentypen”

Position der Konfigurationsdateien mit den Definitionen für Aktionen und Datentyp

Aktionen und Datentypen werden in Konfigurationsdateien definiert. Das einzige Erfordernis bei der Benennung von Dateien, die Definitionen für Aktionen und Datentypen enthalten, ist, daß sie mit dem Suffix `.dt` versehen sein müssen. Konventionellerweise wird die Datei `aktion_name.dt` oder `anwendung_name.dt` genannt.

Dateien, die Aktionen und Datentypen enthalten, werden unter das Anwendungsstammverzeichnis in das Verzeichnis `app_root/dt/appconfig/types/sprache` gestellt. Standardwert für *sprache* ist C.

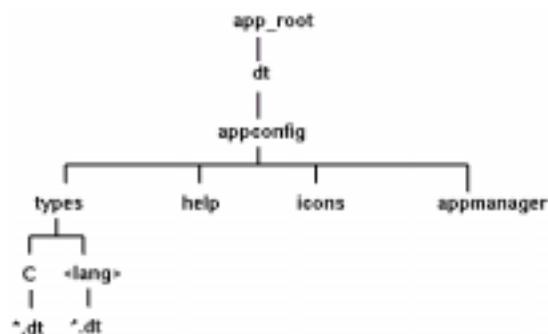


Abbildung 4-6 Definitionsdateien für Aktionen und Datentypen

Möglichkeiten, Aktionen und Datentypen zu erstellen

Definitionen für Aktionen und Datentypen für eine Anwendung können auf zwei verschiedene Art und Weisen erstellt werden:

- mit dem Werkzeug 'Aktion erstellen'

'Aktion erstellen' stellt eine benutzerfreundliche Schnittstelle mit Textfeldern zur Verfügung, die der Benutzer ausfüllen muß. Hier muß der Benutzer bestimmte Einschränkungen hinnehmen.

- oder manuell

Hierfür ist es erforderlich, daß der Benutzer die Syntax für die Erstellung der Definitionen erlernt, aber dafür erhält er Zugriff auf sämtliche Funktionen.

▼ Aktionen und Datentypen mit 'Aktion erstellen' erstellen

Bei dieser Prozedur wird das Dienstprogramm 'Aktion erstellen' verwendet, um eine Aktion und Datentypen für die Anwendung zu erstellen.

Weitere Informationen zu 'Aktion erstellen' können der zugehörigen Online-Hilfefunktion oder Kapitel 9, "Aktionen und Datentypen mit 'Aktion erstellen' erstellen" entnommen werden.

1. Die Gruppe von Anwendungen 'Desktop_Anwendungen' öffnen und 'Aktion erstellen' doppelklicken.
2. Mit 'Aktion erstellen' die Definitionen für Aktionen und Datentypen für die Anwendung und deren Datentyp erstellen.

Die Konfigurationsdatei, die mit 'Aktion erstellen' erstellt wurde, wird in das Verzeichnis *Home-Verzeichnis/.dt/type/aktion_name.dt* geschrieben. Die Aktionsdatei (die ausführbare Datei mit demselben Namen wie die Aktion) wird in das Home-Verzeichnis gestellt.

3. Die Aktion mit der Aktionsdatei aus dem Home-Verzeichnis testen.
4. Die Definitionsdatei für Aktionen, *Home-Verzeichnis/.dt/type/aktion_name.dt*, in das Verzeichnis *app_root/dt/appconfig/types/sprache* kopieren.
5. Nachdem das Verzeichnis für die Gruppe von Anwendungen erstellt wurde (siehe Abschnitt, "Schritt 7: Die Gruppe von Anwendungen erstellen" auf Seite 76), die Aktionsdatei *Home-Verzeichnis/aktion_name* in das Verzeichnis *app_root/dt/appconfig/appmanager/sprache/anwendungsgruppe_name* kopieren.

▼ Aktionen und Datentypen manuell erstellen

- ◆ Eine Konfigurationsdatei mit den Definitionen für Aktionen und Datentypen für die Anwendung erstellen.

Definitionsdateien für Aktionen und Datentypen müssen der Benennungskonvention *name.dt* entsprechen.

Der Benutzer kann alle Definitionen für Aktionen und Datentypen in eine einzige Datei stellen oder er kann sie auf viele verschiedene Dateien verteilen. Für jede Datei sollte ein Dateiname verwendet werden, aus dem für Systemadministratoren die Anwendung leicht erkennbar ist.

Die Namen für Aktionen und Datentypen müssen aus einem Wort bestehen (keine eingegebenen Leerzeichen). Unterstreichungszeichen dürfen verwendet werden. Konventionellerweise wird der erste Buchstabe des Aktions- oder

Datentypnamens als Großbuchstabe geschrieben. Es darf kein bereits vorhandener Aktions- oder Dateiname verwendet werden. Es sollte ein Name verwendet werden, aus dem für fortgeschrittene Benutzer und Systemadministratoren die Anwendung leicht erkennbar ist.

Soll das Symbol der Anwendung mit einem anderen Namen als dem Aktionsnamen bezeichnet werden, muß ein Feld LABEL in der Definition für die Aktion eingefügt werden.

Weitere Informationen zu der Erstellung von Aktionen und Datentypen können folgenden Kapiteln entnommen werden:

- Kapitel 8, "Einführung in Aktionen und Datentypen"
- Kapitel 9, "Aktionen und Datentypen mit 'Aktion erstellen' erstellen"
- Kapitel 10, "Aktionen manuell erstellen"
- Kapitel 11, "Datentypen manuell erstellen"

Schritt 5: Hilfedateien dem Registrierungspaket hinzufügen

Hinweis – Ein Beispiel für das Hinzufügen von Hilfedateien zu einem Registrierungspaket kann Schritt 5 des Abschnitts "Beispiel für die Erstellung eines Registrierungspakets" auf Seite 90 entnommen werden.

Verfügt die Anwendung über einen Desktop-Hilfebund (einen Hilfebund, der mit dem Help Developer's Kit des Desktops erstellt wurde), sollte die Stammdatei für den Hilfebund (*.sdl) in das Verzeichnis *app_root/appconfig/help/sprache* gestellt werden.

Grafiken, die in den Hilfedateien verwendet werden, werden normalerweise in ein eigenes Unterverzeichnis für Grafiken gestellt. Die Grafiken müssen in bezug auf die Stammdatei für den Hilfebund (*.sdl) in demselben Verzeichnis stehen, wie bei der Erstellung des Hilfebands.

Stellt die Anwendung keinen Hilfebund zur Verfügung, kann der Benutzer, falls er über den Help Developer's Kit verfügt, einen Hilfebund erstellen.

Es gibt zwei Stufen der Integration bei einem Hilfebund:

- Vollständige Integration

Ist Desktop-Hilfe vollständig integriert, kann auf den Hilfeband von der Anwendung aus zugegriffen werden, zum Beispiel über 'Kontexthilfe' und das Menü 'Hilfe'. Für die vollständige Integration ist die Änderung der ausführbaren Dateien der Anwendung erforderlich.

- Partielle Integration

Ist die Desktop-Hilfe partiell integriert, ist sie von der obersten Ebene des Hilfemanager verfügbar. Aber von den Fenstern der Anwendung kann nicht auf den Hilfeband zugegriffen werden. Es kann eine Aktion erstellt werden, über die aus der Gruppe von Anwendungen auf die Hilfe zugegriffen werden kann. Das folgende Beispiel für eine Aktion zeigt den Hilfeband in der Hilfestammdatei `MyApp.sdl`:

```
ACTION OpenMyAppHelp
{
    LABEL           MyAppHelp
    ARG_COUNT       0
    TYPE            COMMAND
    WINDOW_TYPE     NO_STDIO
    EXEC_STRING     /usr/dt/bin/dthelpview -helpVolume MyApp
    DESCRIPTION     Anzeige hilft für die Anwendung MyApp.
}
```

Schritt 6: Symbole für die Anwendung erstellen

Hinweis – Ein Beispiel für die Erstellung der Symboldateien für eine Anwendung kann Schritt 6 des Abschnitts "Beispiel für die Erstellung eines Registrierungs pakets" auf Seite 91 entnommen werden.

Das Desktop stellt Standardsymbole für Aktionen, Datentypen und Gruppen von Anwendungen zur Verfügung. Aber möglicherweise hat der Benutzer den Wunsch, eindeutige Symbole für die Anwendung zu erstellen.

Symbole werden in das Verzeichnis `app_root/dt/appconfig/icons/sprache` gestellt.

Für das Desktop erforderliche Symbole

Die Anwendung verwendet folgende Symbolabbilder auf dem Desktop:

- **Aktionssymbol.** Dieses Symbol muß der Benutzer doppelklicken, um die Anwendung (Aktionen) zu starten. Auf dieses Symbol wird im Feld `ICON` der Aktion, die die Anwendung startet, bezug genommen.

Es sollten drei Größen bereitgestellt werden: klein, mittel und groß.

- **Datentypsymboll.** Dieses Symbol wird verwendet, um die Datendateien der Anwendung im Dateimanager darzustellen. Auf dieses Symbol wird im Feld `ICON` der Datentypdefinition bezug genommen.

Unterstützt die Anwendung mehrere Datentypen, sollte für jeden Datentyp ein anderes Symbol zur Verfügung gestellt werden.

Es sollten zwei Größen bereitgestellt werden: klein und mittel.

- **Symbol für Gruppe von Anwendungen.** Dieses Symbol stellt das Verzeichnis auf der obersten Ebene des Anwendungsmanagers dar. Auf dieses Symbol wird im Feld `ICON` der Datentypdefinition für die Gruppe von Anwendungen bezug genommen. (Weitere Informationen können dem Abschnitt "Schritt 7: Die Gruppe von Anwendungen erstellen" auf Seite 76 entnommen werden.)

Es sollten zwei Größen bereitgestellt werden: klein und mittel.

Es ist möglicherweise notwendig, sowohl Pixmap- als auch Bitmap-Versionen jedes Symbols bereitzustellen, um Farb- (acht Bit und größer) und Monochrombildschirme (weniger als acht Bit) zu unterstützen.

Table 4-1 Benennungskonventionen für Symboldateien

Größe	Pixel-Abmessung	Bitmap-Name	Pixmap-Name
klein	16 x 16	<i>Dateiname ohne Pfad</i> . t . bm	<i>Dateiname ohne Pfad</i> . t . pm
mittel	32 x 32	<i>Dateiname ohne Pfad</i> . m . bm	<i>Dateiname ohne Pfad</i> . m . pm
groß	48 x 48	<i>Dateiname ohne Pfad</i> . l . bm	<i>Dateiname ohne Pfad</i> . l . pm

Werden keine Bitmap-Dateien zur Verfügung gestellt, ordnet das Desktop die Farbspezifikationen der Pixmap-Dateien Schwarz- und Weißwerten zu. Dennoch kommt es möglicherweise zu Abweichungen von dem vom Benutzer gewünschten Erscheinungsbild.

Weitere Informationen zu Symbolen können dem Abschnitt “Symbolbilddateien” auf Seite 224 entnommen werden.

Schritt 7: Die Gruppe von Anwendungen erstellen

Hinweis – Ein Beispiel für die Erstellung der Gruppe von Anwendungen kann Schritt 7 dem Abschnitt “Beispiel für die Erstellung eines Registrierungspakets” on Seite 91 entnommen werden.

Nachdem die Definitionen für Aktionen und Datentypen für die Anwendung erstellt wurden, müssen die Konfigurationsdateien für das, was der Benutzer am Bildschirm tatsächlich sieht, nämlich die Gruppe von Anwendungen und deren Inhalt, erstellt werden.

Bei der Gruppe von Anwendungen handelt es sich um ein Verzeichnis auf der obersten Ebene des Anwendungsmanagers (siehe Abbildung 4-1 auf Seite 61).

Die Gruppe von Anwendungen wird in drei Schritten erstellt:

- das Verzeichnis für die Gruppe von Anwendungen im Registrierungspaket erstellen.
- wahlweise: für die Gruppe von Anwendungen ein eindeutiges Symbol erstellen. Hierzu muß die Datentypdefinition für das Verzeichnis für die Gruppe von Anwendungen erstellt werden.
- den Inhalt der Gruppe von Anwendungen erstellen.

Das Verzeichnis für die Gruppe von Anwendungen erstellen

Um eine Gruppe von Anwendungen zu erstellen, die Verzeichnisse in dem Registrierungspaket unter `appmanager`, wie in Abbildung 4-7 gezeigt, erstellen

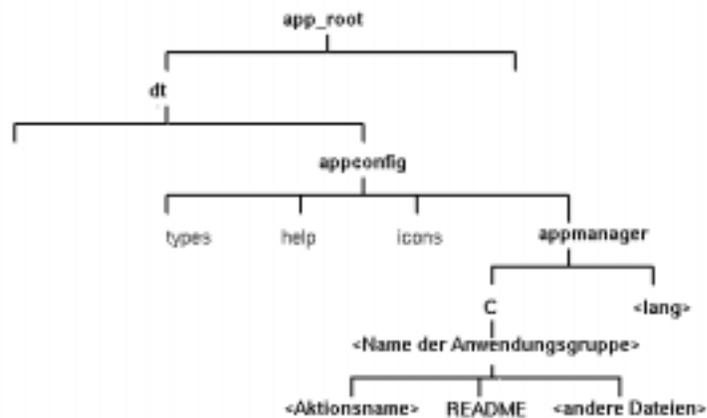


Abbildung 4-7 Das Verzeichnis `appmanager`

Name der Gruppe von Anwendungen

Der Name `<Name der Anwendungsgruppe>` in Abbildung 4-8 ist der Name für die Gruppe von Anwendungen.

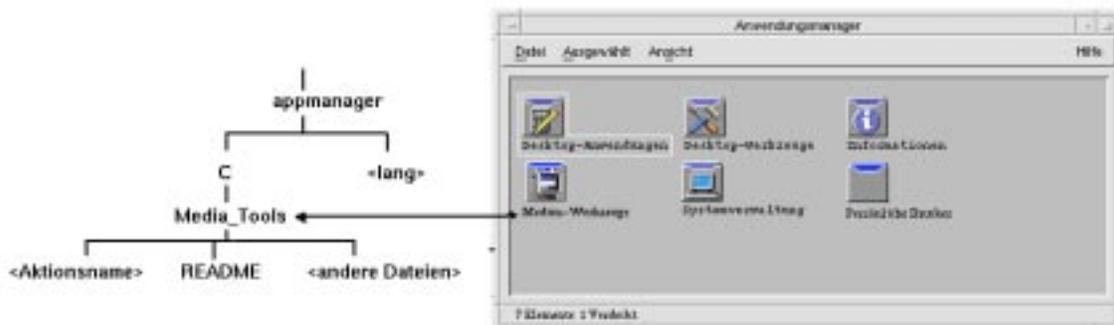


Abbildung 4-8 Der Name der Gruppe von Anwendungen (`<Name der Anwendungsgruppe>`)

Bei dem Namen kann es sich um jeden zulässigen Datei- (bzw. Verzeichnis-) namen handeln. Ein deskriptiver Name ist empfehlenswert.

Eindeutige Symbole für Gruppen von Anwendungen definieren

Das Desktop stellt ein Standardsymbol für Gruppen von Anwendungen zur Verfügung. Dennoch kann der Benutzer ein eigenes Symbol erstellen.

Möchte der Benutzer ein eindeutiges Symbol für die Gruppe von Anwendungen erstellen, muß er

- einen Datentyp für das Verzeichnis, das auf der obersten Ebene des Anwendungsmanager erscheint, erstellen.
- die Aktionen 'Öffnen' und 'Drucken' für den Datentyp erstellen.

Angenommen, der Benutzer möchte eine Gruppe von Anwendungen mit dem Namen Medienwerkzeuge erstellen. Die folgende Datentypdefinition in einer Datei `app_root/dt/appconfig/types/sprache/name.dt` ordnet der Gruppe von Anwendungen ein eindeutiges Symbol zu.

```
DATA_ATTRIBUTES Medien_WerkzeugeAppgroup
{
    ACTIONS      OpenInPlace,OpenNewView
    ICON         Medien_Werkzeuge
    DESCRIPTION  Double-click to open the Medien_Werkzeuge \
                Gruppe von Anwendungen
}
DATA_CRITERIA      Medien_WerkzeugeAppgroupCriteria1
{
    DATA_ATTRIBUTES_NAME Medien_WerkzeugeAppgroup
    MODE              d
    PATH_PATTERN      */appmanager/*/Medien_Werkzeuge
}
```

Der Abschnitt Attribute der Definition gibt an, welches Symbol verwendet werden soll. Der Abschnitt Kriterien der Definition gibt an, daß der Datentyp für jedes Verzeichnis mit Namen `Medien_Werkzeuge` definiert wird, das ein Unterverzeichnis des Verzeichnisses `appmanager` ist.

Abbildung 4-9 zeigt die Beziehung zwischen dem Namen der Gruppe von Anwendungen und der Datentypdefinition. Das Feld `PATH_PATTERN` in der Datentypdefinition verbindet ein eindeutiges Symbol mit der Gruppe von Anwendungen.

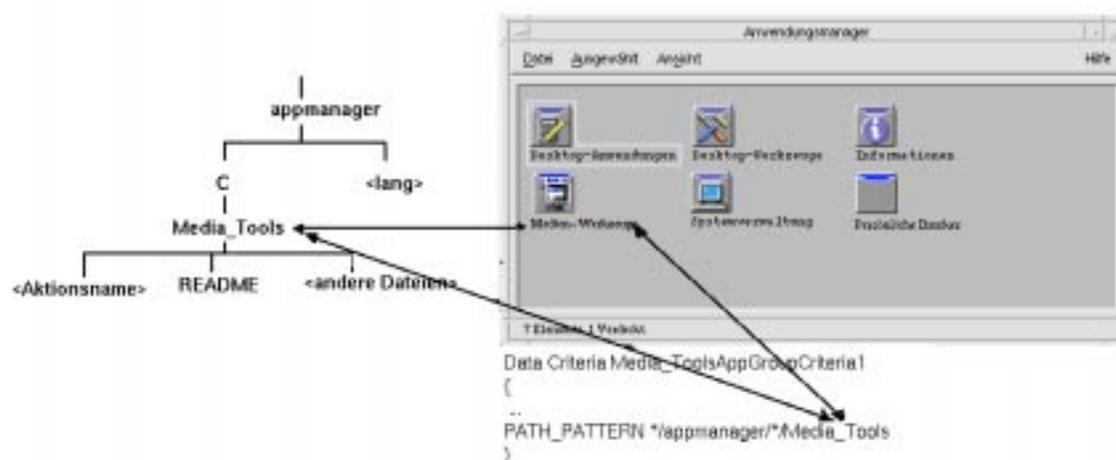


Abbildung 4-9 Zuordnung eines Symbols zu einer Gruppe von Anwendungen

Anschließend sollte der Benutzer die Aktionen 'Öffnen' und 'Drucken' für den Datentyp der Gruppe von Anwendungen erstellen:

```

ACTION Open
{
    ARG_TYPE    Medien_WerkzeugeAppGroup
    TYPE        MAP
    MAP_ACTION  OpenAppGroup
}

ACTION Print
{
    ARG_TYPE    Medien_WerkzeugeAppGroup
    TYPE        MAP
    MAP_ACTION  PrintAppGroup
}

```

Bei den Aktionen OpenAppGroup und PrintAppGroup handelt es sich um integrierte Aktionen, die in `/usr/dt/appconfig/types/sprache/dtappman.dt` definiert sind.

Inhalt der Gruppe von Anwendungen erstellen

Das wichtigste Element in der Gruppe von Anwendungen ist ein Symbol, mit dem die Anwendung gestartet wird (ein Aktionssymbol). Enthält die Gruppe von Anwendungen mehrere Anwendungen, gibt es normalerweise ein Symbol für jede Anwendung.

Zusätzlich zu einem oder mehreren Aktionssymbolen kann die Gruppe von Anwendungen noch folgendes enthalten:

- eine oder mehrere README-Dateien
- eine oder mehrere Beispieldatendateien
- Schablonen
- ein Symbol, das der Benutzer doppelklicken kann, um Hilfeinformationen anzuzeigen
- eine Hilfeseite
- ein spezialisiertes Bedienelement im Bedienfeld

Die Gruppe von Anwendungen kann Unterverzeichnisse enthalten.

Aktionsdatei (Anwendungssymbol) erstellen

Die Gruppe von Anwendungen sollte ein Symbol enthalten, das die Anwendung startet. Besteht die Gruppe aus mehreren Anwendungen, sollte für jede Anwendung ein Symbol vorhanden sein. Diese Symbole werden *Anwendungssymbole* oder *Aktionssymbole* genannt, da sie eine zugrundeliegende Aktion darstellen.

Ein Aktionssymbol wird erstellt, indem eine ausführbare Datei mit demselben Namen wie die Aktion, die sie ausführen soll, erstellt wird:

```
app_root/dt/appconfig/appmanager/anwendungsgruppe_name/aktion_name
```

Die Datei wird *Aktionsdatei* genannt, weil sie eine zugrundeliegende Aktion bildhaft darstellen soll.

Wurde, beispielsweise, eine Aktion mit dem Namen 'BesterTextEditor' erstellt, die die Anwendung 'BesterTextEditor' ausführt, muß eine ausführbare Datei mit Namen 'BesterTextEditor' erstellt werden. Im Dateimanager und dem Anwendungsmanager verwendet die Aktionsdatei das Symbolbild, das in der Definition für die Aktion angegeben wurde.

Ein spezialisiertes Bedienelement im Bedienfeld erstellen

In den meisten Fällen, ist es nicht notwendig, eine Definition für das Bedienelement im Bedienfeld zur Verfügung zu stellen. Der Benutzer kann die Anwendung zum Bedienfeld hinzufügen, indem er das Aktionssymbol dem Bedienelement `Install Icon` in einer Bedientafel übergibt.

Möglicherweise möchte der Benutzer eine Bedienfeldkonfigurationsdatei erstellen, die eine Bedienelementdefinition für eine Anwendung enthält, wenn Benutzer die Möglichkeit haben sollen, ein Bedienelement zu installieren, das sich anders verhält als das Aktionssymbol. Wenn, beispielsweise, das Bedienelement eine Datei überwacht und sein Erscheinungsbild ändern soll, wenn die überwachte Datei sich ändert.

Bedienfeldkonfigurationsdateien werden in das Verzeichnis `app_root/dt/appconfig/types/sprache` gestellt. Die Benennungskonvention lautet `name.fp`.

Wird eine Konfigurationsdatei bereitgestellt, die ein Bedienelement enthält, kann der Benutzer das Bedienelement einer Bedientafel hinzufügen, indem er die Datei `*.fp` dem Bedienelement `'Install Icon'` in der Bedientafel übergibt.

Beispielsweise kann die folgende Definition in eine Bedienfeldkonfigurationsdatei in der Gruppe von Anwendungen gestellt werden. Übergibt der Benutzer diese Datei einem Bedienelement `'Install Icon'` in einer Bedientafel, wird in der Bedientafel ein Bedienelement erstellt, das die Anwendung `'BesterTextEditor'` ein einziges Mal ausführt. Wird `'BesterTextEditor'` bereits ausgeführt, wird das Fenster auf dem Fensterstapel im aktuellen Arbeitsbereich zuoberst angezeigt.

```
CONTROL BesterTextEditorControl
{
    TYPE            icon
    ICON            BTEFPanel
    PUSH_RECALL    True
    CLIENT_NAME    BTEd
    PUSH_ACTION    BTEditor
    DROP_ACTION    BTEditor
    HELP_STRING    Startet die Anwendung 'BesterTextEditor'.
}

```

Weitere Informationen zum Erstellen von Bedienfeldkonfigurationsdateien befinden sich in:

- Kapitel 13, "0Erweiterte Anpassung des Bedienfelds"

- Hilfeseite `dtfpfile(4)`

Schritt 8: Anwendungen mit `dtappintegrate` registrieren

Hinweis – Ein Beispiel für die Registrierung einer Anwendung kann Schritt 8 des Abschnitts “Beispiel für die Erstellung eines Registrierungs pakets” auf Seite 92 entnommen werden.

Sobald ein Registrierungs paket unter einem Anwendungsstammverzeichnis erstellt wurde, kann die tatsächliche Anwendungsregistrierung durchgeführt werden.

Die Anwendungsregistrierung stellt Verbindungen zwischen dem Registrierungs paket und den Verzeichnissen im Desktop-Suchpfad her. (Weitere Informationen können dem Abschnitt “Wie `dtappintegrate` Anwendungen integriert” auf Seite 84 entnommen werden.)

▼ Anwendungen mit `dtappintegrate` registrieren

Ist die Anwendung Desktop-fähig, wird `dtappintegrate` normalerweise als letzter Schritt des Installationsprozesses automatisch ausgeführt. Wird `dtappintegrate` nicht automatisch ausgeführt, oder wurde bei der Erstellung der Konfigurationsdateien eine nicht Desktop-fähige Anwendung integriert, kann `dtappintegrate` auch manuell ausgeführt werden.

1. Als Benutzer ‘root’ anmelden.

2. Folgenden Befehl ausführen:

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

wobei *app_root* das Desktop-Anwendungsstammverzeichnis ist. Weitere Informationen können der Hilfeseite `dtappintegrate(1)` entnommen werden.

3. Die Gruppe von Anwendungen Desktop-Werkzeuge Desktop-Werkzeuge öffnen und ‘Anwendungen erneut laden’ doppelklicken.

4. Überprüfen, ob die Anwendung richtig registriert ist:

- a. Die oberste Ebene des Anwendungsmanagers anzeigen. Die neue Gruppe von Anwendungen sollte nun im Anwendungsmanager erscheinen.
- b. Die Gruppe von Anwendungen öffnen und das Aktionssymbol doppelklicken.

Syntax und Optionen bei dtappintegrate

```
dtappintegrate -s app_root [-t ziel_pfad] [-l sprache] [-u]
```

-s <i>app_root</i>	Erforderlicher Parameter. Gibt das Anwendungsstammverzeichnis an, unter dem die Anwendung installiert wurde.
-t <i>ziel_pfad</i>	Wahlfreier Parameter. Standardmäßig wird die Position <code>/etc/dt/appconfig</code> im System verwendet. Gibt die Position an, mit der die Desktop-Konfigurationsdateien verbunden werden. Es muß eine Position im Anwendungssuchpfad verwendet werden.
-l <i>sprache</i>	Wahlfreier Parameter. Standardmäßig ist jede Sprache möglich. Gibt an, welche sprachabhängigen Desktop-Konfigurationsdateien integriert werden sollen.
-u	Wahlfreier Parameter. Macht die Integrierung der Anwendung rückgängig und entfernt alle Verbindungen, die während der Integrierung gesetzt wurden.

Wie dtappintegrate Anwendungen integriert

Die Funktion von dtappintegrate besteht darin, Verbindungen zwischen den installierten Dateien und den Positionen, wo das Desktop die Konfigurationsdateien sucht, herzustellen.

Aktionen und Datentypen

dtappintegrate erstellt symbolische Verbindungen von den Definitionsdateien für Aktionen und Datentypen im Registrierungspaket zu dem für das System definierten Verzeichnis im Suchpfad der Aktionsdatenbankhilfe. Hierzu werden Verbindungen von

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

zu

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

hergestellt.

Hilfeinformationsdateien

`dtappintegrate` erstellt symbolische Verbindungen von den Hilfe Dateien im Registrierungspaket zu dem für das System definierten Verzeichnis im Hilfesuchpfad. Hierzu werden Verbindungen von

```
app_root/dt/appconfig/help/sprache/help_file.sdl
```

zu

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

hergestellt.

Symboldateien

`dtappintegrate` erstellt symbolische Verbindungen von den Symboldateien im dem Registrierungspaket zu dem für das System definierten Verzeichnis im Symbolsuchpfad. Hierzu werden Verbindungen von

```
app_root/dt/appconfig/icons/sprache/icon_files
```

zu

```
/etc/dt/appconfig/icons/sprache/icon_files
```

hergestellt.

Gruppe von Anwendungen

Um die Gruppe von Anwendungen für die Anwendung in die oberste Ebene des Anwendungsmanagers zu stellen, erstellt `dtappintegrate` eine Verbindung zwischen dem Verzeichnis für die Gruppe von Anwendungen im Registrierungspaket und der für das System definierten Position im Anwendungssuchpfad. Hierzu werden Verbindungen vom Verzeichnis

```
app_root/dt/appconfig/appmanager/sprache/anwendungsgruppe_name
```

zu

```
/etc/dt/appconfig/appmanager/sprache/anwendungsgruppe_name
```

hergestellt.

Beispiel für die Erstellung eines Registrierungs pakets

Mit den folgenden Schritten wird ein Registrierungs paket für eine bereits vorhandene, nicht Desktop-fähige Anwendung mit dem Namen "BesterTextEditor" erstellt.

Notwendige Informationen zu 'BesterTextEditor'

In diesem Beispiel wird für die Anwendung 'BesterTextEditor' folgendes vorausgesetzt:

- sie wurde im Verzeichnis `/usr/BTE` installiert.
- Die Session-Sprache des Benutzers ist der Standardwert `C`.
- 'BesterTextEditor' wird mit folgender Befehlszeile gestartet:

```
BTEd [ dateiname ]
```

wobei *dateiname* der Name der Datendatei ist, die in dem neuen Fenster geöffnet werden soll. 'BesterTextEditor' erstellt sein eigenes Fenster, d.h. diese Anwendung wird nicht innerhalb eines emulierten Terminalfenster ausgeführt.

- 'BesterTextEditor' erstellt und verwendet zwei Arten von Datendateien:
 - Dokumentationsdateien. Hierfür wird die Benennungskonvention `*.bte` verwendet. 'BesterTextEditor' stellt eine Befehlszeile für den Druck der `.bte`-Datendateien.zur Verfügung. Die Syntax dieses Befehls lautet:

```
BTEPrint [-d zieldrucker] [-s] dateiname
```

wobei

`-d zieldrucker` den Zieldrucker angibt.

`-s` angibt, daß sofort gedruckt wird und das Dialogfenster Drucken der Anwendung nicht angezeigt wird.

dateiname die zu druckende Datei angibt.

- Schablonendateien. Hierfür wird die Benennungskonvention `*.tpl` verwendet. Schablonendateien können nicht gedruckt werden.
- Die bereits vorhandenen, nicht Desktop-fähigen Dateien `app-defaults` für 'BesterTextEditor' enthalten Ressourcen für Schnittstellschriftarten und Vordergrund- und Hintergrundfarben.

- Ein Online-Hilfebund für 'BesterTextEditor' wurde mit Hilfe des Help Developer's Kit des Desktop erstellt. Für die Erstellung des Online-Hilfebands wurden folgende Quellendateien verwendet:

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

und die Datei .../BTEHelp.sdl erstellt.

Schritte beim Registrieren von 'BesterTextEditor'

Mit der folgenden schrittweisen Prozedur wird 'BesterTextEditor' registriert:

1. Schriftarten- und Farbressourcen ändern

In der Datei app-defaults des 'BesterTextEditor' die Ressourcen, die

- Schriftarten für Text und
- Farben für Vorder- und Hintergrund

festlegen, entfernen.

2. Anwendungsstammverzeichnis erstellen

Folgendes Verzeichnis erstellen:

```
/desktop_approots/BTE
```

Wird eine bereits vorhandene Anwendung integriert, sollte das Anwendungsstammverzeichnis an einer anderen Stelle erstellt werden, als der Position, an der die Installation erfolgte, da sonst die erstellten Konfigurationsdateien bei einer möglichen Aktualisierung der Anwendung entfernt werden.

3. Registrierungspaketverzeichnisse erstellen

Folgende Verzeichnisse erstellen::

```
/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/BesterTextEditor
```

4. Aktionen und Datentypen für die Anwendung erstellen

- a. Die Konfigurationsdatei für die Definitionen für Aktionen und Datentypen erstellen:

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

- b. Die Aktionsdefinition für die Ausführung von 'BesterTextEditor' erstellen:

```
ACTION BTEditor
{
    WINDOW_TYPE    NO_STDIO
    ICON           BTERun
    DESCRIPTION    Dieses Symbol doppelklicken oder BTE-Daten-
                  \
                  datei übergeben, um BTE auszuführen.
    EXEC_STRING    /usr/BTE/BTEd %Arg_1%
}

```

- c. Den Datentyp für Dateien im Format *.bte erstellen:

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

```

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

```

- d. Den Datentyp für Dateien im Format *.tpl erstellen:

```
DATA_ATTRIBUTES BTETemplateFile
{
    DESCRIPTION    BesterTextEditor template file.
    ICON           BTETempl
}

```

```

        ACTIONS          Open
    }

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

```

e. Die Aktion 'Öffnen' für Dateien im Format *.bte erstellen:

```

ACTION Open
{
    ARG_TYPE      BTEDataFile
    TYPE          MAP
    MAP_ACTION    BTEditor
}

```

f. Die Aktion 'Drucken' für Dateien im Format *.bte erstellen:

Im folgenden handelt es sich um einfache Druckaktionen, mit denen die Datendateien gedruckt werden. Bei diesen Aktionen ist ein Wert für die Umgebungsvariable LPDEST erforderlich und die Druckoption `-s` wird ignoriert. (Ist LPDEST nicht gesetzt, schlägt die Aktion möglicherweise fehl.)

```

ACTION Print
{
    ARG_TYPE      BTEDataFile
    TYPE          MAP
    MAP_ACTION    BTEPrintData
}

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

```

Es folgt eine weitere Version der Aktion 'BTEPrintData' und einer Begleitprozedur. Diese werden eingesetzt, wenn LPDEST nicht gesetzt ist und das Dialogfenster 'Drucken' nicht angezeigt werden soll.

```

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

```

Die Prozedur `/usr/BTE/bin/BTEenvprint` hat folgenden Inhalt:

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

g. Die Aktion 'Öffnen' für Dateien im Format `*.tpl` erstellen:

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

h. Die Aktion 'Drucken' für Dateien im Format `*.tpl` erstellen:

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

Bei 'NoPrint' handelt es sich um eine integrierte Aktion, die ein Dialogfenster anzeigt, in dem dem Benutzer mitgeteilt wird, daß die Datei nicht gedruckt werden kann.

5. Hilfedateien in das Registrierungspaket stellen

a. Die Hilfedateien in folgende Positionen stellen:

```
/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. Folgende Datei erstellen:

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

Die folgende Aktionsdefinition in die Datei stellen

```
ACTION BTEHelp
{
    WINDOW_TYPE NO_STDIO
    EXEC_STRING /usr/dt/bin/dthelpview -helpVolume \
```

```

        BTEHelp.sdl
    DESCRIPTION Öffnet den BesterTextEditor Hilfeband.
}

```

6. Symbole für die Anwendung erstellen

Die Symbole mit dem Symboleditor erstellen. Folgende Größenrichtlinien beachten:

Name	Größe
<i>Dateiname ohne Pfad</i> .t.pm	16 by 16
<i>Dateiname ohne Pfad</i> .m.pm	32 by 32
<i>Dateiname ohne Pfad</i> .l.pm	64 by 64

Diese Symboldateien in folgendem Verzeichnis erstellen:

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

- Symbole für die Darstellung der Aktion, die die Anwendung ausführt: BTERun.t.pm, BTERun.m.pm, BTERun.l.pm
- Symbole, die Dateien im Format *.bte darstellen: BTEData.t.pm, BTEData.m.pm,
- Symbole, die Dateien im Format *.tpl darstellen: BTETempl.t.pm, BTETempl.m.pm
- Symbole, die die Gruppe von Anwendungen (siehe Schritt 7) darstellen: BTEApp.t.pm, BTEApp.m.pm

7. Gruppe von Anwendungen erstellen:

- a. Falls nicht bereits geschehen, folgendes Verzeichnis erstellen:

```

/desktop_approots/BTE/dt/appconfig/appmanager/C/BesterTextEditor

```

- b. Dieser Schritt ist wahlfrei. Hierdurch wird ein eindeutiges Symbol für die Gruppe von Anwendungen zur Verfügung gestellt, indem ein Datentyp und zugehörige Aktionen für die Gruppe von Anwendungen erstellt werden. Wird dieser Schritt nicht ausgeführt, verwendet die Gruppe von Anwendungen das Standardsymbol.

Die folgenden Definitionen für Datentyp und Aktionen der Datei

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

hinzufügen. Der Datentyp gibt an, welches Symbol die Gruppe von

Anwendungen 'BesterTextEditor' verwendet. Die Aktionen verhalten sich beim Öffnen und Drucken genauso wie die integrierten Gruppen von Anwendungen.

```
DATA_ATTRIBUTES BesterTextEditorAppGroup
{
    ACTIONS      OpenInPlace,OpenNewView
    ICON         BTEApp
}
DATA_CRITERIA BesterTextEditorAppGroupCriteria
{
    DATA_ATTRIBUTES_NAME BesterTextEditorAppGroup
    MODE                 d
    PATH_PATTERN         */appmanager/*/BesterTextEditor
}
ACTION Open
{
    ARG_TYPE      BesterTextEditorAppGroup
    TYPE          MAP
    MAP_ACTION    OpenAppGroup
}
ACTION Print
{
    ARG_TYPE      BesterTextEditorAppGroup
    TYPE          MAP
    MAP_ACTION    PrintAppGroup
}
```

- c. Ein Symbol in der Gruppe von Anwendungen erstellen, das die Anwendung startet. Hierzu folgende Datei erstellen:

```
/desktop_approots/BTE/dt/appconfig/appmanager/C/BesterTextEditor/BTEditor
```

und die Datei ausführbar machen.

- d. Die Aktionsdatei in der Gruppe von Anwendungen erstellen, die den Hilfeband erstellt. Hierzu folgende Datei erstellen:

```
/desktop_approots/BTE/dt/appconfig/appmanager/C/BesterTextEditor/BTEHelp
```

und die Datei ausführbar machen.

- e. Weitere Dateien in die Gruppe von Anwendungen stellen, z.B. Readme-Dateien, Beispieldaten- und Schablonendateien.

8. Anwendung registrieren

In einem emulierten Terminalfenster

-
- a. als Benutzer 'root' anmelden.
 - b. Folgenden Befehl ausführen:

```
/usr/dt/bin/dtappintegrate -s /desktop_approots/BTE
```
 - c. Die Gruppe von Anwendungen 'Desktop-Werkzeuge' öffnen und auf 'Anwendungen erneut laden' doppelklicken.

Desktop in einem Netzwerk konfigurieren

Das Desktop ist speziell für den Betrieb in einer stark vernetzten Umgebung ausgerichtet. Die Architektur des Desktops ermöglicht Systemadministratoren Computerressourcen im gesamten Netzwerk zu verteilen. Zu diesen Ressourcen gehören:

- Anwendungen.
- Datendateien für Anwendungen.
- Desktop-Session-Services (Desktop-Anwendungen wie zum Beispiel Anmeldemanager und Dateimanager).
- Hilfe-Services. Hilfedatendateien können auf einen zentralen Hilfe-Server gestellt werden.

<i>Übersicht über das Desktop im Netzwerk</i>	<i>96</i>
<i>Allgemeine Schritte für die Konfiguration eines Desktops im Netzwerk</i>	<i>100</i>
<i>Netzwerk des Basisbetriebssystems für das Desktop konfigurieren</i>	<i>100</i>
<i>Clients und Server für das Desktop konfigurieren</i>	<i>104</i>
<i>Anwendungs-Services verwalten</i>	<i>111</i>

Übersicht über das Desktop im Netzwerk

Das Betriebssystem stellt eine Reihe von Services im Netzwerk, einschließlich verteilter Dateisysteme und der Ausführung auf fernen Systemen zur Verfügung. X-Server stellen zusätzliche Netzwerkfähigkeiten, wie den Zugriff auf ferne Bildschirme und Sicherheitsservices zur Verfügung.

Das Desktop legt eine Benutzerschnittstelle auf diese Netzwerkfunktionen. Ziel dieser Schnittstelle und der ihr zugrundeliegenden Architektur ist

- die einfachere Handhabung von Netzwerksystemen. Benutzer können Anwendungen ausführen und auf Datendateien zugreifen, ohne sich damit beschäftigen zu müssen, wo sich im Netzwerk die Anwendungen und Daten befinden.
- die einfachere Verwaltung von Netzwerksystemen. Das Desktop stellt Werkzeuge zur Anwendungsintegration sowie Netzwerksuchpfade zur Verfügung, die es Systemen leichter machen, ferne Daten und Anwendungen zu finden. Zusätzlich erleichtert der Dateinamenzuordnungsprozeß des Desktops die Verwaltung komplexer Netzwerke mit mehreren Servern.
- Flexibilität. Die Verwaltungsfunktionen des Desktops wurden zwar für bestimmte häufig vorkommende Netzwerksituationen entwickelt, aber das Desktop kann für viele andere angepaßte Netzwerkkonfigurationen eingerichtet werden.

Typen von Desktop Services im Netzwerk

Bei der Arbeit im Netzwerk kann ein Benutzer, der an einem bestimmten Bildschirm sitzt, auf verschiedene Computer-Services, die auf anderen Systemen verteilt sind, zugreifen. Dazu gehören:

- die Desktop-Session und ihre Anwendungen— zum Beispiel Arbeitsbereichsmanager und Dateimanager
- andere Anwendungen
- Datendateien

In der Netzwerkterminologie wird der Begriff *Server* verwendet, um ein System zu beschreiben, das einem oder mehreren Systemen Computerservices zur Verfügung stellt. Empfängt ein System Services von einem Server, wird es als ein *Client* dieses Servers bezeichnet.

In einem komplexen Netzwerk verwendet ein System möglicherweise Services, die sich auf einer Reihe von verschiedenen Systemen im gesamten Netzwerk befinden. Überdies kann ein System als ein bestimmter Servertyp fungieren (zum Beispiel als Session-Server) und gleichzeitig ein Client (zum Beispiel eines Anwendungs-Servers) sein.

Typische Netzwerksituationen

Aus der Perspektive des Desktops enthält eine typische Netzwerkkonfiguration möglicherweise eine Kombination aus den folgenden wichtigen Komponenten:

Bildschirme	an denen der X-Server ausgeführt wird
Anmelde-/Session-Server	an denen die Desktop-Anwendungen (Anmeldemanager, Arbeitsbereichsmanager, usw.) ausgeführt werden
Anwendungs-Server	an denen andere Anwendungen ausgeführt werden
Datei-Server	in dem Anwendungen verwendete Daten gespeichert werden

Zu den am häufigsten vorkommenden Netzwerkkonfigurationen gehören Systeme, die auf einen Anwendungs-Server zugreifen. Abbildung 5-1 zeigt eine Datenstation, die einen Anwendungs-Server verwendet. Der X-Server und die Desktop-Session werden auf der Datenstation ausgeführt.



Abbildung 5-1 Anwendungs-Server stellen Services für die Desktop-Sessions zur Verfügung

Netzwerke verwenden Datei-Server häufig, um große Datenmengen zu steuern. Diese Daten werden von Anwendungen verwendet, die auf einem Anwendungs-Server ausgeführt werden oder von den Desktop-Anwendungen (zum Beispiel benötigt der Dateimanager Zugriff auf Datendateien, um diese im Dateimanagerfenster anzuzeigen).

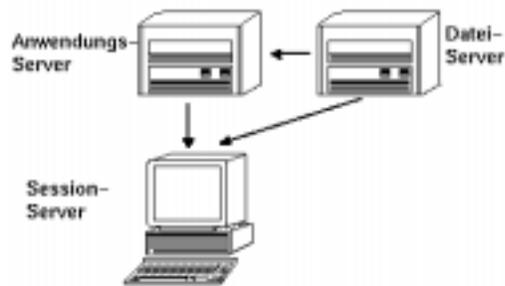


Abbildung 5-2 Datei-Server stellen Daten für Anwendungs-Server und Session-Server zur Verfügung

X-Stations führen den X-Server aus und erhalten Desktop-Session-Services von einem anderen System.

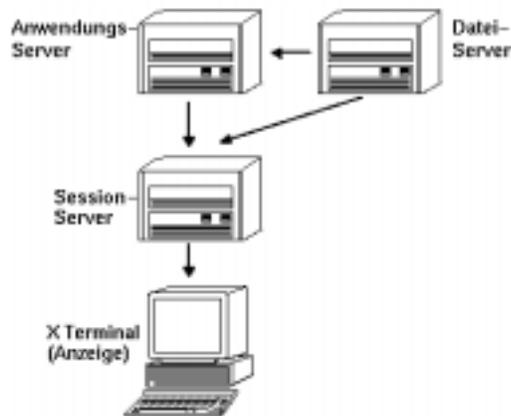


Abbildung 5-3 X-Stationen erhalten Session-Services von einem Session-Server

Weitere Netzwerksituationen

Das Desktop ist flexibel und kann komplexere Netzwerkkonfigurationen unterstützen. Dazu gehört auch, daß verschiedene Services, zusätzlich zu Datei-Servern, für Anwendungs-Server verfügbar gemacht werden.

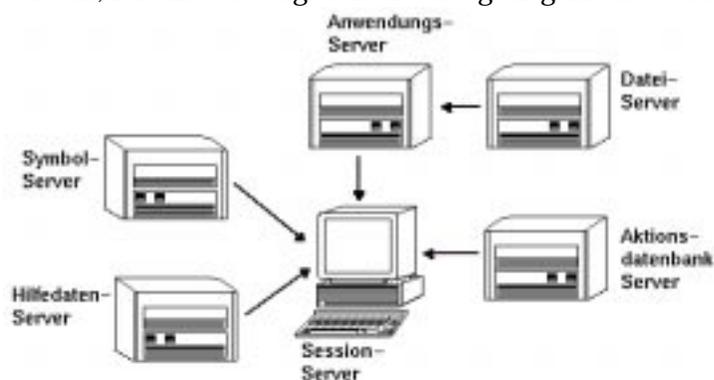


Abbildung 5-4 Services, die für einen Desktop-Anwendungs-Server erforderlich sind, können verteilt sein

Zusammenfassung—Servertypen

Bildschirm	Das System, das den X-Server ausführt.
Anmelde- und Session-Server	Das System, das die Desktop-Session (Anmeldemanager, Session-Manager, Fenstersteuerung, Dateimanager, usw.) ausführt
Anwendungs-Server	Ein System, auf dem eine Anwendung ausgeführt wird. Wird auch als <i>ausführender Rechner</i> bezeichnet.
Datei-Server	Ein System, auf dem Datendateien für Anwendungen gespeichert sind
Hilfe-Server	Ein System, auf dem Hilfedatendateien gespeichert werden
(Aktions-)Datenbank-Server	Ein System, auf dem Dateien mit Definitionen für Aktionen und Datentypen gespeichert werden

Symbol-Server

Ein System, auf dem Symboldateien gespeichert werden

Das Netzwerk enthält möglicherweise noch zusätzliche Server, wie beispielsweise einen Kennwort-Server, einen Post-Server, einen Video-Server, usw.

Allgemeine Schritte für die Konfiguration eines Desktops im Netzwerk

Für die Konfiguration eines Desktops im Netzwerk sind drei grundlegende Schritte erforderlich:

1. Die Netzwerk-Services des Basisbetriebssystems konfigurieren.

Hierbei handelt es sich um die Netzwerk-Services des Betriebssystems, auf die das Desktop angewiesen ist. Weitere Informationen hierzu können dem Abschnitt "Netzwerk des Basisbetriebssystems für das Desktop konfigurieren" auf Seite 100 entnommen werden.

2. Netzwerk-Software und -Services für das Desktop installieren und konfigurieren.

Hierbei handelt es sich um Services, die für das Desktop, unabhängig vom Typ des Clients oder Server-Systems, das eingerichtet wird, erforderlich sind. Weitere Informationen hierzu können dem Abschnitt "Clients und Server für das Desktop konfigurieren" auf Seite 104 entnommen werden.

3. Den genauen Typ von Server oder Client konfigurieren.

Beispielweise erfordert die Konfiguration eines Anwendungs-Servers andere Schritte als die Konfiguration eines Datei-Servers. Weitere Informationen hierzu können dem Abschnitt "Anwendungs-Services verwalten" auf Seite 111 entnommen werden.

Netzwerk des Basisbetriebssystems für das Desktop konfigurieren

Für das Desktop ist folgende Basisnetzwerkconfiguration erforderlich:

- Benutzer müssen über ein Anmeldeabrechnungskonto am Session-Server und auf jedem System, das Desktop-Services für den Session-Server zur Verfügung stellt, verfügen. Dem Benutzer muß an allen Client- und Server-Systemen dieselbe Benutzer- und Gruppen-ID zugeordnet sein.

- Systeme müssen auf ferne Dateisysteme mit Daten, die die Session und andere Anwendungen verwenden, Zugriff haben.
- Der Druck-Spooler `lp` muß für den Zugriff auf ferne Drucker konfiguriert sein.
- `sendmail` muß für E-mail-Services konfiguriert sein.
- X-Berechtigung muß eingerichtet sein.

Anmeldeabrechnungen für Benutzer einrichten

Dieser Abschnitt beschreibt, welche Anmeldeabrechnungen für das Desktop im Netzwerk erforderlich sind.

Anmeldeabrechnungen einrichten

Benutzer müssen ein Anmeldeabrechnungskonto für folgende Server und Systeme haben:

- alle Systeme, die Services für das Desktop zur Verfügung stellen, einschließlich Anwendungs-Server, Datei-Server und Systeme, die Drucker im Netzwerk zur Verfügung stellen.
- alle Session-Server, auf die der Benutzer zugreift. Session-Server werden normalerweise mit Xstations verwendet.

Konsistente Benutzer- und Gruppen-IDs

UNIX-Benutzer werden durch einen Anmeldenamen und eine numerische Benutzer-ID (UID) identifiziert. In einem Desktop-Netzwerk sollten Anmelde-name und UID des Benutzers an allen Client- und Server-Systemen dieselbe sein.

UNIX-Benutzer werden auch einer oder mehreren Anmeldegruppen zugeordnet. Jede Gruppe verfügt über einen Gruppennamen und eine numerische Gruppen-ID (GID). In einem Desktop-Netzwerk sollten alle Systeme konsistente Gruppennamen und Gruppen-IDs verwenden.

Weitere Informationen können den Hilfeseiten `id(1)` or `id(1M)` entnommen werden.

Zugriff auf verteilte Dateisysteme konfigurieren

Das Desktop verwendet NFS[®], damit Dateien von verschiedenen Systemen gemeinsam benutzt werden können. Alle Dateisysteme im Netzwerk, die gemeinsam benutzte Dateien enthalten, müssen identifiziert werden, und es muß sichergestellt werden, daß sie korrekt an allen betreffenden Systemen angehängt sind.

In der Regel muß der folgende Zugriff auf ferne Dateien zur Verfügung gestellt werden:

- Das Home-Verzeichnis des Benutzers muß von allen Client- und Server-Systemen des Desktop gemeinsam benutzt werden. Dies ist notwendig, weil:
 - das Home-Verzeichnis Datendateien enthält, auf die von Anwendungen auf fernen Systemen zugegriffen wird. Beispielsweise verwenden Anwendungen, die mit Datendateien arbeiten, häufig das Home-Verzeichnis als die Standarddateiposition.
 - Das Home-Verzeichnis ist das Standardidentifikationsverzeichnis `dtspcd`. Weitere Informationen zu `dtspcd` können dem Abschnitt "Den Steuerungsdaemonprozeß für Unterprozesse konfigurieren" auf Seite 109 entnommen werden.
- Möchten Benutzer auf Datendateien zugreifen, die sich nicht in ihrem Home-Verzeichnis befinden, müssen diese Datendateien von allen Client- und Server-Systemen des Desktop, die mit den Datendateien arbeiten, gemeinsam benutzt werden.
- Die Desktop-Installations- und Konfigurationsverzeichnisse (`/usr/dt` und `/etc/dt`) müssen von allen Client- und Server-Systemen des Desktop gemeinsam benutzt werden, so daß alle Anwendungen des Benutzers auf dieselben Desktop-Konfigurationsdateien zugreifen.

Home-Verzeichnis im Netzwerk einrichten

Ein Desktop-Netzwerk arbeitet am effektivsten, wenn Benutzer über ein einziges Home-Verzeichnis verfügen, das alle Client- und Server-Systeme im Netzwerk gemeinsam benutzen.

Ein Home-Verzeichnis im Netzwerk erlaubt einem Benutzer, verschiedene Systeme im Netzwerk zu verwenden, ohne daß die von ihm durchgeführten Anpassungen und Konfigurationen verlorengehen. Dies kommt daher, weil die

Anpassungen des Benutzers und Informationen, die für die Wiederherstellung einer vorherigen Session notwendig sind, in Unterverzeichnissen des Home-Verzeichnisses gespeichert sind.

Ein gemeinsames Home-Verzeichnis ist ebenfalls erforderlich für:

- den standardmäßigen X-Identifikationsmechanismus. Weitere Informationen hierzu können dem Abschnitt “X-Berechtigung konfigurieren” auf Seite 104 entnommen werden.
- den Desktop-Steuerungsdaemonprozeß für Unterprozesse, der beim Start ferner Anwendungen beteiligt ist, und in das Home-Verzeichnis des Benutzers schreiben können muß.

Dateinamenkonsistenz

Das Netzwerk sollte so konfiguriert sein, daß Benutzer von allen Systemen aus mit demselben Namen auf ihre Datendateien zugreifen können. Dies wird *Dateinamenkonsistenz* genannt und wird normalerweise erreicht, indem die entsprechenden symbolischen Verbindungen erstellt werden. Beispielsweise kann jedes System so konfiguriert werden, daß das Home-Verzeichnis jedes Benutzers als `/users/anmeldung_name` zur Verfügung steht, indem eine symbolische Verbindung zur tatsächlichen Mount-Position des Verzeichnisses hergestellt wird.

Zugriff auf ferne Drucker konfigurieren

Das Desktop verwendet den Druck-Spooler `lp` für den Zugriff auf lokale oder ferne Drucker. Weitere Informationen zur Konfiguration des Druck-Spoolers `lp` kann der Hilfeseite `lpadmin(1M)` entnommen werden.

Bevor der Benutzer versucht, über die grafische Schnittstelle des Desktops zu drucken, sollte er testen, ob mit dem Befehl `lp` an allen Druckern problemlos gedruckt werden kann.

Es wird unbedingt empfohlen, konsistente Druckereinheitennamen zu verwenden. Ist, beispielsweise, ein bestimmter Drucker dem System, mit dem er direkt verbunden ist, als `Postscript1` bekannt, sollten alle weiteren Systeme, die auf den Drucker fern zugreifen, den Namen `Postscript1` verwenden.

Elektronische Post konfigurieren

Das Desktop-Postprogramm verwendet `sendmail`, um Post zwischen Systemen zu versenden. Weitere Informationen zur Konfigurierung von E-mail-Konnektivität können der Hilfeseite `sendmail(1M)` entnommen werden.

Bevor der Benutzer versucht, Post über das Desktop zu senden oder zu empfangen, sollte er testen, ob er Post problemlos mit dem Befehl `mailx` senden und empfangen kann.

X-Berechtigung konfigurieren

Das Desktop verwendet den standardmäßigen X-Mechanismus, um ferne Anwendungen (X-Clients) zum Zugriff auf einen lokalen Bildschirm zu berechtigen. Der einfachste Weg, dies zu erreichen, ist, ein Home-Verzeichnis im Netzwerk für jeden Benutzer zur Verfügung zu stellen. Hierdurch wird sichergestellt, daß folgende Erfordernisse erfüllt sind:

- Der Benutzer muß über Schreib- und Leseberechtigung für die Datei *Home-Verzeichnis*/`.xauthority` verfügen.
- Die Datei `.xauthority` auf einem Anwendungs-Server muß den Eintrag für den Bildschirm enthalten, auf dem die Anwendung ausgeführt wird.

Weitere Informationen können den Hilfeseiten `x(1)` oder `xauth(1)` entnommen werden.

Clients und Server für das Desktop konfigurieren

In diesem Abschnitt werden Desktop-spezifische Erfordernisse der Netzwerkkonfiguration behandelt. Das heißt, diese Funktionen werden durch das Desktop und nicht durch das Basisbetriebssystem zur Verfügung gestellt.

Der Abschnitt besteht aus zwei Teilen:

- der Konfiguration von Anmelde- und Session-Services.
- der Konfiguration von Services, die für Anwendungen und deren Daten erforderlich sind. Hierzu gehören: Anwendungs-, Datenbank-, Symbol-, Datei- und Hilfe-Server und deren Clients.

Anmelde- und Session-Services konfigurieren

Bei einem Anmelde-/Session-Server handelt es sich um ein System, das Desktop-Services (Anmeldemanager, Session-Manager, Dateimanager, Fenstersteuerung, usw.) einem Bildschirm und X-Server zur Verfügung stellt.

In der Regel stellt ein Session-Server XStations Services zur Verfügung. Aber es kann auch eine Netzwerkkonfiguration eingerichtet werden, die Session-Services auf einem oder mehreren Servern konzentriert, auf die sowohl von XStations als auch von Datenstationen zugegriffen wird.

Beim Anmeldemanager handelt es sich um die Desktop-Komponente, die dafür zuständig ist, daß Anmelde-Services anderen Bildschirmen zur Verfügung gestellt werden. Sobald der Benutzer sich angemeldet hat, wird der Session-Manager für den Benutzer gestartet.

Weitere Informationen zur Konfiguration von Anmelde-/und Session-Servern, sowie XStations können dem Abschnitt "Anmeldefenster an einem Bildschirm im Netz anzeigen" auf Seite 7 entnommen werden.

Weitere anwendungsbezogene Services konfigurieren

Dieser Abschnitt behandelt Erfordernisse des Desktops im Netzwerk für folgende Server:

- Anwendungs-Server
- Datenbank-Server
- Symbol-Server
- Hilfe-Server

▼ *Clients und Server für das Desktop konfigurieren*

1. Die notwendigen Netzwerkkonfigurationen für das Betriebssystem zur Verfügung stellen, die für das Desktop erforderlich sind.

Weitere Informationen hierzu können dem Abschnitt "Netzwerk des Basisbetriebssystems für das Desktop konfigurieren" auf Seite 100 entnommen werden.

2. Das Desktop oder die Mindestdateigruppe installieren :

Es müssen installiert werden:

- die gesamten Dateigruppen von 'Common Desktop Environment runtime'
- *oder*, folgende Dateigruppen: CDE-MIN und CDE-TT

Hinweis – Installation und Dateigruppen können bei verschiedenen Lieferanten unterschiedlich zusammengesetzt sein.

3. Das System für den Dämon des Servers für die ToolTalk-Dateinamendatenbank `rpc.ttdbserver` konfigurieren.

Dies sollte bei der Installation des Desktops automatisch geschehen. Weitere Informationen können dem Abschnitt "ToolTalk-Datenbank-Server konfigurieren" auf Seite 110 entnommen werden.

4. Den Steuerungsdämonprozeß für Unterprozesse (`dtspcd`) installieren und konfigurieren.

Dies sollte bei der Installation des Desktops automatisch geschehen. Weitere Informationen können dem Abschnitt "Den Steuerungsdämonprozeß für Unterprozesse konfigurieren" auf Seite 109 entnommen werden

5. Alle erforderlichen fernen Daten anhängen.

Daten werden als "fern" bezeichnet, wenn sie sich auf einem anderen System als dem System befinden, auf dem die Anwendung, die die Daten verwendet, ausgeführt wird.

Beispiel:

- Verwendet eine Anwendung Daten auf einem Datei-Server, müssen diese Dateien angehängt werden.
- Befinden sich Dateimanagersymbole auf einem Symbol-Server, muß der Session-Server diese Dateien anhängen.
- Verwendet das Netzwerk einen Hilfe-Server für Desktop-Hilfedateien, müssen der Session-Server und alle Anwendungs-Server die Hilfedaten anhängen.

Weitere Informationen zu Einhängepunkten können dem nächsten Abschnitt, "Den Einhängepunkt für ferne Dateisysteme konfigurieren" entnommen werden.

Den Einhängpunkt für ferne Dateisysteme konfigurieren

Wenn das Desktop Dateinamen von einem System zum anderen übermittelt, müssen diese Dateinamen in Namen umgesetzt, bzw. *zugeordnet* werden, die für das Zielsystem verständlich sind. Diese Zuordnung ist notwendig, weil eine Datei möglicherweise an den verschiedenen Systemen an verschiedenen Positionen eingehängt ist, und deshalb verschiedene Namen verwendet werden müssen, um auf die Datei zugreifen zu können. Beispielsweise wird auf die Datei `/projects/big` auf `sysA` von `sysB` aus mit dem Namen `/net/sysA/projects/big` zugegriffen.

Erfordernisse für die Dateinamenzuordnung

Um die Dateinamenzuordnung problemlos durchführen zu können, muß einer der folgenden Punkte zutreffen:

- Der Befehl `mount` wird verwendet, um Dateisysteme statisch einzuhängen. Diese Art des statischen Einhängens wird in der Regel in einer Datei wie beispielsweise `/etc/checklist`, `/etc/mnttab` oder `/etc/filesystems` verwendet.

Damit die Dateinamenzuordnung zwischen Systemen problemlos funktioniert, müssen beim Einhängen von Dateisystemen konsistente Host-Namen verwendet werden. Verfügt ein Host über mehrere Namen, (zum Beispiel Alias-Namen, oder der Host hat mehr als eine LAN-Adresse mit verschiedenen Namen), muß bei allen Einhängvorgängen derselbe Name und dasselbe Format verwendet werden.

- *oder*, der Automounter wird verwendet, um Dateisysteme am Standardeinhängpunkt `/net` einzuhängen.
- *oder*, der Automounter wird verwendet, um Dateisysteme an einer anderen Stelle als `/net` einzuhängen, und die Umgebungsvariable `DTMOUNTPOINT` wird gesetzt, um den Einhängpunkt anzugeben. Weitere Informationen hierzu können dem nächsten Abschnitt, "Einen Wert für `DTMOUNTPOINT` setzen" entnommen werden.

Weitere Informationen zum automounter können der Hilfeseite `automount(1M)` entnommen werden.

Einen Wert für DTMOUNTPOINT setzen

Treffen die beiden folgenden Bedingungen zu, muß die Umgebungsvariable `DTMOUNTPOINT` gesetzt werden:

- Der Automounter wird verwendet, um Dateisysteme einzuhängen
- *und*, ferne Dateisysteme werden an einer anderen Position als `/net` eingehängt.

Zu den Prozessen, für die DTMOUNTPOINT gesetzt werden muß, gehören:

- die Desktop-Prozesse des Benutzers, die automatisch gestartet werden, wenn der Benutzer sich anmeldet, wie zum Beispiel der Arbeitsbereichsmanager (`dtwm`) und der Dateimanager (`dtfile`)
- Systemprozesse, wie zum Beispiel `rpc.ttdbserver` und `dtspcd`, die durch Mechanismen wie `inetd` gestartet werden.
- Anwendungen, die vom Desktop auf lokalen oder fernen Systemen gestartet werden
- Anwendungen, die vom Benutzer über eine Shell-Befehlszeile gestartet werden.

Um DTMOUNTPOINT für diese Prozesse zu setzen, muß der Benutzer

1. die Datei `/etc/inetd.conf` folgendermaßen bearbeiten:

a. den Eintrag `dtspcd` suchen und folgendes hinzufügen:

```
-mount_point einhängpunkt
```

b. den Eintrag `rpc.ttdbserver` suchen und folgendes hinzufügen:

```
-m einhängpunkt
```

Wird, beispielsweise, der Automounter mit dem Einhängpunkt `/nfs` verwendet, sehen die Einträge in `/etc/inetd.conf` folgendermaßen aus:

```
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
```

2. Die Systemprozedur ausführen, mit der `/etc/inetd.conf` erneut gelesen wird. Weitere Informationen können der Hilfeseite `inetd(1M)` entnommen werden.

3. DTMOUNTPOINT so setzen, daß der Wert von Benutzeranmeldungen übernommen wird.

Dies geschieht, indem die Variable in `/etc/dt/config/Xsession.d` gesetzt wird. Weitere Informationen zum Setzen von Umgebungsvariablen können dem Abschnitt "Umgebungsvariablen festlegen" auf Seite 36 entnommen werden.

Den Steuerungsdaemonprozeß für Unterprozesse konfigurieren

Der Unterprozeßsteuerungs-Service (SPC) des Desktops stellt Client-/Server-Befehlsausführungen zur Verfügung.

Der Steuerungsdaemonprozeß für Unterprozesse des Desktops (`dtspcd`) wird vom Desktop verwendet, um ferne Anwendungen zu starten. Es handelt sich um einen `inet`-Dämon, der Anforderungen von fernen Clients zur Befehlsausführung akzeptiert. Weitere Informationen zur Konfiguration von `inet`-Dämonen können der Hilfeseite `inetd.conf(1M)` entnommen werden.

Die Aktionsaufrufbibliothek des Desktop verwendet den SPC-Service, um ferne Aktionen aufzurufen.

Dtspcd konfigurieren

1. Überprüfen, ob `dtspc` sowohl in `/etc/services` als auch in `/etc/inetd.conf` korrekt registriert ist. Weitere Informationen hierzu können dem Abschnitt Hilfeseite `dtspcd(1M)`.
2. Nur für HP-UX: Sicherstellen, daß `/usr/adm/inetd.sec` korrekt konfiguriert ist. Weitere Informationen können der Hilfeseite `inetd.sec(4)` entnommen werden.

SPC-Sicherheit

Die Identifikationsüberprüfung für den Unterprozeßsteuerungsservice basiert auf der Dateisystemidentifikationsüberprüfung. `dtspcd` muß auf ein *Identifikationsüberprüfungsverzeichnis* Zugriff haben, das bei allen SPC-Client-Systemen ebenfalls eingehängt ist.

Standardmäßig handelt es sich bei dem Identitätsüberprüfungsverzeichnis für `dtspcd` um das Home-Verzeichnis des Benutzers. Aber `dtspcd` kann auch so konfiguriert werden, daß ein anderes Verzeichnis gewählt wird, indem die Option `-auth_dir` in dem Verzeichnis `/etc/inetd.conf` gesetzt wird. Weitere Informationen hierzu können der Hilfeseite `dtspcd(1M)` entnommen werden.

Da die SPC-Identitätsüberprüfung auf der Identitätsüberprüfung des Dateisystems basiert, ist der SPC-Service nur so sicher, wie das verteilte Dateisystem. Wird das Desktop in einem Netzwerk verwendet, in dem der

Benutzer dem verteilten Dateisystem nicht traut, kann er `dtspcd` inaktivieren. Hierzu muß der Eintrag `dtspc` in `/etc/services` auf Kommentar gesetzt werden

Umgebungsvariablen für eine Ausführung auf einem fernen System konfigurieren

Verwendet das Desktop eine Aktion, um eine Anwendung auf einem fernen System zu starten, werden die Umgebungsvariablen des Benutzers in das ferne System kopiert und in die Umgebung der Anwendung gestellt.

Standardmäßig werden einige der Umgebungsvariablen geändert, bevor sie in das ferne System kopiert werden. Sowohl die Aktionsaufrufkomponente als auch der Steuerungsdaemonprozeß für Unterprozesse des Desktops können so konfiguriert werden, daß eine zusätzliche Verarbeitung der Umgebungsvariablen erfolgt, bevor die Variablen in die Umgebung der Anwendung gestellt werden.

Weitere Informationen zur Standardkonfiguration und den Möglichkeiten, diese zu ändern, können den Hilfeseiten `dtactionfile(4)` und `dtspcdenv(4)` entnommen werden.

ToolTalk-Datenbank-Server konfigurieren

Eine Komponente von ToolTalk ist der ToolTalk-Datenbank-Server, `/usr/dt/bin/rpc.ttdbserver`.

Der ToolTalk-Datenbank-Server wird vom ToolTalk-Meldungsservice und für die Dateinamenzuordnung verwendet. Dieser Server wird normalerweise bei der Installation des Desktops in `/etc/inetd.conf` registriert und eine zusätzliche Konfiguration ist nicht notwendig.

Weitere Informationen zum ToolTalk-Datenbank-Server und dessen Konfigurationsoptionen können der Hilfeseite `rpc.ttdbserver(1M)` entnommen werden.

ToolTalk-Meldungs-Server konfigurieren

Bei dem ToolTalk-Meldungs-Server handelt es sich um `ttsession`. Standardmäßig muß bei diesem Server keine Konfiguration vorgenommen werden. Er wird während der Anmeldung durch die Prozedur `Xsession` gestartet.

Weitere Informationen zum ToolTalk-Meldungs-Server und dessen Konfigurationsoptionen können der Hilfeseite `ttsession` entnommen werden.

Kalender-Dämon konfigurieren

Eine Komponente der Anwendung 'Kalender' ist der Kalender-Dämon `rpc.cmsd`. Dieser Dämon wird normalerweise bei der Installation des Desktops in `/etc/inetd.conf` registriert und eine zusätzliche Konfiguration ist nicht notwendig.

Weitere Informationen zum Kalender-Dämon und dessen Konfigurationsoptionen können der Hilfeseite `rpc.cmsd(1)` entnommen werden.

Anwendungs-Services verwalten

Dieser Abschnitt behandelt spezifische Konfigurationserfordernisse für:

- Anwendungs-Server und deren Clients
- Desktop-Server, die spezielle Services zur Verfügung stellen, wie z.B. Datenbank-Server, Symbol-Server und Hilfe-Server

Außerdem werden Netzwerkerfordernisse für zwei spezielle Konfigurationen für Anwendungen im Netzwerk behandelt:

- Hosts für Ausführungen auf einem fernen System
- Anwendungen, die über Dateisystemeinhängungen ausgeführt werden

Suchpfadumgebungsvariablen

Das Desktop verwendet eine Gruppe von Umgebungsvariablen, um den Suchpfad anzugeben, über den Desktop-Konfigurationsdateien für Anwendungen, wie z. B. die Datenbank für Aktionen und Datentypen, Hilfedateien oder Symboldateien, gesucht werden.

Weitere Informationen zur Verwendung der Suchpfadumgebungsvariablen können Kapitel 7, "Desktop-Suchpfade" oder der Hilfeseite `dtenvvar(5)` entnommen werden.

Anwendungs-Server und dessen Clients konfigurieren

In der Standardkonfiguration des Anwendungs-Servers enthält der Anwendungs-Server alle zu einer Anwendung gehörenden binären Dateien und Konfigurationsdateien, wie:

- die ausführbare(n) Datei(en) der Anwendung
- Standardanwendungskonfigurationsdateien wie `app-defaults`, Nachrichtenkataloge sowie gemeinsam benutzte Bibliotheken für diese Anwendung
- Desktop- Konfigurationsdateien:
 - Definitionsdateien für Aktionen und Datentyp
 - Symbolabbilddateien
 - Desktop-Hilfedatendateien

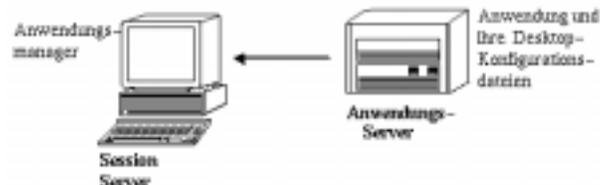


Abbildung 5-5 Standardkonfiguration des Anwendungs-Servers

▼ *Anwendungs-Server konfigurieren*

1. Die für das Desktop erforderlichen Netzwerkkonfigurationen des Betriebssystems einrichten.

Weitere Informationen hierzu können dem Abschnitt “Netzwerk des Basisbetriebssystems für das Desktop konfigurieren” auf Seite 100 entnommen werden.

2. Die für Server erforderliche allgemeine Desktop-Konfiguration einrichten.

Weitere Informationen hierzu können dem Abschnitt “Clients und Server für das Desktop konfigurieren” auf Seite 105 entnommen werden.

3. Die Anwendung(en) installieren.

4. Wird eine Anwendung nicht automatisch registriert, muß die Registrierung durchgeführt werden.

Weitere Informationen hierzu können dem Abschnitt Kapitel 4, “Anwendungen registrieren” entnommen werden.

▼ *Client eines Anwendungs-Servers konfigurieren*

1. Die für das Desktop erforderlichen Netzwerkkonfigurationen des Betriebssystems einrichten.

Weitere Informationen hierzu können dem Abschnitt “Netzwerk des Basisbetriebssystems für das Desktop konfigurieren” auf Seite 100 entnommen werden.

2. Die für Clients erforderliche allgemeine Desktop-Konfiguration einrichten.

Weitere Information hierzu können dem Abschnitt “Clients und Server für das Desktop konfigurieren” auf Seite 105 entnommen werden.

3. Den Anwendungs-Server dem Anwendungssuchpfad entweder für das gesamte System oder für den einzelnen Benutzer hinzufügen:

systemweit Die Variable DTSPSYSAPPHOSTS in
/etc/dt/config/Xsession.d/0010.dtpaths setzen.

persönlich Die Variable DTSPUSERAPPHOSTS in *Home-Verzeichnis*/.dtprofile setzen.

Beispielsweise fügt die folgende Zeile in `/etc/dt/config/Xsession.d/0010.dtpaths` ein System mit den Host-Namen `SysAAA` und `SysBBB` zum Anwendungssuchpfad hinzu:

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

Weitere Informationen zur Einrichtung des Anwendungssuchpfads können folgenden Abschnitten entnommen werden:

- “Anwendungssuchpfad” auf Seite 131
- “Wert eines Suchpfads setzen” auf Seite 129

Datenbank-, Symbol- und Hilfe-Services

Normalerweise werden die zu einer Anwendung gehörenden Definitionen für Aktionen und Datentyp, Symbole und Hilfedatendateien auf dasselbe System wie die Anwendung installiert.

Die typische Konfiguration von Hilfedatendateien sieht folgendermaßen aus:

- Die Hilfedateien für den Dateimanager befinden sich normalerweise auf dem Session-Server. Das Desktop findet sie, weil der Hilfesuchpfad automatisch die richtigen Positionen auf dem Session-Server durchsucht.
- Die Hilfedateien für andere Anwendungen befinden sich normalerweise auf demselben Anwendungs-Server wie die Anwendung. Der Session-Server findet sie, weil eine Änderung des Anwendungssuchpfads automatisch den Hilfesuchpfad ändert.

Möglicherweise gibt es Situationen, in denen der Benutzer Datenbank-(Aktionen und Datentypen), Hilfe- oder Symboldaten an eine anderen Position im Netzwerk stellen möchte. Verwendet das Netzwerk beispielsweise mehrere Session-Server, soll möglicherweise ein Hilfe-Server erstellt werden, auf dem alle Hilfedatendateien für Desktop-Anwendungen (Dateimanager, Umgebungsmanager, usw) gespeichert werden. Dadurch wird Plattenspeicherplatz gespart, da die Hilfedateien nicht auf jedem einzelnen Session-Server vorhanden sein müssen.

▼ **Datenbank-, Hilfe- oder Symbol-Server erstellen**

1. Die für das Desktop erforderlichen Netzwerkkonfigurationen des Betriebssystems einrichten.

Weitere Informationen hierzu können dem Abschnitt “Netzwerk des Basisbetriebssystems für das Desktop konfigurieren” auf Seite 100 entnommen werden.

2. Die für Clients erforderliche allgemeine Desktop-Konfiguration einrichten.

Weitere Informationen hierzu können dem Abschnitt “Clients und Server für das Desktop konfigurieren” auf Seite 105 entnommen werden.

3. Die Datenbank-, Hilfe- oder Symboldateien installieren.

Es ist dem Benutzer freigestellt, wo er die Dateien auf das System stellen möchte. Es empfiehlt sich jedoch, die folgenden Positionen zu verwenden, da diese Verzeichnisse automatisch durchsucht werden, wenn ein System als Anwendungs-Server verwendet wird.

- Datenbankdateien: `/etc/dt/appconfig/types/sprache`
- Hilfedateien: `/etc/dt/appconfig/help/sprache`
- Symboldateien: `/etc/dt/appconfig/icons/sprache`

Wird ein Datenbank-Server eingerichtet, müssen die Aktionen geschrieben werden, um anzugeben, wo die zugehörigen Befehle (`EXEC_STRINGS`) ausgeführt werden. Weitere Informationen hierzu können dem Abschnitt “Rechner für die Ausführung auf einem fernen System angeben” auf Seite 116 entnommen werden.

▼ **Verbindung zwischen Session-Server und Datenbank-, Symbol- oder Hilfe-Server konfigurieren**

1. Die für das Desktop erforderlichen Netzwerkkonfigurationen des Betriebssystems einrichten.

Weitere Informationen hierzu können dem Abschnitt “Netzwerk des Basisbetriebssystems für das Desktop konfigurieren” auf Seite 100 entnommen werden.

2. Die für Clients erforderliche allgemeine Desktop-Konfiguration einrichten.
Weitere Informationen hierzu können dem Abschnitt “Clients und Server für das Desktop konfigurieren” auf Seite 105 entnommen werden.
3. Den Datenbank-, Symbol- oder Hilfe-Server im entsprechenden Suchpfad hinzufügen.
 - Wurden die Datendateien an die Positionen gestellt, die in Schritt 3 des Abschnitts “Datenbank-, Hilfe- oder Symbol-Server erstellen” angegeben wurden, kann der Anwendungssuchpfad geändert werden.
 - Wurden die Datendateien an andere Positionen gestellt, muß der spezifische Suchpfad geändert werden.

Wurden beispielsweise die Hilfedateien in das Verzeichnis `/etc/dt/help on system SysCCC` gestellt, muß die folgende Zeile zu `/etc/dt/config/Xsession.d/0010.dtpaths` hinzugefügt werden:

```
DTSPSYSHELP=/net/SysCCC/etc/dt/help
```

Weitere Informationen zur Einrichtung von Suchpfaden können folgenden Abschnitten entnommen werden:

- “Datenbanksuchpfad (Aktionen/Datentypen)” auf Seite 134
- “Symbolsuchpfad” auf Seite 136
- “Hilfesuchpfad” auf Seite 138
- “Wert eines Suchpfads setzen” auf Seite 129

Spezielle Konfigurationen für Anwendungen im Netzwerk

In diesem Abschnitt wird beschrieben, wie Systeme konfiguriert werden, damit Anwendungen:

- an einem anderen System, als an dem System, das die Aktion enthält, ausgeführt werden, d.h. auf einem Rechner für die Ausführung auf einem fernen System.
- lokal über Dateisystemeinhängungen ausgeführt werden können.

Rechner für die Ausführung auf einem fernen System angeben

In der typischen Anwendungs-Serverkonfiguration befindet sich die Aktionsdefinition auf demselben System wie die ausführbare Datei der Anwendung. Aber es können auch Aktionen geschrieben werden, mit denen

Befehle an anderen Systemen ausgeführt werden können. In einer solchen Konfiguration wird das System, das die Anwendung enthält, als der *ausführende Rechner* bezeichnet.

Die Aktionsdefinition kann sich sowohl auf dem Session-Server als auch auf einem System, das dem Session-Server Aktions- und Datentyp-Services zur Verfügung stellt und das als *Datenbank-Server* oder *Datenbankrechner* bezeichnet wird, befinden.

In Aktionsdefinitionen wird durch das Feld EXEC_HOST angegeben, wo die Befehle (EXEC_STRINGS) ausgeführt werden sollen. Beispielsweise wird in der folgenden Aktionsdefinition angegeben, daß ein Client xload auf einem System mit dem Rechnernamen SysDDD ausgeführt wird:

```
ACTION XloadSysDDD
{
  TYPE          COMMAND
  EXEC_HOST     SysDDD
  EXEC_STRING    /usr/bin/X11/xload -label SysDDD
}
```

In Aktionsdefinitionen wird durch das Feld EXEC_HOST mehr als ein Rechnername angegeben, dann versucht das Desktop, EXEC_STRING nacheinander auf jedem der angegebenen Rechner auszuführen, bis es einen Rechner findet, der die Aktion ausführen kann. Beispielsweise wird im folgenden Feld EXEC_HOST angegeben, daß die Aktion zunächst versuchen soll, EXEC_STRING auf SysDDD, und falls dies fehlschlägt, auf SysEEE auszuführen:

```
EXEC_HOST      SysDDD, SYSEEE
```

Ist das Feld EXEC_HOST für eine Aktion nicht gesetzt, wird der Standardwert %DatabaseHost% gesetzt. Der Wert %DatabaseHost% wird dem Datenbanksuchpfad entnommen.

Angenommen, der Datenbanksuchpfad wurde geändert, indem folgende Zeile zu /etc/dt/config/Xsession.d/0010.dtpaths hinzugefügt wurde:

```
DTSPSYSDATABASEHOSTS=SysAAA: ,/net/SysBBB/etc/dt/appconfig/types/C
```

SysAAA wird mit der qualifizierten Syntax für die Angabe des Rechners, SysAAA: angegeben. Eine Aktion, die dieses Element des Suchpfads verwendet, setzt den Datenbankrechner auf SysAAA. Verwendet eine Aktion den Abschnitt /net/SysBBB... des Suchpfads, wird als Datenbankrechner das lokale System verwendet, da die Syntax das Qualifikationsmerkmal nicht enthält.

Rechner für die Ausführung auf einem fernen System konfigurieren

1. Die für das Desktop erforderlichen Netzwerkkonfigurationen des Betriebssystems einrichten.
Weitere Informationen hierzu können dem Abschnitt “Netzwerk des Basisbetriebssystems für das Desktop konfigurieren” auf Seite 100 entnommen werden.
2. Die für Server erforderliche allgemeine Desktop-Konfiguration einrichten.
Weitere Informationen hierzu können dem Abschnitt “Clients und Server für das Desktop konfigurieren” auf Seite 105 entnommen werden.
3. Sicherstellen, daß die Anwendungen für die Ausführung auf dem lokalen System richtig installiert und konfiguriert sind.

System, das die Aktionsdefinitionen enthält, konfigurieren

1. Die für das Desktop erforderlichen Netzwerkkonfigurationen des Betriebssystems einrichten.
Weitere Informationen hierzu können dem Abschnitt “Netzwerk des Basisbetriebssystems für das Desktop konfigurieren” auf Seite 100 entnommen werden.
2. Die für Server erforderliche allgemeine Desktop-Konfiguration einrichten.
Weitere Informationen hierzu können dem Abschnitt “Clients und Server für das Desktop konfigurieren” auf Seite 105 entnommen werden.
3. Die Aktionsdefinitionen und Anwendungsgruppen erstellen und installieren.
Weitere Informationen hierzu können dem Abschnitt “Aktionen erstellen, die Anwendungen auf fernen Systemen ausführen” auf Seite 198 und “Allgemeine Gruppen von Anwendungen erstellen und verwalten” auf Seite 51 entnommen werden.

Session-Server konfigurieren

1. Die für das Desktop erforderlichen Netzwerkkonfigurationen des Betriebssystems einrichten.

Weitere Informationen hierzu können dem Abschnitt “Netzwerk des Basisbetriebssystems für das Desktop konfigurieren” auf Seite 100 entnommen werden.

2. Die für Clients erforderliche allgemeine Desktop-Konfiguration einrichten.

Weitere Informationen hierzu können dem Abschnitt “Clients und Server für das Desktop konfigurieren” auf Seite 105 entnommen werden.

3. Den Datenbankrechner dem Suchpfad für Aktionen hinzufügen.

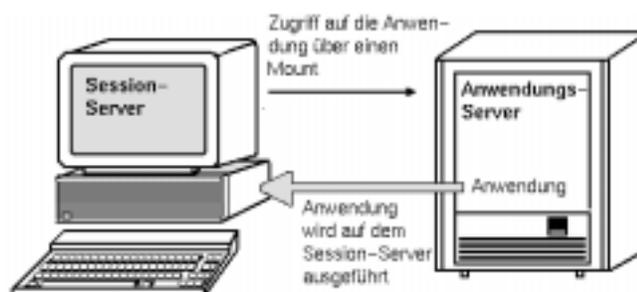
Weitere Informationen hierzu können dem Abschnitt “Datenbanksuchpfad (Aktionen/Datentypen)” auf Seite 134 entnommen werden.

4. Den ausführenden Rechner dem Anwendungssuchpfad hinzufügen.

Weitere Informationen hierzu können dem Abschnitt “Anwendungssuchpfad” auf Seite 131 entnommen werden..

Anwendungen lokal ausführen

Die Server-Standardanwendungsconfiguration führt Anwendungen auf dem Anwendungs-Server aus. Manchmal ist es wünschenswert, daß eine Anwendung auf einem fernen System installiert ist, aber auf dem Session-Server lokal ausgeführt wird..



*Abbildung 5-6*Ausführung über Einhängepunkte

Anwendungs-Server konfigurieren

Eine spezielle Konfiguration ist nicht erforderlich.

Session-Server konfigurieren

- ♦ Den Anwendungssuchpfad ändern. Den lokalen, absoluten Pfad zu der Anwendung verwenden.

Beispielsweise könnte die folgende Variablendefinition verwendet werden, um eine Anwendung zu finden, die auf `sysAAA` registriert ist:

```
DTSPSYSAPPHOSTS=/net/SysAAA/etc/dt/appconfig/appmanager/C
```

Der Session-Server muß auf die Konfigurationsdateien der Anwendung, wie `app-defaults`, Nachrichtenkataloge und gemeinsam benutzte Bibliotheken zugreifen können.

Verwaltung und Drucken über das Desktop konfigurieren

Für den Desktop-Benutzer gibt es verschiedene Möglichkeiten, Dateien zu drucken. Es wird zwischen zwei Kategorien unterschieden: Drucken über das Desktop und Drucken über eine Anwendung.

Zum Drucken über das Desktop gehören:

- Das Auswählen einer Datei im Dateimanager und das Anklicken von 'Drucken' im Menü 'Ausgewählt' oder im (temporär eingeblendeten) Menü des Symbols.
- Das Ziehen einer Datei vom Dateimanager zum Druckersymbol im Bedienfeld oder der Bedientafel 'Persönliche Drucker'.
- das Ziehen einer Datei vom Dateimanager zu einem Drucker im Hauptfenster des Druckmanagers

Um über eine Anwendung zu drucken, den Befehl 'Drucken' auf den über ein Menü oder ein anderes Bedienelement innerhalb des Anwendungsfensters zugegriffen wird, verwenden.

<i>Drucker hinzufügen und löschen</i>	<i>122</i>
<i>Druckersymbolbilder</i>	<i>123</i>
<i>Standarddrucker konfigurieren</i>	<i>125</i>
<i>Druckerkonzepte</i>	<i>126</i>

Drucker hinzufügen und löschen

In diesem Abschnitt werden die Prozeduren erläutert, mit denen Drucker dem Desktop hinzugefügt oder vom Desktop gelöscht werden.

▼ Drucker dem Desktop hinzufügen

1. Einen Drucker der Konfiguration des Systems hinzufügen.
Dabei den Anweisungen in der Dokumentation zur Systemverwaltung folgen.
2. Folgenden Befehl ausführen:

```
env LANG=Wesprache /usr/dt/bin/dtprintinfo -populate
```
3. Den Druckmanager erneut starten oder 'Aktionen erneut laden' in der Gruppe von Anwendungen 'Desktop_Werkzeuge' im Anwendungsmanager doppelklicken. Überprüfen, ob der Drucker angezeigt wird.
4. Über die elektronische Post die Benutzer benachrichtigen, daß sie ebenfalls den Druckmanager erneut starten oder 'Aktionen erneut laden' ausführen sollten.

Bei jedem Aufruf liest der Druckmanager die Konfigurationsliste für Systemdrucker erneut. Wird ein neuer Drucker festgestellt, wird automatisch eine neue Desktop-Druckeraktion und ein Symbol für diesen Drucker erstellt. Der Benutzer muß keine weiteren Schritte unternehmen, damit der Drucker auf dem desktop angezeigt wird.

▼ Drucker vom Desktop löschen

1. Den Drucker aus der Konfiguration des Systems entfernen.
Dabei den Anweisungen in der Dokumentation zur Systemverwaltung folgen.
2. Den Druckmanager erneut starten oder 'Aktionen erneut laden' in der Gruppe von Anwendungen Desktop_Werkzeuge im Anwendungsmanager doppelklicken. Überprüfen, ob der Drucker nicht mehr angezeigt wird.
3. Über die elektronische Post die Benutzer benachrichtigen, daß sie ebenfalls den Druckmanager erneut starten oder 'Aktionen erneut laden' ausführen sollten.

Bei jedem Aufruf liest der Druckmanager die Konfigurationsliste für Systemdrucker. Wird festgestellt, daß ein Drucker von der Liste entfernt wurde, wird automatisch die Druckeraktion und das Symbol für diesen Drucker aus dem Druckmanager und dem Dateimanager entfernt. Der Benutzer muß keine weiteren Schritte unternehmen, damit der Drucker vom Desktop gelöscht wird.

Hinweis – Der Druckmanager kann Drucker nicht aus dem Bedienfeld entfernen. Deshalb sollten, wenn ein Drucker aus der Konfiguration entfernt wird, alle Benutzer über die elektronische Post benachrichtigt werden, daß sie Symbole des gelöschten Druckers im Bedienfeld löschen sollten.

Intervall, in dem Jobs aktualisiert werden, ändern

Um zu ändern, wie häufig die im Druckmanager angezeigten Informationen aktualisiert werden, muß das Intervall für die Aktualisierung von Jobs geändert werden. Standardmäßig fragt der Druckmanager Drucker alle dreißig Sekunden nach dem Status ihrer Druckjobs ab. Dieses Zeitintervall kann mit der Schiebeleiste für das Aktualisierungsintervall im Dialogfenster 'Optionen festlegen' geändert werden. (Dieses Dialogfenster wird angezeigt, indem 'Optionen festlegen' im Menü 'Ansicht' ausgewählt wird.)

Druckersymbolbilder

Wird ein Drucker hinzugefügt, wird ihm automatisch das Standarddruckersymbol zugeordnet. Soll dem Drucker ein anderes Symbol verfügbar gemacht werden, die Symboldateien in `/etc/dt/appconfig/icons/sprache` oder in ein anderes Verzeichnis im Symbolsuchpfad stellen. Die Benutzer können dann dieses Symbol auswählen, um das Standardsymbol für den Drucker zu ersetzen.

Es *muß* ein vollständiger Satz (groß, mittel und klein) der Symbole vorhanden sein, sonst werden sie im Anwendungsauswahlmenü für Symbole des Druckmanagers nicht angezeigt.

Weiter Informationen zum können dem Abschnitt "Symbolsuchpfad" auf Seite 136 entnommen werden.

Symboldateinamen und -größen

Für Symboldateinamen ist folgendes Format erforderlich:

dateiname ohne pfad.größe.typ

wobei

größe die Größe angibt. Möglich sind: *l* (groß), *m* (mittel), *t* (klein). Weitere Informationen zu der Größe von Symbolen können dem Abschnitt "Symbolgrößenkonventionen" auf Seite 225 entnommen werden.

typ angibt, ob es sich um eine Pixmap- (*pm*) oder um eine Bitmap-Datei (*bm*) handelt.

Symboldateinamen für mittelgroße und kleine Pixmap-Symbole für einen Farbdrucker könnten beispielsweise als `ColorPrinter.m.pm` und `ColorPrinter.t.pm` bezeichnet werden..

Weitere Informationen zum Erstellen von Symbolen können Kapitel 12, "Symbole für das Desktop erstellen" entnommen werden.

▼ Symbol, Druckerbezeichnung oder Beschreibung eines Druckers global ändern

Globale Druckermerkmale sollten sofort nach Hinzufügen des Druckers geändert werden, bevor Benutzer sie mit dem Druckmanager ändern. Sobald ein Benutzer die Druckermerkmale mit dem Druckmanager geändert hat, werden nachträgliche globale Änderungen nicht mehr sichtbar.

Die Datei `/etc/dt/appconfig/types/sprache/druckerschlange_name.dt` bearbeiten und die gewünschten Änderungen für das Symbol, die Druckerbeschreibung und die Beschreibung vornehmen:

1. Im Feld `ICON` für *dateiname ohne pfad* den neuen Dateinamen ohne Pfad für das Symbol eingeben.
2. Im Feld `LABEL` für *labelname* die neue Bezeichnung für den Drucker eingeben.

- ◆ Den Text im Feld `DESCRIPTION` aktualisieren.

Hier empfiehlt es sich, die Position des Druckers, den Druckertyp und den zuständigen Hardwarebeauftragten anzugeben. Soll mehr Text als eine Zeile geschrieben werden, am Ende der Zeile einen umgekehrten Schrägstrich (`\`) setzen. Beispiel:

```
DESCRIPTION Dies ist ein PostScript-Drucker in Gebäude 1\Zimmer
              123. Bei Problemen bitte 555-5555 anrufen
```

Standarddrucker konfigurieren

Auf den Standarddrucker wird zugegriffen, wenn der Benutzer

- ein Objekt an das Druckersymbol im Bedienfeld übergibt
- ein Objekt im Dateimanager auswählt und 'Drucken' im Menü 'Ausgewählt' oder im (temporär eingeblendeten) Menü des Symbols anklickt.
- über Anwendungen druckt, die den Standarddrucker verwenden.

▼ Standarddrucker ändern

Der Standarddrucker wird für alle Benutzer geändert, indem

1. die Datei `/etc/dt/config/Xsession.d/0010.dtpaths` geöffnet wird.

Ist `/etc/dt/config/Xsession.d/0010.dtpaths` nicht vorhanden, muß sie aus `/usr/dt/config/Xsession.d/0010.dtpaths` kopiert werden.

2. in der Zeile `LPDEST=drucker` für `drucker` der neue Standarddrucker angegeben wird.

Ist die Zeile nicht vorhanden, eine Zeile `LPDEST=drucker` hinzufügen, in der für `drucker` der Name des Druckers der als Standarddrucker verwendet werden soll, angegeben wird.

3. die Benutzer sich ab- und wieder anmelden.

Der Standarddrucker wird für einen einzelnen Benutzer geändert, indem

- ◆ ein anderer Drucker aus der Bedientafel 'Persönliche Drucker' in das Bedienfeld kopiert wird.

Ein anderer Drucker wird als Standarddrucker angegeben, indem

1. das Home-Verzeichnis angeklickt und die Datei `.dtprofile` geöffnet wird.
2. eine Zeile bearbeitet oder hinzugefügt wird, die einen Wert für die Umgebungsvariable `LPDEST` setzt:

```
LPDEST=drucker_gerät; export LPDEST
```

Wird `csh` verwendet, lautet die Syntax:

```
setenv LPDEST drucker_gerät
```

Beispielsweise setzt die folgende Zeile als Standarddrucker den Drucker mit dem Einheitennamen `laser3d`.

```
LPDEST=laser3d; export LPDEST
```

Wird `csh` verwendet, lautet die Syntax:

```
setenv LPDEST laser3d
```

Druckerkonzepte

Wird eine Druckanforderung eingeleitet, indem eine Datei einem Druckerbedienelement übergeben wird, läuft im System folgende Prozedur ab:

1. Das System durchsucht die Datentypdatenbank nach einer Definition für das übergebene Objekt.
2. Ist eine eindeutige Druckaktion für den Datentyp vorhanden (in dem Feld `ARG_TYPE` in der Druckaktion angegeben), wird diese verwendet. Ansonsten wird die Standarddruckaktion (`dtlp`) verwendet. Handelt es sich bei der Datei, beispielsweise, um eine PostScript®-Datei, verwendet das System die Druckaktion für PostScript-Dateien. (Diese Aktion ist in `/usr/dt/appconfig/types/sprache/dt.dt` definiert.) Wurde das Werkzeug 'Aktion erstellen' für diesen Datentyp verwendet, handelt es sich bei dem Druckbefehl um die eindeutige Druckaktion, die verwendet wird, um Dateien mit diesem Datentyp zu drucken.
3. Die Datei wird über das normale UNIX-Drucksubsystem `lp` an den Drucker geschickt.

Desktop-Suchpfade



Das Desktop verwendet Suchpfade, um die Position von Anwendungen und deren zugehörigen Desktop-Dateien festzustellen.

<i>Desktop-Suchpfade und ihre Umgebungsvariablen</i>	<i>128</i>
<i>Wert eines Suchpfads setzen</i>	<i>129</i>
<i>Anwendungssuchpfad</i>	<i>131</i>
<i>Datenbanksuchpfad (Aktionen/Datentypen)</i>	<i>134</i>
<i>Symbolsuchpfad</i>	<i>136</i>
<i>Hilfesuchpfad</i>	<i>138</i>
<i>Positionen der Suchpfade</i>	<i>140</i>

Das Desktop stellt vier Suchpfade zur Verfügung, die in Tabelle 7-1 beschrieben sind.

Tabelle 7-1 Desktop-Suchpfade

Suchpfad	Beschreibung
Anwendungen	Wird verwendet, um die Position von Anwendungen festzustellen. Der Anwendungsmanager verwendet den Anwendungssuchpfad, um bei der Anmeldung eines Benutzers die oberste Ebene dynamisch zu erstellen.
Datenbank	Wird verwendet, um zusätzliche Positionen für Definitionsdateien für Aktionen und Datentypen (*.dt-Dateien) und Bedienfelddateien (*.fp-Dateien) anzugeben.
Symbole	Wird verwendet, um zusätzliche Positionen für Symbole anzugeben.
Hilfedaten	Wird verwendet, um zusätzliche Positionen für Desktop-Hilfedaten anzugeben.

Die Suchpfade können sowohl lokale als auch ferne Verzeichnisse enthalten. Auf diese Weise spielen Suchpfade eine wichtige Rolle in der Netzwerkarchitektur des Desktops. Beispielsweise findet ein System Anwendungen auf einem Anwendungs-Server, weil dieser Anwendungs-Server im Anwendungssuchpfad aufgelistet ist.

Enthält ein Suchpfad eine ferne Position, muß der ferne Zugriff auf diese Position konfiguriert werden. Weitere Informationen hierzu können. "Zugriff auf verteilte Dateisysteme konfigurieren" auf Seite 102 entnommen werden

Desktop-Suchpfade und ihre Umgebungsvariablen

Die Desktop-Suchpfade werden bei der Anmeldung durch das Desktop-Dienstprogramm `dtsearchpath` erstellt. Dieses Dienstprogramm verwendet eine Kombination aus Umgebungsvariablen und integrierten Positionen, um die Suchpfade zu erstellen.

Die Umgebungsvariablen, die `dtsearchpath` liest, werden als *Eingabevariablen* bezeichnet. Diese Variablen werden vom Systemadministrator oder vom Endbenutzer gesetzt. Die Eingabevariablen verwenden die Benennungskonvention `DTSP*`.

Wird `dtsearchpath` zum Zeitpunkt der Anmeldung ausgeführt, assembliert es die diesen Variablen zugeordneten Werte, fügt integrierte Positionen hinzu, und erstellt Werte für *Ausgabevariablen*. Für jeden Suchpfad gibt es eine Ausgabevariable.

Tabella 7-2 Umgebungsvariablen des Desktop-Suchpfads

Suchpfad für:	Ausgabeumgebungsvariable	Systemweite Eingabevariable	Persönliche Eingabevariable
Anwendungen	DTAPPSEARCHPATH	DTSPSYSAPPHOSTS	DTSPUSERAPPHOSTS
Datenbank	DTDATABASESEARCHPATH	DTSPSYSDATABASEHOSTS	DTSPUSERDATABASEHOSTS
Symbole	XMICONSEARCHPATH, XMICONBMSEARCHPATH	DTSPSYSICON	DTSPUSERICON
Hilfedaten	DTHELPSEARCHPATH	DTSPSYSHELP	DTSPUSERHELP

1. Aktionen, Datentypen und Bedienfelddefinitionen

Komponenten verwenden die Werte der Ausgabevariablen. Zum Beispiel verwendet der Anwendungsmanager den Wert des Anwendungssuchpfads (DTAPPSEARCHPATH), um die Position von Gruppen von Anwendungen festzustellen.

Wert eines Suchpfads setzen

Suchpfade können entweder für das gesamte System oder für einzelne Benutzer geändert werden. Änderungen werden durchgeführt, indem Werte für die für das System definierten oder persönlichen Eingabevariablen gesetzt werden. Sämtliche Änderungen werden den integrierten Suchpfaden hinzugefügt.

▼ Aktuellen Wert für einen Suchpfad anzeigen (Ausgabevariable)

- ◆ Mit dem Befehl `dtsearchpath` die aktuellen Werte für die Suchpfade anzeigen:
 - Um den Wert für den aktuellen Benutzer (der sich gerade anmeldet) zu erhalten, folgenden Befehl eingeben:


```
dtsearchpath -v
```
 - Um den Wert für einen anderen Benutzer zu erhalten, folgenden Befehl eingeben:

`dtsearchpath -u benutzer`

Zu den Suchpfadwerten gehören diese Variablen:

- %H Wird in DTHELPSEARCHPATH verwendet. Es handelt sich um den Namen des Hilfebands.
- %B Wird in XMICONSEARCHPATH verwendet. Es handelt sich um den Dateinamen ohne Pfad einer Symboldatei.
- %M Wird in XMICONSEARCHPATH verwendet. Es handelt sich um die Größe der Symboldatei (.l, .m, .s, .t)
- %L Wert der Umgebungsvariable LANG.

▼ Persönliche Änderungen an einem Suchpfad vornehmen

1. Das Verzeichnis *Home-Verzeichnis*/`.dtprofile` zum Bearbeiten öffnen.
2. Eine Zeile hinzufügen oder Bearbeiten, in der ein Wert für die persönliche Eingabevariable definiert wird.

Beispielsweise fügt die folgende Zeile eine Position zum persönlichen Anwendungssuchpfad des Benutzers hinzu:

```
DTSPUSERAPPHOSTS=/projects1/editors
```

3. Damit die Änderungen angezeigt werden, muß der Benutzer sich ab- und wieder anmelden.

▼ Systemweite Änderungen an einem Suchpfad vornehmen

1. Als Benutzer 'root' anmelden.
2. Ist die Datei `/etc/dt/config/Xsession.d/0010.dtpaths` nicht vorhanden, muß sie durch Kopieren von `/usr/dt/config/Xsession.d/0010.dtpaths` erstellt werden.
3. `/etc/dt/Xsession.d/0010.paths` zum Bearbeiten öffnen. Eine Zeile in der ein Wert für die systemweite Eingabevariable definiert ist, hinzufügen oder bearbeiten.

Beispielsweise fügt die folgende Zeile eine Position zum systemweiten Hilfesuchpfad hinzu:

```
DTSPSYSHELP=/applications/helpdata
```

4. Alle Benutzer im System darüber informieren, daß sie sich ab- und wieder anmelden müssen, damit die Änderungen angezeigt werden.

Anwendungssuchpfad

Bei dem Anwendungssuchpfad handelt es sich um den primären Suchpfad, den das Desktop verwendet, um die Position von Anwendungen im lokalen System und auf Anwendungs-Servern im gesamten Netzwerk festzustellen.

Werden Positionen zum Anwendungssuchpfad hinzugefügt, werden die übrigen Suchpfade (Datenbank-, Symbol- und Hilfesuchpfad) automatisch aktualisiert, um die entsprechenden Positionen für diese Daten anzugeben. Auf diese Weise stellt der Anwendungssuchpfad eine einfache Verwaltung für Anwendungen und deren Desktop-Konfigurationsdateien zur Verfügung. Weitere Informationen hierzu können dem Abschnitt "Wie der Anwendungssuchpfad sich auf die Datenbank-, Symbol- und Hilfesuchpfade auswirkt" auf Seite 133 entnommen werden.

Standardanwendungssuchpfad

Der Standardanwendungssuchpfad enthält persönliche, systemweite und integrierte Positionen. Der Standardwert für *sprache* ist C.

Persönliche Position *Home-Verzeichnis*/.dt/appmanager

Systemweite Position /etc/dt/appconfig/appmanager/*sprache*

Integrierte Position /usr/dt/appconfig/appmanager/*sprache*

Umgebungsvariablen des Anwendungssuchpfads

Der Anwendungssuchpfad wird aus integrierten Positionen und den folgenden Eingabevariablen assembliert:

DTSPSYSAPPHOSTS Eingabevariable für den systemweiten Anwendungssuchpfad

DTSPUSERAPPHOSTS Eingabevariable für den persönlichen Anwendungssuchpfad

Der assemblierte Suchpfad wird durch die Ausgabevariable DTAPPSEARCHPATH angegeben.

Syntax für die Eingabevariablen des Anwendungssuchpfads

Die Syntax für die Variablen DTSPSYSAPPHOSTS und DTSPUSERAPPHOSTS lautet:

VARIABLE=position [, position . . .]

wobei *position* folgende Syntax haben kann:

<i>/pfad</i>	Gibt ein Verzeichnis im lokalen System (Session-Server) an. Diese Syntax wird verwendet, um ein lokales Verzeichnis hinzuzufügen.
<i>hostname:</i>	Gibt das systemweite Verzeichnis <i>/etc/dt/appconfig/appmanager/sprache</i> auf dem System <i>hostname</i> an. Diese Syntax wird verwendet, um einen Anwendungs-Server hinzuzufügen.
<i>hostname: /pfad</i>	Gibt ein Verzeichnis auf einem fernen System <i>hostname</i> an.
<i>localhost:</i>	Gibt die lokale systemweite Position an. Dieses Schlüsselwort wird verwendet, um den Rang der lokalen systemweiten Position zu ändern. Weitere Informationen hierzu können dem Abschnitt "Rangfolge der systemweiten lokalen Positionen ändern" auf Seite 133 entnommen werden.

Wie der Wert des Anwendungssuchpfads assembliert wird

Der Wert des Anwendungssuchpfads (DTAPPSEARCHPATH) wird erstellt, indem folgende Positionen, die ihrem Rang nach aufgelistet sind, assembliert werden:

- Positionen, die über die Variable DTSPUSERAPPHOSTS angegeben sind
- die standardmäßige persönliche Position: *Home-Verzeichnis/.dt/appmanager*
- die Standardposition: */etc/dt/appconfig/appmanager/sprache*
- Positionen, die über die Variable DTSPSYSAPPHOSTS angegeben sind
- */usr/dt/appconfig/appmanager/sprache*

Die Syntax:

hostname:

wird erweitert, um das Verzeichnis */etc/dt/appconfig/appmanager* auf dem System *hostname* anzugeben.

Rangfolge der systemweiten lokalen Positionen ändern

Standardmäßig hat die lokale systemweite Position (`/etc/dt/appconfig/appmanager/sprache`) Vorrang vor fernen Positionen. Auf diese Weise haben lokale Gruppen von Anwendungen Vorrang vor fernen Gruppen von Anwendungen mit demselben Namen. Verfügen beispielsweise sowohl lokale als auch ferne Systeme über Gruppen von Anwendungen mit dem Namen Drucker (`/etc/dt/appconfig/appmanager/sprache/Printers`), wird die lokale Gruppe verwendet.

Die Eingabevariablen des Anwendungssuchpfads stellen eine Syntax, die den Vorrang der lokalen systemweiten Gruppen von Anwendungen angibt, zur Verfügung:

```
localhost:
```

Angenommen, ein System soll auf die Anwendungs-Server `SysA`, `SysB` und `SysC` zugreifen und die systemweiten Gruppen von Anwendungen auf `SysB` sollen Vorrang vor jeder lokalen Gruppe mit demselben Namen haben.

Dies wird erreicht, indem der folgende Wert für `DTSPSYSAPPHOSTS` gesetzt wird:

```
DTSPSYSAPPHOSTS=SysB: ,localhost: ,SysA: ,SysC:
```

Wie der Anwendungssuchpfad sich auf die Datenbank-, Symbol- und Hilfesuchpfade auswirkt

Bei Hinzufügungen zum Anwendungssuchpfad werden die entsprechenden Positionen automatisch auch zu den Datenbank-, Symbol- und Hilfesuchpfaden hinzugefügt. Auf diese Weise kann ein Anwendungs-Server einem Suchpfad hinzugefügt werden, indem lediglich die Eingabevariable für den Anwendungssuchpfad gesetzt wird.

Wird beispielsweise `DTSPSYSAPPHOSTS` wie folgt gesetzt:

```
DTSPSYSAPPHOSTS=servera:
```

sind davon folgende Suchpfade betroffen:

Suchpfad **Dem Suchpfad hinzugefügtes Verzeichnis**

Anwendungen `servera:/etc/dt/appconfig/appmanager/sprache`

Datenbank `servera:/etc/dt/appconfig/types/sprache`

Symbol `servera:/etc/dt/appconfig/icons/sprache`

Hilfe `servera:/etc/dt/appconfig/help/sprache`

So ähnlich sind, wenn DTSPSYSAPPHOSTS wie folgt gesetzt wird:

`DTSPSYSAPPHOSTS=/projects1/apps`

folgende Suchpfade betroffen:

Suchpfad	Dem Suchpfad hinzugefügtes Verzeichnis
-----------------	---

Anwendungen	<code>/projects1/apps/appmanager/<i>sprache</i></code>
-------------	--

Datenbank	<code>/projects1/apps/types/<i>sprache</i></code>
-----------	---

Symbol	<code>/projects1/apps/icons/<i>sprache</i></code>
--------	---

Hilfe	<code>/projects1/apps/help/<i>sprache</i></code>
-------	--

Datenbanksuchpfad (Aktionen/Datentypen)

Der Datenbanksuchpfad weist das Desktop an, angegebene Positionen nach Dateien zu durchsuchen, die folgendes enthalten:

- Definitionen für Aktionen und Datentypen (* .dt-Dateien)
- Bedienfelddefinitionen (* .fp-Dateien).

Möglicherweise muß der Benutzer den Datenbanksuchpfad ändern, wenn er einen Datenbank-Server erstellt oder eine lokale Position für Datenbankdateien hinzufügt.

Standarddatenbanksuchpfad

Der Standarddatenbanksuchpfad enthält persönliche, systemweite und integrierte Positionen. Der Standardwert für *sprache* ist C.

Persönliche Position	<code>Home-Verzeichnis/.dt/types</code>
----------------------	---

Systemweite Position	<code>/etc/dt/appconfig/types/<i>sprache</i></code>
----------------------	---

Integrierte Position	<code>/usr/dt/appconfig/types/<i>sprache</i></code>
----------------------	---

Wie der Anwendungssuchpfad sich auf den Datenbanksuchpfad auswirkt

Wird eine Position zum Anwendungssuchpfad hinzugefügt, wird das entsprechende Datenbankunterverzeichnis automatisch dem Datenbanksuchpfad hinzugefügt. (Weitere Informationen hierzu können dem Abschnitt “Wie der Anwendungssuchpfad sich auf die Datenbank-, Symbol- und Hilfesuchpfade auswirkt” auf Seite 133 entnommen werden.)

Wird beispielsweise der Anwendungs-Server `hosta:` dem Anwendungssuchpfad hinzugefügt, wird das Verzeichnis `hosta:/etc/dt/appconfig/types/sprache` automatisch dem Datenbanksuchpfad hinzugefügt.

Umgebungsvariablen des Datenbanksuchpfads

Der Datenbanksuchpfad wird aus den integrierten Positionen und folgenden Eingabevariablen assembliert:

`DTSPSYSDATABASEHOSTS` Eingabevariable für den systemweiten Datenbanksuchpfad

`DTSPUSERDATABASEHOSTS` Eingabevariable für den persönlichen Datenbanksuchpfad

Diese Eingabevariablen verwenden, um Positionen außerhalb des Anwendungssuchpfads anzugeben.

Der assemblierte Datenbanksuchpfad wird durch die Ausgabevariable `DTDATABASESEARCHPATH` angegeben.

Syntax für die Eingabevariablen des Datenbanksuchpfads

Die Syntax für die Variablen `DTSPSYSDATABASEHOSTS` und `DTSPUSERDATABASEHOSTS` lautet:

`VARIABLE=position [, position . . .]`

wobei *position* folgende Syntax haben kann:

`/pfad` Gibt ein Verzeichnis im lokalen System (Session-Server) an. Diese Syntax wird verwendet, um ein lokales Verzeichnis hinzuzufügen.

hostname: Gibt das systemweite Verzeichnis
 /etc/dt/appconfig/types/*sprache* auf dem System
 hostname an.

hostname: /*pfad* Gibt ein Verzeichnis auf dem fernen System *hostname* an.

Wie der Datenbanksuchpfad assembliert wird

Der Wert des Datenbanksuchpfads (DTDATABASESEARCHPATH) wird erstellt, indem folgende Positionen, die ihrem Rang nach aufgelistet sind, assembliert werden:

- Positionen, die über die Variable DTSPUSERDATABASEHOSTS angegeben sind
- Positionen, die von der Variablen DTSPUSERAPPHOSTS abgeleitet werden.
- die standardmäßige persönliche Position: *Home-Verzeichnis*/.dt/types
- die Standardposition: /etc/dt/appconfig/types/*sprache*
- Positionen, die über die Variable DTSPSYSDATABASEHOSTS angegeben werden
- Positionen, die von der VariablenDTSPSYSAPPHOSTS abgeleitet werden
- /usr/dt/appconfig/types/*sprache*

Die Syntax:

hostname:

wird erweitert, um das Verzeichnis /etc/dt/appconfig/types auf dem System *hostname* anzugeben.

Symbolsuchpfad

Der Symbolsuchpfad weist das Desktop an, angegebene Positionen nach Dateien zu durchsuchen, die Bitmap- und Pixmap-Abbilddateien für das Desktop enthalten.

Standardsymbolsuchpfad

Der Standardsymbolsuchpfad enthält persönliche, systemweite und integrierte Positionen. Der Standardwert für *sprache* ist *C*.

Persönliche Position *Home-Verzeichnis*/.dt/icons

Systemweite Position /etc/dt/appconfig/icons/*sprache*

Integrierte Position `/usr/dt/appconfig/icons/sprache`

Wie der Anwendungssuchpfad sich auf den Symbolsuchpfad auswirkt

Wird eine Position zum Anwendungssuchpfad hinzugefügt, wird das entsprechende Symbolunterverzeichnis automatisch dem Symbolsuchpfad hinzugefügt. (Weitere Informationen hierzu können dem Abschnitt “Wie der Anwendungssuchpfad sich auf die Datenbank-, Symbol- und Hilfesuchpfade auswirkt” auf Seite 133 entnommen werden.)

Wird beispielsweise der Anwendungs-Server `hosta:` dem Anwendungssuchpfad hinzugefügt, wird das Verzeichnis `hosta:/etc/dt/appconfig/icons/sprache` automatisch dem Symbolsuchpfad hinzugefügt.

Umgebungsvariablen des Symbolsuchpfads

Der Symbolsuchpfad wird aus den integrierten Positionen und folgenden Eingabevariablen assembliert:

DTSPSYSICON Eingabevariable für den systemweiten Symbolsuchpfad

DTSPUSERICON Eingabevariable für den persönlichen Symbolsuchpfad

Diese Eingabevariablen verwenden, um Positionen außerhalb des Anwendungssuchpfads anzugeben.

Der assemblierte Symbolsuchpfad wird durch zwei Ausgabevariablen angegeben:

XMICONSEARCHPATH Wird von Farbbildschirmen verwendet.

XMICONBMSEARCHPATH Wird von Monochrombildschirmen verwendet.

Syntax für die Eingabevariablen des Symbolsuchpfads

Die Syntax für die Variablen DTSPSYSICON und DTSPUSERICON lautet:

`VARIABLE=position [, position . . .]`

wobei *position* folgende Syntax haben kann:

/pfad Gibt ein Verzeichnis im lokalen System (Session-Server) an. Diese Syntax wird verwendet, um ein lokales Verzeichnis hinzuzufügen.

Soll eine Position in einem anderen System angegeben werden, muß der Netzwerkdateiname, z. B. */nfs/servera/projects/icons* verwendet werden.

Wie der Symbolsuchpfad assembliert wird

Der Wert des Symbolsuchpfads (XMICONSEARCHPATH und XMICONBMSEARCHPATH) wird erstellt, indem folgende Positionen, die ihrem Rang nach aufgelistet sind, assembliert werden:

- Positionen, die über die Variable DTSPUSERICON angegeben sind
- Positionen, die von der Variablen DTSPUSERAPPHOSTS abgeleitet werden
- die standardmäßige persönliche Position: *Home-Verzeichnis/.dt/icons*
- die Standardposition: */etc/dt/appconfig/icons/sprache*
- Positionen, die über die Variable DTSPSYSICON angegeben sind
- Positionen, die von der Variablen DTSPSYSAPPHOSTS abgeleitet werden
- */usr/dt/appconfig/icons/sprache*

Farb- und Monochromsuchpfade unterscheiden sich lediglich in der Rangfolge, die Pixmap- und Bitmap-Dateien zugeordnet wird. Die Variable XMICONSEARCHPATH listet Pixmap-Dateien vor Bitmap-Dateien, die Variable XMICONBMSEARCPATH listet Bitmap-Dateien vor Pixmap-Dateien.

Hilfesuchpfad

Der Hilfesuchpfad weist das Desktop an, angegebene Positionen nach Dateien mit Hilfeinformationen zu durchsuchen, die auf dem System registriert werden.

Standardhilfesuchpfad

Der Standardhilfesuchpfad enthält persönliche, systemweite und integrierte Positionen. Der Standardwert für *sprache* ist *C*.

Persönliche Position *Home-Verzeichnis/.dt/help*

Systemweite Position */etc/dt/appconfig/.dt/help/sprache*

Integrierte Position `/usr/dt/appconfig/help/sprache`

Wie der Anwendungssuchpfad sich auf den Hilfesuchpfad auswirkt

Wird eine Position zum Anwendungssuchpfad hinzugefügt, wird das entsprechende Hilfeunterverzeichnis automatisch dem Hilfesuchpfad hinzugefügt (Weitere Informationen hierzu können dem Abschnitt “Wie der Anwendungssuchpfad sich auf die Datenbank-, Symbol- und Hilfesuchpfade auswirkt” auf Seite 133 entnommen werden.)

Wird beispielsweise der Anwendungs-Server `hosta:` dem Anwendungssuchpfad hinzugefügt, wird das Verzeichnis `hosta:/etc/dt/appconfig/help/sprache` automatisch dem Hilfesuchpfad hinzugefügt.

Umgebungsvariablen des Hilfesuchpfads

Der Hilfesuchpfad wird aus den integrierten Positionen und folgenden Eingabevariablen assembliert:

`DTSPSYSHELP` Eingabevariable für den systemweiten Hilfesuchpfad

`DTSPUSERHELP` Eingabevariable für den persönlichen Hilfesuchpfad

Diese Eingabevariablen verwenden, um Positionen außerhalb des Anwendungssuchpfads anzugeben.

Der assemblierte Datenbanksuchpfad wird durch die Ausgabevariable `DTHELPPSEARCHPATH` angegeben.

Syntax für die Eingabevariablen des Hilfesuchpfads

Die Syntax für die Variablen `DTSPSYSHELP` und `DTSPUSERHELP` lautet:
`VARIABLE=position [, position . . .]`

wobei *position* folgende Syntax haben kann:

`/pfad` Gibt ein Verzeichnis im lokalen System (Session-Server) an. Diese Syntax wird verwendet, um ein lokales Verzeichnis hinzuzufügen.

Soll eine Position in einem anderen System angegeben werden, muß der Netzwerkdateiname, z. B. `/nfs/servera/projects/help` verwendet werden.

Wie der Hilfesuchpfad assembliert wird

Der Wert des Hilfesuchpfads (DTHELPSEARCHPATH) wird erstellt, indem folgende Positionen, die ihrem Rang nach aufgelistet sind, assembliert werden:

- Positionen, die über die Variable DTSPUSERHELP angegeben werden
- Positionen, die von der Variablen DTSPUSERAPPHOSTS abgeleitet werden
- die standardmäßige persönliche Position: *Home-Verzeichnis*/`.dt/help`
- die Standardposition: `/etc/dt/appconfig/help/sprache`
- Positionen, die über die Variable DTSPSYSHELP angegeben werden
- Positionen, die von der Variablen DTSPSYSAPPHOSTS abgeleitet werden
- `/usr/dt/appconfig/help/sprache`

Positionen der Suchpfade

Die Ausgabevariablen enthalten sowohl Einträge für sprachenspezifische Positionen wie auch für Standardpositionen (C).

Beispielsweise lautet der Standardanwendungssuchpfad:

```

Home-Verzeichnis/.dt/appmanager
/etc/dt/appconfig/appmanager/sprache
/etc/dt/appconfig/appmanager/C
/usr/dt/appconfig/appmanager/sprache
/usr/dt/appconfig/appmanager/C

```

wobei *sprache* den Wert der Umgebungsvariablen LANG darstellt.

Für jeden Bereich (systemweit oder integriert) hat die sprachenspezifische Position Vorrang vor der Standardposition.

Einführung in Aktionen und Datentypen



Bei *Aktionen* und *Datentypen* handelt es sich um wichtige Komponenten bei der Integrierung von Anwendungen auf dem Desktop. Sie stellen eine Möglichkeit zur Verfügung, eine Benutzerschnittstelle zum Starten von Anwendungen und zum Bearbeiten von deren Datendateien zu benutzen.

<i>Einführung in Aktionen</i>	142
<i>Einführung in Datentypen</i>	150

In diesem Kapitel wird das Konzept von Aktionen und Datentypen erläutert. Es wird beschrieben,

- warum es günstig ist, Aktionen und Datentypen für Anwendungen zu erstellen.
- welche Beziehung zwischen Aktionen und Datentypen besteht.
- in welcher Beziehung Aktionen und Datentypen zum Drucken über das Desktop stehen.

Weiterführende Informationen

Die Verfahren und Regeln für das Erstellen von Aktionen und Datentypen werden in diesem Handbuch in drei Kapiteln behandelt:

- In Kapitel 9 wird erläutert, wie Aktionen und Datentypen mit der Desktop-Anwendung 'Aktion erstellen' erstellt werden.

Mit 'Aktion erstellen' können Aktionen und Datentypen für die meisten Anwendungen erstellt werden, ohne daß der Benutzer die Syntaxregeln für deren Definitionen lernen muß.

- In Kapitel 10 und Kapitel 11 wird erläutert, wie Aktionen und Datentypen manuell erstellt werden, indem Konfigurationsdateien erstellt und bearbeitet werden.

Wenn der Benutzer erweiterte Funktionen verwenden möchte, die durch 'Aktion erstellen' nicht unterstützt werden, ist es notwendig, daß er Aktionen und Datentypen manuell erstellt.

Einführung in Aktionen

Bei Aktionen handelt es sich um Instruktionen, mit denen Desktop-Aufgaben wie die Ausführung von Anwendungen und das Öffnen von Datendateien automatisiert werden. Aktionen funktionieren ähnlich wie Anwendungsmakros oder Programmierfunktionen. Jede Aktion verfügt über einen Namen, der für die Ausführung der Aktion verwendet wird.

Sobald eine Aktion definiert wird, kann sie dazu verwendet werden, die Desktop-Benutzerschnittstelle anzupassen, so daß Aufgaben einfacher ausgeführt werden können. Das Desktop verfügt über Benutzerschnittstellenkomponenten wie zum Beispiel Symbole, Bedienelemente im Bedienfeld und Menüelemente für Aktionen zur Verfügung zu stellen.

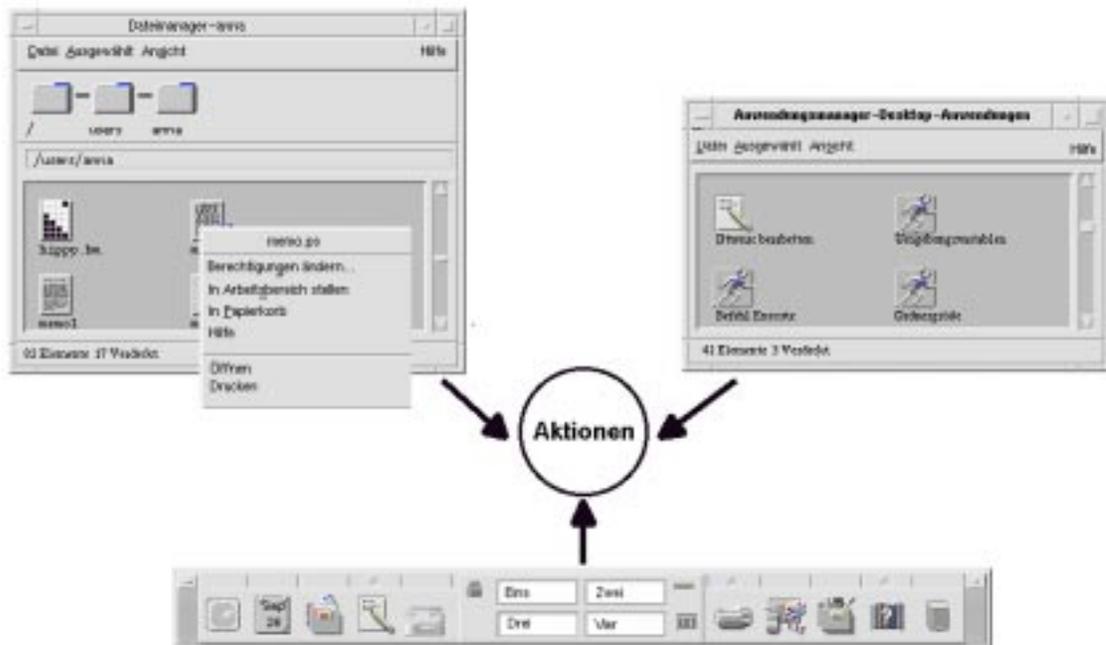


Abbildung 8-1 Verwendungsmöglichkeiten für Aktionen

Beispielsweise enthält die Gruppe von Anwendungen 'Desktop_Werkzeuge' im Anwendungsmanager Symbole, über die verschiedene Dienstprogramme gestartet werden.

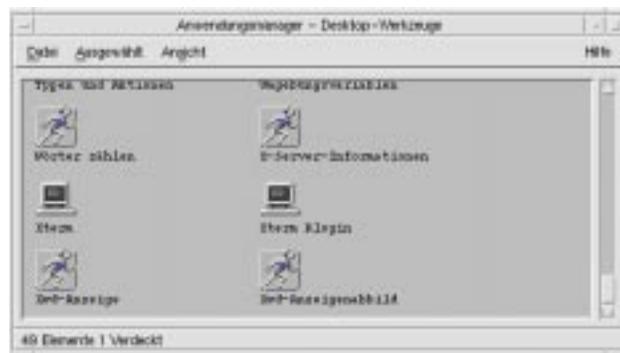


Abbildung 8-2 Symbole für Aktionen in der Gruppe von Anwendungen Desktop_Werkzeuge

Jedes dieser Symbole führt eine Aktion aus, wenn auf diesem Symbol doppelt geklickt wird. Das folgende Beispiel zeigt einen Teil der Definition der Aktion, die ausgeführt wird, wenn der Benutzer auf dem Symbol mit der Bezeichnung Xwd Display doppelt klickt. Die Aktion ist in der Konfigurationsdatei `/usr/dt/appconfig/types/sprache/xclients.dt` definiert.

```

ACTION Xwud
{
    LABEL          Xwd Display
    TYPE           COMMAND
    EXEC_STRING    /usr/bin/X11/xwud -noclick -in \
                  %(File)Arg_1"Xwd File To Display:"%
    ...
}

```

Der Befehl im Feld EXEC_STRING der Aktion wird ausgeführt, wenn der Benutzer dieses Symbol doppelklickt.

Das Bedienfeld verwendet ebenfalls Aktionen. Das folgende Beispiel zeigt einen Teil der Definition des Bedienelements mit der Bezeichnung 'Terminal' in der Bedientafel 'Persönliche Anwendungen'. Das Bedienelement ist in der Konfigurationsdatei `/usr/dt/appconfig/types/sprache/dtwm.fp` definiert.

```

CONTROL Term
{
    ICON           Fpterm
    LABEL          Terminal
    PUSH_ACTION    Dtterm
    ...
}

```

Das Feld `PUSH_ACTION` gibt an, welche Aktion ausgeführt werden soll, wenn der Benutzer das Bedienelement anklickt. In diesem Fall handelt es sich um eine Aktion mit dem Namen `Dtterm`.

Daneben werden Aktionen häufig in Menüs verwendet. Datendateien verwenden normalerweise Aktionen in ihrem Menü 'Ausgewählt' im Dateimanager. Zum Beispiel verfügen XWD-Dateien (Dateien, deren Namen die Erweiterung `.xwd` oder `.wd` haben) über eine Aktion 'Öffnen', die das Anzeigebild anzeigt, indem die Aktion 'Xwud' ausgeführt wird.

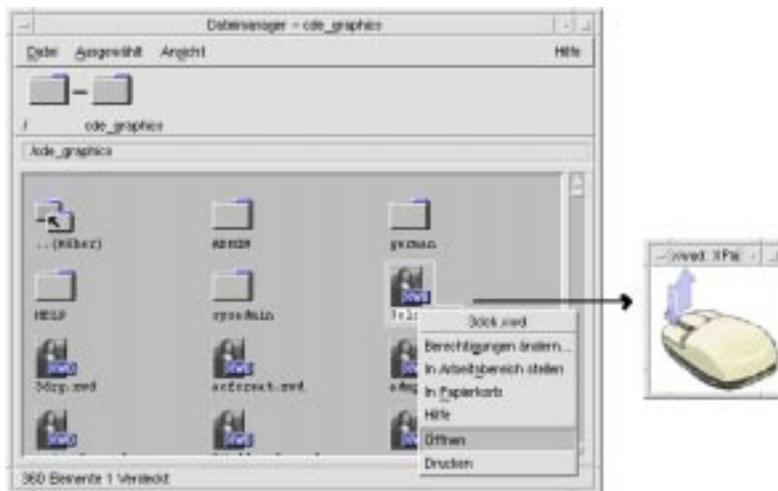


Abbildung 8-3 Aktion 'Öffnen' für Dateien des Datentyps XWD

Die Aktionen im Menü 'Ausgewählt' werden in der Datentypdefinition für XWD-Dateien angegeben. Die Definition befindet sich in der Konfigurationsdatei `/usr/dt/appconfig/types/sprache/xclients.dt`.

```
DATA_ATTRIBUTES XWD
{
    ACTIONS          Open,Print
    ICON             Dtxwd
    ...
}
```

Der Datentyp XWD und die zugehörigen Aktionen 'Öffnen' und 'Drucken', werden im Abschnitt "Wie Datentypen Datendateien mit Aktionen verbinden" auf Seite 152 erläutert.

Wie Aktionen Symbole für Anwendungen erstellen

Als Beispiel soll das Symbol Xwd Display in der Gruppe von Anwendungen Desktop_Werkzeuge betrachtet werden. Wird auf diesem Symbol doppelt geklickt, wird der X-Client `xwud` ausgeführt. Dennoch stellt dieses Symbol nicht direkt die ausführbare Datei `/usr/bin/X11/xwud` dar.

Das Symbol Xwd Display wird in der Gruppe von Anwendungen angezeigt, weil in diesem Verzeichnis eine Datei mit dem Namen `xwud` steht (siehe Abbildung 8-4). Diese Datei stellt eine zugrundeliegende Aktion mit demselben Namen, nämlich `Xwud` dar. In der Aktionsdefinition ist der Aktionsname der Name, der nach dem Schlüsselwort ACTION angegeben ist:

```
ACTION Xwud
{
    LABEL          Xwd Display
    TYPE           COMMAND
    WINDOW_TYPE    NO_STUDIO
    EXEC_STRING     /usr/bin/X11/xwud -noclick -in \
                    %(File)Arg_1"Xwd Datei anzeigen:"%
    DESCRIPTION     Die Xwd-Anzeige (Xwud) 'Aktion Xwd-Anzeige' zeigt
                    eine xwd-Datei, die mit der Aktion 'Xws Capture' (Xwd) erstellt
                    wurde. Es wird der Befehl 'xwud' verwendet.
}
```

Die Datei wird als *Aktionsdatei* bezeichnet, weil sie eine Aktion darstellt. Eine Datei ist eine Aktionsdatei, wenn es sich um eine ausführbare Datei mit demselben Namen wie eine Aktion handelt. Das zugehörige Symbol im

Anwendungsmanager (oder im Dateimanager) wird *Aktionssymbol* oder *Anwendungssymbol* genannt, weil durch Doppelklicken auf diesem Symbol eine Anwendung gestartet wird.

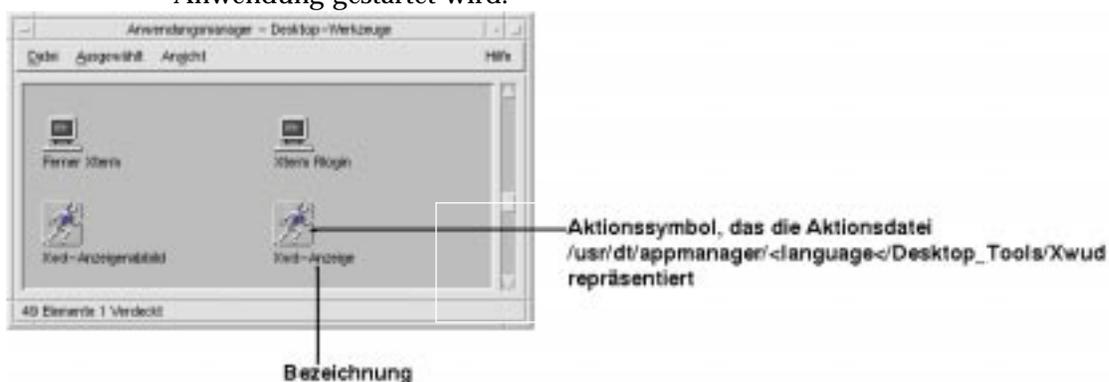


Abbildung 8-4 Anwendungs-(Aktion-)symbol, das eine Aktionsdatei darstellt

Stellt der Anwendungsmanager eine ausführbare Datei fest, durchsucht er die Datenbank für Aktionen, um festzustellen, ob Aktionen vorhanden sind, deren Namen dem Dateinamen entsprechen. Wird eine Entsprechung gefunden, weiss der Anwendungsmanager, daß es sich bei der Datei um eine Aktionsdatei handelt.

Der Inhalt der Aktionsdatei ist irrelevant. Aktionsdateien enthalten normalerweise Kommentare mit der Beschreibung ihrer Desktop-Funktionen.

Hinweis – Bei einer *Aktionsdatei* handelt es sich nicht um eine *Definitionsdatei für Aktionen*. Die *Aktionsdatei* ist eine Datei mit demselben Namen wie die Aktion. Sie wird verwendet, um das *Anwendungssymbol* im Dateimanager oder im Anwendungsmanager zu erstellen. Die *Definitionsdatei für Aktionen* ist die Datei mit dem Namen *name.dt*, die die Definition der Aktion enthält.

Sobald das Desktop festgestellt hat, daß es sich bei einer Datei um eine Aktionsdatei handelt, wird die zugrundeliegende Aktionsdefinition verwendet, um das Erscheinungsbild und das Verhalten der Aktionsdatei zu definieren.

- Im Feld `EXEC_STRING` wird das Verhalten des Anwendungssymbols angegeben. Im Fall des Symbols `Xwd Display` wird im Feld `EXEC_STRING` angegeben, daß das Aktionssymbol den X-Client `xwud` mit bestimmten Befehlszeilenargumenten ausführt.

- Im Feld LABEL wird die Bezeichnung für das Anwendungssymbol angegeben.
- Im Feld DESCRIPTION wird der Text angegeben, der angezeigt wird, wenn der Benutzer die Kontexthilfe aufruft.
- Das Anwendungssymbol Xwud verwendet das Standardsymbolbild für Aktionen, da die Aktionsdefinition kein Feld ICON enthält, in dem ein anderes Abbild definiert ist.

Im Gegensatz dazu verwendet das Symbol mit der Bezeichnung Compress File ein anderes Symbolbild, da die zugrundeliegende Aktionsdefinition ein Feld ICON enthält:

Beispiel:

```

ACTION Compress
  LABEL      Datei komprimieren
  ICON      Dtcmprs
  ...
}
  
```

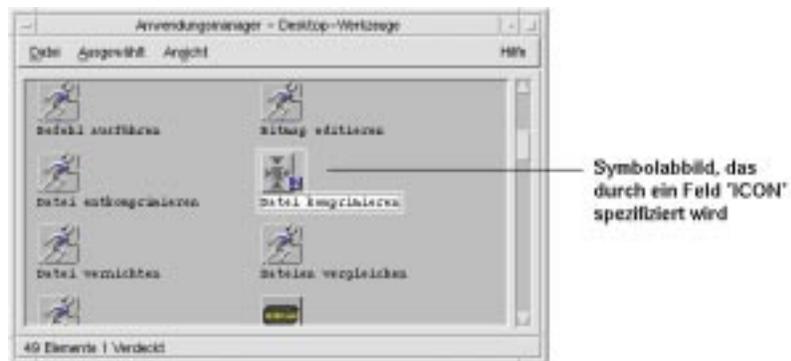


Abbildung 8-5 Symbolbild, das durch das Feld ICON in der Aktionsdefinition angegeben wird

Die Aktion Xwud wird als *Befehlsaktion* bezeichnet, da ihre Definition den auszuführenden Befehl (EXEC_STRING) enthält. Das Feld TYPE in der Aktionsdefinition definiert den Aktionstyp.

Zunächst wird das Symbol Xwd Display in der Gruppe von Anwendungen Desktop_Werkzeuge angezeigt. Der Benutzer kann jedoch zusätzliche Exemplare des Aktionssymbols in jedem Verzeichnis erstellen, für das er

Schreibberechtigung hat. Solange die Aktionsdefinition für Xwud Bestandteil der Datenbank ist, wird jede ausführbare Datei mit dem Namen `xwud`, die der Benutzer erstellt, eine Aktionsdatei sein, die diese Aktion darstellt, und das zugehörige Symbol im Dateimanager oder Anwendungsmanager kann verwendet werden, um die Aktion auszuführen.

Wie Aktionen Datendateien als Argumente verwenden

Ein *Argument* ist das Objekt, normalerweise eine Datei, an dem der Befehl ausgeführt wird. Aktionen können so definiert werden, daß sie Dateiargumente akzeptieren.

Beispielsweise gibt das Feld `EXEC_STRING` der Aktion Xwud an, daß ein Dateiargument erforderlich ist:

```
EXEC_STRING    /usr/bin/X11/xwud -noclick -in \  
               %(File)Arg_1"Xwd Datei anzeigen:"%
```

Der Term `Arg` steht für das Wort *Argument*. Die Syntax `Arg_1` bedeutet das erste Argument, und `(File)` bedeutet, daß die Aktion dieses Argument als eine Datei behandelt.

Die einfachste Weise, in der ein Benutzer ein Dateiargument zur Verfügung stellen kann, besteht darin, eine Datendatei dem Anwendungssymbol zu übergeben. Das Desktop stellt den Pfad der übergebenen Datei fest und setzt diesen in der Befehlszeile anstelle des Texts zwischen den Symbolen `%` (`%(File)Arg_1"Xwd Datei anzeigen:"%`). Dann wird folgender Befehl ausgeführt:

```
/usr/bin/X11/xwud -noclick -in datei_pfad
```

Wenn der Benutzer auf dem Anwendungssymbol doppelt klickt, stellt das Desktop im Feld `EXEC_STRING` fest, daß ein Dateiargument erforderlich ist, und zeigt ein Dialogfenster an, in dem der Benutzer aufgefordert wird, einen Dateinamen oder Pfad einzugeben. Im Fall der Aktion Xwud lautet die Aufforderung:

```
Xwd Datei anzeigen:
```

Der Dateiname oder Pfad, den der Benutzer eingibt, wird als Dateiargument verwendet.

Zusätzliche Verwendung für Aktionen

Zusätzlich zum Starten von Anwendungen werden Aktionen im gesamten Desktop verwendet, um Funktionalität in folgenden Bereichen herzustellen:

- im Bedienfeld

Die Definition für ein Bedienelement im Bedienfeld enthält Felder, in denen die Aktion angegeben wird, die ausgeführt wird, wenn der Benutzer das Bedienelement anklickt oder dem Bedienelement eine Datei übergibt. Weitere Informationen hierzu können dem Abschnitt “Bedienelemente im Bedienfeld definieren” auf Seite 251 entnommen werden

- in Menüs

Die Syntax von Fenstermenü- und Arbeitsbereichsmenüdefinitionen ermöglicht dem Benutzer, die Aktion anzugeben, die über ein Menüelement ausgeführt wird. Weitere Informationen hierzu können dem Abschnitt “Arbeitsbereichsmanagermenüs” auf Seite 271 und der Hilfeseite `dtwmrc(4)` entnommen werden.

- bei der Kommunikation zwischen Anwendungen

Eine Anwendung kann so eingerichtet werden, daß sie mit einem speziellen Typ von Aktion, nämlich ToolTalk message (`TT_MSG`), Informationen sendet und empfängt `TT_MSG`-Aktionen werden in der Entwicklerumgebungsdokumentation für das Desktop beschrieben.

Einführung in Datentypen

Erstellt der Benutzer eine neue Datendatei, kann das Erscheinungsbild und das Verhalten des Symbols der Datei im Dateimanager je nach dem Typ der Datendatei sehr unterschiedlich sein. Diese Fähigkeit, ein angepaßtes Erscheinungsbild und angepaßtes Verhalten für Dateien und Verzeichnisse zu erstellen, wird über den Datentypmechanismus des Desktop zur Verfügung gestellt.

Was ist ein Datentyp?

Bei einem Datentyp handelt es sich um ein Konstrukt, das innerhalb der Desktop-Datenbank definiert ist. Das folgende Beispiel zeigt die Definition des Datentyps `XWD`. Die Definition befindet sich in der Konfigurationsdatei `/usr/dt/appconfig/types/sprache/xclients.dt`.

```
DATA_ATTRIBUTES XWD
{
    ACTIONS          Open,Print
    ICON             Dtxwd
    NAME_TEMPLATE    %s.xwd
    MIME_TYPE        application/octet-stream
    SUNV3_TYPE       xwd-file
    DESCRIPTION      Diese Datei enthält ein Grafikbild im Format XWD.
Diese Dateien werden erstellt, wenn mit der Aktion 'XwdCapture'
Bilder von Fensterausschnitten gemacht werden. Der Datentyp wird XWD
genannt. XWD-Dateinamen enden mit '.xwd' oder '.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
}
```

Jede Datentypdefinition besteht aus zwei Teilen:

DATA_ATTRIBUTES beschreibt das Erscheinungsbild und das Verhalten des Datentyps.

DATA_CRITERIA, gibt an, nach welchen Regeln (Name oder Inhalt) eine Datei diesem Datentyp zugerechnet wird.

Das Feld **DATA_ATTRIBUTES_NAME** verbindet die Kriterien mit den Attributen.

Einem **DATA_ATTRIBUTE** können mehrere **DATA_CRITERIA** zugeordnet sein. Beispielsweise sind für den Datentyp XWD zwei verschiedene Namenskriterien (**NAME_PATTERN**) definiert, nämlich Namen, die in **.xwd** oder in **.wd** enden.

Wie Datentypen Datendateien mit Aktionen verbinden

Als Beispiel soll der Datentyp XWD betrachtet werden. Der Benutzer erstellt eine Datei mit dem Datentyp XWD, indem er der Datei eine der zwei Dateinamenerweiterungen `.xwd` oder `.wd` gibt. Das Desktop verwendet den Dateinamen als das *Kriterium*, nach dem es eine Datei einem Datentyp zuordnet.

Der Datentyp XWD stellt für jede Datei dieses Datentyps folgendes zur Verfügung:

- Ein eindeutiges Symbol, das Benutzern hilft, die Datendateien zu erkennen.
- Kontexthilfe mit Informationen über den Datentyp.
- Ein angepaßtes Menü 'Ausgewählt' im Dateimanager, das die Aktionen 'Öffnen' und 'Drucken' enthält. Die Aktion 'Öffnen' für XWD-Dateien führt die Aktion Xwud aus.

Aktionen über das Menü 'Ausgewählt' ausführen

Das Menü 'Ausgewählt' im Dateimanager ist nur aktiv, wenn eine Datei oder ein Verzeichnis ausgewählt ist. Die Befehle, die unten im Menü 'Ausgewählt' angezeigt werden, sind abhängig vom Datentyp. Wird beispielsweise eine XWD-Datei ausgewählt, enthält das Menü 'Ausgewählt' die Elemente 'Öffnen' und 'Drucken'.

Das Feld `ACTIONS` in der Datentypdefinition gibt an, welche Befehle im Menü 'Ausgewählt' des Datentyps unten angefügt werden.

```
DATA_ATTRIBUTES XWD
{
    ACTIONS          Open,Print
    ...
}
```

Der Inhalt des Menüs 'Ausgewählt' ist abhängig vom Datentyp. Aber viele verschiedene Datentypen stellen eine Aktion 'Öffnen' zur Verfügung. Wird eine Datei dieses bestimmten Datentyps im Dateimanager ausgewählt und das Menü 'Ausgewählt' angezeigt, wird im Menü der Befehl 'Öffnen' angezeigt.

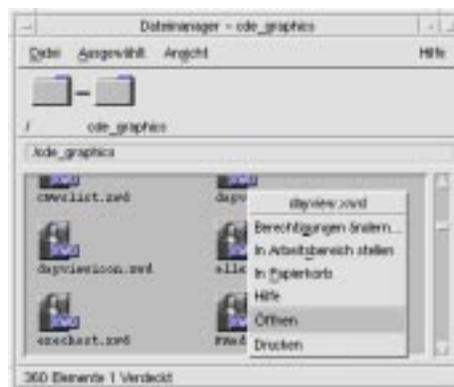


Abbildung 8-6 Das Menü 'Ausgewählt' für eine XWD-Datei

Die Aktion 'Öffnen' führt normalerweise die der Datendatei zugehörige Anwendung aus. Beispielsweise wird durch das Öffnen einer XWD-Datei die Aktion Xwud ausgeführt, die wiederum den X-Client xwud ausführt, um das Anzeigeabbild anzuzeigen. Das heißt, bei dem Datentyp XWD ist die Aktion 'Öffnen' synonym mit der Aktion Xwud. Ebenso wird durch das Öffnen einer Datei vom Datentyp TEXTFILE der Texteditor und durch das Öffnen einer Bitmap- oder Pixmap-Datei der Symboleditor ausgeführt.

Die Möglichkeit, verschiedene Aktionen 'Öffnen' zu erstellen, die unterschiedlich verwendet werden, basiert auf zwei Merkmalen von Aktionsdefinitionen:

- Aktionszuordnung

Die Aktionszuordnung ermöglicht dem Benutzer, eine Aktion zu erstellen, die eine andere Aktion ausführt, anstatt einen Befehl direkt auszuführen. Beispielsweise kann eine Aktion 'Öffnen' erstellt werden, die die Aktion Xwud zuordnet (ausführt).

- Datentypeinschränkungen für eine Aktion

Aktionsdefinitionen können ein Feld `ARG_TYPE` enthalten, das die Aktion auf bestimmte Datentypen begrenzt. Beispielsweise kann angegeben werden, daß die Aktion 'Öffnen', die die Aktion `Xwud` zuordnet, nur auf Dateien des Datentyps `XWD` angewendet wird.

Das folgende Beispiel zeigt die Definition der Aktion, die die Aktion 'Öffnen' für den Datentyp `XWD` der Aktion `Xwud` zuordnet. Sie befindet sich in der Datenbankkonfigurationsdatei

`/usr/dt/appconfig/types/C/xclients.dt:`

```
ACTION Open
{
    LABEL      Öffnen
    ARG_TYPE   XWD
    TYPE       MAP
    MAP_ACTION Xwud
}
```

In dem Feld `TYPE` wird angegeben, daß es sich um eine Zuordnungsaktion (`map`) handelt. Das Feld `MAP_ACTION` gibt an, daß diese Aktion die Aktion `Xwud` ausführt. Das Feld `ARG_TYPE` gibt an, daß diese Aktion nur auf Dateien vom Datentyp `XWD` angewendet wird.

Hierzu zum Vergleich die Definition der Aktion 'Öffnen' in der Datenbankdatei `/usr/dt/appconfig/types/C/dt.dt:`

```
ACTION Open
{
    LABEL      Öffnen
    ARG_TYPE   BM
    TYPE       MAP
    MAP_ACTION Dticon
}
```

Diese Definition bezieht sich auf Dateien des Datentyps (`ARG_TYPE`) `BM` (Bitmap-Dateien). Die Definition ordnet die Aktion 'Öffnen' der Aktion `Dticon` zu, die den Symbol-Editor ausführt.

Verhalten des Datentyps bei Doppelklicken definieren

Das Verhalten des Datentyps bei Doppelklicken wird durch den ersten Eintrag im Feld `ACTIONS` bestimmt. Beispielsweise wird beim Datentyp `XWD` durch Doppelklicken die Aktion 'Öffnen' ausgeführt, die ihrerseits die Aktion `Xwud` ausführt.

Eine Datendatei an ein Aktionssymbol übergeben

Übergibt der Benutzer eine Datendatei an ein Aktionssymbol, führt das System die Aktion aus und verwendet diese Datendatei als das Argument für die Aktion. (Weitere Informationen können dem Abschnitt “Wie Aktionen Datendateien als Argumente verwenden” auf Seite 149 entnommen werden.)

Wird beispielsweise eine XWD-Datendatei an ein Symbol Xwd Display übergeben, wird die Aktion Xwd ausgeführt und das Datendateiargument verwendet. Dadurch wird der X-Client `xwd` mit dieser Datendatei ausgeführt.

Drucken über das Desktop für einen Datentyp einrichten

Über das Desktop bestehen folgende Möglichkeiten, eine Datendatei zu drucken:

- über den Befehl 'Drucken' (falls verfügbar) im Menü 'Ausgewählt' des Dateimanagers.
- durch das Übergeben einer Datendatei an einen Desktop-Druckerübergabebereich (das Druckersymbol im Bedienfeld oder ein Druckersymbol im Druckmanager).

Zusätzlich zum Drucken über das Desktop stellen viele Anwendungen noch eine Möglichkeit zur Verfügung, von innerhalb der Anwendung zu drucken.

Beim Drucken über Desktop werden Aktionen mit dem Namen 'Drucken' verwendet. Bei 'Drucken' handelt es sich, wie bei 'Öffnen', um einen Aktionsnamen, der für viele verschiedene Datentypen verwendet wird. Aus diesem Grund werden bei den Aktionen 'Drucken' die Aktionszuordnung und das Feld `ARG_TYPE` verwendet, um das Drucken für jeden einzelnen Datentyp anzupassen.

Als Beispiel soll die Aktion 'Drucken' für den Datentyp XWD betrachtet werden. Die Definition befindet sich im Verzeichnis `/usr/dt/appconfig/types/sprache/xclients.dt`:

```
ACTION Print
{
    LABEL           Drucken
    ARG_TYPE        XWD
    TYPE            MAP
    MAP_ACTION      NoPrint
}
```

Diese Aktion 'Drucken', die sich speziell auf XWD-Dateien bezieht, ist einer Aktion 'NoPrint' zugeordnet. Bei der Aktion NoPrint handelt es sich um eine spezielle Aktion, die in `/usr/dt/appconfig/types/sprache/dt.dt` definiert ist. Die Aktion 'NoPrint' zeigt ein Dialogfenster an, das dem Benutzer mitteilt, daß dieser Datentyp nicht gedruckt werden kann.

Hierzu zum Vergleich die Aktion 'Drucken' für PCL-Dateien:

```
ACTION Print
    LABEL      Drucken
    ARG_TYPE   PCL
    TYPE       MAP
    MAP_ACTION PrintRaw
}
```

Die Aktion 'PrintRaw', die in der Konfigurationsdatei `/usr/dt/appconfig/types/sprache/print.dt` definiert ist, enthält die Befehlszeile zum Drucken von PCL-Dateien.

```
ACTION PrintRaw
{
    TYPE          COMMAND
    WINDOW_TYPE   NO_STDIO
    EXEC_STRING   /usr/dt/bin/dt1p -w %(File)Arg_1%
}
```

Aktionen und Datentypen mit 'Aktion erstellen' erstellen



Bei 'Aktion erstellen' handelt es sich um ein Werkzeug zum Erstellen

- einer Aktion, mit der eine Anwendung gestartet wird.
- eines oder mehrerer Datentypen für die Datendateien einer Anwendung.
- von Aktionen, mit denen die Datendateien einer Anwendung geöffnet und gedruckt werden.

'Aktion erstellen' kann auch dazu verwendet werden, einfache Aktionen zu erstellen, mit denen Betriebssystembefehle und Shell-Prozeduren ausgeführt werden.

<i>Funktionsweise von 'Aktion erstellen'</i>	<i>157</i>
<i>Einschränkungen von 'Aktion erstellen'</i>	<i>158</i>
<i>Aktionen und Datentypen für eine Anwendung mit 'Aktion erstellen' erstellen</i>	<i>160</i>

Weiterführende Informationen können der Hilfeseite `dtcreate(1X)` entnommen werden.

Funktionsweise von 'Aktion erstellen'

'Aktion erstellen' besteht aus einem Hauptfenster und einer Reihe von Dialogfenstern, über die eine Aktion und die zugehörigen Datentypen erstellt werden.

Das Werkzeug 'Aktion erstellen'

- erstellt eine Aktionsdefinition, die einen Befehl ausführt.
- erstellt eine Datei *Home-Verzeichnis/.dt/types/aktion_name.dt*. Diese Datei speichert die Definitionen für Aktion und Datentyp, die für die Anwendung erstellt wurden.
- erstellt eine *Aktionsdatei* im Home-Verzeichnis des Benutzers. Bei der Aktionsdatei handelt es sich um eine ausführbare Datei mit demselben Namen wie die Aktion.

Die Darstellung der Aktionsdatei im Dateimanager wird als *Anwendungssymbol* bezeichnet, da man durch Doppelklicken auf diesem Symbol die Anwendung starten kann.

Wahlweise kann der Benutzer das Aktionssymbol als Übergabebereich einrichten, indem er beim Erstellen der Aktion für die Übergabe geeignete Datentypen angibt.

- erstellt einen oder mehrere Datentypen für die Datendateien einer Anwendung (optional).
- erstellt eine Aktion 'Öffnen' für jeden Datentyp.
- erstellt eine Aktion 'Drucken' für jeden Datentyp (optional).
- lädt die Datenbank für Aktionen und Datentypen erneut. Auf diese Weise werden Aktionen und Datentypen sofort wirksam.

Einschränkungen von 'Aktion erstellen'

'Aktion erstellen' dient dazu, Aktionen und Datentypen für die Ausführung von Anwendungen zu erstellen. Aber Aktionen und Datentypen sind sehr flexibel und enthalten zusätzliche Funktionen, auf die nur zugegriffen werden kann, wenn die Definitionen manuell erstellt werden.

Weitere Informationen hierzu können folgenden Kapiteln entnommen werden:

- Kapitel 10, "Aktionen manuell erstellen"
- Kapitel 11, "Datentypen manuell erstellen"

Einschränkungen bei Aktionen

'Aktion erstellen' kann nicht verwendet werden, um die Aktion für eine Anwendung zu erstellen, wenn eine der folgenden Bedingungen zutrifft:

- Für die Befehlszeile ist ein Parameter (ein Argument, das 'Aktion erstellen' nicht der Aktionsdatei entnehmen kann) notwendig.

Beispielsweise kann 'Aktion erstellen' nicht verwendet werden, um eine Aktion für folgenden Befehl zu schreiben:

```
lp -dgerät dateiname
```

da der Benutzer bei jeder Ausführung des Befehls einen Wert für *gerät* angeben muß.

- Das Anwendungssymbol muß eine andere Bezeichnung tragen als der Aktionsname.

Beispielsweise kann 'Aktion erstellen' nicht verwendet werden, um eine Version einer bereits vorhandenen Aktion in der lokalen Sprache zu erstellen.

- Für die Aktion ist eine der erweiterten Funktionen der Aktionsdatenbank erforderlich.

Beispiele für diese erweiterten Funktionen sind Aktionen, die

- Befehle auf fernen Systemen, die nicht über die Aktionsdefinition verfügen, starten
- andere Aktionen aufrufen
- mit einer anderen Berechtigung ausgeführt werden müssen (zum Beispiel als Superuser)
- mit der Zuordnungsfunktion (map) ausgeführt werden
- je nach Anzahl der Dateiarumente, die die Aktion zur Verfügung stellt, ein sehr unterschiedliches Verhalten zeigen

Einschränkungen bei Datentypen

'Aktion erstellen' kann nicht verwendet werden, um den Datentyp für eine Anwendung zu erstellen, wenn eine der folgenden Bedingungen zutrifft:

- Für den Datentyp müssen neben 'Öffnen' und 'Drucken' noch weitere Aktionen definiert sein.

- Bei der Aktion 'Öffnen' für den Datentyp handelt es sich nicht um den Befehl der Aktion.

Beispielsweise kann 'Aktion erstellen' nicht verwendet werden, um den Datentyp zu erstellen, mit dem ein eindeutiges Symbol für das Verzeichnis, das die Gruppe von Anwendungen der Anwendung darstellt, zur Verfügung gestellt wird.

Aktionen und Datentypen für eine Anwendung mit 'Aktion erstellen' erstellen

Bevor 'Aktion erstellen' ausgeführt wird, muß der Benutzer einige Informationen über die Anwendung zur Verfügung haben.

- Die Befehlszeile, mit der die Anwendung gestartet wird.

Der Benutzer muß wissen, ob die Befehlszeile ein erforderliches Dateiargument, ein wahlfreies Dateiargument oder kein Dateiargument enthält.

Ist für die Anwendung ein Parameter erforderlich, kann die Aktion nicht mit 'Aktion erstellen' erstellt werden.

- Die Typen von Datendateien, die die Anwendung akzeptieren kann.

Einige Anwendungen können nur einen Datentyp akzeptieren. Andere (zum Beispiel ein ASCII-Editor oder ein Grafikeditor) können verschiedene Datentypen akzeptieren.

- Wie die Anwendung ihre Datendateien identifiziert.

Hierbei kann es sich um eine Benennungskonvention (beispielsweise Dateinamen mit der Erweiterung `.doc`) handeln und/oder vom Inhalt der Datei abhängig sein. Verwendet die Anwendung keine Dateinamenskönvention, kann für das Aktionssymbol dennoch eine Konvention eingerichtet werden.

- Wahlfrei: Die Befehlszeile für den Druck der Dateien

▼ Eine Aktion für eine Anwendung erstellen

1. Auf 'Aktion erstellen' in der Gruppe Desktop-Anwendungen doppelklicken.

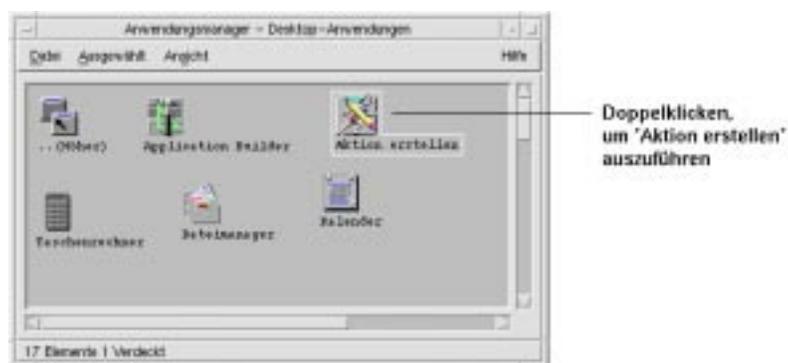


Abbildung 9-1 Das Symbol 'Aktion erstellen' im Anwendungsmanager

Hierdurch wird das Hauptfenster 'Aktion erstellen' angezeigt.



Abbildung 9-2 Hauptfenster 'Aktion erstellen'

2. Den Namen für das Aktionssymbol in das Textfeld 'Aktionsname' eingeben.
3. Die Bedienelemente für 'Aktionssymbole' verwenden, um das Symbol für die Anwendung anzugeben. Zu Beginn wird das Standardsymbol angezeigt.
 - Soll ein anderes, bereits vorhandenes Symbol ausgewählt werden, muß der Benutzer 'Gruppe suchen' anklicken, damit das Dialogfenster 'Gruppe suchen' angezeigt wird. Weitere Informationen hierzu können dem Abschnitt "Symbol über das Dialogfenster 'Gruppe suchen' angeben" auf Seite 171 entnommen werden.
 - Sollen neue Symbole erstellt werden, 'Symbol bearbeiten' auswählen, um den Symboleditor auszuführen.
4. In dem Textfeld 'Befehl beim Öffnen der Aktion (durch Doppelklicken)' den Befehl zum Starten der Anwendung eingeben.

Für ein Dateiarargument die Syntax $\$n$ verwenden. Beispiel:

```
emacs
bitmap $1
diff $1 $2
lp -oraw $1
```

Enthält die Befehlszeile ein Dateiargument ($\$n$), ist das Aktionssymbol ein Übergabebereich für Dateien.

Die Befehlszeilen werden nicht an eine Shell übergeben, es sei denn, die Verwendung einer Shell wird explizit angegeben. Zum Beispiel verwenden folgende Zeilen Shell-Verarbeitung:

```
/bin/sh -c 'ps | lp'
/bin/sh -c 'spell $1 | more'
```

5. Den Kontexthilfetext für das Aktionssymbol im Textfeld 'Hilfetext für Aktionssymbol' eingeben.

Der Text wird im Textfeld automatisch umgebrochen. Diese Zeilenumbrüche werden jedoch online nicht beibehalten. Soll ein fester Zeilenumbruch angegeben werden, muß `\n` verwendet werden.

6. Die für die Aktion erforderliche Fensterunterstützung im Menü 'Optionen' 'Fensterart' auswählen.

Grafisch (X-Window)	Die Anwendung erstellt ihr eigenes Fenster.
Terminal (automatisches Schließen)	Die Anwendung wird in einem Terminal-Emulator-Fenster ausgeführt, das automatisch geschlossen wird, wenn der Benutzer die Anwendung verläßt
Terminal (manuelles Schließen)	Die Anwendung wird in einem Terminal-Emulator-Fenster ausgeführt, das geöffnet bleibt, bis der Benutzer es explizit schließt.
Keine Ausgabe	Die Anwendung zeigt keine Ausgabe am Bildschirm an.

7. Folgendermaßen vorgehen:

- Verfügt die Anwendung über Datendateien und möchte der Benutzer einen oder mehrere Datentypen für diese erstellen, kann er weitere Informationen hierzu dem Abschnitt, "Einen oder mehrere Datentypen für eine Anwendung erstellen" entnehmen.

- Soll kein Datentyp erstellt werden, die Aktion sichern, indem 'Speichern' im Menü 'Datei' ausgewählt wird. Anschließend die neue Aktion durch Doppelklicken auf ihrem Symbol im Home-Verzeichnis des Benutzers testen.

▼ Einen oder mehrere Datentypen für eine Anwendung erstellen

1. Die Aktion für die Anwendung unter Verwendung der Prozedur im vorherigen Abschnitt, "Eine Aktion für eine Anwendung erstellen" definieren.
2. Die Taste 'Erweitert' im Fenster 'Aktion erstellen' anklicken, um das Fenster zu erweitern.



Abbildung 9-3 Erweiterte Funktionen im Hauptfenster 'Aktion erstellen'

3. Soll das Anwendungssymbol nach einem Dateiarargument fragen, nachdem das Symbol doppelgeklickt wurde, den Text der Eingabeaufforderung in dem Textfeld 'Beim Öffnen einer Aktion Benutzer nach folgendem fragen:' eingeben.

Folgende Richtlinien gelten für dieses Textfeld:

- Dieses Feld muß verwendet werden, falls die Befehlszeile der Anwendung ein *erforderliches* Dateiarargument enthält.
- Enthält die Befehlszeile kein Dateiarargument, bleibt das Feld leer.

-
- Ist das Dateiarargument in der Befehlszeile der Anwendung optional, wird, falls ein Text für die Eingabeaufforderung eingegeben wurde, beim Doppelklicken auf dem Symbol nach dem Dateiarargument gefragt. Wurde kein Text für die Eingabeaufforderung eingegeben, wird die Aktion ausgeführt und eine leere Zeichenfolge als Dateiarargument verwendet.
4. Die Dateitypen angeben, die die Aktion als Argumente akzeptiert:
- Akzeptiert die Aktion jeden Datentyp, 'Alle Datentypen' auswählen.
 - Akzeptiert die Aktion nur den/die Datentyp(en), der/die für die Anwendung erstellt wurde(n), 'Nur obige Liste' auswählen.

Zu Beginn ist die Liste 'Datentypen, die diese Aktion verwenden' leer. Erstellt der Benutzer Datentypen für die Anwendung, werden sie zu der Liste hinzugefügt.

- Die Taste 'Hinzufügen' neben dem Fenster mit der Liste 'Datentypen, die diese Aktion verwenden' anklicken, um das Dialogfenster 'Datentyp hinzufügen' anzuzeigen.



Abbildung 9-4 Dialogfenster 'Datentyp Hinzufügen' in 'Aktion erstellen'

- Optional: Soll der Standarddatentypname nicht verwendet werden, einen neuen Namen für den Datentyp im Textfeld 'Name der Datentypfamilie' eingeben.

In dem Namen dürfen keine Leerzeichen vorkommen. Der Datentypname ist für Benutzer der Anwendung nicht sichtbar. Er wird in der Datenbank für Aktionen und Datentypen verwendet, um die Definition für den Datentyp zu identifizieren.

7. Die Taste 'Bearbeiten' neben dem Fenster 'Identifizierende Merkmale' anklicken, um das Dialogfenster 'Identifizierende Merkmale' anzuzeigen.

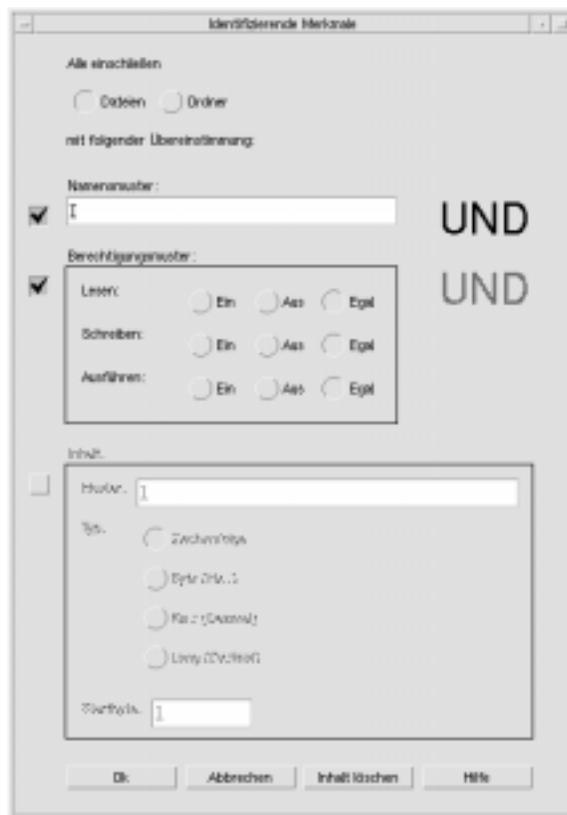


Abbildung 9-5 Dialogfenster 'Identifizierende Merkmale' in 'Aktion erstellen'

Datentypen werden anhand ihrer Merkmale voneinander unterschieden. Der Benutzer kann eines oder mehr der folgenden Kriterien auswählen:

Dateien oder Ordner	Der Datentyp gilt nur für Dateien oder nur für Ordner
Namensmuster	Datentypzuordnung auf der Grundlage des Dateinamens.
Berechtigungsuster	Lese-, Schreib- und Ausführungsberechtigungen
Inhalt	Inhalt eines angegebenen Teils einer Datei

8. Auswählen, ob der Datentyp eine Datei oder einen Ordner darstellt.



Abbildung 9-6 Das Merkmal für Datei oder Verzeichnis für einen Datentyp angeben.

9. Ist die Zuordnung eines Datentyps vom Namen abhängig, das Markierungsfeld 'Namensmuster' auswählen und den Namen im Textfeld eingeben.



Abbildung 9-7 Das Merkmal Dateiname für einen Datentyp angeben.

Hierbei können * und ? als Platzhalterzeichen verwendet werden:

- * Entspricht einer beliebigen Folge von Zeichen
- ? Entspricht einem beliebigen einzelnen Zeichen

10. Ist die Zuordnung eines Datentyps von den Berechtigungen abhängig, das Markierungsfeld 'Berechtigungs muster' auswählen und die Berechtigungen für den Datentyp auswählen..

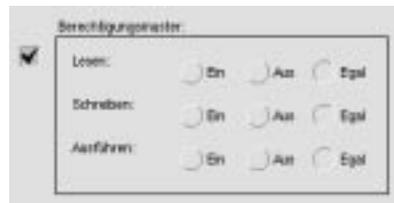
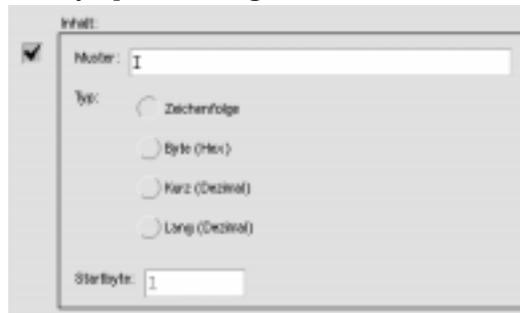


Abbildung 9-8 Das Merkmal Zugriffsrechte für einen Datentyp angeben

- Ein Für die Datei muß die angegebene Berechtigung vorliegen
- Aus Für die Datei muß die angegebene Berechtigung fehlen
- Egal Die angegebene Berechtigung ist unwichtig

11. Ist die Zuordnung eines Datentyps vom Inhalt abhängig, das Markierungsfeld 'Inhalt' auswählen und die erforderlichen Informationen, Suchmuster und Typ des Inhalts, eingeben. Wahlweise kann der Benutzer die Byteposition angeben, an der die Suche starten soll..



The image shows a dialog box titled "Inhalt". It has a checked checkbox in the top left corner. Below the checkbox is a text field labeled "Muster:" containing the character "I". Underneath is a section labeled "Typ:" with four radio button options: "Zeichenfolge", "Byte (Hex)", "Kurz (Decimal)", and "Lang (Decimal)". At the bottom of the dialog is another text field labeled "StartByte:" containing the number "1".

Abbildung 9-9 Das Merkmal Inhalt für einen Datentyp angeben

Hinweis – Die Datentypzuordnung auf der Grundlage des Inhalts beeinträchtigt möglicherweise die Leistung des Systems.

12. 'OK' anklicken, um das Dialogfenster 'Merkmale identifizieren' zu schließen.

Die Merkmale werden im Feld 'Merkmale identifizieren' mit folgender Codierung angezeigt:

- d Verzeichnis
- r Für die Datei liegt Lesezugriff vor
- w Für die Datei liegt Schreibzugriff vor
- x Für die Datei liegt Ausführungszugriff vor
- ! Logischer Operator NICHT
- & Logischer Operator UND

13. Den Hilfetext für die Datendateien in das Textfeld 'Hilfetext' eingeben.
14. Die Bedienelemente für 'Aktionssymbole' verwenden, um das Symbol für die Anwendung anzugeben. Zu Beginn wird das Standardsymbol angezeigt.

- Soll ein anderes, bereits vorhandenes Symbol ausgewählt werden, muß der Benutzer 'Gruppe suchen' anklicken, damit das Dialogfenster 'Gruppe suchen' angezeigt wird. Weitere Informationen hierzu können dem Abschnitt "Symbol über das Dialogfenster 'Gruppe suchen' angeben" auf Seite 171 entnommen werden.
 - Sollen neue Symbole erstellt werden, 'Symbol bearbeiten' auswählen, um den Symboleditor auszuführen.
15. Den Befehl im Textfeld 'Befehl zum Öffnen dieses Datentyps' überprüfen. Dieser Befehl wird ausgeführt, wenn der Benutzer eine Datendatei doppelt anklickt.
 16. Optional: Stellt der Anwendung einen Druckbefehl für das Drucken von Datendateien über die Befehlszeile zur Verfügung, den Befehl in das Textfeld 'Befehl zum Drucken dieses Datentyps' unter Verwendung der Syntax $\$n$ für ein Dateiarargument einzugeben.
 17. Die Datentypdefinition auf eine der folgenden Weisen sichern:
 - 'OK' anklicken, um den Datentyp zu sichern, und das Dialogfenster 'Datentyp hinzufügen' schließen.
 - 'Anwenden' anklicken, um den Datentyp ohne Schließen des Dialogfensters 'Datentyp hinzufügen' zu sichern. Der Benutzer kann sofort mit der Definition eines weiteren Datentyps für die Aktion fortfahren.

Symbol über das Dialogfenster 'Gruppe suchen' angeben

Das Dialogfenster 'Gruppe suchen' wird angezeigt, wenn der Benutzer 'Gruppe suchen' im Hauptfenster 'Aktion erstellen' oder im Fenster 'Datentyp hinzufügen' anklickt. Dieses Dialogfenster wird verwendet, um das Symbol anzugeben, das für die Aktion oder den Datentyp verwendet wird.

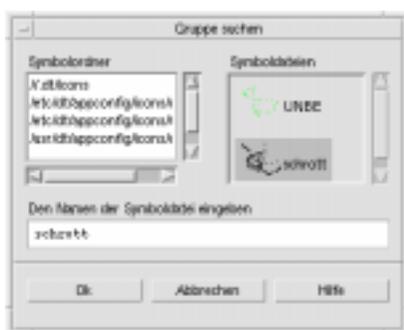


Abbildung 9-10 Dialogfenster 'Gruppe suchen'

Das Dialogfenster 'Gruppe suchen' ermöglicht dem Benutzer, eine Gruppe von Symbolabbilddateien in folgenden Positionen zu erstellen:

- In einem Verzeichnis im Symbolsuchpfad. Die Liste 'Symbolordner' enthält alle Verzeichnisse im Symbolsuchpfad.
- In einem Registrierungspaket, das mit dem Desktop mit `dtappintegrate` integriert wird. Diese Symbole befinden sich noch nicht in einem Verzeichnis im Symbolsuchpfad, werden aber von `dtappintegrate` dorthin gestellt.

Hinweis – Die Definitionen für Aktionen und Datentyp, die mit 'Aktion erstellen' erstellt wurden, verwenden den Dateinamen ohne Pfad für die Symboldateien (den Dateinamen ohne die Erweiterungen für Größe und Typ). Symbole für Aktionen und Datentypen, die mit 'Aktion erstellen' erstellt wurden, werden letztendlich in Verzeichnisse im Symbolsuchpfad gestellt.

▼ **Gruppen von Symbolen im Symbolsuchpfad angeben**

1. In der Liste 'Symbolordner' des Dialogfensters 'Gruppe suchen' auf dem Verzeichnispfad, der das Symbol enthält, doppelklicken.

Die Liste 'Symboldateien' zeigt alle Symboldateien in diesem Ordner an.

2. In der Liste 'Symboldateien' das Symbol anklicken, das verwendet werden soll.

Dadurch wird der Dateiname ohne Pfad der Symboldatei in das Textfeld 'Den Namen der Symboldatei eingeben' gestellt.

3. 'OK' anklicken.

▼ **Symbole in einem Registrierungspaket angeben**

Handelt es sich bei dem Benutzer um einen Systemadministrator oder um einen Programmierer, der ein Registrierungspaket erstellt, werden die Symbolabbilddateien zunächst in ein Verzeichnis im Registrierungspaket gestellt:

`app_root/dt/appconfig/icons/sprache`

Nach der Registrierung mit `dtappintegrate` werden die Symboldateien in das Verzeichnis `/etc/dt/appconfig/icons/sprache`, das sich im Symbolsuchpfad befindet, kopiert.

Diese Prozedur verwenden, um Symbole anzugeben, die Bestandteil eines Registrierungspakets sind:

1. Im Textfeld 'Den Namen der Symboldatei eingeben' des Dialogfensters 'Gruppe suchen' den Dateinamen ohne Pfad der Symboldatei eingeben.
2. 'OK' anklicken.

'Aktion erstellen' zeigt ein Dialogfenster an, in dem der Benutzer darüber informiert wird, daß die Symbole in Verzeichnissen im Symbolpfad nicht gefunden wurden.

3. In dem Dialogfenster 'Information', das angezeigt wird, 'Keine Änderung' auswählen.

Es gibt zwei Möglichkeiten, Aktionen zu erstellen:

- mit der Desktop-Anwendung 'Aktion erstellen'
- durch das manuelle Erstellen einer Aktionsdefinition

Um eine Aktion manuell zu erstellen, ist es erforderlich, eine Datenbankdatei zu bearbeiten. In diesem Kapitel wird beschrieben, wie Aktionsdefinitionen manuell erstellt werden.

<i>Gründe, eine Aktion manuell zu erstellen</i>	<i>174</i>
<i>Aktionen manuell erstellen: Allgemeine Schritte</i>	<i>175</i>
<i>Ausführungszeile für eine Befehlsaktion erstellen</i>	<i>185</i>
<i>Fensterunterstützung und Terminal-Emulatoren für Befehlsaktionen</i>	<i>193</i>
<i>Aktionen auf bestimmte Argumente einschränken</i>	<i>195</i>
<i>Aktionen auf bestimmte Argumente einschränken</i>	<i>195</i>
<i>Variablen in Definitionen für Aktionen und Datentypen verwenden</i>	<i>199</i>
<i>Aktionen über eine Befehlszeile aufrufen</i>	<i>200</i>
<i>Sprachenabhängige Aktionen erstellen</i>	<i>202</i>
<i>Aktionen für ToolTalk-Anwendungen erstellen</i>	<i>203</i>

Weiterführende Informationen

- Eine Einführung in Aktionen kann Kapitel 8, “Einführung in Aktionen und Datentypen” entnommen werden.
- Informationen zur Verwendung von ‘Aktion erstellen’ kann Kapitel 9, “Aktionen und Datentypen mit ‘Aktion erstellen’ erstellen” entnommen werden.
- Referenzinformationen zu Aktionsdefinitionen können der Hilfeseite `dtactionfile(4)` entnommen werden.

Gründe, eine Aktion manuell zu erstellen

Es gibt drei grundlegende Arten von Aktionen:

- `COMMAND`
- `MAP`
- `TT_MSG`

Mit dem Werkzeug ‘Aktion erstellen’ können bestimmte Arten von Befehls- (`COMMAND`) und Zuordnungsaktionen (`MAP`) erstellt werden. Alle ToolTalk-Meldungsaktionen (`TT_MSG`) müssen manuell erstellt werden

Weitere Informationen können dem Abschnitt “Einschränkungen von ‘Aktion erstellen’” auf Seite 158 entnommen werden.

Befehlsaktionen

Eine *Befehlsaktion* (`COMMAND`) führt einen Befehl aus, der eine Anwendung oder ein Dienstprogramm startet, führt eine Shell-Prozedur aus oder führt einen Betriebssystembefehl aus. Die Definition der Aktion enthält den auszuführenden Befehl (`EXEC_STRING`).

Das Werkzeug ‘Aktion erstellen’ kann verwendet werden, um die gebräuchlichsten Befehlsaktionen auszuführen. Aber es gibt Situationen, in denen die Aktion manuell erstellt werden muß. Beispielsweise muß eine Befehlsaktion manuell erstellt werden, wenn folgendes für die Aktion angegeben ist:

- Mehrere Dateiangaben, mit einer anderen Eingabeaufforderung für jedes Argument.
- Aktionsaufruf —die Fähigkeit von Aktionen, andere Aktionen aufzurufen.

- Argumentanzahlabhängiges Verhalten—die Fähigkeit eine Aktion zu erstellen, die bei einer unterschiedlichen Anzahl von Dateiargumenten unterschiedliches Verhalten zeigt.
- Ein Rechner für die Ausführung auf fernen Systemen—die Fähigkeit, eine Anwendung auf einem anderen System als demjenigen, das die Aktionsdefinition enthält, auszuführen.
- Benutzeränderung—die Fähigkeit, die Aktion als ein anderer Benutzer auszuführen (zum Beispiel, sich als Benutzer 'root' anzumelden und die Aktion als 'root' auszuführen).

Zuordnungsaktionen

Bei einer *Zuordnungsaktion* (MAP) handelt es sich um eine Aktion, die einer anderen Aktion zugeordnet wird, statt direkt einen Befehl oder eine ToolTalk-Meldung anzugeben.

Die Zuordnung ermöglicht dem Benutzer, alternative Namen für Aktionen anzugeben. Beispielsweise startet eine integrierte Befehlsaktion mit dem Namen IconEditor den Symboleditor. Die Datenbank enthält auch eine Aktion 'Öffnen', die in der Definition auf Bitmap- und Pixmap-Dateien eingeschränkt ist (durch das Feld ARG_TYPE), die der Aktion IconEditor zugeordnet ist. Dadurch kann der Benutzer den Symboleditor starten, indem er eine Bitmap- oder Pixmap-Datei im Dateimanager auswählt und dann im Menü 'Ausgewählt' die Option 'Öffnen' wählt.

'Aktion erstellen' stellt eine eingeschränkte Zuordnungsfunktion für die Aktionen 'Öffnen' und 'Drucken' zur Verfügung. Alle anderen Zuordnungsaktionen müssen manuell erstellt werden

ToolTalk-Meldungsaktionen

ToolTalk-Meldungsaktionen (TT_MSG) senden eine ToolTalk-Meldung. Sämtliche ToolTalk-Meldungsaktionen müssen manuell erstellt werden

Aktionen manuell erstellen: Allgemeine Schritte

In diesem Abschnitt wird erläutert, wie eine Konfigurationsdatei für eine Aktionsdefinition erstellt wird.

Konfigurationsdateien für Aktionen

Konfigurationsdateien, die Aktionsdefinitionen enthalten, müssen folgenden Anforderungen entsprechen:

- Die Dateien müssen die Namenskonvention *name.dt* einhalten.
- Die Dateien müssen sich im Datenbanksuchpfad (für Aktionen und Datentypen) befinden. Der Standardpfad ist:

Persönliche Aktionen *Home-Verzeichnis/.dt/types*

Systemweite Aktionen */etc/dt/appconfig/types/sprache*

Integrierte Aktionen */usr/dt/appconfig/types/sprache*. Dieses Verzeichnis sollte nicht verwendet werden.

Informationen zur Änderung des Suchpfads für Aktionen und Datentypen können dem Abschnitt “Wert eines Suchpfads setzen” auf Seite 129 entnommen werden.

▼ Aktionen manuell erstellen

1. Eine bereits vorhandene Datenbankdatei öffnen oder eine neue erstellen. Siehe hierzu den vorangegangenen Abschnitt “Konfigurationsdateien für Aktionen”.
2. Die Aktionsdefinition mit folgender Syntax erstellen:

```

ACTION aktion_name
{
  TYPE      aktion_typ
  aktion_feld
  ...
}

```

wobei die Parameter folgende Bedeutung haben:

aktion_name Name, mit dem die Aktion ausgeführt wird.

aktion_typ COMMAND (Standard), MAP oder TT_MSG.

aktion_feld Eines der erforderlichen oder wahlfreien Felder für diesen Aktionstyp. Alle Felder bestehen aus einem Schlüsselwort und einem Wert.

Viele der Aktionsfelder werden in diesem Kapitel behandelt. Weitere Informationen können der Hilfeseite `dtactionfile(4)` entnommen werden.

3. Die Datei sichern.
4. Soll das Aktionssymbol über ein eindeutiges Abbild verfügen, die Symbole für die Aktion erstellen. Die Standardpositionen für Symbole sind:
 - Persönliche Symbole: *Home-Verzeichnis*/`.dt/icons`
 - Systemweite Symbole: `/etc/dt/appconfig/icons/sprache`. Der Standardwert für *sprache* ist `C`.

Weitere Informationen können dem Abschnitt “Symbol angeben, das von einem Bedienelement verwendet werden soll” auf Seite 246 entnommen werden.

5. ‘Aktionen erneut laden’ in der Gruppe von Anwendungen ‘Desktop-Werkzeuge’ doppelklicken.
6. Eine Aktionsdatei für die Aktion erstellen. Die Aktionsdatei erstellt ein Symbol im Dateimanager oder Anwendungsmanager, das die Aktion darstellt. (Dient die Aktion dazu, eine Anwendung zu starten, wird das Symbol als *Anwendungssymbol*) bezeichnet.

Um die Aktionsdatei zu erstellen, muß der Benutzer eine ausführbare Datei mit demselben Namen wie *aktion_name* erstellen. Der Benutzer kann die Datei in jedes Verzeichnis stellen, für das er Schreibberechtigung hat. Es können beliebig viele Aktionsdateien erstellt werden.

Beispiel für das Erstellen einer Befehlsaktion

Mit den folgenden Schritten wird eine persönliche Aktion erstellt, die eine Telefax-Anwendung auf einem fernen System AppServerA startet. Der Befehl zum Starten der Telefax-Anwendung lautet:

```
/usr/fax/bin/faxcompose [dateiname]
```

1. Die Datei *Home-Verzeichnis*/`.dt/types/Fax.dt` erstellen.
2. Die folgende Aktionsdefinition in die Datei einfügen:

```
ACTION FaxComposer
{
    TYPE          COMMAND
    ICON          fax
```

```

WINDOW_TYPE    NO_STDIO
EXEC_STRING    /usr/fax/bin/faxcompose -c %Arg_1%
EXEC_HOST      AppServerA
DESCRIPTION    Führt ein Faxerstellungsprogramm aus
}

```

Die Felder `WINDOW_TYPE` und `EXEC_STRING` beschreiben das Verhalten der Aktion.

<code>WINDOW_TYPE</code>	Das Schlüsselwort <code>NO_STDIO</code> gibt an, daß die Aktion nicht in einem Terminal-Emulator-Fenster ausgeführt werden muß. Weitere Informationen hierzu können dem Abschnitt "Fensterunterstützung für die Aktion angeben" auf Seite 193 entnommen werden.
<code>EXEC_STRING</code>	Die Syntax <code>%Arg_1%</code> akzeptiert eine übergebene Datei. Wird auf dem Aktionssymbol doppelgeklickt, öffnet die Aktion ein leeres Telefaxerstellungsfenster. Weitere Informationen hierzu können dem Abschnitt "Ausführungszeile für eine Befehlsaktion erstellen" auf Seite 185 entnommen werden.

3. Die Datei sichern.
4. Mit dem Symboleditor folgende Symbolbilddateien im Verzeichnis *Home-Verzeichnis/.dt/icons* erstellen:
 - `fax.m.pm`, Größe 32 auf 32 Pixel
 - `fax.t.pm`, Größe 16 auf 16 Pixel
5. 'Aktionen erneut laden' in der Gruppe von Anwendungen 'Desktop-Werkzeuge' doppelklicken.
6. Eine ausführbare Datei mit dem Namen `FaxComposer` in einem Verzeichnis, für das Schreibberechtigung besteht (zum Beispiel im Home-Verzeichnis), erstellen.

Beispiel für das Erstellen einer Zuordnungsaktion

Angenommen, die meisten Dateien, die der Benutzer telefaxt, werden mit Text Editor erstellt und sind vom Datentyp `TEXTFILE` (Dateien mit der Erweiterung `*.txt`):

Mit folgenden Schritten wird ein Menüelement 'Fax' zum Menü 'Ausgewählt' des Datentyps hinzugefügt.

1. Die Datei *Home-Verzeichnis/.dt/types/Fax.dt*, die im vorhergehenden Beispiel erstellt wurde, öffnen.

2. Die folgende Zuordnungsaktionsdefinition der Datei hinzufügen:

```
ACTION Fax
{
  ARG_TYPE   TEXTFILE
  TYPE       MAP
  MAP_ACTION FaxComposer
}
```

3. Die Datei sichern.

4. Die Definition für Datenattribute für TEXTFILE aus */usr/dt/appconfig/types/sprache/dtpad.dt* in eine neue Datei *Home-Verzeichnis/.dt/types/textfile.dt* kopieren. Die Aktion Fax zum Feld ACTIONS hinzufügen.

```
DATA_ATTRIBUTES TEXTFILE
{
  ACTIONS   Open,Print,Fax
  ICON      Dtpenpd
  ...
}
```

5. Die Datei sichern.

6. Den Anwendungsmanager öffnen und 'Aktionen erneut laden' in der Gruppe von Anwendungen 'Desktop-Werkzeuge' doppelklicken.

▼ Datenbank für Aktionen und Datentypen erneut laden

Damit neue oder bearbeitete Aktionsdefinitionen wirksam werden können, muß das Desktop die Datenbank erneut lesen.

◆ Die Gruppe von Anwendungen 'Desktop-Werkzeuge' öffnen und 'Aktionen erneut laden' doppelklicken.

◆ Oder: folgenden Befehl ausführen:

```
dtaction ReloadActions
```

'ReloadActions' ist der Name der Aktion, deren Symbol die Bezeichnung 'Aktionen erneut laden' trägt.

Die Aktionsdatenbank wird ebenfalls erneut gelesen, wenn der Benutzer

- sich anmeldet
- die Arbeitsbereichssteuerung erneuert
- eine Aktion im Fenster 'Aktion erstellen' durch Auswahl von 'Speichern' im Menü 'Datei' sichert.

Aktionsdateien (Symbole) für eine Aktion erstellen

Bei einer *Aktionsdatei* handelt es sich um eine sichtbare Darstellung der Aktion im Dateimanager oder Anwendungsmanager.

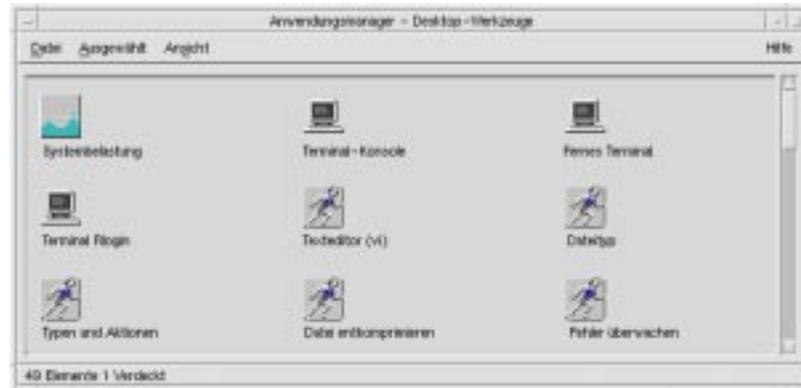


Abbildung 10-1 Aktionsdateien (auch *Aktionssymbole* oder *Anwendungssymbole* genannt) im Anwendungsmanager

Da das Symbol einer Aktionsdatei eine Aktion darstellt, wird es manchmal als *Aktionssymbol* bezeichnet. Wird durch die zugrundeliegende Aktion eine Anwendung gestartet, wird das Aktionsdateisymbol als *Anwendungssymbol* bezeichnet.

Durch Doppelklicken des Aktionssymbols wird die Aktion ausgeführt. Das Aktionssymbol kann auch als Übergabebereich definiert sein.

▼ Aktionsdateien (Aktionssymbol) erstellen

- ♦ Eine ausführbare Datei mit demselben Namen wie dem Aktionsnamen erstellen. Der Inhalt der Datei ist nicht wichtig.

Sieht die Aktionsdefinition beispielsweise folgendermaßen aus,

```
ACTION MyFavoriteApp
{
    EXEC_STRING      Mfa -file %Arg_1%
    DESCRIPTION      Runs MyFavoriteApp
    ICON             Mfapp
}
```

dann handelt es sich bei der Aktionsdatei um eine ausführbare Datei mit dem Namen `MyFavoriteApp`. Im Dateimanager und im Anwendungsmanager verwendet die Datei `MyFavoriteApp` das Symbolbild `Mfapp`. *größe . typ*. Durch Doppelklicken auf dem Symbol für `MyFavoriteApp` wird die Zeile `EXEC_STRING` der Aktion ausgeführt und der Inhalt des Felds `DESCRIPTION` (Führt `MyFavoriteApp` aus) wird als Kontexthilfe des Symbols angezeigt.

Aktionsbezeichnungen

Enthält die Aktionsdefinition ein Feld `LABEL`, wird die Aktionsdatei im Dateimanager und im Anwendungsmanager mit der in diesem Feld angegebenen Bezeichnung versehen, statt mit dem Dateinamen (*aktion_name*). Enthält die Aktionsdefinition beispielsweise

```
ACTION MyFavoriteApp
{
    LABEL           Favorite Application
    ...
}
```

dann erhält das Aktionssymbol die Bezeichnung "Favorite Application."

Das von der Aktion verwendete Symbolbild angeben

Im Feld `ICON` wird das für die Aktionssymbole der Aktion im Dateimanager und im Anwendungsmanager verwendete Symbol angegeben.

Wird kein Symbol angegeben, verwendet das System die standardmäßigen Aktionssymbolabbilddateien
`/usr/dt/appconfig/icons/sprache/Dtactn.*`.



Abbildung 10-2 Standardaktionssymbolbild

Das Standardaktionssymbol kann geändert werden, indem folgende Ressource gesetzt wird:

`*actionIcon: symbol_datei_name`

wobei für *symbol_datei_name* sowohl ein Dateiname ohne Pfad als auch ein absoluter Pfad verwendet werden kann.

Als Wert für das Feld `ICON` können zwei verschiedene Namen eingegeben werden:

- ein Dateiname ohne Pfad

Bei dem Dateinamen ohne Pfad handelt es sich um den Namen der Datei, die das Symbolbild enthält, ohne die Erweiterungen für Größe (`m` und `t`) und Abbildtyp (`bm` und `pm`). Beispielsweise lautet bei Dateien mit den Namen `GameIcon.m.pm` und `GameIcon.t.pm` der Dateinamen ohne Pfad `GameIcon`.

Verwendet der Benutzer den Dateinamen ohne Pfad, müssen die Symboldateien in ein Verzeichnis im Symbolsuchpfad gestellt werden:

- Persönliche Symbole: *Home-Verzeichnis*/`.dt/icons`
- Systemweite Symbole: `/etc/dt/appconfig/icons/sprache`
- ein absoluter Pfad zu der Symboldatei, einschließlich des vollständigen Dateinamens.

Der absolute Pfad sollte nur verwendet werden, wenn die Symboldatei sich nicht im Symbolsuchpfad befindet. Befindet sich die Symboldatei `GameIcon.m.pm` im Verzeichnis `/doc/projects`, das nicht im Symbolsuchpfad steht, wird als Wert für das Feld `ICON` `/doc/projects/GameIcon.m.pm` eingegeben.

Tabelle 10-1 enthält zu erstellende Symbolgrößen und die entsprechenden Dateinamen.

Tabelle 10-1 Symbolnamen und -größen für Aktionssymbole

Größe in Pixel	Bitmap-Name	Pixmap-Name
48 x 48	<i>name.l</i> .bm	<i>name.l</i> .pm
32 x 32	<i>name.m</i> .bm	<i>name.m</i> .pm
16 x 16	<i>name.t</i> .bm	<i>name.t</i> .pm

▼ Eine bereits vorhandene Aktionsdefinition ändern

Alle auf dem System zur Verfügung stehenden Aktionen, einschließlich der integrierten Aktionen, können geändert werden.

Hinweis – Bei der Änderung der Datenbank für integrierte Aktionen sollte vorsichtig vorgegangen werden. Die integrierten Aktionen sind genau auf die Desktop-Anwendungen abgestimmt.

1. Die Position der Definition der zu ändernden Aktion feststellen.

Die Standardpositionen für Aktionsdefinitionen sind:

- Integrierte Aktionen: */usr/dt/appconfig/types/sprache*
- Systemweite Aktionen: */etc/dt/appconfig/types/sprache*
- Persönliche Aktionen: *Home-Verzeichnis/.dt/types*

Für das System sind möglicherweise weitere zusätzliche Positionen definiert. Um eine Liste der Positionen, die das System für Aktionen verwendet, anzuzeigen, folgenden Befehl ausführen:

```
dtsearchpath -v
```

Das System verwendet die unter *DTDATABASESEARCHPATH* aufgelisteten Verzeichnisse.

2. Falls erforderlich, den Text der Aktionsdefinition in eine neue oder bereits vorhandene Datei in einem der folgenden Verzeichnisse kopieren:

- Systemweite Aktionen: */etc/dt/appconfig/types/sprache*
- Persönliche Aktionen: *Home-Verzeichnis/.dt/types*

Integrierte Aktionen müssen kopiert werden, da Dateien im Verzeichnis `/usr/dt/appconfig/types/sprache` nicht bearbeitet werden sollten.

3. Die Aktionsdefinition bearbeiten. Anschließend die Datei sichern.
4. 'Aktionen erneut laden' in der Gruppe von Anwendungen 'Desktop-Werkzeuge' doppelklicken.

Vorrang bei Aktionsdefinitionen

Wenn der Benutzer eine Aktion aufruft, durchsucht das System die Datenbank nach einem übereinstimmenden Aktionsnamen. Ist mehr als eine Aktion mit diesem Namen vorhanden, verwendet das System Vorrangregeln, um zu entscheiden, welche Aktion verwendet wird.

- Werden keine anderen Vorrangregeln angewendet, wird der Vorrang auf der Grundlage der Position der Definition zugeordnet. In der folgenden Liste sind die Aktionen entsprechend ihrem Rang von hoch nach niedrig geordnet:
 - Persönliche Aktionen (*Home-Verzeichnis*/`.dt/types`)
 - Systemweite Aktionen (`/etc/dt/appconfig/types/sprache`)
 - Systemweite Aktionen an einem anderen Rechner (*hostname*:`/etc/dt/appconfig/types/sprache`). Bei den fernen Rechnern, die durchsucht werden, handelt es sich um diejenigen, die im Anwendungssuchpfad aufgelistet sind.
 - Integrierte Aktionen (`/usr/dt/appconfig/types/sprache`)
- Innerhalb eines vorgegebenen Verzeichnisses werden die `*.dt`-Dateien in alphabetischer Reihenfolge gelesen.
- Aktionen, die durch die Felder `ARG_CLASS`, `ARG_TYPE`, `ARG_MODE` oder `ARG_COUNT` eingeschränkt sind, haben Vorrang vor nicht eingeschränkten Aktionen. (Der Standardwert für diese vier Felder ist `*`.)

Sind mehr als eine Einschränkung vorhanden, lautet die Rangfolge von hoch nach niedrig:

- `ARG_CLASS`
- `ARG_TYPE`
- `ARG_MODE`
- `ARG_COUNT`

Ist mehr als ein eingeschränktes Feld `ARG_COUNT` vorhanden, lautet die Rangfolge von hoch nach niedrig:

- Spezifischer ganzzahliger Wert n
- $<n$
- $>n$
- *

Als Beispiel sollen die folgenden Abschnitte aus Aktionsdefinitionen betrachtet werden.

```
ACTION EditGraphics
# EditGraphics-1
{
    ARG_TYPE      XWD
    ...
}
```

```
ACTION EditGraphics
# EditGraphics-2
{
    ARG_COUNT     0
    ...
}
```

```
ACTION EditGraphics
# EditGraphics-3
{
    ARG_TYPE     *
    ...
}
```

Durch Doppelklicken des Aktionssymbols 'EditGraphics' wird EditGraphics-2 gestartet, da kein Argument angegeben ist und ARG_COUNT 0 Vorrang hat. Wird ein Dateiargument vom Typ XWD angegeben, wird EditGraphics-1 verwendet, weil hier XWD ARG_TYPE angegeben ist. Für alle anderen Dateiargumente wird EditGraphics-3 verwendet.

Ausführungszeile für eine Befehlsaktion erstellen

Für eine Befehlsaktion (COMMAND) sind mindestens zwei Felder notwendig, ACTION und EXEC_STRING.

```
ACTION aktion_name
{
    EXEC_STRING ausführung_zeichenfolge
}
```

Bei der Ausführungszeile (`EXEC_STRING`) handelt es sich um den wichtigsten Teil einer Befehlsaktionsdefinition (`COMMAND`). Die verwendete Syntax gleicht der Befehlszeile, die in einem Terminal-Fenster ausgeführt wird, enthält aber zusätzliche Syntax für Datei- und Zeichenfolgenargumente.

Allgemeine Merkmale von Ausführungszeilen

Ausführungszeilen können enthalten:

- Dateiargumente und Parameter
- Shell-Syntax
- Absolute Pfade oder Namen ausführbarer Dateien

Aktionsargumente

Bei einem Argument handelt es sich um eine Information, die für die korrekte Ausführung eines Befehls oder einer Anwendung erforderlich ist. Als Beispiel soll die Befehlszeile betrachtet werden, über die eine Datei im Text Editor geöffnet werden könnte:

```
dtpad dateiname
```

In diesem Befehl ist *dateiname* ein Dateiargument des Befehls `dtpad`.

Aktionen können, wie Anwendungen und Befehle, über Argumente verfügen. Eine Befehlsaktion kann zwei verschiedene Arten von Daten verwenden:

- Dateien
- Zeichenfolgedaten

Shells in Ausführungszeilen verwenden

Die Ausführungszeile wird direkt und nicht über eine Shell ausgeführt. Aber der Benutzer kann auch explizit eine Shell in der Ausführungszeile aufrufen.

Beispiel:

```
EXEC_STRING      \  
                  /bin/sh -c \  
                  'tar -tvf %(File)Arg_1% 2>&1 | \${PAGER:-more};\  
                  echo "\n*** Zum Schließen im Fenstermenü 'Schließen'  
                  auswählen ***"'
```

Name oder absoluter Pfad der ausführbaren Datei

Befindet sich die Anwendung in einem Verzeichnis, das in der Variablen PATH aufgelistet ist, kann der einfache, ausführbare Name verwendet werden. Befindet sich die Anwendung in einem anderen Verzeichnis, muß der absolute Pfad zu der ausführbaren Datei verwendet werden.

Aktionen ohne Argumente erstellen

Dieselbe Syntax für EXEC_STRING verwenden, mit der die Anwendung auch über die Befehlszeile gestartet wird.

Beispiele

- Die folgende Ausführungszeile ist Teil einer Aktion, die den X-Client `xcutsel` startet.
`EXEC_STRING xcutsel`
- Die folgende Ausführungszeile startet den Client `xclock` mit einer Digitaluhr. Die Befehlszeile enthält eine Befehlszeilenoption, aber es werden keine Argumente benötigt.
`EXEC_STRING xclock -digital`

Aktionen erstellen, die übergebene Dateien akzeptieren

Folgende Syntax für das Dateiargument verwenden:

`%Arg_n%`

oder

`%(File)Arg_n%`

`(File)` kann, muß aber nicht angegeben werden, da von Argumenten für `Arg_n` (standardmäßig) angenommen wird, daß es sich um Dateien handelt. (Weitere Informationen zur Verwendung der Syntax `%(String)Arg_n%` können dem Abschnitt "Dateiargumente als Zeichenfolge interpretieren" auf Seite 189 entnommen werden.)

Diese Syntax ermöglicht dem Benutzer, ein Datendateiobjekt an das Aktionssymbol zu übergeben und die Aktion mit diesem Dateiarargument zu starten. Das nte Argument wird in die Befehlszeile übernommen. Bei der Datei kann es sich um eine lokale Datei oder um eine Datei in einem anderen System handeln.

Beispiele

- Die folgende Ausführungszeile führt `wc -w` aus und verwendet eine übergebene Datei als den Parameter `-load`.

```
EXEC_STRING wc -w %Arg_1%
```

- Das folgende Beispiel zeigt einen Teil einer Definition für eine Aktion, die nur für Verzeichnisargumente definiert ist. Wird ein Verzeichnis an das Aktionssymbol übergeben, zeigt die Aktion eine Liste aller Dateien in dem Verzeichnis an, für die Lese-/Schreibberechtigung vorliegt.

```
ACTION Lesbare_Dateien_auflisten
{
    ARG_TYPE      FOLDER
    EXEC_STRING /bin/sh -c 's -l %Arg_1% | grep rw-'
    ...
}
```

Aktionen erstellen, die Dateiarargumente anfordern

Folgende Syntax für das Dateiarargument verwenden:

```
%(File)"prompt"%
```

Mit dieser Syntax wird eine Aktion erstellt, die eine Aufforderung, einen Dateinamen einzugeben, anzeigt, wenn der Benutzer auf dem Aktionssymbol doppelklickt.

Die folgende Ausführungszeile zeigt beispielsweise ein Dialogfenster an, in dem der Benutzer aufgefordert wird, das Dateiarargument des Befehls `wc -w` einzugeben:

```
EXEC_STRING wc -w %(File)"Wörter in Datei zählen:"%
```

Aktionen erstellen, die übergebene Dateien akzeptieren oder anfordern

Folgende Syntax für das Dateiarargument verwenden:

```
%Arg_n"prompt"%
```

oder

```
%(File)Arg_n"prompt"%
```

Mit dieser Syntax wird eine Aktion erstellt, die

- eine übergebene Datei als Dateiarargument akzeptiert.
- ein Dialogfenster anzeigt, in dem der Benutzer aufgefordert wird, einen Dateinamen einzugeben, wenn der Benutzer auf dem Aktionssymbol doppelklickt.

Beispielsweise führt die folgende Ausführungszeile den Befehl `lp -oraw` an einer übergebenen Datei aus. Wird die Aktion durch Doppelklicken auf dem Symbol gestartet, erscheint ein Dialogfenster mit der Aufforderung den Dateinamen einzugeben.

```
EXEC_STRING lp -oraw %Arg_1"Datei drucken:"%
```

Aktion erstellen, die Parameter anfordern

Folgende Syntax für den Parameter verwenden:

```
%"prompt"%
```

oder

```
%(String)"prompt"%
```

Die Angabe `(String)` ist optional, da Text in Anführungszeichen standardmäßig als Zeichenfolge interpretiert wird. Mit dieser Syntax wird ein Dialogfenster angezeigt, das eine Aufforderung enthält, einen Parameter einzugeben. Diese Syntax nicht verwenden, wenn ein Dateiname eingegeben werden muß.

Beispielsweise führt die folgende Ausführungszeile den Befehl `xwd` aus und fordert zur Eingabe eines Werts auf, der jedem Pixel hinzugefügt werden soll:

```
EXEC_STRING xwd -add %"Wert hinzufügen:"% -out %Arg_1"Dateiname:"%
```

Dateiargumente als Zeichenfolge interpretieren

Folgende Syntax für das Dateiargument verwenden:

```
%(String)Arg_n%
```

Beispielsweise druckt folgende Ausführungszeile mit dem Befehl `lp -tbanner dateiname` eine Datei mit einem Deckblatt, das den Dateinamen enthält.

```
EXEC_STRING lp -t%(String)Arg_1% %(File)Arg_1"Datei drucken:"%
```

Shell-Funktionen in einer Aktion zur Verfügung stellen

Die Shell in der Ausführungszeile angeben:

```
/bin/sh -c 'Befehl'  
/bin/ksh -c 'Befehl'  
/bin/csh -c 'Befehl'
```

Beispiele

- Die folgende Ausführungszeile zeigt eine Aktion, die mit Shell-Piping arbeitet.

```
EXEC_STRING /bin/sh -c 'ps | lp'
```

- Die folgende Ausführungszeile ist komplexer. Ein Shell-Prozeß ist erforderlich und ein Dateiarargument wird akzeptiert:

```
EXEC_STRING /bin/sh -c 'tbl %Arg_1"Hilfeseite:"% | troff -man'
```

- Für die folgende Ausführungszeile ist es erforderlich, daß es sich bei dem Argument um eine komprimierte Datei handelt. Die Aktion dekomprimiert die Datei und druckt sie mit dem Befehl `lp -oraw`.

```
EXEC_STRING /bin/sh -c 'cat %Arg_1 "Datei drucken:"% | \  
uncompress | lp -oraw'
```

- Die folgende Ausführungszeile startet eine Shell-Prozedur.

```
EXEC_STRING /usr/local/bin/StartGnuClient
```

Befehlsaktionen mit mehreren Dateiarargumenten erstellen

Aktionen verarbeiten mehrere Dateiarargumente auf drei verschiedene Weisen:

- Die Aktion kann wiederholt ausgeführt werden, jeweils einmal für jedes Argument. Enthält eine Zeile `EXEC_STRING` ein einziges Dateiarargument und es werden mehrere Dateiarargumente zur Verfügung gestellt, indem mehrere Dateien an das Aktionsymbol übergeben werden, wird die Aktion für jedes einzelne Dateiarargument ausgeführt.

Werden beispielsweise mehrere Dateiarargumente an folgende Aktionsdefinition übergeben:

```

ACTION DisplayScreenImage
EXEC_STRING      xwud -in %Arg_1%
...
}

```

wird die Aktion DisplayScreenImage wiederholt ausgeführt.

- Die Aktion kann zwei oder mehrere nicht austauschbare Dateiargumente verwenden. Beispielsweise erfordert

```
xsetroot -cursor zeigerdatei maskendatei
```

zwei eindeutige Dateien in einer bestimmten Reihenfolge.

- Die Aktion kann denselben Befehl sequentiell an jedem Dateiargument ausführen. Beispielsweise druckt

```
pr datei [datei ...]
```

eine oder mehrere Dateien in einem Druckjob.

Eine Aktion mit nicht austauschbaren Dateiargumenten erstellen

Eine der folgenden Syntaxkonventionen verwenden:

- Soll die Aktion zur Eingabe der Dateinamen auffordern, folgende Syntax für jedes Dateiargument verwenden:

```
%(File) "prompt"%
```

Für jedes Argument eine andere Zeichenfolge *prompt* verwenden.

Beispielsweise fordert die folgende Ausführungszeile zur Eingabe von zwei Dateinamen auf.

```
EXEC_STRING xsetroot -cursor %(File)"Zeiger Bitmap:"% \
            %(File)"Maske Bitmap:"%
```

- Damit übergebene Dateien akzeptiert werden, folgende Syntax für jedes Dateiargument verwenden:

```
%Arg_n%
```

wobei für jedes Argument verschiedene Werte *n* verwendet werden. Zum Beispiel:

```
EXEC_STRING diff %Arg_1% %Arg_2%
```

Aktionen mit austauschbaren Dateiargumenten erstellen

Eine der folgenden Syntaxkonventionen verwenden:

- Um eine Aktion zu erstellen, die übergebene Dateien akzeptiert und einen Befehl in der Form *Befehl datei₁ datei₂ ...* absetzt, folgende Syntax für die Dateiarargumente verwenden:

```
%Args%
```

- Um eine Aktion zu erstellen, die mehrere übergebene Dateien akzeptiert oder bei Doppelklicken eine Eingabeaufforderung anzeigt, in der zur Eingabe eines einzelnen Dateinamens aufgefordert wird, folgende Syntax für die Dateiarargumente verwenden:

```
%Arg_1 "prompt"% %Args%
```

Die Aktion setzt den Befehl in der Form *Befehl datei₁ datei₂ ...* ab.

Beispiele

- Die folgende Ausführungszeile erstellt eine Aktion, die folgenden Befehl

```
pr datei1 datei2
```

mit mehreren Dateiarargumenten ausführt.

```
EXEC_STRING pr %Args%
```

- Die folgende Ausführungszeile erstellt eine Aktion, ähnlich dem vorhergehenden Beispiel, bei der jedoch eine Eingabeaufforderung angezeigt wird, wenn das Aktionssymbol angeklickt wird. (keine Dateiarargumente).

```
EXEC_STRING pr %Arg_1 "Datei(en) drucken:"% %Args%
```

Aktionen für mehrere übergebene Dateien erstellen

Um mehrere übergebene Dateiarargumente zu akzeptieren und eine Befehlszeile in der Form

Befehl datei₁ datei₂ ...

auszuführen, folgende Syntax verwenden:

```
%Args%
```

Beispiele

- Die folgende Ausführungszeile führt eine Prozedur namens Checkout für mehrere Dateien aus:

```
EXEC_STRING /usr/local/bin/Checkout \  
            %Arg_1"Welche Datei entnehmen?"% %Args%
```

- Die folgende Ausführungszeile führt lp -oraw mit mehreren Dateien aus:

```
EXEC_STRING lp -oraw %Arg_1"Datei drucken:"% %Args%
```

Fensterunterstützung und Terminal-Emulatoren für Befehlsaktionen

Befehlsaktionen unterstützen Fenster auf dem Desktop auf verschiedene Weise:

- Verfügt die Anwendung über ein eigenes Fenster, muß beim Schreiben der Aktion keine zusätzliche Fensterunterstützung zur Verfügung gestellt werden. Diese Option wird ebenfalls verwendet, wenn eine Aktion einen Befehl ausführt, für den keine direkte Eingabe durch den Benutzer erforderlich ist und der keine Ausgabe hat.
- Muß die Anwendung in einem emulierten Terminalfenster ausgeführt werden, muß die Aktion so geschrieben werden, daß zunächst ein Fenster geöffnet wird und dann die Anwendung ausgeführt wird. Verschiedene Terminal-Optionen stehen zur Verfügung.

Fensterunterstützung für die Aktion angeben

Das Feld `WINDOW_TYPE` verwenden, um die Art von Fensterunterstützung anzugeben, die für die Aktion erforderlich ist.

WINDOW_TYPE	Welche Fensterunterstützung steht zur Verfügung?
<code>NO_STDIO</code>	Keine. <code>NO_STDIO</code> wird verwendet, wenn die Anwendung über ein eigenes Fenster verfügt, oder wenn der Befehl keine sichtbare Ausgabe produziert.
<code>PERM_TERMINAL</code>	Permanentes emuliertes Terminalfenster. Die Aktion öffnet ein Terminal-Fenster, das offen bleibt, bis es vom Benutzer explizit geschlossen wird. Der Benutzer kann in diesem Fenster Daten eingeben. Diesen Fenstertyp für Befehle verwenden, für die eine Eingabe erforderlich ist und die eine Ausgabe produzieren, anschließend das Fenster schließen (zum Beispiel für <code>ls verzeichnis</code>).

TERMINAL	Temporäres emuliertes Terminalfenster. Die Aktion öffnet ein Terminal-Fenster, das geschlossen wird, sobald der Befehl beendet ist. Diesen Fenstertyp für Befehle mit Gesamtanzeige verwenden (zum Beispiel für <code>vi</code>).
----------	--

Befehlszeilenoptionen für den Terminal-Emulator angeben

Das Feld `TERM_OPTS` in der Aktionsdefinition verwenden, um Befehlszeilenoptionen für den Terminal-Emulator anzugeben.

Beispielsweise fordert die folgende Aktion zur Angabe des ausführenden Rechners auf:

```
ACTION OpenTermOnSystemUserChooses
{
    WINDOW_TYPE      PERM_TERMINAL
    EXEC_HOST         %(String)"Fernes Terminal an:"%
    TERM_OPTS         -title %(String)"Fenster titel:"%
    EXEC_STRING       $SHELL
}
```

Andere Standard-Terminal-Emulatoren angeben

Der Standard-Terminal-Emulator für Aktionen ist `dtterm`. Der Benutzer kann auch einen anderen Terminal-Emulator wählen. Der Standard-Terminal-Emulator wird verwendet, wenn für die Aktion kein anderer Terminal-Emulator explizit angegeben ist.

Der Terminal-Emulator für Aktionen muß über folgende Befehlszeilenoptionen verfügen:

- `-title fenster_titel`
- `-e Befehl`

Der Standard-Terminal-Emulator für Aktionen wird durch zwei Ressourcen bestimmt:

- Die Ressource `localTerminal` gibt den Terminal-Emulator an, der von lokalen Anwendungen verwendet wird.

```
*localTerminal:    terminal
```

Beispiel:

```
*localTerminal:xterm
```

- Die Ressource `remoteTerminal` gibt den Terminal-Emulator an, der von Anwendungen auf fernen Systemen verwendet wird.

```
*remoteTerminal:  host: terminal [ , host: terminal...]
```

Beispiel:

```
*remoteTerminal: sysibm1:/usr/bin/xterm,syshp2:/usr/bin/yterm
```

Aktionen auf bestimmte Argumente einschränken

Durch Einschränkung einer Aktion auf einen bestimmten Argumenttyp wird eine Feinanpassung einer Aktion erzielt. Beispielsweise ist es sinnvoll, eine Aktion, die eine Anzeigefunktion für PostScript-Dateien aufruft, auf PostScript-Dateiargumente einzuschränken. Ist eine solche Einschränkung gesetzt, zeigt die Aktion eine Fehlermeldung an, wenn eine Datei angegeben wird, die nicht im PostScript-Format ist.

Aktionen können auf der Grundlage folgender Merkmale eingeschränkt werden:

- dem Datentyp des Dateiarguments.
- der Anzahl der Dateiargumente. Zum Beispiel, keine Argumente im Gegensatz zu einem oder mehreren Argumenten. Dadurch wird unterschiedliches Verhalten bei Übergabe und Doppelklicken bei einem Aktionssymbol bewirkt.
- dem Schreib-/Lesemodus des Arguments.

Aktionen auf einen angegebenen Datentyp einschränken

Das Feld `ARG_TYPE` wird verwendet, um die Datentypen anzugeben, für die die Aktion gültig ist. Den Datenattributnamen verwenden.

Es können mehrere Datentypen angegeben werden. Die Einträge mit Kommas voneinander trennen.

In der folgenden Aktionsdefinition wird angenommen, ein Datentyp `Gif` wurde erstellt.

```
ACTION Open_Gif
{
    TYPE           COMMAND
    LABEL          "Gif anzeigen"
    WINDOW_TYPE    NO_STDIO
}
```

```

ARG_TYPE      Gif
ICON          xgif
DESCRIPTION   Zeigt Gif-Dateien an
EXEC_STRING   xgif
}

```

Aktionen auf der Grundlage der Argumentanzahl einschränken

Das Feld `ARG_COUNT` wird verwendet, um die Anzahl der Argumente anzugeben, die die Aktion akzeptieren kann. Gültige Werte sind:

- * (Standard) Eine beliebige Anzahl von Argumenten. Andere Werte haben Vorrang vor *.
- n* Jede positive ganze Zahl, einschließlich 0.
- >*n* Mehr als *n* Argumente.
- <*n* Weniger als *n* Argumente.

`ARG_COUNT` wird unter anderem verwendet, um unterschiedliches Aktionssymbolverhalten zu bewirken, je nachdem ob der Benutzer auf dem Symbol doppelt klickt oder eine Datei übergibt. Weitere Informationen hierzu enthält der nächste Abschnitt "Unterschiedliches Verhalten bei Doppelklicken und Übergabe zur Verfügung stellen".

▼ Unterschiedliches Verhalten bei Doppelklicken und Übergabe zur Verfügung stellen

Die folgende Prozedur wird verwendet, um eine Aktion zu erstellen, die eine übergebene Datei akzeptiert, aber die nicht zur Angabe einer Datei auffordert, wenn auf dem Aktionssymbol doppeltgeklickt wird.

1. Eine Aktionsdefinition für die Funktionalität für Doppelklicken erstellen.

Im Feld `ARG_COUNT` 0 Argumente angeben. Die Syntax für `EXEC_STRING` verwenden, die kein übergebenes Argument akzeptiert.

2. Eine zweite Aktionsdefinition für die Funktionalität für Übergabe erstellen.

Im Feld `ARG_COUNT` >0 Argumente angeben. Eine Syntax für `EXEC_STRING` verwenden, die ein übergebenes Argument akzeptiert.

Angenommen, die folgenden zwei Befehlszeilen können verwendet werden, um einen Editor mit dem Namen `vedit` zu starten:

- Um den Editor ohne Dateiarargument zu starten:

```
vedit
```

- Um den Editor mit einem Dateiarargument, das als schreibgeschütztes Dokument geöffnet wird, zu starten:

```
vedit -R dateiname
```

Die folgenden zwei Aktionen stellen die Funktionalität für Übergabe und Doppelklicken für eine Aktion mit dem Namen `Vedit` zur Verfügung. Die erste Aktion hat Vorrang, wenn die Datenbank nach einer Entsprechung durchsucht wird, da `ARG_COUNT 0` genauer spezifiziert ist als das implizierte `ARG_COUNT *` der Definition der Funktionalität für Übergabe.

```
# Funktionalität für Doppelklicken
```

```
ACTION Vedit
```

```
{
```

```
    TYPE          COMMAND
    ARG_COUNT      0
    WINDOW_TYPE    PERM_TERMINAL
    EXEC_STRING    vedit
```

```
}
```

```
# Funktionalität für Übergabe
```

```
ACTION Vedit
```

```
{
```

```
    TYPE          COMMAND
    WINDOW_TYPE    PERM_TERMINAL
    EXEC_STRING    vedit -R %Arg_1%
```

```
}
```

Aktionen auf der Grundlage des Argumentmodus einschränken

Das Feld `ARG_MODE` wird verwendet, um den Schreib-/Lesemodus des Arguments anzugeben. Gültige Werte sind:

- * (Standard) Jeder Modus
- !w keine Schreibberechtigung
- w Schreibberechtigung

Aktionen erstellen, die Anwendungen auf fernen Systemen ausführen

Wenn von Aktionen und Ausführung auf fernen Systemen die Rede ist, werden zwei Begriffe häufig verwendet: :

Datenbank-Rechner	Das System, das die Aktionsdefinition enthält
Ausführender Rechner	Das System, auf dem die ausführbare Datei ausgeführt wird

Meistens befinden sich Aktionen und deren Anwendungen auf demselben System. Da der ausführende Rechner für eine Aktion standardmäßig der Datenbank-Rechner ist, ist eine spezielle Syntax nicht erforderlich.

Handelt es sich jedoch bei ausführendem Rechner und Datenbank-Rechner um verschiedene Rechner, muß in der Aktionsdefinition angegeben werden, wo die Ausführungszeile ausgeführt werden soll.

Die Fähigkeit, die Position von Aktionen und Anwendungen in anderen Systemen festzustellen, ist Bestandteil der Client/Server-Architektur des Desktops. Weiterführende Informationen zu Anwendungen im Netzwerk können dem Abschnitt "Anwendungs-Services verwalten" auf Seite 111 entnommen werden.

Aktionen erstellen, die Anwendungen an einem anderen Rechner ausführen

Das Feld `EXEC_HOST` in der Aktionsdefinition wird verwendet, um die Position der Anwendung anzugeben.

Gültige Werte für `EXEC_HOST` sind:

<code>%DatabaseHost%</code>	Der Rechner, auf dem die Aktion definiert ist.
<code>%LocalHost%</code>	Der Rechner, auf dem die Aktion aufgerufen wird (der <i>Session-Server</i>).
<code>%DisplayHost%</code>	Der Rechner, auf dem der X-Server ausgeführt wird (nicht zulässig für Xstations).
<code>%SessionHost%</code>	Der Rechner, auf dem der steuernde Anmeldemanager ausgeführt wird.

<i>hostname</i>	Der benannte Rechner. Dieser Wert wird in Umgebungen verwendet, in denen die Aktion immer auf einem bestimmten Rechner aufgerufen werden soll.
<code>% "prompt" %</code>	Fordert den Benutzer bei jedem Aufrufen des Befehls zur Eingabe des Rechnernamens auf.

Der Standardwert ist `%DatabaseHost%`, `%LocalHost%`. Daher versucht die Aktion, wenn das Feld `EXEC_HOST` übergangen wird, zunächst, den Befehl auf dem Rechner auszuführen, der die Aktionsdefinition enthält. Schlägt dies fehl, versucht die Aktion den Befehl auf dem Session-Server auszuführen.

Beispiele

- Im folgenden Beispiel wird als Rechner `ddsyd` angegeben:

```
EXEC_HOST ddsyd
```

- In diesem Beispiel wird der Benutzer aufgefordert, einen Rechnernamen einzugeben:

```
EXEC_HOST %"Host enthält Anwendung:"%
```

- In diesem Beispiel wird angegeben, daß die Aktion zunächst versucht, die Anwendung auf dem Rechner, der die Aktionsdefinition enthält, auszuführen. Schlägt dies fehl, wird die Aktion versuchen, die Anwendung auf dem Rechner `ddsyd` auszuführen.

```
EXEC_HOST %DatabaseHost%, ddsyd
```

Variablen in Definitionen für Aktionen und Datentypen verwenden

In Definitionsdateien für Aktionen und Datentypen können Zeichenfolgevariablen und Umgebungsvariablen verwendet werden.

Zeichenfolgevariablen in einer Aktion verwenden

Eine Zeichenfolgevariablendefinition ist wirksam von der Position der Definition bis zum Ende der Datei. Es gibt keine globalen Zeichenfolgevariablen für die Datenbank.

Existieren sowohl eine Zeichenfolgevariable als auch eine Umgebungsvariable mit demselben Namen, hat die Zeichenfolgevariable Vorrang.

▼ **Eine Zeichenfolgevariable definieren**

◆ Folgende Syntax verwenden:

```
set variable_name=wert
```

Variablenamen können jedes alphanumerische Zeichen sowie Unterstriche (`_`) enthalten. Jede Variablendefinition muß sich auf einer eigenen Zeile befinden.

Beispiel:

```
set Remote_Application_Server=sysapp
set Remote_File_Server=sysdata
```

▼ **Auf eine Zeichenfolgevariable verweisen**

◆ Folgende Syntax verwenden:

```
${[ ]variable[ ]}
```

Beispiel:

```
EXEC-HOST $Remote_Application_Server
CWD /net/${Remote_File_Server}/doc/project
```

Umgebungsvariablen in Aktionen und Datentypen verwenden

◆ Auf eine Umgebungsvariable mit folgender Syntax verweisen:

```
${[ ]variable[ ]}.
```

Die Variable wird erweitert (durch ihren Wert ersetzt), wenn die Datenbank geladen wird. Existieren sowohl eine Zeichenfolgevariable als auch eine Umgebungsvariable mit demselben Namen, hat die Zeichenfolgevariable Vorrang.

Beispielsweise druckt folgende Ausführungszeile eine Datei mit einem Deckblatt, das den Namen der Datei enthält.

```
EXEC-STRING lp -t$LOGNAME %(File)Arg_1%
```

Aktionen über eine Befehlszeile aufrufen

Das Desktop stellt den Befehl `dtaction` zur Verfügung, mit dem Aktionen über eine Befehlszeile ausgeführt werden können. Mit `dtaction` können Aktionen von folgenden Positionen aus ausgeführt werden:

- Prozeduren
- anderen Aktionen
- der Befehlszeile eines Terminal-Emulators

Syntax von dtaction

```
dtaction [-user benutzer_name] [-execHost hostname] aktion_name [argument [argument]...]
```

<code>-user <i>benutzer_name</i></code>	Stellt die Fähigkeit zur Verfügung, die Aktion als ein anderer Benutzer auszuführen. Wird dtaction von einem anderen als dem durch <i>benutzer_name</i> angegebenen Benutzer aufgerufen, wird dieser zur Eingabe des Kennworts aufgefordert.
<code>-execHost <i>hostname</i></code>	Nur für Befehlsaktionen. Hier wird der Rechner angegeben, auf dem der Befehl ausgeführt wird.
<i>argument</i>	Argumente für die Aktion, normalerweise Dateiargumente.

Der Client dtaction verfügt über weitere Befehlszeilenoptionen. Weitere Informationen hierzu können der Hilfeseite dtaction(1) entnommen werden.

Aktionen erstellen, die andere Aktionen ausführen

In der Zeile EXEC_STRING die Aktion dtaction verwenden.

Beispielsweise verwendet die folgende Aktion eine integrierte Aktion mit dem Namen Spell (die Aktion trägt im Anwendungsmanager die Bezeichnung "Check Spelling"). Die neue Aktion führt den Text Editor und die Aktion Spell aus und zeigt Rechtschreibfehler in einem separaten Terminal-Emulator-Fenster an.

```
ACTION EditAndSpell
{
    WINDOW_TYPE      NO_STDIO
    EXEC_STRING      /bin/sh -c 'dtaction Spell \
                    %Arg_1"Datei:"%; dtpad %Arg_1%'
}
```

Aktionen erstellen, die als andere Benutzer ausgeführt werden

In EXEC_STRING folgende Syntax verwenden:

```
EXEC_STRING dtaction -user benutzer_name aktion_name [datei_argument]
```

Der neue Benutzer (*benutzer_name*) muß durch einen der folgenden Mechanismen Zugriff auf die Anzeige des Systems erhalten:

- Leseberechtigung für die Datei `.Xauthority` des angemeldeten Benutzers
- *Oder*, `xhost`-Berechtigung

Die beiden folgenden Aktionen ermöglichen dem Benutzer, sich als Root anzumelden und eine Datei `app-defaults` zu bearbeiten.

```
ACTION AppDefaults
{
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING    /usr/dt/bin/dtaction -user root \
                  EditAppDefaults %Arg_1"Datei:"%
}
ACTION EditAppDefaults
{
    WINDOW_TYPE    TERMINAL
    EXEC_STRING    /bin/sh -c 'chmod +w %Arg_1%; \
                  vi %Arg_1%; chmod -w %Arg_1%'
}
```

Sprachenabhängige Aktionen erstellen

Der Suchpfad für Datentypen enthält sprachenabhängige Positionen. Das Desktop verwendet den Wert `LANG`, um festzulegen, welche Positionen nach Datentypdefinitionen durchsucht werden sollen.

Positionen für sprachenabhängige Aktionen

Sprachenabhängige Aktionsdefinitionen müssen in die korrekten sprachenabhängigen Verzeichnisse im Aktionssuchpfad gestellt werden.

Der Standardsuchpfad ist:

- Persönliche Aktionen: *Home-Verzeichnis*/`.dt/types`
- Systemweite Aktionen: `/etc/dt/appconfig/types/sprache`
- Integrierte Aktionen: `/usr/dt/appconfig/types/sprache`

▼ Eine bereits vorhandene Aktion sprachenabhängig einrichten

1. Eine Datei in dem entsprechenden sprachenabhängigen Verzeichnis erstellen (zum Beispiel in `/etc/dt/appconfig/types/japanese`).
2. Die Aktionsdefinition in die sprachenabhängige Konfigurationsdatei kopieren.

Beispielsweise kann der Benutzer eine Aktionsdefinition aus

`app_root/dt/appconfig/types/C/datei.dt`

in folgende Datei kopieren:

`app_root/dt/appconfig/types/japanese/neuedatei.dt`

3. Ein Feld LABEL hinzufügen oder ein bereits vorhandenes Feld LABEL ändern.

LABEL *zeichenfolge*

Anwendungsmanager und Dateimanager verwenden die für LABEL gesetzte Zeichenfolge, um das Symbol der Aktion zu identifizieren.

4. Die folgenden Felder in der Aktionsdefinition können sprachenabhängig eingerichtet werden.
 - Symbole sprachenabhängig einrichten: ICON
 - Kontexthilfe sprachenabhängig einrichten: DESCRIPTION
 - Eingabeaufforderungen sprachenabhängig einrichten: jeder Text in Anführungszeichen in der Zeile EXEC_STRING

Aktionen für ToolTalk-Anwendungen erstellen

Hinweis – Folgende Informationen gelten nur für Anwendungen, die das Senden von Nachrichten mit ToolTalk unterstützen.

Mit dem Aktionstyp `TT_MSG` eine Aktion erstellen, die eine ToolTalk-Nachricht sendet.

```
ACTION aktion_name
{
    TYPE     TT_MSG
    ...
}
```

Felder 'addressing' und 'disposition'

- Das ToolTalk-Feld `addressing` ist immer auf `TT_PROCEDURE` gesetzt
- Das ToolTalk-Feld `disposition` wird standardmäßig auf die Spezifikation im statischen Nachrichtenmuster gesetzt.

Nicht unterstützte Nachrichten

Die folgenden Nachrichten werden nicht von Aktionen vom Typ `TT_MSG` unterstützt:

- Objektbezogene Nachrichten in ToolTalk
- kontextargumente in Nachrichten

Schlüsselwörter für ToolTalk-Nachrichtenaktionen

Schlüsselwort	Verwendung
<code>TT_CLASS</code>	Definiert den Wert des ToolTalk-Meldungsfelds <code>class</code>
<code>TT_SCOPE</code>	Definiert den Wert des ToolTalk-Meldungsfelds <code>scope</code>
<code>TT_OPERATION</code>	Definiert den Wert des ToolTalk-Meldungsfelds <code>operation</code> .
<code>TT_FILE</code>	Definiert den Wert des ToolTalk-Meldungsfelds <code>file</code>
<code>TT_ARGn_MODE</code>	Definiert den Wert des ToolTalk-Attributs <code>mode</code> für das <i>n</i> te Meldungsfeld
<code>TT_ARGn_VTYPE</code>	Definiert den Wert des ToolTalk-Attributs <code>vtype</code> des <i>n</i> ten Meldungsfelds
<code>TT_ARGn_VALUE</code>	Definiert den Wert des <i>n</i> ten Meldungsfelds

Es gibt zwei Möglichkeiten, eine Datentypdefinition zu erstellen:

- in dem Werkzeug 'Aktion erstellen'. Die Verwendung von 'Aktion erstellen' wird in Kapitel 9, "Aktionen und Datentypen mit 'Aktion erstellen' erstellen." ausführlich behandelt.
- die Datentypdefinition manuell erstellen.

Um einen Datentyp manuell zu erstellen, ist es erforderlich, eine Datenbankdatei zu bearbeiten.

Im folgenden Kapitel wird beschrieben, wie Datentypdefinitionen manuell erstellt werden.

<i>Gründe, einen Datentyp manuell zu erstellen</i>	<i>206</i>
<i>Komponenten einer Datentypdefinition: Kriterien und Attribute</i>	<i>206</i>
<i>Datentyp manuell erstellen: Allgemeine Schritte</i>	<i>207</i>
<i>Beispiel für die Erstellung einer persönlichen Aktion und eines Datentyps</i>	<i>209</i>
<i>Datenkriterien für einen Datentyp definieren</i>	<i>213</i>

Weiterführende Informationen

- Eine Einführung in Datentypen enthält Kapitel 8, "Einführung in Aktionen und Datentypen."

- Referenzinformationen zu Datentypdefinitionen können der Hilfeseite `dtddsfile(4)` entnommen werden.

Gründe, einen Datentyp manuell zu erstellen

Wird ein Datentyp manuell erstellt, können alle Möglichkeiten, die die Syntax von Datentypdefinitionen bietet, ausgeschöpft werden.

Sollen die folgenden Merkmale von Datentypen verwendet werden, muß der Datentyp manuell erstellt werden:

- Datentypzuordnung auf der Grundlage der Position (des Pfads).
- Die Möglichkeit, dem Datentyp neben 'Öffnen' und 'Drucken' weitere Aktionen zuzuordnen.
- Mehrere Namens-, Muster- oder Inhaltskriterien werden für denselben Datentyp, beispielsweise ein Datentyp, der auf Dateien mit der Erweiterung `*.abc` oder `*.def` endet, verwendet.
- Datentypzuordnung auf der Grundlage von Verweisen.

Komponenten einer Datentypdefinition: Kriterien und Attribute

Eine Datentypdefinition besteht aus zwei separaten Datenbankdefinitionen:

- Die Definition `DATA_ATTRIBUTES`.

Die Definition `DATA_ATTRIBUTES` beschreibt den Namen des Datentyps und das Erscheinungsbild und Verhalten von Dateien dieses Typs.

- Die Definition `DATA_CRITERIA`.

Die Definition `DATA_CRITERIA` beschreibt, nach welchen Kriterien der Datentyp zugeordnet wird. Jede Kriteriendefinition gibt an, auf welche Definition `DATA_ATTRIBUTES` sie angewendet wird.

Für jede Definition `DATA_ATTRIBUTES` muß mindestens eine Definition `DATA_CRITERIA` vorliegen. Es können ihr aber auch mehre Definitionen `DATA_CRITERIA` zugeordnet sein.

Beispielsweise könnte ein Benutzer eine Attributdefinition für PostScript-Dateien erstellen, in der beschrieben wird, wie PostScript-Dateien im Dateimanager angezeigt werden und wie sie sich verhalten. Anschließend

könnte der Benutzer zwei verschiedene Kriterien für den Datentyp PostScript erstellen, einen auf der Grundlage des Dateinamens, und einen auf der Grundlage des Dateinhalts.

Weitere Informationen können dem Abschnitt “Datenkriterien für einen Datentyp definieren” auf Seite 213 entnommen werden.

Datentyp manuell erstellen: Allgemeine Schritte

In diesem Abschnitt wird beschrieben, wie eine Datentypkonfigurationsdatei erstellt wird.

Konfigurationsdateien für Datentypen

Konfigurationsdateien, die Datentypdefinitionen enthalten, müssen folgende Voraussetzungen erfüllen:

- Die Dateien müssen der Namenskonvention *name.dt* entsprechen.
- Die Dateien müssen sich im Datenbanksuchpfad befinden. Der Standardsuchpfad lautet:

Persönliche Datentypen *Home-Verzeichnis/.dt/types*

Systemweite Datentypen */etc/dt/appconfig/types/sprache*

Integrierte Datentypen */usr/dt/appconfig/types/sprache*. Dieses Verzeichnis sollte nicht verwendet werden.

Weitere Informationen zum Ändern des Datenbanksuchpfads können dem Abschnitt “Wert eines Suchpfads setzen” auf Seite 129 entnommen werden.

▼ Datentypdefinition erstellen

1. Eine bereits vorhandene Datenbankdatei öffnen oder eine neue erstellen.

Weitere Informationen können dem vorherigen Abschnitt “Konfigurationsdateien für Datentypen” entnommen werden.

2. Die Datenattribute für den Datentyp unter Verwendung folgender Syntax definieren:

```
DATA_ATTRIBUTES daten_typ_name
{
    ICON           bild_name
```

```

DESCRIPTION    zeichenfolge
attribute_feld
attribute_feld
...
}

```

wobei die Parameter folgende Bedeutung haben:

daten_typ_name Ein eindeutiger Name, der diesem Datentyp zugeordnet wird.

bild_name Dateiname oder Pfad einer Symboldatei. Den Dateinamen ohne Pfad für die Datei verwenden. Beispielsweise wird für die Symboldateien `yimage.m.pm` und `myimage.t.pm` der Name `myimage` verwendet.

attribut_feld Feld, das das Erscheinungsbild oder Verhalten des Datentyps definiert.

zeichenfolge Zeichenfolge. Der Inhalt dieser Zeichenfolge wird für diesen Datentyp als Kontexthilfe angezeigt.

Weitere Informationen hierzu können dem Abschnitt “Beispiel für die Erstellung einer persönlichen Aktion und eines Datentyps” auf Seite 209 entnommen werden.

3. Die Datenkriterien unter Verwendung folgender Syntax definieren:

```

DATA_CRITERIA kriterien_name
{
  DATA_ATTRIBUTES_NAME daten_typ_name
  kriterien_feld
  kriterien_feld
  ...
}

```

wobei die Parameter folgende Bedeutung haben:

kriterien_name Eindeutiger Name für diese Kriteriendefinition.

daten_typ_name Name, der in der Definition `DATA_ATTRIBUTES` verwendet wird.

kriterien_feld Feld, in dem das Kriterium für die Zuordnung einer Datei zu diesem Datentyp definiert wird.

Weitere Informationen hierzu können dem Abschnitt “Datenattribute eines Datentyps definieren” auf Seite 210 entnommen werden.

4. Die Datenbankdatei sichern.
5. Die Symbole für den Datentyp erstellen.
Weitere Informationen hierzu können dem Abschnitt “Symbolbild angeben, das für einen Datentyp verwendet wird” auf Seite 211 entnommen werden.
6. Falls erforderlich, die Aktionen, die im Feld `ACTIONS` der Attributdefinition aufgelistet sind, erstellen.
7. ‘Aktionen erneut laden’ in der Gruppe von Anwendungen ‘Desktop-Werkzeuge’ doppelklicken, um die Datenbank erneut zu laden.

Beispiel für die Erstellung einer persönlichen Aktion und eines Datentyps

Angenommen, das System enthält eine Anwendung mit dem Namen `xgif`, die GIF-Bilder anzeigt. Normalerweise wird das Programm ausgeführt, indem folgendes eingegeben wird:

```
xgif dateiname
```

Nun sollen GIF-Bilder auf verschiedene Weise angezeigt werden:

- durch Doppelklicken einer GIF-Datendatei
- durch Auswählen der Datendatei und Auswählen der Anwendung im Menü ‘Ausgewählt’

1. Eine neue Datei *Home-Verzeichnis*/`.dt/types/GifViewer.dt` zum Bearbeiten öffnen.

2. Die Datentypdefinitionen eingeben:

```
DATA_ATTRIBUTES Gif
{
  DESCRIPTION          Gif image file.
  ICON                 GifIcon
  ACTIONS              View
}
DATA_CRITERIA Gif_Criteria
{
  DATA_ATTRIBUTES_NAME Gif
  NAME_PATTERN          *.gif
}
```

3. Die Aktionsdefinition für die Aktion ‘GifViewer’ eingeben:

```

ACTION GifViewer
{
    EXEC_STRING      xgif %(File)Arg_1"Gif-Datei anzeigen:"
    WINDOW_TYPE      NO_STDIO
    DESCRIPTION      Eine Datei doppelklicken oder übergeben, um
                    die GIF-Anzeige zu starten.
}

```

Da die Definition kein Feld `ICON` enthält, verwendet die Aktion das Standardsymbol des Systems.

- Die folgende Zuordnungsaktion eingeben, um die Aktion `GifViewer` mit der Aktion `View`, die in der Datentypdefinition aufgelistet ist, zu verbinden. Mit dem Feld `ARG_TYPE` wird die Aktion `View` auf Dateien vom Typ `Gif` eingeschränkt.

```

ACTION View
{
    ARG_TYPE          Gif
    TYPE              MAP
    MAP_ACTION        GifViewer
}

```

- Die Datei sichern.
- Auf 'Aktionen erneut laden' in der Gruppe von Anwendungen Desktop-Werkzeuge doppelklicken, um die Datenbank erneut zu lesen.

Datenattribute eines Datentyps definieren

Die Definition `DATA_ATTRIBUTES` definiert das Erscheinungsbild und das Verhalten des Datentyps. Sie gibt den Namen des Datentyps an und bietet die Möglichkeit, zusätzlich folgendes anzugeben:

- das Dateimanagersymbol (Feld `ICON`)
- das Verhalten bei Doppelklicken und den Inhalt des Menüs 'Ausgewählt' (Feld `ACTIONS`)
- die Kontexthilfe des Datentyps (Feld `DESCRIPTION`)

Symbolbild angeben, das für einen Datentyp verwendet wird

Das Feld `ICON` wird verwendet, um das Symbol anzugeben, das im Dateimanager verwendet wird. Wird kein Symbolbild angegeben, wird im Dateimanager nur eine Bezeichnung angezeigt.

Bei dem Wert des Felds `ICON` kann es sich um folgendes handeln:

- Einen Dateinamen ohne Pfad.

Der Dateiname ohne Pfad ist der Name der Datei, die das Symbolbild enthält, ohne die Dateinamenerweiterungen für Größe (`m` und `t`) und Abbildtyp (`bm` und `pm`). Beispielsweise wird für Dateien mit den Namen `GameIcon.m.pm` und `GameIcon.t.pm` der Name `GameIcon` verwendet.

Wird der Dateiname ohne Pfad verwendet, müssen die Symboldateien in ein Verzeichnis im Symbolsuchpfad gestellt werden:

- Persönliche Symbole: *Home-Verzeichnis* / `.dt/icons`
- Systemweite Symbole: `/etc/dt/appconfig/icons/sprache`
- Einen absoluten Pfad zu der Symboldatei, einschließlich des vollständigen Dateinamens.

Der absolute Pfad sollte nur verwendet werden, wenn die Symboldatei sich nicht im Symbolsuchpfad befindet. Befindet sich beispielsweise die Symboldatei `GameIcon.m.pm` im Verzeichnis `/doc/projects`, das sich nicht im Symbolsuchpfad befindet, wird im Feld `ICON` der Wert `/doc/projects/GameIcon.m.pm` gesetzt.

Tabelle 11-1 enthält eine Liste mit Symbolgrößen, die der Benutzer erstellen sollte, und den zugehörigen Dateinamen.

Tabelle 11-1 Symbolnamen und Größen für Datentypsymbole

Größe in Pixel	Bitmap-Name	Pixmap-Name
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>

Datentypen Aktionen zuordnen

Es gibt zwei Möglichkeiten, Datentypen Aktionen zuzuordnen:

- In dem Feld `ACTIONS` in der Definition `DATA_ATTRIBUTES` werden die Aktionen aufgelistet, die im Menü 'Ausgewählt' des Dateimanagers erscheinen. Bei der ersten Aktion in der Liste handelt es sich um die Standardaktion, die auch durch Doppelklicken aktiviert werden kann.
- Aktionen können mit dem Feld `ARG_TYPE` der Aktionsdefinition auf angegebene Datentypen beschränkt werden.

Beispielsweise erstellt die folgende Datentypdefinition einen Datentyp für spezielle, vom Systemadministrator erstellte "Readme"-Dateien mit der Namenskonvention `*.rm`.

```
DATA_ATTRIBUTES SysReadmeFile
{
    ICON                SysReadMe
    ACTIONS              Open, Respond
}
DATA_CRITERIA SysReadmeFileCriteria
{
    NAME_PATTERN        *.rm
    DATA_ATTRIBUTES_NAME SysReadmeFile
}
```

Eine spezielle Aktion 'Respond' für die Datei wird im folgenden definiert. Dadurch wird eine schreibbare Kopie der Datei im Texteditor geöffnet. Wird die Datei gesichert und der Texteditor verlassen, wird die Datei über elektronische Post an den Systemadministrator gesendet (Adresse `sysadmin@utd`).

```
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
}
```

Dateien auf der Grundlage des Datentyps verdecken

Ist eine Datei ein verdeckter Datentyp, wird sie nicht im Dateimanager angezeigt.

Im Feld `PROPERTIES` in der Definition `DATA_ATTRIBUTES` wird angegeben, daß Objekte dieses Typs verdeckt werden:

```
PROPERTIES      invisible
```

Verhalten der Datei bei Bearbeitung angeben

Die folgenden Felder `DATA_ATTRIBUTES` werden in erster Linie von Anwendungsprogrammierern verwendet. Hier wird angegeben, wie sich Dateien verhalten, wenn der Benutzer verschiedene Desktop-Aktivitäten durchführt.

Weitere Informationen können der Publikation *Common Desktop Environment Programmer's Guide*, entnommen werden, das Bestandteil der Entwicklerumgebungsdokumentation ist.

Feld	Beschreibung
<code>MOVE_TO_ACTION</code>	Für Container, wie zum Beispiel Verzeichnisse. Gibt an, daß eine Aktion ausgeführt wird, wenn eine Datei in einen Container dieses Datentyps verschoben wird.
<code>COPY_TO_ACTION</code>	Für Container, wie zum Beispiel Verzeichnisse. Gibt an, daß eine Aktion ausgeführt wird, wenn eine Datei in einen Container dieses Datentyps kopiert wird.
<code>LINK_TO_ACTION</code>	Gibt an, daß eine Aktion ausgeführt wird, wenn eine Datei mit einer Datei dieses Datentyps verbunden wird.
<code>IS_TEXT</code>	Gibt an, daß Dateien dieses Datentyps Text enthalten, der in einem Textfenster angezeigt werden kann.
<code>MEDIA</code>	Gibt den entsprechenden ToolTalk-Mediatyp an.
<code>MIME_TYPE</code>	Gibt den entsprechenden MIME-Typ an.
<code>X400_TYPE</code>	Gibt den entsprechenden X400-Typ an.

Datenkriterien für einen Datentyp definieren

Die Definition `DATA_CRITERIA` definiert die Kriterien für die Zuordnung eines Objekttyps zu einer Datei oder einem Verzeichnis

Folgende Kriterien können für die Objektzuordnung verwendet werden:

Kriterium	Beschreibung
Dateiname	Der Dateiname muß einem angegebenen Muster entsprechen. Das Feld <code>NAME_PATTERN</code> verwenden.
Dateiposition	Der Pfad muß einem angegebenen Muster entsprechen. Das Feld <code>PATH_PATTERN</code> verwenden.
Dateiinhalt	Ein angegebener Teil des Inhalts der Datei muß angegebenen Daten entsprechen. Das Feld <code>CONTENT</code> verwenden.
Dateimodus	Die Datei muß über die angegebenen Berechtigungen verfügen (Lesen, Schreiben, Ausführen, Verzeichnis). Das Feld <code>MODE</code> verwenden.
Symbolische Verbindungen	Die Datentypzuordnung erfolgt auf der Grundlage der Datei, mit der das Objekt verbunden wird.

Für einen Datentyp kann mehr als ein Kriterium verwendet werden. Aber die Kriterien `NAME_PATTERN` und `PATH_PATTERN` sollten nicht in demselben Datentyp verwendet werden.

Datentypenzuordnung auf der Grundlage des Dateinamens

Das Feld `NAME_PATTERN` verwenden, um die erforderlichen Merkmale des Dateinamens anzugeben. Das Feld kann die folgenden Platzhalterzeichen enthalten:

- ? Entspricht einem beliebigen einzelnen Zeichen
- * Entspricht einer beliebigen Folge von Zeichen (einschließlich einer leeren Zeichenfolge)
- [cc...] Entspricht einem beliebigen der in Klammern gesetzten Zeichen (c)
- [c-c] Entspricht einem beliebigen Zeichen aus dem Bereich von c bis c

Beispiele

- Die folgende Datentypdefinition erstellt einen Datentyp auf der Grundlage des Dateinamens. Der Dateiname muß mit `QS` anfangen und mit `.doc` enden.

```

DATA_ATTRIBUTES QS_Doc
{
  DESCRIPTION      Diese Datei enthält ein Dokument zum QS-Projekt.
  ICON             Word_Doc
  ACTIONS          Open
}

DATA_CRITERIA QS_Doc_Criteria
{
  NAME_PATTERN     QS*.doc
  DATA_ATTRIBUTES_NAME QS_Doc
}

```

- Die folgende Definition erstellt einen Datentyp für Verzeichnisse mit dem Namen Demo_n, wobei n den Zahlen 0 bis 9 entspricht.

```

DATA_ATTRIBUTES Demo_directory
{
  DESCRIPTION      Dies ist ein Verzeichnis. Zum Öffnen
doppelklicken.
  ICON             Demo
  ACTIONS          OpenInPlace,OpenNewView
}

DATA_CRITERIA Demo_directory_criteria
{
  NAME_PATTERN     Demo_[0-9]
  MODE             d
  DATA_ATTRIBUTES_NAME Demo_directory
}

```

Datentypenzuordnung auf der Grundlage der Position

Das Feld PATH_PATTERN verwenden, um den Pfad anzugeben. Es können dieselben Platzhalterzeichen verwendet werden, wie bei NAME_PATTERN.

Beispielsweise verwendet der folgende Datentyp ein Kriterium auf der Grundlage des Pfads.

```

DATA_ATTRIBUTES Project_Graphics
{
  DESCRIPTION      Grafikdatei zum QS-Projekt. Symbol doppelklicken, um
die Grafik anzuzeigen.
  ICON             QSgraphics
}
DATA_CRITERIA Project_Graphics_Criteria

```

```
{
  DATA_ATTRIBUTES_NAME   Project_Graphics
  PATH_PATTERN            */projects/QS/graphics/*
}
```

Datentypenzuordnung auf der Grundlage des Dateinamens und der Position

Soll ein Datentyp auf der Grundlage des Dateinamens als auch der Position erstellt werden, muß der Name in dem Wert für das Feld `PATH_PATTERN` enthalten sein. `NAME_PATTERN` und `PATH_PATTERN` können nicht in derselben Kriteriendefinition verwendet werden.

Beispiele

- Der im folgenden definierte Datentyp `QS_Source_Files` gilt für alle Dateien mit dem Namen `appn.c`, wobei $n = 1$ bis 9 , die sich in den Unterverzeichnissen von `*/projects/QS` befinden.

```
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
}
```

- Der folgende Datentyp gilt für alle Dateien im Verzeichnis `/doc/project1` mit dem Namen `chnn.xxx`, wobei $n = 0$ bis 9 ist, und `xxx` eine beliebige aus drei Zeichen bestehende Dateinamenerweiterung ist.

```
DATA_ATTRIBUTES ChapterFiles
{
  DESCRIPTION            Kapiteldatei für das Projektdokument.
  ICON                   chapter
  ACTIONS                Edit,Print
}

DATA_CRITERIA Chapter_Criteria
{
  PATH_PATTERN            /doc/project1/ch[0-9][0-9].???
  DATA_ATTRIBUTES_NAME   ChapterFiles
}
```

Dateimodi als Kriterium der Datentypzuordnung verwenden

Im Feld `MODE` die erforderlichen Berechtigungen angeben.

Moduskriterien werden normalerweise in Kombination mit der Datentypzuordnung auf der Grundlage von Namen, Position oder Inhalt verwendet. Sie ermöglichen dem Benutzer, einen Datentyp auf eine Datei oder ein Verzeichnis einzuschränken oder die erforderlichen Lese-, Schreib- und Ausführungsberechtigungen anzugeben.

Das Feld `MODE` kann die folgenden logischen Operatoren und Zeichen enthalten:

Operator	Beschreibung
-----------------	---------------------

!	Logischer Operator NICHT
&	Logischer Operator UND
	Logisches ODER

Zeichen	Beschreibung
----------------	---------------------

f	Der Datentyp gilt nur für Dateien
d	Der Datentyp gilt nur für Verzeichnisse
r	Die Datei kann von jedem Benutzer gelesen werden
w	In die Datei kann von jedem Benutzer geschrieben werden
x	Die Datei kann von jedem Benutzer ausgeführt werden
l	Die Datei stellt eine Verbindung dar

Standardmäßig hat der Modus keinen Einfluß.

Beispiele

- Die folgenden Modusfelder schränken den Datentyp wie folgt ein:

f&!w	Schreibgeschützte Dateien
!w	Schreibgeschützte Dateien und Verzeichnisse
f&x	Ausführbare Dateien
f&r&x	Dateien, in die geschrieben werden kann, und die ausführbar sind

x|!w Dateien, die ausführbar oder schreibgeschützt sind

- Die folgende Datentypdefinition erstellt einen Datentyp für schreibgeschützte, nicht ausführbare Dateien, deren Namen der Namenskonvention *.doc entsprechen. Es wird angenommen, daß eine Aktion View für den Datentyp definiert wurde.

```
DATA_ATTRIBUTES ReadOnlyDocument
{
    ICON          read_only
    DESCRIPTION   Dieses Dokument kann nicht beschrieben werden.
                  Ein Doppelklick führt den Editor mit einer schreibgeschützten
                  Kopie der Datei aus.
                  file.
    ACTIONS       View
}

DATA_CRITERIA ReadOnlyDocument_Criteria
{
    NAME_PATTERN   *.doc
    MODE           !d&!x&!w
    DATA_ATTRIBUTES_NAME  ReadOnlyDocument
}
```

Datentypzuordnung auf der Grundlage des Inhalts

Im Feld CONTENT wird angegeben, daß die Datentypzuordnung auf der Grundlage des Inhalts der Datei erfolgen soll. Die Datentypzuordnung auf der Grundlage des Inhalts einer Datei kann in Kombination mit der Datentypzuordnung auf der Grundlage von Namen oder Position verwendet werden.

Die Zuordnung kann auf der Grundlage einer Zeichenfolge oder des numerischen Inhalts einer Datei erfolgen. Das erste Byte in der Datei erhält die Zahl 0.

- Für die Zuordnung auf der Grundlage einer Zeichenfolge folgende Syntax verwenden:

```
CONTENT start_byte string zeichenfolge
```

- Für die Zuordnung auf der Grundlage des numerischen Inhalts folgende Syntax verwenden:

```
CONTENT start_byte byte nummer
CONTENT start_byte short nummer
CONTENT start_byte long nummer
```

- Für die Zuordnung auf der Grundlage des Inhalts eines Verzeichnisses folgende Syntax verwenden:

```
CONTENT 0 filename "datei_name"
```

Standard-C-Notation für Oktal- (beginnend mit `o`) und Hexadezimalzahlen (beginnend mit `oX`) verwenden.

Hinweis – Die Datentypzuordnung auf der Grundlage des Inhalts beeinträchtigt die Systemleistung. Der Benutzer sollte, wenn möglich, die Datentypzuordnung auf der Grundlage von Namen und Position verwenden.

Beispielsweise gilt der folgende Datentyp `Writable_Wingz` für alle Dateien mit Schreibberechtigung, die am Anfang der Datei die Zeichenfolge `WNGZ` enthalten.

```
DATA_ATTRIBUTES Writable_Wingz
{
  ...
}

DATA_CRITERIA Writable_Wingz_Criteria
{
  CONTENT          0 string WNGZ
  MODE             w&d
  DATA_ATTRIBUTES_NAME Writable_Wingz
}
```

▼ Datentypen mit mehreren unabhängigen Kriterien erstellen

Es kann ein Datentyp mit mehreren unabhängigen Kriterien erstellt werden, d.h. ein Datentyp, dem die Datei zugeordnet wird, wenn sie *einem* (oder beiden) Kriterien entspricht.

1. Die Definition `DATA_ATTRIBUTES` für den Datentyp erstellen.
2. Eine Definition `DATA_CRITERIA` für jedes Kriterium erstellen.

Das Feld `DATA_ATTRIBUTES_NAME` verwenden, um jedes Kriterium mit derselben Definition `DATA_ATTRIBUTES` zu verbinden.

Beispielsweise erstellen die folgenden Definitionen den Datentyp Mif . Die Datentypzuordnung erfolgt auf der Grundlage von Namen oder Inhalt.

```
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 zeichenfolge MIFFfile
}
```

Sprachenabhängige Datentypen erstellen

Der Suchpfad für Datentypen enthält sprachenabhängige Positionen. Das Desktop verwendet den Wert von LANG, um festzulegen, welche Positionen nach Datentypdefinitionen durchsucht werden.

Positionen für sprachenabhängige Datentypen

Sprachenabhängige Datentypdefinitionen müssen in die entsprechenden sprachenabhängigen Verzeichnisse im Aktionssuchpfad gestellt werden.

Der Standardsuchpfad lautet:

- Persönliche Aktionen: *Home-Verzeichnis*/.dt/types
- Systemweite Aktionen: /etc/dt/appconfig/types/*sprache*
- Integrierte Aktionen: /usr/dt/appconfig/types/*sprache*

▼ Datentypen sprachenabhängig erstellen

1. Eine Datei in dem zugehörigen sprachenabhängigen Verzeichnis erstellen (zum Beispiel in /etc/dt/appconfig/types/japanese) .

-
2. Die Datentypdefinition in die sprachenabhängige Konfigurationsdatei kopieren.
 3. Ein oder mehrere Felder der Datentypdefinition sprachenabhängig anpassen.

Folgenden Objekten werden Desktop-Symbole zugeordnet:

- Aktionsdateien und Datentypen im Dateimanager und im Anwendungsmanager
- Bedienelemente im Bedienfeld
- Anwendungsfenster in Symbolgröße
- Grafiken, die von Anwendungen verwendet werden, wie zum Beispiel Paletten und Funktionsleisten
- Hintergrundmuster des Arbeitsbereichs

<i>Symbolbilddateien</i>	<i>224</i>
<i>Symbolzuordnungen</i>	<i>226</i>
<i>Empfehlungen zur Erstellung von Symbolen</i>	<i>230</i>

Hinweis – Die Dokumentation zur Entwicklungsumgebung enthält zusätzliche Informationen zu Desktop-Symbolen. Weitere Informationen können Kapitel 4, “Visual Design,” der Publikation *Style Guide und Certification Checklist* entnommen werden.

Symbolbilddateien

Damit das Desktop ein Symbolbild verwenden kann, muß die Symbolbilddatei

- im korrekten Format sein.
- die korrekten Dateinamenskonventionen verwenden.
- den Desktop-Größenkonventionen entsprechen.
- sich in einem Verzeichnis im Symbolsuchpfad befinden.
- vom Desktop-Konstrukt mit der korrekten Syntax aufgerufen werden. Wird beispielsweise ein neues Bedienelement für das Bedienfeld erstellt, wird im Feld `ICON` in der Bedienfelddefinition definiert, welches Symbolbild für das Bedienelement verwendet wird.

Symboldateiformate

Für einen Farbbildschirm sollten Symboldateien im Format X-Pixmap (XPM) verwendet werden, die normalerweise die Erweiterung `.pm` haben. Ansonsten werden Dateien im Format X-Bitmap (XBM) verwendet, die die Erweiterung `.bm` haben. Wird in der Pixmap-Datei Transparenz verwendet, wird bei der Erstellung einer `.bm`-Datei eine Maskendatei (`_m.bm`) generiert. Weitere Informationen darüber, wie das Desktop diese Dateien findet, können dem Abschnitt "Symbolsuchpfad" auf Seite 136 entnommen werden.

Symboldateinamen

Jedes Symbol- und Hintergrundmusterbild ist in einer eigenen Datei gespeichert. Normalerweise wird ein Symbol mit dem Dateinamen ohne Pfad angegeben. Beispielsweise wird ein Symbol möglicherweise als `mail` bezeichnet, wenn die Datei unter folgendem Namen gespeichert ist:

```
/usr/dt/appconfig/icons/sprache/mail.1.pm
```

Die Dateinamenskonvention des Hinzufügens von Erweiterungen ermöglicht, Symbole nach Größe und Typ in Gruppen zusammenzufassen. Symbolnamen für Desktop-Komponenten haben im allgemeinen folgende Formate:

dateiname ohne pfad . gröÙe . format

oder

dateiname ohne pfad . format

wobei die Parameter folgende Bedeutung haben:

<i>dateiname ohne pfad</i>	Der Dateiname ohne Pfad des Bilds, mit dem auf das Bild bezug genommen wird
<i>größe</i>	Ein Buchstabe, der die Größe angibt: l (groß) m (mittel) s (klein) t (winzig)
<i>format</i>	Das Dateiformat: pm (Pixmap) bm (Bitmap)

Symbolgrößenkonventionen

Tabelle 12-1 zeigt die empfohlene Größe in Pixel für Desktop-Symbole.

Tabelle 12-1 Symbolgrößen und Dateinamen

Symbolgröße	Bitmap-Name	Pixmap-Name
16 x 16 (winzig)	<i>name.t.bm</i>	<i>name.t.pm</i>
24 x 24 (klein)	<i>name.s.bm</i>	<i>name.s.pm</i>
32 x 32 (mittel)	<i>name.m.bm</i>	<i>name.m.pm</i>
48 x 48 (groß)	<i>name.l.bm</i>	<i>name.l.pm</i>

Tabelle 12-2 zeigt die Symbolgrößen, die die Desktopkomponenten verwenden. In einigen Fällen ist die Größe des verwendeten Symbols vom Auflösungsvermögen des Bildschirms abhängig.

Tabelle 12-2 Desktop-Komponenten und deren Symbolgrößen

Desktop-Komponente	Hohe Auflösung	Mittlere Auflösung	Niedrige Auflösung
Dateimanager und Anwendungsmanager (Ansicht mit Name und Symbol)	mittel	mittel	mittel
Dateimanager und Anwendungsmanager (Ansicht mit Name und kleinem Symbol)	winzig	winzig	winzig
Bedienelemente im Hauptbedienfeld	groß	groß	mittel

Table 12-2 Desktop-Komponenten und deren Symbolgrößen

Desktop-Komponente	Hohe Auflösung	Mittlere Auflösung	Niedrige Auflösung
Bedientafeln des Bedienfelds	mittel	mittel	winzig
Umschalt symbole im Bedienfeld	klein	klein	winzig
Fenster in Symbolgröße	groß	groß	mittel

Wird beispielsweise ein Symbol mit dem Namen `mail` für einen Datentypen an einem Farbbildschirm definiert und ist die Vorgabe für den Dateimanager auf kleine Symbole gesetzt, wird das Symbolbild `mail.t.ppm` verwendet.

Symbolsuchpfad

Das Desktop sucht eine Symboldatei oder ein Bild, indem es eine Liste von Verzeichnissen nach der Datei durchsucht. Diese Liste von Verzeichnissen, die *Symbolsuchpfad* genannt wird, wird durch den Wert mehrerer Umgebungsvariablen bestimmt. Welche Variablen verwendet werden, und wie sie zum Symbolsuchpfad zusammengestellt werden, wird im Abschnitt "Symbolsuchpfad" auf Seite 226 erläutert.

Der Standardsuchpfad lautet:

- Integrierte Symbole: `/usr/dt/appconfig/icons/sprache`
- Systemweite Symbole: `/etc/dt/appconfig/icons/sprache`
- Persönliche Symbole: `Home-Verzeichnis/.dt/icons`

Auf Symbole über das Netzwerk zugreifen

Das Desktop kann auf Symbole in anderen Systemen zugreifen. Weitere Informationen zur Erstellung eines Symbol-Servers können dem Abschnitt "Datenbank-, Symbol- und Hilfe-Services" auf Seite 114 entnommen werden.

Symbolzuordnungen

Um eine schnellere Objekterkennung zu erreichen, können Symbole folgenden Objekten zugeordnet werden:

- Aktionen und Datentypen
- Bedienelementen im Bedienfeld und den Bedientafeln

- Anwendungsfenstern in Symbolgröße

Symboldateien angeben

Für Symbole, die für Aktionen, Datentypen und im Bedienfeld oder den Bedientafeln verwendet werden, muß nur der Dateiname ohne Pfad des Symbols angegeben werden (keine Erweiterungen). Die korrekten Erweiterungen werden je nach Auflösung des Bildschirms, Farbunterstützung, und den Ansichtsoptionen des Dateimanagers (zum Beispiel 'Mit kleinen Symbolen') automatisch hinzugefügt.

Um den Suchpfad außer Kraft zu setzen, den vollständigen Pfad und Namen des Symbols angeben.

▼ Symbole einer Aktion oder einem Datentyp zuordnen

1. Das Symbol in dem Feld `ICON` angeben.

Wird entsprechend den Namenskonventionen für Symboldateien vorgegangen, muß nur der Dateiname ohne Pfad des Symbols angegeben werden. Das korrekte Symbol wird entsprechend der Auflösung und der Farbunterstützung des Bildschirms angezeigt.

2. Die folgenden Symbolgrößen erstellen:
 - Aktionen: groß, mittel, und winzig
 - Datentypen: mittel und winzig

Beispiel für eine Aktionsdefinition

Bei dem folgenden Beispiel handelt es sich um eine Aktionsdefinition zum Starten des Zeichenwerkzeugs Island Paint™. Die Symbole `Ipaint.l` und `Ipaint.s` werden der Aktion zugeordnet.

```
ACTION          IslandPaintOpenDoc
{
    WINDOW_TYPE  NO-STDIO
    ICON         Ipaint
    EXEC_STRING  /usr/bin/IslandPaint %Arg_1"Datei öffnen:"%
}
```

Werden farbige Symbole verwendet, fügt das Desktop zunächst `.pm` an, wenn es nach den zugehörigen Symboldateien sucht. Ansonsten (oder wenn keine Entsprechung mit `.pm` gefunden wird) fügt das Desktop `.bm` an.

Beispiel für eine Datentypdefinition

Die folgende Datentypdefinition ordnet die Symbole `comprsd.l` und `comprsd.s` komprimierten Dateien zu.

```
DATA_ATTRIBUTES      COMPRESSED
{
    ICON              comprsd
    ACTIONS            Uncompress
    DESCRIPTION       EINE KOMPRIMIERTE Datei wurde mit dem Befehl
                      'compress' komprimiert,
                      um weniger Speicherplatz zu belegen.
}
```

▼ Symbole in einem Bedienelement im Bedienfeld anzeigen

1. Mit dem Feld `ICON` den Symbolnamen angeben.

Überwacht das Bedienelement eine Datei (das Feld `MONITOR_TYPE` ist auf `mail` oder `file` gesetzt), im Feld `ALTERNATE_ICON` angeben, welches Symbol verwendet werden soll, wenn eine Änderung festgestellt wird.

Der Animationseffekt kann auch für Tasten und Übergabebereichbedienelemente zur Verfügung gestellt werden.

2. Die folgenden Symbolgrößen erstellen:

- Hauptbedienfeld und Bedientafeln: groß, mittel und winzig
- Arbeitsbereichumschalter: klein

Beispiel

Das folgende Bedienelement ändert sein Erscheinungsbild, wenn eine Datei mit dem Namen `report` in das Verzeichnis `/doc/ftp/pub/` gestellt wird. Befindet sich die Datei nicht dort, wird das Symbol `NoReport.pm` angezeigt. Befindet sich die Datei dort, wird das Symbol `Report.pm` angezeigt.

```
CONTROL MonitorReport
{
    CONTAINER_NAME  container_name
    TYPE            ICON
    MONITOR_TYPE    file
    FILE_NAME       /doc/ftp/pub/report
    ICON            NoReport
    ALTERNATE_ICON  Report
}
```

▼ Symbole einem Anwendungsfenster zuordnen

1. Die Ressource `iconImage` für die Arbeitsbereichssteuerung wie folgt setzen:

```
Dtwm*clientname*iconImage: symbol_datei_name
```

Um den korrekten Wert für *clientname* festzustellen, den Anwendungsmanager öffnen und 'Fenstereigenschaften' in der Gruppe von Anwendungen 'Desktop-Werkzeuge' doppelklicken. Wird ein Fenster ausgewählt, werden dessen Eigenschaften aufgelistet. Das Merkmal `WM_CLASS` zeigt den Klassennamen des Fensters in Anführungszeichen an.

Weitere Informationen zum Festlegen von Ressourcen können dem Abschnitt "Anwendungsressourcen einstellen" auf Seite 282 entnommen werden.

2. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Um zu überprüfen, ob das Symbol vom Arbeitsbereichsmanager erkannt wurde, das Fenster, dessen Symbol geändert werden soll, auf Symbolgröße verkleinern.

Hinweis – Bei einigen Anwendungen ist es nicht möglich, das Standardfenstersymbol außer Kraft zu setzen.

▼ Dateimanager zur Anzeige von Symbolen verwenden

1. Die Datei `/usr/dt/examples/sprache/IconBrowse.dt` in das Verzeichnis `Home-Verzeichnis/.dt/types/Iconbrowse.dt` kopieren.
2. Den Anwendungsmanager öffnen und auf 'Aktionen erneut starten' in der Gruppe von Anwendungen 'Desktop-Werkzeuge' doppelklicken.

Wechselt der Benutzer in ein Verzeichnis, das Symbole enthält (`.bm-` und `.pm-` Dateien), wird jedes Symbol mit seinem Namen dargestellt. Wechselt der Benutzer beispielsweise in das Verzeichnis

```
/usr/dt/appconfig/icons/sprache, werden viele Desktop-Symbole angezeigt.
```

Hinweis – Die Aktivierung der Anzeige von Symbolen an Systemen mit geringer Speicherkapazität kann die Anzeige von Verzeichnissen im Dateimanager verlangsamen.

Bilddateien, die größer als 256 x 256 sind, werden in der Standardkonfiguration nicht angezeigt.

Um die Anzeige von Symbolen zu inaktivieren, muß folgendermaßen vorgegangen werden:

1. Die persönliche Kopie des Benutzers der Datei `IconBrowse.dt` löschen.
2. Den Anwendungsmanager öffnen und auf 'Aktionen erneut laden' in der Gruppe von Anwendungen Desktop-Werkzeuge doppelklicken.

Empfehlungen zur Erstellung von Symbolen

Symbole, die miteinander in Bezug stehen, ähnlich gestalten. Beispielsweise empfiehlt es sich, bei der Erstellung von Symbolen für eine Anwendung darauf zu achten, daß zwischen dem Symbol der Anwendung und den Symbolen für Datendateien zweckmäßige Ähnlichkeiten bestehen.

Sicherstellen, daß die zweifarbige Version eines farbigen Symbols akzeptabel ist. Wird das Symbol an einem Monochrombildschirm oder einem Graustufen-Monitor angezeigt (oder sind an einem Farbbildschirm nicht genügend Farben definiert), wird das Symbol automatisch in seiner zweifarbigen Version angezeigt.

Um die verwendeten Farben im System zu erhalten, sollte der Benutzer die Verwendung von Farben auf diejenigen Farben einschränken, die das Desktop zur Verfügung stellt (Symbole, die mit dem Symboleditor erstellt wurden, verwenden nur Desktop-Farben)

Die Symbolgrößen, die die Desktop-Komponenten verwenden, können Tabelle 12-1, "Symbolgrößen und Dateinamen," auf Seite 225 entnommen werden.

Farbpalette

Für Desktop-Symbole steht eine Palette mit folgenden 22 Farben zur Verfügung:

- Acht statische Grautöne
- Acht statische Farben: Rot, Blau, Grün, Zyanblau, Magenta, Gelb, Schwarz und Weiß
- Sechs dynamische Farben: Vordergrund, Hintergrund, obere Schattierung, untere Schattierung, Auswahl und Transparent

Mit dieser Palette werden attraktive, benutzerfreundliche Symbole erstellt, ohne Farbressourcen, die von anderen Anwendungen benötigt werden, zu übernehmen. Die meisten Symbole, die das Desktop zur Verfügung stellt, verwenden die Farbe Grau, die mit einer anderen Farbe abgesetzt wird.

Die transparente Farbe wird verwendet, um Symbole zu erstellen, die für den Betrachter auf den ersten Blick nicht rechteckig sind, da die Farbe hinter dem Symbol durchscheint.

0Erweiterte Anpassung des Bedienfelds

Der Benutzer kann das Bedienfeld über dessen Menüs und das Bedienelement 'Symbol installieren' in den Bedientafeln anpassen.

In diesem Kapitel wird die Anpassung des Bedienfelds durch das Erstellen und Bearbeiten von Konfigurationsdateien behandelt.

<i>Bedienfeldkonfigurationsdateien</i>	<i>234</i>
<i>Benutzerschnittstellenanpassungen verwalten</i>	<i>236</i>
<i>Organisation der Bedienfelddefinition</i>	<i>238</i>
<i>Hauptanzeige ändern</i>	<i>242</i>
<i>Bedientafeln erstellen und ändern</i>	<i>247</i>
<i>Bedienelemente im Bedienfeld definieren</i>	<i>251</i>
<i>Arbeitsbereichsumschalter anpassen</i>	<i>260</i>
<i>Allgemeine Bedienfeldkonfiguration</i>	<i>261</i>

Weiterführende Informationen

- Referenzinformationen zu Bedienelementen im Bedienfeld und der Konfiguration des Bedienfelds können der Hilfeseite `dtfpfile(4X)` entnommen werden.

- Referenzinformationen zum Arbeitsbereichsmanager können der Hilfeseite `dtwm(1)` and `dtwmrc(4)` entnommen werden.

Bedienfeldkonfigurationsdateien

Das Bedienfeld wird in einer Datenbank mit Konfigurationsdateien definiert.

Über diese Konfigurationsdateien kann das Bedienfeld angepaßt werden. Bestimmte Änderungen können nur durch Bearbeitung der Konfigurationsdatei durchgeführt werden. Dazu gehören

- das Hinzufügen eines neuen Bedienelements im Hauptbedienfeld.
- das Hinzufügen spezieller Bedienelemente, wie zum Beispiel Client-Fenster.
- das Ändern bestimmter Standardverhaltensweisen, zum Beispiel, ob die Bedienelemente im Bedienfeld durch einfaches Klicken oder durch Doppelklicken aktiviert werden.

Um dem Benutzer einen möglichst großen Spielraum bei der Konfiguration des Bedienfelds zu lassen, können die Konfigurationsdateien für einen einzelnen Benutzer definiert (persönlich), für das System definiert (systemweit) oder auf anderen Systemen gespeichert sein.

Das Bedienfeld wird durch den Arbeitsbereichsmanager erstellt und verwaltet.

Standardkonfigurationsdatei des Bedienfelds

Das Standardbedienfeld wird in der Bedienfeldkonfigurationsdatei `/usr/dt/appconfig/types/sprache/dtwm.fp` definiert.

Diese Datei sollte nicht geändert werden.

Suchpfad für Bedienfeldkonfigurationsdateien

Die Bedienfelddefinition kann auf beliebig viele Dateien im lokalen oder in anderen Systemen verteilt sein.

Dateien, die das Bedienfeld definieren, müssen folgenden Anforderungen entsprechen:

- Der Dateiname muß die Erweiterung `.fp` tragen, z. B., `mail.fp`.
- Die Datei muß sich im Datenbanksuchpfad für Aktionen befinden.

Der Standarddatenbanksuchpfad für Aktionen enthält folgende Verzeichnisse, die in der hier aufgeführten Reihenfolge durchsucht werden:

<i>Home-Verzeichnis</i> / .dt/types	Persönliche Anpassungen
/etc/dt/appconfig/types/ <i>sprache</i>	Systemweite Anpassungen
/usr/dt/appconfig/types/ <i>sprache</i>	Integriertes Bedienfeld und Bedienelemente

Ein zusätzliches Verzeichnis, *Home-Verzeichnis* / .dt/types/*fp_dynamic* wird verwendet, um persönliche Anpassungen, die über die Benutzerschnittstelle durchgeführt wurden, zu speichern. Dieses Verzeichnis nicht für manuelle Anpassungen verwenden..

Der Datenbanksuchpfad für Aktionen enthält möglicherweise zusätzliche Verzeichnisse, die das System für das Netzwerk konfigurieren. Insbesondere werden zusätzliche Positionen auf anderen Systemen hinzugefügt, wenn das System für den Zugriff auf einen Anwendungs-Server konfiguriert wird. Weitere Informationen können dem Abschnitt "Datenbanksuchpfad (Aktionen/Datentypen)" auf Seite 134 entnommen werden.

Wie das Bedienfeld assembliert wird: Vorrangregeln

Das Bedienfeld wird aus allen Konfigurationsdateien im Datenbanksuchpfad für Aktionen assembliert.

Kommt es zu einem Konflikt zwischen Komponenten der Definition, bestimmen Vorrangregeln, welche Definition verwendet wird. Zwei Komponenten geraten in Konflikt miteinander, wenn

- sie denselben Bedienelementnamen, CONTAINER_NAME und CONTAINER_TYPE haben
- Oder für sie dieselbe Position definiert ist (zwar unterschiedliche Namen, aber derselbe CONTAINER_NAME, CONTAINER_TYPE und POSITION_HINTS).

Für das Bedienfeld gelten folgende Vorrangregeln:

- Haben Komponenten denselben Bedienelementnamen, Container-Namen und -typ, wird die Komponente verwendet, die zuerst gelesen wird.

Enthalten, beispielsweise, sowohl ein systemweites als auch ein integriertes Bedienelement die folgenden Felder, sind aber ansonsten unterschiedlich,

```
CONTROL TextEditor
{
  CONTAINER_TYPE BOX
  CONTAINER_NAME Top
  ...
}
```

hat das systemweite Bedienelement Vorrang.

- Konkurrieren zwei Komponenten für dieselbe Position, werden sie in der Reihenfolge gesetzt, in der sie gelesen werden.

Erstellt ein Benutzer, beispielsweise, ein neues persönliches Bedienelement für das Hauptbedienfeld (CONTAINER_TYPE BOX und CONTAINER_NAME Top) und ordnet ihm POSITION_HINTS 5 zu, schiebt das persönliche Bedienelement das integrierte Bedienelement und alle anderen Bedienelemente mit höheren Positionsnummern um eine Position nach rechts.

Hinweis – Wird ein Bedienelement geändert, indem eine neue systemweite oder persönliche Version davon erstellt wird, muß in der neuen Bedienelementdefinition derselbe Bedienelement name, CONTAINER_NAME, und CONTAINER_TYPE angegeben werden. Sonst wird das neue Bedienelement dem existierenden Bedienelement hinzugefügt.

Dynamisch erstellte Bedienfelddateien

Paßt der Benutzer das Bedienfeld über das Bedienelement 'Symbol installieren' und Menüs an, werden Dateien in das Verzeichnis *Home-Verzeichnis/.dt/types/fp_dynamic* geschrieben.

Das Bedienfeld erstellt eine zusätzliche Datei, *Home-Verzeichnis/.dt/sessions/dtwmfp.session*, in der der Status des angepaßten Bedienfelds für jede Session gesichert und wiederhergestellt wird.

Benutzerschnittstellenanpassungen verwalten

Über die Menüs der Bedienelemente und die Bedienelemente 'Symbol installieren' kann der Benutzer das Bedienfeld weitreichend anpassen.

Im folgenden Abschnitt wird beschrieben, wie

- bestimmte persönliche Anpassungen ausgeschlossen werden. Beispielsweise kann ausgeschlossen werden, daß ein Benutzer ein Bedienelement löscht.
- persönliche Anpassungen widerrufen werden. Beispielsweise kann ein Benutzer die Wiederherstellung eines versehentlich gelöschten Bedienelements anfordern.

▼ Persönliche Anpassungen ausschließen

1. Handelt es sich bei dem Bedienelement um ein integriertes Bedienelement, die Definition aus `/usr/dt/appconfig/types/sprache/dtwm.fp` in `/etc/dt/appconfig/types/sprache/name.fp` kopieren.
2. Die folgende Zeile zur Bedienelementdefinition hinzufügen:

```
LOCKED    True
```

▼ Gelöschte Bedienelemente oder gelöschte Bedientafeln wiederherstellen

Die Aktion 'Bedienfeld wiederherstellen' in der Gruppe von Anwendungen 'Desktop-Werkzeuge' entfernt alle Bedienfeldanpassungen, die über die Benutzerschnittstelle durchgeführt wurden. Benutzer können mit dieser Aktion alle persönlichen Anpassungen, die sie über die Menüs des Bedienfelds erstellt haben, entfernen.

Mit der folgenden Prozedur ein einzelnes Bedienelement wiederherstellen:

- ♦ Im Verzeichnis *Home-Verzeichnis*/`.dt/types/fp_dynamic` die Datei entfernen, die erstellt wurde, als das Bedienelement vom Benutzer gelöscht wurde. Das Bedienelement trägt denselben Namen, wie das ursprünglich gelöschte.

Wurde, beispielsweise, das Bedienelement `Symboleditor` gelöscht, enthält eine Datei im Verzeichnis `fp_dynamic` folgenden Eintrag:

```
CONTROL IconEditor
{
...
  DELETE    True
}
```

Wird eine Bedientafel gelöscht, wird eine separate dynamische Datei für die Bedientafel und für jedes Bedienelement in der Bedientafel erstellt.

Organisation der Bedienfelddefinition

Das Bedienfeld wird durch Assemblierung der Definitionen seiner Komponenten aufgebaut. Für jede dieser Komponenten ist eine bestimmte Syntax erforderlich, die definiert, wo im Bedienfeld die Komponente steht, wie sie aussieht und wie sie sich verhält.

Bedienfeldkomponenten

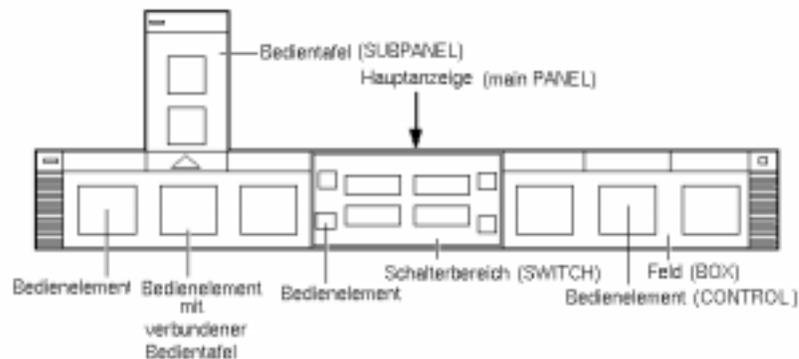


Abbildung 13-1 Bedienfeldkomponenten

Das Bedienfeld ist folgendermaßen aufgebaut:

- Das Bedienfeld (PANEL) ist der Container der obersten Ebene, das übergeordnete Verzeichnis, des gesamten Bedienfelds.
- Das PANEL ist ein Container für ein oder mehrere Felder (BOXen).
- Eine BOX ist ein Container für ein oder mehrere Bedienelemente (CONTROLS).

Es gibt zwei spezielle Typen von Containern:

- Eine Bedientafel (SUBPANEL) ist einem bestimmten Bedienelement zugeordnet (das Bedienelement ist der Container für die Bedientafel). Bedientafeln öffnen sich von dem Bedienelement, dem sie zugeordnet sind, nach oben.

- Der Arbeitsbereichsumschalter (SWITCH) enthält die Tasten zum Wechseln der Arbeitsbereiche sowie zusätzliche Bedienelemente.

Allgemeine Syntax der Bedienfelddefinition

Jede Komponente im Bedienfeld wird einzeln definiert. Dabei wird folgende Syntax verwendet:

```
COMPONENT name
{
  KEYWORD      wert
  KEYWORD      wert
  ...
}
```

Einige Schlüsselwörter sind erforderlich, andere sind wahlfrei. Weitere Informationen können der Hilfeseite `dtfpfile(4X)` entnommen werden.

Bedienfelddefinition (PANEL)

PANEL ist die Komponente der obersten Ebene. Die Definition enthält

- den Bedienfeldnamen
- Felder, die das allgemeine Erscheinungsbild und das Verhalten des gesamten Bedienfelds beschreiben.

```
PANEL bedienfeld_name
{
  KEYWORD      wert
  KEYWORD      wert
  ...
}
```

Der Parameter *bedienfeld_name* ist ein eindeutiger Name für das Bedienfeld. Der Standardname ist "FrontPanel."

Felddefinitionen (BOX)

Eine Definition für BOX beschreibt

- den Namen der BOX
- in welchem Bedienfeld (PANEL) sich das Feld befindet (CONTAINER_NAME)
- die Position des Felds im Bedienfeld (POSITION_HINTS)

- Felder, die das allgemeine Erscheinungsbild und das Verhalten des gesamten Felds beschreiben

```
BOX feld_name
{
    CONTAINER_NAME      bedienfeld_name
    POSITION_HINTS       position
    KEYWORD              wert
    KEYWORD              wert
    ...
}
```

Bedienelementdefinitionen (CONTROL)

Eine Definition für CONTROL beschreibt

- den Namen des Bedienelements (CONTROL)
- ob es sich bei dem Bedienelement um ein Feld, eine Bedientafel oder den Arbeitsbereichsumschalter handelt (CONTAINER_TYPE)
- in welchem/r Feld, Bedientafel oder Arbeitsbereichsumschalter sich das Bedienelement befindet (CONTAINER_NAME)
- die Position des Bedienelements im Feld (POSITION_HINTS).
- Felder, die das allgemeine Erscheinungsbild und das Verhalten des Bedienelements beschreiben

```
CONTROL bedienelement_name
{
    CONTAINER_TYPE      BOX or SUBPANEL or SWITCH
    CONTAINER_NAME      feld_name oder bedientafel_name oder schalter_name
    TYPE                bedienelement_typ
    POSITION_HINTS       position
    KEYWORD              wert
    KEYWORD              wert
    ...
}
```

Bedientafeldefinitionen (SUBPANEL)

Eine Definition für SUBPANEL beschreibt

- den Namen der Bedientafel (SUBPANEL)
- den Namen des Bedienelements, dem die Bedientafel zugeordnet ist (CONTAINER_NAME)

- Felder, die das spezifische Erscheinungsbild und Verhalten der Bedientafel beschreiben

```
SUBPANEL bedientafel_name
{
  CONTAINER_NAME      bedienelement_name
  KEYWORD              wert
  KEYWORD              wert
  ...
}
```

Arbeitsbereichsumschalterdefinition (SWITCH)

Die Definition für SWITCH beschreibt

- den Namen des Arbeitsbereichsumschalters
- in welchem Feld sich derArbeitsbereichsumschalter befindet (CONTAINER_NAME)
- die Position des Arbeitsbereichsumschalters innerhalb des Felds (POSITION_HINTS)
- Felder, die das Erscheinungsbild und Verhalten des Arbeitsbereichsumschalters beschreiben

```
SWITCH schalter_name
{
  CONTAINER_NAME      feld_name
  POSITION_HINTS       position
  KEYWORD              wert
  KEYWORD              wert
  ...
}
```

Hauptanzeige ändern

Bei der Hauptanzeige handelt es sich um das Bedienfeldfenster ohne die Bedientafeln.

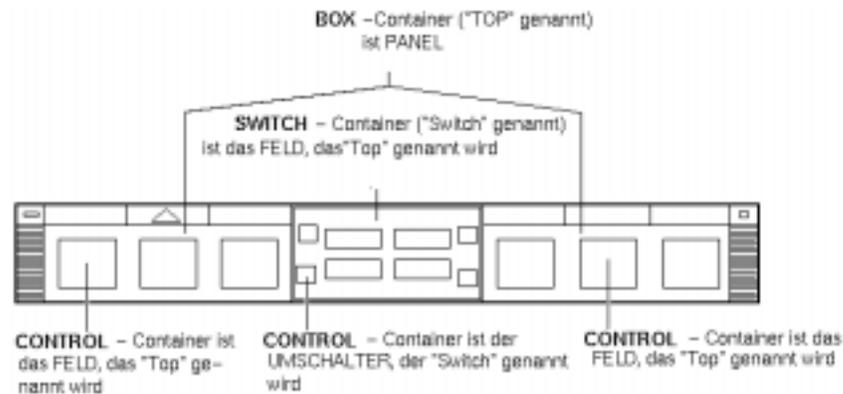


Abbildung 13-2 Container der Hauptanzeige

Folgende Änderungen können vorgenommen werden:

- Hinzufügen oder Entfernen von Bedienelementen
- Austauschen der Positionen von Bedienelementen

▼ Bedienelemente der Hauptanzeige hinzufügen

1. Eine Bedienfeldkonfigurationsdatei erstellen:

- systemweit: `/etc/dt/appconfig/types/sprache/*.fp`
- persönlich: `Home-Verzeichnis/.dt/types/*.fp`

2. Das Bedienelement in der Datei definieren.

Mit den Feldern `CONTAINER_NAME` und `CONTAINER_TYPE` den Container für das Bedienelement spezifizieren:

```
CONTAINER_NAME Top
CONTAINER_TYPE BOX
```

Mit `POSITION_HINTS` die Position von links nach rechts für das Bedienelement angeben. Da Anpassungen Vorrang vor integrierten Bedienelementen haben, schiebt das neue Bedienelement das bereits vorhandene Bedienelement um eine Position nach rechts.

3. Die Konfigurationsdatei sichern.

4. Ein Symbol für das Bedienelement im Bedienfeld erstellen.

Nähere Informationen hierzu können dem Abschnitt “Symbol angeben, das von einem Bedienelement verwendet werden soll” auf Seite 246 entnommen werden.

5. ‘Arbeitsbereichsmanager neu starten’ im Arbeitsbereichsmenü auswählen.

Beispielsweise stellt die folgende Bedienelementdefinition in der Datei `/etc/dt/appconfig/types/sprache/audio.fp` ein Bedienelement für eine Audio-Anwendung zwischen die Bedienelemente für Uhr und Kalender.

```
CONTROL AudioApplication
{
  TYPE                icon
  CONTAINER_NAME      Top
  CONTAINER_TYPE      BOX
  ICON                AudioApp
  POSITION_HINTS        2
  PUSH_ACTION         StartAudioApplication
  PUSH_RECALL         true
}
```

▼ Bedienelemente entfernen

1. Eine Bedienfeldkonfigurationsdatei erstellen:

- systemweit: `/etc/dt/appconfig/types/sprache/name.fp`
- persönlich: `Home-Verzeichnis/.dt/types/name.fp`

2. Die Definition des Bedienelements, das gelöscht werden soll, in die neue Datei kopieren.

Handelt es sich um ein integriertes Bedienelement, befindet sich die Definition in `/usr/dt/appconfig/types/sprache/dtwm.fp`.

Es ist nicht notwendig, die gesamte Definition zu kopieren. Die Felder `CONTAINER_NAME` und `CONTAINER_TYPE` müssen jedoch auf jeden Fall kopiert werden.

3. Das Feld `DELETE` der Definition hinzufügen:

```
DELETE    True
```

4. Die Konfigurationsdatei sichern.

5. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Beispielsweise entfernt die folgende Bedienelementdefinition in der Datei `/etc/dt/appconfig/types/sprache/TrashCan.fp` das Bedienelement 'Papierkorb' aus dem Bedienfeld.

```
CONTROL Trash
{
  CONTAINER_NAME    Top
  CONTAINER_TYPE    BOX
  DELETE             True
}
```

▼ Bedienelemente ändern

Zum Ändern einer Bedienelementdefinition, zum Beispiel eines Symbolbilds, die folgende Prozedur verwenden.

1. Die gesamte Bedienelementdefinition aus `/usr/dt/appconfig/types/sprache/dtwm.fp` in folgende Dateien kopieren:
 - systemweit: `/etc/dt/appconfig/types/sprache/name.fp`
 - persönlich: `Home-Verzeichnis/.dt/types/name.fp`.
2. Das Feld, das geändert werden soll, bearbeiten. Es können auch zusätzliche Felder hinzugefügt werden.
3. Die Datei sichern
4. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

▼ Positionen von Bedienelementen austauschen

1. Die Bedienelementdefinitionen der Bedienelemente, deren Positionen geändert werden sollen, von `/usr/dt/appconfig/types/sprache/dtwm.fp` in folgende Dateien kopieren:

- systemweit: `/etc/dt/appconfig/types/sprache/name.fp`
- persönlich: `Home-Verzeichnis/.dt/types/name.fp`.

Die gesamte Bedienelementdefinition für jedes Bedienelement, das verschoben werden soll, kopieren.

2. Die Werte der Felder `POSITION_HINTS` der Bedienelementdefinitionen austauschen.
3. Die Datei sichern
4. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Beispielsweise tauschen die folgenden Definitionen in einer Datei `/etc/dt/appconfig/types/C/MailHelp.fp` die Positionen der Bedienelemente für Post und Hilfemanager aus und sperren diese gegen persönliche Änderungen einzelner Benutzer.

```
CONTROL Mail
{
    POSITION_HINTS    12
    LOCKED           True
    ...der rest der definition des bedienelements
}

CONTROL Help
{
    POSITION_HINTS    5
    LOCKED           True
    ...der rest der definition des bedienelements
}
```

▼ Ein Bedienelement im Bedienfeld ersetzen

- ◆ Eine weitere Bedienelementdefinition mit denselben Parametern erstellen:
 - `bedienelement_name`
 - `CONTAINER_NAME` Wert

Beispielsweise sind die beiden folgenden Bedienelemente in zwei verschiedenen Konfigurationsdateien definiert. Die Bedienelemente haben denselben Bedienelementnamen und Container-Namen und werden daher als dasselbe Bedienelement betrachtet.

- Definition in `/etc/dt/appconfig/types/C/SysControls.fp`:

```
Control ImportantApplication
{
  CONTAINER_NAME      Top
  CONTAINER_TYPE      BOX
  POSITION_HINTS       2
  ...
```

- Definition in *Home-Verzeichnis*/`.dt/types/MyControls.fp`:

```
Control ImportantApplication
{
  CONTAINER_NAME      Top
  CONTAINER_TYPE      BOX
  POSITION_HINTS       6
  ...
```

Das persönliche Bedienelement hat Vorrang, so daß das Bedienelement an die Position 6 gestellt wird.

Symbol angeben, das von einem Bedienelement verwendet werden soll

Das Feld `ICON` der Bedienelementdefinition definiert, welches Symbolbild für das Bedienelement verwendet wird.

Der Wert für das Feld `ICON` kann auf die folgenden Weisen angegeben werden:

- durch den Dateinamen ohne Pfad

Der Dateiname ohne Pfad ist der Name der Datei, die das Symbolbild enthält, ohne die Dateinamenerweiterungen für Größe (`m` und `t`) und Abbildtyp (`bm` und `pm`). Beispielsweise lautet bei den Dateien mit den Namen `MyGame.l.pm` und `MyGame.m.pm` der Dateiname ohne Pfad `MyGame`.

Wird der Dateiname ohne Pfad verwendet, müssen sich die Symboldateien in einem Verzeichnis im Symbolsuchpfad befinden:

- persönliche Symbole: *Home-Verzeichnis*/`.dt/icons`
- systemweite Symbole: `/etc/dt/appconfig/icons/sprache`

- durch einen absoluten Pfad zu der Symboldatei, einschließlich des vollständigen Dateinamens

Der absolute Pfad sollte nur verwendet werden, wenn sich die Symboldatei nicht im Symbolsuchpfad befindet.

Die Größe des Symbols richtet sich nach der Position des Bedienelements:

Position	Größe
Hauptanzeige	48 x 48 Pixel (<i>name.l.pm</i> or <i>name.l.bm</i>)
Bedientafel	24 x 24 Pixel (<i>name.s.pm</i> or <i>name.s.bm</i>)

Die Symboldatei in eine der folgenden Positionen stellen:

- persönliche Symbole: *Home-Verzeichnis/.dt/icons*
- systemweite Symbole: */etc/dt/appconfig/icons/sprache*

Bedientafeln erstellen und ändern

Der Benutzer kann Bedientafeln über die Menüs des Bedienfelds erstellen und ändern.

Im folgenden Abschnitt wird erläutert, wie durch Änderung der Bedienfeldkonfigurationsdateien eine systemweite Anpassung erreicht wird.

Eine Bedientafel ist einem Bedienelement in der Hauptanzeige zugeordnet.

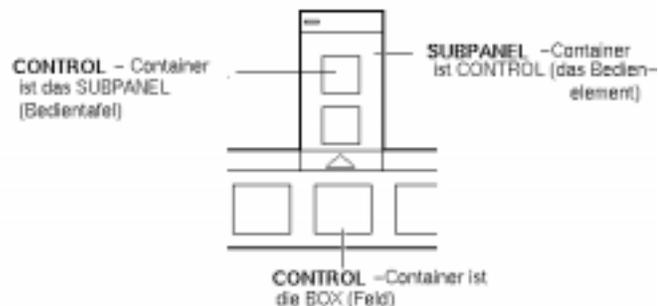


Abbildung 13-3 Der Container einer Bedientafel ist das Bedienelement, dem sie zugeordnet ist.

Die Zuordnung erfolgt über die Bedientafeldefinition. Im Feld CONTAINER_NAME wird angegeben, welchem Bedienelement die Bedientafel zugeordnet ist.

```
CONTROL bedienelement_name
{
    ...
}

SUBPANEL bedientafel_name
{
    CONTAINER_NAME      bedienelement_name
    ...
}
```

▼ Neue systemweite Bedientafeln erstellen

1. Den *bedienelement_name* des Bedienelements in der Hauptanzeige feststellen, dem die Bedientafel zugeordnet werden soll.

Handelt es sich bei dem Bedienelement um eines der integrierten Bedienelemente, befindet sich die Definition in `/usr/dt/appconfig/types/sprache/dtwm.fp`.

2. Eine neue Datei `/etc/dt/appconfig/types/sprache/*.fp` erstellen.
3. Die Bedientafel definieren:

```
SUBPANEL bedientafel_name
{
    CONTAINER_NAME bedienelement_name
    TITLE           wert
    KEYWORD         wert
    ...
}
```

4. Die neue Konfigurationsdatei sichern.
5. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Integrierte Bedientafeln anpassen

Es können allgemeine Merkmale (wie zum Beispiel der Titel) sowie der Inhalt der integrierten Bedientafeln geändert werden.

▼ *Allgemeine Merkmale einer integrierten Bedientafel ändern*

1. Eine neue Bedienfeldkonfigurationsdatei erstellen:
 - systemweit: `/etc/dt/appconfig/types/sprache/name.fp`
 - persönlich: `Home-Verzeichnis/.dt/types/name.fp`.
2. Die gesamte Standardbedientafeldefinition (SUBPANEL) aus der Datei `/usr/dt/appconfig/types/sprache/dtwm.fp` in die neue Datei kopieren:

```
SUBPANEL bedientafel_name
{
    ...
}
```
3. Die Bedientafeldefinition ändern.
4. Die neue Konfigurationsdatei sichern.
5. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Wird, beispielsweise, die folgende Definition in die Datei `/users/janice/.dt/types/PerApps.fp` gestellt, wird dadurch der Name der Bedientafel Persönliche Anwendungen geändert :

```
SUBPANEL PersAppsSubpanel
{
    CONTAINER_NAME      TextEditor
    TITLE                Anwendungen von Janice
}
```

▼ *Systemweite Bedienelemente einer integrierten Bedientafel hinzufügen*

1. Eine Bedienfeldkonfigurationsdatei `/etc/dt/appconfig/types/sprache/name.fp` erstellen.
2. Das systemweite Bedienelement in der Datei definieren.

Mit den Feldern `CONTAINER_NAME` und `CONTAINER_TYPE` den Container für das Bedienelement angeben:

```
CONTROL bedienelement_name
{
    CONTAINER_NAME      bedientafel_name
```

```
CONTAINER_TYPE    SUBPANEL
...
}
```

Weitere Informationen hierzu können dem Abschnitt “Bedienelemente im Bedienfeld definieren” auf Seite 251 entnommen werden.

3. Die Konfigurationsdatei sichern.
4. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Beispielsweise fügt das folgende Bedienelement, das in einer neuen Datei `/etc/dt/appconfig/types/sprache/DigitalClock.fp` definiert ist, die Anwendung Digitaluhr (in der Gruppe von Anwendungen 'Desktop-Werkzeuge') der Bedientafel 'Persönliche Anwendungen' für alle Benutzer hinzu.

```
CONTROL DigitalClockControl
{
    TYPE            icon
    CONTAINER_NAME  PerAppsSubpanel
    CONTAINER_TYPE  SUBPANEL
    ICON            Dtdgclk
    PUSH_ACTION     DigitalClock
    PUSH_RECALL     True
}
```

▼ ***Bedienelemente aus einer integrierten Bedientafel entfernen***

- ◆ Dieselbe Prozedur wie zum Entfernen eines Bedienelements aus der Hauptanzeige verwenden. Weitere Informationen hierzu können dem Abschnitt “Bedienelemente entfernen” auf Seite 243 entnommen werden.

▼ ***Bedienelement 'Symbol installieren' entfernen***

- ◆ Das folgende Feld der Bedientafeldefinition hinzufügen:

```
CONTROL_INSTALL False
```

▼ **Automatisches Schließen von Bedientafeln ändern**

Standardmäßig werden Bedientafeln geschlossen, wenn der Benutzer ein Bedienelement auswählt, es sei denn, der Benutzer hat die Bedientafel aus ihrer ursprünglichen Position verschoben.

Das Bedienfeld kann so konfiguriert werden, daß Bedientafeln geöffnet bleiben, bis der Benutzer sie explizit schließt.

1. Eine neue Bedienfeldkonfigurationsdatei erstellen.:

- systemweit: `/etc/dt/appconfig/types/sprache/*.fp`
- persönlich: *Home-Verzeichnis*/`.dt/types/*.fp`

2. Die Standarddefinition für `PANEL` aus der Datei

`/usr/dt/appconfig/types/sprache/dtwm.fp` in die neue Datei kopieren:

```
PANEL FrontPanel
{
    ...
}
```

3. Das folgende Feld zu der Definition für `PANEL` hinzufügen:

```
SUBPANEL_UNPOST    False
```

4. Die neue Konfigurationsdatei sichern.

5. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Bedienelemente im Bedienfeld definieren

Der Benutzer kann persönliche Bedienelemente erstellen, in dem er Symbole an die Bedienelemente 'Symbol installieren' übergibt.

Dadurch wird die Anpassung zwar wesentlich vereinfacht, aber ein Teil der Funktionalität der Bedienelemente im Bedienfeld geht verloren. Beispielsweise kann ein Bedienelement, das mit dem Bedienelement 'Symbol installieren' erstellt wurde

- keine Animation zur Verfügung stellen
- kein Client-Fenster anzeigen
- nicht bei Auftreten eines Ereignisses das Erscheinungsbild ändern (beispielsweise beim Empfangen von Post)

Im folgenden Abschnitt wird beschrieben, wie Bedienelemente im Bedienfeld manuell erstellt werden.

Referenzinformationen zur Syntax von Bedienelementen im Bedienfeld können der Hilfeseite `dtfile(4X)` entnommen werden.

Definitionen für ein Bedienelement im Bedienfeld

Eine Definition eines Bedienelements im Bedienfeld hat folgende Struktur:

```
CONTROL bedienelement_name
{
  TYPE                control_type
  CONTAINER_NAME     wert
  CONTAINER_TYPE     wert
  andere felder zur definition von erscheinungsbild und verhalten
}
```

Bedienelementtypen

Das Feld `TYPE` in der Bedienelementdefinition gibt das grundlegende Verhalten des Bedienelements an.

Bedienelementtyp (TYPE) Bedienelementverhalten

<code>icon</code>	(Standard). Das Bedienelement führt eine angegebene Aktion aus, wenn der Benutzer das Bedienelement doppelklickt oder dem Bedienelement eine Datei übergibt.
<code>blank</code>	Platzhalter, mit dem der Abstand zwischen Bedienelementen geregelt werden kann.
<code>busy</code>	Betriebsanzeige. Das Bedienelement blinkt (schaltet zwischen Bildschirmen hin und her), wenn eine Aktion aufgerufen wird.
<code>client</code>	Ein Client-Fenster im Bedienfeld.
<code>clock</code>	Uhr.
<code>date</code>	Zeigt das aktuelle Datum an.
<code>file</code>	Stellt eine Datei dar. Durch Auswahl des Bedienelements wird die Standardaktion für die Datei ausgeführt.

▼ Neue Bedienelemente erstellen

Im folgenden Abschnitt wird die Vorgehensweise bei der Definition eines Bedienelements beschrieben und erläutert, wie verschiedene Bedienelementtypen erstellt werden.

1. Soll das Bedienelement über die Aktionen `PUSH_ACTION` und/oder `DROP_ACTION` verfügen, die entsprechenden Aktionsdefinitionen erstellen. Hierbei handelt es sich um die Aktionen, die ausgeführt werden, wenn der Benutzer das Bedienelement anklickt oder ihm eine Datei übergibt.
2. Die Symbolbilddateien für das Bedienelement erstellen.
Informationen zu Symbolgrößen, -namen und -positionen können dem Abschnitt "Symbolbilddateien" auf Seite 224 entnommen werden.
3. Eine neue Bedienfeldkonfigurationsdatei erstellen in:
 - systemweit: `/etc/dt/appconfig/types/sprache/*.fp`
 - persönlich: `Home-Verzeichnis/.dt/types/*.fp`
4. Die Bedienelementdefinition der Datei hinzufügen.
5. Die Datei sichern.
6. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Bedienelement erstellen, das bei Anklicken eine Aktion ausführt

Folgende Felder verwenden, um das Verhalten des Bedienelements zu definieren:

- `TYPE`: Auf `icon` setzen
- `PUSH_ACTION`: Gibt den Namen der Aktion an, die ausgeführt werden soll

Beispielsweise führt das folgende Bedienelement, das in die Bedientafel 'Persönliche Anwendungen' gestellt wird, ein Spiel aus, das der Benutzer am System installiert hat.

```
CONTROL Ball
{
    TYPE                icon
    CONTAINER_NAME      PersAppsSubpanel
    CONTAINER_TYPE      SUBPANEL
    ICON                ball
    PUSH_ACTION         RunBallGame
    HELP_STRING         "Dieses Bedienelement auswählen, um Ball zu
spielen."
}
```

Das folgende Bedienelement wird in die obere linke Ecke des Arbeitsbereichsumschalters gestellt. Es startet eine Aktion mit dem Namen 'CutDisp'.

```
CONTROL StartCutDisp
{
    TYPE            icon
    CONTAINER_NAME  Switch
    CONTAINER_TYPE  SWITCH
    POSITION_HINTS   first
    ICON            cutdisp
    HELP_STRING     "Dieses Bedienelement auswählen, um cutdisp
auszuführen."
    PUSH_ACTION     CutDisp
}
```

Bedienelement erstellen, das eine Datei öffnet

Folgende Felder verwenden, um das Verhalten des Bedienelements zu definieren:

- TYPE: Auf file setzen
- FILE_NAME: Gibt den Pfad der Datei an, die geöffnet werden soll.
- PUSH_ACTION: Auf Open setzen

Eine Aktion 'Öffnen' muß für den Datentyp der Datei definiert sein.

Beispielsweise wird das folgende Bedienelement ganz rechts in die Hauptanzeige gestellt. Es startet den Texteditor mit der Datendatei /users/ellen/PhoneList.txt. Die Aktion 'Öffnen' für Dateien des Typs *.txt ist Bestandteil der Standarddatenbank für Aktionen.

```
CONTROL EditPhoneList
{
    TYPE            file
    FILE_NAME       /users/ellen/PhoneList.txt
    CONTAINER_NAME  Top
    CONTAINER_TYPE  BOX
    POSITION_HINTS   last
    ICON            PhoneBook
    HELP_STRING     "Dieses Bedienelement zeigt Ellens Telefonliste"
    PUSH_ACTION     Open
}
```

Bedienelement erstellen, das als Übergabebereich fungiert

Im Feld `DROP_ACTION` die Aktion angeben, die ausgeführt wird, wenn der Benutzer eine Datei an das Bedienelement übergibt. Die Aktion muß ein Dateiarargument akzeptieren können.

Häufig enthält eine Bedienelementdefinition sowohl ein Feld `PUSH_ACTION` als auch ein Feld `DROP_ACTION`. Dieselbe Aktion kann sowohl für Übergabe als auch für Anklicken verwendet werden.

Beispielsweise führt das folgende Bedienelement in der Bedientafel 'Persönliche Anwendungen' den X-Client `xwud` aus, der ein Dateiarargument benötigt.

```
CONTROL Run_xwud
{
    CONTAINER_NAME      PerAppsSubpanel
    CONTAINER_TYPE      SUBPANEL
    POSITION_HINTS       2
    ICON                 XwudImage
    PUSH_ACTION         RunXwud
    DROP_ACTION         RunXwud
}
```

Bedienelement erstellen, das eine Datei überwacht

Folgende Felder verwenden, um das Verhalten des Bedienelements zu definieren:

- `TYPE`: Einen der folgenden Werte eingeben:

<code>icon</code>	Diesen Typ verwenden, wenn <code>PUSH_ACTION</code> und/oder <code>DROP_ACTION</code> für das Bedienelement angegeben werden soll.
<code>file</code>	Diesen Typ verwenden, wenn das Bedienelement sich nach der Auswahl wie die Datei verhalten soll, wenn das Symbol der Datei im Dateimanager doppelgeklickt wird.

- `ICON` und `ALTERNATE_ICON`: Beschreiben die Bilder, mit denen der ursprüngliche und der geänderte Status der überwachten Datei angegeben wird.
- `MONITOR_TYPE`: Beschreibt die Bedingungen, unter denen ein Bild sich ändert. Einen der folgenden Werte verwenden:

mail	Das Bedienelement ändert sein Erscheinungsbild, wenn der Datei Informationen hinzugefügt werden.
file	Das Bedienelement ändert sich, wenn der angegebenen Datei Informationen hinzugefügt werden.

- **FILE_NAME:** Gibt die zu überwachende Datei an.

Beispielsweise sucht das folgende Bedienelement eine Datei *meetings*, die mit einem anonymen ftp auf das System übertragen werden soll. Das Bedienelement wird an erster Position in die Bedientafel 'Persönliche Anwendungen' gestellt.

```
CONTROL MonitorCalendar
{
    TYPE          file
    CONTAINER_NAME PersAppsSubpanel
    CONTAINER_TYPE SUBPANEL
    POSITION_HINTS first
    FILE_NAME      /users/ftp/meetings
    MONITOR_TYPE   file
    ICON           meetingsno
    ALTERNATE_ICON meetingsyes
}
```

Bedienelement zum Umschalten erstellen

Ein Bedienelement zum Umschalten prüft, ob der Prozeß, der von **PUSH_ACTION** gestartet wurde, bereits ausgeführt wird. Wenn der Prozeß nicht ausgeführt wird, wird **PUSH_ACTION** ausgeführt. Wenn der Prozeß bereits ausgeführt wird, erhält das Fenster die oberste Ebene im Fensterstapel des aktuellen Arbeitsbereichs.

Folgende Felder verwenden, um das Verhalten des Bedienelements zu definieren:

- **PUSH_RECALL:** Auf `True` setzen.
- **CLIENT_NAME:** Gibt den Namen des Client an das Bedienelement.

Der Wert von **CLIENT_NAME** muß mit der ersten Zeichenfolge (*res_name*) der Eigenschaft **WM_CLASS** übereinstimmen. Informationen dazu können der Hilfeseite `xprop(1)` entnommen werden.

- **PUSH_ACTION:** Beschreibt die Aktion 'Ausführen', wenn der Benutzer auf das Bedienelement klickt.

Beispielsweise führt das folgende Bedienelement eine Anwendung mit der Aktion MyEditor einmalig aus.

```
CONTROL MyEditor
{
    TYPE            icon
    CONTAINER_NAME Top
    CONTAINER_TYPE BOX
    POSITION_HINTS  15
    PUSH_RECALL    True
    CLIENT_NAME     BestEditor
    PUSH_ACTION    StartMyEditor
    ICON           MyEd
}
```

▼ Clientfensterbedienelement erstellen

Ein Clientfensterbedienelement ist ein Anwendungsfenster im Bedienfeld. Beispielsweise kann durch das Erstellen eines `xload` Clientfensterbedienelements eine Systemauslastungsanzeige in das Bedienfeld gestellt werden.

1. Das Bedienelement definieren.

Folgende Felder verwenden, um das Verhalten des Bedienelements zu definieren:

- `TYPE`: Auf `client` setzen.
- `CLIENT_NAME`: Gibt den Client an, der gestartet werden soll.

Der Wert von `CLIENT_NAME` muß mit der ersten Zeichenfolge (*res_name*) der Eigenschaft `WM_CLASS` übereinstimmen. Informationen dazu können der Hilfeseite `xprop(1)` entnommen werden.

- `CLIENT_GEOMETRY`: Gibt in Pixel die Größe des Bedienfeldfensters für den Benutzer an.

Die Hilfeseite `xwininfo(1)` beschreibt, wie man die Größe eines Fensters in Pixel erhält.

2. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

3. Den Client aus einer Terminal-Emulatorbefehlszeile starten.

Beispielsweise zeigt das folgende Bedienelement eine Auslastung von 30 x 20 Pixel.

```
CONTROL LoadMeter
{
    TYPE          client
    CONTAINER_NAME Top
    CONTAINER_TYPE BOX
    CLIENT_NAME    xload
    CLIENT_GEOMETRY 30x20
}
```

Wenn der Client zwischen Sessions nicht gespeichert und wiederhergestellt wird, kann das Bedienelement so konfiguriert werden, daß es den Client startet, wenn es vom Benutzer angeklickt wird. Beispielsweise kann das Bedienelement zur Systemauslastung so konfiguriert werden, daß xload gestartet wird durch Hinzufügen der folgenden Zeile in der Definition

```
PUSH_ACTION    StartXload
```

und dem Erstellen dieser Aktion.

```
ACTION StartXload
{
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING    /usr/contrib/bin/X11/xload
}
```

▼ *Bedienelement animieren*

Eine Animationsfolge kann hinzugefügt werden, die ausgeführt wird, wenn der Benutzer das Bedienelement anwählt oder ihm ein Objekt übergibt.

Für eine Animationsfolge muß ein Bedienelement folgende Eigenschaften besitzen:

- Vom Typ `icon` sein
- `PUSH_ACTION` oder über `DROP_ACTION` verfügen

1. Die Animationsfolge mit der Komponente `ANIMATION` angeben:

```
ANIMATION animation_name
{
    symbol_bild          [ verzögerung ]
    symbol_bild          [ verzögerung ]
    ...
}
```

wobei *verzögerung* die Zeitverzögerung in Millisekunden zwischen der Animation von Symbolen angibt.

2. Die Felder `PUSH_ANIMATION` und/oder `DROP_ANIMATION` der Bedienelementdefinition hinzufügen. Der Wert ist der Name der Folge `ANIMATION`.

Beispielsweise animieren die folgenden Zeilen ein Bedienelement, das die Anwendung `BestEditor` startet. Die Zeitverzögerung zwischen Symbolen beträgt 300 Millisekunden. In diesem Beispiel wird angenommen, daß die Symboldateien `frame1`, `frame2`, usw. erstellt wurden.

```
CONTROL BestEditor
{
    ...
    PUSH_ANIMATION BestEdAnimation
    ...
}

ANIMATION BestEdAnimation
{
    frame1      300
    frame2
    ...
}
```

Kontexthilfe für Bedienelemente des Bedienfelds

Es gibt zwei Möglichkeiten der Kontexthilfe für ein Bedienelement:

- Eine Hilfezeichenfolge in der Bedienelementdefinition.

Die Hilfezeichenfolge wird in der Hilfeanzeige angezeigt, wenn der Benutzer Kontexthilfe für das Bedienelement aufruft. Die Hilfezeichenfolge kann keine Formatierungen (z.B. Kopfzeilen) oder Verweise enthalten.

Die Hilfezeichenfolge in der Bedienelementdefinition angeben:

```
HELP_STRING    hilfe_zeichenfolge
```

- Ein Hilfethema in einem registrierten Hilfeband:

Ein Hilfethema sind zusammengefaßte Informationen mit den vollen Möglichkeiten des Hilfesystems. Um ein Hilfethema zu verfassen, ist der Help Developers Kit des Desktops erforderlich.

Den Hilfeband und Kennnummer des Themas in der Bedienelementdefinition angeben.

```
HELP_VOLUME  hilfeband_name
HELP_TOPIC   thema_kennnummer
```

Arbeitsbereichsumschalter anpassen

Es gibt verschiedene Möglichkeiten, den Arbeitsbereichsumschalter anzupassen:

- Anzahl der Arbeitsbereiche ändern
- Anordnung des Schalters ändern
- Bedienelemente im Schalter ändern

▼ Standardanzahl der Arbeitsbereiche ändern

♦ Die folgende Arbeitsbereichsmanagerressource ändern:

```
Dtwm*workspaceCount:n
```

Mehr Informationen können dem Abschnitt “Die Anzahl von Arbeitsplätzen auf einer systemweiten Basis” auf Seite 244 entnommen werden.

▼ Anzahl der Schalterzeilen ändern

♦ Das Feld NUMBER_OF_ROWS in der SWITCH Definition ändern.

Beispielsweise definiert die folgende Definition einen Schalter mit drei Zeilen.

```
SWITCH Switch
{
    CONTAINER_NAME feld_name
    NUMBER_OF_ROWS 3
    ...
}
```

▼ Bedienelemente im Arbeitsbereichsumschalter ändern oder hinzufügen

1. Eine Bedienfeldkonfigurationsdatei mit der Bedienelementdefinition erstellen.

- Das Bedienelement angeben, das im Schalter angezeigt werden soll:

```
CONTAINER_NAME  Switch
CONTAINER_TYPE   SWITCH
```

- Die Position im Schalter angeben:

```
POSITION_HINTS    n
```

wobei n eine ganze Zahl ist. Die Positionen werden durchnummeriert von links nach rechts und von oben nach unten. (Beim standardmäßig eingestellten Schalter mit zwei Zeilen sind dies die Positionen 1 bis 4.)

2. Das Symbol für das Bedienelement erstellen. Die empfohlene Größe beträgt 16 auf 16 Pixel.

Beispielsweise stellt das folgende Bedienelement ein Bedienelement für Terminalfenster in den Schalter.

```
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
}
```

Das Bedienelement verwendet ein integriertes Symbol und dasselbe Hilfethema, das von dem Bedienelement für Terminalfenster und der Bedientafel Persönliche Anwendungen benutzt wird.

Allgemeine Bedienfeldkonfiguration

Die `PANEL` Syntax des Bedienfelds ermöglicht:

- Position des Bedienfelds ändern
- Aussehen des Fensters ändern
- Allgemeines Erscheinungsbild und Verhalten von Bedienelementen einstellen.

Die Standardbeschreibung von `PANEL` befindet sich in der Datei `/usr/dt/appconfig/types/sprache/dtwm.fp`.

Zusätzliche Informationen können der Hilfeseite `dtfpfile(4X)` entnommen werden.

Allgemeine Schritte

1. Eine neue Bedienfeldkonfigurationsdatei erstellen in:
`/etc/dt/appconfig/types/sprache` oder *Home-Verzeichnis*/`.dt/types`.
2. Die Standardbeschreibung von PANEL aus der Datei
`/usr/dt/types/sprache/dtwm.fp` in die neue Datei kopieren.
3. Die PANEL Beschreibung bearbeiten.

Die neue PANEL Beschreibung hat Vorrang vor der Standardbeschreibung.

▼ Standardposition des Bedienfelds ändern

- ◆ Mit dem Feld `PANEL_GEOMETRY` in der Definition für PANEL die Position angeben.

Das folgende Bedienfeld ist zum Beispiel in der oberen rechten Ecke.

```
PANEL SpecialFrontPanel
{
  PANEL_GEOMETRY      -1+1
  ...
}
```

▼ Bezeichnungen für Bedienelemente in der Hauptanzeige erstellen

1. Die folgende Zeile der Definition für PANEL hinzufügen:

```
DISPLAY_CONTROL_LABELS    True
```

2. Jedem Bedienelement ein Feld `LABEL` hinzufügen.

Der *bedienelement_name* wird verwendet, wenn kein `LABEL` angegeben wird.

▼ Klickverhalten von Bedienelementen ändern

- ◆ Mit dem Feld `CONTROL_BEHAVIOR` in der Definition für PANEL angeben, wie der Benutzer die `PUSH_ACTION` eines Bedienelements ausführt. Es gibt folgende Werte für das Feld:

```
single_click    Der Benutzer klickt das Bedienelement an, um
                 PUSH_ACTION auszuführen.
```

`double_click` Der Benutzer doppelklickt das Bedienelement, um `PUSH_ACTION` auszuführen.

▼ Völlig neue Bedienfelder erstellen

Das Erstellen eines neuen Bedienfelds kann von Vorteil sein, wenn umfangreiche Änderungen gemacht werden wollen.

Um nicht mit den integrierten Bedienfeldkomponenten in Konflikt zu geraten, sollte ein völlig neues Bedienfeld neue Namen für das PANEL und andere Container verwenden.

1. Die PANEL Komponente mit einem eindeutigen Namen für das neue Bedienfeld erstellen.

```
PANEL bedienfeld_name
{
  ...
}
```

2. Die neuen Felder und Bedienelemente mit den neuen Container-Namen erstellen.

Wenn existierende Komponenten benutzt werden sollen, müssen ihre Definitionen kopiert und der Wert von `CONTAINER_NAME` geändert werden.

3. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Beispiel für die Erstellung eines persönlichen Bedienfelds mit drei Zeilen

Das folgende Beispiel ändert das Standardbedienfeld, so daß die Bedienelemente in drei Zeilen organisiert werden.

1. Die Datei `/usr/dt/appconfig/types/sprache/dtwm.fp` in die Datei *Home-Verzeichnis*/`.dt.types/MyFrontPanel.fp` kopieren. Der Datei Schreibzugriff geben.

Diese Datei wird zum Erstellen des neuen Bedienfelds bearbeitet.

2. Den Namen des Bedienfelds ändern:

```
PANEL NewFrontPanel
```

3. Den Namen des Feldes Top ändern und den Container-Namen bearbeiten:

```
BOX NewFrontPanelTop
{
    CONTAINER_NAME NewFrontPanel
    POSITION_HINTS first
    ...
}
```

4. Feldefinitionen für die mittlere und untere Zeile hinzufügen:

```
BOX NewFrontPanelMiddle
{
    CONTAINER_NAME    NewFrontPanel
    POSITION_HINTS     second
}
```

```
BOX NewFrontPanelBottom
{
    CONTAINER_NAME    NewFrontPanel
    POSITION_HINTS     second
}
```

5. Den CONTAINER_NAME der folgenden Bedienelemente zu NewFrontPanelTop ändern:

- Clock
- Date
- Home
- TextEditor
- Mail

6. Den CONTAINER_NAME der folgenden Bedienelemente zu NewFrontPanelBottom ändern:

- Printer
- Style
- Applications
- Help
- Trash

7. Den CONTAINER_NAME des Arbeitsbereichsschalters zu NewFrontPanelMiddle ändern.

8. Die Ressource einstellen:

```
Dtwm*frontPanel*name: NewFrontPanel
```

9. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Arbeitsbereichsmanager anpassen

In diesem Kapitel wird die Anpassung des Desktop-Arbeitsbereichsmanagers behandelt.

<i>Konfigurationsdateien des Arbeitsbereichsmanagers</i>	<i>266</i>
<i>Arbeitsbereiche anpassen</i>	<i>269</i>
<i>Arbeitsbereichsmanagermenüs</i>	<i>271</i>
<i>Maustastenbelegungen anpassen</i>	<i>274</i>
<i>Tastaturbelegungen anpassen</i>	<i>277</i>
<i>Wechseln zwischen Standardverhalten und angepasstem Verhalten</i>	<i>279</i>

Der Arbeitsbereichsmanager ist die Fenstersteuerung des Desktops. Wie andere Fenstersteuerungen steuert er:

- das Erscheinungsbild von Komponenten des Fensterrahmens
- das Verhalten von Fenstern, eingeschlossen ihre Stapelordnung und ihr Fokusverhalten
- Tasten- und Maustastenbelegung
- das Erscheinungsbild von Fenstern in Symbolgröße
- Arbeitsbereich und Fenstermenüs

Zusätzlich steuert der Arbeitsbereichsmanager die folgenden Desktop-Komponenten:

- *Arbeitsbereiche.* Der Arbeitsbereichsmanager steuert die Anzahl der Arbeitsbereiche und verwaltet die Anzahl der geöffneten Fenster in jedem Arbeitsbereich.
- *Hintergrundmuster des Arbeitsbereichs.* Der Benutzer ändert Hintergrundmuster mit Hilfe des Umgebungsmanager. Die Verwaltung des Hintergrundmusters ist jedoch eine Funktion des Arbeitsbereichsmanagers.
- *Bedienfeld.* Obwohl das Bedienfeld seine eigenen Konfigurationsdateien hat, wird es vom Arbeitsbereichsmanager erstellt und verwaltet.

Viel von obengenanntem kann mit Hilfe des Umgebungsmanager geändert werden. Mit dem Umgebungsmanager können oft durchgeführte Änderungen schnell und auf einfache Weise gemacht werden. Andere Ressourcen müssen manuell eingestellt werden

Der Arbeitsbereichsmanager ist `dtwm`. Er basiert auf der Grundlage der OSF/Motif-Fenstersteuerung.

Weiterführende Informationen

- Referenzinformationen zum Arbeitsbereichsmanager können den Hilfeseiten `dtwm(1)` und `dtwmrc(4)` entnommen werden.
- Weitere Informationen zum Setzen von Arbeitsbereichsmanagerressourcen können dem Abschnitt “Anwendungsressourcen einstellen” auf Seite 282 entnommen werden.
- Weitere Informationen zu Bedienfeldkonfigurationsdateien können Kapitel 13, “Erweiterte Anpassung des Bedienfelds” entnommen werden.

Zusätzliche Informationen zum Einstellen von Ressourcen können dem Abschnitt “Anwendungsressourcen einstellen” auf Seite 282 entnommen werden.

Konfigurationsdateien des Arbeitsbereichsmanagers

Der Arbeitsbereichsmanager erhält von einer Konfigurationsdatei Informationen über die Fenstermenüs, Arbeitsbereichmenüs, Maustastenbelegung und Tastenbelegung.

Er verwendet eine der folgenden Dateien:

- Persönliche Datei: *Home-Verzeichnis*/`.dt/dtwmrc`
- Systemangepasste Datei: `/etc/dt/config/sprache/sys.dtwmrc`

- **Integrierte Datei:** `/usr/dt/config/sprache/sys.dtwmrc`

Der Arbeitsbereichsmanager sucht in dieser Reihenfolge nach einer Konfigurationsdatei und benutzt die zuerst gefundene Datei.

Für Benutzer, die mehr als eine Session-Sprache verwenden, kann eine sprachenabhängige Konfigurationsdatei *Home-Verzeichnis/.dt/*sprache*/dtwrmc* erstellt werden, die Vorrang vor *Home-Verzeichnis/.dt/dtwmrc* hat.

▼ Persönliche Konfigurationsdatei erstellen oder ändern

Die persönliche Konfigurationsdatei des Arbeitsbereichsmanager heißt *Home-Verzeichnis/.dt/dtwmrc*. Wenn diese Datei existiert, wird sie verwendet.

1. Auf 'Dtwrmc editieren' in der Anwendungsgruppe 'Desktop-Werkzeuge' doppelklicken.

Wenn eine persönliche Datei `dtwrmc` vorhanden ist, wird sie in den Editor geladen. Wenn diese Datei nicht vorhanden ist, wird `sys.dtwmrc` in *Home-Verzeichnis/.dt/dtwmrc* kopiert, und diese wird dann in den Editor geladen.

2. Datei bearbeiten.
3. Den Editor beenden.

Diese Datei wird unbeachtet ihres Ursprungs als persönliche Datei `dtwrmc` abgespeichert.

▼ Systemweite Konfigurationsdatei erstellen

Die systemweite Konfigurationsdatei des Arbeitsbereichsmanager heißt `/etc/dt/config/sprache/sys.dtwmrc`.

- ◆ Die Datei `/usr/dt/config/sprache/sys.dtwmrc` in die Datei `/etc/dt/config/sprache/sys.dtwmrc` kopieren.

Hinweis – Diese Datei kann nicht verwendet werden, wenn *Home-Verzeichnis/.dt/dtwmrc* existiert.

▼ Andere Dateien einfügen

◆ Folgende Syntax verwenden:

```
include
{
    pfad
    pfad
    ...
}
```

Beispielsweise werden folgende Zeilen in die Datei `/users/ellen/mymenu:` eingefügt.

```
include
{
    /users/ellen/mymenu
}
```

Include-Anweisungen sind nützlich für eine zusätzliche Funktionalität, ohne die gesamte Konfigurationsdatei zu kopieren. So möchte zum Beispiel ein Benutzer eine neue Tastenbelegung erstellen, ohne die gesamte Konfigurationsdatei zu verwalten. Der Benutzer kann eine Datei *Home-Verzeichnis*/`.dt/dtwmrc` mit folgendem Inhalt erstellen.

```
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"
    ...
}
```

▼ Arbeitsbereichsmanager neu starten

Der Arbeitsbereichsmanager muß neu gestartet werden, um so die Änderungen in der Konfigurationsdatei wirksam werden zu lassen.

‘Arbeitsbereichsmanager neu starten’ im Arbeitsbereichsmenü auswählen (die dritte Maustaste drücken, wenn der Zeiger im Hintergrund positioniert ist).

Arbeitsbereiche anpassen

Die meisten Anpassungen von Arbeitsbereichen, wie zum Beispiel das Ändern der Arbeitsbereichsnamen und die Anzahl der Arbeitsbereiche, können vom Benutzer mit Hilfe der Desktopschnittstelle durchgeführt werden. Der Arbeitsbereichsmanager besitzt jedoch Ressourcen zur Einstellung von systemweiten Standards.

▼ Anzahl der Arbeitsbereiche auf einer systemweiten Basis ändern

Die Standarddesktopkonfiguration verfügt über vier Arbeitsbereiche. Der Benutzer kann mit dem Menü, das dem Arbeitsbereichsumschalter zugeordnet ist, Arbeitsbereiche hinzufügen und löschen.

Der Arbeitsbereichsmanager besitzt eine Ressource zum Ändern der Standardanzahl der Arbeitsbereiche.

- ♦ Die Ressource `workspaceCount` verwenden, um die Anzahl der Arbeitsbereiche einzustellen:

```
Dtvm*workspaceCount: anzahl
```

Weitere Informationen zum Einstellen der Ressourcen des Arbeitsmanagers können dem Abschnitt “Anwendungsressourcen einstellen” auf Seite 282 entnommen werden.

Die folgende Ressource stellt zum Beispiel die Anzahl der Arbeitsbereiche auf sechs ein:

```
Dtvm*workspaceCount: 6
```

▼ Systemweite Arbeitsbereichsnamen eingeben

Die Arbeitsbereiche werden intern nach der Konvention `wsn` durchnummeriert, wobei `n` für 0, 1, 2, usw. steht. Die vier Standardarbeitsbereiche sind zum Beispiel intern `ws0` bis `ws3` durchnummeriert.

- ♦ Die Ressource `title` verwenden, um den Namen eines angegebenen Arbeitsbereichs zu ändern:

Dtwn*ws*n*: *name*

Weitere Informationen zum Einstellen der Ressourcen des Arbeitsbereichsmanagers können dem Abschnitt “Anwendungsressourcen einstellen” auf Seite 282 entnommen werden.

Beispielsweise setzen die folgenden Ressourcen die vier Standardarbeitsbereiche auf angegebene Namen:

```
Dtwn*ws0*title: Anna
Dtwn*ws1*title: Don
Dtwn*ws2*title: Julia
Dtwn*ws3*title: Patti
```

▼ Zusätzliche Hintergrundmuster erstellen

1. Hintergrundbilder erstellen. Die Bilder können Bitmap- oder Pixmap-Dateien sein.
2. Die Hintergrundmuster in eines der folgenden Verzeichnisse stellen. (Unter Umständen muß das Verzeichnis erstellt werden.)
 - Systemweite Hintergrundmuster: `/etc/dt/backdrops`
 - Persönliche Hintergrundmuster: *Home-Verzeichnis*/`.dt/backdrops`
3. ‘Arbeitsbereichsmanager neu starten’ im Arbeitsbereichsmenü auswählen.

Die systemweiten und die persönlichen Hintergrundmuster werden integrierten Hintergrundmustern in `/usr/dt/backdrops` hinzugefügt.

Ein existierendes integriertes Hintergrundmuster kann ersetzt werden, indem ein persönliches oder ein systemweites Hintergrundmuster mit demselben Namen erstellt wird.

▼ Hintergrundmuster durch eine Grafik ersetzen

Die Hintergrundmuster werden über das Basisfenster der Anzeige gelegt. Das Dialogfenster ‘Hintergrundmuster’ des Umgebungsmanagers enthält eine Einstellung ‘Kein Hintergrundmuster’, bei der das Hintergrundmuster transparent ist.

Es gibt nur ein Basisfenster hinter allen Hintergrundmustern des Arbeitsbereichs. Deshalb bleibt eine Grafik, die über das Basisfenster gestellt wurde, in allen Arbeitsbereichen bestehen. Es kann angegeben werden, bei

welchen Arbeitsbereichen das Basisfenster von einem Hintergrundmuster überlagert wird. Bei der Aktivierung von 'Kein Hintergrundmuster' ist jedoch das angezeigte Bild bei allen Arbeitsbereichen gleich.

1. Die Grafik erstellen. Die Grafik muß in einem Format sein, das auf dem Basisfenster angezeigt werden kann. Wenn zum Beispiel `xsetroot` benutzt werden soll, muß eine Bitmapdatei erstellt werden.
2. Wenn diese Datei noch nicht existiert, muß eine ausführbare Datei `Home-Verzeichnis/.dt/sessions/sessionetc` erstellt werden. Die Datei `sessionetc` wird jedesmal ausgeführt, wenn sich der Benutzer anmeldet.
3. Den Befehl, das Bild anzuzeigen, in die Datei `sessionetc` stellen.

Beispielsweise ordnet der folgende Befehl das Basisfenster und das angegebene Bitmap nebeneinander an:

```
xsetroot -bitmap /users/ellen/.dt/icons/root.bm
```

Arbeitsbereichsmanagermenüs

Der Arbeitsbereichsmanager verfügt über drei Standardmenüs:

Arbeitsbereichsmenü	Wird auch Basismenü genannt. Es wird angezeigt, wenn der Benutzer Maustaste 3 drückt und der Zeiger im Hintergrund positioniert ist. Das Menü ist über Maustastenbelegung der Maustaste zugeordnet.
Fenstermenü	Das Menü wird angezeigt, wenn der Benutzer Maustaste 1 oder 3 drückt und der Zeiger auf der Fenstermenütaste ist (obere linke Ecke des Fensterrahmens). Das Menü ist über die Ressource <code>windowMenu</code> der Taste zugeordnet.
Bedienfeldmenü	Das Menü wird angezeigt, wenn der Benutzer Maustaste 1 oder 3 drückt und der Zeiger über der Fenstermenütaste des Bedienfelds ist.

Syntax des Arbeitsbereichsmanagermenüs

Arbeitsbereichsmanagermenüs haben folgende Syntax:

```
Menu menüname
{
    auswahl1 [mnemonisch] [direktaufruf] funktion [argument]
```

```

    auswahl2 [mnemonisch] [direktaufruf] funktion [argument]
    ...
}

```

Dabei bedeuten:

- auswahl* Der Text oder das Bitmap, die im Menü erscheinen. Wenn der Text Leerstellen enthält, den Text in Anführungszeichen stellen. Bei Bitmaps die Syntax *@/pfad* verwenden.
- mnemonisch* Ein einzelnes Zeichen, das als Kurzwahltaste fungiert, wenn das Menü angezeigt ist. Es wird in der Form *_character* angegeben.
- direktaufruf* Eine Kurzwahltaste, die unabhängig davon, ob das Menü angezeigt ist oder nicht, aktiv ist. Diese Direktaufrufe haben die Syntax *Kombinationstaste<Key> Tastenname*, wobei *Strg*, *Umschalt*, *Alt* (erweitertes Zeichen), oder die *Feststelltaste* die Kombinationstasten sind. Eine Liste aller möglichen Tastennamen kann der Datei *keysymdef.h* im Verzeichnis "X11 include" entnommen werden.
- funktion* Wenn dies ausgewählt wird, wird eine Funktion ausgeführt. Eine Liste der Funktionen kann der Hilfeseite *dtwmrc(4)* entnommen werden.
- argument* Funktionsargumente. Weitere Einzelheiten können der Hilfeseite *dtwmrc(4)* entnommen werden.

Beispielsweise bringt das folgende Menüthema mit der Bezeichnung 'Restore' (Wiederherstellen) das Fenster zur Normalgröße zurück. Wenn das Menü angezeigt ist, stellt auch die Eingabe von "R" das Fenster wieder her. Das Drücken des erweiterten Zeichens F5 stellt ebenfalls das Fenster wieder her.

```
Restore  _R  Alt<Key> F5  f.normalize
```

Hinweis – Vollständige Informationen über die Syntax des Arbeitsbereichsmanagermenüs können der Hilfeseite *dtwmrc(4)* entnommen werden.

▼ Existierendes Arbeitsbereichsmenü (Basismenü) ändern

1. Die entsprechende Datei zur Bearbeitung öffnen:
 - persönlich: *Home-Verzeichnis/.dt/dtwmrc*
 - systemweit: */etc/dt/config/sprache/sys.dtwmrc*

Weitere Informationen zur Erstellung dieser Dateien können dem Abschnitt “Konfigurationsdateien des Arbeitsbereichsmanagers” auf Seite 266 entnommen werden.

2. Die Beschreibung des Arbeitsbereichsmenüs bearbeiten.

Das Standardarbeitsbereichsmenü heißt `DtRootMenu`.

```
Menu DtRootMenu
{
    "Arbeitsbereichsmenü"           f.title
    "Nächstes Fenster"             f.circle_up
    "Vorheriges Fenster"          f.circle_down
    ...
}
```

▼ Neues Arbeitsbereichsmenü (Basismenü) erstellen

1. Die entsprechende Datei zur Bearbeitung öffnen:

- persönlich: `Home-Verzeichnis/.dt/dtwmrc`
- systemweit: `/etc/dt/config/sprache/sys.dtwmrc`

Weitere Informationen zur Erstellung dieser Dateien können dem Abschnitt “Konfigurationsdateien des Arbeitsbereichsmanagers” auf Seite 266 entnommen werden.

2. Das neue Menü erstellen:

```
Menu menü_name
{
    ...
}
```

Weitere Informationen können dem Abschnitt “Syntax des Arbeitsbereichsmanagermenüs” auf Seite 271 entnommen werden.

3. Die Tastenbelegung zum Anzeigen des neuen Menüs erstellen oder bearbeiten.

Wenn das Menü das existierende Menü ersetzt, die Maustastenbelegung bearbeiten, die das Arbeitsbereichsmenü anzeigt.

```
<Btn3Down> root f.menu menü_name
```

Wenn das Menü hinzugefügt wird, eine neue Maustastenbelegung erstellen. Beispielsweise zeigt die folgende Maustastenbelegung das Menü an, wenn die Tastenkombination Umschalt+Maustaste 3 über dem Hintergrund gedrückt werden.

```
Shift<Btn3Down> root f.menu menü_name
```

4. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

▼ Neues Fenstermenü erstellen

Hinweis – Das Fenstermenü ist in den Arbeitsbereichsmanager integriert und gewöhnlich nicht angepasst. Um das Fensterverhalten zwischen Anwendungen beständig zu halten, sollten größere Änderungen im Fenstermenü vermieden werden.

1. Die entsprechende Datei zur Bearbeitung öffnen:

- persönlich: *Home-Verzeichnis*/.dt/dtwmrc
- systemweit: /etc/dt/config/*sprache*/sys.dtwmrc

Weitere Informationen zur Erstellung dieser Dateien können dem Abschnitt "Konfigurationsdateien des Arbeitsbereichsmanagers" auf Seite 266 entnommen werden.

2. Das neue Menü erstellen:

```
Menu menü_name
{
    ...
}
```

3. Mit der Ressource `windowMenu` das neue Menü angeben:

```
Dtwm*windowMenu: menü_name
```

4. 'Arbeitsbereichsmanager neu starten' im Arbeitsbereichsmenü auswählen.

Maustastenbelegungen anpassen

Eine *Maustastenbelegung* ordnet einer Maustaste und möglichen Kombinationstasten der Tastatur eine Fenstersteuerungsfunktion zu.

Die Standardmaustastenbelegungen des Desktop sind in der Arbeitsbereichsmanagerkonfigurationsdatei in einem Maustastenbelegungsset namens `DtButtonBindings` definiert:

```
Buttons DtButtonBindings
{
    ...
}
```

Syntax der Maustastenbelegung

Die Syntax für Maustastenbelegungen lautet:

```
Buttons MaustastenbelegungSetName
{
    [kombinationstaste]<maustaste_mausaktion> kontext funktion [argument]
    [kombinationstaste]<maustaste_mausaktion> kontext funktion [argument]
```

Dabei bedeuten:

<i>maustaste</i>	<p>Btn1—Linke Maustaste Btn2—Mittlere Taste (Maus mit 3 Tasten) oder beide Tasten (Maus mit 2 Tasten) Btn3—Rechte Taste Btn4—Tasten 1 und 2 gleichzeitig bei einer Maus mit 3 Tasten Btn5—Tasten 2 und 3 gleichzeitig bei einer Maus mit 3 Tasten</p>
<i>kombinationstaste</i>	Ctrl, Shift, Alt, Lock
<i>mausaktion</i>	<p>Down—Eine Maustaste gedrückt halten Up—Eine Maustaste loslassen Click—Eine Maustaste drücken und loslassen Click2—Eine Maustaste doppelklicken Drag—Die Maus ziehen, während die Maustaste gedrückt ist</p>
<i>kontext</i>	<p>Zeigt an, an welcher Position sich der Zeiger befinden muß, um die Belegung wirksam zu machen. Wenn nötig, können mehrere Inhalte mit dem Zeichen " " getrennt werden</p> <p>root—Arbeitsbereichsfenster window—Clientfenster oder Fensterrahmen frame—Fensterrahmen, ohne Inhalte icon—Symbol title—Titelleiste app—Client-Fenster (ohne Rahmen)</p>
<i>funktion</i>	Eine der Fenstersteuerungsfunktionen. Eine Liste der gültigen Funktionen kann der Hilfeseite <code>dtwmrc(4)</code> entnommen werden.

argument Alle erforderlichen Fenstersteuerungsfunktionsargumente. Einzelheiten dazu können der Hilfeseite `dtwmrc(4)` entnommen werden.

Beispielsweise wird mit der folgenden Zeile das Menü angezeigt, das in `DtRootMenu` beschrieben ist, wenn Maustaste 3 gedrückt wird und der Zeiger im Arbeitsbereichsfenster ist (jedoch nicht in Client-Fenstern).

```
<Btn3Down>      root      f.menu      DtRootMenu
```

Hinweis – Vollständige Informationen zur Syntax der Maustastenbelegung können der Hilfeseite `dtwmrc(4)` entnommen werden.

▼ Maustastenbelegung hinzufügen

1. Die entsprechende Datei zur Bearbeitung öffnen:

- persönlich: *Home-Verzeichnis*/.dt/dtwmrc
- systemweit: /etc/dt/config/*sprache*/sys.dtwmrc

Weitere Informationen zur Erstellung dieser Dateien können dem Abschnitt “Konfigurationsdateien des Arbeitsbereichsmanagers” auf Seite 266 entnommen werden.

2. Die Maustastenbelegung der Definition zu `DtButtonBindings` hinzufügen.

Die Taste nicht mit verschiedenen Funktionen für das Klicken und Drücken belegen und diesselbe Taste und Kontext nicht mit mehr als einer Funktion belegen.

3. ‘Arbeitsbereichsmanager neu starten’ im Arbeitsbereichsmenü auswählen.

▼ Neue Maustastenbelegungssets erstellen

1. Die entsprechende Datei zur Bearbeitung öffnen:

- persönlich: *Home-Verzeichnis*/.dt/dtwmrc
- systemweit: /etc/dt/config/*sprache*/sys.dtwmrc

Weitere Informationen zur Erstellung dieser Dateien können dem Abschnitt “Konfigurationsdateien des Arbeitsbereichsmanagers” auf Seite 266 entnommen werden.

2. Den neuen Maustastenbelegungsset erstellen. Weitere Informationen können dem Abschnitt “Syntax der Maustastenbelegung” auf Seite 275 entnommen werden.
3. Die Ressource `buttonBindings` auf den neuen Namen setzen:
`Dtwm*buttonBindings: MaustastenbelegungSetName`
4. ‘Arbeitsbereichsmanager neu starten’ im Arbeitsbereichsmenü auswählen.

Hinweis – Die neuen Maustastenbelegungen ersetzen die existierenden Maustastenbelegungen. Alle Maustastenbelegungen, die behalten werden sollen, aus `DtButtonBindings` kopieren.

Tastaturbelegungen anpassen

Eine *Tastaturbelegung*, auch bekannt als *Tastenbelegung*, ordnet Tastenkombinationen Arbeitsbereichsmanagerfunktionen zu. Tastenbelegungen können in allen Arbeitsbereichen angewendet werden.

Hinweis – Es sollte darauf geachtet werden, keine gebräuchliche Tastenkombination neu zu belegen. Beispielsweise setzt die Tastenkombination Umschalt+A den Buchstaben “A” in das aktuelle Fenster. Wenn die Tastenkombination Umschalt+A mit einer Funktion belegt wird, geht der normale Gebrauch dieser Taste verloren.

Standardtastenbelegungen des Desktop

Die Standardtastenbelegungen des Desktop sind in der Arbeitsbereichsmanagerkonfigurationsdatei in einem Tastenbelegungsset namens `DtKeyBindings` definiert:

```
Keys DtKeyBindings
{
  ...
}
```

Syntax der Tastenbelegung

Die Syntax der Tastenbewegung lautet:

```

Keys TastenbelegungSetName
{
  [ direktaufrufe ] <Key> tastenname kontext funktion [ argument ]
  [ direktaufrufe ] <Key> tastenname kontext funktion [ argument ]
  ...
}

```

Dabei bedeuten:

direktaufrufe Ctrl, Shift, Alt, und Lock. Mehrere Kombinationstasten sind erlaubt, sie werden durch Leerstellen getrennt.

key_name Die Taste, der die Funktion zugeordnet ist. Bei Tasten mit Buchstaben oder Zahlen steht der *key_name* (Name der Taste) normalerweise auf der Taste. Zum Beispiel ist "a" der Name der Taste "a", und die Taste "2" wird "2" genannt. Die Tabulatortaste heißt "Tab" und die Funktionstaste "F3" heißt "F3".

Bei anderen Tasten wird der Name ausbuchstabiert - zum Beispiel Plus für die Taste "+". Die Datei `keysymdef.h`, die sich in einem systemabhängigen Verzeichnis befindet, enthält zusätzliche Informationen über Tastennamen.

kontext Dieses Element muß den Tastaturfokus haben, um die Aktion wirksam zu machen. Die Kontexte können miteinander verknüpft sein, wenn sich die Belegung auf mehr als einen Kontext bezieht. Mehrere Kontexte werden durch das Zeichen "|" getrennt.

root-Hintergrundmuster des Arbeitsbereichs
 window-Clientfenster
 icon-Symbol

funktion Eine Fenstersteuerungsfunktion. Eine Liste der gültigen Funktionen kann der Hilfeseite `dtwmrc(4)` entnommen werden.

argument Alle erforderlichen Fenstersteuerungsfunktionsargumente. Einzelheiten dazu können der Hilfeseite `dtwmrc(4)` entnommen werden.

Mit der folgenden Tastenbelegung zum Beispiel kann der Benutzer durch Drücken der Tastenkombination Alt+F6 den Tastaturfokus zum nächsten Fenster im Stapel in einer Anwendung wechseln.

```
Alt<Key>F6      window      f.next_key      transient
```

Hinweis – Vollständige Informationen zur Syntax der Tastenbelegung können der Handseite `dtwmrc(4)` entnommen werden.

▼ Angepasste Tastenbelegungssets erstellen

1. Die entsprechende Datei zur Bearbeitung öffnen:
 - persönlich: *Home-Verzeichnis*/`.dt/dtwmrc`
 - systemweit: `/etc/dt/config/sprache/sys.dtwmrc`

Weitere Informationen zur Erstellung dieser Dateien können dem Abschnitt “Konfigurationsdateien des Arbeitsbereichsmanagers” auf Seite 266 entnommen werden.

1. Einen neuen Tastenbelegungsset mit einem eindeutigen *TastenbelegungSetName* erstellen. Der Standardtastenbelegungsset des Desktop, `DtKeyBindings`, dient zur Anleitung.
2. Die Ressource `keyBindings` auf den neuen Setnamen setzen:
`Dtwm*keyBindings: TastenbelegungSetName`
3. ‘Arbeitsbereichsmanager neu starten’ im Arbeitsbereichsmenü auswählen.

Hinweis – Die neuen Tastenbelegungen ersetzen die existierenden Tastenbelegungen. Alle Dateien, die erhalten bleiben sollen, von `DtKeyBindings` in den neuen Set kopieren.

Wechseln zwischen Standardverhalten und angepasstem Verhalten

Umschalten zwischen dem OSF/Motif 1.2 Standard und dem CDE Desktopfenster:

1. Tastenkombination `Alt+Umschalt+Strg+!` drücken.
2. Im Dialogfenster auf ‘OK’ klicken.

Das Wechseln zum Standardverhalten entfernt das Bedienfeld und alle angepassten Tasten- und Maustastenbelegungen.

Anwendungsressourcen, Schriftarten und Farben verwalten

15 

Mit Hilfe des Umgebungsmanager oder durch Anpassung zusätzlicher Schriftarten und Farbressourcen kann aus einer großen Anzahl von Farben und Schriftarten für die Anzeige ausgewählt werden. Dieses Kapitel behandelt die Anpassung von Schriftarten und Farbressourcen.

Dieses Kapitel beschreibt auch, wie Umsetzungen für den `DtEditor` Fensterobjektanwendungen wie zum Beispiel den Texteditor des Desktop (`dtpad`) und Post (`dtmail`) angegeben werden. Das Kapitel beschreibt ebenfalls Alternativen für Direktaufrufe des `DtEditor` Fensterobjektanwendungsmenü die mit diesen Umsetzungen in Konflikt treten.

<i>Anwendungsressourcen einstellen</i>	<i>282</i>
<i>UNIX-Belegungen definieren</i>	<i>283</i>
<i>Schriftarten verwalten</i>	<i>287</i>
<i>Farben verwalten</i>	<i>292</i>
<i>Schattenstärken für Anwendungsfenster einstellen</i>	<i>301</i>

Anwendungsressourcen einstellen

Ressourcen werden von Anwendungen dazu verwendet, um gewisse Aspekte des Erscheinungsbilds und des Verhaltens einzustellen. Beispielsweise ermöglichen Ressourcen des Umgebungsmanagers (`dtstyle`), dem System anzugeben, wo es nach Dateien mit Informationen über Farbpaletten suchen soll:

```
dtstyle*paletteDirectories: /usr/dt/palettes/C \ Home-Verzeichnis/.dt/palettes
```

Standarddateien für die Desktop-Anwendungen befinden sich im Verzeichnis: `/usr/dt/app-defaults/sprache`

▼ Systemweite Ressourcen einstellen

- ♦ Die Ressourcen der Datei `/etc/dt/config/sprache/sys.resources` hinzufügen. (Möglicherweise muß die Datei erstellt werden.)

Wenn zum Beispiel in `/etc/dt/config/C/sys.resources` folgendes angegeben wird:

```
AnApplication*resource: wert
```

dann wird die Ressource `AnApplication*resource` bei der nächsten Anmeldung in die Eigenschaft `RESOURCE_MANAGER` von jedem Benutzer gestellt.

▼ Persönliche Ressourcen einstellen

1. Die Ressourcen der Datei `Home-Verzeichnis/.xdefaults` hinzufügen.
2. 'Ressourcen erneut laden' in der Anwendungsgruppe 'Desktop_Werkzeuge' doppelklicken.

Wie der Desktop Ressourcen lädt

Beim Neustart einer Session werden die Ressourcen von der Sessionmanager geladen. Mehr Informationen darüber, wie die Sessionmanager die Ressourcen in `RESOURCE_MANAGER` lädt, können dem Abschnitt "Session-Ressourcen starten" auf Seite 32 entnommen werden.

UNIX-Belegungen definieren

UNIX-Belegungen sind standardmäßig nicht aktiviert.

▼ EMACS-Umsetzungen angeben

Die folgende Prozedur gibt an:

- EMACS-Umsetzungen für DtEditor-Fensterobjektanwendungen, wie zum Beispiel den Texteditor des Desktop (dtpad) und Post (dtmail).
- Alternativen für DtEditor-Direktaufrufe des Fensterobjektanwendungsmenüs, die mit diesen Umsetzungen in Konflikt treten.

1. Folgende Zeile der Datei *Home-Verzeichnis*/.Xdefaults hinzufügen:

```
#include "/usr/dt/app-defaults/sprache/UNIXbindings"
```

wobei *sprache* der Wert der LANG-Umgebungsvariablen ist.

2. Session neu starten.

▼ EMACS-Umsetzungen ändern

1. Die Inhalte dieser Datei einfügen:

```
/usr/dt/app-defaults/sprache/UNIXbindings in  
Home-Verzeichnis/.Xdefaults.
```

2. Die Belegungen in der Datei .Xdefaults bearbeiten.

3. Nach Beendigung die Session neu starten.

UNIX-Belegungen über die UNIX-Belegungsdatei

Die Datei /usr/dt/app-defaults/*sprache*/UNIXbindings enthält folgende Belegungen:

Hinweis – Die Löschtaste löscht das vorhergehende Zeichen, wenn die UNIX-Belegungen aktiviert sind, und die Tastenkombination Umschalttaste+Löschtaste löscht das nächste Zeichen.

- Tabelle 15-1 zeigt eine Liste von neuen dtpad Belegungen für Menüdirektaufrufe und Direktaufrufstext, die mit den UNIX-Belegungen in Konflikt treten.

Tabelle 15-1 dtpad Neue Belegungen für

Menüdirektaufrufe und Direktaufrufstext	Neue Belegung
Dtpad*fileMenu.print.acceleratorText:	
Dtpad*fileMenu.print.accelerator:	
Dtpad*editMenu.undo.acceleratorText:	Steuertaste+_
Dtpad*editMenu.undo.accelerator:	Steuertaste<Taste>_
Dtpad*editMenu.paste.acceleratorText:	Umschalttaste+Einfügetaste
Dtpad*editMenu.paste.accelerator:	Umschalttaste<Taste>osflinsert
Dtpad*editMenu.selectAll.acceleratorText:	Steuertaste+/ /
Dtpad*editMenu.selectAll.accelerator:	Steuertaste<Taste>/

- Tabelle 15-2 zeigt eine Liste von neuen dtmail Belegungen beim Erstellen einer Nachricht für Menüdirektbeschleuniger und Direktaufrufstext, die mit den UNIX-Belegungen in Konflikt treten.

Tabelle 15-2 dtmail Neue Belegungen beim Erstellen einer Nachricht

Menüdirektaufrufe und Direktaufrufstext	Neue Belegung
Dtmail*ComposeDialog*menubar*Edit.Undo.acceleratorText:	Steuertaste+_
Dtmail*ComposeDialog*menubar*Edit.Undo.accelerator:	Steuertaste<Taste>_
Dtmail*ComposeDialog*menubar*Edit.Paste.acceleratorText:	Umschalttaste+Einfügetaste
Dtmail*ComposeDialog*menubar*Edit.Paste.accelerator:	Umschalttaste<Taste>osflinsert
Dtmail*ComposeDialog*menubar*Edit.Find/Change.acceleratorText:	Steuertaste+S
Dtmail*ComposeDialog*menubar*Edit.Find/Change.accelerator:	Steuertaste<Taste>s

- Die folgenden Umsetzungen bieten (GNU-Darstellung) EMACS-Steuerung und Metatastenbelegungen, sowie einige zusätzliche Belegungen. Wenn zulässig, ermöglichen sie auch, daß die Umschalttaste in Kombination mit der normalen Belegung benutzt werden kann, um die Richtung des Vorgangs umzukehren. Beispielsweise bewegt dann die Tastenkombination `Steuertaste+Umschalttaste+F` den Cursor ein Zeichen zurück, da `Steuertaste+Umschalttaste+F` den Cursor normalerweise ein Zeichen vorwärts bewegen.

Die zusätzlichen Belegungen sind:

Steuertaste+Komma	letztes Wort
Steuertaste+Umschalttaste+Komma	nächstes Wort
Steuertaste+Punkt	nächstes Wort
Steuertaste+Umschalttaste+Punkt	letztes Wort
Steuertaste+Eingabetaste	Dateiende
Steuertaste+Umschalttaste+Eingabetaste	Dateianfang

GNU EMACS belegt die Löschtaste eher mit Zeichen, die zurück löschen (`^`), als mit Zeichen, die vorwärts löschen (`_`). Die Tastenkombination `Meta+F` ist üblicherweise das mnemonische Zeichen für das Dateimenü, so daß die Belegung mit 'nächstes Wort' ignoriert wird. Für das nächste Wort sollte eine der anderen Belegungen verwendet werden (zum Beispiel `Steuertaste+Punkt`).

- Tabelle 15-3 zeigt eine Liste der `DtEditor.text` Umsetzungen

Tabelle 15-3 `DtEditor.text` Umsetzungen

Kombinationstaste	Taste	Aktionsroutine
<code>c ~s</code>	<code><Taste>a:</code>	<code>Zeilenanfang()\n\</code>
<code>c s</code>	<code><Taste>a:</code>	<code>Zeilenende()\n\</code>
<code>c ~s</code>	<code><Taste>b:</code>	<code>letztes Zeichen()\n\</code>
<code>c s</code>	<code><Taste>b:</code>	<code>nächstes Zeichen()\n\</code>
<code>c ~s</code>	<code><Taste>b:</code>	<code>letztes Zeichen()\n\</code>
<code>c s</code>	<code><Taste>b:</code>	<code>letztes Wort()\n\</code>
<code>m ~s</code>	<code><Taste>b:</code>	<code>letztes Wort()\n\</code>

Tabelle 15-3 DtEditor.text Umsetzungen (Fortsetzung)

Kombinationstaste	Taste	Aktionsroutine
m s	<Taste>b:	nächstes Wort()\n\
c ~s	<Taste>d:	nächstes Zeichen löschen()\n\
c s	<Taste>d:	letztes Zeichen löschen()\n\
m ~s	<Taste>d:	nächstes Wort löschen()\n\
m s	<Taste>d:	letztes Wort löschen()\n\
c ~s	<Taste>e:	Zeilenende()\n\
c s	<Taste>e:	Zeilenanfang()\n\
c ~s	<Taste>f:	nächstes Zeichen()\n\
c s	<Taste>f:	letztes Zeichen()\n\
m ~s	<Taste>f:	nächstes Wort()\n\
m s	<Taste>f:	letztes Wort()\n\
c	<Taste>j:	neue Zeile und einrücken()\n\
c ~s	<Taste>k:	bis Zeilenende löschen()\n\
c s	<Taste>k:	bis Zeilenanfang löschen()\n\
c	<Taste>l:	Anzeige neu aufbauen()\n\
c	<Taste>m:	neue Zeile()\n\
c s	<Taste>n:	Vorgang rückwärts()\n\
c ~s	<Taste>n:	Vorgang vorwärts()\n\
c	<Taste>o:	neue Zeile und sichern()\n\
c ~s	<Taste>p:	Vorgang rückwärts()\n\
c s	<Taste>p:	Vorgang vorwärts()\n\
c ~s	<Taste>u:	bis Zeilenanfang löschen()\n\
c s	<Taste>u:	bis Zeilenende löschen()\n\
c ~s	<Taste>v:	nächste Seite()\n\

Eine *Schriftart* ist eine Darstellungsart, bei der Textzeichen gedruckt oder angezeigt werden. Das Desktop enthält eine Vielfalt von Schriften in verschiedenen Arten und Größen.

Eine *Bitmap-Schriftart* besteht aus einer Matrix aus Punkten. Standardmäßig konfiguriert der Umgebungsmanager nur Bitmap-Schriftarten. Die Schriftart ist vollständig in einer Datei enthalten. Für die Gesamtheit von Schriftgrößen, Zeichenneigungen und Schriftstärken werden viele Dateien benötigt.

Schriftarten sind angegeben als Werte von Ressourcen und als Befehlsparameter. Mit dem XLFD-Namen (X Logical Font Description) wird eine gewünschte Schriftart angefordert. Das System findet die Schriftart, die am besten zu der gegebenen Beschreibung passt.

Schriftartressourcen des Desktop einstellen

Das Dialogfenster 'Schrift' des Umgebungsmanagers erlaubt, Schriftarten (bis zu sieben Größen) für Texteingaben, Bezeichnungen usw. einzustellen.

Vom Schriftart-Dialogfenster eingestellte Ressourcen

Wenn eine Schriftart ausgewählt wird, werden die folgenden Ressourcen in die Eigenschaft RESOURCE_MANAGER geschrieben:

- `SystemFont` wird verwendet für Systembereiche, wie zum Beispiel Menüleisten, Menüteufenster, Tasten, Umschalttasten und Bezeichnungen. Die folgende Ressource wird von `SystemFont` eingestellt:

<code>*FontList</code>	Angezeigt in Systembereichen von Desktop-Clients und anderen erstellten Clients, die die OSF/Motif-Hilfsprogramme benutzen.
------------------------	---
- `UserFont` wird zur Texteingabe in Fenster verwendet. Folgende Ressourcen werden von `UserFont` eingestellt:

<code>*Font</code>	unterstützt frühere Versionen von X-Anwendungen
<code>*FontSet</code>	Haupteinstellung
<code>*XmText*FontList</code>	angezeigt in Texteingabefenstern
<code>*XmTextField*FontList</code>	angezeigt in Texteingabefenstern

Vom Dialogfenster 'Schrift' verwendete Schriftarten

Die Fonts, die für jede Auswahl im Dialogfenster 'Schrift' benutzt werden, sind in der Ressourcendatei `/usr/dt/app-defaults/Dtstyle` angegeben. Bis zu sieben Größen können angegeben werden.

<code>NumFonts</code>	Anzahl der Schriftartgrößen im Dialogfenster
<code>SystemFont[1-7]</code>	bis zu sieben Ressourcen, die eine bestimmte Schriftart einer Dialogfensterauswahl für <code>SystemFont</code> zuordnen
<code>UserFont[1-7]</code>	bis zu sieben Ressourcen, die eine bestimmte Schriftart einer Dialogfensterauswahl für <code>UserFont</code> zuordnen.

Hinweis – Die Standardschriftarten für diese Ressourcen wurden für Lesbarkeit auf verschiedenen Bildschirmen ausgewählt. Wenn für eine Anwendung eine besondere Schriftart benötigt wird, sollte die Schriftart mit einer Schriftartressource der Anwendung eingestellt werden und nicht mit einer Änderung der Schriftarten des Desktop.

Mehr Informationen zu Anwendungsschriftarten können den Handseiten `DtStdAppFontNames (5)` und `DtStdInterfaceFontNames (5)` entnommen werden.

▼ Verfügbare Schriftarten auflisten

1. Folgendes eingeben:

```
xlsfonts [-optionen] [-fn muster]
```

Eine Liste mit XLFD-Namen und Alias-Namen von Schriftarten, die auf dem System zur Verfügung stehen, wird angezeigt. Bitmap-Schriftarten zeigen Werte in allen vierzehn XLFD-Feldern. Skalierbare Schriften zeigen Nullen in den Positionen *Pixelgröße*, *Punktgröße*, *AuflösungX*, und *AuflösungY*.

2. Um nach bestimmten Schriftarten zu suchen, sucht `xlsfonts` nach übereinstimmenden Mustern. Mit Platzhalterzeichen wird der Teil des Musters ersetzt, der nicht übereinstimmen soll.

3. Wenn `xlsfonts` keine Schriftartennamen zeigt, die mit `dt` beginnen, enthält der Schriftartenpfad keine Desktop-Schriftarten. Folgenden Befehl eingeben, um die Desktop-Schriftarten in die verfügbaren Schriftarten miteinzubeziehen:

```
xset +fp verzeichnisname
```

wobei *verzeichnisname* das Verzeichnis mit den Desktopschriftarten ist. Die Standardposition beim Neustart einer Session ist

```
/usr/dt/config/xfonts/sprache.
```

Zusätzliche Informationen:

- Die `xset` und `xlsfonts` Handseiten enthalten eine Liste der verfügbaren Optionen.
- *Das X-Fenster-System benutzen* erklärt Alias-Namen von Schriftarten und den `xset` Client.

▼ Schriftarten auf der Befehlszeile angeben

- ◆ Mit der `-xrm` Befehlszeilenoption wird eine Schriftartenressource für einen bestimmten Client angegeben. Beispiel:

```
anwendungsname -xrm "*bitstream-charter-medium-r-normal-8-88-75-75-p-45-iso8859-1"
```

X Logical Font Description (XLFD)

Eine Schriftart wird durch eine Liste mit vierzehn verschiedenen Kenndaten, die durch Bindestriche (-) getrennt sind, angegeben. Dies wird X Logical Font Description (XLFD) genannt. In einigen Fällen kann eine Eigenschaft in der Liste durch ein * Platzhalterzeichen ersetzt werden, und ein Zeichen in einer Eigenschaft kann durch ein ? Platzhalterzeichen ersetzt werden. Tabelle 15-4 enthält eine Liste von Spezifikationen der Eigenschaftszeichenfolgen für Schriftarten.

Die Form der Spezifikation der Eigenschaftszeichenfolge lautet:

"Entwickler-Markename-Schriftstärke-Zeichenneigung-Zeichenbreite-Zusätzliche Informationen-Pixelgröße-Punktgröße-AuflösungX-AuflösungY-Zeichenabstand-Durchschnittsbreite-Registrierungsberechtigter-Zeichensatz"

Table 15-4 Spezifikationen der Eigenschaftszeichenfolgen für Schriftarten

Eigenschafts- zeichenfolge	Definition
<i>Entwickler</i>	Zeichenfolge, die den Entwickler der Schriftart identifiziert
<i>Markename</i>	Zeichenfolge, die den Markennamen der Schriftart identifiziert
<i>Schriftstärke</i>	Zeichenfolge, die die relative Schriftstärke der Schriftart angibt, z.B. fett
<i>Zeichenneigung</i>	Code, der die Richtung der Zeichenneigung beschreibt R (Roman–keine Neigung) I (Italic–Neigung rechts) O (Oblique–Neigung rechts) RI (Reverse Italic–Neigung links) RO (Reverse Oblique–Neigung links)
<i>Zeichenbreite</i>	Zeichenfolge, die die Breite angibt, z.B. komprimiert oder erweitert
<i>Zusätzliche Informationen</i>	Zeichenfolge mit zusätzlichen Informationen, die gebraucht werden, um die Schriftart eindeutig zu identifizieren
<i>Pixelgröße</i>	Eine ganze Zahl, die die Größe eines Quadrats in Pixel angibt
<i>Punktgröße</i>	Eine ganze Zahl, die die Größe eines Quadrats in Dezimalzeichen angibt
<i>AuflösungX</i>	Eine ganze Zahl, die die horizontale Auflösung in Pixel angibt
<i>AuflösungY</i>	Eine ganze Zahl, die die vertikale Auflösung in Pixel angibt
<i>Zeichenabstand</i>	Ein Code, der den Abstand zwischen den Zeichen angibt M (Monospace--feste Zeichenbreite) P (Proportionaler Abstand--variable Zeichenbreite) C (Zeichenrahmen)

Tabelle 15-4 Spezifikationen der Eigenschaftszeichenfolgen für Schriftarten

Eigenschafts- zeichenfolge	Definition
<i>Durchschnittsbreite</i>	Eine ganze Zahl, die die Durchschnittsbreite in 1/10 Pixel angibt
<i>Registrierungsberechtigter</i>	Zeichenfolge, identifiziert den Registrierungsberechtigten, der den verschlüsselten Code registriert hat
<i>Zeichensatz</i>	Zeichenfolge, identifiziert den Zeichensatz im angegebenen Register

Beispiel

Der folgende XLFD-Name beschreibt die Schriftart `charter` von Bitstream, die die ISO8859-1 Standardverschlüsselung unterstützt:

```
-bitstream-charter-medium-r-normal--8-80-75-75-p-45-iso8859-1
```

Die Schriftart hat eine mittlere Schriftstärke, keine spezielle Neigung und eine normale Breite. Sie ist eine proportionale Schriftart mit einem quadratischen Rahmen von 8 Pixel oder 8,0 Punkten. Die horizontale und vertikale Auflösung ist bei beiden 75 Pixel. Die Durchschnittsbreite eines Zeichens beträgt 45 1/10 Pixel oder 4,5 Pixel.

Teile dieser Zeichenfolge können durch Platzhalterzeichen ersetzt werden. Das System verwendet die erste Schriftart, die es findet, die mit den angegebenen Teilen übereinstimmt.

Wenn nur eine acht-Pixel Schriftart `charter` benötigt wird, folgendes eingeben:

```
*-charter-*-*-*-*8-*
```

Farben verwalten

Dieser Abschnitt beschreibt:

- Wie der Umgebungsmanager Bildschirmfarben einstellt.
- Wie der Umgebungsmanager Ressourcen verwendet, um die Farben des Desktops zu steuern.

Farbpaletten

Eine Palette besteht aus einer Gruppe von Farbskalen. Die Farbskalen für die aktuelle Palette werden im Farbdialogfenster des Umgebungsmanager gezeigt.

Für jede Palette existiert eine Datei. Die Ressource `paletteDirectories` gibt die Verzeichnisse an, die Palettendateien enthalten. Standardmäßig enthält diese Ressource:

- integrierte Paletten: `/usr/dt/palettes`
- systemweite Paletten: `/etc/dt/palettes`
- persönliche Paletten: *Home-Verzeichnis*/`.dt/palettes`

Farbskalen

Jede Farbskala in der aktuellen Palette wird durch eine Farbtaste im Farbdialogfenster des Umgebungsmanagers dargestellt. Jede Farbe wird durch eine Farbskala-Kennung identifiziert - eine Zahl von 1 bis 8.



Abbildung 15-1 Farbskala-Kennungswerte für HIGH_COLOR

Jede Farbskala ist aus maximal fünf Farben zusammengesetzt. Jede Farbtaste zeigt die Hintergrundfarbe der Farbskala. Die fünf Farben in jeder Farbskala stellen die folgenden Ressourcen für Anzeigekomponenten dar:

<code>foreground</code>	Der Vordergrund eines Anwendungsfensters oder Fensterrahmens. Er ist entweder schwarz oder weiß. Er wird im allgemeinen für Text in Fenstern und Titel benutzt.
<code>background</code>	Der Hintergrund der Anwendung oder des Fensterrahmens.
<code>topShadowColor</code>	Die Farbe der oberen und linken Schattierungen der Bedienelemente von Anwendungen (z.B. Tasten) und Fensterrahmen.
<code>bottomShadowColor</code>	Die Farbe der unteren und rechten Schattierungen der Bedienelemente von Anwendungen und Fensterrahmen.
<code>selectColor</code>	Die Farbe, die den aktiven Status von bestimmten Bedienelementen anzeigt, z.B. aktive Umschalttasten und Tasten.

Die Anzahl von Farbskalen, die von jeder Palette benutzt werden, wird bestimmt von der Ressource `colorUse`, die der Benutzer mit dem Dialogfenster 'Anzahl der benutzten Farben' im Umgebungsmanager einstellen kann.

Farbwert angeben

Der Umgebungsmanager verwendet RGB-Werte, wenn er Farbinformationen in seine Palettendateien schreibt. Die Syntax für RGB-Zahlen lautet:

`#RedGreenBlue`

Red, *Green*, und *Blue* sind hexadezimale Zahlen, jede mit 1 bis 4 Stellen, die die Menge der benutzten Farbe anzeigen. Jede Farbe muß die gleiche Anzahl von Stellen haben. Deshalb bestehen gültige Farbwerte aus 3, 6, 9 oder 12 hexadezimalen Stellen.

Weiß könnte zum Beispiel auf jede der folgenden Möglichkeiten angegeben werden:

```
#fff
#ffffff
#ffffffffff
#ffffffffffff
```

Wenn eine Farbressource direkt eingestellt wird, kann entweder der Farbname oder der RGB-Wert verwendet werden. Die Datei `/usr/lib/X11/rgb.txt` enthält eine Liste aller benannten Farben.

Wie Farbskalen Ressourcen zugeordnet werden

Das Desktop ordnet Farbskalen über Ressourcen verschiedenen Bildelementen zu. Folgende Zuordnungen werden gemacht:

Ressource	Bildschirmelement
<code>activeColorSetId</code>	Farbe des aktiven Fensterrahmens
<code>inactiveColorSetId</code>	Farbe des inaktiven Fensterrahmens
<code>textColorSetId</code>	Bereiche zur Texteingabe
<code>primaryColorSetId</code>	Hauptintergrundbereiche der Anwendung
<code>secondaryColorSetId</code>	Menüleiste, Menüs und Dialogfenster der Anwendung

Diese Ressourcen nehmen eine Farbskalen-Kennung als ihren Wert an. Bildelemente mit Farbskalen-Kennungen farbig zu machen, ermöglicht dem Element, dynamisch zu dem neuen Farbschema zu wechseln, wenn eine neue Palette mit dem Umgebungsmanager ausgewählt wird.

Diese Ressourcen können für einzelne Anwendungen verwendet werden. Beispielsweise zeigt die folgende Zeile, wie man alle `dtterm`-Fenster visuell gruppieren könnte, indem Farbskala 8 als Grundfarbe verwendet wird.

```
dtterm*primaryColorSetId: 8
```

Zuordnungen der Standardfarbskala

Die Farbskalen-Kennung, die für Bildelemente verwendet wird, hängt davon ab, wieviel Farben im Umgebungsmanager eingestellt sind.

- Maximale Anzahl Farben (8 Farbskalen)—Umgebungsmanager Einstellung ‘Mehr Farben für Desktop’:

Farbskalen-Kennung	Anzeigeelement
1	Farbe des aktiven Fensterrahmens
2	Farbe des inaktiven Fensterrahmens

3	Standardmäßig nicht belegt
4	Bereiche zur Texteingabe
5	Haupt hintergrundbereiche der Anwendung
6	Menüleiste, Menüs und Dialogfenster der Applikation
7	Standardmäßig nicht belegt
8	Hintergrund des Bedienfelds
<ul style="list-style-type: none"> • Mittlere Farbe (4 Farbskalen)—Umgebungsmanager Einstellung ‘Mehr Farben für Anwendungen’: 	
Farbskalen-Kennung	Bildschirmelement
1	Farbe des aktiven Fensterrahmens
2	Farbe des inaktiven Fensterrahmens
3	Hintergrundfarbe der Anwendung und des Bedienfelds
4	Bereiche zur Texteingabe
<ul style="list-style-type: none"> • Niedrige Farbe (2 Farbskalen)—Umgebungsmanager Einstellung ‘Die meisten Farben für Anwendungen’: 	
Farbskalen-Kennung	Bildschirmelement
1	Aktiver Fensterrahmen, Auswahl-tasten Arbeitsbereich
2	Alle anderen Bildschirmelemente

Farbe mit dem Umgebungsmanager steuern

Mit dem Umgebungsmanager können Farben für Desktop-Anwendungen und andere kooperierende Anwendungen dynamisch geändert werden. Die Vorder- und Hintergrundfarben, die vom Umgebungsmanager eingestellt werden, stehen nicht-kooperierenden Anwendungen zur Verfügung.

- Damit ein Client die Farbänderungen des Umgebungsmanagers nutzen kann, muß er die Motifbibliothek des Desktop verwenden. Clients, die mit anderen Hilfsprogrammen geschrieben wurden, können die Farbe nicht dynamisch ändern. Farbänderungen für diese Clients haben keine Auswirkung, bevor der Client neu gestartet wird.
- Es dürfen keine anderen spezifischen Farbressourcen für den Client angewendet werden. Dies beinhaltet vom Benutzer angegebene Ressourcen, Standardwerte für Anwendungen und Ressourcen, die in die Anwendung integriert sind
- Clients können die Ressourcen `primaryColorSetId` und `secondaryColorSetId` angeben, um bestimmte Farben einer Desktop-Palette zu verwenden.

Anzahl der vom Umgebungsmanager benutzten Farben

Die Anzahl der Farben, die vom Umgebungsmanager benutzt werden, hängt von den Werten der folgenden Ressourcen ab:

<code>colorUse</code>	Konfiguriert die Anzahl der Farben, die vom Desktop benutzt werden.
<code>shadowPixmap</code>	Befiehlt dem Desktop, die zwei Schattenfarben durch Pixmaps zu ersetzen.
<code>foregroundColor</code>	Gibt an, ob sich die Vordergrundfarbe dynamisch ändert.
<code>dynamicColor</code>	Steuert, ob Anwendungen beim Wechseln von Paletten die Farbe ändern.

Tabelle 15-5 zeigt eine Liste der maximalen Anzahl von Farben, die vom Desktop zugeordnet werden.

Tabelle 15-5 Anzahl der Desktopfarben

Anzeige	Maximale Anzahl der Farben	Anzahl abgeleitet von
B_W (SCHWARZ-WEIß)	2	Schwarz und Weiß
LOW_COLOR (NIEDRIGE FARBE)	12	Zwei Farbskalen mal fünf Farben plus Schwarz und Weiß
MEDIUM_COLOR (MITTLERE FARBE)	22	Vier Farbskalen mal fünf Farben plus Schwarz und Weiß
HIGH_COLOR (MAXIMALE ANZAHL FARBEN)	42	Acht Farbskalen mal fünf Farben plus Schwarz und Weiß

Die maximale Anzahl von Farben bestimmen:

1. Die Anzahl der Farbskalen in der Palette mit der Anzahl der Farben in jeder Farbskala multiplizieren.
2. 2 addieren (für Schwarz und Weiß).

Bei der folgenden Konfiguration wären jedoch nur zehn Farben in der Palette; vier Farbskalen mal zwei Farben in jeder Skala (background und selectColor) plus Schwarz und Weiß.

```
*colorUse:MEDIUM_COLOR
*shadowPixmap: True
*foregroundColor:White
```

Hinweis – Mehrfarbige Symbole benötigen vierzehn zusätzliche Farben.

Ressource colorUse

Der Standardwert der Ressource `colorUse` ist `MEDIUM_COLOR` (mittlere Farbe). Der Wert dieser Ressource beeinflusst die Anzahl der Farbskalen in einer Palette. Andere Ressourcen beeinflussen die Anzahl der Farben für Schattierungen. Der Wert der Ressource `colorUse` beeinflusst ebenfalls die Benutzung von mehrfarbigen Symbolen.

Wert	Beschreibung
<code>B_W</code>	Umgebungsmanager Einstellung "Schwarz und Weiß" Anzeige mit 1 bis 3 Farbspeichern Anzahl der Farbskalen: 2 Maximale Anzahl der Farben: 2 Standardanzahl der Farben: 2 Keine mehrfarbigen Symbole
<code>LOW_COLOR</code>	Umgebungsmanager Einstellung "Die meisten Farben für Anwendungen" Anzeige mit 4 bis 5 Farbspeichern Anzahl der Farbskalen: 2 Maximale Anzahl der Farben: 12 Standardanzahl der Farben: 12 Keine mehrfarbigen Symbole
<code>MEDIUM_COLOR</code>	Umgebungsmanager Einstellung "Mehr Farben für Anwendung" Anzeige mit 6 Farbspeichern Anzahl der Farbskalen: 4 Maximale Anzahl der Farben: 22 Standardanzahl der Farben: 22 Mehrfarbige Symbole
<code>HIGH_COLOR</code>	Umgebungsmanager Einstellung "Mehr Farben für das Desktop" Anzeige mit 7 oder mehr Farbspeichern Anzahl der Farbskalen: 8 Maximale Anzahl der Farben: 42 Standardanzahl der Farben: 42 Mehrfarbige Symbole\
<code>default</code>	Das Desktop wählt den richtigen Wert für die Anzeige. (Um die Anzahl der Farben zu verringern, die vom Desktop bei der Anzeige von maximalen Farben benutzt wird, ist die Standardressource <code>colorUse</code> auf <code>MEDIUM_COLOR</code> gesetzt.)

Ressource shadowPixmap

Die Ressource `shadowPixmap` ersetzt die zwei Schattenfarben auf dem Desktop durch Pixmap. Diese Pixmap mischen die Hintergrundfarbe mit Schwarz oder Weiß, um obere oder untere Schattierungswerte zu simulieren. Damit wird die Anzahl der benötigten Farben um zwei verringert, da Farbzellen nicht den Schattenfarben zugeordnet werden müssen.

Wert	Beschreibung
True	Das Desktop erstellt ein <code>topShadowPixmap</code> und ein <code>bottomShadowPixmap</code> anstatt die Schattenfarben zu verwenden.
False	<code>topShadowColor</code> und <code>bottomShadowColor</code> aus der Palette werden verwendet.

Der Standardwert für `shadowPixmap` hängt von der verwendeten Ressource `colorUse` und der Hardwareunterstützung für den Bildschirm ab.

Ressource foregroundColor

Die Ressource `foregroundColor` gibt an, wie der Vordergrund in einer Palette konfiguriert ist.

Einstellung	Ergebnis
White	Der Vordergrund ist auf Weiß eingestellt.
Black	Der Vordergrund ist auf Schwarz eingestellt.
Dynamic	Der (Standard)-Vordergrund ist dynamisch auf Schwarz oder Weiß eingestellt, abhängig von dem Wert von <code>background</code> . Wenn zum Beispiel weiße Buchstaben auf einem gelben Hintergrund schwer zu lesen sind, wählt das System schwarz.

Wenn `foregroundColor` entweder auf Schwarz oder Weiß eingestellt ist, wird die Anzahl der Farben in der Farbskala um eins verringert, und der Vordergrund ändert sich nicht, wenn sich die Hintergrundfarbe ändert.

Der Standardwert für `foregroundColor` ist `Dynamic`, ausgenommen `colorUse` hat den Wert `B_W`.

Ressource `dynamicColor`

Die Ressource `dynamicColor` steuert, ob Anwendungen die Farbe dynamisch ändern; das heißt, ob die Clients die Farbe ändern, wenn die Paletten gewechselt werden.

Wert	Beschreibung
True	Clients ändern die Farbe dynamisch, wenn eine neue Palette ausgewählt wird. Dies ist der Standardwert.
False	Clients ändern die Farbe nicht dynamisch. Wenn eine neue Palette ausgewählt wird, verwenden die Clients die neuen Farben erst nach einem Neustart der Session.

Wenn der Wert der Ressource `dynamicColor` True ist, ordnen Clients, die Farben nicht dynamisch ändern können (Nicht-Motif Anwendungen), andere Zellen der Farbtabelle zu als Clients, die die Farbe dynamisch ändern können.

Hinweis – Da sich alle Clients dieselben Farbzellen teilen können, wird die Anzahl der vom Desktop verarbeiteten Farben reduziert, wenn `dynamicColor` auf False eingestellt wird.

Schattenstärken für Anwendungsfenster einstellen

Das Desktop definiert eine Standardschattenstärke von einem Pixel für die Komponenten in Anwendungsfenstern, wie zum Beispiel Tastenschatten und die Hervorhebung von bestimmten Bereichen. Motif 1.2-Anwendungen verwenden diesen Ressourcenwert; andere Anwendungen können diesen Ressourcenwert nicht verwenden und erscheinen deshalb verschieden auf dem Bildschirm.

Die Schattenstärke für Nicht-Motif 1.2-Anwendungen auf ein Pixel einstellen:

1. Als Benutzer `'root'` anmelden.
2. Die Datei `/etc/dt/config/sprache/sys.resources` erstellen.
3. Die applikationsspezifische Ressource in `/etc/dt/config/sprache/sys.resources` wie folgt angeben:
`application_class_name*XmCascadeButton*shadowThickness: 1`

Mehr Informationen über die Neubelegung von Standardressourcen des Systems und das Angeben zusätzlicher Ressourcen für alle Desktopbenutzer können dem Abschnitt “Session-Ressourcen starten” auf Seite 32 entnommen werden.

Länderspezifische Desktop-Sessions konfigurieren

16 

Um länderspezifische Desktop-Sessions zu konfigurieren, müssen folgende Schritte unternommen werden:

- Die LANG-Umgebungsvariable und andere NLS- (National Language Support) Umgebungsvariablen einstellen
- Auf sprachenabhängige Nachrichtenkataloge und Ressourcendateien zugreifen
- Fernausführung von Anwendungen über internationalisierte Systeme.

<i>LANG-Umgebungsvariable verwalten</i>	<i>304</i>
<i>Schriftarten finden</i>	<i>307</i>
<i>Ressourcendateien für Standardwerte von Anwendungen eingrenzen</i>	<i>307</i>
<i>Aktions- und Datenarten eingrenzen</i>	<i>308</i>
<i>Symbole und Bitmaps eingrenzen</i>	<i>308</i>
<i>Hilfebände eingrenzen</i>	<i>310</i>
<i>Nachrichtenkataloge eingrenzen</i>	<i>310</i>
<i>Fernausführung von länderspezifischen Desktopanwendungen</i>	<i>310</i>
<i>Tastaturbelegung zurücksetzen</i>	<i>311</i>

LANG-Umgebungsvariable verwalten

Die LANG-Umgebungsvariable muß für das Desktop eingestellt werden, das die sprachenspezifischen Routinen des Betriebssystems benutzt. Das Desktop unterstützt:

- Westeuropäische und Romanische Sprachen
- Japanisch
- Traditionelles Chinesisch
- Vereinfachtes Chinesisch
- Koreanisch

Hinweis – Möglicherweise wurde eine Unterstützung für andere Sprachen bereits vom Händler installiert.

LANG kann auf jeden Wert eingestellt werden, der vom Betriebssystem unterstützt wird. Das Menü 'Optionen' des Anmeldebildschirms zeigt die Liste der unterstützten Sprachen und Gebiete.

Es gibt vier Möglichkeiten, LANG für das Desktop einzustellen:

- Eine Ressource in der Datei `xconfig` bearbeiten
- Mit dem Menü 'Optionen' des Anmeldebildschirms
- Eine ausführbare Prozedur `sh` oder `ksh` `xsession.d` erstellen. (Mehr Informationen zum Benutzen einer `Xsession.d` Prozedur können dem Abschnitt "Prozeduren `Xsession.d` als Ausgangsobjekte verwenden" auf Seite 28 entnommen werden.)
- Die Datei `.dtprofile` des Benutzers bearbeiten.

Wenn LANG eingestellt ist, verwendet das Desktop die folgenden sprachenspezifischen Dateien, um die länderspezifische Schnittstelle zu bestimmen.

Farben `/usr/dt/palettes/desc.sprache`

Hintergrundmuster `/usr/dt/backdrops/desc.sprache`

Sprache für mehrere Benutzer einstellen

Wenn die Sprache mit einer `xconfig` Datei eingestellt wird, ist der Anmeldebildschirm länderspezifisch, und `LANG` ist für alle Benutzer eingestellt. Dies ist die einzige Möglichkeit, `LANG` für alle Bildschirme in Systemen mit mehreren Bildschirmen zu ändern. (Um `xconfig`, zu ändern, `/usr/dt/config/Xconfig` in `/etc/dt/config/Xconfig` kopieren.)

Die Sprache wird eingestellt, indem die folgende Zeile in `/etc/dt/config/Xconfig` gestellt wird:

```
dtlogin.host_bildschirm.language: sprache
```

Zum Beispiel setzt die folgende Zeile `LANG` auf `Swedish_locale` bei der Anzeige `my_host:0`.

```
dtlogin.my_host_0.language: Swedish_locale
```

Der `dtlogin`-Client liest den entsprechenden Nachrichtenkatalog für diese Sprache und baut den länderspezifischen Anmeldebildschirm auf. Der `dtlogin`-Client bestimmt dann die Liste der länderspezifischen Angaben mit Hilfe der folgenden Ressourcen in der Ressourcendatei `/etc/dt/config/Xresources`:

- `dtlogin*language`
- `dtlogin*languageList`
- `dtlogin*languageName`

Die Datei `xconfig` muß möglicherweise die `NLSPATH`-Umgebungsvariable entsprechend der ausgewählten Sprache einstellen. Wenn dies nicht der Fall ist oder wenn `NLSPATH` selbst eingestellt werden soll, können mehr Informationen dem Abschnitt "NLSPATH-Umgebungsvariable" auf Seite 307 entnommen werden.

Sprache für eine Session einstellen

Um die Sprache für eine Session einzustellen, das Menü 'Optionen' des Anmeldebildschirms verwenden. Der Anmeldebildschirm ist länderspezifisch und `LANG` ist für den Benutzer eingestellt. Am Ende der Session kehrt `LANG` zum Standardwert (eingestellt in `dtlogin`) zurück.

Sprache für einen Benutzer einstellen

Ein Benutzer kann die Einstellung des LANG bei der Anmeldung in der Datei *Home-Verzeichnis/.dtprofile* neu belegen. Der Anmeldebildschirm ist nicht länderspezifisch und LANG wird für den Benutzer eingestellt.

- Wenn `sh` oder `ksh` benutzt wird:

```
LANG=sprache
export LANG
```

- Wenn `csh` benutzt wird:

```
setenv LANG sprache
```

LANG-Umgebungsvariable und Session-Konfiguration

Die LANG-Umgebungsvariable ändert den Verzeichnisnamen, der für die Session-Konfigurationsdateien gesucht wird.

Es gibt folgende länderspezifische Session-Konfigurationsdateien:

- `/usr/dt/config/sprache/Xresources` (Anmeldemanager-Ressourcendatei)
- `/usr/dt/config/sprache/sys.font` (Sessionmanager-Ressourcendatei)
- `/usr/dt/config/sprache/sys.resources` (Sessionmanager-Ressourcendatei)
- `/usr/dt/config/sprache/sys.session` (Sessionmanager ausführbare Shell)
- `/usr/dt/config/sprache/sys.dtwmrc` (Fenstersteuerung-Ressourcendatei)
- `/usr/dt/appconfig/types/sprache/dtwm.fp` (Fenstersteuerung-Bedienfeld)

Andere NLS-Umgebungsvariablen einstellen

Neben LANG gibt es andere NLS-Umgebungsvariablen, wie zum Beispiel LC_CTYPE und LC_ALL. Diese Variablen werden weder von der Sprachenressource `dtlogin` beeinflusst, noch von dem Menü Optionen des Anmeldebildschirms. Sie müssen in die folgenden Dateien gestellt werden:

- systemweite Variablen: `/etc/dt/config/Xsession.d`
- persönliche Variablen: *Home-Verzeichnis/.dtprofile*

NLSPATH-Umgebungsvariable

Die NLSPATH-Umgebungsvariable bestimmt die Verzeichnispfade, die Anwendungen nach Nachrichtenkatalogen durchsuchen. Sowohl LANG als auch NLSPATH müssen eingestellt sein, um diese Nachrichtenkataloge zu benutzen. Mehr Informationen über die Position von länderspezifischen Nachrichten können dem Abschnitt “Nachrichtenkataloge eingrenzen” auf Seite 310 entnommen werden. Die meisten Desktop-Clients werden beim Neustart den Pfad vor NLSPATH setzen.

Schriftarten finden

Die im Desktop enthaltenen Schriftarten befinden sich im Verzeichnis `/usr/lib/X11/fonts`. Jedes Verzeichnis enthält eine Verzeichnisdatei `fonts.dir` und eine Alias-Datei `fonts.alias`. Mehr Informationen über das Erstellen der Dateien `fonts.dir` und `fonts.alias` können der Hilfeseite `mkfontdir` entnommen werden..

Mit dem Befehl `xlsfonts` können alle verfügbaren Schriftarten auf einem Server aufgelistet werden. Mit dem Befehl `xset` können dem Server Schriftarten hinzugefügt oder von ihm gelöscht werden.

Ressourcendateien für Standardwerte von Anwendungen eingrenzen

Die Standardposition der Datei `app-defaults` für den Desktop-Client ist `/usr/dt/app-defaults/sprache`. Wenn zum Beispiel LANG auf `Swedish_locale` eingestellt ist, dann suchen Anwendungen nach ihrer `app-defaults` Datei in `/usr/dt/app-defaults/Swedish_locale`. Wenn LANG nicht eingestellt ist, wird *sprache* ignoriert und die Anwendungen suchen ihre `app-defaults` Datei in `/usr/app-defaults/C`.

Um die Position der Datei `app-defaults` zu ändern, die `XFILESEARCHPATH`-Umgebungsvariable verwenden. Um zum Beispiel `app-defaults` nach `/users` zu verschieben, `XFILESEARCHPATH` in `/usr/app-defaults/sprache/klassenname` stellen.

Wenn `XFILESEARCHPATH` in `Home-Verzeichnis/.dtprofile` gestellt wird, gilt der Wert für alle Desktop- und X-Clients, die ausgeführt werden. Nicht-Clients finden ihre Ressourcendateien nicht, bevor sie in das von `XFILESEARCH` angegebene Verzeichnis kopiert oder mit ihm verbunden werden.

Aktions- und Datenarten eingrenzen

Hinweis – Um eine Datei in das Verzeichnis `/usr/dt/appconfig` anzupassen, die Datei vor dem Anpassen in das Verzeichnis `/etc/dt/appconfig` kopieren.

Der Suchpfad für Aktions- und Datenartendefinitionsdateien enthält sprachenspezifische Verzeichnisse in:

- persönlich: *Home-Verzeichnis*/dt/types
- systemweit: `/etc/dt/appconfig/types/sprache`
- integriert: `/usr/dt/appconfig/types/sprache`

Der Suchpfad für Konfigurationsdateien des Anwendungsmanagers lautet:

- persönlich: *Home-Verzeichnis*/dt/appmanager
- systemweit: `/etc/dt/appconfig/appmanager/sprache`
- integriert: `/usr/dt/appconfig/appmanager/sprache`

Datei- und Verzeichnisnamen in diesem Verzeichnis sind länderspezifisch.

Symbole und Bitmaps eingrenzen

Um ein Symbol einzugrenzen, das Symbol mit dem Symboleditor bearbeiten und speichern in:

`/etc/dt/appconfig/icons/sprache`

Wenn es in einem anderen Verzeichnis gespeichert wurde, die `XMICONSEARCHPATH`-Umgebungsvariable so einstellen, daß sie das Verzeichnis mit dem gespeicherten Symbol enthält. Die `XMICONBMSEARCHPATH`-Umgebungsvariable steuert den Pfad, mit dem nach Symbolen gesucht wird.

Namen von Hintergrundmustern eingrenzen

Die Eingrenzung von Hintergrundmustern wird mit Hilfe von Beschreibungsdateien durchgeführt (`desc.sprache` and `desc.backdrops`). Es gibt kein spezifisch eingegrenztes Verzeichnis für Hintergrundmusterdateien (wie z.B. `/usr/dt/backdrops/sprache`). Alle länderspezifischen Angaben

benutzen diesselben Hintergrundmusterdateien, haben aber ihre eigene desc.*sprache* Datei, die die übersetzten Namen der Hintergrundmuster enthält.

Die Beschreibungsdatei enthält Ressourcenspezifikationen für die übersetzten Namen der Hintergrundmuster. Beispiel:

```
Backdrops*Corduroy.desc:  Velours
Backdrops*DarkPaper.desc: PapierKraft
Backdrops*Foreground.desc: AvantPlan
```

Die Datei desc.*sprache* wird dazu benutzt, die Beschreibung der Hintergrundmuster für länderspezifische Angaben *sprache* abzurufen, um das Hintergrundmuster im Style Manager anzuzeigen. Wenn es eine Beschreibungsspezifikation gibt, wird sie in der Hintergrundmusterliste des Style Manager angezeigt. Sonst wird der Name der Hintergrundmusterdatei benutzt.

Benutzer können ihre eigenen Hintergrundmusterbeschreibungen der Datei *Home-Verzeichnis/.dt/backdrops/desc.backdrops* hinzufügen. Mit dieser Datei werden Hintergrundmusterbeschreibungen von allen Hintergrundmustern abgerufen, die der Benutzer hinzugefügt hat, unabhängig von den länderspezifischen Angaben.

Der Suchpfad für die description Dateien lautet:

- persönlich: *Home-Verzeichnis/.dt/backdrops/desc.backdrops*
- systemweit: */etc/dt/backdrops/desc.sprache*
- integriert: */usr/dt/backdrops/desc.sprache*

Palettennamen eingrenzen

Die Eingrenzung von Paletten wird mit Hilfe von Beschreibungsdateien durchgeführt (desc.*sprache* und desc.*palettes*). Es gibt kein spezifisch eingegrenztes Verzeichnis (wie z.B. */usr/dt/palettes/sprache*). Alle länderspezifischen Angaben benutzen diesselben Palettendateien, haben aber ihre eigene desc.*palettes* Datei mit den übersetzten Namen der Paletten.

Die Beschreibungsdatei enthält Ressourcenspezifikationen für die übersetzten Palettennamen. Beispiel:

```
Palettes*Cardamon.desc:  Cardamone
Palettes*Cinnamon.desc: Cannelle
Palettes*Clove.desc:    Brun
```

Die Datei `desc.sprache` wird dazu benutzt, die Beschreibung der Paletten für länderspezifische Angaben `sprache` abzurufen, um die Palette in der Liste des Style Manager anzuzeigen. Wenn es eine Beschreibungsspezifikation gibt, wird sie in der Palettenliste des Style Managers angezeigt. Sonst wird der Name der Palettendatei benutzt.

Benutzer können ihre eigenen Palettenbeschreibungen der Datei `Home-Verzeichnis/.dt/palettes/desc.palettes` hinzufügen. Mit dieser Datei werden Palettenbeschreibungen von allen Paletten abgerufen, die der Benutzer hinzugefügt hat, unabhängig von den länderspezifischen Angaben.

Der Suchpfad für die Beschreibungsdateien lautet:

- persönlich: `Home-Verzeichnis/.dt/palettes/desc.palettes`
- systemweit: `/etc/dt/palettes/desc.sprache`
- integriert: `/usr/dt/palettes/desc.sprache`

Hilfebände eingrenzen

Wenn ein Hilfeband eingegrenzt wurde, muß er in einem der folgenden Verzeichnisse gespeichert werden. Der erste gefundene Hilfeband wird benutzt. Die Verzeichnisse werden in der folgenden Reihenfolge durchsucht:

- persönlich: `Home-Verzeichnis/.dt/help`
- systemweit: `/etc/dt/appconfig/help/sprache`
- integriert: `/usr/dt/appconfig/help/sprache`

Nachrichtenkataloge eingrenzen

Wenn ein Nachrichtenkatalog eingegrenzt wurde, muß er im folgenden Verzeichnis gespeichert werden:

`/usr/dt/lib/nls/msg/sprache.`

Diese Verzeichnisse enthalten die `*.cat` Dateien.

Fernausführung von länderspezifischen Desktopanwendungen

Länderspezifische Desktopanwendungen können an jedem fernen Ausführungshost aufgerufen werden, der eine ähnliche länderspezifische Desktopinstallation hat. Die Werte der NLS-ähnlichen Umgebungsvariablen

am Host, der die Anwendung aufruft, werden dem entfernten Host übergeben, wenn die Anwendung startet. Die Umgebungsvariablen enthalten jedoch keine Hostinformationen.

Tastaturbelegung zurücksetzen

Wenn unerwartete Zeichen oder Verhaltensmuster erscheinen oder wenn Zeichen nicht angezeigt oder eingegeben werden können, muß entweder die Tastaturbelegung zurückgesetzt oder installiert werden oder die Eingabemethode muß geändert werden.

Die Eingabemethode wird bestimmt von den LC_CTYPE-, LANG- oder LC_ALL-Umgebungsvariablen oder der Sprache, die von der Option `-lang` angegeben wird.

Wenn der Benutzer zum Beispiel ein Terminal mit der länderspezifischen Angabe in der C-Umgebung in einer POSIX-Shell öffnen möchte, wie z.B.:

```
LANG=C dtterm
```

Dieses neue Terminal verwendet die länderspezifische Angabe in der C-Umgebung, einschließlich der C-Eingabemethode und Schriftarten. Wenn eine sprachenspezifische Tastatur benutzt wird, akzeptiert die Eingabemethode möglicherweise keine erweiterten Zeichen zur Eingabe. Wenn die länderspezifische Angabe in der C-Umgebung mit einer sprachenspezifischen Tastatur benutzt wird, muß der Benutzer die LC_CTYPE- (oder LANG oder LC_ALL) Umgebungsvariable vor dem Aufrufen des Terminals auf einen entsprechenden Wert setzen.

Um beispielsweise die länderspezifische Angabe in der C-Umgebung mit der deutschen Tastatur zu benutzen, folgendes eingeben:

```
LANG=C LC_CTYPE=DeDE dtterm
```

Wenn der X-Server zurückgesetzt wurde und die Tastenbelegungen initialisiert wurden, kann die richtige Tastenbelegung am Server mit dem Befehl `xmodmap` zurückgesetzt werden.

Index

Symbols

%B 130
%DatabaseHost% 198
%DisplayHost% 198
%H 130
%L 130
%LocalHost% 198
%M 130
%SessionHost% 198
* Platzhalterzeichen 214
? Platzhalterzeichen 214

A

Abstand zwischen Bedienelementen
regeln 252
actionIcon, Ressource 182
actions
search path, See database search path
ACTIONS Feld 212
activeColorSetId Ressource 295
Aktion erstellen
Aktionsbefehlssyntax 162
Aktionsname 162
Dateiargument angeben 163
Dateieingabe 164
Datentyp erstellen 150, 164
Datentypname 166
Einführung 157
Einschränkungen 158
erstellte Konfigurationsdatei 158
Funktionsweise 157
Hauptfenster 162
starten 161
Symbol angeben 171
Verwendung 160
Aktionen
als anderer Benutzer ausführen 201
andere Aktionen ausführen 200, 201
ändern 183
Argumentanzahl einschränken 196
Argumente 159, 186
Argumente einschränken 195
Ausführungszeile 185
austauschbare Argumente 192
bearbeiten 183
Bedienfeld 144
Beispiel 177, 178
benötigt für Registrierung 70
Bezeichnungen 181, 203
COMMAND 174
Datei anfordern 188
Dateiargumente 149
Dateien darstellen, siehe Aktionsdatei
180
Datentypeinschränkungen 153

Datentypen 152
 Datentypen zuordnen 211
 Einführung 141
 Einführungi 142
 Einschränkungen 159
 erneut laden 179
 Felder 176
 Fensterunterstützung 163, 193
 ferne Anwendungen ausführen 198
 integriert mit dtappintegrate 84
 keine Bildschirmausgabe 163
 Konfigurationsdateien 176
 manuell erstellen 176
 manuell erstellen 173, 175
 mehrere übergebene Dateien
 akzeptieren 192
 Name 158, 176, 181
 nicht austauschbare Argumente 191
 Nicht-Datei Argumente 189
 ohne Argumente 187
 ohne Daten 187
 Parameter, siehe Argumente
 Positionen 202
 Server 114
 Shells zur Verfügung stellen 190
 Standardsymbol 182
 Standardterminal 194
 Suchpfad, siehe Datenbanksuchpfad
 Symbol zuordnen 227
 Symboldarstellung 180
 Symbole 171, 181
 Symbole für Anwendungen erstellen
 146
 Terminaloptionen 194
 Terminalunterstützung 194
 TT_MSG 175
 Typen 174, 176
 übergebene Datei akzeptieren oder
 anfordern 188
 übergebene Dateien akzeptieren 149,
 187
 Umgebungsvariable 200
 Variable in Definition 199
 Verhalten bei Doppelklicken und
 Übergabe 196
 verwendet in Menüs 145
 Verwendungen 150
 vom Datentyp eingeschränkt 195
 Vorrangregeln 184
 Zeichenfolgevariable 199
 zuordnen 153
 Zuordnungen 175
 Aktionen anfordern 188
 Aktionen bearbeiten 183
 Aktionen erneut laden Aktion 179
 Aktionsdatei 158, 180
 Definition 146
 erstellen 80, 180
 Inhalt 147
 Aktionsdefinitionsdatei, mit Aktion
 erstellt 158
 Aktionsnamenfeld in Aktion erstellen 162
 Aktions-Server, siehe Datenbank-Server
 Aktionssuchpfad, siehe
 Datenbanksuchpfad
 Aktionssymbol 147, 180
 erstellen 180
 startet Anwendung 75
 Aktionssymbolsteuerung, in Aktion
 erstellen 162
 Aktuelle Session 26
 aktuelle Session 26
 ALTERNATE_ICON Feld 255
 ANIMATION Definition 258
 Animation für Bedienfeld 258
 Anmeldeabrechnungen 101
 Anmeldefenster
 anpassen 12
 Anzeige an einem Bildschirm im Netz
 7
 Begrüßung 14
 Erscheinungsbild ändern 13
 Inhalt des Menüs Sprache ändern 18
 Ressourcen 13
 Schriftarten 14
 Standardsprache ändern 17
 Verhalten an einzelnen Bildschirmen
 16
 Verhalten ändern 15

-
- X -Server Umgebung 17
 - X -Server Zugriff 16
 - Anmeldefenster, Verhalten ändern 15
 - Anmelde-Manager
 - anpassen 1
 - Befehle eingeben 18
 - Definition 1
 - Fehlerprüfung 11
 - Konfigurationsdateien 23
 - Ressourcen 13
 - Anmelde-Server
 - Anmeldefenster 1
 - Anmeldung über Befehlszeile 6
 - Benutzeridentifikation 1
 - Benutzerpfad 21
 - Bildschirmanschluß 1
 - Fehlerprüfung 11
 - kein Bitmap-Bildschirm 5
 - konfigurieren 105
 - Konsole mit zeichenorientierter Anzeige 7
 - ohne lokalen Bildschirm ausführen 5
 - Prozeß-ID 3
 - Prozeß-ID löschen 11
 - Session starten 1
 - starten 2
 - stoppen 11
 - System-Shell 21
 - Überblick 2
 - Umgebung 20
 - von der Befehlszeile starten 2
 - Zeitzone ändern 22
 - Zugriff steuern 9
 - Anmeldung (getty) 27
 - Anmeldung über Befehlszeile 6
 - Anmeldung über Befehlszeile (getty) 7
 - Anwendungen
 - app_root Verzeichnis 66
 - bei Anmeldung starten 28, 35
 - dem Anwendungsmanager hinzufügen 48
 - Desktop-fähig 50
 - entfernen 54
 - erfassen von der Session-Steuerung 31
 - erforderliche Aktionen 70
 - erforderliche Datentypen 70
 - Funktionen der
 - Anwendungsregistrierung 61
 - hinzufügen 48
 - im Anwendungsmanager zusammenstellen 45
 - neu laden 55
 - ohne Registrierungs paket hinzufügen 49
 - Registrierung 48
 - Stammverzeichnis 66
 - Symbole erstellen 146
 - zusammenstellen 47
 - Anwendungen Erneut Laden Aktion 55
 - Anwendungsfenster, Symbol zuordnen 229
 - Anwendungsgruppe benennen 53
 - Anwendungsgruppen
 - Aktionen 79
 - anpassen 52
 - benennen 52
 - Datentypen 79
 - Definition 44
 - erstellen im Registrierungs paket 76
 - Inhalt 80
 - integriert mit dtappintegrate 85
 - Namen 77
 - persönlich 52
 - Readme-Dateien 81
 - Standard 46
 - Symbole für 78
 - systemweit 51
 - verwalten 51
 - zusammenstellen 45
 - Anwendungsmanager
 - aktualisieren 55
 - allgemeine Verwaltung 56
 - Allgemeine Verwaltung des 54
 - Anwendungen erfassen 31
 - Anwendungen hinzufügen 48
 - Anwendungen integrieren 61
 - Anwendungen zusammenstellen 45

-
- Beschreibung 43
 - symbolische Verbindungen 45
 - Verzeichnisposition 44
 - Vorrangsregeln 46
 - Anwendungspaket, siehe
 - Registrierungspaket
 - Anwendungsregistrierung, siehe
 - Registrierung
 - Anwendungsressourcen, siehe
 - Ressourcen
 - Anwendungs-Server
 - Client 113
 - Client konfigurieren 113
 - Definition 97
 - konfigurieren 105, 113
 - verwalten 111
 - Anwendungs-Servers
 - Verfügbarkeit von Anwendungen 63
 - Anwendungsstammverzeichnis 66
 - Anwendungssuchpfad
 - ändern 53
 - Anwendungen suchen und
 - zusammenstellen 45
 - Anwendungs-Server hinzufügen 53
 - assemblieren 132
 - Definition 131
 - persönlich 54
 - Position 140
 - Rangfolge ändern 133
 - Standard 53, 131
 - Syntax 132
 - systemweit 53
 - Umgebungsvariable 131
 - Anwendungssymbol 177
 - doppelklicken 149
 - erstellen 147, 180
 - übergebene Dateien 149
 - Verwendung von Aktion erstellen 158
 - Anzeigetyp foreign 6
 - Anzeigetyp local 6
 - app_root, siehe
 - Anwendungsstammverzeichnis
 - Application Manager
 - general administration 54
 - precedence rules 46
 - applications
 - registering, See registration
 - Arbeitsbereiche
 - anpassen 269
 - Hintergrundmuster, siehe
 - Hintergrundmuster
 - Namen 269
 - Arbeitsbereichsmanager
 - andere Dateien einfügen 268
 - Bedienfeld verwalten 266
 - Definition 265
 - Funktionen 272
 - Konfigurationsdateien 266
 - Menüs 271
 - neu starten 268
 - persönliche Anpassung 267
 - starten 28, 35
 - systemweite Anpassung 267
 - wechseln zu OSF/Motif 279
 - Arbeitsbereichsmenü
 - ändern 272
 - Definition 271
 - erstellen 273
 - Arbeitsbereichsumschalter
 - Definition 239
 - Syntax der Definition 241
 - Arg_1 Syntax 149
 - ARG_CLASS Feld 184
 - ARG_COUNT Feld 184
 - ARG_COUNT field 196
 - ARG_MODE Feld 184
 - Arg_n Syntax 187
 - ARG_TYPE Feld 184, 195, 212
 - drucken 126
 - Argumente
 - anfordern 188
 - Anzahl für Aktionen 196
 - austauschbar für Aktion 192
 - einschränken für Aktionen 195
 - für Aktionen 186
 - mehrere für Aktionen 190
 - nicht austauschbar, für Aktionen 191
 - Nicht-Datei 189

-
- von Aktionen 149
 - auf Name basierende Datentypen 214
 - auf Pfad basierender Datentyp 215
 - auf Position basierender Datentyp 215
 - ausführbare Datei, Dateitypkriterien 217
 - Ausführender Host
 - Aktionen erstellen 198
 - Ausführender Hostost
 - angegeben vom EXEC_HOST Feld 198
 - Ausführungszeile 185
 - absoluter Pfad 187
 - allgemeine Merkmale 186
 - Argumente mit mehreren Dateien 190
 - ausführbare Datei angeben 187
 - Datei anfordern 188
 - ohne Argumente 187
 - Shellsyntax 186
 - übergebene Dateien 187
 - Zeichenfolge anfordern 189
 - Ausgabevariable 129
 - Ausgewählt Menü 145, 152, 153
 - authorize, Ressource 17
 - Automounter 107
- B**
- background, Ressource 294
 - Basisdateiname 182, 211
 - Bedienelement
 - ändern 244
 - Animation 258
 - bezeichnen 262
 - Client 257
 - Datei überwachen 255
 - definieren 251
 - der Hauptanzeige hinzufügen 242
 - Erscheinungsbild 246
 - ersetzen 245
 - erstellen 252
 - persönliche Anpassungen
 - ausschließen 237
 - Positionen von Bedienelementen
 - austauschen 245
 - Symbol 246
 - Typen 252
 - Übergabebereich 255
 - Umschalten 256
 - von Bedienfeld entfernen 243
 - wiederherstellen 237
 - zum Öffnen einer Datei 254
 - zum Umschalten 256
 - Bedienelementverhalten 252
 - Bedienfeld 262
 - ändern 242
 - Animation 258
 - Anpassung 233
 - Bedienelement , siehe Bedienelement
 - Bedienelement als Übergabebereich 255
 - Client 257
 - dynamische Anpassungen 235, 236
 - Komponenten 238
 - Konfigurationsdateien 234
 - Menü 271
 - Namenskonvention für Dateien 234
 - neu erstellen 263
 - Organisation der Definition 238
 - persönliche Anpassungen 237
 - Position 262
 - Registrierungspaket 82
 - Suchpfad 234
 - Symbol anzeigen 228
 - Syntax 239
 - verwaltet vom
 - Arbeitsbereichsmanager 266
 - Verwendung von Aktionen 144
 - Vorrangregeln beim Assemblieren 235
 - Bedienfeld wiederherstellen, Aktion 237
 - Bedientafel
 - ändern 247
 - automatisches Schließen ändern 250
 - Container 240
 - Definition 240
 - erstellen 247
 - gelöschte Bedientafel
 - wiederherstellen 237
 - integrierte Bedientafeln anpassen 248
 - neu erstellen 248

-
- Syntax 240
 - systemweite Anpassung 248
 - Zuordnung mit Hauptanzeige 248
 - Befehlszeile für Aktionen 185
 - Begrüßungsnachricht
 - ändern 14, 29
 - anzeigen 27, 29
 - Standard 14
 - Benutzer, für Aktion ändern 201
 - Benutzer-ID 101
 - Berechtigtmuster, angegeben mit
 - Aktion erstellen 167
 - Betriebsanzeige 252
 - Bezeichnungen
 - Aktionen 181, 203
 - Bedienelemente des Bedienfelds 262
 - Bezeichnungen für Bedienelemente 262
 - Bilder, siehe Symbole
 - Bildschirmfarben
 - maximale Anzahl 298
 - bildschirmspezifische Ressourcen 37
 - bildschirmspezifische Session 39
 - Bitmap-Bildschirm, Anmelde-Server ohne
 - lokalen Bildschirm 5
 - Bitmaps
 - siehe auch Symbole
 - Dateien finden 224
 - Namenskonventionen 225
 - Suchpfad 226
 - Wie Dateien gefunden werden 226
 - .bm Dateinamenerweiterung 224
 - bottomShadowColor, Ressource 294
 - BOX Definition 238
 - Syntax 239
 - BROADCAST, verwendet im indirekten
 - XDMCP-Modus 10
 - buttonBindings, Ressource 277
- C**
- CDE-MIN Dateien 106
 - CDE-TT Dateien 106
 - CHOOSER Zeichenfolge
 - Definition 10
 - Identifikation 10
 - CLIENT_GEOMETRY Feld 257
 - CLIENT_NAME Feld 256, 257
 - Client-Fenster im Bedienfeld 252
 - Clients
 - Definition 96
 - eines Server konfigurieren 104
 - Fenster in Bedienfeld 257
 - Client-Server Konfiguration, siehe
 - Netzwerk
 - colorUse, Ressource 34, 297, 299
 - COMMAND Aktion 174
 - Ausführungszeile 185
 - Beispiel 177
 - benötigte Felder 185
 - configuration files
 - data types 207
 - in registration package 62
 - CONTAINER_NAME Feld 240
 - CONTAINER_NAME, Feld 239, 240, 241, 245
 - CONTAINER_TYPE Feld 240
 - CONTENT, Feld 214, 218
 - CONTROL Definition, Syntax 240
 - CONTROL_BEHAVIOR, Feld 262
 - COPY_TO_ACTION, Feld 213
 - current.old, Verzeichnis 40
- D**
- DATA_ATTRIBUTES
 - definieren 210
 - Definition 206
 - Syntax 207
 - DATA_CRITERIA
 - definieren 213
 - Definition 206
 - mehrere 219
 - paarig mit DATA_ATTRIBUTES 206
 - Syntax 208
 - database search path
 - See also search path

- Datei, Datentypkriterien 217
- Dateiargument
 - angegeben mit Aktion erstellen 163
 - verwendet in Aktionen 149
- Dateieingabe, angegeben mit Aktion
 - erstellen 164
- Dateien
 - anhängen 102
 - auf der Grundlage des Datentyps
 - verstecken 212
 - Einhängepunkt 107
 - ferne Daten 106
 - ferner Zugriff 102
 - Namenkonsistenz 103
 - Netzwerk 105
 - Zugriff auf verteilte Dateisysteme 102
- Dateiinhalt 167
- Dateimanager, Symbole anzeigen 229
- Dateinamen zuordnen 107
- Dateinamen, Symbol 224
- Dateinamendatenbank-Server 106
- Dateinamenkonsistenz 103
- Datei-Server 97, 98
- Dateisteuerung 252
- Dateitypbedienelement 255
- Dateitypen
 - auf Name basierend 214
 - auf Pfad basierend 215
 - auf Position basierend 215
 - Hilfe 210
 - mehrere Kriterien 219
 - Moduskriterium 217
 - Positionen 220
 - Symbole 211
 - verdeckt 212
- Datenbank
 - Aktionen erneut laden 179
 - erneut laden 179
- Datenbank-Host 117, 198
- Datenbank-Host Schlüsselwort 198
- Datenbank-Server 117
 - Client 115
 - erstellen 115
 - konfigurieren 105, 114
- Datenbank-Suchpfad
 - Auswirkung auf EXEC_HOST 117
- Datenbanksuchpfad 134, 176
 - assemblieren 136
 - Auswirkung des
 - Anwendungssuchpfads 133
 - bezogen auf Anwendungssuchpfad
 - 135
 - Standard 134
 - Syntax 135
 - Umgebungsvariable 135
- Datentypen
 - Aktionen 152
 - Aktionen erneut laden 179
 - Aktionen zuordnen 211
 - Attribute 210
 - ausführbar 217
 - Bedingungen für manuelle Erstellung
 - 159
 - Beispiel 209
 - Berechtigungs muster 167, 168
 - Datendateien 62
 - definieren 207
 - differenzieren 213
 - drucken 155
 - Einführung 141, 150
 - einschränkende Aktionen 195
 - Einschränkungen 159
 - erforderlich für Registrierung 70
 - Erfordernisse für manuelle Erstellung
 - 206
 - erneut starten 179
 - Inhalt 167, 218
 - Konfigurationsdateien 207
 - Kriterien 213
 - Kriterien definieren 213
 - manuell erstellen 205
 - mit Aktion erstellt 150, 164
 - mit dtappintegrate integrieren 84
 - Name 167
 - schreibgeschützt 217
 - Server 114
 - Suchpfad, siehe Datenbanksuchpfad
 - Symbole 75, 171

Symbole zuordnen 227
 Variable in Definition 199
 Verhalten bei Doppelklicken 154
 Datentypenliste in Aktion erstellen 165
 DELETE, Feld 237
 DESCRIPTION, Feld 181, 210
 Desktop_Tools Anwendungsgruppe
 ändern 52
 Desktop-fähige Anwendung 50
 Desktop-Suchpfade 27
 DISPLAY Variable, gesetzt vom
 Anmeldemanager 20
 DISPLAY_displayname Makro 37
 DisplayHost, Schlüsselwort 198
 DROP_ACTION Feld 255
 DROP_ANIMATION Feld 259
 Drucken
 Datentypen 126
 konfigurieren für Datentyp 155
 Konzepte 126
 Standard 125
 Test 103
 Verwaltung 121
 Drucker
 Bezeichnungen 124
 Einheitennamen 103
 ferner Zugriff 103
 Hilfe 125
 hinzufügen 122
 Jobaktualisierungsintervall 123
 löschen 122
 Standard 125
 Symbolabbilder 123
 Druckjob-Aktualisierungsintervall 123
 Druckmanager 122
 Jobaktualisierungsintervall 123
 dt Dateien 176
 dtaction
 Benutzer ändern 201
 Syntax 201
 dtappgather Desktop-Dienstprogramm 45
 dtappgather Variable 31
 dtappintegrate 83
 Anwendung entfernen 54
 Funktion 84
 Syntax 84
 DTAPPSEARCHPATH Variable
 assemblieren 131, 132
 Definition 129
 DtButtonBindings 275
 dtchooser Datei 23
 dtconfig Befehl 11
 dtCreate, siehe Aktion erstellen
 DTDATABASESEARCHPATH Variable
 assemblieren 135, 136
 Definition 129
 Verwendung 183
 Dterror Datei 11
 dtgreet Datei 23
 DTHELPPSEARCHPATH Variable
 assemblieren 140
 Definition 129
 Dtlogin*language Ressource 18
 dtlogin, siehe Anmelde-Manager
 DTMOUNTPOINT Variable
 Prozesse 107, 108
 Wert setzen 107
 Dtpid Datei 4
 .dtprofile Datei
 LANG einstellen 306
 erstellen 28
 Umgebungsvariable setzen 36
 Verwendung als Ausgangsobjekt 28,
 27
 Syntax 37
 DtRoot Menü 273
 dtsearchpath 183
 dtsearchpath Variable 30, 129
 dtsmcmd Befehl 38
 DTSOURCEPROFILE Variable 31
 dtspcd 106, 108, 109
 Identifikationsüberprüfungsverzeich-
 nis 109
 Identifikationsverzeichnis 102
 konfigurieren 109
 DTSPSYSAPPHOSTS Variable 113

- ändern 53
- Definition 129
- Syntax 132
- DTSPSYSDATABASEHOSTS Variable
 - Auswirkung auf EXEC_HOST 117
 - Definition 129, 135
 - Syntax 135
- DTSPSYSHelp Variable 139
 - Definition 129
 - Syntax 139
- DTSPSYSICON Variable
 - Definition 129
 - Syntax 137
- DTSPUSERAPPHOSTS Variable 113
 - ändern 54
 - Definition 129
 - Syntax 132
- DTSPUSERDATABASEHOSTS Variable
 - Definition 129, 135
 - Syntax 135
- DTSPUSERHELP Variable
 - Definition 129
 - Syntax 139
- DTSPUSERICON Variable
 - Definition 129
 - Syntax 137
- dtstart_appgather Variable 31
- dtstart_searchpath Variable 30
- dtstart_ttsession Variable 32
- dtwm.fp Datei 234
- dtwmfp.session Datei 236
- dtwmrc Datei 266
 - bearbeiten 267
- Dtwmrc editieren, Aktion 267
- dynamicColor Ressource 34, 297, 301

E

- Einfache Motif-Session 19
- Eingabemethode 311
- Eingabevariable 128
- Eingrenzen
 - Standardwerte 307

- Einhängpunkt für ferne Dateien 107
- Umgebungsvariable
 - in .dtprofile 28
- EXEC_HOST Feld 117, 198
 - Standardwert 117, 199
- EXEC_HOST field 198
- EXEC_HOST, siehe Execution Host
- EXEC_STRING Feld, siehe
 - Ausführungszeile
- Execution Host
 - angeben 117
 - konfigurieren 118

F

- Farbe
 - Farbskalen 293
 - maximale Anzahl 298
 - Schatten mit der Ressource
 - shadowPixmap erstellen 300
 - steuern mit der Ressource
 - dynamicColor 301
 - Verwendung mit Symbolen 231
- Farben
 - aktiver Fensterrahmen 295
 - Anwendungsfenster 295
 - benutzte Anzahl 297
 - Paletten 293
 - Standard 295
 - Texteingabebereiche 295
- Farbpaletten 293
- Farbressourcen ändern 66
- Farb-Server 28
 - Ressourcen 34
 - starten 34
- Farbskalen 293
 - Standard 295
- Fenstermenü
 - Definition 271
- Fenstersteuerung
 - ändern 35
- Fensterunterstützung für Aktionen 163
- Ferne Ausführung

Anwendungs-Server konfigurieren
112
von Aktionen 198
file types, See data types
FILE_NAME Feld 254, 256
Font Ressource 288
FontSet Ressource 288
foreground Ressource 294
foregroundColor Ressource 34, 300
fp_dynamic Verzeichnis 235

G

gemeinsam benutzte Dateien 102
GID 101
Gruppe suchen Dialogfenster 171
Gruppen von Anwendungen
benennen 52
Symbole für 78
Gruppen-ID 101

H

Hilfe
siehe auch Hilfeband, Hilfedateien
Help Developer's Kit 73
HELP_STRING Feld 259
HELP_TOPIC Feld 260
HELP_VOLUME Feld 260
Hilfe
Aktionsdatei 181
angegeben mit Aktion erstellen 163
Datentyp 210
Druckersymbole 125
partielle Integration 74
vollständige Integration 73
Hilfeband
Hilfezugriff 74
im Registrierungspaket 73
Integrationsstufen 73
Hilfedateien
im Registrierungspaket 73
integriert mit dtappintegrate 85

Hilfe-Server 99
Client 115
erstellen 115
konfigurieren 105, 114
Hilfesuchpfad 138
siehe auch Suchpfad
assemblieren 140
Auswirkung des
Anwendungssuchpfads 133
bezogen auf Anwendungssuchpfad
139
Standard 138
Syntax 139
Umgebungsvariable 139
Hintergrundmuster 266
Dateipositionen 270
Home -Session 26
HOME Variable 21
home.old Verzeichnis 40
Home-Session 26
Home-Verzeichnis
gemeinsame Nutzung 102
Netzwerk 102

I

icon Bedienelement 252
ICON Feld
erlaubte Werte 182
für Datentyp 211
gültige Werte 211, 212
im Bedienfeld 246
icon type Bedienelement 255
Identifikationsüberprüfungsverzeichnis
109
Identifikationsverzeichnis 102
identifizierende Merkmale
Dialogfenster 167
inactiveFrameColorId, Ressource 295
Include-Anweisungen in Dateien des
Arbeitsbereichsmanagers 268
-indirect Option 8
inetd.conf Datei 108

inetd.sec Datei 109

IS_TEXT Feld 213

K

Kalender-Dämon 111

keyBindings, Ressource 279

Konfigurations-Dateien

Anmeldemanager 23

Konfigurationsdateien

Aktion 176

Arbeitsbereichsmanager 266

Bedienfeld 234

Fenstersteuerung 266

im Registrierungspaket 76

Session-Steuerung 41

Konsole mit zeichenorientierter Anzeige 7

L

LABEL Aktionsfeld 181

LANG Variable 304

Auswirkung auf Datentypen 220

in .dtprofile 306

vom Anmelde-Manager eingestellt 20

LANG Variable verwalten 304

LINK_TO_ACTION Feld 213

localTerminal Ressource 194

LOCKED Feld 237

login Datei 27

.login Datei

nicht vom Anmelde-Manager gelesen
36

Verwendung als Ausgangsobjekt 31

Anmelde-Manager

siehe Anmeldeserver

LOGNAME Variable 21

LOW_COLOR 299

lp Befehl 103

lp Druck-Spooler 103

LPDEST Variable 125

M

mail type Bedienelement 256

mailx Befehl 104

MAP Aktionen

Definition 175

Maustastenbelegung

Syntax 275

MEDIA Feld 213

Mehrere Bildschirme

Anmelde-Manager 16

Menü Sprache, Ändern 18

Menüs

Arbeitsbereichsmanager 271

Verwendung von Aktionen 145

MIME_TYPE_MEDIA Feld 213

mkfontdir Befehl 307

MODE Feld

Syntax 217

MODE, Feld 214

monitor types Bedienelemente 256

MONITOR_TYPE Feld 255

MOVE_TO_ACTION Feld 213

N

Name des Datentypentextfelds 166

NAME_PATTERN Feld 214

Namensmuster von Datentypen 167

Netzwerk

siehe auch Server

allgemeine Schritte für die
Konfiguration 100

Basiskonfiguration 100

Clients und Server konfigurieren 104

erforderliche Dateien 105

Servicetypen 96

Übersicht 96

NFS 102

NICHT Operator im MODE Feld 217

NLS Umgebungsvariable 306

NO_STDIO Fensterunterstützung 193

NoPrint Aktion 156

NUMBER_OF_ROWS Feld 260

NumFonts, Ressource 289

O

ODER Operator im MODE Feld 217

Öffnen Aktion 153

P

Paletten 293

PANEL Definition

Syntax 239

PANEL definition 238

PANEL_GEOMETRY Feld 262

PATH Variable 187

eingestellt vom Anmelde-Manager 20

PATH_PATTERN Feld

Syntax 215

PATH_PATTERN field 214

PERM_TERMINAL Fensterunterstützung

193

Persönliche Aktion und Datentypen

erstellen 209

Persönliche Anwendungsgruppe 52

Persönliche Anwendungsgruppen 54

Persönliche Gruppe von Anwendungen

52

Pfad

Benutzer 21

System 21

Pixmaps

Dateien finden 224

Namenskonventionen 225

Suchpfad 226

Platzhalterzeichen in Datentypen 214

.pm Dateinamenerweiterung 224

Position

Aktionen 202

POSITION_HINTS Feld 245

Positionen

Aktionsbezeichnung 203

Datentyp 220

primaryColorSetId Ressource 295, 297

profile Datei 27

.profile Datei

nicht vom Anmelde-Manager gelesen

36

Verwendung als Ausgangsobjekt 31

PUSH_ACTION Feld 253

PUSH_ANIMATION Feld 259

Q

-query Option 7

R

readme files 81

registration

See also registration package

modifying resources 65

registration package

See also registration

application icon 80

Registrierung

allgemeine Schritte 64

Anwendungsgruppe 76

Anwendungsstammverzeichnis 66

Beispiel 86–93

benötigte Datentypen 70

Definition 48, 63

dtappintegrate 83

erforderliche Aktionen 70

Farbe ändern 66

Funktionen 61

Hilfdateien 73

Ressourcen ändern 65

Schriftart ändern 65

Symbole 74

Übersicht 60

Zweck 62

Registrierungspaket 49

Bedienfeld steuern 82

Definition 63

erstellen 86

Inhalt der Anwendungsgruppe 80

Readme-Dateien 81

-
- Verzeichnisse 67
 - Zweck 61
 - Reload Actions, Aktion 180
 - RESOURCE_MANAGER Eigenschaften 33, 37
 - Ressourcen
 - bildschirmspezifisch 37
 - colorUse 299
 - einstellen 282
 - foregroundColor 300
 - laden 32
 - Session 28
 - setzen 37
 - shadowPixmap 300
 - sprachenabhängig 307
 - Standarddesktop 33
 - Standardwerte 282
 - RGB Farbwert 294
 - rgb.txt Datei 295
 - root Menü, siehe Arbeitsbereichsmenü
 - rpc.cmsd 111
 - rpc.ttdbserver 106, 108
- S**
- Schreibgeschützte Datentypkriterien 217
 - Schriftarten
 - Anzahl im Umgebungsmanager 289
 - Benutzer, im Umgebungsmanager 289
 - Bitmap 288
 - Eigenschaftszeichenfolge angeben 291
 - finden mit Aliasdatei 307
 - finden mit dem Befehl mkfontdir 307
 - finden mit Verzeichnisdatei 307
 - Hauptverzeichnis 307
 - Ressourcen für Registrierung ändern 65
 - System, im Umgebungsmanager 289
 - X Logical Font Description 290
 - sdl Dateien 74
 - secondaryColorSetId Ressource 295, 297
 - selectColor Ressource 294
 - sendmail, Funktion 104
 - Server
 - Aktionen 114
 - Anmeldung 105
 - Anwendung 97, 105
 - Datei 97
 - Datenbank 105
 - Datentypen 114
 - Definition 96
 - Hilfe 99, 105
 - konfigurieren 104
 - Session 105
 - Server
 - Anmeldung 97
 - Symbol 100, 105
 - Typen 99
 - Session
 - aktuell 26
 - Befehle bei Abmeldung durchführen 40
 - Befehle beim Start ausführen 39
 - bildschirmspezifisch 27, 39
 - Definition 25
 - einfache Motif-Session 19
 - erste Anmeldung 26
 - Home-Session 26
 - Probleme beim Start 41
 - Prozedur am Ende ausführen 20
 - Ressourcen 28
 - Sicherungskopie 40
 - Standard 26
 - starten 27
 - Startsession 38
 - wiederherstellen 40
 - Session-Steuerung
 - siehe auch Session
 - sessionetc Datei 39
 - sessionexit Datei 70
 - Session-Host Schlüsselwort 198
 - Sessionmanager
 - Suchpfade einrichten 30
 - sessions Verzeichnis 39
 - Session-Server, siehe Anmeldung Server
 - Session-Steuerung

Anwendungen erfassen 31
 Anwendungen starten 35
 Arbeitsbereichsmanager starten 35
 Befehle bei der Abmeldung ausführen
 40
 Befehle beim Start ausführen 39
 Begrüßungsnachricht 27
 Client 32
 Dateien 41
 Einführung 25
 Probleme 41
 Ressourcen laden 32
 Sicherungskopie 40
 starten 27
 systemweite Anpassung 29
 Verzeichnisse 41
 shadowPixmaps Ressource 34, 300
 Shell
 in einer Aktion 190
 persönliche Anpassung 28
 Syntax in Ausführungszeile 186
 systemweite Anpassung 28
 Verwendung von .profile oder .login
 als Ausgangsprojekte 31
 siehe auch Umgebungsvariable
 SHELL Variable, eingestellt vom
 Anmelde-Manager 20
 Shells
 in Aktionen 190
 SPC 109
 Sicherheit 109
 Standarddateien
 Desktopanwendungen 282
 Standardwert
 sprachenabhängig 307
 startlog Datei 41
 Steuerungsdaemon für Unterprozesse,
 siehe dtspcd
 SUBPANEL Definition 238
 Suchpfad
 Anwendungen 45
 eingestellt von der Session-Steuerung
 30
 Suchpfade

 Aktionen 176
 aktueller Wert 129
 Anwendungen 131
 Ausgabevariable 129
 Bedienfelddefinitionen 234
 Desktop 128
 Eingabevariable 128
 Hilfe 138
 Position 140
 Symbole 226
 Umgebungsvariable 129
 Wert setzen 129
 SWITCH Definition 241
 Symbole
 Aktionen 171, 181
 Aktionssymbole 180
 Anwendungsgruppen 75
 Bedienfeld 228, 246
 Darstellung von Anwendungen 88
 Dateien finden 224
 Dateiformat 224
 Dateimanager zum Anzeigen
 verwenden 229
 Dateiname ohne Pfad 211
 Dateinamen 224
 Datentyp 75
 Datentypen 171, 211
 die Anwendungen repräsentieren 80
 Druckerabbilder 123
 einem Anwendungsfenster zuordnen
 229
 einer Aktion oder einem Datentyp
 zuordnen 227
 Empfehlungen zur Erstellung 230
 erforderlich für Registrierung 74
 Farbpalette 231
 für Anwendungen 146
 für Anwendungen erstellen 91
 Größenkonventionen 225
 Gruppe suchen Dialogfenster 171
 Gruppe von Anwendungen 78
 integriert mit dtappintegrate 85
 mit dem Dateimanager anzeigen 229
 Namenskonvention 224
 Server, siehe Symbolserver

-
- Suchpfad, siehe Symbolsuchpfad
 - Zuordnungen 226
 - Symbole mit dem Dateimanager anzeigen 229
 - Symbolgrößen 225
 - Symbolische Verbindung
 - erstellt bei Registrierung 84
 - Symbolische Verbindungen
 - Dateinamenkonsistenz 103
 - Datentypkriterien 214
 - Symbol-Server 100
 - Client 115
 - erstellen 115
 - Konfigurieren 105
 - Symbol-Serveri
 - konfigurieren 114
 - Symbolsuchpfad 136
 - assemblieren 138
 - Auswirkung des
 - Anwendungssuchpfads 133
 - bezogen auf Anwendungssuchpfad 137
 - Standard 136
 - Syntax 137
 - Umgebungsvariable 137
 - sys.dtpfile Datei 28
 - sys.dtwmrc Datei 266, 267
 - sys.resources Datei 33, 282
 - sys.resources file 33, 37
 - sys.session Datei 35, 38
 - systemPath Ressource 21
- T**
- Tastenbelegung
 - Syntax 277
 - TERMINAL Fensterunterstützung 194
 - Terminal-Emulator
 - ändern 55
 - automatisches Schließen 163
 - Befehlszeilenoptionen für Aktionen 194
 - für Aktionen 193
 - manuelles Schließen 163
 - Standard für Aktionen 194
 - Text -Editor, ändern 55
 - textColorSetId Ressource 295
 - timeZone Ressource 22
 - title Ressource 269
 - ToolTalk
 - Aktion, siehe TT_MSG Aktion
 - Anwendungen, Aktionen 203
 - Datenbank-Server, siehe
 - rpc.ttdbserver
 - Nachrichtendämon 27, 32
 - ToolTalk Meldungs-Server, siehe ttsession
 - topShadowColor Ressource 294
 - TT_MSG Aktion
 - erstellen 203
 - Schlüsselwörter 204
 - ttsession 111
 - starten 32
 - TYPE Feld 252
 - TZ Variable 20
- U**
- Übergabebereich
 - Aktionssymbol 187
 - Bedienfeldsteuerung 255
 - Übergebene Datei, Aktion akzeptiert 187
 - Übertragungsprotokollprogramm 37
 - Uhr 252
 - UID 101
 - Umgebungsmanager
 - Farben angeben 294
 - Farbintegration 66
 - Schriftartintegration 65
 - Umgebungsvariable
 - Aktionsdefinitionen 200
 - Anmeldemanager 21
 - Anwendungssuchpfad 131
 - Bitmapsuchpfad 137
 - Datenbanksuchpfad 135
 - Einstellung 36
 - ferne Ausführung 110
 - Hilfesuchpfad 139

in .dtpfile 28
 persönlich 36
 Pixmapsuchpfad 137
 setzen 36
 Standard 28
 Suchpfad 129
 Symbolsuchpfad 137
 systemweit 36
 Verwendung von .login or .profile als
 Ausgangsobjekte 31
 UND Operator im MODE Feld 217
 UNIX Tastenbelegung 281
 Unterprozeßsteuerungs-Service, siehe
 SPC
 USER Variable 21
 userPath, Ressource 21
 user-prefs.dt Datei 57

V

Variable
 in Aktionsdefinitionen 199
 Umgebung, siehe
 Umgebungsvariable
 Verbindung, Datentypkriterien 217
 Verzeichnis
 Datentypkriterien 217
 Vorrang
 Aktionsdatenbankassemblierung 184
 Vorrangregeln
 Bedienfeld assemblieren 235

W

WINDOW_TYPE Feld 193
 windowMenu Ressource 271
 WM_CLASS Eigenschaft 229
 wmStartupCommand Ressource 35
 workspaceCount Ressource 269
 writeXrdbColors Ressource 34

X

X -Berechtigung 104
 X Ressourcen, siehe Ressourcen
 X -Server
 Zugriffsrechte ändern 16
 X station
 Bildschirme ohne XDMCP 8
 Xaccess Liste 9
 XDMCP-direkt 7
 X -Stationen 105
 X Terminals
 XDMCP-Direktmodus 9
 X terminals
 CHOOSER string 10
 possible login server configurations 3
 XDMCP-indirect 8
 XDMCP-indirekter Modus 10
 X400_TYPE Feld 213
 Xaccess Datei 9
 XAUTHORITY Variable, eingestellt vom
 Anmelde-Manager 20
 X-Berechtigung 104
 Xconfig Datei
 ändern 4
 Ressourcen setzen 15
 .Xdefaults Datei 282, 33
 XDMCP 2
 Definition 7
 Direktanforderung 7
 Direktanforderung von Bildschirmen
 im Netz 7
 Direktmodus 9
 indirekte Anforderungen 8, 23
 indirekter Modus 10
 Xerrors Datei 11
 Xfailsafe Datei 19, 21, 23
 xlsfonts Befehl 289
 XMICONBMSEARCHPATH Variable
 assemblieren 138
 Definition 129
 Verwendung 137
 XMICONSEARCHPATH Variable
 assemblieren 138
 Verwendung 137
 XMICONSEARCHPATH variable

- definition 129
- XmText*FontList Ressource 288
- XmTextField*FontList Ressource 288
- X-Netzwerk
 - X -Berechtigung 104
- Xpid Datei 3
- Xreset Datei 20
- Xresources Datei 13, 14
- Xservers Datei
 - lokale Bildschirmverwaltung 22
 - Server starten 4
 - Standard 5
 - Syntax 4
- Xsession Datei 28
 - ausgeführt vom Anmelde-Server 19
 - Pfad setzen 21
 - Session-Steuerung starten 27
 - systemweite Anpassung 29
- Xsession.d directory 36
- Xsession.d Verzeichnis 27
 - anpassen 29
 - Prozeduren 28
- Xsetup Datei 18
- Xstartup Datei 19
- X-Stations
 - Session-Services 98
- XUSERFILESEARCHPATH Variable 307

Z

- Zeichenfolge-Aktionsargument 189
- Zeichenfolgevariable in
 - Aktionsdefinitionen 199
- Zuordnungsaktionen 153
 - Beispiel 178