



Common Desktop Environment: Användarhandbok för systemadministratörer

Sun Microsystems, Inc.
901 San Antonio Blvd.
Palo Alto, CA 94303
USA

Artikelnummer: 805-0020
September, 1997

Copyright 1997 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303-4900 USA. Med ensamrätt.

Den här produkten och tillhörande dokumentation är skyddad genom copyright och distribueras under licens, vilket begränsar hur det får användas, kopieras, distribueras och delas upp. Ingen del av denna produkt eller tillhörande dokument får återges i någon form utan uttryckligt skriftligt tillstånd från Sun eller från licensgivare.

Delar av denna produkt kan komma från UNIX[®] systemet, licensierad från Novell, Inc och från Berkeley BSD system, licensierad från University of California. UNIX är ett registrerat varumärke i USA och i andra länder och är exklusivt licensierat genom X/Open Company, Ltd. Tredjepartsprogram, inklusive teckensnittsteknik i produkten är skyddade genom copyright och licensierade från leverantörer till Sun.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227 - 14(g)(2)(6/87) and FAR 52.227 - 19(6/87), or DFAR 252.227 - 7015(b)(6/95) and DFAR 227.7202 - 3(a).

Sun, Sun Microsystems, Sun-logotypen, SunSoft, SunDocs, SunExpress och Solaris är varumärken, registrerade varumärken eller servicemärken från Sun Microsystems, Inc. i USA eller i andra länder. Alla SPARC-varumärken används under licens och är varumärken eller registrerade varumärken från SPARC International, Inc. i USA eller i andra länder. Produkter som bär SPARC-varumärken baseras på arkitektur utvecklad av Sun Microsystems, Inc. Koden och dokumentationen för bildmodulerna DtComboBox och DtSpinBox tillhandahålls av Interleaf, Inc. Copyright 1993, Interleaf, Inc. , Interleaf, Inc.

De grafiska användargränssnitten OPEN LOOK[®] och Sun[®] är utvecklade av Sun Microsystems, Inc. för dess användare och licenstagare. Sun vill tacka Xerox för de banbrytande insatser i forskning och utveckling av visuellt och grafiskt användargränssnitt som de gjort för datorindustrin. Sun innehar en licens utan ensamrätt från Xerox till Xerox grafiska användargränssnitt. Licensen täcker också Suns licensinnehavare som implementerar OPEN LOOK GUI:s och på annat sätt samtycker till Suns skriftliga licensavtal.

DENNA PUBLIKATION TILLHANDAHÅLLS "I BEFINTLIGT SKICK" UTAN NÅGRA SOM HELST GARANTIER, VARKEN UNDERFÖRSTÅDDA ELLER UTTRYCKLIGA, INKLUSIVE, MEN INTE BEGRÄNSAD TILL, DE UNDERFÖRSTÅDDA FÖRSÄLJNINGSGARANTIERNÄ, LÄMPLIGHET FÖR ETT SÄRSKILT ÄNDAMÅL, ELLER ATT DE INTE INKRÄKTAR PÅ ANDRA FÖRETAGS RÄTTIGHETER.



Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou de sa documentation associée ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Des parties de ce produit pourront être dérivées du système UNIX[®] licencié par Novell, Inc. et du système Berkeley BSD licencié par l'Université de Californie. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

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

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

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, NI EXPRESSE NI IMPLICITE, Y COMPRIS, ET SANS QUE CETTE LISTE NE SOIT LIMITATIVE, DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DES PRODUITS A RÉPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ILS NE SOIENT PAS CONTREFAISANTS DE PRODUITS DE TIERS.

Innehåll

Förord xxi

1. Konfigurera inloggningshanteraren 1

Starta inloggningsservern 2

Hantera lokala skärmar och nätverksskärmar 2

 Söka efter inloggningsserverns process-ID 3

 Visa en inloggningsbild på en lokal skärm 3

 Köra inloggningsservern utan lokal skärm 4

 Kommandoradsinloggning på lokala skärmar 4

 Anpassa textskärmen 5

 Visa en inloggningsbild på en nätverksskärm 5

 Kontrollera tillgång till inloggningsservern 7

Felsökning 8

Inaktivera inloggningsservern 9

Inloggningsbilden 9

Ändra utseendet på inloggningsbilden 10

▼ Ändra logotyp 11

▼ Ändra välkomstmeddelandet 11

▼ Ändra teckensnitt 11

▼ Ange olika texter för olika språk 12

Ändra hur inloggningsbilden fungerar skärmvis	13
Ändra X-serverbehörighet	13
Starta kommandon innan inloggningsbilden visas	15
Starta en Felsäker session	15
När användarsessionen avslutas	16
Inloggningsserverns miljö	16
Ändra användar- eller systemsökvägen	17
▼ Ändra systemskalet	18
▼ Ändra tidszon	18
Administrera inloggningshanteraren	18
Inloggningshanterarens filer	19
2. Konfigurera sessionshanteraren	21
Vad är en session?	21
Startsession	22
Aktiv session	22
Hemsession	22
Skärmspecifika sessioner	22
Starta en session	23
När sessionen startar	23
Läsa in och köra skriptet .dtprofile	24
Läsa in och köra skriptet Xsession.d	24
Visa välkomstmeddelandet	25
Ställa in skrivbordets sökvägar	25
Samla ihop tillgängliga program	26
Läsa in och köra skripten .profile och .login (valfritt)	26
Starta ToolTalk	26
Starta sessionshanterarklienten	27
Ladda sessionsresurser	27

Starta färgservern	28
Starta arbetsytehanteraren	29
Starta sessionsprogram	29
Ytterligare anpassningar av sessionsstarten	29
▼ Ange miljövariabler	30
▼ Ange resurser	30
▼ Ange skärmspecifika resurser	31
▼ Ändra program för startsessionen	31
▼ Ange en skärmspecifik session	32
Köra ytterligare kommandon vid sessionsstart och utloggning	32
▼ Köra ytterligare kommandon vid sessionsstart	33
▼ Köra ytterligare kommandon vid utloggning	33
▼ Återskapa en session från säkerhetskopia	33
▼ Undersöka problem vid sessionsstart	34
Filer och mappar för sessionshanteraren	34
3. Felsökning för inloggning och problem vid sessionstart	35
Startfiler för inloggning	35
Fellogningskatalog	36
Användarstartfil	37
Solaris CDE, startexempel	37
4. Lägga till och hantera program	39
Programhanterarens struktur	39
Programhanterarens plats i filsystemet	40
Hur programhanteraren hittar och samlar ihop program	40
Prioritetsregler för insamling av program	41
Programgrupper på standardskrivbordet	41
Exempel på hur programgrupper samlas in	42
Lägga till program i programhanteraren	43

	Lägga till program i programhanteraren	43
▼	Lägga till skrivbordsanpassade program i Programhanteraren	44
▼	Registrera befintliga program eller program som inte är skrivbordsanpassade	45
▼	Lägga till en programikon i befintlig programgrupp	45
	Skapa och hantera allmänna programgrupper	46
▼	Skapa en systemtäckande allmän programgrupp	46
▼	Skapa en egen allmän programgrupp	46
▼	Anpassa en inbyggd programgrupp	46
	Ändra sökvägen som används för att söka efter program	47
	Standardsökvägen	47
	Lägga till en programserver i programsökvägen	48
	Allmän administration av Programhanteraren	49
▼	Ta bort ett program	49
▼	Uppdatera programhanteraren under en session	49
	Ändra textredigeraren och terminalemulatorn	50
▼	Ändra standardtextredigerare och standardterminalemulator	50
5.	Registrera ett program	53
	Programregistrering – översikt	54
	Funktioner som medföljer programregistrering	54
	Syftet med programregistrering	56
	Generella steg för registrering av program	57
	Steg 1: Ändra teckensnitts- och färgresurser	58
	Steg 2: Skapa skrivbordsprogrammets rot	59
	Steg 3: Skapa registreringspaketets mappar	60
	Steg 4: Skapa funktionerna och datatyperna för programmet	63
	Steg 5: Placera hjälpfilerna i registreringspaketet	66
	Steg 6: Skapa ikoner för programmet	66
	Steg 7: Skapa programgruppen	68

Steg 8: Registrera programmet med hjälp av dtappintegrate	73
Exempel på hur du kan skapa ett registreringspaket	76
Information om "BästaTextredigerare"	76
Steg för att registrera "BästaTextredigerare"	77
6. Diverse konfigurationer	83
Solaris CDE – mappstruktur	83
/usr/dt	84
/etc/dt	84
/var/dt	85
\$Hemmapp	85
Huvudkonfigurationsfiler	85
Xconfig	85
Xservers	85
Starta inloggningsservern	86
Ansluta en installerad CDE från en annan installationsadress för en arbetsstation eller nätverksserver	88
▼ Ansluta en installerad CDE	88
▼ Koppla ur en ansluten CDE-mapp	89
Konfigurera skrivbordet för att använda flera skärmar	89
▼ Starta skrivbordet på flera skärmar	89
Nätverksanslutna skrivbord	91
Använda X-terminaler	92
Inloggningsspråkområde och teckensnittssökväg	93
Använda arbetsstationer som X-terminaler	94
▼ Använda Väljaren för att markera en värd-CDE-inloggning	94
▼ Använda en specifik värd-CDE-inloggning	95
▼ Använda den första tillgängliga värdinloggningen	95
Särskilda CDE-konfigurationer	95

	Anpassa utskrift av brev	96
	Konvertera kalendrar till nya dataformat	97
	Lägga till AnswerBook-paketet från nätverket	97
	Installera CDE-miljö utanför CDE-skrivbordet	98
	Skrivbordsmiljöfil	98
	Använda diskett och CD-skiva med Apple Macintosh programmiljö	99
	XServer och drivrutinen Power Management	100
7.	Konfigurera skrivbordet i nätverk	103
	Nätverk på skrivbord – översikt	103
	Typer av nätverkstjänster för skrivbord	104
	Typiska nätverkssituationer	104
	Andra nätverkssituationer	106
	Sammanfattning – Typer av servrar	107
	Generella steg för att konfigurera nätverket	108
	Konfigurera operativsystemets basnätverkstjänster för skrivbordet	108
	Inloggningskonto för användare	109
	Konfigurera distribuerad filsystemåtkomst	109
	Konfigurera åtkomst av fjärrskrivare	110
	Konfigurera e-post	111
	Konfigurera som X-verifiering	111
	Konfigurera skrivbordsklienter och servrar	111
	Konfigurera inloggnings- och sessionstjänster	111
	Konfigurera andra programrelaterade tjänster	112
	Administrera programtjänster	116
	Miljövariabler för sökvägar	117
	Konfigurera en programserver och dess klienter	117
	Konfigurera databas-, ikon- och hjälptjänster	118
	Speciella konfigurationer för nätverksprogram	120

- 8. Konfigurera och administrera utskrifter från skrivbordet 125**
 - Lägga till och ta bort skrivare 125
 - ▼ Lägga till en skrivare till skrivbordet 125
 - ▼ Ta bort en skrivare från skrivbordet 126
 - Ändra intervall för jobbuppdatering 127
 - Ikonbilder till skrivaren 127
 - Ikonfilnamn och ikonfilstorlekar 127
 - ▼ Ändra ikon, skrivarnamn och beskrivning globalt 127
 - Konfigurera standardskrivare 128
 - ▼ Ändra destination för standardutskrift 128
 - Vad som sker vid utskrift 129
- 9. Skrivbordets sökvägar 131**
 - Skrivbordets sökvägar och deras miljövariabler 132
 - Ange värde för en sökväg 133
 - ▼ Visa aktuellt värde för en sökväg (utdatavariabel) 133
 - ▼ Göra egna ändringar av sökvägen 133
 - ▼ Göra systemtäckande ändringar av sökvägen 134
 - Programsökväg 134
 - Standardsökväg för program 134
 - Miljövariabler för programsökvägar 135
 - Syntax för indatavariabler för programsökväg 135
 - Hur programsökvägens värde sätts samman 135
 - Ändra prioritetsordning på den lokala systemtäckande adressen 136
 - Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp 136
 - Databassökväg (funktion/datatyper) 137
 - Databasens standardsökväg 138
 - Hur programsökvägen påverkar databassökvägen 138

	Databassökvägens miljövariabler	138
	Syntax för databassökvägens indatavariabler	138
	Hur databassökvägen sätts samman	139
	Ikonsökväg	139
	Standardikonsökväg	139
	Hur programsökvägen påverkar ikonsökvägen	140
	Ikonsökvägens miljövariabler	140
	Syntax för ikonsökvägens indatavariabler	140
	Hur ikonsökvägen sätts samman	141
	Hjälpsökväg	141
	Standardhjälpökväg	141
	Hur programsökvägen påverkar hjälpökvägen	141
	Hjälpsökvägens miljövariabler	142
	Syntax för hjälpökvägens indatavariabler	142
	Hur ikonsökvägen sätts samman	142
	Lokaliserade sökvägar	143
10.	Introduktion till funktioner och datatyper	145
	Introduktion till funktioner	146
	Hur funktioner skapar programikoner	148
	Hur funktioner använder datafiler som argument	150
	Ytterligare användningsområden för funktioner	151
	Introduktion till datatyper	151
	Vad är en datatyp?	152
	Hur datatyper ansluter datafiler till funktioner	152
	Skriva ut datatyper från skrivbordet	155
11.	Skapa funktioner och datatyper med Skapa funktion	157
	Vad Skapa funktion gör	157
	Begränsningar i Skapa funktion	158

Funktionsbegränsningar	158
Datatypsbegränsningar	159
Skapa en funktion och datatyp för ett program med Skapa funktion	159
▼ Skapa en funktion för ett program	160
▼ Skapa en eller flera datatyper för programmet	162
Använd dialogrutan Sök ikonuppsättning för att ange en ikon	168
12. Skapa funktioner manuellt	171
När funktioner måste skapas manuellt	172
COMMAND-funktioner	172
MAP-funktioner	172
TT_MSG-funktioner (ToolTalk-meddelanden)	173
Skapa en funktion manuellt: Generella steg	173
Konfigurationsfiler för funktioner	173
▼ Skapa en funktion manuellt	173
Exempel på hur du skapar en COMMAND-funktion	174
Exempel på hur du skapar en MAP-funktion	175
▼ Starta om databasen för funktioner och datatyper	176
Skapa funktionsfil (ikon) för funktion	177
Ange funktionens ikonbild	178
▼ Ändra en befintlig funktionsdefinition	179
Prioritet i funktionsdefinitioner	180
Skapa en exekveringssträng för COMMAND-funktioner	181
Generella egenskaper för exekveringssträngar	182
Skapa funktioner som inte använder argument	183
Skapa funktioner som accepterar släppta filer	183
Skapa en funktion som frågar efter ett filargument	184
Skapa funktioner som accepterar eller frågar efter en släppt fil	184
Skapa en funktion som frågar efter icke-filargument	184

Tolka filargument som en sträng	185
Skalmöjligheter för funktioner	185
Skapa COMMAND-funktioner för flera filargument	186
Fönsterstöd och terminalemulatorer för COMMAND-funktioner	188
Ange fönsterstöd för funktionen	188
Ange kommandoradsalternativ för terminalemulatorn	189
Ange en annan standardterminalemulator	189
Begränsa funktioner för vissa argument	190
Begränsa funktioner till en viss datatyp	190
Begränsa funktioner med antalet argument	190
▼ Ange metod för dubbelklickning och släppfunktionen	191
Begränsa funktioner baserat på argumentläge	192
Skapa funktioner som kör program i fjärrsystem	192
Skapa en funktion som kör ett fjärrprogram	192
Använda variabler i funktions- och datatypsdefinitioner	193
Använda strängvariabler i en funktion	193
Använda miljövariabler i funktioner och datatyper	194
Starta funktioner från en kommandorad	195
Syntax för <code>dtaction</code>	195
Skapa en funktion som kör en annan funktion	195
Skapa funktioner som kör som andra användare	195
Skapa språkanpassade funktioner	196
Adresser för språkanpassade funktioner	196
▼ Språkanpassa en befintlig funktion	196
Skapa funktioner för ToolTalk-program	197
Fälten <code>addressing</code> och <code>disposition</code>	197
Meddelanden som inte stöds	198
13. Skapa datatyper manuellt	199

Varför du måste skapa datatyper manuellt	199
Komponenter i datatypsdefinitionen: Villkor och attribut	200
Skapa datatyper manuellt: Allmänna steg	200
Konfigurationsfiler för datatyper	201
▼ Skapa en datatypsdefinition	201
Exempel på hur du skapar en egen funktion och datatyp	202
Definiera datatypens dataattribut	203
Ange ikonbild för datatypen	203
Associera datatyper med funktioner	204
Dölja filer baserat på datatyp	205
Ange metoder när en fil manipuleras	205
Definiera dataattribut för en datatyp	206
Namnbaserade datatyper	207
Sökvägsbaserade datatyper	208
Datatyper baserade på namn och sökväg	208
Använda fillägen som typvillkor	209
Innehållsbaserad datatypsbestämning	211
▼ Skapa en datatyp med flera oberoende villkor	211
Skapa språkanpassade datatyper	212
Adresser för språkanpassade datatyper	212
▼ Språkanpassa en datatyp	212
14. Skapa skrivbordsikoner	215
Ikonbilsfiler	215
Ikonfilformat	216
Ikonfilnamn	216
Storlekskonventioner	216
Ikonsökvägen	218
Komma åt ikoner via nätverket	218

Associera ikoner	218
Ange ikonfiler	218
▼ Associera en ikon med en funktion eller datatyp	218
▼ Visa en ikon i Frontpanelen	219
▼ Associera en ikon med ett programfönster	220
▼ Använda Filhanteraren som ett ikonfönster	220
Rekommendationer hur du skapar ikoner	221
Hur färger används	221
15. Särskild anpassning av Frontpanelen	223
Konfigurationsfiler för Frontpanelen	223
Standardfrontpanelens konfigurationsfil	224
Sökvägen till Frontpanelens konfigurationsfiler	224
Hur Frontpanelen sätts samman: prioritetsregler	225
Skapa frontpanelsfiler dynamiskt	225
Administrera anpassning av användargränssnitt	226
▼ Förhindra egna anpassningar	226
▼ Återställa en borttagen kontroll eller panelundermeny	226
Hur frontpanelsdefinitionen är organiserad	227
Komponenter på Frontpanelen	227
Allmän syntax för frontpanelsdefinitionen	227
Anpassa huvudpanelen	230
▼ Lägg till en kontroll på huvudpanelen	230
▼ Ta bort en kontroll	231
▼ Ändra en kontroll	232
▼ Byta position för kontrollen	232
▼ Ersätta en frontpanelkontroll	233
Ange vilken ikonbild som ska användas av kontrollen	234
Skapa och ändra panelundermenyer	234

- ▼ Skapa en ny systemtäckande panelundermeny 235
 - Anpassa inbyggda panelundermenyer 236
- ▼ Så här tar du bort den automatiska stängningen (Auto-Close) av panelundermenyer 238
- Definiera kontroller på Frontpanelen 238
 - Definitioner till kontroller på Frontpanelen 239
 - Kontrolltyper 239
- ▼ Skapa en ny kontroll 239
- Anpassa området för växling av arbetsyta 245
- ▼ Ändra standardantalet arbetsytor 245
- ▼ Ändra antalet knapprader i växlingsområdet 245
- ▼ Ändra eller lägga till kontroller i växlingsområdet 246
- Allmän konfiguration av Frontpanelen 246
 - Allmänna steg 247
- ▼ Ändra Frontpanelens standardposition 247
- ▼ Namnge kontroller på huvudpanelen 247
- ▼ Ändra klickmetod för kontroller 248
- ▼ Skapa en helt ny frontpanel 248
 - Exempel på hur du skapar en egen Frontpanel med tre rader 248
- 16. Anpassa arbetsythanteraren 251**
 - Arbetsythanterarens konfigurationsfiler 252
 - ▼ Skapa och ändra en egen konfigurationsfil 252
 - ▼ Skapa en systemtäckande konfigurationsfil 253
 - ▼ Inkludera andra filer 253
 - ▼ Starta om Arbetsythanteraren 254
 - Anpassa arbetsytor 254
 - ▼ Ändra antalet systemtäckande arbetsytor 254
 - ▼ Namnge systemtäckande arbetsytor 255

- ▼ Skapa ytterligare bakgrunder 255
- ▼ Ersätta bakgrunden med en grafisk bild 255
- Arbetsytehanterarens menyer 256
 - Arbetsytehanterarens menysyntax 256
- ▼ Anpassa Arbetsyta-menyn 257
- ▼ Skapa en ny Arbetsyta-meny 258
- ▼ Skapa en ny Fönster-meny 258
- Anpassa knappkopplingar 259
 - Syntax för knappkoppling 259
- ▼ Lägg till knappkopplingar 260
- ▼ Skapa nya knappkopplingsuppsättningar 260
- Anpassa tangentkopplingar 261
 - Förinställda tangentkopplingar för skrivbordet 261
 - Syntax för tangentkopplingar 261
 - ▼ Skapa en anpassad tangentkopplingsuppsättning 262
- Växla mellan de förinställda och egna uppsättningarna 263
- 17. Hantera programresurser, teckensnitt och färger 265**
 - Ange programresurser 265
 - ▼ Ange systemtäckande resurser 266
 - ▼ Anger egna resurser 266
 - Hur skrivbordet startar resurser 266
 - Definiera UNIX-kopplingar 266
 - ▼ Ange EMACS-stildefinitioner 266
 - ▼ Ändra EMACS-stildefinitioner 267
 - UNIX-kopplingar i filen UNIXbindings 267
 - Hantera teckensnitt 271
 - Ange teckensnittsresurser för skrivbordet 271
 - ▼ Visa tillgängliga teckensnitt 272

- ▼ Ange teckensnitt på kommandoraden 273
 - X Logical Font Description (XLFD) 273
 - Lagring av användares filsystem för teckensnittgrupper 275
 - Skapa teckensnittgrupp för systemadministratör 276
- Hantera färger 276
 - Färgpaletter 276
 - Färguppsättningar 277
 - Kontrollera färg med Inställningar 280
 - Antal färger som används i Inställningar 281
- Ange skuggtjocklek för programfönster 284
- 18. Konfigurera språkpassade skrivbordsessioner 285**
 - Hantera miljövariabeln LANG 285
 - Ange språk för flera användare 286
 - Ange språk för en session 287
 - Ange språk för en användare 287
 - Miljövariabeln LANG och sessionskonfiguration 287
 - Ange andra NLS-miljövariabler 288
 - Söka efter teckensnitt 288
 - Språkpassa resursfilerna app-defaults 288
 - Språkpassa funktioner och datatyper 289
 - Språkpassa ikoner och bitmappar 289
 - Språkpassa bakgrundsnamn 290
 - Språkpassa palettnamn 290
 - Språkpassa hjälpvolym 291
 - Språkpassa meddelandemappar 291
 - Köra språkpassade fjärranslutna skrivbordsprogram 292
 - Återställa tangentbordstabellen 292
- A. Direkthjälp (man) för dtconfig(1) 293**

Förord

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

- hur du anpassar systemstart, inloggning och startsession
- hur du lägger till program och förser program och dess data med gränssnitt
- hur du konfigurerar skrivbordsprocesser, program och data via nätverk
- hur du anpassar skrivbordet, t ex fönsterhantering, utskrift, färger och fonter

Obs! Termen "x86" refererar till Intel 8086-familjen med mikroprocessorer, inklusive processorerna Pentium och Pentium Pro samt kompatibla mikroprocessorer gjorda av AMD och Cyrix. I det här dokumentet refererar termen "x86" även till alla plattformar där "*Intel Platform Edition*" finns med som produktnamn.

Den här boken riktar sig till

Bokens målgrupp:

- Systemadministratörer. Många av uppgifterna i boken kräver rotbehörighet.
- Vana användare som vill utföra anpassningar av systemet som inte går att utföra med hjälp av skrivbordets verktyg. Skrivbordet ger användarspecifika sökvägar till de flesta konfigurationsfilerna.

Innan du läser denna bok

Användare bör känna till innehållet i följande böcker:

- *Solaris Common Desktop Environment: User's Guide*
- *Solaris Common Desktop Environment: User's Transition Guide*

Hur boken är upplagd

Handboken innehåller följande kapitel:

Kapitel 1 behandlar hur du konfigurerar utseendet och beteendet hos Inloggningshanteraren.

Kapitel 2 behandlar hur skrivbordet sparar/hämtar sessioner och hur du anpassar startsessionen.

Kapitel 3 beskriver startfiler i Solaris CDE, möjliga startproblem med Solaris CDE och förslag till lösningar för undvika startproblem.

Kapitel 4 behandlar hur programhanteraren hämtar program och förklarar hur du lägger till nya program.

Kapitel 5 beskriver hur man skapar ett registreringspaket för ett program.

Kapitel 6 riktar sig till mer avancerad konfiguration, t ex egna inloggningskonfigurationer, inställningar för flera fönster, skrivbord och Xterminal som ingår i ett nätverk, ändring av punktfiler för användaren, utskriftsinställningar för e-post, inställningar av skrivbordsmiljö och olika typer av felloggfiler.

Kapitel 7 beskriver hur du distribuerar tjänster, program och data från skrivbordet över ett nätverk.

Kapitel 8 behandlar hur du lägger till och tar bort skrivare och hur man anger en standardskrivare.

Kapitel 9 beskriver hur skrivbordet hittar program, hjälpfiler, ikoner och andra skrivbordsrelaterade data över nätverket.

Kapitel 10 introducerar nya koncept om funktioner och datatyper och om hur dessa förser användaren med gränssnitt gentemot program.

Kapitel 11 beskriver hur du använder programmet Skapa funktion för att skapa funktioner och datatyper.

Kapitel 12 beskriver hur du skapar funktionsdefinitioner genom att redigera konfigurationsfilen i en databas.

Kapitel 13 beskriver hur du skapar datatypsdefinitioner genom att redigera konfigurationsfilen i en databas.

Kapitel 14 beskriver hur du använder ikonredigeraren och hur man namnger, ändrar storlek och anger sökväg för ikoner för skrivbordet.

Kapitel 15 beskriver hur du skapar nya heltäckande kontroller och panelundermenyer även andra panelanpassningar.

Kapitel 16 beskriver hur du anpassar fönster, koppling av musknappar, koppling av tangentbordet och menyer i Arbetsytehanterare.

Kapitel 17 beskriver hur du ställer in programresurser, färger och fonter.

Kapitel 18 beskriver systemadministrationens uppgifter för system som kör internationella sessioner.

Bilaga A är en kopia av direkthjälpen (man) till dtconfig(1).

Beställa dokument till Sun

SunDocsm-programmet innehåller fler än 250 manualer från Sun Microsystems, Inc. Om du bor i USA, Kanada, Europa eller Japan kan du köpa dessa samlingsdokument eller enskilda manualer med hjälp av detta program.

Listan på dokument och hur du ska beställa dem hittar du i avsnittet i mappen från SunExpressTM på internet <http://www.sun.com/sunexpress>.

Typografiska ändringar

Följande tabell beskriver de typografiska ändringar som används i den här boken.

TABELL P-1 Typografiska konventioner

Teckensnitt eller Symbol	Betydelser	Exempel
AaBbCc123	Lista på kommandon, filer och mappar; utmatning på skärmen	Redigera din <code>.login</code> -fil. Använd <code>ls -a</code> för att lista alla filer. <code>maskinnamn% Du har fått brev.</code>
AaBbCc123	Vad du skriver i jämförelse med vad som matas ut på skärmen	<code>maskinnamn% su</code> <code>Password:</code>
AaBbCc123	Kommandoradens platshållare: ersätt med ett riktigt namn eller värde	Om du vill ta bort en fil, skriv <code>rm filnamn.</code>
AaBbCc123	Boktitlar, nya ord, nya termer eller framhävda ord	Läs kapitel 6 i <i>Användarhandbok</i> . De här kallas för <i>klassalternativ</i> . Du <i>måste</i> ha rotbehörighet för att utföra detta.

Ledtext i skalet för kommandoexempel

Följande tabell visar standardledtexten och superanvändarledtexten för C-skal, Bourne-skal och Korn-skal.

TABELL P-2 Ledtext i skalet

skal	Ledtext
Ledtext för C-skal	<code>maskinnamn%</code>
Ledtext för superanvändare med C-skal	<code>maskinnamn#</code>

TABELL P-2 Ledtext i skalet (forts.)

skal	Ledtext
Ledtext för Bourne-skal och Korn-skal	\$
Ledtext för superanvändare med Bourne-skal och Korn-skal	#

Konfigurera inloggningshanteraren

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, till en X-terminal eller till arbetsstation i nätverket.

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

- "Starta inloggningsservern" på sidan 2
- "Hantera lokala skärmar och nätverksskärmar" på sidan 2
- "Felsökning" på sidan 8
- "Inaktivera inloggningsservern" på sidan 9
- "Ändra utseendet på inloggningsbilden" på sidan 10
- "Administrera inloggningshanteraren" på sidan 18
- "Inloggningshanterarens filer" på sidan 19

Inloggningsservern:

- kan visa en inloggningsbild på bitmappskärmar utan begäran eller på begäran av bitmappskärmar lokalt eller i nätverk
- förser systemfönster med direktanslutna tecken
- kan visa en valskärm där användaren kan visa inloggningsbilder från andra inloggningsservrar i nätverket
- tillåter kontrollerad åtkomst 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 inloggningssbild. För nätverksskärmar (t ex X-terminaler) stöds X Display Manager Protocol (XDMCP) 1.0, med vilket skärmar kan begära att inloggningsservern visar en inloggningssbild.

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 kommer sedan att startas automatiskt när du startar om.

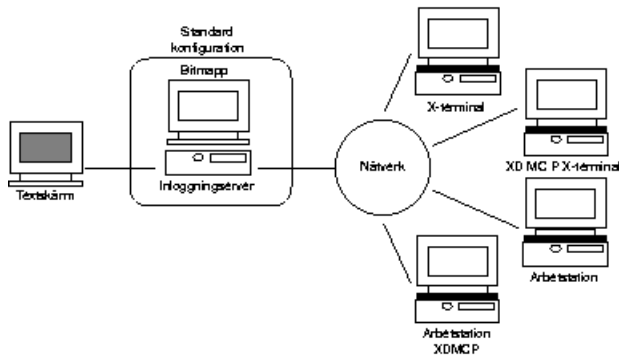
Mer information om skrivbordskonfigurationen `dtconfig` finns i **Bilaga A**. Där finns en kopia av direkthjälpen `(man)` för `dtconfig.1`.

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

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

Figur 1-1 visar en möjlig 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 filen ändras måste mappen finnas när inloggningsservern startar.

Om du vill ändra `Xconfig`, kopierar du `Xconfig` från `/usr/dt/config` och placerar den i `/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 kör kommandot `kill -HUP login_server_process_ID`.

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

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

När inloggningsservern startas om lagras dess process-ID i `/var/myserver/Dtpid`. Mappen `/var/myserver` 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 samt om och i så fall hur inloggningsbilder ska visas på lokala skärmar och nätverksskärmar.

Om du vill 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 kör kommandot `kill -HUP login_server_process_ID`

Formatet på raden för `Xservers` är:

```
skärnmamn skärmklass skärmtyp X-serverkommando
```

där

skärnmamn – meddelar inloggningsservern vilket anslutningsnamn som ska användas när du ansluter till X-servern (:0 i följande exempel). Värdet * (asterisk) utökas till *värddatornamn:0*. Det angivna numret måste matcha det nummer som anges i anslutningsnumret för *X-serverkommando*.

skärmklass – identifierar resurser som är anknutna till skärmen (`Local` i följande exempel).

skärmtyp – meddelar inloggningsservern om skärmen är en lokal skärm eller en nätverksskärm och hur alternativen för Kommandoradsinloggning i inloggningsfönstret (`local@console` i följande exempel) ska hanteras.

X-serverkommando – identifierar kommandoraden, anslutningsnummer och andra alternativ som inloggningsservern kommer använda för att starta X-servern (`/usr/bin/X11/X: 0` i följande exempel). Anslutningsnumret måste matcha det nummer som är angivet i *skärnmamn*.

Standardraden för `Xservers` ser ut ungefär så här:

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

Köra inloggningsservern utan lokal skärm

Om inloggningssystemet inte har någon bitmappskärm, kör du inloggningsservern utan lokal skärm genom att kommentera bort raden `Xservers` för lokal skärm med hjälp av tecknet # (pundtecken). Exempel:

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

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

Kommandoradsinloggning på lokala skärmar

När användaren väljer Kommandoradsinloggning på inloggningsbilden avbryts X-servern tillfälligt, vilket gör att den traditionella kommandoradsinloggningen på bitmappskärmen 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.

Skärmtypen kontrollerar hur Kommandoradsinloggning fungerar. Formatet för *skärmtyp* är:

- `local@skärmenhet`
- `local`
- `främmande`

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 kommer X-servern att avslutas och då ge återkomst till kommandoradsinloggning (`getty`) på `/dev/skärmenhet`.

Om du vill inaktivera alternativet Kommandoradsinloggning på en skärm anger du `none` som *skärmenhet*. Standardraden för *skärmenhet* är `console`. När `local` anges, ges *skärmenhet* standardvärdet `console`. När `foreign` anges kan Kommandoradsinloggning inte användas.

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

Anpassa textskärmen

Om inloggningsserversystemet har en direktansluten textskärm som fungerar som en konsol, kan du ange *skärmenhet* till `none` för att inaktivera Kommandoradsinloggning på bitmappskärmen.

Om en kommandoradsinloggning (`getty`) körs på både textskärmen och bitmappskärmen kan du ändra *skärmenhet* till enheten för kommandoradsinloggning (`getty`) på bitmappskärmen.

Exempel: Om bitmappskärmens kommandoradsinloggning (`getty`) finns på enheten `/dev/tty01` ändrar du *skärmtyp* till `local@tty01`.

Visa en inloggningsbild på en nätverksskärm

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

Om du ska hantera förfrågningar från nätverksskärmen måste inloggningsservern stödja XDMCP (X Display Manager Protocol) 1.0. Detta protokoll gör att

inloggningsservern kan förhandla med nätverksskärmen genom att acceptera eller avslå förfrågningar. XDMCP finns inbyggt i de flesta X-terminaler.

XDMCP direktförfrågningar från nätverksskärmar

När du konfigurerar X-terminalen för att använda XDMCP direkt (frågeläge), anger du namnet på inloggningssvården. 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 inloggningsserverns värddator direkt med en begäran om att inloggningsbilden ska visas på X-servern. Om du startar X-servern på en bitmappskärm på arbetsstationen `birgitta`, visar inloggningsservern `anita` en inloggningsbild på X-servern:

```
X -query anita
```

XDMCP, indirekta förfrågningar från nätverksskärmar

När du konfigurerar X-terminalen att använda XDMCP indirekt läge, anger du namnet på inloggningssvården. När X-terminaler startas kontaktas inloggningsservern. Denna visar, via en skärm, en lista med andra inloggningssvårder 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--indirekt`, med vilket X-servern kontaktar inloggningsservern i XDMCP indirekt läge.

Hantera nätverksskärmar utan XDMCP

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

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

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
```

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 kör kommandot `kill -HUP inloggningsserverns process-ID`.

XDMCP Direct

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. Alla `Xaccess`-poster är värddatornamn, inklusive jokertecknen `*` (asterisk) och `?` (frågetecken). En `*` (asterisk) motsvarar noll eller flera tecken och ett `?` (frågetecken) motsvarar ett enstaka valfritt tecken. Ett `!` (utropstecken) före posten innebär att värddatorn inte har tillgång till servern, medan om det inte finns något utropstecken har värddatorn tillgång till servern.

`Xaccess`, t ex, innehåller följande tre poster:

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

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

XDMCP Indirect

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, inklusive jokertecken. Undantaget är att alla poster markeras med strängen CHOOSEER. Exempel:

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

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

Något av följande steg visas efter CHOOSEER:

- BROADCAST
- *lista över värddatornamn*

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

```
amazon.waterloo.com CHOOSEER shoal.waterloo.com alum.waterloo.com
*.dept5.waterloo.com CHOOSEER BROADCAST
!* CHOOSEER 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 inloggningssvärdar 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 CHOOSEER %list1
```

Felsökning

Normalt loggar inloggningsservern fel i filen `/var/dt/Xerrors`. Om du vill ändra detta anger du resursen `Dtlogin.errorLogFile` i filen `Xconfig`. Angiven mapp måste finnas när inloggningsservern startar.

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

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

När du startar om inloggningsservern loggas fel i filen `/var/mylogs/Derrors`. Mappen `/var/mylogs` måste finnas när inloggningsservern startar.

Inaktivera inloggningsservern

- Om du vill inaktivera startprocessen för inloggningsservern när du startar om datorn, skriver du:

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

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

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

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

Detta kör kommandot (`kill login_server_process_ID`)

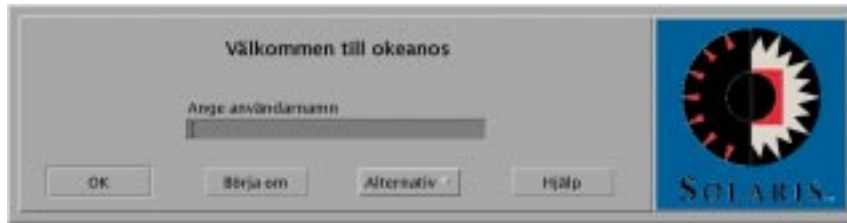
Obs! Om du tar bort inloggningsserverns process avbryts alla användarsessioner som hanteras av inloggningsservern.

Du kan också avbryta inloggningsservern genom att använda kommandot `kill` för att avsluta processen. Inloggningsserverns process-ID lagras i `/var/dt/Xpid` eller i den fil som anges i `Xconfig` av resursen `Dtlogin.pidFile`.

Om du har loggat in på skrivbordet samtidigt som du har använt programmet `kill` för att avbryta inloggningsservern, kommer skrivbordssessionen att avslutas omedelbart.

Inloggningsskärmen

Inloggningsskärmen som inloggningsservern visar är ett attraktivt alternativ till den traditionella teckenbaserade inloggningsskärmen, och den 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 startas 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 här användarens session har avslutats

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

Ändra utseendet på inloggningsbilden

Du kan anpassa inloggningsbilden genom att ändra logotypen, grafiken, välkomstmeddelandet och teckensnitten.

Om du vill ändra Xresources, kopierar Xresources från `/usr/dt/config/språk` till `/etc/dt/config/språk`. Inloggningsskärmen kommer visa alla ändringar nästa gång som inloggningsskärmen visas. 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 lägg till:

`Dtlogin*logo*bitmapFile` - en bitmapfil eller pixmapfil kan visas som logobild

`Dtlogin*greeting*persLabelString` - personligt välkomstmeddelande

Dtlogin*greeting*labelString - välkomstmeddelande
Dtlogin*greeting*fontList - Teckensnitt för välkomstmeddelande
Dtlogin*labelFont - Teckensnitt för knappar och namn
Dtlogin*textFont - Teckensnitt för hjälp- och felmeddelanden
Dtlogin**språk**languageName - Alternativ text för lokalt *språk*

▼ Ändra logotyp

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

Logotypen kan vara en färgad bildpunktsmönsterfil eller en bitmapsfil.

I följande exempel används bitmappen Mylogo som logo:

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

▼ Ändra välkomstmeddelandet

Normalt visas meddelandet Välkommen till *värddatornamn* på inloggningsskärmen. Så här kan du ändra meddelandet:

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

Värdet på resursen labelString innehåller %%LocalHost%{>%, vilket ersätts med inloggningsserverns värddatornamn och %DisplayName%, som i sin tur ersätts med X-serverns skärmmamn.

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* som standard. Du kan ändra meddelande genom att ändra resursen Dtlogin*greeting*persLabelString i Xresources. Värdet på resursen persLabelString kan innehålla %s, vilket ersätts med *användarnamn*.

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

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

▼ Ändra teckensnitt

Du ändrar teckensnittet som används i inloggningsskärmen genom att ange följande resurs i Xresources:

Dtlogin*greeting*fontList - teckensnitt för välkomstmeddelandet

Dtlogin*labelFont - teckensnitt för knappar och namn

Dtlogin*textFont - teckensnitt för hjälp- och felmeddelanden

Om du vill se en lista över tillgängliga teckensnitt skriver du:

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

I följande exempel används ett stort teckensnitt till välkomstmeddelandet (värdet du anger måste rymmas på en rad):

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

▼ Ange 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 resursens namn

Dtlogin**språk**languageName i resursen 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 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*)

Resurser som anges i filen Xconfig innehåller:

Dtlogin*authorize - Xaccess filspecifikation

Dtlogin*environment - X-server miljö

Dtlogin*language - standardspråk

Dtlogin*languageList - lista på språk i Språk-menyn på inloggningsskärmen

Dtlogin*resources - Xresources specifikation

Dtlogin*setup - Xsetup filspecifikation

Dtlogin*startup - Xstartup filspecifikation

Dtlogin*session - Xsession filspecifikation
Dtlogin*failsafeClient - Xfailsafe skriptspecifikation
Dtlogin*reset - Xreset skriptspecifikation
Dtlogin*userPath - PATH för Xsession och Xfailsafe
Dtlogin*systemPath - PATH för Xsetup, Xstartupoch Xfailsafe
Dtlogin*systemShell - SHELL för Xsetup, Xstartupoch Xfailsafe
Dtlogin.timeZone - TZ för alla skript

Ändra hur inloggningsskärmen fungerar skärmvis

I exemplet nedan påverkar ändringar av resursen Xconfig beteendet i inloggningsskärmen. Resurser som visas med en asterisk (*) kan anges per skärm. Detta gör att du kan ange anpassade inloggningsskärmar för vissa skärmar. Om du vill ange en resurs för en viss skärm, anges resursen som Dtlogin**skärmanamn**. Om du t ex vill stänga av användarbaserad behörighetskontroll för skärm expo:0 men vill ha det aktiverat för andra skärmar, skriver du:

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

Obs! Alla specialtecken i skärmanamnet som : (kolon) eller . (punkt), byts ut mot ett _ (understräck).

Ändra X-serverbehörighet

Inloggningsserven gör det möjligt med åtkomstkontroll i X-servern, utgående ifrån den informationen som sparas i filen *Hemmapp/.Xauthority*. Endast användare som har tillgång till den här filen har också tillgång till X-server. Detta är den vanligaste metoden för att kontrollera 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. Värddatorbaserad kontroll används när:

- Äldre R2- och R3 X-klienter inte kan ansluta till en X-server med användarbaserad behörighetskontroll.
- Obehöriga personer 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
```

Ändra 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
```

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

Ändra standardspråk

När användaren loggar in på 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.

Ändra 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` när användarsessionen avslutas.

Starta kommandon innan användarsessionen

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` när användarsessionen startas.

Starta en skrivbordssession

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

Obs! Uppdatera inte skriptet `Xsession` direkt.

Se Kapitel 2 för mer information om hur du ändrar användarens startsession på skrivbordet.

Starta en Felsäker session

Om användaren väljer Felsäker session på undermenyn Sessioner på inloggningsbildens Alternativ-meny, körs skriptet `Xfailsafe`. `Xfailsafe` körs med användarbehörighet och startar de kommandon som behövs för att starta en minimal fönstermiljö, oftast 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 det ändrade `Xfailsafe` när användarsessionen avslutas.

När användarsessionen avslutas

När användaren avslutar skrivbordet, eller Felsäker session, 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` när användarsessionen avslutas.

Inloggningsserverns miljö

Inloggningsservern omfattar en miljö som exporteras till skripten `Xsetup`, `Xstartup`, `Xsession`, `Xfailsafe` och `Xreset`. Mer information om de här miljöerna finns i Tabell 1-1. Ytterligare variabler kan också exporteras av inloggningsservern.

TABELL 1-1 Inloggningsserverns miljöer

Miljö Variabel	Xsession				Beskrivning
	Xsetup	Xstartup	Xfailsafe	Xreset	
LANG	X	X	X	X	Standard eller valt språk
XAUTHORITY	X	X	X	X	Alternativ X-behörighetsfil
PATH	X	X	X	X	Värdet på Dtlogin*userPath- resursen (Xsession, Xfailsafe) eller Dtlogin*systemPath- resursen (Xsetup, Xstartup, Xreset)
DISPLAY	X	X	X	X	X-server- anslutningsnummer
SHELL	X	X	X	X	Skal som anges i <code>/etc/ passwd</code> (Xsession, Xfailsafe) eller i resursen Dtlogin*systemShell (Xsetup, Xstartup, Xreset)

TABELL 1-1 Inloggningsserverns miljöer (forts.)

Miljö Variabel	Xsession				Beskrivning
	Xsetup	Xstartup	Xfailsafe	Xreset	
TZ	X	X	X	X	Värdet av resursen eller tidszonen Dtlogin.timeZone bestäms av systemet
USER		X	X	X	Användarnamn
HOME		X	X	X	Hemmappen anges i /etc/passwd
LOGNAME		X	X	X	Användarnamn

Ändra användar- eller systemsökvägen

Inloggningsservern anger miljövariabeln PATH när skripten Xsession och Xfailsafe körs. Du kan bifoga en alternativ sökväg för dessa skript.

Ändra användarsökväg

- ♦ **Ändra resursen** Dtlogin*userPath i Xconfig. **Exempel:**

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

Ändra 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
```

▼ Ändra tidszon

Inloggningsservern anger miljövariabeln TZ när skripten Xsetup, Xstartup, Xsession, Xfailsafe och Xreset körs. Standardvärdet kommer från systemet så du behöver oftast inte ändra dess upplägg. 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, startar också processen dtlogin. Processen dtlogin 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 den läser också filen Xaccess för att kontrollera behörighet till inloggningsservern.

Om inloggningsservern får information från filen Xservers om 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 om att den måste hantera en nätverksskärm, antas det att X-servern redan körs med angivet skärmanamn 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 för den skärmen. Detta betyder att om inloggningsservern hanterar n skärmar, finns det $n+1$ dtlogin-processer. Inloggningsservern kommer köra skriptet Xsetup, ladda filen Xresources och sedan köra 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 direkt. När användaren har avslutat sessionen kommer inloggningsservern att köra skriptet Xreset.

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

För konfigurationsfilerna `Xaccess`, `Xconfig`, `Xfailsafe`, `Xreset`, *språk*/`Xresources`, `Xservers`, `Xsetup` och `Xstartup` kommer inloggningsservern som standard att först öppna `/etc/dt/config/`, sedan `/usr/dt/config/` och använda 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 inloggningsskärmbild för en skärm
- `/usr/dt/bin/dtchooser` – visar en valskärm för en skärm
- `/usr/dt/bin/Xsession` – startar en skrivbords-session
- `/usr/dt/config/Xfailsafe` – startar en session med endast ett fönster
- `/usr/dt/config/Xconfig` – konfigurationsfiler för inloggningsservern
- `/usr/dt/config/Xservers` – inloggningsserver, skärmbeskrivningsfil
- `/usr/dt/config/Xaccess` – inloggningsserver, åtkomstbeskrivningsfil
- `/usr/dt/config/språk/Xresources` – visar utformningsresurser
- `/usr/dt/config/Xsetup` – visar inställningsfilen
- `/usr/dt/config/Xstartup` – för-sessionens startfil
- `/usr/dt/config/Xreset` – efter-sessionens återställningsfil
- `/var/dt/Xpid` – process-ID för inloggningsservern
- `/var/dt/Xerrors` – felloggningsfil för inloggningsservern

Konfigurera sessionshanteraren

Sessionshanterarens uppgift är att starta skrivbordet och automatiskt spara och återställa färger, teckensnitt, musfunktioner, ljudvolym och tangentbordsklickningar samt program som körs.

Här följer en lista över funktioner i sessionshanteraren och avsnitt i detta kapitel:

- 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?" på sidan 21
- "Starta en session" på sidan 23
- "När sessionen startar" på sidan 23
- "Ytterligare anpassningar av sessionsstarten" på sidan 29
- "Filer och mappar för sessionshanteraren" på sidan 34

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 en användarsession kan sparas och återställas. En användare kan logga in i systemet och ta del av samma program, inställningar och resurser som fanns när användaren loggade ut. När användaren loggar in på skrivbordet för första

gången kommer en standardsession att laddas. Därefter används begreppen aktiv session och hemsession.

Startsession

När en användare loggar in för första gången kommer sessionshanteraren att skapa användarens startsession med 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 den aktiva sessionen, baserat på användarens Startinställningar i Inställningar, när användaren avslutar sessionen. När användaren loggar in nästa gång startas 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 Inställningar, 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 mappen *Hemmapp/.dt/sessions* till *Hemmapp/.dt/-skärm*, där *skärm* är det verkliga, ogiltiga värddatornamnet (`pablo:0` är giltigt, `pablo.gato.com:0` eller `unix:0` är det inte). När användaren loggar in på skärmen `pablo:0`, kommer Sessionshanteraren att starta de skärmspecifika sessionerna.

Starta en session

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

Användaren kan också logga in med den traditionella teckenbaserade inloggningen (`getty`) och starta sessionshanterare manuellt med verktygen som startar X-hanteraren, `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 `Hemmapp/.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) `Hemmapp/.profile` eller `Hemmapp.login`**
7. **Startar ToolTalk[®] meddelande-daemon**
8. **Laddar sessionsresurser**
9. **Startar färghanteraren**
10. **Startar arbetsytahanteraren**
11. **Startar sessionens program**

Följande avsnitt beskriver stegen ovanför.

Läsa in och köra skriptet .dtprofile

När sessionen startar, körs skriptet `Xsession`, som läser in och kör användarens `Hemmapp/.dtprofile`-skript. `Hemmapp/.dtprofile`-skript är ett `/bin/sh` eller `/bin/ksh`-skript som gör att användaren kan ange miljövariabler för sessionen. Mer information om hur du anger miljövariabler finns i "Ytterligare anpassningar av sessionsstarten" på sidan 29.

Om `Hemmapp/.dtprofile`-skriptet inte finns när användaren loggar in på skrivbordet för första gången kommer `Xsession` att kopiera skrivbordsstandarden `sys.dtprofile` till `Hemmapp/.dtprofile`.

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

Läsa in och köra skriptet Xsession.d

Efter att ha läst in och kört skriptet `Hemmapp/.dtprofile` kommer skriptet `Xsession` att läsa in och köra skriptet `Xsession.d`. Detta skript används för att allokera miljövariabler och starta eventuella bakgrundsprogram för användarens session. Standardbeskrivningen för `Xsession.d`-skript är:

0010.dtpaths – anpassningsbara dokument sökvägar för skrivbordet

0020.dtims – startar valfri metodserver för indata

0030.dttmpdir – skapar en temporär mapp per användare och session

0040.xmbind – ställer in `$XMBINDDIR` till standardvärde för skrivbordet

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

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

Skrivbordets standardskript `Xsession.d` finns i mappen `/usr/dt/config/Xsession.d`. Om du vill anpassa ett `Xsession.d`-skript, kopierar du skriptet 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 eget skript, kopierar du ditt eget skript 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 ha läst in och kört skripten *Hemmapp/.dtprofile* och *Xsession.d* kommer *Xsession* visa ett välkomstmeddelande på skärmen. Du kan ändra välkomstmeddelandet eller stänga av det helt och hållet. Klienten *dthello* används för att visa meddelanden.

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 ett */etc/dt/config/Xsession.d*-skript som anger det nya värdet. Om du vill visa dagens meddelande för alla användare, skapar du ett körbart *sh* eller *ksh*-skript, t ex */etc/dt/config/Xsession.d/minavar* och ställer in *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 *Hemmapp/.dtprofile*.

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

Mer information om *dthello* finns i direkthjälpen (man) på *dthello*.

Ställa in skrivbordets sökvägar

Skrivbordssökvägar skapas vid inloggning av *dtsearchpath*. Det finns två kategorier av miljövariabler som används av *dtsearchpath*:

Inmatningsvariabel – Systemtäckande och egna miljövariabler vars värden sätts av systemadministratören eller av slutanvändaren.

Utmatningsvariabel – Variabler skapade och tilldelade med värden från *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 körbart *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*-alternativ för sina egna sessioner genom att lägga *dtstart_searchpath* i *Hemmapp/.dtprofile*.

Mer information om *dtsearchpath* finns i Kapitel 9. Mer information om *dtsearchpath* alternativ, finns i direkthjälpen (man) på *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 kommandoradsalternativ för `dtappgather` ändrar du variabeln `dtstart_appgather`. Om du vill ändra variabeln `dtstart_appgather` för alla användare, skapar du ett körbart `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 lägga `dtstart_appgather` i *Hemmapp*/.`dtprofile`.

Mer information om `dtappgather`-alternativ finns i direkthjälpen (`man`) på `dtappgather(4)`.

Läsa in och köra skripten `.profile` och `.login` (valfritt)

`Xsession` kan läsa in och köra användarnas traditionella *Hemmapp*/.`profile` eller *Hemmapp*/.`login`-skript. Som standard gäller inte de här skripten. Om du vill göra så att `Xsession` läser in och kör skripten `.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 ställer in det nya värdet. Om du vill ange `DTSOURCEPROFILE` till `true` för alla användare, skapar du ett körbart `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 *Hemmapp*/.`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 körbart `sh` eller `ksh`-skript (t ex `/etc/dt/config/Xsession.d/minavar`) 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 *Hemmapp/.dtprofile*.

Mer information om `ttsession`-alternativet finns i direkthjälpen (man) `ttsession`. Mer information om `ttsession` finns i *Common Desktop Environment: ToolTalk Messaging Overview*.

Starta sessionshanterarklienten

Här startar `Xsession /usr/dt/bin/dtsession`, som innehåller sessionens startprocess.

Ladda sessionsresurser

Sessionshanteraren använder `X-serverRESOURCE_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 bör inte redigeras eftersom den skrivs över vid varje skrivbordsinstallation.

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 som anges i den här filen har företräde framför de som anges i resursfilen för standardskrivbordet.

Användare kan utvidga skrivbordets standardresurser och systemtäckande resurser med filen *Hemmapp/.Xdefaults* fil. Resurser som anges i den här filen blir tillgängliga för denna användares session via egenskapen `RESOURCE_MANAGER`. Resurser som anges i den här filen har företräde framför de som anges i resursfiler för standardskrivbordet eller resursfiler från systemadministratörer.

Obs! Hjälpprogrammet X Toolkit Intrinsic anger att det laddar resurser för ett program från antingen `RESOURCE_MANAGER` eller från `Hemmapp/.Xdefaults`, men inte från båda. Vanligtvis innebär detta att användarens `Hemmapp/.Xdefaults`-fil ignoreras. Men sessionshanteraren kan innehålla `Hemmapp/.Xdefaults` genom att den slås samman med `RESOURCE_MANAGER` när sessionen startar, vilket beskrivs ovan. Om en användare ändrar `Hemmapp/.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 användarspecifiserade resurser. På detta sätt görs ändringar av systemtäckande och egna resursfiler tillgängliga för alla program.

Mer information finns i

-
- Direkthjälpen (man) för `dtresourcesfile(4)`

Starta färgservern

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

`foregroundColor` – kontrollerar om en bildpunkt finns tillgänglig för förgrundsfärgen

`dynamicColor` – anger om färger som inte kan redigeras finns tillgängliga

`shadowPixmap` – anger om färger ska ha en övre eller undre skugga

`colorUse` – begränsar färgintervall

`writeXrdbColors` – anger om `*background` eller `*foreground` placeras i resursdatabasen

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

Användarna kan på ett liknande sätt ange färghanteraresurser för sina egna sessioner genom att ange sådana resurser i `Hemmapp/.Xdefaults`.

Mer information om hur du anger färghanteraresurser finns i "Hantera färger" på sidan 276.

Starta arbetsytehanteraren

Sessionshanteraren 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 *Hemmapp*/`.Xdefaults`.

Mer information om fönsterhanterare finns i Kapitel 16.

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 initiala 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 direkthjälpen `(man) dtactionfile(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

▼ Ange miljövariabler

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

Om du t ex skapar ett körbart 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 **Hemmapp**/`.dtprofile`.

Exempel:

```
export MINVARIABEL="värde"
```

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

Obs! Sessionshanteraren läser inte automatiskt filerna `.profile` eller `.login`. Du kan konfigurera om den så att dessa filer kan användas; se .

▼ Ange resurser

- ◆ Om du vill ange systemtäckande resurser lägger du till resurserna i filen `/etc/dt/config/språk/sys.resources`. (Först måste du skapa filen.)

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

Om du 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.

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

▼ Ange 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 *Hemmapp*/`.Xdefaults`.**

Du avgränsar resurser genom att omsluta dem med villkorsatsen `cpp` villkorlig programsats. Makrot `DISPLAY_skrärmnamn` 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 `$DISPLAY` av `:0` visar `DISPLAY_0`, och `$DISPLAY` av `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 du:

```
Myapp*resource: värde

#ifdef DISPLAY_blanco_gato_com_0
    Myapp*resource: specialvalue1
#endif

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

Ange resursen `MyApp*resource` i `RESOURCE_MANAGER` till `specialvalue1` när användaren loggar in på skärmen `blanco.gato.com:0`; `specialvalue2` när användaren loggar in på `pablo.gato.com:0`; och `värde` när användaren loggar in på en annan skärm.

▼ Ändra 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 command_and_options
```

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.session`:

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

▼ Ange en skärmspecifik session

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

- ♦ **Kopiera mappen `Hemmapp/.dt/sessions` till `Hemmapp/.dt/skärm` där `skärm` är det verkliga, absoluta värddatornamnet (`pablo:0` är giltigt, `pablo.gato.com:0` eller `unix:0` är det inte).**

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

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

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

Köra ytterligare kommandon vid sessionsstart och utloggning

Användarna kan ange att ytterligare kommandon ska startas när de loggar in till skrivbordsessionen. 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.

▼ Köra ytterligare kommandon vid sessionsstart

- ◆ Skapa filen *Hemmapp/.dt/sessions/sessionetc* som innehåller kommandona.

Normalt är den här filen ett skript som 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.

▼ Köra ytterligare kommandon vid utloggning

En stödfil för *sessionetc* är *sessionexit*. Använd *sessionexit* för att utföra vissa bearbetningar när sessionen avslutas som inte hanteras av sessionshanteraren.

- ◆ Skapa filen *Hemmapp/.dt/sessions/sessionexit* som innehåller kommandona.

Liksom *sessionetc* är den här filen normalt ett skript med exekveringsbehörighet.

▼ Återskapa en session från säkerhetskopia

När sessionshanteraren sparar en session lagras sessionens information i mappen *Hemmapp/.dt/sessions*, eller i mappen *Hemmapp/.dt/skärm* om en skärmspecifik session används. I dessa mappar skapas en undermapp med namnet *current* eller *Home* som lagrar information för den aktiva sessionen eller hemsessionen. Innan sessionsinformationen lagras, skapar sessionshanteraren en säkerhetskopia av den tidigare sessionen med det namnet och lagrar den i *current.old* eller *home.old*.

1. Logga in med Felsäker session eller Kommandorad från inloggningsfönstret.
2. Kopiera mappen för den säkerhetskopierade sessionen till aktivt namn, t ex till säkerhetskopierad hemsession:

```
cp -r Hemmapp/.dt/sessions/home.old \  
  Hemmapp/.dt/sessions/home
```

En skärmspecifik session kan återskapas på samma sätt.

▼ Undersöka problem vid sessionsstart

- ◆ Undersök den här filen *Hemmapp/.dt/startlog*.

Sessionshanteraren loggar varje användares startsession i den här filen.

Filer och mappar för sessionshanteraren

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

Felsökning för inloggning och problem vid sessionstart

I det här kapitlet finns flera avsnitt med information om Solaris CDEs startfiler, möjliga problem vid start av Solaris CDE och förslag på lösningar av startproblem.

- "Startfiler för inloggning" på sidan 35
- "Felloggningskatalog" på sidan 36
- "Användarstartfil" på sidan 37
- "Solaris CDE, startexempel" på sidan 37

Startfiler för inloggning

Efter att Solaris CDE Inloggningshanterare har verifierat användaren, anropar den följande startskript för att starta skrivbordet:

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

Den första användarspecifika fil som `Xsession` anropar är *Hemmapp/.dtprofile*.

Första gången en ny användare loggar in i Solaris CDE, kopieras filen `.dtprofile` till användarens hemkatalog. Filen är förinställd att inte utföra någonting. Den innehåller däremot ett antal kommentarer om på vilka sätt den kan redigeras. Användaren kan redigera den här filen och lägga till användarspecifika miljövariabler.

Obs! Precis som `Xsession`-skriptet som anropar `.dtprofile`, använder den här filen ett `ksh`-syntax.

En användbar redigeringsfunktion är att avkommentera sista raden i `.dtprofile`-filen:

```
DTSOURCEPROFILE=true
```

Den här raden anger användarens *Hemmapp*/`.login` (för `ssh`-användare) eller *Hemmapp*/`.profile` (för övriga skal-användare) som startkatalog vid startprocessen.

Felloggningskatalog

Välj Felsäker session på Alternativ-menyn i Solaris CDE inloggningsbild om du vill kringgå den normala Xsession-starten och undersöka felloggningskataloger samt åtgärda problem med användarpunktfiler. Tabell 3-1 visar felloggningsfiler och deras kataloger.

TABELL 3-1 Felloggningskataloger

Katalog	Felloggningsfiler
<code>/var/dt/Xerrors</code>	Solaris CDE inloggningsfönster systemfel med tidigare ursprung än användarens inloggning
<i>Hemmapp</i> / <code>.dt/startlog</code>	Solaris CDE startfel under Xsession, <code>.dtprofile</code> , <code>.login</code> eller <code>.profile</code>
<i>Hemmapp</i> / <code>.dt/errorlog</code>	Solaris CDE startfel efter Xsession start
<i>Hemmapp</i> / <code>.dt/sessionlogs</code>	Kataloger för sessionsloggning av fel i Sessionshanteraren och Fönsterhanteraren

Vissa fel visas även i Systemfönstret. Om inte Systemfönstret är aktiverat är namnet på systemets reservloggningfil `wscn` med tid och visningskoden sammanlänkad. Till exempel:

```
/usr/tmp/wscnAAAA004EE:0.0
```

Användarstartfil

Om du vill ange `.login` (för `csh`-användare) eller `.profile` (för `sh`- eller `ksh`-användare) som källfiler, avkommenterar du den sista raden i filen `.dtprofile` till:

```
DTSOURCEPROFILE=true
```

I normala fall är detta den enda åtgärd som behövs. I vissa fall behöver du ändra i `.login` eller `.profile`-filerna eftersom de kan innehålla kommandon som inte är kompatibla med Solaris CDE:s inloggningshanterare. Om det finns ett fel i något kommando kan det bero på att filen förväntar sig en indatakälla från terminalens tangentbord, t ex `stty`, `tset` eller ett "vänta på indata"-kommando.



Varning! Om `.dtprofile` är inställt för att ange en `.login` eller `.profile`-fil som källa, och den innehåller felkommandon som kraschar skalet, kommer försöket att starta skrivbordet att misslyckas. Skrivbordet kommer inte att visas. Istället kommer Solaris CDE:s inloggningsbild att visas på nytt. Startfel från `.login` eller `.profile` visas i *Hemmapp*/`.dt/startlog`. Använd en felsäker inloggnings-session eller kommandoradsinloggning för att söka efter felkommandon i `.login` eller `.profile`.

Om du behöver mer information om olika problem och deras lösningar läser du filen `.dtprofile`. Normalt sett är felaktiga kommandon kopplade till terminalinformation och styrning.

Solaris CDE, startexempel

Det här avsnittet innehåller förslag på hur du kan redigera följande användarstartfiler:

- `.login` (för `csh` användare)
- `.profile` (för `sh` eller `ksh` användare)
- `.xdefaults`

Solaris CDE:s `startprocess` anger en skalvariabel som heter `DT` i `.login` eller `.profile`-skripten så att du kan genomföra en kontroll under Solaris CDE:s `startprocess`. Detta förhindrar att terminalrelaterade kommandon som t ex `stty` och `stty` utförs. Hänvisa till följande skriptexempel:

```
.login (C shell)
```

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

.profile (sh or ksh)

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

Denna extra syntax runt DT stoppar varningsmeddelanden från C shell när DT inte är definierad. DT är inte definierad när användaren loggar in från en textbaserad inloggningsledning i systemfönstret. Se *Hemmapp*/.dtprofile-filen om vill ha mer information om hur du ställer in punktfiler.

Obs! Du anger Solaris CDE tty-inställningar i .Xdefaults.

Användarens .Xdefaults-fil används som källfil för användarangivna resurser under Solaris CDE:s start. Till exempel anger, ttyModes att användarens favoritinställningar för tty används i terminalemulerings- fönster som dtterm och xterm. Följande rad visar en typisk ttyModes-inställning i filen .Xdefaults:

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

Obs! Favoritresurser för Solaris CDE skiljer sig från standardinställningarna.

Fönsterhanteraren i Skrivbordets placering av ikoner är ett exempel på en standardinställning. Följande rad i filen .Xdefaults visar standardinställningen för ikonplacering:

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

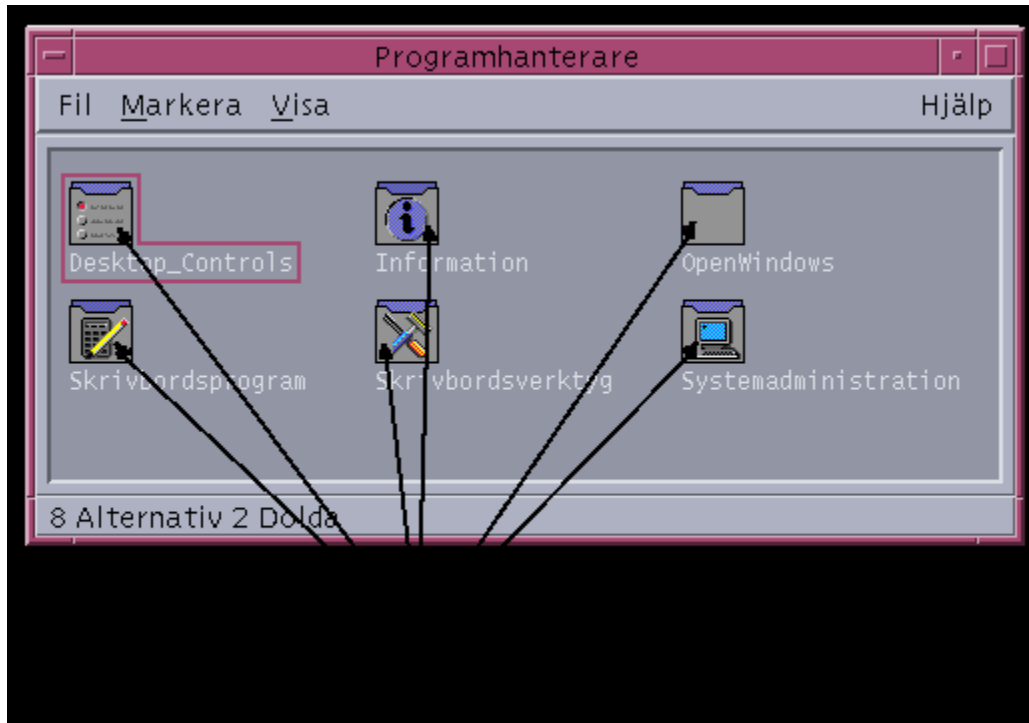

Lägga till och hantera program

Programhanteraren är den funktion på skrivbordet som innehåller program som är tillgängliga för användaren.

- "Programhanterarens struktur" på sidan 39
- "Lägga till program i programhanteraren" på sidan 43
- "Skapa och hantera allmänna programgrupper" på sidan 46
- "Ändra sökvägen som används för att söka efter program" på sidan 47
- "Allmän administration av Programhanteraren" på sidan 49
- "Ändra textredigeraren och terminalemulatorn" på sidan 50

Programhanterarens struktur

Programhanterarens översta nivå innehåller normalt mappar. Var och en av dessa mappar, med innehåll, kallas för en *programgrupp*.



Figur 4-1 Programgrupper i programhanteraren

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

Programhanterarens plats i filsystemet

I filsystemet ligger programhanteraren i mappen

`/var/dt/appconfig/appmanager/inloggningsnamn-värdatornamn-skärm`. Mappen skapas dynamiskt varje gång användaren loggar in.

Om användaren `karinj` t ex loggar in från skärmen `wxyz:0`, skapas mappen

`/var/dt/appconfig/appmanager/karinj-wxyz-0`.

Hur programhanteraren hittar och samlar ihop program

Programhanteraren utgörs av insamlade lokala programgrupper och programgrupper i nätverket. Programgrupperna samlas in från mappar i programsökvägen.

Standardsökvägen för program består av platserna i Tabell 4-1.

TABELL 4-1 Standardsökväg för program

Definitionsområde	Plats
Inbyggd	<code>/usr/dt/appconfig/appmanager/språk</code>
Systemtäckande	<code>/etc/dt/appconfig/appmanager/språk</code>
Egen	<i>Hemmapp</i> / <code>.dt/appmanager</code>

Vid inloggningen skapas programhanterarens översta nivå genom att länkar skapas från programgrupperna (mappar) i programsökvägens mappar, till programhanterarens mapp

`/var/dt/appconfig/appmanager/inloggningsnamn-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/Skrivbordsverktyg`

Vid inloggningen skapas en symbolisk länk till:

`/var/dt/appconfig/appmanager/inloggningsnamn-värddatornamn-skärm/Skrivbordsverktyg`

Programsökvägen kan omfatta mappar på nätverket. På detta sätt kan programgrupper samlas in från olika system i nätverket. Mer information finns i "Lägga till en programserver i programsökvägen" på sidan 48.

Prioritetsregler för insamling av program

Vid dubbleringar 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/sv/Skrivbordsverktyg` och `/etc/dt/appconfig/appmanager/sv/Skrivbordsverktyg` finns, används programgruppen under `/etc`.

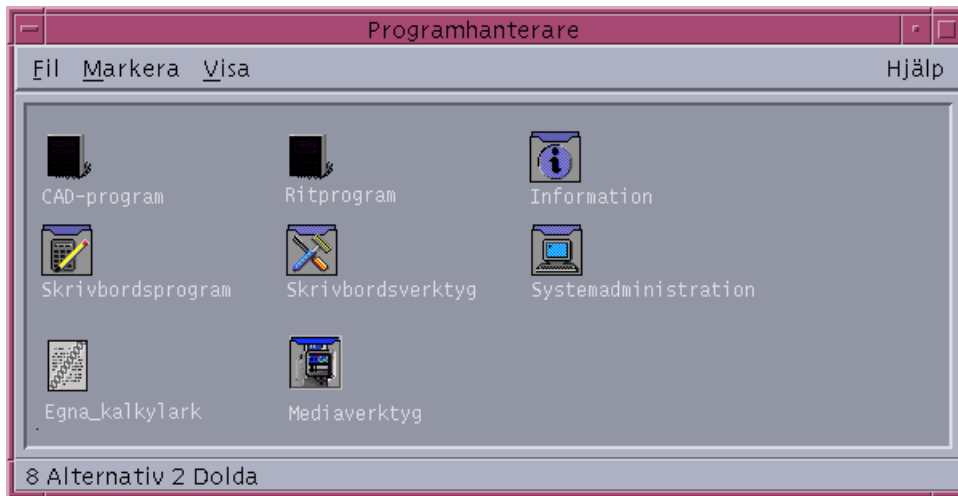
Programgrupper på standardskrivbordet

På ett skrivbord som inte är anpassat finns fyra programgrupper.

- Skrivbordsprogram
- Skrivbordsverktyg
- Information
- Systemadministration

Exempel på hur programgrupper samlas in

Figur 4-2 visar ett fönster i programhanteraren som innehåller flera olika programgrupper. Tabell 4-2 visar de mappar som programgrupperna samlades in från.



Figur 4-2 Ett typiskt programhanterarfönster

TABELL 4-2 Källor till programgrupperna i Figur 4-2

Namn	Insamlad mapp
CAD-program	/net/ApServA/etc/dt/appconfig/appmanager/C/CAD_App
Ritprogram	/etc/dt/appconfig/appmanager/C/DrawingApp
Skrivbordsprogram	/usr/dt/appconfig/appmanager/C/Desktop_Apps

TABELL P-2 Källor till programgrupperna i Figur 4-2 (forts.)

Namn	Insamlad mapp
Skrivbordsverktyg	/usr/dt/appconfig/appmanager/C/Desktop_Tools
Information	/usr/dt/appconfig/appmanager/C/Information
Systemadministration	/etc/dt/appconfig/appmanager/C/System_Admin
Egna_kalkylblad	/users/anna/.dt/appmanager/MySpreadSheet
Mediaverktyg	/etc/dt/appconfig/appmanager/C/Media_Tools

Om programgrupperna Information eller Systemadministration har anpassats, kommer de i stället att samlas in från /etc/dt/appconfig/appmanager/C.

Gruppen CAD-program samlas in eftersom systemet ApServA har lagts till programsökvägen (se "Lägga till en programserver i programsökvägen" på sidan 48). Egna_kalkylblad är en personlig programgrupp som bara är tillgänglig för användaren anna.

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 mapp på Programhanterarens översta nivå som innehåller programikon och andra filer som hör till programmet.

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

Lägga till program i programhanteraren

Det finns två sätt att lägga till program i programhanteraren:

- Registrera programmet

- Lägg till en programikon utan att registrera programmet

Registrera program

Programregistrering ger en fullständig programintegration.

Ett registrerat program:

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

Program kan registreras på två sätt:

- När du installerar ett skrivbordsanpassat program sker registreringen automatiskt. Mer information finns i "Lägga till skrivbordsanpassade program i Programhanteraren" på sidan 44.
- Genom att skapa ett registreringspaket kan befintliga program registreras. Mer information finns i "Registrera befintliga program eller program som inte är skrivbordsanpassade" på sidan 45.

Med registreringspaket går det lättare att administrera program på skrivbordet. Registreringspaket skapas inte på samma plats i filsystemet som används för skrivbordets konfigurationsfiler.

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.

Program som har lagts till utan registreringspaket:

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

Mer information finns i "Lägga till en programikon i befintlig programgrupp" på sidan 45.

▼ Lägga till skrivbordsanpassade program i Programhanteraren

Ett skrivbordsanpassat program är ett program som registreras automatiskt i programhanteraren när det installeras. Bland programmets filuppsättningar finns ett registreringspaket för skrivbordet.

- 1. Installera programmet genom att följa instruktionerna till programmet.**

2. När installationen är klar dubbelklickar du på **Ladda om program i programgruppen Skrivbordsverktyg**.
3. **Kontrollera att installationen är klar genom att:**
 - a. **Öppna programhanteraren och kontrollera att den nya programgruppen finns där.**
 - b. **Öppna programmet genom att öppna programgruppen och dubbelklicka på dess programikon.**

▼ Registrera befintliga program eller program som inte är skrivbordsanpassade

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

På skrivbordet finns verktyget `dtappintegrate` som skapar länkar mellan registreringspaketets filer och mapparna i skrivbordets sökväg.

Mer information om skrivbordsregistrering finns i Kapitel 5.

▼ Lägga till en programikon i befintlig programgrupp

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

På skrivbordet finns bl a programgruppen Systemadministration som reserverats för olika program och skript som hjälper dig att administrera system. Om du har ett skript som användarna ofta använder kanske du vill att de ska kunna köra skriptet genom att dubbelklicka på en ikon i programgruppen Systemadministration.

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

Mer information om Skapa funktion finns i Kapitel 11.

2. **Skapa en körbar fil med samma namn som funktionen i programgruppens mapp. Innehållet i filen spelar ingen roll.**

Om du t ex har skapat en funktion med namnet "Ordna" som startar ett verktyg för systemadministration, skapar du den körbara filen /
`etc/dt/appconfig/appmanager/språk/Systemadministration/Ordna.`

Skapa och hantera allmänna programgrupper

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

Du kan skapa flera allmänna programgrupper. Du kan till exempel skapa en grupp som heter Spel som innehåller alla spel i systemet.

En allmän programgrupp kan vara systemtäckande eller egen.

▼ Skapa en systemtäckande allmän programgrupp

1. **Logga in som root.**
2. **Skapa en mapp i `/etc/dt/appconfig/appmanager/språk`.**
Programgruppen får samma namn som mappen.
3. **Dubbelklicka på Ladda om program i programgruppen Skrivbordsverktyg.**

▼ Skapa en egen allmän programgrupp

1. **Skapa en mapp i `Hemmapp/.dt/appmanager`.**
Programgruppen får samma namn som mappen.
2. **Dubbelklicka på Ladda om program i programgruppen Skrivbordsverktyg.**

▼ Anpassa en inbyggd programgrupp

1. **Logga in som root.**
2. **Om programgruppen finns i `/usr/dt/appconfig/appmanager/språk`, kopierar du programgruppen till `/etc/dt/appconfig/appmanager/språk`.**
Följande kommando kopierar till exempel programgruppen Skrivbordsverktyg:

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


Den nya programgruppen kommer att prioriteras framför den inbyggda.

3. Ändra kopian av programgruppen. Du kan t ex lägga till nya funktionsfiler (körbara filer med samma namn som funktionen).
4. Kontrollera ändringarna genom att logga ut och sedan in igen.

Ändra sökvägen som används för att söka efter program

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

Mer information om programsökvägen finns i "Programsökväg" på sidan 134.

Standardsökvägen

Standardsökvägen innehåller mapparna i Tabell 4-3.

TABELL 4-3 Mapper i standardsökvägen

Definitionsområde	Mapp i sökvägen
Egen	<i>Hemmapp</i> /.dt/appmanager
Systemtäckande	/etc/dt/appconfig/appmanager/ <i>språk</i>
Inbyggd	/usr/dt/appconfig/appmanager/ <i>språk</i>

Lägga till en programserver i programsökvägen

Det kan hända att du, utöver ändringen av programsökvägen, också måste utföra andra konfigurationer för att kontakten med programservern ska fungera. Mer information finns i "Administrera programtjänster" på sidan 116.

Ange en systemtäckande programsökväg

1. **Logga in som root.**

2. **Om inte filen `/etc/dt/config/Xsession.d/0010.dtpaths` redan 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 och exporterar variabeln `DTSPSYSAPPHOSTS`:**

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

I följande rad läggs systemet `ApServA` till i programsökvägen:

```
export DTSPSYSAPPHOSTS=ApServA:
```

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

Ange en egen programsökväg

1. **Öppna *Hemmapp*/`.dtprofile` för redigering.**

2. **Lägg till eller redigera en rad som anger och exporterar variabeln `DTSPUSERAPPHOSTS`:**

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

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

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

3. **Logga ut och sedan in igen.**

Allmän administration av Programhanteraren

Allmän administration av Programhanteraren omfattar:

- Ta bort program
- Ladda om programdatabasen under en session

▼ Ta bort ett program

Om ett program har registrerats med verktyget `dtappintegrate`, kan du också använda `dtappintegrate` för att göra det motsatta. När ett program avregistreras tas dess programgrupp bort från Programhanteraren, och dess funktioner, datatyper och ikoner är inte längre tillgängliga.

1. **Logga in som root.**
2. **Kör kommandot:**

```
dtappintegrate -s programrot -u
```

▼ Uppdatera 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 program.**

Ladda om program är användbart för att uppdatera programhanteraren när program har lagts till på en programserver. Ladda om program upptäcker dock inte program som har tagits bort från en programserver eller program som har flyttats från en plats 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 dubbelklicka på en ikon i Programhanteraren.

Dessa program startas också av andra skrivbordsaktiviteter.

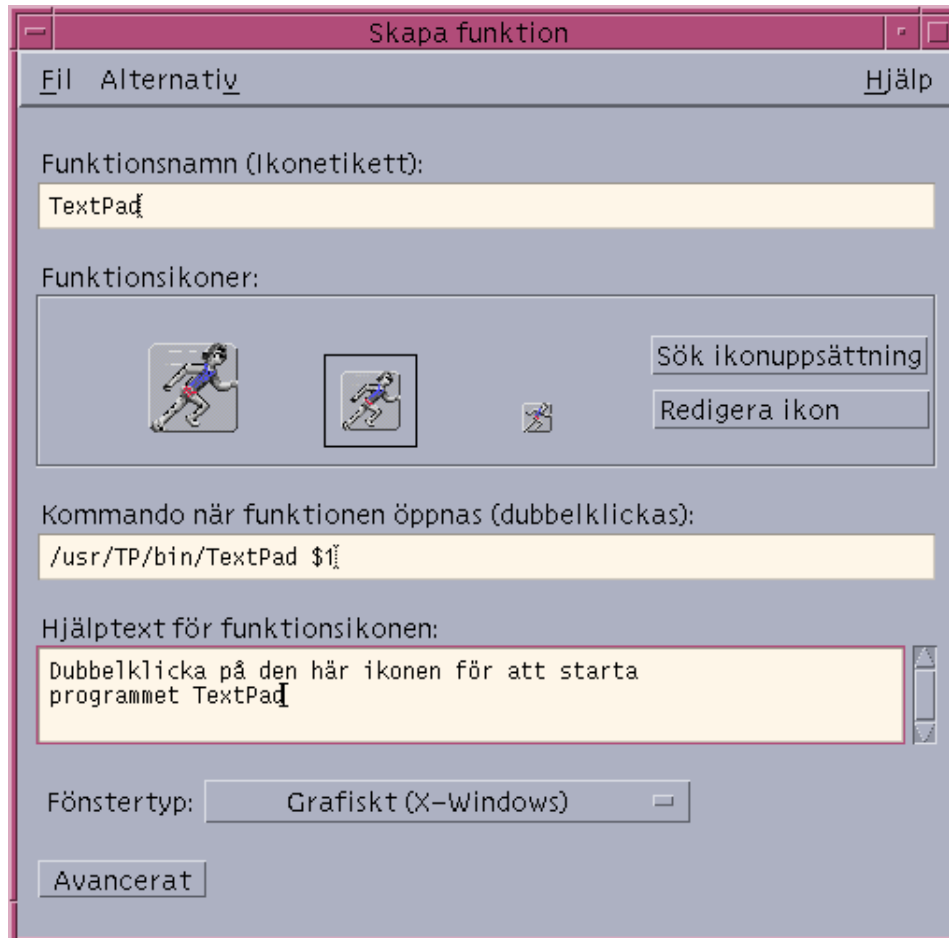
- Textredigeraren öppnas när användaren markerar textfilen i filhanteraren och väljer Öppna på Markera-menyn. Standardredigeraren är `dtpad`.
- En terminalemulator körs när användaren väljer Öppna terminal på filhanterarens Fil-menü eller när en funktion öppnar terminalemulatorfönstret. Standardterminalen är `dtterm`.

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

▼ Ändra standardtextredigerare och standardterminalemulator

1. **Logga in som root om ändringen ska vara systemtäckande.**
2. **Skapa en funktion för den nya textredigeraren eller terminalemulatorn.**

- Du kan använda programmet Skapa funktion. Figur 4-3 visar ett fönster i Skapa funktion som fyllts i för ett program som heter TextPad. Mer information om Skapa funktion finns i Kapitel 11.



Figur 4-3 Fönstret Skapa funktion

- *Eller, skapa funktionen manuellt:*

```

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

```

}

Mer information om hur du manuellt skapar funktionsdefinitioner finns i Kapitel 12.

3. Placera konfigurationsfilen med den nya funktionen i rätt mapp:

- Systemtäckande: `/etc/dt/appconfig/types/språk`
- Egen: `Hemmapp/.dt/types`

4. Om den inte redan finns, skapar du en lämplig `user-prefs.dt` -fil genom att kopiera `/usr/dt/appconfig/types/språk/user-prefs.dt` till:

- Systemtäckande: mappen `/etc/dt/appconfig/types/språk`.
- Egen: mappen `Hemmapp/.dt/types`.

5. Redigera funktionen `TextEditor` eller `Terminal` i den systemtäckande eller egna filen `user-prefs.dt`. Ändra `MAP_ACTION` till att tilldela funktionen den nya funktionen.

Ändra t ex raden:

```
MAP_ACTION Dtpad
```

till:

```
MAP_ACTION TxtPd
```

6. Spara filen `user-prefs.dt`.

7. Uppdatera funktionsdatabasen genom att dubbelklicka på `Ladda om funktioner` i programgruppen `Skrivbordsverktyg`.

Registrera ett program

Det här kapitlet beskriver hur du skapar ett registreringspaket för ett program och hur du registrerar programmet.

- "Programregistrering – översikt" på sidan 54
- "Generella steg för registrering av program" på sidan 57
- "Steg 1: Ändra teckensnitts- och färgresurser" på sidan 58
- "Steg 2: Skapa skrivbordsprogrammets rot" på sidan 59
- "Steg 3: Skapa registreringspaketets mappar" på sidan 60
- "Steg 4: Skapa funktionerna och datatyperna för programmet" på sidan 63
- "Steg 5: Placera hjälpfilerna i registreringspaketet" på sidan 66
- "Steg 6: Skapa ikoner för programmet" på sidan 66
- "Steg 7: Skapa programgruppen" på sidan 68
- "Steg 8: Registrera programmet med hjälp av dtappintegrate" på sidan 73
- "Exempel på hur du kan skapa ett registreringspaket" på sidan 76

När ett program är helt registrerat på skrivbordet har det:

- En egen programgrupp på översta nivån i Programhanteraren
- En funktion som startar programmet. Funktionen representeras av en ikon i programgruppen
- Datatyper för sina datafiler (valfritt)

Programregistrering är en icke inträngande åtgärd i programmet:

- Den ändrar inte programmets körbara filer. Du kan därför registrera befintliga program i ett system.
- Den kräver inte att någon av programmets inbyggda filer (t ex körbara och `app-defaults`) ska flyttas.

- Registrering är enkel att ångra. Verktöget `dtappintegrate`, som används för att registrera program, innehåller ett kommandoradsalternativ som upphäver processen.

Du kan behöva skapa ett registreringspaket om du är:

- En systemadministratör som vill registrera ett befintligt program på skrivbordet
- En programmerare som vill skapa ett installationspaket till ett skrivbordsprogram

Programregistrering – översikt

I det här avsnittet förklaras:

- Syftet med programregistrering
- Funktioner som tillförs programmet genom programregistrering

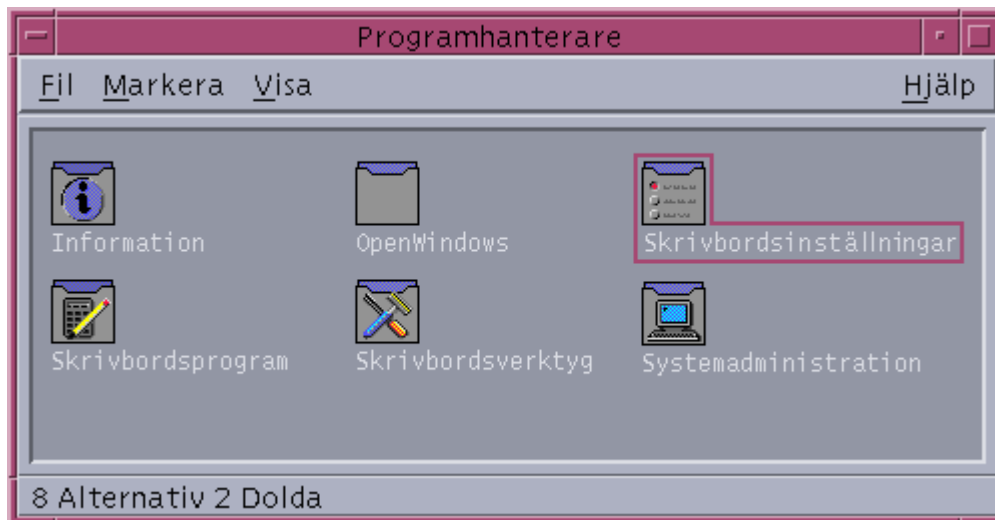
Obs! Ett utförligt exempel som visar hur du registrerar ett befintligt program finns i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

Funktioner som medföljer programregistrering

Programregistrering erbjuder ett grafiskt sätt för användare att:

- Leta rätt på programmet.

I samband med installationen "registreras" programmet i Programhanteraren och får en egen programgrupp.



Figur 5-1 Programgrupper på översta nivån i Programhanteraren

- Starta programmet.

Programgruppen med ditt program innehåller en ikon som användaren kan dubbelklicka på och starta programmet.



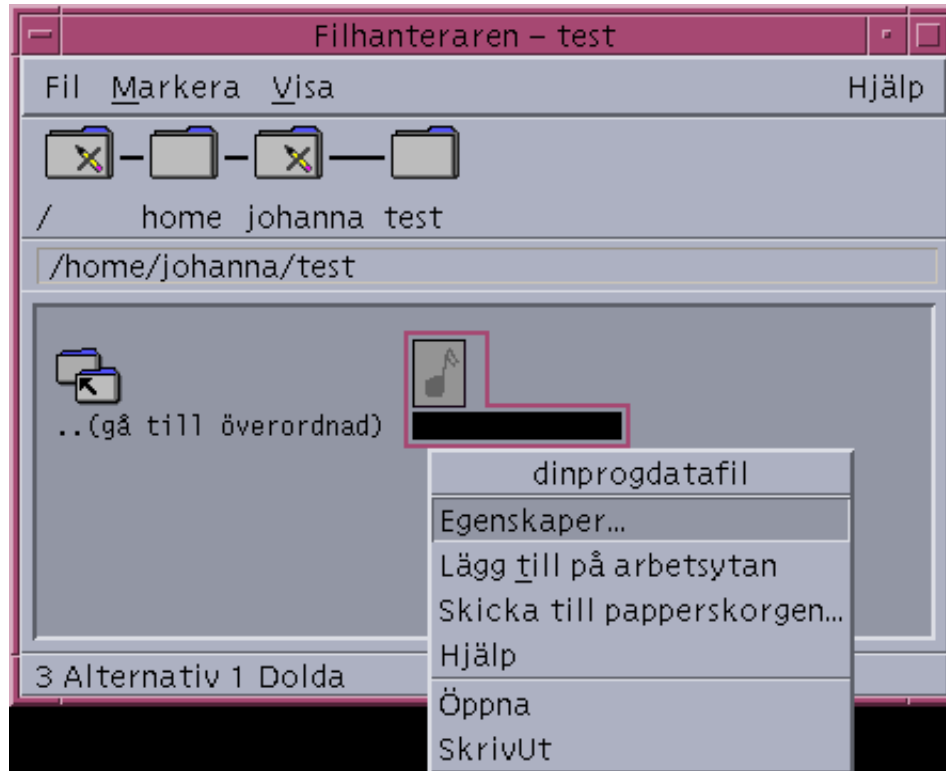
Figur 5-2 En programgrupp som innehåller en ikon som startar programmet

- Identifiera och ändra datafiler. Programmets datafiler kommer att ha en unik ikon i Filhanteraren.

Användaren kan använda datafilikoner för att:

- Starta (Öppna) programmet

- Skriva ut datafiler



Figur 5-3 En datafiles snabbmeny med "Öppna" and "Skriv ut"

- Utföra andra åtgärder, t ex posthantering, komprimering, visning eller uppspelning av data (ljud).

Syftet med programregistrering

Ett registrerat skrivbordsprogram innehåller vissa konfigurationsfiler som används av skrivbordet för att visa programmets användargränssnitt:

- Filer med funktions- och datatypsdefinitioner
- Ikonbildfiler (bildpunktsbeskrivning eller bitmapp)
- En mapp och filer som skapar programgruppen
- Skrivbordets hjälpfiler och Frontpanelens definitionsfiler (valfritt)

För att dessa filer ska kännas igen och användas av skrivbordet måste de finnas i vissa mappar som anges i skrivbordets sökvägar.

Det kan vara svårt att administrera ett program om dess konfigurationsfiler är utspridda i flera mappar. Därför kan ett program lagra alla skrivbordskonfigurationsfiler i en och samma mapp. Den här grupperingen av filer kallas *registreringspaket*.

Om programmet är skrivbordsanpassat innehåller det ett registreringspaket som en del av installationspaketet. Om du är systemadministratör och skapar konfigurationsfilerna själv kan du skapa registreringspaketet själv.

Konfigurationsfilerna i registreringspaketet är inte tillgängliga på skrivbordet eftersom de inte finns i lämpliga sökvägsmappar. Processen att placera de här filerna på lämpliga adresser kallas *registrering* eller *integrering* av programmet.

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

Många skrivbordsanpassade program kör `dtappintegrate` automatiskt under installationen. Om du är systemadministratör och integrerar ett befintligt program kan du köra det själv när du har skapat registreringspaketet.

När ett program är registrerat på ett systems skrivbord är programmet tillgängligt för alla användare i systemet. Om systemet är konfigurerat som en skrivbordsprogramserver blir programmet också tillgängligt i andra system i nätverket.

Verktyget `dtappintegrate` har ett kommandoradsalternativ som upphäver processen genom att bryta länkarna. Detta gör det enkelt att ta bort programmet från Programhanteraren så att det kan flyttas till en annan programserver eller uppdateras.

Generella steg för registrering av program

Obs! Ett utförligt exempel på hur du kan använda de här stegen för att skapa ett programpaket finns i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

1. **Ändra alla programresurser som anger teckensnitt och färger. Annars kommer inte skrivbordets dynamiska teckensnitt och färger att fungera korrekt.**
Se "Steg 1: Ändra teckensnitts- och färgresurser" på sidan 58.
2. **Skapa en rotadress för programmet.**
Se "Steg 2: Skapa skrivbordsprogrammets rot" på sidan 59.

3. Skapa mappstrukturen under programroten.

Se "Steg 3: Skapa registreringspaketets mappar" på sidan 60

4. Skapa funktionerna och datatyperna för programmet.

Se "Steg 4: Skapa funktionerna och datatyperna för programmet" på sidan 63.

5. Placera hjälpfilerna i lämplig mapp.

Se "Steg 5: Placera hjälpfilerna i registreringspaketet" på sidan 66.

6. Skapa ikonerna för programmet

Se "Steg 6: Skapa ikoner för programmet" på sidan 66.

7. Skapa programgruppen för programmen.

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

8. Registrera programmet med hjälp av dtappintegrate.

Se "Steg 8: Registrera programmet med hjälp av dtappintegrate" på sidan 73.

Steg 1: Ändra teckensnitts- och färgresurser

Obs! Ett exempel på hur du kan ändra resurser för ett program finns i Steg 1 i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

Skrivbordet tillhandahåller mekanismer för att ange och ändra gränssnittsteckensnitt och fönsterfärger. För att ett program ska kunna använda de här mekanismerna korrekt kan du behöva ändra programmets `programstandardfiler`.

Ändra teckensnittsresurser

Obs! Det här avsnittet gäller program som är skapade med OSF/Motif 1.2™ (eller senare versioner). Inställningar kan inte ange gränssnittsteckensnitt för program som är skrivna med tidigare versioner av OSF/Motif.

Inställningar på skrivbordet anger gränssnittsteckensnitt för program som är skrivna med OSF/Motif 1.2 (eller senare versioner) om programmet inte anger programspecifika gränssnittsteckensnitt.

Inställningar tillhandahåller två teckensnitt:

systemteckensnitt – används i systemområden som t ex etiketter, menyer och knappar

användarteckensnitt – används i redigerbara områden, t ex textfält

Varje tillhandahålls i sju storlekar, med etiketten 1 t o m 7 i dialogrutan Teckensnitt. Teckensnitten i Inställningar är anslutna till faktiska teckensnitt i systemet via Inställningar som anges i `/usr/dt/app-defaults/språk/Dtstyle`.

Om du vill att programmet ska använda teckensnitten i Inställningar, bör du ta bort alla programresurser som anger teckensnitt för gränssnittet. Skrivbordet anger automatiskt korrekta programresurser:

`*FontList` - ange som systemteckensnitt

`XmText*FontList` - ange som användarteckensnitt

`XmTextField*FontList` - ange som användarteckensnitt

Ändra färgresurser

Inställningar erbjuder möjlighet att ändra programfärger dynamiskt. Programmet måste vara en OSF/Motif 1.1- eller OSF/Motif 1.2-klient. Klienter som skrivs med andra verktygslådor kan inte ändra färg dynamiskt. Färgändringarna träder i kraft när klienten startas om.

Det enklaste sättet att använda de dynamiska färger som tillhandahålls av skrivbordet är att ta bort alla programfärgresurser för bakgrunds- och förgrundsfärg.

Steg 2: Skapa skrivbordsprogrammets rot

Obs! Ett exempel på hur du kan skapa en rotmapp för ett skrivbordsprogram finns i Steg 2 i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

Registreringspaketets filer för programmet är grupperade under en mapp som kallas programrot, eller *programrot*. Mappen *programrot* som används för skrivbordskonfigurationsfilerna kan vara samma mapp som programmets installations-*programrot* eller någon annan mapp.

Anta exempelvis att ett program är installerat under mappen `/usr/BTE`. Samma mapp kan användas som *programrot* för skrivbordskonfigurationsfilerna. Om du däremot integrerar ett befintligt program som inte är skrivbordsanpassat, skapar du en annan *programrot*-mapp för skrivbordet. Det här förhindrar att konfigurationsfilerna som du skapar blir överskrivna när du uppdaterar programmet.

En systemadministratör kan t ex vilja skapa en mapp som heter `/etc/desktop_approots/BTE` som skrivbordets *programrot*-mapp.

Steg 3: Skapa registreringspaketets mappar

Obs! Ett exempel på hur du kan skapa registreringspaketmappar för ett program finns i Steg 3 i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

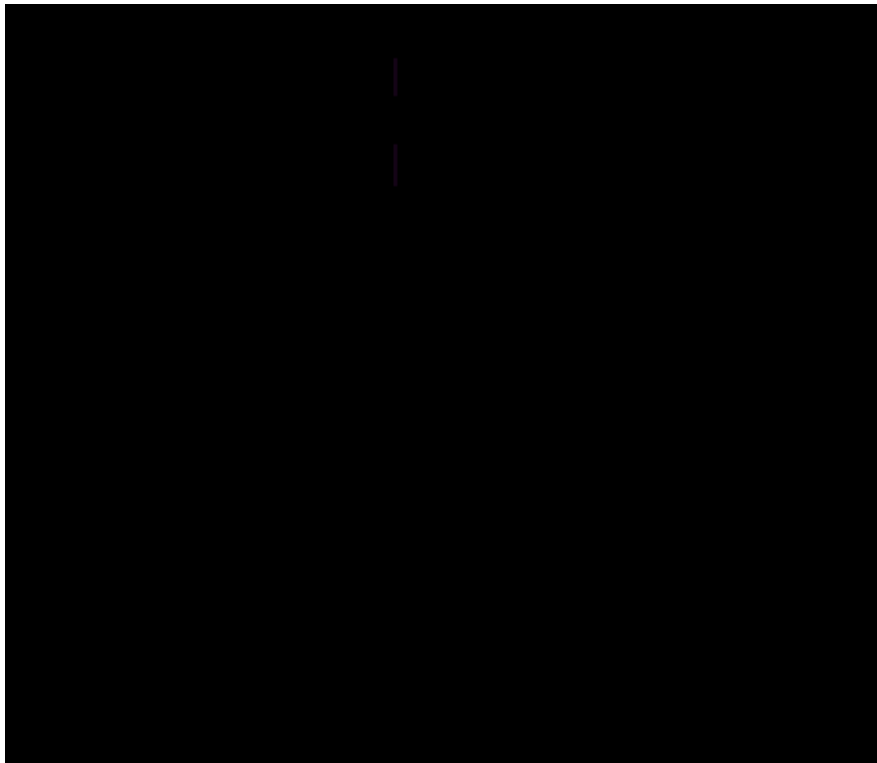
Registreringspaketet är den grupp med skrivbordskonfigurationsfiler som används av skrivbordet för att tillhandahålla ett grafiskt gränssnitt för programmet.

Registreringspaketet – innehåll

Skrivbordskonfigurationsfilerna innehåller:

- Filer med funktions- och datatypsdefinitioner
- Filer med ikonbilder
- En programgruppsmapp och dess innehåll
- Hjälpdatafiler och en konfigurationsfil för Frontpanelen (valfritt)

Registreringspaketet samlas under mappen på högsta nivå, kallad programroten eller *programrot*.



Figur 5-4 Ett registreringspaket under en programrotsmapp

Huvudkategorierna av konfigurationsfält under mappen *programrot/dt/appconfig* visas i Tabell 5-1.

TABELL 5-1 Konfigurationsfält – huvudkategorier

Undermapp	Innehåll
types	Filer för funktions- och datatypsdefinitioner
help	Skrivbordshjälpfiler
icons	Bitmapp- och bildpunktsbeskrivningsfiler som används av programmets funktioner och datatyper
appmanager	Mappen och innehållet som skapar programgruppen

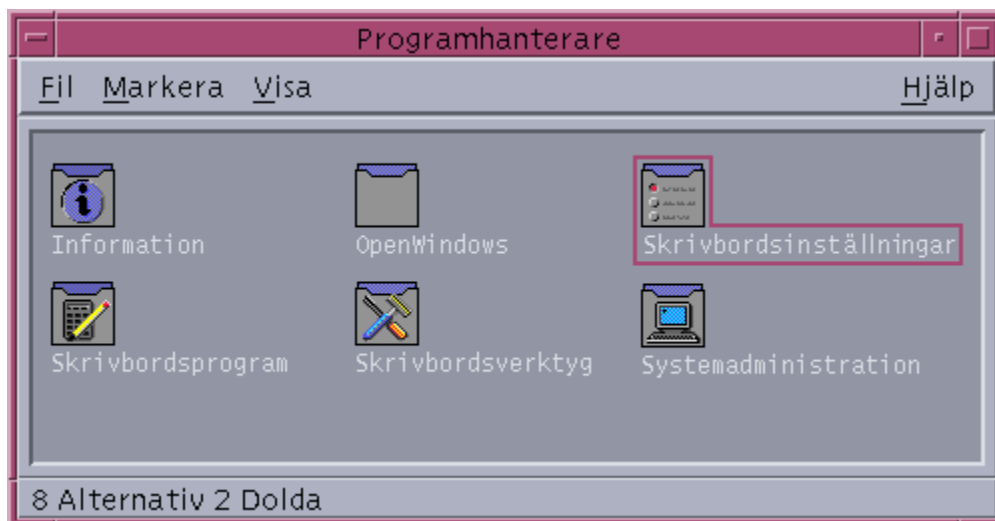
Var och en av huvudkategorierna har undermappar för språkberoende filer. Filer på standardspråket placeras i mappen C

Skapa registreringspaketet

- ◆ **Skapa de här mapparna. Om du tillhandahåller språkberoende konfigurationsfiler skapar du en separat mapp för varje språk. Om du bara tillhandahåller ett språk placerar du filerna i mappen C.**

- *programrot/dt/appconfig/types/språk*
- *programrot/dt/appconfig/help/språk*
- *programrot/dt/appconfig/icons/språk*
- *programrot/dt/appconfig/appmanager/språk/proggruppnamn*, där *proggruppnamn* är namnet på programgruppen.

Exempel: Figur 5-5 visar Programhanteraren med en grupp vars *proggruppnamn* är "Skrivbordsprogram".



Figur 5-5 Programgrupp på högsta nivån i Programhanteraren

Verktyget dtappintegrate fungerar bara på skrivbordskonfigurationsfilerna i mapparna types, help, icons och appmanager. Programmets binära körbara programstandardfiler och meddelandemappfiler administreras separat.

Steg 4: Skapa funktionerna och datatyperna för programmet

Obs! Ett exempel på hur du kan skapa funktionerna och datatyperna för ett program finns i Steg 4 i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

Funktioner och datatyper tillhandahåller ett användargränssnitt för programmet.

- Funktioner tillhandahåller ett användargränssnitt där kommandot kan starta programmet.
- Datatyper tillhandahåller anpassade utseenden och beteenden för programmets datafiler.

Nödvändiga funktioner och datatyper för ett program

Ett typiskt program kan kräva följande funktions- och datatypsdefinitioner:

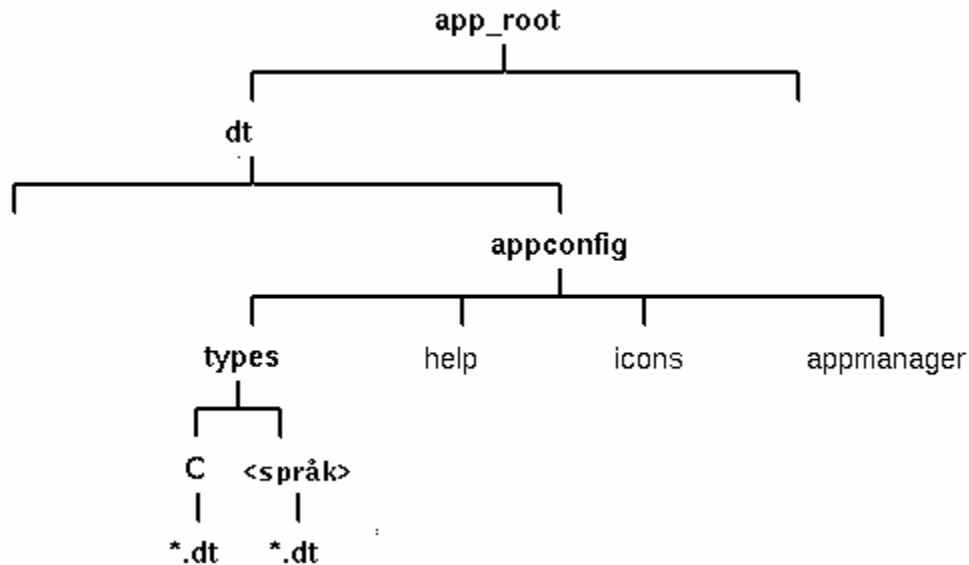
- En funktion som öppnar programmet.
- En datatyp för datafilerna i programmet. Om du skapar en datatyp behöver du också skapa:
 - En Open-funktion för datafilerna i programmet
 - En Print-funktion för datafilerna i programmet
- En datatyp för programgruppen (se "Konfigurera programgruppen så att den använder en unik ikon" på sidan 69).

Du får en beskrivning av hur funktioner och datatyper används på skrivbordet i Kapitel 10.

Adress för konfigurationsfiler för funktions- och datatypsdefinitioner

Funktioner och datatyper definieras i konfigurationsfiler. Det enda namnkravet för filer som innehåller funktions- och datatypsdefinitioner är att de måste ha ett `.dt`-suffix. Du kan vilja namnge filen konventionsenligt till *funktionsnamn.dt* eller *prognamn.dt*.

Placera filer som innehåller funktioner och datatyper under programroten i mappen *programrot/dt/appconfig/types/språk*. Standardbeskrivningen för *språk* är `C`.



Figur 5-6 Filer med funktions- och datatypsdefinitioner

Olika sätt att skapa funktioner och datatyper

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

- Använd verktyget Skapa funktion.

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

- Skapa definitionerna manuellt.

Det här kräver att du lär dig syntax för att skapa definitionerna, men ger dig samtidigt tillgång till hela uppsättningen funktioner.

Skapa funktioner och datatyper med hjälp av Skapa funktion

Den här proceduren använder hjälpmedlet Skapa funktion för att skapa funktioner och datatyper för programmet.

Mer information om Skapa funktion finns i direkthjälpen och i Kapitel 11.

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

Konfigurationsfilen som skapas av Skapa funktion kommer att skrivas till *Hemmapp/.dt/types/funktionsnamn.dt*. Funktionsfilen (den körbara filen med samma namn som funktionen) placeras i din hemmapp.

3. Testa funktionen med hjälp av funktionsfilen som har skapats i din hemmapp.
4. Kopiera funktionsdefinitionsfilen *Hemmapp/.dt/types/funktionsnamn.dt* till mappen *programrot/dt/appconfig/types/språk*.
5. När programgrupsmappen har skapats (se "Steg 7: Skapa programgruppen" på sidan 68) kopierar du funktionsfilen *Hemmapp/funktionsnamn* till mappen *programrot/dt/appconfig/appmanager/språk/progrupp*.

Skapa funktioner och datatyper manuellt

- ◆ Skapa en konfigurationsfil som innehåller funktions- och datatypsdefinitioner för programmet.

Funktions- och datatypsdefinitionsfilerna måste följa namnkonventionen *namn.dt*.

Du kan placera alla funktions- och datatypsdefinitioner i en fil eller distribuera dem bland flera filer. För varje fil använder du ett filnamn som systemadministratörer enkelt kopplar till ditt program.

Funktions- och datatypsnamn måste bestå av ett ord (inga inbäddade blanksteg). Du kan använda ett understrykningstecken. Enligt konventionen är den första bokstaven i funktions- eller datatypsnamnet versal. Använd inte ett befintligt funktionsnamn eller filnamn. Använd ett namn som avancerade användare och systemadministratörer enkelt kopplar till ditt program.

Om du vill att programmets ikon ska ha ett annat namn än funktionsnamnet inkluderar du fältet `LABEL` i funktionsdefinitionen.

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

- Kapitel 10
- Kapitel 11
- Kapitel 12
- Kapitel 13

Steg 5: Placera hjälpfilerna i registreringspaketet

Obs! Ett exempel på hur du kan lägga till hjälpfiler i ett registreringspaket finns i Steg 5 i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

Om programmet innehåller en skrivbordshjälpvolym (en hjälpvolym som är skapad med skrivbordets Help Developer's Kit), bör hjälpvolymens huvudfil (*.sdl) placeras i mappen *programrot/appconfig/help/språk*.

Grafik som används av hjälpfilerna placeras vanligen i undermappen *graphics*. Grafiken måste finnas i samma mapp relativt huvudhjälpvolymens (*.sdl)-fil som när hjälpvolymen skapades.

Om programmet inte innehåller en hjälpvolym kan du skapa en sådan, om du har tillgång till Help Developer's Kit.

Det finns två integrationsnivåer för en hjälpvolym:

- Full integration.

När skrivbordshjälpen är fullt integrerad kan hjälpvolymen kommas åt från programmet – exempelvis via objekthjälp och Hjälp-menyn. Full integrering innebär att programmets körbara filer ändras.

- Delvis integration.

När skrivbordshjälpen är delvis integrerad är den tillgänglig från högsta nivån i Hjälpmantern. Du kan emellertid inte komma åt hjälpvolymen från programmets fönster. Du kan också låta en funktion komma åt hjälpen från programgruppen. I följande exempel visar funktionen hjälpvolymen som finns i hjälpens huvudfil *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 ikoner för programmet

Obs! Ett exempel på hur du kan skapa ikonfiler för ett program finns i Steg 6 i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

Skrivbordet tillhandahåller standardikoner för funktioner, datatyper och programgrupper. Troligen vill du emellertid skapa unika ikoner för programmet.

Ikoner placeras i mappen *programrot/dt/appconfig/icons/språk*.

Nödvändiga ikoner för skrivbordet

Programmet använder de här ikonbilderna på skrivbordet:

- **Funktionsikon** Det här är ikonen som användaren dubbelklickar på för att starta programmet (funktionerna). Den refereras till i funktionens `ICON`-fält som startar programmet.

Den finns i tre storlekar: mycket liten, medelstor och stor.

- **Datatypesikon**. Den här ikonen används för att representera programmets datafiler i Filhanteraren. Den refereras till i datatypesdefinitionens `ICON`-fält.

Om programmet stöder flera datatyper bör du ha en särskild ikon för varje datatyp.

Den finns i två storlekar: mycket liten och medelstor.

- **Programgruppikon**. Det här är ikonen som representerar mappen på högsta nivån i Programhanteraren. Den refereras till i datatypesdefinitionens `ICON`-fält för programgruppen. (I "Steg 7: Skapa programgruppen" på sidan 68.)

Den finns i två storlekar: mycket liten och medelstor.

Du kan behöva stödja både bildpunktsbeskrivnings- och bitmappsversioner av varje ikon för att stödja både färgskärmar (åtta bitar och större) och monokroma skärmar (färre än åtta bitar).

TABELL 5-2 Namnkonventioner för ikonfiler

storlek	Bildpunktsstorlek	Bitmappsnamn	Bildpunktsmönster
mycket liten	16 gånger 16	<i>basnamn</i> .t.bm	<i>basnamn</i> .t.pm
medelstor	32 gånger 32	<i>basnamn</i> .m.bm	<i>basnamn</i> .m.pm
stor	48 gånger 48	<i>basnamn</i> .l.bm	<i>basnamn</i> .l.pm

Om du inte tillhandahåller bitmappsfiler anger skrivbordet färgspecifikationerna för bildpunktsbeskrivningsfilerna till svartvitt. Det är dock möjligt att den här tilldelningen inte ger det utseende du vill ha.

Mer information om ikoner finns i "Ikonbildsfiler" på sidan 215.

Steg 7: Skapa programgruppen

Obs! Ett exempel på hur du skapar en programgrupp finns i Steg 7 i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

När du har skapat funktions- och datatypsdefinitionerna för programmet måste du skapa konfigurationsfilerna som ska skapa det som användaren ser på skärmen – programgruppen och dess innehåll.

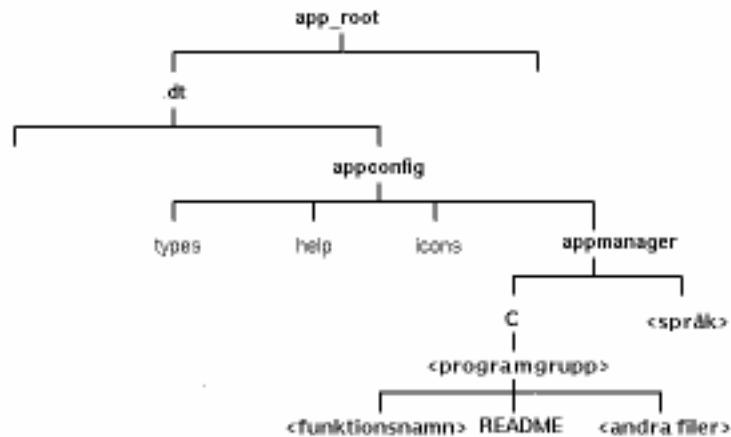
Programgruppen är en mapp på högsta nivån i Programhanteraren (se Figur 5-1).

Du skapar programgruppen i tre steg:

- Skapa programgruppsmappen i registreringspaketet.
- Valfritt: konfigurera programgruppen så att den använder en unik ikon. Det innebär att du skapar datatypsdefinitionen för programgruppsmappen.
- Skapa innehållet i programgruppen.

Skapa programgruppsmappen

Om du vill skapa en programgrupp skapar du mapparna i registreringspaketet under `appmanager`, som det visas i Figur 5-7.



Figur 5-7 Mappen `appmanager`

Programgruppsnamn

`<programgruppsnamn>` i Figur 5-7 är namnet på programgruppen.



Figur 5-8 Programgruppnamnet (<programgruppnamn>)

Namnet kan vara ett valfritt tillåtet fil- eller mappnamn. Använd ett namn som beskriver programmet.

Konfigurera programgruppen så att den använder en unik ikon

Skrivbordet tillhandahåller en standardikon för programgrupper. Troligen vill du emellertid använda en anpassad ikon.

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

- En datatyp för mappen som visas på översta nivån i Programhanteraren.
- Open- och Print-funktioner för datatypen.

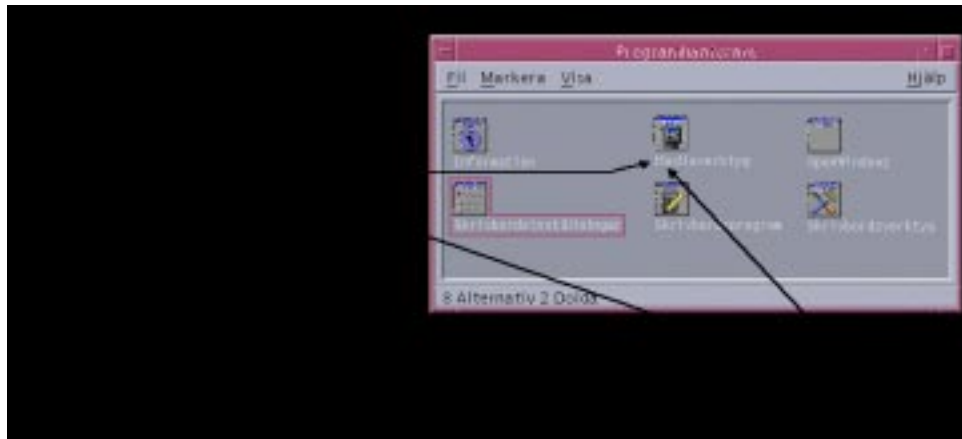
Anta exempelvis att du vill skapa en programgrupp som heter Mediaverktyg. Följande datatypsdefinition, placerad i filen *programrot/dt/appconfig/types/språk/namn.dt*, anger en unik ikon som programgruppsikon.

```
DATA_ATTRIBUTES  MediaverktygProggrupp
{
    ACTIONS      OpenInPlace,OpenNewView
    ICON         Mediaverktyg
    DESCRIPTION  Dubbelklicka för att öppna programgruppen \
                Mediaverktyg
}

DATA_CRITERIA  MediaverktygProggruppKriterial
{
    DATA_ATTRIBUTES_NAME  MediaverktygProggrupp
    MODE                  d
    PATH_PATTERN          */appmanager/*/Mediaverktyg
}
```

Attributavsnittet i definitionen anger vilken ikon som ska användas. Villkorsavsnittet i definitionen anger att datatypen ska definieras till valfri mapp som heter Mediaverktyg och som är en undermapp till en mapp som heter appmanager.

Figur 5-9 visar sambandet mellan programgruppnamnet och datatypsdefinitionen. Fältet PATH_PATTERN i datatypsdefinitionen ansluter en unik ikon till programgruppen.



Figur 5-9 Hur en programgrupp får en unik ikon

Du bör också skapa en Open- och Print-funktion för programgruppens datatyp:

```
ACTION Open
{
  ARG_TYPE  MediaverktygProggrupp
  TYPE      MAP
  MAP_ACTION OpenAppGroup
}
```

```
ACTION Print
{
  ARG_TYPE  MediaverktygProggrupp
  TYPE      MAP
  MAP_ACTION PrintAppGroup
}
```

OpenAppGroup- och PrintAppGroup är inbyggda funktioner som finns definierade i /usr/dt/appconfig/types/språk/dtappman.dt.

Skapa innehållet i programgruppen

Det viktigaste elementet i programgruppen är den ikon som startar programmet (en funktionsikon). Om programgruppen innehåller en uppsättning program finns det vanligen en ikon för varje program.

Förutom en eller flera funktionsikoner kan programgruppen innehålla:

- En eller flera README-filer
- Ett eller flera exempel på datafiler
- Mallar
- En ikon som användaren kan dubbelklicka på för att visa hjälpinformation
- En direkthjälp

- En särskild Frontpanelkontroll

Programgruppen kan innehålla undermappar.

Skapa funktionsfilen (programikon)

Programgruppen bör innehålla en ikon som startar programmet. Om gruppen tillhandahåller en uppsättning program bör det finnas en ikon för varje program. De här ikonerna kallas *programikoner* eller *funktionsikoner* eftersom de representerar en underliggande funktion.

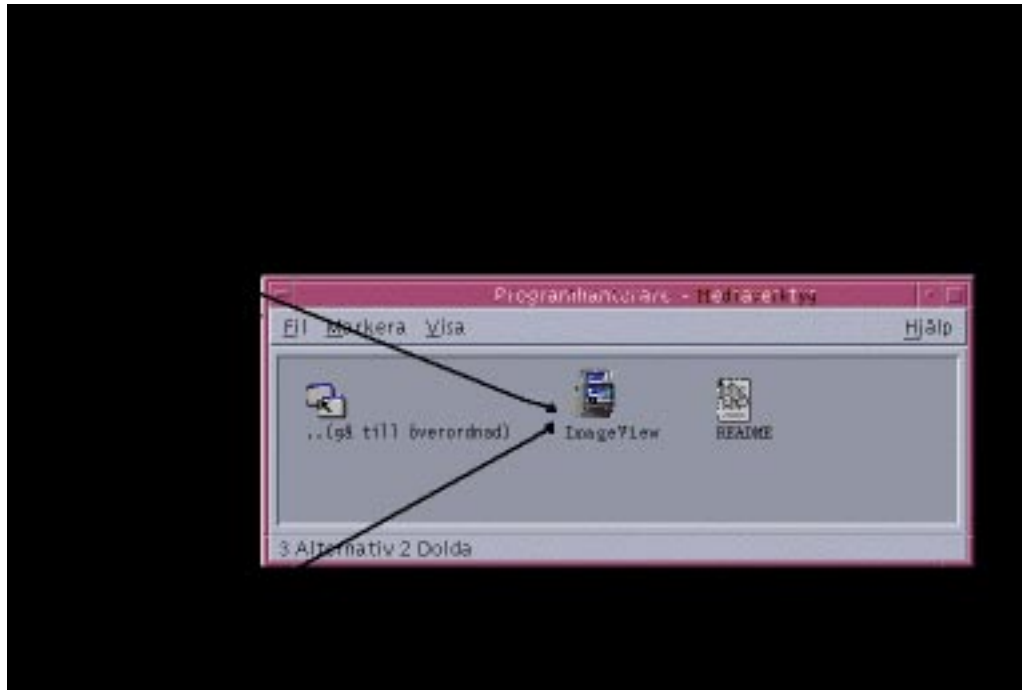
Du skapar en funktionsikon genom att skapa en körbar fil med samma namn som funktionen den kommer att köra:

```
programrot/dt/appconfig/appmanager/progrupp/funktionsnamn
```

Filen kallas en *funktionsfil* eftersom dess syfte är att skapa en visuell representation av den underliggande funktionen.

Om du t ex har skapat en funktion som heter BästaTextredigerare som kör programmet BästaTextredigerare, skapar du en körbar fil som heter BästaTextredigerare. I Filhanteraren och Programhanteraren använder funktionsfilen ikonbilden som anges i funktionsdefinitionen.

Figur 5–10 illustrerar förhållandet mellan funktionsdefinitionen, funktionsfilen och faktiska inmatningar i programhanterarfönstret.



Figur 5-10 Programikonen är en fil i programgruppen

Readme-filer

Till skrivbordet finns datatypen README som du använder till programmets README-filer. Uttrycket Readme förknippas oftast med något som är viktigt, dvs en fil eller något som ska läsas så snart som möjligt. Använd en av följande namngivningskonventioner:

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

Skapa en särskild kontroll för Frontpanelen

För det mesta behöver du inte tillhandahålla en kontrolldefinition för Frontpanelen. Användaren kan lägga till programmet i Frontpanelen genom att släppa funktionsikonen på installationsikonen på en panelundermeny.

Du kan vilja skapa en konfigurationsfil för Frontpanelen som innehåller en kontrolldefinition för programmet, om du vill att användare ska kunna installera en kontroll som fungerar annorlunda än funktionsikonen – om exempelvis kontrollen övervakar en fil och ändrar utseende när den övervakade filen ändras.

Frontpanelens konfigurationsfiler placeras i mappen `programrot/dt/appconfig/types/språk`. Namnkonventionen är `namn.fp`.

Om du tillhandahåller en konfigurationsfil som innehåller en kontroll kan användaren lägga till kontrollen i en panelundermeny genom att släppa filen `*.fp` på installationsikonen på panelundermenyn.

Följande definition kan exempelvis placeras i en konfigurationsfil på Frontpanelen i programgruppen. Om användaren släpper den här filen på en installationsikon på en panelundermeny skapas en kontroll i panelundermenyn, som kör en upplaga av programmet BästaTextredigerare. Om BästaTextredigerare redan körs flyttas fönstret högst upp i fönsterstacken på den aktuella arbetsytan.

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

Mer information om hur du skapar konfigurationsfiler för Frontpanelen finns i:

- Kapitel 15
- direkthjälpen (man) till `dtfpfil(4)`

Steg 8: Registrera programmet med hjälp av dtappintegrate

Obs! Ett exempel på hur du kan registrera ett program finns i Steg 8 i "Exempel på hur du kan skapa ett registreringspaket" på sidan 76.

När du har skapat ett registreringspaket under en programrot kan du utföra själva programregistreringen.

Programregistrering skapar länkar mellan registreringspaketet och mapparna som finns i skrivbordets sökvägar (se "Hur dtappintegrate integrerar program" på sidan 75).

Registrera ett program med dtappintegrate

Om programmet är skrivbordsanpassat körs vanligen dtappintegrate automatiskt som sista steg i installationsprocessen. Om det inte körs automatiskt, eller om du har skapat konfigurationsfiler för att integrera ett icke skrivbordsanpassat program, kan du köra dtappintegrate manuellt.

1. Logga in som root.

2. Kör kommandot:

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

där *programrot* är skrivbordets programrotsmapp. Mer information finns i direkthjälpen (man) till dtaction(1).

3. Öppna programgruppen Skrivbordsverktyg och dubbelklicka på Ladda om funktioner.

4. Kontrollera att programmet är korrekt registrerat:

- a. Visa 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 programrot [-t målsökväg ] [-l språk ] [-u]
```

<code>-s <i>programrot</i></code>	Nödvändig parameter som anger programroten under vilken programmet är installerat.
<code>-t <i>målsökväg</i></code>	Valfri parameter. Som standard är systemadressen / etc/dt/appconfig. Anger adressen dit skrivbordets konfigurationsfiler är länkade. Du måste använda en adress i programmets sökväg.
<code>-l <i>språk</i></code>	Valfri parameter. Som standard alla språk. Anger vilka språkberoende skrivbordskonfigurationsfiler som ska integreras.
<code>-u</code>	Valfri parameter, avintegrerar programmet och tar bort alla länkar som skapades under integreringen.

Hur dtappintegrate integrerar program

Funktionen hos dtappintegrate är att skapa länkar mellan de installerade filerna och adresserna där skrivbordet letar efter konfigurationsfiler.

Funktioner och datatyper

dtappintegrate skapar symboliska länkar från funktions- och datatypsdefinitionsfilerna i registreringspaketet till den systemtäckande mappen längs funktionsdatabasens hjälpsökväg. Detta åstadkoms genom att länkar skapas från

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

till

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

Hjälpinformationsfiler

dtappintegrate skapar symboliska länkar från hjälpfilerna i registreringspaketet till den systemtäckande mappen längs hjälpsökvägen. Detta åstadkoms genom att länkar skapas från

```
programrot/dt/appconfig/help/språk/hjälpfil.sdl
```

till

```
/etc/dt/appconfig/icons/språk/hjälpfil.sdl.
```

Ikonfiler

dtappintegrate skapar symboliska länkar från ikonfilerna i registreringspaketet till den systemtäckande mappen längs ikonsökvägen. Detta åstadkoms genom att länkar skapas från

```
programrot/dt/appconfig/icons/språk/ikonfiler
```

till

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

Programgrupp

För att placera programgruppen för programmet på översta nivån i Programhanteraren skapar dtappintegrate en länk mellan programgruppsmappen i registreringspaketet och den systemtäckande adressen längs programmets sökväg. Detta åstadkoms genom att länkar skapas från mappen

```
programrot/dt/appconfig/appmanager/språk/proggrupp
```

till

/etc/dt/appconfig/appmanager/språk/proggrupp.

Exempel på hur du kan skapa ett registreringspaket

Följande steg skapar ett registreringspaket för ett befintligt, icke skrivbordsanpassat program som heter BästaTextredigerare.

Information om ”BästaTextredigerare”

I exemplet görs följande antaganden om programmet BästaTextredigerare:

- Det är installerat i mappen /usr/BTE.
- Användarens sessionsspråk är standardspråket, C.
- Kommandoraden som startar BästaTextredigerare är:

```
BTEd {filnamn}
```

där *filnamn* är namnet på datafilen som ska öppnas i det nya fönstret. BästaTextredigerare skapar ett eget fönster – d v s det körs inte i ett terminalemulatorfönster.

- BästaTextredigerare skapar och använder två typer av datafiler:

- Dokumentationsfiler. De använder namnkonventionen *.bte. BästaTextredigerare tillhandahåller en kommandorad för utskrift av sina .bte-datafiler. Syntax för det här kommandot är:

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

där:

–d *mål* anger skrivare.

–s anger tyst utskrift. Programmets dialogruta för utskrift visas inte.

filnamn anger vilken fil som ska skrivas ut.

- Mallfiler. De använder namnkonventionen *.tpl. Mallfiler kan inte skrivas ut.
- De befintliga standardfilerna som inte är desktopprogram för BästaTextredigerare innehåller resurser för gränssnittsteckensnitt samt förgrunds- och bakgrundsfärger.

- En hjälpvolym för BästaTextredigerare skapades med skrivbordets Help Developer's Kit. När hjälpvolymen skapades användes följande källfiler:

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

och genererade filen .../BTEHelp.sdl.

Steg för att registrera "BästaTextredigerare"

Följande stegvisa procedur registrerar BästaTextredigerare.

1. Ändra teckensnitt- och färgresurser.

I BästaTextredigerares app-defaults-fil tar du bort resurser som anger:

- Teckensnitt för text
- Färger för förgrund och bakgrund

2. Skapa programroten.

Skapa mappen:

```
/desktop_approots/BTE
```

Om du integrerar ett befintligt program bör du skapa programrotsmappen någon annanstans än på installationsplatsen för programmet. Annars kan konfigurationsfilerna som du skapar tas bort när du uppdaterar programmet.

3. Skapa registreringspaketets mappar.

Skapa följande mappar:

```
/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/BästaTextredigerare
```

4. Skapa funktionerna och datatyperna för programmet.

a. Skapa konfigurationsfilen för funktions- och datatypsdefinitionerna:

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

b. Skapa funktionsdefinitionen för att köra BästaTextredigerare:

```
ACTION BTEditor
{
    WINDOW_TYPE    NO_STDIO
    ICON           BTERun
    DESCRIPTION    Dubbelklicka på den här ikonen eller släpp \
```

```

        en BTE-datafil på den för att köra \
        BästaTextredigerare.
EXEC_STRING    /usr/BTE/BTEd %Arg_1%
}

```

c. Skapa datatypen för *.bte-filer:

```

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

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

```

d. Skapa datatypen för *.tpl-filer:

```

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

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

```

e. Skapa Open-funktionen för *.bte-filer.

```

ACTION Open
{
    ARG_TYPE    BTEDataFile
    TYPE       MAP
    MAP_ACTION BTEditor
}

```

f. Skapa Print-funktionen för *.bte-filer.

Här följer enkla Print-funktioner som skriver ut datafiler. De här funktionerna kräver ett värde för miljövariabeln LPDEST och ignorerar utskriftsalternativet -s. (Om LPDEST inte anges kan funktionen misslyckas.)

```

ACTION Print
{

```



```

        ARG_TYPE    BTEDataFile
        TYPE        MAP
        MAP_ACTION  BTEPrintData
    }

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

```

Här följer en annan version av funktionen BTEPrintData och ett medföljande skript. Tillsammans hanterar de situationer där LPDEST inte anges eller 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 Open-funktionen för *.tpl-filer:

```

ACTION Open
{
    ARG_TYPE    BTETemplateFile
    TYPE        MAP
    MAP_ACTION  BTEditor
}

```

h. Skapa Print-funktionen för *.tpl-filer:

```

ACTION Print
{
    ARG_TYPES    BTETemplateFile
    TYPE        MAP
    MAP_ACTION    NoPrint
}

```

NoPrint är en inbyggd funktion som visar en dialogruta där det står att filen inte går att skriva ut.

5. Placera hjälpfilerna i registreringspaketet.

a. Placera hjälpfilerna på följande adresser:

```
/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 BästaTextredigerares hjälpvolym.
}
```

6. Skapa ikoner för programmet.

Använd Ikonredigeraren för att skapa ikonerna. Tabell 5-3 visar vilken storlek på rutnät som ska användas.

TABELL 5-3 Rutnät för olika storlekar på ikoner

Namn	Storlek
<i>basnamn</i> .t.pm	16 gånger 16
<i>basnamn</i> .m.pm	32 gånger 32
<i>basnamn</i> .l.pm	48 gånger 48

Skapa de här ikonfilerna i mappen

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

- Ikoner som representerar 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 mappen om du inte redan har gjort det.

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

b. Det här steget är valfritt. Det skapar en unik ikon för programgruppen genom att skapa en datatyp och associerade funktioner för programgruppen. Om du utelämnar det här steget använder programgruppen standardikonen.

Lägg till följande datatyp- och funktionsdefinitioner i filen

/desktop_approots/BTE/dt/appconfig/types/C/BTE.dt. Datatypen anger vilken ikon som ska användas av programgruppen BästaTextredigerare. Funktionerna ger samma Open- och Print-beteende som hos de inbyggda programgrupperna.

```
DATA_ATTRIBUTES BästaTextredigerareProggrupp
{
    ACTIONS  OpenInPlace,OpenNewView
    ICON     BTEApp
}

DATA_CRITERIA BästaTextredigerareProggruppVillkor
{
    DATA_ATTRIBUTES_NAME BästaTextredigerareProggrupp
    MODE                  d
    PATH_PATTERN          */appmanager/*/BästaTextredigerare
}

ACTION Open
{
    ARG_TYPE    BästaTextredigerareProggrupp
    TYPE        MAP
    MAP_ACTION  OpenAppGroup
}

ACTION Print
{
    ARG_TYPE    BästaTextredigerareProggrupp
    TYPE        MAP
    MAP_ACTION  PrintAppGroup
}
```

- c. Skapa en ikon i programgruppen som ska starta programmet. För att göra det skapar du filen:**

```
/desktop_approots/BTE/dt/appconfig/appmanager/C \  
/BastaTextredigerare/BTEditor
```

och gör filen körbar.

- d. Skapa funktionsfilen i programgruppen som ska öppna hjälpvolyten. För att göra det skapar du filen:**

```
/desktop_approots/BTE/dt/appconfig/appmanager/C \  
/BastaTextredigerare/BTEHelp
```

och gör filen körbar.

- e. Placera andra filer i programgruppen. Exempelvis "Readme"-filer, exempeldata och mallfiler.**

8. Registrera programmet.

I ett terminalemuleringsfönster:

- a. Logga in som root.**

- b. Kör kommandot:**

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

- c. Öppna programgruppen Skrivbordsverktyg och dubbelklicka på Ladda om funktioner.**

Diverse konfigurationer

Det här kapitlet handlar om installation och systemadministration.

- "Solaris CDE – mappstruktur" på sidan 83
- "Huvudkonfigurationsfiler" på sidan 85
- "Starta inloggningsservern" på sidan 86
- "Ansluta en installerad CDE från en annan installationsadress för en arbetsstation eller nätverksserver" på sidan 88
- "Konfigurera skrivbordet för att använda flera skärmar" på sidan 89
- "Nätverksanslutna skrivbord" på sidan 91
- "Använda X-terminaler" på sidan 92
- "Inloggningsspråkområde och teckensnittssökväg" på sidan 93
- "Använda arbetsstationer som X-terminaler" på sidan 94
- "Särskilda CDE-konfigurationer" på sidan 95
- "XServer och drivrutinen Power Management" på sidan 100

Solaris CDE – mappstruktur

Det här avsnittet beskriver huvudmapparna i skrivbordsmiljön.

/usr/dt

Den här mappen är Solaris CDE:s installationsadress. Den kan också vara en anslutningspunkt för en fjärrhanterare. Tabell 6-1 beskriver undermapparna /usr/dt.

TABELL 6-1 /usr/dt Undermappar

Undermapp	Beskrivning
/bin	SCDE-program och SCDE-hjälpmedel
/lib	SCDE körtidsdelade bibliotek
/config	Standardfiler för systemkonfiguration
/man	[Valfritt] direkthjälp (man)
/program- standardfiler	Standardprogramresurser
/appconfig	Standardprogramikoner, typer och funktioner
/exempel	[Valfritt] CDE-kod/CDE-program, exempel
/inkludera	[Valfritt] Tilläggsfiler (include files) för utvecklare
/paletter	Färger
/share	CDE AnswerBook, dokumentation och standardbakgrund

/etc/dt

Den här mappen innehåller anpassade arbetsstationsspecifika konfigurationsfiler. Filerna gör att du kan anpassa din miljö på följande sätt:

- Ange konfigurationsalternativ för X-server
- Använda flera skärmar
- Anpassa arbetsstationens funktionsfiler, datatyper, ikoner och teckensnitt

/var/dt

Den här mappen används för att lagra temporära filer för Solaris CDE-program t ex Inloggningshanteraren och Programhanteraren.

\$Hemmapp

Den här mappen innehåller användarspecifika filer som gäller användarens skrivbordsinställningar. Där ingår ändringar av program, färginställningar, Arbetsyta-menyn, Frontpanelen och felloggar.

Huvudkonfigurationsfiler

De flesta anpassningar av din skrivbordsmiljö kan inkludera många filer. Följande filer är två exempel:

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

Xconfig

Xconfig är huvudkonfigurationsfilen som används av dtlogin. Den anger inloggningsresurser och adresser för ytterligare filer som krävs av dtlogin. Följande exempel visar några rader som finns i standard-Xconfig-filen:

```
Dtlogin.errorLogFile:      /var/dt/Xerrors

Dtlogin.servers:          /usr/dt/config/Xservers

Dtlogin*session:          /usr/dt/bin/Xsession
```

Mer information om den här filen finns i Bilaga A". Du kan också granska de förklarande kommentarerna i själva filen.

Xservers

dtlogin, liksom XDM (X Display Manager) som den bygger på, använder filen Xservers för att ange hur den lokala Xservern startas. Som standard ser filens sista rad ut så här:

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

- `:0` innebär att Xserverns skärm är `<localhost:0>`.
- `local` anger att den nya X11-servern ska startas lokalt.

Obs! Ersätt `local` med `foreign` om du vill ansluta till en aktiv X11-server.

- `console` innebär att kommandoradinloggningens användare är `/dev/console`.
- `/usr/openwin/bin/Xsun` är sökvägen till X11-servern.

Obs! Miljöerna Solaris CDE och OpenWindowsTM kör samma X11-server.

Mer information om den här filen finns i direkthjälpen (`man`) till `dtlogin(1)`. Du kan också granska de förklarande kommentarerna i själva filen.

Starta inloggningsservern

Inloggningsservern startas för det mesta automatiskt när systemet startas. Du kan också starta inloggningsservern från en kommandorad, men du måste logga in som rotanvändare först.

Så här anger du att inloggningsservern ska starta när systemet startas:

- ◆ **Skriv följande kommando och tryck på Retur:**

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

Det här lägger till filen `S99dtlogin` i mappen `/etc/rc2.d`. Inloggningsservern startar automatiskt när du startar om.

Så här hindrar du att inloggningsservern startar automatiskt när systemet startas:

- ◆ **Skriv följande kommando och tryck på Retur:**

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

Så här startar du inloggningsservern från en kommandorad:

- ◆ **Skriv följande kommando och tryck på Retur:**

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

Obs! Även om det går att starta inloggningsservern från kommandoraden för tillfällig konfigurationstestning, bör inloggningsservern normalt startas när systemet startas.

Så här avbryter du inloggningsservern, Xserver, och hela Solaris CDE-skrivbordet:

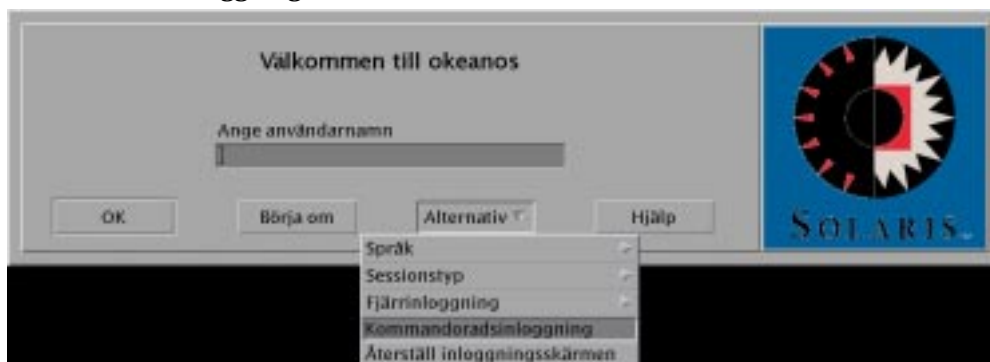
- ◆ **Skriv följande kommando och tryck på Retur:**

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

Mer information om skrivbordets konfigurationshjälpmedel, dtconfig, finns i **Bilaga A**. Där finns en kopia a

Så här avslutar du Solaris CDE:s inloggningsskärm med hjälp av alternativet – Kommandoradsinloggning:

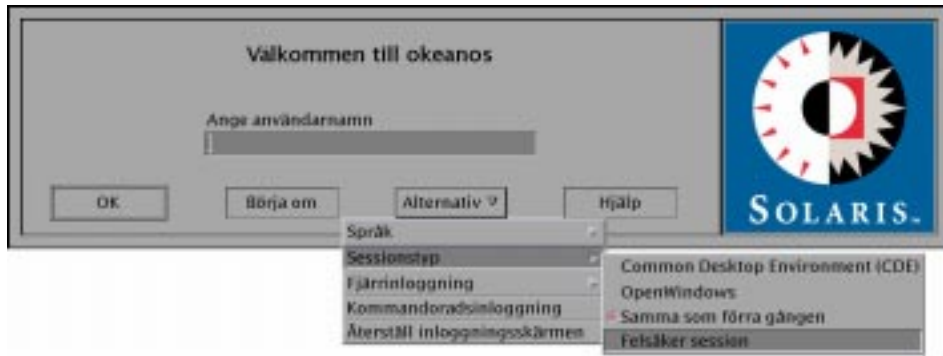
- ◆ **Välj Kommandoradsinloggning på Alternativ-menyn på Solaris CDE:s inloggningsskärm. När skärmen är tom trycker du på Retur för att visa en ledtext för inloggning.**



Obs! Skrivbordets inloggningsdemon kommer att starta skrivbordets inloggningsskärm automatiskt när du är färdig med kommandoradsinloggningen.

Så här startar du ett enstaka terminalemuleringsfönster:

- ◆ **Välj Felsäker session på undermenyn Session på Alternativ-menyn på Solaris CDE:s inloggningsskärm.**



Om du vill låta Xservern stå på väljer du Felsäker session på undermenyn Session. Det här startar ett `xterm`-fönster. Alternativet Felsäker session är alltid tillgängligt, även när alternativet Kommandoradsinloggning inte är tillgängligt.

Ansluta en installerad CDE från en annan installationsadress för en arbetsstation eller nätverksserver

Du kan montera en installerad CDE från en annan installationsadress för arbetsstation eller nätverksserver om du inte har det nödvändiga diskutrymme på din arbetsstation, eftersom den proceduren inte använder något lokalt diskutrymme.

Obs! Eftersom mappstrukturen `/usr/dt` skiljer sig åt mellan olika versioner av Solaris och klientarbetsstationer (SPARC-mappstrukturen skiljer sig från Intel-mappstrukturen), måste klientarbetsstationen ansluta en lämplig `/usr/dt`-bild för NFS-server. Ett SPARC-system med Solaris 2.5 bör exempelvis ansluta `/usr/dt` från ett annat Solaris 2.5-system med `/usr/dt`-installerad CDE.

▼ Ansluta en installerad CDE

1. Anslut en `/usr/dt`-mapp från en tidigare installerad arbetsstations- eller nätverksserver till mappen `/usr/dt` på din arbetsstation.
2. Skriv `/usr/dt/bin/dtconfig -inetd`
3. Aktivera Solaris skrivbordsinloggning genom att skriva:

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

4. Starta om arbetsstationen.

Mer information om skrivbordets konfigurationshjälpmedel, `dtconfig`, finns i Bilaga A. Där finns en kopia av direkthjälpen (man) till `dtconfig.1`.

▼ Koppla ur en ansluten CDE-mapp

1. Koppla ur Solaris skrivbordsinloggning genom att skriva:

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

2. Skriv `/usr/dt/bin/dtconfig -inetd.ow`

3. Koppla ur `/usr/dt`.

4. Starta om arbetsstationen.

Mer information om skrivbordskonfigurationshjälpmedlet `dtconfig` finns i Bilaga A. Där finns en kopia av direkthjälpen (man) till `dtconfig.1`.

Konfigurera skrivbordet för att använda flera skärmar

Standardinloggningen startar skrivbordet på en skärm. Du kan redigera filen `Xconfig` så att skrivbordet startar på flera skärmar. Du måste vara inloggad som rotanvändare för att ändra den här filen.

Obs! Om du vill redigera information om inloggningskonfiguration snabbare, använder du alternativet `Felsäker session` för att köra ett enda `xterm`-fönster i stället för hela skrivbordet.

▼ Starta skrivbordet på flera skärmar

1. Skapa en kopia av filen `Xserver` genom att skriva följande kommando och trycka på Retur:

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

Obs! /etc/dt/config/Xservers åsidosätter /usr/dt/config/Xservers.

2. Redigera filen /etc/dt/config/Xservers och ange två bildskärmsminnen (skärmar):

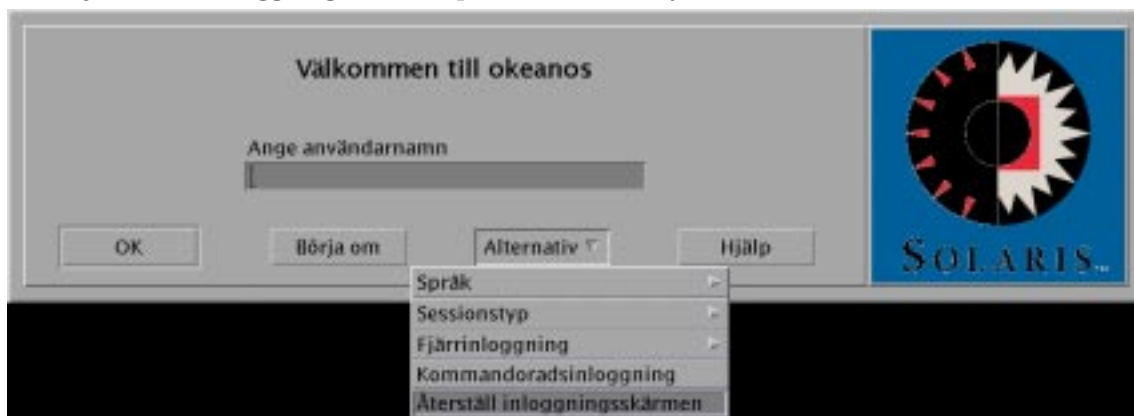
a. Leta rätt på raden:

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

b. Lägg till två -dev-alternativ i slutet av raden. I följande exempel heter de /dev/fb0 och /dev/fb1. Lägg till följande, föregånget av ett blanksteg:

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

3. Välj Återställ inloggningsskärmen på Alternativ-menyn.



Obs! När du kör Solaris CDE på flera skärmar visas bara Frontpanelen på den första skärmen. Om du vill starta program på de andra skärmarna använder du undermenyn Program på menyn Arbetsyta. Du visar undermenyn Program på menyn Arbetsyta genom att trycka på musknapp 3 över ett tomt avsnitt av skrivbordet. Om du använder en mus med två knappar trycker du på musknapp 2.

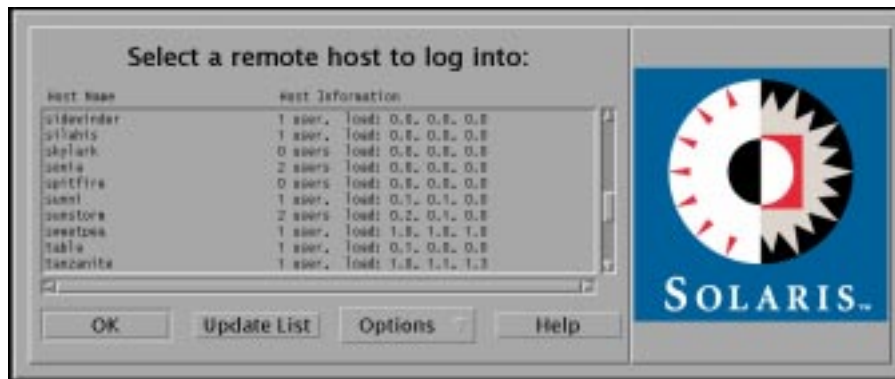
Nätverksanslutna skrivbord

Solaris CDE Inloggningshanteraren är nätverksmedveten. Inloggningskärmar svarar som standard på frågor från Väljaren i Solaris CDE.

Så här visar du en lista med fjärranslutna värddatorer:

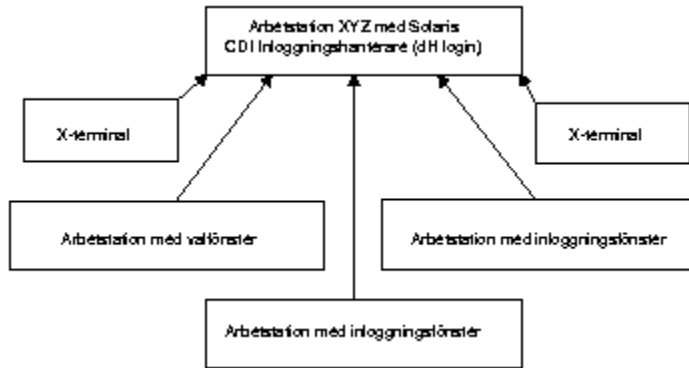
- ♦ **Välj Välj server från en lista på Fjärrinloggning-menyn på Solaris CDE inloggningskärm.**

Mer information om hur du kan köra Väljaren oberoende av inloggningskärmen finns i "Använda arbetsstationer som X-terminaler" på sidan 94. Följande skärm visar ett exempel på en lista i Väljaren över tillgängliga servrar.



Du kan markera en ledig server i listan och använda dess resurser via Väljaren för att logga in på Solaris CDE.

En nätverksansluten arbetsstation som kör Solaris CDE Inloggningshanteraren kan både stödja en lokal skrivbordsanvändare och flera fjärranvändare inklusive Xterminal-användare genom att arbetsstationen används som Solaris CDE-skrivbordsserver. (Se Figur 6-1.)

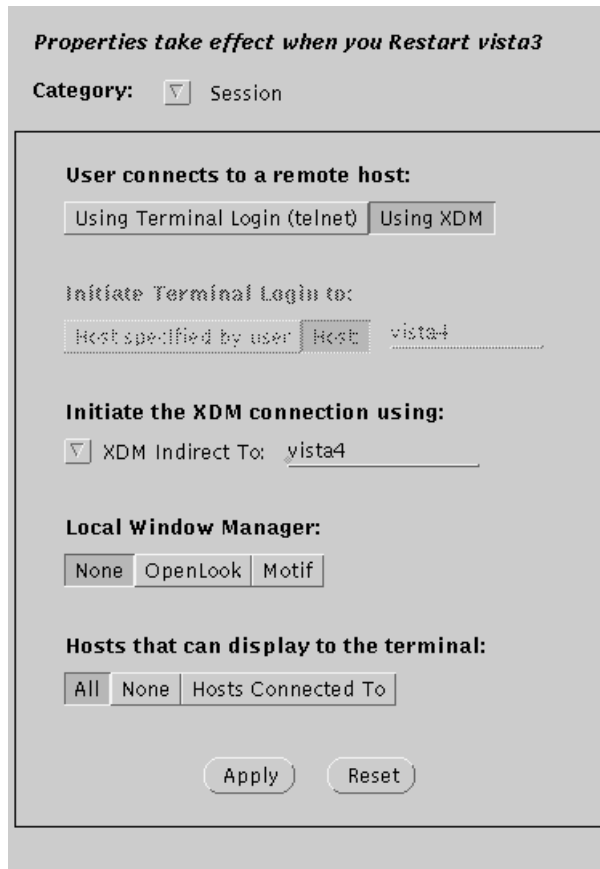


Figur 6-1 Nätverksanslutna skrivbord

Obs! Figur 6.1 kan beskriva en blandad maskinvarumiljö.

Använda X-terminaler

Alla X-terminaler som stöder XDM-protokollet kan använda Väljaren för att logga in på Solaris CDE. Mer information om hur du kan köra Väljaren finns i "Använda arbetsstationer som X-terminaler" på sidan 94. Programvaran till SPARC Xterminal™ version 2.0 (eller senare) fungerar bra med Solaris CDE. Klicka på Egenskaper på Xterminalens kontrollfönster och välj Session som kategori för att visa installationsskärmen som visas i Figur 6-2.

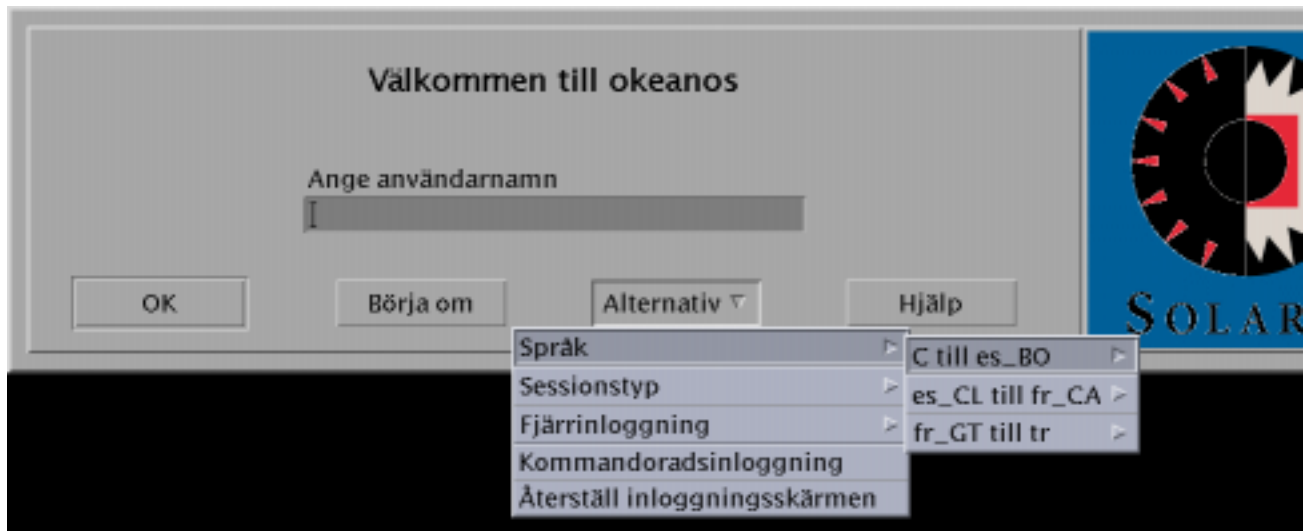


Figur 6-2 Session, installationskärm

Välj Ingen som Lokal fönsterhanterare, och XDM indirekt som XDM-anlutning till fjärrsystemet som kör Solaris CDE.

Inloggningspråkområde och teckensnittssökväg

Markera det språk du vill ha när du loggar in på Solaris CDE. Följande Solaris CDE-inloggningskärm visar alternativet Språk på Alternativ-menyn.



På arbetsstationer innehåller teckensnittssökvägar automatiskt teckensnitt (och aliasnamn) som är associerade med både det språk du vill använda och systemets baspråksområde, "C". För X-terminaler tillhandahålls de här teckensnitten automatiskt av en X11-teckensnittsserver.

Använda arbetsstationer som X-terminaler

Om du har äldre arbetsstationer eller arbetsstationer med 16 MB minne eller mindre, kan du använda dem som X-terminaler när de laddas med Solaris 2.4-programvara eller senare (eller Solaris 2.3-programvara med Xservers stora korrigeringsfil).

Obs! Om teckensnitten på arbetsstationens skärm inte visas normalt när du utför nedanstående procedur kanske inte Solaris CDE är installerat på värdarbetsstationen. Använd kommandoradsalternativet `pkgadd` för att lägga till Solaris CDE:s teckensnittspaket `SUNWdtft()` separat från en utgåva av Solaris CDE 1.0.x.

▼ Använda Väljaren för att markera en värd-CDE-inloggning

1. Avsluta det fönstersystem som körs.

2. Starta Väljaren. Skriv kommandon (med hjälp av `ssh`) från systemfönstrets kommandorad:

```
setenv OPENWINHOME /usr/openwin
```

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

Xserver startar och visar Val-fönstret från värddatorn.

3. Välj den inloggningsvärd du vill ha från Väljarskärmen.

▼ Använda en specifik värd-CDE-inloggning

- ◆ Skriv följande kommando och tryck på Retur:

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

Xserver startar och visar inloggningsskärmen från värddatorn. Alternativet `-once` avslutar servern efter en inloggnings-/utloggningssession. Om alternativet `-once` inte läggs till visas inloggningsskärmen igen efter utloggningen från Solaris CDE.

▼ Använda den första tillgängliga värdinloggningen

- ◆ Skriv följande kommando och tryck på Retur:

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

Det här startar Xserver. Kommandot skickar en fråga på det lokala delnätet efter en XDM-inloggningstjänst (X Display Manager). Om några system på delnätet kör Solaris CDE Inloggningshanteraren (eller något annat XDM-baserat inloggningsfönster), placerar den första värddator som svarar sitt inloggningsfönster på skrivbordet.

Särskilda CDE-konfigurationer

Det här avsnittet handlar om särskilda konfigurationer.

Anpassa utskrift av brev

CDE Postverktyg skriver ut brev med hjälp av funktionen Print som är definierad för datatypen DTMAIL_FILE. Den använder inte utskriftskommandot som anges av OpenWindows Postverktyg. Om du vill ändra inställningarna för utskrift måste du ändra funktionen Print.

Ändra funktionen Print

1. Skapa följande fil i en textredigerare:

```
Hemmapp/.dt/types/dtmail.dt
```

2. Skriv följande rader i den här filen:

```
#

# Åsidosätter standardinställningarna för funktionen Print för brevlådor

#

ACTION Print

{

LABEL          Print

ARG_TYPE       DTMAIL_FILE

TYPE           COMMAND

WINDOW_TYPE    NO_STDIO

EXEC_STRING    sh -c ' \

                dtmailpr -p -f %(File)Arg_1% | mp -m -l | \

                dtlp -u %(File)Arg_1%;'
```

Obs! `dtmailpr` är ett utskriftsfilter som sorterar bort bilagor, vilket resulterar i en ström av oformaterad text. `dtlp` är CDE:s standardgränssnitt för `lp`. `%(File)Arg_1%` är filen som skrivs ut.

3. Ändra fältet `EXEC_STRING` så att det innehåller de utskriftskommandon du vill ha.

4. Starta om Postverktyget.

Konvertera kalendrar till nya dataformat

Det tånjbara dataformatet i version 4 är ett nytt format som stöds av CDE Kalender. OpenWindows Kalender kan inte läsa det här formatet. Om du behöver växla fram och tillbaka mellan plattformarna OpenWindows och CDE, bör du inte konvertera kalendern till dataformatet version 4. Du kan emellertid använda kommandofilen `sdtcm_convert` för att rensa i kalendern.

Obs! Du kan ta reda på kalenderns dataversion genom att välja Om Kalender på Kalenderns Hjälp-meny.

Använd annars kommandofilen `sdtcm_convert` för att konvertera kalendern till dataformatet version 4.

Du får mer information om den här filen i direkthjälpen (man) till `sdtcm_convert(1)`.

Obs! Om du bara installerade CDE-paketet för slutanvändare installerades inte direkthjälpen. För att kunna visa direkthjälpen måste du installera den med hjälpmedlet `pkgadd`. Mer information finns i *Solaris CDE: Installationshandbok*.

Lägga till AnswerBook-paketet från nätverket

När du installerar Solaris CDE ger installationsmenyn möjlighet att installera AnswerBook:s CDE-paket. Om du väljer YES installeras AnswerBook-paketet i mappen `/usr/dt/share/answerbooks/språk` (*språk* är det lokala namnet på den specifika AnswerBook-översättningen).

Eftersom AnswerBook-paketet kräver 109 MB diskutrymme kanske du inte vill installera det. Följande två alternativ är emellertid tillgängliga:

- Du kan ansluta (eller länka från `/net/. . .`) AnswerBook-paketet till `/usr/dt/share/answerbooks/språk`.

- Du kan lägga till miljövariabeln `AB_CARDCATALOG` i *Hemmapp*/.dtprofile. Om du t ex har en exporterad version i ditt nätverk lägger du till följande rad i *Hemmapp*/.dtprofile :

```
export AB_CARDCATALOG=/net/värddatornamn/usr/dt/share/answerbooks \  
  
/språk/ab_cardcatalog
```

Installera CDE-miljö utanför CDE-skrivbordet

Miljövariabler som används av dina Solaris CDE-program kan installeras utanför CDE-skrivbordet. Du kan t ex logga in på en fjärrarbetsstation och visa ett CDE-program tillbaka till din arbetsstation. Använd CDE-hjälpmidlet `dtsearchpath` för att installera olika CDE-skalmiljövariabler.

För Bourne- och Korn-skal skriver du följande kommando:

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

För C-skal skriver du följande kommando:

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

Du kan följa det här kommandot genom att ange `DISPLAY` tillbaka till den lokala arbetsstationen och sedan fjärrköra CDE-programmet. Resultaten visas på den lokala arbetsstationen.

Obs! I det här exemplet ingår Bourne- och Korn-skalets `dtsearchpath`-syntax i CDE:s exempelimplementering, som är tillgänglig på alla plattformar som kör CDE. C-skalets alternativ (`-c`) är emellertid bara tillgängligt på Sun-plattformar.

Skrivbordsmiljöfil

CDE-skrivbordet tillhandahåller en körtidsmiljö för OpenWindows DeskSetTM-programmen. Det krävs inga särskilda CDE-installationer för den här funktionen. Vissa av miljöns inställningar kommer från följande fil:

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

Om du vill lägga till eller ta bort miljöinställningar för en viss arbetsstation, kan den här filen antingen redigeras på plats eller först kopieras till följande filadress:

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

Ett exempel på ett tillägg kan vara att starta OpenWindows gamla hjälpprogram Tangentbord för användning men OpenWindows DeskSet-program. Eftersom de flesta program på ett CDE-skrivbord (och de flesta Sun-användare) inte använder

hjälpprogrammet Tangentbord, har detta utelämnats från standardstartsekvensen för att förbättra skrivbordets startprestanda.

Se även kommentarerna i filen `0015.sun.env` om du vill ha ytterligare information om valfri start av OpenWindows Tangentbord (`vkdb`).

Använda diskett och CD-skiva med Apple Macintosh programmiljö

Om du har installerat Apple Macintosh Application Environment (MAE) version 1.0 och tänker använda flyttbara media (disketter eller CD-skivor) via programmet OpenWindows Filhanteraren, måste du redigera filen `/etc/rmmount.conf`. Det säkerställer att MAE fungerar korrekt med flyttbara media i Filhanteraren.

Ändra filen `/etc/rmmount.conf`

1. **Bli rotanvändare genom att skriva `su` och skriv därefter ditt lösenord.**
2. **Ändra till mappen `/etc`. Skriv följande kommando och tryck på Retur:**

```
# cd /etc
```

3. **Öppna filen `/etc/rmmount.conf` i en textredigerare.**

4. **Flytta följande rad i funktioner till slutet av listan:**

```
action floppy action_macfs.so
```

Exempel:

```
# more rmmount.conf
```

```
# @(#)rmmount.conf 1.2      92/09/23 SMI
```

```
#
```

```
# Konfigurationsfil för anslutning av flyttbara media.
```

```
#
```

```
# Filsystemidentifiering
```

```
ident hsfs ident_hsfs.so cdrom

ident ufs ident_ufs.so cdrom floppy

ident pcfs ident_pcfs.so floppy

ident macfs ident_macfs.so floppy

# Funktioner

action cdrom action_filemgr.so

action floppy action_filemgr.so

action floppy action_macfs.so
```

5. Spara och avsluta.

När MAE är installerat placeras raden `macfs action` i filen `rmmount.conf`. Om MAE avinstalleras tas raden bort.

Den här ändringen gör att du kan använda flyttbara media i OpenWindows eller CDE:s Filhanteraren och fortfarande använda Macintosh-disketter i MAE. Den hindrar dock MAE från att ta emot oformaterade disketter, oläsbara disketter eller DOS-disketter. Alla dessa disketter går till OpenWindows Filhanteraren med den här installationsproceduren.

Om du vill att MAE ska ha åtkomst till oläsbara disketter, oformaterade disketter eller DOS-disketter, måste raden `action floppy action action_macfs.so` placeras före raden `action floppy action_filemgr.so`.

XServer och drivrutinen Power Management

Drivrutinen Power Management kontrollerar strömmen till skärmen Solaris 2.6. Om skärmen står oanvänd under en period försätts den i viloläge för att minska elförbrukningen. När du använder Solaris CDE-skrivbordet sköter Xserver om Power Management, men skärmen försätts bara i viloläge (efter en period som oanvänd) om CDE:s Inställningar har skärmläckaren angiven till en tom skärm (det här är

standard). Skärmen återställs till normalt tillstånd efter ett par sekunder när den återaktiveras av användaren.

Om du vill avaktivera Power Management när du använder Solaris CDE, ändrar du skärmläckaren i Inställningar från standardinställningen (tom skärm) till ett skärmmönster. Den kan också avaktiveras genom att du kör följande kommando i terminalemuleringsfönstret:

```
% xset s noblank
```


Konfigurera skrivbordet i nätverk

Skrivbordet är utformat för att fungera i nätverksintensiva miljöer. Skrivbordets arkitektur gör att systemadministratörer kan distribuera dataresurser över hela nätverket, inklusive:

- Program.
- Datafiler för program.
- Skrivbordssessionstjänster (skrivbordsprogram som Inloggningshanteraren och Filhanteraren).
- Hjälptjänster (hjälpfiler kan placeras på en central hjälpserver)
- "Nätverk på skrivbord – översikt" på sidan 103
- "Generella steg för att konfigurera nätverket" på sidan 108
- "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108
- "Konfigurera skrivbordsklienter och servrar" på sidan 111
- "Administrera programtjänster" på sidan 116

Nätverk på skrivbord – översikt

Operativsystemet erbjuder flera nätverkstjänster, inklusive distribuerade filsystem och fjärrkörning. X-hanteraren erbjuder ytterligare nätverksfunktioner, inklusive tillgång till fjärrskärmar och tjänster för säkerhetshantering.

Skrivbordet placerar ett användargränssnitt ovanpå de här nätverksfunktionerna. Avsikten med gränssnittet och dess underliggande arkitektur är att göra nätverkssystem:

- Enklare att använda. Användare kan köra program och komma åt datafiler utan att veta exakt var i nätverket programmen och filerna finns.
- Enklare att administrera. Skrivbordet innehåller verktyg för programintegrering och nätverkssökvägar som gör det enklare för systemet att hitta fjärrdata och fjärrinstallerade program. Dessutom gör skrivbordets filnamnsavbildningsprocess det lättare att administrera komplexa nätverk som innehåller flera servrar.
- Flexibelt. Skrivbordets administrativa funktioner är utformade för vissa vanliga nätverkssituationer, men kan även användas för många andra anpassade nätverkskonfigurationer.

Typer av nätverkstjänster för skrivbord

Via ett nätverk kan en användare komma åt olika datatjänster som distribueras på andra system, t ex:

- Skrivbordssessioner och dess program – exempelvis Arbetsytehanteraren och Filhanteraren
- Andra program
- Datafiler

I nätverksterminologi används uttrycket *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:

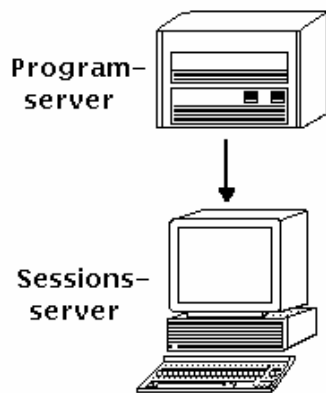
Skärmar – där X-hanteraren körs

Inloggnings-/Sessionsserverar – där skrivbordsprogrammen (Inloggningshanteraren, Arbetsytehanteraren och liknande) körs

Programserverar – där andra program körs

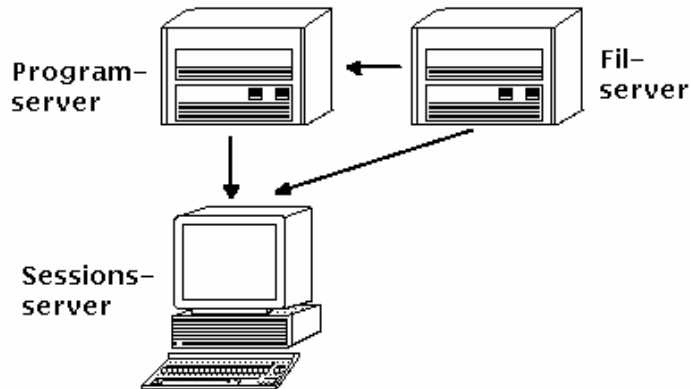
Filservrar – där data som används av program finns

En av de vanligaste nätverkskonfigurationerna innebär att system ansluter till en programserver. Figur 7-1 föreställer en arbetsstation som använder en programserver. X-hanteraren och skrivbordssessionen körs på arbetsstationen.



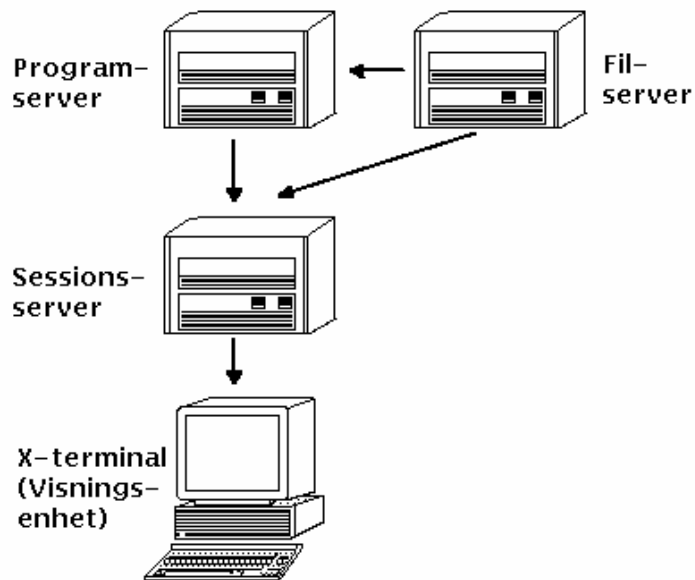
Figur 7-1 Programserver ger en skrivbordssession tillgång till olika tjänster.

Nätverk brukar också använda filserver till att lagra stora mängder data. Ett program som körs på en programserver eller ett skrivbordsprogram (Filhanteraren behöver t ex tillgång till datafiler för att kunna visa dem i ett filhanterarfönster) kan använda sig av dessa data.



Figur 7-2 Filserver ger program- och sessionsserver tillgång till data.

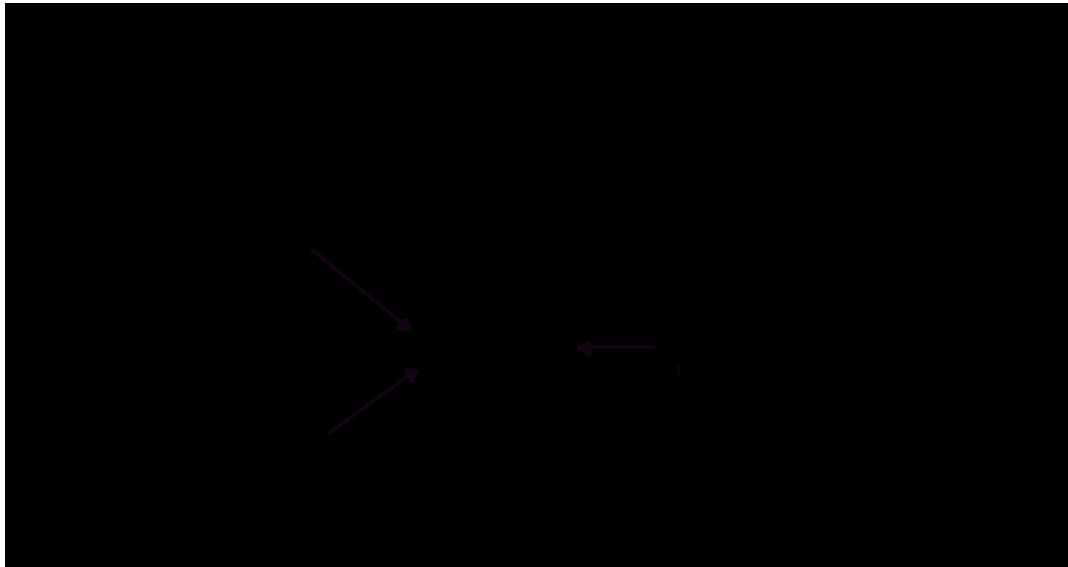
X-terminaler kör X-hanteraren och kommer åt skrivbordssessionstjänster från ett annat system.



Figur 7-3 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 filserverar, görs tillgängliga för programserverar.



Figur 7-4 De tjänster som behövs för en skrivbordsprogramserver kan distribueras

Sammanfattning – Typer av servrar

Skärm – systemet som kör X-hanteraren.

Inloggnings- och sessionsserver – systemet som kör skrivbordssessionen (Inloggningshanteraren, Sessionshanteraren, Fönsterhanteraren, Filhanteraren och liknande).

Programserver – ett system där ett program körs. Kallas också *värddator för exekvering*.

Filserver – ett system där datafiler för program lagras.

Hjälpsserver – ett system där hjälpfiler lagras.

Funktionsdatabasserver – ett system där filer som innehåller funktions- och datatypsdefinitioner lagras.

Ikonservrar – ett system där ikonfiler lagras.

Nätverket kan innehålla ytterligare servrar, exempelvis en lösenordsserver, postserver, videoservert osv.

Generella steg för att konfigurera nätverket

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

1. Konfigurera operativsystemets basnätverkstjänster.

Dessa tjänster tillhandahålls av operativsystemet och är nödvändiga för skrivbordet. Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.

2. Installera och konfigurera skrivbordets program och tjänster för nätverk.

Detta är tjänsterna som behövs för skrivbordet, oavsett vilken typ av klient och server som installeras. Se "Konfigurera skrivbordsklienter och servrar" på sidan 111.

3. Konfigurera en viss typ av server eller klient.

När du t ex konfigurerar en programserver krävs andra metoder än när du konfigurerar en filserver. Se "Administrera programtjänster" på sidan 116.

Konfigurera operativsystemets basnätverkstjänster 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.
- Kommandot `lp:s` skrivarkö 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 programserverar, filserverar och system som innehåller nätverksskrivare.
- Alla sessionsserverar som användaren använder. Sessionsserverar används normalt med X-terminaler.

Konsekvent användar- och grupp-ID

UNIX-användare identifieras av ett inloggningsnamn och ett numeriskt användar-ID (UID). 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 (GID). I ett nätverk ska alla system använda samma gruppnamn och grupp-ID.

Mer information finns i direkthjälpen (man) för `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 monterade i alla lämpliga system.

Normalt måste du tillhandahålla följande fjärrfilåtkomst:

- Användarens hemmapp måste delas med alla skrivbordsklienter och serversystem. Detta är nödvändigt eftersom:
 - Hemmappen innehåller datafiler som program i fjärrsystem ska ha tillgång till. Program som använder datafiler har ofta hemmappen som standardadress för datafilerna.
 - Hemmappen är standardverifieringsmapp för `dtspc`. Mer information om `dtspc` finns i "Konfigurera styrdemon för underprocesser" på sidan 115.
- Om användaren kräver åtkomst till datafiler som inte finns i hemmappen, måste dessa datafiler delas av alla skrivbordsklienter och serversystem som arbetar med datafilerna.

- Mapper för skrivbordsinstallation och skrivbordskonfiguration (`/usr/dt` och `/etc/dt`) måste delas av alla skrivbordsklienter och serversystem, så att alla användarens program har tillgång till samma skrivbordskonfigurationsfiler.

Hemmapp för nätverk

Arbetsstationer i nätverk fungerar mest effektivt när användaren har en enstaka hemmapp som delas mellan alla klienter och serversystem i nätverket.

Med en nätverksansluten hemmapp kan användaren använda olika system i nätverket utan att egna anpassningar och konfigurationer går förlorade. Detta beror på att egna anpassningar, och informationen som krävs för att återställa föregående session, sparas i undermapper till hemmappen.

En hemmapp behövs också för:

- Den standardmässiga X-behörighetsmekanismen (se även "Konfigurera som X-verifiering" på sidan 111).
- Styr demon för underprocesser. Denna demon startar fjärrprogram och måste kunna skriva i användarens hemmapp.

Konsekventa filnamn

Du bör 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 hemmapp är tillgänglig som `/users/inloggningsnamn` genom att skapa en symbolisk länk till mappens verkliga plats.

Konfigurera åtkomst av fjärrskrivare

Skrivbordet använder kommandot `lp:s` skrivarkö för att komma åt lokala skrivare och fjärrskrivare. Mer information om hur du konfigurerar kommandot `lp` finns i direkthjälpen (`man`) för `lpadmin`.

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

Se till att använda konsekventa namn på utskriftsenheter. Om t ex en viss skrivare kallas `skrivare` i systemet som den är direktansluten till, bör samma namn användas av alla system som har fjärråtkomst till skrivaren `skrivare`.

Konfigurera e-post

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

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

Konfigurera som X-verifiering

Skrivbordet använder standard `x`-mekanism för att verifiera om fjärrprogram (X-klienter) har åtkomst till en lokal skärm. Det enklaste sättet att göra detta är ge alla användare en nätverkshemmapp. Detta garanterar att följande krav uppfylls:

- Användaren måste ha läs- och skrivbehörighet till filen `Hemmapp/.Xauthority`.
- Filen `.Xauthority` på en programserver måste innehålla en "magic cookie" för skärmen där programmet kommer att köras.

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

Konfigurera skrivbordsklienter och servrar

Det här avsnittet beskriver krav för nätverkskonfiguration som är specifika för arbetsstationen – det vill säga, de här funktionerna tillhandahålls av skrivbordet snarare än av det grundläggande 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 deras klienter.

Konfigurera inloggnings- och sessionstjänster

En inloggnings- och sessionsserver är ett system som tillhandahåller skrivbordstjänster (Inloggningshanteraren, Sessionshanteraren, Filhanteraren, Fönsterhanteraren osv) på en skärm med X-hanterare.

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 tillgängliga för både X-terminaler och arbetsstationer.

Inloggningshanteraren är en skrivbordskomponent som svarar för inloggnings tjänster på andra skärmar. När användaren har loggat in startas Sessionshanteraren åt användaren.

Mer information om hur du konfigurerar inloggnings- och sessionsserverar samt X-terminaler finns i "Visa en inloggningsbild på en nätverksskärm" på sidan 5.

Konfigurera andra programrelaterade tjänster

Om ikoner i Filhanteraren finns på en ikonserver, måste sessionsservern montera dessa filer.

- Programserverar
- Databasserverar
- Hjälpservrar
- Hjälpservrar

Konfigurera skrivbordsklienter och -serverar

1. Gör operativsystemets nätverkskonfiguration tillgänglig för skrivbordet.

Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.

2. Installera skrivbordet eller minimiuppsättningen med filer:

Du måste installera:

- Hela uppsättningen körtidsfiler i Common Desktop Environment
- dessa uppsättningar filer: CDE-MIN och CDE-TT

Obs! Installation och filuppsättningar kan variera mellan olika leverantörer.

3. Konfigurera systemet för filnamnsdatabasens serverdemon i

ToolTalk `rpc.ttdbserver`.

Detta ska ske automatiskt när skrivbordet installeras. Mer information finns i "Konfigurera databasservern ToolTalk " på sidan 116.

4. Installera och konfigurera styrdemon för underprocesser (`dtspc`).

Detta ska ske automatiskt när skrivbordet installeras. Mer information finns i "Konfigurera styrdemon för underprocesser" på sidan 115.

5. Montera alla nödvändiga fjärrdata.

Data betraktas som "fjärrdata" om de finns i ett annat system än programmet som använder dessa data.

Exempel:

- Om ett program använder data som finns på en filserver, måste programmet montera dessa filer.
- Om ikoner till Filhanteraren finns på en ikonserver måste sessionsservern montera de filerna.
- Om ett nätverk använder en hjälpservare för hjälpfiler, måste sessionsservern och alla programservrar montera hjälpdata.

Mer information om monteringspunkter finns i nästa avsnitt. "Konfigurera monteringspunkter för fjärrfilsystem" på sidan 113.

Konfigurera monteringspunkter 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 monteras på olika platser i olika system, och därför måste gå att komma åt med olika namn. Filen `projekt/stor` på `sysA` kan t ex nås som `/net/sysA/projekt/stor` på `sysB`.

Krav för tilldelning av filnamn

För att tilldelningen ska göras korrekt måste något av följande vara uppfyllt:

- Kommandot `mount` används för att statiskt montera filsystem. Dessa typer av statisk montering är normalt konfigurerade i filer som `/etc/checklist`, `/etc/mnttab` eller `/etc/filesystems`.

För att tilldelningen av filnamn ska fungera korrekt mellan olika system, måste monterade filsystem 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 monteringar.

- Automontering används för att montera filsystem vid standardmonteringspunkten `/net`
- Automontering används för att montera filsystem till andra adresser än `/net` och miljövariabeln `DTMOUNTPOINT` ställs in för att ange monteringspunkten. Mer information finns i nästa avsnitt, "Ange värde på `DTMOUNTPOINT`" på sidan 114.

Mer information om automontering finns i direkthjälpen `(man) till automount(1M)`.

Ange värde på DTMOUNTPOINT

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

- Automonteringen används för att montera filsystem.
- Och fjärrfilssystem har monterats till andra adresser än /net.

DTMOUNTPOINT måste anges för processer, inklusive:

- Användarens skrivbordsprocesser som startas automatiskt när användaren loggar in, t ex Skrivbordshanteraren (dtwm) och Filhanteraren (dtfile)
- Systemprocesser, t ex rpc.ttdbserver och dtspc, 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 anger du DTMOUNTPOINT för de här processerna:

1. Redigera filen /etc/inetd.conf:

a. Leta reda på posten dtspc och lägg till:

```
-mount_point monteringspunkt
```

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

```
-m monteringspunkt
```

Om t ex automontering används med monteringspunkten /nfs, är posterna i /etc/inetd.conf:

```
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 proceduren på det system som läser om /etc/inetd.conf. Mer information finns i direkthjälpen (man) till inetd(1M).

3. Ange DTMOUNTPOINT på ett sådant sätt att dess värde ärvs av en användare som loggar in.

Detta kan utföras genom att ange variabeln i /etc/dt/config/Xsession.d. Mer information om olika miljöinställningar finns i .

Konfigurera styrdemon för underprocesser

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

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

Skrivbordets funktionsanropsbibliotek använder tjänsten SPC för att anropa fjärrfunktioner.

Konfigurera `dtspcd`

- ◆ **Bekräfta att `dtspc` är korrekt registrerad i både `/etc/services` och `/etc/inetd.conf`.**

Se direkthjälpen (`man`) till `dtspcd(1M)`.

SPC-säkerhet

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

Normalt är verifieringsmappen för `dtspcd` användarens hemmapp. Du kan dock konfigurera `dtspcd` till att använda en annan adress genom att ange alternativet `-auth_dir` i mappen `/etc/inetd.conf`. Mer information finns i direkthjälpen (`man`) till `dtspcd(1M)`.

Eftersom SPC-verifieringen baseras på filsystemverifiering, är tjänsten SPC bara lika 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 `dtspcd` i `/etc/services`.

Konfigurera miljövariabler för fjärrkörning

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 underprocessstyrningen för att bearbeta ytterligare miljövariabler innan variablerna placeras i programmets miljö.

Mer information om standardkonfiguration och hur du ändrar den finns i direkthjälparna (man) 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 ToolTalk databasserver och dess konfigurationsalternativ finns i direkthjälp (man) till `rpc.ttdbserver(1M)`.

Konfigurera meddelandeservern ToolTalk

Meddelandeservern ToolTalk är `ttsession`. Som standard krävs ingen konfiguration. Den startas av kommandofilen `Xsession` vid inloggningen.

Mer information om ToolTalk-meddelandeservern och dess konfigurationsalternativ finns i direkthjälp (man) till `ttsession`.

Konfigurera Kalenderdemon

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

Mer information om Kalenderdemon och dess konfigurationsalternativ finns i direkthjälp (man) till `rpc.cmsd(1)`.

Administrera programtjänster

I det här avsnittet beskrivs vissa konfigurationskrav för:

- Programservrar och deras klienter
- Skrivbordsservrar som tillhandahåller särskilda tjänster – databasservrar, ikonservrar och hjälpservrar

Här beskrivs också nätverkskrav för två speciella konfigurationer av nätverksprogram:

- Värddatorer för fjärrkörning

- Program som kör över olika monterade filsystem

Miljövariabler för sökvägar

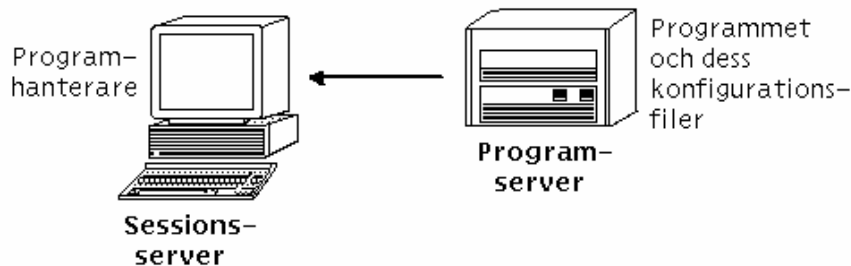
Skrivbordet använder en uppsättning miljövariabler för att ange vilken sökväg som används för att hitta skrivbordskonfigurationsfiler som exempelvis funktionsdatabaser, datatypsdata-baser, hjälpfiler och ikonfiler.

Mer information om hur miljövariabler för sökvägar används finns i direkthjälpen (man) till Kapitel 9, eller i direkthjälpen (man) till dtenvvar(5).

Konfigurera en programserver och dess klienter

I standardkonfigurationen för programservern innehåller programservern alla binära filer och konfigurationsfiler som är associerade med programmet, inklusive:

- Programmets körbara filer
- Standardkonfigurationsfilerna för program, t ex programstandardfiler, meddelandemapp och delade bibliotek.
- Skrivbordets konfigurationsfiler:
 - Funktions- och datatypsdefinitionsfiler
 - Ikonbilsdfiles
 - Hjälpdatafiler för skrivbordet.



Figur 7-5 Standardkonfiguration för programservern

Konfigurera en programserver

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

Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.

2. **Gör den allmänna skrivbordskonfiguration som krävs för servrar tillgänglig.**
Se "Konfigurera skrivbordsklienter och -servrar" på sidan 112.
3. **Installera programmet eller programmen.**
4. **Om ett program inte registreras automatiskt måste du registrera det.**
Se Kapitel 5.

Konfigurera klienten till en programserver

1. **Lägg till programservern i programmets sökväg.**
Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.
2. **Gör den allmänna skrivbordskonfiguration som krävs för klienter tillgänglig.**
Se "Konfigurera skrivbordsklienter och -servrar" på sidan 112.
3. **Gör den allmänna skrivbordskonfiguration som krävs för klienter tillgänglig.**
Hela systemet – ange och exporter variabeln `DTSPSYSAPPHOSTS` i `/etc/dt/config/Xsession.d/0010.dtpaths`
Egen – ange och exporter variabeln `DTSPUSERAPPHOSTS` i `Hemmapp/.dtprofile`
Följande rad i `/etc/dt/config/Xsession.d/0010.dtpaths` lägger t ex till ett system med värddatarnamnet `SysAAA` och `SysBBB` till programsökvägen:

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

Mer information om hur du anger programsökväg finns i:

- "Programsökväg" på sidan 134
- "Ange värde för en sökväg" på sidan 133

Konfigurera databas-, ikon- och hjälptjänster

Normalt installeras ett programs funktions- och datatypsdefinitioner, ikon- och hjälpdatafiler till samma system som programmet själv installeras till.

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.

- Hjälppfilerna 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 nätverket t ex använder flera sessionsservrar kan du skapa en hjälpservrar där alla hjälppfiler för skrivbordsprogram (Filhanteraren, Inställningar och liknande) lagras. Detta sparar diskutrymme eftersom hjälppfilerna då inte behöver dupliceras på varje sessionsserver.

Skapa en databas-, hjälp- eller ikonservrar

1. **Gör nätverkskonfiguration för operativsystemet som krävs av skrivbordet tillgänglig.**

Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.

2. **Gör den allmänna skrivbordskonfiguration som krävs för klienter tillgänglig.**

Se "Konfigurera skrivbordsklienter och -servrar" på sidan 112.

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 de mappar som automatiskt söks när systemet har utnämnts till programserver.

- Databasfiler: `/etc/dt/appconfig/types/språk`
- Hjälppfiler: `/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 fjärrkörning" på sidan 120.

Konfigurera sessionsservern att hitta en databas-, ikon- eller hjälpservrar

1. **Gör nätverkskonfiguration för operativsystemet som krävs av skrivbordet tillgänglig.**

Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.

2. **Gör den allmänna skrivbordskonfiguration som krävs för klienter tillgänglig.**

Se "Konfigurera skrivbordsklienter och -servrar" på sidan 112.

3. Lägg till databas-, ikon- eller hjälpdatabasen i lämplig sökväg.

- Om du har placerat datafilerna på den plats som anges i i "Skapa en databas-, hjälp- eller ikonserver" på sidan 119, kan du ändra programmets sökväg.
- Om du har placerat datafilerna någon annanstans måste du ändra den specifika sökvägen.

Om du t ex har placerat hjälpfilerna i mappen `/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 137
- "Ikonsökväg" på sidan 139
- "Hjälpsökväg" på sidan 141
- "Ange värde för en sökväg" på sidan 133

Speciella konfigurationer för nätverksprogram

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

- På en annan plats i systemet som innehåller funktionen – på en värddator för fjärrkörning
- Lokalt via filsystemmonteringar

Ange en värddator för fjärrkörning

I den typiska programserverkonfigurationen finns funktionsdefinitionen i samma system som programmets körbara filer. Funktioner kan dock skrivas för att köra 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 åt sessionsservern, även kallat en *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 köra `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 på SysDDD och, om detta inte går, försöka i SysEEEE.

```
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 hjälp av den värddatorqualificerade syntaxen - SysAAA:. En funktionsdefinition som hittas med det här elementet i sökvägen anger databasvärden till SysAAA. En funktion som hittas med delen /net/SysBBB... av sökvägen anger databasvärden till det lokala systemet eftersom syntaxen inte innehåller värddatorqualificeraren.

Konfigurera värddatorn för fjärrkörning

- 1. Gör nätverkskonfigurationen för operativsystemet som krävs av skrivbordet tillgänglig.**
Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.
- 2. Gör den allmänna skrivbordskonfiguration som krävs för servrar tillgänglig.**
Se "Konfigurera skrivbordsklienter och -servrar" på sidan 112.
- 3. Kontrollera att programmen är rätt installerade och konfigurerade för lokal exekvering.**

Konfigurera systemet som innehåller funktionsdefinitionen

- 1. Gör den nätverkskonfiguration för operativsystemet som krävs av skrivbordet tillgänglig.**
Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.
- 2. Gör den allmänna skrivbordskonfiguration som krävs för servrar tillgänglig.**
Se "Konfigurera skrivbordsklienter och -servrar" på sidan 112.
- 3. Skapa och installera funktionsdefinitioner och programgrupper.**

Se "Skapa funktioner som kör program i fjärrsystem" på sidan 192 och "Skapa och hantera allmänna programgrupper" på sidan 46.

Konfigurera sessionsservern

1. **Gör den nätverkskonfiguration för operativsystemet som krävs av skrivbordet tillgänglig.**

Se "Konfigurera operativsystemets basnätverkstjänster för skrivbordet" på sidan 108.

2. **Gör den allmänna skrivbordskonfiguration som krävs för klienter tillgänglig.**

Se "Konfigurera skrivbordsklienter och -servrar" på sidan 112.

3. **Ändra funktionssökvägen till att omfatta databasvärden.**

Se "Databassökväg (funktion/datatyper)" på sidan 137.

4. **Ändra programsökvägen till att omfatta värddatorn för exekvering.**

Se "Programsökväg" på sidan 134.

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 7-6 Exekvering via monteringspunkter

Konfigurera programservern

Ingen speciell konfiguration behövs.

Konfigurera 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, meddelandemappar och delade bibliotek.

Konfigurera och administrera utskrifter från skrivbordet

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 välj Skriv ut från Markera-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" på sidan 125
 - "Ikonbilder till skrivaren" på sidan 127
 - "Konfigurera standardskrivare" på sidan 128
 - "Vad som sker vid utskrift" på sidan 129

Lägga till och ta bort skrivare

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

▼ Lägga till en skrivare till skrivbordet

1. **Lägg till en skrivare till systemets konfiguration.**

Följ instruktionerna i dokumentationen för systemadministratörer 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.

▼ Ta bort en skrivare från skrivbordet

1. Ta bort skrivaren från systemkonfigurationen.

Följ instruktionerna i dokumentationen för systemadministratörer för operativsystemet.

2. Starta om Utskriftshanteraren genom att dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg 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 informationen 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 mapp 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 139.

Ikonfilnamn och ikonfilstorlekar

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 216.

typ – pm (färgat bildpunktsmönster), bm (bitmapp).

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

Mer information om hur du skapar ikoner finns i Kapitel 14.

▼ Ändra ikon, skrivarnamn och beskrivning globalt

Du ska ändra de globala skrivaregenskaperna så fort du har lagt till skrivaren, innan användarna har ändrat dem med utskriftshanteraren. När användarna väl har ändrat skrivaregenskaperna 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. I fältet `ICON` uppdaterar du *basnamn* till den nya ikonens basnamn.
2. I fältet `LABEL` uppdaterar du *etikettnamn* till den nya titeln för skrivaren.
3. Uppdatera texten i fältet `DESCRIPTION`.
Här kan du skriva var skrivaren finns, typen av skrivare och skrivarkontakter.
Om du vill skriva mer än en rad, placerar du ett \ i slutet av raden. Exempel:

```
DESCRIPTION    Det här är en PostScript-skrivare i byggnad 1 \
                Rum 123. Ring 555-5555 vid problem.
```

Konfigurera standardskrivare

Standardskrivaren används när användaren:

- Släpper ett objekt på skrivarikonen på Frontpanelen
- Markerar ett objekt i Filhanteraren sedan väljer Skriv ut från vald meny eller från ikonens objektmeny
- Skriver ut från program som använder standardskrivare

▼ Ändra 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 ska du kopiera den från `/usr/dt/config/Xsession.d/0010.dtpaths`.
2. I fältet `LPDEST=skrivare` uppdaterar du *skrivare* till den nya destinationen för standardutskrift.
Om raden inte finns ska du lägga 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 enskild användare standardskrivare:

- ◆ **Kopiera en annan skrivare till Frontpanelen från panelundermenyn Egen skrivare.**
Om du vill ändra standardskrivaren för en enskild användare måste du:

1. Gå till hemmappen och öppna filen `.dtprofile`.
2. Lägg till eller redigera en rad som anger värden för LPDEST miljövariabel:

```
LPDEST=skrivarenhet export LPDEST
```

Om du använder `bash` är syntaxen:

```
setenv LPDEST skrivarenhet
```

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

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

Om du använder `bash` är syntaxen:

```
setenv LPDEST laser3d
```

Vad som sker vid utskrift

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

1. Systemet söker i datatyps-databasen efter det släppta objektets definition.
2. Om det finns en unik skrivarfunktion för datatypen (anges med fältet `ARG_TYPE` i utskriftsfunktionen) kommer den användas, annars kommer utskriftsfunktionen (`dtlp`) att användas. Om t ex 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` för utskrift.

Skrivbordets sökvägar

Skrivbordet använder sökvägar för att hitta program och tillhörande skrivbordsfiler.

- "Skrivbordets sökvägar och deras miljövariabler" på sidan 132
- "Ange värde för en sökväg" på sidan 133
- "Programsökväg" på sidan 134
- "Databassökväg (funktion/datatyper)" på sidan 137
- "Ikonsökväg" på sidan 139
- "Hjälpsökväg" på sidan 141
- "Lokaliserade sökvägar" på sidan 143

Skrivbordet tillhandahåller fyra sökvägar, enligt beskrivningen i Tabell 9-1.

TABELL 9-1 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) samt frontpanelsfiler (*.fp-filer).
Ikoner	Används för att ange ytterligare adresser för ikoner.
Hjälpdata	Används för att ange ytterligare adresser för hjälpdata.

Sökvägen kan innehålla både lokala mappar och fjärrmappar. När en sökväg omfattar fjärrmappar måste du konfigurera fjärrfilsåtkomst för dessa adresser. Ett system hittar t ex program på en programserver eftersom programservern är listad i programmets sökväg.

När en sökväg innehåller en fjärradress måste du konfigurera fjärrfilsåtkomst för adressen. Mer information finns i "Konfigurera distribuerad filsystemåtkomst" på sidan 109.

Skrivbordets sökvägar och deras miljövariabler

Skrivbordets sökvägar skapas vid inloggningen av skrivbordshjälpmedlet `dtsearchpath`. Hjälpmedlet `dtsearchpath` använder en kombination av miljövariabler och inbyggda adresser för att skapa sökvägarna.

Miljövariabler som `dtsearchpath` kallas för *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 9-2 Miljövariabler för skrivbordets sökvägar

Sökväg för:	Utdatamiljövariabel	Systemtäckande indatavariabel	Egen indatavariabel
Program	DTAPPSEARCHPATH	DTSPSYSAPPHOSTS	DTSPUSERAPPHOSTS
Databas ¹	DTDATABASESEARCHPATH	DTSPSYSDATABASEHOSTS	DTSPUSERDATABASEHOSTS
Ikoner	XMICONSEARCHPATH, XMICONMSEARCHPATH	DTSPSYSICON	DTSPUSERICON
Hjälpdata	DTHELPSEARCHPATH	DTSPSYSHELP	DTSPUSERHELP

1. Definitioner av funktioner, datatyper och Fronpanel

Komponenter använder utdatavariablernas värden. Programhanteraren använder t ex värdet för programmets sökväg (`DTAPPSEARCHPATH`) för att hitta programgrupper.

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.

▼ Visa aktuellt värde för en sökväg (utdatavariabel)

- ◆ **Använd kommandot `dtsearchpath` för att visa aktuella värden för sökvägarna:**

- Om du vill ta reda på värdet för den aktuella (inloggade) användaren skriver du kommandot:

```
dtsearchpath -v
```

- Om du vill ta reda på värdet för en annan användare skriver du kommandot:

```
dtsearchpath -u användare
```

Sökvägens värden omfattar dessa variabler:

%H – används i `DHELPSEARCHPATH`. Hjälppvolymens namn.

%B – används i `XMICONSEARCHPATH`. Basnamnet på en ikonfil.

%M – storleken på ikonfilen (`.l`, `.m`, `.s`, `.t`) används i `XMICONSEARCHPATH`.

%L – värdet på miljövariabeln `LANG`.

▼ Göra egna ändringar av sökvägen

1. **Öppna *Hemmapp*/`.dtprofile` för redigering.**

2. **Lägga till eller redigera en rad som definierar och exporterar den egna indatavariabeln.**

Följande rad lägger t ex till en adress i användarens egen programsökväg:

```
export DTSPUSERAPPHOSTS=/projekt1/editors
```

3. **Logga ut och in igen för att aktivera ändringen.**

▼ Göra 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 och exporterar den systemtäckande indatavariabeln.**
Följande rad lägger t ex till en adress i den systemtäckande hjälpsökvägen:

```
export DTSPYSHELLP=/applications/helpdata
```
4. **Informera alla användare i systemet att de måste logga ut och in igen för att ändringarna ska träda i kraft.**

Programsökväg

Programmets sökväg är den primära sökväg som används av skrivbordet för att hitta program i det lokala systemet och på programserverar i hela nätverket.

När adresser läggs till i programmets sökväg uppdateras andra sökvägar (databas, ikon och hjälp) automatiskt så att de visar adresserna för dessa data. Programmets sökväg erbjuder alltså relativt enkel administration för program och deras skrivbordskonfigurationsfiler. Se "Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp" på sidan 136.

Standardsökväg för program

Standardsökvägen för program innehåller egna, systemtäckande och inbyggda adresser. Standardspråk är C.

Egen adress - *Hemmapp* / `.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 sammanlagda sökvägen anges av utdatavariabeln DTAPPSEARCHPATH.

Syntax för indatavariabler för programsökväg

Syntax för variablerna DTSPSYSAPPHOSTS och DTSPUSERAPPHOSTS är normalt

`VARIABLE=adress [, adress . . .]`

där *adress* har syntaxen:

/sökväg – anger en mapp i det lokala (sessionsserver-) systemet. Använd den här syntaxen för att lägga till en lokal mapp.

värddatornamn: – anger den systemtäckande mappen `/etc/dt/appconfig/appmanager/språk` i systemet *värddatornamn*. Använd den här syntaxen för att lägga till en programserver.

värddatornamn:/sökväg – anger en mapp på fjärrsystemet *värddatornamn*.

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 136.

Hur programsökvägens värde sätts samman

Värdet på programsökvägen (DTAPPSEARCHPATH) skapas genom att sätta samman följande adresser, listade i prioritetsordning:

- Adresser som anges med variabeln DTSPUSERAPPHOSTS
- Egen standardadress: *Hemmapp*/`.dt/appmanager`
- Standardadressen: `/etc/dt/appconfig/appmanager/språk`
- Adresser som anges med variabeln DTSPSYSAPPHOSTS
- `/usr/dt/appconfig/appmanager/språk`

Syntax:

värddatornamn:

utökas till att ange mappen `/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 programserverna `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` ger det här beteendet:

```
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` enligt följande:

```
export DTSPSYSAPPHOSTS=servera:
```

så påverkas sökvägarna enligt beskrivningen i Tabell 9-3.

TABELL 9-3 Påverkade sökvägar

Sökvägar	Mapp som lagts till sökväg
Program	<code>servera:/etc/dt/appconfig/appmanager/språk</code>
Databas	<code>servera:/etc/dt/appconfig/types/språk</code>
Ikon	<code>servera:/etc/dt/appconfig/icons/språk</code>
Hjälp	<code>servera:/etc/dt/appconfig/help/språk</code>

Och om du anger DTSPSYSAPPHOSTS enligt följande:

```
export DTSPSYSAPPHOSTS=/projekt1/apps
```

så påverkas sökvägarna enligt beskrivning i Tabell 9-4.

TABELL 9-4 Påverkade sökvägar

Sökvägar	Mapp som lagts till sökväg
Program	<code>/projekt1/apps/appmanager/språk</code>
Databas	<code>/projekt1/apps/types/språk</code>
Ikon	<code>/projekt1/apps/icons/språk</code>
Hjälp	<code>/projekt1/apps/help/språk</code>

Databassökväg (funktion/datatyper)

Databassökvägen hjälper skrivbordet att söka i vissa adresser efter filer som innehåller:

- Funktions- och datatypsdefinitioner (*.dt -filer)
- Frontpanelsdefinitioner (*.fp files).

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. Standardspråket är C.

Egen adress – *Hemmapp*/.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 undermapp för databasen till i databassökvägen (se "Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp" på sidan 136).

Om t ex programservern *hosta*: läggs till i programsökvägen, läggs mappen *hosta:/etc/dt/appconfig/types/språk* automatiskt till i databassökvägen.

Databassökvägens miljövariabler

Databassökvägen sätts samman utifrån inbyggda adresser och följande indatavariabler:

DTSPSYSDATABASEHOSTS – indatavariabel för sökväg till systemtäckande databas

DTSPUSERDATABASEHOSTS, *variabel* – indatavariabel för sökväg till egen databas

Följande indatavariabler används för att ange adresser utanför programsökvägen.

Den sammansatta databassökvägen anges av utdatavariabeln
DTDATABASESEARCHPATH.

Syntax för databassökvägens indatavariabler

Syntax för variablerna DTSPSYSDATABASEHOSTS och DTSPUSERDATABASEHOSTS är normalt

VARIABLE=adress [, *adress* . . .]

adress har syntaxen:

/sökväg – anger en mapp i det lokala (sessionsserver-) systemet. Använd denna syntax för att lägga till en lokal mapp.

värddatornamn: – anger den systemtäckande mappen */etc/dt/appconfig/types/språk* i systemet *värddatornamn*.

värddatornamn:/sökväg – anger en mapp i fjärrsystemet *värddatornamn*.

Hur databassökvägen sätts samman

Värdet för databassökvägen (DTDATABASESEARCHPATH) skapas genom att följande adresser sätts samman, listade i prioritetsordning:

- Adresser som anges med variabeln DTSPUSERDATABASEHOSTS
- Adresser som fås från variabeln DTSPUSERAPPHOSTS
- Egen standardadress: *Hemmapp/.dt/types*
- Standardadressen: */etc/dt/appconfig/types/språk*
- Adresser som anges med variabeln DTSPSYSDATABASEHOSTS
- Adresser som fås från variabeln DTSPSYSAPPHOSTS
- */usr/dt/appconfig/types/språk*

Syntaxen:

värddatornamn:

utökas till att ange mappen */etc/dt/appconfig/types* i systemet *värddatornamn*.

Ikonsökväg

Ikonsökvägen hjälper skrivbordet att söka i angivna adresser efter filer som innehåller bitmapp- och bildpunktsmönsterfiler.

Standardikonsökväg

Standardikonsökvägen innehåller egna, systemtäckande och inbyggda adresser. Standard*språket* är C.

Egen adress – *Hemmapp/.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 ikonundermapp automatiskt till i ikonsökvägen (se "Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp" på sidan 136).

Om t ex programservern *hosta*: läggs till i programsökvägen, läggs mappen *hosta:/etc/dt/appconfig/icons/språk* automatiskt till i ikonsökvägen.

Ikonsökvägens miljövariabler

Ikonsökvägen sätts samman utifrån inbyggda adresser och följande indatavariabler:

DTSPSYSICON - indatavariabel för sökväg till systemtäckande ikon

DTSPUSERICON - indatavariabel för egen ikonsökväg

Följande indatavariabler anger adresser utanför programsökvägen.

Den sammansatta ikonsökvägen anges av två utdatavariabler:

XMICONSEARCHPATH - används av färgbildskärmar

XMICONBMSEARCHPATH - används av monokroma bildskärmar

Syntax för ikonsökvägens indatavariabler

Syntax för variablerna DTSPSYSICON och DTSPUSERICON är normalt

VARIABLE=adress [,adress . . .]

adress har syntaxen:

/sökväg anger en mapp i det lokala (sessionsserver-) systemet. Använd den här syntaxen för att lägga till en lokal mapp.

Om du vill ange en adress i ett annat system använder du nätverksfilnamnet, exempelvis, */nfs/servera/projects/icons*.

Hur ikonsökvägen sätts samman

Värdet för ikonsökvägen (XMICONSEARCHPATH och XMICONBMSEARCHPATH) skapas genom sammansättning av följande adresser, listade i prioritetsordning:

- Adresser som anges med variabeln DTSPUSERICON
- Adresser som fås från variabeln DTSPUSERAPPHOSTS
- Egen standardadress: *Hemmapp/.dt/icons*
- Standardadressen: */etc/dt/appconfig/icons/språk*
- Adresser som anges med variabeln DTSPSYSICON
- Adresser som fås från variabeln DTSPSYSAPPHOSTS
- */usr/dt/appconfig/icons/språk*

Sökvägarna för färgskärm och monokrom skärm skiljer sig bara åt genom prioriteten som tilldelas bildpunktsbeskrivningar och bitmappar. Variabeln XMICONSEARCHPATH listar bildpunktsbeskrivningar före bitmappar. XMICONBMSEARCHPATH listar bitmappar före bildpunktsbeskrivningar.

Hjälpsökväg

Hjälpsökvägen hjälper skrivbordet att söka i angivna adresser efter filer som innehåller hjälpinformation som ska registreras i systemet.

Standardhjälpsökväg

Standardhjälpsökvägen innehåller egna, systemtäckande och inbyggda adresser. Standard*språket* är C.

Egen adress – *Hemmapp/.dt/help*

Systemtäckande adress – */etc/dt/appconfig/help/språk*

Inbyggd adress – */usr/dt/appconfig/help/språk*

Hur programsökvägen påverkar hjälpsökvägen

När en adress läggs till i programsökvägen, läggs en lämplig hjälpundermapp automatiskt till i hjälpsökvägen (se "Hur programsökvägen påverkar sökvägarna för databas, ikon och hjälp" på sidan 136).

Om t ex programservern *hosta:* läggs till i programsökvägen, läggs mappen *hosta:/etc/dt/appconfig/help/språk* automatiskt till i hjälpsökvägen.

Hjälpsökvägens miljövariabler

Hjälpsökvägen sätts samman utifrån inbyggda adresser och följande indatavariabler:

DTSPSYSHELP – indatavariabel för sökväg till systemtäckande hjälp

DTSPUSERHELP – indatavariabel för sökväg till egen hjälp

Följande indatavariabler anger adresser utanför programsökvägen.

Den sammansatta databassökvägen anges av utdatavariabeln DTHELPPSEARCHPATH.

Syntax för hjälpsökvägens indatavariabler

Syntax för variablerna DTSPSYSHELP och DTSPUSERHELP är normalt

`VARIABLE=adress [, adress . . .]`

där *adress* har syntaxen:

/sökväg – anger en mapp i det lokala (sessionsserver-) systemet. Använd den här syntaxen för att lägga till en lokal mapp.

Om du vill ange en adress i ett annat system använder du dess nätverksfilnamn, exempelvis `/nfs/servera/projects/help`.

Hur ikonsökvägen sätts samman

Värdet för hjälpsökvägen (DTHELPPSEARCHPATH) skapas genom sammansättning av följande adresser, listade i prioritetsordning:

- Platser som anges med variabeln DTSPUSERHELP
- Platser som fås från variabeln DTSPUSERAPPHOSTS
- Egen standardadress: *Hemmapp*/.dt/help
- Standardadressen: `/etc/dt/appconfig/help/språk`
- Platser som anges med variabeln DTSPSYSHELP
- Platser som fås från variabeln DTSPSYSAPPHOSTS
- `/usr/dt/appconfig/help/språk`

Lokaliserade sökvägar

Utdatavariablerna innehåller poster för både lokaliserade adresser och standardadresser (C).

Standardprogramsökvägen är t ex:

```
Hemmapp/ .dt/appmanager  
/etc/dt/appconfig/appmanager/språk  
/etc/dt/appconfig/appmanager/C  
/usr/dt/appconfig/appmanager/språk  
/usr/dt/appconfig/appmanager/C
```

där *språk* ersätts med värdet i miljövariabeln LANG

För varje definitionsområde (systemtäckande och inbyggda) har den språkspecifika adressen prioritet över standardadressen.

Introduktion till funktioner och datatyper

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 ändra deras datafiler.

- "Introduktion till funktioner" på sidan 146
- "Introduktion till datatyper" på sidan 151

Det här kapitlet introducerar begreppen funktioner och datatyper. Det 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 utskrifter från skrivbordet.

Procedurerna och reglerna för att skapa funktioner och datatyper beskrivs i tre kapitel i handboken.

- Kapitel 11 beskriver hur du kan skapa funktioner och datatyper genom att använda skrivbordsverktyget Skapa funktion.

Du använder Skapa funktion för att skapa funktioner och datatyper till de flesta program, utan att du behöver lära dig syntaxreglerna för definitionerna.

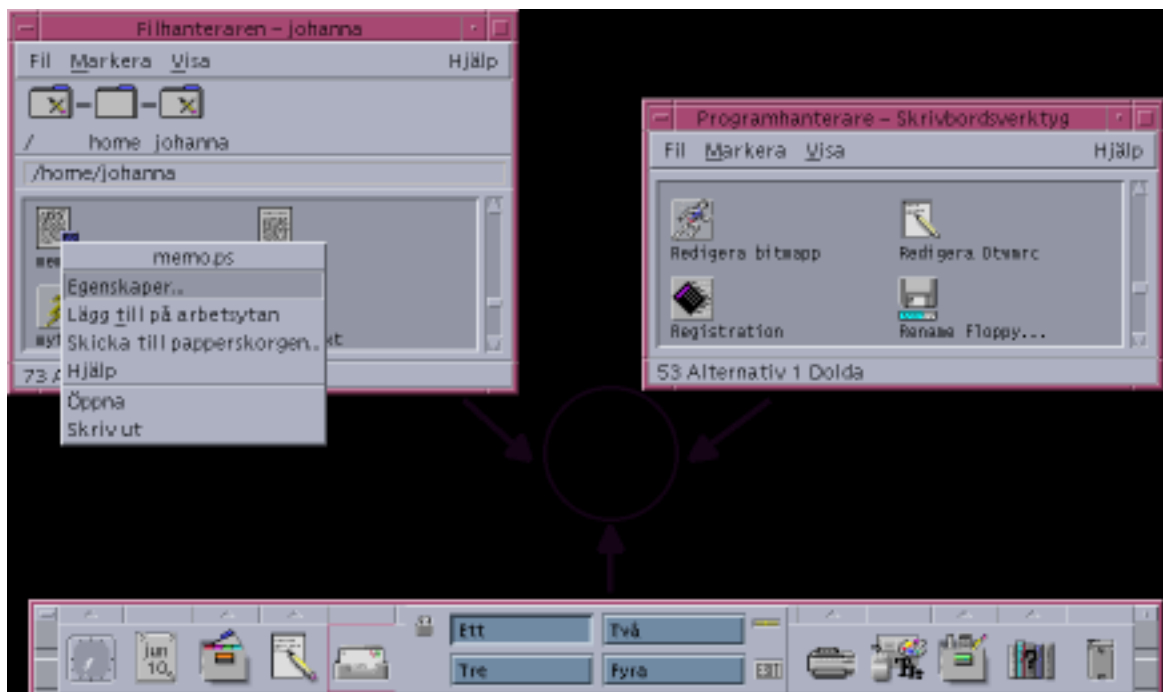
- Kapitel 12 och Kapitel 13 beskriver hur du skapar funktioner och datatyper manuellt genom att skapa och redigera konfigurationsfilerna.

När du vill använda andra funktioner som inte stöds av Skapa funktion måste du skapa dessa funktioner och datatyper manuellt.

Introduktion till funktioner

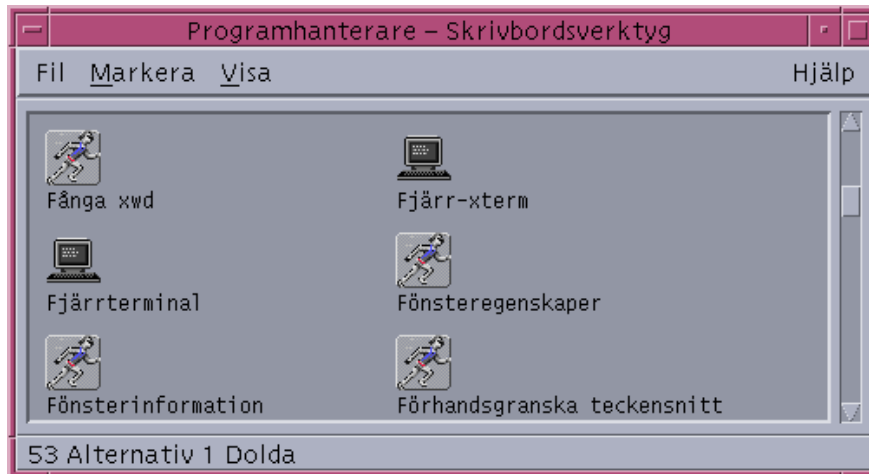
Funktioner är instruktioner som automatiserar skrivbordsuppgifter, t ex att köra program och öppna datafiler. Funktioner fungerar på ungefär samma sätt som makron och programmeringsfunktioner. Varje funktion har ett namn som används för att köra funktionen.

När du definerar en funktion kan den användas för att anpassa användargränssnittet så att olika aktiviteter blir lättare att utföra. Skrivbordet kan ansluta nya komponenter för användargränssnittet, t ex ikoner, kontroller för Frontpanelen och menyalternativ för funktioner.



Figur 10-1 Hur funktioner används

Programgruppen Skrivbordsverktyg i programhanteraren innehåller t ex ikoner som startar olika hjälpmedel.



Figur 10-2 Funktionsikoner i programgruppen Skrivbordsverktyg

När du dubbelklickar på en ikon kommer en funktion att köras. 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"Xwd-fil som ska visas:"%
  ...
}
```

Kommandot i funktionens EXEC_STRING körs när användaren dubbelklickar på ikonen.

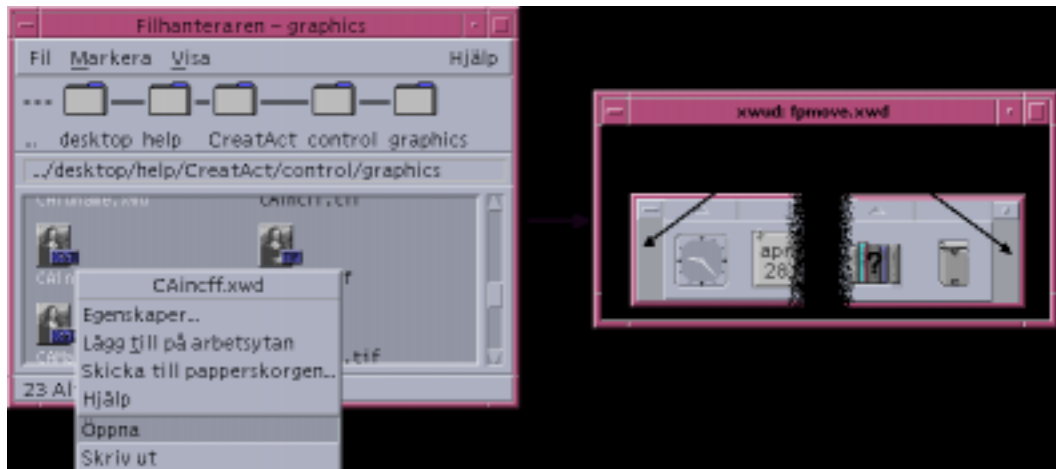
Frontpanelen använder också funktioner. Här är t ex en del av definitionen till ikonen Terminal på panelundermenyn Egna program. Ikonen definieras i konfigurationsfilen `/usr/dt/appconfig/types/språk/dtwm.fp:`

```
CONTROL Term
{
  ICON     Fpterm
  LABEL    Terminal
  PUSH_ACTION Dtterm
  ...
}
```

Fältet PUSH_ACTION anger vilken funktion som ska köras när användaren klickar på kontrollen – i detta fall en funktion kallad Dtterm.

Ett annat vanligt användningsområde för funktioner är menyer. Datafiler har normalt funktioner på Markera-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 10-3 Funktionen Öppna för filer med datatypen XWD

Funktionerna på Markera-menyn anges i datatypsdefinitionen för XWD-filen. Definitionen finns i konfigurationsfilen

```
/usr/dt/appconfig/types/språk/xclients.dt.
```

```
DATA_ATTRIBUTES XWD
{
  ACTIONS      Open,Print
  ICON         Dtxwd
  ...
}
```

Datatypen XWD och tillhörande Open- och Print-funktioner förklaras i .

Hur funktioner skapar programikoner

Titta på ikonerna Visa xwd i programgruppen Skrivbordsverktyg. Om du dubbelklickar på den körs X-klienten xwud. Ikonerna representerar dock inte direkt det körbara programmet xwud i /usr/bin/X11/xwud.

Ikonerna med namnet Visa xwd visas i programgruppen eftersom det finns en fil i den mappen med namnet Xwud (se Figur 10-4). Den filen motsvarar underliggande funktioner med samma namn - Xwud. I funktionsdefinitionen är funktionsnamnet namnet efter nyckelordet ACTION :

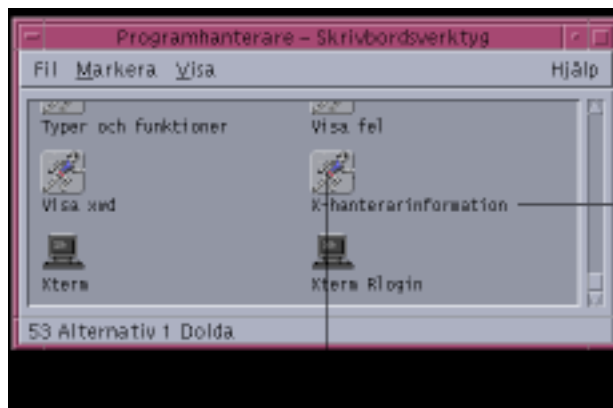
```
ACTION Xwud
{
  LABEL      Xwd Display
```

```

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 har skapats med funktionen \
             Xwd Capture (Xwd). Den använder \
             kommandot 'xwud'.
}

```

Filen kallas för en *funktionsfil* eftersom den är knuten till en funktion. En fil är en funktionsfil när den är körbar och har samma namn som funktionen. Ikonen i Programhanteraren (eller filhanteraren) kallas för en *funktionsikon* eller *programikon*, eftersom programmet startar då du dubbelklickar på den.



Figur 10-4 Programikonen (funktionsikonen) representerar en funktionsfil

När programhanteraren hittar en körbar 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 en funktionsfil är ointressant eftersom den innehåller kommentarer till skrivbordsfunktionen.

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 Visa xwd anger EXEC_STRING att funktionsikonen ska köra X-klienten xwud 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`, vilken 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 10-5 Ikonbild som anges med fältet `ICON` i funktionsdefinitionen

Funktionen `Xwud` kallas en *kommandofunktion* eftersom dess definition innehåller kommandot som ska köras (`EXEC_STRING`). Fältet `TYPE` i funktionsdefinitionen definierar funktionstypen.

Inledningsvis visas ikonen `Visa xwd` i programgruppen `Skrivbordsverktyg`. Du kan dock skapa ytterligare kopior av funktionsikonen i valfri mapp där du har skrivbehörighet. Så länge funktionsdefinitionen för `Xwud` är en del av databasen, blir alla körbara 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 arbetar med. Funktioner kan skrivas för att acceptera filargument.

`EXEC_STRING` för funktionen `Xwud` anger t ex 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 körs blir då:

```
/usr/bin/X11/xwd -noclick -in sökväg
```

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 Xwd blir ledtexten:

```
Xwd-fil som ska visas:
```

Filnamnet eller sökvägen som användaren anger används som filargument.

Ytterligare användningsområden för funktioner

Förutom för 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 238.

- Meny

Med syntaxen till definitionerna för Fönster- och Arbetsyta-menyn kan du ange vilken funktion som ska köras av ett menyalternativ. Mer information finns i "Arbetsytahanterarens meny" på sidan 256 och i direkthjälpen (man) för 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 utvecklingsmiljön.

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 datatypmekanism går det att skapa anpassat utseende och en anpassad metod för filer och mappar.

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  Denna fil innehåller en grafisk bild med XWD-format. \
               Dessa filer skapas vanligtvis genom att \
               funktionen XwdCapture tar en bild av ett fönster. \
               Dess datatyp heter XWD. XWD-filer \
               har namn som slutar med '.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 utseendet och egenskaper hos datatypen.

`DATA_CRITERIA` – bestämmer regler (namn och innehåll) för hur en fil ska kategoriseras så att den tillhör en datatyp.

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 med `.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 filtypstilläggen `.xwd` eller `.wd`. Skrivbordet använder filnamnet som *kriterium* vid bestämningen av filens datatyp.

Varje fil med datatypen XWD får:

- En unik ikonbild som hjälper användaren att känna igen datafilerna.
- Objekthjälp som informerar om datatypen.
- En egen Markera-menyn i filhanteraren som innehåller funktionerna Öppna och Skriv ut. Funktionen Open (Öppna) för XWD-filer kör funktionen Xwud.

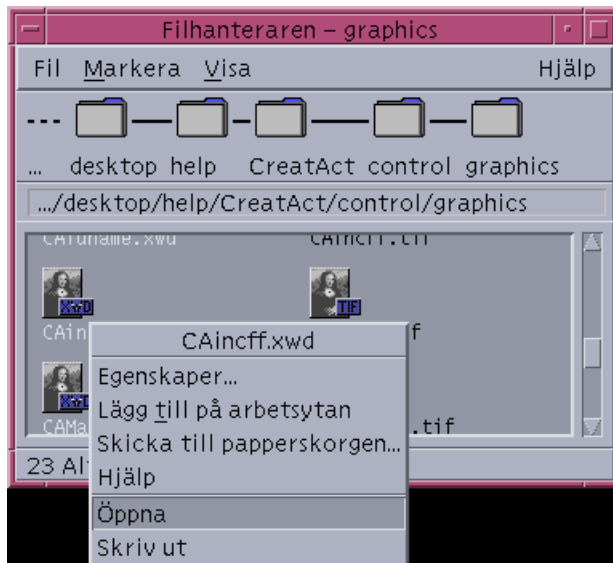
Köra funktioner från Markera-menyn

Markera-menyn i filhanteraren är bara aktiv när en fil eller en mapp är markerad. Vilket kommando som finns längst ned på Markera-menyn beror på datatypen. Om en XWD-fil är markerad finns alternativet Öppna och Skriv ut på Markera-menyn.

Fältet ACTIONS i datatypsdefinitionen anger vilka kommandon som läggs till längst ned på datatypens Markera-menyn.

```
DATA_ATTRIBUTES XWD { ACTIONS Open,Print ... }
```

Innehållet på Markera-menyn beror på datatypen. Det finns många datatyper som innehåller funktionen Open (Öppna). Det innebär att när du markerar en fil av en sådan datatyp i filhanteraren och visar Markera-menyn, finns kommandot Öppna på menyn.



Figur 10-6 Markera-menyn för en XWD-fil

Funktionen Open (Öppna) kör normalt programmet som datafilen hör till. Om du till exempel öppnar en XWD-fil körs funktionen Xwud, vilken i sin tur kör X-klienten xwud för att visa skärmbilden. Med andra ord är funktionen Open (Öppna) synonym med funktionen Xwud, för datatypen XWD. Om du öppnar en fil av typen TEXTFILE

öppnas Textredigeraren. När du öppnar en BM- (bitmapp) 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 direkt. Du kan t ex skapa en Öppna-funktion som tilldelas (kör) funktionen Xwud.

- Datatypsrestriktioner för en funktion.

Funktionsdefinitioner kan innehålla 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 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. `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 (bitmappfiler). Definitionen tilldelar funktionen Öppna 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 150).

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 om möjligt kommandot Skriv ut på filhanterarens Markera-menyn.
- Släpp en datafil på skrivbordets målområde för utskrift (skrivarikonerna på Frontpanelen eller en skrivarikon i utskriftshanteraren).

Dessutom går det att skriva ut från många program.

När du skriver ut från skrivbordet används funktionen 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      SkrivUt
  ARG_TYPE   XWD
  TYPE       MAP
  MAP_ACTION NoPrint
}
```

Den här Print (Skriv ut)-funktionen, som är specifik för XWD-filer, tilldelas 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 Print-funktionen för XWD med följande Print-funktion för PCL-filer:

```
ACTION Print
{
  LABEL      SkrivUt
  ARG_TYPE   PCL
  TYPE       MAP
  MAP_ACTION PrintRaw
}
```

Funktionen PrintRaw, som definieras i konfigurationsfilen

`/usr/dt/appconfig/types/språk/print.dt`, innehåller följande kommandorad för att skriva PCL-filerna:

```
ACTION PrintRaw
{
  TYPE          COMMAND
  WINDOW_TYPE   NO_STDIO
  EXEC_STRING   /usr/dt/bin/dt1p -w %(File)Arg_1%
}
```

Skapa funktioner och datatyper med Skapa funktion

Skapa funktion är ett verktyg 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. Mer information hittar du i:

- "Vad Skapa funktion gör" på sidan 157
- "Begränsningar i Skapa funktion" på sidan 158
- "Skapa en funktion och datatyp för ett program med Skapa funktion" på sidan 159

Referensinformation finns i direkthjälpen (man) för 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 *Hemmapp/.dt/types/funktionsnamn.dt*. Den här filen lagrar funktionen och de datatypsdefinitioner som skapas för programmet.
- Skapar en *funktionsfil* i användarens hemmapp. Funktionsfilen är en körbar fil med samma namn som funktionen.

Funktionsfilen representeras i filhanteraren av en *programikon* som startar programmet när du 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 för funktioner och datatyper. Detta gör att funktionerna och datatyperna träder i kraft omedelbart.

Begränsningar i 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 12
- Kapitel 13

Funktionsbegränsningar

Du kan inte använda Skapa funktion för att skapa funktioner för ett program om något av följande villkor är sanna:

- Kommandoraden kräver ett argument som inte är ett filargument (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 funktion för att skapa datatypen till ett program om något av följande villkor är sanna:

- Datatypen har 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 den mapp 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.

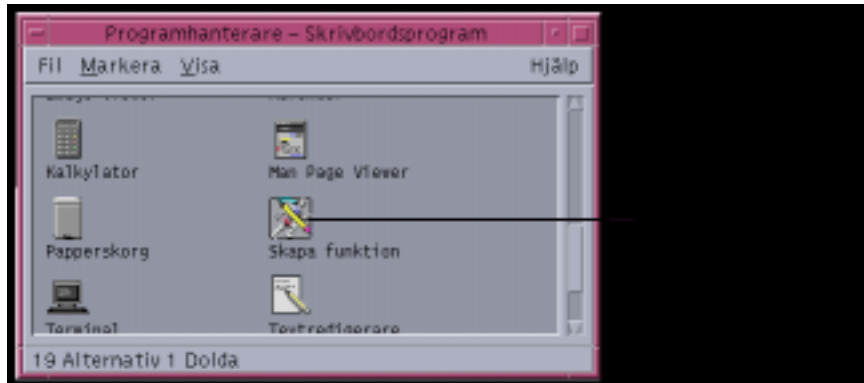
- På vilket sätt programmet identifierar sina datafiler.

Detta kan vara en namngivningskonvention (t ex filnamn som slutar med `.doc`). Det kan också bero på innehållet i filen. Även om programmet inte använder någon namngivningskonvention, kan du ange en sådan för funktionsikonen.

- Kommandoraden för att skriva ut filer (valfritt)

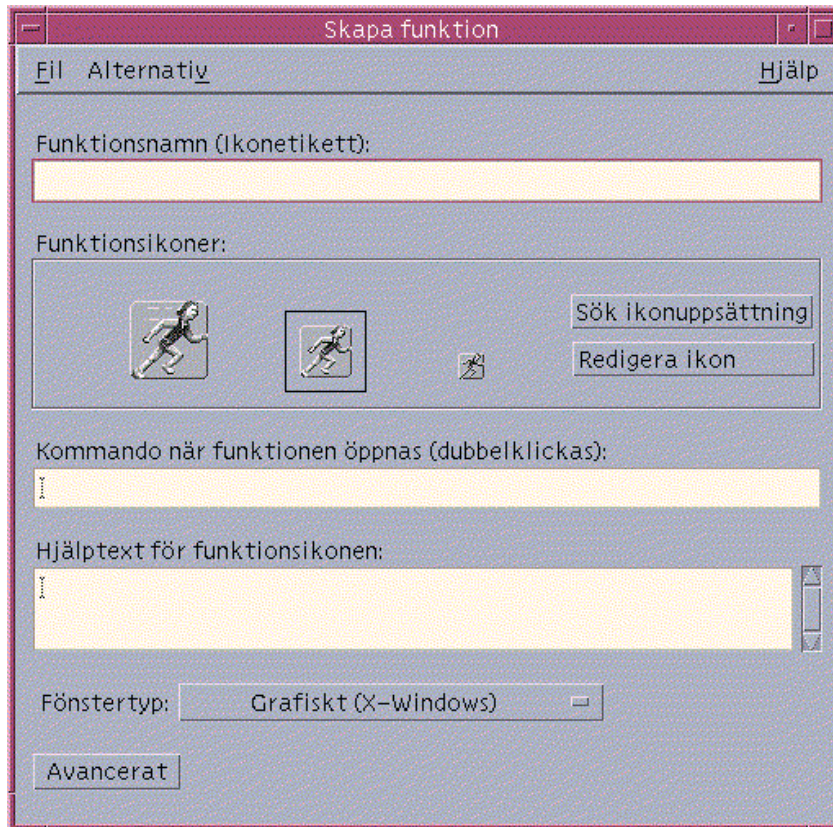
▼ Skapa en funktion för ett program

1. Dubbelklicka på Skapa funktion i programgruppen Skrivbordsprogram.



Figur 11-1 Ikonen för Skapa funktion i programhanteraren

Detta öppnar huvudfönstret i Skapa funktion.



Figur 11-2 Huvudfönstret i Skapa funktion

2. **Skriv namnet på funktionsikonen i textfältet Funktionsnamn.**
3. **Använd någon av funktionsikonerna för att ange ikon för programmet. Från början visas standardikonen.**
 - Om du vill välja en annan ikon som finns tillgänglig, kan du klicka på Sök ikonuppsättning för att visa dialogrutan Sök ikonuppsättning. Se "Använd dialogrutan Sök ikonuppsättning för att ange en ikon" på sidan 168.
 - Om du vill skapa nya ikoner väljer du Redigera ikon för att köra Ikonredigeraren.
4. **Skriv vilket kommando som ska starta programmet i textfältet Kommando när funktionen öppnas.**

Använd syntaxen $\$n$ som filargument, t ex:

```
emacs
  bitmap $1
  diff $1 $2
  lp -oraw $1
```

Om kommandoraden innehåller ett filargument (n), blir funktionsikonen 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 för 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 i menyfältet Fönstertyp.

Grafiskt (X-Windows) – programmet skapar sina egna fönster

Terminal (automatisk stängning) – programmet körs i ett kommandofönster som automatiskt kommer att stängas när programmet avslutas

Terminal (manuell stängning) – programmet körs i ett kommandofönster som kommer vara öppet ända tills användaren stänger det

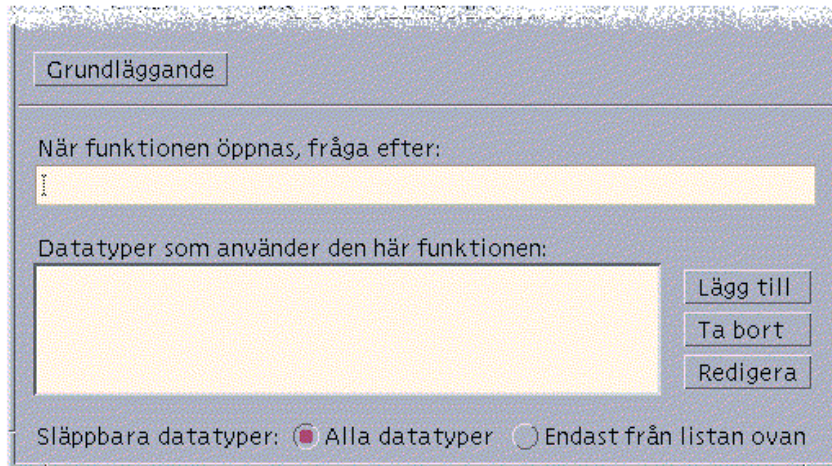
Ingen visning – programmet kommer inte visa någon utmatning på skärmen

7. Utför följande:

- Om programmet har datafiler och du vill skapa en eller fler datatyper för dem, läser du nästa avsnitt "Skapa en eller flera datatyper för programmet" på sidan 162.
- Om du inte vill skapa en datatyp kan du spara funktionen genom att välja Spara från Fil-menyn. Testa sedan den nya funktionen genom att dubbelklicka på ikonen i hemmappen.

▼ Skapa en eller flera datatyper för programmet

1. Definera en funktion för ett program genom att använda procedurerna från föregående avsnitt, "Skapa en funktion för ett program" på sidan 160.
2. Klicka på Avancerat i fönstret Skapa funktion för att visa fler alternativ.



Figur 11-3 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 "När funktionen öppnas, fråga efter".

Använd följande riktlinjer för textfältet:

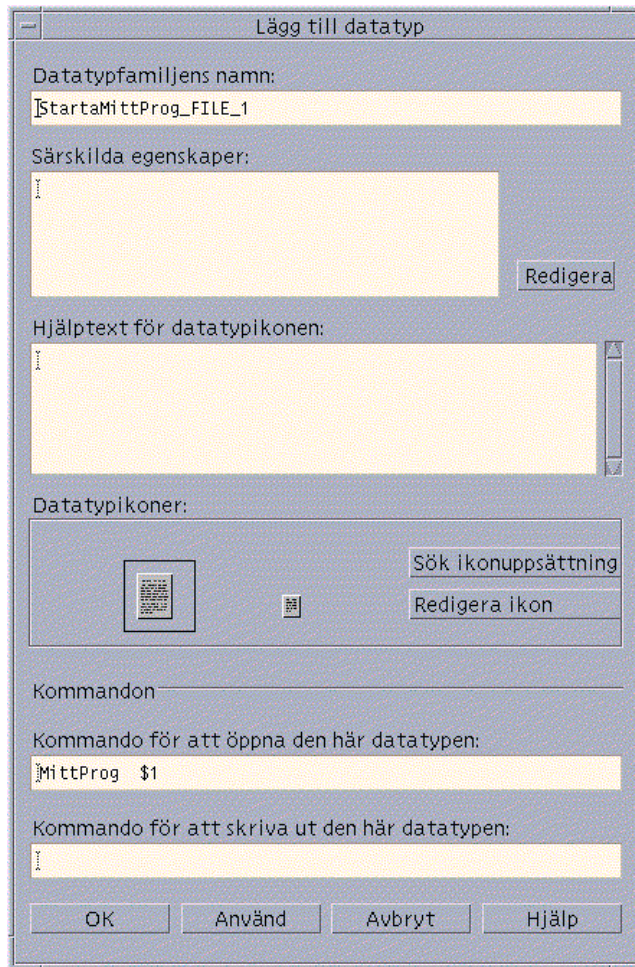
- Du måste använda det här textfältet om programmets kommandorad har ett *obligatoriskt* filargument.
- Lämna detta fält tomt om kommandoraden inte innehåller något filargument.
- Om filargumentet är valfritt i programmets kommandorad kan du välja om du vill använda det. Om du bifogar ledtext kommer funktionsikonen att fråga efter filnamnet när du dubbelklickar på ikonen. Om du inte bifogar ledtext kommer programmet att utföras med en nollvariabel som filargument.

4. Ange de filtyper som funktionen kommer att acceptera som argument:

- Om funktionen ska acceptera alla datatyper markerar du Alla datatyper.
- Om funktionen bara ska acceptera de datatyper som du har skapat för programmet markerar du Endast från listan ovan.

Inledningsvis kommer listan Datatyper som använder den här funktionen att vara tom. Efterhand som du skapar datatyper för program kommer de att läggas till i listan.

5. Klicka på **Lägg till** (bredvid listan Datatyper som använder den här funktionen) för att visa dialogrutan **Lägg till datatyp**.

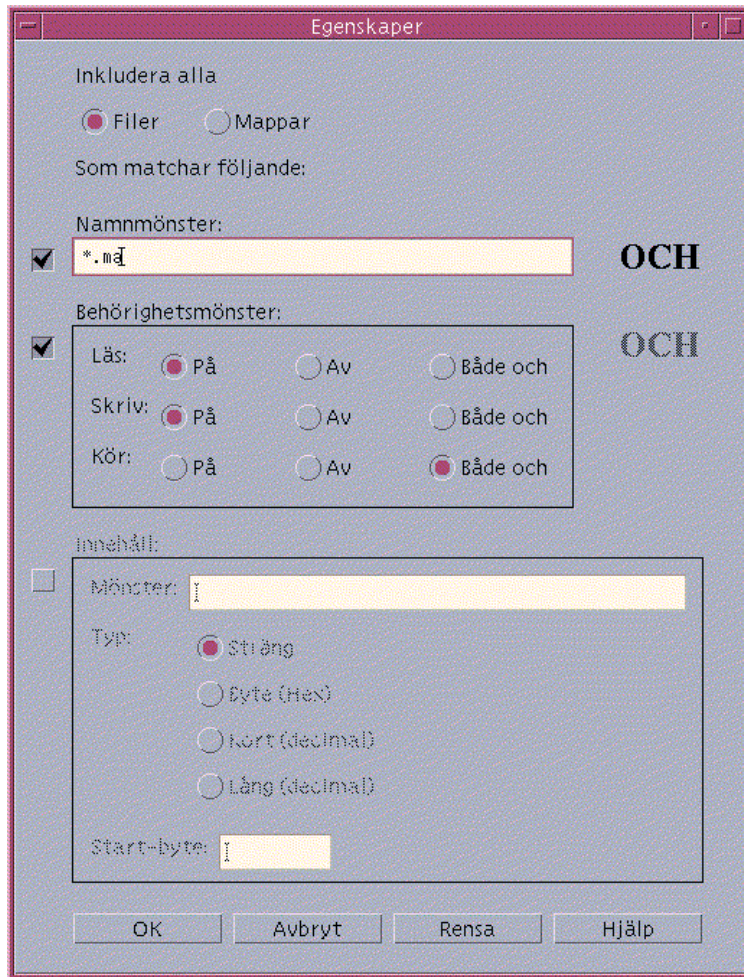


Figur 11-4 Dialogrutan Lägg till datatyp i Skapa funktion

6. Valfritt: Om du inte vill använda datatypens standardnamn skriver du ett nytt namn på datatypen i textfältet Datatypfamiljens namn.

Namnet får inte innehålla blanksteg. Datatypsnamnet är inte synligt för programmets användare, utan används i funktions- och datatypsdata-baser för att identifiera datatypsdefinitionen.

7. Klicka på Redigera (bredvid rutan Särskilda egenskaper) för att visa dialogrutan Egenskaper.



Figur 11-5 Dialogrutan Egenskaper i Skapa funktion

Datatypens egenskaper är de villkor som används för att skilja en datatyp från en annan. Du kan välja ett eller flera av följande villkor:

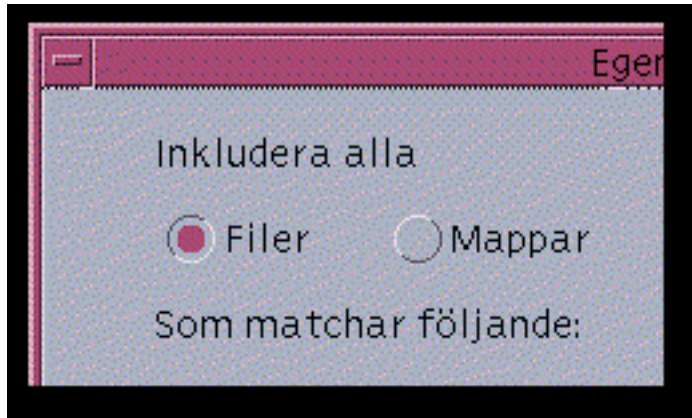
Filer eller mappar – datatypen används endast på filer eller på mappar

Namnmönster – datatypen baseras på filnamnet

Behörighetsmönster – rättighet att läsa, skriva och använda

Innehåll – innehåll i en specifik del av en fil

8. Ange om datatypen ska representera en fil eller en mapp.



Figur 11-6 Ange en fil- eller mappegenskap för datatypen.

9. Om datatypen är beroende av namnet markerar du kryssrutan **Nammönster** och fyller i textfältet.



Figur 11-7 Ange filnamnsegenskap för en datatyp

Du kan använda * och ? som jokertecken:

- * – matchar en hel serie tecken
- ? – matchar endast ett tecken

10. Om datatypen är beroende av behörigheter markerar du kryssrutan **Behörighetsmönster** och markerar behörighet för datatypen.

Behörighetsmönster:

OCH

Läs:	<input checked="" type="radio"/> På	<input type="radio"/> Av	<input type="radio"/> Både och
Skriv:	<input checked="" type="radio"/> På	<input type="radio"/> Av	<input type="radio"/> Både och
Kör:	<input type="radio"/> På	<input type="radio"/> Av	<input checked="" type="radio"/> Både och

Figur 11-8 Ange behörighetsegenskaper för datatypen

På – filen måste ha specificerad behörighet

Av – filen får inte ha specificerad behörighet

Både och – den specificerad behörighet spelar ingen roll

- 11. Om datatypen beror på dess innehåll kan du markera kryssrutan Innehåll och bifoga den information du vill ge – klicka på Mönster och Typ. Du kan också ange i vilken position som sökningen ska starta.**

Contents:

Pattern:

Type:

String

Byte (Hex)

Short (Decimal)

Long (Decimal)

Start Byte:

Figur 11-9 Ange innehållsegenskaper för en datatyp

Obs! Om du använder en innehållsbaserad datatyp kan det ta påverka systemets prestanda.

- 12. Klicka på OK för att stänga dialogrutan Egenskaper.**

Egenskaperna visas i fältet Särskilda egenskaper med dessa koder:

d – en mapp

- r – fil med läsrättigheter
- w – fil med skrivrättigheter
- x – fil med rättighet att använda
- ! – den logiska operatoren NOT
- & – den logiska operatoren AND

13. Skriv hjälptexten för datafilen i fältet Hjälptext för datatypikonen.

14. Använd någon av funktionsikonerna för att ange ikon för programmet. Inledningsvis visas standardikonen.

- Om du vill välja en annan tillgänglig ikon kan du klicka på Sök ikonuppsättning för att visa dialogrutan Sök ikonuppsättning. Se ”Använd dialogrutan Sök ikonuppsättning för att ange en ikon” på sidan 168.
- Om du vill skapa nya ikoner klickar du på Redigera ikon för att köra Ikonredigeraren.

15. Bekräfta kommandot i textfältet Kommando för att öppna den här datatypen. Detta är kommandot som kommer att användas när användaren dubbelklickar på datafilen.

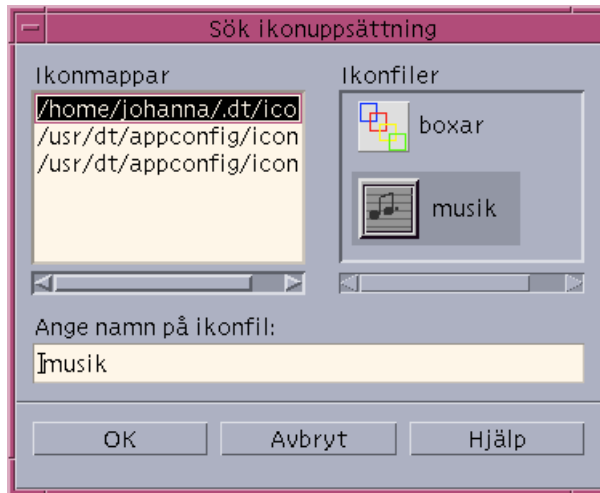
16. Valfritt: Om programmet har ett kommando för att skriva ut datafiler från kommandoraden kan du skriva kommandot i textfältet Kommando när du vill skriva ut den här datatypen. Använd följande syntax (\$n) för filargument, t ex:

17. Utför något av följande steg för att spara datatypsdefinitionen:

- Klicka på OK för att spara datatypen och stäng dialogrutan Lägg till datatyp.
- Klicka på Använd för att spara datatypen utan att stänga dialogrutan Lägg till datatyp. Detta gör att du omedelbart kan definiera ytterligare en ny datatyp för en funktion.

Använd dialogrutan Sök ikonuppsättning för att ange en ikon

Dialogrutan Sök ikonuppsättning visas när du klickar på Sök ikonuppsättning 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 11-10 Dialogrutan Sök ikonuppsättning

I dialogrutan Sök ikonuppsättning kan du ange en uppsättning ikonbildsfilerna som finns i:

- En mapp i ikonsökvägen. I listan Ikonmappar finns alla mappar i ikonsökvägen.
- Ett registreringspaket som ska integreras med skrivbordet med hjälp av dtappintegrate. Dessa ikoner finns ännu inte i någon mapp 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 mappar i ikonsökvägen.

Ange en uppsättning ikoner i ikonsökvägen

1. **I listan Ikonmappar i dialogrutan Sök ikonuppsättning dubbelklickar du på mappsökvägen som innehåller ikonerna.**
Listan Ikonfiler visar alla ikoner som finns i mappen.
2. **Klicka på ikonerna du vill använda i listan Ikonfiler.**
Basnamnet på ikonfilen placeras i textfältet Ange namn på ikonfil.
3. **Välj OK.**

Ange en ikon i ett registreringspaket

För systemadministratörer eller programmerare som skapar ett registreringspaket finns ikonfilerna inledningsvis i en mapp i registreringspaketet:

programrot/dt/appconfig/icons/*språk*

När du har registrerat ikonerna med `dtappintegrate`, kopieras ikonfilerna till `/etc/dt/appconfig/icons/språk`, i ikonsökvägen.

Använd proceduren nedan för att ange ikoner som är en del av registreringspaketet:

- 1. I textfältet Ange namn på ikonfil i dialogrutan Sök ikonuppsättning skriver du ett basnamn på ikonfilen.**
- 2. Välj OK.**
Skapa funktion visar en dialogruta som meddelar att ikonerna inte fanns i någon mapp längs ikonsökvägen.
- 3. Välj Ingen ändring i dialogrutan som visas.**

Skapa funktioner manuellt

Det finns två sätt att skapa funktioner:

- Använda skrivbordsprogrammet Skapa funktion
- Skapa en funktionsdefinition manuellt

När du skapar en funktion manuellt måste du redigera en databasfil. Det här kapitlet beskriver hur du skapar funktionsdefinitioner manuellt.

- "När funktioner måste skapas manuellt" på sidan 172
- "Skapa en funktion manuellt: Generella steg" på sidan 173
- "Skapa en exekveringssträng för COMMAND-funktioner" på sidan 181
- "Fönsterstöd och terminalemulatorer för COMMAND-funktioner" på sidan 188
- "Begränsa funktioner för vissa argument" på sidan 190
- "Skapa funktioner som kör program i fjärrsystem" på sidan 192
- "Använda variabler i funktions- och datatypsdefinitioner" på sidan 193
- "Starta funktioner från en kommandorad" på sidan 195
- "Skapa språkanpassade funktioner" på sidan 196
- "Skapa funktioner för ToolTalk-program" på sidan 197
- En introduktion till funktioner finns i Kapitel 10.
- Mer information om Skapa funktion finns i Kapitel 11.
- Referensinformation om funktionsdefinitioner finns i direkthjälpen (man) för `dtactionfile(4)`.

När funktioner måste skapas 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-funktioner. Alla TT_MSG-funktioner måste skapas manuellt.

Mer information finns i "Begränsningar i Skapa funktion" på sidan 158.

COMMAND-funktioner

En *kommandofunktion* utför ett kommando som startar ett program eller ett hjälpmedel, kör en kommandofil i skal eller utför ett operativsystemkommando. Definitionen för funktionen innehåller kommandot som ska köras (EXEC_STRING).

Verktyget Skapa funktion kan användas till att skapa de vanligaste typerna av kommandofunktioner. Det kan dock finnas situationer där du måste skapa funktionen manuellt. Du måste exempelvis skapa en COMMAND-funktion manuellt om funktionen anger:

- Flerfilsargument med olika ledtexter för varje argument.
- Starta funktioner – funktioners förmåga att starta andra funktioner.
- Argumentberoende funktion – förmågan att skapa en åtgärd som fungerar olika för olika antal filargument.
- Värddator för fjärrkörning – förmågan att köra ett program på ett annat system än det som innehåller funktionsdefinitionen.
- Ändra användare – förmågan att köra funktionen som en annan användare (exempelvis fråga efter superanvändarens lösenord och därefter köra som root).

MAP-funktioner

En *tilldelningsfunktion* är en funktion som tilldelas en annan funktion i stället för att den direkt anger ett kommando eller ToolTalk-meddelande.

Med tilldelning kan du ange alternativa namn på funktioner. En inbyggd kommandofunktion med namnet IconEditor startar Ikonredigeraren. Databasen innehåller också en Öppna-funktion (i definitionen begränsad till bitmapps- och bildpunktsfiler (av fältet ARG_TYPE)) som tilldelats funktionen IconEditor. På så sätt

kan användaren starta Ikonredigeraren genom att markera en bitmapps- eller bildpunktsfil i Filhanteraren och därefter välja Öppna på Markera-menyn.

Skapa funktion erbjuder begränsad tilldelning för Öppna- och Skriv ut-funktioner. Alla andra tilldelningsfunktioner måste skapas manuellt.

TT_MSG-funktioner (ToolTalk-meddelanden)

TT_MSG-funktioner skickar ToolTalk-meddelanden. Alla TT_MSG-funktioner måste skapas manuellt.

Skapa en funktion manuellt: Generella steg

Det här avsnittet beskriver hur du skapar en konfigurationsfil för en funktionsdefinition.

Konfigurationsfiler för funktioner

Konfigurationsfiler som innehåller funktionsdefinitioner måste uppfylla följande krav:

- Filerna måste använda namnkonventionen *namn.dt*
- Filerna måste finnas i databasens (funktioner och datatyper) sökväg. Standardsökvägen är:

Egna funktioner – *hemmapp/.dt/types*

Systemfunktioner – */etc/dt/appconfig/types/språk*

Inbyggda funktioner – */usr/dt/appconfig/types/språk*. Du bör inte använda den här mappen.

Mer information om hur du ändrar sökväg för funktioner och datatyper finns i "Ange värde för en sökväg" på sidan 133.

▼ Skapa en funktion manuellt

1. **Öppna en befintlig databasfil eller skapa en ny.**
Se föregående avsnitt, "Konfigurationsfiler för funktioner" på sidan 173.
2. **Skapa en funktionsdefinition med syntaxen:**

```
ACTION funktionsnamn
{
    TYPE funktionstyp
    funktionsfält
    ...
}
```

där:

funktionsnamn är namnet som används för att köra funktionen.

funktionstyp är COMMAND (standard), MAP, eller TT_MSG.

funktionsfält är ett av de nödvändiga 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. Du får mer information i direkthjälpen (man) för 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: *Hemmapp/.dt/icons*
- Systemtäckande ikoner: */etc/dt/appconfig/icons/språk*. Standardvärdet för *språk* är C.

Mer information finns i "Ange funktionens ikonbild" på sidan 178.

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 körbar fil med samma namn som ett *funktionsnamn*. Du kan placera den här filen i en valfri mapp, 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 ProgServerA. Kommandot som startar faxprogrammet är:

```
/usr/fax/bin/faxcompose [filnamn]
```

1. Skapa filen *Hemmapp/.dt/types/Fax.dt*.

2. Placera följande funktionsdefinition i filen:


```

ACTION Faxassistent
{
    TYPE          COMMAND
    ICON          fax
    WINDOW_TYPE   NO_STDIO
    EXEC_STRING   /usr/fax/bin/faxcompose -c %Arg_1%
    EXEC_HOST     ProgServerA
    DESCRIPTION    Kör Faxassistenten
}

```

Fälten WINDOW_TYPE och EXEC_STRING beskriver funktionens beteende.

WINDOW_TYPE – nyckelordet NO_STDIO anger att funktionen inte behöver köras i ett terminalemulatorfönster.

Se "Ange fönsterstöd för funktionen" på sidan 188.

EXEC_STRING – syntaxen %Arg_1% accepterar en släppt fil. Om någon dubbelklickar på funktionsikonen, öppnar funktionen ett tomt faxassistentfönster.

Se "Skapa en exekveringssträng för COMMAND-funktioner" på sidan 181.

3. Spara filen

4. Skapa följande ikonbildsfil i *Hemmapp*/.dt/icons med hjälp av Ikonredigeraren:

- fax.m.pm, 32 gånger 32 bildpunkter
- fax.t.pm, 16 gånger 16 bildpunkter

5. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

6. Skapa en körbar fil med namnet Faxassistent i en mapp som du har skrivbehörighet till (t ex hemmappen).

Exempel på hur du skapar en MAP-funktion

Anta att de flesta filer du faxar skapas med Textredigeraren och är av datatypen TEXTFILE (filer med namnet *.txt).

Stegen nedan lägger till menyalternativet "Fax" på datatypens Markera-menyn.

1. Öppna filen *Hemmapp*/.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 Faxassistent
}
```

3. Spara filen

4. Kopiera dataattributsdefinition för TEXTFILE från /usr/dt/appconfig/types/*språk*/dtpad.dt till den nya filen **Hemmapp/.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.

▼ Starta 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* också kör du kommandot:

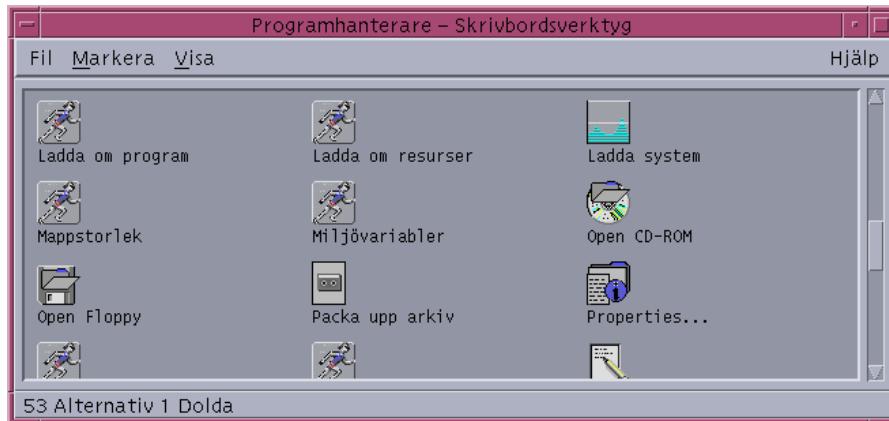
```
dtaction ReloadActions
```

ReloadActions är namnet på den funktion vars ikon heter "Ladda om funktioner". Funktionsdatabasen läses också om när användaren:

- Loggar in
- Startar om arbetsyttehanteraren
- Sparar en funktion i fönstret Skapa funktion genom att välja Spara på Fil-menyn.

Skapa funktionsfil (ikon) för funktion

En *funktionsfil* är en fil som skapas för att visuellt representera funktionen i Filhanteraren och Programhanteraren.



Figur 12-1 Funktionsfiler (funktionsikoner) i Programhanteraren

Eftersom en funktionsfils ikon representerar en funktion, kallas den ibland *funktionsikon*. Om den underliggande funktionen startar ett program, kallas funktionsfilsikonen för *programikon*.

Om du dubbelklickar på funktionsikonen körs funktionen. Funktionsikonen kan också vara ett släppområde.

Skapa en funktionsfil (funktionsikon)

- ◆ Skapa en körbar fil med samma namn som funktionsnamnet. Innehållet i filen har ingen betydelse.

Om funktionsdefinitionen är:

```

ACTION Favoritprogram
{
  EXEC_STRING Fav-fil %Arg_1%
  DESCRIPTION Kör Favoritprogram
  ICON Favp
}

```

blir funktionsfilen en körbar fil med namnet `Favoritprogram`. I Filhanteraren och Programhanteraren använder filen `Favoritprogram` ikonbilden `Favp.storlek.typ`. Om du dubbelklickar på ikonen för `Favoritprogram` körs funktionens exekveringssträng, och ikonens objekthjälp får innehållet i fältet `DESCRIPTION` ("Kör `Favoritprogram`").

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 t ex innehåller:

```
ACTION Favoritprogram
{
  LABEL      Mitt favoritprogram
  ...
}
```

kommer funktionsikonen att heta "Mitt favoritprogram".

Ange funktionens ikonbild

Använd fältet `ICON` för att ange vilken ikonbild som ska användas i Filhanteraren och Programhanteraren för den funktionsikon som skapas för funktionen.

Om du inte anger någon ikon används bildfilerna för standardfunktionsikoner `/usr/dt/appconfig/icons/sprak/Dtactn.*`.



Figur 12-2 Standardbild för funktionsikon

Standardfunktionsikonen ändras med resursen:

```
*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 suffixen för storlek (m och t) och bildtyp (bm och pm). Om filer t ex får namnet `Spelikon.m.pm` och `Spelikon.t.pm`, använder du `Spelikon`.

Om du använder basfilnamnet måste ikonfilen placeras i en mapp i ikonsökvägen:

- Egna ikoner: *Hemmapp/.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 ikonerna inte finns i ikonsökvägen. Om ikonfilen *Spelikon.m.pm* placeras i mappen */dok/projekt*, som inte finns i ikonsökvägen, blir värdet i fältet *ICON /dok/projekt/Spelikon.m.pm*.

Tabell 12-1 visar ikonstorlekar som du bör skapa samt motsvarande filnamn.

TABELL 12-1 Ikonnamn och storlek på funktionsikoner

Storlek i bildpunkter	Bitmappsnamn	Bildpunktsmönsternamn
48 gånger 48	<i>namn.l.bm</i>	<i>namn.l.pm</i>
32 gånger 32	<i>namn.m.bm</i>	<i>namn.m.pm</i>
16 gånger 16	<i>namn.t.bm</i>	<i>namn.t.pm</i>

▼ Ändra 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.

Standardsökvägen för funktionsdefinitioner är:

- Inbyggda funktioner: */usr/dt/appconfig/types/språk*
- Systemtäckande funktioner: */etc/dt/appconfig/types/språk*
- Egna funktioner: *Hemmapp/.dt/types*

Det kan finnas fler sökvägar i systemet. Om du vill visa en lista med de sökvägar som systemet använder för funktioner, skriver du kommandot:

```
dtsearchpath -v
```

Systemet använder mapparna som listas under `DTDATABASESEARCHPATH`.

2. Om det behövs kopierar du texten till funktionsdefinitionen till en ny eller befintlig fil i någon av dessa mappar:

- Systemtäckande funktioner: `/etc/dt/appconfig/types/språk`
- Egna funktioner: `Hemmapp/.dt/types`

Du måste kopiera inbyggda funktioner eftersom du inte bör redigera filer i mappen `/usr/dt/appconfig/types/språk`.

3. Redigera funktionsdefinitionen.

4. När du är färdig med redigeringen sparar du filen.

5. 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 funktion som ska användas.

- Om inga andra prioritetsregler gäller, baseras prioriteten på definitionens sökväg. Följande lista är rangordnad från högsta till lägsta prioritet:

- Egna funktioner (`Hemmapp/.dt/types`)
- Systemtäckande funktioner (`/etc/dt/appconfig/types/språk`)
- Systemtäckande fjärfunktioner (`värddatornamn:/etc/dt/appconfig/types/språk`). De fjärranslutna värddatorer som genomsöks är de som listas i programmets sökväg.
- Inbyggda funktioner (`/usr/dt/appconfig/types/språk`)

- Inom en given mapp 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 fler än en begränsning gäller är prioritetsordningen (från hög till låg):

- `ARG_CLASS`
- `ARG_TYPE`
- `ARG_MODE`
- `ARG_COUNT`

När det finns fler än en begränsad `ARG_COUNT` är prioritetsordningen (från hög till låg):

- 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    *
  ...
}
```

När du dubbelklicka på funktionen EditGraphics startar EditGraphics-2 eftersom inga argument har angivits och ARG_COUNT 0 har prioritet. När filargument av XWD-typ tillhandahålls, används EditGraphics-1 eftersom det angav XWD som ARG_TYPE. EditGraphics-3 används för alla andra filargument.

Skapa en exekveringssträng för COMMAND-funktioner

Minimikraven för en COMMAND-funktion är två fält – ACTION och EXEC_STRING.

```
ACTION funktionsnamn
{
  EXEC_STRING exekveringssträng
}
```

Exekveringssträngen är den viktigaste delen av en `COMMAND`-funktionsdefinition. Den använder syntax som liknar den på kommandoraden i ett terminalfönster, men innehåller ytterligare syntax för hantering av fil- och strängargument.

Generella egenskaper för exekveringssträngar

Exekveringssträngar kan innehålla:

- Fil och icke-filargument
- Skalsyntax
- Absoluta sökvägar eller namn på körbara 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 kommandon, ha argument. Det finns två typer av data som en `COMMAND`-funktion kan använda:

- Filer
- Strängdata

Använda skal i exekveringssträngar

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.

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 ***"'
```

Skapa en funktion som kör ett fjärrprogram

Om ditt program finns i en mapp som är listad i variabeln `PATH`, kan du ange den körbara filens enkla namn. Om programmet finns någon annanstans måste du använda den absoluta sökvägen till den körbara filen.

Skapa funktioner som inte använder argument

Använd samma syntax i fältet `EXEC_STRING` som du skulle göra på kommandoraden.

Exempel

- Den här körbara strängen är en del av funktionen som startar X-klienten `xcutssel`.

```
EXEC_STRING xcutssel
```

- Den här körbara 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 följande syntax för filargumentet:

```
%Arg_n%
```

eller

```
%(Fil)Arg_n%
```

(File) är valfritt eftersom argument för `Arg_n` antas (som standard) vara filer. (I "Tolka filargument som en sträng" på sidan 185 beskrivs hur du använder syntaxen `%(String)Arg_n%`.)

Med den här syntaxen kan användarna släppa ett datafilobjekt på funktionsikonen för att starta funktionen med filargumentet. Den ersätter det *n*:te argumentet på kommandoraden. Filen kan vara en lokal fil eller en fjärrfil.

Exempel

- Exekveringssträngen kör `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 mappargument. När mappen släpps på funktionsikonen, visas en lista över alla filer i mappen som har läs- och skrivbehörighet.

```
ACTION List_Writable_Files
{
    ARG_TYPE      FOLDER
    EXEC_STRING /bin/sh -c 's -l %Arg_1% | grep rw-'
    ...
}
```

```
}
```

Skapa en funktion som frågar efter ett filargument

Använd följande syntax 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 följande syntax 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 startas genom att någon dubbelklickar på ikonen visas en dialogruta som frågar efter filnamnet.

```
EXEC_STRING lp -oraw %Arg_1"Fil att skriva ut:" %
```

Skapa en funktion som frågar efter icke-filargument

Använd följande syntax 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 inte syntaxen för att fråga efter ett 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 följande syntax 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 som ska skrivas 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 skalprogramkoppling.

```
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"Hjälpfil (man):"% | 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 som ska skrivas ut:"% | \  
uncompress | lp -oraw'
```

- Denna exekveringssträng startar ett skalskript:

```
EXEC_STRING /usr/local/bin/StartGnuClient
```

Skapa COMMAND-funktioner för flera filargument

Funktioner kan använda något av följande tre sätt att hantera flera filargument:

- Funktionen utförs en gång för varje argument: När en EXEC_STRING innehåller ett enskilt 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
```

som kräver två unika filer i en viss ordning.

- Funktionen kan utföra samma kommando för varje filargument, exempelvis:

```
pr fil [fil ...]
```

som skriver ut en eller flera filer.

Skapa en funktion för icke utbytbara argument

Använd någon av följande syntaxer:

- Om du vill att funktionen ska fråga efter filnamnen använder du den här syntaxen för varje filargument:

```
%(File)"ledtext" %
```

Använd en särskild *ledtext* för varje argument.

Följande exekveringssträng frågar efter två filer.

```
EXEC_STRING      xsetroot -cursor %(File)"Markörbitmapp:"% \
                  %(File)"Maskbitmapp:"%
```

- Om följande syntax används för filargument, accepteras släppta filer:

```
%Arg_n%
```

Använd olika värden på *n* för varje argument. Exempel:

```
EXEC_STRING      diff %Arg_1% %Arg_2%
```

Skapa COMMAND-funktioner för utbytbara filargument

Använd någon av följande syntaxer:

- Om du vill skapa en funktion som accepterar släppta filer och som skapar ett kommando av formen *kommando fil 1 fil 2 ...*, ska följande syntax för filargument användas:

```
%Args%
```

- Om du vill skapa en funktion som accepterar flera släppta filer, eller visar en ledtext för en enstaka fil när du dubbelklickar på den, använder du följande syntax för filargumenten:

```
%Arg_1"ledtext"% %Args%
```

Funktionen startar kommandot i formatet: *kommando fil 1 fil 2 ...*

Exempel

- Den här exekveringssträngen skapar en funktion som kör:

```
pr 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 du dubbelklickar (inga filargument).

```
EXEC_STRING pr %Arg_1"Fil(er) som ska skrivas ut:"% %Args%
```

Skapa funktioner för flera släppta filer

För att acceptera flera argument för släppta filer och köra en kommandorad i formatet:

```
kommando fil 1 fil 2 ...
```

använder du syntaxen:

```
%Args%
```

Exempel

- Den här exekveringssträngen kör skriptet Checkout för flera filer:

```
EXEC_STRING /usr/local/bin/Checkout \ %Arg_1"Vilken fil ska hämtas ut?"% %Args%
```

- Den här exekveringssträngen kör `lp -oraw` med flera filer.

```
EXEC_STRING lp -oraw %Arg_1"Fil som ska skrivas ut:"% %Args%
```

Fönsterstöd och terminalemulatorer för COMMAND-funktioner

Det finns flera sätt på vilka funktionen `COMMAND` stödjer fönster på skrivbordet

- Om programmet har ett eget fönster kan funktionen skrivas så att den inte ger ytterligare fönsterstöd. Det här alternativet används också när en funktion kör ett kommando som inte kräver någon direkt användarindata och inte har några utdata.
- Om programmet måste köras i ett terminalfönster kan funktionen skrivas så att den startar ett terminalfönster och sedan kör programmet. Det finns flera terminalalternativ.

Ange fönsterstöd för funktionen

Använd fältet `WINDOW_TYPE` för att ange den typ av fönsterstöd som krävs av funktionen enligt Tabell 12-2.

TABELL 12-2 Fältet `WINDOW_TYPE` och tillgängligt fönsterstöd

<code>WINDOW_TYPE</code>	Tillgängligt fönsterstöd
<code>NO_STDIO</code>	Inget. Använd <code>NO_STDIO</code> om programmet har ett eget fönster eller om kommandot inte har synliga utdata.
<code>PERM_TERMINAL</code>	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 <code>ls mapp</code>).
<code>TERMINAL</code>	Temporärt terminalemulatorfönster. Funktionen öppnar ett terminalfönster som stängs så fort kommandot är färdigt. Används tillsammans med fullskärmskommandon (t ex <code>vi</code>).

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 ÖppnaTermPåValfrittSystem
{
  WINDOW_TYPE      PERM_TERMINAL
  EXEC_HOST        %(String)"Fjärrterminal på:"%
  TERM_OPTS        -title %(String)"fönstertitel:"%
  EXEC_STRING      $SHELL
}
```

Ange en annan standardterminalemulator

Standardterminalemulator för 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: sysibml:/usr/bin/xterm,syshp2:/usr/bin/yterm
```

Begränsa funktioner för vissa argument

När du begränsar en funktion till en viss typ av argument förfinas funktionen. Du bör t ex begränsa en funktion som startar en läsare för PostScript-filer till endast PostScript-filargument. Med en sådan begränsning returnerar funktionen ett felmeddelande om en icke-PostScript-fil anges.

Du kan begränsa funktioner baserat på:

- Filargumentets datatyp.
- Antalet filargument – exempelvis inga argument jämfört med ett eller flera argument. Detta gör att funktionsikonen fungerar på olika sätt när någon släpper filer eller 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 ange en lista med datatyper. Avgränsa posterna med kommatecken.

Följande funktionsdefinition antar att en gif-datatyp har skapats.

```
ACTION Open_Gif
{
    TYPE      COMMAND
    LABEL     "Visa gif"
    WINDOW_TYPE NO_STDIO
    ARG_TYPE  Gif
    ICON     xgif
    DESCRIPTION  Visar gif-filer
    EXEC_STRING  xgif
}
```

Begränsa funktioner med antalet argument

Använd fältet `ARG_COUNT` för att ange hur många argument som funktionen ska acceptera. Giltiga värden är:

* (Standard) – valfritt antal argument. Andra värden har företräde före *.

n – valfritt positivt 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. Se även nästa avsnitt "Ange metod för dubbelklickning och släppfunktionen".

▼ Ange metod för dubbelklickning och släppfunktionen

Använd följande procedur 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 dubbelklickningsfunktionalitet.

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 släppfunktionen.

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:

- Starta redigeraren utan filargument:

```
vedit
```

- Starta redigeraren med ett filargument som öppnas som skrivskyddat dokument:

```
vedit -R filnamn
```

Följande två funktioner skapar släpp- och dubbelklickningsfunktionalitet 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.

```
# Dubbelklickningsfunktionalitet
ACTION Vedit
{
  TYPE          COMMAND
  ARG_COUNT     0
  WINDOW_TYPE   PERM_TERMINAL
  EXEC_STRING   vedit
}

# Släppfunktionalitet
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:

* (Standard) – valfritt läge

!w – icke skrivbar

w – skrivbar

Skapa funktioner som kör program i fjärrsystem

När funktioner och fjärrexeckvering behandlas finns det två termer som ofta används:

databasvärd – systemet som innehåller funktionsdefinitionen

körvärd – systemet där den körbara filen körs

I de flesta situationer finns funktioner och deras program i samma system. Eftersom databasvärden är standardkörvärd för en funktion, krävs ingen särskild syntax.

När värd datorn för execkvering däremot är en annan än databasvärden, måste funktionsdefinitionen ange var execkveringssträngen 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 116.

Skapa en funktion som kör ett fjärrprogram

Använd fältet `EXEC_HOST` i funktionsdefinitionen för att ange platsen för programmet.

Giltiga värden för `EXEC_HOST` är:

`%DatabaseHost%` – värd datorn där funktionen har definierats.

`%LocalHost%` – värd datorn där funktionen startas (*sessionsservern*).

`%DisplayHost%` – värd datorn som kör X-hanteraren (ej tillåtet för X-terminaler).

`%SessionHost%` – värd datorn där den kontrollerande 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 startas.

Standardvärdet är `%DatabaseHost%`, `%LocalHost%`. Det innebär att när fältet `EXEC_HOST` utelämnas försöker dock funktionen först köra kommandot på den värddator 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 har samma namn som en miljövariabel har strängvariabeln företräde.

Definiera en strängvariabel

◆ Använd syntaxen:

```
set variabelnamn=värde
```

Variabelnamn kan innehålla alla alfanumeriska tecken samt understreck (_). Varje variabeldefinition måste stå på en separat rad.

Exempel:

```
set Remote_Application_Server=sysapp
set Remote_File_Server=sysdata
```

Referera till en strängvariabel

◆ Använd syntaxen:

```
${[variabel]}
```

Exempel:

```
EXEC-HOST ${Remote_Application_Server}
CWD /net/${Remote_File_Server}/doc/project
```

Använda miljövariabler i funktioner och datatyper

◆ Referera till en miljövariabel med syntaxen:

```
${[variabel]}.
```

Variabeln ersätts med dess värde när databasen laddas. Om en strängvariabel och en miljövariabel har samma namn används strängvariabeln.

Den här exekveringssträngen skriver exempelvis ut en fil med loginnamnet på försättsbladet.

```
EXEC-STRING lp -t$LOGNAME %(File)Arg_1%
```

Starta funktioner från en kommandorad

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 för `dtaction`

```
dtaction [-user användarnamn] [-execHost värddatornamn] \  
    funktionsnamn [argument [argument]...]
```

`-user användarnamn` – ger dig möjlighet att köra funktionen som en annan användare. Om `dtaction` anropas av en annan användare än `användarnamn`, ombeds användaren ange lösenord.

`-execHost värddatornamn` – för `COMMAND`-funktionen enbart. Anger värddatorn där kommandot kommer att köras.

`argument` – argument till funktionen. Vanligen filargument.

Klienten `dtaction` har ytterligare kommandoradsalternativ. Mer information finns i direkthjälpen (`man`) för `dtaction(1)`.

Skapa en funktion som kör en annan funktion

Använd `dtaction` i funktionens `EXEC_STRING`.

Följande funktion använder `tex` det inbyggda funktionsnamnet `Spell` (funktionen "Kontrollera stavning" i Programhanteraren). Den nya funktionen kör `Textredigeraren` 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"File:"%; dtpad %Arg_1%'  
}
```

Skapa funktioner som kör som andra användare

Använd följande syntax i fältet `EXEC_STRING`:

```
EXEC_STRING dtaction -user användardnamn funktionsnamn [filargument]
```

Användaren (*användardnamn*) måste ha skärmbehörighet för systemet via någon av följande mekanismer:

- Läsrättigheter till den inloggade användarens .Xauthority-fil
- *Eller* xhost-behörighet

Följande funktioner ger exempelvis möjligheten att bli superanvändare och redigera parameterfiler.

```
ACTION AppDefaults
{
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING    /usr/dt/bin/dtaction -user root \
                  EditAppDefaults %Arg_1"File:"%
}
ACTION EditAppDefaults
{
    WINDOW_TYPE    TERMINAL
    EXEC_STRING    /bin/sh -c 'chmod +w %Arg_1%; \
                  vi %Arg_1%; chmod -w %Arg_1%'
}
```

Skapa språkanpassade funktioner

Sökvägen till datatyper omfattar språkberoende adresser. Skrivbordet använder värdet på LANG för att bestämma sökta adresser för datatypsdefinitioner.

Adresser för språkanpassade funktioner

Språkanpassade funktionsdefinitioner måste placeras i rätt språkberoende mapp i funktionens sökväg.

Standardsökvägen är:

- Egna funktioner: *Hemmapp*/.dt/types
- Systemtäckande funktioner: /etc/dt/appconfig/types/*språk*
- Inbyggda funktioner: /usr/dt/appconfig/types/*språk*

▼ Språkanpassa en befintlig funktion

1. **Skapa en fil i rätt språkberoende mapp (t ex i /etc/dt/appconfig/types/sv).**

2. Anta att du kopierar funktionsdefinitionen från

```
programrot/dt/appconfig/types/C/fil.dt
```

till

```
programrot/dt/appconfig/types/sv/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. Språkanpassa något av följande fält i funktionsdefinitionen:

- Språkanpassade ikoner: ICON
- Språkanpassad objekthjälp: DESCRIPTION
- Språkanpassade ledtexter: 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
    ...
}
```

Fälten addressing och disposition

- 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-typfunktioner:

- Objektorienterade ToolTalk-meddelanden
- Sammanhangsargument i meddelanden

Nyckelord för TT_MSG -funktioner

Tabell 12-3 visar nyckelord och användning för TT_MSG-funktioner.

TABELL 12-3 TT_MSG Funktionsnyckelord och användning

Nyckelord	Användning
TT_CLASS	Definierar värdet på ToolTalk-meddelandefältet class
TT_SCOPE	Definierar värdet på ToolTalk-meddelandefältet scope
TT_OPERATION	Definierar värdet på ToolTalk-meddelandefältet operation
TT_FILE	Definierar värdet på ToolTalk-meddelandefältet file
TT_ARGn_MODE	Definierar värdet på ToolTalk-attributet mode för det <i>n</i> :te meddelandeargumentet
TT_ARGn_VTYPE	Definierar värdet på ToolTalk-attributet vtype för det <i>n</i> :te meddelandeargumentet
TT_ARGn_VALUE	Definierar värdet på det <i>n</i> :te meddelandeargumentet

Skapa datatyper manuellt

Du kan skapa datatypsdefinitioner på två sätt:

- Med verktyget Skapa funktion. Skapa funktion beskrivs i Kapitel 11.
- Manuellt.

När du skapar en datatyp manuellt måste du redigera en databasfil.

I det här kapitlet beskrivs hur du manuellt skapar datatypsdefinitioner i följande avsnitt:

- "Varför du måste skapa datatyper manuellt" på sidan 199
- "Komponenter i datatypsdefinitionen: Villkor och attribut" på sidan 200
- "Skapa datatyper manuellt: Allmänna steg" på sidan 200
- "Exempel på hur du skapar en egen funktion och datatyp" på sidan 202
- "Definiera dataattribut för en datatyp" på sidan 206
- En introduktion till datatyper finns i Kapitel 10".
- Mer information om datatypsdefinitioner finns i direkthjälpen (man) för dtddsfile(4).

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 andra funktioner än Open (Öppna) och Print (Skriv ut) som associeras med datatypen.
- Flera namn-, mönster- eller innehållsvillkor för samma datatyp – exempelvis en datatyp som är baserad på filer som heter *.abc eller *.def
- Länkbaserade datatyper

Komponenter i datatypsdefinitionen: Villkor och attribut

En datatypsdefinition består av två separata databasdefinitioner:

- DATA_ATTRIBUTES -definitionen.

Definitionen av DATA_ATTRIBUTES beskriver datatypens namn samt utseende och egenskaper hos filer av den här typen.

- Fältet DATA_CRITERIA -definitionen.

Fältet 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-definition. 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. Sedan kan du skapa två separata villkor för PostScript-datatypen – ett som är baserat på filnamn och ett som är baserat på filinnehåll.

Mer information finns i "Definiera dataattribut för en datatyp" på sidan 206.

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 – *Hemmapp*/.dt/types

Systemtäckande datatyper – /etc/dt/appconfig/types/*språk*

Inbyggda datatyper – /usr/dt/appconfig/types/*språk*. Den här mappen ska du inte använda.

Mer information om hur du ändrar databassökvägen finns i "Ange värde för en sökväg" på sidan 133.

▼ Skapa en datatypsdefinition

1. Öppna en befintlig databasfil eller skapa en ny.

Mer information finns i föregående avsnitt "Konfigurationsfiler för datatyper" på sidan 201.

2. Definiera dataattribut för datatypen med följande syntax:

```
DATA_ATTRIBUTES datatypsnamn
{
    ICON          bildnamn
    DESCRIPTION   sträng
    attributfält
    attributfält
    ...
}
```

där

datatypsnamn är ett unikt namn som ges till den här datatypen.

bildnamn är ett filnamn eller en sökväg för en ikonfil. Använd filens basnamn. För ikonfilerna *minbild.m.pm* och *minbild.t.pm* används namnet *myimage*.

attributfält definierar utseende eller beteende för datatypen.

sträng är en teckensträng. Innehållet blir objekthjälp för datatypen.

Se "Exempel på hur du skapar en egen funktion och datatyp" på sidan 202.

3. Definiera datatypens datavillkor med hjälp av följande syntax:

```
DATA_CRITERIA villkorsnamn
{
    DATA_ATTRIBUTES_NAME   datatypsnamn
    villkorsfält
    villkorsfält
    ...
}
```

, där

villkorsnamn är ett unikt namn för den här villkorsdefinitionen

datatypsnamn är ett namn som används i definitionen `DATA_ATTRIBUTES`

villkorsfält är ett fält som används för att definiera villkoren för tilldelning av en fil till den här datatypen

Se: "Definiera dataattribut för en datatyp" på sidan 206.

4. Spara databasfilen.

5. Skapa datatypens ikoner

Mer information finns i "Ange ikonbild för datatypen" på sidan 203.

6. Skapa de funktioner som finns i fältet `ACTIONS` i attributdefinitionen.

7. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg för att läsa om databasen.

Exempel på hur du skapar en egen funktion och datatyp

Anta att det i ditt system finns ett program som heter `xgif`, som visar GIF-bilder. Normalt startas programmet med kommandot

```
xgif filnamn.
```

Du vill kunna visa GIF-bilder på flera olika sätt:

- Genom att dubbelklicka på en GIF-datafil
 - Genom att markera datafilen och välja programmet från Markera-menyn
1. **Öppna den nya filen *Hemmapp*/.dt/types/Gifvisning.dt för redigering.**

2. Skriv datatypsdefinitionerna:

```
DATA_ATTRIBUTES Gif
{
  DESCRIPTION  GIF-bildfil.
  ICON         GifIcon
  ACTIONS      View
}
DATA_CRITERIA Gifvillkor
{
  DATA_ATTRIBUTES_NAME  Gif
  NAME_PATTERN           *.gif
}
```

3. Skriv funktionsdefinitionen för funktionen Gifvisning:

```
ACTION Gifvisning
{
  EXEC_STRING      xgif %(File)Arg_1"Gif-fil som ska visas:"
  WINDOW_TYPE      NO_STDIO
  DESCRIPTION       Starta Gif-läsaren genom att \
                    dubbelklicka eller släppa filen.
}
```

Eftersom definitionen inte omfattar något `ICON`-fält använder funktionen systemets standardikon.

4. Skapa följande tilldelningsfunktion för att koppla funktionen Gifvisning till funktionen View i datatypsdefinitionen. Använd fältet `ARG_TYPE` för att begränsa denna visningsfunktion till GIF-filer.

```
ACTION View
{
  ARG_TYPE      Gif
  TYPE          MAP
  MAP_ACTION    Gifvisning
}
```

5. Spara filen.

6. Dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

Definiera datatypens dataattribut

Definitionen `DATA_ATTRIBUTES` definierar utseende och beteende för datatypen. Den anger namn på datatypen och ger dig möjlighet att ange:

- Ikonen Filhanteraren (fältet `ICON`)
- Metoden dubbelklick och innehåll på Markera-menyn (fältet `ACTIONS`)
- Datatypens objekthjälp (fältet `DESCRIPTION ACTIONS`, fält

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 t ex filer får namnet `Spelikon.m.pm` och `Spelikon.t.pm`, använder du `Spelikon`.

Om du använder basfilnamnet måste ikonfilerna placeras i en mapp i ikonsökvägen:

- Egna ikoner: `Hemmapp/.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 `Spelikon.m.pm` finns i mappen `/doc/projects`, som inte finns i ikonsökvägen, ska värdet i fältet `ICON` vara `/doc/projects/Spelikon.m.pm`.

Tabell 13-1 listar de ikonstorlekar du ska skapa och motsvarande filnamn.

TABELL 13-1 Ikonnamn och storlekar för datatypsikoner

Storleken i bildpunkter	Bitmappsnamn	Bildpunktsmönsternamn
32 gånger 32	<code>namn.m.bm</code>	<code>namn.m.pm</code>
16 gånger 16	<code>namn.t.bm</code>	<code>namn.t.pm</code>

Associera datatyper med funktioner

Du kan associera datatyper med funktioner på två sätt:

- Fältet `ACTIONS` i definitionen `DATA_ATTRIBUTES` listar de funktioner som visas på Filhanterarens Markera-menyn. Den första funktionen i listan är standardfunktionen (dubbelklick).
- Funktioner begränsas till angivna datatyper med funktionsdefinitionens `ARG_TYPE`-fält.

Följande datatypsdefinition skapar t ex en datatyp för särskilda "viktig"-filer (readme-filer) som skapas av systemadministratören och använder namnkonventionen `*.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, skickas 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

Tabell 13-2 visar `DATA_ATTRIBUTES`-fälten, vilka mest används 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.

TABELL 13-2 DATA_ATTRIBUTES Fält och beskrivningar

Fält	Beskrivning
MOVE_TO_ACTION	För behållare som t ex mappar. Anger en funktion som ska köras när en fil flyttas till en behållare av den här datatypen.
COPY_TO_ACTION	För behållare som t ex mappar. 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 typ av ToolTalk-medium.
MIME_TYPE	Anger motsvarande MIME-typ.
X400_TYPE	Anger motsvarande X400-typ.

Definiera dataattribut för en datatyp

Definitionen för DATA_CRITERIA anger de villkor som används för att tilldela en fil eller mapp en datatyp

Du kan använda villkoren som definieras i Tabell 13-3 för objekttypsbestämning.

TABELL 13-3 DATA_CRITERIA Villkor och beskrivningar

Villkor	Beskrivning
Filnamn	Filnamnet måste matcha ett angivet mönster. Använd fältet NAME_PATTERN.
Filens sökväg	Sökvägen måste matcha ett angivet mönster. Använd fältet PATH_PATTERN.
Filinhåll	En angiven del av filens innehåll måste matcha angivna data. Använd fältet CONTENT.

TABELL P-3 DATA_CRITERIA Villkor och beskrivningar (forts.)

Villkor	Beskrivning
Filläge	Filen måste ha de angivna behörigheterna (läsa, ändra, använda, mapp). Använd fältet <code>MODE</code>
Symboliska länkar	Typen beror på filen som objektet har länkats till.

Du kan använda fler än ett villkor för en datatyp. Du bör däremot inte använda villkoren `NAME_PATTERN` eller `PATH_PATTERN` i samma datatyp.

Namnbaseerade datatyper

Använd fältet `NAME_PATTERN` för att ange namngivningskrav. Fältvärdet kan innehålla följande jokertecken:

? – matchar ett valfritt tecken

* – matchar noll eller flera tecken i följd

[*tt*...] – matchar något av de tecken (*t*) som omsluts av hakparentes

[*t-t*] – matchar något av tecknen i intervallet *t t o m 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 \
                 projektet QS.
  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 mappar med namnet `Demo_n` där *n* är i intervallet mellan 0 och 9.

```
DATA_ATTRIBUTES Demomapp
{
```

```

DESCRIPTION    Det här är en mapp. 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    Demomapp
}

```

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 Projektgrafik
{
  DESCRIPTION    Grafikfil för projektet QS. Dubbelklicka på ikonen \
                om du vill visa grafiken.
  ICON           QSgraphics
}
DATA_CRITERIA Projektgrafikvillkor
{
  DATA_ATTRIBUTES_NAME    Projektgrafik
  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_källfiler` som definieras nedan gäller för alla filer med namnet `appn.c` (där `n` ligger inom intervallet 1 till 9) som finns i undermappar till `*/projects/QS`.

```

DATA_ATTRIBUTES QS_källfiler
{
  ...
}
DATA_CRITERIA QS_källfilsvillkor
{
  PATH_PATTERN            */projects/QS/*/app[1-9].c
  DATA_ATTRIBUTES_NAME    QS_källfiler
}

```

- Följande datatyp gäller för alla filer i mappen /doc/project1 som heter *chm.xxx* där *n* är i intervallet mellan 0 och 9 och *xxx* är ett valfritt filnamnsuffix på tre tecken.

```
DATA_ATTRIBUTES Kapitelfiler
{
  DESCRIPTION  Kapitelfil för projektdokumentet.
  ICON         chapter
  ACTIONS      Edit,Print
}

DATA_CRITERIA Kapitelvillkor
{
  PATH_PATTERN  /doc/project1/ch[0-9][0-9].???
  DATA_ATTRIBUTES_NAME  Kapitelfiler
}
```

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 mapp, eller ange om det krävs behörighet för att läsa, skriva eller använda.

Fältet `MODE` kan innehålla logiska operatorer (Tabell 13-4) och tecken (Tabell 13-5).

TABELL 13-4 `MODE` Logiska operatorer

Operator	Beskrivning
!	NOT, logisk operator
&	AND, logisk operator
	OR, logiskt eller

TABELL 13-5 MODE Fälttecken och beskrivningar

Tecken	Beskrivning
f	Datatypen gäller bara filer
d	Datatypen gäller bara mappar
r	Filen kan läsas av alla användare
w	Filen kan ändras av alla användare.
x	Filen kan köras av alla användare.
l	Filen är en länk

Standardinställningen för ett behörighetsläge är att läget inte spelar någon roll.

Exempel

- Följande fält begränsar datatypen enligt följande:

f&!w – skrivskyddade filer

!w – skrivskyddade filer och mappar

f&x – körbara filer

f&r&x – filer som både kan ändras och köras

x|!w – filer som är körbara eller skrivskyddade

- Följande datatypsdefinition skapar en datatyp för skrivskyddade, icke körbara filer vars filnamn följer namnkonventionen *.doc. Den förutsätter att det redan har skapats en visningsfunktion för datatypen.

```
DATA_ATTRIBUTES Skrivskyddat_dokument
{
    ICON      read_only
    DESCRIPTION  Det här dokumentet kan inte ändras. Om du \
                dubbelklickar körs redigeraren med en \
                skrivskyddad kopia av filen.
    ACTIONS    View
}

DATA_CRITERIA Skrivskyddat_dokument_villkor
{
    NAME_PATTERN      *.doc
    MODE              !d&!x&!w
    DATA_ATTRIBUTES_NAME  Skrivskyddat_dokument
}
```

```
}
```

Innehållsbaserad datatypsbestämning

Använd fältet `CONTENT` för att ange datatyp baserat på filens innehåll. Innehållsbaserad datatypsbestämning kan användas i kombination med namn- eller platsbaserad datatypsbestämning.

Typbestämningen kan baseras på texten i filen eller på dess numeriska innehåll. Numret på filens första byte är 0.

- Typbestämning med hjälp av filens innehåll använder syntaxen:

```
CONTENT startbyte string sträng
```

- För numeriskt innehåll använder du syntaxen:

```
CONTENT startbyte byte antal  
CONTENT startbyte short antal  
CONTENT startbyte long antal
```

- För mappinnehåll använder du syntaxen:

```
CONTENT 0 filename "filnamn"
```

Använd den vanliga C-beteckningen för oktala (inledande `o`) 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.

Datatypen `Skrivbar_Wingz` gäller för alla filer med skrivbehörighet som innehåller strängen `WNGZ` i början av filen.

```
DATA_ATTRIBUTES Skrivbar_Wingz  
{  
  ...  
}  
  
DATA_CRITERIA Skrivbar_Wingz_villkor  
{  
  CONTENT      0 string WNGZ  
  MODE         w&!d  
  DATA_ATTRIBUTES_NAME Skrivbar_Wingz  
}
```

▼ Skapa en datatyp med flera oberoende villkor

Du kan skapa en datatyp med flera oberoende villkor – dvs filen tilldelas datatypen om den minst motsvarar ett av villkoren.

1. Skapa en `DATA_ATTRIBUTES`-definition 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_namnsvillkor
{
    DATA_ATTRIBUTES_NAME Mif
    NAME_PATTERN          *.mif
}

DATA_CRITERIA Mif_innehållsvillkor
{
    DATA_ATTRIBUTES_NAME Mif
    CONTENT               1 string MIFfile
}
```

Skapa språkpassade datatyper

Sökvägen för datatyper omfattar språkberoende adresser. Skrivbordet använder värdet på LANG för att bestämma på vilka adresser datatypsdefinitioner söks.

Adresser för språkpassade datatyper

Språkpassade datatypsdefinitioner måste placeras i rätt språkberoende mapp i funktionssökvägen.

Standardsökvägen är:

- Egna funktioner: *Hemmapp/.dt/types*
- Systemtäckande funktioner: */etc/dt/appconfig/types/språk*
- Inbyggda funktioner: */usr/dt/appconfig/types/språk*

▼ Språkpassa en datatyp

1. **Skapa en fil i rätt språkberoende mapp (t ex i */etc/dt/appconfig/types/sv*).**

2. Kopiera datatypsdefinitionen till den språkberoende konfigurationsfilen.
3. Språkänpassa ett eller flera fält i datatypsdefinitionen.

Skapa skrivbordsikoner

Skrivbordsikoner kan kopplas till:

- Funktionsfiler och datatyper i Filhanteraren och Programhanteraren
- Ikoner på Frontpanelen
- Minimerade programfönster
- Grafik som används av program, till exempel paletter och verktygsfält.
- Arbetsytans bakgrund

Läs mer om skrivbordsikoner i följande avsnitt:

- "Ikonbilsfiler" på sidan 215
- "Associera ikoner" på sidan 218
- "Rekommendationer hur du skapar ikoner" på sidan 221

Obs! Dokumentationen för utvecklingsmiljön innehåller ytterligare information om skrivbordsikoner. Se kapitel 4, "Visual Design," i *Common Desktop Environment: Style Guide and Certification Checklist*.

Ikonbilsfiler

Om skrivbordet ska använda ikonbilder måste filerna:

- Ha rätt format
- Använda rätt filnamnsregler
- Använda skrivbordets storlekskonventioner
- Finnas i en mapp i ikonsökvägen

- Anropas från skrivbordet med rätt syntax. Om du t ex vill skapa en ny kontroll för Frontpanelen ska du använda `ICON`-fältet i definitionen för Frontpanelen för att ange den ikonbild som ska användas för kontrollen.

Ikonfilformat

För en färgskärm kan du använda ikonfiler med bildpunktsmönsterformatet X-pixmap (XPM), som normalt har suffixet `.pm`. Annars använder du filer med bitmappsformatet X-bitmap (XBM), vilka normalt har suffixet `.bm`. Om genomskinlighet används i pixmap-filen genereras en ikonmaskfil (`_m.bm`) när filen `.bm` skapas. Mer information om hur skrivbordet hittar dessa filer finns i "Ikonsökväg" på sidan 139.

Ikonfilnamn

Alla ikoner och bakgrundsbilder lagras som separata filer. Normalt visar basdelen i filnamnet att den innehåller en ikon. En ikon kan t ex refereras till med hjälp av 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

Eller

basnamn.format

där:

basnamn är bildens basnamn som används som referens till bilden

storlek är en bokstav som anger storleken: `l`=large (stor), `m`=medium (normal), `s`=small (liten) och `t`=tiny (mycket liten)

format är filformatet: `pm` (bildpunktsmönster), `bm` (bitmapp)

Storlekskonventioner

Tabell 14–1 visar rekommenderad bildpunktsstorlek för skrivbordsikonen.

TABELL 14-1 Ikonstorlekar och filnamn

Ikonstorlek	Bitmappsnamn	Bildpunktmönsternamn
16 gånger 16 (mycket liten)	<i>namn</i> .t.bmp	<i>namn</i> .t.ppm
24 gånger 24 (liten)	<i>namn</i> .s.bmp	<i>namn</i> .s.ppm
32 gånger 32 (normal)	<i>namn</i> .m.bmp	<i>namn</i> .m.ppm
48 gånger 48 (stor)	<i>namn</i> .l.bmp	<i>namn</i> .l.ppm

Tabell 14-2 visar ikonstorleken som används av skrivbordets komponenter. I vissa fall bestäms ikonstorleken av upplösningen på skärmen.

TABELL 14-2 Skrivbordets komponenter och deras ikonstorlekar

Skrivbordskomponent	Hög upplösning	Normal upplösning	Låg upplösning
Filhanteraren och Programhanteraren (namn och ikon)	normal	normal	normal
Filhanteraren och Programhanteraren (namn och liten ikon)	mycket liten	mycket liten	mycket liten
Frontpanelens huvudkontroller	stor	stor	normal
Frontpanelens panelundermenyer	normal	normal	mycket liten
Frontpanelens arbetsytteomkopplare	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.ppm`.

Ikonsökvägen

Skrivbordet söker efter en ikonfil eller en bild genom att leta efter filen i en lista över mappar. Denna lista kallas *ikonsökväg* och bestäms av värdet på ett flertal miljövariabler. Mer information om vilka variabler som används och hur de läggs ihop för att skapa ikonsökvägen finns i "Ikonsökväg" på sidan 139.

Standardsökväg:

- Inbyggda ikoner: `/usr/dt/appconfig/icons/språk`
- Systemtäckande ikoner: `/etc/dt/appconfig/icons/språk`
- Egna ikoner: `Hemmapp/.dt/icons`

Komma åt ikoner via nätverket

Skrivbordet kan även använda ikoner i fjärrsystem. Mer information om hur du skapar en ikonserver finns i "Konfigurera databas-, ikon- och hjälptjänster" på sidan 118.

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 i funktioner, datatyper och på Frontpanelen eller panelundermenyer, anger du bara ikonens basnamn (inga suffix). Rätt suffix läggs till automatiskt utifrån skärmutplösning, färgstöd och alternativ för filhanterarfönstret (t ex Små ikoner).

För att åsidosätta sökvägen anger du fullständig sökväg och namn på ikonen.

▼ Associera 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 funktionsdefinitioner för att köra 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"Öppna fil"%
}
```

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 komprimerad fil har blivit komprimerad av \
                kommandot 'compress' för att ta upp mindre utrymme.

}
```

▼ Visa en ikon i Frontpanelen

1. Ange bildens namn i 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:

- Frontpanelen och panelundermenyer: stor, normal och mycket liten

- Arbetsyteomkopplaren: liten

Exempel

Följande ikon ändrar utseende när en fil med namnet `rapport` placeras i mappen `/doc/ftp/pub/`. När filen inte hittas kommer `NoReport.pm`-ikonen att visas. När filen hittas kommer `Report.pm` att visas.

```
CONTROL MonitorReport
{
  CONTAINER_NAME    namn
  TYPE              ICON
  MONITOR_TYPE      file
  FILE_NAME         /doc/ftp/pub/report
  ICON              NoReport
  ALTERNATE_ICON    Report
}
```

▼ Associera 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 Fönsteregenskaper i programgruppen Skrivbordsverktyg. När du markerar ett fönster visas dess egenskaper. Egenskapen `WM_CLASS` visar fönstrets klassnamn inom citattecken.

Mer information om hur du anger resurser finns i .

2. Välj Starta om Arbetsytehanteraren på 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.

▼ Använda Filhanteraren som ett ikonfönster

1. Kopiera filen `/usr/dt/examples/språk/IconBrowse.dt` till mappen `Hemmapp/.dt/types/Iconbrowse.dt`.

2. Öppna Programhanteraren och dubbelklicka på Ladda om funktioner i programgruppen Skrivbordsverktyg.

När du växlar till en mapp som innehåller ikoner (.bm och .pm-filer) visas varje ikon intill sitt namn. Om du t ex växlar till mappen `/usr/dt/appconfig/icons/språk`, visas många av skrivbordsikonerna.

Obs! Om du aktiverar ikonfönstret i ett system med lite minne kan det hända att Filhanteraren visar mappar mycket långsammare. Bilder som är större än 256 x 256 bildpunkter 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.

Rekommendationer hur 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 som du skapar är giltig. Om ikonen visas på en monokrom skärm eller gräskaleskärm (eller om det inte finns tillräckligt med tillgängliga färger) visas ikonen automatiskt i tvåfärgsformatet.

Om du vill sänka antalet färger kan du begränsa färgintervallet i en ikon genom att använda färgerna från skrivbordet. (Ikoner som är skapade med ikonredigeraren kommer endast att använda skrivbordsfärger.)

Mer information om vilka storlekar som används av skrivbordets komponenter finns i Tabell 14-1.

Hur färger används

Skrivbordet använder en palett med 22 färger, bl a:

- Åtta statiska gräskalor
- 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 snygga 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 använder du när du vill skapa ikoner som verkar icke-rektangulära, eftersom färgen bakom ikonerna skiner igenom.

Särskild anpassning av Frontpanelen

Användarna kan anpassa Frontpanelen och dess meny och använda funktionen Installera Ikon på panelundermenyerna.

I det här kapitlet beskrivs hur du anpassar Frontpanelen genom att skapa och redigera konfigurationsfiler.

- "Konfigurationsfiler för Frontpanelen" på sidan 223
- "Administrera anpassning av användargränssnitt" på sidan 226
- "Hur frontpanelsdefinitionen är organiserad" på sidan 227
- "Anpassa huvudpanelen" på sidan 230
- "Skapa och ändra panelundermenyer" på sidan 234
- "Definiera kontroller på Frontpanelen" på sidan 238
- "Anpassa området för växling av arbetsyta" på sidan 245
- "Allmän konfiguration av Frontpanelen" på sidan 246
- Referensinformation om Frontpanelens kontroller och konfiguration finns i direkthjälpen (man) för `dtfpfile(4X)`.
- Referensinformation om Arbetsytahanteraren finns i direkthjälpen (man) för `dtwm(1)` och `dtwmrc(4)`.

Konfigurationsfiler för Frontpanelen

Frontpanelen definieras i en databas med konfigurationsfiler.

Med konfigurationsfilerna anpassar du Frontpanelen. Vissa ändringar kan du bara göra genom att redigera en konfigurationsfil, bl a:

- Lägga till en ny kontrollposition på huvudpanelen
- Lägga till speciella typer av kontroller, t ex klientfönster
- Ändra vissa standardbeteenden – exempelvis om Frontpanelens kontroller ska svara på klick eller dubbelklick

För att konfigureringen ska vara så flexibel som möjligt kan dessa filer vara egna eller 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

Frontpanelsdefinitionen kan distribueras mellan ett stort antal filer som finns lagrade lokalt eller i fjärrsystem.

Filer som används för att definiera Frontpanelen måste uppfylla följande krav:

- Filnamnet måste sluta på `.fp`, exempelvis `mail.fp`
- Filen måste finnas i funktionsdatabasens sökväg

Sökvägen till standardfunktionsdatabasen omfattar följande mappar, vilka söks i denna ordning:

Egna anpassningar – *Hemmapp*/`.dt/types`

Systemtäckande anpassningar – `/etc/dt/appconfig/types/språk`

Inbyggd panel och kontroll – `/usr/dt/appconfig/types/språk`

Ytterligare en mapp, *Hemmapp*/`.dt/types/fp_dynamic`, används för egna anpassningar som har utförts med hjälp av användargränssnittet. Använd inte den här mappen för manuella anpassningar.

Sökvägen till funktionsdatabasen kan innehålla ytterligare mappar som har lagts till för att konfigurera det system 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 137.

Hur Frontpanelen sätts samman: prioritetsregler

Frontpanelen sätts ihop ifrån alla konfigurationsfiler som finns i funktionsdatabasens sökväg.

När det uppstår en konflikt mellan olika delar av definitionen, bestämmer prioritetsreglerna vilken definition som ska användas. Två komponenter krockar med varandra när de:

- har samma kontrollnamn, CONTAINER_NAME och CONTAINER_TYPE.
- tävlar om samma placering (genom att ha olika namn men samma CONTAINER_NAME, CONTAINER_TYPE och POSITION_HINTS).

Frontpanelen använder följande prioritetsregler:

- Om olika komponenter har samma kontrollnamn, behållarnamn (container_name) och behållartyp (container_type), används den komponent som läses först

Om t ex både en systemtäckande och en inbyggd kontroll innehåller de här fälten 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, flyttar 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 kontroll 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 mappen *Hemmapp/.dt/types/fp_dynamic*.

Frontpanelen skapar ytterligare en fil, *Hemmapp/.dt/sessions/dtwmfp.session*, som används för att spara och återställa status för anpassade Frontpanelen för varje session.

Administrera anpassning av användargränssnitt

Användare kan använda snabbmenyerna och kontrollerna för ikoninstallation på Frontpanelen när de vill göra betydande anpassningar av 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.
- Ångrar egna anpassningar. En användare kanske vill att du ska återställa en kontroll som har 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/namn.fp`.

2. Lägg till följande rad i kontrolldefinitionen:

```
LOCKED True
```

▼ Återställa en borttagen kontroll eller panelundermeny

Frontpanelsfunktionen 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 har gjorts med Frontpanelens menyer.

Använd proceduren nedan för att återställa individuella kontroller.

- ◆ I mappen `Hemmapp/.dt/types/fp_dynamic` tar du bort filen som skapades när användaren tog bort kontrollen. Kontrollen har samma namn som den originalkontroll som togs bort.

Om användaren t ex tog bort kontrollen för Ikonredigeraren, innehåller en fil i mappen `fp_dynamic`:

```
CONTROL IconEditor  
{
```

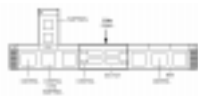
```
...  
DELETE True  
}
```

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 läggs ihop. 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 15-1 Komponenter på Frontpanelen

Frontpanelen sätts samman från utsidan och inåt:

- En panel (**PANEL**) är behållaren på Frontpanelens översta nivå (den överordnade).
- En **PANEL** är en behållare som innehåller en eller flera rutor (**BOXes**).
- En **BOX** är en behållare som innehåller en eller flera kontroller (**CONTROLS**).

Det finns två specialtyper av behållare:

- En panelundermeny (**SUBPANEL**) är kopplad till en viss kontroll (kontrollen är behållaren för panelundermenyn). Panelundermenyer "glider upp" från den kontroll som de är associerade till.
- En arbetsytväxlare (**SWITCH**) innehåller knappar för att byta arbetsyta och 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 direkthjälpen (man) för dtfpfile(4X).

PANEL-definition

En PANEL är komponenten på den översta nivån. Dess definition innehåller:

- Frontpanelens namn
- 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".

BOX-definitioner

En BOX-definition beskriver:

- Rutans namn
- vilken PANEL rutan finns i (CONTAINER_NAME)
- Rutans placering på PANEL (POSITION_HINTS)
- Fält som beskriver utseende och metod för hela rutan

```

BOX rutans namn
{
  CONTAINER_NAME      frontpanelnamn
  POSITION_HINTS       position
  KEYWORD              värde
  KEYWORD              värde
  ...
}

```

CONTROL, definitioner

Definitionen för en kontroll (CONTROL) beskriver:

- Kontrollens namn

- Om kontrollen finns i en ruta (BOX), panelundermeny (PANEL) eller arbetsytväxlare (SWITCH) (CONTAINER_TYPE)
- Namnet på den ruta, panelundermeny eller arbetsytväxlare kontrollen finns i (CONTAINER_NAME)
- Kontrollens position i rutan (POSITION_HINTS).
- Fält som beskriver utseende och metod för kontrollen

```
CONTROL Kontrollnamn
{
  CONTAINER_TYPE      BOX eller SUBPANEL eller SWITCH
  CONTAINER_NAME      rutans namn eller panelundermenyns namn eller arbetsytväxlarens namn
  TYPE                kontrolltyp
  POSITION_HINTS       position
  KEYWORD              värde
  KEYWORD              värde
  ...
}
```

SUBPANEL, definitioner

Definitionen för en panelundermeny (SUBPANEL) beskriver:

- panelundermenyns namn
- Namnet på kontrollen som panelundermenyn är kopplad till (CONTAINER_NAME)
- Fält som beskriver utseende och metod för panelundermenyn

```
SUBPANEL panelundermenyns namn
{
  CONTAINER_NAME      kontrollnamn
  KEYWORD              värde
  KEYWORD              värde
  ...
}
```

SWITCH, definitioner

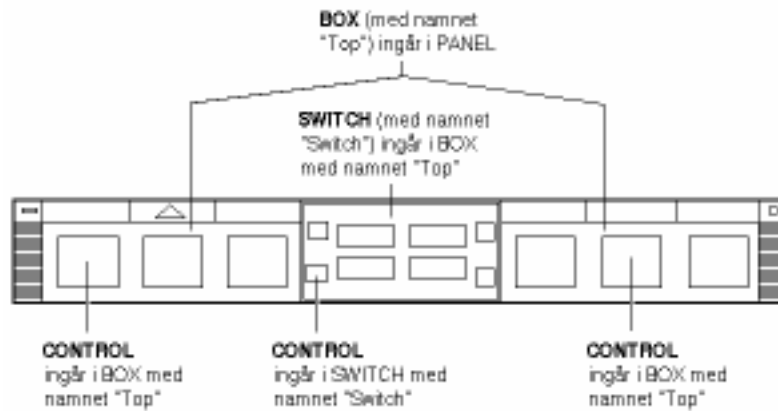
Definitionen av en arbetsytväxlare (SWITCH) beskriver:

- arbetsytväxlarens namn
- Den ruta som arbetsytväxlaren finns i (CONTAINER_NAME)
- Arbetsytväxlarens position i rutan (POSITION_HINTS)
- Fält som beskriver utseende och metod för arbetsytväxlaren

```
SWITCH namn
{
  CONTAINER_NAME      rutans namn
  POSITION_HINTS       position
  KEYWORD              värde
  KEYWORD              värde
  ...
}
```

Anpassa huvudpanelen

Huvudpanelen är Frontpanelen förutom dess underpanelmenyer.



Figur 15-2 Behållare på huvudpanelen

De ändringar du kan göra är bl a:

- Lägg till och ta bort kontroller
- Byta plats på kontroller

▼ Lägg till en kontroll på huvudpanelen

1. Skapa Frontpanelens konfigurationsfil:

- Systemtäckande: `/etc/dt/appconfig/types/språk/*.fp`
- Egen: `Hemmapp/.dt/types/*.fp`

2. Definiera kontrollen i filen.

Använd fältet `CONTAINER_NAME` and `CONTAINER_TYPE` för att ange behållare för kontrollen:

```
CONTAINER_NAME Top
CONTAINER_TYPE BOX
```

Använd `POSITION_HINTS` för att ange kontrollens placering (från vänster till höger). Eftersom anpassningar har högre prioritet än inbyggda kontroller kommer den nya kontrollen att "knuffa" en befintlig kontroll på samma plats åt höger.

3. Välj Starta om Arbetsytahanteraren på Arbetsyta-menyn.

4. Skapa en ikon för frontpanelkontrollen.

Se "Ange vilken ikonbild som ska användas av kontrollen" på sidan 234.

5. Välj Starta om Arbetsyتهanternaren på Arbetsyta-menyn.

Följande kontrolldefinition som finns 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
}
```

▼ Ta bort en kontroll

1. Skapa Frontpanelens konfigurationsfil:

- Systemtäckande: `/etc/dt/appconfig/types/språk/namn.fp`
- Egen: `Hemmapp/.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. Den del 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 Arbetsyتهanternaren på Arbetsyta-menyn.

Följande kontrolldefinition som finns i filen
`/etc/dt/appconfig/types/språk/TrashCan.fp` tar bort kontrollen för
papperskorgen från Frontpanelen.

```
CONTROL Trash
{
  CONTAINER_NAME Top
```

```
CONTAINER_TYPE BOX
DELETE True
}
```

▼ Ändra en kontroll

Använd den här proceduren om du behöver ändra en kontrolldefinition – exempelvis ändra dess ikonbild.

1. **Kopiera hela kontrolldefinitionen från**
`/usr/dt/appconfig/types/språk/dtwm.fp` till:
 - Systemtäckande: `/etc/dt/appconfig/types/språk/namn.fp`
 - Egen: `Hemmapp/.dt/typesnamn.fp`.
2. Redigera fältet du vill ändra. Du kan också lägga till ytterligare fält.
3. Spara filen.
4. Välj Starta om Arbetsytahanteraren på Arbetsyta-menyn.

▼ Byta 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: `Hemmapp/.dt/typesnamn.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 Arbetsytahanteraren 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 läser 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
}
```

▼ Ersätta 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.

- Definitionen `/etc/dt/appconfig/types/C/SysControls.fp`:

```
Control ImportantApplication
{
  CONTAINER_NAME      Top
  CONTAINER_TYPE      BOX
  POSITION_HINTS       2
  ... }

```

- Definitionen `Hemmapp/.dt/types/MyControls.fp`:

```
Control ImportantApplication
{
  CONTAINER_NAME      Top
  CONTAINER_TYPE      BOX
  POSITION_HINTS       6
  ... }

```

Den egna kontrollen har prioritet, så kontrollen läses 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 ikonerna minus filnamnsuffixet för storlek (m och t) och bildtyp (bm och pm). Om t ex filer kallas `MittSpel.l.pm` och `MittSpel.m.pm`, används `MittSpel`.

Om du använder basfilnamnet måste ikonfilerna placeras i en mapp i ikonsökvägen:

- Behållare på huvudpanelen: `Hemmapp/.dt/icons`
- Systemtäckande ikoner: `/etc/dt/appconfig/icons/språk`

- En absolut sökväg till ikonerna, inklusive fullständigt filnamn.

Du ska bara använda den absoluta sökvägen om ikonfilerna inte finns i ikonsökvägen.

Vilken ikonstorlek på ikonerna som behövs beror på var kontrollen placeras:

Platsstorlek

Huvudpanel – 48 gånger 48 bildpunkter (`namn/dtwm.fp` or `namn/.dt.types/MyFrontPanel.fp`)

Panelundermeny – 24 gånger 24 bildpunkter (`namn.l.pm` or `namn.l.bm`)

Huvudpanelen är Frontpanelens fönster minus panelundermenyerna.

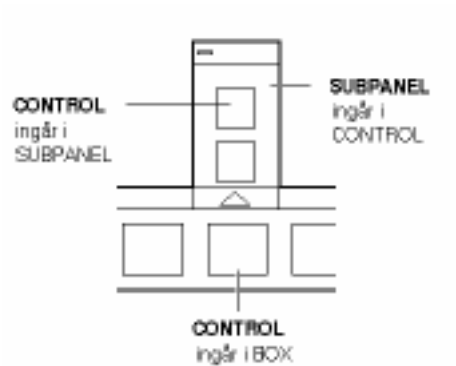
- Behållare på huvudpanelen `Hemmapp/.dt/icons`
- Systemtäckande ikoner: `/etc/dt/appconfig/icons/språk`

Skapa och ändra panelundermenyer

Användarna kan skapa och ändra panelundermenyerna med hjälp av Frontpanelens menyer.

I det här avsnittet beskrivs hur du gör systemtäckande anpassningar, vilket kräver att du ändrar Frontpanelens konfigurationsfiler.

En panelundermeny är ”kopplad” till en kontroll i huvudpanelen.



Figur 15-3 En panelundermenys behållare är kontrollen som den är kopplad till

Kopplingen görs i panelundermenyns definition. CONTAINER_NAME anger vilken kontroll som panelundermenyn är kopplad till:

```
CONTROL kontrollnamn
{
  ...
}

SUBPANEL panelundermenyns namn
{
  CONTAINER_NAME      kontrollnamn
  ...
}
```

▼ Skapa en ny systemtäckande panelundermeny

1. Leta reda på *namnet* på den kontroll 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 panelundermenyns namn
{
  CONTAINER_NAME      kontrollnam

  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.

Ändra allmänna egenskaper för en inbyggd panelundermeny

1. Skapa en ny konfigurationsfil för Frontpanelen:

- Systemtäckande: `/etc/dt/appconfig/types/språk/namn.fp`
- Egen: `Hemmapp/.dt/typesnamn.fp`.

2. Kopiera hela standarddefinitionen för SUBPANEL från `/usr/dt/appconfig/types/språk/dtwm.fp` till den nya filen:

```
SUBPANEL panelundermenys namn {  
    ...  
}
```

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  
}
```

Lägga till systemtäckande kontroller i en inbyggd panelundermeny

1. Skapa Frontpanelens konfigurationsfil: `/etc/dt/appconfig/types/språk/namn.fp`.
2. Definiera den systemtäckande kontrollen i filen.

Använd fältet CONTAINER_NAME och CONTAINER_TYPE för att ange behållare för kontrollen:

```
CONTROL kontrollnamn
{
  CONTAINER_NAME   panelundermenyns namn
  CONTAINER_TYPE   SUBPANEL
  ...
}
```

Se "Definiera kontroller på Frontpanelen" på sidan 238.

3. Välj Starta om Arbetsythanteraren på Arbetsyta-menyn.

4. Välj Starta om Arbetsythanteraren på Arbetsyta-menyn.

Följande kontroll som definieras i en ny fil /etc/dt/appconfig/types/*språk*/DigitalClock.fp lägger till DigitalClock (i programgruppen Skrivbordsverktyg) på panelundermenyn Egna program för alla program.

```
CONTROL DigitalClockControl
{
  TYPE          icon
  CONTAINER_NAME PerAppsSubpanel
  CONTAINER_TYPE SUBPANEL
  ICON          Dtdgclk
  PUSH_ACTION   DigitalClock
  PUSH_RECALL   True
}
```

Ta bort en kontroll från en inbyggd panelundermeny

- ◆ Använd samma procedur som för att ta bort kontroller på Huvudpanelen. Se "Ta bort en kontroll" på sidan 231.

Ta 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 den ursprungliga positionen.

Frontpanelen kan konfigureras så att panelundermenyerna hålls öppna tills användaren stänger dem.

1. Skapa en ny konfigurationsfil för Frontpanelen i:

- Systemtäckande: `/etc/dt/appconfig/types/språk/*.fp`
- Egen: `Hemmapp/.dt/types/*.fp`

2. Kopiera standardbeskrivningen för PANEL från

`/usr/dt/appconfig/types/språk/dtwm.fp` till den nya filen:

```
PANEL FrontPanel
{
  ...
}
```

3. Lägg till följande fält i definitionerna för PANEL:

```
SUBPANEL_UNPOST    False
```

4. Spara den nya konfigurationsfilen.

5. Välj Starta om Arbetsytahanteraren 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 har skapats med Installera ikon kan t ex inte:

- Innehålla 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 direkthjälpen (man) för 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 beteende
}
```

Kontrolltyper

Fältet `TYPE` i kontrolldefinitionen anger grundläggande beteende för den kontrollen.

Kontrolltyp (TYPE)	Beskrivning
<code>icon</code> (Standard).	Kontrollen kör en angiven funktion när användaren klickar på kontrollen eller släpper en fil på den.
<code>blank</code>	Platshållare för att justera avstånd mellan kontroller.
<code>busy</code>	Upptaget-lampa. Kontrollen blinkar (växlar bilder) när en funktion startas
<code>client</code>	Ett klientfönster i Frontpanelen.
<code>clock</code>	Klocka.
<code>date</code>	Visar aktuellt datum.
<code>file</code>	Representerar en fil. När du väljer kontrollen körs standardfunktionen för den filen.

▼ Skapa en ny kontroll

Detta avsnitt beskriver hur en kontroll och olika kontrolltyper skapas.

1. Om kontrollen ska ha någon avfälten `PUSH_ACTION` eller `DROP_ACTION` definierade skapar du motsvarande funktionsdefinitioner. Dessa funktioner körs när användaren klickar på kontrollen eller släpper en fil på den.
2. Skapa ikonbildens fil för kontrollen.
Mer information om ikonstorlek, ikonnamn och ikonadresser finns i "Ikonbilsfiler" på sidan 215.
3. Skapa en ny konfigurationsfil för Frontpanelen i:
 - Systemtäckande: `/etc/dt/appconfig/types/språk/*.fp`
 - Egen: `Hemmapp/.dt/types/*.fp`
4. Lägg till kontrolldefinitionen i filen.
5. Spara filen.
6. Välj Starta om Arbetsyتهanteraren på Arbetsyta-menyn.

Skapa en kontroll som kör en funktion när du klickar på den

Använd följande fält när du definierar kontrollens funktion:

- `TYPE`: Anges som `icon`
- `PUSH_ACTION`: Anger namnet på funktionen som ska köras

Följande kontroll, som placeras på panelundermenyn Egna program, kör ett spel som användaren har lagt in:

```
CONTROL Kula
{
  TYPE          icon
  CONTAINER_NAME  PersAppsSubpanel
  CONTAINER_TYPE  SUBPANEL
  ICON          kula
  PUSH_ACTION    RunBallGame
  HELP_STRING    "Välj den här kontrollen om du vill spela Kulspelet."
}
```

Följande kontroll placeras i det övre vänstra hörnet av arbetsyتهväxlaren. 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 om du vill köra cutdisp."
  PUSH_ACTION    CutDisp
}
```

Skapa en kontroll som öppnar en fil

Använd dessa fält om du vill definiera kontrollens funktion.

- **TYPE:** Anges som file
- **FILE_NAME:** Anger sökvägen till filen som ska öppnas
- **PUSH_ACTION:** Anges som 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 ett målområde

Använd fältet `DROP_ACTION` för att ange vilken funktion som ska köras när användaren släpper en fil på kontrollen. Funktionen måste kunna acceptera filargument.

En kontrolldefinition innehåller ofta fälten `PUSH_ACTION` och `DROP_ACTION`. Du kan använda samma funktion för en dra och släpp-funktion.

Följande kontroll på panelundermenyn Eegna 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 om du vill definiera kontrollens funktion.

- **TYPE:** Anges 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 en fil när du dubbelklickar på filens ikon i Filhanteraren.

- `ICON` och `ALTERNATE_ICON`: Beskriver de bilder som används för att ange ändrad och icke ändras status för 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 den angivna filen blir icke-tom.

- `FILE_NAME`: Anger namnet på den fil som ska övervakas.

Följande kontroller letar exempelvis efter en fil med namnet `meetings` som ska överföras till ditt system med hjälp av anonym ftp. Kontrollen placeras i underpanelmenyn Egna program.

```
CONTROL MonitorCalendar
{
    TYPE          file
    CONTAINER_NAME    PersonalApps
    CONTAINER_TYPE    SUBPANEL
    POSITION_HINTS     first
    FILE_NAME        /users/ftp/meetings
    MONITOR_TYPE     file
    ICON             meetingsno
    ALTERNATE_ICON   meetingsyes
}
```

Skapa en växlingskontroll

En växlingskontroll kontrollerar om den process som ska startas av `PUSH_ACTION` redan körs. Om processen inte körs, körs `PUSH_ACTION`. Om processen redan körs flyttas fönstret högst upp i fönsterstacken på den aktuella arbetsytan.

Använd dessa fält om du vill definiera kontrollens funktion.

- `PUSH_RECALL`: Anges till `True`.
- `CLIENT_NAME`: Anger namnet på kontrollens klient.

Värdet på `CLIENT_NAME` måste matcha den första strängen (*res_name*) i egenskapen `WM_CLASS` i programmets primära fönster. Mer information finns i direkthjälpen (`man`) för `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
}

```

Skapa en klientfönsterkontroll

En klientfönsterkontroll är ett programfönster som är inbäddat på Frontpanelen. Du kan t ex placera en systembelastningsmätare i Frontpanelen genom att skapa en klientfönsterkontroll för `xload`.

1. Använd följande fält för att definiera kontrollens funktion:

- `TYPE`: Anges till `client`.
- `CLIENT_NAME`: Anger vilken klient som ska startas.
Värdet på `CLIENT_NAME` måste matcha den första strängen (`res_name`) i egenskapen `WM_CLASS` i programmets primära fönster. Mer information finns i direkthjälpen (`man`) för `xprop(1)`.
- `CLIENT_GEOMETRY`: Anger den storlek, (i bildpunkter) som är nödvändig för klientens frontpanelfönster.
Direkthjälpen (`man`) för `xwininfo(1)` beskriver hur du tar reda på storleken på ett fönster i bildpunkter.

2. Välj Starta om Arbetsytahanteraren på Arbetsyta-menyn.

3. Starta klienten från kommandoraden i ett terminalfönster.

Följande kontroll visar t ex en belastningsmätare på 30×20 bildpunkter.

```

CONTROL LoadMeter
{
  TYPE      client
  CONTAINER_NAME  Top
  CONTAINER_TYPE  BOX
  CLIENT_NAME    xload
  CLIENT_GEOMETRY 30x20
}

```

Om klienten inte sparas och återställs mellan sessionerna, vill du kanske konfigurera kontrollen så att klienten startar när användaren klickar på den. Du kan t ex konfigurera kontrollen för `LoadMeter` så att den startar `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
}

```

Animera en kontroll

Du kan skapa en animeringssekvens som används när användaren väljer kontrollen eller släpper ett objekt på den.

För att kunna innehålla 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
{
    icon_image    [fördröjning]
    icon_image    [fördröjning]
    ...
}
```

där *fördröjning* 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 hjälp till en kontroll tillgänglig:

- Lägga till 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:

- 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.

När du vill lägga in ett hjälpavsnitt anger du hjälpvolym och avsnitts-ID i kontrolldefinitionen:

```
HELP_VOLUME    hjälpvolymnamn
HELP_TOPIC     avsnitts-ID
```

Anpassa området för växling av arbetsyta

Det finns flera sätt att anpassa området för växling av arbetsyta:

- Ändra antalet arbetsytor
- Ändra utseende på växlingsområde
- Ändra kontrollerna i växlingsområde

▼ Ändra standardantalet arbetsytor

- ◆ Ändra följande resurs för Arbetsytehanteraren:

```
Dtwm*workspaceCount :      n
```

Mer information finns i "Ändra antalet systemtäckande arbetsytor" på sidan 254.

▼ Ändra antalet knapprader i växlingsområdet

- ◆ Ändra fältet `NUMBER_OF_ROWS` i `SWITCH` -definitionen.

Följande definition definierar ett treraders växlingsområde.

```
SWITCH Switch
{
  CONTAINER_NAME    rutans namn
```

```
NUMBER_OF_ROWS    3
...
}
```

▼ Ändra eller lägga till kontroller i växlingsområdet

1. Skapa en konfigurationsfil för Frontpanelen med kontrolldefinitionen.

- Ange att kontrollen ska finnas inuti växlingsområdet:

```
CONTAINER_NAME    Switch
CONTAINER_TYPE    SWITCH
```

- Ange positionen i arbetsytväxlaren:

```
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 kontrollens ikon. Storleken bör vara 16 gånger 16 bildpunkter.

Följande kontroll placerar en Terminal-kontroll i arbetsytväxlaren.

```
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 ikon och samma hjälpavsnitt som används av Terminal-kontrollen på panelundermenyn Egna program.

Allmän konfiguration av Frontpanelen

Med Frontpanelens syntax för PANEL 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 direkthjälpen (`man`) för `dtfpfile(4X)`.

Allmänna steg

1. **Skapa en ny konfigurationsfil till Frontpanelen i**
`/etc/dt/appconfig/types/språk eller Hemmapp/.dt/types`.
2. **Kopiera standardbeskrivningen för `PANEL` från**
`/usr/dt/appconfig/nyspråk/dtwm.fp` **till den nya filen.**
3. **Redigera `PANEL`-beskrivningen.**
Den nya `PANEL`-beskrivningen har högre prioritet än standardbeskrivningen.

▼ Ändra 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 till definitionen av `PANEL`:**

```
DISPLAY_CONTROL_LABELS    True
```

2. **Lägg till fältet `LABEL` för varje kontroll.**
Kontrollnamn används om `LABEL` inte anges.

▼ Ändra klickmetod för kontroller

- ◆ **Använd fältet CONTROL_BEHAVIOR i PANEL-definitionen för att ange hur användaren kör PUSH_ACTION. Tänkbara värden är:**
 - single_click - användaren klickar på kontrollen för att köra PUSH_ACTION
 - double_click - användaren dubbelklickar på kontrollen för att köra PUSH_ACTION

▼ Skapa en helt ny frontpanel

När du vill göra mycket omfattande ändringar kan det vara bättre att skapa en helt ny frontpanel.

För att undvika konflikt med Frontpanelens inbyggda komponenter bör en helt ny frontpanel använda nya namn för 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 Arbetsytahanteraren 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 *Hemmapp*/.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_NAME      NewFrontPanel
  POSITION_HINTS       first
  ...
}
```

4. Lägg till rutdefinitioner för den mellersta och undre raden:

```
BOX NewFrontPanelMiddle
{
  CONTAINER_NAME      NewFrontPanel
  POSITION_HINTS       second
}
```

```
BOX NewFrontPanelBottom
{
  CONTAINER_NAME      NewFrontPanel
  POSITION_HINTS       second
}
```

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 arbetsytväxlaren till NewFrontPanelMiddle.

8. Ange resursen:

```
Dtwm*frontPanel*name: NewFrontPanel
```

9. Välj Starta om Arbetsythanteraren på Arbetsyta-menyn.

Anpassa arbetsytehanteraren

I följande avsnitt beskrivs hur du anpassar skrivbordets arbetsytehanterare.

- "Arbetsytehanterarens konfigurationsfiler" på sidan 252
- "Anpassa arbetsytor" på sidan 254
- "Arbetsytehanterarens menyer" på sidan 256
- "Anpassa knappkopplingar" på sidan 259
- "Anpassa tangentkopplingar" på sidan 261
- "Växla mellan de förinställda och egna uppsättningarna" på sidan 263

Arbetsytehanteraren är en fönsterhanterare som finns på skrivbordet. På samma sätt som andra fönsterhanterare hanterar den följande:

- Hur olika delar av fönsterramen visas
- Fönstrets olika visningsinställningar, bl a staplingsordning och fokusering
- Tangentbords- och knappkopplingar
- Visning av minimerade fönster
- Arbetsyta- och Fönster-menyerna

Arbetsytehanteraren hanterar dessutom följande skrivbordskomponenter:

- *Arbetsytor* Arbetsytehanteraren hanterar antalet arbetsytor och håller reda på vilka fönster som är öppna i varje arbetsyta.
- *Arbetsytans bakgrund* Användaren ändrar bakgrund med hjälp av Inställningar. Hantering av bakgrunder är dock en funktion i arbetsytehanteraren.
- *Frontpanelen*. Trots att Frontpanelen har egna konfigurationsfiler, skapas och hanteras den med arbetsytehanteraren.

Flera av dessa komponenter kan ändras med Inställningar. Med Inställningar går det snabbt och lätt att redigera sådant som du behöver ändra ofta. Övriga resurser måste anges manuellt.

Arbetsytehanteraren är `dtwm`. Den baseras på fönsterhanteraren OSF/Motif.

- Om du behöver referensinformation om Arbetsytehanteraren läser du `dtwm(1)` och `dtwmrc(4)` i direkthjälpen.
- Mer information om hur du ställer in Arbetsytehanterarens resurser finns i "Ange programresurser" på sidan 265.
- Mer information om konfigurationsfiler för Frontpanelen finns i Kapitel 15.

Mer information om hur du anger resurser finns i.

Arbetsytehanterarens konfigurationsfiler

Arbetsytehanteraren hämtar information om Fönster-menyerna, arbetsytemenyerna samt knapp- och tangentkopplingar från en konfigurationsfil.

Den använder någon av följande filer:

- Egen fil: *Hemmapp*/`.dt/dtwmrc`
- Egen systemfil: `/etc/dt/config/språk/sys.dtwmrc`
- Inbyggd fil: `/usr/dt/config/språk/sys.dtwmrc`

Arbetsytehanteraren söker efter en konfigurationsfil enligt ovan nämnda ordning och använder den första som den hittar.

För användare som använder fler än ett språk under en session, kan en egen språkberoende konfigurationsfil med namnet *Hemmapp**.`dt.språk/dtwmrc` skapas. Den prioriteras över *Hemmapp*/`.dt/dtwmrc`.

▼ Skapa och ändra en egen konfigurationsfil

Den egna konfigurationsfilen för Arbetsytehanteraren är *Hemmapp*/`.dt/dtwmrc`. Om den här filen finns är det den som används.

1. Dubbelklicka på Redigera Dtwmrc i tillbehörsggruppen Skrivbordsverktyg.

Om det redan finns en egen `dtwmrc`-fil hämtas den till redigeraren. Annars kopieras `sys.dtwmrc` till *Hemmapp*/`.dt/dtwmrc`, som sedan hämtas till redigeraren.

2. Redigera filen.

3. Avsluta redigeraren.

Filen sparas som din egen `dtwmrc`, oavsett ursprungskälla.

▼ Skapa en systemtäckande konfigurationsfil

Den systemtäckande konfigurationsfilen för Arbetsyterhanteraren 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 *Hemmapp*/`.dt/dtwmrc` finns.

▼ Inkludera andra filer

- ◆ **Använd syntaxen:**

```
include
{
    sökväg
    sökväg
    ... }
```

Följande rader inkluderar t ex filen `/users/ellen/mymenu`:

```
include
{
    /users/ellen/mymenu
}
```

Använd programsatserna `Include` när du vill skaffa ytterligare funktionalitet utan att hela konfigurationsfilen behöver kopieras. En användare vill kanske skapa en ny tangentkoppling utan att administrera hela konfigurationsfilen. Användaren skapar då filen *Hemmapp*/`.dt/dtwmrc` med följande innehåll:

```
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"
    ...
}
```

▼ Starta 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 befinner sig över bakgrunden).**

Anpassa arbetsytor

De flesta anpassningar av arbetsytor, t ex ändring av namn och antalet arbetsytor, kan göras av användaren med hjälp av skrivbordets gränssnitt. Arbetsytehanteraren omfattar dock resurser för att ange systemtäckande standardvärden.

▼ Ändra antalet systemtäckande arbetsytor

Standardskrivbordskonfigurationen omfattar fyra arbetsytor. Användaren kan lägga till och ta bort arbetsytor med hjälp av menyerna som hör till arbetsyteomkopplaren.

Filen `/usr/dt/app-defaults/C/Dtwm` ställer in `workspaceCount`-resursen med följande antal arbetsytor:

```
Dtwm*0*workspaceCount: 4
Dtwm*workspaceCount: 1
```

Flera arbetsytor anges på skärm 0. Enstaka arbetsytor anges på övriga skärmar.

Du kan skapa filen (eller redigera om den befintliga) `/etc/dt/config/C/sys.resources` för att ändra det förvalda antalet arbetsytor för alla nya användare på en arbetstation.

- ◆ **Använd resursen `0*workspaceCount` om du vill ställa in den förvalda systemtäckande inställningen på den primära skärmen:**

```
Dtwm*0*workspaceCount: antal
```

Följande resurs ställer in antalet systemtäckande arbetsytor på den primära skärmen till sex:

```
Dtwm*0*workspaceCount: 6
```


Mer information om hur du ställer in resurser i arbetsythanteraren finns i .

Följande resurs ställer in antalet arbetsytor till sex:

```
Dtwm*workspaceCount: 6
```

▼ Namnge systemtäckande arbetsytor

Arbetsytor numreras internt med numreringskonventionen wsn , där n är 0, 1, 2, osv. Standardantalet på fyra arbetsytor numreras internt med $ws0$ till $ws3$.

◆ **Använd resursen `title` om du vill ändra namnet på en angiven arbetsyta:**

```
Dtwm*wsn: namn
```

Mer information om hur du ställer in arbetsythanterarens resurser finns i .

Följande resurser kopplar de fyra förvalda arbetsytorna till de angivna namnen:

```
Dtwm*ws0*title: Anna  
Dtwm*ws1*title: Bo  
Dtwm*ws2*title  Julia  
Dtwm*ws3*title  Emma
```

▼ Skapa ytterligare bakgrunder

1. **Skapa bakgrundsbilderna. De kan vara i bitmapps- eller pixmappsformat.**
2. **Placera bakgrunden i någon av följande mappar. (Du kanske först måste skapa mappen.)**
 - Systemtäckande bakgrunder: `/etc/dt/backdrops`
 - Egna bakgrunder: `Hemmapp/.dt/backdrops`
3. **Välj Starta om arbetsythanteraren på Arbetsyta-menyn.**

Systemtäckande och egna bakgrunder läggs till de inbyggda bakgrunderna i `/usr/dt/backdrops`.

Du kan ersätta en befintlig inbyggd bakgrund genom att skapa en egen eller systemtäckande bakgrund med samma namn.

▼ Ersätta bakgrunden med en grafisk bild

Bakgrunder läggs som skikt ovanpå skärmens rotfönster. I dialogrutan Inställningar - Bakgrund finns inställningen `NoBackdrop` som gör bakgrunden genomskinlig.

Det finns bara ett rotfönster bakom alla arbetsyteteksturer. Därför finns en grafisk bild som har 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.

Bilden måste vara i ett format som det finns ett verktyg till för att den ska kunna visas i rotfönstret. Om du t ex tänker använda `xsetroot`, måste du skapa en bitmapsfil.

2. Skapa den körbara filen *Hemmapp*/.dt/sessions/sessionetc.

Filen `sessionetc` körs varje gång som användaren loggar in.

3. Placera kommandot som visar bilden i filen `sessionetc`.

Följande kommando visar rotfönstret sida vid sida med angiven bitmapsfil:

```
xsetroot -bitmap /users/ellen/.dt/icons/root.bm
```

Arbetsyteteksturerens menyer

Arbetsytetekstureraren har tre standardmenyer:

Arbetsyta-menyn – kallas även rotmenyn. Den visas när du trycker på musknapp 3 med pekaren placerad över bakgrunden. Menyn är kopplad till musknappen med en knappkoppling.

Fönster-menyn – menyn visas när användaren trycker ned musknapp 1 eller musknapp 3 med pekaren på fönstermenyknappen (i övre vänstra hörnet i fönsterramen). Menyn är kopplad till knappen med `windowMenu`-resursen.

Frontpanelmenyn – menyn visas när användaren trycker ned musknapp 1 eller musknapp 3 med pekaren på frontpanelknappen.

Arbetsyteteksturerens menysyntax

Arbetsyteteksturerarens menyer har följande syntax:

```
Menu Menynamn
{
  val1 [snabbtang] [kortkommando] funktion [argument]
  val2 [snabbtang] [kortkommando] funktion [argument]
  ...
}
```

```
}
```

där:

val är den text eller bitmapp som visas på menyn. Om texten innehåller blanksteg innesluter du texten med citattecken. För bitmappar använder du syntaxen *@/sökväg*.

snabbtangenter är ett tecken som fungerar som en tangentbordsgenväg när menyn visas. Det anges som *_tecken*.

kortkommando – en tangentbordsgenväg som är aktiverad även om inte menyn visas. Kortkommandon har syntaxen *styrtingent<Key> Tangentnamn*, där styrtingenten är Ctrl+_ , Shift, Alt (Extend char) eller Lock. Om du vill ha en lista över alla möjliga tangentnamn undersöker du filen *keysymdef.h* i din "X11 include"-mapp.

funktion – den funktion som ska utföras när markeringen görs. Se sidan (4) i *dtwmrc* om du vill se en lista med funktioner.

argument – funktionsargument. Se sidan (4) i *dtwmrc* för mer utförlig information.

Följande objekt med etiketten Återställ återställer fönstret. När menyn visas kan du återställa fönstret genom att trycka på "R". Du kan även återställa fönstret genom att trycka på Extend char F5.

```
Restore _R Alt<Key> F5 f.normalize
```

Obs! Fullständig information om Arbetsyتهanternrens menysyntax hittar du i *dtwmrc(4)* i direkt hjälpen.

▼ Anpassa Arbetsyta-menyn

1. Öppna lämplig fil för redigering:

- Egen: *Hemmapp/.dt/dtwmrc*
- Systemtäckande: */etc/dt/config/språk/sys.dtwmrc*
Mer information om hur du skapar de här filerna finns i "Arbetsyتهanternrens konfigurationsfiler" på sidan 252.

2. Redigera beskrivningen av Arbetsyta-menyn.

Den förvalda Arbetsyta-menyn heter *DtRootMenu*.

```
Menu DtRootMenu
{
    "Workspace Menu"           f.title
    "Shuffle Up"              f.circle_up
    "Shuffle Down"           f.circle_down
    ...
}
```

▼ Skapa en ny Arbetsyta-meny

1. Öppna lämplig fil för redigering:

- Egen: *Hemmapp*/.dt/dtwmrc
- Systemtäckande: /etc/dt/config/*språk*/sys.dtwmrc

Mer information om hur du skapar de här filerna finns i "Arbetsytehanterarens konfigurationsfiler" på sidan 252.

2. Skapa den nya menyn:

```
Menu menynamn
{
    ...
}
```

Se "Arbetsytehanterarens menysyntax" på sidan 256.

3. Skapa eller redigera knappkopplingarna för att visa den nya menyn.

Om menyn ersätter den befintliga menyn redigerar du knappkopplingen som visar Arbetsyta-menyn.

```
<Btn3Down> root f.menu menynamn
```

Om menyn är en tilläggsmeny skapar du en ny musknappskoppling. Den följande knappkopplingen visar menyn när Skift och musknapp 3 hålls ned över bakgrunden:

```
Shift<Btn3Down> root f.menu menynamn
```

4. Välj Starta om arbetsytehanteraren på Arbetsyta-menyn.

▼ Skapa en ny Fönster-meny

Obs! Fönster-menyn är inbyggd i Arbetsytehanteraren och anpassas normalt inte. Om du vill att fönstrens uppträdande i de olika programmen ska vara konstant bör du inte ändra inställningarna på fönster-menyn.

1. Öppna en lämplig fil för redigering:

- Egen: *Hemmapp*/.dt/dtwmrc
- Systemtäckande: /etc/dt/config/*språk*/sys.dtwmrc

Mer information om hur du skapar de här filerna finns i "Arbetsyterhanterarens konfigurationsfiler" på sidan 252.

2. Skapa den nya menyn:

```
Menu menynamn
{
    ...
}
```

3. Använd resursen `windowMenu` för att ange den nya menyn:

```
DtWm*windowMenu: menynamn
```

4. Välj Starta om arbetsyterhanteraren på Arbetsyta-menyn.

Anpassa knappkopplingar

En *knappkoppling* kopplar en musknapp, eventuellt tillsammans med en ändringstangent från tangentbordet, till en funktion i Fönsterhanteraren. Knappkopplingar gäller i alla arbetsytor.

Skrivbordets standardknappkopplingar definieras i arbetsytans konfigurationsfil i en knappkopplingsuppsättning med namnet `DtButtonBindings`:

```
Buttons DtButtonBindings
{
    ...
}
```

Syntax för knappkoppling

Syntaxen för knappkoppling är:

```
Buttons ButtonBindingSetName
{
    [modifier]<button_nameMouse_action> sammanhang funktion [argument]
    [modifier]<button_nameMouse_action> sammanhang funktion [argument]
```

där:

button_name – Btn1 är Vänster musknapp Btn2 är Mittenknappen (3-knapps-mus) eller båda knapparna (2-knapps-mus) Btn3 – höger musknapp Btn4 är Knapparna 1 och 2 tillsammans på en 3-knapps-mus Btn5 är Knapparna 2 och 3 tillsammans på en 3-knapps-mus

styr tangent – Ctrl, Shift, Alt, Lock

mouse_action – Down – Hålla ned en mus; knapp Up – Släppa en musknapp; Click – Trycka och släppa en musknapp; Click2 – Dubbelklicka med en musknapp; Drag – Dra musen medan du håller ned musknappen

sammanhang – indikerar pekarens nödvändiga placering vid koppling. Dela upp innehållet med "|" tecknet.

root – arbetsyfefönstret; *window* – Klientfönster eller fönsterram; *frame* – fönsterram, exkludera innehållet; *icon* – Ikon; *-title* – Titelfältet; *app* – Klientfönster (utan ram)

funktion – en av fönsterhanterarens funktioner. I `dtwmrc(4)` i direkthjälpen hittar du en lista med giltiga funktioner.

argument – de funktionsargument i Fönsterhanteraren som krävs. I `dtwmrc(4)` i direkthjälpen hittar du ytterligare detaljer.

Följande rad visar menyn som beskrivs i `DtRootMenu` när musknapp 3 trycks ned och pekaren är i arbetsyfefönstret (men inte i ett klientfönster).

```
<Btn3Down>      root      f.menu      DtRootMenu
```

Obs! Fullständig information om syntax för knappkopplingar hittar du i `dtwmrc(4)` i direkthjälpen.

▼ Lägga till knappkopplingar

1. Öppna lämplig fil för redigering:

- Egen: *Hemmapp*/.dt/dtwmrc
- Systemtäckande: /etc/dt/config/*språk*/sys.dtwmrc

Mer information om hur du skapar de här filerna finns i "Arbetsyfehänterarens konfigurationsfiler" på sidan 252.

2. Lägga till knappkopplingen i `DtButtonBindings`-definitionen

Koppla inte samma knapp till olika funktioner och koppla inte fler än en funktion till samma knapp eller sammanhang.

3. Välj Starta om arbetsyfehänteraren på Arbetsyta-menyn.

▼ Skapa nya knappkopplingsuppsättningar

1. Öppna lämplig fil för redigering:

- Egen: *Hemmapp*/.dt/dtwmrc

- Systemtäckande: `/etc/dt/config/språk/sys.dtmrc`

Information om hur du skapar de här filerna finns i "Arbetsyterhanterarens konfigurationsfiler" på sidan 252.

2. **Skapa en ny knappkopplingsuppsättning. Se "Syntax för knappkoppling" på sidan 259.**

3. **Ställ in `buttonBindings`-resursen till det nya namnet:**

```
Dtwm*buttonBindings: ButtonBindingsSetName
```

4. **Välj Starta om arbetsyterhanteraren på Arbetsyta-menyn.**

Obs! De nya knappkopplingarna ersätter de befintliga. Kopiera de knappkopplingar som du vill behålla från `DtButtonBindings`.

Anpassa tangentkopplingar

En *tangentkoppling* kopplar olika kombinationer av tangentnedtryckningar till funktioner i Arbetsyterhanteraren. Tangentkopplingar gäller för alla arbetsytor.

Obs! Använd inte en vanlig tangentkombination som tangentkoppling. Om du kopplar Skift-A till en funktion förlorar du möjligheten att skriva "A".

Förinställda tangentkopplingar för skrivbordet

De förinställda tangentkopplingarna anges i konfigurationsfilen för Arbetsyterhanteraren i en tangentkopplingsuppsättning som heter `DtKeyBindings`:

```
Keys DtKeyBindings
{
  ...
}
```

Syntax för tangentkopplingar

Syntaxen för tangentkopplingar är:

```

Keys KeyBindingSetName
{
  [Styrtangenter]<Key>key_name context function [argument]
  [Styrtangenter]<Key>key_name context function [argument]
  ...
}

```

där:

Styrtangenter är `Ctrl`, `Shift`, `Alt` och `Lock`. Flera styrtangenter är tillåtna, dela av dem med blanksteg.

tangent_namn är tangenten som funktionen kopplas till. Tangenter med bokstäver eller siffror har oftast sitt *tangent_namn* skrivet på tangenten. Tangenten "a" heter till exempel "a" och "2"-tangenten heter "2". "Tab"-tangenten heter "Tab". "F3"-tangenten heter "F3".

För övriga tangenter skrivs namnet oftast ut, till exempel `plus` för "+"-tangenten. Filen `keysymdef.h`, som finns i en systemberoende mapp, innehåller mer information om tangentnamn.

sammanhang – det element som måste vara kopplat till ett tangentbordsfokus för att operationen ska fungera. De kan slås ihop om kopplingen ska gälla i mer än ett sammanhang. Olika sammanhang delas upp med "|" tecknet.

root – Arbetsytebakgrund; *window* – Klientfönster; *icon* – Ikon

funktion – en funktion i Fönsterhanteraren. I `dtwmrc(4)` i direkthjälpen hittar du en lista med giltiga funktioner.

argument – de funktionsargument i fönsterhanteraren som krävs för operationen. Se `dtwmrc(4)` i direkthjälpen för ytterligare detaljer.

Följande tangentkoppling gör att användaren kan byta tangentbordsfokus till nästa transienta fönster i ett program genom att trycka `Alt+F6`.

```
Alt<Key>F6 window f.next_key transient
```

Obs! Om du behöver mer information om syntax för tangentkopplingar läser du `dtwmrc(4)` i direkthjälpen.

▼ Skapa en anpassad tangentkopplingsuppsättning

1. Öppna en lämplig fil för redigering:

- Egen: *Hemmapp*/`.dt/dtwmrc`
- Systemtäckande: `/etc/dt/config/språk/sys.dtwmrc`

Mer information om hur du skapar de här filerna finns i "Arbetsyterhanterarens konfigurationsfiler" på sidan 252.

2. **Skapa en ny tangentkopplingsuppsättning med en unik *KeyBindingSetName*.**
Använd skrivbordets förinställda tangentkopplingsuppsättning, *DtKeyBindings*, som mall.

3. **Ställ in *keyBindings*-resursen med det nya namnet:**

`Dtwm*keyBindings: KeyBindingSetName`

4. **Välj Starta om arbetsyterhanteraren på Arbetsyta-menyn.**

Obs! De nya tangentkopplingarna ersätter de befintliga tangentkopplingarna. Kopiera de tangentkopplingar som du vill behålla från *DtKeyBindings* till din nya uppsättning.

Växla mellan de förinställda och egna uppsättningarna

Växla mellan den förinställda uppsättningen OSF/Motif 1.2 och uppsättningen CDE för skrivbordsfönster:

1. **Tryck på Alt+Shift+Ctrl+!**

2. **Klicka på OK i dialogrutan.**

Om du växlar till den förinställda metoden tas Frontpanelen samt de anpassade tangent- och knappkopplingarna bort.

Hantera programresurser, teckensnitt och färger

Du kan välja mellan ett flertal färger och teckensnitt för skärmen med hjälp av Inställningar eller genom att anpassa ytterligare teckensnitts- och färgresurser. I det här kapitlet beskrivs hur du anpassar teckensnitt och färgresurser.

I följande avsnitt beskrivs också hur du anger stildefinitioner för DtEditor-bildmodulsprogram, t ex skrivbordets textredigerare (dtpad) och Elektronisk post (Ctrl+S) samt hur du ändrar snabbval för DtEditor-bildmodulsprogram som krockar med dessa definitioner.

- "Ange programresurser" på sidan 265
- "Definiera UNIX-kopplingar" på sidan 266
- "Hantera teckensnitt" på sidan 271
- "Hantera färger" på sidan 276
- "Ange skuggtjocklek för programfönster" på sidan 284

Ange programresurser

Resurser används av program för att ange vissa aspekter i utseende och metod. Inställningen (dtstyle) tillhandahåller exempelvis resurser som gör att du kan ange var systemet ska leta efter filer som innehåller information om färgpaletter:

```
dtstyle*paletteDirectories: /usr/dt/palettes/C \ Hemmapp/.dt/palettes
```

Programstandardfiler för skrivbordsprogram finns i mappen /usr/dt/app-defaults/*språk*.

▼ Ange systemtäckande resurser

- ◆ **Lägg till resurserna i filen `/etc/dt/config/språk/sys.resources`. (Du kan behöva skapa filen.)**

Exempelvis i `/etc/dt/config/C/sys.resources` anger du värdet:

```
AnApplication*resource
```

Då anges resursen `AnApplication*resource` i varje användares `RESOURCE_MANAGER`-egenskap vid nästa inloggning.

▼ Anger egna resurser

1. **Lägg till resurserna i filen *Hemmapp*/`.Xdefaults`.**
2. **Dubbelklicka på Ladda om resurser i programgruppen Skrivbordsverktyg.**

Hur skrivbordet startar resurser

Resurser laddas när sessionen startas av Sessionshanteraren. Mer information om hur Sessionshanteraren laddar resurserna i `RESOURCE_MANAGER` finns i "Ladda sessionsresurser" på sidan 27.

Definiera UNIX-kopplingar

UNIX-kopplingar är normalt inte aktiverade.

▼ Ange EMACS-stildefinitioner

Följande procedur:

- Anger EMACS-stildefinitioner för bildelement av typen `DtEditor`, t ex skrivbordets `Textredigerare` och (`dtpad`) och Elektronisk post (`Ctrl+S`)
- Ändrar snabbval för `DtEditor`-bildelement som krockar med dessa definitioner.

1. **Lägg till följande rad i filen *Hemmapp*/`.Xdefaults`:**

```
#include "/usr/dt/app-defaults/språk/UNIXbindings"
```

där *språk* är värdet på miljövariabeln `LANG`.

2. **Starta om sessionen.**

▼ Ändra EMACS-stildefinitioner

1. **Infoga innehållet i filen** `/usr/dt/app-defaults/språk/UNIXbindings i Hemmapp/.Xdefaults.`
2. **Redigera kopplingarna i filen** `.Xdefaults.`
3. **Starta om sessionen när du är klar.**

UNIX-kopplingar i filen UNIXbindings

Filen `/usr/dt/app-defaults/språk/UNIXbindings` tillhandahåller de kopplingar som beskrivs i tabellen nedan.

Obs! Tangenten Delete tar bort föregående tecken när UNIX-kopplingar aktiveras och Skift-Delete tar bort nästa tecken.

Tabell 17-1 listar de dtpad-menysnabbval och den snabbvalstext som krockar med UNIX-kopplingarna.

TABELL 17-1 dtpad Åsidosätter

Snabbval för menyer och snabbvalstext	Åsidosätter
<code>Dtpad*fileMenu.print.acceleratorText:</code>	
<code>Dtpad*fileMenu.print.accelerator:</code>	
<code>Dtpad*editMenu.undo.acceleratorText:</code>	Ctrl+_
<code>Dtpad*editMenu.undo.accelerator:</code>	Ctrl<Key>_
<code>Dtpad*editMenu.paste.acceleratorText:</code>	Skift+Insert
<code>Dtpad*editMenu.paste.accelerator:</code>	Skift<Key>osffInsert
<code>Dtpad*editMenu.findChange.acceleratorText:</code>	Ctrl+S
<code>Dtpad*editMenu.findChange.snabbkommando:</code>	Ctrl<Key>s

Tabell 17-2 listar vilka snabbkommandon skrivfönstret dtmail åsidosätter och vilken snabbvalstext som är oförenlig med UNIX-kopplingarna.

TABELL 17-2 dtmail Vad skrivfönstret åsidosätter

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 ctrl-kopplingar enligt GNU EMACS-formatet samt några ytterligare kopplingar. När det behövs kan också tangenten Skift användas i kombination med den normala kopplingen för att skapa en motsatt effekt. Exempelvis flyttar Ctrl+Skift+F markören bakåt ett tecken eftersom Ctrl+F normalt flyttar den framåt ett tecken.

Ytterligare kopplingar:

Ctrl+komma – backward-word

Ctrl+Skift+komma – forward-word

Ctrl+punkt – forward-word

Ctrl+Skift+punkt – backward-word

Ctrl+Retur – end-of-file

Ctrl+Skift+Retur – beginning-of-file

GNU EMACS kopplar delete-previous-character() snarare än delete-next-character() till tangenten Delete. Meta+F är vanligen kortkommando för Fil-menyn, så kopplingen till forward-word() kommer att ignoreras. Använd någon av de andra kopplingarna för forward-word (exempelvis Ctrl+punkt).

Tabell 17-3 listar översättningarna för DtEditor.text

TABELL 17-3 DtEditor.text Översättningar

Ändringstangent Tangent Åtgärd		
c ~s	<Key>a:	beginning-of-line()\n\
c s	<Key>a:	end-of-line()\n\
c ~s	<Key>b:	backward-character()\n\
c s	<Key>b:	forward-character()\n\
c ~s	<Key>b:	backward-character()\n\
c s	<Key>b:	backward-word()\n\
m ~s	<Key>b:	backward-word()\n\
m s	<Key>b:	forward-word()\n\
c ~s	<Key>d:	delete-next-character()\n\
c s	<Key>d:	delete-previous-character()\n\
m ~s	<Key>d:	kill-next-word()\n\
m s	<Key>d:	kill-previous-word()\n\
c ~s	<Key>e:	end-of-line()\n\
c s	<Key>e:	beginning-of-line()\n\
c ~s	<Key>f:	forward-character()\n\
c s	<Key>f:	backward-character()\n\
m ~s	<Key>f:	forward-word()\n\
m s	<Key>f:	backward-word()\n\
c	<Key>j:	newline-and-indent()\n\
c ~s	<Key>k:	kill-to-end-of-line()\n\
c s	<Key>k:	kill-to-start-of-line()\n\
c	<Key>l:	redraw-display()\n\
c	<Key>m:	newline()\n\

Ändringstangent Tangent Åtgärd

c s	<Key>n:	process-up()\n\
c ~s	<Key>n:	process-down()\n\
c	<Key>o:	newline-and-backup()\n\
c ~s	<Key>p:	process-up()\n\
c s	<Key>p:	process-down()\n\
c ~s	<Key>u:	kill-to-start-of-line()\n\
c s	<Key>u:	kill-to-end-of-line()\n\
c ~s	<Key>v:	next-page()\n\
c s	<Key>v:	previous-page()\n\
m ~s	<Key>v:	previous-page()\n\
m s	<Key>v:	next-page()\n\
c	<Key>w:	kill-selection()\n\
c ~s	<Key>y:	unkill()\n\
m	<Key>]:	forward-paragraph()\n\
m	<Key>[:	backward-paragraph()\n\
c ~s	<Key>comma:	backward-word()\n\
c s	<Key>comma:	forward-word()\n\
m	<Key>\\<:	beginning-of-file()\n\
c ~s	<Key>period:	forward-word()\n\
c s	<Key>period:	backward-word()\n\
m	<Key>\\>:	end-of-file()\n\
c ~s	<Key>Return:	end-of-file()\n\
c s	<Key>Return:	beginning-of-file()\n\

Ändringstangent Tangent Åtgärd		
~c ~s ~m ~a	<Key>osfDelete:	delete-previous-character()\n\
~c s ~m ~a	<Key>osfDelete:	delete-next-character()

Hantera teckensnitt

Med dialogrutan Teckensnitt i Inställningar kan du välja den teckensnittsgrupp och teckensnittsstorlek du vill ha för alla program. Du kan också ange teckensnitt på kommandoraden eller använda resurser för att:

- Ange teckensnittsresurser för enstaka program
- Tilldela de olika teckensnitt som ska användas av dialogrutan Teckensnitt

Ett *teckensnitt* är det typsnitt och storlek som tecken skrivs och visas med. Skrivbordet omfattar en mängd teckensnitt med olika stil och storlek.

En *bitmapsteckensnitt* skapas från en punktmatris. (Inställningar konfigurerar normalt bara bitmapsteckensnitt.) Teckensnitten är begränsade till en fil. Många filer behöver en fullständig uppsättning storlekar, lutningar och tjocklekar.

Teckensnitt anges som värden på resurser och parametrar till kommandon. Namnet X Logical Font Description (XLFD) är den metod som ett teckensnitt efterfrågas med. Systemet hittar det teckensnitt som bäst motsvarar den angivna beskrivningen.

Ange teckensnittsresurser för skrivbordet

Med dialogrutan Teckensnitt i Inställningar kan du markera teckensnitt (upp till sju storlekar) för textposter och etiketter. Du kan också lägga till eller ta bort teckensnittsgrupper.

Resurser som anges i dialogrutan Teckensnitt

När ett teckensnitt markeras skrivs följande resurser till egenskapen RESOURCE_MANAGER:

- `SystemFont` används för systemområden, t ex menyrader, menyrutor, tryckknappar, växlingsknappar och etiketter. Följande resurs anges av `SystemFont`:
 - *`FontList` – visas i systemområden hos skrivbordsklienter och andra klienter som är skapade med verktygslådan OSF/Motif.
- `UserFont` används för text som skrivs i fönster. Följande resurser anges av `UserFont`:
 - *`Font` – stöder tidigare versioner av X-program
 - *`FontSet` – den primära inställningen
 - *`XmText*FontList` – visas i textinmatningsrutor
 - *`XmTextField*FontList` – visas i textinmatningsrutor

Resurser som används i dialogrutan Teckensnitt

Teckensnitt som används för varje val i dialogrutan Teckensnitt anges i resursfilen `/usr/dt/app-defaults/Dtstyle`. Upp till sju storlekar kan anges.

`NumFonts` – antal teckensnittstorlekar i dialogrutan Teckensnitt

`SystemFont[1-7]` – upp till sju resurser som tilldelar ett specifikt teckensnitt till en markering i dialogrutan Teckensnitt för `SystemFont`

`UserFont[1-7]` – upp till sju resurser som tilldelar ett specifikt teckensnitt till en markering i dialogrutan Teckensnitt för `UserFont`

Obs! Standardteckensnitten för dessa resurser har valts för läsbarheten på olika skärmar. Om du vill använda ett visst teckensnitt i ett program, anger du det med programmets teckensnittsresurs i stället för att ändra skrivbordets teckensnitt.

Du får mer information om programteckensnitt i kommandobeskrivningarna till `DtStdAppFontNames(5)` och `DtStdInterfaceFontNames(5)`.

▼ Visa tillgängliga teckensnitt

1. Skriv följande:

```
xlsfonts [alternativ] [-fn mönster]
```

En lista över XLFD-namn och teckensnittalias som är tillgängliga i systemet visas. Bitmappsteckensnitt 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 teckensnitt 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 teckensnittnamn som börjar på `dt`, innehåller inte teckensnittsökvägen skrivbordsteckensnitt. Skriv följande kommando för att få med skrivbordsteckensnitten bland de tillgängliga teckensnitten:

```
xset +fp mappnamn
```

där *mappnamn* är mappen som innehåller skrivbordsteckensnitten. Standardpositionen som anges vid sessionsstarten är `/usr/dt/config/xfonts/språk`.

Ytterligare information hittar du i:

- Kommandobeskrivningarna till `xset` och `xlsfonts`, som listar tillgängliga alternativ.
- *Using the X Window System*, som beskriver teckensnittalias och `xset`-klienten.

▼ Ange teckensnitt på kommandoraden

- ◆ Använd kommandoradsalternativet `--xrm` för att ange en teckensnittsresurs för en viss klient. Exempel:

```
programnamn -xrm "*bitstream-charter-medium-r-normal-8-88-75-75-p-45-iso8859-1"
```

X Logical Font Description (XLFD)

Ett teckensnitt 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 ett jokertecken (*), och ett tecken i en egenskap kan ersättas av jokertecknet ?. Tabell 17-4 listar specifikationer för teckensnittegenskapssträngar.

Formatet på egenskapssträngarnas specifikationer är:

```
"-Upphovsman-Familj-Tjocklek-Lutning-AngeBredd-LäggTillStil-Bildpunktstorlek-  
Punktstorlek-UpplösningX-UpplösningY--UpplösningY-  
Snittbredd-Teckenregister-Teckenkodning"
```

TABELL 17-4 Specifikationer till teckensnittens egenskapssträngar

Egenskaps- sträng	Beskrivning
<i>Upphovsman</i>	En sträng som anger vem som har utformat teckensnitten
<i>Familj</i>	En sträng som anger varumärkesnamnet på teckensnitten
<i>Tjocklek</i>	En sträng som anger relativ tjocklek på teckensnitten, t ex fetstil
<i>Lutning</i>	En kod som beskriver riktning på lutningen R (Romersk – ingen lutning) I (Kursiv – högerlutning) O (Snedställd – högerlutning) RI (Omvänd kursiv – vänsterlutning) RO (Omvänd snedställd – vänsterlutning)
<i>AngeBredd</i>	En sträng som beskriver bredd, t ex komprimerad eller utökad
<i>LäggTillStil</i>	En sträng som innehåller eventuell ytterligare information som behövs för unik identifiering av teckensnitten
<i>Bildpunktstorlek</i>	Ett heltal som anger storleken på en så kallad em-fyrkant i bildpunkter
<i>Punktstorlek</i>	Ett heltal som anger storleken på en så kallad em-fyrkant i decipunkter
<i>UpplösningX</i>	Ett heltal som anger vågrät upplösning i bildpunkter
<i>UpplösningY</i>	Ett heltal som anger lodrät upplösning i bildpunkter
<i>Avstånd</i>	En kod som anger mellanrum mellan enheter M (Fast teckenindelning–fast breddsteg) P (Proportionellt breddsteg–proportionellt breddsteg) C (Teckencell)
<i>Snittbredd</i>	Ett heltal som anger snittbredden i 1/10 bildpunkter
<i>Teckenregister</i>	En sträng som anger den myndighet som har auktoriserat teckensnittets kodning
<i>Teckenkodning</i>	En sträng som anger registrets teckenuppsättning

Exempel

Följande XLFD-namn beskriver ett teckensnitt med namnet charter som har skapats av Bitstream och stöder standardkod ISO8859-1:

```
-bitstream-charter-medium-r-normal--8-80-75-75-p-45-iso8859-1
```

Teckensnittet har normal tjocklek utan någon speciell lutning och normal bredd. Teckensnittet är proportionellt med en 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 det första teckensnitt som påträffas som matchar de delar du anger.

Om du bara vill ha ett åtta bildpunkters charter-teckensnitt använder du:

```
*-charter-*-*-*-*8-*
```

Visa markerade teckensnittgruppsattribut

Genom att välja knappen Attribut i dialogrutan Teckensnitt i Inställningar, kan du visa följande teckensnittgruppsattribut:

- Teckensnittgrupp
- Storlek
- Alias
- Alias XLFD
- Aliasadress
- Teckensnitt
- Font XLFD

Lagring av användares filsystem för teckensnittgrupper

När en användare lägger till en teckensnittgrupp lagras den på följande adress:

```
Hemmapp/.dt/sdtfonts/värddator/språkområde/teckensnitt-nnnnnn
```

värddator är den lokala arbetsstationens värddatornamn.

språkområde är användarens aktuella språkområde som t ex "C" eller "ja".

teckensnitt-nnnnnn är ett namn som fås från de markerade teckensnitten och ett unikt genererat tal.

Den här teckensnittgruppsmappen innehåller följande tre filer

- `fonts.alias`
- `fonts.dir`
- `sdtfonts.group`

Filerna `fonts.alias` och `fonts.dir` är normala X11-teckensnittfiler som kan läggas till i X-serverns teckensnittsökväg. Filen `sdtfonts.group` innehåller det användardefinierade teckensnittgruppsnamnet.

Skapa teckensnittgrupp för systemadministratör

För att andra användare ska kunna komma åt teckensnittgrupper på en arbetsstation kan systemadministratören kopiera teckensnittgrupper till mappen

`/etc/dt/sdtfonts/språkområde` eller till mappen

`/usr/openwin/lib/X11/stdfonts/språkområde`. Sessionshanteraren letar först i

Hemmapp/`.dt/stdfonts/värddator/språkområde`, därefter i

`/etc/dt/sdtfonts/språkområde` och till sist i

`/usr/openwin/lib/X11/stdfonts/språkområde`.

Hantera färger

I det här avsnittet beskrivs:

- Hur du anger skärmfärger i Inställningar
- Resurser som används av Inställningar för att kontrollera hur skrivbordet använder färger

Färgpaletter

En palett består av en grupp färguppsättningar. Färguppsättningen för den aktuella paletten visas i dialogrutan Inställningar - Färg.

För varje palett finns en fil. Resursen `paletteDirectories` anger mapparna som innehåller palettfiler. Normalt innehåller den här resursen:

- Inbyggda paletter: `/usr/dt/palettes`
- Systemtäckande paletter: `/etc/dt/palettes`
- Egna paletter: *Hemmapp*/`.dt/palettes`

Färguppsättningar

Varje färguppsättning i aktuell palett representeras av en färgknapp i dialogrutan Inställningar - Färg. Varje färg identifieras av ett färguppsättnings-ID (ett tal mellan 1 och 8).



Figur 17-1 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 - förgrunden i ett programfönster eller en fönsterram. Den är alltid svart eller vit. Den används ofta för text i fönster och rubriker.

background - bakgrunden i ett programfönster eller en fönsterram.

topShadowColor - färgen på övre och vänstra ytan för programkontroller (t ex tryckknappar) och fönsterramar.

bottomShadowColor - färgen på nedre och högra ytan för programkontroller och fönsterramar.

`selectColor` – färgen som anger aktivt läge för vissa kontroller, t ex aktiva växlar och knappar.

Antalet färguppsättningar som används av varje palett bestäms av resursen `colorUse`, som användaren kan ange med hjälp av dialogrutan `Antal färger i Inställningar`.

Ange färgvärden

Inställningar använder RGB-värden när information skrivs till palettfilerna. Syntaxen för RGB-tal är:

```
#RödGrönBlå
```

Röd, *Grön* och *Blå* är hexadecimala tal som vart och ett 1 till 4 siffror långt. De 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  
#ffffffffffff
```

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 tilldelar färguppsättningar till olika skärmelement via resurser. Tilldelningarna visas i Tabell 17-5.

TABELL 17-5 Färguppsättningar som tilldelats resurser

Resurs	Visningselement
<code>activeColorSetId</code>	Ramfärg för det aktiva fönstret
<code>inactiveColorSetId</code>	Ramfärg för inaktivt fönster
<code>textColorSetId</code>	Områden för textinmatning

TABELL P-5 Färguppsättningar som tilldelats resurser (forts.)

Resurs	Visningselement
primaryColorSetId	Huvudbakgrunder för program
secondaryColorSetId	Menyfält, menyer och dialogrutor för program

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 Inställningar.

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 Inställningar:

Tabell 17-6 visar färguppsättnings-ID för många färger (8 färguppsättningar) – Inställningen Fler skrivbordsfärger i Inställningar.

TABELL 17-6 Fler skrivbordsfärger

Färguppsättnings-ID	Visningselement
1	Ramfärg för det aktiva fönstret.
2	Ramfärg för inaktiva fönster
3	Oanvända (standard)
4	Ytor för inmatning av text
5	Huvudbakgrunder för program
6	Menyfält, menyer och dialogrutor för program
7	Oanvänd som standard
8	Frontpanelens bakgrund

Tabell 17-7 visar färguppsättnings-ID för normalt antal färger (4 färguppsättningar) – Inställningen Fler programfärger i Inställningar.

TABELL 17-7 Fler programfärger

Färguppsättnings-ID	Visningsselement
1	Ramfärg för det aktiva fönstret.
2	Ramfärg för inaktiva fönster
3	Bakgrundsfärg för program och frontpanel
4	Ytor för textinmatning

Tabell 17-8 visar färguppsättnings-ID för få färger (2 färguppsättningar) – Inställningen Ännu fler programfärger i Inställningar.

TABELL 17-8 Ännu fler programfärger

Färguppsättnings-ID	Standardvärdet på resursen
1	Aktiv fönsterram, knappar för markering av arbetsyta
2	Alla andra skärmelement

Kontrollera färg med Inställningar

Du kan dynamiskt ändra färg på skrivbordsprogram och andra samverkande program via Inställningar. De förgrunds- och bakgrundsfärger som anges med Inställningar är även tillgängliga för icke samverkande program.

För att klienten ska reagera på färgändringar i Inställningar, måste den använda skrivbordsbiblioteket Motif. Klienter som skrivs med andra verktyg kan inte ändra färg dynamiskt efter ändringar i Inställningar. 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 som är inbyggda i programmet.

Klienter kan ange resurserna `primaryColorSetId` och `secondaryColorSetId` för att använda vissa färger i en skrivbordspalette.

Antal färger som används i Inställningar

Antalet färger som används i Inställningar är beroende av värdena på följande resurser:

`colorUse` – konfigurerar antalet färger som skrivbordet använder

`shadowPixmaps` – instruerar skrivbordet att ersätta de två skuggningsfärgerna med bildpunktsbeskrivningar

`foregroundColor` – anger om förgrundsfärgen ska ändras dynamiskt

`dynamicColor` – bestämmer om program ska ändra färg när du växlar paletter

Tabell 17-9 listar det högsta tillåtna antal färger som kan tilldelas av skrivbordet.

TABELL 17-9 Antal skrivbordsfärger

Skärm	Maximalt antal färger	Antalet är härlett från
B_W	2	Svart och vitt
LOW_COLOR	12	Två färguppsättningar gångar fem färger plus svart och vitt
MEDIUM_COLOR	22	Fyra färguppsättningar gångar fem färger plus svart och vitt
HIGH_COLOR	42	Åtta färguppsättningar gångar fem färger plus svart och vitt.

Avgöra maximalt antal färger

1. **Multiplitera antalet färguppsättningar i paletten med antalet färger i varje färguppsättning.**

2. Lägg till 2 (för svart och vitt).

Med följande konfiguration har du emellertid bara tio färger i paletten: fyra färguppsättningar gånger två färger i varje uppsättning: (background och selectColor) plus svart och vitt:

```
*colorUse MEDIUM_COLOR
*shadowPixmap: True
*foregroundColor: White
```

Obs! Flerfärgade ikoner 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 flerfärgade ikoner.

Värde	Beskrivning
B_W	"Svartvit", Inställningar, Visas med 1 till 3 färgplan, Antal färguppsättningar: 2, Maximalt antal färger: 2, Standardantal färger: 2, Inga flerfärgade ikoner
LOW_COLOR	"Flest programfärger", Inställningar, Visas med 4 till 5 färgplan, Antal färguppsättningar: 2, Maximalt antal färger: 12, Standardantal färger: 12, Inga flerfärgade ikoner
MEDIUM_COLOR	"Fler programfärger", Inställningar, Visas med 6 färgplan, Antal färguppsättningar: 4, Maximalt antal färger: 22, Standardantal färger: 22, Flerfärgade ikoner
HIGH_COLOR	"Fler skrivbordsfärger", Inställningar, Visas med 7 eller fler färgplan, Antal färguppsättningar: 8, Maximalt antal färger: 42, Standardantal färger: 42, Flerfärgade ikoner
standard	Skrivbordet väljer korrekt värde för den skärmen. (Om du vill minska antalet färger som används av skrivbordet för skärmar med många färger anger du standardresursen <code>colorUse</code> till <code>MEDIUM_COLOR</code> .)

Resursen shadowPixmap

Resursen `shadowPixmap` gör så att skrivbordet ersätter 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 <code>topShadowPixmap</code> och <code>bottomShadowPixmap</code> att använda i stället för skuggningsfärgerna.
False	<code>topShadowColor</code> och <code>bottomShadowColor</code> från paletten används.

Standardvärdet för `shadowPixmap`s beror på din `colorUse`-resurs och på maskinvarustödet för skärmen.

Resursen `foregroundColor`

Resursen `foregroundColor` anger hur förgrunden är konfigurerad i en palett.

Inställning	Resultat
White	Förgrunden anges till vit.
Black	Förgrunden anges till svart.
Dynamic (Standard)	Förgrunden anges dynamiskt till svart eller vit, beroende på värdet för <code>background</code> . Vita tecken på en gul bakgrund är svåra att läsa så systemet väljer då automatiskt alternativet <code>black</code> .

Om `foregroundColor` anges till antingen `Black` eller `White`, minskas antalet färger i färguppsättningen med en, och förgrunden ändras inte när bakgrundsfärgen ändras.

Standardvärdet för `foregroundColor` är `Dynamic`, utom när värdet för `colorUse` är `B_W`.

Resursen `dynamicColor`

Resursen `dynamicColor` kontrollerar om program ändrar färg dynamiskt, dvs om klienter ändrar färg när du växlar palett.

Värde	Beskrivning
True	Klienter ändrar färg dynamiskt när en ny palett markeras. Det här är standardvärdet.
False	Klienter ändrar inte färg dynamiskt. När en ny palett markeras använder klienter de nya färgerna när sessionen startas om.

När värdet på resursen `dynamicColor` är `True`, tilldelas klienter som inte kan ändra färger dynamiskt (icke-Motif-program) andra celler i färgkartan än klienter som kan ändra färger dynamiskt, även om du ser samma färg.

Obs! Eftersom alla klienter kan dela samma färgceller kan du minska antalet färger som ditt skrivbord använder genom att ange `dynamicColor` 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. Andra program kan ha svårt att hitta resursvärdet och visas därför annorlunda på skärmen.

Om du vill ange skuggans tjocklek till en bildpunkt för icke-Motif 1.2-program gör du så här:

1. **Logga in som root.**
2. **Skapa filen** `/etc/dt/config/språk/sys.resources`.
3. **Ange den programspecifika resursen i**
`/etc/dt/config/språk/sys.resources` **så här:**

```
application_class_name*XmCascadeButton*shadowThickness: 1
```

Mer information om hur du åsidosätter systemets standardresurser och anger ytterligare resurser för alla skrivbordsanvändare finns i "Ladda sessionsresurser" på sidan 27

Konfigurera språkanpassade skrivbordssessioner

Om du vill konfigurera språkanpassade skrivbordssessioner måste du:

- Ställa in LANG-miljövariabeln och övriga variabler för NLS (National Language Support)
- Använda språkberoende meddelandemappar och resursfiler
- Köra fjärrprogram via internationella system
- "Hantera miljövariabeln LANG" på sidan 285
- "Söka efter teckensnitt" på sidan 288
- "Språkanpassa resursfilerna app-defaults" på sidan 288
- "Språkanpassa funktioner och datatyper" på sidan 289
- "Språkanpassa ikoner och bitmappar" på sidan 289
- "Språkanpassa hjälpolymer" på sidan 291
- "Språkanpassa meddelandemappar" på sidan 291
- "Köra språkanpassade fjärranslutna skrivbordsprogram" på sidan 292
- "Återställa tangentbordstabellen" på sidan 292

Hantera miljövariabeln LANG

Kommandot LANG måste vara inställt för att använda operativsystemets språkanpassade 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 ställa in LANG till alla värden som stöds av operativsystemet. Alternativ-menyn på inloggningsbilden visar en lista över språk och områden som stöds.

Det finns fyra olika sätt att ange LANG för skrivbordet:

- Redigera en resurs i filen `Xconfig`-filen
- Använda Alternativ-menyn på inloggningsbilden
- Genom att skapa ett körbart `sh` eller `ksh` `Xsession.d`-skript. (I "Läsa in och köra skriptet `Xsession.d`" på sidan 24 finns mer information om hur du använder `Xsession.d`-skript.)
- Redigera användarens `dtprofile`-fil

När LANG är inställt använder skrivbordet följande språkberoende filer för att bestämma det lokala gränssnittet.

Färger `~/usr/dt/palettes/descspråk`

Bakgrunder `~/usr/dt/backdrops/descspråk`

Ange språk för flera användare

Om du anger språk med hjälp av en `Xconfig`-fil, lokaliseras systemet och LANG anges för alla användare. Detta är det enda sätt på vilket du kan ange LANG för alla visningsenheter i ett system med flera visningsenheter. (Om du vill modifiera `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.host_display.språk språk
```

Följande rad anger LANG till `Swedish_locale` i visningsenheten `my_host:0`.

```
dtlogin.my_host_0.language: Swedish_locale
```

Klienten `dtlogin` läser in den lämpliga meddelandemappen för det språket och visar den språkanpassade inloggningsbilden. Klienten `dtlogin` anger en lista över lokala enheter som använder de följande resurserna i resursfilen `/etc/dt/config/Xresources`:

- `dtlogin*language`
- `dtlogin*languageList`
- `dtlogin*languageName`

Filen `Xconfig` kan behöva ställa in miljövariabeln `NLSPATH` enligt det valda språket. Om detta inte är fallet, eller om du vill ställa in `NLSPATH` själv, läser du "Miljövariabeln `NLSPATH`" på sidan 288.

Ange språk för en session

Om du vill ange språk för en session använder du Alternativ-menyn på inloggningsbilden. Inloggningsbilden är språkanpassad och `LANG` är inställd för användaren. `LANG` använder standardinställningen (som anges `dtlogin`) när sessionen avslutas.

Ange språk för en användare

En användare kan åsidosätta `LANG`-inställningen i filen *Hemmapp*`etc/dt/config/Xsession.d/0010.dtpaths`. Inloggningsbilden är inte språkanpassad och `LANG` är inställd 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 namnet på den mapp som ska innehålla konfigurationsfilerna för din session.

Den språkanpassade 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 körbara 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/inetd.conf
- Egna variabler: *Hemmapp*/.dtprofile

Miljövariabeln NLSPATH

Miljövariabeln NLSPATH bestämmer vilka sökvägar som program ska söka efter meddelandemappar i. Både LANG och NLSPATH måste anges för att det ska gå att använda dessa meddelandemappar. I "Språkanpassa meddelandemappar" på sidan 291 hittar du sökvägar till språkanpassade meddelanden. De flesta skrivbordsklienter kommer att använda sökvägsprefixet NLSPATH vid start.

Söka efter teckensnitt

Skrivbordets teckensnitt finns i mappen /usr/lib/X11/fonts. Varje mapp innehåller en katalogfil (fonts.dir) och en aliasfil (fonts.alias). I mkfontdir direkthjälpen finns information om hur du skapar fonts.dir- och fonts.alias-filer.

Om du vill visa en lista med alla tillgängliga teckensnitt använder du kommandot xlsfonts. Om du vill lägga till eller ta bort teckensnitt till/från servern använder du kommandot xset.

Språkanpassa resursfilerna app-defaults

Standardsökvägen för app-defaults-filen för skrivbordsklienten är /usr/dt/app-defaults/*språk*. Om till exempel LANG är angiven som Swedish_locale, kommer program att söka efter app-defaults-filen i /usr/dt/app-defaults/Swedish_locale. Om LANG inte är angiven, ignoreras *språk*, och programmen kommer att söka efter app-defaults-filen i /usr/app-defaults/C.

Om du vill ändra sökvägen till app-defaults, använder du miljövariabeln XFILESEARCHPATH. Om du vill flytta app-defaults till /users/, ställer du in XFILESEARCHPATH till /usr/app-defaults/*språk*/*classname*.

Om du anger `XFILESEARCHPATH` till `Hemmappetc/dt/config/Xsession.d/0010.dtpaths`, gäller det värdet för alla skrivbords- och X-klienter som du kör. Filer som inte är klienter kommer inte att hitta sina resursfiler om du inte länkar eller kopierar dem till mappen som anges i `XFILESEARCHPATH`.

Språkanpassa funktioner och datatyper

Obs! Om du vill anpassa en fil i mappen `/usr/dt/appconfig`, kopierar du filen till mappen `/etc/dt/appconfig` innan du anpassar den.

Sökvägen för funktions- och datatypsdefinitionsfiler omfattar språkberoende mappar i:

- Egen: `Hemmapp/.dt/icons`
- Systemtäckande: `/etc/dt/appconfig/types/språk`
- Inbyggd `/usr/dt/appconfig/types/språk`

Sökvägen för programhanterarens konfigurationsfiler är:

- Egen: `Hemmapp/.dt/appmanager`
- Systemtäckande: `/etc/dt/appconfig/appmanager/språk`
- Inbyggd: `/usr/dt/appconfig/appmanager/språk`

Fil- och mappnamn i den här mappen är språkanpassade.

Språkanpassa ikoner och bitmappar

Om du vill språkanpassa en ikon redigerar du den i ikonredigeraren och sparar den i:

`/etc/dt/appconfig/icons/språk`

Om du sparar den i en annan mapp inkluderar du den nya mappen i miljövariabeln `XMICONSEARCHPATH`. Miljövariabeln `XMICONBMSEARCHPATH` anger vilken sökväg som ska användas för att söka efter ikoner.

Språkanpassa bakgrundsnamn

För att språkanpassa bakgrundsfiler använder du beskrivningsfilerna `desc.språk` och `desc.backdrops`. Det finns ingen språkanpassad mapp 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 bakgrunden i Inställningar. Om det finns någon beskrivningsspecifikation visas den i bakgrundslistan i Inställningar. I annat fall används bakgrundens filnamn.

Användarna kan lägga till egna bakgrundsbeskrivningar i filen `Hemmapp/.dt/backdrops/desc.backdrops`. Filen används för att hämta bakgrundsbeskrivningar för alla bakgrunder oavsett språk.

Sökvägen för filen `description` är:

- Egen: `Hemmapp/.dt/backdrops/desc.backdrops`
- Systemtäckande: `/etc/dt/backdrops/desc.språk`
- Inbyggd: `/usr/dt/backdrops/desc.språk`

Språkanpassa palettnamn

Språkanpassning av paletter utförs med hjälp av beskrivningsfiler `desc.språk` och `desc.palettes`. Det finns ingen språkanpassad mapp som exempelvis `/usr/dt/palettes/språk`. Alla språk använder samma uppsättning palettfiler men har sina egna `desc.palettes`-filer som innehåller de översatta namnen på paletterna.

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 Inställningar. Om det finns en beskrivningsspecifikation visas den i palettlistan i Inställningar. I annat fall används palettens filnamn.

Användarna kan lägga till egna palettbeskrivningar i filen `Hemmapp/.dt/palettes/desc.palettes`. Filen används för att hämta palettbeskrivningar för alla paletter oavsett språk.

Sökvägen för beskrivningsfilerna är:

- Egen: `Hemmapp/.dt/palettes/desc.palettes`
- Systemtäckande: `/etc/dt/palettes/desc.språk`
- Inbyggd: `/usr/dt/palettes/desc.språk`

Språkanpassa hjälpvolym

Om du har språkanpassat en hjälpvolym måste du lagra den i någon av följande mappar. Den första hjälpvolymen som påträffas används. Mapparna söks igenom i följande ordning:

- Egen: `Hemmapp/.dt/icons`
- Systemtäckande: `/etc/dt/appconfig/icons/språk`
- Inbyggd: `/usr/dt/appconfig/icons/språk`

Språkanpassa meddelandemappar

Om du har språkanpassat en meddelandemapp lagrar du den i följande mapp:

`/usr/dt/lib/nls/msg/språk`.

Dessa mappar innehåller *.cat-filer.

Köra språkpassade fjärranslutna skrivbordsprogram

Du kan anropa språkpassade skrivbordsprogram på alla värddatorer för fjärrkörning som har en liknande språkpassad skrivbordsinstallation. Värdena på de NLS-relaterade miljövariablerna på värddatorn som anropar programmet skickas till fjärrvärddatorn när programmet startas. Miljövariablerna innehåller dock ingen 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 avgörs av miljövariablerna `LC_CTYPE`, `LANG` eller `LC_ALL`, eller det språk som är angivet i alternativet `--lang`.

Om användaren till exempel vill öppna en terminal med hjälp av språket `C` i ett POSIX-skal:

```
LANG=C dtterm
```

Den nya terminalen använder språket `C` med `C`-inmatningsmetoder och `C`-teckensnitt. Om du använder ett språkpassat tangentbord är det möjligt att inmatningsmetoden inte stöder utökade tecken. När du använder språket `C` med ett språkpassat tangentbord måste du i någon av miljövariablerna `LC_CTYPE` eller `LANG` eller `LC_ALL` ange ett lämpligt värde innan du anropar terminalen.

Om du till exempel vill använda språket `C` med ett tyskt tangentbord, skriver du:

```
LANG=C LC_CTYPE=DeDE dtterm
```

Om `X`-servern har startats om och tangenttabeller har initierats, kan du återställa den korrekta tangenttabellen på servern med kommandot `xmodmap`.

Direkthjälp (man) för dtconfig(1)

NAME
dtconfig - desktop configuration utility

SYNOPSIS
dtconfig [-d |-e |-kill |-reset |-p |-inetd |-inetd.ow]

DESCRIPTION
Desktop configuration utility. Integrates CDE with the operating system of the underlying platform. System root login privilege is required to use dtconfig.

OPTIONS

- d Disables desktop auto-start feature. At end of boot cycle, platform's native text based login mechanism will be used.
- e Enable's desktop auto-start feature. Desktop login
- kill Kill desktop (window based) login process and any user sessions associated with it. Return control to system's native text based console.
- reset Tell desktop (window based) login process to reread its configuration file to incorporate any changes.
- p Printer actions for any printer known to platform will be created if such print actions do not already exist in the platform's actions database. This option is executed automatically at boot time if desktop auto-start has been enabled.
- inetd Adds /usr/dt/bin daemons to the /etc/inetd.conf file. Specific CDE background daemon setup includes rpc.ttdbserverd (ToolTalk), rpc.cmsd (Calendar Manager), and dtspcd (subprocess control). This -inetd option is called automatically by Solaris CDE package installs. This -inetd option is also useful for CDE daemon setup outside of normal Solaris CDE install, including system setup where /usr/dt has simply been mounted from some remote fileserver exporting the

(forts.)

/usr/dt directory.

-inetd.ow
Switches the ToolTalk and Calendar Manager daemons (rpc.ttdbserverd & rpc.cmsd) start lines in /etc/inetd.conf back to the older /usr/openwin/bin area. This option is called automatically by Solaris CDE package remove scripts when needed. It is also useful outside of normal Solaris package remove operations when /usr/dt is about to be manually removed or unmounted.

RETURN VALUES

- 0 Successful completion
- >0 Error condition

FILES

- /usr/dt/bin/dtconfig location of dtconfig utility

SEE ALSO

- dtlogin (1), dtprintinfo (1)

Index

Special Characters

* , jokertecken, 207
? , jokertecken, 207

A

actionIcon, resurs, 178
ACTIONS, fält, 204
aktiv session, 22
ALTERNATE_ICON, fält, 242
Alternativ-menyn
 språk, 93
AND, operator i fältet MODE, 209
ANIMATION, definition, 244
Animering i Frontpanelen, 244
anpassa utskrift av brev, 96
ansluta
 en installerad CDE, 88
AnswerBook-dokumentation, lägga till från
 nätverk, 97
använda X-terminaler, 92
användar-ID, 109
användare, ändra funktion, 195
app-defaults
 språkberoende, 288
/appconfig, 84
/programstandardfiler, 84
Apple Macintosh-programmiljön, 99
Arbetsinställningar

 färgintegrering med, 59
 teckensnittsintegrering med, 58
arbetsstationer, som X-terminaler, 94
Arbetsyta-menyn
 anpassa, 257
 definition, 256
 skapa, 258
 syntax, 256
Arbetsyتهanteraren
 byta till OSF/Motif, 263
 definition, 252
 egna anpassningar, 252
 funktioner, 257
 hantera Frontpanelen, 252
 inkludera andra filer, 253
 knappkopplingar, 259
 konfigurationsfiler, 252
 menyer, 256
 starta, 29
arbetsyتهanteraren
 starta, 29
Arbetsyتهanteraren
 starta om, 254
 startar, 23
 systemtäckande anpassning, 253
arbetsyتهväxlare
 definition, 227
 definitionens syntax, 229
arbetsytor

- anpassa, 254
- antal, 254
- namn, 255
- ändra standardantal, 245
- arbetsytor, *Se* bakgrunder
- bakgrunder,
 - argument
 - antal, för funktioner, 190
 - begränsa för funktioner, 190
 - exekveringssträng, 182
 - flera, för funktioner, 186
 - fråga efter fil, 184
 - i funktioner, 150
 - icke utbytbara, för funktioner, 186
 - icke-fil, 184
 - utbytbara, för funktioner, 187
 - Arg_1, syntax, 150
 - ARG_CLASS, fält, 180
 - ARG_COUNT, fält, 180, 190, 191
 - Arg_n, syntax, 183
 - ARG_TYPE, fält, 180, 190, 204
 - utskrift, 129
 - authorize, resurs, 14
 - automontering, 113

B

- %B, 133
- bakgrunder, 252
 - använda grafiska bilder, 255
 - lägga till, 255
 - sökvägar, 255
- bakgrundsresurs, 277
- basfilnamn, 178, 204
- behörighetsmönster, anges med Skapa funktion, 165
- bilder, , *Se* ikoner,
- bildpunktsmönster
 - hur filer hittas, 216
 - namngivningskonventioner, 216
- /bin, 84
- bitmapp, 216
 - hur filer hittas, 216, 218
 - namngivningskonventioner, 216
 - sökväg, 218
- bitpunktsmönster
 - hur filer hittas, 218
 - sökväg, 218

- blank, kontrolltyp, 239
- bläddra ikoner med Filhanteraren, 220
- .bm-namntillägg, 216
- bottomShadowColor, resurs, 278
- BOX, definition, 227
 - syntax, 228
- BROADCAST, används i XDMCP-indirekt, 8
 - broadcast, flagga, 95
- buttonBindings, resurs, 261

C

- C, språkområde, 94
- CDE-MIN, filer, 112
- CDE-TT, filer, 112
- CHOOSE, sträng, 8
- client, kontrolltyp, 239
- CLIENT_GEOMETRY, fält, 243
- CLIENT_NAME, fält, 242, 243
- clock, kontrolltyp, 239
- colorUse, resurs, 28, 281, 282
- COMMAND, funktion, 172
 - exekveringssträng, 181
 - exempel, 174
 - nödvändiga fält, 181
- /config, 84
- CONTAINER_NAME, fält, 228, 229, 233
- CONTAINER_TYPE, fält, 229
- CONTENT, fält, 206, 211
- CONTROL, syntax för definition, 228
- CONTROL_BEHAVIOR, fält, 248
- COPY_TO_ACTION, fält, 206
- cpp statements, 31
- current.old, mapp, 33

D

- databas
 - ladda om, 176
 - starta om funktioner, 176
- %DatabaseHost%, 192
- DataBaseHost, nyckelord, 192
- databasservrar, 107, 120
 - klient till, 119
 - konfigurera, 112, 118
 - skapa, 119
- databassökväg, 137, 173

- förhållande till programsökväg, 136, 138
- inverkan på EXEC_HOST, 121
- miljövariabler, 138
- sammansättning, 139
- standard, 138
- syntax, 138
- databasvärd, 120, 192
- datatyper
 - associera ikoner med, 219
 - associera med funktioner, 204
 - attribut, 203
 - begränsa funktioner baserat på, 190
 - begränsning av Skapa funktion, 159
 - behörighetsmönster, 167
 - definiera, 201
 - definition, 104
 - dolda, 205
 - exempel, 202
 - flera villkor, 211
 - förhållande till funktioner, 152
 - hjälp med, 203
 - ikon, 107
 - ikoner för, 67, 169, 203
 - innehållsbaserad, 211
 - integrerade med dtappintegrate, 75
 - introduktion, 146, 151
 - kategoriseringsvillkor, 206
 - konfigurationsfiler, 201
 - krav för att skapa manuellt, 159, 199
 - körbar, 210
 - ladda om, 176
 - Ladda om funktioner, 176
 - lägesvillkor, 209
 - metod för att släppa, 155
 - metod för dubbelklickning, 154
 - namnbaserade, 165, 207
 - nödvändiga för registrering, 63
 - platsbaserade, 208
 - program, 104
 - servrar för, 118
 - session, 112
 - servrar, 104
 - skapa manuellt, 199
 - skapa med Skapa funktion, 152, 162
 - skilja, 206
 - skriva ut, 155
 - skrivskydd, 210
 - språkanpassade, 212
 - syfte med att skapa, 55
 - sökvägsbaserade, 208
 - variabler i definitioner, 193
 - villkor, 206
- datatyper, *Se* databassökväg
- sökväg,
- Datatypfamiljens namn, textfält, 164
- DATA_ATTRIBUTES
 - definiera, 203
 - definition, 200
 - syntax, 201
- DATA_CRITERIA
 - definiera, 206
 - definition, 200
 - flera, 212
 - syntax, 202
 - tillsammans med
 - DATA_ATTRIBUTES, 200
- date, kontrolltyp, 239
- DELETE, fält, 227
- DESCRIPTION, fält, 178, 203
- desktop-smart-program, 44
- /dev/console, 86
- Dialogrutan Sök ikonuppsättning, 168
- DISPLAY, variabel, anges av
 - inloggningshanteraren, 16
- %DisplayHost%, 192
- DisplayHost, nyckelord, 192
- DISPLAY_skärmmamn, makro, 31
- DROP_ACTION, fält, 241
- DROP_ANIMATION, fält, 244
- dt, filer, 173
- dtaction
 - använd för att ändra användare, 195
 - syntax, 195
- dtappgather, 26, 41
- dtappintegrate, 73
 - funktionalitet, 75
 - syntax, 74
 - ta bort program, 49
- DTAPPSEARCHPATH, variabel
 - definition, 132
 - sammansättning, 135
- DtButtonBindings, 259
- dtchooser, fil, 19
- dtconfig(1) man page, 293
- dtconfig, kommando, 9

dtCreate, , *Se* funktion,
DTDATABASESEARCHPATH, variabel
 användning, 180
 definition, 132
 sammansättning, 138, 139
DtEditor, stildefinitioner för, 266
Dterrors, fil, 9
dtgreet, fil, 19
DTHELPSEARCHPATH, variabel
 definition, 132
 sammansättning, 142
dtlogin, 85
Dtlogin*language, 14
dtlogin, , *Se* Inloggningshanterare,
dtlp, 97
dtmailpr, 97
DTMOUNTPOINT, variabel
 ange, 114
 processer som använder, 114
 processer som behöver, 114
 ärvt av användare, 114
Dtpid, fil, 3
.dtprofile, 35 to 37
.dtprofile, fil
 skapa, 24
 ange egna miljövariabler i, 30
 Läsa in och köra i, 24
 läser in och kör skriptet i, 23
 syntax, 30
.dtprofile file
 ange LANG, 287
DtRootMenu, 257
dtsearchpath, 25, 133, 179
dtsmcmd, kommando, 32
DTSOURCEPROFILE, variabel, 26
dtspc, 112, 114
dtspcd, 115
 konfigurera, 115
 verifieringsmapp, 109, 115
DTSPSYSAPPHOSTS, variabel
 definition, 132
 syntax, 135
 ändra, 48
DTSPSYSDATABASEHOSTS, variabel
 definition, 132, 138
 effekt på fältet EXEC_HOST, 121
 syntax, 138
DTSPSYSHELP, variabel, 142
 definition, 132
 syntax, 142
DTSPSYSICON, variabel
 definition, 132
 syntax, 140
DTSPUSERAPPHOSTS, variabel
 definition, 132
 syntax, 135
 ändra, 48
DTSPUSERDATABASEHOSTS, variabel
 definition, 132, 138
 syntax, 138
DTSPUSERHELP, variabel
 definition, 132
 syntax, 142
DTSPUSERICON, variabel
 definition, 132
 syntax, 140
dtstart_appgather, variabel, 26
dtstart_searchpath, variabel, 25
dtstart_ttsession, variabel, 27
dtterm, 38
dtwm , *Se* Arbetsyttehanteraren,
dtwm.fp, fil, 224
dtwmfp.session, fil, 225
dtwmrc-file
 redigering, 252
dtwmrc-filen, 252
dynamicColor, resurs, 28, 281, 283
dölja filer med datatyper, 205

E

e-post, konfigurera, 111
egen datatyp och funktion, skapa, 202
egen programgrupp, 46
Egenskaper
 dialogruta, 165
EMACS definitioner, 266
EMACS-stildefinitioner, 266
/etc/dt, 84
/etc/rmmount.conf, 99
etiketter
 funktioner, 178
 /exempel, 84
EXEC_HOST
 flera värden, 121

- påverkas av databassökvägen, 121
- EXEC_HOST , Se värddator för exekvering,
- EXEC_HOST, fält, 192
 - standardvärde, 193
- EXEC_STRING, 97
- EXEC_STRING, fält, , Se exekveringssträng,
- exekveringssträng, 181
 - absolut sökväg i, 182
 - allmänna egenskaper, 182
 - ange körbar fil, 182
 - flera filargument, 186
 - fråga efter fil, 184
 - fråga efter sträng, 184
 - släppta filer, 183
 - syntax i exekveringssträng, 182
 - utan argument, 183

F

- felloppningsfil
 - Sessionshanteraren, 36
 - start, 36
 - Xsession, 36
- felloppningsfiler
 - inloggning, 36
 - kataloger, 36
- felloppningsfiler för inloggning, 36
- felloppningsfiler för start, 36
- felsäker session, 15
- Felsäker session, alternativ, 36
 - inloggningsskärm, 88
- felsökning skrivbordstart, 35
- fil, datatypsvillkor, 210
- filargument
 - ange med Skapa funktion, 162
 - i funktioner, 150
- filindelning, 109
- file, kontrolltyp, 239, 242
- filer
 - dölja baserat på datatyp, 205
 - fjärrdata, 113
 - fjärråtkomst, 109
 - konsekventa namn, 110
 - montera, 109
 - monteringspunkt, 113
 - nödvändiga för nätverk, 112
 - åtkomst till distribuerade, 109
- filer, starta inloggning, 35

- FILE_NAME, fält, 241, 242
- filnamn, ikon, 216
- filnamnsdatabasserver, 112
- filnamnstilldelning, 113
- filserverar, 104, 105
- filtyper, , Se datatyper,
- fjärrexekvering
 - med funktioner, 192
- fjärrkörning
 - konfigurera programserver, 117
 - lokalt språkstöd, 292
 - med en fjärransluten funktion till programmet, 120
- flagga, -broadcast, 95
- flera skärmar, 89
 - Inloggningshanterare, 13
- Font, resurs, 272
- FontSet, resurs, 272
- foregroundColor, resurs, 28, 283
- fp_dynamic, mapp, 224
- Frontpanelen, 247
 - animering, 244
 - anpassa, 223
 - använda funktioner, 147
 - dynamiska anpassningar, 224, 225
 - fönster i Frontpanelen, 243
 - hjälp, 244
 - hur definitioner organiseras, 227
 - komponenter, 227
 - konfigurationsfilen, 223
 - kontroll i registreringspaketet, 72
 - kontrollera egna anpassningar, 226
 - läge på skärmen, 247
 - lägga till rader, 248
 - meny, 256
 - målområde, 241
 - namngivningskonvention för filer, 224
 - ny, 248
 - prioritet vid sammansättning, 225
 - stys av arbetsyttehanteraren, 252
 - syntax, 227
 - sökvägar, 224
 - visa ikoner i, 219
 - ändra, 230
- Frontpanelen, Se ikon
- kontroll,,
 - främmande skärmtyp, 5

- fråga efter funktioner, 184
- funktioner
 - acceptera flera släppta filer, 187
 - acceptera släppta filer, 151, 183
 - acceptera släppta filer eller fråga efter, 184
 - använda, 151
 - används av Frontpanelen, 147
 - används på menyer, 148
 - argument, 182
 - argument, begränsande antal, 190
 - argument, icke-fil, 184
 - associera ikoner med, 218
 - associera med datatyper, 204
 - begränsa argument, 190
 - begränsade av datatyp, 190
 - begränsade av en datatyp, 154
 - begränsningar i Skapa funktion, 158
 - COMMAND, 172
 - etiketter, 178
 - exekveringssträng, 181
 - exempel, 174, 175
 - filargument, 150
 - fråga efter fil, 184
 - fönsterstöd, 162
 - fönsterstöd för, 188
 - förhållande till datatyper, 152
 - icke utbytbara argument, 186
 - icke-filargument, 158, 184
 - ikoner för, 169, 178
 - ikonrepresentation av, 177
 - integrerade med dtappintegrate, 75
 - introduktion, 146
 - konfigurationsfiler, 173
 - kör som andra användare, 195
 - köra andra funktioner, 195
 - köra fjärrprogram, 192
 - ladda om, 176
 - MAP, 172
 - miljövariabler., 194
 - namn, 158, 177, 196
 - nödvändiga för registrering, 63
 - olika metoder för dubbelklickning och släpp, 191
 - prioritetsregler, 180
 - redigera, 179
 - servrar för, 118
 - skal för, 185
 - skapa ikoner för program, 148
 - skapa manuellt, 171, 173
 - som kör fjärrprogram, 120
 - språkanpassade, 196
 - standardikon, 178
 - standardterminal, 189
 - strängvariabler i, 193
 - terminalstöd för, 189
 - terminalalternativ, 189
 - tilldelning, 154
 - TT_MSG, 173
 - typer av, 172, 174
 - utan argument, 183
 - utan data, 183
 - utbytbara argument, 187
 - varför de måste skapas manuellt, 172
 - variabler i definitioner, 193
 - visa inga utdata, 162
 - ändra, 179
- funktioner, *Se* argument
- parametrar.,
- funktioner, *Se* databassökväg
- sökväg.,
- funktioner, *Se* funktionsfil
- filer som representerar.,
- funktionsdefinitionsfil, skapad av Skapa funktion, 157
- funktionsfil, 157, 177
 - definition, 149
 - i funktioner, 150
 - innehåll, 149
 - skapa, 177
- Funktionsfilen
 - skapa, 71
- funktionsikon, 149, 177
 - nödvändig för skrivbordet, 67
 - skapa, 177
- Funktionsikoner, verktyg i Skapa funktion, 161
- Funktionsnamn, fält i Skapa funktion, 161
- funktionsserver, . *Se* databasserver,
- funktionsökväg, . *Se* databassökväg,
- färg
 - ange förgrund, 283
 - använda i ikoner, 221
 - färguppsättningar, 277
 - kontrollera, med Inställningar, 280
 - kontrollera, med resursen dynamicColor, 284

- maximalt antal allokerade, 281
- skapa skuggor med resurserna
 - shadowPixmaps, 283
- färger, 276, 278
 - administrera, 276
 - aktiv fönsterram, 278
 - antal som används, 281
 - färguppsättningar, 277
 - områden för textinmatning, 279
 - paletter, 276
 - programfönster, 279
 - ram på inaktivt fönster, 278
 - resurser, 277
 - standard, 279
 - värden, 278
- färghanteraren, 23
- färgresurser, ändra för registrering, 59
- färgservern
 - resurser, 28
 - starta, 28
- färguppsättningar, 276, 277
 - standard, 279
 - tilldela för skärmelement, 278
- Fönster-menyn
 - definition, 256
 - ny, 258
 - syntax, 256
- fönsterhanterare
 - ändra, 29
- Fönsterhanteraren, 252
- Fönsterhanteraren OSF/Motif, , Se Arbetsytahanteraren,
- fönsterstöd för funktioner, 162
- förgrundsresurs, 277

G

- getty, 5, 23
- GID, 109
- granska ikoner med Filhanteraren, 220
- grupp-ID, 109

H

- %H, 133
- Help Developer's Kit, 66
- HELP_STRING, fält, 245
- HELP_TOPIC, fält, 245

- HELP_VOLUME, fält, 245
- Hemmapp, 85
- hemmapp
 - delad, 109
 - nätverksansluten, 110
- hemsession, 22
- HIGH_COLOR, 282
- hjälp, 66
 - ange med Skapa funktion, 162
 - delvis integrering, 66
 - Frontpanelen, 244
 - full integrering, 66
 - för skrivarikon, 128
 - med datatyper, 203
 - objekt-, fil, 178
- Hjälpfiler:
 - i registreringspaketet, 66
 - integrerade med dtappintegrate, 75
- hjälpserver, 107
 - klient till, 119
 - konfigurera, 112, 118
 - skapa, 119
- hjälpsoökväg, 132, 141
 - förhållande till programsökväg, 136, 141
- Hjälpsoökväg
 - hjälp, 141
- hjälpsoökväg
 - miljövariabler, 142
 - sammansättning, 142
 - standard, 141
 - syntax, 142
- hjälpvolym
 - adress i registreringspaketet, 66
 - huvudhjälpfil, 66
 - integreringsnivåer, 66
- hjälpvolym
 - språkpassa, 291
- HOME, variabel, 17
- home.old, mapp, 33

I

- ICON, fält
 - för datatyper, 203
 - giltiga värden, 178, 204
 - i Frontpanelen, 234
- icon, kontrolltyp, 242

- ikon
 - byta plats med andra ikoner, 232
 - definiera, 238
 - ersätta, 233
 - lägga till i huvudpanelen, 230
 - som öppnar en fil, 241
 - ta bort från Frontpanelen, 231
 - utseende, 234
 - ändra, 232
 - övervaka, 241
- ikoner
 - använda Filhanteraren som
 - ikonfönster, 220
 - associera, 218
 - associera med funktioner och
 - datatyper, 218
 - associera med programfönster, 220
 - basfilnamn, 204
 - datatyp, 67
 - datatyper, 169, 203
 - Dialogrutan Sök ikonuppsättning, 168
 - filformat, 216
 - Filhanteraren som ikonfönster, 220
 - filnamn, 216
 - Frontpanelen, 219, 234
 - funktioner, 169, 178
 - funktionsikoner, 177
 - färganvändning, 221
 - för program, 148
 - hur filer hittas, 216
 - icke-engelska, 289, 290
 - integrerade med dtappintegrate, 75
 - lokaliserade, 290
 - namngivningskonventioner, 216
 - nödvändiga för registrering, 66
 - programgrupper, 67, 69
 - rekommendationer när du skapar, 221
 - representera program, 71
 - skrivarbilder, 127
 - språkanpassa, 289
 - språkanpassade, 290
 - starta program, 67
 - storlekskonventioner, 216
- ikoner, *Se* ikonservrar
 - servrar,,
- ikoner, *Se* ikonsökväg
 - sökväg,
- ikonplacering, 38
- ikonservrar, 107
 - klient till, 119
 - konfigurera, 112, 118
 - skapa, 119
- ikonstorlekar, 216
- ikonsökväg, 139
 - förhållande till programsökväg, 136, 140
 - miljövariabler, 140
 - sammansättning, 141
 - standard, 139
 - syntax för, 140
- inactiveFrameColorId, resurs, 278
- /inkludera, 84
- indatavariabler, 132
 - indirekt, alternativet, 6
- inetd.conf, 114
- inkludera instruktionen i
 - arbetsyتهanterarfiler, 253
- inloggning äkthetskontroll, 35
- Inloggningshanterare, 2
 - administration, 18
 - definition, 2
 - fel, 8
 - konfigurationsfilen, 19
 - resurser, 10, 12
 - starta kommandon, 15
 - ändra, 2
- inloggningskonton, 109
- inloggningsserver
 - avbryta, 87
 - konfigurationer, 3
 - koppla ur vid systemstart, 86
 - starta från kommandorad, 86
 - starta vid systemstart, 86
- inloggningsservrar
 - ansluta skärmar, 2
 - användarsökväg, 17
 - avsluta process-ID, 9
 - felsökning, 8
 - inaktivera, 9
 - ingen bitmappskärm, 4
 - kommandoradsinloggning, 4
 - konfigurera, 111
 - kontrollera behörighet, 7
 - köra utan lokal skärm, 4
 - miljö, 16, 17
 - process-ID, 3

- starta, 2
- starta en session, 2
- starta från kommandoraden, 2
- systemskal, 18
- textskärmen, 5
- verifiera användare, 2
- visa inloggningsskärmen, 2
- ändra tidszon, 18
- översikt, 2
- inloggningsskärm
 - anpassning, 10
 - avsluta, 87
 - Felsäker session, alternativ, 88
 - lokaliserar, 12
 - resurs, 11
 - skärmberoende funktionalitet, 13
 - teckensnitt, 11
 - visa på en nätverksskärm, 5
 - välkommen, 11
 - X-serverbehörighet, 13
 - X-servermiljö, 14
 - ändra funktion, 12
 - ändra innehåll på Språk-menyn, 14
 - ändra standardspråk, 14
 - ändra utseende, 10
- inloggningsskärm
 - ändra välkomstmeddelande, 11
- inloggningsspråkområde, 93
- inmatningsmetod, lokalisering, 292
- innehållsbaserad datatyp, 211
- installation
 - mappadress, 84
- Installera ikon, ta bort, 237
- Inställningar
 - använda för att ange färger, 278
- IS_TEXT, fält, 206

J

- jokertecken i datatyper, 207

K

- Kalenderdemon, 116
- keyBindings-resurs, 263
- klient-serverkonfiguration, , Se nätverk,
- klienter
 - definition, 104
 - fönster i Frontpanelen, 243
 - till servrar, konfigurera, 111
- knappkoppling
 - syntax, 259
- knappkopplingar, 259
 - lägga till, 260
 - skapa ny uppsättning, 260
- kommandorad för funktioner, 181
- kommandoradsinloggning, 5
- konfigurationsfilen, 85
 - adress, 84
 - Frontpanelen, 223
 - i registreringspaket, 56
 - Sessionshanteraren, 34
- konfigurationsfiler
 - Arbetsytehanteraren, 252
 - datatyper, 201
 - funktion, 173
 - Fönsterhanteraren, 252
 - Inloggningshanterare, 19
- konsekventa filnamn, 110
- kontroll
 - animering, 244
 - en förekomst, 242
 - i område för byte av arbetsyta, 246
 - ikon, 234
 - klicka eller dubbelklicka, 248
 - klient, 243
 - läsa, 226
 - målområde, 241
 - namnge, 247
 - objekthjälp, 244
 - skapa, 239
 - typer, 239
 - växling, 242
 - återställa, 226
- koppla ur en ansluten CDE-mapp, 89
- körbar fil, datatypsvillkor, 210
- körvärd
 - anges av fältet EXEC_HOST, 192
 - skapa funktioner för, 192

L

- %L, 133
- LABEL, funktionsfält, 178
- Ladda om funktion, funktion, 176

- Ladda om funktioner, funktion, 176
- Ladda om program, funktion, 49
- Ladda om resurser, funktion, 28
- LANG, variabel, 285
 - anges av inloggningshanteraren, 16
 - effekt på datatyper, 212
 - i .dtprofile, 287
- ledtext, ange med Skapa funktion, 163
- /lib, 84
- LINK_TO_ACTION, fält, 206
- listan Datatyper i Skapa funktion, 163
- %LocalHost%, 192
- localTerminal, resurs, 189
- LOCKED, fält, 226
- .login, 37
- .login, fil, 23
- .login, fil
 - inte läsbar av Inloggningshanterare, 30
 - läs in och köra, 26
- LOGNAME, variabel, 17
- lokal skärmtyp, 5
- lokalisera, 290
 - app-defaults, 288
 - ikoner, 289
 - meddelandemappar, 291
 - teckensnitt, 288
- lokalisering
 - felsökning, 292
 - inloggningsskärm, 12
 - LANG, variabel, 285
 - NLS-miljövariabler, 288
 - ställa in språk, 286
- LOW_COLOR, 282
- lp
 - kommando, 110
 - skrivarspooler, 110
- LPDEST, variabel, 128
- länk, datatypsvillkor, 210

M

- %M, 133
- mail, kontrolltyp, 242
- mailx, 111
- /man, 84
- MAP, funktioner, 154
 - definition, 172
 - exempel, 175

- mapp
 - datatypsvillkor, 210
- Markera-menyn, 148, 153
- meddelandemappar, 288
- MEDIA, fält, 206
- MEDIUM_COLOR, 282
- menyer
 - använda funktioner, 148
 - Arbetsyterhanteraren, 256
- miljöfil, skrivbord, 98
- miljövariabler
 - ange, 30
 - bildpunktsmönstersökväg, 140
 - bitmappsökväg, 140
 - databassökväg, 138
 - egna, 30
 - i .dtprofile, 24
 - exportera, 30
 - fjärrkörning, 115
 - hjälpökväg, 142
 - i .dtprofile, 24
 - ikonsökväg, 140
 - Inloggningshanterare, 17
 - läs in och köra .login eller .profile, 26
 - programsökväg, 135
 - standard, 24
 - systemtäckande, 30
 - sökväg, 132, 207
- miljövariabler.
 - funktionsdefinitioner, 194
- MIME_TYPE_MEDIA, fält, 206
- mkfontdir-kommandot, kompilera filer, 288
- MODE, fält, 207
 - syntax, 209
- MONITOR_TYPE, fält, 242
- monteringar, köra program via, 122
- monteringspunkt för fjärrfiler, 113
- MOVE_TO_ACTION, fält, 206
- muskopplingar, . Se knappkopplingar,
- målområde
 - ikon på Frontpanelen, 241

N

- NAME_PATTERN, fält, 206
- namn
 - funktioner, 196

- kontroller på Frontpanelen, 247
- namnbaserade datatyper, 207
- namnge kontroller, 247
- NFS, 109
- NLS
 - lokalisering, 285
- NLS-fjärrkörning, 292
- NLS-miljövariabler, 288
- NoBackdrop, inställning, 255
- NoPrint, funktion, 155
- NOT, operator i fältet MODE, 209
- NO_STDIO, fönsterstöd, 188
- NUMBER_OF_ROWS, fält, 245
- nyckelkonfigurationsfiler, 85
- nätverk, 103, 111
 - allmänna steg för att konfigurera, 108
 - baskonfiguration, 108
 - e-post, 111
 - filer som är nödvändiga för, 112
 - konfigurera klienter och servrar, 111
 - köra program via monteringar, 122
 - typer av tjänster, 104
 - X-verifiering, 111
 - översikt, 103
- nätverksanslutna skrivbord, 91

O

- Område för växling av arbetsyta
 - anpassa, 245
- område för växling av arbetsyta
 - antal rader, 245
 - lägga till kontroller, 246
- once, alternativ, 95
- Open, funktion, 154
- OpenWindows
 - miljö, 86
- OR, operator i fältet MODE, 209

P

- paletter, 276
 - språkanpassa namn, 290
- /paletter, 84
- PANEL, definition, 227
 - syntax, 228
- panelundermeny
 - anpassa inbyggd panel, 236

- associera med huvudpanelen, 235
- behållare, 229
- definition, 229
- ny, 235
- skapa, 234
- syntax, 229
- ändra, 234, 235
- ändra metoden auto-close, 238
- återställa borttagna, 226
- PANEL_GEOMETRY, fält, 247
- PATH, variabel, 182
 - anges av inloggningshanteraren, 16
- PATH_PATTERN, fält, 206
 - syntax, 208
- PERM_TERMINAL, fönsterstöd, 188
- placering, ikoner, 38
- platsbaserad datatyp, 208
- .pm-namntillägg, 216
- POSITION_HINTS, fält, 232
- prioritet
 - i funktionsdatabas, 180
 - sammansättning av Frontpanelen, 225
- problem, skrivbordstart, 35
- .profile, 37
- .profile, fil, 23
- .profile, fil
 - inte läsbar av Inloggningshanterare, 30
 - läs in och köra, 26
- program
 - avregistrera, 49
 - funktioner med registrering, 54
 - köra lokalt via monteringar, 122
 - lägga till, 43
 - lägga till i befintliga programgrupper, 45
 - lägga till i programhanteraren, 43
 - lägga till utan registrering, 44
 - nödvändiga datatyper, 63
 - nödvändiga funktioner, 63
 - programrot, mapp, 59
 - registrerad, definition, 44
 - rotmapp, 59
 - samla ihop i programhanteraren, 40
 - samla in, 42
 - samlade av sessionshanteraren, 26
 - skapa ikoner för, 148
 - skrivbordsanpassad, 44
 - starta vid inloggning, 23, 29

- syfte med datatyper, 55
- sökväg, 40
- ta bort, 49
- återskapa, 49
- program, *Se* registrering
 - registrera,,
- programfönster, associera ikoner med, 220
- programgrupper
 - anpassa, 46
 - datatyper för, 70
 - definition, 39
 - egen, 46
 - exempel på hur du kan skapa, 81
 - funktion för, 70
 - hantera, 46
 - ikoner för, 67, 69
 - innehåll, 70
 - integrerade med dtappintegrate, 75
 - mappar för, 68
 - namn, 68
 - namnge, 46
 - prioritetsregler, 41
 - readme-filer, 72
 - samla ihop, 40
 - skapa i registreringspaketet, 68
 - standard, 41
 - systemtäckande, 46
- Programhanteraren
 - allmän administration, 49
 - beskrivning, 39
 - filsystemplacering, 40
 - integrera program med, 54
 - lägga till program, 43
 - prioritetsregler, 41
 - samla ihop program, 40
 - samla program, 26
 - symboliska länkar, 41
 - uppdatera, 49
- programikon, 174
 - använda Skapa funktion, 158
 - dubbelklicka, 151
 - nödvändig för skrivbordet, 67
 - skapa, 149, 177
 - släppta filer, 151
- programpaket, , *Se* registreringspaket,
- programregistrering, , *Se* registrering,
- programresurser, , *Se* resurser,
- programrot , *Se* programrotsmapp,

- programrotsmapp, 59
- programservrar
 - administrera, 116
 - definition, 104
 - klient till, 118
 - konfigurera, 112, 117
 - konfigurera klient till, 118
 - lägga till, 47
 - standardkonfiguration, 117
 - tillgänglighet hos program, 57
- programstandardfiler
 - skrivbordsprogram, 265
- programsökväg
 - används för att samla ihop program, 40
 - definition, 134
 - egen, 48
 - lokaliserad, 143
 - miljövariabler, 135
 - sammansättning, 135
 - skäl att ändra, 47
 - standard, 47, 134
 - syntax, 135
 - systemtäckande, 48
 - ändra, 48
 - ändra prioritet, 136
- protokoll, XDM, 92
- punktuppbyggd skärm, utan
 - inloggningsserver, 4
- PUSH_ACTION, fält, 240
- PUSH_ANIMATION, fält, 244

Q

- query, alternativ, 6

R

- readme-filer, 72
- Redigera Dtwmrc, åtgärd, 252
- redigera funktioner, 179
- registrering, 71
 - allmänna steg, 57
 - definition, 44, 57
 - dtappintegrate, 73
 - exempel, 76, 83
 - funktioner som följer med, 54
 - färgändringar, 59

- Hjälpfiler:, 66
- ikonkrav, 66
- nödvändiga datatyper, 63
- nödvändiga funktioner, 63
- programgrupp, 68
- programrotsmapp, 59
- resursändring, 58
- syftet med, 56
- teckensnittsändringar, 58
- ändra resurser, 58
- översikt, 54
- registreringspaket, 44, 71
 - definition, 57
 - exempel på hur du skapar, 76
 - kontroll för Frontpanelen, 72
 - mappar, 60
 - programgruppsinnehåll, 70
 - programikon, 71
 - readme-filer, 72
 - syftet med, 54
- Reload Actions, funktion, 176
- RESOURCE_MANAGER, egenskap, 28, 30
- resurser
 - ange, 30, 265
 - colorUse, 282
 - egna, 266
 - foregroundColor, 283
 - ladda, 27
 - programstandardfiler, 265
 - session, 23
 - shadowPixmaps, 282
 - skuggtjocklek i fönster, 284
 - skärmspecifika, 31
 - språkberoende, 288
 - standardskrivbord, 27
 - systemtäckande, 266
 - teckensnitt, 271
- RGB-färgvärde, 278
- RGB-värden, 278
- rgb.txt, fil, 278
- rotfönster, 255
- rotmenyn, , Se Arbetsyta-menyn,
- rpc.cmsd, 116
- rpc.ttdbserver, 112, 114

S

- .sdl, filer, 66

- sdtcm_convert, kommandofil, 97
- secondaryColorSetId, resurs, 279, 281
- selectColor, resurs, 278
- sendmail, 111
- servrar
 - databas, 112
 - datatyper, 118
 - fil, 104
 - funktioner, 118
 - hjälp, 107, 112
 - ikon, 112
 - inloggning, 112
 - konfigurera, 111
 - program, 112
 - typer, 107
- session
 - aktiv, 22
 - definition, 21
 - felsäker, 15
 - först, 31
 - hem, 22
 - kör kommandon vid start, 33
 - köra kommandon vid utloggning, 33
 - loggning av fel, 34
 - resurser, 23
 - skript som körs vid avslut, 16
 - skärmspecifika, 22, 32
 - standard, 22
 - start, 22
 - starta, 23
 - säkerhetskopior, 33
 - återskapa, 33
- sessionetc, fil, 33
- sessionexit, fil, 33
- %SessionHost%, 192
- SessionHost, nyckelord, 193
- sessionshanterare, 23
 - starta, 23
- Sessionshanteraren
 - ange sökvägar, 25
 - anpassa program vid start, 29
 - fellogg, 34
 - felloggningsfil, 36
- sessionshanteraren
 - felsökning, 33
- Sessionshanteraren
 - felsökning, 34

- filer, 34
- introduktion, 21
- klient, 27
- kör ytterligare kommandon, 33
- köra kommandon vid utloggning, 33
- ladda resurser, 27
- mappar, 34
- sessionshanteraren
 - samla ihop program, 26
- Sessionshanteraren
 - starta arbetsyttehanteraren, 29
 - starta program, 29
 - systemtäckande anpassning, 24
 - välkomstmeddelande, 23
- sessionsmapp, 32
- sessionsservrar, , *Se* inloggningsservrar,
- shadowPixmaps, resurs, 28, 282
- /share, 84
- SHELL, variabel, anges av
 - inloggningshanteraren, 16
- skal
 - använda i funktioner, 185
 - egen anpassning, 24
 - i funktioner, 185
 - läs in och köra .profile eller .login, 26
 - Se* även miljövariabler, .profile, .login, .dtprofile, 30
 - skapa funktioner för, 182
 - systemtäckande anpassning, 24
- Skapa funktion, 157
 - ange filargument, 162
 - ange ikon, 168
 - använda, 159
 - begränsningar, 158
 - datatypsnamn, 164
 - funktionalitet, 157
 - funktionsnamn, 161
 - huvudfönster, 161
 - introduktion, 157
 - ledtext, 163
 - skapa datatyper, 152, 162
 - skapad konfigurationsfil, 157
 - starta, 160
 - syntax för funktionskommando, 162
- skriva ut
 - konfigurera för datatyp, 155
- skrivare
 - enhetsnamn, 110
 - etiketter, 128
 - fjärråtkomst, 110
 - ikonbilder, 127
 - intervall för jobbuppdatering, 127
 - lägg till, 125
 - objekthjälp, 128
 - standard, 128
 - ta bort, 126
- skrivarspooler, lp, 110
- skrivbord, nätverksanslutna, 91
- skrivbordsmiljöfil, 98
- skrivbordssökvägar, 23
- skrivbordstart
 - problem, 35
- Skrivbordsverktyg, programgrupp, ändra, 46
- skrivskyddad, datatypsvillkor, 210
- skuggtjocklek, fönster, 284
- skärmfärger
 - maximalt antal allokerade, 281
- skärmspecifik session, 32
- skärmspecifika resurser, 31
- släppområde
 - funktionsikon, 183
- släppt fil, funktion som accepterar, 183
- Solaris CDE
 - avbryta skrivbord, 87
- SPC, 115
 - säkerhet, 115
- språk, ange med filen Xconfig, 286
- Språk, menyalternativ, 93
- Språk-menyn, anpassa, 14
- språkanpassa
 - ikoner, 290
 - palettnamn, 290
- språkanpassning
 - datatyp, 212
 - funktioner, 196
 - funktionsnamn, 196
- språkområde, C, 94
- starta skrivbord
 - flera skärmar, 89
- startfiler för inloggning, 35
- startlog, fil, 34
- strängfunktionsargument, 184
- strängvariabler i funktionsdefinitioner, 193
- stty, 37
- styrdaemon för underprocesser, , *Se* dtspcd,

SUBPANEL, definition, 227
SWITCH, definition, 229
symboliska länkar
 datatypsvillkor, 207
 konsekventa filnamn, 110
 skapade under registrering, 75
sys.dtpofile, fil, 24
sys.dtwmrc-filen, 252, 253
sys.resources, fil, 27, 30, 266
sys.session, fil, 29, 32
systemPath, resurs, 17
Särskilda egenskaper
 fält, 167
sökväg
 användare, 17
 system, 17
sökvägar
 aktuellt värde, 133
 ange, 133
 ange med sessionshanteraren, 25
 definierade av skrivbordet, 132
 frontpanelsdefinitioner, 224
 funktioner, 173
 hur filer hittas, 218
 indatavariabler, 132
 lokaliserade, 143
 miljövariabler, 132, 207
 program, 40, 134
 utdatavariabler, 132
sökvägsbaserad datatyp, 208

T

tangentkoppling
 skapa ny uppsättning, 262
 syntax, 261
tangentkopplingar
 förinställda, 261
teckensnitt
 administrera, 271
 ange egenskapssträngar, 273
 antal i Inställningar, 272
 bitmapp, 271
 huvudkatalog, 288
 lista tillgängliga, 272
 resurser, ange, 271
 resurser, ändra för registrering, 58
 system-, i Inställningar, 272

 söka med aliasfilen, 288
 söka med katalogfilen, 288
 söka med mkfontdir-kommandot, 288
 X Logical Font Description, 271, 273
 xlsfonts, kommando, 288
Teckensnitt, dialogruta, 271
teckensnittssökväg, 93
TERMINAL, fönsterstöd, 188
terminalemulator
 för funktioner, 188
 kommandoradsalternativ för
 funktioner, 189
 som stängs automatiskt, 162
 standard för funktioner, 189
 ändra, 50
terminalemulering, 38
textColorSetId, resurs, 278
textredigerare, ändra, 50
textskärmen, 5
tidszon, ändra, 18
timeZone, resurs, 18
title, resurs, 255
tjänst för underprocesskontroll, , *Se* SPC,
ToolTalk
 meddelande-daemon, 23, 26
 program, funktioner för, 197
ToolTalk Message Server, , *Se* ttsession,
ToolTalk, *Se* rpc.ttdbserver
 Databasserver,,
ToolTalk, *Se* TT_MSG, funktion
 funktion,,
topShadowColor, resurs, 277
tset, 37
ttsession, 116
 starta, 27
tty inställningar, 38
tty-inställningar, 38
ttyModes, 38
TT_MSG, funktion
 nyckelord, 198
 skapa, 197
TYPE, fält, 239
TZ, variabel, 17, 18

U

UID, 109

- UNIX-tangentkopplingar, 265, 266
- UNIXbindings, fil, 267
- uppdateringsintervall för utskriftsjobb, 127
- USER, variabel, 17
- user-prefs.dt, fil, 52
- userPath, resurs, 17
- /usr/dt, 84
- utdatavariabler, 132
- utskrift
 - administration, 125
 - begrepp, 129
 - standarddestination, 128
 - testa, 110
- utskrift av brev, anpassa, 96
- utskrift pågår
 - olika datatyper, 129
- Utskriftshanteraren, 126
 - intervall för jobbuppdatering, 127

V

- /var/dt, 85
- variabler
 - i funktionsdefinitioner, 193
 - miljö, se miljövariabler, 30
- verifieringsmapp, 109, 115
- Väljaren, 91, 92
- välkomstmeddelande
 - standard, 11
 - Visar, 23
 - ändra, 11
- välkomstmeddelandet
 - anpassa, 25
 - visar, 25
- värddator för exekvering
 - ange, 120
 - konfigurera, 121
- växlingsområde
 - antal arbetsytor, 245
- växlingsområde, *Se* Område för växling av arbetsyta,

W

- windowMenu, resursen, 256
- WINDOW_TYPE, fält, 188
- wmStartupCommand, resurs, 29
- WM_CLASS, egenskap, 220

- workspaceCount-resursen, 254
- writeXrdbColors, resurs, 28
- wscon systemfönstrets loggningsfil, 36

X

- X Logical Font Description, 271
- X terminaler
 - få sessionstjänster, 105
- X-resurser, , *Se* resurser,
- X-server
 - ändra behörighet, 13
 - ändra miljö, 14
- X-terminaler, 91, 94, 112
 - använda, 92
 - CHOOSER, sträng, 8
 - möjliga konfigurationer av
 - inloggningsservrar, 3
 - skärmar utan XDMCP, 6
 - som arbetsstationer, 94
 - Xaccess, lista, 7
 - XDMCP direkt, 6, 7
 - XDMCP indirekt, 6, 7
- X-verifiering, 111
- X11 server, 86
- X400_TYPE, fält, 206
- Xaccess, fil, 7
- XAUTHORITY, variabel anges av
 - inloggningshanteraren, 16
- Xconfig, 85, 89
- Xconfig, fil
 - ange resurser i, 12
 - ändra, 3
- ?Xconfig file
 - ange språk med, 286
- .Xdefaults, 37, 38
- .Xdefaults, fil, 28, 266
- XDM, 85, 92, 95
 - protokoll, 92
- XDMCP, 2
 - definition, 6
 - direkt förfrågan, 6
 - direkt åtkomst, 7
 - frågeläge, 6
 - indirekt förfrågan, 6, 19
 - indirekt åtkomst, 8
- Xerrors, fil, 8

- Xfailsafe, fil, 15, 17, 18
- xlsfonts, kommando
 - installation, 288
 - lista teckensnitt på servern, 288
- XMICONBMSEARCHPATH, variabel
 - användning, 140
 - definition, 132
 - sammansättning, 141
- XMICONSEARCHPATH, variabel
 - användning, 140
 - definition, 132
 - sammansättning, 141
- XmText*FontList, resurs, 272
- Xpid, fil, 3
- Xreset, fil, 16
- Xresources, fil, 10, 11
- Xserver, 95
 - avbryta, 87
- Xservers, fil, 85
 - hantera lokal skärm, 18
 - standard, 4
 - starta en server, 3
 - syntax, 4

- Xsession
 - skript, 35
- Xsession, fil, 24
 - körs av inloggningsservern, 15
 - starta sessionshanteraren, 23
 - systemtäckande anpassning, 24
 - ändra PATH, 17
- Xsession.d, mapp, 23, 30
 - anpassa, 24
 - skript i, 24
- Xsetup, fil, 15
- Xstartup, fil, 15
- xterm, 38
 - starta, 88
- XUSERFILESEARCHPATH-variabeln, 288

Special Characters

- Återställ, funktion på Frontpanelen, 226
- äktthetskontroll, inloggning, 35
- översättningstabell, återställa, 292
- övervakning, kontrolltyper för, 242