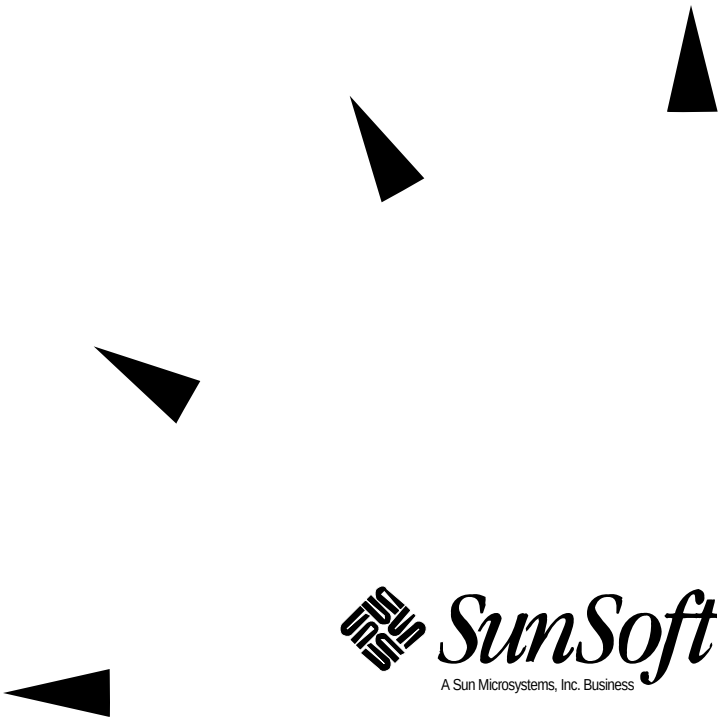
Common Desktop Environment: Användarhandbok för systemadministratörer



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

SunSoft
A Sun Microsystems, Inc. Business

Artikelnummer: 802-4176-11
Utgåva A, juni 1996

De nombreuses appellations utilisées par les constructeurs et les distributeurs sont des marques déposées. Dans la présente publication, les termes en majuscules ou commençant par une majuscule peuvent être des marques déposées.

LA PRESENTE PUBLICATION EST FOURNIE "DANS L'ETAT". LES SOCIETES DETENTRICES DU COPYRIGHT DECLINENT TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE ET A L'APTITUDE A UNE UTILISATION PARTICULIERE. PAR AILLEURS, CES SOCIETES NE SAURAIENT ETRE TENUES POUR RESPONSABLES D'UN DOMMAGE RESULTANT DE L'UTILISATION DES INFORMATIONS DU PRESENT DOCUMENT.

Le code et la documentation des widgets DtComboBox et DtSpinBox ont été fournis par Interleaf, Inc. Copyright 1993, Interleaf, Inc.

Copyright © 1993, 1994, 1995, 1996 Hewlett-Packard Company
Copyright © 1993, 1994, 1995, 1996 International Business Machines Corp.
Copyright © 1993, 1994, 1995, 1996 Sun Microsystems, Inc.
Copyright © 1993, 1994, 1995, 1996 Novell, Inc.

Tous droits réservés. Ce produit et sa documentation sont protégés par copyright et des licences en restreignent la distribution, l'utilisation, la copie et la décompilation. Aucune partie de ce produit ou de sa documentation ne peut être reproduite, sous quelque forme et par quelque moyen que ce soit, sans autorisation écrite préalable.

UNIX est une marque dont la licence est détenue exclusivement par

X / Open Company, Ltd.

OSF / Motif et Motif sont des marques d'Open Software Foundation, Ltd.

X Window System est une marque de X Consortium, Inc.

PostScript est une marque d'Adobe Systems, Inc., qui peut être enregistrée dans certains pays.

Innehållsförteckning

Förord xvii

1. Konfigurera inloggningshanteraren 1

Starta inloggningsservern 2

Hantera lokala skärmar och nätverksskärmar 3

Söka efter inloggningsserverns process-ID 3

Visa en inloggningsbild på en lokal skärm 4

Köra inloggningsservern utan lokal skärm 5

Kommandoradsinloggning på lokala skärmar 5

Anpassa textskärmen 6

Visa en inloggningsbild på en nätverksskärm 6

Kontrollera tillgång till inloggningsservern 8

Felsökning 10

Avaktivera inloggningsservern 10

Inloggningsbilden 11

Ändra hur inloggningsbilden ser ut 12

Ändra hur inloggningsbilden fungerar skärmvis 15
Ändra X-serverbehörighet 15
Starta kommandon innan inloggningsbilden visas 17
Starta en session med endast ett fönster 17
När användarsessionen avslutas 18
Inloggningsserverns miljö 18
Ändra användar- och systemsökväg 19
Administrera inloggningshanteraren 20
Inloggningshanterarens filer 21
2. Konfigurera sessionshanteraren 23
Vad är en session? 23
Startsession 24
Aktiv session 24
Hemsession 24
Skärmspecifika sessioner 24
Starta en session 25
När sessionen startar 25
Läsa in och köra skriptet .dtprofile 26
Läsa in och köra skriptet Xsession.d 26
Visa välkomstmeddelandet 27
Ange skrivbordssökvägar 27
Samla ihop tillgängliga program 28
Valfritt läsa in och köra skripten .profile och .logint 29
Starta ToolTalk 29

Starta sessionshanterarklienten 30
Ladda sessionsresurser 30
Starta färgservern. 31
Starta arbetsytehanteraren 32
Starta sessionsprogram 32
Ytterligare anpassningar av sessionsstarten. 32
Exekvera ytterligare kommandon vid sessionsstart och utloggning 36
Filer och kataloger i sessionshanteraren 37
3. Lägga till och administrera program 39
Programhanterarens struktur. 40
Katalogplacering i programhanteraren. 40
Hur programhanteraren hittar och samlar program. 40
Prioritetsregler vid insamling av program 41
Programgrupper på standardskrivbordet 41
Exempel på hur programgrupper samlas in 42
Lägga till program i programhanteraren 43
Olika sätt att lägga till program i programhanteraren 43
Skapa och administrera allmänna programgrupper 45
Ändra sökvägen som används för att söka efter program. . . . 47
Standardsökvägen 47
Lägga till en programserver till programsökvägen. 47
Allmän programhantering 48
Ändra textredigeraren och terminalemulatorn 49

4. Registrera program 53

Översikt över programregistrering 54

Funktioner som tillförs genom programregistrering 54

Syftet med programregistrering. 56

Allmänna steg för att registrera program 57

Steg 1: Ändra font- och färgresurser 58

Ändra fontresurser 58

Ändra färgresurser 59

Steg 2: Skapa skrivbordets rot 59

Steg 3: Skapa kataloger för registreringspaket 60

Steg 4: Skapa funktioner och datatyper för programmet 62

Funktioner och datatyper som är nödvändiga för ett program. 63

Adresser för konfigurationsfiler till funktions- och datatypsdefinitioner 63

Olika sätt att skapa funktioner och datatyper 64

Steg 5: Placera hjälpfiler i registreringspaket 66

Steg 6: Skapa programikoner 67

Ikoner som behövs för skrivbordet 67

Steg 7: Skapa programgruppen 68

Skapa programgruppskatalogen 69

Programgruppsnamn 69

Konfigurera programgruppen att använda en unik ikon . . 70

Skapa innehållet i programgruppen 72

Steg 8: Registrera programmet med dtappintegrate 75

Hur dtappintegrate integrerar program 76
Exempel på hur du skapar ett registreringspaket 77
5. Konfigurera skrivbordet i nätverk 85
Översikt över skrivbordet i nätverk 85
Typer av nätverkstjänster 86
Typiska nätverkssituationer 86
Andra nätverkssituationer 88
Sammanfattning — typer av tjänster. 89
Allmänna steg för att konfigurera nätverket 89
Konfigurera nätverkets basoperativsystem för skrivbordet. . . 90
Inloggningskonto för användare 90
Konfigurera distribuerad filsystemåtkomst 91
Konfigurera åtkomst av fjärrskrivare 92
Konfigurera elektronisk post 92
Konfigurera X-verifiering 92
Konfigurera skrivbordsklienter- och servrar 93
Konfigurera inloggnings- och sessionstjänster. 93
Konfigurera andra programrelaterade tjänster 93
Administrera programtjänster 98
Miljövariabler för sökvägar 99
Konfigurera en programserver och dess klienter 99
Konfigurera databas-, ikon- och hjälptjänster 101
Speciella konfigurationer för nätverksprogram. 102

6. Konfigurera och administrera utskrifter från skrivbordet. . . 107
Lägga till och ta bort skrivare. 108
Ändra intervall för jobbuppdatering. 109
Ikonbilder till skrivaren 109
Ikonfilnamn- och storlekar 110
Konfigurera standardskrivare 110
Vad som sker vid utskrift 112
7. Skrivbordets sökvägar 113
Skrivbordets sökvägar och deras miljövariabler 114
Ange värde för en sökväg. 115
Programsökväg 116
Standardprogramsökvägen 117
Miljövariabler för programsökvägar. 117
Syntax för indatavariabler till programsökvägen 117
Hur programsökvägens värde sätts samman. 118
Ändra prioritetsordning på den lokala systemtäckande adressen 118
Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp. 119
Databassökväg (funktion / datatyper) 120
Databasens standardsökväg. 120
Hur programsökvägen påverkar databassökvägen 120
Databassökvägens miljövariabler 120
Syntax för databassökvägens indatavariabler 121
Hur databassökvägen sätts samman. 121

Ikonsökväg. 121
Standardikonsökväg 122
Hur programsökvägen påverkar ikonsökvägen 122
Ikonsökvägens miljövariabler 122
Syntax för ikonsökvägens indatavariabler 122
Hur ikonsökvägen sätts samman. 123
Hjälpsökväg. 123
Standardhjälpsökväg. 123
Hur programsökvägen påverkar hjälpsökvägen. 124
Hjälpsökvägens miljövariabler 124
Syntax för hjälpsökvägens indatavariabler 124
Hur hjälpsökvägen sätts samman 124
Lokalanpassade sökvägar. 125
8. Introduktion till funktioner och datatyper. 127
Introduktion till funktioner 128
Hur funktioner skapar programikoner. 131
Hur funktioner använder datafiler som argument 133
Ytterligare användningsområden för funktioner 134
Introduktion till datatyper 135
Vad är en datatyp?. 135
Hur datatyper ansluter datafiler till funktioner. 136
Skriva ut datatyper från skrivbordet 139
9. Skapa funktioner och datatyper med Skapa funktion. 141

Vad Skapa funktion gör. 141
Begränsningar för Skapa funktion 142
Funktionsbegränsningar 142
Datatypsbegränsningar 143
Skapa en funktion och datatyp för ett program med Skapa funktion 143
Ange ikon med dialogrutan Sök uppsättning 153
10. Skapa funktioner manuellt 155
Varför du måste skapa funktioner manuellt 156
COMMAND-funktioner 156
MAP-funktioner 157
TT_MSG-funktioner (ToolTalk-meddelande) 157
Skapa en funktion manuellt: Allmänna steg 157
Konfigurationsfiler för funktioner 157
Exempel på hur du skapar en COMMAND-funktion 159
Exempel på hur du skapar en MAP-funktion 160
Skapa en funktionsfil (ikon) för en funktion 162
Ange ikonbilden som används av en funktion 163
Prioritet i funktionsdefinitioner 165
Skapa en exekveringsträng för COMMAND-funktioner 167
Allmänna egenskaper för exekveringssträngar 167
Skapa funktioner som inte använder argument 168
Skapa funktioner som accepterar släppta filer 168
Skapa en funktion som frågar efter filargument 169

Skapa funktioner som accepterar eller frågar efter en släppt fil 169
Skapa funktioner som frågar efter icke-filargument 170
Tolka filargument som en sträng 170
Skalmöjligheter för funktioner 170
Skapa COMMAND-funktioner för flera filargument 171
Fönsterstöd och terminalemulatorer för COMMAND-funktioner 173
Ange fönsterstöd för funktionen 174
Ange kommandoradsalternativ för terminalemulatorn . . 174
Ange en annan standardterminalemulator 174
Begränsa funktioner till vissa argument 175
Begränsa funktioner till en viss datatyp 176
Begränsa funktioner baserat på antal argument 176
Begränsa funktioner baserat på argumentläge. 178
Skapa funktioner som kör program i fjärrsystem 178
Skapa en funktion som kör ett fjärrprogram 178
Använda variabler i funktions- och datatypsdefinitioner 179
Använda strängvariabler i en funktion. 179
Använda miljövariabler i funktioner och datatyper 180
Anropa funktioner från kommandoraden 181
Syntax till dtaction. 181
Skapa en funktion som kör en annan funktion 181
Skapa en funktion som körs som en annan användare. . . . 182
Skapa lokalanpassade funktioner 182

Adresser för lokalanpassade funktioner 182
Skapa funktioner för ToolTalk-program 183
11. Skapa datatyper manuellt 187
Varför du måste skapa datatyper manuellt 188
Komponenter i datatypsdefinitionen: Villkor och attribut 188
Skapa datatyper manuellt: Allmänna steg 189
Konfigurationsfiler för datatyper. 189
Exempel på hur du skapar en egen funktions- och datatyp 190
Definiera dataattribut för en datatyp 192
Ange ikonbild för datatypen 192
Associera datatyper med funktioner. 193
Dölja filer baserat på datatyp 194
Ange metoder när en fil manipuleras 194
Definiera datavillkor för en datatyp 195
Namnbaserade datatyper 195
Sökvägsbaserade datatyper 196
Datatyper baserade på namn och sökväg. 197
Använda fillägen som typvillkor. 198
Innehållsbaserad datatyp 199
Skapa lokalanpassade datatyper 201
Adresser för lokalanpassade datatyper. 201
12. Skapa skrivbordsikoner. 203
Ikonbildsfiler 203
Ikonfilformat. 204

Ikonfilnamn 204
Storlekskonventioner 205
Ikonsökvägen 206
Komma åt ikoner via nätverket 206
Associera ikoner 206
Ange ikonfiler 206
Rekommendation när du skapar ikoner 209
Hur färger används 210
13. Särskild anpassning av frontpanelen 211
Konfigurationsfiler för frontpanelen 212
Standardfrontpanelens konfigurationsfil 212
Sökvägen till frontpanelens konfigurationsfiler 212
Hur frontpanelen sätts samman: prioritetsregler 213
Skapa frontpanelsfiler dynamiskt 214
Administrera anpassning av användargränssnitt 214
Hur frontpanelsdefinitionen är organiserad 215
Komponenter på frontpanelen 216
Allmän syntax för frontpanelsdefinitionen 216
Ändra huvudpanelen 219
Ange vilken ikonbild som ska användas av kontrollen . . . 223
Skapa och ändra panelundermenyer 224
Anpassa inbyggda panelundermenyer 225
Definiera kontroller på frontpanelen 228

Definitioner till kontroller på frontpanelen 228
kontrolltyper. 229
Anpassa arbetsyteomkopplaren 236
Allmän konfiguration av frontpanelen 237
Allmänna steg 237
Exempel på hur du skapar en egen frontpanel med tre rader 239
14. Anpassa arbetsytehanteraren 241
Arbetsytehanterarens konfigurationsfiler. 242
Anpassa arbetsytor 244
Arbetsytehanterarens menyer 246
Arbetsytehanterarens menysyntax 247
Anpassa knappkopplingar 249
Syntax för knappkoppling 250
Anpassa tangentkopplingar 252
Skrivbordets standardtangentkopplingar. 252
Syntax för tangentkoppling 253
Växla mellan standardmetod och anpassad metod. 254
15. Administrera programresurser, fonter och färger 257
Ange programresurser 257
Hur skrivbordet startar resurser 258
Definiera UNIX-kopplingar 258
UNIX-kopplingar i filen i UNIXbindings 259
Administrera fonter. 263
Ange fontresurser för skrivbordet. 263

X Logical Font Description (XLFD) 265
Administrera färger 267
Färgpaletter 267
Färguppsättningar 268
Kontrollera färg med Arbetsinställningar 271
Antal färger som används i Arbetsinställningar 271
Ange skuggtjocklek för programfönster 275
16. Konfigurera lokalanpassade skrivbordssessioner 277
Hantera miljövariabeln LANG 278
Ange språk för flera användare 278
Ange språk för en session 279
Ange språk för en användare 279
Miljövariabeln LANG och sessionskonfiguration 280
Ange andra NLS-miljövariabler 280
Söka efter fonter 280
Lokalanpassa resursfilerna app-defaults 281
Lokalanpassa funktioner och datatyper 281
Lokalanpassa ikoner och bitmönster 282
Lokalanpassa bakgrundsnamn 282
Lokalanpassa palettnamn 283
Lokalanpassa hjälpvolymer 283
Lokalanpassa meddelandekataloger 284
Exekvera lokalanpassade fjärranslutna skrivbordsprogram . . 284

Återställa tangentbordstabellen 284

Förord

I den här handboken beskrivs hur Common Desktop Environment (CDE) ser ut och fungerar för avancerade användare. Boken tar bl a upp:

- Hur du anpassar systemstart, inloggning och sessionsstart
- Hur du lägger till program och hur dessa och data visas i det grafiska gränssnittet
- Hur du konfigurerar skrivbordsprocesser, program och data via nätverk
- Hur du anpassar skrivbordet, t ex fönsterhantering, utskrift, färger och fonter

Bokens målgrupp

Den här boken riktar sig till:

- Systemadministratörer. Många av uppgifterna i boken kräver rotbehörighet.
- Avancerade användare som vill anpassa sådant som inte kan uföras via skrivbordet. Skrivbordet tillhandahåller användarspecifika lägen för många av konfigurationsfilerna.

Hur boken är upplagd

Handboken innehåller följande kapitel:

Kapitel 1, ”Konfigurera inloggningshanteraren”, beskriver hur du konfigurerar hur skrivbordets inloggningshanterare ska se ut och fungera.

Kapitel 2, ”Konfigurera sessionshanteraren”, beskriver hur skrivbordet lagrar och hämtar sessioner, och hur du anpassar sessionsstarten.

Kapitel 3, ”Lägga till och administrera program”, beskriver hur programhanteraren samlar ihop program och hur du lägger till program.

Kapitel 4, ”Registrera program”, beskriver hur du skapar ett registreringspaket till ett program.

Kapitel 5, ”Konfigurera skrivbordet i nätverk”, beskriver hur du distribuerar skrivbordstjänster, program och data via nätverk.

Kapitel 6, ”Konfigurera och administrera utskrifter från skrivbordet”, beskriver hur du lägger till och tar bort skrivare och hur du anger standardskrivare för skrivbordet.

Kapitel 7, ”Skrivbordets sökvägar”, beskriver hur skrivbordet söker efter program, hjälpfiler, ikoner och andra skrivbordsdata via nätverk.

Kapitel 8, ”Introduktion till funktioner och datatyper”, beskriver principerna för funktioner och datatyper samt förklarar hur de används för att ge program ett användargränssnitt.

Kapitel 9, ”Skapa funktioner och datatyper med Skapa funktion”, beskriver hur du använder programmet Skapa funktion för att skapa funktioner och datatyper.

Kapitel 10, ”Skapa funktioner manuellt”, beskriver hur du skapar funktionsdefinitioner genom att redigera en databaskonfigurationsfil.

Kapitel 11, ”Skapa datatyper manuellt”, beskriver hur du skapar datatypsdefinitioner genom att redigera en databaskonfigurationsfil.

Kapitel 12, ”Skapa skrivbordsikoner”, beskriver hur du använder ikonredigeraren, namngivningskonventioner, storlek och sökväg för skrivbordsikoner.

Kapitel 13, ”Särskild anpassning av frontpanelen”, beskriver hur du skapar nya systemtäckande kontroller och panelundermenyer samt anpassar andra paneler.

Kapitel 14, ”Anpassa arbetsytehanteraren”, beskriver hur du anpassar fönster, kopplingar till musknappar och tangentbord samt menyer i arbetsytehanteraren.

Kapitel 15, ”Administrera programresurser, fonter och färger”, beskriver hur du anger programresurser och hur fonter och färger används av skrivbordet.

Kapitel 16, ”Konfigurera lokalanpassade skrivbordssessioner”, beskriver systemadministrationsuppgifter för system som kör internationella sessioner.

Typsnitt och symboler i boken

I tabellen nedan beskrivs hur olika typsnitt och symboler används i boken.

tabell P-1 Typografiska konventioner

Typsnitt eller symbol	Betyder	Exempel
`AaBbCc123`	Namn på kommandon, filer kataloger och nyckelord	Redigera `.dtprofile` . Använd `ls -a` för att lista alla filer.
AaBbCc123	Parametrar eller variabler som ska ersättas med ett riktigt värde.	Om du vill ta bort en fil skriver du `rm` *filnamn*`.`
AaBbCc123	Boktitlar, nya ord eller termer, eller andra viktiga ord	Läs kapitel 6 i *Användarhandboken*`.` Dessa kallas *klassalternativ.* Du *måste* ha rotbehörighet för att göra det här.

Konfigurera inloggningshanteraren 1

Inloggningshanteraren är en server som visar en inloggningsbild, verifierar användarna och startar användarens session. Den här grafiska inloggningen är enklare än den traditionella teckenbaserade inloggningen. Skärmarna som hanteras av inloggningsservern kan direkt anslutas till inloggningsservern eller till en X-terminal eller arbetsstation i nätverket.

Obs! – Du måste vara rotanvändare för att kunna starta, stoppa eller anpassa inloggningsservern.

Starta inloggningsservern	*2*
Hantera lokala skärmar och nätverksskärmar	*3*
Felsökning	*10*
Avaktivera inloggningsservern	*10*
Ändra hur inloggningsbilden ser ut	*12*
Administrera inloggningshanteraren	*20*
Inloggningshanterarens filer	*21*

1

Inloggningsservern:

- Kan visa en inloggningsbild på bitmönsterskärmar ovillkorligt, eller på begäran på lokala bitmönsterskärmar eller bitmönsterskärmar i nätverk
- Omfattar direkt anslutna textskärmar
- Kan visa en valskärm där användaren kan visa inloggningsbilder från andra inloggningsservrar i nätverket
- Kontrollerar vem som kan få tillgång till inloggningsservern
- Omfattar även traditionell teckenbaserad inloggning

Skärmar som hanteras av inloggningshanteraren kan anslutas direkt till inloggningshanterarservern eller till en X-terminal eller arbetsstation i nätverket. För lokala skärmar startar inloggningsservern automatiskt en X-server och visar en inloggningsbild. För nätverksskärmar, t ex X-terminaler, stöds X Display Manager Protocol (XDMCP) 1.0, med vilket skärmar kan begära att inloggningsservern visar en inloggningsbild.

Starta inloggningsservern

Inloggningsservern startas normalt när systemet startas. Du kan också starta inloggningsservern från en kommandorad.

- Om du vill att inloggningsservern ska starta när systemet startas, skriver du `/usr/dt/bin/dtconfig -e`

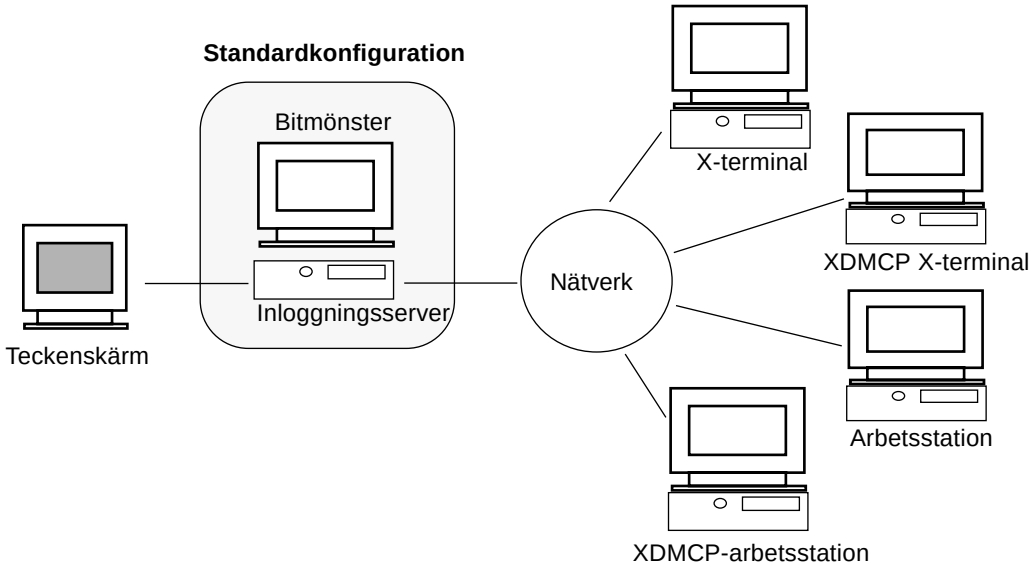
  Inloggningsservern startas automatiskt när du startar om.

- Om du vill starta inloggningsservern från kommandoraden, skriver du `/usr/dt/bin/dtlogin -daemon`

Obs! – Även om det är möjligt att starta inloggningsservern från kommandoraden när du ska testa konfigurationen, bör du normalt starta inloggningsservern i samband med att systemet startas.

Hantera lokala skärmar och nätverksskärmar

I figur 1-1 visas ett exempel på en konfiguration av inloggningsservern.



figur 1-1 Exempel på konfiguration av inloggningsserver

Söka efter inloggningsserverns process-ID

Normalt lagras inloggningsserverns process-ID i `/var/dt/Xpid`.

Om du vill ändra detta anger du resursen `Dtlogin.pidFile` i filen `Xconfig`. Om du ändrar denna måste den angivna katalogen finnas när inloggningsservern startas.

Om du vill ändra `Xconfig` kopierar du `Xconfig` från `/usr/dt/config` till `/etc/dt/config`. När du har ändrat `/etc/dt/config/Xconfig` uppmanar du inloggningsservern att läsa `Xconfig` genom att skriva:

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

Detta startar kommandot `kill -HUP` *inloggningsserverns process-ID*.

Om du t ex vill lagra inloggningsserverns process-ID i `/var/myservers/Dtpid` anger du följande i filen `Xconfig`:

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

När inloggningsservern startas om lagras dess process-ID i `/var/myservers/Dtpid`. Katalogen `/var/myservers` måste finnas när inloggningsservern startas.

Visa en inloggningsbild på en lokal skärm

När inloggningsservern startar kontrolleras filen `Xservers` för att avgöra om en X-server måste startas och om och hur inloggningsbilder ska visas på lokala skärmar och nätverksskärmar.

För att ändra `Xservers` kopierar du `Xservers` från `/usr/dt/config` till `/etc/dt/config`. När du har ändrat `/etc/dt/config/Xservers` uppmanar du inloggningsservern att läsa `Xservers` genom att skriva:

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

Detta startar kommandot `kill -HUP` *inloggningsserverns process-ID*

Formatet på raden för `Xservers` är:

skärmnamn skärmklass skärmtyp X-serverkommando

där

skärmnamn — Anger för inloggningsservern vilket anslutningsnamn som ska användas när den ansluter till X-servern (:0 i följande exempel). Värdet * (asterisk) utökas till *värddatornamn*:0. Talet som anges måste motsvara det som anges i anslutningsnumret för *X-serverkommando*.

skärmklass — Identifierar resurser som är specifika för den här skärmen (`Local` i följande exempel).

skärmtyp — Anger för inloggningsservern om skärmen är lokal eller är en nätverksskärm, och hur alternativet Kommandoradsinloggning på inloggningsbilden ska hanteras (`local@console` i följande exempel).

X-serverkommando Identifierar kommandoraden, anslutningsnummer och andra alternativ som inloggningsservern använder för att starta X-servern (`/usr/bin/X11/X: 0` i följande exempel). Angivet anslutningsnummer måste motsvara det som anges i *skärmnamn*.

Standardraden för `Xservers` ser ut ungefär så här:

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

Köra inloggningsservern utan lokal skärm

Om inloggningsserversystemet inte har någon bitmönsterskärm, kör du inloggningsservern utan lokal skärm genom att kommentera bort raden `Xservers` för lokal skärm med hjälp av tecknet #. Exempel:

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

När inloggningsservern startar körs den i bakgrunden och väntar på en förfrågan från nätverksskärmar.

Kommandoradsinloggning på lokala skärmar

När användaren väljer Kommandoradsinloggning på inloggningsbilden avbryts X-servern tillfälligt, vilket gör att den traditionella kommandoradsinloggningen på bitmönsterskärmen kan användas. När användaren har loggat in och sedan ut, eller efter angiven time-out, startas X-servern om.

Obs! – Alternativet Kommandoradsinloggning kan inte användas på nätverksskärmar.

Skärmtyp kontrollerar hur Kommandoradsinloggning fungerar. Formatet är:

- `local@`*skärmenhet*
- `local`
- `foreign`

När `local@`*skärmenhet* anges antar inloggningsservern att X-servern och `/dev/`*skärmenhet* finns på samma fysiska enhet och att kommandoradsinloggningen (normalt `getty`) körs på enheten. När

användaren väljer Kommandoradsinloggning avbryts X-servern och kommandradsinloggningen (`getty`) som körs på `/dev/`*skärmenhet* kan användas.

Om du vill avaktivera alternativet Kommandoradsinloggning på en skärm anger du `none` som *skärmenhet*. Standardvärdet är `console`. När `local` anges får *skärmenhet* också standardvärdet `console`. När `foreign` anges kan Kommandoradsinloggning inte användas.

Obs! – Alternativet Kommandoradsinloggning avaktiveras på den lokala skärmen när inloggningsservern startar från kommandoraden.

Anpassa textskärmen

Om inloggningsserversystemet har en direkt ansluten textskärm som fungerar som en konsol, kan du ange *skärmenhet* till `none` för att avaktivera Kommandoradsinloggning på bitmönsterskärmen.

Om en kommandoradsinloggning (`getty`) körs på både text- och bitmönsterskärmen, kan du också ändra *skärmenhet* till enheten för kommandoradsinloggning (`getty`) på bitmönsterskärmen.

Om kommandoradsinloggningen (`getty`) på bitmönsterskärmen finns på enheten `/dev/tty01`, ändrar du *skärmtyp* till `local@tty01`.

Visa en inloggningsbild på en nätverksskärm

Inloggningsservern kan acceptera förfrågningar från nätverksskärmar att visa en inloggningsbild på en viss skärm. Nätverksskärmen är normalt en X-terminal men kan också vara en arbetsstation.

För att kunna hantera förfrågningar från nätverksskärmar stöder inloggningsservern X Display Manager Protocol (XDMCP) 1.0. Med detta protokoll kan inloggningsservern förhandla och acceptera eller avslå förfrågningar från nätverksskärmar. XDMCP finns inbyggt i de flesta X-terminaler.

XDMCP direktförfrågningar från nätverksskärmar

När du konfigurerar X-terminalen att använda XDMCP direkt (frågeläge), anger du namnet på inloggningsvärden. När X-terminalen startas kontaktas inloggningsservern automatiskt, och en inloggningsbild visas på X-terminalen. Mer information om hur du konfigurerar X-terminalen för XDMCP direktläge finns i dokumentationen till X-terminalen.

De flesta X-servrar stöder också alternativet `-query`. I det här läget fungerar X-servern som om den var en X-terminal, och kontaktar inloggningserverns värddator direkt med en begäran om att inloggningsbilden ska visas på X-servern. Om du startar X-servern på en bitmönsterskärm på arbetsstationen `birgitta`, visar inloggningsservern `anita` en inloggningsbild på X-servern:

```
X -query anita
```

XDMCP indirekta förfrågningar från nätverksskärmar

När du konfigurerar X-terminlen att använda XDMCP indirekt läge, anger du namnet på inloggningsvärden. När X-terminaler startas, kontaktas inloggningsservern. Denna visar, via en valskärm, en lista med andra inloggningsvärdar i nätverket. I denna lista kan användaren välja en värddator, och värddatorn visar en inloggningsbild på användarens X-terminal. Mer information om hur du konfigurerar X-terminalen för XDMCP indirekt läge finns i dokumentationen till X-terminalen.

I detta fall stöder X-servrar alternativet X `-indirect`, med vilket X-servern kontaktar inloggningsservern i XDMCP indirekt läge.

Nätverksskärmar som inte stöder XDMCP

Det kan hända att äldre X-terminaler inte stöder XDMCP. Om inloggningsservern ska kunna visa en inloggningsbild på den här typen av terminal, listar du X-terminalnamnet i filen `Xservers`.

Exempel

Följande rader i filen `Xservers` tvingar inloggningsservern att visa två X-terminaler som inte stöder XDMCP, `rut` och `kurt`:

```
rut.blackdog.com:0 AcmeXsta foreign
kurt:0 PandaCo foreign
```

Eftersom skärmen finns i nätverket ingår värddatornamnet som en del i *skärmnamn*. *Skärmklass* används för att ange resurser som är specifika för en viss klass av X-terminaler. (I dokumentationen till X-teminalen finns information om vilken klass X-terminalen hör till.) *Skärmtyp* för `foreign` anger att inloggningsservern ska anslutas till en befintlig X-terminal i stället för att starta en egen. I detta fall anges inte *X-serverkommando*.

Kontrollera tillgång till inloggningsservern

Normalt kan vilken värddator som helst i nätverket som har tillgång till inloggningsservern begära att en inloggningsbild ska visas. Du kan begränsa tillgången till inloggningsservern genom att ändra filen `Xaccess`.

Om du vill ändra `Xaccess` kopierar du `Xaccess` från `/usr/dt/config` till `/etc/dt/config`. När du har ändrat `/etc/dt/config/Xaccess` uppmanar du inloggningsservern att läsa `Xaccess` genom att skriva:

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

Detta startar kommandot `kill -HUP` *inloggningsserverns process-ID*.

XDMCP direkt

När en värddator försöker ansluta till en inloggningsserver via XDMCP direkt, jämförs värddatornamnet med posten `Xaccess` för att avgöra om värddatorn har tillgång till inloggningsservern. Varje `Xaccess`-post är ett värddatornamn, inklusive jokertecknen `*` (asterisk) och `?` (frågetecken). En `*` (asterisk) motsvarar noll eller flera tecken och ett `?` (frågetecken) motsvarar ett enstaka valfritt tecken. Ett `!` (utropstecken) före en post hindrar åtkomst. Finns det inget utropstecken före posten har värddatorn tillgång till servern.

Anta att `Xaccess` innehåller följande tre poster:

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

Den första posten ger tillgång till inloggningsservern från värddatorn `amazon.waterloo.com`, den andra posten innebär tillgång från alla värddatorer vars fullständiga domännamn slutar med `dept5.waterloo.com`, och den sista posten hindrar åtkomst från alla andra värddatorer.

XDMCP indirekt

När en värddator försöker ansluta till en inloggningsserver via XDMCP indirekt, jämförs värddatornamnet med posten Xaccess för att avgöra om värddatorn har tillgång till inloggningsservern. Alla Xaccess-poster liknar posterna för XDMCP direkt, inlusive jokertecken. Undantaget är att alla poster markeras med strängen CHOOSER. Exempel:

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

Den första posten ger även här tillgång till inloggningsservern från värddatorn amazon.waterloo.com. Den andra posten innebär tillgång från alla värddatorer vars fullständiga domännamn slutar med dept5.waterloo.com och den sista posten hindrar åtkomst från alla andra värddatorer.

Något av följande kan finnas efter CHOOSER:

- BROADCAST
- *lista över värddatornamn*

BROADCAST uppmanar inloggningsservern att meddela (broadcast) inloggningsserverns undernätverk att skapa en lista över tillgängliga inloggningsvärdar. En lista över värddatornamn talar om för inloggningsservern att den ska använda denna lista som listan över tillgängliga inloggningsvärdar. Exempel:

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

Om amazon.waterloo.com ansluter via XDMCP indirekt, visas en lista som innehåller shoal och alum. Om alice.dept5.waterloo.com ansluter visas en lista över alla tillgängliga inloggningsvärdar i inloggningsserverns undernätverk. Andra förfrågningar för XDMCP indirekt avvisas.

Ett annat sätt är att definiera ett eller fler makron som innehåller en lista över värddatornamn. Exempel:

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

1

Felsökning

Normalt loggar inloggningsservern fel i filen `/var/dt/Xerrors`. Om du vill ändra detta anger du resursen `Dtlogin.errorLogFile` i filen `Xconfig`. Katalogen som anges måste finnas när inloggningsservern startas.

Om du vill att inloggningsservern ska lagra fel i filen `/var/mylogs/Dterrors` anger du följande i filen `Xconfig`:

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

När du startar om inloggningsservern loggas fel i filen `/var/mylogs/Dterrors`. Katalogen `/var/mylogs` måste finnas när inloggningsservern startas.

Avaktivera inloggningsservern

- Om du inte vill att inloggningsservern ska starta när systemet startar, skriver du:

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

  Detta uppmanar systemet att inte starta inloggningsservern när du startar om.

- Om du vill avaktivera inloggningsservern genom att ta bort dess process-ID, skriver du:

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

  Detta kör kommandot `kill` *inloggningsserverns process-ID*.

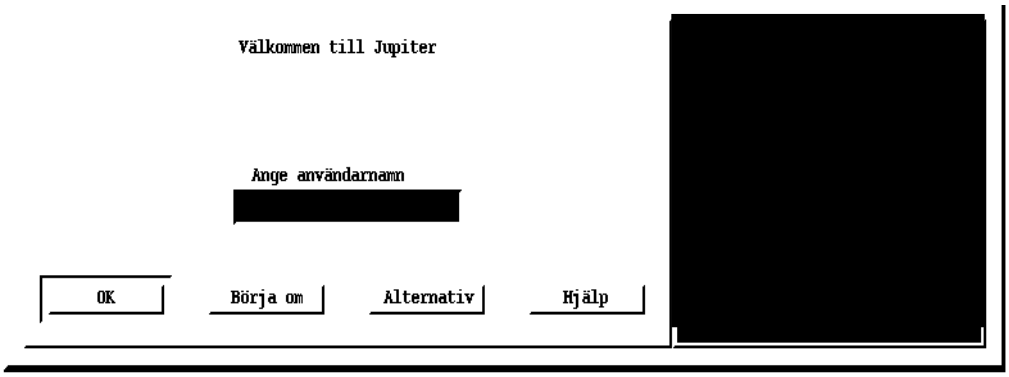
Obs! – Om du tar bort inloggningsserverns process avbryts alla användarsessioner som hanteras av inloggningsservern.

Du kan också avaktivera inloggningsservern genom att ta bort dess process-ID. Inloggningsserverns process-ID lagras i `/var/dt/Xpid` eller i den fil som anges i `Xconfig` av resursen `Dtlogin.pidFile`.

Om du loggas in till skrivbordet samtidigt som du tar bort inloggningsservern, avbryts skrivbordssessionen omedelbart.

Inloggningsbilden

Inloggningsbilden som inloggningsservern visar är ett attraktivt alternativ till den traditionella teckenbaserade inloggningsbilden, och omfattar dessutom fler funktioner.



figur 1-2 Inloggningsbilden

På samma sätt som vid en teckenbaserad inloggning skriver användaren ett användarnamn följt av ett lösenord. Om användaren verifieras startar en skrivbordssession för användaren. När användaren avslutar sessionen visas en ny inloggningsbild, och processen börjar om igen.

Om du vill anpassa inloggningsbilden kan du:

- Ändra hur inloggningsbilden ser ut
- Konfigurera X-serverbehörighet
- Ändra standardspråk
- Starta kommandon innan inloggningsbilden visas
- Ändra innehållet på inloggningsbildens Språk-meny
- Ange vilket kommando som ska starta användarens session
- Exekvera kommandon innan användarens skrivbordssession startar
- Exekvera kommandon efter användarens session slutar

Allt detta kan göras för alla skärmar eller för en skärm i taget.

1

Ändra hur inloggningsbilden ser ut

Om du vill anpassa hur inoggningsbilden ser ut ändrar du logotypen eller grafiken, välkomstmeddelandet och fonterna.

Om du vill ändra `Xresources` kopierar du `Xresources` från `/usr/dt/config/`*språk* till `/etc/dt/config/`*språk*. Om du vill tvinga inloggningsbilden att visas en gång till, markerar du Återställ inloggningsskärm på inloggningsbildens Alternativ-meny.

Attribut till inloggningsbilden som kan bestämmas av resursspecifikationer i filen `Xresources` omfattar:

`Dtlogin*logo*bitmapFile`	Bitmönster- eller bildpunktsmönsterfil som ska visas som en logotyp
`Dtlogin*greeting*labelString`	Välkomstmeddelande
`Dtlogin*greeting*persLabelString`	Personligt välkomstmeddelande
`Dtlogin*greeting*fontList`	Fonter för välkomstmeddelanden
`Dtlogin*labelFont`	Fonter för knappar och namn
`Dtlogin*textFont`	Fonter för hjälp- och felmeddelanden
`Dtlogin*`*språk*`*languageName`	Alternativ text för lokalt *språk*

▼ Så här ändrar du logotyp

- Ange resursen `Dtlogin*logo*bitmapFile` i `Xresources`.

  Logotypen kan vara en färgad bildpunktsmönsterfil eller en bitmönsterfil.

I följande exempel används bitmönstret `Mylogo` som logotyp:

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

▼ Så här ändrar du välkomstmeddelandet

Normalt visas meddelandet `Välkommen till` *värddatornamn* på inloggningsbilden. Om du vill ändra meddelandet:

- Ange resursen `Dtlogin*greeting*labelString` i `Xresources`.

  Värdet på resursen `labelString` kan innehålla %*LocalHost*%, vilket ersätts med inloggningsserverns värddatornamn och %*DisplayName*%, vilket ersätts med X-serverns skärmnamn.

I följande exempel ändras välkomstmeddelandet till `Detta är` *värddatornamn*`!`:

```
Dtlogin*greeting*labelString: Detta är %LocalHost%!
```

När användarnamnet har skrivits in visas normalt meddelandet Välkommen *användarnamn*. Du ändrar det här meddelandet genom att ange resursen `Dtlogin*greeting*persLabelString` i `Xresources`. Värdet på `persLabelString` kan innehålla `%s`, vilket ersätts med *användarnamn*.

I följande exempel ändras det personliga välkomstmeddelandet till Hej *användarnamn*.

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

▼ Så här ändrar du fonter

Du ändrar fonterna som används i inloggningsbilden genom att ange följande fontresurs i `Xresources:`

Om du vill lista tillgängliga fonter skriver du:

`xlsfonts [-`*alternataiv*`] [-fn` *mönster*`]`

`Dtlogin*greeting*fontList` Font för välkomstmeddelande

`Dtlogin*labelFont` Font för knappar och namn

`Dtlogin*textFont` Font för hjälp- och felmeddelanden

I följande exempel används en stor font till välkomstmeddelandet (värdet du anger måste rymmas på en rad):

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

▼ *Så här anger du olika texter för olika språk*

Om du vill att texten på inloggningsbildens Språk-meny ska vara på ett visst språk i stället för standardspråket, ändrar du resursen `Dtlogin*`*språk*`*languageName` i `Xresources`:

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

Texten `American` visas i stället för det lokala namnet `En_US`.

Ändra hur inloggningsbilden fungerar

Om du vill anpassa hur inloggningsbilden fungerar ändrar du resurserna som anges i filen `Xconfig`.

Om du vill ändra `Xconfig` kopierar du den från `/usr/dt/config` till `/etc/dt/config`**.** När du har ändrat `/etc/dt/config/Xconfig` uppmanar du inloggningsservern att läsa `Xconfig` igen genom att skriva:

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

Detta startar kommandot `kill -HUP` *inloggningsserverns process-ID*

Resurser som anges i filen `Xconfig` omfattar:

`Dtlogin*authorize`	`Xaccess`-filens specifikation
`Dtlogin*environment`	X-servermiljö
`Dtlogin*language`	Standardspråk
`Dtlogin*languageList`	Språklista för inloggningsbildens Språk-meny
`Dtlogin*resources`	Specifikation till `Xresources`
`Dtlogin*setup`	`Xsetup`-filens specifikation
`Dtlogin*startup`	`Xstartup`-filens specifikation
`Dtlogin*session`	`Xsession`-filens specifikation
`Dtlogin*failsafeClient`	`Xfailsafe`-skriptets specifikation
`Dtlogin*reset`	`Xreset`-skriptets specifikation
`Dtlogin*userPath`	PATH för `Xsession` och `Xfailsafe`

`Dtlogin*systemPath`	PATH för `Xsetup`, `Xstartup` och `Xfailsafe`
`Dtlogin*systemShell`	SHELL för `Xsetup`, `Xstartup` och `Xfailsafe`
`Dtlogin.timeZone`	TZ för alla skript

Ändra hur inloggningsbilden fungerar skärmvis

I exemplet nedan påverkar ändringar av resursen `Xconfig` hur inloggningsbilden fungerar på alla skärmar. Resurser som listas med en asterisk (*) kan anges per skärm. Detta gör att du kan ange anpassade inloggningsbilder för vissa skärmar. Om du vill ange en resurs för en viss skärm, anges resursen som `DtLogin`*Skärmnamn*resurs*. Om du t ex vill koppla från användarbaserad behörighetskontroll för skärmen `expo:0,` men ha det på för andra skärmar, skriver du:

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

Obs! – Alla specialtecken i skärmnamnet som: (kolon) och . (punkt) ersätts med _ (understreck).

Ändra X-serverbehörighet

Normalt kontrollerar inloggningsservern X-serverbehörighet per användare, där kontrollen baseras på verifieringsdata som lagras och skyddas i filen *Hemkatalog*`/.Xauthority`. Endast användare som kan läsa den här filen får ansluta till X-servern. Detta är den vanligaste metoden för kontroll av X-serverbehörighet.

Ett alternativ till användarbaserad behörighetskontroll är värddatorbaserad behörighetskontroll. Om en värddator är garanterad behörighet till X-servern, kan alla användare på den värddatorn ansluta till X-servern. Du använder värddatorbaserad kontroll när:

- Äldre R2- och R3 X-klienter inte kan ansluta till en X-server med användarbaserad behörighetskontroll.
- Obehöriga kan snappa upp behörighetsdata som skickas mellan X-klienten och X-servern i ett oskyddat nätverk.

Resursen `Xconfig Dtlogin*authorize` uppmanar inloggningsservern att använda användarbaserad kontroll av X-serverbehörighet. Om du vill använda värddatorbaserad behörighetskontroll ändrar du värdet på verifieringsresursen till `False`, t ex:

```
Dtlogin*authorize: False
```

▼ *Så här ändrar du X-servermiljö*

Om du vill ge X-servern en eller flera miljövariabler och värden när den startas av inloggningsservern, anger du dem med resursen `Dtlogin*environment` i `Xconfig`. Exempel:

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

gör variabeln `VAR1` och `VAR2` tillgängliga för den lokala X-serverprocessen. Dessa variabler exporteras också till skripten `Xsession` och `Xfailsafe`.

▼ *Så här ändrar du standardspråk*

När användaren loggar in till skrivbordet från inloggningsbilden körs användarsessionen med det språk som markeras på undermenyn Språk till Alternativ-menyn. Om användaren inte väljer något språk används inloggningsserverns standardspråk. Du styr värdet på standardspråket genom att ställa in resursen `Dtlogin*language` i `Xconfig`. Exempel:

```
Dtlogin*language: Ja_JP
```

Information om hur du kontrollerar vilka språk som är installerade i systemet finns i systemdokumentationen.

▼ *Så här ändrar du innehållet på inloggningsbildens Språk-meny*

Normalt skapas inloggningsbildens Språk-meny av en lista över de språk som finns installerade i systemet. När användaren väljer ett språk från menyn, visas inloggningsbilden på nytt med det valda språket. När användaren loggar in nästa gång startas en skrivbordssession för användaren med det språket.

Du kan ange en egen lista över språk genom att ändra resursen `Dtlogin*languageList` i `Xconfig`:

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

Nu visas bara `En_US` och `De_DE` på inloggningsbildens Språk-meny.

Starta kommandon innan inloggningsbilden visas

Efter att X-servern har startat, men innan inloggningsbilden visas, körs skriptet `Xsetup`. `Xsetup` körs med rotbehörighet och startar kommandon som måste köras innan inloggningsbilden kan visas.

Om du vill ändra `Xsetup` kopierar du `Xsetup` från `/usr/dt/config` till `/etc/dt/config`**.** Nästa gång inloggningsbilden visas körs det ändrade `Xsetup`.

Starta kommandon innan användarsessionen börjar

Efter att användaren har skrivit användarnamnet och lösenordet och de har verifierats, men innan användarsessionen börjar, körs skriptet `Xstartup`. `Xstartup` körs med rotbehörighet och startar kommandon som måste köras innan användarsessionen kan börja.

Om du vill ändra `Xstartup` kopierar du `Xstartup` från `/usr/dt/config` till `/etc/dt/config`**.** Nästa gång användaren loggar in körs det ändrade `Xstartup`.

Starta en skrivbordssession

Normalt startas användarsessionen genom att skriptet `Xsession` körs. `Xsession` körs med användarens behörighet och startar de kommandon som behövs för att starta skrivbordet.

Obs! – Uppdatera inte `Xsession` direkt.

Mer information om hur du anpassar start av användarens skrivbordssession finns i Kapitel 2, ”Konfigurera sessionshanteraren”.

Starta en session med endast ett fönster

Om användaren väljer Session med endast ett fönster på undermenyn Sessioner till inloggningsbildens Alternativ-meny, körs skriptet `Xfailsafe`. `Xfailsafe` körs med användarbehörighet och startar kommandon som behövs för att starta en minimal fönstermiljö, normalt ett terminalfönster och ytterligare en fönsterhanterare.

Om du vill ändra `Xfailsafe` kopierar du `Xfailsafe` från `/usr/dt/config` till `/etc/dt/config`**.** Nästa gång användaren loggar in körs `Xfailsafe`.

När användarsessionen avslutas

När användaren avslutar skrivbordet, eller sessionen med endast ett fönster, körs skriptet `Xreset`. `Xreset` körs med rotbehörighet och startar kommandon som måste köras som rot efter att användarsessionen avslutas.

Om du vill ändra `Xreset` kopierar du `Xreset` från `/usr/dt/config` till `/etc/dt/config`**.** Nästa gång användaren loggar in körs det ändrade `Xreset`.

Inloggningsserverns miljö

Inloggningsservern omfattar en miljö som exporteras till skripten `Xsetup`, `Xstartup`, `Xsession`, `Xfailsafe` och `Xreset`. Miljön beskrivs i tabell 1-1. Ytterligare variabler kan också exporteras av inloggningsservern.

tabell 1-1 Inloggningsserverns miljöer

Miljö-variabel	`Xsetup`	`Xstartup`	`Xsession`	`Xreset`	**Beskrivning**
LANG	X	X	X	X	Standardspråk eller valt språk
XAUTHORITY	X	X	X	X	Alternativ X-behörighetsfil (valfri)
PATH	X	X	X	X	Värdet på resursen `Dtlogin*userPath` (`Xsession`, `Xfailsafe`) eller `Dtlogin*systemPath` (`Xsetup`, `Xstartup`, `Xreset`)
DISPLAY	X	X	X	X	X-serverns anslutningsnummer
SHELL	X	X	X	X	Skal som anges i resursen `/etc/passwd` (`Xsession`, `Xfailsafe`) eller `Dtlogin*systemShell` (`Xsetup`, `Xstartup`, `Xreset`)
TZ	X	X	X	X	Värdet på resursen `Dtlogin.timeZone` eller tidszon som bestäms av systemet

tabell 1-1 Inloggningsserverns miljöer *(Continua)*

Miljö-variabel	Xsetup	Xstartup	Xsession	Xreset	Beskrivning
USER		X	X	X	Användarnamn
HOME		X	X	X	Hemkatalog som anges i `/etc/passwd`
LOGNAME		X	X	X	Användarnamn

Ändra användar- och systemsökväg

Inloggningsservern anger miljövariabeln PATH när skripten `Xsession` och `Xfailsafe` körs. Du kan ange en annan sökväg för dessa skript.

▼ *Så här ändrar du användarsökväg*

♦ Ange resursen `Dtlogin*userPath` i `Xconfig`. Exempel:

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

▼ *Så här ändrar du systemsökväg*

♦ Ange resursen `Dtlogin*systemPath` i `Xconfig`. Exempel:

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

▼ Ändra systemskalet

Inloggningsservern anger miljövariabeln SHELL när skripten `Xsetup`, `Xstartup` och `Xfailsafe` körs. Standardvärdet är `/bin/sh`. Om du vill ange ett annat skal för dessa skript, anger du resursen `Dtlogin*systemShell` i `Xconfig`. Exempel:

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

▼ Så här ändrar du tidszon

Inloggningsservern anger miljövariabeln TZ när skripten `Xsetup`, `Xstartup`, `Xsession`, `Xfailsafe` och `Xreset` körs. Standardvärdet härleds från systemet, så normalt behöver du inte ändra detta. Om du vill ange en annan tidszon för dessa skript anger du resursen `Dtlogin.timeZone` i `Xconfig`. Exempel:

```
Dtlogin.timeZone: CST6CDT
```

Administrera inloggningshanteraren

När inloggningsservern startar börjar en `dtlogin`-process. Denna process läser filen `Xconfig` för att bestämma initial serverkonfiguration och hitta andra konfigurationsfiler för inloggningsserven. Inloggningsservern läser sedan filen `Xservers` för att kontrollera om den ska hantera några skärmar, och läser också filen `Xaccess` för att kontrollera behörighet till inloggningsservern.

Om inloggningsservern får information från filen `Xservers` att den måste hantera en lokal skärm, startas en X-server på det sätt som instrueras i filen `Xservers`, och sedan visas inloggningsbilden på den skärmen.

Om inloggningsservern får information från filen `Xservers` att den måste hantera en nätverksskärm, antas det att X-serven redan körs med angivet skärmnamn och inloggningsbilden visas på den skärmen.

Inloggningsservern väntar sedan på en XDMCP-förfrågan från nätverket.

För varje skärm som hanteras skapar inloggningsservern först en ny `dtlogin`-process. Detta innebär att om inloggningsservern hanterar *n* skärmar, finns det *n+1* `dtlogin`-processer. Inloggningsservern kör skriptet `Xsetup`, laddar filen `Xresources` och kör sedan `dtgreet` för att visa inloggningsbilden. När användaren har skrivit användarnamnet och lösenordet och har blivit verifierad, körs skriptet `Xstartup` och sedan skriptet `Xsession` eller `Xfailsafe`. När användaren avslutar sessionen körs skriptet `Xreset`.

Om inloggningsservern får en XDMCP indirekt förfrågan, körs `dtchooser` för att visa en lista över inloggningsvärdar på den skärmen. När användaren väljer en värddator från listan, kommer inloggningsservern på den värddatorn att hantera skärmen.

För konfigurationsfilerna `Xaccess`, `Xconfig`, `Xfailsafe`, `Xreset`, *language*`/Xresources`, `Xservers`, `Xsetup` och `Xstartup` letar inloggningsservern normalt först i `/etc/dt/config`, sedan i `/usr/dt/config` och använder den första filen som hittas.

Inloggningshanterarens filer

Standardsökvägen till inloggningshanterarens filer är:

`/usr/dt/bin/dtlogin`	Inloggningsservern och skärmhanteraren
`/usr/dt/bin/dtgreet`	Visar en inloggningsbild för en skärm
`/usr/dt/bin/dtchooser`	Visar en valskärm för en skärm
`/usr/dt/bin/Xsession`	Startar en skrivbordssession
`/usr/dt/config/Xfailsafe`	Startar en session med endast ett fönster
`/usr/dt/config/Xconfig`	Inloggningsserverns konfigurationsfil
`/usr/dt/config/Xservers`	Inloggningsserverns skärmbeskrivningsfil
`/usr/dt/config/Xaccess`	Inloggningsserverns behörighetbeskrivningsfil
`/usr/dt/config/`*språk*`/Xresources`	Skärmlayoutresurser
`/usr/dt/config/Xsetup`	Skärminstallationsfil
`/usr/dt/config/Xstartup`	Pre-session startfil
`/usr/dt/config/Xreset`	Post-session återställningsfil
`/var/dt/Xpid`	Inloggningsserverns process-ID
`/var/dt/Xerrors`	Inloggningsserverns fellogg

1

Konfigurera sessionshanteraren 2

Sessionshanteraren ansvarar för att starta skrivbordet och automatiskt spara och återställa program som körs, samt färger, fonter, musfunktioner, ljudvolym och tangentbordsklickningar.

Med sessionshanteraren kan du:

- Anpassa startsessionen för alla skrivbordsanvändare
- Anpassa miljö och resurser för alla skrivbordsanvändare
- Ändra sessionens startmeddelande
- Ändra parametrar för sessionens startverktyg och daemons
- Anpassa användningen av skrivbordsfärger för alla användare

Vad är en session?	*23*
Starta en session	*25*
När sessionen startar	*25*
Ytterligare anpassningar av sessionsstarten	*32*
Filer och kataloger i sessionshanteraren	*37*

Vad är en session?

En session är en mängd program, inställningar och resurser som visas på användarens skrivbord. Sessionshanteraren är en uppsättning konventioner och protokoll som gör att sessionshanteraren kan spara och återställa en

användarsession. En användare kan logga in till systemet och få samma uppsättning program, inställningar och resurser som fanns när han eller hon loggade ut. När en användare loggar in till skrivbordet för första gången, laddas en startsession av standardtyp. Därefter används begreppen aktiv session och hemsession.

Startsession

När en användare loggar in till skrivbordet första gången, skapar sessionshanteraren användarens startsession med hjälp av systemets standardvärden. Normalt startas filhanteraren och hjälpvolymen Introduktion till skrivbordet.

Aktiv session

Den session som användaren kör betraktas alltid som aktiv session, oavsett om den har återställts vid inloggning från en sparad hemsession, en sparad aktiv session eller från systemets standardstartsession. Sessionshanteraren sparar automatiskt aktiv session, baserat på användarens Startinställningar i Arbetsinställningar, när användaren avslutar sessionen. När användaren loggar in nästa gång till skrivbordet startar den tidigare sparade aktiva sessionen, vilket innebär att skrivbordet återställs till samma status som när användaren loggade ut.

Hemsession

Du kan också låta skrivbordet återställas till samma status varje gång användaren loggar in, oavsett vilken status det hade när användaren loggade ut. Användaren kan spara status i aktiv session och sedan, med Startinställningar i Arbetsinställningar, låta sessionshanteraren starta den sessionen varje gång användaren loggar in.

Skärmspecifika sessioner

Om en användare vill köra en viss session för en viss skärm, kan han eller hon skapa en skärmspecifik session. Användaren ska då kopiera katalogen *Hemkatalog*`/.dt/sessions` till *Hemkatalog*`/.dt/`*skärm*, där *skärm* är det

verkliga, ogiltiga värddatornamnet (t ex, `pablo:0` är giltigt, `pablo.gato.com:0` eller `unix:0` är det inte). När användaren loggar in till skärmen `pablo:0` startar sessionshantaren den skärmspecifika sessionen.

Starta en session

Sessionshanteraren startas via `/usr/dt/bin/Xsession`. När användaren loggar in med hjälp av inloggningshanteraren startas `Xsession` som standard.

Användaren kan också logga in med den traditionella teckenbaserade inloggningen (`getty`) och starta sessionshanteraren manuellt, med hjälp av de verktyg som startar en X-server, t ex `xinit`. Exempel: `xinit /usr/dt/bin/Xsession`.

När sessionen startar

När sessionshanteraren startar går den igenom följande steg för att starta användarens session:

1. Läser in och kör skriptet *Hemkatalog*`/.dtprofile`
2. Läser in och kör skriptet `Xsession.d`
3. Visar ett välkomstmeddelande
4. Ställer in skrivbordets sökvägar
5. Samlar ihop tillgängliga program
6. Läser in och kör (valfritt) *Hemkatalog*`/.profile` eller *Hemkatalog*`/.login`
7. Startar ToolTalk®
8. Laddar sessionsresurser
9. Startar färghanteraren
10. Startar arbetsytehanteraren
11. Startar sessionens program

Dessa steg beskrivs i avsnitten nedan.

Läsa in och köra skriptet .dtprofile

När sessionen startar, läser den in och kör skriptet `Xsession` i användarens *Hemkatalog*`/.dtprofile`-skript. Skriptet *Hemkatalog*`/.dtprofile` är ett `/bin/sh` eller `/bin/ksh`-skript som gör att användaren kan ange miljövariabler för sessionen. Mer information om hur man anger miljövariabler finns i ”Ytterligare anpassningar av sessionsstarten” på sidan 32.

Om skriptet *Hemkatalog*`/.dtprofile` inte finns, t ex när användaren loggar in till skrivbordet första gången, kopierar `Xsession` skrivbordets normala `sys.dtprofile` till *Hemkatalog*`/.dtprofile`.

Skrivbordets standardvärde är `/usr/dt/config/sys.dtprofile`. Om du vill anpassa skriptet `sys.dtprofile` kopierar du `sys.dtprofile` från `/usr/dt/config` till `/etc/dt/config` och redigerar den nya filen.

Läsa in och köra skriptet Xsession.d

När skriptet *Hemkatalog*`/.dtprofile har lästs in och körts`, läser skriptet `Xsession` in skriptet `Xsession.d` och kör det. Dessa skript används för att ange ytterligare miljövariabler och starta fler daemons för användarens session. Standardskripten för `Xsession.d` är:

`0010.dtpaths`	Dokumenterar anpassningsbara skrivbordssökvägar
`0020.dtims`	Startar (valfritt) ims (input method server)
`0030.dttmpdir`	Skapar temporär katalog per användare och per session
`0040.xmbind`	Anger `$XMBINDDIR` till skrivbordets standardvärde

Det kan finnas ytterligare leverantörsspecifika skript i `Xsession.d`.

`Xsession` läser först in och kör alla filer i katalogen `/etc/dt/config/Xsession.d`, följt av de i katalogen `/usr/dt/config/Xsession.d`.

Skrivbordets standardskript, `Xsession.d`, finns i katalogen `/usr/dt/config/Xsession.d`. Om du vill anpassa `Xsession.d` kopierar du skripet från `/usr/dt/config/Xsession.d` till `/etc/dt/config/Xsession.d` och redigerar den nya filen. Du måste ha exekveringsbehörighet för att kunna göra detta.

Om du vill att Xsession automatiskt ska läsa in och köra ett av dina egna skript, kopierar du det till /etc/dt/config/Xsession.d.

Obs! – När du ändrar eller skapar ett Xsession.d-skript, ska du kontrollera att alla förgrundskommandon du kör är korta, eftersom tiden som kommandot tar upp direkt påverkar starttiden. Om ett förgrundskommando inte avslutas hänger sig sessionsstarten. Kommandon som körs i ett Xsession.d-skript och som du vill ska fortsätta köras under hela sessionen, ska köras i bakgrunden.

Visa välkomstmeddelandet

Efter att Xsession har lästs in och startat skripten *Hemkatalog*/.dtprofile och Xsession.d, visas ett välkomstmeddelande som täcker skärmen. Du kan anpassa välkomstmeddelandet eller stänga av det. Klienten dthello används för att visa meddelandet.

Om du vill ändra meddelandetexten ändrar du alternativet dthello genom att modifiera variabeln dtstart_hello[0].

Om du vill ändra dtstart_hello[0] skapar du skriptet /etc/dt/config/Xsession.d som anger det nya värdet. Om du vill visa dagens meddelande för alla användare skapar du ett exkeverbart sh- eller ksh-skript, t ex /etc/dt/config/Xsession.d/*minavar*, och anger dtstart_hello[0] så här:

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

På samma sätt kan användarna ändra välkomstmeddelandet för sina egna sessioner genom att ange dtstart_hello[0] i *Hemkatalog*/.dtprofile.

Om du vill stänga av välkomstmeddelandet anger du dtstart_hello[0]="".

Mer information om dthello finns i kommandobeskrivningen till dthello .

Ange skrivbordssökvägar

Skrivbordssökvägarna skapas av dtsearchpath vid inloggningen. Det finns två kategorier av miljövariabler som används av dtsearchpath:

Indatavariabler	Systemtäckande och egna miljövariabler vars värden anges av systemadministratören eller slutanvändaren.

Utdatavariabler	Variabler som skapats och tilldelats värden av `dtsearchpath`. Värdet på varje variabel är sökvägen till skrivbordssessionen.

Om du vill ändra kommandoradsalternativ för `dtsearchpath` ändrar du variabeln `dtstart_searchpath`. Om du vill ändra variabeln `dtstart_searchpath` för alla användare, skapar du ett exekverbart `sh-` eller `ksh`-skript (t ex `/etc/dt/config/Xsession.d/`*minavar*) och anger `dtstart_searchpath` så här:

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

Användarna kan på ett liknande sätt ändra `dtsearchpath`-alternativen för sina egna sessioner genom att ange `dtstart_searchpath` i *Hemkatalog*`/.dtprofile`.

Mer information om `dtsearchpath` finns i Kapitel 7, ”Skrivbordets sökvägar”. Mer information om `dtsearchpath`-alternativ finns i kommandobeskrivningen till `dtsearchpath` .

Samla ihop tillgängliga program

Nästa steg efter att ha ställt in skrivbordssökvägar är att samla ihop tillgängliga program, med hjälp av `dtappgather`. Om du vill ändra kommandoradsalternativen för `dtappgather`, ändrar du variabeln `dtstart_appgather`. Om du vill ändra variabeln `dtstart_appgather` för alla användare, skapar du ett exekverbart `sh-` eller `ksh`-skript (t ex `/etc/dt/config/Xsession.d/`*minavar*) och anger `dtstart_appgather` så här:

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

Användarna kan på ett liknande sätt ändra `dtappgather`-alternativ för sina egna sessioner genom att ange `dtstart_appgather` i *Hemkatalog*`/.dtprofile`.

Mer information om `dtappgather`-alternativ finns i kommandobeskrivningen till `dtappgather (4)`.

Valfritt läsa in och köra skripten .profile och .logint

Xsession kan läsa in och köra användarnas traditionella *Hemkatalog*/.profile- och *Hemkatalog*/.login-skript. Normalt är denna funktion avstängd. Om du vill att Xsession ska läsa in och köra skriptet .profile eller .login anger du DTSOURCEPROFILE till true.

Om du vill ändra DTSOURCEPROFILE för alla användare skapar du skriptet etc/dt/config/Xsession.d som anger det nya värdet. Om du vill ange DTSOURCEPROFILE till true för alla användare, skapar du ett exekverbart sh- eller ksh-skript, t ex /etc/dt/config/Xsession.d/*minavar*, och anger DTSOURCEPROFILE så här:

```
DTSOURCEPROFILE=true
```

Användarna kan på ett liknande sätt ange DTSOURCEPROFILE för sina egna sessioner genom att ange DTSOURCEPROFILE till true i *Hemkatalog*/.dtprofile.

Starta ToolTalk

ToolTalk-meddelandedaemonen ttsession gör att oberoende program kan kommunicera utan att ha direkt kännedom om varandra. Program skapar och skickar ToolTalk-meddelanden för att kommunicera med varandra. ttsession kommunicerar i nätverket för att leverera meddelanden.

Om du vill ändra kommandoradsalternativ för ttsession ändrar du variabeln dtstart_ttsession. Om du vill ändra variabeln dtstart_ttsession för alla användare, skapar du ett exekverbart sh- eller ksh-skript (t ex /etc/dt/config/Xsession.d/*minavar*) och och anger dtstart_ttsession så här:

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

Användarna kan på ett liknande sätt ändra ttsession-alternativ för sina egna sessioner genom att ange dtstart_ttsession i *Hemkatalog*/.dtprofile.

Mer information om ttsession-alternativ finns i kommadobeskrivningen till ttsession. Mer information om ttsession finns i *Getting Started Using ToolTalk Messaging*.

Starta sessionshanterarklienten

`Xsession` startar vid den här punkten `/usr/dt/bin/dtsession`, vilket innehåller sessionens startprocess.

Ladda sessionsresurser

Sessionshanteraren använder X-serveregenskapen RESOURCE_MANAGER för att göra skrivbordsresurser tillgängliga för alla program. Sessionshanteraren laddar RESOURCE_MANAGER genom att:

- Ladda systemets standardresurser
- Slå samman alla systemtäckande resurser som anges av systemadministratören.
- Slå samman användarspecificerade resurser

Skrivbordets standardresurser finns i `/usr/dt/config/`*språk*`/sys.resources`. Dessa resurser görs tillgängliga för alla användare via egenskapen RESOURCE_MANAGER. Den här filen ska inte redigeras eftersom den skrivs över vid senare skrivbordsinstallationer.

Du kan utvidga systemets standardresurser genom att skapa `/etc/dt/config/`*språk*`/sys.resources`. I den här filen kan du åsidosätta standardresurserna eller ange ytterligare resurser för alla skrivbordsanvändare. Eftersom den här filen kopplas samman med skrivbordets standardresurser när sessionen startar, ska endast nya eller uppdaterade resursspecifikationer placeras i den här filen. Resurser som specificeras i filen blir tillgängliga för alla användares sessioner via egenskapen RESOURCE_MANAGER. Resurser i denna fil gäller före de som anges i skrivbordets standardresursfil.

Användare kan utvidga skrivbordets standardresurser och systemtäckande resurser med filen *Hemkatalog*`/.Xdefaults`. Resurser som anges i den här filen blir tillgängliga för denna användares session via egenskapen RESOURCE_MANAGER. Resurser i den här filen gäller före de som anges i skrivbordets standardresursfil och systemadministratörens resursfil.

Obs! – Hjälpprogrammet X Toolkit Intrinsics anger att det laddar resurser för ett program från antingen RESOURCE_MANAGER eller från *Hemkatalog*`/.Xdefaults`, men inte båda. Vanligtvis innebär detta att användarens *Hemkatalog*`/.Xdefaults`-fil ignoreras. Men sessionshanteraren kan innehålla *Hemkatalog*`/.Xdefaults` genom att den slås samman med RESOURCE_MANAGER när sessionen startar, vilket beskrivs ovan. Om en

användare ändrar *Hemkatalog*/.Xdefaults, blir ändringarna inte synliga för nya program förrän användaren anropar funktionen Ladda om resurser. Funktionen Ladda om resurser instruerar sessionshanteraren att starta om RESOURCE_MANAGER med standardresurser, systemtäckande resurser eller användarspecificerade resurser. På detta sätt görs ändringar av systemtäckande och egna resursfiler tillgängliga för alla program.

Mer information finns i:

- ”Ange programresurser” på sidan 257
- Kommandobeskrivningen till dtresourcesfile(4)

Starta färgservern

Sessionshanteraren fungerar som en färgservern för skrivbordet och tillhandahåller följande uppsättningar dtsession-resurser som kan användas för att konfigurera den.

foregroundColor	Kontrollerar om en bildpunkt allokeras till förgrundsfärgen
dynamicColor	Anger om skrivskyddade färger allokeras
shadowPixmaps	Anger om färger allokeras för övre skugga eller undre skugga
colorUse	Begränsar färgallokering
writeXrdbColors	Anger om resurserna *background och *foreground placeras i resursdatabasen

Du anger färghanterarresurser för alla användare genom att skapa /etc/dt/config/*språk*/sys.resources och ange färghanterarresurser i den filen.

Användarna kan på ett liknande sätt ange färghanterarresurser för sina egna sessioner genom att ange sådana resurser i *Hemkatalog*/.Xdefaults.

Mer information om hur du anger färghanterarresurser finns i ”Administrera färger” på sidan 267.

Starta arbetsytehanteraren

Sessionhanteraren startar arbetsytehanteraren. Normalt startas `/usr/dt/bin/dtwm`. En alternativ fönsterhanterare kan anges med resursen `wmStartupCommand`.

Du anger en alternativ fönsterhanterare för alla användare genom att skapa `/etc/dt/config/`*`språk`*`/sys.resources` och ange fullständig sökväg samt alternativ för fönsterhanteraren med resursen `Dtsession*wmStartupCommand` i den filen.

Användarna kan på ett liknande sätt ange en alternativ fönsterhanterare för sina egna sessioner genom att ange resursen `Dtsession*wmStartupCommand` i *Hemkatalog*`/.Xdefaults`.

Mer information om fönsterhanterare finns i Kapitel 14, ”Anpassa arbetsytehanteraren”.″

Starta sessionsprogram

När sessionen startar, startar sessionshanteraren om alla program som sparades som en del av sessionen. Systemets standardprogramuppsättning, som ska återställas som en del av användarens intiala session, finns i `/usr/dt/config/`*`språk`*`/sys.session`. Den här filen ska inte redigeras eftersom den skrivs över vid kommande skrivbordsinstallationer.

Mer information finns i kommandbeskrivningen till `dtsessionfile(4)`.

En systemadministratör kan ersätta den programuppsättning som startas som en del av användarens startsession genom att kopiera `/usr/dt/config/`*`språk`*`/sys.session` till `/etc/dt/config/`*`språk`*`/sys.session` och ändra den senare filen. Till skillnad från resursfilen, används den här filen som en fullständig ersättning till skrivbordets standardfil, så du kan skapa en kopia av systemets standardfiler och göra nödvändiga ändringar.

Ytterligare anpassningar av sessionsstarten

I det här avsnittet beskrivs hur du:

- Anger miljövariabler
- Anger resurser

- Använder skärmberoende sessioner
- Kör skript vid inloggning
- Återskapar en säkerhetskopierad session

▼ Så här anger du miljövariabler

♦ Om du vill ange systemtäckande miljövariabler skapar du en fil i katalogen `/etc/dt/config/Xsession.d` som anger och exporterar variabeln.

Om du t ex skapar ett exekverbart `sh-` eller `ksh`-skript, `/etc/dt/config/Xsession.d/`*minavar*,

som innehåller:

```
export MINVARIABEL="värde"
```

anges variabeln `MINVARIABEL` i alla användares miljö vid nästa inloggning.

♦ Om du vill ange egna miljövariabler anger du variabeln i *Hemkatalog*`/.dtprofile`.

Exempel:

```
export MINVARIABEL="värde"
```

anger variabeln `MINVARIABEL` i alla användares miljö vid nästa inloggning.

Obs! – Sessionhanteraren läser inte automatiskt filen `.profile` och `.login`. Den kan dock konfigureras att använda dessa filer. Mer information finns i ”Valfritt läsa in och köra skripten .profile och .logint” på sidan 29.

▼ Så här anger du resurser

♦ Om du vill ange systemtäckande resurser lägger du till resurserna i filen `/etc/dt/config/`*språk*`/sys.resources`. (Du kanske först måste skapa filen.)

Obs! – `.dtprofile` stöder bara syntaxerna `/bin/sh` eller `/bin/ksh`.

I `/etc/dt/config/C/sys.resources` kan du t ex ange:

```
AnApplication*resource: värde
```

så anges resursen `AnApplication*resource` i alla användares `RESOURCE_MANAGER`-egenskap vid nästa inloggning.

- Om du vill ange egna resurser lägger du till dem i filen *Hemkatalog*`/.Xdefaults`.

▼ Så här anger du skärmspecifika resurser

Du kan ange skärmspecifika resurser för alla användare i systemet. Användarna kan också ange skärmspecifika resurser för sina egna sessioner. Du kan därför ange resurser beroende på vilken skärm som användaren använder för att logga in till skrivbordet.

- Om du vill ange skärmspecifika resurser för alla användare i systemet skapar du filen `/etc/dt/config/`*språk*`/sys.resources` som anger den skärmspecifika resursen.
- Om du vill ange egna skärmspecifika resurser anger du resursen i *Hemkatalog*`/.Xdefaults`.

Du avgränsar resurser genom att omsluta dem med villkorssatsen `cpp`. Makrot `DISPLAY_`*skärmnamn* definieras beroende på värdet i variabeln `$DISPLAY`. Detta görs genom att alla punkt- och kolontecken omvandlas till understrykning (_), alla skärmspecifikationer rensas och slutligen läggs `DISPLAY_` till som prefix till resultatet.

Exempelvis `$DISPLAY` med `:0` visar `DISPLAY_0`, och `$DISPLAY` med `blanco.gato.com:0.0` visar `DISPLAY_blanco_gato_com_0`. Det resulterande värdet kan användas som en del i ett `cpp`-test i en sessionsresursfil. Om du i t ex `/etc/dt/config/C/sys.resources` anger:

```
Myapp*resource: value

#ifdef DISPLAY_blanco_gato_com_0
Myapp*resource: specialvalue1
#endif

#ifdef DISPLAY_pablo_gato_com_0
Myapp*resource: specialvalue2
#endif
```

anges resursen `MyApp*resource` i `RESOURCE_MANAGER` till `specialvalue1` när användaren loggar in till skärmen `blanco.gato.com:0`, `specialvalue2` när användaren loggar in till `pablo.gato.com:0` och `value` när användaren loggar in till en annan skärm.

▼ Så här ändrar du program för startsessionen

Du kan ange att andra program ska starta som en del av användarens startsession.

1. Kopiera `/usr/dt/config/`*`språk`*`/sys.session` till `/etc/dt/config/`*`språk`*`/sys.session.`
2. Ändra den nya filen `sys.session`.

   Alla poster i `sys.session` ser ut så här:

   `dtsmcmd -cmd` *`kommando_och_alternativ`*

   Om du vill starta ytterligare ett program som en del av användarens startsession, anger du en ny `sys.session`-post med fullständig sökväg. Om du t ex vill starta `/usr/bin/X11/xclock` som en del av användarens startsession, lägger du till posten `xclock` till `/etc/dt/config/C/sys.resources`:

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

▼ Så här anger du en skärmspecifik session

En användare kan ange en skärmspecifik session för att justera sessionen för en viss skärm.

- Kopiera katalogen *Hemkatalog*`/.dt/sessions` till *Hemkatalog*`/.dt/`*`skärm`* där *skärm* är det verkliga, absoluta värddatornamnet (`pablo:0` är giltigt, `pablo.gato.com:0` och `unix:0` är det inte).

Du kan t ex skapa en skärmspecifik session för skärmen `pablo.gato.com:0`:

`cp -r` *Hemkatalog*`/.dt/sessions` *Hemkatalog*`/.dt/pablo:0`

När användaren loggar in till skärmen `pablo.gato.com:0`, startar sessionshanteraren den skärmspecifika sessionen.

Exekvera ytterligare kommandon vid sessionsstart och utloggning

Användarna kan ange att ytterligare kommandon ska startas när de loggar in till skrivbordssessionen. Detta är användbart vid X-inställningar som inte sparas av sessionshanteraren. Användaren kan t ex använda `xsetroot` för att anpassa rotpekaren (arbetsyta). Ett annat användningsområde är att starta program som inte kan sparas och återställas av sessionshanteraren. Om ett program inte vill starta när sessionen startar, kan användaren starta klienten med den här metoden.

▼ Så här exekverar du ytterligare kommandon vid sessionsstart

♦ Skapa filen *Hemkatalog*`/.dt/sessionetc` som innehåller kommandona.

Normalt är den här filen ett skript och måste ha exekveringsbehörighet. Processer i `sessionetc` ska köras i bakgrunden.

Obs! – Använd inte `sessionetc` för att starta klienter som automatiskt återställs av sessionshanteraren. Om du gör det kan flera kopior av programmet startas. Du kanske inte ser alla kopior omedelbart eftersom fönstren kan staplas på varandra.

▼ Så här exekverar du ytterligare kommandon vid utloggning

`Sessionexit` är en systerfil till `sessionetc`. Använd `sessionexit` för att utföra vissa bearbetningar när sessionen avslutas som inte hanteras av sessionshanteraren.

♦ Skapa filen *Hemkatalog*`/.dt/sessionexit`.

På samma sätt som `sessionetc` är den här filen normalt ett skript med exekveringsbehörighet.

▼ Så här återställer du en session från en säkerhetskopia

När sessionshanteraren sparar en session lagras sessionens information i katalogen *Hemkatalog*`/.dt/sessions`, eller i katalogen *Hemkatalog*`/.dt/`*skärm* om en skärmspecifik session används. I dessa kataloger skapas en underkatalog med namnet `current` eller `home` som lagrar information för

aktiv session eller hemsessionen. Innan sessionsinformationen lagras, skapar sessionshanteraren en säkerhetskopia av tidigare session med det namnet och lagrar den i `current.old` eller `home.old`.

1. Logga in med Session med endast ett fönster eller Kommandradsinloggning från inloggningsbilden.
2. Kopiera katalogen för den säkerhetskopierade sessionen till aktivt namn. Om du t ex vill återskapa den säkerhetskopierade hemsessionen:

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

Skärmspecifika sessioner kan återskapas på samma sätt.

▼ Så här undersöker du problem vid sessionsstart

♦ Kontrollera filen *Hemkatalog*`/.dt/startlog`.

Sessionshanteraren loggar alla användares sessionsstarter i den här filen.

Filer och kataloger i sessionshanteraren

- `/usr/dt/bin/Xsession`
- `/usr/dt/config/Xsession.d/*`
- `/usr/dt/bin/dtsession`
- `/usr/dt/bin/dtsession_res`
- *Hemkatalog*`/.dt/sessions/current`
- *Hemkatalog*`/.dt/sessions/home`
- *Hemkatalog*`/.dt/`*skärm*`/current`
- *Hemkatalog*`/.dt/`*skärm*`/home`

2

Lägga till och administrera program

3

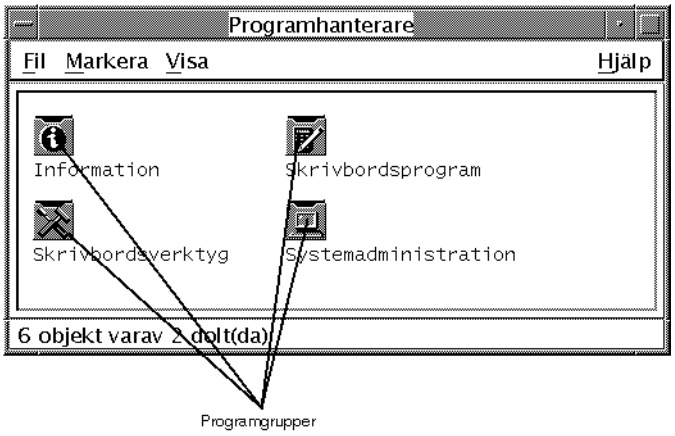
Programhanteraren är skrivbordets behållare för de program som är tillgängliga för användaren.

Programhanterarens struktur	*40*
Lägga till program i programhanteraren	*43*
Skapa och administrera allmänna programgrupper	*45*
Ändra sökvägen som används för att söka efter program	*47*
Allmän programhantering	*48*
Ändra textredigeraren och terminalemulatorn	*49*

3

Programhanterarens struktur

Den översta nivån i programhanteraren innehåller normalt kataloger. Var och en av dessa kataloger, och dess innehåll, kallas en *programgrupp*.



figur 3-1 Programgrupper i programhanteraren

Programgrupperna och deras innehåll samlas in från olika platser i nätverket.

Katalogplacering i programhanteraren

I filsystemet är programhanteraren katalogen `/var/dt/appconfig/appmanager/`*inloggning-värddatornamn-skärm*`.` Katalogen skapas dynamiskt varje gång användaren loggar in.

Om t ex användaren `karinj` loggar in från skärmen `wxyz:0`, skapas katalogen `/var/dt/appconfig/appmanager/karinj-wxyz-0`.

Hur programhanteraren hittar och samlar program

Programhanteraren utgörs av insamlade lokala programgrupper och fjärrprogramgrupper. Programgrupperna samlas in från kataloger i programsökvägen.

Standardprogramsökvägen består av följande adresser:

Omfattning	Adress
Inbyggd	`/usr/dt/appconfig/appmanager/`*språk*
Systemtäckande	`/etc/dt/appconfig/appmanager/`*språk*
Egen	*Hemkatalog*`/.dt/appmanager`

För att skapa den översta nivån i programhanteraren skapas länkar vid inloggningen från programgrupperna (kataloger), som finns i kataloger i programsökvägen, till programhanterarens katalog `/var/dt/appconfig/appmanager/`*inloggning-värddatornamn-skärm*. Insamlingen utförs av skrivbordshjälpmedlet `dtappgather`, som körs automatiskt av inloggningshanteraren efter att användaren har loggat in.

På skrivbordet finns t ex den inbyggda programgruppen:

`/usr/dt/appconfig/appmanager/`*språk*`/Desktop_Tools`

Vid inloggningen skapas en symbolisk länk till:

`/var/dt/appconfig/appmanager/`*inloggning-värddatornamn-skärm*`/Desktop_Tools`

Programsökvägen kan omfatta fjärrkataloger. På detta sätt kan programgrupper från olika system i nätverket samlas. Mer information finns i ”Lägga till en programserver till programsökvägen” på sidan 47.

Prioritetsregler vid insamling av program

När det finns duplikat i sökvägen har egna programgrupper prioritet över systemtäckande grupper, och systemtäckande grupper har prioritet över inbyggda grupper. Om t ex både `/usr/dt/appconfig/appmanager/C/Desktop_Tools` och `/etc/dt/appconfig/appmanager/C/Desktop_Tools` finns, används programgruppen under `/etc` .

Programgrupper på standardskrivbordet

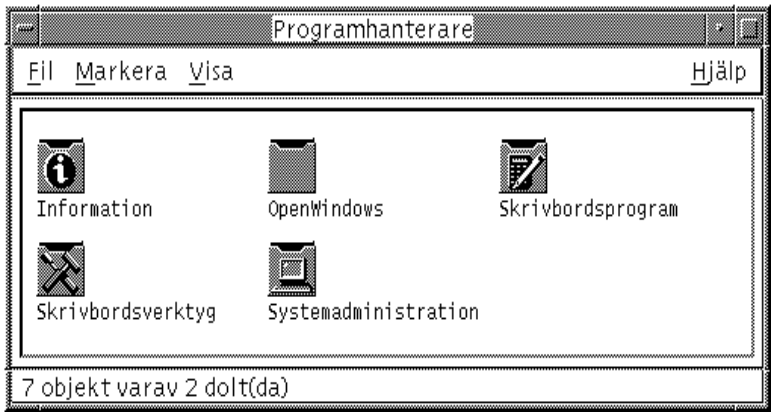
På ett skrivbord som inte är anpassat finns fyra programgrupper.

- Skrivbordsprogram
- Skrivbordsverktyg

- Information
- Systemadmin

Exempel på hur programgrupper samlas in

I figur 3-2 visas ett programhanterarfönster som innehåller olika programgrupper. I tabell 3-1 visas katalogerna som programgrupperna samlas in från.



figur 3-2 Ett typiskt programhanterarfönster

tabell 3-1 Källor till programgrupperna i figur 3-2

Namn	Katalog
Information	`/usr/dt/appconfig/appmanager/C/Information`
OpenWindows	`/usr/dt/appconfig/appmanager/C/OpenWindows`
Skrivbordsprogram	`/usr/dt/appconfig/appmanager/C/Desktop_Apps`
Skrivbordsverktyg	`/usr/dt/appconfig/appmanager/C/Desktop_Tools`
Systemadministration	`/etc/dt/appconfig/appmanager/C/System_Admin`

Om programgruppen Information eller System_Admin har anpassats samlas de i stället in från `/etc/dt/appconfig/appmanager/C`.

Om du vill lägga till ett system till programsökvägen, se ”Lägga till en programserver till programsökvägen” på sidan 47.

Lägga till program i programhanteraren

När ett program har lagts till i programhanteraren kan det startas med en ikon i en programgrupp.

Många program har egna programgrupper. Programgruppen är en katalog på den översta nivån i programhanteraren, och innehåller programikonen och andra filer som hör till programmet.

Vissa program har inga egna programgrupper. I stället finns ikonen i en allmän programgrupp. Du kan t ex skapa en tom programgrupp med namnet ”Spel” som du använder som en behållare för alla spel du installerar i systemet.

Olika sätt att lägga till program i programhanteraren

Det finns två sätt att lägga till program i progamhanteraren:

- Registrera programmet
- Lägga till en programikon utan att registrera programmet

Registrera programmet

Programregistrering ger fullständig programintegration.

Ett registrerat program:

- Har sin egen programgrupp.
- Har sina skrivbordskonfigurationsfiler samlade på ett ställe. Dessa konfigurationsfiler kallas *registreringspaket*.
- Kan ha en registrerad hjälpvolym.

Program kan registreras på två sätt:

- När du installerar ett desktop-smart program sker registreringen automatiskt. Mer information finns i ”Så här lägger du till ett Desktop-Smart program i programhanteraren” på sidan 44.
- Ett befintligt program kan registreras med ett registreringspaket. Mer information finns i ”Så här registrerar du ett befintligt program eller ett program som inte är ett Desktop Smart program” på sidan 44.

Med registreringspaket går det lättare att administrera programmet på skrivbordet. Registreringspaketet skapas någonstans i filsystemet, men inte där skrivbordets konfigurationsfiler finns.

Lägga till program utan registreringspaket

Det här är det bästa sättet att lägga till program när du vill att programhanteraren bara ska innehålla en ikon som startar programmet.

Ett progam som har lagts till utan registreringspaket:

- Kan ha sin egen programgrupp, men ikonen placeras normalt i en befintlig grupp
- Har sina skrivbordskonfigurationsfiler placerade direkt i skrivbordets sökväg.

Mer information finns i ”Så här lägger du till en programikon i en befintlig programgrupp” på sidan 45.

▼ Så här lägger du till ett Desktop-Smart program i programhanteraren

Ett desktop-smart program är ett program som registreras automatiskt i programhanteraren när det installeras. Programmets filuppsättningar omfattar registreringspaketet som behövs för skrivbordet.

1. Installera programmet med hjälp av instruktionerna i programmet.
2. När installationen är klar dubbelklickar du på Ladda om funktioner i programgruppen Skrivbordsverktyg.
3. Kontrollera att installationen är klar:
   a. Öppna programhanteraren och kontrollera att den nya programgruppen finns där.
   b. Öppna programmet genom att öppna programgruppen och dubbelklicka på programikonen.

▼ Så här registrerar du ett befintligt program eller ett program som inte är ett Desktop Smart program

Det här är det bästa sättet att fullständigt integrera program med skrivbordet.

På skrivbordet finns ett verktyg, `dtappintegrate`, som skapar länkar mellan registreringspaketet och katalogerna i skrivbordssökvägen.

Skrivbordsregistrering beskrivs i Kapitel 4, ”Registrera program”.

▼ Så här lägger du till en programikon i en befintlig programgrupp

I den här proceduren beskrivs hur du lägger till en programikon i en befintlig programgrupp.

På skrivbordet finns bl a programgruppen System_Admin som är reserverad för olika program och skript som rör administrationssystem. Om du har ett skript som användarna ofta kör, kanske du vill att de ska kunna köra skriptet genom att dubbelklicka på en ikon i programgruppen System_Admin.

1. Använd Skapa funktion för att skapa en funktionsdefinition för programmet.

   Mer information om Skapa funktion finns i Kapitel 9, ”Skapa funktioner och datatyper med Skapa funktion”.

2. Skapa en exekverbar fil med samma namn som funktionen i programgruppens katalog. Innehållet i filen är irrelevant.

   Om du t ex skapar en funktion med namnet ”Cleanup” som kör ett systemadministrationsverktyg, skapar du den exekverbara filen `/etc/dt/appconfig/appmanager/`*`språk`*`/System_Admin/Cleanup.`

Skapa och administrera allmänna programgrupper

En allmän programgrupp är en programgrupp (katalog) som inte hör till någon viss programprodukt. Den inbyggda programgruppen Skrivbordsverktyg är en allmän programgrupp som innehåller ikoner för ett stort antal program som är likartade, men som inte hör till någon enstaka produkt.

Du kan skapa ytterligare allmänna programgrupper. Du kanske t ex vill skapa en grupp som heter Spel för att samla olika spel i systemet.

En programgrupp kan vara systemtäckande eller egen.

3

▼ Så här skapar du en systemtäckande allmän programgrupp

1. Logga in som rot.
2. Skapa en katalog i `/etc/dt/appconfig/appmanager/`*språk*.

   Programgruppen får samma namn som katalogen.
3. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

▼ Så här skapar du en egen allmän programgrupp

1. Skapa en katalog i *Hemkatalog*`/.dt/appmanager`.

   Programgruppen får samma namn som katalogen.
2. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

▼ Så här anpassar du en inbyggd programgrupp

1. Logga in som rot.
2. Om programgruppen finns i `/usr/dt/appconfig/appmanager/`*språk*, kopierar du programgruppen till `/etc/dt/appconfig/appmanager/`*språk*.

   Följande kommando kopierar t ex programgruppen Skrivbordsverktyg:

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

   Den nya programgruppen gäller före den inbyggda versionen.
3. Ändra kopian av programgruppen. Du kan t ex lägga till nya funktionsfiler (exekverbara filer med samma namn som funktionen).
4. Kontrollera ändringarna genom att logga ut och in igen.

Ändra sökvägen som används för att söka efter program

Det viktigaste skälet för att ändra programsökvägen är att lägga till en programserver. När du lägger till en programserver i sökvägen samlas serverns alla systemtäckande programgrupper.

Mer information om programsökvägen finns i ”Programsökväg” på sidan 116.

Standardsökvägen

Standardprogramsökvägen omfattar följande kataloger:

Omfattning	**Katalog i sökvägen**
Egen	*Hemkatalog*/.dt/appmanager
Systemtäckande	/etc/dt/appconfig/appmanager/*språk*
Inbyggd	/usr/dt/appconfig/appmanager/*språk*

Lägga till en programserver till programsökvägen

Förutom att ändra programsökvägen kan du behöva konfigurera ytterligare för att aktivera kommunikationen med programservern. Mer information finns i ”Administrera programtjänster” på sidan 98 .

▼ *Så här anger du en systemtäckande programsökväg*

1. Logga in som rot.
2. Om filen /etc/dt/config/Xsession.d/0010.dtpaths inte finns skapar du den genom att kopiera /usr/dt/config/Xsession.d/0010.dtpaths.
3. Öppna /etc/dt/Xsession.d/0010.paths för redigering. Lägg till eller redigera en rad som anger variabeln DTSPSYSAPPHOSTS:

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

   I följande rad läggs systemet ApServA till i programsökvägen:

   ```
   export DTSPSYSAPPHOSTS=ApServA:
   ```

4. Informera alla användare i systemet att de måsta logga ut och in igen för att ändringen ska träda i kraft.

▼ *Så här anger du en egen programsökväg*

1. Öppna *Hemkatalog*/.dtprofile för redigering.
2. Lägg till eller redigera en rad som anger variabeln DTSPUSERAPPHOSTS:

   `export DTSPUSERAPPHOSTS=`*värddatornamn*`:[,hostname]`

   I följande rad läggs systemet ApServB och ApServC till i programsökvägen:

   `export DTSPUSERAPPHOSTS=ApServB:,ApServC:`
3. Logga ut och sedan in igen.

Allmän programhantering

Allmän programhantering omfattar att:

- Ta bort program
- Läsa om databasen med program under en session

▼ Så här tar du bort ett program

Om ett program har registrerats med verktyget dtappintegrate, kan du också använda dtappintegrate för att göra det motsatta. När ett program inte längre är registrerat, tas dess programgrupp bort från programhanteraren, samt dess funktioner, datatyper och ikoner. Hjälpen är heller inte längre tillgänglig.

1. Logga in som rot.
2. Kör kommandot:

   `dtappintegrate -s` *progrot* `-u`

▼ Så här uppdaterar du programhanteraren under en session

Du måste återskapa programhanteraren om du lägger till program och vill att dessa ändringar ska träda i kraft omedelbart.

♦ Öppna programgruppen Skrivbordsverktyg och dubbelklicka på Ladda om funktioner.

Ladda om funktioner är användbart för att uppdatera programhanteraren när program läggs till en programserver. Ladda om funktioner upptäcker dock inte program som har tagits bort från en programserver eller program som har flyttats från en position till en annan. Dessa ändringar träder i kraft när användaren loggar ut och sedan in igen.

Ändra textredigeraren och terminalemulatorn

Både textredigeraren och terminalemulatorn kan startas med en ikon på frontpanelen eller genom att du dubbelklickar på en ikon i programhanteraren.

Dessa program startas också av andra skrivbordsaktiviteter.

- Textredigeraren öppnas när användaren markerar textfilen i filhanteraren och väljer Öppna på Markerade-menyn. Standardredigeraren är `dtpad`.
- En terminalemulator körs när användaren väljer Öppna terminal på filhanterarens Fil-meny eller när en funktion öppnar teminalemulatorfönstret. Standardemulatorn är `dtterm`.

Du kan konfigurera skrivbordet så att andra textredigerare eller terminalemulatorer används i dessa situationer.

▼ Så här ändrar du standardtextredigerare och -terminalemulator

1. Logga in som rot om ändringen är systemtäckande.
2. Skapa en funktion för nästa textredigerare eller terminalemulator.

- Du kan använda programmet Skapa funktion. I figur 3-3 visas ett Skapa funktion-fönster som är ifyllt för ett program som heter TextPad. Mer information om Skapa funktion finns i Kapitel 9, ”Skapa funktioner och datatyper med Skapa funktion”.



figur 3-3 Skapa funktion-fönstret

- *Alternativt* kan du skapa en funktionsdefinition manuellt, t ex:

```
ACTION TextPad
{
   LABEL          TextPad
   TYPE           COMMAND
   WINDOW_TYPE    NO_STDIO
   EXEC_STRING    /usr/TP/bin/TextPad %(File)Arg_1%
   DESCRIPTION    Dubbelklicka på den här ikonen för att \
                  starta programmet TextPad.
}
```

  Mer information om hur du manuellt skapar en funktionsdefinition finns i Kapitel 10, ”Skapa funktioner manuellt”.

3. Placera konfigurationsfilen som innehåller den nya funktionen i rätt katalog:
   - Systemtäckande: `/etc/dt/appconfig/types/`*språk*
   - Egen: *Hemkatalog*`/.dt/types`
4. Om den inte redan finns, skapar du lämplig `user-prefs.dt`-fil genom att kopiera `/usr/dt/appconfig/types/`*språk*`/user-prefs.dt` till:
   - Systemtäckande: katalogen `/etc/dt/appconfig/types/`*språk*
   - Egen: katalogen *Hemkatalog*`/.dt/types`
5. Redigera funktionen TextEditor eller Terminal i den systemtäckande eller egna filen `user-prefs.dt`. Ändra raden `MAP_ACTION` till att tilldela funktionen till den nya funktionen.

   Ändra t ex raden:

   ```
   MAP_ACTION Dtpad
   ```

   till:

   ```
   MAP_ACTION TxtPd
   ```
6. Spara filen `user-prefs.dt`.
7. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg och ladda funktionsdatabasen.

3

Registrera program 4

I det här kapitlet beskrivs hur du skapar ett registreringspaket för ett program och hur du registrerar programmet på skrivbordet.

Översikt över programregistrering	*54*
Allmänna steg för att registrera program	*57*
Steg 1: Ändra font- och färgresurser	*58*
Steg 2: Skapa skrivbordets rot	*59*
Steg 3: Skapa kataloger för registreringspaket	*60*
Steg 4: Skapa funktioner och datatyper för programmet	*62*
Steg 5: Placera hjälpfiler i registreringspaket	*66*
Steg 6: Skapa programikoner	*67*
Steg 7: Skapa programgruppen	*68*
Steg 8: Registrera programmet med dtappintegrate	*75*
Exempel på hur du skapar ett registreringspaket	*77*

4

När ett program är fullständigt registrerat på skrivbordet har det:

- En egen programgrupp på den översta nivån i programhanteraren
- En funktion som startar programmet. Funktionen representeras av en ikon i programgruppen
- Eventuellt datatyper för sina datafiler

Registreringen är en icke destruktiv operation för programmet:

- Den ändrar inte programmets exekverbara filer. Därför kan du registrera befintliga program i systemet.
- Den kräver inte att någon av programmets inbyggda filer (t ex exekverbara och `app-defaults`) ska flyttas.
- Den kan lätt ångras. Verktyget `dtappintegrate` som används för att registrera program, omfattar även ett kommadoradsalternativ för den motsatta processen.

Du skapar registreringspaket om du är:

- En systemadministratör som vill registrera ett befintligt program på skrivbordet
- En programmerare som vill skapa ett installationspaket för ett desktop-smart program

Översikt över programregistrering

I det här avsnittet beskrivs:

- Syftet med programregistrering
- Funktioner som tillförs programmet genom programregistrering

---

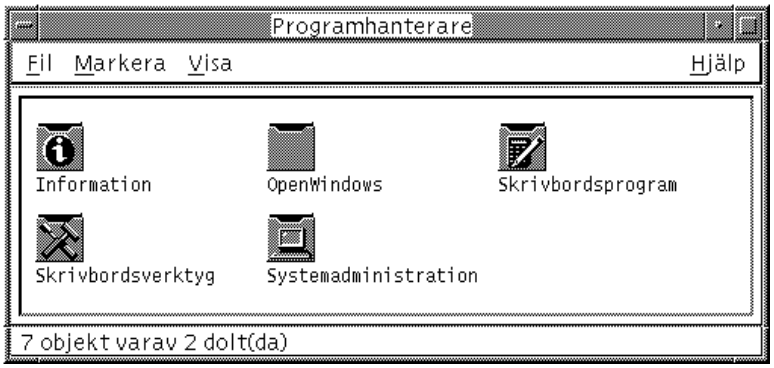
Obs! – Ett detaljerat exempel på hur du registrerar ett befintigt program finns i ”Exempel på hur du skapar ett registreringspaket” på sidan 77.

Funktioner som tillförs genom programregistrering

Om ett program registreras får användaren ett grafiskt sätt att:

- Hitta programmet.

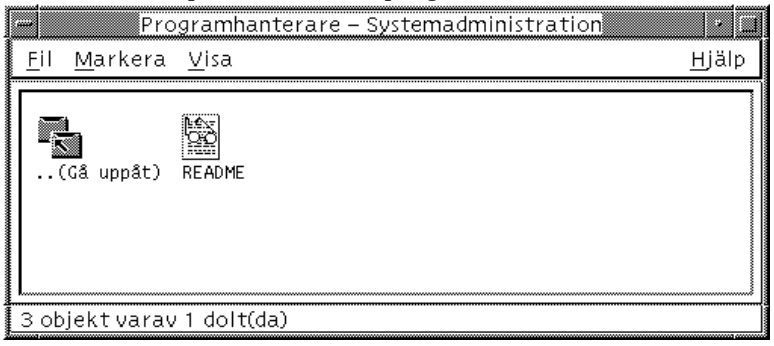
Vid installationen registreras programmet i programhanteraren och får sin egen programgrupp.



figur 4-1 Programgrupper på den översta nivån i programhanteraren

- Starta programmet.

  Programmets programgrupp innehåller en ikon som användaren dubbelklickar på för att starta programmet.

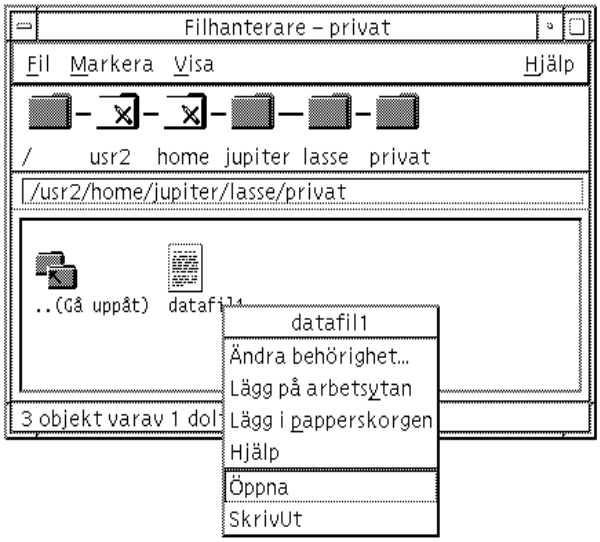


figur 4-2 En programgrupp innehåller en ikon som startar programmet

- Identifiera och manipulera datafiler. Programmets datafiler får en unik ikon i filhanteraren.

Användaren använder datafilsikoner för att:

- Starta (öppna) programmet
- Skriva ut datafiler



figur 4-3 En datafils meny innehåller ”Öppna” och ”SkrivUt”

- Utföra andra operationer, t ex skicka, komprimera, visa och spela upp (ljud-)data

Syftet med programregistrering

Ett registrerat skrivbordsprogram har vissa konfigurationsfiler som skrivbordet använder för att skapa programmets användargränssnitt:

- Filer med funktions- och datatypsdefinitioner
- Filer med ikonbilder (bildpunktsmönster och bitmönster)
- En katalog och filer som skapar programgruppen
- Eventuella hjälpfiler och definitionsfiler till frontpanelen

För att skrivbordet ska kunna känna igen och använda dessa filer måste de finnas i vissa kataloger som anges i skrivbordets sökvägar.

Det kan vara svårt att administrera ett program när konfigurationsfilerna har spridits ut på en mängd kataloger. Därför kan programmet ha alla konfigurationsfiler samlade i en enda katalog. Den här grupperingen av filer kallas ett *registreringspaket*.

Om programmet är ett desktop smart-program ingår registreringspaketet som en del av installationspaketet. Om du är en systemadministratör som skapar konfigurationsfilerna själv, kan du också skapa registreringspaketet.

Konfigurationsfilerna i registreringspaketet är inte tillgängliga för skrivbordet eftersom de inte finns i rätt sökvägskatalog. När filerna placeras på rätt plats kallas detta *registrering* eller *integrering* av programmet.

På skrivbordet finns ett verktyg, `dtappintegrate`, som utför registreringen genom att skapa symboliskt länkade representationer av filerna i rätt sökvägskataloger.

Många desktop-smart program kör `dtappintegrate` automatiskt under installationen. Om du är en systemadministratör som integrerar ett befintligt program, kan du köra det själv efter att du har skapat registreringspaketet.

När programmet är registrerat på systemets skrivbord är det tillgängligt för alla användare i systemet. Om systemet är konfigurerat som en skrivbordsprogramserver är programmet också tillgängligt för andra system via nätverket.

Verktyget `dtappintegrate` har ett kommandoradsalternativ som utför den motsatta processen genom att länkarna bryts. På detta sätt är det enkelt att ta bort programmet från programhanteraren så att det kan flyttas till en annan programserver eller uppdateras.

Allmänna steg för att registrera program

Obs! – Ett detaljerat exempel med dessa steg för att skapa ett programpaket finns i ”Exempel på hur du skapar ett registreringspaket” på sidan 77.

1. Ändra alla programresurser som anger fonter och färger. Om du inte gör det fungerar inte skrivbordets dynamiska fonter och färger som de ska.

   Se ”Steg 1: Ändra font- och färgresurser” på sidan 58.

2. Skapa en rotadress för programmet.

   Se ”Steg 2: Skapa skrivbordets rot” på sidan 59.

3. Skapa katalogstrukturen under programroten.

   Se ”Steg 3: Skapa kataloger för registreringspaket” på sidan 60

4. Skapa programmets funktioner och datatyper.

   Se ”Steg 4: Skapa funktioner och datatyper för programmet” på sidan 62.

5. Placera hjälpfilerna i rätt katalog.

   Se ”Steg 5: Placera hjälpfiler i registreringspaket” på sidan 66.

6. Skapa programikonerna

   Se ”Steg 6: Skapa programikoner” på sidan 67.

7. Skapa programmets programgrupp.

   Se ”Steg 7: Skapa programgruppen” på sidan 68.

8. Registrera programmet med `dtappintegrate`.

   Se ”Steg 8: Registrera programmet med dtappintegrate” på sidan 75.

Steg 1: Ändra font- och färgresurser

Obs! – Ett exempel på hur du ändrar resurser för ett program finns i Steg 1 i “Exempel på hur du skapar ett registreringspaket” på sidan 78.

På skrivbordet finns ett verktyg för att ange och manipulera gränssnittets fonter och fönsterfärger. För att ett program ska kunna använda detta verktyg på rätt sätt måste du ändra programmets `app-defaults`-fil.

Ändra fontresurser

Obs! – Det här avsnittet gäller för program som skapats med OSF/Motif 1.2™ (eller senare version). Arbetsinställningar kan inte användas för att ange gränssnittsfonter för program som skrivits i tidigare versioner av OSF/Motif.

Skrivbordets Arbetsinställningar anger gränssnittsfonter för program som skapats med OSF/Motif 1.2 (eller senare version) om programmet inte har programspecifika gränssnittsfonter.

Arbetsinställningar innehåller två fonter:

systemfont	Används i systemområden, t ex etiketter, menyer och knappar
användarfont	Används till redigerbara områden, t ex textfält

Varje font finns i sju storlekar, rubricerade 1 t o m 7 i dialogrutan Fonter. Fonterna i Arbetsinställningar är anslutna till verkliga fonter i systemet via Arbetsinställningars resurser som anges i `/usr/dt/app-defaults/`*`språk`*`/Dtstyle`.

Om du vill att programmet ska använda fonterna i Arbetsinställningar tar du bort alla programresurser som anger gränssnittsfonter. Skrivbordet anger då programmets resurser på rätt sätt:

`FontList`	Anges till systemfont
`XmText*FontList`	Anges till användarfont
`XmTextField*FontList`	Anges till användarfont

Ändra färgresurser

Med Arbetsinställningar ändrar du programmets färger dynamiskt. Programmet måste vara en OSF/Motif 1.1- eller 1.2-klient. Klienter som skrivits med andra verktyg kan inte ändra färg dynamiskt. Färgerna träder i kraft när klienten startas om.

Det lättaste sättet att använda de dynamiska färgerna som finns på skrivbordet är att ta bort alla programfärgresurser för bakgrunds- och förgrundsfärg.

Steg 2: Skapa skrivbordets rot

Obs! – Ett exempel på hur du skapar skrivbordets rotkatalog till ett program finns i Steg 2 i “Exempel på hur du skapar ett registreringspaket” på sidan 78.

Filerna i programmets registreringspaket grupperas under en katalog som kallas programrot, eller *progrot*. Katalogen *progrot*, som används till skrivbordskonfigurationsfiler, kan vara samma katalog som programmets installations-*progrot* eller finnas någon annanstans.

Anta att ett program installeras under katalogen `/usr/BTE`. Denna katalog kan också användas som *progrot* för skrivbordskonfigurationsfilerna. Om du integrerar ett program som inte är ett desktop smart program, rekommenderas det dock att du skapar en annan katalog för *progrot*. Detta hindrar konfigurationsfilerna du skapar från att skrivas över när programmet uppdateras.

En systemadministratör kan t ex skapa katalogen `/etc/desktop_approots/BTE` som katalogen för *progrot*.

Steg 3: Skapa kataloger för registreringspaket

Obs! – Ett exempel på hur du skapar kataloger för registreringspaket finns i Steg 3 i “Exempel på hur du skapar ett registreringspaket” på sidan 79.

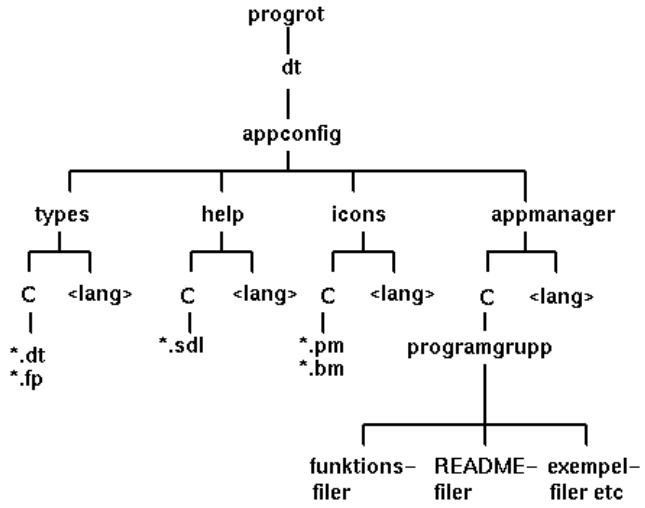
Registreringspaket är en grupp skrivbordskonfigurationsfiler som skrivbordet använder för att ge programmet ett grafiskt gränssnitt.

Innehåll i registreringspaket

Skrivbordskonfigurationsfilerna omfattar:

- Filer med funktions- och datatypsdefinitioner
- Filer med ikonbilder
- En programgruppskatalog och dess innehåll
- Eventuella hjälpdatafiler och en konfigurationsfil för frontpanelen

Registreringspaket samlas under katalogen på högsta nivån, kallad programroten eller *progrot*.



figur 4-4 Ett registreringspaket under katalogen för programroten

Huvudkategorierna med konfigurationfält under katalogen *progrot*`/dt/appconfig` är:

Underkatalog	**Innehåll**
`types`	Filer med funktions- och datatypsdefinitioner
`help`	Hjälpfiler
`icons`	Bitmönster- och bildpunktsmönsterfiler som används av programmets funktioner och datatyper
`appmanager`	Katalogen och innehållet som utgör programgruppen

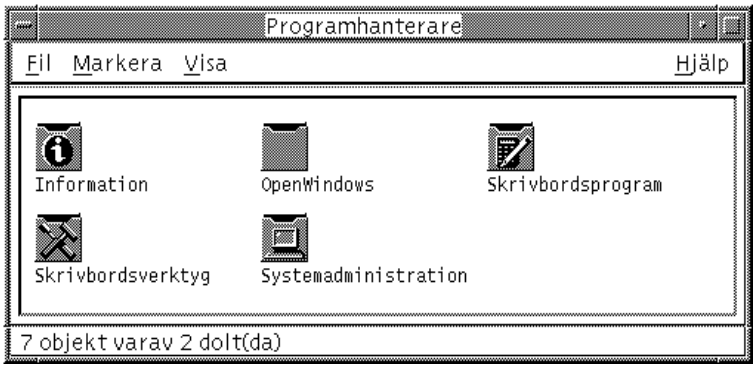
Alla huvudkategorier har underkategorier för språkberoende filer. Standardspråket placeras i katalogen `C`.

Skapa registreringspaket

♦ Skapa följande kataloger. Om du vill använda språkberoende filer, skapar du en separat katalog för varje språk. Om du endast vill använda ett språk placerar du filerna i katalogen `C`.

- *progrot*`/dt/appconfig/types/`*språk*
- *progrot*`/dt/appconfig/help/`*språk*
- *progrot*`/dt/appconfig/icons/`*språk*
- *progrot*`/dt/appconfig/appmanager/`*språk/programgrupp*, där *programgrupp* är namnet på programgruppen.

I figur 4-5 visas en programhanterare som innehåller en grupp vars *programgrupp*-namn är ”Skrivbordsverktyg”.



figur 4-5 Programgrupp på den översta nivån i programhanteraren

Verktyget `dtappintegrate` arbetar bara med skrivbordskonfigurationsfilerna i katalogerna `types`, `help`, `icons` och `appmanager`. Programmets binära exekverbara fil, `app-defaults` och meddelandekatalogfiler administreras separat.

Steg 4: Skapa funktioner och datatyper för programmet

Obs! – Ett exempel på hur du skapar funktioner och datatyper för ett program finns i Steg 4 i “Exempel på hur du skapar ett registreringspaket” på sidan 79.

Funktioner och datatyper ger programmet ett användargränssnitt.

- Funktioner tillhandahåller ett användargränssnitt för kommandot som startar programmet.
- Datatyper tillhandahåller anpassade utseenden och metoder för programmets datafiler.

Funktioner och datatyper som är nödvändiga för ett program

För typiska program är följande funktions- och datatypsdefinitioner nödvändiga:

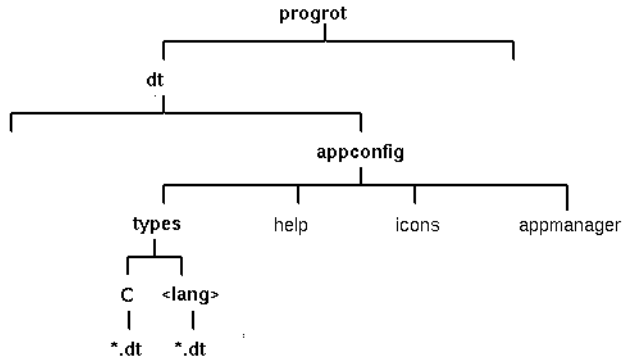
- En funktion som öppnar programmet.
- En datatyp för programmets datafiler. Om du skapar en datatyp måste du också skapa:
  - Funktionen Open för datafilerna i programmet
  - Funktionen Print för datafilerna i programmet
- En datatyp för programgruppen (se ”Konfigurera programgruppen att använda en unik ikon” på sidan 70).

En introduktion till hur funktioner och datatyper används på skrivbordet finns i Kapitel 8, ”Introduktion till funktioner och datatyper”.

Adresser för konfigurationsfiler till funktions- och datatypsdefinitioner

Funktioner och datatyper definieras i konfigurationsfiler. Det enda namnkravet på filer som innehåller funktions- och datatypsdefinitioner är att de måste ha ett `dt`-suffix. Därför ska du namnge filerna med *funktion*`.dt` eller *program*`.dt`.

Placerera filerna med funktioner och datatyper under roten i katalogen *progrot*`/dt/appconfig/types/`*språk*. Standardkatalogen för *språk* är `C`.



figur 4-6 Filer med funktions- och datatypsdefinitioner

Olika sätt att skapa funktioner och datatyper

Du kan skapa funktions- och datatypsdefinitioner för ett program på två sätt:

- Använda verktyget Skapa funktion.

  Skapa funktion har ett lättanvänt gränssnitt med textfält som du fyller i. Verktyget har dock vissa begränsningar.

- Skapa definitioner manuellt.

  Detta kräver att du måste lära dig syntaxen till definitionerna, men ger tillgång till fullständig funktionalitet.

▼ Så här skapar du funktioner och datatyper med Skapa funktion

I den här proceduren används Skapa funktion för att skapa en funktion och datatyper för programmet.

Mer information om Skapa funktion finns i direkthjälpen och i Kapitel 9, ”Skapa funktioner och datatyper med Skapa funktion”.

1. Öppna programgruppen Skrivbordsprogram och dubbelklicka på Skapa funktion.
2. Använd Skapa funktion för att skapa funktions- och datatypsdefinitionerna för programmet och dess datatyp.

Konfigurationfilerna som skapas av Skapa funktion skrivs till *Hemkatalog*/.dt/type/*funktion*.dt. Funktionsfilen (den exekverbara filen med samma namn som funktionen) placeras i hemkatalogen.

3. Testa funktionen med funktionsfilen som skapats i hemkatalogen.
4. Kopiera funktionsdefinitionsfilen *Hemkatalog*/.dt/type/*funktion*.dt till katalogen *progrot*/dt/appconfig/types/*språk*.
5. När programgruppskatalogen har skapats (se avsnittet, ”Steg 7: Skapa programgruppen,” på sidan 68) kopierar du funktionsfilen *Hemkatalog/funktion* till katalogen *progrot*/dt/appconfig/appmanager/*språk*/*programgrupp*.

▼ Så här skapar du funktioner och datatyper manuellt

♦ Skapa en konfigurationsfil som innehåller funktions- och datatypsdefinitionerna för programmet.

Funktions- och datatypsdefinitionsfiler måste följa namngivningskonventionen *namn*.dt.

Du kan placera alla funktions- och datatypsdefinitioner i en fil eller distribuera dem mellan flera filer. För varje fil måste du använda ett filnamn som systemadministratören lätt kan koppla till ditt program.

Funktions- och datatypsnamn måste vara ett ord (inga inbäddade mellanslag), men kan innehålla understreck. Enligt konventionen är den första bokstaven i funktions- och datatypsnamnet versalt. Använd inte ett befintligt funktions- eller datatypsnamn. Använd namn som avancerade användare och systemadministratörer lätt kan koppla till ditt program.

Om du vill att programikonens namn ska vara ett annat än namnet på funktionen, inkluderar du fältet LABEL i funktionens definition.

Mer information om hur du skapar funktioner och datatyper finns i:

- Kapitel 8, ”Introduktion till funktioner och datatyper”
- Kapitel 9, ”Skapa funktioner och datatyper med Skapa funktion”
- Kapitel 10, ”Skapa funktioner manuellt”
- Kapitel 11, ”Skapa datatyper manuellt”

4

Steg 5: Placera hjälpfiler i registreringspaket

Obs! – Ett exempel på hur du lägger till hjälpfiler i ett registreringspaket finns i Steg 5 i “Exempel på hur du skapar ett registreringspaket” på sidan 81.

Om programmet omfattar en hjälpvolym för skrivbordet (en hjälpvolym skapas med skrivbordets Help Developer’s Kit), ska hjälpvolymens masterfil (`*.sdl`) placeras i katalogen *progrot*`/appconfig/help/`*språk*.

Grafik som används av hjälpfilerna placeras normalt i underkatalogen `graphics`. Grafiken måste placeras i samma katalog i förhållande till masterfilen för hjälpvolymen (`*.sdl`) när hjälpvolymen skapades.

Om programmet inte har någon hjälpvolym kan du skapa en om du har Help Developer’s Kit.

Det finns två nivåer av integration för en hjälpvolym:

- Fullständig integration.

  När skrivbordshjälpen är fullständigt integrerad kan hjälpvolymen nås från programmet, t ex via objekthjälpen och Hjälp-menyn. Fullständig integration omfattar modifiering av programmets exekverbara filer.

- Delvis integration.

  När skrivbordet är delvis integrerat, är hjälpvolymen tillgänglig från den översta nivån i hjälphanteraren. Du kan dock inte nå hjälpvolymen från programmets fönster. Du kan också använda en funktion för att nå hjälpen från programgruppen. I följande exempelfunktion visas hjälpvolymen som finns i hjälpmasterfilen `MyApp.sdl`:

```
ACTION OpenMyAppHelp
{
  LABEL           MyAppHelp
  ARG_COUNT       0
  TYPE            COMMAND
  WINDOW_TYPE     NO_STDIO
  EXEC_STRING     /usr/dt/bin/dthelpview -helpVolume MyApp
  DESCRIPTION     Visar hjälp för programmet MyApp.
}
```

Steg 6: Skapa programikoner

Obs! – Ett exempel på hur du skapar ikonfiler för ett program finns i Steg 6 i “Exempel på hur du skapar ett registreringspaket” på sidan 82.

På skrivbordet finns standardikoner för funktioner, datatyper och programgrupper. Du kan dock skapa unika ikoner för programmet.

Ikoner placeras i katalogen *progrot*`/dt/appconfig/icons/`*språk*.

Ikoner som behövs för skrivbordet

Programmet använder tre ikonbilder på skrivbordet:

- **Funktionsikon**. På den här ikonen dubbelklickar användaren för att starta programmet (funktionen). Den refereras i fältet `ICON` för funktionen som startar programmet.

  Finns i tre storlekar: mycket liten, normal och stor.

- **Datatypsikon**. Den här ikonen används för att representera programmets datafiler i filhanteraren. Den refereras i fältet `ICON` för datatypsdefinitionen.

  Om programmet stöder flera datatyper ska du ha olika ikoner för varje datatyp.

  Finns i två storlekar: mycket liten och normal.

- **Programgruppsikon**. Den här ikonen representerar katalogen på den översta nivån i programhanteraren. Den refereras i fältet `ICON` för programgruppens datatypsdefinition. (Se ”Steg 7: Skapa programgruppen” på sidan 68.)

  Finns i två storlekar: mycket liten och normal.

Du kanske måste ha både bitmönster- och bildpunktsmönsterversioner av varje ikon för att kunna använda både färgskärm (åtta-bitars och större) och monokrom skärm (färre än åtta bitar).

tabell 4-1 Namngivningskonventioner för ikonfiler

Storlek	Bildpunkts-dimension	Bitmönsternamn	Bildpunkts-mönsternamn
mycket liten	16 gånger 16	*basnamn*`.t.bm`	*basnamn*`.t.pm`
normal	32 gånger 32	*basnamn*`.m.bm`	*basnamn*`.m.pm`
stor	48 gånger 48	*basnamn*`.l.bm`	*basnamn*`.l.pm`

Om du inte använder bitmönsterfiler, tilldelar skrivbordet färgspecifikationerna för bildpunktsmönsterfilerna till svart och vitt. Den här tilldelningen kanske inte ger det utseende som du vill ha.

Mer information om ikoner finns i ”Ikonbildsfiler” på sidan 203.

Steg 7: Skapa programgruppen

Obs! – Ett exempel på hur du skapar programgruppen finns i Steg 7 i “Exempel på hur du skapar ett registreringspaket” på sidan 82.

När du har skapat funktions- och datatypsdefinitioner för programmet, måste du skapa konfigurationsfilerna som skapar det som användaren ser - programgruppen och dess innehåll.

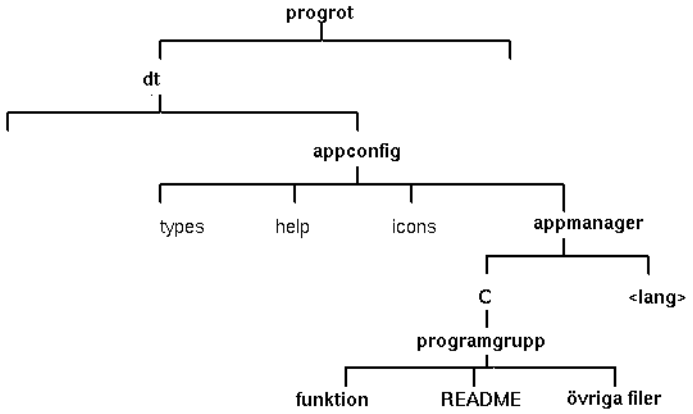
Programgruppen är en katalog på den översta nivån i programhanteraren (se figur 4-1 på sidan 55).

Du skapar programgruppen i tre steg:

- Skapa programgruppskatalogen i registreringspaketet.
- Valfritt: Konfigurera programgruppen att använda en unik ikon. Detta omfattar att skapa datatypsdefinitionen för programgruppskatalogen.
- Skapa innehållet i programgruppen.

Skapa programgruppskatalogen

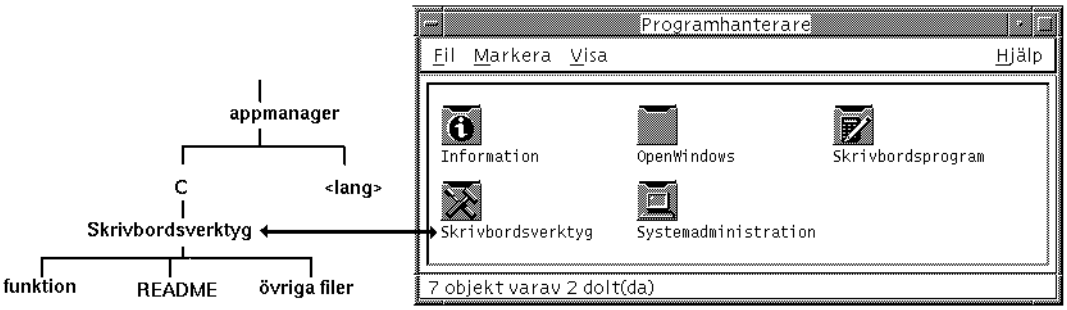
Om du vill skapa en programgrupp skapar du katalogerna i registreringspaketet under appmanager, vilket visas i figur 4-7.



figur 4-7 Katalogen appmanager

Programgruppsnamn

Programgrupp i figur 4-7 är namnet på en programgrupp .



figur 4-8 Namnet på programgruppen.

Namnet kan vara valfritt giltigt filnamn (katalognamn). Använd ett namn som beskriver programmet.

Konfigurera programgruppen att använda en unik ikon

Till skrivbordet finns standardikoner för programgrupper. Du kan dock skapa egna, anpassade ikoner.

Om du vill använda en unik ikon för programgruppen måste du skapa:

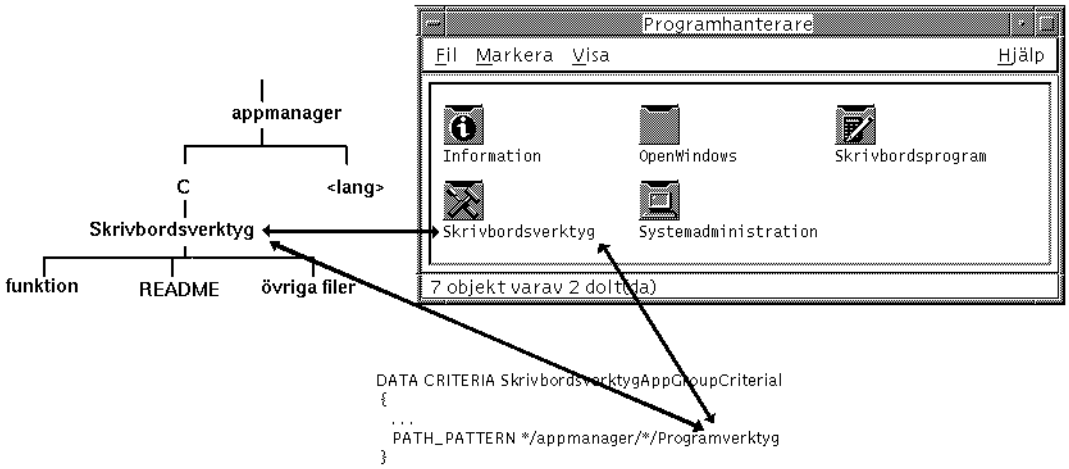
- En datatyp för katalogen som visas på den översta nivån i programhanteraren.
- Funktionerna Open och Print för datatypen.

Anta att du vill skapa programgruppen Skrivbordsverktyg. Följande datatypsdefinition, placerad i filen *progrot*/dt/appconfig/types/*språk*/*namn*.dt, tilldelar en unik ikon till programgruppen.

```
DATA_ATTRIBUTES  Desktop_ToolsAppgroup
{
   ACTIONS       OpenInPlace,OpenNewView
   ICON          Desktop_Tools
   DESCRIPTION   Dubbelklicka för att öppna programgruppen \
                 Skrivbordsverktyg
}
DATA_CRITERIA           Desktop_ToolsAppgroupCriteria1
{
   DATA_ATTRIBUTES_NAME Desktop_ToolsAppgroup
   MODE                 d
   PATH_PATTERN         */appmanager/*/Desktop_Tools
}
```

Attributavsnittet i definitionen anger vilken ikon som ska användas. Villkorsavsnittet anger att datatypen ska definieras för alla kataloger som heter Skrivbordsverktyg och är underkataloger till en katalog som heter appmanager.

I figur 4-9 visas förhållandet mellan programgruppsnamnet och datatypsdefinitionen. Fältet PATH_PATTERN i datatypsdefinitionen kopplar en unik ikon till programgruppen.



figur 4-9 Hur programgruppen får en unik ikon

Du måste också skapa funktionen Open och Print för programgruppens datatyp:

```
ACTION Open
{
    ARG_TYPE    Desktop_ToolsAppGroup
    TYPE        MAP
    MAP_ACTION  OpenAppGroup
}
ACTION Print
{
    ARG_TYPE    Desktop_ToolsAppGroup
    TYPE        MAP
    MAP_ACTION  PrintAppGroup
}
```

Funktionerna OpenAppGroup och PrintAppGroup är inbyggda funktioner som definieras i `/usr/dt/appconfig/types/`*`språk`*`/dtappman.dt`.

Skapa innehållet i programgruppen

Det viktigaste alternativet i programgruppen är den ikon som startar programmet (en funktionsikon). Om programgruppen innehåller flera program finns det normalt en ikon för varje program.

Förutom en eller flera funktionsikoner kan programgruppen innehålla:

- En eller flera README-filer
- En eller flera exempeldatafiler
- Mallar
- En ikon som användaren klickar på för att få hjälp
- En kommandobeskrivning
- En specialiserad frontpanel

Programgruppen kan innehålla underkataloger.

Skapa funktionsfilen (programikon)

Programgruppen ska innehålla en ikon som startar programmet. Om gruppen innehåller flera program, ska det finnas en ikon för varje program. Dessa ikoner kallas *programikoner* eller *funktionsikoner,* eftersom de representerar en underliggande funktion.

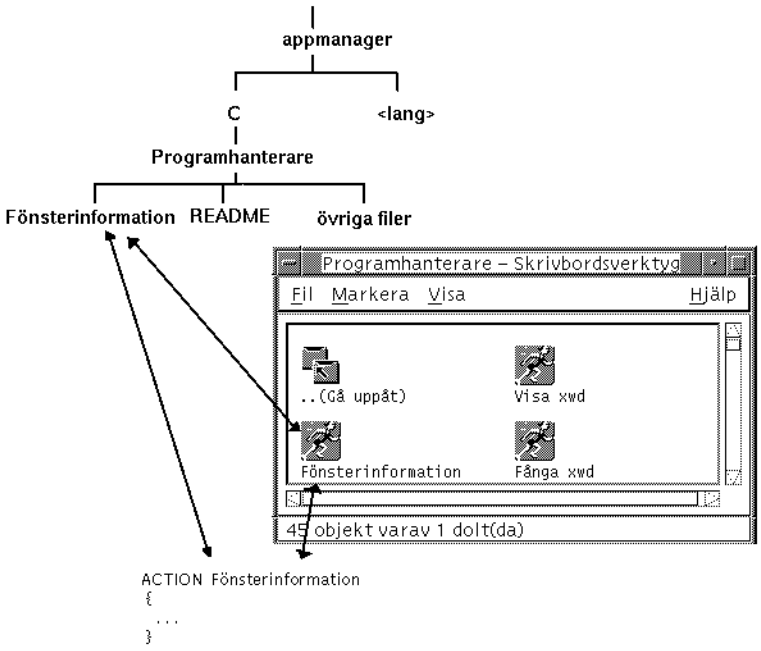
En funktionsikon skapas genom att du skapar en exekverbar fil med samma namn som funtionen den ska köra:

progrot`/dt/appconfig/appmanager/`*programgrupp/funktion*

Filen kallas en *funktionsfil* eftersom syftet med den är att skapa en visuell representation av underliggande funktion.

Om du t ex har skapat en funktion med namnet BestTextEditor som kör programmet BestTextEditor, ska du skapa en exekverbar fil som heter BestTextEditor. I filhanteraren och programhanteraren använder funktionsfilen den ikonbild som anges i funktionsdefinitionen.

I figur 4-10 visas förhållandet mellan funktionsdefinitionen, funktionsfilen och vad som visas i programhanteraren.



figur 4-10 Programikonen är en fil i programgruppen

Read Me-filer

Till skrivbordet finns datatypen README som du använder till programmets README-filer. Använd en av följande namngivningskonventioner:

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

Skapa en specialiserad frontpanel

I de flesta fall behöver du inte ange en frontpanelsdefinition. Användaren kan lägga till programmet på frontpanelen genom att släppa funktionsikonen på Installera ikon på en panelundermeny.

Du kan skapa en konfigurationfil för frontpanelen som innehåller ikondefinitioner för programmet om du vill att användarna ska kunna installera en ikon som uppträder annorlunda än funktionsikonen. Ikonen kan t ex övervaka en fil och ändra utseende när den övervakade filen ändras.

Konfigurationsfilerna för frontpanelen placeras i katalogen *progrot*`/dt/appconfig/types/`*språk*. Namngivningskonventionen är *namn*`.fp`.

Om du skapar en konfigurationsfil som innehåller en ikon, kan användaren lägga till ikonen på en panelundermeny genom att släppa filen `*.fp` på Installera ikon på panelundermenyn.

Följande definition kan t ex placeras i frontpanelens konfigurationsfil i programgruppen. Om användaren släpper den här filen på Installera ikon på en panelundermeny, skapas en ikon på panelundermenyn som kör en förekomst av programmet BestTextEditor. Om BestTextEditor redan körs, flyttas fönstret överst bland fönstren i aktiv arbetsyta.

```
CONTROL BestTextEditorControl
{
    TYPE           icon
    ICON           BTEFPanel
    PUSH_RECALL    True
    CLIENT_NAME    BTEd
    PUSH_ACTION    BTEditor
    DROP_ACTION    BTEditor
    HELP_STRING    Startar programmet BestTextEditor.
}
```

Mer information om hur du skapar konfigurationsfiler till frontpanelen finns i:

- Kapitel 13, ”Särskild anpassning av frontpanelen”
- Kommandobeskrivningen för `dtfpfile(4)`

Steg 8: Registrera programmet med dtappintegrate

Obs! – Ett exempel på hur du registrerar ett program finns i Steg 8 i “Exempel på hur du skapar ett registreringspaket” på sidan 84.

När du har skapat ett registreringspaket under programroten är du klar att utföra själva programregistreringen.

Programregistreringen skapar länkar mellan registreringspaketet och de kataloger som finns i skrivbordssökvägarna (se ”Hur dtappintegrate integrerar program” på sidan 76).

▼ Så här registrerar du ett program med dtappintegrate

Om programmet är ett desktop smart-program, körs `dtappintegrate` normalt automatiskt som det sista steget i installationen. Om det inte körs automatiskt, eller om du har skapat konfigurationsfiler för att integrera ett program som inte är ett desktop smart-program, kör du `dtappintegrate` manuellt.

1. Logga in som rot.
2. Kör kommandot:

   `/usr/dt/bin/dtappintegrate -s` *progrot*

   där *progrot* är skrivbordets rotkatalog. Mer information finns i kommandobeskrivningen till `dtappintegrate(1)`.
3. Öppna programgruppen Skrivbordsverktyg och dubbelklicka på Ladda om program.
4. Kontrollera att programmet är korrekt registrerat:
   a. Visa den översta nivån i programhanteraren. Den nya programgruppen ska visas i programhanteraren.
   b. Öppna programgruppen och dubbelklicka på funktionsikonen.

Syntax och alternativ för dtappintegrate

`dtappintegrate -s` *progrot* `[-t` *målsökväg* `] [-l` *språk* `] [-u]`

`-s` *progrot*	Obligatorisk parameter. Anger programroten under vilken programmet har installerats.
`-t` *målsökväg*	Valfri parameter, där standardvärdet är systemadressen `/etc/dt/appconfig`. Anger den adress som skrivbordskonfigurationsfilerna är länkade till. Du måste använda en adress i programsökvägen.
`-l språk`	Valfri parameter som använder standardvärdet för alla språk. Anger vilka språkberoende skrivbordskonfigurationsfiler som ska integreras.
`-u`	Valfri parameter. Löser upp programmet och tar bort alla länkar som upprättats under integrationen.

Hur dtappintegrate integrerar program

Uppgiften för `dtappintegrate` är att upprätta länkar mellan de installerade filerna och de adresser som skrivbordet söker efter konfigurationsfiler på.

Funktioner och datatyper

`dtappintegrate` skapar symboliska länkar från funktions- och datatypdefinitionsfilerna i registreringspaketet till den systemtäckande katalogen i sökvägen för aktionsdatabasens hjälp. Detta görs genom att länkar skapas från

progrot`/dt/appconfig/types/`*språk*`/*.dt`

till

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

Hjälpfiler

`dtappintegrate` skapar symboliska länkar från hjälpfilerna i registreringspaketet till den systemtäckande katalogen i hjälpsökvägen. Detta görs genom att länkar skapas från

progrot`/dt/appconfig/help/`*språk/hjälpfil*`.sdl`

till

`/etc/dt/appconfig/help/`*språk/hjälpfil*`.sdl`

Ikonfiler

`dtappintegrate` skapar symboliska länkar från ikonfilerna i registreringspaketet till den systemtäckande katalogen i ikonsökvägen. Detta görs genom att länkar skapas från

progrot`/dt/appconfig/icons/`*språk/ikonfiler*

till

`/etc/dt/appconfig/icons/`*språk/ikonfiler*

Programgrupp

För att placera programmets programgrupp i den översta nivån i programhanteraren, skapar `dtappintegrate` en länk mellan programgruppskatalogen i registreringspaketet och den systemtäckande adressen i programsökvägen. Detta görs genom att länkar skapas från katalogen

progrot`/dt/appconfig/appmanager/`*språk/programgrupp*

till

`/etc/dt/appconfig/appmanager/`*språk/programgrupp*

Exempel på hur du skapar ett registreringspaket

I följande steg skapas ett registreringspaket för ett befintligt program som heter BestTextEditor och som inte är ett desktop smart program.

Information du behöver om "BestTextEditor"

I exemplet förutsätts följande fakta om programmet BestTextEditor:

- Det har installerats i katalogen `/usr/BTE`.
- Användarens sessionsspråk är standardvärdet, `C`.
- Kommandoraden som startar BestTextEditor är:

  `BTEd {`*filnamn*`]`

  där *filnamn* är namnet på datafilen som ska öppnas i det nya fönstret. BestTextEditor skapar ett eget fönster, dvs det körs inte inuti terminalemulatorfönstret.
- BestTextEditor skapar och använder två typer av datafiler:

- Dokumentationsfiler. De använder namngivningskonventionen `*.bte`. I BestTextEditor finns en kommandorad för att skriva ut `.bte`-datafiler. Syntaxen för detta kommando är:

  `BTEPrint [-d` *mål*`] [-s]` *filnamn*

  där:

`-d` mål	Anger målskrivaren.
`-s`	Anger tyst utskrift. Dialogrutan Skriv ut i programmet visas inte.
filnamn	Anger vilken fil som ska skrivas ut.

- Mallfiler. Dessa använder namngivningskonventionen `*.tpl`. Mallfiler kan inte skrivas ut.

- De befintliga icke-desktop program-standardfilerna för BestTextEditor innehåller resurser för gränssnittsfonter samt förgrunds- och bakgrundsfärger.

- En hjälpvolym för BestTextEditor har skapats med Help Developer's Kit. När hjälpvolymen skapades användes följande källfiler:

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

  och skapade filen `.../BTEHelp.sdl`.

Steg för att registrera "BestTextEditor"

Följande stegvisa procedur registrerar BestTextEditor.

1. **Ändra font- och färgresurser.**

   I BestTextEditors programstandardfil tar du bort resurserna som anger:
   - Fonter för text.
   - Färger på bakgrund och förgrund.

2. **Skapa programroten.**

   Skapa katalogen:

   `/desktop_approots/BTE`

Om du integrerar ett befintligt program ska du skapa programrotkatalogen någon annanstans än vid programmets installationsadress. I annat fall kan konfigurationsfilerna du skapar tas bort när programmet uppdateras.

3. **Skapa kataloger för registreringspaket**.

   Skapa dessa kataloger:

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

4. **Skapa funktioner och datatyper för programmet.**

   a. Skapa konfigurationsfilen för funktions- och datatypsdefinitioner:

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

   b. Skapa funktionsdefinitionen för att köra BestTextEditor:

   ```
   ACTION BTEditor
   {
      WINDOW_TYPE    NO_STDIO
      ICON           BTERun
      DESCRIPTION    Dubbelklicka på den här ikonen eller släpp\
                     en BTE-datafil på den för att köra
                     BestTextEditor.
      EXEC_STRING    /usr/BTE/BTEd %Arg_1%
   }
   ```

   c. Skapa datatyper för `*.bte`-filerna:

   ```
   DATA_ATTRIBUTES BTEDataFile
   {
      DESCRIPTION    Datafil för BestTextEditor.
      ICON           BTEData
      ACTIONS        Open,Print
   }

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

d. Skapa datatypen för `*.tpl`-filerna:

```
DATA_ATTRIBUTES BTETemplateFile
{
   DESCRIPTION    Mallfil för BestTextEditor.
   ICON           BTETempl
   ACTIONS        Open
}

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

e. Skapa funktionen Open för `*.bte`-filerna.

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

f. Skapa funktionen Print för `*.bte`-filerna.

Det här är enkla utskriftsfunktioner som skriver ut datafilerna. Dessa funktioner kräver ett värde på miljövariabeln LPDEST och ignorerar utskriftsalternativet `-s`. (Om LPDEST inte anges kanske funktionen inte fungerar.)

```
ACTION Print
{
   ARG_TYPE       BTEDataFile
   TYPE           MAP
   MAP_ACTION     BTEPrintData
}

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

Detta är ytterligare en version av funktionen BTEPrintData och ett tillhörande skript. Tillsammans hanterar de situationer där LPDEST inte anges eller där tyst utskrift begärs.

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

Innehållet i skriptet `/usr/BTE/bin/BTEenvprint` är:

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

g. Skapa funktionen Open för `*.tpl`-filer:

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

h. Skapa funktionen Print för `*.tpl`-filerna:

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

NoPrint är en inbyggd funktion som visar en dialogruta som anger för användaren att filen inte kan skrivas ut.

5. **Placera hjälpfilerna i registreringspaket**.

a. Placera hjälpfilerna på följande platser:

```
/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. Skapa filen:

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

Placera följande funktionsdefinition i filen:

```
ACTION BTEHelp
{
  WINDOW_TYPE NO_STDIO
  EXEC_STRING /usr/dt/bin/dthelpview -helpVolume \
              BTEHelp.sdl
  DESCRIPTION Öppnar hjälpvolym till BestTextEditor.
}
```

6. **Skapa ikoner för programmet.**

   Använd Ikonredigeraren för att skapa ikonerna. Använd följande riktlinjer för storlekar:

Namn	**Storlek**
basnamn`.t.pm`	16 gånger 16
basnamn`.m.pm`	32 gånger 32
basnamn`.l.pm`	64 gånger 64

   Skapa dessa ikonfiler i katalogen `/desktop_approots/BTE/dt/appconfig/icons/C`:

   - Ikoner för funktionen som kör programmet: `BTERun.t.pm`, `BTERun.m.pm`, `BTERun.l.pm`
   - Ikoner som representerar `*.bte`-filer: `BTEData.t.pm`, `BTEData.m.pm`,
   - Ikoner som representerar `*.tpl-filer`: `BTETempl.t.pm`, `BTETempl.m.pm`
   - Ikoner som representerar programgruppen (används i steg 7): `BTEApp.t.pm`, `BTEApp.m.pm`

7. **Skapa programgruppen**.

   a. Skapa katalogen om du inte redan har gjort det.

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

   b. Det här steget är valfritt. Det skapar en unik ikon för programgruppen genom att en datatyp och tillhörande funktioner för programgruppen skapas. Om du inte utför det här steget används standardikonen.

   Lägg till följande datatyps- och funktionsdefinitioner till filen `/desktop_approots/BTE/dt/appconfig/types/C/BTE.dt`. Datatypen anger vilken ikon som ska användas till programgruppen BestTextEditor. Funktionerna skapar samma Open- och Print-funktionalitet som de inbyggda programgrupperna.

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

c. Skapa en ikon i programgruppen som startar programmet. Det gör du genom att skapa filen:

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

och göra filen exekverbar.

d. Skapa den funktionsfil i programmgruppen som ska öppna hjälpvolymen. Det gör du genom att skapa filen:

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

och göra filen exekverbar.

e. Placera andra filer i programgruppen, t ex “read me”-filer, exempeldata och mallfiler.

8. **Registrera programmet.**

   I ett terminalemulatorfönster:

   a. Logga in som rot.

   b. Kör kommandot:

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

   c. Öppna programgruppen Skrivbordsverktyg och dubbelklicka på Ladda om program.

Konfigurera skrivbordet i nätverk

5

Skrivbordet är utformat för att fungera i avancerade nätverksmiljöer. Skrivbordets arkitektur gör att systemadministratören kan distribuera datorresurser via nätverket, bl a:

- Program.
- Datafiler för program.
- Skrivbordssessionstjänster (skrivbordsprogram som inloggningshanteraren och filhanteraren).
- Hjälptjänster. Hjälpdatafiler kan placeras på en central hjälpserver.

Översikt över skrivbordet i nätverk	*85*
Allmänna steg för att konfigurera nätverket	*89*
Konfigurera nätverkets basoperativsystem för skrivbordet	*90*
Konfigurera skrivbordsklienter- och servrar	*93*
Administrera programtjänster	*98*

Översikt över skrivbordet i nätverk

Operativsystemet innehåller en mängd olika nätverkstjänster, bl a distribuerat filsystem och fjärrexekvering. X-servrar har ytterligare nätverksfunktioner, bl a åtkomst till fjärrskärmar och säkerhetstjänster.

Skrivbordet placerar ett användargränssnitt ovanpå dessa nätverksfunktioner. Syftet med detta gränssnitt och dess underliggande arkitektur är att skapa nätverkssystem som är:

- Enklare att använda. Användarna kan köra program och komma åt datafiler utan att behöva tänka på var i nätverket programmet och data finns.
- Enklare att administrera. Skrivbordet innehåller verktyg för programintegration och nätverkssökvägar som gör det enklare för systemet att hitta fjärrdata och -program. Dessutom innebär skrivbordets filnamnstilldelning att det blir enklare att administrera komplexa nätverk som innehåller flera servrar.
- Flexibelt. Förutom att skrivbordets administrationsfunktioner har utformats för vissa nätverkssituationer, kan skrivbordet innehålla många andra anpassade nätverkskonfigurationer.

Typer av nätverkstjänster

I nätverket kan användarna sitta framför en viss skärm och komma åt olika datatjänster som distribueras från andra system, t ex:

- Skrivbordssessionen och dess program—t ex arbetsytehanteraren och filhanteraren
- Andra program
- Datafiler

I nätverksterminologi används termen *server* för att beskriva ett system som tillhandahåller datatjänster för ett eller flera system. När ett system tar emot tjänster från en server, kallas det *klient* till den servern.

I ett komplext nätverk kan ett system använda tjänster som finns på flera olika system i nätverket. Ett system kan också fungera som en viss typ av server (t ex en sessionsserver) och samtidigt vara en klient (t ex till en programserver).

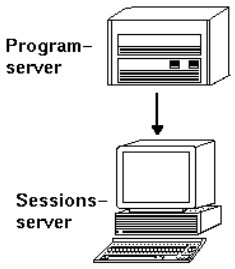
Typiska nätverkssituationer

Från ett skrivbordsperspektiv kan en typisk nätverkskonfiguration innehålla kombinationer av dessa huvudkomponenter:

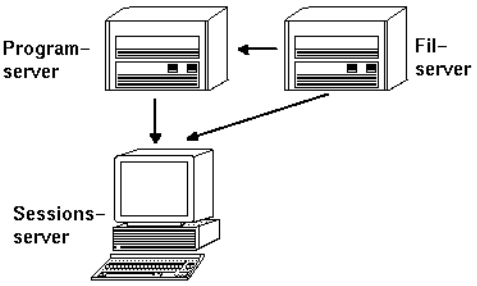
Skärmar	Där X-servern körs

Inloggnings-/sessionsservrar	Där skrivbordsprogrammen (inloggningshanteraren, arbetsytehanteraren, osv) körs
Programservrar	Där andra program körs
Filservrar	Där data som används av programmet finns

En av de vanligaste nätverkskonfigurationerna omfattar system med tillgång till en programserver. I figur 5-1 visas en arbetsstation som använder en programserver. X-servern och skrivbordssessionen körs på arbetsstationen. Nätverk använder också ofta filservrar för att lagra mängder av data. Dessa

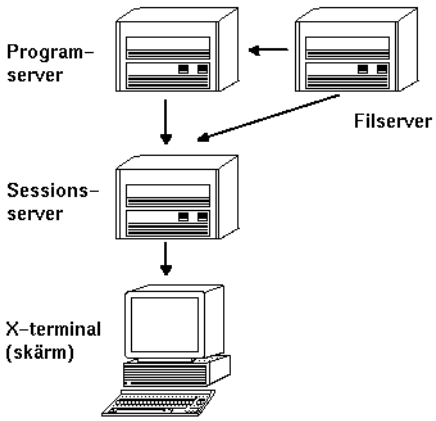


data kan användas av program som körs på en programserver, eller av skrivbordsprogram (filhanteraren behöver t ex tillgång till datafiler för att visa dem i filhanterarens fönster).



figur 5-1 Filservrar innehåller data för programservrar och sessionsservrar

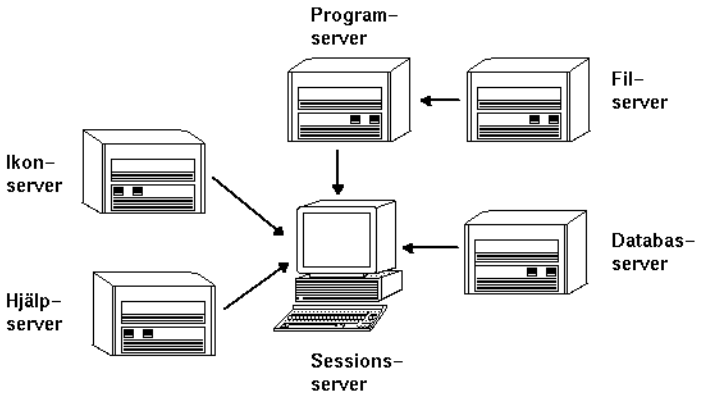
X-terminaler kör X-servern och hämtar tjänster för skrivbordssessionen från andra system.



figur 5-2 X-terminaler får sessionstjänster från en sessionsserver

Andra nätverkssituationer

Skrivbordet är flexibelt och kan stödja mer komplexa nätverkskonfigurationer. Dessa omfattar normalt olika tjänster som, tillsammans med filservrar, är tillgängliga för programservrar.



figur 5-3 Tjänster som behövs för en skrivbordsprogramserver kan distribueras

Sammanfattning — typer av tjänster

Skärmar	Systemet som kör X-servern.
Inloggnings- och sessionsserver	Systemet som kör skrivbordssessionen (inloggningshanteraren, sessionshanteraren, fönsterhanteraren, filhanteraren, osv)
Programserver	Ett system där programmet körs. Kallas också *värddator för exekvering*.
Filserver	Ett system där datafiler för program lagras
Hjälpserver	Ett system där hjälpdatafiler lagras
(Funktions-)databasserver	Ett system där filer som innehåller funktions- och datatypsdefinitioner lagras
Ikonserver	Ett system där ikonfiler lagras

Nätverket kan också omfatta ytterligare servrar, t ex en lösenordsserver, brevserver, bildserver, osv.

Allmänna steg för att konfigurera nätverket

Det finns tre allmänna steg för att konfigurera skrivbord i nätverket:

1. Konfigurera basnätverkstjänsterna i operativsystemet.

   Dessa tjänster tillhandahålls av operativsystemet och är nödvändiga för skrivbordet. Mer information finns i ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Installera och konfigurera program och tjänster för skrivbordsnätverket.

   Detta är tjänsterna som behövs för skrivbordet, oavsett typ av klient och server som installeras. Mer information finns i ”Konfigurera skrivbordsklienter- och servrar” på sidan 93.

3. Konfigurera en viss typ av server eller klient.

   När du t ex konfigurerar en programserver krävs andra steg än när du konfigurerar en filserver. Mer information finns i ”Administrera programtjänster” på sidan 98.

5

Konfigurera nätverkets basoperativsystem för skrivbordet

Skrivbordet kräver följande basnätverkskonfiguration:

- Användaren måste ha ett inloggningskonto på sessionsservern och på varje system som tillhandahåller skrivbordstjänster för sessionsservern. Användaren måste ha samma användar-ID och grupp-ID på alla klienter och serversystem.
- Systemen måste ha tillgång till fjärrfilsystem som innehåller data som används av sessionen och andra program.
- Skrivarspoolern `lp` måste vara konfigurerad för åtkomst av fjärrskrivare.
- `sendmail` måste vara konfigurerad för e-posttjänster.
- X-verifiering måste vara angivet.

Inloggningskonto för användare

I det här avsnittet beskrivs vad som krävs för inloggningskonton på skrivbord i nätverket.

Inloggningskonton

Användaren måste ha inloggningskonto på:

- Alla system som tillhandahåller tjänster för skrivbordet, inklusive programservrar, filservrar och system som innehåller nätverksskrivare.
- Alla sessionsservrar som användaren använder. Sessionsservrar används normalt med X-terminaler.

Konsekvent användar- och grupp-ID

UNIX-användare identifieras av ett inloggningsnamn och ett numeriskt användar-ID. Användaren måste ha samma inloggningsnamn och användar-ID på alla klienter och serversystem i nätverket.

UNIX-användare kopplas också till en eller flera inloggningsgrupper. Varje grupp har ett gruppnamn och ett numeriskt grupp-ID. I ett nätverk ska alla system använda samma gruppnamn och grupp-ID.

Mer information finns i kommandobeskrivningen till `id(1)` och `id(1M)`.

Konfigurera distribuerad filsystemåtkomst

Skrivbordet använder NFS® för att dela filer mellan system. Du måste identifiera alla filsystem i nätverket som innehåller delade filer och kontrollera att de är korrekt anslutna till rätt system.

Normalt måste du tillhandahålla följande fjärrfilåtkomst:

- Användarens hemkatalog måste delas av alla skrivbordsklienter och serversystem. Detta är nödvändigt eftersom:
  - Hemkatalogen innehåller datafiler som program på fjärrsystem ska ha tillgång till. Program som använder datafiler har ofta hemkatalogen som standardadressen för datafilerna.
  - Hemkatalogen är standardverifieringskatalog för `dtspcd`. Mer information om `dtspcd` finns i ”Konfigurera styrdaemon för underprocesser” på sidan 96.
- Om användaren kräver åtkomst till datafiler som inte finns i hemkatalogen, måste dessa datafiler delas av alla skrivbordsklienter och serversystem som arbetar med datafilerna.
- Kataloger för skrivbordsinstallationen och -konfigurationen (`/usr/dt` och `/etc/dt`) måste delas av alla skrivbordsklienter och serversystem, så att alla av användarens program har tillgång till samma skrivbordskonfigurationsfiler.

Hemkatalog för nätverk

Ett skrivbord i nätverk fungerar mest effektivt när användaren har en enstaka hemkatalog som delas mellan alla klienter och serversystem i nätverket.

Med en nätverkshemkatalog kan användaren använda olika system i nätverket utan att egna anpassningar och konfigurationer försvinner. Detta beror på att egna anpassningar, och informationen som krävs för att återställa föregående session, sparas i underkataloger till hemkatalogen.

En hemkatalog behövs också för:

- Standardmekanismen för X-verifiering. Se ”Konfigurera X-verifiering” på sidan 92.
- Skrivbordets styrdaemon för underprocesser, som startar fjärrprogram måste kunna skriva till användarens hemkatalog.

Konsekventa filnamn

Du ska konfigurera nätverket så att användarna kan komma åt sina datafiler från alla system med samma namn. Detta kallas för att använda *konsekventa filnamn*, och uppnås normalt genom att du skapar lämpliga symboliska länkar. Du kan t ex konfigurera alla system så att varje användares hemkatalog är tillgänglig som `/users/`*inloggningsnamn* genom att skapa en symbolisk länk till katalogens verkliga anslutning.

Konfigurera åtkomst av fjärrskrivare

Skrivbordet använder skrivarspoolern `lp` för att komma åt lokala skrivare och fjärrskrivare. Mer information om hur du konfigurerar spoolern `lp` finns i kommandobeskrivningen till `lpadmin(1M)`.

Innan du försöker skriva ut med skrivbordets grafiska gränssnitt, ska du testa att du kan skriva ut till alla skrivare med hjälp av kommandot `lp`.

Det rekommenderas starkt att du använder konsekventa skrivarnamn. Om t ex en viss skrivare kallas `Postscript1` i systemet som den är direkt ansluten till, bör samma namn användas av alla system som har fjärråtkomst till skrivaren.

Konfigurera elektronisk post

Skrivbordets brevlåda använder `sendmail` för att leverera post mellan system. Mer information om hur du konfigurerar anslutningsbarhet för e-post finns i kommandobeskrivningen till `sendmail(1M)`.

Innan du försöker skicka och ta emot post från skrivbordet, ska du testa att du kan skicka och ta emot post med hjälp av kommandot `mailx`.

Konfigurera X-verifiering

Skrivbordet använder standardmekanismen för att verifiera fjärrprogram (X-klienter) för åtkomst av en lokal skärm. Det enklaste sättet att göra detta är ge alla användare en nätverkshemkatalog. Detta garanterar att följande krav uppfylls:

- Användaren måste ha läs- och skrivbehörighet till filen *Hemkatalog*`/.Xauthority`.

- Filen `.Xauthority` på en programserver måste innehålla ”magic cookie” för skärmen som programmet ska köras på.

Mer information finns i kommandobeskrivningen till `X(1)` och `xauth(1)`.

Konfigurera skrivbordsklienter- och servrar

I det här avsnittet beskrivs nätverkets konfigurationskrav som är specifika för skrivbordet, dvs de funktioner som tillhandahålls av skrivbordet i stället för operativsystemet.

Avsnittet består av två delar:

- Konfigurera inloggnings- och sessionstjänster.
- Konfigurera tjänster som är nödvändiga för program och dess data. Detta omfattar program-, databas-, ikon-, fil- och hjälpservrar och dess klienter.

Konfigurera inloggnings- och sessionstjänster

En inloggnings/sessionsserver är ett system som innehåller skrivbordstjänster (inloggningshanteraren, sessionshanteraren, filhanteraren, fönsterhanteraren, osv) för en skärm och X-server.

En sessionsserver innehåller normalt tjänster för X-terminaler. En nätverkskonfiguration kan dock ställas in så att sessionstjänsterna koncentreras till en eller flera servrar som är tillgänglig för både X-terminaler och arbetsstationer.

Inloggningshanteraren är den komponent på skrivbordet som ansvarar för inloggningstjänster till andra skärmar. När användaren har loggat in startas sessionshanteraren för användaren.

Mer information om hur du konfigurerar inloggnings/sessionsservrar och X-terminaler finns i ”Visa en inloggningsbild på en nätverksskärm” på sidan 6.

Konfigurera andra programrelaterade tjänster

I det här avsnittet beskrivs vanliga nätverkskrav för skrivbordets:

- Programservrar
- Databasservrar
- Ikonservrar

- Hjälpservrar

▼ *Så här konfigurerar du skrivbordsklienter och -servrar*

1. Skapa de nätverkskonfigurationer till operativsystemet som krävs för skrivbordet.

   Se ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Installera skrivbordet eller minimiuppsättningen med filer:

   Du måste installera:

   - Hela uppsättningen körtidsfiler för Common Desktop Environment
   - *Eller* dessa uppsättningar filer: CDE-MIN och CDE-TT

   Obs! – Installation och filuppsättningar kan variera mellan olika leverantörer.

3. Konfigurera systemet för ToolTalk-databasserver-deamon `rpc.ttdbserver`.

   Detta ska ske automatiskt när skrivbordet installeras. Mer information finns i ”Konfigurera databasservern ToolTalk” på sidan 98.

4. Installera och konfigurera styrdaemon för underprocesser (`dtspcd`).

   Detta ska ske automatiskt när skrivbordet installeras. Mer information finns i ”Konfigurera styrdaemon för underprocesser” på sidan 96.

5. Anslut alla nödvändiga fjärrdata.

   Data betraktas som ”fjärrdata” när de finns i ett annat system än det som programmet, som använder data, körs på.

   Exempel:
   - Om ett program använder data som finns på en filserver, måste programmet ansluta dessa filer.
   - Om ikoner i filhanteraren finns på en ikonserver, måste sessionsservern ansluta dessa filer.
   - Om ett nätverk använder en hjälpserver för hjälpfiler, måste sessionsservern och alla programservrar ansluta hjälpdata.

Mer information om anslutningspunkter finns i nästa avsnitt ”Konfigurera anslutningspunkter för fjärrfilsystem”.

Konfigurera anslutningspunkter för fjärrfilsystem

När skrivbordet skickar filnamn från ett system till ett annat måste det överföra, eller *tilldela*, dessa filnamn till namn som målsystemet känner igen. Tilldelningen är nödvändig eftersom en fil kan anslutas på olika platser i olika system, och därför måste gå att komma åt med olika namn. Filen `/projects/big` på `sysA` kan t ex nås som `/net/sysA/projects/big` på `sysB`.

Krav för tilldelning av filnamn

För att tilldelningen ska göras korrekt måste följande vara sant:

- Kommandot `mount` används för att statiskt ansluta filsystem. Dessa typer av statisk anslutning är normalt konfigurerade i en fil som `/etc/checklist`, `/etc/mnttab` eller `/etc/filesystems`.

  För att tilldelningen av filnamn ska fungera korrekt mellan olika system, måste filsystemanslutningar använda konsekventa värddatornamn. Om en värddator har flera namn (t ex alias, eller om värddatorn har fler än en LAN-adress som är känd under olika namn) måste du använda samma namn och format på namnet för alla anslutningar.

- *Eller* att autoanslutning används för att ansluta filsystem vid standardanslutningspunkten `/net`.

- *Eller* att autoanslutning används för att ansluta filsystem till andra adresser än `/net` och att miljövariabeln DTMOUNTPOINT ställs in för att ange anslutningspunkten. Mer information finns i nästa avsnitt ”Ange värde på DTMOUNTPOINT”.

Mer information om autoanslutning finns i kommandobeskrivningen till `automount(1M)`.

Ange värde på DTMOUNTPOINT

Du måste ange miljövariabeln DTMOUNTPOINT om båda nedanstående villkor är sanna:

- Autoanslutningen används för att ansluta filsystem.
- *Och* fjärrfilsystem ansluts till andra adresser än `/net`.

DTMOUNTPOINT måste ställas in för processer, bl a:

- Användarens skrivbordsprocesser som startas automatiskt när användaren loggar in, t ex skrivbordshanteraren (`dtwm`) och filhanteraren (`dtfile`)
- Systemprocesser, t ex `rpc.ttdbserver` och `dtspcd`, som startas av mekanismer som `inetd`
- Program som startas av skrivbordet på lokala system eller fjärrsystem
- Program som startas av användaren från skalets kommandorad

Så här ställer du in DTMOUNTPOINT för alla dessa processer:

1. Redigera filen `/etc/inetd.conf`:

   a. Leta reda på posten `dtspcd` och lägg till:

      `-mount_point` *anslutningspunkt*

   b. Leta reda på posten `rpc.ttdbserver` och lägg till:

      `-m` *anslutningspunkt*

   Om t ex autoanslutning används med anslutningspunkten `/nfs`, är posterna i `/etc/inetd.conf` dessa:

```
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. Utför den procedur i ditt system som läser om `/etc/inetd.conf`. Mer information finns i kommandobeskrivningen till `inetd(1M)`.

3. Ställ in DTMOUNTPOINT så att värdet hämtas från användarens inloggningar.

   Detta gör du genom att ange variabeln i `/etc/dt/config/Xsession.d`. Mer information om hur du anger miljövariabler finns i ”Så här anger du miljövariabler” på sidan 33.

Konfigurera styrdaemon för underprocesser

Skrivbordets tjänst för underprocesskontroll (subprocess control, SPC) omfattar kommandoexekvering för klient/server.

Styrdaemon för underprocesser (`dtspcd`) används av skrivbordet för att starta fjärrprogram. Det är en `inet`-daemon som accepterar förfrågningar från fjärrklienter att exekvera kommandon. Mer information om hur du konfigurerar `inet`-daemons finns i kommandobeskrivningen till `inetd.conf(1M)`.

Skrivbordets funktionsanropsbibliotek använder tjänsten SPC för att anropa funktioner.

Så här konfigurerar du dtspcd

1. Bekräfta att `dtspc` är korrekt registrerad i både `/etc/services` och `/etc/inetd.conf`. Se kommandobeskrivningen till `dtspcd(1M)`.
2. Endast HP-UX: Kontrollera att `/usr/adm/inetd.sec` är rätt konfigurerad. Se kommandobeskrivningen till `inetd.sec(4)`.

SPC-säkerhet

Verifieringen för tjänsten underprocesstyrning baseras på filsystemverifiering. `dtspcd` måste ha tillgång till en *verifieringskatalog* som också är ansluten till alla SPC-klientsystem.

Normalt är verifieringskatalogen för `dtspcd` användarens hemkatalog. Du kan dock konfigurera `dtspcd` till att använda en annan adress genom att ange alternativet `-auth_dir` i katalogen `/etc/inetd.conf`. Mer information finns i kommandobeskrivningen till `dtspcd(1M)`.

Eftersom SPC-verifieringen baseras på filsystemverifiering, är tjänsten SPC bara så säker som det distribuerade filsystemet. Om du använder skrivbordet i ett nätverk där du inte litar på det distribuerade filsystemet, bör du avaktivera `dtspcd`. Detta gör du genom att kommentera bort posten `dtspc` i `/etc/services`.

Konfigurera miljövariabler för fjärrexekvering

När skrivbordet använder en funktion för att starta ett program i ett fjärrsystem, kopieras användarens miljövariabler till fjärrsystemet och placeras i programmets miljö.

Normalt ändras några av miljövariablerna innan de kopieras till fjärrsystemet. Du kan konfigurera både komponenten för funktionsanropet och underprocesstyrning för att bearbeta ytterligare miljövariabler innan variablerna placeras i programmets miljö.

Mer information om standardkonfigurationen och hur den ändras finns i kommandobeskrivningen till `dtactionfile(4)` och `dtspcdenv(4)`.

Konfigurera databasservern ToolTalk

En komponent i ToolTalk är databasservern ToolTalk, `/usr/dt/bin/rpc.ttdbserver`.

Databasservern ToolTalk används av meddelandetjänsten ToolTalk och till filnamnstilldelning. Den registreras normalt i `/etc/inetd.conf` när skrivbordet installeras och behöver ingen ytterligare konfiguration.

Mer information om databasservern ToolTalk och dess konfigurationsalternativ finns i kommandobeskrivningen till `rpc.ttdbserver(1M)`.

Konfigurera meddelandeservern ToolTalk

Meddelandeservern ToolTalk är `ttsession`. Normalt kräver den ingen konfiguration, utan startas av skriptet `Xsession` vid inloggningen.

Mer information om meddelandeservern ToolTalk och dess konfigurationsalternativ finns i kommandobeskrivningen till `ttsession`.

Konfigurera Kalender-daemon

En komponent i programmet Kalender är en Kalender-daemon, `rpc.cmsd`. Den registreras normalt i `/etc/inetd.conf` när skrivbordet installeras och behöver ingen ytterligare konfiguration.

Mer information om Kalender-daemon och dess konfigurationsalternativ finns i kommandobeskrivningen till `rpc.cmsd(1)`.

Administrera programtjänster

I det här avsnittet beskrivs vissa konfigurationskrav för:

- Programservrar och deras klienter

- Skrivbordsservrar som tillhandahåller vissa tjänster - databasservrar, ikonservrar och hjälpservrar

Här beskrivs också nätverkskrav för två speciella konfigurationer av nätverksprogram:

- Värddatorer för fjärrexekvering
- Program som körs via filsystemanslutningar

Miljövariabler för sökvägar

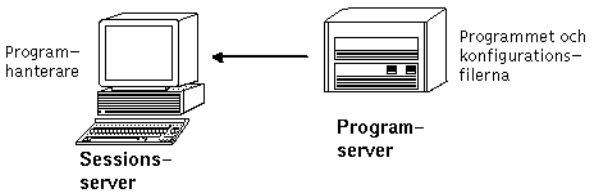
Skrivbordet använder en uppsättning miljövariabler för att ange vilken sökväg som ska användas för att hitta programmets skrivbordskonfigurationsfiler, t ex funktioner och datatyper, hjälpfiler och ikonfiler.

Mer information om hur du använder miljövariabler till sökvägar finns i Kapitel 7, ”Skrivbordets sökvägar” och i kommandobeskrivningen till `dtenvvar(5)`.

Konfigurera en programserver och dess klienter

I standardkonfigurationen till programservern, innehåller programservern alla binära filer och konfigurationsfiler som hör till programmet, inklusive:

- Programmets exekverbara filer
- Standardkonfigurationsfilerna för program, t ex programstandardfiler, meddelandekatalog och delade bibliotek.
- Skrivbordskonfigurationsfiler:
  - Funktions- och datatypsdefinitionsfiler
  - Ikonbildfiler
  - Hjälpdatafiler för skrivbordet



figur 5-4 Standardkonfiguration av programserver

▼ *Så här konfigurerar du en programserver*

1. Gör nätverkskonfigurationen för operativsystemet som krävs av skrivbordet tillgänglig.

   Se ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Gör den allmänna skrivbordskonfigurationen som krävs för servrar tillgänglig.

   Se ”Så här konfigurerar du skrivbordsklienter och -servrar” på sidan 94.

3. Installera programmet eller programmen.

4. Om ett program inte registreras automatiskt, måste du registrera det.

   Se Kapitel 4, ”Registrera program”.

▼ *Så här konfigurerar du klienten till en programserver*

1. Gör nätverkskonfigurationen för operativsystemet som krävs av skrivbordet tillgänglig.

   Se ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Gör den allmänna skrivbordskonfigurationen som krävs för klienter tillgänglig.

   Se ”Så här konfigurerar du skrivbordsklienter och -servrar” på sidan 94.

3. Lägg till programservern till den systemtäckande eller egna programsökvägen:

Systemtäckande	Ange variabeln DTSPSYSAPPHOSTS i `/etc/dt/config/Xsession.d/0010.dtpaths`
Personlig	Ange variabeln DTSPUSERAPPHOSTS i *Hemkatalog*`/.dtprofile`

   Följande rad i `/etc/dt/config/Xsession.d/0010.dtpaths` lägger t ex till ett system med värddatornamnet `SysAAA` och `SysBBB` till programsökvägen:

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

   Mer information om hur du anger programsökväg finns i:

   - ”Programsökväg” på sidan 116
   - ”Ange värde för en sökväg” på sidan 115

Konfigurera databas-, ikon- och hjälptjänster

Normalt installeras filer med funktions- och datatypsdefinitioner, ikoner och hjälpdata som hör till programmet, i samma system som programmet.

Detta kan vara en typisk konfiguration av hjälpdatafiler:

- Hjälpdatafilerna för filhanteraren finns normalt på sessionsservern. Skrivbordet kan hitta dem eftersom hjälpsökvägen automatiskt söker på rätt plats på sessionsservern.
- Hjälpfilerna till andra program finns normalt på samma programserver som programmet. Sessionsservern kan hitta dem eftersom en ändring av programsökvägen automatiskt innebär en ändring av hjälpsökvägen.

Ibland kanske du vill placera databas- (funktioner och datatyper), hjälp- eller ikondata någon annanstans i nätverket. Om t ex nätverket använder flera sessionsservrar kanske du vill skapa en hjälpserver där alla hjälpdatafiler för skrivbordsprogrammen (filhanteraren, Arbetsinställningar, osv) lagras. Detta sparar diskutrymme eftersom hjälpfilerna inte behöver dupliceras på varje sessionsserver.

▼ *Så här skapar du en databas-, hjälp- eller ikonserver*

1. Gör nätverkskonfigurationen för operativsystemet som krävs av skrivbordet tillgänglig.

   Se ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Gör den allmänna skrivbordskonfigurationen som krävs för klienter tillgänglig.

   Se ”Så här konfigurerar du skrivbordsklienter och -servrar” på sidan 94.

3. Installera databas-, hjälp- eller ikonfilerna.

   Filerna kan finnas var som helst i systemet. Det kan dock underlätta att använda följande adresser, eftersom detta är katalogerna som automatiskt söks när systemet har utnämnts till programserver.

   - Databasfiler: `/etc/dt/appconfig/types/`*`språk`*
   - Hjälpfiler: `/etc/dt/appconfig/help/`*`språk`*
   - Ikonfiler: `/etc/dt/appconfig/icons/`*`språk`*

Om du anger en databasserver måste funktionerna skrivas för att ange var deras kommandon (EXEC_STRING) ska köras. Se ”Ange en värddator för fjärrexekvering” på sidan 103.

▼ *Så här konfigurerar du sessionservern att hitta en databas-, ikon- eller hjälpserver*

1. Gör nätverkskonfigurationen för operativsystemet som krävs av skrivbordet tillgänglig.

   Se ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Gör den allmänna skrivbordskonfigurationen som krävs för klienter tillgänglig.

   Se ”Så här konfigurerar du skrivbordsklienter och -servrar” på sidan 94.

3. Lägg till databas-, ikon- eller hjälpdatabasen i lämplig sökväg.
   - Om du har placerat datafilerna på den plats som anges i Steg 3 i ”Så här skapar du en databas-, hjälp- eller ikonserver”, ändrar du programsökvägen.
   - Om du har placerat datafilerna någon annanstans måste du ändra den specifika sökvägen.

     Om du t ex har placerat hjälpfilerna i katalogen `/etc/dt/help` i systemet SysCCC lägger du till följande rad i `/etc/dt/config/Xsession.d/0010.dtpaths`:

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

   Mer information om hur du anger sökvägar finns i:
   - ”Databassökväg (funktion / datatyper)” på sidan 120
   - ”Ikonsökväg” på sidan 121
   - ”Hjälpsökväg” på sidan 123
   - ”Ange värde för en sökväg” på sidan 115

Speciella konfigurationer för nätverksprogram

I det här avsnittet beskrivs hur du konfigurerar system att köra program:

- Någon annanstans än i systemet som innehåller funktionen - på en värddator för fjärrexekvering
- Lokalt via filsystemanslutningar

Ange en värddator för fjärrexekvering

I den typiska programserverkonfigurationen finns funktionsdefinitionen i samma system som programmets exekverbara filer. Funktioner kan dock skrivas för att exekvera kommandon i andra system. I den här konfigurationen kallas systemet som innehåller programmet för *värddator för exekvering*.

Funktionsdefinitionen kan finnas på sessionsservern eller i ett system som tillhandahåller funktions- och datatypstjänster för sessionsservern - som kallas *databasserver* eller *databasvärd*.

Funktionsdefinitioner använder fältet EXEC_HOST för att ange var deras kommandon (EXEC_STRING) ska köras. Följande funktionsdefinition anger t ex att en xload-klient ska köras i systemet med värddatornamnet SysDDD:

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

Om fältet EXEC_HOST anger fler än ett värddatornamn, försöker skrivbordet att exekvera EXEC_STRING på alla värddatorer i tur och ordning tills det hittar en sträng som kan köra funktionen. Följande EXEC_HOST-fält anger t ex att funktionen först ska försöka köra EXEC_STRING i SysDDD och, om detta inte går, försöka i SysEEE.

```
    EXEC_HOST       SysDDD,SYSEEE
```

Om fältet EXEC_HOST inte anges för någon funktion får det standardvärdet %DatabaseHost%. Värdet på %DatabaseHost% hämtas från databassökvägen.

Anta att databasens sökväg har ändrats genom att följande rad har lagts till i /etc/dt/config/Xsession.d/0010.dtpaths:

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

SysAAA anges med värddatorsyntaxen—SysAAA:. En funktionsdefinition som hittas med det här elementet i sökvägen anger databasvärden till SysAAA. En funktion som hittas med delen /net/SysBBB... i sökvägen, anger dock databasvärden till det lokala systemet, eftersom syntaxen inte omfattar värddatorn.

Så här konfigurerar du värddatorn för fjärrexekvering

1. Gör nätverkskonfigurationen för operativsystemet som krävs av skrivbordet tillgänglig.

   Se ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Gör den allmänna skrivbordskonfigurationen som krävs för servrar tillgänglig.

   Se ”Så här konfigurerar du skrivbordsklienter och -servrar” på sidan 94.

3. Kontrollera att programmen är rätt installerade och konfigurerade för lokal exekvering.

Så här konfigurerar du systemet som innehåller funktionsdefinitionen

1. Gör nätverkskonfigurationen för operativsystemet som krävs av skrivbordet tillgänglig.

   Se ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Gör den allmänna skrivbordskonfigurationen som krävs för servrar tillgänglig.

   Se ”Så här konfigurerar du skrivbordsklienter och -servrar” på sidan 94.

3. Skapa och installera funktionsdefinitionerna och programgrupperna.

   Se ”Skapa funktioner som kör program i fjärrsystem” på sidan 178 och ”Skapa och administrera allmänna programgrupper” på sidan 45.

Så här konfigurerar du sessionsservern

1. Gör nätverkskonfigurationen för operativsystemet som krävs av skrivbordet tillgänglig.

   Se ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

2. Gör den allmänna skrivbordskonfigurationen som krävs för klienter tillgänglig.

   Se ”Så här konfigurerar du skrivbordsklienter och -servrar” på sidan 94.

3. Ändra funktionssökvägen till att omfatta databasvärden.

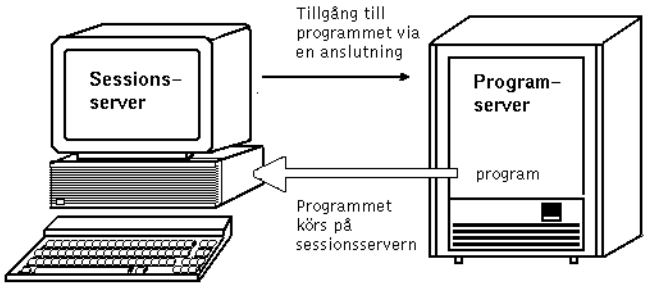
   Se ”Databassökväg (funktion / datatyper)” på sidan 120.

4. Ändra programsökvägen till att omfatta värddatorn för exekvering.

   Se ”Programsökväg” på sidan 116.

Köra program lokalt

Standardkonfigurationen för programservrar kör program på programservern. Ibland kan det dock vara bättre att programmet är installerat i ett fjärrsystem men körs lokalt på sessionsservern.



figur 5-5 Exekvering via anslutningspunkter

Så här konfigurerar du programservern

Ingen speciell konfiguration behövs.

Så här konfigurerar du sessionsservern

♦ Ändra programsökvägen. Använd den lokala absoluta sökvägen till programmet.

Du kan t ex använda följande variabeldefinition för att söka efter ett program som har registrerats i `sysAAA`:

```
DTSPSYSAPPHOSTS=/net/SysAAA/etc/dt/appconfig/appmanager/C
```

Sessionsservern måste ha tillgång till programmets konfigurationsfiler, t ex programstandardfiler, meddelandekataloger och delade bibliotek.

5

Konfigurera och administrera utskrifter från skrivbordet 6

En skrivbordsanvändare kan skriva ut filer på flera olika sätt. De delas in i två huvudkategorier: skriva ut från skrivbordet och från ett program.

Du kan skriva ut från skrivbordet genom att:

- Markera en fil i filhanteraren och välja Skriv ut från Markerade-menyn eller ikonens meny
- Dra en fil från filhanteraren till frontpanelens skrivarikon eller panelundermenyn Egen skrivare
- Dra en fil från filhanteraren till en skrivare i utskriftshanterarens huvudfönster

Om du vill skriva ut från ett program använder du kommandot Skriv ut, som normalt finns på en meny eller nås via en ikon i programfönstret.

Lägga till och ta bort skrivare	*108*
Ikonbilder till skrivaren	*109*
Konfigurera standardskrivare	*110*
Vad som sker vid utskrift	*112*

6

Lägga till och ta bort skrivare

I det här avsnittet beskrivs hur du lägger till och tar bort skrivare från skrivbordet.

▼ Så här lägger du till en skrivare till skrivbordet

1. Lägg till en skrivare till systemets konfiguration.

   Följ instruktionerna i systemadministrationsdokumentationen för operativsystemet.

2. Kör kommandot:

   `env LANG=`*`språk`* `/usr/dt/bin/dtprintinfo -populate`

3. Starta om utskriftshanteraren genom att dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg i programhanteraren. Kontrollera att skrivaren visas.

4. Skicka ett meddelande till användarna att även de ska starta om utskriftshanteraren och köra Ladda om funktioner.

Varje gång utskriftshanteraren anropas läser den systemets skrivarkonfigurationslista. Om en ny skrivare upptäcks skapas automatiskt en ny skrivarfunktion och ikon för skrivaren till skrivbordet. Du behöver inte göra något mer för att skrivaren ska visas på skrivbordet.

▼ Så här tar du bort en skrivare från skrivbordet

1. Ta bort skrivaren från systemkonfigurationen.

   Följ instruktionerna i systemadministrationsdokumentationen för operativsystemet.

2. Starta om utskriftshanteraren genom att dubbelklicka på Ladda om funktioner i programgruppen Skrivbordverktyg i programhanteraren. Kontrollera att skrivaren är borta.

3. Skicka ett meddelande till användarna att även de ska starta om utskriftshanteraren och köra Ladda om funktioner.

Varje gång utskriftshanteraren anropas läser den systemets skrivarkonfigurationslista. Om den upptäcker att en skrivare har tagits bort från listan, tas skrivarens funktion och ikon automatiskt bort från utskriftshanteraren och filhanteraren. Du behöver inte göra något mer för att ta bort skrivaren från skrivbordet.

Obs! – Utskriftshanteraren kan inte ta bort skrivare från frontpanelen. Därför ska du skicka ett meddelande till alla användare när du tar bort en skrivare, så att de tar bort alla ikoner för den skrivaren på frontpanelen.

Ändra intervall för jobbuppdatering

Om du vill ändra hur ofta information i utskriftshanteraren ska uppdateras, ändrar du intervallet för jobbuppdatering. Normalt begär utskriftshanteraren information om utskriftsjobb från skrivaren var 30:e sekund. Du ändrar detta med skjutreglaget Uppdatera intervall i dialogrutan Välj alternativ (visas när du väljer Välj alternativ på Visa-menyn).

Ikonbilder till skrivaren

När du lägger till en skrivare tilldelas den automatiskt standardskrivarikonen. Om du vill göra en annan ikon tillgänglig placerar du ikonfilerna i `/etc/dt/appconfig/icons/`*`språk`*, eller i någon annan katalog i ikonsökvägen. Användarna kan sedan markera den här ikonen för att ersätta standardikonen.

Du *måste* skapa en komplett uppsättning (mycket liten, normal och stor) ikoner, annars visas de inte i ikonväljaren i utskriftshanteraren.

Mer information om ikonsökvägen finns i ”Ikonsökväg” på sidan 121.

Ikonfilnamn- och storlekar

Ikonernas filnamn ska se ut så här:

basnamn.storlek`.`*typ*

där:

storlek	`l` (stor), `m` (normal), `t` (mycket liten). Mer information om ikonstorlekar finns i ”Storlekskonventioner” på sidan 205.
typ	`pm` (färgat bildpunktsmönster), `bm` (bitmönster).

Ikonfilnamnet på en medelstor och mycket liten bildpunktsmönsterikon för en färgskrivare blir då `Färgskrivare.m.pm` och `Färgskrivare.t.pm`.

Mer information finns i Kapitel 12, ”Skapa skrivbordsikoner”.

▼ Så här ändrar du ikon, skrivarnamn och beskrivning globalt

Du ska ändra de globala skrivaregenskaperna så fort du har lagt till skrivaren, innan användarna har ändrat dem med utskrifthanteraren. När användarna väl har ändrat skrivaregenskaper med utskriftshanteraren visas inte de ändringar som du gör.

Redigera filen `/etc/dt/appconfig/types/`*språk*`/`*utskriftskö*`.dt` med önskad information för ikon, skrivarnamn och beskrivning:

1. Uppdatera *basnamn* i fältet `ICON` till ikonens nya basnamn.
2. Uppdatera *namn* i fältet `LABEL` till skrivarens nya namn.

♦ Uppdatera texten i fältet `DESCRIPTION`.

Här kan du skriva var skrivaren finns, typ av skrivare och skrivarkontakter. Om du vill skriva mer än en rad, placerar du ett \ i slutet av raden. Exempel:

```
DESCRIPTION    Detta är en PostScript-skrivare i hus 1,\
               rum 123. Ring 123 om problem uppstår.
```

Konfigurera standardskrivare

Standardskrivaren används när användaren:

- Släpper ett objekt på skrivarikonen på frontpanelen
- Markerar ett objekt i filhanteraren och väljer Skriv ut på Markerade-menyn eller på ikonens meny
- Skriver ut från program som använder standardskrivaren

▼ Så här ändrar du destination för standardutskrift

Så här ändrar du standardskrivare för alla användare:

1. Öppna filen `/etc/dt/config/Xsession.d/0010.dtpaths`.

   Om `/etc/dt/config/Xsession.d/0010.dtpaths` inte finns kopierar du den från `/usr/dt/config/Xsession.d/0010.dtpaths`

2. På raden `LPDEST=`*skrivare* uppdaterar du *skrivare* till den nya destinationen för standardutskrift.

   Om raden inte finns lägger du till raden `LPDEST=`*skrivare*, där *skrivare* är namnet på skrivaren som ska vara standardskrivare.

3. Användarna måste logga ut och in igen.

Så här ändrar en enstaka användare standardskrivare:

♦ Kopiera en annan skrivare till frontpanelen från panelundermenyn Egen skrivare.

Så här anger du en annan skrivare som standardskrivare:

1. Gå till hemkatalogen och öppna filen `.dtprofile`.
2. Lägg till eller redigera raden som anger värde på miljövariabeln LPDEST:

   `LPDEST=`*skrivarenhet;* `export LPDEST`

   `Om du använder csh är syntaxen:`

   `setenv LPDEST` *skrivarenhet*

Följande rad ändrar t ex standardskrivaren till skrivaren vars enhetsnamn är `laser3d`.

```
LPDEST=laser3d; export LPDEST
```

Om du använder `csh` är syntaxen:

```
setenv LPDEST laser3d
```

Vad som sker vid utskrift

När en utskriftsbegäran initieras av att en fil släpps på skrivarikonen, utför systemet följande:

1. Systemet söker i datatypsdatabasen efter det släppta objektets definition.
2. Om det finns en unik skrivarfunktion för datatypen (anges med fältet `ARG_TYPE` i skrivarfunktionen) används den. I annat fall används standardskrivarfunktionen (`dtlp`). Om filen är en PostScript®-fil använder systemet funktionen Print för PostScript-filer. (Den här funktionen definieras i `/usr/dt/appconfig/types/`*`språk`*`/dt.dt`.) Om du använde verktyget Skapa funktion för den här datatypen, används det utskriftskommando du skrev som den unika utskriftsfunktionen, vilket används för att skriva ut filer av den här typen.
3. Filen levereras till skrivaren med det vanliga UNIX-undersystemet `lp`.

Skrivbordets sökvägar 7

Skrivbordet använder sökvägar för att hitta program och tillhörande skrivbordsfiler.

Skrivbordets sökvägar och deras miljövariabler	*114*
Ange värde för en sökväg	*115*
Programsökväg	*116*
Databassökväg (funktion/datatyper)	*120*
Ikonsökväg	*121*
Hjälpsökväg	*123*
Lokalanpassade sökvägar	*125*

Skrivbordet tillhandahåller fyra sökvägar som beskrivs i tabell 7-1.

tabell 7-1 Skrivbordets sökvägar

Sökväg	Beskrivning
Program	Används för att söka efter program. Programhanteraren använder programsökvägen för att dynamiskt fylla den översta nivån när användaren loggar in.
Databas	Används för att ange ytterligare adresser för funktions- och datatypsdefinitionsfiler (`*.dt`) och frontpanelsfiler (`*.fp`).
Ikoner	Används för att ange ytterligare adresser för ikoner.
Hjälpdata	Används för att ange ytterligare adresser för skrivbordets hjälpdata.

Sökvägarna kan omfatta både kataloger och fjärrkataloger. Därför spelar sökvägarna en viktig roll i skrivbordets nätverksarkitektur. Ett system hittar t ex ett program på programservern eftersom programservern listas i programsökvägen.

När en sökväg omfattar fjärrkataloger måste du konfigurera fjärrfilsåtkomst för dessa adresser. Mer information finns i ”Konfigurera nätverkets basoperativsystem för skrivbordet” på sidan 90.

Skrivbordets sökvägar och deras miljövariabler

Skrivbordets sökvägar skapas vid inloggningen av hjälpmedlet `dtsearchpath`. Hjälpmedlet `dtsearchpath` använder en kombination av miljövariabler och inbyggda adresser för att skapa sökvägarna.

Miljövariablerna som `dtsearchpath` läser kallas *indatavariabler*. Dessa anges av systemadministratören eller slutanvändaren. Indatavariabler använder namngivningskonventionen `DTSP*`.

När `dtsearchpath` körs vid inloggningen sätts värdena som tilldelas dessa variabler samman, inbyggda adresser läggs till och värden för *utdatavariabler* skapas. Det finns en utdatavariabel för varje sökväg.

tabell 7-2 Miljövariabler för skrivbordets sökvägar

Sökväg för:	Utdatamiljövariabel	Systemtäckande indatavariabel	Egen indatavariabel
Program	DTAPPSEARCHPATH	DTSPSYSAPPHOSTS	DTSPUSERAPPHOSTS
Databas	DTDATABASESEARCHPATH	DTSPSYSDATABASEHOSTS	DTSPUSERDATABASEHOSTS
Ikoner	XMICONSEARCHPATH, XMICONBMSEARCHPATH	DTSPSYSICON	DTSPUSERICON
Hjälpdata	DTHELPSEARCHPATH	DTSPSYSHELP	DTSPUSERHELP

1. Funktions-,datatyps- och frontpanelsdefinitioner

Komponenter använder utdatavariablernas värden. Programhanteraren använder t ex programsökvägens värde (DTAPPSEARCHPATH) för att hitta programgruppen.

Ange värde för en sökväg

Du kan ändra systemtäckande och egna sökvägar. Ändringarna görs genom att värden anges för systemtäckande eller egna indatavariabler. Alla ändringar du gör läggs till de inbyggda sökvägsadresserna.

▼ Så här visar du aktuellt värde för en sökväg (utdatavariabel)

♦ Använd kommandot `dtsearchpath` för att visa aktuella värden för sökvägarna:

- Om du vill hämta värdet för aktuell (inloggad) användare exekverar du:

  ```
  dtsearchpath -v
  ```

- Om du vill hämta värdet för en annan användare exekverar du:

  ```
  dtsearchpath -u användare
  ```

Sökvägens värden omfattar dessa variabler:

`%H` Används i DTHELPSEARCHPATH. Namn på hjälpvolymen.

%B	Används i XMICONSEARCHPATH. Ikonfilens basnamn.
%M	Används i XMICONSEARCHPATH. Storleken på ikonfilen (.l, .m, .s, .t)
%L	Värde på miljövariabeln LANG.

▼ Så här gör du egna ändringar av sökvägen

1. Öppna *Hemkatalog*/.dtprofile för redigering.
2. Lägg till eller redigera en rad som definierar ett värde för den egna indatavariabeln.

   Följande rad lägger t ex till en adress i användarens egen programsökväg:

   ```
   DTSPUSERAPPHOSTS=/projects1/editors
   ```

3. Logga ut och in igen för att aktivera ändringen.

▼ Så här gör du systemtäckande ändringar av sökvägen

1. Logga in som rot.
2. Om filen /etc/dt/config/Xsession.d/0010.dtpaths inte finns skapar du den genom att kopiera /usr/dt/config/Xsession.d/0010.dtpaths.
3. Öppna /etc/dt/Xsession.d/0010.paths för redigering. Lägg till eller redigera en rad som definierar ett värde på den systemtäckande indatavariabeln.

   Följande rad lägger t ex till en adress i den systemtäckande hjälpsökvägen:

   ```
   DTSPSYSHELP=/applications/helpdata
   ```

4. Informera alla användare i systemet att de måste logga ut och in igen för att ändringarna ska träda i kraft.

Programsökväg

Programsökvägen är den primära sökvägen som används av skrivbordet för att hitta program i lokala system och på programservrar i nätverket.

När adresser läggs till i programsökvägen uppdateras de andra sökvägarna (databas, ikon och hjälp) automatiskt för att återspegla motsvarande position för data. Därför är administration av program och deras skrivbordskonfigurationsfiler relativt enkel. Mer information finns i ”Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp” på sidan 119.

Standardprogramsökvägen

Standardprogramsökvägen omfattar egna, systemtäckande och inbyggda adresser. *Språk* är normalt `C`.

Egen adress	*Hemkatalog*`/.dt/appmanager`
Systemtäckande adress	`/etc/dt/appconfig/appmanager/`*språk*
Inbyggd adress	`/usr/dt/appconfig/appmanager/`*språk*

Miljövariabler för programsökvägar

Programsökvägen sätts samman utifrån inbyggda adresser och följande indatavariabler:

DTSPSYSAPPHOSTS	Indatavariabel för systemtäckande programsökväg
DTSPUSERAPPHOSTS	Indatavariabel för egen programsökväg

Den sammansatta sökvägen anges av utdatavariabeln DTAPPSEARCHPATH.

Syntax för indatavariabler till programsökvägen

Syntaxen för variablerna DTSPSYSAPPHOSTS och DTSPUSERAPPHOSTS är:

VARIABLE=adress `[,`*adress*`...]`

där *adress* har syntaxen:

`/`*sökväg*	Anger en katalog i det lokala systemet (sessionsserver). Använd den här syntaxen för att lägga till en lokal katalog.
värddatornamn`:`	Anger den systemtäckande katalogen `/etc/dt/appconfig/appmanager/`*språk* i systemet *värddatornamn*. Använd den här syntaxen för att lägga till en programserver.

värddatornamn`:/`*sökväg* Anger en katalog på *värddatorn* i fjärrsystemet.

`localhost:` Den lokala systemtäckande adressen. Det här nyckelordet används för att ändra prioritetsordning på den lokala systemtäckande adressen. Se ”Ändra prioritetsordning på den lokala systemtäckande adressen” på sidan 118.

Hur programsökvägens värde sätts samman

Värdet för programsökvägen (DTAPPSEARCHPATH) skapas genom att följande adresser sätts samman, listade i prioritetsordning:

- Adresser som anges med variabeln DTSPUSERAPPHOSTS
- Egen standardadress: *Hemkatalog*`/.dt/appmanager`
- Standardadressen: `/etc/dt/appconfig/appmanager/`*språk*
- Adresser som anges med variabeln DTSPSYSAPPHOSTS
- `/usr/dt/appconfig/appmanager/`*språk*

Syntaxen:

värddatornamn`:`

utökas till att ange katalogen `/etc/dt/appconfig/appmanager` i systemet *värddatornamn*.

Ändra prioritetsordning på den lokala systemtäckande adressen

Normalt har den lokala systemtäckande adressen (`/etc/dt/appconfig/appmanager/`*språk*) prioritet över fjärradresser. Lokala programgrupper har därför prioritet över fjärrgrupper med samma namn. Om både det lokala systemet och fjärrsystemet innehåller programgruppen Skrivare (`/etc/dt/appconfig/appmanager/`*språk*`/Skrivare`), används den lokala gruppen.

Programsökvägens indatavariabler tillhandahåller syntax för att ange prioritetsordning för de lokala systemtäckande programgrupperna:

`localhost:`

Anta att systemet måste ha tillgång till programservrarna SysA, SysB och SysC, och du vill att de systemtäckande programgrupperna i SysB ska ha prioritet över alla lokala grupper med samma namn.

Följande värde för DTSPSYSAPPHOSTS skapar detta:

```
DTSPSYSAPPHOSTS=SysB:,localhost:,SysA:,SysC:
```

Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp

När ett tillägg görs i programsökvägen läggs motsvarande adresser automatiskt till även för databas-, ikon- och hjälpsökvägen. På det här sättet kan du lägga till en programserver till en sökväg genom att bara ange indatavariabeln i programsökvägen.

Om du t ex anger DTSPSYSAPPHOSTS så här:

```
DTSPSYSAPPHOSTS=servera:
```

påverkas följande sökvägar:

Sökväg	**Katalog som läggs till i sökvägen**
Program	`servera:/etc/dt/appconfig/appmanager/`*språk*
Databas	`servera:/etc/dt/appconfig/types/`*språk*
Ikon	`servera:/etc/dt/appconfig/icons/`*språk*
Hjälp	`servera:/etc/dt/appconfig/help/`*språk*

Samma sak gäller om du anger DTSPSYSAPPHOSTS så här:

```
DTSPSYSAPPHOSTS=/projects1/apps
```

så påverkas följande sökvägar:

Sökväg	**Katalog som läggs till i sökvägen**
Program	`/projects1/apps/appmanager/`*språk*
Databas	`/projects1/apps/types/`*språk*
Ikon	`/projects1/apps/icons/`*språk*
Hjälp	`/projects1/apps/help/`*språk*

7

Databassökväg (funktion/datatyper)

Databassökvägen hjälper skrivbordet att söka i vissa addesser efter filer som innehåller:

- Funktions- och datatypsdefinitioner (`*.dt`-filer)
- Frontpanelsdefinitioner (`*.fp`-filer).

Du kanske måste ändra databassökvägen när du skapar en databasserver, eller när du lägger till en lokal adress för databasfiler.

Databasens standardsökväg

Standardsökvägen för databasen omfattar egna, systemtäckande och inbyggda adresser. *Språk* är normalt `C`.

Egen adress	*Hemkatalog*`/.dt/types`
Systemtäckande adress	`/etc/dt/appconfig/types/`*språk*
Inbyggd adress	`/usr/dt/appconfig/types/`*språk*

Hur programsökvägen påverkar databassökvägen

När en adress läggs till i programsökvägen läggs lämplig underkatalog för databasen till i databassökvägen (se ”Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp” på sidan 119).

Om t ex programservern `hosta:` läggs till i programsökvägen, läggs katalogen `hosta:/etc/dt/appconfig/types/`*språk* automatiskt till i databassökvägen.

Databassökvägens miljövariabler

Databassökvägen sätts samman utifrån inbyggda adresser och följande indatavariabler:

DTSPSYSDATABASEHOSTS Indatavariabel för den systemtäckande databassökvägen

DTSPUSERDATABASEHOSTS Indatavariabel för den egna databassökvägen

Använd dessa indatavariabler för att ange adresser utanför programsökvägen.

Den assemblerade databassökvägen anges av utdatavariabeln DTDATABASESEARCHPATH.

Syntax för databassökvägens indatavariabler

Syntaxen för variablerna DTSPSYSDATABASEHOSTS och DTSPUSERDATABASEHOSTS är:

VARIABEL=adress [,*adress*...]

där *adress* har syntaxen:

/sökväg	Anger en katalog i det lokala systemet (sessionsserver). Använd den här syntaxen för att lägga till en lokal katalog.
värddatornamn:	Anger den systemtäckande katalogen /etc/dt/appconfig/types/*språk* i systemet *värddatornamn*.

värddatornamn:/*sökväg*Anger en katalog i fjärrsystemet *värddatornamn*.

Hur databassökvägen sätts samman

Värdet för databassökvägen (DTDATABASESEARCHPATH) skapas genom att följande adresser sätts samman, listade i prioritetsordning:Ads som anges med variabeln DTSPUSERDATABASEHOSTS

- Adresser som härleds från variabeln DTSPUSERAPPHOSTS
- Egen standardadress: *Hemkatalog*/.dt/types
- Standardadressen: /etc/dt/appconfig/types/*språk*
- Adresser som anges med variabeln DTSPSYSDATABASEHOSTS
- Adresser som härleds från variabeln DTSPSYSAPPHOSTS
- /usr/dt/appconfig/types/*språk*

Syntaxen:

värddatornamn:

utökas till att ange katalogen /etc/dt/appconfig/types i systemet *värddatornamn*.

Ikonsökväg

Ikonsökvägen hjälper skrivbordet att söka i angivna adresser efter filer som innehåller bitmönster- och bildpunktsmönsterfiler.

Standardikonsökväg

Standardikonsökvägen omfattar egna, systemtäckande och inbyggda adresser. *Språk* är normalt `C`.

Egen adress	*Hemkatalog*`/.dt/icons`
Systemtäckande adress	`/etc/dt/appconfig/icons/`*språk*
Inbyggd adress	`/usr/dt/appconfig/icons/`*språk*

Hur programsökvägen påverkar ikonsökvägen

När en adress läggs till i programsökvägen, läggs en lämplig ikonunderkatalog automatiskt till i ikonsökvägen (se ”Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp” på sidan 119).

Om t ex programservern `hosta:` läggs till i programsökvägen, läggs katalogen `hosta:/etc/dt/appconfig/icons/`*språk* automatiskt till i ikonsökvägen.

Ikonsökvägens miljövariabler

Ikonsökvägen sätts samman utifrån inbyggda adresser och följande indatavariabler:

DTSPSYSICON	Indatavariabel för systemtäckande ikonsökväg
DTSPUSERICON	Indatavariabel för egen ikonsökväg

Använd dessa indatavariabler för att ange adresser utanför programsökvägen.

Den assemblerade ikonsökvägen anges av två utdatavariabler:

XMICONSEARCHPATHAnvänds av färgskärmar

XMICONBMSEARCHPATHAnvänds av monokroma skärmar

Syntax för ikonsökvägens indatavariabler

Syntaxen för variablerna DTSPSYSICON och DTSPUSERICON är:

VARIABEL=adress `[,`*adress*`...]`

där *adress* har syntaxen:

/sökväg	Anger en katalog i det lokala systemet (sessionsserver). Använd den här syntaxen för att lägga till en lokal katalog.

Om du vill ange en adress i ett annat system använder du dess nätverksfilnamn, t ex `/nfs/servera/projects/icons`.

Hur ikonsökvägen sätts samman

Värdet för ikonsökvägen (XMICONSEARCHPATH och XMICONBMSEARCHPATH) skapas genom att följande adresser sätts samman, listade i prioritetsordning:

- Adresser som anges med variabeln DTSPUSERICON
- Adresser som härleds från variabeln DTSPUSERAPPHOSTS
- Egen standardadress: *Hemkatalog*`/.dt/icons`
- Standardadressen: `/etc/dt/appconfig/icons/`*språk*
- Adresser som anges med variabeln DTSPSYSICON
- Adresser som härleds från variabeln DTSPSYSAPPHOSTS
- `/usr/dt/appconfig/icons/`*språk*

Färg- och monokromsökvägar skiljer sig bara åt när det gäller prioritetsordning mellan bildpunktsmönster och bitmönster. Variabeln XMICONSEARCHPATH listar bildpunktsmönster före bitmönster, och XMICONBMSEARCPATH listar bitmönster före bildpunktsmönster.

Hjälpsökväg

Hjälpsökvägen hjälper skrivbordet att söka i angivna adresser efter filer som innehåller hjälpinformation som ska registreras i systemet.

Standardhjälpsökväg

Standardhjälpsökvägen omfattar egna, systemtäckande och inbyggda adresser. *Språk* är normalt `C`.

Egen adress	*Hemkatalog*`/.dt/help`
Systemtäckande adress	`/etc/dt/appconfig/.dt/help/`*språk*
Inbyggd adress	`/usr/dt/appconfig/help/`*språk*

Hur programsökvägen påverkar hjälpsökvägen

När en adress läggs till i programsökvägen, läggs en lämplig hjälpunderkatalog automatiskt till i hjälpsökvägen (se ”Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp” på sidan 119).

Om t ex programservern `hosta:` läggs till i programsökvägen, läggs katalogen `hosta:/etc/dt/appconfig/help/`*språk* automatiskt till i hjälpsökvägen.

Hjälpsökvägens miljövariabler

Hjälpsökvägen sätts samman utifrån inbyggda adresser och följande indatavariabler:

DTSPSYSHELP	Indatavariabel för systemtäckande hjälpsökväg
DTSPUSERHELP	Indatavariabel för egen hjälpsökväg

Använd dessa indatavariabler för att ange adresser utanför programsökvägen.

Den assemblerade hjälpsökvägen anges av utdatavariabeln DTHELPSEARCHPATH.

Syntax för hjälpsökvägens indatavariabler

Syntaxen för variablerna DTSPSYSHELP och DTSPUSERHELP är:

VARIABEL=adress `[,adress...]`

där *adress* har syntaxen:

/sökväg	Anger katalogen i det lokala systemet (sessionsserver). Använd den här syntaxen för att lägga till en lokal katalog.

Om du vill ange en adress i ett annat system använder du dess nätverksfilnamn, t ex `/nfs/servera/projects/help`.

Hur hjälpsökvägen sätts samman

Värdet för hjälpsökvägen (DTHELPSEARCHPATH) skapas genom att följande adresser sätts samman, listade i prioritetsordning:

- Adresser som anges med variabeln DTSPUSERHELP
- Adresser som härleds från variabeln DTSPUSERAPPHOSTS

- Egen standardadress: *Hemkatalog*/.dt/help
- Standardadress: /etc/dt/appconfig/help/*språk*
- Adresser som anges med variabeln DTSPSYSHELP
- Adresser som härleds från variabeln DTSPSYSAPPHOSTS
- /usr/dt/appconfig/help/*språk*

Lokalanpassade sökvägar

Utdatavariablerna omfattar poster för både lokalanpassade adresser och standardadresser (C).

Standardprogramsökvägen är t ex:

Hemkatalog/.dt/appmanager
/etc/dt/appconfig/appmanager/*språk*
/etc/dt/appconfig/appmanager/C
/usr/dt/appconfig/appmanager/*språk*
/usr/dt/appconfig/appmanager/C

där *språk* är värdet för miljövariabeln LANG.

För varje definitionsområde (systemtäckande och inbyggt) har den språkspecifika adressen prioritet över standardadressen.

7

Introduktion till funktioner och datatyper

8

Funktioner och *datatyper* är kraftfulla komponenter som integrerar program med skrivbordet. Med hjälp av dessa kan du skapa ett användargränssnitt för att starta program och manipulera deras datafiler.

Introduktion till funktioner	*128*
Introduktion till datatyper	*135*

Det här kapitlet handlar om:

- Varför du kan behöva skapa funktioner och datatyper för program.
- Hur funktioner och datatyper förhåller sig till varandra.
- Hur funktioner och datatyper förhåller sig till utskrift från skrivbordet.

Se även

Procedurerna och reglerna för att skapa funktioner och datatyper beskrivs i tre kapitel i handboken.

- I Kapitel 9 beskrivs hur du skapar funktioner och datatyper med skrivbordsprogrammet Skapa funktion.

  Du använder Skapa funktion för att skapa funktioner och datatyper till de flesta progam, utan att du behöver lära dig syntaxreglerna för definitionerna.

8

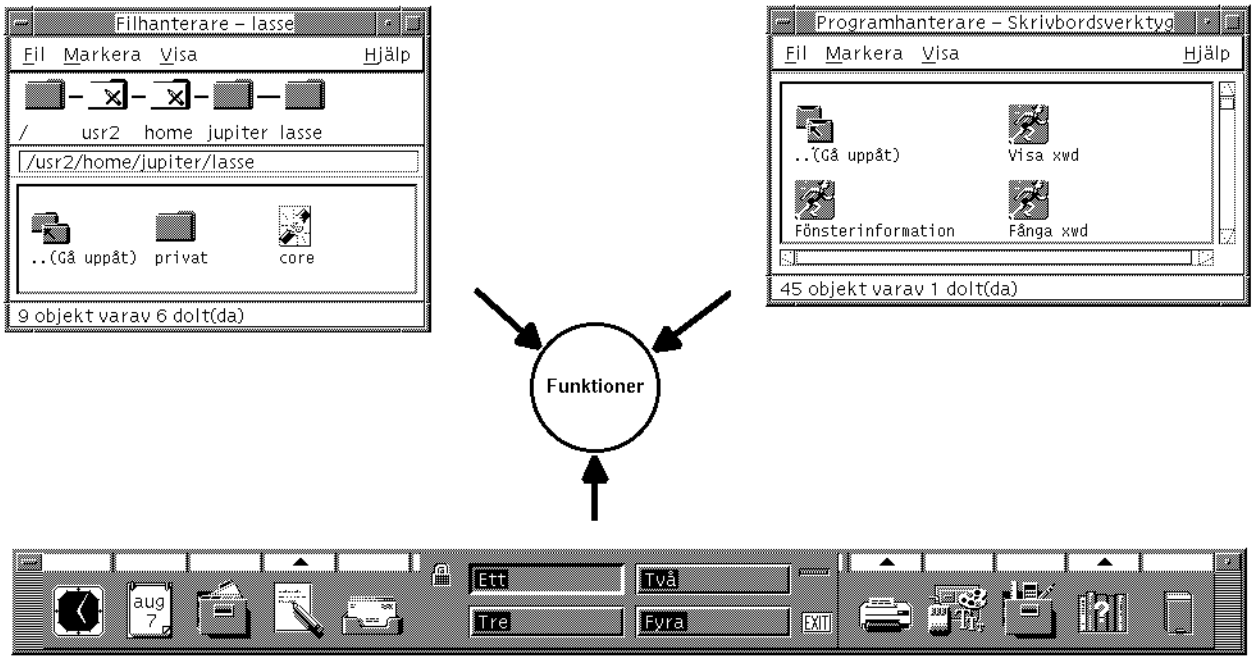
- I Kapitel 10 och Kapitel 11 förklaras hur du skapar funktioner och datatyper manuellt genom att skapa och redigera konfigurationsfiler.

  Du måste skapa funktioner och datatyper manuellt när du vill använda avancerade funktioner som inte kan användas i Skapa funktion.

Introduktion till funktioner

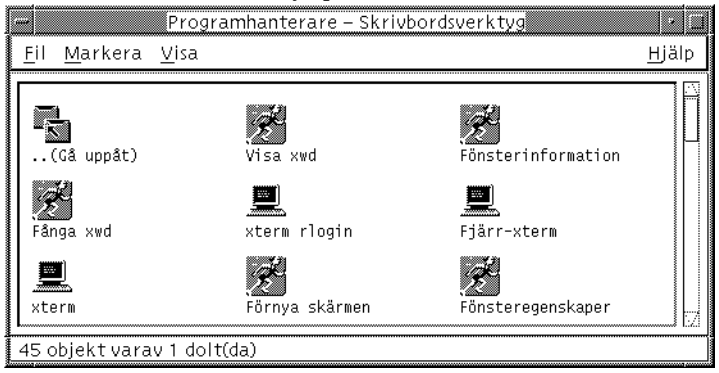
Funktioner är instruktioner som automatiserar skrivbordsuppgifter, t ex att köra program och öppna datafiler. Funktioner fungerar på ungefär samma sätt som makron och programmeringsfunktioner. Varje funktion har ett namn som används för att köra funktionen.

När du definierar en funktion kan den användas för att anpassa skrivbordets användargränssnitt så att det blir enklare att utföra olika uppgifter. På skrivbordet kan du också koppla komponenter i användargränssnittet, t ex ikoner, frontpanelsikoner och menyalternativ till funktioner.



figur 8-1 Hur funktioner används

Programgruppen Skrivbordsverktyg i programhanteraren innehåller t ex ikoner som startar olika hjälpmedel.



figur 8-2 Funktionsikoner i programgruppen Skrivbordsverlktyg

Var och en av dessa ikoner kör en funktion när man dubbelklickar på ikonen. Här är t ex en del av definitionen till funktionen som körs när användaren dubbelklickar på ikonen för Xwd Display. Funktionen definieras i konfigurationsfilen `/usr/dt/appconfig/types/`*språk*`/xclients.dt`.

```
ACTION Xwud
{
    LABEL           Xwd Display
    TYPE            COMMAND
    EXEC_STRING     /usr/bin/X11/xwud -noclick -in \
                    %(File)Arg_1"Visa Xwd-filen:"%
    …
}
```

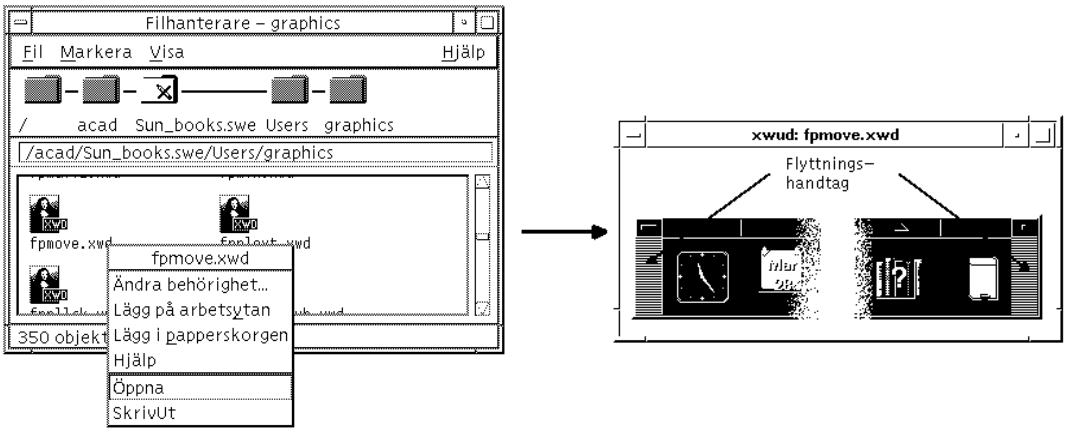
Kommandot i funktionens `EXEC_STRING` körs när användaren dubbelklickar på ikonen.

Frontpanelen använder också funktioner. Här är t ex en del av definitionen till ikonen Terminal på panelundermenyn Egna program. Ikonen definieras i konfigurationsfilen `/usr/dt/appconfig/types/`*språk*`/dtwm.fp`:

```
CONTROL Term
{
    ICON            Fpterm
    LABEL           Terminal
    PUSH_ACTION     Dtterm
    …
}
```

Fältet PUSH_ACTION anger att funktionen ska köras när användaren klickar på kontrollen, i det här fallet funktionen Dtterm.

Ett annat vanligt användningsområde för funktioner är i menyer. Datafiler har normalt funktioner på Markerade-menyn i filhanteraren. XWD-filer (filer med namn som slutar på .xwd eller .wd) har t ex funktionen Öppna som visar skärmbilden genom att köra funktionen Xwud.



figur 8-3 Funktionen Öppna för filer med datatypen XWD

Funktionerna på Markerade-menyn anges i datatypsdefinitionen för XWD-filen. Definitionen finns i konfigurationsfilen /usr/dt/appconfig/types/*språk*/xclients.dt.

```
DATA_ATTRIBUTES XWD
{
    ACTIONS        Öppna,SkrivUt
    ICON           Dtxwd
    ...
}
```

Datatypen XWD och tillhörande Öppna- och SkrivUt-funktioner förklaras i ”Hur datatyper ansluter datafiler till funktioner” på sidan 136.

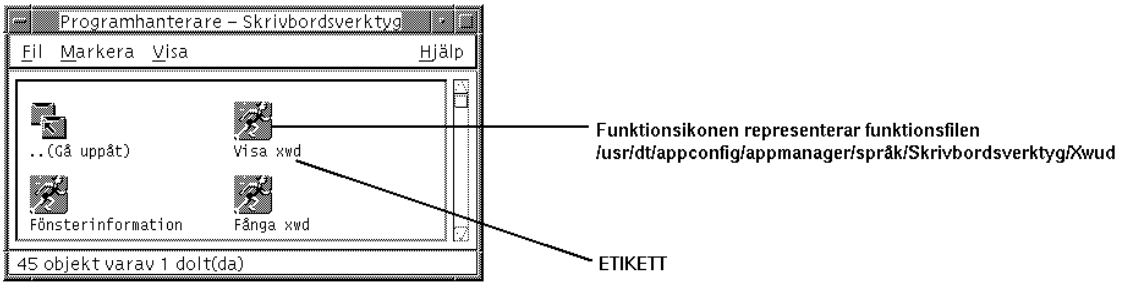
Hur funktioner skapar programikoner

Titta på ikonen Visa xwd i programgruppen Skrivbordsverktyg. Om du dubbelklickar på den körs X-klienten `xwud`. Ikonen representerar dock inte direkt den exekverbara `/usr/bin/X11/xwud` för `xwud`.

Ikonen med namnet Visa xwd visas i programgruppen eftersom det finns en fil i katalogen `Xwud` (se figur 8-4). Den här filen representerar en underliggande funktion med samma namn (Xwud). I funktionsdefinitionen är funktionsnamnet det namn som följder efter `ACTION`-nyckelord:

```
ACTION Xwud
{
    LABEL           Visa xwd
    TYPE            COMMAND
    WINDOW_TYPE     NO_STDIO
    EXEC_STRING     /usr/bin/X11/xwud -noclick -in \
                    %(File)Arg_1"Xwd-fil som ska visas:"%
    DESCRIPTION     Funktionen Visa xwd (Xwud) \
                    visar en xwd-fil som skapades med \
                    funktionen Fånga xwd (Xwd). Den använder \
                    kommandot xwud.
}
```

Filen kallas en *funktionsfil* eftersom den representerar en funktion. En fil är en funktionsfil när den är en körbar fil med samma namn som en ikon. Ikonen i programhanteraren (eller filhanteraren) kallas en *funktionsikon*, eller *programikon*, eftersom den startar ett program när du dubbelklickar på den.



figur 8-4 Programikonen (funktionsikonen) representerar en funktionsfil

När programhanteraren känner av en exekverbar fil letar den igenom funktionens databas för att se om det finns några funktioner vars namn matchar filnamnet. Om så är fallet vet programhanteraren att filen är en funktionsfil.

Innehållet i funktionfilen är irrelevant. En funktionsfil innehåller normalt kommentarer som beskriver dess skrivbordsfunktioner.

Obs! – *Funktionsfilen* är inte detsamma som *funktionsdefinitionsfilen*. *Funktionsfilen* är en fil med samma namn som funktionen. Den används för att skapa *programikonen* i filhanteraren eller programhanteraren. Funktionsdefinitionsfilen är filen med namnet *namn*`.dt` som innehåller funktionens definition.

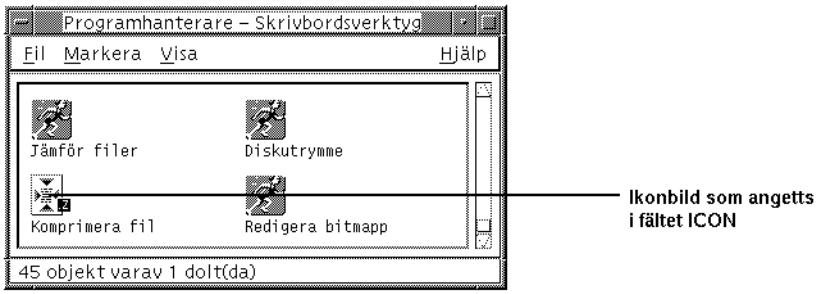
När skrivbordet upptäcker att en fil är en funktionsfil, används den underliggande funktionen för att definiera funktionsfilens utseende och metod.

- Fältet `EXEC_STRING` anger hur funktionsikonen ska fungera. I fallet med ikonen Xwd Display anger `EXEC_STRING` att funktionsikonen ska köra X-klienten `xwud` X med vissa kommandoradsargument.
- Fältet `LABEL` anger namnet på funktionsikonen.
- Fältet `DESCRIPTION` beskriver texten som visas när användaren begär objekthjälp.
- Programikonen Xwud använder standardikonbilden för funktioner, eftersom dess ikondefinition inte innehåller fältet `ICON` som anger en annan bild.

  Ikonen Compress File använder däremot en annan ikonbild eftersom dess underliggande funktionsikon innehåller ett `ICON-fält`:

  Exempel:

```
ACTION Compress
  LABEL      Komprimera fil
  ICON       Dtcmprs
  …
}
```



figur 8-5 Ikonbild som anges med fältet ICON i funktionsdefinitionen

Funktionen Xwud kallas en *kommandofunktion* eftersom dess definition innehåller kommandot (EXEC_STRING) som ska köras. Fältet TYPE i funktionsdefinitionen definierar funktionstypen.

Inledningsvis visas ikonen Visa xwd i programgruppen Skrivbordsverktyg. Du kan dock skapa ytterligare kopior av funktionsikonen i valfri katalog där du har skrivbehörighet. Så länge funktionsdefinitionen för Xwud är en del av databasen, blir alla exekverbara filer du skapar med namnet Xwud en funktionsfil som representerar den funktionen. Ikonen i program- eller filhanteraren används för att köra funktionen.

Hur funktioner använder datafiler som argument

Ett *argument* till ett kommando är normalt en fil som kommandot verkar med. Funktioner kan skrivas för att acceptera filargument.

T ex EXEC_STRING för funktionen Xwud anger att ett filargument krävs:

```
EXEC_STRING     /usr/bin/X11/xwud -noclick -in \
                %(File)Arg_1"Xwd-fil som ska visas:"%
```

Termen Arg står för ordet *argument*. Syntaxen Arg_1 innebär det första argumentet, och (File) innebär att funktionen behandlar argumentet som en fil.

Det enklaste sättet för användaren att tillhandahålla ett filargument är att släppa en datafil på programikonen. Skrivbordet känner av sökvägen till den släppta filen och placerar den i kommandoraden i stället för texten mellan symbolerna % (`%(File)Arg_1"Xwd-fil som ska visas:"%`). Kommandot som exekveras blir då:

`/usr/bin/X11/xwud -noclick -in` *filsökväg*

När användaren dubbelklickar på programikonen känner skrivbordet av från `EXEC_STRING` att ett filargument krävs, och visar en dialogruta som ber användaren att skriva ett filnamn eller en sökväg. I fallet med funktionen Xwud blir ledtexten:

`Xwd-fil som ska visas:`

Filnamnet eller sökvägen som användaren anger används som filargument.

Ytterligare användningsområden för funktioner

Förutom att starta program, används funktioner på skrivbordet för att skapa funktionalitet i:

- Frontpanelen.

  Definitionen av en ikon på frontpanelen omfattar fält som anger vilken funktion som körs när användaren klickar på ikonen eller släpper en fil på den. Mer information finns i ”Definiera kontroller på frontpanelen” på sidan 228.

- Menyer.

  Med syntaxen till definitionerna för Fönster- och Arbetsyta-menyn kan du ange vilken funktion som ska köras av ett menyalternativ. Mer information finns i ”Arbetsytehanterarens menyer” på sidan 246 och kommandobeskrivningen till `dtwmrc(4)`.

- Kommunikation mellan program.

  Ett program kan vara utformat för att skicka och ta emot information med en speciell typ av funktion som kallas ToolTalk-meddelande (`TT_MSG`). `TT_MSG`-funktioner beskrivs i skrivbordsdokumentationen till utvecklarmiljön.

Introduktion till datatyper

När användaren skapar en ny datafil varierar filikonens utseende och metod i filhanteraren, beroende på den typ av datafil som användaren har skapat. Med skrivbordets datatypsmekanism går det att skapa anpassat utseende och metod för filer och kataloger.

Vad är en datatyp?

En datatyp är en konstruktion som definieras i skrivbordets databas. Nedan finns t ex definitionen till datatypen XWD. Definitionen finns i konfigurationsfilen `/usr/dt/appconfig/types/`*`språk`*`/xclients.dt`:

```
DATA_ATTRIBUTES XWD
{
    ACTIONS        Open,Print
    ICON           Dtxwd
    NAME_TEMPLATE  %s.xwd
    MIME_TYPE      application/octet-stream
    SUNV3_TYPE     xwd-file
    DESCRIPTION    Den här filen innehåller en bild i formatet XWD. \
                   Dessa filer skapas normalt genom att \
                   bilder tas av fönster med funktionen XwdCapture. \
                   Dess datatyp heter XWD. Namn på XWD-filer \
                   slutar på ‘.xwd’ eller ‘.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
}
```

Alla datatypsdefinitioner består av två delar:

DATA_ATTRIBUTES	Beskriver datatypens utseende och metod.
DATA_CRITERIA,	Anger regler (namn eller innehåll) för hur filer som hör till datatypen kategoriseras.

Fältet DATA_ATTRIBUTES_NAME innehåller villkor till attributen.

Det kan finnas flera DATA_CRITERIA för ett DATA_ATTRIBUTE. Datatypen XWD har t ex två villkor för att ange två olika namnvillkor (NAME_PATTERN)—namn som slutar på .xwd eller .wd.

Hur datatyper ansluter datafiler till funktioner

Ta datatypen XWD som exempel. Användaren skapar en fil av XWD-typ genom att ge filen något av filtypstillägggen .xwd eller.wd. Skrivbordet använder filnamnet som ett *villkor* för att beteckna filen som den typen.

Varje fil av datatypen XWD får:

- En unik ikonbild som hjälper användaren att känna igen datafilerna.
- Objekthjälp som beskriver datatypen.
- En anpassad Markerade-meny i filhanteraren som innehåller funktionerna Open och Print. Funktionen Open för XWD-filer kör funktionen Xwud.

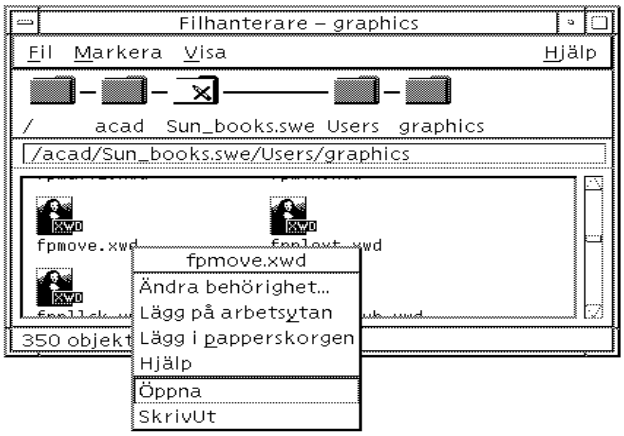
Köra funktioner från Markerade-menyn

Markerade-menyn i filhanteraren är bara aktiv när en fil eller en katalog är markerad. Vilket kommando som finns längst ned på Markerade-menyn beror på datatypen. Om en XWD-fil är markerad finns alternativen Open och Print på Markerade-menyn.

Fältet ACTIONS i datatypsdefinitionen anger vilka kommandon som läggs till längst ned på datatypens Markerade-meny.

```
DATA_ATTRIBUTES XWD
{
          ACTIONS           Open,Print
          …
}
```

Innehållet på Markerade-menyn beror på datatypen. Många olika datatyper har dock funktionen Öppna. Det innebär att när du markerar en fil med en viss datatyp i filhanteraren och visar Markerade-menyn, finns kommandot Öppna på menyn.



figur 8-6 Markerade-menyn för en XWD-fil

Funktionen Öppna kör normalt programmet som datafilen hör till. Om du t ex öppnar en XWD-fil körs funktionen Xwud, vilken i sin tur kör X-klienten xwud för att visa skärmbilden. Med andra ord, för datatypen XWD är funktionen Öppna synonym med funktionen Xwud. Samma sak gäller när du öppnar en fil med datatypen TEXTFILE, då textredigeraren körs. När du öppnar en BM- (bitmönster) eller PM-fil (bildpunktsmönster) körs ikonredigeraren.

Du kan skapa olika Öppna-funktioner som utför olika saker med två möjligheter till funktionsdefinitioner:

- Funktionstilldelning.

  Med funktionstilldelning skapar du en funktion som kör en annan funktion, i stället för att köra ett kommando. Du kan t ex skapa en Öppna-funktion som tilldelar (kör) funktionen Xwud.

- Datatypsrestriktioner för en funktion.

  Funktionsdefinitioner kan omfatta fältet ARG_TYPE som begränsar funktionen till vissa datatyper. Du kan t ex ange att Öppna-funktionen som kör funktionen Xwud bara gäller för fält med datatypen XWD.

Nedan finns definitionen av den funktion som tilldelar Öppna-funktionen till funktionen Xwud för datatypen XWD. Den finns i databaskonfigurationsfilen `/usr/dt/appconfig/types/C/xclients.dt`:

```
ACTION Open
{
    LABEL       Open
    ARG_TYPE    XWD
    TYPE        MAP
    MAP_ACTION  Xwud
}
```

Fältet `TYPE` anger att detta är en tilldelningsfunktion. Fältet `MAP_ACTION` anger att funktionen kör funktionen Xwud. Fältet `ARG_TYPE` anger att funktionen bara gäller för filer vars datatyp är XWD.

Jämför föregående definition av funktionen Öppna med nästa definition, som finns i databasfilen `/usr/dt/appconfig/types/C/dt.dt`:

```
ACTION Open
{
    LABEL           Open
    ARG_TYPE        BM
    TYPE            MAP
    MAP_ACTION      Dticon
}
```

Den här definitionen gäller för filer med datatypen (`ARG_TYPE`) BM (bitmönsterfiler). Definitionen tilldelar funktionen Öppna till funktionen Dticon, som kör ikonredigeraren.

Definiera hur datatypens dubbelklickningar fungerar

Vad som händer när du dubbelklickar för en viss datatyp definieras av den första posten i fältet `ACTIONS`. För datatypen XWD innebär ett dubbelklick att funktionen Öppna körs, vilken i sin tur kör funktionen Xwud.

Släppa en datafil på en funktionsikon

När användaren släpper en datafil på en funktionsikon körs funktionen som använder den datafilen som ett argument (se ”Hur funktioner använder datafiler som argument” på sidan 133).

När t ex datafilen XWD släpps på ikonen Visa xwd körs funktionen Xwud med den datafilen som argument. Detta kör X-klienten xwud med den datafilen.

Skriva ut datatyper från skrivbordet

En datafil kan skrivas ut på följande sätt från skrivbordet:

- Använd kommandot Skriv ut, om möjligt, på filhanterarens Markerade-meny.
- Släpp en datafil på skrivbordets målområden för utskrift (skrivarikonen på frontpanelen eller en skrivarikon i utskriftshanteraren).

Dessutom går det att skriva ut från många program.

När du skriver ut från skrivbordet används funktioner som heter Skriv ut. Skriv ut är, på samma sätt som Öppna, ett funktionsnamn som används till flera olika typer av data. Därför använder Skriv ut-funktioner funktionstilldelning och fältet ARG_TYPE för att anpassa utskriften efter varje datatyp.

Nedan finns t ex funktionen Skriv ut för datatypen XWD. Definitionen finns i /usr/dt/appconfig/types/*språk*/xclients.dt:

```
ACTION Print
{
    LABEL          Print
    ARG_TYPE       XWD
    TYPE           MAP
    MAP_ACTION     NoPrint
}
```

Den här Skriv ut-funktionen, som är specifik för XWD-filer, tilldelas till en NoPrint-funktion. NoPrint är en speciell funktion som definieras i /usr/dt/appconfig/types/*språk*/dt.dt. Funktionen NoPrint visar en dialogruta som anger att den här datatypen inte kan skrivas ut.

Jämför funktionen XWD Print med följande Print-funktion för PCL-filer:

```
ACTION Print
    LABEL      Print
    ARG_TYPE   PCL
    TYPE       MAP
    MAP_ACTION PrintRaw
}
```

Funktionen PrintRaw, som definieras i konfigurationsfilen `/usr/dt/appconfig/types/`*språk*`/print.dt`, innehåller kommandoraden för att skriva PCL-filerna:

```
ACTION PrintRaw
{
    TYPE           COMMAND
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING    /usr/dt/bin/dtlp -w %(File)Arg_1%
}
```

Skapa funktioner och datatyper med Skapa funktion 9

Skapa funktion är ett verktyg som används för att skapa:

- En funktion som startar ett program
- En eller flera datatyper för ett programs datafiler
- Funktioner för att öppna och skriva ut programmets datafiler

Skapa funktion är också användbart för att skapa enkla funktioner som kör operativsystemkommandon och skalskript.

Vad Skapa funktion gör	*141*
Begränsningar för Skapa funktion	*142*
Skapa en funktion och datatyp för ett program med Skapa funktion	*143*

Referensinformation finns i kommandobeskrivningen till `dtcreate(1X)`.

Vad Skapa funktion gör

Skapa funktion omfattar ett huvudfönster och en uppsättning dialogrutor som du använder för att skapa en funktion och tillhörande datatyper.

Skapa funktion gör följande:

- Skapar en funktionsdefinition som kör ett kommando.
- Skapar filen *Hemkatalog*`/.dt/types/`*funktionsnamn*`.dt`. Den här filen lagrar funktionen och datatypsdefinitionerna som skapas för programmet.

- Skapar en *funktionsfil* i användarens hemkatalog. Funktionsfilen är en exekverbar fil med samma namn som funktionen.

  Funktionsfilen representeras i filhanteraren av en *programikon* som startar programmet när man dubbelklickar på den.

  Du kan också göra funktionsikonen till ett målområde genom att ange en släppbar datatyp när du skapar funktionen.
- Skapar en eller flera datatyper för programmets datafiler (valfritt).
- Skapar en Öppna-funktion för varje datatyp.
- Skapar en Skriva ut-funktion för varje datatyp (valfritt).
- Startar om databasen med funktioner och datatyper. Detta gör att funktionerna och datatyperna träder i kraft omedelbart.

Begränsningar för Skapa funktion

Skapa funktion är utformat för att skapa funktioner och datatyper för program som ska köras. Funktioner och datatyper kan dock vara mycket flexibla, och omfatta ytterligare funktionalitet som bara kan nås om du skapar definitionerna manuellt.

Mer information finns i:

- Kapitel 10, ”Skapa funktioner manuellt”
- Kapitel 11, ”Skapa datatyper manuellt”

Funktionsbegränsningar

Du kan inte använda Skapa funktion för att skapa funktioner för ett program om följande villkor är sanna:

- Kommandoraden kräver ett argument som inte består av en fil (parameter).

  Du kan t ex inte använda Skapa funktion för att skriva en funktion för kommandot:

  `lp -d`*`enhet filnamn`*

  där användaren måste ange *enhet* varje gång kommandot ska köras.
- Programikonen måste ha ett annat namn än funktionen.

Du kan t ex inte använda Skapa funktion för att tillhandahålla en lokal språkversion av en befintlig funktion.

- Funktionen kräver någon av specialfunktionerna i funktionsdatabasen.

  Exempel på sådana specialfunktioner är funktioner som:
  - Startar kommandon i fjärrsystem
  - Anropar andra funktioner
  - Måste köras som en annan användare (t ex en superanvändare)
  - Använder tilldelningsfunktionen
  - Fungerar olika beroende på antalet filargument som anges för funktionen

Datatypsbegränsningar

Du kan inte använda Skapa funkton för att skapa datatypen till ett program om något av följande villkor är sanna:

- Datatypen måste ha andra funktioner än Öppna och Skriv ut kopplade till sig.
- Funktionen Öppna för datatypen är inte funktionens kommando.

  Du kan t ex inte använda Skapa funktion för att skapa datatypen som tillhandahåller en unik ikon för katalogen som representerar programmets programgrupp.

Skapa en funktion och datatyp för ett program med Skapa funktion

Det finns några saker som du måste känna till om programmet innan du kör Skapa funktion.

- Kommandoraden för att starta programmet.

  Du måste veta om kommandoraden innehåller ett obligatoriskt filargument, ett valfritt filargument eller inget filargument.

  Om programmet kräver argument som inte består av en fil, kan du inte använda Skapa funktion.

- De typer av datafiler som ett program accepterar.

  Vissa program accepterar bara en typ av data. Andra (t ex en ASCII-redigerare eller grafikredigerare) accepterar flera datatyper.

- På vilket sätt programmet identifierar sina datatyper.

  Detta kan vara en namngivningskonvention (t ex filnamn som slutar med `.doc`), och/eller kan bero på innehållet i filen. Även om programmet inte använder någon namngivningskonvention, kan du ange en för funktionsikonen.

- Valfritt: Kommandoraden för att skriva ut filer

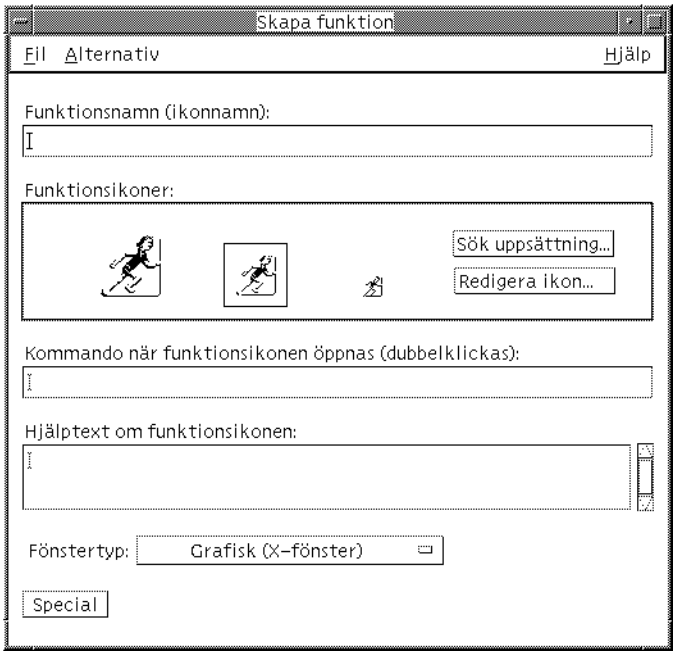
▼ Så här skapar du en funktion för ett program

1. Dubbelklicka på Skapa funktion i programgruppen Skrivbordsprogram.



figur 9-1 Ikonen för Skapa funktion i programhanteraren

Detta är huvudfönstret i Skapa funktion.



figur 9-2 Huvudfönstret i Skapa funktion

2. Skriv namnet på funktionsikonen i textfältet Funktionsnamn.
3. Använd verktygen Funktionsikoner för att ange ikon för programmet. Inledningsvis visas standardikonen.
   - Om du vill välja en annan, befintlig ikon klickar du på Sök uppsättning för att visa dialogrutan Sök uppsättning. Mer information finns i ”Ange ikon med dialogrutan Sök uppsättning” på sidan 153.
   - Om du vill skapa nya ikoner väljer du Redigera ikon för att köra ikonredigeraren.
4. Skriv i textfältet Kommando när funktionsikonen öppnas, vilket kommando som ska starta programmet.

   Använd syntaxen $*n* för filargument, t ex:

```
emacs
bitmap $1
diff $1 $2
lp -oraw $1
```

Om kommandoraden innehåller ett filargument f ($*n*), blir funktionikonen ett målområde för filer.

Kommandoraden skickas inte till ett skal såvida du inte särskilt anger att skal ska användas. Följande rader använder skalbearbetning:

```
/bin/sh -c 'ps | lp'
/bin/sh -c 'spell $1 | more'
```

5. Skriv texten till objekthjälpen för funktionsikonen i textfältet Hjälptext om funktionsikonen.

   Texten radbryts automatiskt i textfältet. Dessa radbrytningar bevaras dock inte när objekthjälpen visas på skärmen. Om du vill infoga hårda radbrytningar använder du `\n`.

6. Välj det fönsterstöd som funktionen kräver på alternativmenyn Fönstertyp.

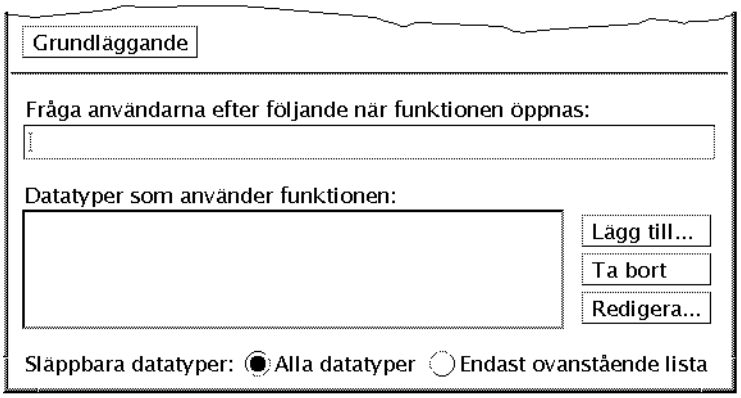
Grafiskt (X-fönster)	Programmet skapar sitt eget fönster
Terminal (stängs automatiskt)	Programmet körs i ett terminalemulatorfönster som stängs automatiskt när användaren avslutar programmet
Terminal (stängs manuellt)	Programmet körs i ett terminalemulatorfönster som är öppet tills användaren stänger det
Inget fönster	Programmet skapar inte något fönster

7. Fortsätt så här:
   - Om programmet har datafiler och du vill skapa en eller flera datatyper för dem, läser du nästa avsnitt “Så här skapar du en eller flera datatyper för programmet”.
   - Om du inte vill skapa datatyper, sparar du funktionen genom att välja Spara på Fil-menyn. Testa sedan den nya funktionen genom att dubbelklicka på ikonen i hemkatalogen.

▼ Så här skapar du en eller flera datatyper för programmet

1. Definiera funktionen för programmet med proceduren i förgående avsnitt, “Så här skapar du en funktion för ett program.”

2. Klicka på Special i fönstret Skapa funktion för att visa fler alternativ.



figur 9-3 Specialfunktioner i huvudfönstret till Skapa funktion

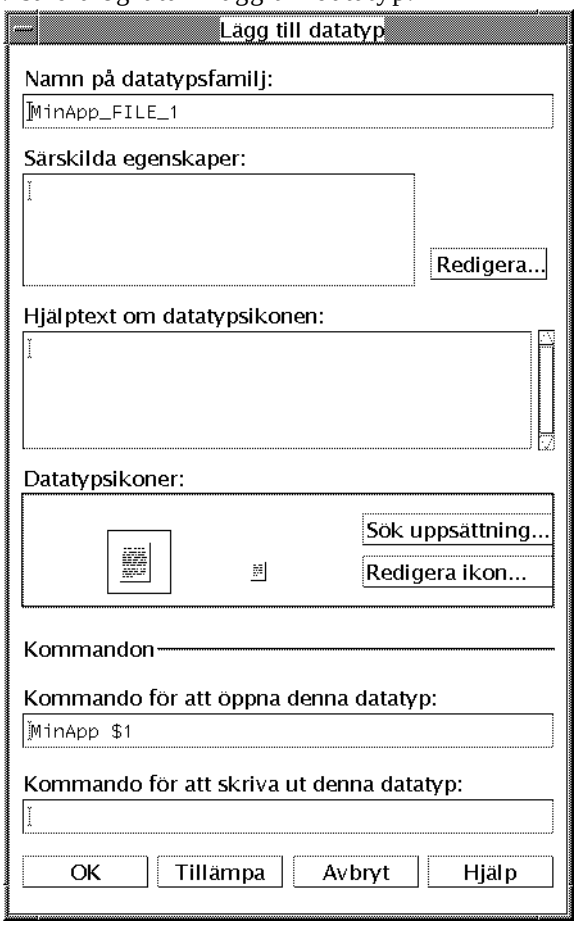
3. Om du vill att programikonen ska fråga efter ett filargument när någon dubbelklickar på den, skriver du texten till ledtexten i textfältet ”Fråga användarna följande när funktionen öppnas”.

   Använd följande riktlinjer för textfältet:
   - Du måste använda det här textfältet om programmets kommandorad har ett *obligatoriskt* filargument.
   - Fältet måste vara tomt om kommandoraden inte omfattar filargument.
   - Om filargumentet i programmets kommandorad är valfritt kan du välja. Om du anger en ledtext visar funktionsikonen den när användarna dubbelklickar på ikonen. Om du inte anger någon ledtext, körs funktionen med en tom sträng som filargument.

4. Ange de typer av filer som funktionen accepterar som argument:
   - Om funktionen accepterar alla datatyper väljer du Alla datatyper.
   - Om funktionen endast accepterar de datatyper du skapar för programmet, väljer du Endast ovanstående lista.

   Inledningsvis är listan Datatyper som använder funktionen tom. Allt eftersom du skapar datatyper för programmet läggs de till i listan.

5. Klicka på Lägg till bredvid listan Datatyper som använder funktionen för att visa dialogrutan Lägg till datatyp.

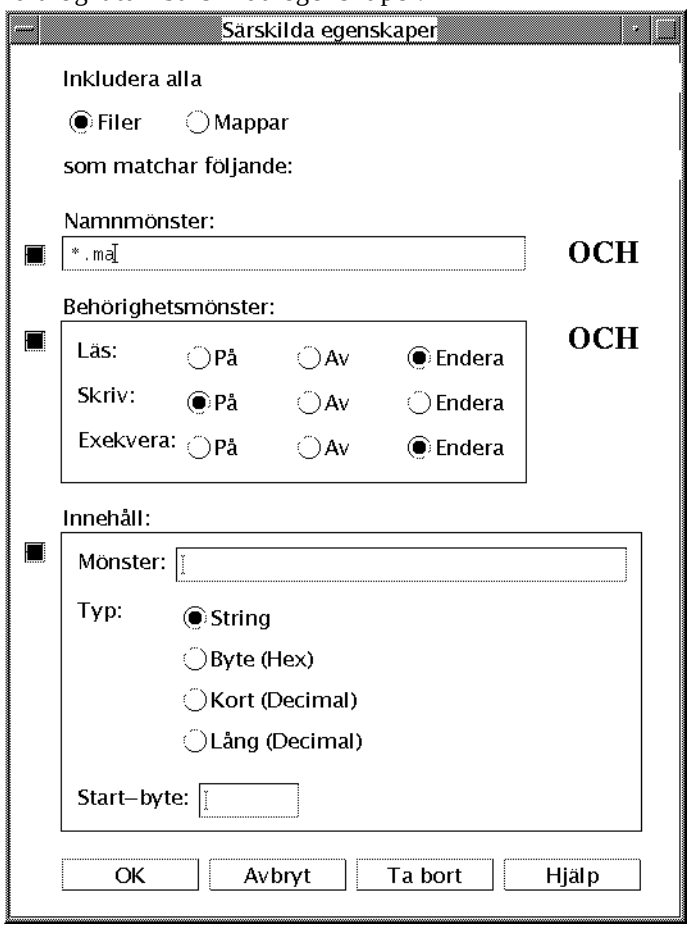


figur 9-4 Dialogrutan Lägg till datatyp i Skapa funktion

6. Valfritt: Om du inte vill använda datatypens standardnamn skriver du ett nytt namn på datatypen i textfältet Namn på datatypsfamilj.

   Namnet får inte innehålla mellanslag. Datatypsnamnet är inte synligt för programmets användare, utan används i funktions-/datatypsdatabaser för att identifiera datatypsdefinitionen.

7. Klicka på Redigera bredvid rutan Särskilda egenskaper för att visa dialogrutan Särskilda egenskaper.

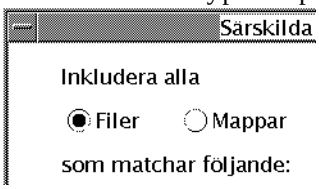


figur 9-5 Dialogrutan Särskilda egenskaper i Skapa funktion

Datatypens egenskaper är villkoren som används för att skilja en datatyp från en annan. Du kan välja ett eller flera av följande villkor:

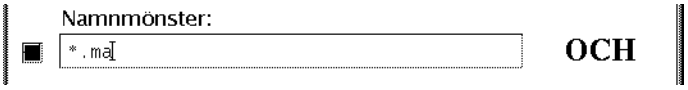
Filer eller Mappar	Datatypen gäller endast för filer och mappar
Namnmönster	Datatypsnamnet baseras på filnamnet
Behörighetsmönster	Behörigheterna läsa, skriva och exekvera
Innehåll	Innehållet i angiven del av filen

8. Markera om datatypen representerar en fil eller mapp.



figur 9-6 Ange en fil- eller katalogegenskap för datatypen.

9. Om datatypen är beroende av namnet markerar du kryssrutan Namnmönster och fyller i textfältet.

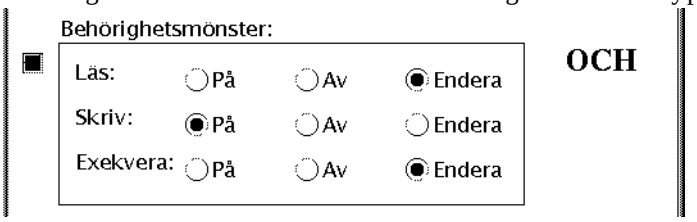


figur 9-7 Ange filnamnsegenskap för en datatyp

Du kan använda jokertecknen * och ?:

*	Motsvarar valfri teckensekvens
?	Motsvarar ett enstaka tecken

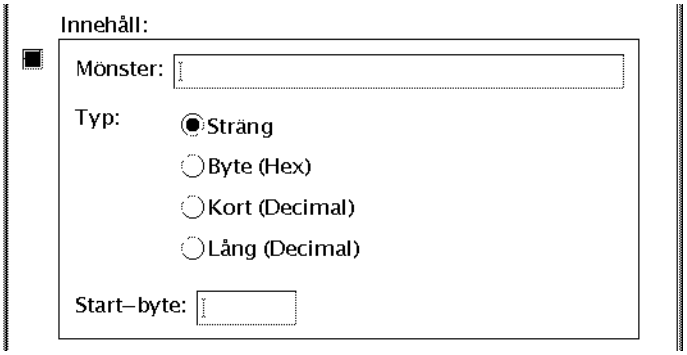
10. Om datatypen är beroende av behörigheter markerar du kryssrutan Behörighetsmönster och markerar behörighet för datatypen.



figur 9-8 Ange behörighetsegenskaper för datatypen

På	Filen måste ha angiven behörighet
Av	Filen får inte ha angiven behörighet
Endera	Angiven behörighet har ingen betydelse

11. Om datatypen är beroende av innehållet markerar du kryssrutan Innehåll och anger den information som krävs: Mönster att söka efter och typ av innehåll. Du kan också ange i vilken byte-position som sökningen ska starta.



figur 9-9 Ange innehållsegenskaper för en datatyp

Obs! – Om du använder innehållsbaserad datatyp kan det påverka systemets prestanda.

---

12. Klicka på OK för att stänga dialogrutan Särskilda egenskaper.

    Egenskaperna visas i fältet Särskilda egenskaper med dessa koder:

d	En katalog
r	Filen har läsbehörighet
w	Filen har skrivbehörighet
x	Filen har exekveringsbehörighet
!	Logiska operatorn INTE
&	Logiska operatorn OCH

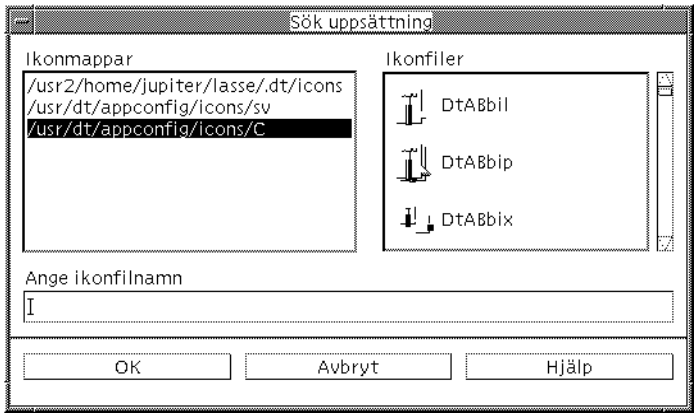
13. Skriv hjälptexten för datafilen i textfältet Hjälptext.

14. Använd verktyget Datatypsikoner för att specificera ikon för programmet. Inledningsvis visas standardikonen.
    - Om du vill välja en annan, befintlig ikon klickar du på Sök uppsättning för att visa dialogrutan Sök uppsättning. Mer information finns i ”Ange ikon med dialogrutan Sök uppsättning” på sidan 153.
    - Om du vill skapa nya ikoner klickar du på Redigera ikon för att köra ikonredigeraren.
15. Bekräfta kommandot i textfältet Kommando för att öppna denna datatyp. Detta kommando exekveras när användaren dubbelklickar på en datafil.
16. Valfritt: Om programmet har ett utskriftskommando för att skriva ut datafiler från kommandoraden, skriver du kommandot i textfältet Kommando för att skriva ut denna datatyp, med syntaxen $*n* för ett filargument.
17. Gör något av följande för att spara datatypsdefinitionen:

- Klicka på OK för att spara datatypen och stäng dialogrutan Lägg till datatyp.
- Klicka på Tillämpa för att spara datatypen utan att stänga dialogrutan Lägg till datatyp. Då kan du fortsätta att definiera andra datatyper för funktionen.

Ange ikon med dialogrutan Sök uppsättning

Dialogrutan Sök uppsättning visas när du klickar på Sök uppsättning i huvudfönstret till Skapa funktion, eller i fönstret Lägg till datatyp. Använd den här dialogrutan för att ange vilken ikon som ska användas till funktionen eller datatypen.



figur 9-10 Dialogrutan Sök uppsättning

I dialogrutan Sök uppsättning kan du ange en uppsättning ikonbildsfiler som finns i:

- En katalog i ikonsökvägen. I listan Ikonmappar finns alla kataloger i ikonsökvägen.
- Ett registreringspaket som ska integreras med skrivbordet med hjälp av `dtappintegrate`. Dessa ikoner finns ännu inte i någon katalog i ikonsökvägen, men placeras där av `dtappintegrate`.

Obs! – Funktions- och datatypsdefinitioner som skapas med Skapa funktion använder basnamnet på ikonfilen (namnet minus suffixet för typ och storlek). Ikoner för funktioner och datatyper som skapas med Skapa funktion måste placeras i kataloger i ikonsökvägen.

▼ *Så här anger du en uppsättning ikoner som finns i ikonsökvägen*

1. I listan Ikonmappar i dialogrutan Sök uppsättning dubbelklickar du på mappsökvägen som innehåller ikonen.

   Listan Ikonfiler visar alla ikoner som finns i mappen.

2. Klicka på ikonen du vill använda i listan Ikonfiler.

   Basnamnet på ikonfilen placeras i textfältet Ange ikonfilnamn.

3. Klicka på OK.

▼ *Så här anger du en ikon i ett registreringspaket*

Om du är systemadministratör eller programmerare som skapar ett registreringpaket, finns ikonfilerna inledningsvis i en katalog i registreringspaketet:

progrot`/dt/appconfig/icons/`*språk*

När du har registrerat med `dtappintegrate` kopieras ikonfilerna till `/etc/dt/appconfig/icons/`*språk*, som finns i ikonsökvägen.

Använd proceduren nedan för att ange ikoner som är en del av registreringspaketet:

1. I textfältet Ange ikonfilnamn i dialogrutan Sök uppsättning skriver du ett basnamn på ikonfilen.

2. Välj OK.

   Skapa funktion visar en dialogruta som meddelar att ikonerna inte fanns i katalogen i ikonsökvägen.

3. Välj Ingen ändring i dialogrutan som visas.

Skapa funktioner manuellt 10

Du kan skapa funktioner på två sätt:

- Använda skrivbordsprogrammet Skapa funktion
- Skapa en funktionsdefinition manuellt

För att skapa en funktion manuellt måste du redigera en databasfil. I det här kapitlet beskrivs hur du skapar funktionsdefinitioner manuellt.

Varför du måste skapa funktioner manuellt	*156*
Skapa en funktion manuellt: Allmänna steg	*157*
Skapa en exekveringsträng för COMMAND-funktioner	*167*
Fönsterstöd och terminalemulatorer för COMMAND-funktioner	*173*
Begränsa funktioner till vissa argument	*175*
Skapa funktioner som kör program i fjärrsystem	*178*
Använda variabler i funktions- och datatypsdefinitioner	*179*
Anropa funktioner från kommandoraden	*181*
Skapa lokalanpassade funktioner	*182*
Skapa funktioner för ToolTalk-program	*183*

10

Se även

- En introduktion till funktioner finns i Kapitel 8, ”Introduktion till funktioner och datatyper”.
- Mer information om hur du använder Skapa funktion finns i Kapitel 9, ”Skapa funktioner och datatyper med Skapa funktion”.
- Referensinformation om funktionsdefinitioner finns i kommandobeskrivningen till `dtactionfile(4)`.

Varför du måste skapa funktioner manuellt

Det finns tre grundtyper av funktioner:

- `COMMAND`
- `MAP`
- `TT_MSG`

Verktyget Skapa funktion är utformat för att skapa vissa typer av `COMMAND`- och `MAP`-funktoner. Alla `TT_MSG`-funktioner måste skapas manuellt.

Mer information finns i ”Begränsningar för Skapa funktion” på sidan 142.

COMMAND-funktioner

En *kommandofunktion* exekverar ett kommando som startar ett program eller hjälpmedel, kör ett skalskript eller exekverar ett operativsystemkommando. Definitionen på funktionen omfattar kommandot som ska exekveras (`EXEC_STRING`).

Verktyget Skapa funktion kan användas till att skapa de vanligaste typerna av kommandofunktioner. Det kan dock finnas tillfällen när du måste skapa funktionen manuellt. Du måste t ex skapa en `COMMAND`-funktion manuellt om funktionen anger:

- Flerfilsargument med olika ledtexter för varje argument.
- Funktionsanrop – möjligheten för funktioner att anropa andra funktioner.
- En metod beroende av antal argument – möjligheten att skapa en funktion som fungerar på olika sätt för olika antal filargument.
- En fjärrvärddator för exekvering – möjligheten att köra ett program i ett system som inte innehåller funktionsdefinitionen.

- Annan användare – möjlighet att köra funktionen som en annan användare (för att t ex fråga efter rotlösenord och sedan köra som rot).

MAP-funktioner

En *tilldelningsfunktion* är en funktion som är ‘‘tilldelad’’ en annan funktion i stället för att direkt ange ett kommando eller ett ToolTalk-meddelande.

Med tilldelning kan du ange alternativa namn på funktioner. En inbyggd kommandofunktion med namnet IconEditor startar Ikonredigeraren. Databasen innehåller också funktionen Open, som begränsas i definitionen till bitmönster- och bildpunktmönsterfiler (med fältet ARG_TYPE), som tilldelas funktionen IconEditor. På så sätt kan användaren starta Ikonredigeraren genom att välja en bitmönster- eller bildpunktsmönsterfil i filhanteraren och välja Öppna på Markerade-menyn.

Skapa funktion ger begränsad tilldelning för funktionerna Open och Print. Alla andra tilldelningsfunktioner måste skapas manuellt.

TT_MSG-funktioner (ToolTalk-meddelande)

TT_MSG-funktioner skickar ToolTalk-meddelanden. Alla TT_MSG-funktioner måste skapas manuellt.

Skapa en funktion manuellt: Allmänna steg

I det här avsnittet beskrivs hur du skapar en konfigurationsfil för en funktionsdefinition.

Konfigurationsfiler för funktioner

Konfigurationsfiler som innehåller funktionsdefinitioner måste uppfylla följande krav:

- Filerna måste följa namngivningskonventionen *namn*`.dt`
- Filerna måste finnas i databassökvägen (funktioner och datatyp). Standardsökvägen är:

Egna funktioner *Hemkatalog*`/.dt/types`

Systemtäckande funktioner `/etc/dt/appconfig/types/`*språk*

Inbyggda funktioner `/usr/dt/appconfig/types/`*språk*. Den här katalogen ska du inte använda.

Mer information om hur du ändrar sökvägen till funktioner och datatyper finns i ”Ange värde för en sökväg” på sidan 115.

▼ Så här skapar du en funktion manuellt

1. Öppna en befintlig databasfil eller skapa en ny. Se föregående avsnitt ”Konfigurationsfiler för funktioner”.
2. Skapa funktionsdefinitionen med syntaxen:

```
ACTION funktionsnamn
{
  TYPE      funktionstyp
  funktionsfält
  …
}
```

   där:

funktionsnamn Namn som används för att köra funktionen.

Funktionstyp `COMMAND` (standard), `MAP` eller `TT_MSG`.

funktionsfält Ett av de obligatoriska eller valfria fälten för den här typen av funktion. Alla fält består av ett nyckelord och ett värde.

Många av funktionsfälten beskrivs i det här kapitlet. Mer information finns i kommandobeskrivningen till dtactionfile(4).

3. Spara filen.
4. Om du vill att funktionsikonen ska ha en unik bild skapar du en ikon till funktionen. Standardadressen för ikoner är:
   - Egna ikoner: *Hemkatalog*`/.dt/icons`
   - Systemtäckande ikoner: `/etc/dt/appconfig/icons/`*språk*. Standardvärdet för *språk* är `C`.

   Mer information finns i ”Ange ikonbilden som används av en funktion” på sidan 163.
5. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

6. Skapa funktionsfilen för funktionen. Funktionsfilen skapar en ikon i filhanteraren eller programhanteraren som representerar funktionen. (Om funktionen är skriven för att starta ett program kallas ikonen för *programikon*.)

   Om du vill skapa en funktionsfil, skapar du en exekverbar fil med samma namn som ett *funktionsnamn*. Du kan placera den här filen i valfri katalog som du har skrivbehörighet till. Du kan skapa så många funktionsfiler du vill.

Exempel på hur du skapar en COMMAND-funktion

I följande steg skapas en egen funktion som startar ett faxprogram i fjärrsystemet AppServerA. Kommandot som startar faxprogrammet är:

```
/usr/fax/bin/faxcompose [filnamn]
```

1. Skapa filen *Hemkatalog*`/.dt/types/Fax.dt`.
2. Placera följande funktionsdefinition i filen:

```
ACTION FaxComposer
{
  TYPE           COMMAND
  ICON           fax
  WINDOW_TYPE    NO_STDIO
  EXEC_STRING    /usr/fax/bin/faxcompose -c %Arg_1%
  EXEC_HOST       AppServerA
  DESCRIPTION     Kör FaxComposer
}
```

   Fälten `WINDOW_TYPE` och `EXEC_STRING` beskriver hur funktionen fungerar.

`WINDOW_TYPE` — Nyckelordet `NO_STDIO` anger att funktionen inte behöver köras i ett terminalemulatorfönster.

Se ”Ange fönsterstöd för funktionen” på sidan 174.

`EXEC_STRING` — Syntaxen `%Arg_1%` accepterar en släppt fil. Om någon dubbelklickar på funktionsikonen, öppnar funktionen ett tomt faxkompositionsfönster.

Se ”Skapa en exekveringsträng för COMMAND-funktioner” på sidan 167.

3. Spara filen.

4. Använd ikonredigeraren för att skapa följande bildfiler i katalogen *Hemkatalog*/.dt/icons:
   - fax.m.pm, storlek 32 gånger 32 bildpunkter
   - fax.t.pm, storlek 16 gånger 16 bildpunkter
5. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.
6. Skapa en exekverbar fil med namnet FaxComposer i en katalog som du har skrivbehörighet till (hemkatalogen, t ex).

Exempel på hur du skapar en MAP-funktion

Anta att de flesta filer du faxar skapas med Textredigaren och är av datatypen TEXTFILE (filer med namnet *.txt).

Stegen nedan lägger till menyalternativet ”Fax” på datatypens Markerade-meny.

1. Öppna filen *Hemkatalog*/.dt/types/Fax.dt som skapades i föregående exempel.
2. Lägg till den här tilldelningsdefinitionen i filen:

```
ACTION Fax
{
  ARG_TYPE   TEXTFILE
  TYPE       MAP
  MAP_ACTION FaxComposer
}
```

3. Spara filen.
4. Kopiera definitionen för dataattribut för TEXTFILE från /usr/dt/appconfig/types/*språk*/dtpad.dt till en ny fil *Hemkatalog*/.dt/types/textfile.dt. Lägg till funktionen Fax i fältet ACTIONS.

```
DATA_ATTRIBUTES TEXTFILE
{
  ACTIONS    Open,Print,Fax
  ICON       Dtpenpd
  …
}
```

5. Spara filen.

6. Öppna programhanteraren och dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

▼ Så här startar du om databasen för funktioner och datatyper

För att nya och redigerade funktionsdefinitioner ska träda i kraft måste skrivbordet läsa om databasen.

- Öppna programgruppen Skrivbordsverktyg och dubbelklicka på Ladda om funktioner.
- *Eller*, exekvera kommandot:

  ```
  dtaction ReloadActions
  ```

ReloadActions är namnet på den funktion vars ikon heter ’’Ladda om funktioner’’.

Funktionsdatabasen läses också om när användaren:

- Loggar in
- Startar om arbetsytehanteraren
- Sparar en funktion i fönstret Skapa funktion genom att välja Spara på Fil-menyn

Skapa en funktionsfil (ikon) för en funktion

En *funktionsfil* är en fil som skapas för att representera funktionen visuellt i filhanteraren eller programhanteraren.



figur 10-1 *Funktionsfiler* (kallas också *funktionsikoner* eller *programikoner*) i programhanteraren

Eftersom en funktionsfils ikon representerar en funktion, kallas den ibland för *funktionsikon*. Om den underliggande funktionen startar ett program, kallas funktionsfilsikonen för *programikon*.

Om du dubbelklickar på funktionsikonen körs funktionen. Funktionsikonen kan också vara ett målområde.

▼ *Så här skapar du en funktionsfil (funktionsikon)*

♦ Skapa en exekverbar fil med samma namn som funktionsnamnet. Innehållet i filen har ingen betydelse.

Om funktionsdefinitionen är:

```
ACTION MyFavoriteApp
{
    EXEC_STRING     Mfa -file %Arg_1%
    DESCRIPTION     Kör MyFavoriteApp
    ICON            Mfapp
}
```

blir funktionsfilen en exekverbar fil med namnet `MyFavoriteApp`. I filhanteraren och programhanteraren använder filen `MyFavoriteApp` ikonbilden `Mfapp.`*`storlek`*`.`*`typ`*. Om du dubbelklickar på ikonen för `MyFavoriteApp` körs funktionens exekveringssträng, och ikonens objekthjälp får innehållet i fältet `DESCRIPTION` (`Kör MyFavoriteApp`).

Funktionsnamn

Om funktionsdefinitionen innehåller fältet `LABEL`, får funktionsfilen innehållet i detta fält som namn i filhanteraren och programhanteraren, i stället för filnamnet (*funktionsnamn*). Om funktionsdefinitionen innehåller:

```
ACTION MyFavoriteApp
{
    LABEL      Mitt favoritprogram
    …
}
```

får funktionsikonen namnet ‘‘Mitt favoritprogram.’’

Ange ikonbilden som används av en funktion

Använd fältet `ICON` för att ange vilken ikonbild som ska användas i filhanteraren och programhanteraren för funktionsikonen som skapas för funktionen.

Om du inte anger någon ikon används bildfilerna för standardfunktionsikoner `/usr/dt/appconfig/icons/`*`språk`*`/Dtactn.*`.



figur 10-2 Standardbild för funktionsikon

Standardfunktionsikonen ändras med resurserna:

`*actionIcon:` *ikonfilnamn*

där *ikonfilnamn* kan vara ett basnamn eller en absolut sökväg.

Värdet på fältet `ICON` kan vara:

- Ett basfilnamn.

Basfilnamnet är namnet på filen som innehåller ikonbilden minus suffixet för storlek (`m` och `t`) och bildtyp (`bm` och `pm`). Om filer heter `Spelikon.m.pm` och `Spelikon.t.pm`, använder du `Spelikon`.

Om du använder basfilnamnet måste ikonfilen placeras i en katalog i ikonsökvägen:

- Egna ikoner: *Hemkatalog*`/.dt/icons`
- Systemtäckande ikoner: `/etc/dt/appconfig/icons/`*språk*

- En absolut sökväg till ikonfilen, inklusive fullständigt filnamn.

  Du ska bara använda den absoluta sökvägen om ikonen inte finns i ikonsökvägen. Om ikonfilen `Spelikon.m.pm` placeras i katalogen `/doc/projekt`, som inte finns i ikonsökvägen blir värdet i fältet `/doc/projekt/Spelikon.m.pm`.

I tabell 10-1 listas de ikonstorlekar du ska skapa och motsvarande filnamn.

tabell 10-1 Ikonnamn och storlekar på funktionsikoner

Storlek i bildpunkter	**Bitmönsternamn**	**Bildpunkts-mönsternamn**
48 gånger 48	*namn*`.l.bm`	*namn*`.l.pm`
32 gånger 32	*namn*`.m.bm`	*namn*`.m.pm`
16 gånger 16	*namn*`.t.bm`	*namn*`.t.pm`

▼ Så här ändrar du en befintlig funktionsdefinition

Du kan ändra alla funktioner som finns tillgängliga i systemet, inklusive inbyggda funktioner.

Obs! – Var försiktig när du ändrar databasen för inbyggda funktioner. De inbyggda funktionerna är utformade speciellt för att fungera med skrivbordsprogrammen.

1. Sök reda på funktionsdefinitionen till funktionen du vill ändra.

   Standardadresserna för funktionsdefinitioner är:

   - Inbyggda funktioner: `/usr/dt/appconfig/types/`*språk*
   - Systemtäckande funktioner: `/etc/dt/appconfig/types/`*språk*

- Egna funktioner: *Hemkatalog*/.dt/types

Det kan finnas fler adresser i systemet. Om du vill visa en lista över adresser som används för funktioner exekverar du kommandot:

```
dtsearchpath -v
```

Systemet använder katalogerna som listas under DTDATABASESEARCHPATH.

2. Om det behövs kopierar du texten till funktionsdefinitionen till en ny eller befintlig fil i någon av dessa kataloger:
   - Systemtäckande funktioner: /etc/dt/appconfig/types/*språk*
   - Egna funktioner: *Hemkatalog*/.dt/types

   Du måste kopiera inbyggda funktioner eftersom du inte ska redigera filer i katalogen /usr/dt/appconfig/types/*språk*.
3. Redigera funktionsdefinitionen. Spara filen när du är klar.
4. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

Prioritet i funktionsdefinitioner

När användarna anropar en funktion söker systemet i databasen efter ett matchande funktionsnamn. När det finns fler än en funktion med samma namn används prioritetsregler för att bestämma vilken som ska användas.

- Om inga andra prioritetsregler gäller, baseras prioriteten på definitionens adress. Följande lista är rangordnad från högsta till lägsta prioritet:
  - Egna funktioner (*Hemkatalog*/.dt/types)
  - Systemtäckande funktioner (/etc/dt/appconfig/types/*språk*)
  - Systemtäckande fjärrfunktioner (*värddatornamn*:/etc/dt/appconfig/types/*språk*). De sökta fjärrvärddatorerna listas i programsökvägen.
  - Inbyggda funktioner (/usr/dt/appconfig/types/*språk*)
- Inom en given katalog läses filerna *.dt i alfabetisk ordning.
- Funktioner som begränsas av ARG_CLASS, ARG_TYPE, ARG_MODE eller ARG_COUNT prioriteras framför obegränsade funktioner. (Standardvärdet för dessa fyra fält är *.)

När mer än en begränsning gäller är prioritetsordningen:

- ARG_CLASS
- ARG_TYPE
- ARG_MODE
- ARG_COUNT

När det finns mer än en begränsad ARG_COUNT är prioritetsordningen:

- Särskilda heltalsvärdet *n*
- <*n*
- >*n*
- *

Ta följande del av en funktionsdefinition som exempel:

```
ACTION EditGraphics
# EditGraphics-1
{
    ARG_TYPE       XWD
    …
}

ACTION EditGraphics
# EditGraphics-2
{
    ARG_COUNT      0
    …
}

ACTION EditGraphics
# EditGraphics-3
{
    ARG_TYPE  *
    …
}
```

Om du dubbelklickar på funktionsikonen EditGraphics startar EditGraphics-2, eftersom det inte finns några argument och ARG_COUNT 0 prioriteras. När ett filargument av XWD-typ finns används EditGraphics-1, eftersom det angav XWD ARG_TYPE. EditGraphics-3 används för alla andra filargument.

Skapa en exekveringsträng för COMMAND-funktioner

De lägsta kraven för en COMMAND-funktion är två fält—ACTION och EXEC_STRING.

```
ACTION funktionsnamn
{
    EXEC_STRING exekveringssträng
}
```

Exekveringssträngen är den viktigaste delen av en COMMAND-funktion. Den använder en syntax som påminner om kommandoraden du exekverar i terminalfönstret, men omfattar ytterligare syntax för att hantera fil- och strängargument.

Allmänna egenskaper för exekveringssträngar

Exekveringssträngar kan innehålla:

- Fil och icke-filargument
- Skalsyntax
- Absoluta sökvägar eller namn på exekverbara filer

Funktionsargument

Ett argument är information som är nödvändig för att ett kommando eller program ska kunna köras. Kommandoraden nedan kan du t ex använda för att öppna en fil i Textredigeraren:

```
dtpad filnamn
```

I det här kommandot är *filnamn* ett filargument till kommandot dtpad.

Funktioner kan, på samma sätt som program och kommadon, ha argument. Det finns två typer av data som en COMMAND-funktion kan använda:

- Filer
- Strängdata

Använda skal i exekveringssträngar

Exekveringssträngen exekveras direkt, i stället för via ett skal. Du kan dock särskilt anropa ett skal i exekveringssträngen.

Exempel:

```
EXEC_STRING    \
       /bin/sh -c \
       'tar -tvf %(File)Arg_1% 2>&1 | \${PAGER:-more};\
       echo "\\n*** Välj Stäng på Fönster-menyn för att stänga ***"'
```

Namn eller absolut sökväg till exekverbara filer

Om programmet finns i en katalog som listas i variabeln PATH kan du använda det enkla exekverbara namnet. Om programmet finns någon annanstans måste du använda den absoluta sökvägen till den exekverbara filen.

Skapa funktioner som inte använder argument

Använd samma syntax för `EXEC_STRING` som du använder för att starta programmet från kommandoraden.

Exempel

- Den här exekveringssträngen är en del av funktionen som startar X-klienten `xcutsel`.

  ```
  EXEC_STRING xcutsel
  ```

- Den här exekverbara strängen startar klienten `xclock` med en digital klocka. Kommandoraden omfattar ett kommandoradsalternativ men kräver inga argument.

  ```
  EXEC_STRING xclock -digital
  ```

Skapa funktioner som accepterar släppta filer

Använd den här syntaxen för filargumentet:

`%Arg_`*n*`%`

eller

`%(File)Arg_`*n*`%`

`(File)` är valfritt eftersom argument för `Arg_`*n* antas (som standard) vara filer. (I ”Tolka filargument som en sträng” på sidan 170 finns information om hur du använder syntaxen `%(String)Arg_`*n*`%`.)

Med den här syntaxen kan användarna släppa ett datafilsobjekt på funktionsikonen för att starta funktionen med filargumentet. Den ersätter det *n*:e argumentet i kommandoraden. Filen kan vara lokal eller vara en fjärrfil.

Exempel

- Den här exekveringssträngen exekverar `wc -w` med en släppt fil som parametern `-load`.

  ```
  EXEC_STRING wc -w %Arg_1%
  ```

- I det här exemplet visas en del av en definition för en funktion som bara arbetar med katalogargument. När katalogen släpps på funktionsikonen, visas en lista över alla filer i katalogen med läs- och skrivbehörighet.

  ```
  ACTION  List_Writable_Files
  {
      ARG_TYPE     FOLDER
      EXEC_STRING /bin/sh -c 's ls -l %Arg_1% | grep rw-'
      …
  }
  ```

Skapa en funktion som frågar efter filargument

Använd den här syntaxen för filargumentet:

`%(File)"`*ledtext*`"%`

Den här syntaxen skapar en funktion som visar en ledtext för ett filnamn när användaren dubbelklickar på funktionsikonen.

Den här exekveringssträngen visar t ex en dialogruta som frågar efter filargumentet till kommandot `wc -w` :

```
EXEC_STRING wc -w %(File)"Räknar ord i filen:"%
```

Skapa funktioner som accepterar eller frågar efter en släppt fil

Använd den här syntaxen för filargumentet:

`%Arg_`*n*`"`*ledtext*`"%`

eller

`%(File)Arg_`*n*`"`*ledtext*`"%`

Syntaxen skapar en funktion som:

- Accepterar en släppt fil som filargument.
- Visar en dialogruta som frågar efter ett filnamn när användaren dubbelklickar på funktionsikonen.

Den här exekveringssträngen utför t ex `lp -oraw` på en släppt fil. Om funktionen startar när någon dubbelklickar på ikonen visas en dialogruta som frågar efter filnamnet.

```
EXEC_STRING lp -oraw %Arg_1"Fil att skriva ut:"%
```

Skapa funktioner som frågar efter icke-filargument

Använd den här syntaxen för icke-filparametern:

`%"`*ledtext*`"%`

eller

`%(String)"`*ledtext*`"%`

`(String)` är valfritt eftersom text inom citattecken normalt tolkas som strängdata. Den här syntaxen visar en dialogruta som frågar efter icke-fildata. Använd den inte för att fråga efter filnamn.

Den här exekveringssträngen kör t ex kommandot `xwd` och frågar efter ett värde som ska läggas till varje bildpunkt:

```
EXEC_STRING xwd -add %"Lägg till värde:"% -out %Arg_1"Filnamn:"%
```

Tolka filargument som en sträng

Använd den här syntaxen för argumentet:

`%(String)Arg_`*n*`%`

Den här exekveringssträngen skriver t ex ut en fil med en titel som innehåller filnamn, med hjälp av kommandot `lp -t`*titel filnamn*.

```
EXEC_STRING lp -t%(String)Arg_1% %(File)Arg_1"Fil att skriva ut:"%
```

Skalmöjligheter för funktioner

Ange skalet i exekveringssträngen:

```
/bin/sh -c 'kommando'
/bin/ksh -c 'kommando'
/bin/csh -c 'kommando'
```

Exempel

- Den här exekveringssträngen illustrerar en funktion som använder skalrörledning.

  ```
  EXEC_STRING /bin/sh -c 'ps | lp'
  ```

- Det här är en mer komplicerad exekveringssträng som kräver skalbearbetning och accepterar filargument.

  ```
  EXEC_STRING /bin/sh -c 'tbl %Arg_1"Kommandobeskrivning:"% | troff
  -man'
  ```

- Den här exekveringssträngen kräver att argumentet ska vara en komprimerad fil. Funktionen packar upp filen och skriver ut den med `lp -oraw`.

  ```
  EXEC_STRING  /bin/sh -c 'cat %Arg_1 "Fil att skriva ut:"% | \
               uncompress | lp -oraw'
  ```

- Den här exekveringssträngen startar ett skalskript.

  ```
  EXEC_STRING /usr/local/bin/StartGnuClient
  ```

Skapa COMMAND-funktioner för flera filargument

Det finns flera sätt för funktioner att hantera flera filargument:

- Funktionen kan köras upprepade gånger, en gång för varje argument. När en `EXEC_STRING` innehåller ett enstaka filargument, och flera filargument tillhandahålls genom att flera filer släpps på funktionsikonen, körs funktionen separat för varje filargument.

  Om det t ex finns flera filargument för följande funktionsdefinition:

  ```
  ACTION DisplayScreenImage
   EXEC_STRING        xwud -in %Arg_1%
   …
  }
  ```

  körs funktionen DisplayScreenImage upprepade gånger.

- Funktionen kan använda två eller flera icke utbytbara filargument, exempelvis:

  `xsetroot -cursor` *markörfil maskeringsfil*

  kräver två unika filer i en viss ordning.

- Funktionen kan utföra samma kommando och sekventiellt för varje filargument, exempelvis:

  `pr` *fil* [*fil* …]

  skriver ut en eller flera filer i ett utskriftsjobb.

Skapa en funktion för icke utbytbara argument

Använd någon av följande syntaxkonventioner:

- Om du vill att funktionen ska fråga efter filnamn, använder du den här syntaxen för varje filargument:

  `%(File)"`*ledtext*`"%`

  Använd en annan *ledtext*-sträng för varje argument.

  Den här exekveringssträngen frågar t ex efter två filer.

```
EXEC_STRING  xsetroot -cursor %(File)"Markörbitmönster:"% \
             %(File)"Maskeringsbitmönster:"%
```

- För att acceptera släppta filer använder du den här syntaxen för varje filargument:

  `%Arg_`*n*`%`

  använder olika värden på *n* för varje argument, t ex:

```
EXEC_STRING  diff %Arg_1% %Arg_2%
```

Skapa funtioner med utbytbara filargument

Använd någon av följande syntaxkonventioner:

- För att skapa en funktion som accepterar släppta filer och startar ett kommando i formatet *kommando* fil_1 fil_2 …, använder du den här syntaxen för filargumenten:

  `%Args%`

- För att skapa en funktion som accepterar flera släppta filer, eller visar en ledtext för en enstaka fil när man dubbelklickar på den, använder du den här syntaxen för filargumenten:

  `%Arg_1"`*ledtext*`"% %Args%`

  Funktionen startar kommandot i formatet: *kommando* fil_1 fil_2 ….

Exempel

- Den här exekveringssträngen skapar en funktion som exekverar:

  `lpr` *fil*$_1$ *fil*$_2$

  med flera filargument.

  ```
  EXEC_STRING pr %Args%
  ```

- Den här exekveringssträngen skapar en funktion som liknar den i föregående exempel, med undantaget att den visar en ledtext när den dubbelklickas (inga filargument).

  ```
  EXEC_STRING  pr %Arg_1"Fil(er) att skriva ut:"% %Args%
  ```

Skapa funktioner för flera släppta filer

För att acceptera flera argument för släppta filer och exekvera en kommandorad i formatet:

kommando *fil*$_1$ *fil*$_2$ *...*

använder du syntaxen:

```
%Args%
```

Exempel

- Den här exekveringssträngen exekverar skriptet Checkout för flera filer:

  ```
  EXEC_STRING  /usr/local/bin/Checkout \
               %Arg_1"Vilken fil ska kontrolleras?"% %Args%
  ```

- Den här exekveringssträngen exekverar `lp -oraw` med flera filer:

  ```
  EXEC_STRING  lp -oraw %Arg_1"Fil att skriva ut:"% %Args%
  ```

Fönsterstöd och terminalemulatorer för COMMAND-funktioner

`COMMAND`-funktioner stöder fönster på skrivbordet på flera sätt.

- Om programmet har ett eget fönster, kan funktioner skrivas utan ytterligare fönsterstöd. Det här alternativet används också när en funktion kör ett kommando som inte kräver någon direkt användarinmatning och inte har några utdata.
- Om programmet måste köras i ett terminalemulatorfönster, kan funktionen skrivas för att öppna fönstret och sedan köra programmet. Det finns flera terminalalternativ.

Ange fönsterstöd för funktionen

Använd fältet WINDOW_TYPE för att ange typen av fönsterstöd som funktionen kräver.

WINDOW_TYPE	Fönsterstöd
NO_STDIO	Inget. Använd NO_STDIO om programmet har ett eget fönster eller om kommandot inte har synliga utdata.
PERM_TERMINAL	Permanent terminalemulatorfönster. Funktionen öppnar ett terminalfönster som är öppet tills användaren stänger det. Användaren kan skriva data i fönstret. Används tillsammans med kommandon som tar emot vissa indata, skapar vissa utdata och sedan avslutas (t ex ls *katalog*).
TERMINAL	Temporärt terminalemulatorfönster. Funktionen öppnar ett terminalfönster som stängs så fort kommandot är färdigt. Används tillsammans med fullskärmskommandon (t ex vi).

Ange kommandoradsalternativ för terminalemulatorn

Använd fältet TERM_OPTS i funktionsdefinitionen för att ange kommandoradsalternativ för terminalemulatorn.

Följande funktion frågar t ex efter värddatorn för exekvering:

```
ACTION OpenTermOnSystemUserChooses
{
    WINDOW_TYPE    PERM_TERMINAL
    EXEC_HOST      %(String)"Fjärrterminal på:"%
    TERM_OPTS      -title %(String)"Fönstertitel:"%
    EXEC_STRING    $SHELL
}
```

Ange en annan standardterminalemulator

Standardterminalemulatorn som används av funktioner är dtterm. Du kan ändra detta till en annan terminalemulator. Standardterminalemulatorn används när funktionen inte särskilt anger vilken terminalemulator som ska användas.

Terminalemulatorn som används av funktionerna måste ha dessa kommandoradsalternativ:

- `-title` *fönstertitel*
- `-e` *kommando*

Två resurser bestämmer vilken standardterminalemulator som används av funktionerna:

- Resursen `localTerminal` anger vilken terminalemulator som används av lokala program.

  `*localTerminal:` *terminal*

  Exempel:

  `*localTerminal:xterm`

- Resursen `remoteTerminal` anger vilken terminalemulator som används av fjärrprogram.

  `*remoteTerminal:` *värddator*`:`*terminal* `[,`*värddator*`:`*terminal*`...]`

  Exempel:

  `*remoteTerminal: sysibm1:/usr/bin/xterm,syshp2:/usr/bin/yterm`

Begränsa funktioner till vissa argument

Genom att begränsa funktionen till en viss typ av argument finjusteras funktionen. Det kan t ex vara bra att begränsa en funktion som anropar en vy för PostScript-filer till endast filargument för PostScript. Med denna begränsning returnerar funktionen ett fel om en fil som inte är PostScript anges.

Du kan begränsa funktioner baserat på:

- Filargumentets datatyp.
- Antalet filargument – t ex inga filargument gentemot ett eller flera argument. Detta gör att funktionsikonen fungerar på olika sätt när någon släpper filer och dubbelklickar på den.
- Argumentets läs- och skrivläge.

Begränsa funktioner till en viss datatyp

Använd fältet ARG_TYPE för att ange vilken datatyp som funktionen gäller för. Använd datanamnsattributet.

Du kan skriva en lista över datatyper och avgränsa posterna med kommatecken.

Följande funktionsdefinition antar att en Gif-datatyp har skapats.

```
ACTION Open_Gif
{
    TYPE           COMMAND
    LABEL          "Visar Gif"
    WINDOW_TYPE    NO_STDIO
    ARG_TYPE       Gif
    ICON           xgif
    DESCRIPTION    Visar gif-filer
    EXEC_STRING    xgif
}
```

Begränsa funktioner baserat på antal argument

Använd fältet ARG_COUNT för att ange antal argument som funktionen kan acceptera. Giltiga värden är:

* (Standardvärde) Valfritt antal argument. Andra värden har prioritet över *.

n Alla positiva heltal, inklusive 0.

>*n* Fler än *n* argument.

<*n* Färre än *n* argument.

Använd ARG_COUNT för att ange olika metoder för funktionsikoner, beroende på om användaren dubbelklickar på ikonen eller släpper en fil på den. Mer information finns i nästa avsnitt, ”Så här anger du metod för dubbelklickning och släpp”.

▼ Så här anger du metod för dubbelklickning och släpp

Använd den här proceduren för att skapa en funktion som accepterar en släppt fil, men inte frågar efter en fil när någon dubbelklickar på funktionsikonen.

1. Skapa en funktionsdefinition för dubbeklickfunktionalitet.

   Använd fältet ARG_COUNT för att ange 0 argument. Använd en syntax för EXEC_STRING som inte accepterar släppta argument.

2. Skapa en andra funktionsdefinition för släppfunktionalitet.

   Använd fältet ARG_COUNT för att ange >0 argument. Använd en syntax för EXEC_STRING som accepterar en släppt fil.

Anta att följande två kommandorader kan användas för att starta en redigerare med namnet vedit:

- För att starta redigeraren utan filargument:

  vedit

- För att starta redigeraren med ett filargument som öppnas som skrivskyddat dokument:

  vedit -R *filnamn*

Följande två funktioner skapar släpp- och dubbelklickfunktionalitet för en funktion som heter Vedit. Den första funktionen prioriteras när databasen letar efter en matchning, eftersom ARG_COUNT 0 är mer specifik än ARG_COUNT * i definitionen av släppfunktionaliteten.

```
# Double-click functionality
ACTION Vedit
{
    TYPE           COMMAND
    ARG_COUNT      0
    WINDOW_TYPE    PERM_TERMINAL
    EXEC_STRING    vedit
}

# Drop functionality
ACTION Vedit
{
    TYPE           COMMAND
    WINDOW_TYPE    PERM_TERMINAL
    EXEC_STRING    vedit -R %Arg_1%
}
```

Begränsa funktioner baserat på argumentläge

Använd fältet ARG_MODE för att ange läs- och skrivläge för argumentet. Giltiga värden är:

*	(Standardvärde) Valfritt läge
!w	Icke skrivbar
w	Skrivbar

Skapa funktioner som kör program i fjärrsystem

När funktioner och fjärrexekvering behandlas finns det två termer som ofta används:

databasvärd	Systemet innehåller funktionsdefinitionen
värddator för exekvering	Systemet där de exekverbara filerna körs

I de flesta fall finns funktioner och deras program i samma system. Eftersom den normala värddatorn för exekvering av en funktion är databasvärden, behövs ingen särskild syntax.

När värddatorn för exekvering däremot är en annan än databasvärden, måste funktionsdefinitionen ange var exekveringssträngen ska köras.

Möjligheten att placera funktioner och program på olika system är en del av skrivbordets klient/server-arkitektur. Mer information om nätverksprogram finns i ”Administrera programtjänster” på sidan 98.

Skapa en funktion som kör ett fjärrprogram

Använd fältet EXEC_HOST i funktionsdefinitionen för att ange var programmet finns.

Giltiga värden för EXEC_HOST är:

%DatabaseHost%	Värddatorn där funktionen är definierad.
%LocalHost%	Värddatorn där funktionen anropas (*sessionsservern*).
%DisplayHost%	Värddatorn som kör X-servern (gäller inte för X-terminaler).

%SessionHost%	Värddatorn där den styrande inloggningshanteraren körs.
värddatornamn	Den namngivna värddatorn. Använd det här värdet för miljöer där funktionen alltid ska anropas på en viss värddator.
%"*ledtext*"%	Frågar användaren efter värddatornamnet varje gång funktionen anropas.

Standardvärdet är %DatabaseHost%, %LocalHost%. När fältet EXEC_HOST utelämnas försöker dock funktionen först köra kommandot på värddatorn som innehåller funktionsdefinitionen. Om detta misslyckas försöker funktionen köra kommandot på sessionsservern.

Exempel

- Detta fält anger värddatorn ddsyd:

  ```
  EXEC_HOST  ddsyd
  ```

- Detta fält frågar efter värddatornamnet:

  ```
  EXEC_HOST  %"Värddator som innehåller programmet:"%
  ```

- Detta fält anger att funktionen försöker köra programmet på värddatorn som innehåller funktionsdefinitionen. Om detta misslyckas försöker funktionen köra programmet på värddatorn ddsyd.

  ```
  EXEC_HOST  %DatabaseHost%, ddsyd
  ```

Använda variabler i funktions- och datatypsdefinitioner

Du kan ta med strängvariabler och miljövariabler i filer med funktions- och datatyper.

Använda strängvariabler i en funktion

En strängvariabeldefinition gäller från definitionens adress till slutet av filen. Det finns inga globala strängvariabler för databasen.

Om en strängvariabel och miljövariabel har samma namn, har strängvariabeln prioritet.

▼ *Så här definierar du en strängvariabel*

♦ Använd syntaxen:

`set` *variabelnamn=värde*

Variabelnamn kan innehålla alla alfanumeriska tecken och understreck (_). Varje variabeldefinition måste finnas på en separat rad.

Exempel:

```
set Remote_Application_Server=sysapp
set Remote_File_Server=sysdata
```

▼ *Så här refererar du en strängvariabel*

♦ Använd syntaxen:

`$[{]`*variabelnamn*`[}]`

Exempel:

```
EXEC-HOST  $Remote_Application_Server
CWD        /net/${Remote_File_Server}/doc/project
```

Använda miljövariabler i funktioner och datatyper

♦ Referera en miljövariabel med syntaxen:

`$[{]`*variabel*`[}].`

Variabeln utökas (ersätts med sitt värde) när databasen startas. Om en strängvariabel och miljövariabel har samma namn, prioriteras strängvariabeln.

Den här exekveringssträngen skriver t ex ut en fil med en titelsida som innehåller inloggningsnamnet.

```
EXEC-STRING lp -t$LOGNAME %(File)Arg_1%
```

Anropa funktioner från kommandoraden

På skrivbordet finns kommandot `dtaction` som används för att köra funktioner från kommandoraden. Du använder `dtaction` för att köra funktioner från:

- Skript
- Andra funktioner
- Kommandoraden i en terminalemulator

Syntax till dtaction

`dtaction [-user` *användarnamn*`] [-execHost` *värddatornamn*`]` *funktionsnamn* `[`*argument* `[`*argument*`]…]`

`-user` *användarnamn*	Ger möjlighet att köra funktionen som en annan användare. Om `dtaction` anropas av en annan användare än *användarnamn*, ombeds användaren ange lösenord.
`-execHost` *värddatornamn*	Endast för `COMMAND`-funktioner. Anger vilken värddator kommandot körs på.
argument	Argument för funktionen, normalt filargument.

Klienten `dtaction` har ytterligare kommandoradsalternativ. Mer information finns i kommandobeskrivningen till `dtaction(1)`.

Skapa en funktion som kör en annan funktion

Använd `dtaction` i funktionens `EXEC_STRING`.

Följande funktion använder t ex det inbyggda funktionsnamnet Spell (funktionen ‘‘Kontrollera stavning’’ i programhanteraren). Den nya funktionen kör Textredigaren och funktionen Spell, samt visar stavfel i ett separat terminalemulatorfönster.

```
ACTION EditAndSpell
{
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING    /bin/sh -c 'dtaction Spell \
                   %Arg_1"Fil:"%; dtpad %Arg_1%'
}
```

Skapa en funktion som körs som en annan användare

Använd följande syntax i EXEC_STRING:

EXEC_STRING dtaction -user *användarnamn funktionsnamn* [*filargument*]

Den nya användaren (*användarnamn*) måste ha läsrättigheter i systemet via någon av följande mekanismer:

- Läsbehörighet till inloggande användares .Xauthority-fil
- *Eller* xhost-behörighet

Följande två funktioner ger t ex möjlighet att bli rot och redigera en programstandardfil.

```
ACTION AppDefaults
{
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING    /usr/dt/bin/dtaction -user root \
                   EditAppDefaults %Arg_1"Fil:"%
}
ACTION EditAppDefaults
{
    WINDOW_TYPE    TERMINAL
    EXEC_STRING    /bin/sh -c 'chmod +w %Arg_1%; \
                   vi %Arg_1%; chmod -w %Arg_1%'
}
```

Skapa lokalanpassade funktioner

Sökvägen till datatyper omfattar språkberoende adresser. Skrivbordet använder värdet på LANG för att bestämma sökta adresser för datatypsdefinitioner.

Adresser för lokalanpassade funktioner

Lokalanpassade funktionsdefinitioner måste placeras i rätt språkberoende katalog i funktionens sökväg.

Standardsökvägen är:

- Egna funktioner: *Hemkatalog*/.dt/types
- Systemtäckande funktioner: /etc/dt/appconfig/types/*språk*
- Inbyggda funktioner: /usr/dt/appconfig/types/*språk*

▼ Så här lokalanpassar du en befintlig funktion

1. Skapa en fil i rätt språkberoende katalog (t ex i `/etc/dt/appconfig/types/jp_JP`).
2. Kopiera funktionsdefinitionen till den språkberoende konfigurationsfilen.

   Anta att du kopierar funktionsdefinitionen från

   progrot`/dt/appconfig/types/C/`*fil*`.dt`

   till

   progrot`/dt/appconfig/types/jp_JP/`*nyfil*`.dt`
3. Lägg till fältet `LABEL` eller ändra det befintliga fältet `LABEL`.

   `LABEL` *sträng*

   Programhanteraren och filhanteraren använder namnsträngen för att identifiera funktionens ikon.
4. Lokalanpassa något av följande fält i funktionsdefinitionen:
   - Lokalanpassade ikoner: `ICON`
   - Lokalanpassad objekthjälp: `DESCRIPTION`
   - Lokalanpassade: all text inom citattecken i `EXEC_STRING`

Skapa funktioner för ToolTalk-program

Obs! – Följande information gäller bara för program som stöder ToolTalk-meddelanden.

Använd funktionstypen `TT_MSG` för att skapa en funktion som skickar ett ToolTalk-meddelande.

```
ACTION funktionsnamn
{
    TYPE    TT_MSG
    …
}
```

Adressera och disponera fält

- Fältet `addressing` i ToolTalk anges alltid till `TT_PROCEDURE`.

- Fältet `disposition` i ToolTalk använder som standard specifikationen i det statiska meddelandemönstret.

Meddelanden som inte stöds

Följande stöds inte av `TT_MSG`-typsfunktionerna:

- Objektorienterade ToolTalk-meddelanden
- Sammanhangsargument i meddelanden

Nyckelord för TT_MSG-funktioner

Nyckelord	Användningsområde
TT_CLASS	Definierar värdet i meddelandefältet `class` message i ToolTalk
TT_SCOPE	Definierar värdet i meddelandefältet `scope` i ToolTalk
TT_OPERATION	Definierar värdet i meddelandefältet `operation` i ToolTalk.
TT_FILE	Definierar värdet i meddelandefältet `file` i ToolTalk
TT_ARGn_MODE	Definierar värdet på attributet `mode` för det *n*:e meddelandeargumentet i ToolTalk
TT_ARGn_VTYPE	Definierar värdet på attributet `vtype` för det *n*:e meddelandeargumentet i ToolTalk
TT_ARGn_VALUE	Definierar värdet på det *n*:e meddelandeargumentet

10

Skapa datatyper manuellt 11

Du kan skapa datatypsdefinitioner på två sätt:

- Använda verktyget Skapa funktion. Skapa funktion beskrivs i Kapitel 9, ”Skapa funktioner och datatyper med Skapa funktion”.
- Skapa datatypsdefinitioner manuellt.

När du skapar en datatyp manuellt måste du redigera en databasfil.

I det här kapitlet beskrivs hur du manuellt skapar datatypsdefinitioner.

Varför du måste skapa datatyper manuellt	*188*
Komponenter i datatypsdefinitionen: Villkor och attribut	*188*
Skapa datatyper manuellt: Allmänna steg	*189*
Exempel på hur du skapar en egen funktions- och datatyp	*190*
Definiera datavillkor för en datatyp	*195*

Se även

- En introduktion till datatyper finns i Kapitel 8, ”Introduktion till funktioner och datatyper”.
- Referensinformation om datatypsdefinitioner finns i kommandobeskrivningen till `dtddsfile(4)`.

11

Varför du måste skapa datatyper manuellt

Med manuellt skapade datatyper kan du använda alla möjligheter som finns inbyggda i datatypsdefinitionernas syntax.

Du måste skapa en datatyp manuellt om du vill kunna använda följande funktionalitet:

- Sökvägsbaserade datatyper
- Möjlighet att ange funktioner som associeras med en annan datatyp än Open och Print
- Flera namn-, mönster- och innehållsvillkor för samma datatyp, t ex en datatyp baserad på filer med namnet `*.abc` eller `*.def`
- Länkbaserade datatyper

Komponenter i datatypsdefinitionen: Villkor och attribut

En datatypsdefinition består av två separata databasdefinitioner:

- Definitionen `DATA_ATTRIBUTES`.

  `DATA_ATTRIBUTES` beskriver datatypens namn samt utseende och metod för filer av den här typen.

- Definitionen `DATA_CRITERIA`.

  `DATA_CRITERIA` beskriver typens villkor. Varje villkorsdefinition anger `DATA_ATTRIBUTES`-definitionen som villkoret gäller för.

Det måste finnas minst en `DATA_CRITERIA`-definition för varje `DATA_ATTRIBUTES`: en `DATA_ATTRIBUTES`-definition kan ha flera `DATA_CRITERIA` associerade.

Du kan t ex skapa en attributdefinition för PostScript-filer som beskriver hur PostScript-filerna ser ut och fungerar i filhanteraren. Därefter kan du skapa två separata villkor för PostScript-datatypen— ett baserat på filnamn och det andra på filinnehåll.

Mer information finns i ”Definiera datavillkor för en datatyp” på sidan 195.

Skapa datatyper manuellt: Allmänna steg

I det här avsnittet beskrivs hur du skapar en datatypskonfigurationsfil.

Konfigurationsfiler för datatyper

Kraven för konfigurationsfiler som innehåller datatypsdefinitioner är:

- Filerna måste använda namngivningskonventionen *namn*`.dt`
- Filerna måste finnas i databassökvägen. Standardsökvägen är:

Egna datatyper	*Hemkatalog*`/.dt/types`
Systemtäckande datatyper	`/etc/dt/appconfig/types/`*språk*
Inbyggda datatyper	`/usr/dt/appconfig/types/`*språk*. Den här katalogen ska du inte använda.

Information om hur du ändrar databassökvägen finns i ”Ange värde för en sökväg” på sidan 115.

▼ Så här skapar du en datatypsdefinition

1. Öppna en befintlig databasfil eller skapa en ny.

   Mer information finns i föregående avsnitt ”Konfigurationsfiler för datatyper”.

2. Definiera dataattribut för datatypen med syntaxen:

```
DATA_ATTRIBUTES datatypsnamn
{
  ICON            bildnamn
  DESCRIPTION     sträng
  attributfält
  attributfält
  …
}
```

   där:

datatypsnamn	Ett unikt namn för den här datatypen.
bildnamn	Filnamn på eller sökväg för en ikonfil. Använd filens basnamn. För t ex ikonfilerna `myimage.m.pm` och `myimage.t.pm` använder du `myimage`.

attributfält	Fält som definierar utseende och metod för datatypen.
sträng	Teckensträng. Innehållet utgör objekthjälpen för datatypen.

Mer information finns i ”Exempel på hur du skapar en egen funktions- och datatyp” på sidan 190.

3. Definiera datavillkoret för datatypen med syntaxen:

```
DATA_CRITERIA villkorsnamn
{
  DATA_ATTRIBUTES_NAME   datatypsnamn
  villkorsfält
  villkorsfält
  …
}
```

där:

villkorsnamn	Ett unikt namn på villkorsdefinitionen
datatypsnamn	Namn som används i definitionen `DATA_ATTRIBUTES`
villkorsfält	Fält som används för att definiera villkor för tilldelning av en fil till den här datatypen

Mer information finns i ”Definiera datavillkor för en datatyp” på sidan 195.

4. Spara databasfilen.
5. Skapa ikonerna för den här datatypen.

   Mer information finns i ”Ange ikonbild för datatypen” på sidan 192.

6. Om det behövs skapar du funktionerna som listas i fältet `ACTIONS` för attributdefinitionen.
7. Dubbelklicka på Ladda funktioner i programgruppen Skrivbordsverktyg för att ladda om databasen.

Exempel på hur du skapar en egen funktions- och datatyp

Anta att systemet innehåller ett program med namnet `xgif` som visar GIF-bilder. Normalt kör du programmet genom att exekvera:

`xgif` *filnamn*

Du vill kunna visa GIF-bilder på flera sätt:

- Genom att dubbelklicka på en GIF-datafil
- Genom att markera datafilen och välja programmet från Markerade-menyn

1. Öppna den nya filen *Hemkatalog*`/.dt/types/GifViewer.dt` för redigering.

2. Skriv datatypsdefinitionerna:

```
DATA_ATTRIBUTES Gif
{
 DESCRIPTION           Gif-bildfil.
 ICON                  GifIcon
 ACTIONS               View
}
DATA_CRITERIA Gif_Criteria
{
 DATA_ATTRIBUTES_NAME  Gif
 NAME_PATTERN          *.gif
}
```

3. Skriv funktionsdefinitionen för funktionen GifViewer:

```
ACTION GifViewer
{
 EXEC_STRING       xgif %(File)Arg_1"Gif-fil som ska visas:"
 WINDOW_TYPE       NO_STDIO
 DESCRIPTION       Dubbelklicka eller släpp en fil \
                   för att starta Gif-granskaren.
}
```

   Eftersom definitionen inte omfattar något `ICON`-fält använder funktionen systemets standardikon.

4. Skriv följande tilldelningsfunktion för att ansluta funktionen GifViewer till funktionen View som listas i datatypsdefinitionen. Använd fältet `ARG_TYPE` för att begränsa den här visningsfunktionen till Gif-filer.

```
ACTION View
{
 ARG_TYPE       Gif
 TYPE           MAP
 MAP_ACTION     GifViewer
}
```

5. Spara filen.

6. Dubbelklicka på Ladda funktioner i programgruppen Skrivbordsverktyg för att läsa om databasen.

Definiera dataattribut för en datatyp

Definitionen DATA_ATTRIBUTES definierar datatypens utseende och metod. Den anger namn på datatypen och tillhandahåller möjligheten att ange:

- Ikon i filhanteraren (fältet ICON)
- Metoden dubbelklick och innehåll i Markerade-menyn (fältet ACTIONS)
- Datatypens objekthjälp (fältet DESCRIPTION)

Ange ikonbild för datatypen

Använd fältet ICON för att ange vilken ikon som ska användas i filhanteraren. Om du inte anger någon ikonbild visas bara namnet i filhanteraren.

Värdet på fältet ICON kan vara:

- Ett basfilnamn.

  Basfilnamnet är namnet på filen som innehåller ikonbilden, minus suffix för storlek (m och t) och bildtyp (bm och pm). Om filer har namnen GameIcon.m.pm och GameIcon.t.pm, använder du GameIcon.

  Om du använder basfilnamnet måste ikonfilerna placeras i en katalog i ikonsökvägen:
  - Egna ikoner: *Hemkatalog*/.dt/icons
  - Systemtäckande ikoner: /etc/dt/appconfig/icons/*språk*

- En absolut sökväg till ikonfilen, inklusive fullständigt filnamn.

  Du ska bara använda absolut sökväg om ikonfilen inte finns i ikonsökvägen. Om ikonfilen GameIcon.m.pm finns i katalogen /doc/projects, som inte finns i ikonsökvägen, ska värdet i fältet ICON vara /doc/projects/GameIcon.m.pm.

I tabell 11-1 listas de ikonstorlekar du ska skapa och motsvarande filnamn.

tabell 11-1 Ikonnamn och storlekar för datatypsikoner

Storlek i bildpunkter	Bitmönsternamn	Bildpunkts-mönsternamn
32 gånger 32	*namn*.m.bm	*namn*.m.pm
16 gånger 16	*namn*.t.bm	*namn*.t.pm

Associera datatyper med funktioner

Du kan associera datatyper med funktioner på två sätt:

- I fältet ACTIONS i definitionen DATA_ATTRIBUTES listas de funktioner som visas i filhanterarens Markerade-meny. Den första funktionen i listan är standardfunktionen (dubbelklick).
- Funktionstyper begränsas till angivna datatyper med funktionsdefinitionens ARG_TYPE-fält.

Följande datatypsdefinition skapar en datatyp för vissa ”readme”-filer som skapas av systemadministratören. Filerna använder namngivningskonventionen *.rm.

```
DATA_ATTRIBUTES SysReadmeFile
{
    ICON                    SysReadMe
    ACTIONS                 Open,Respond
}
DATA_CRITERIA SysReadmeFileCriteria
{
    NAME_PATTERN            *.rm
    DATA_ATTRIBUTES_NAME    SysReadmeFile
}
```

Nedan definieras en speciell Respond-funktion för filen. Den öppnar en skrivbar kopia av filen i textredigeraren. När filen sparas och textredigeraren avslutas, postas filen till systemadministratören (adressen 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
}
```

Dölja filer baserat på datatyp

Om en fil är en osynlig datatyp visas den inte i filhanteraren.

Använd fältet PROPERTIES i definitionen DATA_ATTRIBUTES för att ange att objekt av den här typen ska döljas:

```
PROPERTIES      invisible
```

Ange metoder när en fil manipuleras

Följande DATA_ATTRIBUTES-fält används huvudsakligen av programmerare. De anger hur filer ska fungera när användaren utför olika skrivbordsaktiviteter.

Mer information finns i *Common Desktop Environment Programmer's Guide*, som ingår i dokumentationen till utvecklingsmiljön.

Fält	Beskrivning
MOVE_TO_ACTION	För behållare såsom kataloger. Anger en funktion som ska köras när en fil flyttas till en behållare av den här datatypen.
COPY_TO_ACTION	För behållare såsom kataloger. Anger en funktion som ska köras när en fil kopieras till en behållare av den här datatypen.
LINK_TO_ACTION	Anger en funktion som ska köras när en fil länkas till en fil av den här datatypen.
IS_TEXT	Anger att filer av den här datatypen innehåller text som kan visas i en textruta.
MEDIA	Anger motsvarande ToolTalk-mediatyp.
MIME_TYPE	Anger motsvarande MIME-typ.
X400_TYPE	Anger motsvarande X400-typ.

Definiera datavillkor för en datatyp

Definitionen DATA_CRITERIA definierar vilka villkor som används för att tilldela en objekttyp till en fil eller katalog.

Du använder följande villkor för objekttyper:

Villkor	Beskrivning
Filnamn	Filnamnet måste matcha ett visst mönster. Använd fältet NAME_PATTERN.
Sökväg	Sökvägen måste matcha ett visst mönster. Använd fältet PATH_PATTERN.
Filinnehåll	En angiven del av filens innehåll måste matcha angivna data. Använd fältet CONTENT.
Filläge	Filen måste ha angiven behörighet (läs, skriv, exekvera eller katalog). Använd fältet MODE.
Symboliska länkar	Typen baseras på filen som objektet är länkat till.

Du kan använda flera villkor för en datatyp. Du kan dock inte använda villkoren NAME_PATTERN och PATH_PATTERN i samma datatyp.

Namnbaserade datatyper

Använd fältet NAME_PATTERN för att ange namngivningskrav. Fältvärdet kan innehålla följande jokertecken:

?	Motsvarar valfritt, enstaka tecken
*	Motsvarar valfri teckensekvens (inklusive en tom sträng)
[*tt*…]	Motsvarar det tecken (*t*) som omsluts av hakparentes
[*t*–*t*]	Motsvarar de tecken som finns i intervallet mellan *t* ooh *t*

Exempel

- Följande datatypsdefinition skapar en datatyp baserat på filnamnet. Filnamnet måste börja med QS och sluta med .doc.

```
DATA_ATTRIBUTES QS_Doc
{
 DESCRIPTION    Den här filen innehåller ett dokument för QS-
projektet.
```

```
  ICON           Word_Doc
  ACTIONS        Open
}

DATA_CRITERIA QS_Doc_Criteria
{
  NAME_PATTERN   QS*.doc
  DATA_ATTRIBUTES_NAME QS_Doc
}
```

- Följande definition skapar en datatyp för kataloger med namnet Demo_*n* där *n* är i intervallet mellan 0 och 9.

```
DATA_ATTRIBUTES Demo_directory
{
  DESCRIPTION    Detta är en katalog. Dubbelklicka för att öppna
den.
  ICON           Demo
  ACTIONS        OpenInPlace,OpenNewView
}

DATA_CRITERIA Demo_directory_criteria
{
  NAME_PATTERN           Demo_[0-9]
  MODE                   d
  DATA_ATTRIBUTES_NAME   Demo_directory
}
```

Sökvägsbaserade datatyper

Använd fältet PATH_PATTERN för att ange sökväg. Du kan använda samma jokertecken som med NAME_PATTERN.

Följande datatyp använder ett villkor baserat på sökvägen.

```
DATA_ATTRIBUTES Project_Graphics
{
    DESCRIPTION    Grafikfiler för QS-projektet. Dubbelklicka \
                   på ikonen för att visa grafiken.
    ICON           QSgraphics
}
DATA_CRITERIA Project_Graphics_Criteria
{
    DATA_ATTRIBUTES_NAME   Project_Graphics
    PATH_PATTERN           */projects/QS/graphics/*
}
```

Datatyper baserade på namn och sökväg

Om du vill skapa en datatyp baserat på både filnamn och sökväg, tar du med namnet i värdet PATH_PATTERN. Du kan inte använda både NAME_PATTERN och PATH_PATTERN i samma villkorsdefinition.

Exempel

- Datatypen QS_Source_Files som definieras nedan gäller för alla filer med namnet app*n*.c, där *n* = intervallet mellan 1 och 9, som finns i underkataloger till */projects/QS.

```
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
}
```

- Följande datatyp gäller för alla filer i katalogen /doc/project1 med namnet ch*nn*.*xxx*, där *n* är i intervallet mellan 0 och 9 och *xxx* är ett valfritt filnamnssuffix på tre tecken.

```
DATA_ATTRIBUTES ChapterFiles
{
 DESCRIPTION        Kapitelfil för projektdokumentet.
 ICON               chapter
 ACTIONS            Edit,Print
}

DATA_CRITERIA Chapter_Criteria
{
 PATH_PATTERN           /doc/project1/ch[0-9][0-9].???
 DATA_ATTRIBUTES_NAME   ChapterFiles
}
```

Använda fillägen som typvillkor

Använd fältet `MODE` för att ange vilka behörigheter som krävs.

Lägesvillkor används normalt i kombination med namnbaserade, sökvägsbaserade eller innehållsbaserade datatyper. Med dessa kan du begränsa en datatyp till en fil eller katalog, eller ange om läs-, skriv- eller exekveringsbehörighet krävs.

Fältet `MODE` kan innehålla följande logiska operatorer och tecken:

Operator	**Beskrivning**
`!`	Logiska operatorn ICKE
`&`	Logiska operatorn OCH
`|`	Logiskt ELLER
Tecken	**Beskrivning**
`f`	Datatypen gäller bara för filer
`d`	Datatypen gäller bara för kataloger
`r`	Filen kan läsas av alla användare
`w`	Filen kan skrivas av alla användare
`x`	Filen kan exekveras av alla användare
`l`	Filen är en länk

Standardvärdet för ett visst läge, är att läget inte spelar någon roll.

Exempel

- Följande lägesfält begränsar datatypen enligt beskrivningen:

`f&!w`	Skrivskyddade filer
`!w`	Skrivskyddade filer och kataloger
`f&x`	Exekverbara filer
`f&r&x`	Filer som är både skriv- och exekverbara
`x|!w`	Filer som är exekverbara eller skrivskyddade

- Följande datatypsdefinition skapar en datatyp för skrivskyddade, icke exekverbara filer vars namn följer namngivningskonventionen `*.doc`. Det antas att funktionen View har definierats för datatypen.

```
DATA_ATTRIBUTES ReadOnlyDocument
{
 ICON           read_only
 DESCRIPTION    Dokumentet är inte skrivbart. Om du \
                dubbelklickar körs redigeraren med en \
                skrivskyddad kopia av filen.
 ACTIONS        View
}

DATA_CRITERIA ReadOnlyDocument_Criteria
{
 NAME_PATTERN           *.doc
 MODE                   !d&!x&!w
 DATA_ATTRIBUTES_NAME   ReadOnlyDocument
}
```

Innehållsbaserad datatyp

Använd fältet `CONTENT` för att ange datatyp baserat på filens innehåll. Innehållsbaserade datatyper kan användas i kombination med namn- och sökvägsbaserade datatyper.

Typen kan baseras på stränginnehåll eller numeriskt innehåll i filerna. Filens första byte har nummer 0.

- För stränginnehåll använder du syntaxen:

  `CONTENT` *startbyte* `string` *sträng*

- För numeriskt innehåll använder du syntaxen:

  `CONTENT` *startbyte* `byte` *nummer*
  `CONTENT` *startbyte* `short` *nummer*
  `CONTENT` *startbyte* `long` *nummer*

- För kataloginnehåll använder du syntaxen:

  `CONTENT 0 filename "`*filnamn*`"`

  Använd den vanliga C-beteckningen för oktala (inledande `0`) och hexadecimala (inledande `0X`) tal.

Obs! – Om du använder en innehållsbaserad datatyp kan systemets prestanda sjunka. Använd därför namn- eller sökvägsbaserade datatyper när det är möjligt.

Följande datatyp, Writable_Wingz, gäller för alla filer med skrivbehörighet som innehåller strängen WNGZ i början av filen.

```
DATA_ATTRIBUTES Writable_Wingz
{
  …
}

DATA_CRITERIA Writable_Wingz_Criteria
{
  CONTENT               0 string WNGZ
  MODE                  w&!d
  DATA_ATTRIBUTES_NAME  Writable_Wingz
}
```

▼ Så här skapar du en datatyp med flera oberoende villkor

Du kan skapa en datatyp med flera oberoende villkor, vilket innebär att filen tilldelas till datatypen om den uppfyller något av (eller båda) villkoren.

1. Skapa definitionen DATA_ATTRIBUTES för datatypen.
2. Skapa en DATA_CRITERIA-definition för varje villkor.

   Använd fältet DATA_ATTRIBUTES_NAME för att ansluta alla villkor till samma DATA_ATTRIBUTES-definition.

Följande definitioner skapar datatypen Mif. Typen baseras på namn eller innehåll.

```
DATA_ATTRIBUTES Mif
{
    ICON                  Frame
    ACTION_LIST           Open,Print
}

DATA_CRITERIA Mif_Name_Criteria
{
    DATA_ATTRIBUTES_NAME      Mif
    NAME_PATTERN              *.mif
```

```
}

DATA_CRITERIA Mif_Content_Criteria
{
    DATA_ATTRIBUTES_NAME      Mif
    CONTENT                   1 string MIFFile
}
```

Skapa lokalanpassade datatyper

Sökvägen för datatyper omfattar språkberoende adresser. Skrivbordet använder värdet på LANG för att bestämma på vilka adresser datatypsdefinitioner söks.

Adresser för lokalanpassade datatyper

Lokalanpassade datatypsdefinitioner måste placeras i rätt språkberoende katalog i funktionssökvägen.

Standardsökvägen är:

- Egna funktioner: *Hemkatalog*`/.dt/types`
- Systemtäckande funktioner: `/etc/dt/appconfig/types/`*språk*
- Inbyggda funktioner: `/usr/dt/appconfig/types/`*språk*

▼ Så här lokalanpassar du en datatyp

1. Skapa en fil i rätt språkberoende katalog (t ex i `/etc/dt/appconfig/types/svenska`).
2. Kopiera datatypsdefinitionen till den språkberoende konfigurationsfilen.
3. Lokalanpassa ett eller flera fält i datatypsdefinitionen.

11

Skapa skrivbordsikoner 12

Skrivbordsikoner kan kopplas till:

- Funktionsfiler och datatyper i filhanteraren och programhanteraren
- Ikoner på frontpanelen
- Minimerade programfönster
- Grafik som används av program, t ex paletter och verktygsfält
- Arbetsytans bakgrund

Ikonbildsfiler	*203*
Associera ikoner	*206*
Rekommendation när du skapar ikoner	*209*

Obs! – Dokumentationen till utvecklingsmiljön innehåller mer information om skrivbordsikoner. Se Kapitel 4, ”Visual Design” i *Style Guide and Certification Checklist*.

Ikonbildsfiler

För att skrivbordet ska kunna använda en ikonbild, måste ikonbildsfilen:

- Ha rätt format
- Använda rätt namngivningskonvention
- Använda skrivbordets storlekskonventioner

- Finnas i en katalog i ikonsökvägen
- Anropas av skrivbordstolken med rätt syntax. Om du t ex skapar en ny ikon för frontpanelen, använder du fältet `ICON` i frontpanelsdefinitionen för att ange vilken ikonbild som ska användas

Ikonfilformat

För en färgskärm använder du ikonfiler med formatet X pixmap (XPM), som normalt har suffixet `.pm`. Annars använder du filer med formatet X bitmap (XBM), vilka normalt har suffixet `.bm`. Om transparens används i bildpunktsmönsterfilen, skapas en maskeringsfil (`_m.bm`) när filen `.bm` skapas. Mer information om hur skrivbordet hittar dessa filer finns i ”Ikonsökväg” på sidan 121.

Ikonfilnamn

Alla ikoner och bakgrundsbilder lagras som separata filer. Normalt anges en ikon med basdelen i filnamnet. en ikon kan t ex refereras med namnet `mail` när filen lagras som:

`/usr/dt/appconfig/icons/`*`språk`*`/mail.l.pm`

Namngivningskonventionen med tillagda suffix grupperar ikoner efter storlek och typ. Ikonnamn för skrivbordskomponenter finns i följande allmänna format:

basnamn`.`*storlek*`.`*format*

Alternativt

basnamn`.`*format*

där:

basnamn	Bildens basnamn används för att referera till bilden
storlek	En bokstav som anger storlek: `l` (stor) `m` (normal) `s` (liten) `t` (mycket liten)

format	Filformat: `pm` (bildpunktsmönster) `bm` (bitmönster)

Storlekskonventioner

I tabell 12-1 visas rekommenderade bildpunktsdimensioner för skrivbordsikoner.

tabell 12-1 Ikonstorlekar och filnamn

Ikonstorlek	Bitmönsternamn	Bildpunktsmönstern amn
16 gånger 16 (mycket liten)	*namn*`.t.bm`	*namn*`.t.pm`
24 gånger 24 (liten)	*namn*`.s.bm`	*namn*`.s.pm`
32 gånger 32 (normal)	*namn*`.m.bm`	*namn*`.m.pm`
48 gånger (stor)	*namn*`.l.bm`	*namn*`.l.pm`

I tabell 12-2 visas vilka ikonstorlekar som används av skrivbordets komponenter. I vissa fall beror storleken som används på skärmens upplösning.

tabell 12-2 Skrivbordets komponenter och deras ikonstorlekar

Del av skrivbordet	Hög uppslösning	Normal uppslösning	Låg upplösning
Filhanteraren och programhanteraren (Namn och Ikon)	normal	normal	normal
Filhanteraren och programhanteraren (Små ikoner)	mycket liten	mycket liten	mycket liten
Ikoner på huvudfrontpanelen	stor	stor	medel
Frontpanelens panelundermenyer	normal	normal	mycket liten
Frontpanelens växlingsikoner	liten	liten	mycket liten
Minimerade fönster	stor	stor	normal

Om du t ex anger en ikon med namnet `mail` för en datatyp, förutsatt att du har en färgskärm och har angett filhanterarens inställningar till små ikoner, används ikonbilden `mail.t.pm`.

Ikonsökvägen

Skrivbordet söker efter en ikonfil, eller bild, genom att leta efter filen i en lista över kataloger. Den här listan som kallas *ikonsökvägen* bestäms utifrån värdet på flera miljövariabler. Vilka variabler som används och hur de läggs ihop för att skapa ikonsökvägen beskrivs i ”Ikonsökväg” på sidan 121.

Standardsökvägen är:

- Inbyggda ikoner: `/usr/dt/appconfig/icons/`*språk*
- Systemtäckande ikoner: `/etc/dt/appconfig/icons/`*språk*
- Egna ikoner: *Hemkatalog*`/.dt/icons`

Komma åt ikoner via nätverket

Skrivbordet kan även få använda ikoner i fjärrsystem. Mer information om hur du skapar en ikonserver finns i ”Konfigurera databas-, ikon- och hjälptjänster” på sidan 101.

Associera ikoner

För att objekt snabbare ska kunna kännas igen associerar du ikoner med:

- Funktioner och datatyper
- Ikoner på frontpanelen och panelundermenyer
- Minimerade programfönster

Ange ikonfiler

För ikoner som används till funktioner, datatyper och i frontpanelen eller panelundermenyer, anger du bara ikonens basnamn (inga suffix). Rätt suffix läggs till automatiskt baserat på skärmupplösning, färgstöd och alternativ för filhanterarfönstret (t ex Små ikoner).

För att åsidosätta sökvägen anger du fullständig sökväg och namn på ikonen.

▼ Så här associerar du en ikon med en funktion eller datatyp

1. Ange ikonen med fältet `ICON`.

   Om du följer lämplig namngivningskonvention för ikonfiler, anger du bara ikonens basnamn. Rätt ikon visas baserat på skärmens upplösning och färgstöd.

2. Skapa följande ikonstorlekar:
   - Funktioner: stor, normal och mycket liten
   - Datatyper: normal och mycket liten

Exempel på en funktionsdefinition

Följande exempel är en funktionsdefinition för att starta ritverktyget Island Paint™. Ikonerna `Ipaint.l` och `Ipaint.s` associeras med funktionen:

```
ACTION             IslandPaintOpenDoc
{
    WINDOW_TYPE    NO-STDIO
    ICON           Ipaint
    EXEC_STRING     /usr/bin/IslandPaint %Arg_1"Fil att öppna:"%
}
```

Om du använder färgikoner lägger skrivbordet först till `.pm` när det söker efter ikonfilen. Annars (eller om ingen matchning hittades med `.pm`) lägger skrivbordet till `.bm`.

Exempel på datatypsdefinition

Följande datatypsdefinition associerar ikonerna `comprsd.l` och `comprsd.s` med komprimerade filer:

```
DATA_ATTRIBUTES     COMPRESSED
{
    ICON           comprsd
    ACTIONS        Uncompress
    DESCRIPTION    En COMPRESSED-fil har komprimerats med \
                   kommandot 'compress' för att ta upp mindre plats.
}
```

▼ Så här visar du en ikon i frontpanelen

1. Ange bildens namn med fältet `ICON`.

   Om ikonen övervakar en fil (`MONITOR_TYPE` anges till mail eller file) använder du fältet `ALTERNATE_ICON` för att ange vilka ikoner som ska användas när förändringen upptäcks.

   Du kan också ange animeringar för ikoner till knappar och målområden.

2. Skapa följande ikonstorlekar:
   - Huvudpanelen och panelundermenyer: stor, normal och mycket liten
   - Arbetsyteomkopplaren: liten

Exempel

Följande ikon ändrar utseende när en fil med namnet `report` placeras i katalogen `/doc/ftp/pub/`. När filen inte finns där visas ikonen `NoReport.pm`. När filen finns där visas `Report.pm`.

```
CONTROL MonitorReport
{
    CONTAINER_NAME     behållare
    TYPE               ICON
    MONITOR_TYPE       file
    FILE_NAME          /doc/ftp/pub/report
    ICON               NoReport
    ALTERNATE_ICON     Report
}
```

▼ Så här associerar du en ikon med ett programfönster

1. Ange resursen `iconImage` för arbetsytehanteraren så här:

   `Dtwm*`*klientnamn*`*iconImage:` *ikonfilnamn*

   För att bestämma rätt värde på *klientnamn* öppnar du programhanteraren och dubbelklickar på Fönsterinställningar i programgruppen Skrivbordsverktyg. När du markerar ett fönster listas dess egenskaper. Egenskapen WM_CLASS visar fönstrets klassnamn inom citattecken.

   Mer information om hur du anger resurser finns i ”Ange programresurser” på sidan 257.

2. Välj Starta om arbetsytehanteraren från Arbetsyta-menyn.

För att kontrollera att ikonen har känts igen av arbetsytehanteraren minimerar du fönstret vars ikon du försöker ändra.

Obs! – I vissa program kan inte standardfönsterikonen åsidosättas.

▼ Så här använder du filhanteraren som ett ikonfönster

1. Kopiera filen `/usr/dt/examples/`*språk*`/IconBrowse.dt` till katalogen *Hemkatalog*`/.dt/types/Iconbrowse.dt`.
2. Öppna programhanteraren och dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

När du växlar till en katalog som innehåller ikoner (`.bm`- och `.pm`-filer) visas varje ikon intill sitt namn. Om du t ex växlar till katalogen `/usr/dt/appconfig/icons/`*språk* visas många av skrivbordsikonerna.

Obs! – Om du aktiverar ikonfönstret i ett system med lågt minne kan det hända att filhanteraren visar kataloger mycket långsammare.

Bilder större än 256 x 256 visas inte i standardkonfigurationen.

Så här avaktiverar du ikonfönstret:

1. Ta bort den egna kopian av filen `IconBrowse.dt`.
2. Öppna programhanteraren och dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

Rekommendation när du skapar ikoner

Använd ett tema för närliggande ikoner. Om du t ex skapar ikoner för ett program bör programikonen och ikonen för datafilerna likna varandra.

Kontrollera att den tvåfärgsversion av färgikonen du skapar är giltig. Om ikonen visas på en monokrom skärm eller gråskaleskärm (eller om det inte finns tillräckligt med färger tillgängliga) visas ikonen automatiskt i tvåfärgsformatet.

Om du vill spara på färgerna som används, kan du begränsa de ikonfärger som används till de som finns på skrivbordet. (Ikoner som skapas med ikonredigeraren använder bara dessa färger.)

Information om vika storlekar som används av skrivbordets komponenter finns i tabell 12-1, ”Ikonstorlekar och filnamn,” på sidan 205.

Hur färger används

Skrivbordet använder en palett med 22 färger, bl a:

- Åtta statiska gråskalor
- Åtta statiska färger: röd, blå, grön, cyan, magenta, gul, svart och vit
- Sex dynamiska färger: förgrund, bakgrund, övre skugga, undre skugga, markering och genomskinlig

Med den här paletten skapar du tilltalande och lättlästa ikoner som inte inkräktar på de färgresurser som behövs av andra program. För de flesta ikoner på skrivbordet används gråskalor som betonas med färg.

Den genomskinliga färgen är användbar för att skapa ikoner som ska verka icke-rektangulära, eftersom färgen bakom ikonen skiner igenom.

Särskild anpassning av frontpanelen

13

Användarna kan anpassa frontpanelen med dess meny och Installera Ikon på panelundermenyerna.

I det här kapitlet beskrivs hur du anpassar frontpanelen genom att skapa och redigera konfigurationsfiler.

Konfigurationsfiler för frontpanelen	*212*
Administrera anpassning av användargränssnitt	*214*
Hur frontpanelsdefinitionen är organiserad	*215*
Ändra huvudpanelen	*219*
Skapa och ändra panelundermenyer	*224*
Definiera kontroller på frontpanelen	*228*
Anpassa arbetsyteomkopplaren	*236*
Allmän konfiguration av frontpanelen	*237*

Se även

- Referensinformation för kontroller på frontpanelen och konfiguration finns i kommandobeskrivningen till `dtfpfile(4X)`.
- Referensinformation om arbetsytehanteraren finns i kommandobeskrivningen till `dtwm(1)` och `dtwmrc(4)`.

13

Konfigurationsfiler för frontpanelen

Frontpanelen definieras i en databas med konfigurationsfiler.

Med konfigurationsfilerna anpassar du frontpanelen. Vissa ändringar kan bara göras genom att en konfigurationsfil redigeras, bl a:

- Lägga till en ny kontrollposition på huvudpanelen
- Lägga till speciella typer av kontroller, t ex klientfönster
- Ändra vissa standardmetoder – t ex om kontrollerna på frontpanelen ska reagera på enkel- eller dubbelklickningar

För att konfigureringen ska vara så flexibel som möjligt kan dessa filer vara egna, systemtäckande eller finnas i andra system.

Frontpanelen skapas och hanteras av arbetsytehanteraren.

Standardfrontpanelens konfigurationsfil

Standardfrontpanelen definieras i konfigurationsfilen `/usr/dt/appconfig/types/`*`språk`*`/dtwm.fp`.

Den här filen får inte ändras.

Sökvägen till frontpanelens konfigurationsfiler

Frontpaneldefinitionen kan distribueras mellan ett stort antal filer som finns lokalt eller i fjärrsystem.

Filer som används för att definiera frontpanelen måste uppfylla följande krav:

- Filnamnet måste sluta på `.fp`, t ex `mail.fp`
- Filen måste finnas i funktionsdatabasens sökväg

Sökvägen till standardfunktionsdatabasen omfattar följande kataloger, vilka söks i denna ordning:

Hemkatalog`/.dt/types`	Egna anpassningar
`/etc/dt/appconfig/types/`*`språk`*	Systemtäckande anpassningar
`/usr/dt/appconfig/types/`*`språk`*	Inbyggda paneler och kontroller

Ytterligare en katalog, *Hemaktalog*/.dt/types/fp_dynamic, används för egna anpassningar gjorda med hjälp av användargränssnittet. Använd inte den här katalogen för manuella anpassningar.

Sökvägen till funktionsdatabasen kan innehålla ytterligare kataloger som lagts till för att konfigurera systemet som ska användas i nätverk. Ytterligare fjärradresser läggs speciellt till när systemet konfigureras för åtkomst av en programserver. Mer information finns i ”Databassökväg (funktion / datatyper)” på sidan 120.

Hur frontpanelen sätts samman: prioritetsregler

Frontpanelen assembleras av alla konfigurationsfiler som finns i funktionsdatabasens sökväg.

När det uppstår en konflikt mellan olika delar av definitionen, bestämer prioritetsregler vilken definition som ska användas. Två komponenter krockar med varandra när de:

- har samma kontrollnamn, CONTAINER_NAME och CONTAINER_TYPE
- konkurrerar om samma position (genom att ha olika namn men samma CONTAINER_NAME, CONTAINER_TYPE och POSITION_HINTS).

Frontpanelen använder dessa prioritetsregler:

- Om olika komponenter har samma kontrollnamn, behållarnamn och -typ, används den komponent som läses först

  Om t ex både en systemtäckande och inbyggd kontroll innehåller dessa fält men i övrigt är olika:

  ```
  CONTROL TextEditor
  {
    CONTAINER_TYPE BOX
    CONTAINER_NAME Top
    …
  }
  ```

  prioriteras den systemtäckande kontrollen.

- Om två olika komponenter konkurrerar om samma position, placeras de i den ordning de läses.

Om en användare t ex skapar en ny egen kontroll för huvudpanelen (CONTAINER_TYPE BOX och CONTAINER_NAME Top) och kopplar den till POSITION_HINTS 5, knuffar den egna kontrollen den inbyggda och alla andra kontroller med högre positionsnummer en position åt höger.

Obs! – När du ändrar en kontroll genom att skapa en ny systemtäckande eller egen version av den, måste den nya kontrolldefinitionen ange samma kontrollnamn, CONTAINER_NAME och CONTAINER_TYPE. I annat fall visas både den nya kontrollen och den befintliga.

Skapa frontpanelsfiler dynamiskt

När användaren anpassar frontpanelen med Installera ikon och dess egna menyer, skrivs filer till katalogen *Hemkatalog*/.dt/types/fp_dynamic.

Frontpanelen skapar ytterligare en fil, *Hemkatalog*/.dt/sessions/dtwmfp.session, som används för att spara och återställa status på den anpassade frontpanelen för varje session.

Administrera anpassning av användargränssnitt

Användarna använder kontrollernas menyer och Installera ikon för att anpassa frontpanelen.

I det här avsnittet beskrivs hur du:

- Förhindrar vissa egna anpassningar. Du kanske vill göra det omöjligt för användarna att ta bort en kontroll.
- Ångra egna anpassningar. En användare kanske vill att du ska återställa en kontroll som tagits bort av misstag.

▼ Förhindra egna anpassningar

1. Om kontrollen är inbyggd kopierar du dess definition från /usr/dt/appconfig/types/*språk*/dtwm.fp till /etc/dt/appconfig/types/*språk*/*name*.fp.
2. Lägg till följande rad i kontrolldefinitionen:

```
LOCKED    True
```

▼ Så här återställer du en kontroll eller panelundermeny som tagits bort

Frontpanelfunktionen Återställ frontpanelen i programgruppen Skrivbordsverktyg tar bort alla anpassningar av användargränssnittet. Användarna kan använda den här funktionen för att ta bort alla egna anpassningar som gjorts med frontpanelens menyer.

Använd proceduren nedan för att återställa individuella kontroller.

- I katalogen *Hemkatalog*`/.dt/types/fp_dynamic` tar du bort filen som skapades när användaren tog bort kontrollen. Kontrollen har samma namn som originalkontroollen som togs bort.

  Om användaren t ex tog bort kontrollen för ikonredigeraren innehåller en fil i katalogen `fp_dynamic`:

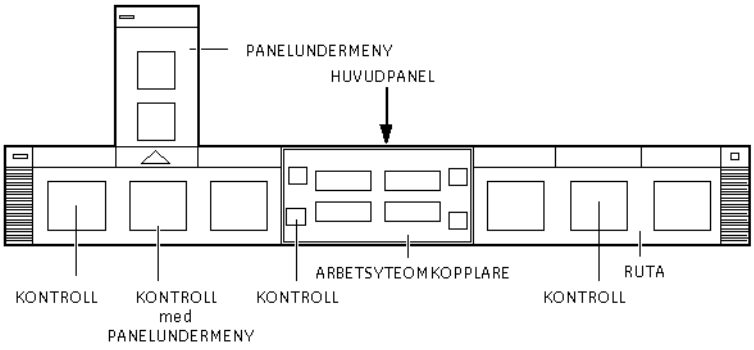
```
CONTROL IconEditor
{
  ...
   DELETE   True
}
```

När användaren tar bort en panelundermeny skapas en separat dynamisk fil för panelundermenyn och för varje kontroll på den.

Hur frontpanelsdefinitionen är organiserad

Frontpanelen skapas genom att definitioner för dess komponenter assembleras. Var och en av dessa komponenter har en obligatorisk syntax som definierar var komponenten placeras på frontpanelen, hur den ser ut och hur den fungerar.

Komponenter på frontpanelen



figur 13-1 Komponenter på frontpanelen

Frontpanelen sätts ihop så här:

- PANEL är behållaren på den översta nivån, eller överordnad, för hela frontpanelen.
- PANEL är en behållare för en eller flera rutor.
- En RUTA är en behållare för en eller flera kontroller.

Det finns två specialtyper av behållare:

- En PANELUNDERMENY är associerad med en viss kontroll (kontrollen är behållaren för panelundermenyn). Panelundermenyer ”viks upp” från kontrollen som den är associerad till.
- ARBETSYTEOMKOPPLAREN innehåller knappar för att ändra arbetsytor samt ytterligare kontroller.

Allmän syntax för frontpanelsdefinitionen

Varje komponent i frontpanelen definieras separat med syntaxen:

```
COMPONENT  namn
{
 KEYWORD          värde
 KEYWORD          värde
   …
}
```

Vissa nyckelord är obligatoriska och andra valfria. Mer information finns i kommandobeskrivningen till `dtfpfile(4X)`.

PANEL-definition

`PANEL` är komponenten på den översta nivån. Dess definition innehåller:

- Frontpanelnamnet
- Fält som beskriver allmänt utseende och allmänna metoder för hela frontpanelen

```
PANEL frontpanelnamn
{
    KEYWORD        värde
    KEYWORD        värde
    …
```

Frontpanelnamn är ett unikt namn på frontpanelen. Standardnamnet är ”FrontPanel”.

RUTA-definitioner

En `RUTA`-definition beskriver:

- Namnet på `RUTAN`
- Vilken `PANEL` rutan finns i (`CONTAINER_NAME`)
- Rutans placering på `PANELEN` (`POSITION_HINTS`)
- Fält som beskriver utseende och metod som gäller för hela rutan

```
BOX rutans_namn
{
    CONTAINER_NAME      frontpanelnamn
    POSITION_HINTS      position
    KEYWORD             värde
    KEYWORD             värde
    …
}
```

KONTROLL-definitioner

En `KONTROLL`-definition beskriver:

- Namnet på `KONTROLl`

- Om kontrollen finns i en ruta, panelundermeny eller arbetsyteomkopplare (CONTAINER_TYPE)
- Vilken ruta, panelundermeny eller arbetsyteomkopplare kontrollen finns i (CONTAINER_NAME)
- Placeringen av KONTROLL i RUTAN (POSITION_HINTS)
- Fält som beskriver utseende och metod för kontrollen

```
CONTROL kontrollnamn
{
    CONTAINER_TYPE      RUTA eller UNDERPANEL eller ARBETSYTEOMKOPPLARE
    CONTAINER_NAME      rutans namn eller panelundermenyns namn eller namnet på
arbetsyteomkopplaren
    TYPE                kontrolltyp
    POSITION_HINTS      position
    KEYWORD             värde
    KEYWORD             värde
    …
}
```

UNDERPANEL-definitioner

En UNDERPANEL-definition beskriver:

- Namnet på UNDERPANELEN
- Namnet på kontrollen som panelundermenyn är kopplad till (CONTAINER_NAME)
- Fält som beskriver utseende och metod för panelundermenyn

```
SUBPANEL panelundermenynamn
{
    CONTAINER_NAME      kontrollnamn
    KEYWORD             värde
    KEYWORD             värde
   …
}
```

Definition för Arbetsyteomkopplaren

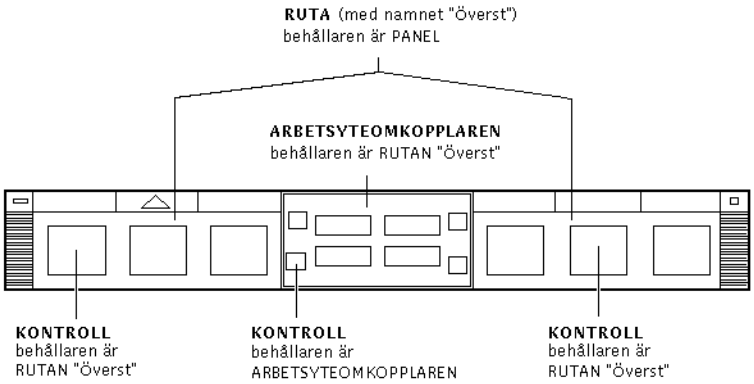
Definitionen för ARBETSYTEOMKOPPLAREN beskriver:

- Namnet på ARBETSYTEOMKOPPLAREN
- Vilken RUTA som ARBETYTEOMKOPPLAREN finns i (CONTAINER_NAME)
- Positionen för ARBETSYTEOMKOPPLAREN i RUTAN (POSITION_HINTS)
- Fält som beskriver utseende och metod för ARBETSYTEOMKOPPLAREN

```
SWITCH namnet på arbetsyteomkopplaren
{
    CONTAINER_NAME      rutans_namn
    POSITION_HINTS      position
    KEYWORD             värde
    KEYWORD             värde
   …
}
```

Ändra huvudpanelen

Huvudpanelen är frontpanelens fönster minus panelundermenyerna.



figur 13-2 Behållare på huvudpanelen

De ändringar du kan göra är bl a:

- Lägga till och ta bort kontroller
- Byta plats på kontroller

▼ Så här lägger du till en kontroll på huvudpanelen

1. Skapa en konfigurationsfil för frontpanelen:
   - Systemtäckande: `/etc/dt/appconfig/types/`*`språk`*`/*.fp`
   - Egen: *Hemkatalog*`/.dt/types/*.fp`
2. Definiera kontrollen i filen.

   Använd fälten `CONTAINER_NAME` och `CONTAINER_TYPE` för att ange behållare för kontrollen:

   ```
   CONTAINER_NAME  Top
   CONTAINER_TYPE  BOX
   ```

   Använd `POSITION_HINTS` för att ange placering från vänster till höger för kontrollen. Eftersom anpassade kontroller har prioritet över inbyggda, kommer den nya kontrollen att ”knuffa” den befintliga kontrollen med den positionen en position åt höger.
3. Spara konfigurationsfilen.
4. Skapa en ikon för frontpanelkontrollen.

   Se ”Ange vilken ikonbild som ska användas av kontrollen” på sidan 223.
5. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Följande kontrolldefinition i filen `/etc/dt/appconfig/types/`*`språk`*`/audio.fp` infogar en ljudprogramkontroll mellan kontrollerna för Klocka och Kalender.

```
CONTROL AudioApplication
{
 TYPE             icon
 CONTAINER_NAME   Top
 CONTAINER_TYPE   BOX
 ICON             AudioApp
 POSITION_HINTS   2
 PUSH_ACTION      StartAudioApplication
 PUSH_RECALL      true
}
```

▼ Så här tar du bort en kontroll

1. Skapa frontpanelens konfigurationsfil:

- Systemtäckande: /etc/dt/appconfig/types/*språk*/*namn*.fp
- Egen: *Hemkatalog*/.dt/types/*namn*.fp

2. Kopiera definitionen för kontrollen som du vill ta bort till den nya filen.

   Om kontrollen är inbyggd finns definitionen i /usr/dt/appconfig/types/*språk*/dtwm.fp.

   Du behöver inte kopiera hela definitionen. Delen du kopierar måste dock innehålla fälten CONTAINER_NAME och CONTAINER_TYPE.

3. Lägg till fältet DELETE i definitionen:

   ```
   DELETE   True
   ```

4. Spara konfigurationsfilen.

5. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Följande kontrolldefinition som finns i filen /etc/dt/appconfig/types/*språk*/TrashCan.fp tar bort kontrollen för papperskorg från frontpanelen.

```
CONTROL Trash
{
 CONTAINER_NAME  Top
 CONTAINER_TYPE  BOX
 DELETE          True
}
```

▼ Så här ändrar du en kontroll

Använd den här proceduren om du behöver ändra kontrolldefinitionen, t ex för att ändra ikonbilden.

1. Kopiera hela kontrolldefinitionen från /usr/dt/appconfig/types/*språk*/dtwm.fp till:
   - Systemtäckande: /etc/dt/appconfig/types/*språk*/*namn*.fp
   - Egen: *Hemaktalog*/.dt/types/*namn*.fp.

2. Redigera fältet du vill ändra. Du kan också lägga till ytterligare fält.

3. Spara filen.

4. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

▼ Så här byter du position för kontrollen

1. Kopiera kontrolldefinitionerna för kontrollerna vars positioner du vill ändra från `/usr/dt/appconfig/types/`*språk*`/dtwm.fp` till:
   - Systemtäckande: `/etc/dt/appconfig/types/`*språk*`/`*namn*`.fp`
   - Egen: *Hemkatalog*`/.dt/types/`*namn*`.fp`.

   Du måste kopiera hela kontrolldefinitionen för alla kontroller som ska flyttas.
2. Byt ut värdena på fälten `POSITION_HINTS` mellan kontrolldefinitionerna.
3. Spara filen.
4. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Följande kontrolldefinitioner som finns i filen `/etc/dt/appconfig/types/C/MailHelp.fp` byter plats på kontrollerna för post och hjälphanteraren och låser dessa kontroller för egna ändringar.

```
CONTROL Mail
{
    POSITION_HINTS     12
    LOCKED             True
    …resten av kontrolldefinitionen
}

CONTROL Help
{
    POSITION_HINTS     5
    LOCKED             True
    …resten av kontrolldefinitionen
}
```

▼ Så här ersätter du en frontpanelkontroll

♦ Skapa en annan kontrolldefinition med samma namn:
- *kontrollnamn*
- `CONTAINER_NAME` värde

Följande två kontroller definieras i två olika konfigurationsfiler. Dessa kontroller har samma kontrollnamn och behållarnamn, och betraktas därför som samma kontroll.

- Definition i `/etc/dt/appconfig/types/C/SysControls.fp`:

```
Control ImportantApplication
{
  CONTAINER_NAME     Top
  CONTAINER_TYPE     BOX
  POSITION_HINTS     2
  …
```

- Definition in *Hemkatalog*`/.dt/types/MyControls.fp`:

```
Control ImportantApplication
{
  CONTAINER_NAME     Top
  CONTAINER_TYPE     BOX
  POSITION_HINTS     6
  …
```

Den egna kontrollen har prioritet, så kontrollen låses vid position 6.

Ange vilken ikonbild som ska användas av kontrollen

Kontrolldefinitionens `ICON`-fält definierar vilken ikonbild som ska användas till kontrollen.

Värdet på fältet `ICON` kan vara:

- Ett basfilnamn.

  Basfilnamnet är namnet på filen som innehåller ikonen minus filnamnssuffixet för storlek (`m` och `t`) och bildtyp (`bm` och `pm`). Om filer t ex heter `MittSpel.l.pm` och `MittSpel.m.pm`, används `MittSpel`.

  Om du använder basfilnamnet måste ikonfilerna placeras i en katalog i ikonsökvägen:
  - Egna ikoner: *Hemkatalog*`/.dt/icons`
  - Systemtäckande ikoner: `/etc/dt/appconfig/icons/`*språk*

- En absolut sökväg till ikonen, inklusive fullständigt filnamn.

  Du ska bara använda den absoluta sökvägen om ikonfilen inte finns i ikonsökvägen.

Vilken ikonstorlek på ikonen som behövs beror på var kontrollen placeras:

Position	**Storlek**

Huvudpanel	48 gånger 48 bildpunkter (*namn*`.l.pm` eller *namn*`.l.bm`)
Panelundermeny	24 gånger 24 bildpunkter (*namn*`.s.pm` eller *namn*`.s.bm`)

Placera ikonfilen på någon av följande adresser:

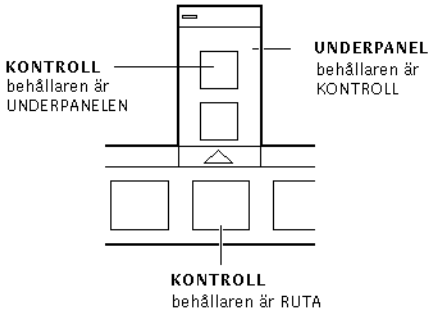
- Egna ikoner: *Hemkatalog*`/.dt/icons`
- Systemtäckande ikoner: `/etc/dt/appconfig/icons/`*språk*

Skapa och ändra panelundermenyer

Användarna kan skapa och ändra panelundermenyer med frontpanelens menyer.

I det här avsnittet beskrivs hur du gör systemtäckande anpassningar, vilka kräver att du ändrar frontpanelens konfigurationsfiler.

En panelundermeny är ”kopplad” till en kontroll på huvudpanelen.



figur 13-3 En panelundermenys behållare är kontrollen som den är kopplad till

Kopplingen görs i panelundermenyns definition. Fältet `CONTAINER_NAME` anger vilken kontroll som panelundermenyn är kopplad till:

```
CONTROL kontrollnamn
{
    …
}

SUBPANEL panelundermenynamn
{
```

```
    CONTAINER_NAME      kontrollnamn
    …
}
```

▼ Så här skapar du en ny systemtäckande panelundermeny

1. Leta reda på kontrollens *kontrollnamn* i huvudpanelen som du vill koppla panelundermenyn till.

   Om kontrollen är inbyggd finns definitionen i `/usr/dt/appconfig/types/`*språk*`/dtwm.fp`.

2. Skapa en ny fil `/etc/dt/appconfig/types/`*språk*`/*.fp`.

3. Definiera panelundermenyn:

```
SUBPANEL panelundermenynamn
{
  CONTAINER_NAME kontrollnamn
  TITLE          värde
  KEYWORD        värde
  …
}
```

4. Spara den nya konfigurationsfilen.

5. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Anpassa inbyggda panelundermenyer

Du kan ändra allmänna egenskaper för (t ex rubrik) och innehåll i inbyggda panelundermenyer.

▼ *Så här ändrar du allmänna egenskaper för en inbyggd panelundermeny*

1. Skapa en ny konfigurationsfil för frontpanelen:
   - Systemtäckande: `/etc/dt/appconfig/types/`*språk*`/`*namn*`.fp`
   - Egen: *Hemkatalog*`/.dt/types/`*namn*`.fp`.

2. Kopiera hela standarddefinitionen för `SUBPANEL` från `/usr/dt/appconfig/types/`*språk*`/dtwm.fp` till den nya filen:

```
SUBPANEL    panelundermenynamn
{
  …
}
```

3. Ändra definitionen till panelundermenyn.
4. Spara den nya konfigurationsfilen.
5. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Följande definition som finns i filen `users/janne/.dt/types/PerApps.fp` ändrar namn på panelundermenyn Egna program:

```
SUBPANEL PersAppsSubpanel
{
    CONTAINER_NAME     TextEditor
    TITLE              Jannes program
}
```

▼ *Så här lägger du till systemtäckande kontroller i en inbyggd panelundermeny*

1. Skapa en konfigurationsfil för frontpanelen `/etc/dt/appconfig/types/`*språk*`/`*namn*`.fp.`
2. Definiera den systemtäckande kontrollen i filen.

   Använd fälten `CONTAINER_NAME` och `CONTAINER_TYPE` för att ange behållare för kontrollen:

   ```
   CONTROL kontrollnamn
   {
    CONTAINER_NAME      panelundermenynamn
    CONTAINER_TYPE      SUBPANEL
    …
   }
   ```

   Se ”Definiera kontroller på frontpanelen” på sidan 228.
3. Spara konfigurationsfilen.
4. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Följande kontroll som definieras i den nya filen `/etc/dt/appconfig/types/`*språk*`/DigitalClock.fp` lägger till en digital klocka (i programgruppen Skrivbordsverktyg) på panelundermenyn Egna program för alla användare.

```
CONTROL DigitalClockControl
{
    TYPE           icon
    CONTAINER_NAME PerAppsSubpanel
    CONTAINER_TYPE SUBPANEL
    ICON           Dtdgclk
    PUSH_ACTION    DigitalClock
    PUSH_RECALL    True
}
```

▼ *Så här tar du bort en kontroll från en inbyggd panelundermeny*

♦ Använd samma procedur som för att ta bort kontrooller på huvudpanelen. Se ”Så här tar du bort en kontroll” på sidan 220.

▼ *Så här tar du bort kontrollen Installera ikon*

♦ Lägg till följande fält i panelundermenyns definition:

```
CONTROL_INSTALL False
```

▼ Så här tar du bort den automatiska stängningen (Auto-Close) av panelundermenyer

Standardmetoden för panelundermenyer är att de stängs när användaren väljer en kontroll, såvida inte användaren har flyttat panelundermenyn från det ursprungliga positionen.

Frontpanelen kan konfigureraras att hålla panelundermenyerna öppna tills användaren uttryckligen stänger dem.

1. Skapa en konfigurationsfil för frontpanelen i:
   - Systemtäckande: `/etc/dt/appconfig/types/`*språk*`/*.fp`
   - Egen: *Hemaktalog*`/.dt/types/*.fp`
2. Kopiera standarddefinitionen för `PANEL` från `/usr/dt/appconfig/types/`*språk*`/dtwm.fp` till den nya filen:

```
PANEL FrontPanel
{
  ...
}
```

3. Lägg till följande fält i definitioner för `PANEL`:

   ```
   SUBPANEL_UNPOST    False
   ```

4. Spara den nya konfigurationsfilen.

5. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Definiera kontroller på frontpanelen

Användarna kan skapa egna knappar genom att släppa ikoner på Installera ikon-kontrollerna.

Det här sättet att anpassa är enkelt, men funktionaliteten omfattar bara en del av vad som kan göras med kontroller på frontpanelen. En kontroll som skapats med Installera ikon kan t ex inte:

- Ha animeringar
- Visa ett klientfönster
- Ändra utseende när något inträffar (t ex för att ange att ny post har tagits emot)

I det här avsnittet beskrivs hur du manuellt skapar kontroller på frontpanelen.

Referensinformation om syntaxen för frontpanelskontroller finns i kommandobeskrivningen till `dtfpfile(4X)`.

Definitioner till kontroller på frontpanelen

Strukturen på en definition till en frontpanelskontroll är:

```
CONTROL kontrollnamn
{
  TYPE             kontrolltyp
  CONTAINER_NAME   värde
  CONTAINER_TYPE   värde
  andra fält som definierar utseende och metod
}
```

kontrolltyper

Fältet TYPE i kontrolldefinitionen anger basmetod för kontrollen.

TYPE	Kontrollmetod
`icon`	(Standard). kontrollen kör en angiven funktion när användaren klickar på kontrollen eller släpper en fil på den.
`blank`	Platshållare används för att justera mellanrum mellan kontroller.
`busy`	Upptagetsignal. kontrollen blinkar (växlande bilder) när en funktion anropas
`client`	Ett klientfönster i frontpanelen.
`clock`	Klocka.
`date`	Visar aktuellt datum.
`file`	Representerar en fil. Om kontrollen väljs körs filens standardfunktion.

▼ Så här skapar du en ny kontroll

I det här avsnittet beskrivs allmänna steg för hur du definierar en kontroll och hur du skapar olika typer av kontroller.

1. Om kontrollen ska ha en `PUSH_ACTION` och/eller `DROP_ACTION` skapar du funktionsdefinitionerna. Dessa funktioner körs när användaren klickar på kontrollen eller släpper en fil på den.
2. Skapa ikonbildens fil för kontrollen.

   Mer information om ikonstorlek, -namn och -adresser finns i ”Ikonbildsfiler” på sidan 203.
3. Skapa en ny konfigurationsfil för frontpanelen i:
   - Systemtäckande: `/etc/dt/appconfig/types/`*`språk`*`/*.fp`
   - Egen: *Hemkatalog*`/.dt/types/*.fp`
4. Lägg till kontrolldefinitionen i filen.
5. Spara filen.
6. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Skapa en kontroll som kör en funktion när den markeras

Använd dessa fält för att definiera kontrollens metod:

- TYPE: Anges till icon
- PUSH_ACTION: Anger namnet på funktionen som ska köras

Följande kontroll, som placeras på panelundermenyn Egna program, kör ett spel som användaren har skaffat:

```
CONTROL Ball
{
    TYPE               icon
    CONTAINER_NAME     PersAppsSubpanel
    CONTAINER_TYPE     SUBPANEL
    ICON               ball
    PUSH_ACTION        RunBallGame
    HELP_STRING        "Väj den här kontrollen för att spela Ball."
}
```

Följande kontroll placeras i det övre vänstra hörnet av arbetsyteomkopplaren. Den startar en funktion som heter CutDisp.

```
CONTROL StartCutDisp
{
  TYPE             icon
  CONTAINER_NAME   Switch
  CONTAINER_TYPE   SWITCH
  POSITION_HINTS   first
  ICON             cutdisp
  HELP_STRING      "Välj den här kontrollen för att köra cutdisp."
  PUSH_ACTION      CutDisp
}
```

Skapa en kontroll som öppnar en fil

Använd dessa fält för att definiera kontrollens metod:

- TYPE: Anges till file
- FILE_NAME: Anger sökvägen till filen som ska öppnas
- PUSH_ACTION: Anges till Open

Det måste finnas en Open-funktion definierad för filens datatyp.

Följande kontroll placeras längst till höger på huvudpanelen. Den startar textredigeraren med datafilen /users/ellen/PhoneList.txt. Funktionen Open för *.txt-filer är en del av standardfunktionsdatabasen.

```
CONTROL EditPhoneList
{
    TYPE           file
    FILE_NAME      /users/ellen/PhoneList.txt
    CONTAINER_NAME Top
    CONTAINER_TYPE BOX
    POSITION_HINTS last
    ICON           PhoneBook
    HELP_STRING    "Den här kontrollen visar Ellens telefonlista."
    PUSH_ACTION    Open
}
```

Skapa en kontroll som fungerar som målområde

Använd fältet DROP_ACTION för att ange funktionen som körs när användaren släpper en fil på kontrollen. Funktionen måste kunna acceptera ett filargument.

För det mesta innehåller en kontrolldefinition både fältet PUSH_ACTION och DROP_ACTION. Du måste använda samma funktion för funktionerna push och drop.

Följande kontroll som finns på panelundermenyn Egna program kör X-klienten xwud, som använder filargument.

```
CONTROL Run_xwud
{
    CONTAINER_NAME     PerAppsSubpanel
    CONTAINER_TYPE     SUBPANEL
    POSITION_HINTS     2
    ICON               XwudImage
    PUSH_ACTION        RunXwud
    DROP_ACTION        RunXwud
}
```

Skapa en kontroll som övervakar filer

Använd dessa fält för att definiera kontrollens metod:

- TYPE: Ange något av följande värden:

`icon`	Använd den här typen om du vill ange en `PUSH_ACTION` och/eller `DROP_ACTION` för kontrollen.
`file`	Använd den här typen om du vill att kontrollen, när den väljs, ska fungera som filen när någon dubbelklickar på filens ikon i filhanteraren.

- `ICON` och `ALTERNATE_ICON:` Beskriver de bilder som används för att ange ändrad och icke ändras status på den övervakade filen.
- `MONITOR_TYPE`: Beskriver vilka villkor som gör att bilden ändras. Använd något av följande värden:

`mail`	Kontrollen ändrar utseende när information läggs till i filen.
`file`	Kontrollen ändras när angiven fil fylls med något.

- `FILE_NAME`: Anger vilken fil som ska övervakas.

Följande kontroll letar efter en fil som heter `möten` som du förväntar dig ska överföras till ditt system med en anonym ftp. Kontrollen placeras högst upp i panelundermenyn Egna program.

```
CONTROL MonitorCalendar
{
    TYPE           file
    CONTAINER_NAME PersonalApps
    CONTAINER_TYPE SUBPANEL
    POSITION_HINTS first
    FILE_NAME      /users/ftp/möten
    MONITOR_TYPE   file
    ICON           mötenno
    ALTERNATE_ICON mötenyes
}
```

Skapa en växlingskontroll

En växlingskontroll kontrollerar för att se om den process som startas av `PUSH_ACTION` redan körs. Om processen inte körs, körs `PUSH_ACTION`. Om processen redan körs, placeras fönstret ovanpå alla andra fönster i aktuell arbetsyta.

Använd dessa fält för att definiera kontrollens metod:

- `PUSH_RECALL`: Anges till `True`.
- `CLIENT_NAME`: Anger namnet på kontrollens klient.

Värdet på CLIENT_NAME måste matcha den första strängen (*res_namn*) i egenskapen WM_CLASS i fönstret på programmets översta nivå. Mer information finns i kommandobeskrivningen till xprop(1).

- PUSH_ACTION: Beskriver funktionen som körs när användaren klickar på kontrollen.

Följande kontroll kör en förekomst av ett program vars funktion heter MyEditor.

```
CONTROL MyEditor
{
    TYPE           icon
    CONTAINER_NAME Top
    CONTAINER_TYPE BOX
    POSITION_HINTS 15
    PUSH_RECALL    True
    CLIENT_NAME    BestEditor
    PUSH_ACTION    StartMyEditor
    ICON           MyEd
}
```

▼ *Så här skapar du en klientfönsterkontroll*

En klientfönsterkontroll är ett programfönster som är inbäddat i frontpanelen. Du kan t ex placera en belastningsmätare på frontpanelen genom att skapa en xload-klientfönsterkontroll.

1. Definiera kontrollen.

   Använd dessa fält för att definiera kontrollens metod:

   - TYPE: Anges till client.
   - CLIENT_NAME: Anger klienten som ska startas.

     Värdet på CLIENT_NAME måste matcha den första strängen (*res_namn*) i egenskapen WM_CLASS i fönstret på programmets översta nivå. Mer information finns i kommandobeskrivningen till xprop(1).
   - CLIENT_GEOMETRY: Anger storlek, i bildpunkter, som behövs för klientens frontpanelfönster.

     I kommandobeskrivningen till xwininfo(1) beskrivs hur du får reda på fönsterstorleken i bildpunkter.

2. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

3. Starta om klienten från kommandoraden i en terminalemulator.

Följande kontroll visar en belastningsmätare på 30 × 20 bildpunkter.

```
CONTROL LoadMeter
{
    TYPE           client
    CONTAINER_NAME Top
    CONTAINER_TYPE BOX
    CLIENT_NAME    xload
    CLIENT_GEOMETRY30x20
}
```

Om klienten inte sparas och återställs mellan sessionerna, kanske du vill konfigurera kontrollen att starta klienten när användaren klickar på den. Du kan t ex konfigurera kontrollen för LoadMeter att starta `xload` genom att lägga till följande rad i definitionen:

```
PUSH_ACTION    StartXload
```

och skapa funktionen:

```
ACTION StartXload
{
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING    /usr/contrib/bin/X11/xload
}
```

▼ *Så här animerar du en kontroll*

Du kan skapa en animeringssekvens som används när användaren väljer kontrollen eller släpper ett objekt på den.

För att kunna ha en animering måste kontrollen:

- Vara av typen `icon`
- Ha en `PUSH_ACTION` eller `DROP_ACTION`

1. Ange animeringssekvensen med hjälp av komponenten `ANIMATION`:

```
ANIMATION animeringsnamn
{
  ikonbild          [förskjutning]
  ikonbild          [förskjutning]
  …
}
```

där *förskjutning* är tidsförskjutningen mellan animeringsikonerna i millisekunder.

2. Lägg till fältet PUSH_ANIMATION och/eller DROP_ANIMATION till kontrolldefinitionen. Värdet är namnet på sekvensen ANIMATION.

Följande rad animerar en kontroll som startar programmet BestEditor. Tidsförskjutningen mellan ikonerna är 300 millisekunder. I exemplet antas det att du har skapat ikonfilerna frame1, frame2, osv.

```
CONTROL BestEditor
{
    …
    PUSH_ANIMATION BestEdAnimation
    …
}

ANIMATION BestEdAnimation
{
    frame1     300
    frame2
    …
}
```

Objekthjälp för kontroller på frontpanelen

Det finns två sätt att göra kontrollhjälp tillgänglig:

- Lägga in en hjälpsträng i kontrolldefinitionen.

  Hjälpsträngen visas i hjälpfönstret när användaren anropar objekthjälp till kontrollen. Hjälpsträngen kan inte innehålla formatering (t ex rubriker) eller länkar.

  Om du vill lägga till en hjälpsträng anger du den i kontrolldefinitionen:

  HELP_STRING *hjälpsträng*

- Ange ett hjälpavsnitt i en registrerad hjälpvolym.

  Ett hjälpavsnitt är skriven information som utnyttjar alla möjligheter som finns i hjälpsystemet. För att kunna skriva hjälpavsnitt måste du använda skrivbordets Help Developer's Kit.

  För att lägga in ett hjälpavsnitt anger du hjälpvolym och avsnitts-ID i kontrolldefinitionen:

```
HELP_VOLUME  hjälpvolymnamn
HELP_TOPIC   avsnitts-ID
```

Anpassa arbetsyteomkopplaren

Du kan anpassa arbetsyteomkopplaren på flera sätt:

- Ändra antalet arbetsytor
- Ändra knappens utseende
- Ändra antalet kontroller i knappen

▼ Så här ändrar du antalet arbetsytor

♦ Ändra följande resurs för arbetsytehanteraren:

```
Dtwm*workspaceCount:n
```

Mer information finns i ”Så här ändrar du antalet systemtäckande arbetsytor” på sidan 244.

▼ Så här ändrar du antalet knapprader

♦ Ändra fältet NUMBER_OF_ROWS i definitionen SWITCH.

Följande definition definierar en treraders knapp.

```
SWITCH Switch
{
    CONTAINER_NAME rutans_namn
    NUMBER_OF_ROWS 3
  …
}
```

▼ Så här ändrar du eller lägger du till kontroller i arbetsyteomkopplaren

1. Skapa en konfigurationsfil för frontpanelen med kontrolldefinitionen.
   - Ange att kontrollen ska finnas inuti knappen:

     ```
     CONTAINER_NAME   Switch
     CONTAINER_TYPE   SWITCH
     ```

   - Ange positionen i arbetsyteomkopplaren:

     ```
     POSITION_HINTS     n
     ```

där *n* är ett heltal. Positionerna numreras i ordning från vänster till höger, uppifrån och ned. (För den normala tvåraders knappen är positionerna mellan 1 och 4.)

2. Skapa kontrollbilden för kontrollen. Storleken bör vara 16 gånger 16 bildpunkter.

Följande kontroll placerar en Terminal-kontroll i knappen.

```
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
}
```

Kontrollen använder en inbyggd ikonbild och samma hjälpavsnitt som används av Terminal-kontrollen på panelundermenyn Egna program.

Allmän konfiguration av frontpanelen

Med frontpanelens PANEL-syntax kan du:

- Ändra positionen för frontpanelen
- Ändra fönstrets detaljer
- Ange utseende och metod för kontrollerna

Standardbeskrivningen för PANEL finns i /usr/dt/appconfig/types/*språk*/dtwm.fp.

Mer information finns i kommandobeskrivningen till dtfpfile(4X).

Allmänna steg

1. Skapa en ny konfigurationsfil till frontpanelen i /etc/dt/appconfig/types/*språk* eller *Hemkatalog*/.dt/types.

2. Kopiera standardbeskrivningen för PANEL från /usr/dt/types/*språk*/dtwm.fp till den nya filen.
3. Redigera PANEL-beskrivningen.

Den nya PANEL-beskrivningen har högre prioritet än standardbeskrivningen.

▼ Så här ändrar du frontpanelens standardposition

♦ Använd fältet PANEL_GEOMETRY i PANEL-definitionen för att ange positionen.

Följande panel finns i det övre högra hörnet.

```
PANEL SpecialFrontPanel
{
  PANEL_GEOMETRY      -1+1
   …
}
```

▼ Namnge kontroller på huvudpanelen

1. Lägg till följande rad i PANEL-definitionen:

   ```
   DISPLAY_CONTROL_LABELS   True
   ```

2. Lägg till fältet LABEL för varje kontroll.

Kontrollnamn används om LABEL inte anges.

▼ Så här ändrar du klickmetod för kontroller

♦ Använd fältet CONTROL_BEHAVIOR i PANEL-definitionen för att ange hur användaren kör PUSH_ACTION för en kontroll. Värden på fältet är:

single_click	Användaren klickar på kontrollen för att köra PUSH_ACTION
double_click	Användaren dubbelklickar på kontrollen för att köra PUSH_ACTION

▼ Så här skapar du en helt ny frontpanel

När du vill göra mycket omfattande ändringar kan det vara bättre att skapa en helt ny frontpanel.

För att undvika krockar med komponenter i den inbyggda frontpanelen, ska den nya frontpanelen ha nya namn på PANEL och andra behållare.

1. Skapa PANEL-komponenten för den nya frontpanelen. Ge den ett unikt namn:

   ```
   PANEL frontpanelnamn
   {
       …
   }
   ```

2. Skapa nya rutor och kontroller. Använd nya namn på behållarna.

   Om du vill använda befintliga komponenter måste du kopiera deras definitioner och ändra värdet CONTAINER_NAME.

3. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Exempel på hur du skapar en egen frontpanel med tre rader

I följande exempel ändras standardfrontpanelen så att dess kontroller ordnas i tre rader.

1. Kopiera /usr/dt/appconfig/types/*språk*/dtwm.fp till *Hemkatalog*/.dt.types/MyFrontPanel.fp. Tilldela filen skrivbehörighet.

   Det är den här filen du redigerar för att skapa en ny frontpanel.

2. Ändra namnet på frontpanelen:

   ```
   PANEL NewFrontPanel
   ```

3. Ändra namnet på rutan som heter Top och redigera behållarens namn:

   ```
   BOX NewFrontPanelTop
   {
     CONTAINER_NAMENewFrontPanel
     POSITION_HINTSfirst
     …
   }
   ```

4. Lägg till rutdefinitioner för den mellersta och undre raden:

```
BOX NewFrontPanelMiddle
{
  CONTAINER_NAME    NewFrontPanel
  POSITION_HINTS    first
}

BOX NewFrontPanelBottom
{
  CONTAINER_NAME    NewFrontPanel
  POSITION_HINTS    first
}
```

5. Ändra `CONTAINER_NAME` för följande kontroller till `NewFrontPanelTop`:
   - Clock
   - Date
   - Home
   - TextEditor
   - Mail
6. Ändra `CONTAINER_NAME` för följande kontroller till `NewFrontPanelBottom`:
   - Printer
   - Style
   - Applications
   - Help
   - Trash
7. Ändra `CONTAINER_NAME` för arbetsyteomkopplaren till `NewFrontPanelMiddle`.
8. Ange resursen:

   ```
   Dtwm*frontPanel*name:  NewFrontPanel
   ```
9. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Anpassa arbetsytehanteraren 14

I det här kapitlet beskrivs hur du anpassar skrivbordets arbetsytehanterare.

Arbetsytehanterarens konfigurationsfiler	*242*
Anpassa arbetsytor	*244*
Arbetsytehanterarens menyer	*246*
Anpassa knappkopplingar	*249*
Anpassa tangentkopplingar	*252*
Växla mellan standardmetod och anpassad metod	*254*

Arbetsytehanteraren är fönsterhanteraren som finns på skrivbordet. På samma sätt som andra fönsterhanterare kontrollerar den:

- Hur olika delar av fönsterramen ser ut
- Fönstrets metoder, bl a staplingsordning och fokusering
- Tangentbords- och knappkopplingar
- Hur minimerade fönster ser ut
- Arbetsyta- och Fönster-menyn

Arbetsytehanteraren kontrollerar dessutom följande skrivbordskomponenter:

- *Arbetsytor.* Arbetsytehanteraren kontrollerar antalet arbetsytor och håller reda på vilka fönster som är öppna i varje arbetsyta.

- *Arbetsytans bakgrund*. Användaren ändrar bakgrund med hjälp av Arbetsinställningar. Hantering av bakgrunder är dock en funktion för arbetsytehanteraren.
- *Frontpanelen*. Trots att frontpanelen har sina egna konfigurationsfiler, skapas och hanteras den med arbetsytehanteraren.

Fler av dessa komponenter kan ändras med Arbetsinställningar. Med Arbetsinställningar kan du snabbt och enkelt ändra sådant som behöver ändras ofta. Andra resurser måste anges manuellt.

Arbetsytehanteraren är `dtwm`. Den baseras på OSF/Motif Window Manager.

Se även

- Referensinformation om arbetsytehanteraren finns i kommandobeskrivningen till `dtwm(1)` och `dtwmrc(4)`.
- Information om hur du anger resurser för arbetsytehanteraren finns i ”Ange programresurser” på sidan 257.
- Information om frontpanelens konfigurationsfiler finns i Kapitel 13, ”Särskild anpassning av frontpanelen””.

Ytterligare information om hur du anger resurser finns i ”Ange programresurser” på sidan 257.

Arbetsytehanterarens konfigurationsfiler

Arbetsytehanteraren får information om fönstermenyer, arbetsytemenyer samt knapp- och tangentkopplingar från en konfigurationsfil.

Någon av följande filer används:

- Egen fil: *Hemkatalog*`/.dt/dtwmrc`
- Systemtäckande fil: `/etc/dt/config/`*språk*`/sys.dtwmrc`
- Inbyggd fil: `/usr/dt/config/`*språk*`/sys.dtwmrc`

Arbetsytehanteraren söker efter konfigurationsfilen i ordningen som anges ovan, och använder den första som påträffas.

För användare som använder mer än ett sessionsspråk, kan en egen, språkberoende konfigurationsfil med namnet *Hemkatalog*`/.dt/`*språk*`/dtwrmc` få prioritet över *Hemkatalog*`/.dt/dtwmrc`.

▼ Så här skapar du och ändrar en egen konfigurationsfil

Den egna konfigurationsfilen för arbetsytehanteraren är *Hemkatalog*/.dt/dtwmrc. Om filen finns är det den som används.

1. Dubbelklicka på Redigera Dtwmrc i programgruppen Skrivbordsverktyg.

   Om du redan har en egen dtwmrc-fil hämtas den till redigeraren. Om inte, kopieras sys.dtwmrc till *Hemkatalog*/.dt/dtwmrc som sedan hämtas till redigeraren.

2. Redigera filen.

3. Avsluta redigeraren.

   Filen sparas som en egen dtwmrc, oavsett ursprungskälla.

▼ Så här skapar du en systemtäckande konfigurationsfil

Den systemtäckande konfigurationsfilen för arbetsytehanteraren är /etc/dt/config/*språk*/sys.dtwmrc.

♦ Kopiera /usr/dt/config/*språk*/sys.dtwmrc till /etc/dt/config/*språk*/sys.dtwmrc.

Obs! – Den här filen används inte om *Hemkatalog*/.dt/dtwmrc finns.

▼ Så här inkluderar du andra filer

♦ Använd syntaxen:

```
include
{
   sökväg
  sökväg
   …
}
```

Följande rader inkluderar filen /users/ellen/mymenu:

```
include
{
   /users/ellen/mymenu
}
```

Programsatserna Include är användbara för att skaffa ytterligare funktionalitet utan att hela konfigurationsfilen behöver kopieras. En användare kanske vill skapa en ny tangentkoppling utan att administrera hela konfigurationsfilen. Användaren skapar då filen *Hemkatalog*/.dt/dtwmrc med det här innehållet:

```
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"
  …
}
```

▼ Så här startar du om arbetsytehanteraren

Arbetsytehanteraren måste startas om för att ändringarna i konfigurationsfilen ska träda i kraft.

♦ Välj Starta om arbetsytehanteraren på Arbetsyta-menyn (klicka på tredje musknappen när pekaren finns på bakgrunden).

Anpassa arbetsytor

De flesta anpassningar av arbetsytor, t ex ändringar av namn och antalet arbetsytor, kan göras av användaren med skrivbordets gränssnitt. Arbetsytehanteraren omfattar dock resurser för att ange systemtäckande standardvärden.

▼ Så här ändrar du antalet systemtäckande arbetsytor

Standardskrivbordskonfigurationen omfattar fyra arbetsytor. Användaren kan lägga till och ta bort arbetsytor med menyerna som hör till arbetsyteomkopplaren.

Arbetsytehanteraren innehåller en resurs för att ändra standardantalet arbetsytor.

♦ Använd resursen `workspaceCount` för att ange antalet arbetsytor:

`Dtwm*workspaceCount:` *antal*

Information om hur du anger resurser för arbetsytehanteraren finns i ”Ange programresurser” på sidan 257.

Följande resurs anger antalet arbetsytor till sex:

```
Dtwm*workspaceCount: 6
```

▼ Namnge systemtäckande arbetsytor

Internt numreras arbetsytor med numreringskonventionen `ws`*n*, där *n* är 0, 1, 2, osv. Standardantalet på fyra arbetsytor numreras internt `ws0` t o m `ws3`.

♦ Använd resursen `title` för att ändra namn på en viss arbetsyta:

`Dtwm*ws`*n*`:` *namn*

Information om hur du anger resurser för arbetsytehanteraren finns i ”Ange programresurser” på sidan 257.

Följande resurser anger namn på de fyra standardarbetsytorna:

```
Dtwm*ws0*title:  Anna
Dtwm*ws1*title:  Dag
Dtwm*ws2*title   Julia
Dtwm*ws3*title   Peter
```

▼ Så här skapar du ytterligare bakgrunder

1. Skapa bakgrundsbilderna. De kan vara bitmönster- eller bildpunktsmönster filer.
2. Placera bakgrunden i någon av följande kataloger. (Du kanske först måste skapa katalogen.)
   - Systemtäckande bakgrunder: `/etc/dt/backdrops`
   - Egna bakgrunder: *Hemkatalog*`/.dt/backdrops`
3. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Systemtäckande och egna bakgrunder läggs till de inbyggda i `/usr/dt/backdrops`.

Du ersätter en befintlig inbyggd bakgrund genom att skapa en egen eller systemtäckande bakgrund med samma namn.

▼ Så här ersätter du bakgrunden med en grafisk bild

Bakgrunder läggs som skikt ovanpå skärmens rotfönster. I dialogrutan Arbetsinställningar - Bakgrund finns inställningen NoBackdrop som gör bakgrunden genomskinlig.

Det finns bara ett rotfönster bakom alla arbetsytebakgrunder. Därför finns en grafisk bild som placerats på rotfönstret kvar för alla arbetsytor. Du kan ange vilka arbetsytor som ska täcka rotfönstret med en bakgrund. Bilden som visas när NoBackdrop är aktivt är dock densamma för alla arbetsytor.

1. Skapa den grafiska bilden. Den måste ha ett format som kan visas på rotfönstret med något av de tillgängliga verktygen. Om du t ex tänker använda `xsetroot` skapar du en bitmönsterfil.
2. Skapa den exekverbara filen *Hemkatalog*`/.dt/sessions/sessionetc` om den inte redan finns. Filen `sessionetc` körs varje gång användaren loggar in.
3. Placera kommandot som visar bilden i filen `sessionetc`.

   Följande kommando visar rotfönstret sida vid sida med angiven bitmönsterfil:

   ```
   xsetroot -bitmap /users/ellen/.dt/icons/root.bm
   ```

Arbetsytehanterarens menyer

Arbetsytehanteraren har tre standardmenyer:

Arbetsyta-menyn	Kallas också rotmenyn. Den visas när användren klickar på tredje musknappen när pekaren finns på bakgrunden. Menyn är associerad med musknappen via en knappkoppling.

Fönster-menyn	Menyn visas när användaren klickar med första eller tredje musknappen och pekaren finns på fönstermenyknappen (i det övre vänstra hörnet på fönsterramen). Menyn är associerad med knappen via resursen `windowMenu`.
Frontpanel-menyn	Menyn visas när användaren klickar med första eller tredje musknappen och pekaren finns på frontpanelens fönstermenyknapp.

Arbetsytehanterarens menysyntax

Arbetsytehanterarens menyer har syntaxen:

```
Menu Menynamn
{
    val1 [alternativbokstav] [snabbval] funktion [argument]
    val2 [alternativbokstav] [snabbval] funktion [argument]
    …
}
```

där:

val	Texten eller bitmönstret som visas på menyn. Om texten innehåller mellanslag, omsluter du den med citattecken. För bitmönster använder du syntaxen **@/***sökväg*.
alternativbokstav	Ett enstaka tecken som fungerar som ett snabbval med tangentbordet. Det anges i formatet *_tecken*.
snabbval	Ett snabbval med tangentbordet är aktivt oavsett om menyn visas eller inte. Snabbval har syntaxen *styrtangent*`<Key>` *Tangentnamn*, där *styrtangent* är `Ctrl`, `Skift`, `Alt` eller `Lock`. En lista över alla tangentnamn finns i filen `keysymdef.h` i katalogen "X11 include".
funktion	Funktionen som ska utföras när valet är gjort. En lista över funktioner finns i kommandobeskrivningen till `dtwmrc(4)`.
argument	Funktionsargument. Mer information finns i kommandobeskrivningen till `dtwmrc(4)`.

Följande menyalternativ med namnet Återställ normaliserar fönstret. När menyn visas och du skriver "R" återställs också fönstret. Om du trycker på `Alt`+ F5 händer samma sak.

```
Återställl   _R   Alt<Key> F5   f.normalize
```

Obs! – Fullständig information om arbetsytehanterarens menysyntax finns i kommandobeskrivningen till `dtwmrc(4)`.

▼ Så här ändrar du den befintliga Arbetsyta-menyn (rotmenyn)

1. Öppna rätt fil för redigering:
   - Egen: *Hemkatalog*`/.dt/dtwmrc`
   - Systemtäckande: `/etc/dt/config/`*språk*`/sys.dtwmrc`

   Information om hur du skapar dessa filer finns i ”Arbetsytehanterarens konfigurationsfiler” på sidan 242.

2. Redigera beskrivningen av Arbetsyta-menyn.

   Standardmenyn heter DtRootMenu.

```
Menu DtRootMenu
{
    "Arbetsyta-menyn"                  f.title
    "Lägg överst"                      f.circle_up
    "Lägg underst"                    f.circle_down
    …
}
```

▼ Så här skapar du en ny Arbetsyta-meny (rotmeny)

1. Öppna rätt fil för redigering:
   - Egen: *Hemkatalog*`/.dt/dtwmrc`
   - Systemtäckande: `/etc/dt/config/`*språk*`/sys.dtwmrc`

   Information om hur du skapar dessa filer finns i ”Arbetsytehanterarens konfigurationsfiler” på sidan 242.

2. Skapa den nya menyn:

```
Menu menynamn
{
        …
}
```

   Se ”Arbetsytehanterarens menysyntax” på sidan 247.

3. Skapa eller redigera knappkopplingen för att visa den nya menyn.

Om menyn ersätter den befintliga, redigerar du knappkopplingen som visar Arbetsyta-menyn.

```
<Btn3Down> root   f.menu   menynamn
```

Om det är en ytterligare meny skapar du en ny knappkoppling. Följande knappkoppling visar menyn när Skift-musknapp 3 klickas på bakgrunden:

```
Shift<Btn3Down> root   f.menu   menynamn
```

4. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

▼ Så här skapar du en ny Fönster-meny

Obs! – Fönster-menyn är inbyggd i arbetsytehanteraren och anpassas normalt inte. Om du vill att fönstrets metoder ska vara konsekvent mellan olika program, ska du undvika att göra några större ändringar av Fönster-menyn.

1. Öppna rätt fil för redigering:
   - Egen: *Hemkatalog*`/.dt/dtwmrc`
   - Systemtäckande: `/etc/dt/config/`*språk*`/sys.dtwmrc`

   Information om hur du skapar dessa filer finns i ”Arbetsytehanterarens konfigurationsfiler” på sidan 242.

2. Skapa den nya menyn:

```
Menu menynamn
{
        …
}
```

3. Använd resursen `windowMenu` för att ange den nya menyn:

```
Dtwm*windowMenu: menynamn
```

4. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Anpassa knappkopplingar

En *knappkoppling* associerar en bearbetning med musknappen och möjliga styrtangenter till en fönsterhanterarfunktion. Knappkopplingarna gäller alla arbetsytor.

Skrivbordets standardknappkopplingar definieras i arbetsytans konfigurationsfil i en knappkopplingsuppsättning med namnet `DtButtonBindings`:

```
Buttons DtButtonBindings
{
  ...
}
```

Syntax för knappkoppling

Syntaxen för knappkopplingar är:

```
Buttons Knappkopplingsuppsättning
{
    [styrtangent]<knappnamnMusfunktion>  sammanhang  funktion  [argument]
    [styrtangent]<knappnamnMusfunktion>  sammanhang  funktion  [argument]
```

där:

knappnamn
- `Btn1`–Vänster musknapp
- `Btn2`–Mellersta musknappen (mus med tre knappar) eller båda knapparna (mus med två knappar)
- `Btn3`–Höger knapp
- `Btn4`–Knapp 1 och 2 samtidigt på en mus med tre knappar
- `Btn5`–Knapp 2 och 3 samtidigt på en mus med tre knappar

styrtangent
- `Ctrl, Skift, Alt, Lock`

musfunktion
- `Down`—Hålla ned musknappen
- `Up`—Släppa upp musknappen
- `Click`—Trycka och släppa upp musknappen
- `Click2`—Dubbelklicka på musknappen
- `Drag`—Dra med musen och samtidigt hålla ned musknappen

sammanhang Anger var pekaren måste vara för att kopplingen ska fungera. Om det behövs avgränsar du flera sammanhang med tecknet "|".

`root`—Arbetsytefönstret
`window`—Klientfönster eller fönsterram
`frame`–Fönsterram, utan innehållet
`icon`–Ikon
`title`–Namnlist
`app`–Klientfönster, utan ramen

funktion Någon av fönsterhanterarfunktionerna. En lista över funktioner finns i kommandobeskrivningen till `dtwmrc(4)`.

argument Valfritt argument till fönsterhanterarfunktionen som behövs. Mer information finns i kommandobeskrivningen till `dtwmrc(4)`.

Följande rad visar menyn som beskrivs i `DtRootMenu` när tredje musknappen klickas och pekaren finns i arbetsytefönstret (men inte i något klientfönster).

```
<Btn3Down>      root      f.menu      DtRootMenu
```

Obs! – Fullständig information om knappkopplingens syntax finns i kommandobeskrivningen till `dtwmrc(4)`.

▼ Så här lägger du till en knappkoppling

1. Öppna rätt fil för redigering:
   - Egen: *Hemkatalog*`/.dt/dtwmrc`
   - Systemtäckande: `/etc/dt/config/`*språk*`/sys.dtwmrc`

   Information om hur du skapar dessa filer finns i ”Arbetsytehanterarens konfigurationsfiler” på sidan 242.

2. Lägg till knappkopplingen i definitionen `DtButtonBindings`.

   Koppla inte samma knapp till olika funktioner för både klick- och tryckbearbetningar, och koppla inte mer än en funktion till samma knapp och sammanhang.

3. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

▼ Så här skapar du en ny knappkopplingsuppsättning

1. Öppna rätt fil för redigering:
   - Egen: *Hemkatalog*`/.dt/dtwmrc`
   - Systemtäckande: `/etc/dt/config/`*språk*`/sys.dtwmrc`

   Information om hur du skapar dessa filer finns i ”Arbetsytehanterarens konfigurationsfiler” på sidan 242.

2. Skapa den nya knappkopplingsuppsättningen. Se ”Syntax för knappkoppling” på sidan 250.

3. Ange resursen `buttonBindings` till det nya namnet:

   `Dtwm*buttonBindings:` *Knappkopplingsuppsättning*

4. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Obs! – De nya knappkopplingarna ersätter de befintliga. Kopiera alla knappkopplingar du vill ha kvar från `DtButtonBindings`.

Anpassa tangentkopplingar

En *tangentbordskoppling*, som också kallas *tangentkoppling*, associerar en tangentkombination med arbetsytehanterarens funktioner. Tangentkopplingar gäller för alla arbetsytor.

Obs! – Var försiktig med att använda en vanlig tangentkombination som tangentkoppling. Kombinationen Skift-A, t ex, placerar normalt bokstaven "A" i aktuellt fönster. Om du kopplar Skift-A till en funktion åsidosätts den normala användningen.

Skrivbordets standardtangentkopplingar

Skrivbordets standardtangentkopplingar definieras i arbetsytehanterarens konfigurationsfil i en tangentkopplingsuppsättning med namnet `DtKeyBindings`:

```
Keys DtKeyBindings
{
 …
}
```

Syntax för tangentkoppling

Syntaxen för tangentkoppling är:

```
Keys Tangentkopplingsuppsättning
{
    [Styrtangent]<Key>tangentnamn  sammanhang  funktion  [argument]
    [Styrtangent]<Key>tangentnamn  sammanhang  funktion  [argument]
    …
}
```

där:

Styrtangent `Ctrl`, `Skift`, `Alt` och `Lock`. Du kan använda flera styrtangenter, avgränsa dem med mellanslag.

tangentnamn Tangenten som funktionen tilldelas. För tangenter med bokstäver och tal är *tangentnamn* normalt tryckt på tangenten. Namnet på tangenten "a" är "a" och tangenten "2" heter "2". Tangenten "Tab" heter "Tab". Tangenten "F3" heter "F3".

För andra tangenter uttalas namnet – t ex plus för tangenten "+". Filen `keysymdef.h`, som finns i en systemberoende katalog, innehåller ytterligare information om tangentnamn.

sammanhang Elementet som måste vara tangentbordsaktiverat för att funktionen ska fungera. Dessa kan vara ihoplänkade om kopplingen gäller för mer än ett sammanhang. Flera sammanhang avgränsas med tecknet "`|`".

`root`–Arbetsytans bakgrund
`window`–Klientfönster
`icon`–Ikon

funktion En fönsterhanterarfunkton. En lista över giltiga funktioner finns i kommandobeskrivningen till `dtwmrc(4)`.

argument Valfritt argument till fönsterhanterarfunktionen som behövs. Mer information finns i kommandobeskrivningen till `dtwmrc(4)`.

Med följande tangentkoppling kan användarna byta vad som är tangentbordsaktiverat till nästa tillfälliga fönster i programmet genom att trycka på Alt+F6.

```
Alt<Key>F6      window      f.next_key      transient
```

Obs! – Fullständig information om tangentkopplingens syntax finns i kommandobeskrivningen till `dtwmrc(4)`.

▼ Så här skapar du en anpassad tangentkopplingsuppsättning

1. Öppna rätt fil för redigering:
   - Egen: *Hemkatalog*`/.dt/dtwmrc`
   - Systemtäckande: `/etc/dt/config/`*språk*`/sys.dtwmrc`

   Information om hur du skapar dessa filer finns i ”Arbetsytehanterarens konfigurationsfiler” på sidan 242.

1. Skapa en ny tangentkopplingsuppsättning med ett unikt namn på *Tangentkopplingsuppsättning*. Använd skrivbordets standardkopplingsuppsättning, `DtKeyBindings`, som riktlinje.

2. Ange resursen `keyBindings` till det nya namnet:

   `Dtwm*keyBindings:` *Tangentkopplingsuppsättning*

3. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

Obs! – De nya tangentkopplingarna ersätter de befintliga. Kopiera alla tangentkopplingar du vill ha kvar från `DtKeyBindings` till den nya uppsättningen.

Växla mellan standardmetod och anpassad metod

Om du vill växla mellan standardmetoden för OSF/Motif 1.2 och metoden för skrivbordsfönster i CDE:

1. Tryck på Alt+Skift+Ctrl+!

2. Klicka på OK i dialogrutan.

När du växlar till standardmetoden tas frontpanelen och alla anpassade tangent- och knappkopplingar bort.

14

Administrera programresurser, fonter och färger 15

Du kan välja mellan ett flertal färger och fonter för skärmen med hjälp av Arbetsinställningar eller genom att anpassa ytterligare font- och färgresurser. I det här kapitlet beskrivs hur du anpassar fonter och färgresurser.

Här beskrivs också hur du definierar stilen för byggklossprogram av typen `DtEdito`r, som skrivbordets Textredigerare (`dtpad`) och Elektronisk post (`dtmail`), och hur du ändrar snabbval för `DtEditor` som krockar med dessa definitioner.

Ange programresurser	*257*
Definiera UNIX-kopplingar	*258*
Administrera fonter	*263*
Administrera färger	*267*
Ange skuggtjocklek för programfönster	*275*

Ange programresurser

Resurser används av program för att ange vissa aspekter i utseende och metod. I Arbetsinställningar (`dtstyle`) finns t ex resurser med vilka du kan ange var systemet ska söka efter filer som innehåller information om färgpaletter:

```
dtstyle*paletteDirectories: /usr/dt/palettes/C \
Hemkatalog/.dt/palettes
```

Programstandardfiler för skrivbordsprogram finns i katalogen `/usr/dt/app-defaults/`*språk*.

▼ Så här anger du systemtäckande resurser

♦ Lägg till resurserna i filen `/etc/dt/config/`*språk*`/sys.resources`. (Du kanske först måste skapa filen.)

I t ex `/etc/dt/config/C/sys.resources` anger du:

`AnApplication*resource:` *värde*

då anges resursen `AnApplication*resource` i alla användares RESOURCE_MANAGER-egenskap vid nästa inloggning.

▼ Så här anger du egna resurser

1. Lägg till resurserna i filen *Hemkatalog*`/.Xdefaults`.
2. Dubbelklicka på Ladda om resurser i programgruppen Skrivbordsverktyg.

Hur skrivbordet startar resurser

Resurser startas när sessionen startas av sessionshanteraren. Mer information om hur sessionshanteraren laddar resurser till RESOURCE_MANAGER finns i ”Ladda sessionsresurser” på sidan 30.

Definiera UNIX-kopplingar

UNIX-kopplingar är normalt inte aktiverade.

▼ Så här anger du EMACS-Style Translations

Följande procedur anger:

- EMACS stildefinitioner för byggklossar av typen `DtEditor`, t ex skrivbordets Textredigerare och (`dtpad`) och Elektronisk post (`dtmail`).
- Ändrar snabbval för `DtEditor`-byggklossar som krockar med dessa definitioner.

1. Lägg till följande rad i filen *Hemkatalog*/.Xdefaults:

   #include "/usr/dt/app-defaults/*språk*/UNIXbindings"

   där *språk* är värdet på miljövariabeln LANG.

2. Starta om sessionen.

▼ Så här ändrar du EMACS-Style Translations

1. Infoga innehållet i filen /usr/dt/app-defaults/*språk*/UNIXbindings i *Hemkatalog*/.Xdefaults.
2. Redigera kopplingarna i filen .Xdefaults.
3. Starta om sessionen när du är klar.

UNIX-kopplingar i filen i UNIXbindings

Filen /usr/dt/app-defaults/*språk*/UNIXbindings omfattar följande kopplingar:

Obs! – Tangenten Delete tar bort föregående tecken när UNIX-kopplingar aktiveras, och Skift-Delete tar bort nästa tecken.

- I tabell 15-1 listas de dtpad menysnabbval och snabbvalstext som krockar med UNIX-kopplingarna.

tabell 15-1 dtpad

Snabbval för menyer och snabbvalstext	Åsidosätter
Dtpad*fileMenu.print.acceleratorText:	
Dtpad*fileMenu.print.accelerator:	
Dtpad*editMenu.undo.acceleratorText:	Ctrl+_
Dtpad*editMenu.undo.accelerator:	Ctrl<Key>_
Dtpad*editMenu.paste.acceleratorText:	Skift+Insert

tabell 15-1 `dtpad`

Snabbval för menyer och snabbvalstext	Åsidosätter
`Dtpad*editMenu.paste.accelerator:`	Skift<Key>osfInsert
`Dtpad*editMenu.selectAll.acceleratorText:`	Ctrl+/
`Dtpad*editMenu.selectAll.accelerator:`	Ctrl<Key>/

- I tabell 15-2 listas de `dtmail` Meddelandefönster menysnabbval och snabbvalstext som krockar med UNIX-kopplingarna.

tabell 15-2 `dtmail` Meddelandefönster

Snabbval för menyer och snabbvalstext	Åsidosätter
`Dtmail*ComposeDialog*menubar*Edit.Undo.acceleratorText:`	Ctrl+_
`Dtmail*ComposeDialog*menubar*Edit.Undo.accelerator:`	Ctrl<Key>_
`Dtmail*ComposeDialog*menubar*Edit.Paste.acceleratorText:`	Skift+Insert
`Dtmail*ComposeDialog*menubar*Edit.Paste.accelerator:`	Skift<Key>osfInsert
`Dtmail*ComposeDialog*menubar*Edit.Find/Change.acceleratorText:`	Ctrl+S
`Dtmail*ComposeDialog*menubar*Edit.Find/Change.accelerator:`	Ctrl<Key>s

- Följande tabell tillhandahåller meta- och ctrlkopplingar enligt GNU EMACS-format samt några ytterligare kopplingar. När det behövs kan också tangenten Skift användas i kombination med den normala kopplingen för att skapa en motsatt bearbetning. Exempelvis Ctrl+Skift+F flyttar markören bakåt ett tecken eftersom Ctrl+F normalt flyttar den framåt ett tecken.

  Ytterligare kopplingar är:

Ctrl+komma	bakåt ett ord
Ctrl+Skift+komma	framåt ett ord
Ctrl+punkt	framåt ett ord
Ctrl+Skift+punkt	bakåt ett ord
Ctrl+Retur	slutet av filen
Ctrl+Skift+Retur	början av filen

GNU EMACS kopplar delete-previous-character() i stället för delete-next-character() till tangenten Delete. Meta+F är normalt alternativbokstav för Fil-menyn, så kopplingen till ”framåt ett ord”() ignoreras. Använd någon av de andra kopplingarna för ”framåt ett ord” (t ex Ctrl+punkt).

- I tabell 15-3 listas `DtEditor.text`-tabellen

tabell 15-3 `DtEditor.text-tabellen`

Styrtangent	Tangent	Funktion
c ~s	<Key>a:	början på raden()\n\
c s	<Key>a:	slutet på raden()\n\
c ~s	<Key>b:	framåt ett tecken()\n\
c s	<Key>b:	framåt ett tecken()\n\
c ~s	<Key>b:	bakåt ett tecken()\n\
c s	<Key>b:	bakåt ett ord()\n\
m ~s	<Key>b:	bakåt ett ord()\n\
m s	<Key>b:	framåt ett ord()\n\
c ~s	<Key>d:	radera följande tecken()\n\
c s	<Key>d:	radera föregående tecken()\n\
m ~s	<Key>d:	radera följande ord()\n\
m s	<Key>d:	radera föregående ord()\n\
c ~s	<Key>e:	slutet på raden()\n\
c s	<Key>e:	början på raden()\n\
c ~s	<Key>f:	framåt ett tecken()\n\
c s	<Key>f:	bakåt ett tecken()\n\
m ~s	<Key>f:	framåt ett ord()\n\
m s	<Key>f:	bakåt ett ord()\n\
c	<Key>j:	ny rad och indrag()\n\
c ~s	<Key>k:	radera slutet av raden()\n\
c s	<Key>k:	radera början av raden()\n\
c	<Key>l:	uppdatera skärmen()\n\

tabell 15-3 `DtEditor.text-tabellen` *(Continua)*

Styrtangent	Tangent	Funktion
c	<Key>m:	ny rad()\n\
c s	<Key>n:	en process uppåt()\n\
c ~s	<Key>n:	en process nedåt()\n\
c	<Key>o:	ny rad och säkerhetskopia()\n\
c ~s	<Key>p:	en process uppåt()\n\
c s	<Key>p:	en process nedåt()\n\
c ~s	<Key>u:	radera början av raden()\n\
c s	<Key>u:	radera slutet på raden()\n\
c ~s	<Key>v:	nästa sida()\n\
c s	<Key>v:	föregående sida()\n\
m ~s	<Key>v:	föregående sida()\n\
m s	<Key>v:	nästa sida()\n\
c	<Key>w:	radera det valda avsnittet()\n\
c ~s	<Key>y:	återfå det raderade()\n\
m	<Key>]:	ett stycke framåt()\n\
m	<Key>[:	ett stycke bakåt()\n\
c ~s	<Key>komma:	ett ord bakåt()\n\
c s	<Key>komma:	ett ord framåt()\n\
m	<Key>\\<:	början av filen()\n\
c ~s	<Key>punkt:	framåt ett ord()\n\
c s	<Key>punkt:	bakåt ett ord()\n\
m	<Key>\\>:	slutet på filen()\n\
c ~s	<Key>Retur:	slutet på filen()\n\

tabell 15-3 `DtEditor.text-tabellen` *(Continua)*

Styrtangent	Tangent	Funktion
c s	<Key>Retur:	början av filen()\n\
~c ~s ~m ~a	<Key>osfDelete:	radera föreggående tecken()\n\
~c s ~m ~a	<Key>osfDelete:	radera följande tecken()

Administrera fonter

Med dialogrutan Arbetsinställningar - Font kan du välja vilken fontstorlek du vill ha till alla program. Du kan också ange fonter på kommandoraden eller använda resurser för att:

- Ange fontresurser för enstaka program
- Tilldela olika fonter som ska användas av dialogrutan Font

En *font* är det typsnitt + storlek som tecken skrivs och visas med. Skrivbordet omfattar en mängd fonter med olika stil och storlek.

En *bitmönsterfont* skapas från en punktmatris. (Arbetsinställningar konfigurerar normalt bara bitmönsterfonter.) Fonten är begränsad till en fil. Många filer behöver en fullständig uppsättning storlekar, lutningar och tjocklekar.

Fonter anges som värden på resurser och parametrar till kommandon. Namnet X Logical Font Description (XLFD) är metoden som en font efterfrågas med. Systemet hittar den font som bäst motsvarar den angivna beskrivningen.

Ange fontresurser för skrivbordet

I dialogrutan Arbetsinställningar - Font kan du ange fonter (upp till sju storlekar) för t ex textinmatningar och namn.

Resurser som anges i dialogrutan Font

När en font väljs skrivs följande resurser till egenskapen RESOURCE_MANAGER:

- `SystemFont` används för systemområden, t ex menyrader, menyrutor, knappar, växlingsknappar och namn. Följande resurs anges av `SystemFont`:

`*FontList`	Visas i systemområden på skrivbordsklienter och andra klienter som skapats med OSF/Motif-verktyg.

- `UserFont` används för text som skrivs i fönster. Följande resurs anges av `UserFont`:

`*Font`	Stöder tidigare versioner av X-program
`*FontSet`	Huvudinställningen
`*XmText*FontList`	Visas i textinmatningsrutor
`*XmTextField*FontList`	Visas i textinmatningsrutor

Resurser som används i dialogrutan Font

Fonterna som används för varje val i dialogrutan Font anges i resursfilen `/usr/dt/app-defaults/Dtstyle`. Upp till sju storlekar kan anges.

`NumFonts`	Antal fontstorlekar i dialogrutan Font
`SystemFont[1-7]`	Upp till sju resurser som tilldelar en viss font till valet i dialogrutan Font för `SystemFont`
`UserFont[1-7]`	Upp till sju resurser som tilldelar en viss font till valet i dialogrutan Font för `UserFont`

Obs! – Standardfonterna för dessa resurser har valts för läsbarheten på olika skärmar. Om du vill ha en viss font för ett program, anger du den med programmets fontresurs i stället för att ändra skrivbordets fonter.

Mer information om programfonter finns i kommandobeskrivningen till `DtStdAppFontNames` (5) och `DtStdInterfaceFontNames` (5).

▼ Så här listar du tillgängliga fonter

1. Skriv följande:

   `xlsfonts [-`*alternativ*`] [-fn` *mönster*`]`

En lista över XLFD-namn och fontalias som är tillgängliga i systemet visas. Bitmönsterfonter visar värden i alla fjorton XLFD-fält. Skalbara typsnitt visar noll i positionerna för *Bildpunktstorlek*, *Punktstorlek*, *UpplösningX* och *UpplösningY*.

2. Om du letar efter vissa fonter använder du mönstermatchningen för `xlsfonts`. Använd jokertecken för att ersätta den del av mönstret som du inte försöker matcha.

3. Om `xlsfonts` inte visar några fontnamn som börjar på `dt`, innehåller inte fontsökvägen skrivbordsfonter. Skriv följande kommando för att få med skrivbordsfonterna i de tillgängliga fonterna:

   `xset +fp` *katalognamn*

   där *katalognamn* är katalogen som innehåller skrivbordsfonterna. Standardpositionen som anges vid sessionsstarten är `/usr/dt/config/xfonts/`*språk*.

Mer information finns i:

- Kommandobeskrivningen till `xset` och `xlsfonts` som listar tillgängliga alternativ.
- *Using the X Window System* som beskriver fontalias och `xset`-klienten.

▼ Så här anger du fonter på kommandoraden

♦ Använd kommandoradsalternativet `-xrm` för att ange en viss klient. Exempel:

programnamn `-xrm "*bitstream-charter-medium-r-normal-8-88-75-75-p-45-iso8859-1"`

X Logical Font Description (XLFD)

En font anges genom att fjorton olika egenskaper listas, avgränsade med bindestreck (-). Detta kallas X Logical Font Description (XLFD). I vissa fall kan en egenskap i listan ersättas av jokertecknet *, och ett tecken i egenskapen ersättas med jokertecknet ?. I tabell 15-4 listas specifikationer för fontegenskapernas strängar.

Formatet på egenskapssträngarnas specifikationer är:

"*-Upphovsman-Familjenamn-Tjocklek-Lutning-AngeBredd-LäggTillStil-Bildpunktstorlek-Punktstorlek-UpplösningX-UpplösningY-Mellanrum-Snittbredd-TeckenuppsättningRegister-TeckenuppsättningKodning*"

tabell 15-4 Specifikationer till fonternas egenskapssträngar

Egenskapssträng	Definition
Upphovsman	En sträng som anger vem som utformat fonten
Familjenamn	En sträng som anger varumärkesnamnet på fonten
Tjocklek	En sträng som anger relativ tjocklek på fonten, t ex fetstil
Lutning	En kod som beskriver riktning på lutningen R (Roman–ingen lutning) I (Kursiv–höger lutning) O (Oblique–höger lutning) RI (Omvänd kursiv–vänster lutning) RO (Omvänd Oblique–vänster lutning)
AngeBredd	En sträng som beskriver bredd, t ex komprimerad eller utökad
LäggTillStil	En sträng som innehåller ytterligare information som behövs för att göra fonten unik
Bildpunktstorlek	Ett heltal som anger storleken på en s k em-square i bildpunkter
Punktstorlek	Ett heltal som anger storleken på en s k em-square i decipunkter
UpplösningX	Ett heltal som anger vågrät upplösning i bildpunkter
ResolutionY	Ett heltal som anger lodrät upplösning i bildpunkter
Mellanrum	En kod som anger mellanrum mellan enheter *M* (Fast teckenindelning--fast breddsteg) *P* (Proportionellt breddsteg--proportionellt breddsteg) *C* (Teckencell)
Snittbredd	Ett heltal som anger snittbredden i 1/10 bildpunkter
TeckenuppsättningRegister	En sträng som anger hur fontkoden har registrerats
TeckenuppsättningKodning	En sträng som anger teckenuppsättningen i angivet register

Exempel

Följande XLFD-namn beskriver en font med namnet `charter` som skapats av Bitstream och stöder standardkod ISO8859-1:

```
-bitstream-charter-medium-r-normal--8-80-75-75-p-45-iso8859-1
```

Den har normal tjocklek, utan någon speciell lutning och normal bredd. Fonten är proportionell, med en s k em-square box på 8 bildpunkter eller 8,0 punkter. Den vågräta och lodräta upplösningen är båda på 75 bildpunkter. Snittbredden på tecknen är 45 tiondelsbildpunkter eller 4,5 bildpunkter.

Delar av den här strängen kan ersättas med jokertecken. Systemet använder den första fonten som påträffas som matchar de delar du anger.

Om allt du vill ha är en åtta bildpunkters `charter`-font använder du:

```
*-charter-*-*-*-*-8-*
```

Administrera färger

I det här avsnittet beskrivs:

- Hur du anger skärmfärger i Arbetsinställningar
- Resurser som används av Arbetsinställningar för att kontrollera hur skrivbordet använder färger

Färgpaletter

En palett består av en grupp färguppsättningar. Färguppsättningen för aktuell palett visas i dialogrutan Arbetsinställningar - Färg.

För varje palett finns en fil. Resursen `paletteDirectories` anger vilka kataloger som innehåller palettfiler. Normalt innehåller den här resursen:

- Inbyggda paletter: `/usr/dt/palettes`
- Systemtäckande paletter: `/etc/dt/palettes`
- Egna paletter: *Hemkatalog*`/.dt/palettes`

Färguppsättningar

Varje färguppsättning i aktuell palett representeras av en färgknapp i dialogrutan Arbetsinställningar - Färg. Varje färg identifieras med ett färguppsättnings-ID, som är ett tal från 1 till 8.



Figure 15-1 Värden på färguppsättnings-ID för HIGH_COLOR

Varje färguppsättning utgörs av upp till fem färger. Varje färgknapp visar bakgrundsfärgen på färguppsättningen. De fem färgerna i varje färguppsättning representerar följande komponenter:

`foreground`	Förgunden till ett programfönster eller fönsterram. Den är alltid svart eller vit. Normalt används den för text i fönster och rubriker.
`background`	Bakgrunden till programfönstret eller fönsterramen.
`topShadowColor`	Färgen på den övre och vänstra avfasningen för programikoner (t ex knappar) och fönsterramar.
`bottomShadowColor`	Färgen på den undre och högra avfasningen för programikoner och fönsterramar.
`selectColor`	Färgen som anger aktiv status på vissa ikoner, t ex aktiva växlingsikoner och knappar.

Antalet färguppsättningar som används av varje palett bestäms av resursen `colorUse`, som användaren anger i dialogrutan Arbetsinställningar - Ange antal färger.

Ange färgvärden

Arbetsinställningar använder RGB-värden när information skrivs till palettfilerna. Syntaxen för RGB-tal är:

#RödGrönBlå

Röd, *Grön* och *Blå* är hexadecimala tal, vart och ett 1 till 4 siffror långt, som anger vilken mängd färg som ska användas. Du måste ange samma antal siffror för var och en av färgerna. Giltiga färgvärden består därför av 3, 6, 9 eller 12 hexadecimala siffror.

Vit kan t ex anges på något av följande sätt:

```
#fff
#ffffff
#fffffffff
#fffffffffffff
```

Om du anger färgresurser direkt använder du antingen färgnamnet eller RGB-värdet. I filen `/usr/lib/X11/rgb.txt` listas alla namngivna färger.

Hur färguppsättningar tilldelas resurser

Skrivbordet kopplar färguppsättningar till olika skärmelement via resurser, och gör följande tilldelningar:

Resurs	Skärmelement
`activeColorSetId`	Aktivt fönsters ramfärg
`inactiveColorSetId`	Inaktivt fönsters ramfärg
`textColorSetId`	Textinmatningsområden
`primaryColorSetId`	Områden i programmets huvudbakgrund
`secondaryColorSetId`	Programmets menyrad, menyer och dialogrutor

Dessa resurser använder ett färguppsättnings-ID som värde. Genom att skärmelement anges med färguppsättnings-ID, kan elementet dynamiskt ändras till det nya färgschemat när en ny palett markeras i Arbetsinställningar.

Du använder de här resurserna för enstaka program. Följande rad visar hur du visuellt grupperar alla `dtterm`-fönster genom att använda färguppsättning 8 för huvudfärgen.

```
dtterm*primaryColorSetId:   8
```

Standardtilldelningar av färguppsättningar

Färguppsättnings-ID som används till skärmelement är beroende av inställningen Ange antal färger i Arbetsinställningar:

- Maximalt antal färger (8 färguppsättningar)—inställningen Fler färger för skrivbordet:

Färg-ID	Skärmelement
1	Aktivt fönsters ramfärg
2	Inaktivt fönsters ramfärg
3	Används inte (standardvärde)
4	Textinmatningsområden
5	Områden i programmets huvudbakgrund
6	Programmets menyrad, menyer och dialogrutor
7	Används normalt inte
8	Frontpanelens bakgrund

- Normalt antal färger (4 färguppsättningar)—inställningen Fler färger för program:

Färg-ID	Skärmelement
1	Aktivt fönsters ramfärg
2	Inaktivt fönsters ramfärg
3	Färg på programmets och frontpanelens bakgrund
4	Textinmatningsområden

- Minimalt antal färger (2 färguppsättningar)—inställningen Flest färger för program.

Färg-ID	Skärmelement
1	Ram på aktivt fönster, arbetsytans markeringsknappar

2 Alla andra skärmelement

Kontrollera färg med Arbetsinställningar

Du kan dynamiskt ändra färg på skrivbordsprogram och andra samverkande program via Arbetsinställningar. De förgrunds- och bakgrundsfärger som anges med Arbetsinställningar är även tillgängliga för icke samverkande program.

- För att klienten ska reagera på färgändringar i Arbetsinställningar, måste den använda Motif. Klienter som skrivs med andra verktyg kan inte ändra färg dynamiskt efter ändringar i Arbetsinställningar. Färgändringar för dessa klienter träder inte i kraft förrän klienten startas om.
- Det får inte finnas några andra specifika färgresurser för klienten. Detta omfattar användarspecificerade resurser, programstandardfiler och resurser inbyggda i programmet.
- Klienter kan ange resurserna `primaryColorSetId` och `secondaryColorSetId` för att använda vissa färger i en skrivbordspalett.

Antal färger som används i Arbetsinställningar

Antalet färger som används i Arbetsinställningar är beroende av värdena på följande resurser:

`colorUse`	Konfigurerar antalet färger som används av skrivbordet
`shadowPixmaps`	Dirigerar skrivbordet att ersätta två skuggfärger med bildpunktsmönster
`foregroundColor`	Anger om förgrundsfärgen ändras automatiskt.
`dynamicColor`	Kontrollerar om program ändrar färg när du byter palett.

I tabell 15-5 listas maximalt antal färger som allokeras av skrivbordet.

tabell 15-5 Antal skrivbordsfärger

Skärm	Maximalt antal färger	Antalet härleds från
B_W	2	Svart och vitt
LOW_COLOR	12	Två färguppsättningar gånger fem färger plus svart och vitt
MEDIUM_COLOR	22	Fyra färguppsättningar gånger fem färger plus svart och vitt
HIGH_COLOR	42	Åtta färguppsättningar gånger fem färger plus svart och vitt

Så här bestämmar du maximalt antal färger:

1. Multiplicera antalet färguppsättningar i paletten med antalet färger i varje färguppsättning.
2. Lägg till två (för svart och vitt).

Med följande konfiguration har du bara tio färger i paletten: fyra färguppsättningar gånger två färger i varje uppsättning (`background` och `selectColor`) samt svart och vitt:

```
*colorUse:MEDIUM_COLOR
*shadowPixmaps: True
*foregroundColor:White
```

Obs! – Flerfärgsikoner använder ytterligare fjorton färger.

Resursen colorUse

Standardvärdet på resursen `colorUse` är MEDIUM_COLOR. Värdet på den här resursen påverkar antalet färger som används i paletten. Andra resurser påverkar antalet färger som används för att skapa skuggor. Värdet på resursen `colorUse` påverkar också användningen av flerfärgsikoner.

Värde	Beskrivning
B_W	Inställningen ”Svartvitt” i Arbetsinställningar Skärmar med 1 till 3 färgskikt Antal färguppsättningar: 2 Maximalt antal färger: 2 Standardantal färger: 2 Inga flerfärgsikoner
LOW_COLOR	Inställningen ”Flest färger för program” i Arbetsinställningar Skärmar med 4 till 5 färgskikt Antal färguppsättningar: 2 Maximalt antal färger: 12 Standardantal färger: 12 Inga flerfärgsikoner
MEDIUM_COLOR	Inställningen ”Flest färger för program” i Arbetsinställningar Skärmar med 6 färgskikt Antal färguppsättningar: 4 Maximalt antal färger: 22 Standard antal färger: 22 Flerfärgsikoner
HIGH_COLOR	Inställningen ”Flest färger för program” i Arbetsinställningar Skärmar med 7 eller fler färgskikt Antal färguppsättningar: 8 Maximalt antal färger: 42 Standardantal färger: 42 Flerfärgsikoner\
standardvärde	Skrivbordet väljer korrekt värde för skärmen. (För att minska antalet färger som används av skrivbordet för skärmar med många färger, anges standardresursen `colorUse` till MEDIUM_COLOR.)

Resursen shadowPixmaps

Resursen `shadowPixmaps` dirigerar skrivbordet att ersätta de två skuggfärgerna med bildpunktsmönster. Dessa bildpunktsmönster blandar bakgrundsfärgen med svart eller vitt för att simulera övre och undre skugga. Detta minskar antalet färger som behövs med två, eftersom färgceller inte måste allokeras för skuggfärgerna.

Värde	Beskrivning
True	Skrivbordet skapar en `topShadowPixmap` och `bottomShadowPixmap` och använder dem i stället för skuggfärger.
False	`topShadowColor` och `bottomShadowColor` från paletten används.

Standardvärdet på `shadowPixmaps` är beroende av vilken `colorUse`-resurs du har och maskinvarustödet för skärmen.

Resursen foregroundColor

Resursen `foregroundColor` anger hur förgrunden konfigureras i en palett.

Inställning	Resultat
White	Förgrunden anges till vit.
Black	Förgrunden anges till svart.
Dynamic	(Standardvärde) Förgrunden anges dynamiskt till svart eller vitt, beroende på värdet på `background`. Vit text på gul bakgrund är t ex svårläst, så systemet väljer svart.

Om `foregroundColor` anges till Black eller White, minskas antalet färger i färguppsättningen med en, och förgrunden ändras inte efter ändringarna i bakgrundsfärgen.

Standardvärdet på `foregroundColor` är Dynamic, utom när värdet på `colorUse` är B_W.

Resursen dynamicColor

Resursen `dynamicColor` kontrollerar om program ändrar färger dynamiskt, dvs om klienterna ändrar färg när du byter paletter.

Värde	Beskrivning
True	Klienter ändrar färg dynamiskt när en ny palett markeras. Detta är standardvärdet.
False	Klienter ändrar inte färg dynamiskt. När en ny palett markeras, använder klienterna de nya färgerna när sessionen startas om.

När värdet på resursen `dynamicColor` är True, kan klienter som inte ändrar färger dynamiskt (icke-Motif-program) allokera andra celler i färgkartan än klienter som ändrar färg dynamiskt, även om samma färg visas.

Obs! – Eftersom alla klienter kan dela samma färgceller, minskas antalet färger som skrivbordet använder om resursen `dynamicColor` anges till False.

Ange skuggtjocklek för programfönster

Skrivbordet definierar standardtjockleken för skugga till en bildpunkt för komponenterna i programfönstret, t ex knappskuggor och fokusmarkering. Motif 1.2-program använder det här resursvärdet, medan andra program inte kan det och därför visar skärmen annorlunda.

Om du vill ange skuggans tjocklek till en bildpunkt för icke-Motif 1.2-program gör du så här:

1. Logga in som rot.
2. Skapa filen `/etc/dt/config/`*`språk`*`/sys.resources`.
3. Ange den programspecifika resursen i `/etc/dt/config/`*`språk`*`/sys.resources` så här:

   `programklass``*XmCascadeButton*shadowThickness: 1`

Mer information om hur du åsidosätter systemets standardresurser och anger ytterligare resurser för alla skrivbordsanvändare finns i ”Ladda sessionsresurser” på sidan 30.

15

Konfigurera lokalanpassade skrivbordssessioner

16

För att konfigurera lokalanpassade skrivbordssessioner måste du:

- Ange miljövariabeln LANG och andra miljövariabler för National Language Support (NLS)
- Använda språkberoende meddelandekataloger och resursfiler
- Exekvera fjärrprogram via internationella system

Hantera miljövariabeln LANG	*278*
Söka efter fonter	*280*
Lokalanpassa resursfilerna app-defaults	*281*
Lokalanpassa funktioner och datatyper	*281*
Lokalanpassa ikoner och bitmönster	*282*
Lokalanpassa hjälpvolymer	*283*
Lokalanpassa meddelandekataloger	*284*
Exekvera lokalanpassade fjärranslutna skrivbordsprogram	*284*
Återställa tangentbordstabellen	*284*

16

Hantera miljövariabeln LANG

Miljövariabeln LANG måste anges att använda operativsystemets språkkänsliga rutiner. Skrivbordet stöder:

- Västeuropeiska språk som skrivs med latinska bokstäver
- Japanska
- Traditionell kinesiska
- Förenklad kinesiska
- Koreanska

Obs! – Stöd för andra språk kan ha lagts till av skrivbordsleverantören.

Du kan ange LANG till ett valfritt värde som stöds av operativsystemet. Alternativ-menyn på inloggningsbilden visar en lista över språk och områden som stöds.

Det finns fyra sätt att ange LANG för skrivbordet:

- Genom att redigera en resurs i filen `Xconfig`
- Använda Alternativ-menyn på inloggningsbilden
- Genom att skapa ett exekverbart `sh`- eller `ksh` `Xsession.d`-skript. (Mer information om hur du använder skriptet Xsession.d finns i ”Läsa in och köra skriptet Xsession.d” på sidan 26.)
- Genom att redigera användarens `.dtprofile`-fil

När LANG anges använder skrivbordet följande språkberoende filer för att bestämma det lokalanpassade gränssnittet.

Färger	`/usr/dt/palettes/desc.`*språk*
Bakgrunder	`/usr/dt/backdrops/desc.`*språk*

Ange språk för flera användare

Om du anger språk med hjälp av en `Xconfig`-fil, lokalanpassas inloggningsbilden och LANG anges för alla användare. Det här är det enda sättet att ändra LANG för alla skärmar i flerskärmssystem. (För att ändra `Xconfig` kopierar du `/usr/dt/config/Xconfig` till `/etc/dt/config/Xconfig.`)

Språket anges genom att du placerar följande rad i `/etc/dt/config/Xconfig`:

`dtlogin.`*värddatorskärm*`.language:` *språk*

Följande rad anger LANG till `Swedish_locale` på skärmen `my_host:0`.

```
dtlogin.my_host_0.language: Swedish_locale
```

Klienten `dtlogin` läser lämplig meddelandekatalog för språket och hämtar den lokalnpassade inloggningsbilden. Klienten `dtlogin` bestämmer sedan listan över språk med följande resurs i resursfilen `/etc/dt/config/Xresources`:

- `dtlogin*language`
- `dtlogin*languageList`
- `dtlogin*languageName`

Filen `Xconfig` kanske måste ange miljövariabeln NLSPATH för det valda språket. Om så inte är fallet, eller om du själv vill ange NLSPATH, läser du ”Miljövariabeln NLSPATH” på sidan 280.

Ange språk för en session

Om du vill ange språk för en session använder du Alternativ-menyn på inloggningsbilden. Inloggningsbilden lokalanpassas och LANG anges för användaren. LANG återgår till standardvärdet (anges i `dtlogin`) när sessionen avslutas.

Ange språk för en användare

En användare åsidosätter inloggningens LANG-inställning i filen *Hemkatalog*`/.dtprofile`. Inloggningsbilden lokalanpassas inte och LANG anges inte för användaren.

- Om du använder `sh` eller `ksh`:

  ```
  LANG=språk
  export LANG
  ```

- Om du använder `csh`:

  ```
  setenv LANG språk
  ```

Miljövariabeln LANG och sessionskonfiguration

Miljövariabeln LANG ändrar katalognamnet som söks för sessionens konfigurationsfiler.

Den lokalanpassade sessionens konfigurationsfiler är:

- `/usr/dt/config/`*språk*`/Xresources` (inloggningshanterarens resursfil)
- `/usr/dt/config/`*språk*`/sys.font` (sessionshanterarens resursfil)
- `/usr/dt/config/`*språk*`/sys.resources` (sessionshanterarens resursfil)
- `/usr/dt/config/`*språk*`/sys.session` (sessionshanterarens exekverbara skal)
- `/usr/dt/config/`*språk*`/sys.dtwmrc` (fönsterhanterarens resursfil)
- `/usr/dt/appconfig/types/`*språk*`/dtwm.fp` (fönsterhanterarens frontpanel)

Ange andra NLS-miljövariabler

Förutom LANG finns det andra NLS-miljövariabler, t ex LC_CTYPE och LC_ALL. Dessa variabler påverkas inte av språkresursen `dtlogin` eller av inloggningsbildens Alternativ-meny. De måste anges i följande filer:

- Systemtäckande variabler: `/etc/dt/config/Xsession.d`
- Egna variabler: *Hemkatalog*`/.dtprofile`

Miljövariabeln NLSPATH

Miljövariabeln NLSPATH bestämmer katalogsökväg där programmen söker efter meddelandekataloger. Både LANG och NLSPATH måste anges till att använda dessa meddelandekataloger. I ”Lokalanpassa meddelandekataloger” på sidan 284 beskrivs adresserna för lokalanpassade meddelanden. De flesta skrivbordsklienter lägger till NLSPATH till sökvägen vid start.

Söka efter fonter

Skrivbordets fonter finns i katalogen `/usr/lib/X11/fonts`. Varje katalog innehåller en katalogfil, `fonts.dir`, och en aliasfil, `fonts.alias`. Information om hur du skapar filerna `fonts.dir` och `fonts.alias` finns i kommandobeskrivningen till `mkfontdir`.

Om du vill lista alla fonter som är tillgängliga för en server, använder du kommandot `xlsfonts`. Om du vill lägga till och ta bort fonter för en server använder du kommandot `xset`.

Lokalanpassa resursfilerna app-defaults

Standardadressen för skrivbordklientens `app-defaults`-fil är `/usr/dt/app-defaults/`*språk*. Om LANG anges till `Swedish_locale` söker programmen efter sina `app-defaults`-filer i `/usr/dt/app-defaults/Swedish_locale`. Om LANG inte anges, ignoreras *språk* och programmen letar efter `app-defaults`-filen i `/usr/app-defaults/C`.

Om du vill ändra adress för `app-defaults` använder du miljövariabeln XFILESEARCHPATH. För att t ex flytta `app-defaults` till `/users`, anger du XFILESEARCHPATH till `/usr/app-defaults/`*språk/klassnamn*.

Om du anger XFILESEARCHPATH i *Hemkatalog*`/.dtprofile`, gäller värdet för alla skrivbord och X-klienter du kör. Icke-klienter hittar inte sina resursfiler såvida du inte länkar eller kopierar dem till katalogen som anges av XFILESEARCHPATH.

Lokalanpassa funktioner och datatyper

Obs! – Om du vill anpassa en fil i katalogen `/usr/dt/appconfig` kopierar du den till katalogen `/etc/dt/appconfig` innan du anpassar den.

Sökvägen för funktions- och datatypsdefinitionsfiler omfattar språkberoende kataloger i:

- Egna: *Hemkatalog*`/dt/types`
- Systemtäckande: `/etc/dt/appconfig/types/`*språk*
- Inbyggda: `/usr/dt/appconfig/types/`*språk*

Sökvägen för programhanterarens konfigurationsfiler är:

- Egna: *Hemkatalog*/dt/appmanager
- Systemtäckande: /etc/dt/appconfig/appmanager/*språk*
- Inbyggda: /usr/dt/appconfig/appmanager/*språk*

Fil- och katalognamn i den här katalogen är lokalanpassade.

Lokalanpassa ikoner och bitmönster

Om du vill lokaanpassa en ikon redigerar du den i ikonredigeraren och sparar den i:

/etc/dt/appconfig/icons/*språk*

Om du sparar den i en annan katalog anger du miljövariabeln XMICONSEARCHPATH att omfatta katalogen där du sparar ikonen. Miljövariabeln XMICONBMSEARCHPATH kontrollerar sökvägen som används för att söka efter ikoner.

Lokalanpassa bakgrundsnamn

Du lokalanpassar bakgrundsnamn med beskrivningsfiler (desc.*språk* och desc.backdrops). Inga speciella lokalanpassade kataloger finns (som /usr/dt/backdrops/*språk*) för bakgrundsfiler. Alla språk använder samma uppsättning bakgrundsfiler men har sina egna desc.*språk*-filer som innehåller de översatta namnen på bakgrunderna.

Beskrivningsfilen innehåller resursspecifikationer för de översatta bakgrundsnamnen. Exempel:

```
Backdrops*Corduroy.desc:  Velours
Backdrops*DarkPaper.desc: PapierKraft
Backdrops*Foreground.desc:AvantPlan
```

Filen desc.*språk* används för att hämta beskrivningsfilen för det lokala *språket* och visa bakgunden i Arbetsinställningar. Om det finns någon beskrivningsspecifikation visas den i bakgrundslistan i Arbetsinställningar. I annat fall används bakgrundens filnamn.

Användarna kan lägga till egna bakgrundsbeskrivningar i filen *Hemkatalog*/.dt/backdrops/desc.backdrops. Den här filen används för att hämta bakgrundsbeskrivningar för alla bakgrunder som användaren har lagt till, oavsett språk.

Sökvägen för filen `description` är:

- Egen: *Hemkatalog*`/.dt/backdrops/desc.backdrops`
- Systemtäckande: `/etc/dt/backdrops/desc.`*språk*
- Inbyggd: `/usr/dt/backdrops/desc.`*språk*

Lokalanpassa palettnamn

Du lokalanpassar palettnamn med beskrivningsfilerna (`desc.`*språk* och `desc.palettes`). Ingen speciell lokalanpassad katalog finns (som `/usr/dt/palettes/`*språk*). Alla språk använder samma uppsättning palettfiler, men har sina egna `desc.palettes`-filer som innehåller de översatta palettnamnen.

Beskrivningsfilen innehåller resursspecifikationer för de översatta palettnamnen. Exempel:

```
Palettes*Cardamon.desc:    Cardamone
Palettes*Cinnamon.desc:    Cannelle
Palettes*Clove.desc:       Brun
```

Filen `desc.`*språk* används för att hämta beskrivningen av paletterna för det lokala *språket* och visa paletten i listan i Arbetsinställningar. Om det finns en beskrivningsspecifikation visas den i palettlistan i Arbetsinställningar. I annat fall används palettens filnamn.

Användarna kan lägga till egna palettbeskrivningar i filen *Hemkatalog*`/.dt/palettes/desc.palettes`. Den här filen används för att hämta palettbeskrivningar för alla paletter som användaren har lagt till, oavsett språk.

Sökvägen för beskrivningsfilerna är:

- Egna: *Hemkatalog*`/.dt/palettes/desc.palettes`
- Systemtäckande: `/etc/dt/palettes/desc.`*språk*
- Inbyggda: `/usr/dt/palettes/desc.`*språk*

Lokalanpassa hjälpvolymer

Om du har lokalanpassat en hjälpvolym måste du lagra den i någon av följande kataloger. Den första hjälpvolymen som påträffas används. Katalogerna söks i följande ordning:

- Egna: *Hemkatalog*`/.dt/help`
- Systemtäckande: `/etc/dt/appconfig/help/`*språk*
- Inbyggda: `/usr/dt/appconfig/help/`*språk*

16

Lokalanpassa meddelandekataloger

Om du har lokalanpassat en meddelandekatalog lagrar du den i följande katalog:

`/usr/dt/lib/nls/msg/`*språk*.

Dessa kataloger innehåller `*.cat`-filer.

Exekvera lokalanpassade fjärranslutna skrivbordsprogram

Du kan anropa lokalanpassade skrivbordsprogram på alla värddatorer för fjärrexekvering som har en liknande lokalnpassad skrivbordsinstallation. Värdena på de NLS-relaterade miljövariablerna på värddatorn som anropar programmet skickas till fjärrvärddatorn när programmet startas. Miljövariablerna innehåller dock inte någon värddatorinformation.

Återställa tangentbordstabellen

Om tecken ser konstiga ut och oväntade saker händer på skärmen, eller om tecken inte kan visas eller skrivas, måste du kanske återställa eller installera om tangentbordstabellen eller ändra inmatningsmetoden.

Inmatningsmetoden bestäms av miljövariabeln LC_CTYPE, LANG eller LC_ALL, eller språket som anges med alternativet `-lang`.

Anta att användaren vill öppna en terminal med språket för C i ett POSIX-skal, t ex:

```
LANG=C   dtterm
```

Den här nya terminalen använder språket för C inklusive inmatningsmetoden och fonterna för C. Om du använder ett landsspecifikt tangentbord, kanske inmatningsmetoden inte accepterar utökade tecken. När språkmiljön C används med ett landsspecifikt tangentbord, måste användarna ange miljövariabeln LC_CTYPE (eller LANG eller LC_ALL) till ett lämpligt värde innan terminalen anropas.

Om du vill använda språkmiljön C med ett tyskt tangentbord skriver du:

```
LANG=C LC_CTYPE=DeDE dtterm
```

Om X-servern har återställts och keymaps har initierats, återställer du lämplig tangentbordstabell på servern med kommandot `xmodmap`.

16

16

Index

Symbols

%B, 116
%DatabaseHost%, 178
%DisplayHost%, 178
%H, 115
%L, 116
%LocalHost%, 178
%M, 116
%SessionHost%, 178
*, jokertecken, 195
/ / / fileame, databasserver, 94
/ / / foreign skärmtyp, 5
?, jokertecken, 195

A

actionIcon, resurs, 163
ACTIONS, fält, 193
activeColorSetId, resurs, 269
aktiv session, 24
ALTERNATE_ICON, fält, 232
AND, operator i fältet MODE, 198
ANIMATION, definition, 234
animering för frontpanelen, 234
anslutningar,köra program via, 105
anslutningspunkt för fjärrfiler, 95
användare, ändra för funktion, 182
användar-ID, 90
app-defaults
  språkberoende, 281
Arbetsinställningar
  använda för att ange färger, 269
  integrera färg med, 59
  integrera fonter med, 58
Arbetsyta-menyn
  definition, 246
  skapa, 248
  syntax, 247
Arbetsytehanterare
  starta, 25
Arbetsytehanteraren
  starta, 32
arbetsytehanteraren
  ändra, 248
  ändra till OSF / Motif, 254
  definition, 241
  egna anpassningar, 243
  funktion, 247
  hantera frontpanelen, 242
  inkludera andra filer, 243
  knappkopplingar, 249
  konfigurationsfiler, 242
  menyer, 246
  starta om, 244

systemtäckande anpassning, 243
arbetsytor
ändra standardantal, 236
antal, 244
bakgrunder, Se bakgrunder
namn, 245
arbetytor
anpassa, 244
Arg_1, syntax, 133
ARG_CLASS, fält, 165
ARG_COUNT, fält, 165, 176
ARG_MODE, fält, 165
Arg_n, syntax, 168
ARG_TYPE, fält, 165, 176, 193
skriva ut, 112
argument
antal för funktioner, 176
flera för funktioner, 171
för funktioner, 167
fråga efter, 169
icke utbytbara, för funktioner, 172
icke-fil, 170
till funktioner, 133
utbytbara för funktioner, 172
Återställ, funktion på frontpanelen, 215
authorize, resurs, 16
autoanslutning, 95

B

background, resurs, 268
bakgrunder, 242
använda grafiska bilder, 246
fillägen, 245
lägga till, 245
basfilnamn, 164, 192
behörighetsmönster, anges med Skapa funktion, 149
bilder, Se ikoner
bildpunktsmönster
hur filer hittas, 204, 206
namngivningskonventioner, 205
sökväg, 206
bitmönster
Se även ikoner
hur filer hittas, 204, 206
namngivningskonventioner, 205
sökväg, 206
bitmönsterskärm, köra inloggningshanteraren utan, 5
blank, ikontyp, 229
.bm, filtypstillägg, 204
bottomShadowColor, resurs, 268
BOX, definition, 216
syntax, 217
BROADCAST, används i XDMCP indirekt, 9
busy, ikontyp, 229
buttonBindings, resurs, 252

C

CDE-MIN, filer, 94
CDE-TT, filer, 94
CHOOSER, sträng
definition, 9
identifiera, 9
client, ikontyp, 229
CLIENT_GEOMETRY, fält, 233
CLIENT_NAME, fält, 232, 233
clock, ikontyp, 229
color sets, 268
colorUse, resurs, 31, 271, 273
COMMAND, funktion, 156
exekveringssträng, 167
exempel, 159
obligatoriska fält, 167
CONTAINER_NAME, fält, 217, 218, 219, 222
CONTAINER_TYPE, fält, 218
CONTENT, fält, 195, 199
CONTROL, definition, syntax, 217
CONTROL_BEHAVIOR, fält, 238
COPY_TO_ACTION, fält, 194
cpp, programsatser, 34

current.old, katalog, 37

D

DATA_ATTRIBUTES
definiera, 192
definition, 188
syntax, 189
DATA_CRITERIA
definiera, 195
definition, 188
flera, 200
syntax, 190
tillsammans med DATA_ATTRIBUTES, 188
databas
starta om, 161
starta om funktioner, 161
DataBaseHost, nyckelord, 178
databasservrar, 103
klient till, 102
konfigurera, 93, 101
skapa, 101
databassökväg, 120, 157
Se även sökväg
assemblera, 121
förhållande till programsökväg, 119, 120
inverkan på EXEC_HOST, 103
miljövariabler, 120
standard-, 120
syntax, 121
databasvärd, 103, 178
datatyper
associera ikoner med, 207
associera med funktioner, 193
attribut, 192
begränsa funktioner baserat på, 176
begränsning av Skapa funktion, 143
behörighetsmönster, 149, 150
definiera, 189
dolda, 194
exekverbara, 198
exempel, 190
flera villkor, 200
förhållande till funktioner, 136
hjälp med, 192
ikoner för, 67, 153, 192
innehållsbaserade, 149, 199
integrera med dtappintegrate, 76
introduktion, 127, 135
kategorisera villkor, 195
konfigurationsfiler, 189
krav för att skapa manuellt, 188
lägesvillkor, 198
lokaliserade, 201
metod för att släppa, 138
metod för dubbelklickning, 138
namnbaserade, 149, 195
nödvändiga för att skapa manuellt, 143
nödvändiga för registreringen, 62
Reload Actions, 161
servrar för, 101
skapa manuellt, 187
skapa med Skapa funktion, 135, 146
skilja mellan, 195
skriva ut, 139
skrivskyddade, 198
sökväg, Se databassökväg
sökvägsbaserade, 196
starta om, 161
syfte för att skapa, 55
variabler i definitioner, 179
villkor, 195
Datatyper, lista i Skapa funktion, 147
date, ikontyp, 229
DELETE, fält, 215
DESCRIPTION, fält, 163, 192
Desktop_Tools, programgrupp, ändra, 46
desktop-smart program, 44
DISPLAY, variabel, anges av inloggningshanteraren, 18
DISPLAY_skärmnamn, makro, 34
DisplayHost, nyckelord, 178
dölja filer med datatyper, 194
DROP_ACTION, fält, 231
DROP_ANIMATION, fält, 235
dt, filer, 157

dtaction
    använda för att ändra användare, 182
    syntax, 181
dtappgather, 28, 41
dtappintegrate, 75
    funktionalitet, 76
    syntax, 75
    ta bort program, 48
DTAPPSEARCHPATH, variabel
    assemblera, 117, 118
    definition, 115
DtButtonBindings, 250
dtchooser, fil, 21
dtconfig, kommando, 10
dtCreate, Se Skapa funktion
DTDATABASESEARCHPATH, variabel
    använda, 165
    assemblera, 121
    definition, 115
DtEditor, stildefinitioner för, 258
Dterrors, fil, 10
dtgreet, fil, 21
DTHELPSEARCHPATH, variabel
    assemblera, 124
    definition, 115
Dtlogin*language, resurs, 16
dtlogin, Se Inloggningshanteraren
DTMOUNTPOINT, variabel
    ange, 95
    hämtas från användare, 96
    processer som använder, 95
    processer som behöver, 96
Dtpid, fil, 4
.dtprofile, fil
    / / /sourcing i, 25
    ange miljövariabler i, 33
    ange LANG, 279
    skapa, 26
    läsa in och köra, 26
    syntax, 33
DtRootMenu, 248
dtsearchpath, 27, 115, 165
dtsmcmd, kommando, 35
DTSOURCEPROFILE, variabel, 29
dtspcd, 94, 96, 97
    konfigurera, 97
    verifieringskatalog, 91, 97
DTSPSYSAPPHOSTS, variabel, 100
    ändra, 47
    definition, 115
    syntax, 117
DTSPSYSDATABASEHOSTS, variabel
    definition, 115, 120
    effekt på fältet EXEC_HOST, 103
    syntax, 121
DTSPSYSHELP, variabel, 124
    definition, 115
    syntax, 124
DTSPSYSICON, variabel
    definition, 115
    syntax, 122
DTSPUSERAPPHOSTS, variabel, 100
    ändra, 48
    definition, 115
    syntax, 117
DTSPUSERDATABASEHOSTS, variabel
    definition, 115, 120
    syntax, 121
DTSPUSERHELP, variabel
    definition, 115
    syntax, 124
DTSPUSERICON, variabel
    definition, 115
    syntax, 122
dtstart_appgather, variabel, 28
dtstart_searchpath, variabel, 28
dtstart_ttsession, variabel, 29
dtwm, Se arbetsytehanteraren
dtwm.fp, fil, 212
dtwmfp.session, fil, 214
dtwmrc, fil, 242
    redigera, 243
dynamicColor, resurs, 31, 271, 274

E

Edit Dtwmrc, funktion, 243
egen programgrupp, 46
egna datatyper och funktioner, skapa, 190
elektronisk post, konfigurera, 92
EMACS definitioner, 258
EMACS stildefinitioner, 258
EXEC_HOST
    påverkas av databassökvägen, 103
EXEC_HOST, fält, 178
    flera värden, 103
    standaradvärde, 179
EXEC_HOST, Se värddator för exekvering
EXEC_STRING, fält, Se exekveringssträng
exekverbara filer, datatypsvillkor, 198
exekveringssträng, 167
    absolut sökväg i, 168
    allmänna egenskaper, 167
    ange exekverbara filer, 168
    flerfilsargument, 171
    fråga efter fil, 169
    fråga efter sträng, 170
    inga argument, 168
    skalsyntax, 167
    släppta filer, 168

F

färg
    ange förgrund, 274
    använda i ikoner, 210
    färguppsättningar, 268
    kontrollera, med resursen
        dynamicColor, 274
    skapa skuggor med resurserna
        shadowPixmaps, 274
färger
    administrera, 267
    antal som används, 271
    färguppsättningar, 268
    kontrollera, med
        Arbetsinställningar, 271
    maximalt antal allokerade, 272
    paletter, 267
    programfönster, 269
    ram på aktivt fönster, 269
    ram på inaktivt fönster, 269
    resurser, 268
    standard-, 270
    textinmatningsområden, 269
    värden, 269
färghanterare, 25
färghanteraren
    resurser, 31
    starta, 31
färgpaletter, 267
färgresurser,ändra för registrering, 59
färguppsättningar, 267
    standardvärde, 270
    tilldela för skärmelement, 269
fil, datatypsvillkor, 198
filargument
    ange med Skapa funktion, 146
    används i funktioner, 133
fildelning, 91
file, ikontyp, 229, 232
FILE_NAME, fält, 230, 232
filer
    ansluta, 91
    anslutningspunkt, 95
    åtkomst till distribuerade, 91
    dölja baserat på datatyp, 194
    fjärråtkomst, 91
    fjärrdata, 94
    konsekventa namn, 92
    nödvändiga för nätverk, 94
Filhanteraren, använda som
        ikonfönster, 209
filnamn, ikon, 204
filnamnstilldelning, 95
filservrar, 87
filtyper, Se datatyper
fjärrexekvering
    av funktioner, 178
    konfigurera programserver, 99
    lokalt språk, 284

med en fjärransluten funktion till programmet, 103
flera skärmar
inloggningshanteraren, 15
fönsterhantarere
Se även arbetsytehanterare
Fönsterhanteraren
ändra, 32
Fönster-menyn
definition, 247
ny, 249
syntax, 247
fönsterstöd för funktioner, 146
Font, dialogruta, 263
Font, resurs, 264
fonter
administrera, 263
ange egenskapssträngar, 266
antal i Arbetsinställningar, 264
användar-, i Arbetsinställningar, 264
bitmönster, 263
huvudkatalog, 280
lista tillgängliga, 264
resurser, ändra för registrering, 58
resurser, ange, 263
söka med aliasfil, 280
söka med katalogfil, 280
söka med kommandot mkfontdir, 280
system-, i Arbetsinställningar, 264
X Logical Font Description, 263, 265
xlsfonts, kommando, 281
FontSet, resurs, 264
foreground, resurs, 268
foregroundColor, resurs, 31, 274
fp_dynamic, katalog, 213
fråga efter funktioner, 169
Frontpanel
ikoner i registreringspaketet, 74
meny, 247
frontpanel
hur definitioner organiseras, 215
frontpanelen, 238
ändra, 219
animering, 234
anpassa, 211
använda funktioner, 129
dynamiska anpassningar, 213, 214
hanteras av arbetsytehanteraren, 242
hjälp, 235
ikon för målområde, 231
ikoner, Se ikoner
komponenter, 216
konfigurationsfilen, 212
kontrollera egna anpassningar, 214
läge på skärmen, 238
lägga till rader, 239
namngivningskonvention för filer, 212
ny, 239
prioritet i assemblering, 213
sökvägar, 212
syntax, 216
visa ikoner i, 208
funktioner
acceptera flera släppta filer, 173
acceptera släppta filer, 134, 168
acceptera släppta filer eller fråga efter, 169
ändra, 164
använda, 134
används av frontpanelen, 129
används på menyer, 130
argument, 167
argument, icke-fil, 170
associera ikoner med, 207
associera med datatyper, 193
begränsade av antal argument, 176
begränsade av datatyp, 176
begränsade av en datatyp, 137
begränsningar i Skapa funktion, 142
COMMAND, 156
exekveringssträng, 167
exempel, 159, 160
fält, 158
filargument, 133
filer som representerar, Se funktionsfil, 162

fönsterstöd, 146
fönsterstöd för, 173
förhållande till datatyper, 136
fråga efter fil, 169
icke utbytbara argument, 172
icke-filargument, 142, 170
ikoner för, 153, 163
ikoner som representerar, 162
inga argument, 168
inget fönster, 146
integrera med dtappintegrate, 76
introduktion, 127, 128
konfigurationsfiler, 157
köra andra funktioner, 181
köra fjärrprogram, 178
köra som andra användare, 182
lokaliserade, 182
MAP, 157
miljövariabler, 180
namn, 142, 158, 162, 163, 183
nödvändiga för registreringen, 62
olika metoder för dubbelklickning och släpp, 176
parametrar, Se argument
prioritetsregler, 165
redigera, 164
servrar för, 101
skal för, 170
skapa ikoner för program, 131
skapa manuellt, 155, 157, 158
sökväg, Se databassökväg
som kör fjärrprogram, 103
standardikon, 163
standardterminal, 174
starta om, 161
strängvariabler i, 179
terminalstöd för, 174
terminalternativ, 174
tilldelning, 137
TT_MSG, 157
typer av, 156, 158
utan data, 168
utbytbara argument, 172
variabler i definitioner, 179
funktionsdefinitionsfil, skapad Skapa funktion, 141
funktionsfil, 142, 162
  definition, 131
  innehåll, 132
  skapa, 72, 162
funktionsikon, 131, 162
  dubbelklicka, 134
  nödvändig för skrivbordet, 67
  skapa, 162
Funktionsikoner, verktyg i Skapa funktion, 145
Funktionsnamn, fält i Skapa funktion, 145
funktionsservrar, Se databasservrar
funktionssökväg, Se databassökväg

G

getty, 6, 25
GID, 90
granska ikoner med filhanteraren, 209
grupp-ID, 90

H

Help Developer’s Kit, 66
HELP_STRING, fält, 235
HELP_TOPIC, fält, 236
HELP_VOLUME, fält, 236
hemkatalog
  delad, 91
  nätverk, 91
hemsession, 24
HIGH_COLOR, 273
hjälp
  Se även hjälpvolym, hjälpfiler
  ange med Skapa funktion, 146
  delvis integration, 66
  för skrivarikoner, 110
  frontpanelen, 235
  fullständig integration, 66
  med datatyper, 192
  objekt-, fil, 163
hjälpfiler

i registreringspaket, 66
integrera med dtappintegrate, 76
hjälpservrar, 89
klient till, 102
konfigurera, 94, 101
skapa, 101
hjälpsökväg, 123
Se även sökväg
assemblera, 124
förhållande till programsökväg, 119, 124
miljövariabler, 124
standard-, 123
syntax, 124
hjälpvolym
hjälpmasterfilen, 66
nivåer av integration, 66
placera i registreringspaketet, 66
hjälpvolymer
lokalisera, 283
HOME, variabel, 19
home.old, katalog, 37

I

ICON, fält
för datatyper, 192
giltiga värden, 163, 192
i frontpanelen, 223
icon, ikontyp, 232
ikon
ändra, 221
animering, 234
byta plats med andra ikoner, 222
definiera, 228
ersätta, 222
i knapp för byte av arbetsyta, 236
ikonbild, 223
klicka eller dubbelklicka, 238
klient, 233
lägga till i huvudpanelen, 220
låsa, 215
målområde, 231
namnge, 238
objekthjälp, 235
övervaka, 231
skapa, 229
som öppnar en fil, 230
ta bort från frontpanelen, 220
typer, 229
utseende, 223
växla, 232
växlings-, 232
ikoner
använda filhanteraren som ikonfönster, 209
associera, 206
associera med funktioner och datatyper, 207
associera med programfönster, 208
återställa, 215
basfilnamn, 192
datatyp, 67
datatyper, 153, 192
färganvändning, 210
filformat, 204
filhanteraren som ikonfönster, 209
filnamn, 204
för program, 131
frontpanelen, 208, 223
funktioner, 153, 163
funktionsikoner, 162
hur filer hittas, 204
icke-engelska, 282, 283
integrera med dtappintegrate, 77
lokalisera, 282, 283
namngivningskonventioner, 204
nödvändiga för registreringen, 67
programgrupper, 67, 70
rekommendationer när du skapar, 209
representerar program, 72
servrar, Se ikonservrar
skrivarbilder, 109
Sök uppsättning, dialogruta, 153
sökväg, Se ikonsökväg
starta program, 67
storlekskonventioner, 205
ikoner för frontpanelen, 229
ikonservrar, 89

klient till, 102
konfigurera, 93, 101
skapa, 101
ikonsökväg, 121
assemblera, 123
förhållande till programsökväg, 119, 122
miljövariabler, 122
standard-, 122
syntax för, 122
ikonstorlekar, 205
inactiveFrameColorId, resurs, 269
include, programsats, i arbetsytehanterarens filer, 243
indatavariabler, 114
-indirect, alternativ, 7
inetd.conf, 96
inetd.sec, 97
inloggningsbild
ändra funktion, 14
ändra standardspråk, 16
ändra utseende, 12
ändra välkomstmeddelande, 13
anpassa, 11
fonter, 13
hälsning, 13
lokalisera, 14
resurser, 12
skärmberoende funktionalitet, 15
Inloggningshanteraren
Se även inloggningsserver
anpassa, 1
definition, 1
fel, 10
konfigurationsfiler, 21
resurser, 12, 14
starta kommandon, 17
inloggningskonton, 90
inloggningsservrar
ändra tidszon, 20
ansluta skärmar, 1
användarsökväg, 19
avaktivera, 10
avsluta, 10
avsluta process-ID, 10
felsöka, 10
ingen bitmönsterskärm, 5
kommandoradsinloggning, 5
konfigurera, 93
kontrollera behörighet, 8
köra utan lokal skärm, 5
miljöer, 18
översikt, 2
process-ID, 3
starta, 2
starta en session, 1
starta från kommandoraden, 2
systemskal, 19
textskärm, 6
verifiera användare, 1
visa inloggningsbild, 1
visa på en nätverksskärm, 6
X-servermiljö, 16
inloggningsskärm
ändra innehåll på Språk-menyn, 16
inmatningsmetod, lokalisera, 284
innehållsbaserad datatyp, 199
innehållsbaserade datatyper, 149
Installera ikon, ta bort, 227
IS_TEXT, fält, 194

J

jokertecken i datatyper, 195

K

Kalender-daemon, 98
katalog
datatypsvillkor, 198
keyBindings, resurs, 254
klient/server-konfiguration, Se nätverk
klienter
definition, 86
fönster i frontpanelen, 233
till servrar, konfigurera, 93
knapp för byte av arbetsyta
anpassa, 236

antal arbetsyta, 236
antal rader, 236
definition, 216
definition av syntax, 219
lägga till ikoner, 236
knappkoppling, 249
lägga till, 251
skapa ny uppsättning, 252
syntax, 250
kommandorad för funktioner, 167
kommandoradsinloggning, 5
konfigurationsfiler
arbetsytehanteraren, 242
datatyper, 189
fönsterhanteraren, 242
frontpanelen, 212
funktion, 157
i registreringspaketet, 56
inloggningshanteraren, 21
sessionshanteraren, 37
konsekventa filnamn, 92

L

LABEL, funktionsfält, 163
LANG, variabel, 278
anges av inloggningshanteraren, 18
effekt på datatyper, 201
i .dtprofile, 279
länk, datatypsvillkor, 198
ledtext, ange med Skapa funktion, 147
LINK_TO_ACTION, fält, 194
lokalisering
Se även internationalisering
localTerminal, resurs, 175
LOCKED, fält, 215
.login, fil, 25
läses inte av inloggningshanteraren, 33
läsa in och köra, 29
LOGNAME, variabel, 19
lokal skärmtyp, 5
lokalisera
ange språk, 278
app-defaults, 281
datatyp, 201
felsökning, 284
fonter, 280
ikoner, 282, 283
inloggningsbild, 14
meddelandekataloger, 284
NLS-miljövariabler, 280
variabeln LANG, 278
lokalisering
funktioner, 182
funktionsnamn, 183
palettnamn, 283
LOW_COLOR, 273
lp, kommando, 92
lp, skrivarspooler, 92
LPDEST, variabel, 111

M

mail, typ av ikon, 232
mailx, 92
målområde
funktionsikon, 168
ikon på frontpanelen, 231
MAP, funktioner, 137
definition, 157
exempel, 160
Markerade-menyn, 130, 136, 137
meddelandekataloger, 280
MEDIA, fält, 194
MEDIUM_COLOR, 273
menyer
använda funktioner, 130
arbetsytehanteraren, 246
miljlövariabler
funktionsdefinitioner, 180
miljövariabler
ange, 33
bildpunktsmönstersökväg, 122
bitmönstersökväg, 122
databassökväg, 120
i .dtprofile, 26
egna, 33

exportera, 33
fjärrexekvering, 97
hjälpsökväg, 124
i .dtprofile, 26
ikonsökväg, 122
inloggningshanteraren, 19
läsa in och köra .login och .profile, 29
programsökväg, 117
sökväg, 115
standardvärden, 26
miljövaribler
systemtäckande, 33
MIME_TYPE_MEDIA, fält, 194
mkfontdir, kommando, kompilera filer, 280
MODE, fält, 195
syntax, 198
monitor, typ av ikoner, 232
MONITOR_TYPE, fält, 232
MOVE_TO_ACTION, fält, 194
muskopplingar, Se knappkopplingar

N

NAME_PATTERN, fält, 195
namn
funktioner, 163, 183
ikoner på frontpanelen, 238
Namn på datatypsfamilj, textfält, 148
namnbaserade datatyper, 149, 195
namnge ikoner, 238
National Language Support
lokalisera, 277
nätverk
Se även servrar
allmänna steg för att konfigurera, 89
baskonfiguration, 90
elektronisk post, 92
filer som är nödvändiga för, 94
konfigurera klinter och servrar, 93
köra program via anslutningar, 105
översikt, 85
typer av tjänster, 86
X-verifiering, 92
NFS, 91
NLS-fjärrexekvering, 284
NLS-miljövariabler, 280
NO_STDIO, fönsterstöd, 174
NoBackdrop, inställning, 246
NoPrint, funktion, 139
NOT, operator i fältet MODE, 198
NUMBER_OF_ROWS, fält, 236
NumFonts, resurs, 264

O

Open, funktion, 137
OR, operator i fältet MODE, 198
OSF/Motif Window Manager, Se arbetsytehanterare
översättningstabell, återställa, 284

P

paletter, 267
lokalisera namn, 283
PANEL, definition, 216
syntax, 217
PANEL_GEOMETRY, fält, 238
panelundermeny
ändra, 224
ändra metoden auto-close, 227
anpassa inbyggd panel, 225
associera med huvudpanelen, 224
återsälla borttagna, 215
behållare, 218
definition, 218
ny, 225
skapa, 224
syntax, 218
systemtäckande anpassningar, 225
PATH, variabel, 168
anges av inloggningshanteraren, 18
PATH_PATTERN, fält, 195
syntax, 196
PERM_TERMINAL, fönsterstöd, 174
.pm, fitypstillägg, 204

POSITION_HINTS, fält, 222
primaryColorSetId, resurs, 269, 271
prioritet
  assemblera frontpanelen, 213
  i funktionsdatabas, 165
.profile, fil, 25
  läses inte av inloggningshanteraren, 33
  läsa in och köra, 29
Progamhanteraren
  beskrivning, 39
program
  avregistrera, 48
  datatyper som behövs, 63
  desktop-smart, 44
  funktioner som behövs, 63
  köra lokalt via anslutningar, 105
  lägga till i programhanteraren, 43
  lägga till utan registrering, 44
  progrot, katalog, 59
  registrerade, definition, 43
  registrering, Se registrering
  registreringsfuntioner, 54
  rotkatalog, 59
  samla ihop, 42
  samla ihop i programhanteraren, 40
  samlade av sessionshanteraren, 28
  sätt att lägga till, 43
  skapa ikoner för, 131
  sökväg, 41
  starta om, 48
  starta vid inloggning, 25, 32
  syfte för datatyper, 55
  ta bort, 48
programfönster, associera ikoner med, 208
programgrupper
  administrera, 45
  anpassad, 46
  datatyper för, 71
  definition, 40
  egen, 46
  exempel på att skapa, 82
  funktioner för, 71
  ikoner för, 67, 70
  innehåll, 72
  integrera med dtappintegrate, 77
  kataloger för, 68
  namn, 69
  namnge, 46
  prioritet, 41
  readme-filer, 73
  samla ihop, 40
  skapa i registreringspaket, 68
  standard, 41
  systemtäckande, 46
Programhanteraren
  allmän hantering, 48
  filsystemplacering, 40
  integrera program med, 54
  prioritetsregler, 41
  samla ihop program, 40
  samla program, 28
  symboliska länkar, 41
  uppdatera, 48
programhanteraren
  lägga till program, 43
programikon, 159
  nödvändig för skrivbordet, 67
  skapa, 131, 162
  släppta filer, 134
programikoner
  använda Skapa funktion, 142
programmets rotkatalog, 59
programpaket, Se registreringspaket
programregistrering, Se registrering
programresurser, Se resurser
programservrar
  administrera, 98
  definition, 87
  klient till, 100
  konfigurera, 93, 100
  konfigurera klient till, 100
  lägga till, 47
  standardkonfiguration, 99
  tillgänglighet för program, 57
programsökväg
  ändra, 47
  ändra prioritet, 118

används för att samla ihop program, 40
assemblera, 118
definition, 116
egen, 48
lokaliserad, 125
miljövariabler, 117
skäl för att ändra, 47
standard, 47
standard-, 117
syntax, 117
systemtäckande, 47
programstandardfiler
skrivbordsprogram, 258
progrot, Se programmets rotkatalog
PUSH_ACTION, fält, 230
PUSH_ANIMATION, fält, 235

Q

-query, alternativ, 7

R

readme, filer, 73
redigera funktioner, 164
registrera
exempel, 77 to ??
registrering
Se även registreringspaket
allmänna steg, 57
ändra färg, 59
ändra fonter, 58
ändra resurser, 58
datatyper som behövs, 62
definition, 43, 57
dtappintegrate, 75
exempel, ?? to 84
funktioner som behövs, 62
funktioner som tillförs vid, 54
hjälpfiler, 66
ikonkrav, 67
översikt, 54
programgrupp, 68
programmets rotkatalog, 59
syfte med, 56
registreringspaket, 43
Se även registrering
definition, 57
exempel på att skapa, 77
ikoner på frontpanelen, 74
kataloger, 60
progamgruppsinnehåll, 72
programikon, 72
readme-filer, 73
syfte med, 54
reglage, Se knapp för byte av arbetsyta
Reload Actions, funktion, 161
Reload Applications, funktion, 49
RESOURCE_MANAGER, egenskap, 30, 34
resurser
ange, 33, 257
colorUse, 273
egna, 258
fonter, 263
foregroundColor, 274
ladda, 30
programstandardfiler, 258
session, 25
shadowPixmaps, 274
skärmspecifika, 34
skuggtjocklek i fönster, 275
språkberoende, 281
standard för skrivbordet, 30
systemtäckande, 258
rgb.txt, fil, 269
RGB-färgvärde, 269
RGB-värden, 269
rotfönster, 246
rotmenyn, Se Arbetsyta-menyn
rpc.cmsd, 98
rpc.ttdbserver, 94, 96

S

Särskilda egenskaper
dialogruta, 149
fält, 151

.sdl, filer, 66
secondaryColorSetId, resurs, 269, 271
selectColor, resurs, 268
sendmail, 92
servrar
  databas-, 93
  datatyper, 101
  definition, 86
  fil-, 87
  funktioner, 101
  hjälp-, 89, 94
  ikon-, 89, 93
  inloggnings-, 93
  konfigurera, 93
  program-, 87, 93
  sessions-, 93
    servrar
      inloggnings-, 87
  typer, 89
session
  aktiv, 24
  återställa, 36
  definition, 23
  exekvera kommandon vid start, 36
  exekvera kommandon vid utloggning, 36
  första, 35
  hem, 24
  initial, 24
  logga fel, 37
  med endast ett fönster, 17
  resurser, 25
  säkerhetskopia, 36
  skärmspecifik, 24, 35
  skript som körs vid avslut, 18
  standardvärde, 24
  starta, 25
sessioner med endast ett fönster, 17
sessionetc, fil, 36
sessionexit, fil, 36
Sessionhanteraren
  inledning, 23
SessionHost, nyckelord, 179
Sessionshanteraren
  Se även, session
  ange sökvägar, 27
  anpassa program vid start, 32
  exekvera kommandon vid utloggning, 36
  exekvera ytterligare kommandon, 36
  fellogg, 37
  felsökning, 37
  filer, 37
  kataloger, 37
  klient, 30
  ladda resurser, 30
  säkerhetskopiera sessioner, 36
  samla program, 28
  starta, 25
  starta arbetsytehanteraren, 32
  starta program, 32
  systemtäckande anpassning, 27
  välkomstmeddelande, 25
sessionskatalog, 35
sessionsservrar, Se inloggningsservrar
shadowPixmaps, resurs, 31, 274
SHELL, variabel, anges av inloggningshanteraren, 18
skal
  Se även miljövariabler, .profile, .login, .dtprofile
  använda i funktioner, 170
  egen anpassning, 26
  i funktioner, 170
  läsa in och köra .profile eller .login, 29
  syntax i exekveringssträng, 167
  systemtäckande anpassning, 26
Skapa, 203
Skapa funktion, 141
  ange filargument, 146
  ange ikoner, 153
  använda, 143
  begränsningar, 142
  datatypsnamn, 148
  funktionalitet, 141
  funktionsnamn, 145
  huvudfönster, 145
  introduktion, 141

ledtext, 147
skapa datatyper, 135, 146
skapa konfigurationsfil, 141
starta, 144
syntax till funktionskommando, 145
skärmfärger
maximalt antal allokerade, 272
skärmspecifik session, 35
skärmspecifika resurser, 34
skriva ut
administration, 107
skrivare
enhetsnamn, 92
fjärråtkomst, 92
ikonbilder, 109
intervall för jobbuppdatering, 109
lägga till, 108
objekthjälp, 110
standard-, 110
ta bort, 108
skrivbordssökvägar, 25
skrivskyddad
datatypsvillkor, 198
skuggtjocklek, fönster, 275
släppt fil, funktion som accepterar, 168
Sök uppsättning, dialogruta, 153
sökväg
användare, 19
system, 19
sökvägar
aktuellt värde, 115
ange, 115
ange med sessionshanteraren, 27
definierade av skrivbordet, 114
frontpaneldefinitioner, 212
funktioner, 157
hjälp, 123
ikoner, 206
indatavariabler, 114
lokaliserade, 125
miljövariabler, 115
program, 41
program-, 116
utdatavariabler, 115
sökvägsbaserad datatyp, 196
SPC, 96
säkerhet, 97
språk, ange med filen Xconfig, 278
Språk-menyn, anpassa, 16
startlog, fil, 37
strängfunktionsarguments, 170
strängvariabler i funktionsdefinitioner, 179
styrdaemon för underprocesser, Se dtspcd
SUBPANEL, definition, 216
SWITCH, definition, 219
symboliska länkar
datatypsvillkor, 195
konsekventa filnamn, 92
skapa vid registreringen, 76
sys.dtprofile, fil, 26
sys.dtwmrc ,fil, 243
sys.dtwmrc, fil, 242
sys.resources, fil, 30, 33, 258
sys.session, fil, 32, 35
systemPath, resurs, 19

T

tangentkoppling
skapa ny uppsättning, 254
standard-, 252
syntax, 253
TERMINAL, fönsterstöd, 174
terminalemulator
ändra, 49
för funktioner, 173
kommandoradsalternataiv för funktioner, 174
som stängs automatiskt, 146
som stängs manuellt, 146
standard- för funktioner, 174
textColorSetId, resurs, 269
textredigerare, ändra, 49
textskärm, 6
tidszon, ändra, 20
timeZone, resurs, 20

title, resurs, 245
tjänst för underprocesskontroll, Se SPC
ToolTalk
  databasserver, Se rpc.ttdbserver
  funktion, Se TT_MSG, funktion
  meddelande-daemon, 25, 29
  program, funktioner för, 183
ToolTalk meddelandeserver, Se ttsession
topShadowColor, resurs, 268
TT_MSG, funktion
  nyckelord, 185
  skapa, 183
ttsession, 98
  starta, 29
TYPE, fält, 229
TZ, variabel, 18, 20

U

UNIXbindings, fil, 259
UNIX-tangentkopplingar, 257, 258
uppdateringsintervall för utskriftsjobb, 109
USER, variabel, 19
userPath, resurs, 19
user-prefs.dt, fil, 51
utdatavariabler, 115
utskrift
  konfigurera för datatyp, 139
  olika datatyper, 112
  standarddestination, 111
  testa, 92
  vad som händer, 112
Utskriftshanteraren, 108
  intervall för jobbuppdatering, 109

V

välkomstmeddelande
  ändra, 13
  anpassa, 27
  standardvärde, 13
  visa, 25, 27
värddator för exekvering
  ange, 103
  anges av fältet EXEC_HOST, 178
  konfigurera, 104
  skapa funktioner för, 178
variabler
  i funktionsdefinitioner, 179
  miljö, Se miljövariabler
verifiering, X, 92
verifieringskatalog, 91, 97

W

WINDOW_TYPE, fält, 174
windowMenu, resurs, 247
WM_CLASS, egenskap, 208
wmStartupCommand, resurs, 32
workspaceCount, resurs, 245
writeXrdbColors, resurs, 31

X

X Logical Font Description, 263
X resources, Se resources
X400_TYPE, fält, 194
Xaccess, fil, 8
XAUTHORITY, variabel, anges av inloggningshanteraren, 18
Xconfig, fil
  ändra, 3
  ange resurser i, 14
  ange språk med, 278
.Xdefaults, fil, 30, 258
XDMCP, 2
  definition, 6
  direkt åtkomst, 8
  direkt förfrågan, 7
  frågeläge, 7
  indirekt åtkomst, 9
  indirekt förfrågan, 7, 20
Xerrors, fil, 10
Xfailsafe, fil, 17, 19, 20
xlsfonts, kommando, 264
  installation, 281

lista fonter på servern, 281
xlsfonts, kommanod
installation, 281
XMICONBMSEARCHPATH, variabel
användning, 122
assemblera, 123
definition, 115
XMICONSEARCHPATH, variabel
användning, 122
assemblera, 123
definition, 115
XmText*FontList, resurs, 264
XmTextField*FontList, resurs, 264
Xpid, fil, 3
Xreset, fil, 18
Xresources, fil, 12, 13
X-server
ändra behörighet, 15
ändra miljö, 16
Xservers, fil
hantera lokal skärm, 20
standardvärde, 5
starta en server, 4
syntax, 4
Xsession, fil, 26
ange PATH, 19
körs av inloggningsservern, 17
starta sessionshanteraren, 25
systemtäckande anpassning, 27
Xsession.d, katalog, 25, 33
anpassa, 26
skript i, 26
Xsetup, fil, 17
Xstartup, fil, 17
X-terminaler, 93
CHOOSER, sträng, 9
hämta sessionstjänster, 88
listan Xaccess, 8
möjliga konfigurationer av inloggningsservrar, 3
skärmar utan XDMCP, 7
XDMCP direkt, 7, 8
XDMCP indirekt, 7, 9
XUSERFILESEARCHPATH, variabel, 281
X-verifiering, 92

De nombreuses appellations utilisées par les constructeurs et les distributeurs sont des marques déposées. Dans la présente publication, les termes en majuscules ou commençant par une majuscule peuvent être des marques déposées.

LA PRESENTE PUBLICATION EST FOURNIE "DANS L'ETAT". LES SOCIETES DETENTRICES DU COPYRIGHT DECLINENT TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE ET A L'APTITUDE A UNE UTILISATION PARTICULIERE. PAR AILLEURS, CES SOCIETES NE SAURAIENT ETRE TENUES POUR RESPONSABLES D'UN DOMMAGE RESULTANT DE L'UTILISATION DES INFORMATIONS DU PRESENT DOCUMENT.

Le code et la documentation des widgets DtComboBox et DtSpinBox ont été fournis par Interleaf, Inc. Copyright 1993, Interleaf, Inc.

Copyright © 1993, 1994, 1995, 1996 Hewlett-Packard Company
Copyright © 1993, 1994, 1995, 1996 International Business Machines Corp.
Copyright © 1993, 1994, 1995, 1996 Sun Microsystems, Inc.
Copyright © 1993, 1994, 1995, 1996 Novell, Inc.

Tous droits réservés. Ce produit et sa documentation sont protégés par copyright et des licences en restreignent la distribution, l'utilisation, la copie et la décompilation. Aucune partie de ce produit ou de sa documentation ne peut être reproduite, sous quelque forme et par quelque moyen que ce soit, sans autorisation écrite préalable.

UNIX est une marque dont la licence est détenue exclusivement par

X / Open Company, Ltd.

OSF / Motif et Motif sont des marques d'Open Software Foundation, Ltd.

X Window System est une marque de X Consortium, Inc.

PostScript est une marque d'Adobe Systems, Inc., qui peut être enregistrée dans certains pays.