

HP DECwindows Motif for OpenVMS

管理ガイド

2006年10月

本書では、HP DECwindows Motif for OpenVMSソフトウェアの設定および管理方法について説明します。

改訂/更新情報:

本書は、『日本語 DECwindows Motif for OpenVMS 環境設定の手引き』の代わりとなるドキュメントです。

オペレーティング・システム: OpenVMS I64 V8.3

OpenVMS Alpha V8.3

ソフトウェア・バージョン: DECwindows Motif for OpenVMS I64 V1.6

DECwindows Motif for OpenVMS Alpha V1.6

© Copyright 2006 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

本書の著作権は Hewlett-Packard Development Company, L.P. が保有しており、本書中の解説および図、表は Hewlett-Packard Development Company, L.P. の文書による許可なしに、その全体または一部を、いかなる場合にも再版あるいは複製することを禁じます。

また、本書に記載されている事項は、予告なく変更されることがありますので、あらかじめご承知おきください。万一、本書の記述に誤りがあった場合でも、弊社は一切その責任を負いかねます。

本書で解説するソフトウェア (対象ソフトウェア) は、所定のライセンス契約が締結された場合に限り、その使用あるいは複製が許可されます。

日本ヒューレット・パカードは、弊社または弊社の指定する会社から納入された機器以外の機器で対象ソフトウェアを使用した場合、その性能あるいは信頼性について一切責任を負いかねます。

Intel および Itanium は、米国ならびにその他の国における、Intel Corporation またはその関連会社の商標または登録商標です。

Java は、米国における Sun Microsystems, Inc. の商標です。

UNIX は、The Open Group の登録商標です。

X デバイスは、英国ならびにその他の国における、X/Open Company Ltd. の商標です。

原典： HP DECwindows Motif for OpenVMS Management Guide
© Copyright 2006 Hewlett-Packard Development Company, L.P.

ZK6300

目次

まえがき	ix
1 システムの概要	
1.1 DECwindows のクライアント/サーバ処理モデル	1-2
1.1.1 クライアント	1-3
1.1.2 トランスポート	1-3
1.1.3 ディスプレイ・サーバ	1-3
1.2 オプションのサーバ構成	1-4
1.2.1 フォント・サーバ	1-4
1.2.2 プロキシ・サーバ	1-5
1.3 DECwindows Motif の構成要素	1-7
1.3.1 レイヤード・プロダクトの構成要素	1-7
1.3.2 オペレーティング・システムの構成要素	1-7
2 DECwindows Motif の起動	
2.1 スタートアップ・プロシージャの説明	2-2
2.2 DECW\$STARTUP.COM プロシージャの使用法	2-6
2.3 システム・スタートアップの留意事項	2-7
2.3.1 DECwindows システム論理名の定義	2-7
2.3.2 システム・パラメータの調整	2-7
3 ディスプレイ・サーバの構成	
3.1 DECwindows ディスプレイ・サーバのカスタマイズ	3-2
3.1.1 DECW\$PRIVATE_SERVER_SETUP ファイルの使用法	3-2
3.1.2 ディスプレイ・サーバのカスタマイズ・パラメータ	3-2
3.1.2.1 サーバ・プロセス	3-4
3.1.2.2 拡張	3-7
3.1.2.3 セキュリティ	3-10
3.1.2.4 デバイス	3-12
3.1.2.5 トランスポートおよびネットワーク接続	3-18
3.1.2.6 フォント	3-19
3.1.2.6.1 フォント・パス	3-19
3.1.2.6.2 フォント・キャッシュ	3-22
3.1.2.6.3 フォント・スケーリング	3-22
3.1.2.7 キーボード	3-23
3.1.2.8 マウス	3-26
3.1.2.9 カラー・データベース	3-27
3.1.2.10 スクリーン・セーブ	3-27
3.1.2.11 パッキング・ストアとセーブ・アンダ	3-28
3.1.2.12 エラー報告	3-30
3.1.3 クラスタ共通パラメータの設定	3-34

3.1.4	スタンドアロン・システム・パラメータの設定	3-34
3.1.5	現在のサーバ・パラメータの確認	3-35
3.1.5.1	サーバ論理名テーブルの表示	3-35
3.1.5.2	X Display Information ユーティリティ (xdpyinfo) の利用	3-37
3.1.5.3	X 設定ユーティリティ (xset) の利用	3-39
3.2	ネットワーク・トランスポートの指定	3-40
3.2.1	ローカル・トランスポートの利用	3-40
3.2.2	DECnet トランスポートの利用	3-41
3.2.3	TCP/IP トランスポートの利用	3-41
3.2.4	LAT トランスポートの利用	3-41
3.2.5	デフォルト・トランスポートの変更	3-42
3.3	サーバ・アクセス制御の設定	3-42
3.3.1	ユーザ・ベースのアクセス制御	3-42
3.3.2	トークン・ベースのアクセス制御	3-43
3.3.2.1	Magic Cookie	3-43
3.3.2.2	Kerberos	3-44
3.3.3	X authority ファイル	3-45
3.3.4	アクセス許可ファイル	3-46
3.3.5	アクセス信頼ファイル	3-47
3.3.6	アクセス制御方法の選択	3-47
3.3.6.1	デスクトップ・セッション外の接続にアクセス制御を適用する	3-48
3.3.6.2	デスクトップ・セッション内の接続にアクセス制御を適用する	3-48
3.3.7	ユーザ・ベースのアクセス制御を有効にする	3-48
3.3.8	Magic Cookie によるアクセス制御を有効にする	3-51
3.3.9	Kerberos によるアクセス制御を有効にする	3-52
3.3.10	セキュリティ拡張の利用	3-55
3.3.10.1	セキュリティ拡張を有効にする	3-56
3.3.10.2	セキュリティ・ポリシー・ファイルの利用	3-56
3.4	マルチヘッド・システムの設定	3-56
3.4.1	システム設定	3-57
3.4.2	単純なマルチヘッド・システムの構成	3-57
3.4.3	XINERAMA を使用したマルチヘッド・システムの構成	3-57
3.4.3.1	ハードウェアおよび構成の要件	3-58
3.4.3.2	XINERAMA 拡張を有効にする	3-58
3.4.3.3	モニタの配置	3-58
3.5	デフォルトのキーボード・レイアウトの変更	3-59
3.5.1	DECwindows キーマップ・ファイルの利用	3-59
3.5.2	X Keyboard キーマップ・ファイルの利用	3-60
3.5.2.1	X Keyboard のコンポーネント・データベース	3-61
3.5.2.2	X Keyboard のキーマップ・ファイルの作成	3-63
3.5.2.3	コンパイル済みキーマップ・ファイルをロードする	3-64
3.5.2.4	AccessX キー機能を有効にする	3-64
3.6	新しいフォントの指定	3-65
3.6.1	他社製のフォントの使用	3-65
3.6.2	ユーロ通貨記号のサポートを有効にする	3-66
3.6.2.1	DECwindows Motif アプリケーションでのユーロ記号の表示	3-67
3.6.2.2	キーボードを使ったユーロ記号の入力 (Alpha のみ)	3-67
3.6.3	フォント・サーバのサポートを有効にする	3-67

3.7	LBX プロキシ・サーバの設定	3-68
3.7.1	LBX 拡張を有効にする	3-69
3.7.2	プロキシ・サーバの起動	3-69
3.7.2.1	プロキシ・サーバ環境での認証	3-69
3.7.2.2	管理対象プロキシ・サーバの起動	3-70
3.7.2.3	スタンドアロン・プロキシ・サーバの起動	3-70
3.7.3	IPv6 環境でのプロキシ・サーバの利用	3-71
3.7.4	デフォルトのプロキシ・サーバ・プロセス属性の変更	3-71
3.7.5	プロキシ・サーバの停止	3-72
3.7.5.1	自動停止	3-72
3.7.5.2	手動停止	3-72
3.7.6	プロキシ・マネージャ構成ファイル	3-72
3.7.7	プロキシ・マネージャの起動	3-73
3.7.7.1	DECwindows Motif のスタートアップ時に自動的に起動する	3-74
3.7.7.2	手動起動	3-75
4	DECwindows の使用方法	
4.1	ディスプレイの設定	4-2
4.1.1	ディスプレイ名の形式	4-2
4.1.2	TCP/IP ホスト名の変換	4-5
4.2	ログイン・プロセスの説明	4-5
4.2.1	New Desktop のログイン・シーケンス	4-5
4.2.2	Traditional DECwindows Desktop のログイン・シーケンス	4-7
4.3	ログイン環境のカスタマイズ	4-8
4.3.1	アプリケーションのスタートアップ性能の向上	4-9
4.3.2	ログイン画面のカスタマイズ (Traditional DECwindows Desktop のみ)	4-9
4.3.2.1	ロゴおよびログイン画面の色のカスタマイズ	4-10
4.3.2.2	「Start Session」ダイアログ・ボックスと「Set Password」ダイアログ・ボックスの位置の変更	4-10
4.3.2.3	「Start Session」ダイアログ・ボックスでのノード名表示を無効にする	4-11
4.3.3	ログイン前にカスタム・メッセージを表示する (New Desktop のみ)	4-11
4.3.4	推奨パスワード・リストを無効にする (New Desktop のみ)	4-12
4.3.5	UNIX スタイルのファイル名のサポートを有効にする (New Desktop のみ)	4-12
4.3.5.1	「File Selection」ダイアログ・ボックスで有効にする	4-12
4.3.5.2	ファイル・マネージャで有効にする	4-13
4.4	スタートアップ環境のカスタマイズ	4-13
4.4.1	DECW\$PRIVATE_APPS_SETUP ファイルの利用	4-13
4.4.2	デスクトップの切り替え	4-16
4.4.3	IPv6 サポートを有効にする	4-17
4.4.4	デフォルトのロゴの変更 (Traditional DECwindows Desktop のみ)	4-18
4.4.5	個人使用ライセンスを使用したシステムでのロゴの表示 (Traditional DECwindows Desktop のみ)	4-19
4.4.6	コンソール・メッセージの表示	4-19

4.4.7	専用アカウントの作成 (Traditional DECwindows Desktop のみ)	4-21
4.4.7.1	セッション・マネージャのコマンド・プロシーダを変更する	4-21
4.4.7.2	セッション・マネージャの実行可能ファイルを変更する	4-23
4.4.7.3	セッション・マネージャのプロファイル・ファイルを変更する	4-25
4.4.8	カスタム Bookreader ディレクトリの作成	4-27
4.5	セッション・マネージャの動作の変更 (Traditional DECwindows Desktop のみ)	4-28
4.6	システム・リソース・ファイルの変更	4-28
4.7	クライアント・アクセス制御の指定	4-29
4.7.1	セキュリティ・オプションの設定	4-30
4.7.2	セッションの最中にセキュリティ・オプションを更新する	4-30
4.7.3	ログイン時のアクセス制御を有効または無効にする	4-31
4.7.4	信頼ユーザによる一時停止状態のデスクトップ・セッションのアンロックを有効にする	4-32
4.8	プリント・フォーマットのカスタマイズ	4-32
4.8.1	プリント・フォーマットの定義	4-32
4.8.2	論理名とプリント・フォーマット	4-33

A DECwindows システムのチューニング

A.1	DECwindows アプリケーション向けの UAF パラメータの設定	A-1
A.2	DECwindows アプリケーションおよびディスプレイ・サーバ用のシステム・パラメータの設定	A-2
A.3	非 VGA デバイス向けのサーバ・パラメータの設定	A-4

B DECwindows Motif キーマップ名

索引

例

3-1	トランスポート接続の設定例	3-42
3-2	オランダ語キーボードと 100 dpi モニタの設定例	3-60
4-1	プライベート・セッション・マネージャ・コマンド・ファイルの使用例	4-22
4-2	セッション・マネージャの実行可能ファイルを変更する	4-24

図

1-1	DECwindows のシステム・アーキテクチャ: 基本	1-2
1-2	DECwindows のシステム・アーキテクチャ: フォント・サーバ	1-5
1-3	DECwindows のシステム・アーキテクチャ: プロキシ・サーバ	1-6
2-1	DECwindows のスタートアップ・コマンド・プロシーダの流れ	2-2

表

2-1	スタートアップ・パラメータの値	2-6
2-2	システム論理名	2-7
3-1	DECwindows ディスプレイ・サーバのカスタマイズ・パラメータ	3-3
3-2	ロード可能なディスプレイ・サーバ拡張	3-8
3-3	組み込みディスプレイ・サーバ拡張	3-9
3-4	LBXPROXY プロセス属性論理名	3-71
3-5	プロキシ・マネージャを制御するグローバル・シンボル	3-74
4-1	サポートされている DECwindows Motif のトランスポート名	4-3
4-2	サポートされている IPv6 ホスト名とアドレスの形式	4-4
4-3	サポートされる IPv4 ホスト名とアドレスの形式	4-4
4-4	ロゴの移動とログイン画面の色の変更	4-10
4-5	「Start Session」ダイアログ・ボックスと「Set Password」ダイアログ・ボックスの位置の変更	4-10
4-6	変更可能なクライアント領域	4-13
4-7	セッション・マネージャの論理名	4-28
4-8	「Print」ダイアログ・ボックスの論理名	4-33
A-1	推奨される UAF リミットの設定値	A-1
A-2	Alpha システムでのシステム・パラメータの推奨される設定	A-2
A-3	I64 システムでのシステム・パラメータの推奨される設定	A-3
A-4	非 VGA デバイス向けの最小サーバ制限値	A-5
B-1	DECwindows Motif キーマップ	B-1

対象読者

本書は、OpenVMS I64 プラットフォームまたは OpenVMS Alpha プラットフォームを使用したクラスタ・システムまたはスタンドアロン・システム上で、HP DECwindows Motif for OpenVMS (DECwindows Motif) ソフトウェアの管理とカスタマイズを行う OpenVMS システム管理者を対象としています。

本書の構成

本書の構成は以下のとおりです。

- 第 1 章では、DECwindows Motif のアーキテクチャの概要とさまざまな構成要素について説明します。
- 第 2 章では、DECwindows Motif のスタートアップ処理と、スタートアップ・コマンド・プロシージャのカスタマイズ方法および調整方法について説明します。
- 第 3 章では、サーバ・プロセスの動作、デバイスの設定、フォントの設定、バッキング・ストア、拡張、エラー・レポートのカスタマイズなど、シンボルを使用した X ディスプレイ・サーバの設定方法について説明します。また、サーバのアクセス制御の設定、代替トランスポート手段を指定する方法、フォント・サーバとプロキシ・サーバを有効にする方法、マルチヘッド・ディスプレイを設定する方法についての例も記載しています。
- 第 4 章では、クライアントのログイン手順と、セッション・マネージャ環境をカスタマイズする方法について説明します。これには、クライアント・アプリケーションが使用するアクセス制御方法を指定して管理する方法、代替ロゴを指定する方法、専用のアカウントを作成する方法、プリント・フォーマットをカスタマイズする方法が含まれます。
- 付録 A では、UAF リミットと OpenVMS システムでのシステム・パラメータについて、推奨される最低の設定を示します。また、非 VGA 構成におけるチューニング時の推奨事項も示します。
- 付録 B では、DECwindows Motif でサポートされているすべてのキーマップを示します。この付録は、アルファベット順に並んでおり、言語ごとにグループ分けされています。

関連資料

OpenVMS や DECwindows Motif 製品とサービスに関するその他の情報については、次の Web サイトをご覧ください。

<http://www.hp.com/jp/openvms>

<http://www.hp.com/go/openvms>

本書で使用する表記法

本書では以下の製品名が使用されています。

- HP OpenVMS Industry Standard 64 for Integrity サーバ
- OpenVMS I64
- I64

これら 3 つの名前 (長い形式と 2 つの省略形式) は、どれも Intel Itanium アーキテクチャ上で動作する OpenVMS オペレーティング・システムを指しています。

DECwindows および DECwindows Motif は、HP DECwindows Motif for OpenVMS ソフトウェアを指します。また、X サーバと X ディスプレイ・サーバは、すべて DECwindows X11 Display Server を指します。

本書では、以下の表記法を使用しています。

Ctrl/ <i>x</i>	Ctrl/ <i>x</i> という表記は、Ctrl キーを押しながら別のキーまたはポインティング・デバイス・ボタンを押すことを示します。
PF1 <i>x</i>	PF1 <i>x</i> という表記は、PF1 に定義されたキーを押してから、別のキーまたはポインティング・デバイス・ボタンを押すことを示します。
Return	例の中で、キー名が四角で囲まれている場合には、キーボード上でそのキーを押すことを示します。(本文中では、キー名は四角で囲まれていません。) HTML 形式のドキュメントでは、キー名は四角ではなく、大括弧で囲まれています。
...	例の中の水平方向の反復記号は、次のいずれかを示します。 <ul style="list-style-type: none">• 文中のオプションの引数が省略されている。• 前出の 1 つまたは複数の項目を繰り返すことができる。• パラメータや値などの情報をさらに入力できる。
.	垂直方向の反復記号は、コードの例やコマンド形式の中の項目が省略されていることを示します。このように項目が省略されるのは、説明している内容にとってその項目が重要ではないからです。

()	コマンドの形式の説明において、括弧は、複数のオプションを選択した場合に、選択したオプションを括弧で囲まなければならないことを示しています。
[]	コマンドの形式の説明において、大括弧で囲まれた要素は省略可能です。オプションをすべて選択しても、いずれか1つを選択しても、あるいは1つも選択しなくても構いません。ただし、OpenVMS ファイル指定のディレクトリ名の構文や、代入文の部分文字列指定の構文の中では、大括弧は省略できません。
[]	コマンド形式の説明において、大括弧内の要素を分けている縦線はオプションを1つまたは複数選択するか、または何も選択しないことを意味します。
{ }	コマンドの形式の説明において、中括弧で囲まれた要素は必須オプションです。いずれか1つのオプションを選択しなければなりません。
太字テキスト	太字のテキストは、新しい用語、引数名、属性、条件を示しています。 オンライン版のマニュアルでは、ユーザ入力を示すためにも太字が使用されます。
<i>italic text</i>	イタリックのテキストは、重要な情報を示したり、変数やマニュアルの完全なタイトルを示します。また、システム・メッセージ (たとえば内部エラー <i>number</i>)、コマンド行 (たとえば <i>/PRODUCER=name</i>)、文中のコマンド・パラメータなどの変化する部分を示す場合にも使用されます。
Monospace type	モノスペース・タイプの文字は、コード例および会話型の画面表示を示します。 C プログラミング言語では、テキスト中のモノスペース・タイプの文字は、キーワード、別々にコンパイルされた外部関数およびファイルの名前、構文の要約、または例に示された変数や識別子の実際の文字列などを示します。
UPPERCASE TEXT	英大文字のテキストは、コマンド、ルーチン名、ファイル名、システム特権の短縮形を示します。
-	コード例中のハイフンは、要求に対する引数とその次の行に続くことを示します。
数	特に明記しない限り、本文中の数はすべて10進数です。10進数以外(2進数、8進数、16進数)のときは、その旨を明記してあります。

システムの概要

この章では、HP DECwindows Motif for OpenVMS (DECwindows Motif) ソフトウェアの概要について説明します。以下のトピックについて説明します。

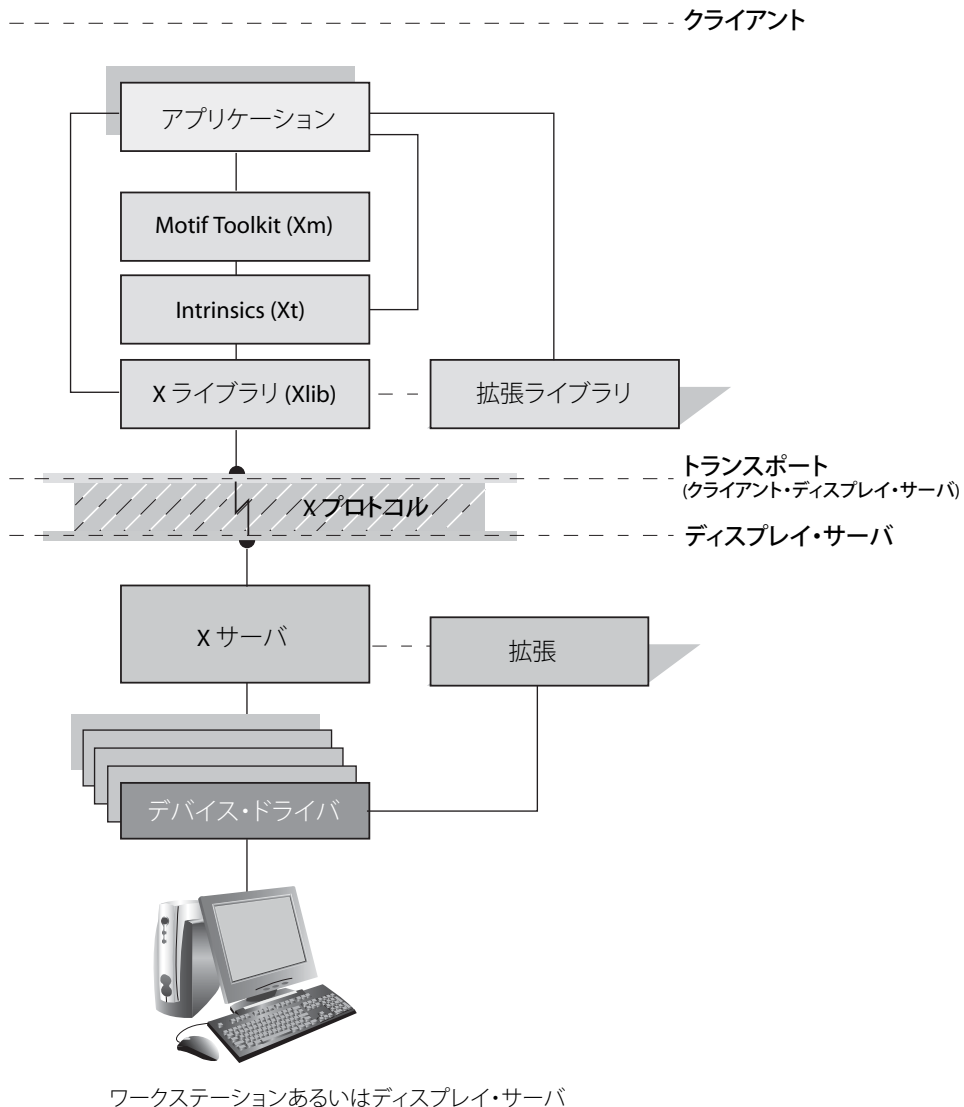
- DECwindows Motif の処理モデルと、基本的なシステム・アーキテクチャ
- オプションのフォント・サーバとプロキシ・サーバの構成
- クライアントと X ディスプレイ・サーバ・ソフトウェアの構成要素

1.1 DECwindows のクライアント/サーバ処理モデル

DECwindows Motif ソフトウェアでは、クライアント/サーバの処理モデルが採用されています。サーバは、1つの共有プロセスであり、多数のクライアント・プロセスの要求に基づいて処理を行います。

図 1-1 に、DECwindows Motif のクライアントとディスプレイ・サーバの基本的なアーキテクチャを示します。

図 1-1 DECwindows のシステム・アーキテクチャ: 基本



ZK-5222A-AI

ほとんどのクライアント/サーバの関係では、クライアント・システムはデスクトップにあり、サーバ・システムはネットワーク上にあります。DECwindows Motif で

は、すべての X ウィンドウ・システム環境と同じく、サーバ・システムはデスクトップにあり、画面上にグラフィックスを表示します。

1.1.1 クライアント

クライアントは、デスクトップ・アプリケーションや X ウィンドウ・システム・ユーティリティなどのプロセスであり、X プロトコル・リクエストを送信します。たとえば、DECwindows Motif 環境では、デスクトップ・アプリケーション (DECterm など) や X ウィンドウ・システムのユーティリティ (xlsfonts など) は、X ディスプレイ・サーバと通信するクライアントです。

クライアントは、ディスプレイ・サーバ・システム上の表示内容を制御し、ユーザと対話するためのグラフィック・インタフェースを生成します。

1.1.2 トランスポート

DECwindows アーキテクチャでは、ほとんどのクライアント/サーバ処理モデルと同様に、クライアントとディスプレイ・サーバはそれぞれ別のシステムに存在できます。これらのシステムは、ユーザからは見えないネットワーク・トランスポートで互いに接続されます。

トランスポートは、クライアントとサーバ・システム間でのデータのやり取りだけを担当します。データの変更は一切行いません。

DECwindows Motif では、以下の伝送手段がサポートされています。

- ローカル (共用メモリ)
- LAT (Local Area Transport)
- DECnet
- IPv4 (Internet Protocol Version 4) または IPv6 (Internet Protocol Version 6) のホスト名とアドレス形式を使用した TCP/IP

クライアントとサーバは、ネットワーク・トランスポートに対する固有のインタフェースをそれぞれ維持管理しています。

1.1.3 ディスプレイ・サーバ

ディスプレイ・サーバは、クライアント・アプリケーションが、サポートされているデバイスと一貫した方法でやり取りできるようにします。ディスプレイ・サーバは、クライアント・アプリケーションに代わって物理的なグラフィックス・ディスプレイと周辺デバイスを管理します。トランスポート層を通じてクライアント・アプリケーションからの X プロトコル・リクエストを受信し、特定のデバイスに対する要求を満たすために必要な機能を実行します。

システムの概要

1.1 DECwindows のクライアント/サーバ処理モデル

基本的に、サーバは、要求を表すデータを該当するグラフィックス・デバイスが実行できるコマンドに変換します。ユーザが入力デバイス(マウス、キーボード、タッチパッドなど)を使用してアプリケーション・データを入力すると、ディスプレイ・サーバがデバイス・ドライバから入力を受信し、トランスポート層を通じてプロトコル・パケットを X ライブラリ (Xlib) および X Toolkit Intrinsics (Xt) ルーチンに返します。

正常に通信するためには、クライアントとディスプレイ・サーバ間でアクセス制御方法、通信プロトコル、ホスト名の形式などの設定に関して、互換性が必要となります。

1.2 オプションのサーバ構成

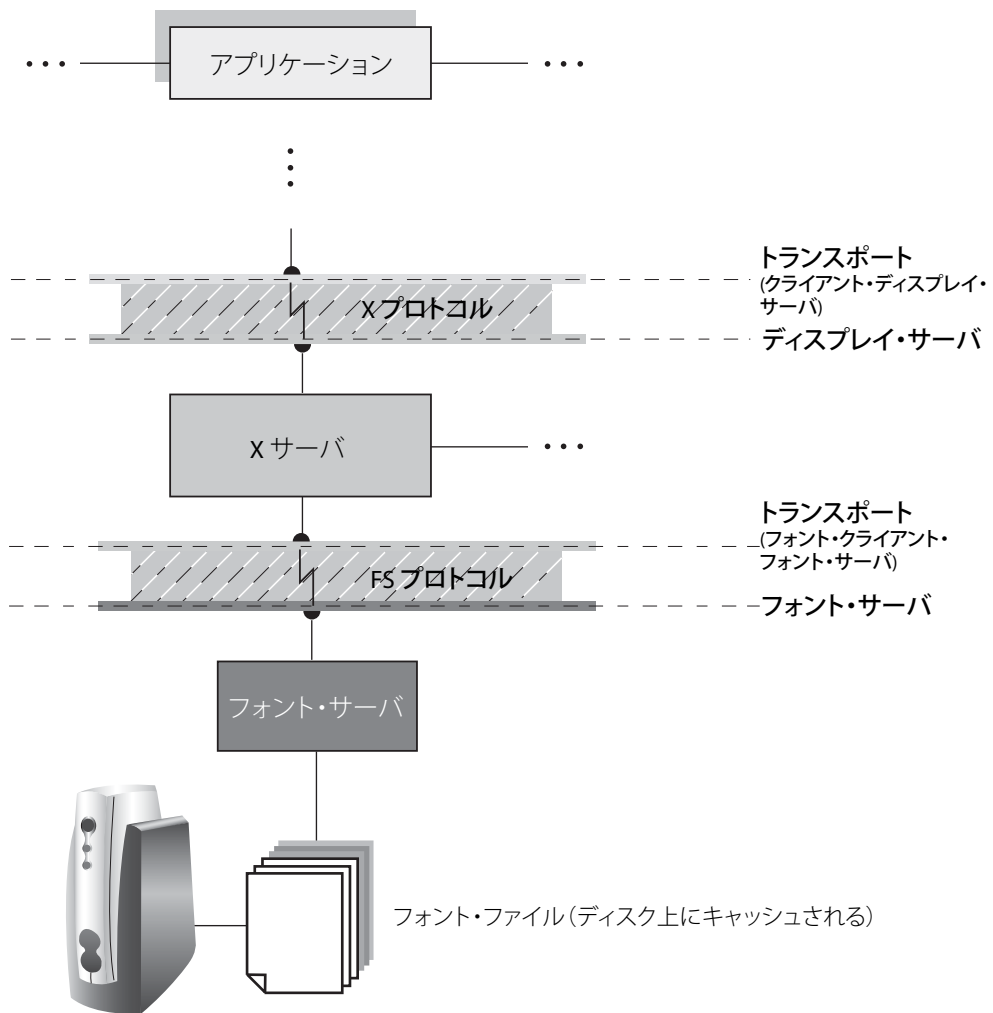
以降の項で説明するように、基本システム・アーキテクチャを拡張して、他の種類のサーバを含めることができます。

1.2.1 フォント・サーバ

X ディスプレイ・サーバに組み込まれているフォント・レンダラでは、1つ以上のフォント・サーバを使用して、ディスプレイ・サーバが動作しているシステム以外のシステムに分散されたフォント・ファイルにアクセスすることができます。ディスプレイ・サーバとフォント・サーバの間の通信では、X フォント・サーバ (FS) プロトコルが使用されます。

図 1-2 に、フォント・サーバを追加した DECwindows Motif アーキテクチャを示します。

図 1-2 DECwindows のシステム・アーキテクチャ: フォント・サーバ
----- クライアント



VM-1162A-AI

1.2.2 プロキシ・サーバ

プロキシ・サーバは、クライアントからは他の X サーバと同じに見えます。プロキシ・サーバはクライアントからの接続要求を受け付け、クライアントと X サーバ間の仲介者として動作します。プロキシ・サーバとクライアント間の通信では、標準の X プロトコルが使用されます。プロキシと X サーバ間の通信では、LBX (Low-Bandwidth X) プロトコルが使用されます。

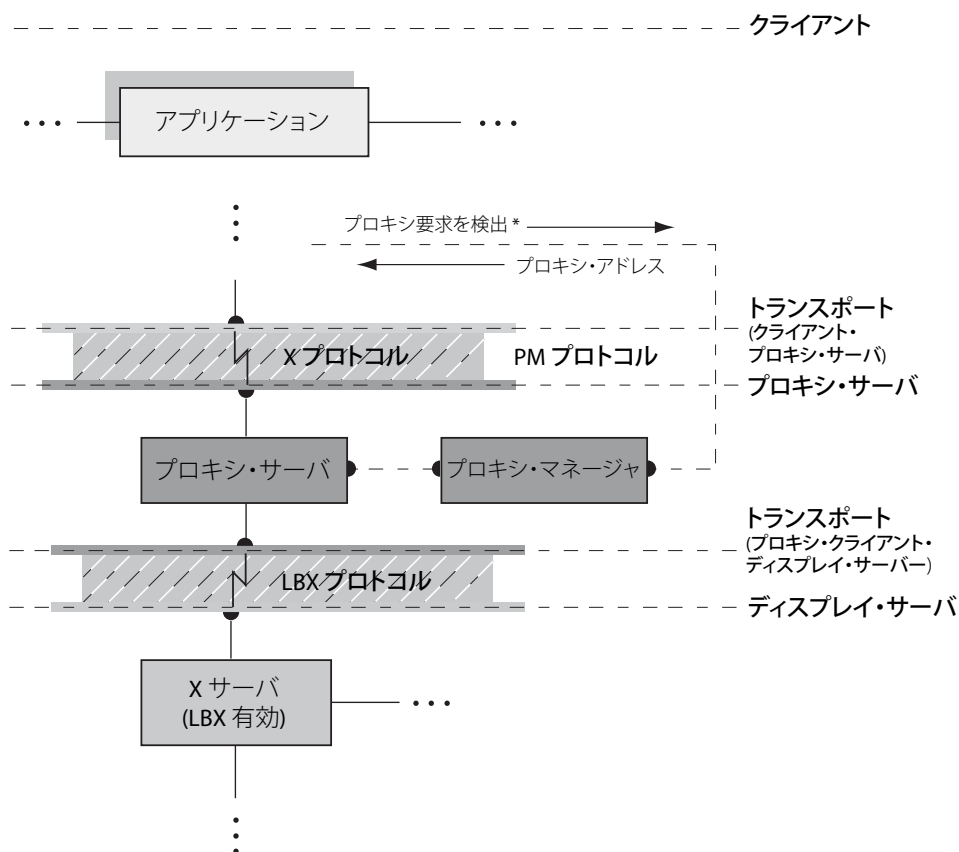
LBX は、ディスプレイ・サーバとクライアントが、56K bps のダイヤルイン・モデムやワイド・エリア・ネットワーク (WAN) などの低速回線で接続されているような構成用に設計されています。X プロトコルの開発当初は、主にローカル・エリア・ネットワーク (LAN) 上で使用されており、低速の接続用に最適化されていませんでした。

システムの概要
1.2 オプションのサーバ構成

た。LBXでは、クライアントとサーバ間のデータ・フローを最小化するように設計された圧縮とキャッシュを使用することでこの欠点に対処しています。

図 1-3 に、プロキシ・サーバを追加した DECwindows Motif サーバ・アーキテクチャを示します。

図 1-3 DECwindows のシステム・アーキテクチャ: プロキシ・サーバ



* 通常は SET DISPLAY コマンドで DECW\$DISPLAY 論理名をリフレッシュして行う。
この接続は一時的なもので、プロキシ・サーバのアドレスを見つけるためだけに使用される。

VM-1083A-AI

プロキシ・サーバは、プロキシ・マネージャ・アプリケーションで管理することもできます。クライアント・アプリケーションは、X デisplay・サーバへの要求をプロキシ・マネージャに渡します。プロキシ・マネージャは、適切な既存のプロキシ・サーバを探るか、プロキシ・サーバの新しいインスタンスを自動的に起動します。

1.3 DECwindows Motif の構成要素

ここでは、レイヤード・プロダクト・ソフトウェア DECwindows Motif を構成するクライアントの構成要素、ディスプレイ・サーバ、OpenVMS オペレーティング・システムで提供されている共通の構成要素を示します。

1.3.1 レイヤード・プロダクトの構成要素

DECwindows Motif クライアント・ソフトウェアを構成するコンポーネントは以下のとおりです。

- デスクトップ・アプリケーション
- X ウィンドウ・システム・ユーティリティ
- セッション管理ユーティリティ
- プログラミング・ライブラリ
 - X ライブラリ (Xlib)
 - X Toolkit Intrinsic (Xt)
 - OSF/Motif ツールキット (Xm)
 - HP Extensions to Motif (DXm)
 - ICE (Inter-Client Exchange) および XSMP (X Session Management) 拡張
- サンプル・プログラムとサンプル・ウィジェット

1.3.2 オペレーティング・システムの構成要素

以下の共通コンポーネントとディスプレイ・サーバ・コンポーネントが、OpenVMS オペレーティング・システムで提供されています。

- X ディスプレイ・サーバ

以下の共有イメージで構成されます。

メイン・エントリ・ポイント・スタブ
デバイス非依存サーバ (DIX) イメージ
デバイス依存サーバ (DDX) イメージ
動的にロード可能な拡張

メインのイメージと DIX イメージは一緒にリンクされます。初期化の際に、どのグラフィックス・デバイスが使用可能で選択されているかに応じて、DDX イメージが動的にアクティブ化されます。ロード可能な拡張は、オペレーティング・システムの初期化時、または最初に使用したときに動的にアクティブ化されます。

- トランスポート・インタフェース

単一の共有イメージと、トランスポート固有のイメージの集合で構成されます。

システムの概要

1.3 DECwindows Motif の構成要素

共通のトランスポート・イメージはディスプレイ・サーバに組み込まれています。トランスポートの初期化パラメータに応じて、必要なトランスポート固有イメージが動的にアクティブ化されます。

- デバイス・ドライバ

DECwindows ディスプレイ・サーバは、入力ドライバとグラフィックス/ビデオ・ドライバの 2 種類のドライバを使用します。

- データ・ファイル

ディスプレイ・サーバは、以下の種類のデータ・ファイルを参照します。

X authority およびアクセス・リスト (サーバ・システムへのアクセスを制御)

フォント (詳細なフォント記述を格納)

キーマップ (キーボードの各キーの解釈方法を定義)

カラー・データベース (カラー名と RGB 値の対応を定義)

DECwindows Motif の起動

この章では、DECwindows システムのスタートアップ処理について説明します。スタートアップ処理は、最初の DECwindows のスタートアップ・コマンド・ファイルを実行してから DECwindows のログイン・ダイアログ・ボックスが表示されるまでの処理です。

この章では、以下のトピックについて説明します。

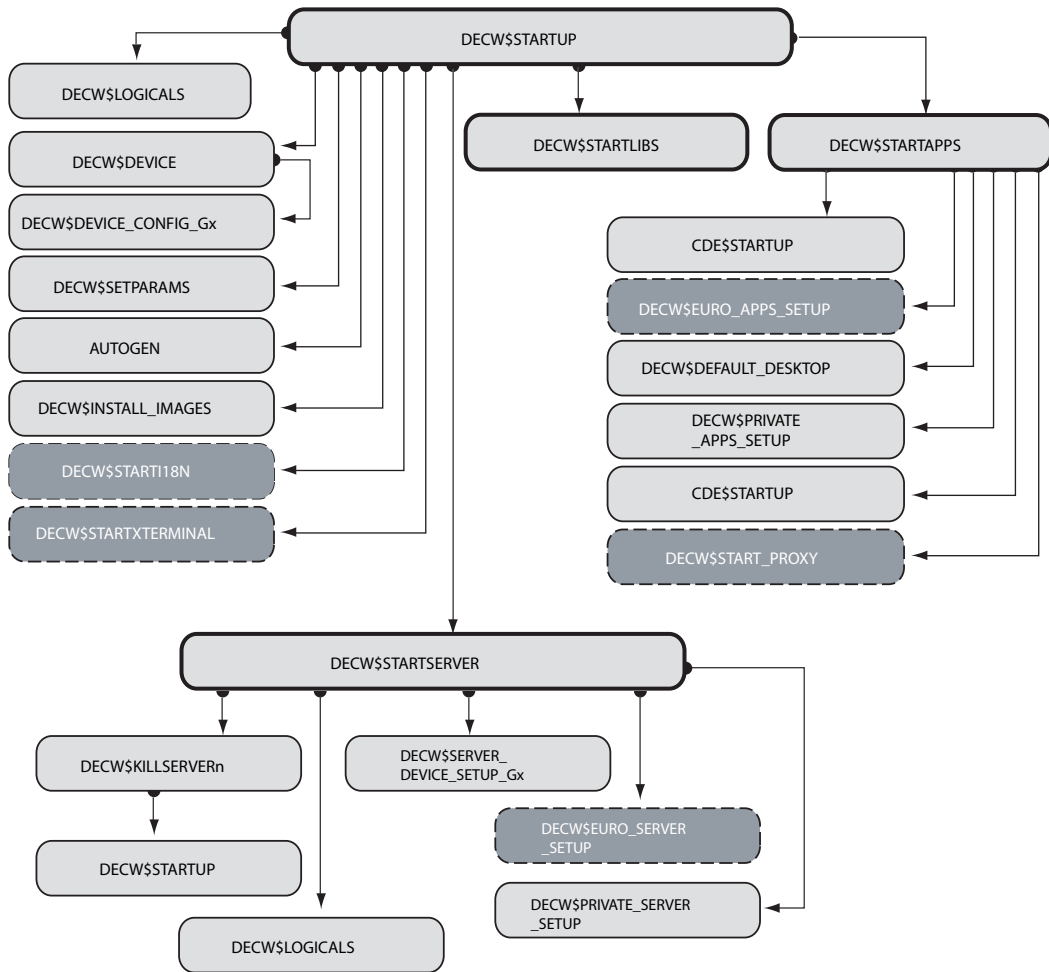
- スタートアップ・プロシージャの説明
- DECWS\$STARTUP.COM プロシージャの使用方法
- システム・スタートアップの留意事項

2.1 スタートアップ・プロセスの説明

ここでは、DECwindows Motif のスタートアップ処理について説明します。
 図 2-1 に、DECwindows Motif のスタートアップ・コマンド・プロセス
 DECW\$STARTUP を起動した時点からのスタートアップ・シーケンスを示します。

プロセスのいくつかはオプションです。たとえば、DECW\$STARTI18N は、
 OpenVMS 国際化 (I18N) キットがインストールされている場合にだけ起動されま
 す。次の図では、これらオプションのプロセスは、濃い灰色で表示されていま
 す。

図 2-1 DECwindows のスタートアップ・コマンド・プロセスの流れ



ZK-6831A-AI

1. DECW\$LOGICALS.COM: DECW\$LOGICAL_NAMES テーブルを作成します。

DECwindows アプリケーションのスタートアップ・パラメータと設定パラメータは、このステップで定義されます。クラス単位（データ・ファイル（キーマップやフォント・ファイルなど）の設定を個々のシステムで設定変更できるようにするための論理名 DECWSSYSCOMMON も定義されます。

2. DECW\$DEVICE.COM: DECwindows デバイス・ドライバをロードして設定します。

このプロセスは、DECwindows のスタートアップ・シーケンスの以降の処理で使用するシンボルも定義します。特定のデバイス・ドライバをロードする前に、DECW\$PRIVATE_SERVER_SETUP ファイル（存在する場合）の INIT 部分を実行し、システムが自動的なデバイス選択を変更できるようにします。

次に、ディスプレイ・サーバで使用できるグラフィックス・デバイスの一覧が格納されたシンボル DECW\$DEVICE を設定します。DECW\$DEVICE.COM はこの一覧を使用して、デバイス固有の設定プロセス DECW\$DEVICE_CONFIG_Gx (xは、GZA0 を示す Z など、デバイス・タイプ識別子) を起動します。

3. DECW\$SETPARAMS.COM: 必要なシステム・パラメータ値を設定します。

このプロセスは、DECwindows のスタートアップ時に必要なシステム・パラメータに適切な値を設定します。

正しく設定されていないシステム・パラメータがあると、変更が必要なパラメータの一覧が表示されます。DECW\$STARTUP.COM プロセスは、AUTOGEN を実行するかどうかを質問します。DECW\$STARTUP.COM を RESTART パラメータ付きで実行すると、質問はシステム・コンソールに表示されます。

この質問に NO と答えると、DECW\$STARTUP.COM は、システム・パラメータを変更するまで DECwindows が起動できないことを示すメッセージを表示して終了します。YES と答えると、DECW\$STARTUP.COM は AUTOGEN を GETDATA フェーズから REBOOT まで実行します。これによりシステム・パラメータが変更され、システムがリブートします。

システム・パラメータのチェックを省略するには、DECW\$STARTUP.COM を実行する前に論理名 DECW\$IGNORE_AUTOGEN を定義します（表 2-2 を参照）。

4. DECW\$INSTALL_IMAGES.COM: システム・イメージとトランスポート共有イメージをロードします。

デフォルトでは、以下の共有イメージがスタートアップ時に SYSSHARE にインストールされます。

DECW\$XPORT_SERVICES.EXE (メモリに常駐)
DECW\$LCNLIBSHR.EXE (メモリに常駐)
DECW\$XPORT_PTHREAD.EXE
DECW\$TRANSPORT_COMMON.EXE (メモリに常駐)
DECW\$TRANSPORT_DECNET.EXE
DECW\$TRANSPORT_LOCAL.EXE

DECW\$TRANSPORT_TCPIP.EXE
DECW\$SECURITY.EXE
DECW\$SETSHODIS.EXE
DECW\$SETSHODISSHR.EXE (メモリに常駐)
DECW\$XAUSHR.EXE (メモリに常駐)
DECW\$TRANSPORT_LAT.EXE

5. DECW\$STARTI18N.COM: 言語ロケールを設定します (オプション)。
DECwindows のインストール時に I18N オプションを選択した場合、システムのスタートアップ時に以下の機能が実行されます。
 - a. 言語固有のディレクトリの作成
 - b. デフォルトのシステム言語の指定
 - c. システム論理名を、言語固有の適切なディレクトリ指定を反映するように変更
 - d. 言語固有のスタートアップ・プロシージャ DECW\$STARTI18N_XX_XX.COM の実行。ここで XX_XX は言語ロケール (FR_FR など) を表します。
6. DECW\$STARTXTERMINAL.COM: X ターミナル・エミュレータのサポート・ファイルをインストールします (オプション)。
システム・スタートアップ・プロシージャ内で論理名 DECW\$INSTALL_XTERMINAL が定義されている場合は、以下の機能が実行されます。
 - a. LAT トランスポートを使用したクライアント・サポートの提供
 - b. LT ドライバと通信するための Xlib 用クラス・ドライバとして XTDRIVER を追加
 - c. DECW\$TRANSPORT_LAT.EXE をインストール
 - d. DECW\$FD プロセスを通じたフォント・ファイルの共用のための設定
 - e. フォント・デーモン・プロセスを独立プロセスとして実行
 - f. SYSNAM 特権を持つ DECnet アクセス・ゲートウェイ・サーバ・イメージをインストール
7. DECW\$STARTSERVER.COM: DECwindows ディスプレイ・サーバを起動します。
このプロシージャは以下の処理を行います。
 - a. DECW\$STARTUP プロシージャに対する RESTART オプションを処理します。RESTART を選択すると、DECW\$KILLSERVER n .COM という名前のコマンド・ファイルが作成され、“Server n Restart” というプロセス名で独立プロセスとして実行されます。このプロシージャは、現在のサーバ・プロセスを停止し、P1 にヌルを設定して DECW\$STARTUP.COM を実行します。
 - b. スタートアップ・ログ・ファイル SYSSMANAGER:DECW\$STARTUP_ n .LOG に進行状況を書き込み、論理名 DECW\$KEEPLIMIT で指定された値に基づいて以前のログをパージします。デフォルト値は 2 です。

- c. サーバ固有の論理名テーブル DECW\$SERVER_n_TABLE を作成します。
 - d. インストール時にユーロ・オプションを選択した場合には、ファイル SYSS\$MANAGER:DECW\$EURO_SERVER_SETUP.COM が実行されます。これにより、フォント・ファイルとキーマップ・ファイルに対するデフォルトのパスが、ユーロ対応のファイルに変更されます。
 - e. サーバ・カスタマイズ・ファイル DECW\$PRIVATE_SERVER_SETUP.COM を実行します。ディスプレイ・サーバをカスタマイズする方法については、第 3.1 節を参照してください。
 - f. DECW\$DEVICE.COM, DECW\$PRIVATE_SERVER_SETUP.COM, および DECW\$EURO_SERVER_SETUP.COM (存在する場合) で定義されたシンボルを基に、サーバ固有の論理名テーブルに値を設定します。
 - g. フォント・サーバが定義されている場合は、その存在を確認して起動します。
 - h. 以前のバージョンのサーバ・エラー・ログ・ファイル DECW\$SERVER_n_ERROR.LOG をパーズします。
 - i. サーバがクラッシュした場合にプロセス・ダンプ・ファイルを保存するかどうかを示す論理名 DECW\$SERVER_DUMP を確認します。この論理名を設定すると、サーバの条件ハンドラが無効になります。
 - j. ユーザ指定のサーバ・プロセス・パラメータを確認します。変更可能なサーバ・プロセス・パラメータの一覧については、第 3.1.2.1 項を参照してください。
 - k. サーバの実行可能ファイル名を決定します。通常は、SYSS\$SYSTEM:DECW\$SERVER_MAIN.EXE です。以下の条件を満たす場合は、代替サーバ・イメージが実行されます。
 - 論理名 DECW\$SERVER_MAIN に、代替サーバの実行可能ファイルが定義されている場合。
 - SYSS\$SYSTEM:DECW\$SERVER_MAIN_XX.EXE という名前のファイルが存在する場合。ここで XX は、シンボル DECW\$DEVICE で定義されたプライマリ・グラフィックス・デバイスの最初の 2 文字。代替サーバ・イメージは、バグ・フィックス・リリースや特殊ハードウェア・リリースの場合にのみ実行します。
 - l. 収集したデバイス名とプロセス・パラメータ情報を使用して、ディスプレイ・サーバ・イメージを独立プロセスとして実行します。
8. DECW\$STARTLIBS.COM: クライアントと共用ライブラリをロードして構成します。

このプロセスは以下の処理を実行します。

- a. DECwindows の論理名を定義する。
- b. Xlib, Xt などの共用ライブラリをインストールする。

- c. DCL コマンド SET DISPLAY 用の WSDRIVER を接続する。
9. DECW\$STARTAPPS.COM: ユーザ・アプリケーション開発環境を設定します。
- このプロシージャは、WSAn: デバイスを実行可能モードで作成し、「Start Session」ダイアログ・ボックスを作成します。カスタマイズ・コマンド・ファイル DECW\$PRIVATE_APPS_SETUP.COM および DECW\$EURO_APPS_SETUP.COM (ユーロ・オプションがインストールされている場合) を作成した場合は、ここで起動されます。このファイルは、ログイン・シーケンスをカスタマイズするために使用します。アプリケーションのスタートアップをカスタマイズする方法については、第 4.4 節を参照してください。

2.2 DECW\$STARTUP.COM プロシージャの使用方法

通常、DECW\$STARTUP.COM は、システム管理者が直接起動する唯一の DECwindows コマンド・プロシージャです。このプロシージャは SYSSMANAGER ディレクトリにあり、以下のように呼び出されます。

- システムのスタートアップ時に、DECwindows Motif を起動するために呼び出される。
- 特権ユーザによって、ディスプレイ・サーバを再起動するために、必要に応じて呼び出される。
- 特権ユーザによって、DECwindows サーバ、ライブラリ、アプリケーションを起動するために呼び出される。たとえばライブラリの「起動」は、論理名の定義とイメージのインストールで構成される。

DECW\$STARTUP.COM は、1 つのパラメータを受け付けます。表 2-1 に、指定可能なパラメータ値とそれによって起動されるプロシージャの一覧を示します。

表 2-1 スタートアップ・パラメータの値

P1 の値	起動されるプロシージャ
" " (なし)	完全なスタートアップ処理
RESTART	SYSSMANAGER:DECW\$STARTSERVER.COM
I18N	SYSSMANAGER:DECW\$STARTI18N.COM (存在する場合)
XTERMINAL	SYSSSTARTUP:DECW\$STARTXTERMINAL.COM (存在する場合)
SERVER	SYSSMANAGER:DECW\$STARTSERVER.COM
LIBS	SYSSMANAGER:DECW\$STARTLIBS.COM
APPS	SYSSMANAGER:DECW\$STARTAPPS.COM

2.3 システム・スタートアップの留意事項

ここでは、DECwindows Motif のシステム・スタートアップ・プロセスを起動する前に推奨される作業について説明します。

2.3.1 DECwindows システム論理名の定義

DECwindows を起動する前に、SYSSMANAGER ディレクトリにあるシステム・スタートアップ・ファイル SYSTARTUP_VMS.COM を編集して、論理名を変更または追加することができます。

表 2-2 に、各論理名とその意味を示します。

表 2-2 システム論理名

論理名	意味
DECW\$DEFAULT_TRANSPORT	DECwindows セッションの開始時に、初期ワークステーションのディスプレイ・デバイスが使用するデフォルトのトランスポートを設定する。
DECW\$IGNORE_AUTOGEN	SYSGEN パラメータを確認しない。
DECW\$IGNORE_DECWINDOWS	DECwindows を起動しない。
DECW\$IGNORE_SHARE_ADDRESS	常駐または共用アドレス・データを使用したイメージをインストールしない。
DECW\$IGNORE_SUBPROCESS	サブプロセスを生成しない。
DECW\$IGNORE_WORKSTATION	ワークステーション固有のスタートアップを実行しない。
DECW\$INSTALL_TCPIP	サーバの TCP/IP トランスポートのサポートを有効にする。
DECW\$PARAMS_BEFORE_DEVICE	ディスプレイ・デバイスを決定する前に SYSGEN パラメータを確認する。
DECW\$SETDISPLAY_DEFAULT_TRANSPORT	DCL コマンド SET DISPLAY/CREATE で使用するデフォルトのトランスポートを設定する。
DECW\$USEXLIBPG4	Xlib C ランタイム・ライブラリ (C RTL) のロケール関数を使用する。
OPCSOPA0_ENABLE	オペレータ・コンソール (OPA0) にブロードキャスト・メッセージを送信する。

2.3.2 システム・パラメータの調整

DECwindows Motif では、いくつかのシステム・パラメータに特定の最小値を設定する必要があります。値が適切に設定されていないと、AUTOGEN を実行して値を変更するように促すメッセージが表示されます。値の詳細は、付録 A を参照してください。

警告

SYSSYSTEM:MODPARAMS.DAT に設定したパラメータ値は、他の設定より優先されます。このファイルに値を設定すると、DECwindows Motif が起動しなくなるおそれがあります。

システムの性能と物理メモリの使用量を最適化するために、システムのスタートアップ時に多くの DECwindows イメージが (共用アドレス・データとともに) 常駐イメージとしてインストールされます。

DECwindows Motif ソフトウェアを、通常 of システム・スタートアップではなく手動で開始すると、以下のようなエラー・メッセージが表示されることがあります。

```
$ @SYS$MANAGER:DECW$STARTUP
%INSTALL-I-NONRES, image installed ignoring '/RESIDENT' DISK$ALPHA:
<SYS0.SYSCOMMON.SYSLIB>DECW$XPORT_SERVICES.EXE
-INSTALL-E-NOGHREG, insufficient memory in the code or data granularity hint region
%INSTALL-I-NONRES, image installed ignoring '/RESIDENT' DISK$ALPHA:
<SYS0.SYSCOMMON.SYSLIB>DECW$LCNLIBSHR.EXE
-INSTALL-E-NOGHREG, insufficient memory in the code or data granularity hint region
%INSTALL-I-NONRES, image installed ignoring '/RESIDENT' DISK$ALPHA:
<SYS0.SYSCOMMON.SYSLIB>DECW$TRANSPORT_COMMON.EXE
-INSTALL-E-NOGHREG, insufficient memory in the code or data granularity hint region
.
.
.
```

これらのメッセージは、粒度ヒント領域に、イメージを常駐イメージとしてインストールできるだけのメモリがないことを示します。その結果、イメージは、共用アドレス・データなしで非常駐イメージとしてインストールされます。これらのメッセージは、重大なエラーではありませんが、これらのイメージの性能に影響します (特に起動時)。

スタートアップ時に上記のメッセージが表示される場合は、1 つ以上の GH_* システム・パラメータ (たとえば、GHR\$RVPGCNT) の値を増やし、DECwindows Motif を再起動して、常駐イメージが正常にインストールされるようにします。

ディスプレイ・サーバの構成

この章では、DECwindows ディスプレイ・サーバの構成方法に関する以下のトピックについて説明します。

- ディスプレイ・サーバのカスタマイズ。DECW\$PRIVATE_SERVER_SETUP ファイルの使用方法を説明し、指定可能なカスタマイズ・パラメータの一覧を示します。
- ネットワーク・トランスポートの指定
- アクセス制御の設定
- マルチヘッド・システムの設定
- デフォルトのキーボード・レイアウトの変更
- 新しいフォントの指定
- LBX プロキシ・サーバの設定

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

ここでは、サーバ・カスタマイズ・パラメータを使用して、DECwindows ディスプレイ・サーバのデフォルトの動作を変更する方法について説明します。また、指定可能なパラメータと、クラスタ共通およびスタンドアロン・システムのパラメータを設定する方法について説明します。

3.1.1 DECW\$PRIVATE_SERVER_SETUP ファイルの使用方法

DECW\$PRIVATE_SERVER_SETUP.TEMPLATE ファイルは、SYSSMANAGER ディレクトリにあります。このテンプレート・ファイルには、DECwindows Motif ディスプレイ・サーバ環境をカスタマイズするために必要な情報が格納されています。

ディスプレイ・サーバを起動すると、ディスプレイ・サーバは、論理名テーブル (DECW\$SERVER n _TABLE) 内でデフォルトの属性を設定変更する論理名を探します。第2章で説明したように、論理名テーブルは、DECwindows のスタートアップ処理で作成され値が設定されます。DECW\$PRIVATE_SERVER_SETUP.COM ファイルで定義されている各パラメータについて、スタートアップ・プロシージャがディスプレイ・サーバ論理名テーブル中に論理名を作成します。

いくつかのパラメータは、スタートアップ・プロシージャ中でのみ使用されるため、ディスプレイ・サーバ論理名テーブルには一致する論理名がありません。これらのパラメータは、ディスプレイ・サーバ・プロセスを起動する際に使用されるシステム・パラメータを識別するか、他のパラメータと組み合わせて、ディスプレイ・サーバ論理名テーブルに追加する論理名の値を得るために使用されます。

プライベート論理名テーブル中に論理名が見つからないと、ディスプレイ・サーバはシステム論理名テーブルを参照します。サポートされている機能のうち、使用している OpenVMS のバージョンでは対応するパラメータがない機能については、DECW\$PRIVATE_SERVER_SETUP.COM ファイル内で DCL コマンド DEFINE /SYSTEM を使用して論理名を定義します。

3.1.2 ディスプレイ・サーバのカスタマイズ・パラメータ

表 3-1 に、パラメータとその型、デフォルト値、範囲をアルファベット順に示します。

表 3-1 DECwindows ディスプレイ・サーバのカスタマイズ・パラメータ

パラメータ	型	デフォルト値	範囲
DECW\$CLIENT_ERROR_THRESHOLD	整数	1	
DECW\$CURSOR_SIZE	整数	32	16,32
DECW\$DEFAULT_KEYBOARD_MAP	文字列	NORTH_AMERICAN_ LK401AA	
DECW\$DEVICE	文字列リスト		
DECW\$FONT_SERVERS	文字列リスト		
DECW\$IPV6_FONT_SUPPORT	文字列		
DECW\$KEYMAP	文字列		
DECW\$MONITOR_DENSITY	整数リスト	100	
DECW\$PRIMARY_DEVICE	文字列		
DECW\$RGBPATH	文字列	SYSSMANAGER:DECW\$RGB.DAT	
DECW\$SECURITY_POLICY	文字列		
DECW\$SERVER_ACCESS_TRUSTED	文字列	SYSSMANAGER:DECW\$SERVER_ ACCESS_TRUSTED.DAT	
DECW\$SERVER_ACCESS_ALLOWED	文字列	SYSSMANAGER:DECW\$SERVER_ ACCESS_ALLOWED.DAT	
DECW\$SERVER_AUDIT_LEVEL	整数	0	0,1,2,4
DECW\$SERVER_BELL_BASE_VOLUME	整数	50	0...100
DECW\$SERVER_BUG_COMPATIBILITY	論理値	True	
DECW\$SERVER_DEFAULT_BACKING_STORE	整数	0	0...2
DECW\$SERVER_DEFAULT_VISUAL_CLASS	整数	デバイス依存	0...5
DECW\$SERVER_DENSITY	整数	100	75,100
DECW\$SERVER_DISABLE_BACKING_STORE	論理値	False	
DECW\$SERVER_DISABLE_SAVE_UNDER	論理値	False	
DECW\$SERVER_DISABLESCREEN	整数		0...15
DECW\$SERVER_DISABLE_TEST	論理値	False	
DECW\$SERVER_DUMP	論理値	False	
DECW\$SERVER_EDGE_BOTTOM	整数リスト		
DECW\$SERVER_EDGE_LEFT	整数リスト		
DECW\$SERVER_EDGE_RIGHT	整数リスト		
DECW\$SERVER_EDGE_TOP	整数リスト		
DECW\$SERVER_ENABLE_ACCESSX	整数	0	0,1
DECW\$SERVER_ENABLE_KB_AUTOREPEAT	論理値	True	
DECW\$SERVER_ENABLESCREEN	整数		0...15
DECW\$SERVER_ENQUEUE_LIMIT	整数	付録 A を参照	
DECW\$SERVER_ERROR_LOG_TO_KEEP	整数	2	
DECW\$SERVER_ERROR_THRESHOLD	整数	10	

(次ページに続く)

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

表 3-1 (続き) DECwindows ディスプレイ・サーバのカスタマイズ・パラメータ

パラメータ	型	デフォルト値	範囲
DECW\$SERVER_EXTENSIONS	文字列リスト	XIE,DEC-XTRAP,MULTI-BUFFERING,SEC_XAG	
DECW\$SERVER_FILE_LIMIT	整数	200	
DECW\$SERVER_FONT_CACHE_SIZE	整数	0	
DECW\$SERVER_FONT_CACHE_UNIT	整数	128	
DECW\$SERVER_KEYCLICK_VOLUME	整数	0	0...100
DECW\$SERVER_KEY_REPEAT_DELAY	整数	660	0...1000
DECW\$SERVER_KEY_REPEAT_INTERVAL	整数	40	0...1000
DECW\$SERVER_MOUSE_ACCELERATION	整数	2	0..2
DECW\$SERVER_MOUSE_THRESHOLD	整数	4	
DECW\$SERVER_ONLYSCREEN	整数		0...15
DECW\$SERVER_PAGE_FILE	整数	付録 A を参照	
DECW\$SERVER_PRIORITY	整数	6	1...15
DECW\$SERVER_SCALE_BITMAP FONTS	論理値	False	
DECW\$SERVER_SCREENSAVERS	文字列リスト		
DECW\$SERVER_SCREEN_SAVER_INTERVAL	整数	600	
DECW\$SERVER_SCREEN_SAVER_PREFER_BLANKING	論理値	True	
DECW\$SERVER_SCREEN_SAVER_TIMEOUT	整数	600	
DECW\$SERVER_TRANSPORTS	文字列リスト		
DECW\$SERVER_WSDEF	整数	付録 A を参照	
DECW\$SERVER_WSEXTENT	整数	付録 A を参照	
DECW\$SERVER_WSQUOTA	整数	付録 A を参照	
DECW\$SERVER_XAUTHORITY	文字列		
DECW\$SERVER_XKEYBOARD_COMPILED_DIR	文字列	SYSSCOMMON: [SYSSKEYMAP.XKB .COMPILED]	
DECW\$SERVER_XKEYBOARD_DIRECTORY	文字列	DECW\$SYSSCOMMON: [SYSSKEYMAP.XKB]	
DECW\$SERVER_XKEYBOARD_LOAD_MAP	整数	0	0,1
DECW\$SERVER_XKEYBOARD_MAP	文字列	DIGITAL_US_LK201	
DECW\$START_FONT_SERVER	論理値		False
DECW\$EXPORT_SYNC_TIMEOUT	整数	30000	

以降の項では、表 3-1 に示したすべてのパラメータの定義と例を示します。

3.1.2.1 サーバ・プロセス

DECwindows のスタートアップ・プロセス内で、ディスプレイ・サーバが独立プロセスとして起動されます。通常は、サーバ・プロセスに割り当てられるデフォルトの制限値で十分です。ただし、場合によっては、特定のパラメータを増やさなくてはならないことがあります。DECW\$PRIVATE_SERVER_SETUP.COM でシンボルを定義することで、サーバ・プロセスの制限値を調整することができます。

たとえば、以下の場合に制限値を増やす必要があります。

- 大量のメモリの使用 – これは、大きなピクスマップを多数使用するアプリケーションの実行によって起こります。

大量にメモリを使用する環境向けに調整するには、DECW\$SERVER_PAGE_FILE に割り当てられた値を増やします。これにより、サーバ・プロセスの PGFLQUOTA が調整されます。このパラメータは、システム・パラメータ VIRTUALPAGECNT とシステム・ページ・ファイルのサイズによって制限される点に注意してください。また、DECW\$SERVER_WSDEF と DECW\$SERVER_WSQUOTA の値を増やすことも検討してください。

- 多数のフォントの同時利用 – 参照されているフォント・ファイルのうち、フォント・キャッシュが有効になっているものは、そのフォントを参照しているすべてのクライアント・アプリケーションが終了するまで開かれたままになります。

この状況に合わせて調整するには、DECW\$SERVER_FILE_LIMIT の値を増やします。この値は、同時にアクティブになる可能性があるクライアント・アプリケーションのあらゆる組み合わせで使用されるフォントの合計数よりも大きくする必要があります。

- 頻繁に要求を行うアプリケーション – クライアント・アプリケーションによっては、X ディスプレイ・サーバからの応答が不要な要求を連続的に送るものがあります。これらの要求により、他のアプリケーションに割り当てられる処理時間に影響が与えることがあります。これにより、ウィンドウ・マネージャなどの他のアプリケーションの動作が遅くなることがあります。

応答時間を改善するには、DECW\$SERVER_PRIORITY の値を変更することで、ディスプレイ・サーバ・プロセスの優先順位を調整します。

- 多数のアプリケーションの同時使用 – 多数のクライアント・アプリケーションが同じディスプレイ・サーバに同時に接続される場合は、ネットワーク・トランスポート用にサーバのリソースの追加が必要になる場合があります。

DECwindows ディスプレイ・サーバのチューニングについての詳細は、付録 A を参照してください。

DECW\$SERVER_PRIORITY

DECW\$SERVER_PRIORITY パラメータは、ディスプレイ・サーバ・プロセスの優先順位を制御します。このパラメータを使用すると、サーバ・プロセスの優先順位を下げることができ、大量の要求によって応答時間が長くなる状況でのシステムの性能を改善することができます。

サーバ・プロセスの最適な優先順位を推定します。有効な値は 1 (低い) から 15 (高い) です。最善の結果を得るには、中間値 4, 5, または 6 (デフォルト) を使用することをお勧めします。優先順位を下げすぎると、入力デバイスの反応 (キーボードやマウスの動作) が悪くなります。

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

次のシンボル定義は、DECwindows のディスプレイ・サーバに優先順位 4 を割り当てています。

例

```
$ DECW$SERVER_PRIORITY == "4"
```

DECW\$SERVER_WSDEF

このパラメータは、DECwindows のディスプレイ・サーバ・プロセスに適用するプロセス・リミット (ページレット単位) を定義します。DECW\$SERVER_WSDEF の値が DECW\$SERVER_WSQUOTA よりも大きいと、DECW\$SERVER_WSQUOTA の値が適用されます。ワーキング・セット・サイズを決定する方法については、『Guide to OpenVMS Performance Management』を参照してください。

次の論理名定義は、5000 ページレットのワーキング・セット・サイズを DECwindows ディスプレイ・サーバに割り当てています。

例

```
$ DEFINE DECW$SERVER_WSDEF 5000
```

DECW\$SERVER_WSQUOTA

このパラメータは、DECwindows サーバに割り当てることができる最大の物理メモリ量 (ワーキング・セット) を定義します。ワーキング・セット・サイズの決定方法についての詳細は、『Guide to OpenVMS Performance Management』を参照してください。

次の論理名定義は、DECwindows サーバに割り当てられる最大ページレット数として 10000 を指定します。

例

```
$ DEFINE DECW$SERVER_WSQUOTA 10000
```

DECW\$SERVER_WSEXTENT

このパラメータは、DECW\$SERVER_WSDEF によって割り当てられる領域よりも多くの領域が DECwindows ディスプレイ・サーバで必要な場合に、物理メモリの絶対的な上限を定義します。DECwindows サーバに対して割り当てられるページレットの総数は、DECW\$SERVER_WSQUOTA を超えても構いません (追加のページレットが使用可能な場合は最大 DECW\$SERVER_WSEXTENT の値まで)。ワーキング・セット・サイズの決定方法についての詳細は、『Guide to OpenVMS Performance Management』を参照してください。

次の論理名定義は、必要に応じて DECW\$SERVER_WSQUOTA を超えない範囲で、DECwindows サーバに 20000 ページレットを割り当てます。

例

```
$ DEFINE DECW$SERVER_WSEXTENT 20000
```

DECW\$SERVER_PAGE_FILE

このパラメータは、DECwindows ディスプレイ・サーバが使用できる最大の仮想メモリ量 (ページレット単位) を定義します。

次の論理名定義は、ページ・ファイルのサイズを 1000000 ブロックに増やします。

例

```
$ DEFINE DECW$SERVER_PAGE_FILE 1000000
```

DECW\$SERVER_FILE_LIMIT

このパラメータは、サーバが同時にオープンできるファイルの最大数を定義します。また、最大同時クライアント接続数も表します。デフォルトは 200 ファイルです。

次の論理名定義は、サーバがオープンできるファイルの最大数を 275 に増やします。

例

```
$ DEFINE DECW$SERVER_FILE_LIMIT 275
```

DECW\$SERVER_ENQUEUE_LIMIT

このパラメータは、プロセス間でリソース (特にファイル) を共有するために使用する未処理ロックの最大数を定義します。デフォルトは 512 ロックです。

次の論理名定義は、未処理ロック・リミットをデフォルトの倍の 1024 ロックに増やします。

例

```
$ DEFINE DECW$SERVER_ENQUEUE_LIMIT 1024
```

3.1.2.2 拡張

すべての X ディスプレイ・サーバが持っている機能として、サーバ拡張のサポートがあります。拡張は、X ディスプレイ・サーバに対する追加であり、追加のプロトコル・リクエストを解釈し、新しい機能や改良された機能を実行します。

拡張の一部は組み込みで常に有効になっていますが、パラメータを定義して有効にする必要があるものもあります。OpenVMS ディスプレイ・サーバは、動的にロードされる拡張をサポートしています。これらの拡張は独立した共有イメージに格納されており、要求に応じてアクティブ化されます。動的にロードされる拡張のメリットは、拡張を使用しないかぎり、その拡張に必要なリソースが消費されない点です。

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

ディスプレイ・サーバでは、サーバの初期化時に、特定の動的拡張がロードされている必要があります。ロードする拡張は、シンボル `DECW$SERVER_EXTENSIONS` で識別します。

`DECW$SERVER_EXTENSIONS`

このパラメータは、サーバの初期化時にロードする拡張の一覧を定義するために使用します。このパラメータは、動的にロード可能な拡張イメージ (組み込みの拡張に対する追加) のリストに変換されます。

表 3-2 に、利用可能なサーバ拡張の一覧を示します。このパラメータのデフォルト値は "XIE,DEC-XTRAP,MULTI-BUFFERING,SEC_XAG" です。

表 3-2 ロード可能なディスプレイ・サーバ拡張

拡張名	パラメータ値	説明
Digital 2D 拡張	D2DX	2D グラフィックス環境でのサーバの性能が向上します。
Digital Trapping 拡張	DEC-XTRAP ¹	イベントのトラップとシミュレーションを行います。
Low-Bandwidth X 拡張	LBX	LBX プロキシ機能を有効にします。
X Double-Buffering 拡張	DBE	ちらつきのないウィンドウ表示と滑らかなアニメーションを可能にします。
X Imaging 拡張	XIE ¹	イメージング操作をローカルに実行します。
X Keyboard 拡張	XKB	X キーマップ、キーボード・コンパイラ、および運動障害のあるユーザ向けの AccessX キーボード機能の使用を有効にします。
X Multi-Buffering 拡張	MULTI-BUFFERING ¹	滑らかなアニメーションのためのマルチ・バッファ・ウィンドウを有効にします。
X Security 拡張および Application Group 拡張	SEC_XAG ¹	X Security 拡張と Application Group 拡張を有効にします。
Xinerama 拡張	XINERAMA	XINERAMA プロトコルに基づいたマルチヘッド・システムの構成を可能にします。

¹デフォルトで事前にロードされます。

表 3-3 に、組み込みディスプレイ・サーバ拡張の一覧を示します。

表 3-3 組み込みディスプレイ・サーバ拡張

拡張名	パラメータ値	説明
Big Requests	BIG-REQUESTS	プロトコル・リクエストの長さフィールドを 32 ビット値に拡張します。
Colormap Utilization Policy	TOG-CUP	ディスプレイ・サーバにカラーマップ管理ポリシーを提供します。
DEC-Server-Mgmt-Extension	-	セッション・マネージャだけが使用するサーバ管理機能を提供します。
Extended Visual Information	EVI	カラーマップ情報やフレームバッファ・レベルなどのコア X ビジュアルに関する情報を、クライアントがサーバに対して照会できるようにします。
MIT Miscellaneous	MIT-SUNDRY-NONSTANDARD	その他のバグ互換モード制御機能を提供します。
MIT Screen Saver	MIT-SCREEN-SAVER	スクリーン・セーバが起動または切り替えられたときにクライアントが通知を受け取ることができるようにします。
MIT Shared Memory	MIT-SHM	共用メモリ、高速な PutImage のサポートを有効にします。
Non-Rectangular Window Shape	SHAPE	長方形でないウィンドウの表示を可能にします。
X Miscellaneous	XC-MISC	以前使用したリソース ID の範囲をディスプレイ・サーバから取得できるようにします。
X Synchronization	SYNC	さまざまなオペレーティング・システム上の複数のクライアントからの要求を同期させるプリミティブ呼び出しを提供します。
X Test	XTEST	単純なイベント・トラップとシミュレーション機能を提供します。

ユーザ作成や他社製などの X ウィンドウ・システム拡張のイメージがある場合は、このパラメータを使用して、これらの拡張をサーバのスタートアップ時に有効にすることもできます。グラフィックス・デバイスによっては、デバイスが使用する追加の拡張 (XFree86-DRI, GLX, および SGI-GLX など) がロードされる点に注意してください。

注意

サーバ・リソースの競合を防ぐため、同じディスプレイ・サーバ・システムにロードしてはならない拡張の組み合わせがあります。拡張の制限事項についての詳細は、『日本語 HP DECwindows Motif for OpenVMS リリース・ノート』を参照してください。

次のシンボル定義は、スタートアップ時に有効にするサーバ拡張のリストを指定します。

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

例

```
$ DECW$SERVER_EXTENSIONS == "XIE,DEC-XTRAP,XINERAMA,SEC_XAG,DBE"
```

DECW\$SERVER_DISABLE_TEST

このパラメータは、テスト拡張 XTEST および DEC-XTRAP を有効にするかどうかを制御します。このパラメータに指定できる値は、T (True: 無効) または F (False: 有効) です。デフォルト値は F です。

次のシンボル定義は、すべてのテスト拡張を無効にします。

例

```
$ DECW$SERVER_DISABLE_TEST == "T"
```

3.1.2.3 セキュリティ

DECwindows では、X ディスプレイ・サーバへのアクセスを制御するために、以下のメカニズムがサポートされています。

- 許可ユーザ・リストを使用した、ユーザ・ベースのアクセス制御
- X authority ファイルと Magic Cookie (MIT-MAGIC-COOKIE-1) または Kerberos (MIT-KERBEROS-5) プロトコルを使用した、トークン・ベースのアクセス制御
- セキュリティ拡張 (SECURITY) の利用

これらの方法では、X ディスプレイ・サーバへの接続を許可するクライアントを定義する手段と、接続後に実行可能な操作を定義する手段が追加で提供されています。これらのメカニズムで使用するファイル (セキュリティ・ポリシー・ファイル、X authority ファイル、アクセス許可ファイル、アクセス信頼ファイル) の場所を指定するには、この項のパラメータを使用します。

ディスプレイ・サーバのアクセス制御方式の実装方法についての詳細は、第 3.3 節を参照してください。

DECW\$SECURITY_POLICY

セキュリティ拡張を使用する場合に、このパラメータはセキュリティ・ポリシー・ファイルの名前を示します。デフォルトでは、ファイルは指定されていません。

次のシンボル定義は、セキュリティ・ポリシー・ファイル SYSS\$MANAGER:DECW\$SECURITY_POLICY.DAT を指定します。

例

```
$ DECW$SECURITY_POLICY == "SYSS$MANAGER:DECW$SECURITY_POLICY.DAT"
```

DECW\$SERVER_XAUTHORITY

このパラメータは、ディスプレイ・サーバが参照する X authority ファイルの名前を指定します。このファイルには、クライアントからサーバへの接続を許可するために使用するレコードが格納されています。デフォルトでは、ファイルは指定されていません。これにより、ローカルの SYSTEM アカウントからサーバへのクライアント・アクセス (DECnet またはローカル・トランスポート経由) が、追加認証なしで許可されます。

ファイルが指定されていると、このファイルの値がサーバにロードされ、すべてのクライアント接続で使用されます。通常のログイン処理を行うためには、DECW\$SERVER_ACCESS_TRUSTED.DAT ファイルを使用して、信頼関係によるアクセス許可を明示的に設定する必要があります。

DECW\$SERVER_XAUTHORITY で指定された X authority ファイルの設定は、ユーザが DECwindows デスクトップにログインする前に確立されるサーバ接続に適用されます。ユーザがデスクトップにログインすると、ユーザの X authority 設定が適用されます。

次のシンボル定義は、X authority ファイル SYSS\$MANAGER:DECW\$XAUTH.DAT を指定します。

例

```
$ DECW$SERVER_XAUTHORITY == "SYSS$MANAGER:DECW$XAUTH.DAT"
```

DECW\$SERVER_ACCESS_TRUSTED

このパラメータは、アクセス信頼ファイルの名前を指定します。このファイルには、信頼関係によるサーバへのアクセスが許されるクライアントの一覧を記載します。デフォルトのファイルは SYSS\$MANAGER:DECW\$SERVER_ACCESS_TRUSTED.DAT です。

DECW\$SERVER_ACCESS_TRUSTED で指定されたアクセス信頼ファイル内の設定は、ユーザが DECwindows デスクトップにログインする前のサーバ接続に適用される点に注意してください。ユーザがデスクトップにログインした後は、ユーザのアクセス設定が適用されます。

次のシンボル定義は、アクセス信頼ファイルの指定を DECW\$SERVER1_ACCESS_TRUSTED.DAT に変更します。

例

```
$ DECW$SERVER_ACCESS_TRUSTED == "SYSS$MANAGER:DECW$SERVER1_ACCESS_TRUSTED.DAT"
```

DECW\$SERVER_ACCESS_ALLOWED

このパラメータは、アクセス許可ファイルの名前を指定します。このファイルには、追加の認証なしにサーバへのアクセスが自動的に許可されるクライアントの一覧を記述します。デフォルトのファイルはSYS\$MANAGER:DECW\$SERVER_ACCESS_ALLOWED.DATです。

DECW\$SERVER_ACCESS_ALLOWED で指定されたアクセス許可ファイル内の設定は、ユーザがDECwindows デスクトップにログインする前のサーバ接続に対して適用される点に注意してください。ユーザがデスクトップにログインした後は、ユーザのアクセス設定が適用されます。

次のシンボル定義は、アクセス許可ファイルの指定を DECW\$SERVER1_ACCESS_ALLOWED.DAT に変更します。

例

```
$ DECW$SERVER_ACCESS_ALLOWED == "SYS$MANAGER:DECW$SERVER1_ACCESS_ALLOWED.DAT"
```

3.1.2.4 デバイス

スタートアップの際に、DECwindows のスタートアップ・プロシージャは、システムが認識しているすべてのグラフィック・デバイスを管理するために、デバイス固有のサーバ構成要素を識別して起動しようとしています。ここで説明するシンボルと論理名を使用すると、ディスプレイ・サーバでどのデバイスを使用するかを制御することができます。

サーバがデバイスから取得できない情報や、ディスプレイ・デバイスに関して設定変更が必要な情報があります。たとえば、表示可能な色が限られている特別な種類のモニタに関する情報を指定しなくてはならない場合があります。また、単一の仮想ディスプレイとして機能する複数のモニタで構成されるマルチヘッド・システムを構成する場合も、情報を指定する必要があります。

DECwindows では、2 種類のマルチヘッド構成がサポートされています。

- DECW\$MULTI_HEAD パラメータを使用した単純な構成
- Xinerama 拡張 (XINERAMA) とパラメータに基づいた高度な構成

XINERAMA (以前は Panoramix と呼ばれていました) を使用すると、マルチヘッド・システムの構成が可能となり、またディスプレイ内に画面を定義して有効にし、その順序を制御し、ディスプレイの境界と形状を設定するための複数のパラメータがあります。

デフォルトでは、マルチヘッド・ディスプレイのすべての画面が有効になっています。DECW\$SERVER_ONLYSCREEN および DECW\$SERVER_DISABLESCREEN を使用して、1 つ以上の画面をディスプレイから削除することができます。無効にされた画面は初期化されず、画面番号も割り当てられません。

マルチヘッド・ディスプレイの構成方法については、第 3.4 節を参照してください。

DECW\$MULTI_HEAD

このパラメータは、システムのマルチヘッド・サポートを構成します。
DECW\$MULTI_HEAD シンボルは、SYSSMANAGER:DECW\$PRIVATE_
SERVER_SETUP.TEMPLATE ファイルで設定済みです。

このパラメータを有効にするには、
SYSSMANAGER:DECW\$PRIVATE_SERVER_SETUP.TEMPLATE を
DECW\$PRIVATE_SERVER_SETUP.COM にコピーします。

DECW\$PRIMARY_DEVICE

サーバは、このパラメータを使用して特定の DECW\$DEVICE_XX.COM プロシージャをアクティブにすることで、デバイス名を確認します。ここで、XXはシンボルに付加される文字列です。次のシンボル定義では、プライマリ・デバイスとして GXA0 を割り当てます。

例

```
$ DECW$PRIMARY_DEVICE == "GXA0"
```

DECW\$DEVICE

このパラメータを使用すると、単純なマルチヘッド構成で使用するグラフィックス・デバイスを識別し、その順序を指定することができます。デフォルトでは、DECW\$DEVICE の値はシステムに実装されているグラフィックス・デバイスによって決まります。

次の例では、1 台のサーバ、1 つのマウス、1 つのキーボードで制御されるグラフィックス・デバイスの並びを指定します。

例

```
$ DECW$DEVICE == "GAA0,GAB0"
```

DECW\$DEFAULT_VISUAL_CLASS

このパラメータは、マルチヘッド・システムの各ヘッドのデフォルトのビジュアル・クラスを指定変更します。ビジュアル・クラスは数値であり、DECW\$INCLUDE:X.h で以下のように定義されています。

- 0 = StaticGray
- 1 = GrayScale
- 2 = StaticColor
- 3 = PseudoColor
- 4 = TrueColor
- 5 = DirectColor

個々のデバイス・タイプでのデフォルトはハードウェアに依存し、通常、8 プレーンのカラー・ボードでは PseudoColor、24 プレーンでは TrueColor です。モノクロ・ディスプレイを使用する場合は、デフォルトのビジュアル・クラスを GrayScale にすることで、カラーが階調表示に変換されます。GrayScale の出力は緑のリード線です。このパラメータでは、マルチヘッド・システムの各ヘッドに対応して複数の値を割り当てることができます。

次のシンボル定義は、ヘッド 0 に対して PseudoColor、ヘッド 1 に対して TrueColor、ヘッド 2 に対して StaticGray を指定します。

例

```
$ DECW$SERVER_DEFAULT_VISUAL_CLASS == "3,4,0"
```

DECW\$MONITOR_DENSITY

モニタ密度は、モニタのインチあたりのドット数 (dpi) を定義し、DECwindows Motif アプリケーションが画面の実際の幅を判断できるようにします。

モニタ密度のデフォルト値はサーバ密度です。実際に 75 dip や 100 dpi (フォント・サイズに関して DECW\$SERVER_DENSITY で使用する値) のモニタはほとんどないため、これらの値を使用して画面上のアイテムの実際の幅と高さを正確に計算することはできません。DECW\$MONITOR_DENSITY に実際の値を設定することで、Xlib ルーチンを使用して画面の幅と高さの正しい値を得ることができます。

実際のモニタ密度を計算するには以下の方法を使用します。

1. 画面のピクセル幅を確認します。

一般に、システムのグラフィックス・アダプタに応じて、ピクセル数は 1024 または 1280 となります。X Display Information ユーティリティ (xdpyinfo) を使用して現在のディスプレイのピクセルの幅と高さを得ることができます (第 3.1.5.2 項を参照)。

2. 画面の表示部分の幅を測定します (インチ単位)。
3. ピクセル値を画面の値で割ります。

VRT19 モニタと SPX グラフィックスを使用している場合は、次の計算を行います。

$$1280 \text{ ピクセル} / 13.5 \text{ インチ} = 94.81 \text{ dpi}$$

dpi 値を最も近い整数に丸め、次の例に示すように DECW\$MONITOR_DENSITY に 95 を設定します。

例

```
$ DECW$MONITOR_DENSITY == "95"
```

注意

モニタ密度とサーバ密度に異なる値を設定すると、75 dpi および 100 dpi のフォントの縮尺を実際のモニタ密度に合わせて調整することができないため、表示上の問題が発生する場合があります。

DECW\$MONITOR_DENSITY はモニタごとに設定することができます。次の例では、デュアル・ヘッド・ワークステーションのモニタ密度を設定する方法を示します。画面 0 を 95 dpi とし、画面 1 を 75 dpi としています。

例

```
$ DECW$MONITOR_DENSITY == "95,75"
```

DECW\$SERVER_SCREEN

XINERAMA を使用したマルチヘッド・システムでは、画面は、デバイス名とその物理的な位置に応じて、アルファベット順に初期化されます。このパラメータを使用すると、画面が初期化される順序を変更することができます。

次のシンボル定義は、4 画面のマルチヘッド・ディスプレイでの初期化の順序を変更します。

例

```
$ DECW$SERVER_SCREEN == "GYB0,GYA0,GYD0,GYC0"
```

DECW\$SERVER_ENABLESCREEN

XINERAMA を使用したマルチヘッド・システムでは、ディスプレイ中の無効になっている画面を個別に再度有効にすることができます。このパラメータを使用すると、指定した画面が有効になります。有効な値は 0 ~ 15 です。15 は、拡張によってサポートされる最大画面番号です。

次の例では、4 つの画面 (0,1,2,3) で構成されるマルチヘッド・ディスプレイの 2 番目の画面 (1) を有効にします。

例

```
$ DECW$SERVER_ENABLESCREEN == "1"
```

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

DECW\$SERVER_DISABLESCREEN

XINERAMA を使用したマルチヘッド・システムでは、ディスプレイ中の各画面を個別に無効にすることができます。このパラメータを使用すると、指定した画面が無効になります。有効な値は 0 ~ 15 です。15 は、拡張によってサポートされる最大画面番号です。

無効にされた画面は、ディスプレイの一部として初期化されなくなり、画面番号も割り当てられません。これにより既存の画面の順序が変わり、ディスプレイの定義済みのエッジ連結が変わります。

次のシンボル定義は、4 つの画面 (0,1,2,3) で構成されるマルチヘッド・ディスプレイの 3 番目の画面 (2) を無効にします。

例

```
$ DECW$SERVER_DISABLESCREEN == "2"
```

DECW\$SERVER_ONLYSCREEN

XINERAMA を使用したマルチヘッド・システムでは、ディスプレイ中の特定の画面を有効にし、それ以外をすべて無効にすることができます。このパラメータは、指定した画面を明示的に有効にし、それ以外をすべて無効にします。有効な値は 0 ~ 15 です。15 は、拡張によってサポートされる最大画面番号です。

次のシンボル定義は、4 つの画面 (0,1,2,3) で構成されるマルチヘッド・ディスプレイの 2 番目の画面 (1) を有効にし、それ以外の画面 (0,2,3) をすべて無効にします。

例

```
$ DECW$SERVER_ONLYSCREEN == "1"
```

DECW\$SERVER_EDGE_LEFT

XINERAMA を使用したマルチヘッド・システムでは、エッジ制御を使用して仮想ディスプレイの境界が定義されます。このパラメータは、ディスプレイの左側の境界をどの画面に連結するかを決定します。値は画面番号で指定します。例を示します。

left-screen#, *index-screen#*, *right-screen#*

ここで *index-screen#* は、境界を連結する画面の番号です。 *left-screen#* は、インデックスのすぐ左の画面番号を示し、 *right-screen#* はインデックスのすぐ右の画面番号を示します。境界を結びつけるすべての画面に対してこのパターンを繰り返します。値 -1 は指定なしと同じです。

下記のシンボル定義は、4 つの画面が次の順で配置された正方形のディスプレイの左エッジを指定します。

```
0 1
2 3
```

2 番目と 4 番目の画面 (インデックス 1 および 3) の左エッジが, 1 番目と 3 番目の画面 (0,2) に連結されます。

例

```
$ DECW$SERVER_EDGE_LEFT == "-1,0,-1,2"
```

DECW\$SERVER_EDGE_RIGHT

XINERAMA を使用したマルチヘッド・システムでは, エッジ制御を使用して仮想ディスプレイの境界が定義されます。このパラメータは, ディスプレイの右側の境界をどの画面に連結するかを決定します。値は画面番号で指定します。例を示します。

right-screen#, index-screen#, left-screen#

ここで *index-screen#* は, 境界を連結する画面の番号です。 *right-screen#* はインデックスのすぐ右の画面番号を示し, *left-screen#* は, インデックスのすぐ左の画面番号を示します。境界を結びつけるすべての画面に対してこのパターンを繰り返します。値 -1 は指定なしと同じです。

下記のシンボル定義は, 4 つの画面が次の順で配置された正方形のディスプレイの右エッジを指定します。

```
0 1
2 3
```

1 番目と 3 番目の画面 (インデックス 0 および 2) の右エッジが, 2 番目と 4 番目の画面 (1,3) に連結されます。

例

```
$ DECW$SERVER_EDGE_RIGHT == "1,-1,3,-1"
```

DECW\$SERVER_EDGE_TOP

XINERAMA を使用したマルチヘッド・システムでは, エッジ制御を使用して仮想ディスプレイの境界が定義されます。このパラメータは, ディスプレイの上部の境界をどの画面に連結するかを決定します。値は画面番号で指定します。例を示します。

top-screen#, index-screen#, bottom-screen#

ここで *index-screen#* は, 境界を連結する画面の番号です。 *top-screen#* は, インデックスのすぐ上の画面番号を示し, *bottom-screen#* はインデックスのすぐ下の画面番号を示します。境界を結びつけるすべての画面に対してこのパターンを繰り返します。値 -1 は指定なしと同じです。

下記のシンボル定義は, 4 つの画面が次の順で配置された正方形のディスプレイの上エッジを指定します。

```
0 1
```

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

2 3

3番目と4番目の画面(インデックス2および3)の上エッジが, 1番目と2番目の画面(0,1)に連結されます。

例

```
$ DECW$SERVER_EDGE_TOP == "-1,-1,0,1"
```

DECW\$SERVER_EDGE_BOTTOM

XINERAMA を使用したマルチヘッド・システムでは, エッジ制御を使用して仮想ディスプレイの境界が定義されます。このパラメータは, ディスプレイの下部の境界をどの画面に連結するかを決定します。値は画面番号で指定します。例を示します。

bottom-screen#, index-screen#, top-screen#

ここで*index-screen#*は, 境界を連結する画面の番号です。*bottom-screen#*はインデックスのすぐ下の画面番号を示し, *top-screen#*は, インデックスのすぐ上の画面番号を示します。境界を結びつけるすべての画面に対してこのパターンを繰り返します。値-1は指定なしと同じです。

下記のシンボル定義は, 4つの画面が次の順で配置された正方形のディスプレイの下エッジを指定します。

0 1
2 3

1番目と2番目の画面(インデックス0および1)の下エッジが, 3番目と4番目の画面(2,3)に連結されます。

例

```
$ DECW$SERVER_EDGE_BOTTOM == "2,3,-1,-1"
```

3.1.2.5 トランスポートおよびネットワーク接続

トランスポートは, DECwindows クライアント・アプリケーションとディスプレイ・サーバ間のリンクです。ここで説明するシンボルは, 使用可能なネットワーク・トランスポート・リンクとリンクのタイミング属性を制御するために使用します。

DECW\$SERVER_TRANSPORTS

サーバが着信接続を監視するネットワーク・トランスポートを指定することができます。DECW\$SERVER_TRANSPORTS に指定できる値は, "DECNET", "LOCAL", "TCPIP", および "LAT" です。このパラメータのデフォルト値は, システム論理名 DECW\$INSTALL_TCPIP の値によって変わります。この論理名に True が設定されていると, デフォルト値は "DECNET,LOCAL,TCPIP" となります。論理名が設定されていない場合のデフォルト値は "DECNET,LOCAL" です。

次の例では、使用するトランスポートとして DECnet , local , および TCP/IP を指定する方法を示します。

例

```
$ DECW$SERVER_TRANSPORTS == "DECNET,LOCAL,TCPIP"
```

DECW\$XPORT_SYNC_TIMEOUT

このパラメータは、トランスポートのタイムアウト値 (ミリ秒単位) を定義します。デフォルトは 30000 ミリ秒 (30 秒) です。このタイムアウト時間が経過する前にクライアントが自身のバッファを空にしないと、サーバはクライアントを切断します。

次の論理名定義は、タイムアウト値を 60000 ミリ秒 (1 分) に延長します。

例

```
$ DEFINE/SYSTEM DECW$XPORT_SYNC_TIMEOUT 60000
```

3.1.2.6 フォント

テキストの表示は、ディスプレイ・サーバの主な仕事の 1 つです。テキストを表示するために、ディスプレイ・サーバにフォント名と描画する文字セットが渡されます。通常は、フォント名はサーバ・システム (ディスプレイ・サーバまたはフォント・サーバ) 上にある特定のフォント・ファイルに対応します。フォント名とフォント・ファイルの対応付けを行う際に、ディスプレイ・サーバはフォント・パスを検索します。フォント・パスは、順序付けられたディレクトリ・リストです。有効なフォント・ディレクトリには、フォント・ファイル名と対応するフォント名の一覧が記述されたフォント・ディレクトリ・ファイル (DECW\$FONT_DIRECTORY*.DAT) がそれぞれ格納されています。ディスプレイ・サーバは、一致するフォント名が見つかるまで各ディレクトリ・ファイルを順に検索します。

また、DECwindows Motif ではフォント別名ファイル (DECW\$FONT_ALIAS*.DAT) をサポートしています。このファイルは、有効なフォントディレクトリに格納しておくことができます。これらのファイルにより、あるフォント名が別のフォント名にマッピングされます。ディスプレイ・サーバは、あるフォント・ディレクトリ内のフォント・ディレクトリ・ファイルを確認しても一致するフォントが見つからないと、フォント別名ファイルを検索します。別名ファイル中に一致するフォントが見つかり、変換されたフォント名に置き換えられ、フォント・パスの最初から再度検索が開始されます。

3.1.2.6.1 フォント・パス 第 3.1.2.6 項で説明したように、フォント・パスは、ディスプレイ・サーバから利用できるフォント・ファイルがあるディレクトリを順に列挙したリストです。フォント・パスは、いくつかの設定パラメータから構成されます。

利用可能なグラフィックス・デバイスの種類もフォント・パスの内容に影響を与えます。サーバの論理名テーブル中の論理名 DECW\$FONT にフォント・パスが格納されています。フォント・パスは、以下のディレクトリのサブセットです。

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

```
SYSS$COMMON:[SYSFONT.DECW.CURSOR32]
SYSS$COMMON:[SYSFONT.DECW.CURSOR16]
SYSS$COMMON:[SYSFONT.DECW.100DPI]
SYSS$COMMON:[SYSFONT.DECW.75DPI]
SYSS$COMMON:[SYSFONT.DECW.COMMON]
```

フォント・パスには以下のディレクトリも含まれています。

```
SYSS$COMMON:[SYSFONT.DECW.SPEEDO]
SYSS$COMMON:[SYSFONT.DECW.TRUETYPE]
SYSS$COMMON:[SYSFONT.DECW.TYPE1]
```

フォント・パスの最後には、フォント・サーバへのポインタを追加することもできます。詳細は、DECW\$FONT_SERVERS を参照してください。

各ディレクトリの改良版として、ユーザ定義のフォントとユーロ対応のフォントがあります。たとえば、SYSS\$COMMON:[SYSFONT.DECW.USER_100DPI]があります。詳細は、第 3.6 節を参照してください。

DECW\$SERVER_DENSITY

75 dpi と 100 dpi のどちらのフォントを使用するかは、サーバの密度値によって決まります。100 dpi のモニタを使用している場合、画面の密度は 75 dpi または 100 dpi になります。75 dpi のモニタ用のフォントを 100 dpi のモニタで表示すると、小さく表示されるため読みづらくなります。

次のシンボル定義は、100 dpi フォントを使用するように画面密度を設定します。

例

```
$ DECW$SERVER_DENSITY == "100"
```

DECW\$CURSOR_SIZE

このパラメータは、フォント・パスに含めるカーソル・フォント・ディレクトリを定義します。16 を設定すると 16x16 のカーソルだけが対象となり、32 を設定すると、32x32 と 16x16 の両方のカーソルが対象となります。値 32 を設定することで表示される大きなサイズのカーソルは、100 dpi のディスプレイに適しています。

次のシンボル定義は、カーソル・サイズとして 16x16 を設定します。

例

```
$ DECW$CURSOR_SIZE == "16"
```


DECW\$FONT_SERVERS

フォント・パスにフォント・サーバを追加するには、ファイル SYSSMANAGER:DECW\$PRIVATE_SERVER_SETUP.COM のサイト固有のサーバ・セクションでシンボル DECW\$FONT_SERVERS を定義します。

コンマで区切ったリストとしてシンボルを定義することで、複数のフォント・サーバを追加することができます。このシンボルでは大文字と小文字は区別されません。

次のシンボル定義は、フォント・パスにフォント・サーバ ASHFLD::FONTSRV を追加します。

例

```
$ DECW$FONT_SERVERS == "DECNET/ASHFLD::FONTSRV"
```

DECW\$IPV6_FONT_SUPPORT

このパラメータは、クライアントが明示的なフォント・サーバ・パスを指定した場合に、DECW\$FONT_SERVER 定義中でのトランスポート名 TCP および TCPIP の解釈方法を制御します。このパラメータには以下のいずれかの値を設定します。

- TCP_IS_INET6

トランスポート名 TCP および TCPIP は、INET6 トランスポートの同義語として解釈されます。この値を使用すると、トランスポート名 TCP または TCPIP を指定した場合に、ディスプレイ・サーバとそのフォント・サーバ間のデフォルトのトランスポートとして、IPv6 が有効になります。この設定は、IPv6 トランスポートをデフォルトとして使用することを要求する点に注意してください。指定したノード上で IPv6 トランスポートが利用できないと、IPv4 トランスポートが使用されます。

- DISABLED

トランスポート名 TCP および TCPIP は、INET トランスポートの同義語として解釈されます。これは、パラメータが定義されていない場合のデフォルト値です。この値を使用すると、トランスポート名 TCP または TCPIP を指定した場合に、ディスプレイ・サーバとそのフォント・サーバ間のトランスポートとして、IPv4 が設定されます。この場合も、クライアントがフォント・サーバ・パスで明示的に INET6 トランスポートを指定することで、IPv6 トランスポートを要求することができます。

次の例では、トランスポート名 TCP および TCPIP を INET6 のトランスポートとして解釈することを指定しています。

例

```
$ DECW$IPV6_FONT_SUPPORT == TCP_IS_INET6
```

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

3.1.2.6.2 フォント・キャッシュ フォント・ファイルには、そのフォント内のすべてのグリフ (文字) を表す圧縮されたバイナリ・データが格納されています。たとえば、次のフォントファイルには、Helvetica 12 ポイント・フォントの各グリフのすべての情報が格納されています。

```
DECW$SYSCOMMON: [SYSFONT.DECW.100DPI]HELVETICA12_100DPI.DECW$FONT
```

DECwindows ディスプレイ・サーバは、X11R5 のディスプレイ・サーバで採用された PCF (Portable Compiled format) をサポートしています。PCF ではフォント形式が変更されており、フォント・データの一部としてビット/バイトの指示とアラインメント情報が格納されているため、ベンダに依存しないフォント・サポートが可能になっています。

DECW\$SERVER_FONT_CACHE_SIZE

このパラメータは、各フォントに割り当てる単位数を定義し、16 ビットの PCF フォントだけに適用されます。デフォルトは 1024 です。

次の論理名定義は、フォント・キャッシュのサイズを 512 に減らします。

例

```
$ DEFINE/SYSTEM DECW$SERVER_FONT_CACHE_SIZE 512
```

DECW\$SERVER_FONT_CACHE_UNIT

このパラメータは、単位あたりのバイト数を定義します。デフォルトは 128 です。

次の論理名定義は、単位あたりのバイト数を 512 に増やします。

例

```
$ DEFINE/SYSTEM DECW$SERVER_FONT_CACHE_UNIT 512
```

3.1.2.6.3 フォント・スケーリング いくつかの形式のスケーラブル・アウトライン・フォントのサポートに加え、DECwindows のディスプレイ・サーバは、あらゆるビットマップ・フォントを任意のサイズにスケーリングすることができます。

ただし、これらのスケーリングされたビットマップ・フォントの品質は、元のビットマップ・フォントはもちろん、それに相当するスケーリングされたアウトライン・フォントよりも大きく劣ります。デフォルトでは、この機能は無効になっています。いくつかのケースでは、この品質劣化は許容範囲にあり、スケーリングによりフォント・サイズの柔軟性が増します。

DECW\$SERVER_SCALE_BITMAP_FONTS

このパラメータを使用すると、ビットマップ・フォントのスケールリングによる、任意のポイント・サイズの文字の表示が有効になります。デフォルト値は False (無効) です。

次の論理名定義は、ビットマップ・フォントのスケールリングを有効にします。

例

```
$ DEFINE/SYSTEM DECW$SERVER_SCALE_BITMAP_FONTS TRUE
```

3.1.2.7 キーボード

サーバ・システムに接続されているキーボードのデフォルト属性のいくつかを変更することができます。これには、キークリック音やベルの音量、オートリピートなどのキーボード動作に加えて、キーボードの各キーを、クライアント・アプリケーションが使用する、キーボードに依存しないシンボルにマッピングする方法も含まれます。

また、X Keyboard 拡張 (XKB) では、キーボード・レイアウトと音響フィードバックを定義するための高度な機能が提供されます。XKB を使用して X キーボード・レイアウト・ファイルの設定を指定するには、ここで示すパラメータを使用します。カスタム・キーマップ・ファイルおよびキーボード・レイアウト・ファイルをロードする方法については、第 3.5 節を参照してください。

ここに示すキーボード設定のいくつかは、セッション・マネージャで設定変更することができます。これらのシンボルの定義は、ユーザがログインする前または DECwindows セッション・マネージャを使用していない場合に有効になります。

DECW\$KEYMAP

DECW\$KEYMAP パラメータは、キーボード・マッピング・ファイルがあるディレクトリの指定です。これは参照のためだけにあります。この値は変更しないでください。

DECW\$DEFAULT_KEYBOARD_MAP

キーボードの言語を指定することができます。DECW\$DEFAULT_KEYBOARD_MAP に指定できる値は、SYS\$COMMON:[SYS\$KEYMAP.DECW.SYSTEM]ディレクトリおよび SYS\$COMMON:[SYS\$KEYMAP.DECW.USER]ディレクトリ内のファイル名 (ファイル・タイプを除く) です。

有効なキーマップ名の一覧については、付録 B を参照してください。

次のシンボル定義は、キーボード・レイアウトをオランダ語のタイプライタ・レイアウトに変更します。

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

例

```
$ DECW$DEFAULT_KEYBOARD_MAP == "DUTCH_LK201LH_TW"
```

DECW\$SERVER_BELL_BASE_VOLUME

このパラメータは、キーボードのベル音の音量を決定します。値は 0 ~ 100 で、100 が最大の音量になります。デフォルトの音量は 50 です。

次の論理名定義は、ボリュームを最大音量の 1/4 にします。

例

```
$ DEFINE/SYSTEM DECW$SERVER_BELL_BASE_VOLUME 25
```

DECW\$SERVER_ENABLE_ACCESSX

このパラメータは、スティッキー・キーやスロー・キーなど、障害を持ったユーザ向けの AccessX キーボード機能を有効にします。指定可能な値は 0 (無効) または 1 (有効) です。デフォルトは 0 です。

次の例では、AccessX 機能を有効にしています。

例

```
$ DECW$SERVER_ENABLE_ACCESSX == "1"
```

DECW\$SERVER_ENABLE_KB_AUTOREPEAT

キーボードのオートリピートは、キーを押している間は文字が自動的に繰り返されるようにするためのオプションです。このパラメータに True を指定することでこのオプションを有効にすることができます。

次の論理名定義は、キーボードから入力した文字のオートリピートを有効にします。

例

```
$ DEFINE/SYSTEM DECW$SERVER_ENABLE_KB_AUTOREPEAT T
```

DECW\$SERVER_KEYCLICK_VOLUME

このパラメータは、キーボードのクリック音の音量を決定します。値は 0 ~ 100 で、100 が最も大きな音量になります。デフォルトは 0 です。

次のシンボル定義は、ボリュームを最大音量の 1/4 にします。

例

```
$ DEFINE/SYSTEM DECW$SERVER_KEYCLICK_VOLUME 25
```

DECW\$SERVER_KEY_REPEAT_DELAY

XKB を使用する場合、このパラメータはキーストロークのリピートを開始するまでの時間をミリ秒単位で指定します。このパラメータで有効な値は 0 ~ 1000 です。デフォルトは 660 です。

次のシンボル定義は、キーストロークのリピートを開始するまでの遅延を指定します。

例

```
$ DECW$SERVER_KEY_REPEAT_DELAY == "800"
```

DECW\$SERVER_KEY_REPEAT_INTERVAL

XKB を使用する場合、このパラメータは、リピートされるキーストローク間の時間をミリ秒単位で指定します。このパラメータで有効な値は 0 ~ 1000 です。デフォルトは 40 です。

次のシンボル定義は、キーストロークのリピート間隔を指定します。

例

```
$ DECW$SERVER_KEY_REPEAT_INTERVAL == "20"
```

DECW\$SERVER_XKEYBOARD_COMPILED_DIR

XKB を使用する場合、このパラメータは、すべてのコンパイル済み X キーボード・ファイルのデフォルト・ディレクトリを指定します。このディレクトリには、サーバが必要に応じてコンパイルするキーマップ・ファイルも格納されます。デフォルトは SYSS\$COMMON:[SYSS\$KEYMAP.XKB.COMPILED] です。

次のシンボル定義は、ルート・ディレクトリを SYSS\$COMMON:[SYSS\$KEYMAP.XKB.SERVER1] に変更します。

例

```
$ DECW$SERVER_XKEYBOARD_COMPILED_DIR == "SYSS$COMMON:[SYSS$KEYMAP.XKB.SERVER1]"
```

DECW\$SERVER_XKEYBOARD_DIRECTORY

XKB を使用する場合、このパラメータは、すべての X キーボード・ファイルのデフォルト・ルート・ディレクトリを指定します。構成要素となるすべてのソース X キーボード・ファイルは、このルート・ディレクトリの下の子ディレクトリに格納されます。デフォルトは DECW\$SYSSCOMMON:[SYSS\$KEYMAP.XKB] です。

次のシンボル定義は、ルート・ディレクトリを SYSS\$COMMON:[SYSS\$KEYMAP.XKB] に変更します。

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

例

```
$ DECW$SERVER_XKEYBOARD_DIRECTORY == "SYS$COMMON:[SYS$KEYMAP.XKB]"
```

DECW\$SERVER_XKEYBOARD_LOAD_MAP

XKBを使用する場合、このパラメータを設定すると、DECW\$SERVER_XKEYBOARD_MAPで指定されるすべてのXキーボード・レイアウトがロードされます。このパラメータに指定できる値は0(無効)または1(有効)です。デフォルトは0です。このパラメータを無効にすると、DECwindowsのキーボード・マップが使用されます。

次のシンボル定義は、デフォルトのDECwindowsキーボード・マップ・ファイルをロードします。

例

```
$ DECW$SERVER_XKEYBOARD_LOAD_MAP == "1"
```

DECW\$SERVER_XKEYBOARD_MAP

XKBを使用する場合、このパラメータには、使用しているキーボード用のデフォルトのXキーボード・レイアウト・ファイルを指定します。デフォルトはDIGITAL_US_LK201です。コンパイル済みのレイアウト・ファイルがDECW\$SERVER_XKEYBOARD_COMPILED_DIRで指定された場所がない場合は、ディスプレイ・サーバはXキーボード・コンポーネント・データベース内のデータに基づいてオンデマンドでファイルをコンパイルします。

次のシンボル定義は、Xキーボード・レイアウトを別のキーボード・レイアウトに変更します。

例

```
$ DECW$SERVER_XKEYBOARD_MAP == "DIGITAL_US_LK401"
```

3.1.2.8 マウス

マウス・ポインタ属性を変更するには、ここに示すシンボルを使用します。ここに示すマウス・ポインタ設定よりも、セッション・マネージャでの設定の方が優先されます。これらのシンボルの設定は、ユーザがログインする前、またはDECwindowsセッション・マネージャを使用していない場合に有効です。

DECW\$SERVER_MOUSE_ACCELERATION

このパラメータは、マウスの移動とポインタの移動の間の関係を定義します。以下の値を指定することができます。

2 (高速)

1 (標準)

0 (低速)

次の論理名を定義すると、マウスの移動と比べて高速にポインタが移動するようになります。

例

```
$ DEFINE/SYSTEM DECW$SERVER_MOUSE_ACCELERATION 2
```

DECW\$SERVER_MOUSE_THRESHOLD

このパラメータは、DECwindows サーバが移動の通知を受け取るマウスの最小移動量 (ピクセル単位) を定義します。デフォルトは4ピクセルです。

次の論理名を定義すると、マウスの移動の感度が非常に高くなり、マウスの移動がDECwindows サーバに頻繁に通知されるようになります。

例

```
$ DEFINE/SYSTEM DECW$SERVER_MOUSE_THRESHOLD 1
```

3.1.2.9 カラー・データベース

ディスプレイ・サーバは、カラー・データベース・ファイルを使用してクライアント・アプリケーションからサーバに渡された色の名前を RGB 値に変換します。カラー・データベースの属性を変更するには、ここで説明するシンボルを使用します。

DECW\$RGBPATH

このパラメータは、ディスプレイ・サーバが使用する RGB データベースのファイル名を定義します。デフォルトの名前は SYS\$MANAGER:DECW\$RGB.DAT です。

次のシンボル定義は、ファイル名の値を DECW\$RGBPATH.DAT に変更します。

例

```
$ DECW$RGBPATH == "SYS$MANAGER:DECW$RGBPATH.DAT"
```

3.1.2.10 スクリーン・セーバ

スクリーン・セーバを有効にした場合、指定した時間 (デフォルトでは 10 分間) ユーザからの入力がないと、モニタの表示が消えます。実際の値は秒単位で指定します。

また、スクリーン・セーバのタイムアウト時間が経過したときに、サーバが黒または白の X というイメージを画面上にランダムなサイズと位置で表示するようにすることもできます。2 番目のパラメータには、スクリーン・セーバ・パターンの再描画間隔を指定します。

注意

画面の背景が表示されたままになるため、通常の使用ではこのスクリーン・セーバはお勧めできません。

DECW\$SERVER_SCREEN_SAVER_PREFER_BLANKING

このパラメータは、スクリーン・セーバの実行方法を決定します。この値が True (デフォルト) の場合、DECwindows サーバは、スクリーン・セーバタイムアウト時間が経過すると、ビデオ・デバイス・ドライバのビデオ信号をオフにします。この値が False の場合、DECwindows サーバは、タイムアウト時間が経過すると、画面を見えなくします。

次の論理名を定義すると、スクリーン・セーバ・タイムアウト時間が経過したときに、画面が消去されます。

例

```
$ DEFINE/SYSTEM DECW$SERVER_SCREEN_SAVER_PREFER_BLANKING F
```

DECW\$SERVER_SCREEN_SAVER_TIMEOUT

このパラメータは、スクリーン・セーバが起動されるまでの時間 (初期時間, 分単位) を定義します。その後、スクリーン・セーバ間隔が有効になります。デフォルトは 600 秒 (10 分) です。

次の論理名を定義すると、スクリーン・セーバの初期時間が 5 分になります。

例

```
$ DEFINE/SYSTEM DECW$SERVER_SCREEN_SAVER_TIMEOUT 5
```

DECW\$SERVER_SCREEN_SAVER_INTERVAL

このパラメータは、サーバが画面の背景を再描画するまでの時間 (分単位) を定義します。スクリーン・セーバは、画面の蛍光体が焼き付かないように、画面のピクセルの配置を変更します。サーバによっては、泳いでいる魚やロゴなど、無表示状態にしないスクリーン・セーバを使用します。デフォルトは 600 秒 (10 分) です。

次の論理名を定義すると、背景画面が 7 分おきに再描画されます。

例

```
$ DEFINE/SYSTEM DECW$SERVER_SCREEN_SAVER_INTERVAL 7
```

3.1.2.11 バッキング・ストアとセーブ・アンド

一般に、画面の隠れた領域が表示されたときにその表示を復元するのはクライアント・アプリケーションの責任です。ただし、ディスプレイ・サーバでは、クライアントに代わってこの操作を実行するためのいくつかの手法が利用できます。そのためにあるのがバッキング・ストア・オプションとセーブ・アンド・オプションです。

バッキング・ストアでは、ウィンドウの各部分が隠れる直前に、その内容がサーバのメモリに保存されます。後でその部分が表示されたときに、クライアントとは無関係にサーバが再描画することができます。これにより、特に複雑なグラフィックスを含

むウィンドウや、クライアント/サーバ間のリンクが低速な環境で、再描画に要する時間が大幅に短縮されます。

セーブ・アンドは同様のメカニズムの1つで、ウィンドウを表示する前に、それによって隠れる画面の部分がサーバによって保存されます。

これらのメカニズムは、どちらもデフォルトで利用可能で、ウィンドウごとにオンにすることができます。これらのメカニズムを頻繁に使用すると、サーバのメモリ使用量が増加します。クライアント・アプリケーションのデバッグ時に、アプリケーションがすべてのエクスポーズ・イベントを正しく処理することを確認する場合には、これらの機能を無効にすることをお勧めします。

注意

バックング・ストアとセーブ・アンドでは、どちらも DECwindows サーバに非常に高い負荷がかかるため、性能の低下を招くおそれがあります。

DECW\$SERVER_DEFAULT_BACKING_STORE

3種類のウィンドウ状態でこのオプションを有効にすることができます。次の表に、シンボルに指定できる値とその意味を示します。

値	意味
2	常にバックング・ストアを有効にする。
1	ウィンドウがマップされているときだけバックング・ストアを有効にする。
0	不定。使用するサーバとデバイスに依存する。

DECW\$SERVER_DISABLE_BACKING_STORE に値 True を設定すると、バックング・ストア・オプションが使用されなくなります。DECW\$SERVER_DEFAULT_BACKING_STORE の値は、アプリケーション内でウィンドウごとに設定変更することができます。

次の論理名を定義すると、ウィンドウが画面にマップされているときにサーバがウィンドウの保存および復元を行うようになります。

例

```
$ DEFINE/SYSTEM DECW$SERVER_DEFAULT_BACKING_STORE 1
```

DECW\$SERVER_DISABLE_BACKING_STORE

バックング・ストア機能を無効にする場合はこのパラメータを使用します。デフォルト値は False で、バックング・ストアは有効です。

DECW\$SERVER_DISABLE_BACKING_STORE に True (バッキング・ストアを無効にする) を設定してサーバを再起動すると、バッキング・ストア・オプションが無効になります。その後、バッキング・ストアを再度有効にするには、DECW\$SERVER_DISABLE_BACKING_STORE の割り当てを解除するか再定義してサーバを再起動します。

アプリケーション・ウィンドウが空白で表示される場合は、アプリケーションのバッキング・ストアの要求が正しくなく、エクスポーズ・イベントの処理が開始されるのを待っています。アプリケーションを修正してバッキング・ストアを要求しないようにするか、DECW\$SERVER_DISABLE_BACKING_STORE に True を設定して、DECwindows ディスプレイ・サーバを再起動します。

次のシンボルを定義すると、バッキング・ストアが無効になります。

例

```
$ DECW$SERVER_DISABLE_BACKING_STORE == "T"
```

DECW\$SERVER_DISABLE_SAVE_UNDER

セーブ・アンダ・オプションを有効にすると、別のウィンドウがその上に表示されたときに隠れる可能性があるウィンドウの情報が記録されます。このパラメータに True を指定すると、このオプションを無効にすることができます。

セーブ・アンダ・オプションは、バッキング・ストア・オプションに似ていますが、ウィンドウの隠れた部分だけがサーバによって保存されるため、ウィンドウのその部分が見えるようになると、サーバはウィンドウのその部分を再描画します。

次のようにシンボルを定義すると、サーバのセーブ・アンダ・オプションが無効になります。

例

```
$ DEFINE/SYSTEM DECW$SERVER_DISABLE_SAVE_UNDER T
```

3.1.2.12 エラー報告

ディスプレイ・サーバは、問題のトラブルシューティングを支援するために、情報メッセージとエラー・メッセージを記録するためのログ・ファイルを作成します。ログ・ファイルの名前は SYSSMANAGER:DECW\$SERVER_n_ERROR.LOG で、n はサーバ番号 (通常は 0) です。

ディスプレイ・サーバは、条件ハンドラを使用してエラー状態 (アクセス違反など) をトラップします。仮にこの処理がないと、ディスプレイ・サーバは停止します。条件ハンドラは、回復可能なエラーを検出すると、ディスプレイ・サーバの動作を続行させようとしています。条件ハンドラは、常にエラー・ログ・ファイルにエラーを記録します。条件ハンドラは、1 つのクライアントに対して複数 (デフォルトは 2) のエラーを検出すると、クライアントを切断します。

条件ハンドラがエラーから復旧すると、ディスプレイ・サーバはメモリなどのいくつかのリソースの状況が分からなくなります。そのため、このような例外が何度か (デフォルトは 10) 発生すると、条件ハンドラはシステム上のすべてのユーザに対して、ディスプレイ・サーバが縮退モードで動作している可能性があることを示すメッセージをブロードキャストして、ディスプレイ・サーバの再起動を提案します。このようなメッセージが表示された場合は、機会を見てディスプレイ・サーバを再起動してください。サーバの再起動方法については第 2.2 節を参照してください。

DECW\$SERVER_AUDIT_LEVEL

このパラメータは、通常のクライアント接続/切断メッセージをエラー・ログ・ファイルに記録するかどうかを制御します。このパラメータに指定できる値は以下のとおりです。

- 0 (無効)
- 1 (有効)
- 2 (有効: 成功メッセージを含む)
- 4 (有効: セキュリティ・ログを含む)

デフォルト値は 0 です。

次のようにシンボルを定義すると、最低限の監査ログが有効になります。

例

```
$ DECW$SERVER_AUDIT_LEVEL == "1"
```

DECW\$SERVER_BUG_COMPATIBILITY

X11R4 よりも前のサーバと X11 プロトコルの間には整合性がないため、いくつかの X リクエストで未定義のビットがオンになることがあります。しかし、プロトコルのこの部分を厳密に強制すると、未定義のビットをオンにしているアプリケーションは動作しなくなります。バグ互換性に True (デフォルト) を設定することで、サーバはこれらのアプリケーションの動作を引き続き許可します。ただし、X11 プロトコルに準拠するようにアプリケーションのコードを変更することをお勧めします。

次のようにシンボルを定義すると、X11R4 よりも前のサーバで X11 プロトコル・アプリケーションが正しく動作します。

例

```
$ DECW$SERVER_BUG_COMPATIBILITY == "T"
```

DECW\$SERVER_DUMP

このパラメータに True を設定すると、DCL RUN コマンドに/DUMP 修飾子が追加されます。これにより、サーバがクラッシュしたときにプロセス・ダンプが行われます。また、True を指定すると、サーバの条件ハンドラが自動的に無効になります (DECW\$SERVER_DISABLE_CH に True が設定されます)。

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

次のように論理名を定義すると、DCL RUN コマンドに/DUMP 修飾子が追加され、サーバの条件ハンドラが無効になります。

例

```
$ DEFINE DECW$SERVER_DUMP T
```

DECW\$SERVER_ERROR_LOG_TO_KEEP

このパラメータは、保存するエラー・ログ・ファイルのバージョン数を定義します。デフォルトは2バージョンです。

次のようにシンボルを定義すると、DECwindows サーバによって最新の3バージョンのエラー・ログ・ファイルが保存されるようになります。

例

```
$ DECW$SERVER_ERROR_LOG_TO_KEEP == "3"
```

DECW\$CLIENT_ERROR_THRESHOLD

このパラメータは、1つのクライアントでプロトコル・エラーがいくつ発生したらクライアントを終了させるかを定義します。デフォルトは1つのプロトコル・エラーです。

次のように論理名を定義すると、DECwindows ディスプレイ・サーバは、10個以上のプロトコル・エラーが発生したときにクライアントを終了させます。

例

```
$ DEFINE/SYSTEM DECW$CLIENT_ERROR_THRESHOLD 10
```

DECW\$SERVER_ERROR_THRESHOLD

このパラメータは、サーバ・エラーの総数がいくつになったら次のメッセージを報告するかを定義します (デフォルトは10)。

```
Server internal runtime error threshold exceeded (code = %x),  
server performance may be degraded.
```

シンボル%*x*は、エラーの原因となった条件コード (16進数) で置き換えられます。最も一般的な値は“c”で、アクセス違反を示します。

また、ログインしているすべての端末に次のメッセージがブロードキャストされます。

```
DECW$SERVER_ERROR: internal runtime error threshold exceeded.  
Performance may be degraded, restart DECwindows software when  
convenient by @SYS$MANAGER:DECW$STARTUP RESTART.
```

しきい値を超えるサーバ・エラーが5回発生するたびに、次のメッセージがログ・ファイル SYSSMANAGER:DECW\$SERVER_0_ERROR.LOG に書き込まれ、サーバに接続している端末にブロードキャストされます。

Server performance still degrading...

以下の3つのカテゴリのエラーがサーバの条件ハンドラで処理され、ログ・ファイルに記録されます。%dは10進数値、%xは16進数値です。

- クライアント・エラーしきい値に対してカウントされるエラー

- エラー・カウントがクライアント・エラーしきい値よりも小さい場合は、次のエラーが報告されます。

```
opcode %d is ignored due to internal runtime error %x
for client %d (#error = %d)
```

- エラー・カウントがクライアント・エラーしきい値以上の場合は、次のエラーが報告されます。

```
Client %d has made too many runtime errors %d,
its connection is marked for termination
```

- クライアントの切断原因となるエラー

```
Client %d has been disconnected due to unrecoverable
runtime error %x detected while processing opcode %d
```

サーバがサーバ拡張のオペレーション・コードを処理していた場合は、次のメッセージが表示されます。

```
Exception trapped while processing extension opcode %d
(extension id=%d)
```

- サーバの停止原因となるエラー

```
Unrecoverable server internal error (error code = %d) found,
terminating all connections.
```

```
.
.
.
```

```
List of active images and
call chain at time of the error
```

```
.
.
.
```

```
** SERVER INTERNAL RUNTIME ERROR ENCOUNTERED,
SERVER HAS JUST CRASHED!! **
*****
```

次のように論理名を定義すると、サーバ・エラーが20回発生した後は、DECwindows サーバはシステム縮退メッセージを報告します。

例

```
$ DEFINE/SYSTEM DECW$SERVER_ERROR_THRESHOLD 20
```

3.1.3 クラスタ共通パラメータの設定

DECW\$PRIVATE_SERVER_SETUP.COM を SYSS\$COMMON:[SYSMGR]に置くと、このファイル内のシンボル定義がシステム・ディスクを共有するすべてのシステム (通常はクラスタの全メンバ) に適用されます。

OpenVMS クラスタの全システムの DECwindows 環境をカスタマイズするには、以下の手順を実行します。

1. テンプレート・ファイルを SYSS\$COMMON:[SYSMGR]DECW\$PRIVATE_SERVER_SETUP.COM にコピーします。
2. コマンド・ファイルを編集して、*Cluster-Common or Standalone Workstation Setup* というセクションを探します。このセクションがない場合は、`node_list = ""`を探します。
3. このセクションにカスタマイズ内容を追加します。
4. 次のコマンドを実行して DECwindows ディスプレイ・サーバを再起動します。

```
$ @SYSS$STARTUP:DECW$STARTUP RESTART
```

カスタマイズ内容を有効にするシステムそれぞれで DECwindows を再起動する必要があります。

警告

DECwindows サーバを再起動すると、現在接続されているクライアント・プロセスがすべて切断されます。

3.1.4 スタンドアロン・システム・パラメータの設定

場合によっては、クラスタ内の 1 台以上のシステムに特殊なサーバ要件があることがあります。クラスタ内の特定のシステムをカスタマイズするには、以下の方法のいずれかを使用します。

- DECW\$PRIVATE_SERVER_SETUP.COM ファイルの専用のコピーを作成し、カスタマイズする各システムの SYSS\$SPECIFIC:[SYSMGR]に置きます。

- 以下のようにして、DECW\$PRIVATE_SERVER_SETUP.COM ファイル内のノード・リスト・ファシリティを使用します。
 1. DECW\$PRIVATE_SERVER_SETUP.COM を編集して、node_list = ""を探します。
 2. 次のように、引用符の間にクラスタ内のノードを追加します。

```
$ node_list = "NODE1/NODE2/NODE3"
```
 3. コマンド・プロシージャの最後に、ラベルとシステム固有の定義を追加します。

次の例は、NODE1、NODE2、および NODE3 のカスタマイズ方法を示します。

```
$!  
$! server symbol definitions for NODE1  
$!  
$DO_NODE1:  
$ DECW$SERVER_TRANSPORTS == "DECNET,LOCAL,TCPIP,LAT"  
$ EXIT  
$!  
$! server symbol definitions for NODE2  
$!  
$DO_NODE2:  
$ DECW$SERVER_DENSITY == "75"  
$ EXIT  
$!  
$! server symbol definitions for NODE3  
$!  
$DO_NODE3:  
$ DECW$DEFAULT_KEYBOARD_MAP == "US_LK201AA"  
$ EXIT
```

3.1.5 現在のサーバ・パラメータの確認

ここでは、現在のサーバ設定を確認するために使用できるコマンドの一覧を示します。

3.1.5.1 サーバ論理名テーブルの表示

DECwindows のスタートアップ処理で指定されたシンボルで定義されるサーバ論理名テーブル中のすべての論理名を表示するには、DCL コマンド SHOW LOGICAL を使用します。

以下の例は、コマンドの入力方法と、一般的な表示内容を示します。

```
$ SHOW LOGICAL/TABLE=DECW$SERVER0_TABLE  
(DECW$SERVER0_TABLE)
```

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

```
"DECW$COLOR" = "TRUE"
"DECW$DEFAULT_FONT" = "FIXED"
"DECW$DEFAULT_KEYBOARD_MAP" = "NORTH_AMERICAN_LK401AA"
"DECW$FONT" = "DECW$SYSCOMMON:[SYSFONT.DECW.USER_CURSOR32]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.CURSOR32]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.USER_CURSOR16]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.CURSOR16]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.USER_100DPI]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.100DPI]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.USER_75DPI]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.75DPI]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.USER_COMMON]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.COMMON]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.SPEEDO]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.TYPE1]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.USER_TYPE1]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.TRUE_TYPE]"
    = "DECW$SYSCOMMON:[SYSFONT.DECW.USER_TRUE_TYPE]"
"DECW$KEYBOARD" = "OPA2:"
"DECW$KEYMAP" = "DECW$SYSCOMMON:[SYS$KEYMAP.DECW.USER]"
    = "DECW$SYSCOMMON:[SYS$KEYMAP.DECW.SYSTEM]"
"DECW$MONITOR_DENSITY" = "100"
"DECW$POINTER" = "TTA0:"
"DECW$RGBPATH" = "SYS$MANAGER:DECW$RGB.DAT"
"DECW$SERVER_AUDIT_LEVEL" = "4"
"DECW$SERVER_BUG_COMPATIBILITY" = "Y"
"DECW$SERVER_DISABLE_BACKING_STORE" = "FALSE"
"DECW$SERVER_DISABLE_CH" = "N"
"DECW$SERVER_DISABLE_TEST" = "F"
"DECW$SERVER_ENABLE_ACCESSX" = "0"
"DECW$SERVER_EXTENSIONS" = "Xie"
    = "Multi-Buffering"
    = "Sec Xag"
"DECW$SERVER_KEY_REPEAT_DELAY" = "660"
"DECW$SERVER_KEY_REPEAT_INTERVAL" = "40"
"DECW$SERVER_SCREEN" = "GYA0"
"DECW$SERVER_TRANSPORTS" = "DECNET"
    = "LOCAL"
    = "TCPIP"
    = "LAT"
"DECW$SERVER_XKEYBOARD_COMPILED_DIR" = "SYS$COMMON:[SYS$KEYMAP.XKB.COMPILED]"
"DECW$SERVER_XKEYBOARD_DIRECTORY" = "DECW$SYSCOMMON:[SYS$KEYMAP.XKB]"
"DECW$SERVER_XKEYBOARD_LOAD_MAP" = "0"
"DECW$SERVER_XKEYBOARD_MAP" = "DIGITAL_US_LK201"
"DECW$SYSCOMMON" = "SYS$SYSROOT"
"DECW$XPORT_LRP_SIZE" = "32768"
"DECW$XSIZE_IN_PIXELS" = "1280"
"DECW$YSIZE_IN_PIXELS" = "1024"
```


3.1.5.2 X Display Information ユーティリティ (xdpyinfo) の利用

X Display Information ユーティリティ (xdpyinfo) を使用すると、サーバに直接照会し、さまざまなサーバ・パラメータを表示することができます。

このユーティリティを実行する前に、SET DISPLAY コマンドを使用して、正しいディスプレイが選択されていることを確認してください。

以下の例は、xdpyinfo の起動方法と、一般的な表示内容を示します。

```
$ SET DISPLAY/CREATE/NODE=node_name
$ RUN DECW$UTILS:XDOPYINFO

name of display:      _WSA1:
version number:      11.0
vendor string:       DECWINDOWS Hewlett-Packard Development Company OpenVMS
vendor release number:  8002
maximum request size: 65535 longwords (262140 bytes)
motion buffer size:  0
bitmap unit, bit order, padding:  32, LSBFirst, 32
image byte order:    LSBFirst
number of supported pixmap formats:  6
supported pixmap formats:
    depth 1, bits_per_pixel 1, scanline_pad 32
    depth 4, bits_per_pixel 8, scanline_pad 32
    depth 8, bits_per_pixel 8, scanline_pad 32
    depth 12, bits_per_pixel 32, scanline_pad 32
    depth 24, bits_per_pixel 32, scanline_pad 32
    depth 32, bits_per_pixel 32, scanline_pad 32
keycode range:      minimum 8, maximum 255
number of extensions:  17
    DEC-Server-Mgmt-Extension
    ServerManagementExtension
    SHAPE
    MIT-SHM
    Extended-Visual-Information
    XTEST
    BIG-REQUESTS
    MIT-SUNDRY-NONSTANDARD
    MIT-SCREEN-SAVER
    SYNC
    XC-MISC
    TOG-CUP
    Xie
    DEC-XTRAP
    Multi-Buffering
    SECURITY
    XC-APPGROUP
default screen number:  0
number of screens:     1
```

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

```
screen #0:
  dimensions: 1280x1024 pixels (325x260 millimeters)
  resolution: 100x100 dots per inch
  depths (1): 8
  root window id: 0x2e
  depth of root window: 8 planes
  number of colormaps: minimum 1, maximum 1
  default colormap: 0x21
  default number of colormap cells: 256
  preallocated pixels: black 0, white 1
  options: backing-store YES, save-unders YES
  current input event mask: 0x0
  number of visuals: 10
  default visual id: 0x22
  visual:
    visual id: 0x22
    class: PseudoColor
    depth: 8 planes
    size of colormap: 256 entries
    red, green, blue masks: 0x0, 0x0, 0x0
    significant bits in color specification: 8 bits
  visual:
    visual id: 0x23
    class: PseudoColor
    depth: 8 planes
    size of colormap: 256 entries
    red, green, blue masks: 0x0, 0x0, 0x0
    significant bits in color specification: 8 bits
  visual:
    visual id: 0x24
    class: DirectColor
    depth: 8 planes
    size of colormap: 8 entries
    red, green, blue masks: 0xe0, 0x1c, 0x3
    significant bits in color specification: 3 bits
  visual:
    visual id: 0x25
    class: GrayScale
    depth: 8 planes
    size of colormap: 256 entries
    red, green, blue masks: 0x0, 0x0, 0x0
    significant bits in color specification: 8 bits
  visual:
    visual id: 0x26
    class: StaticGray
    depth: 8 planes
    size of colormap: 256 entries
    red, green, blue masks: 0x0, 0x0, 0x0
    significant bits in color specification: 8 bits
  visual:
    visual id: 0x27
    class: StaticColor
    depth: 8 planes
    size of colormap: 256 entries
```

```

red, green, blue masks:  0xe0, 0x1c, 0x3
significant bits in color specification:  8 bits
visual:
visual id:  0x28
class:  TrueColor
depth:  8 planes
size of colormap:  8 entries
red, green, blue masks:  0xe0, 0x1c, 0x3
significant bits in color specification:  3 bits
visual:
visual id:  0x29
class:  TrueColor
depth:  8 planes
size of colormap:  8 entries
red, green, blue masks:  0xe0, 0x1c, 0x3
significant bits in color specification:  3 bits
visual:
visual id:  0x2a
class:  TrueColor
depth:  8 planes
size of colormap:  8 entries
red, green, blue masks:  0xe0, 0x1c, 0x3
significant bits in color specification:  3 bits
visual:
visual id:  0x2b
class:  TrueColor
depth:  8 planes
size of colormap:  8 entries
red, green, blue masks:  0xe0, 0x1c, 0x3
significant bits in color specification:  3 bits

```

3.1.5.3 X 設定ユーティリティ (xset) の利用

サーバに直接パラメータ設定を照会するには、X 設定 (xset) ユーティリティを使用します。現在のフォント・パスを確認するには、xset を実行することをお勧めします。

このユーティリティを実行する前に、SET DISPLAY コマンドを使用して、正しいディスプレイが選択されていることを確認してください。

以下の例は、xset の起動方法と、一般的な表示内容を示します。

ディスプレイ・サーバの構成

3.1 DECwindows ディスプレイ・サーバのカスタマイズ

```
$ SET DISPLAY/CREATE/NODE=node_name
$ MCR DECW$UTILS:XSET Q

Keyboard Control:
  auto repeat: on    key click percent: 25    LED mask: 00000000
  auto repeating keys: 000000000000000000
                    0000c0ffffffffffff
                    ffffffff27f8ff
                    ffffffff
  bell percent: 0    bell pitch: 400    bell duration: 100
Pointer Control:
  acceleration: 7/1    threshold: 3
Screen Saver:
  prefer blanking: yes    allow exposures: yes
  timeout: 600    cycle: 600
Colors:
  default colormap: 0x21    BlackPixel: 0    WhitePixel: 1
Font Path:
DECW$SYSCOMMON: [SYSFONT.DECW.CURSOR32],DECW$SYSCOMMON: [SYSFONT.DECW.CURSOR16],
DECW$SYSCOMMON: [SYSFONT.DECW.100DPI],DECW$SYSCOMMON: [SYSFONT.DECW.75DPI],
DECW$SYSCOMMON: [SYSFONT.DECW.USER_COMMON],DECW$SYSCOMMON: [SYSFONT.DECW.COMMON],
DECW$SYSCOMMON: [SYSFONT.DECW.SPEEDO],DECW$SYSCOMMON: [SYSFONT.DECW.TYPE1],
DECW$SYSCOMMON: [SYSFONT.DECW.TRUETYPE],
CDE$SYSTEM_DEFAULTS: [CONFIG.XFONTS.C.100DPI],
CDE$SYSTEM_DEFAULTS: [CONFIG.XFONTS.C.75DPI],
CDE$SYSTEM_DEFAULTS: [CONFIG.XFONTS.C]
Bug Mode: compatibility mode is enabled
```

3.2 ネットワーク・トランスポートの指定

第 1 章で概要を説明したように、DECwindows トランスポート・インタフェースは、X プロトコル・リクエストをクライアントとサーバの間でやり取りするための、汎用のデータ伝送メカニズムです。

ここでは、サポートされるトランスポートの概要と、X ディスプレイ・サーバで利用できるようにする方法について説明します。クライアントのトランスポートを設定する方法については、第 4.1 節を参照してください。

3.2.1 ローカル・トランスポートの利用

ローカル・トランスポートはデフォルトのネットワーク・トランスポートであり、常に利用可能です。DECwindows クライアントとディスプレイ・サーバが同じ OpenVMS システムで動作している場合には、ローカル・トランスポートを使用します。ローカル・トランスポートでは、クライアントとサーバの間で共用メモリを介してより直接的にデータが転送されるため、一般に性能が向上します。このメカニズムにより、システム内でデータをコピーする回数が減り、ネットワーク・アクセスによる余分なオーバーヘッドが排除できます。

3.2.2 DECnet トランスポートの利用

DECwindows では、DECnet トランスポートもサポートされています。このトランスポートを有効にして使用する前に、HP DECnet Phase IV for OpenVMS ソフトウェアまたは HP DECnet-Plus (Phase V) for OpenVMS ソフトウェアをシステムにインストールして動作させておく必要があります。

ネットワーキング・ソフトウェアのインストールおよび実行方法については、DECnet 製品のマニュアルを参照してください。

注意

サーバに接続しているときに DECnet または TCP/IP がシャットダウンされると、ネットワークが再開されたときに再接続するために、トランスポートはネットワークに対して継続的にポーリングを行います。

3.2.3 TCP/IP トランスポートの利用

第 4 章で説明したように、DECwindows Motif は、IPv4 (Internet Protocol Version 4) と IPv6 (Internet Protocol Version 6) の両方について TCP/IP トランスポートの実装をサポートします。このトランスポートを有効にして使用する前に、HP TCP/IP Services for OpenVMS ソフトウェアまたはサポートされている他社製の TCP/IP 製品をインストールし、システムのスタートアップの一環として実行する必要があります。

TCP/IP Services ソフトウェアは、デフォルトでシステムのスタートアップ時に起動するように構成されています。他社の製品を使用している場合は、ソフトウェアのインストール方法と構成方法について、製品のマニュアルを参照してください。

HP TCP/IP Services for OpenVMS ソフトウェアを、X プロトコルをサポートするための最小限の DECwindows の要件を満たすように構成することで、メモリとプロセス・スロットを有効に利用することができます。DECwindows に必要なものは、INET_ACP が動作していることだけです。TCP/IP の概念とネットワーク・ソフトウェアの構成方法についての詳細は、TCP/IP Services for OpenVMS のマニュアルを参照してください。

3.2.4 LAT トランスポートの利用

DECwindows では、他の OpenVMS Alpha ワークステーションおよび OpenVMS I64 ワークステーションに表示するためのネットワーク・トランスポート・メカニズムとして、LAT をサポートしています。LAT トランスポートを有効にして使用する前に、DECwindows のクライアントとサーバ・システムの両方で LAT ソフトウェアを起動する必要があります。LAT ソフトウェアの起動と構成についての詳細は、『LATCP Utility Reference Manual』を参照してください。

ディスプレイ・サーバの構成

3.2 ネットワーク・トランスポートの指定

LAT をトランスポートとして使用するためには、ディスプレイ・サーバ・システムで LAT サービス XSERVER が動作している必要があります。DECwindows のスタートアップ・プロシージャで強制的に LAT サービスを作成するには、次の論理名を定義します。

```
$ DEFINE /SYSTEM DECW$INSTALL_XTERMINAL SERVER
```

この論理名は、DECwindows を起動する前に定義する必要があります。この論理名は、SYSSMANAGER:SYLOGICALS.COM で定義することをお勧めします。

3.2.5 デフォルト・トランスポートの変更

DECnet トランスポートとローカル・トランスポートはデフォルトで有効になっています。ディスプレイ・サーバで使用するトランスポートを有効または無効にするには、DECW\$PRIVATE_SERVER_SETUP.COM ファイルを変更して、DECW\$SERVER_TRANSPORTS パラメータを再定義します。

例 3-1 に、TCP/IP 接続とローカル接続を使用し、DECnet 接続は使用しないようにシステムを設定する例を示します。

例 3-1 トランスポート接続の設定例

```
$do TCP/IP:  
$ decw$server_transports == "TCP/IP,LOCAL"  
$ exit  
$ !
```

3.3 サーバ・アクセス制御の設定

ここでは、DECwindows Motif がサポートしているアクセス制御方式と、それを使用して X ディスプレイ・サーバへのアクセスを制御する方法について説明します。

3.3.1 ユーザ・ベースのアクセス制御

ユーザ・ベースのアクセス制御では、ホスト、トランスポート、およびユーザ名の組み合わせ (たとえば DECNET ZEPHYR JONES) に対して X ディスプレイ・サーバへのアクセスが許可されます。指定したユーザ名、ノード名、およびトランスポート情報は、選択したユーザ・クラス以外のユーザを遮断するフィルタとして機能します。

ユーザ・ベースのアクセス制御は、許可ユーザ・リストまたはアクセス許可リストのどちらかにエントリがあるかぎり常に有効です。この形式のアクセス制御は、暗号化機能がなく、TCP/IP 環境でユーザ名を指定することができず、最もセキュリティが低いいため、ローカル、DECnet、または LAT 環境でアクセスを許可するためにだけ使用することをお勧めします。

3.3.2 トークン・ベースのアクセス制御

トークン・ベースのアクセス制御では、接続要求時にクライアント・アプリケーションから提示されたパスワードまたはトークンに基づいて X ディスプレイ・サーバへのアクセスが許可されます。クライアントの認証レベルと認証方法は、Magic Cookie (MIT-MAGIC-COOKIE-1) と Kerberos (MIT-KERBEROS-5) のどちらのプロトコルを使用するかで変わります。

一般に、トークン・ベースのアクセス制御で保護されたサーバにクライアント・アプリケーションがアクセスしようとするたびに、サーバは X authority ファイルを参照して、使用する適切なプロトコルと、接続を許可するために使用する認証方法を判断します。

トークン・ベースのプロトコルでは、より高い保護が得られるだけでなく、オープンされたサーバ接続上で実行できる操作についてもより詳細な制御が可能です。たとえば、トークンを使用して信頼特権を許可または拒否することができます。X ディスプレイ・サーバへの信頼されていない接続では、その接続上で実行できる操作が大幅に制限されます。

注意

XDM-AUTHORIZATION-1 や SUN-DES-1 などのその他の X ウィンドウ・システムのセキュリティ・プロトコルは、現在サポートされていません。これらのプロトコルを使用して DECwindows の X ディスプレイ・サーバにアクセスする他社製のクライアント・アプリケーションは、デフォルトでユーザ・ベースのアクセス制御になります。

3.3.2.1 Magic Cookie

Magic Cookie は、ホスト・ベースのセキュリティ・メカニズムに代わるより安全なメカニズムを提供するために設計されたもので、X ウィンドウ・システムの以前のリリースでも利用可能です。このプロトコルは、トークン・ベースのアプローチとしては最初のものであり、X ディスプレイ・サーバへのアクセスをユーザ・レベルで制限するための標準的な手段として初めて提供されました。

Magic cookie では、cookie と呼ばれる、クライアントがサーバに対して提示するバイナリ数値に基づいてサーバへの接続が許可されます。通常、クライアントはこの cookie 値を X authority ファイルから取得しますが、プログラムによっては他の方法も使用します。

X authority ファイル内の Magic Cookie アクセス制御の各エントリには、以下の内容を指定します。

X ディスプレイ・デバイスのアドレス
プロトコル名 (MIT-MAGIC-COOKIE-1)
ランダムな数値の cookie 値

ディスプレイ・サーバの構成

3.3 サーバ・アクセス制御の設定

DECwindows Motif セッションで接続を許可するために Magic Cookie を使用する場合、ユーザがローカルの DECwindows Motif デスクトップに正常にログインするたびに cookie が生成されます。ローカル接続を許可する magic cookie は、デバイス、トランスポート、プロトコル名とともに X ディスプレイ・サーバに渡され、最新の X authority ファイル (SYSS\$LOGIN:DECWS\$XAUTHORITY.DECWS\$XAUTH) に保存されます。

セッション中にクライアント・アプリケーションがディスプレイ・サーバに接続を試みるたびに、アプリケーションは接続要求とともに有効な cookie をサーバに提示する必要があります。cookie が、X ディスプレイ・サーバが保持しているいずれかの cookie と一致した場合、接続とアクセスが許可されて、ディスプレイがオープンされます。ユーザが DECwindows Motif セッションからログアウトすると、サーバはリセットされ、cookie は破棄されます。

ランダムに生成された値を使用するため、Magic Cookie はユーザ・ベースの方式よりも安全な形のアクセス制御を提供します。ただし、cookie は暗号化されずにネットワーク上でやり取りされるため、傍受されやすくなっています。したがって、この形のアクセス制御は、ローカル・エリア・ネットワーク (LAN)、限定された DECnet 環境、他の方法で保護された TCP/IP ネットワーク上で接続を許可するために使用することをお勧めします。

3.3.2.2 Kerberos

Kerberos は、X ディスプレイ・サーバへの接続を、以下の要素の組み合わせに基づいて承認します。

X authority ファイル中のプロトコル名 (MIT-KERBEROS-5)
有効な Kerberos プリンシパルとそれに関連付けられたパスワードのリスト
有効な証明書の提示

Kerberos 証明書 (チケット) は、プリンシパルの身元を確認するために使用する一連の電子情報です。これらのプリンシパルは、サーバ・システムに保管されている許可プリンシパル・リストに格納されます。Kerberos では、有効なプリンシパルで動作するクライアント・アプリケーションは、Kerberos で保護された X ディスプレイ・サーバに接続しようとするたびに、Kerberos KDC (Key Distribution Center) に対して、チケット要求を送信します。

サーバで Kerberos のアクセス制御を有効にすると、ユーザがローカルのデスクトップにログインするたびに、自動的に新しいチケットが KDC に対して要求されます。KDC は、ユーザのプリンシパル名に関連付けられたチケット保証チケット (TGT: Ticket Granting Ticket) を作成し、パスワードを鍵としてそれを暗号化し、暗号化された TGT を返します。

TGT が正常に復号化されると、ユーザは認証され TGT はキャッシュされます。TGT は認証されたプリンシパルが追加のチケットを取得するのを許可します。これらの追加のチケットは、特定のサービス (この場合は他のクライアント・アプリケーション

からの X ディスプレイ・サーバへのアクセス) を許可します。これらの追加のチケットの要求と許可は、透過的に行われます。

DECwindows Motif では、ユーザごとの認証が使用されています。このモデルでは、クライアントとサーバの両方が接続の両端で Kerberos クライアントを使用して、ユーザの身元 (プリンシパル) を確認します。接続の両端でプリンシパルが認証されると、サーバへのアクセスが許可されます。

デフォルトでは、各 TGT は一定の時間で期限切れとなります。TGT が期限切れとなるか信用できなくなった場合、Kerberos ログインを強制することで現在の TGT を破棄して新しい TGT を生成することができます。

Kerberos は、要求元のクライアントとサーバ・システム間での初期認証情報を暗号化するため、最も安全なアクセス制御方式です。したがって、インターネットなどの保護されていないネットワーク上でリモート・クライアント接続を許可するための方法としてお勧めします。

注意

Kerberos は、ネットワーク接続上でやり取りされるすべてのデータを暗号化するために使用するセッション鍵を生成するように設計されています。X ウィンドウ・システムでは、この鍵を使用して最初の認証メッセージだけが暗号化されます。いったんクライアントの身元が確認されると、以降のデータは暗号化されずにネットワーク・チャネル上で送信されます。そのため、サーバ自体はネットワーク・レベルの攻撃を受けることがあります。

3.3.3 X authority ファイル

X authority ファイルは、X ディスプレイ・サーバへの接続を許可するために使用する情報が格納されたバイナリ・データ・ファイルです。クライアント・アプリケーションは、X サーバに接続しようとするたびに、最新の X authority ファイルを参照して、接続を認証するために適用する適切な承認キーを決定します。

各承認キーは、プロトコル名とトークンで構成され、使用するプロトコルに応じて以下のいずれかになります。

- MIT-MAGIC-COOKIE-1 + cookie 値
- MIT-KERBEROS-5 + 暗号化された TGT (個別にキャッシュされる)

デフォルトでは、Magic Cookie または Kerberos アクセス制御が構成されたシステム上のデスクトップにユーザが最初にログインしたときに、自動的に X authority ファイルが作成されます。このファイルはそのユーザの OpenVMS ログイン・ディレクトリ (SYS\$LOGIN:DECW\$XAUTHORITY.DECW\$XAUTH) に保存されます。以降ユーザがそのシステムのデスクトップにログインするたびに、新しい承認キーが生成され、X ディスプレイ・サーバに渡されて、ユーザの X authority ファイルに書き込ま

れます。このキーは、DECwindows Motif セッションの間、X ディスプレイ・サーバへのアクセスを制御します。

通常の DECwindows Motif のログイン処理とは異なるサーバへのアクセスが必要なクライアント・アプリケーション用に、別の X authority ファイルを手動で定義することができます (DECW\$SERVER_XAUTHORITY シンボルを使用します)。

セキュリティ拡張が有効な場合、承認キーを手動で生成することもできます。手動で生成したキーを使用して、サーバへのアクセスをさらに制限することができます。生成されたキーはクライアント・システム上の X authority ファイルに格納され、指定されたディスプレイ・サーバに対する既存の値を上書きします。特定のサーバへの接続を許可するために、キーをさまざまなクライアント・システムに配布したり、以降の接続を止めるために破棄することができます。

生成されたキーには、承認 ID が割り当てられます。これによって、キーとそれを生成したユーザが関連付けられます。その結果、キーを生成したユーザだけがキーを破棄することができます。

3.3.4 アクセス許可ファイル

アクセス許可ファイルは、X ディスプレイ・サーバへの追加の OpenVMS ユーザ・アクセスをサーバのスタートアップ時に自動的に許可するための ASCII テキスト・ファイルです。

アクセス許可設定は、DECwindows Motif デスクトップにユーザがログインするまで有効です。ユーザがデスクトップにログインして DECwindows Motif セッションが開始されると、セッション・マネージャでそのユーザ用に定義されたセキュリティ・オプションが適用されます。

ユーザがセッションを終了すると、サーバは再度初期化され、アクセス許可設定が復元されます。

警告

アクセス許可ファイルは、通常は DECwindows Motif のログイン処理を使用しないワークステーションで使用するために設計されています。このファイルを使用すると DECwindows Motif システムのセキュリティが低下するおそれがあるため、DECwindows Motif のログイン処理を使用して X サーバへのアクセスを制限するシステムでは、このファイルを使用しないでください。

たとえば、アクセス許可ファイルでアクセスを許可されたユーザがログイン・ウィンドウを偽装して、DECwindows Motif デスクトップにログインしようとした他のユーザのパスワードを読み取るおそれがあります。

3.3.5 アクセス信頼ファイル

信頼されたユーザとは、セキュリティ設定の変更を許可されたユーザです。アクセス信頼ファイルは、特定のディスプレイ・サーバのアクセス制御設定を変更することができる OpenVMS ユーザを識別する ASCII テキスト・ファイルです。

デフォルトでは、ローカルの SYSTEM アカウントに対して信頼権限が与えられます (ローカル・トランスポートまたは DECnet トランスポート上で)。このファイルのエントリは、トークン・ベースの認証方式が有効になっていないかぎり、アクセス許可ファイルに自動的に追加されます。トークン・ベースの認証方式が有効な場合は、アクセス許可リストに手動でエントリを追加するか、適切な X authority ファイルにエントリを追加することで、信頼されたユーザによる X ディスプレイ・サーバへのアクセスを許可する必要があります。

アクセス信頼設定は、アクセス許可ファイルの設定と同様に、ユーザが DECwindows Motif デスクトップにログインするまで有効です。

3.3.6 アクセス制御方法の選択

X ディスプレイ・サーバのアクセス制御を構成するときには、ネットワーク環境に応じて 1 つ以上の方式を組み合わせて適用します。たとえば、TCP/IP 経由でのすべてのリモート・サーバ接続を Kerberos で承認し、LAN ネットワーク接続を Magic Cookie で承認することができます。

組み合わせて使用する場合は、クライアントから提示された最も制限の厳しいアクセス制御方式が常に優先されます。たとえば、サーバで 3 種類の方式がすべて有効になっている場合で、要求元のクライアントが Magic Cookie を使用している場合、サーバは Magic Cookie で接続を承認しようとします。Magic Cookie のアクセス制御では、デフォルトではユーザ・ベースのアクセス制御が使用可能である点に注意してください。クライアントがトークン・ベースの方式を使用してサーバに接続しようとして失敗した場合でも、許可ユーザ・リストのメンバになっていれば、アクセスが許可されます。

ただし、アクセス制御方式を有効にする前に、以降の項で説明するように、まずサーバ接続環境を確認する必要があります。たとえば、いくつかの DECwindows Motif システムは、デスクトップ・セッション外でのみアプリケーションを実行します。そのようなシステムでは、DECwindows Motif セッション外で行われる接続だけにアクセス制御を適用する必要があります。デスクトップ・セッションの内部と外部の両方の接続に対してアクセス制御を適用すると、最初の DECwindows Motif ログイン処理でログインできなくなる可能性があります。

3.3.6.1 デスクトップ・セッション外の接続にアクセス制御を適用する

DECwindows Motif デスクトップ・セッション外でアクセス制御を有効にすると、許可された OpenVMS ユーザが、ログイン処理を行わずにシステム上でクライアント・アプリケーションを実行できるようになります。このタイプのアクセス制御は、一般に、対話型の DECwindows Motif ワークステーションではなく、スタンドアロンの X ディスプレイ・サーバとして機能するシステムで使用されます。

サーバ・カスタマイズ・パラメータとアクセス許可ファイルまたは X authority ファイルを使用してアクセス制御を設定します。

3.3.6.2 デスクトップ・セッション内の接続にアクセス制御を適用する

DECwindows Motif セッション内でアクセス制御方式を有効にするには、「Security Options」ダイアログ・ボックスを使用します。ダイアログ・ボックスのオプションでは、ローカルの X ディスプレイ・サーバで使用するアクセス制御方式の設定、ディスプレイ・サーバにアクセスする他のユーザの許可、ローカルのクライアント・アプリケーションがディスプレイ・サーバに接続する際に使用する方式の指定を行うことができます。

セッション・マネージャ (Traditional Desktop) とスタイル・マネージャ (New Desktop) のどちらからアクセスした場合でも、「Security Options」ダイアログ・ボックスの設定は同じです。ただし、デスクトップによって設定の保存方法が異なります。

- Traditional Desktop での設定は、変更が適用されるとすぐに DECWSSMB_SECURITY.DAT ファイルに保存されます。
- New Desktop での設定は、セッションが保存される際 (新しいホーム・セッションを保存する際または現在のセッションを保存する際) に保存されます。New Desktop では、次回ログイン時にホーム・セッションの復元を選択すると、ホーム・セッションを更新して保存しないかぎり、変更内容は失われます。

3.3.7 ユーザ・ベースのアクセス制御を有効にする

ここでは、ユーザ・ベースのアクセス制御を、デスクトップ・セッションの外部と内部のサーバ接続に適用する方法を説明します。

TCP/IP サービス・トランスポート上でのユーザ・ベースのアクセス制御では、`getnameinfo` 関数を使用してピア・システムの名前を取得し、この値とユーザが指定した値を突き合わせます。アドレスが IPv4 射影 IPv6 アドレスの場合、`AF_INET` ファミリと、IPv4 射影 IPv6 アドレスから抽出した IPv4 アドレスを使用して `getnameinfo` が呼び出されます。それ以外の場合は、`AF_INET6` と IPv6 アドレスを使用して `getnameinfo` 関数が呼び出されます。名前の逆引きに失敗した IPv4 接続は、IPv4 形式で表現されます。名前の逆引きに失敗したその他すべての接続は、IPv6 形式で表現されます。

使用しているアドレスがリンク・ローカルの場合、TCP/IP サービスの `getnameinfo` 関数が返す文字列にはスコープ ID が含まれます。サーバはこれを許可ユーザ・リスト内のホスト名と突き合わせます。これは、エントリにスコープ ID が含まれているかどうかには関係なく実行されます。たとえば、戻り値 `test12i6%WE1` は、`test12i6%WE1` または `test12i6` のどちらかに一致します。

デスクトップ・セッション外のサーバ接続の場合

ユーザ・ベースのアクセス制御を、DECwindows Motif セッション外のサーバ接続に対して適用するには、以下の手順を実行します。

警告

アクセス許可ファイルのエントリを使用して X サーバに対する TCPIP ホスト接続を許可すると、DECwindows Motif システムに対して認証されていないアクセスが可能になります。これにより、システムが意図しない侵入、サービス停止 (DoS) 攻撃、データ損失に対して無防備になります。

適切なレベルのシステム・セキュリティを保つには、トークン・ベースの方式 (Magic Cookie や Kerberos など) を使用して TCP/IP 上での X サーバに対するリモート・アクセスを制限することをお勧めします。これらの方式では、システムの高度な保護が可能となるだけでなく、ユーザごとにアクセスを許可 (または拒否) することも可能になります。

1. `SYSSMANAGER:DECW$PRIVATE_SERVER_SETUP.COM` ファイルを変更して、ファイルが格納されている場所を指すように `DECW$SERVER_ACCESS_ALLOWED` パラメータおよび `DECW$ACCESS_TRUSTED` パラメータの値を定義します。以下に例を示します。

```
$ DECW$SERVER_ACCESS_ALLOWED == "SYSSMANAGER:DECW$SERVER1_ACCESS_ALLOWED.DAT"  
$ DECW$SERVER_ACCESS_TRUSTED == "SYSSMANAGER:DECW$SERVER1_ACCESS_TRUSTED.DAT"
```

2. アクセス許可ファイルおよびアクセス信頼ファイルを作成して編集し、適切なユーザ・エントリを追加します。各エントリは、*transport-host-username* の形式で記述します。トランスポートとして指定できるのは、DECNET、TCPIP、および LOCAL です。表 4-1 に示すトランスポートの同義語 TCP、INET、INET6、および DNET はサポートされていません。

たとえば、次のエントリはユーザ JONES にサーバへのローカル・アクセスと DECnet トランスポート経由でのノード ZEPHYR からのネットワーク・アクセスを許可します。

```
DECNET ZEPHYR JONES  
LOCAL 0 JONES  
.  
.  
.
```

ネットワーク・トランスポートとしてTCP/IPを使用する場合は、アクセス特権と信頼特権は、個別のユーザに対してではなくホストに対してだけ割り当てることができる点に注意してください。TCP/IPでは、リモート接続時に渡されるデータの中にユーザ指定が含まれていません。

その結果、TCP/IPを使用したホストのファイル・エントリでは、ユーザ指定にアスタリスク(*)を含める必要があります。これにより、TCP/IPを使用してXディスプレイ・サーバに接続する際に、特定のホスト・システムのすべてのユーザにアクセス特権が与えられます。たとえば、次のエントリは、ノードZEPHYR上のすべてのユーザに対してTCP/IPトランスポート経由でのアクセスを許可します。

```
TCP/IP ZEPHYR *  
.  
.  
.
```

3. ファイルを保存してサーバを再起動します。新しいアクセス特権と信頼特権は、システムのスタートアップ時に自動的に適用されます。

デスクトップ・セッション内の接続の場合

DECwindows Motif デスクトップ・セッション内のサーバ接続に対してユーザ・ベースのアクセス制御を適用するには、以下の手順を実行します。

1. デスクトップに応じて以下のどちらかを実行します。
 - Traditional Desktop の場合、セッション・マネージャの[オプション]メニューから[セキュリティ...]を選択します。
 - New Desktop の場合、スタイル・マネージャの[J-Security]コントロールをクリックします。「Security Options」ダイアログ・ボックスが表示されます。
2. 「Server Access Control」領域で、[Users...]をクリックし、「Configure Users」ダイアログ・ボックスを表示します。
3. 承認するユーザのノード、ユーザ名、トランスポートを入力します。
4. [Add]ボタンをクリックします。ユーザが「Authorized Users」リストに追加されます。
5. [了解]をクリックして変更を保存および適用し、「Configure Users」ダイアログ・ボックスを閉じます。

ユーザ・ベースのアクセス制御を無効にするには、すべてのユーザを「Authorized Users」リストから削除する必要があります。

ユーザ名を削除するにはまず削除するユーザ名をクリックし、[Remove]ボタンをクリックします。最後に[了解]または[適用]をクリックします。これでユーザはシステムへのアクセスが許可されなくなります。

3.3.8 Magic Cookie によるアクセス制御を有効にする

ここでは、デスクトップ・セッションの外部と内部のサーバ接続に対して Magic Cookie によるアクセス制御を適用する方法を説明します。

デスクトップ・セッション外の接続の場合

Magic Cookie によるアクセス制御を DECwindows Motif セッション外のサーバ接続に適用するには、以下の手順を実行します。

1. SYSTEM アカウントまたは特権を持つ別のアカウントでログインします。
2. DECW\$PRIVATE_SERVER_SETUP.COM ファイルを変更し、X authority ファイルが格納される場所を指すように DECW\$SERVER_XAUTHORITY パラメータの値を定義します。次に例を示します。

```
$ DECW$SERVER_XAUTHORITY == "SYS$MANAGER:SERVER_ZEPHYR.DECW$XAUTH"
```

3. X authority ファイル・ユーティリティ (xauth) を使用して、X authority ファイルをサーバ用に手動で作成し、適切なエントリを追加します。たとえば、次のコマンドは新しい X authority ファイル SERVER_ZEPHYR.DECW\$XAUTH を作成し、ローカル・トンランスポート用のエントリを追加し、Magic Cookie プロトコルを指定し、cookie 値 12345abcdef56789 を割り当てます。

```
$ XAUTH -f SYS$SYSROOT:[SYSMGR]SERVER_ZEPHYR.DECW$XAUTH ADD -  
_ $ :0 MIT-MAGIC-COOKIE-1 12345abcdef56789
```

このファイル内の cookie は X サーバのスタートアップ時にロードされ、すべてのクライアント接続を承認するために使用されます。

4. ファイルを保存しサーバを再起動します。
5. xauth ユーティリティを使用して、キーをすべてのクライアント・システムに配布します。xauth ユーティリティの使用方法についての詳細は、『HP DECwindows Motif for OpenVMS New Features』またはオンライン・ヘルプを参照してください。

デスクトップ・セッション内の接続の場合

DECwindows Motif セッション内のサーバ接続に対して、Magic Cookie によるアクセス制御を適用するには、以下の手順を実行します。

1. デスクトップに応じて以下のどちらかを実行します。
 - Traditional Desktop の場合、セッション・マネージャの[オプション]メニューから[セキュリティ...]を選択します。
 - New Desktop の場合、スタイル・マネージャの[J-Security]コントロールをクリックします。「Security Options」ダイアログ・ボックスが表示されます。
2. 「Server Access Control」領域で、「Magic Cookie」オプションを選択します。
3. [了解]をクリックして変更を保存および適用し、「Security Options」ダイアログ・ボックスを閉じます。

4. 有効にすると、デスクトップにログインするたびに cookie が生成されます。他のユーザに対して X ディスプレイ・サーバへのアクセスを許可するには、xauth ユーティリティを使用して、cookie をそのユーザの X authority ファイルに配布します。

xauth ユーティリティの使用方法についての詳細は、『HP DECwindows Motif for OpenVMS New Features』またはオンライン・ヘルプを参照してください。

Magic Cookie を無効にするには、「Magic Cookie」オプションの選択を解除し、[了解]または[適用]をクリックします。

他のユーザが現在の cookie 値を使用して現在のセッションにアクセスできないようにするには、[Create Cookie]ボタンをクリックします。新しい cookie 値がデフォルトの X authority ファイルに追加されます。

注意

新しい cookie が生成されるときに X ディスプレイ・サーバに接続しているクライアント・アプリケーションは接続されたままです。認証は、最初にサーバに接続する際にだけ実行されます。

3.3.9 Kerberos によるアクセス制御を有効にする

Kerberos を有効にするためには、サーバ・システムであらかじめ以下の手順を実行しておく必要があります。

1. TCP/IP for OpenVMS ソフトウェアを、ドメイン名サーバを有効にしてインストールし構成します。
2. 『Kerberos Client for OpenVMS Installation Guide and Release Notes』に従って、Kerberos Client for OpenVMS ソフトウェアをインストールして構成します。
3. 以下の情報を準備します。
 - KDC の場所
 - プリンシパルを追加するために使用するノード、ドメイン、レルム情報
 - プリンシパル名とパスワード
4. SYSSMANAGER:DECW\$PRIVATE_SERVER_SETUP 内で DECW\$SERVER_TRANSPORTS パラメータを定義し、サーバを再起動して TCP/IP トランスポートを有効にします。

Kerberos によるアクセス制御は、IPv6 を使用する TCP/IP 環境ではサポートされていません。Kerberos と IPv6 が有効になっていると、サーバに接続しようとしても、ネットワーク・アドレスが無効であることを示す KERBEROS エラーで失敗します。Kerberos によるアクセス制御を使用するには、INET トランスポート

ト名(表 4-1 を参照)を指定します。これにより、IPv4 アドレスの使用が強制されます。

以降の項では、デスクトップ・セッションの外部と内部のサーバ接続に対して Kerberos によるアクセス制御を適用する方法を説明します。

デスクトップ・セッション外の接続の場合

DECwindows Motif セッション外の接続に対して Kerberos によるアクセス制御を適用するには、以下の手順を実行します。

1. 次のようにして Kerberos Administration ユーティリティを起動します。

```
$ KERBEROS/INTERFACE=DECWINDOWS/ADMIN
```

2. 以下のプリンシパル、keytab ファイル、keytab ファイル・エントリを作成します。Kerberos Administration ユーティリティの使用方法については、Kerberos Client for OpenVMS のマニュアルを参照してください。

- プリンシパル `x0/host@REALM` を作成します。次に例を示します。

```
x0/system@ORG.COMPANY.COM
```

- keytab ファイル `SYSSYSROOT:[SYSMGR]DECW$X0.KEYTAB` を作成します。
- keytab ファイル内にプリンシパル `x0` 用のエントリを作成します。

3. `DECW$PRIVATE_SERVER_SETUP.COM` ファイルを変更し、X authority ファイル、アクセス許可ファイル、アクセス信頼ファイルが格納される場所を指すように以下のパラメータの値を定義します。以下に例を示します。

```
$ DECW$SERVER_XAUTHORITY == "SYS$MANAGER:SERVER_ZEPHYR.DECW$XAUTH"  
$ DECW$SERVER_ACCESS_ALLOWED == "SYS$MANAGER:DECW$SERVER_ZEPHYR_ACCESS_ALLOWED.DAT"  
$ DECW$SERVER_ACCESS_TRUSTED == "SYS$MANAGER:DECW$SERVER_ZEPHYR_ACCESS_TRUSTED.DAT"
```

4. `xauth` ユーティリティを使用して、サーバ用の X authority ファイルを手動で作成し、適切なエントリを追加します。たとえば、下記のコマンドは、新しい X authority ファイル `SERVER_ZEPHYR.DECW$XAUTH` を作成し、ローカル・トランスポート用のエントリを追加して、Kerberos プロトコルを指定し、keytab ファイルを示す値を割り当てます。

```
$ XAUTH -f SYSSYSROOT:[SYSMGR]SERVER_ZEPHYR.DECW$XAUTH -  
_ $ ADD :0 MIT-KERBEROS-5 -  
_ $ ""CS:X0,SYSSYSROOT:[SYSMGR]DECW$X0.KEYTAB""
```

5. `DECW$SERVER_ACCESS_TRUSTED` パラメータで指定した場所にアクセス信頼ファイルを手動で作成し、次のように SYSTEM アカウント用のエントリを追加します。

```
* SYSTEM 0
```

6. DECW\$SERVER_ACCESS_ALLOWED パラメータで指定した場所にアクセス許可ファイルを手動で作成し、サーバへのアクセスを許可するそれぞれの Kerberos プリンシパルについてエントリを追加します。アクセス許可ファイル内の Kerberos プリンシパルに対する各エントリは、*protocol-principal@realm-accessrights*の形式に従います。ここで、*accessrights*は NONE、ALL、または*です。

たとえば、アクセス許可ファイル内の次のエントリは、プリンシパル JONES に対して、TCP/IP トランスポート経由でのサーバ・アクセスを許可します。

```
KERBEROS jones@ORG.COMPANY.COM ALL
.
.
.
```

7. ファイルを保存してサーバを再起動します。

デスクトップ・セッション内の接続の場合

DECwindows Motif セッション内のサーバ接続に対して Kerberos によるアクセス制御を適用するには、以下の手順を実行します。

1. デスクトップに応じて以下のどちらかを実行します。
 - Traditional Desktop の場合、セッション・マネージャの[オプション]メニューから[セキュリティ...]を選択します。
 - New Desktop の場合、スタイル・マネージャの[J-Security]コントロールをクリックします。

「Security Options」ダイアログ・ボックスが表示されます。
2. [Configure Principals]ボタンをクリックします。
3. 追加する Kerberos プリンシパルに対する指定を、「Authorized Principals」リストに追加します。

一般的な Kerberos プリンシパルの形式は *primary/instance@REALM* です。
4. [Add]ボタンをクリックします。プリンシパルが「Authorized Principals」ボックスに追加されます。
5. [了解]をクリックして変更を保存および適用し、「Configure Principals」ダイアログ・ボックスを閉じます。
6. 「Server Access Control」領域で「Kerberos」を選択し、[了解]をクリックします。

「Kerberos Login」ダイアログ・ボックスが表示され、ログインして Kerberos 証明書を確認するように促されます。
7. Kerberos プリンシパル名とパスワードを入力して[了解]をクリックします。プリンシパル名とパスワードでは大文字と小文字が区別される点に注意してください。

Kerberos を無効にするには、Kerberos オプションの選択を解除し、一覧からすべてのプリンシパルを削除して、[了解]または[適用]をクリックします。

1 つ以上のプリンシパルがセッションにアクセスできないようにするには、まず削除する名前をクリックし、次に[Remove]ボタンをクリックします。最後に、[了解]または[適用]をクリックします。そのプリンシパルはワークステーションにアクセスする許可がなくなります。

現在のチケットが信頼できなくなったと思われる場合は、[Revoke Ticket]ボタンを押すことで、セッションへのアクセスを拒否し、プリンシパルに対してログインおよび認証手順を繰り返すように強制することができます。

3.3.10 セキュリティ拡張の利用

セキュリティ拡張 (SECURITY) を使用すると、xauth コーティリティまたは SET DISPLAY/GENERATE コマンドを使用して承認キーを手動で生成することができます。これにより、以下の追加属性のいずれかをサーバ接続に適用するように指定することができます。

- UNTRUSTED – この接続が信頼されない接続であることを示します。信頼されない接続では、接続上で実行できる操作が大幅に制限されます。信頼されない接続上で動作しているクライアント・アプリケーションでは、X サーバ拡張に対する限られたアクセスだけが許可され、他のアプリケーションが作成したウィンドウへのアクセスが禁止されます。これは、すべての承認キーでのデフォルトの属性です。
- TRUSTED – この接続が信頼される接続であることを示します。信頼される接続では、接続上ですべてのクライアント操作 (アクセス制御パラメータを変更する操作を除く) を実行することができます。
- TIMEOUT – トークンの有効期間を設定します。
- GROUP – トークンを適用するアプリケーション・グループを示します。

注意

信頼されない接続上での使用を許可するように作成されていないクライアント・アプリケーションは予想しない動作をすることがあります。信頼されない接続での制限事項の説明については、X.Org Foundation が発行しているセキュリティ拡張の仕様を参照してください。

3.3.10.1 セキュリティ拡張を有効にする

セキュリティ拡張を有効にするには、SYSS\$MANAGER:DECW\$PRIVATE_SERVER_SETUP.COM ファイルを変更して、値 "SEC_XAG" を含むように DECW\$SERVER_EXTENSIONS パラメータを再定義します。次に例を示します。

```
$ decw$server_extensions == "SEC_XAG"
```

ファイルを保存してサーバを再起動します。

3.3.10.2 セキュリティ・ポリシー・ファイルの利用

セキュリティ・ポリシー・ファイルを使用すると、信頼されないネットワーク接続上で特定の動作 (X アトム・レベル) を許可するようにサーバを構成することができます。このファイルは、一連のフィールド定義を通じて、許可する動作を規定する 1 つ以上のサイト・ポリシーを確立します。

DECwindows Motif では、DECW\$EXAMPLES:DECW\$SECURITY_POLICY.TXT という名前のサンプル・ファイルが提供されています。ポリシー・ファイルを作成する際にはこのファイルをテンプレートとして使用します。セキュリティ・ポリシーについては、X.Org Foundation が発行している『Security Extension Specification』に説明があります。セキュリティ・ポリシーの使用方法和定義についての詳細は、この仕様書を参照してください。

DECwindows Motif システム上でセキュリティ・ポリシー・ファイルを設定するには、以下の手順を実行します。

1. DECW\$EXAMPLES:DECW\$SECURITY_POLICY.TXT を別のファイルにコピーし、必要な変更を行って、システム上の別の場所に保存します。
2. DECW\$PRIVATE_SERVER_SETUP.COM ファイルを変更し、セキュリティ・ポリシー・ファイルがある場所を指すように DECW\$SECURITY_POLICY パラメータの値を変更します。
3. ファイルを保存してサーバを再起動します。

3.4 マルチヘッド・システムの設定

マルチヘッド・システムでは、複数のモニタをシステムに接続して、単一のディスプレイを構成することができます。ここでは、システム設定の前提条件と、マルチヘッド・システムを構成するためにサポートされている手法について説明します。

3.4.1 システム設定

マルチヘッド・システムを設定する前に、まず以下の作業を実行する必要があります。

1. VGA サービスを無効にする – ビデオ・カードによっては、必要に応じてVGA サービスを動的に無効または有効にできるものもありますが、そうでない場合は、ビデオ・カード上のジャンパ設定を使用して手作業でVGAを無効にする必要があります。このような変更が必要かどうかについては、ビデオ・カードのドキュメントを参照してください。変更が必要な場合は、システムにカードを装着する前にこの変更を行います。

警告

システムに複数のビデオ・カードを装着し、複数のカードでVGAサービスを有効にしたままにすると、ブート時にカード全体でビデオ・サブシステムの制御が競合するため、システムが損傷する可能性があります。

2. ビデオ・カードを装着する – OpenVMSシステムをシャットダウンし、ハードウェア・ドキュメントの説明に従ってビデオ・カードを装着します。

電源をオンにし、オペレーティング・システムをリブートします。スタートアップ中にOpenVMSオペレーティング・システムはビデオ・カードが正しく装着されていることを確認します。

3.4.2 単純なマルチヘッド・システムの構成

DECW\$PRIVATE_SERVER_SETUP.TEMPLATE ファイルには、システムにマルチヘッドの利用を設定するための以下のコマンドが含まれています。

```
$ IF DECW$DEVICE_COUNT .GT. 1 THEN DECW$MULTI_HEAD == 1
```

テンプレート・ファイルはSYSSMANAGERディレクトリにあります。マルチヘッドのサポートを起動するには、テンプレート・ファイルをSYSSMANAGER:DECW\$PRIVATE_SERVER_SETUP.COMにコピーします。このファイルを編集する必要はありません。

3.4.3 XINERAMA を使用したマルチヘッド・システムの構成

Xinerama 拡張 (XINERAMA) を使用すると、1 台のシステムに複数のモニタを接続して、統合された仮想ディスプレイを作成することができます。単純なマルチヘッド・システムと異なり、XINERAMA では、画面およびデスクトップの配置を制御することができます。ディスプレイ中の画面の数、順序、構成をカスタマイズしたり、デスクトップ上の画面間でウィンドウや文字をドラッグすることができます。

ディスプレイ・サーバの構成

3.4 マルチヘッド・システムの設定

ここでは、XINERAMA を使用したマルチヘッド・システムの構成方法について説明します。

3.4.3.1 ハードウェアおよび構成の要件

XINERAMA は、同質のグラフィックス環境でのみサポートされます。マルチヘッドの各画面は、共通のビデオ・カード、ビット深度、ビジュアル・クラス、画面解像度、同サイズのモニタで構成する必要があります。また、モニタは、隙間のない長方形として配置する必要があります。

現在サポートされているビデオ・グラフィックス・カードについては、『日本語 HP DECwindows Motif for OpenVMS ソフトウェア使用書』を参照してください。これらのグラフィックス設定のデフォルト値を変更するために使用できる論理名の説明については、第 3.1.2.4 項を参照してください。

ディスプレイ・サーバは、マルチヘッド構成で最大 16 台のモニタをサポートします。実際のモニタの数は、利用可能なオプション・カード・スロットの数によっても制限される点に注意してください。

3.4.3.2 XINERAMA 拡張を有効にする

この拡張は X ディスプレイ・サーバの一部ですが、デフォルトでは有効になっていません。

XINERAMA を有効にするには、DECW\$PRIVATE_SERVER_SETUP.COM ファイルを変更し、値 "XINERAMA" を含むようにパラメータ DECW\$SERVER_EXTENSIONS を再定義します。次に例を示します。

```
$ DECW$SERVER_EXTENSIONS == "DEC-XTRAP,XINERAMA"
```

ファイルを保存してサーバを再起動します。

3.4.3.3 モニタの配置

デフォルトでは、システム・バス上のビデオ・カードの物理位置を使用してデバイス名 (GYA0, GYB0 など) が割り当てられます。その後デバイスに画面番号が割り当てられ、デバイス名のアルファベット順に初期化されます。

例として、4 つのモニタからなるマルチヘッド構成では、正しい順序でビデオ・カードにケーブルを接続し、モニタを横に並べたとすると、画面番号は昇順 (0, 1, 2, 3) または降順 (3, 2, 1, 0) となります。

OpenVMS システムでは、エッジとポインタを正しく連結するために、画面の初期化順序が非常に重要です。画面 0 を最初に初期化し (仮想ディスプレイの左上)、次に残りの画面を、順次初期化し、最後に最も画面番号が大きな画面 (仮想ディスプレイの右下) を初期化する必要があります。画面の順序が正しくない場合は、画面構成に応じた以下のいずれかを実行します。

- 正しい位置に物理的にモニタを移動する。
- 正しい順序でケーブルを再接続する。

- DECW\$PRIVATE_SERVER_SETUP.COM ファイルを編集して、デフォルトの画面初期化順序を設定変更するように DECW\$SERVER_SCREEN\$S パラメータを定義する。

画面の順序が正しくなったら、DECW\$PRIVATE_SERVER_SETUP.COM 内で以下のエッジ連結パラメータを使用して、さらに仮想ディスプレイをカスタマイズします。

```
DECW$SERVER_EDGE_LEFT  
DECW$SERVER_EDGE_RIGHT  
DECW$SERVER_EDGE_TOP  
DECW$SERVER_EDGE_BOTTOM
```

これらのパラメータは、仮想ディスプレイの各エッジを連結する場所を制御します。

セットアップ処理が完了すると、すべてのモニタがアクティブになり、適切に配置された状態になります。DECwindows Motif を再起動すると、仮想ディスプレイの中央にセッションのログイン・ダイアログ・ボックスが表示され、アプリケーション・ウィンドウを開いたり、画面間でウィンドウをドラッグできるようになります。

3.5 デフォルトのキーボード・レイアウトの変更

DECwindows Motif で利用可能なキーボード・レイアウトには、DECwindows キーマップと X Keyboard キーマップの 2 種類があります。ここでは、デスクトップ・セッション外でデフォルトのキーボード設定を変更する方法について説明します。『日本語 DECwindows Motif for OpenVMS 操作ガイド』および『日本語 DECwindows Motif New Desktop 使用概説書』で説明されているように、デスクトップ・セッションの間は、セッション・マネージャによってこれらの設定が設定変更される可能性がある点に注意してください。

言語、キーボード・モデル、キーマップ名の一覧については、付録 B を参照してください。

3.5.1 DECwindows キーマップ・ファイルの利用

デフォルトのキーボード・レイアウトを設定変更して、DECwindows キーマップ・ファイルを指定するには、以下のようにして、使用しているキーボードのモデル番号から正しい名前を確認します。

1. キーボードを裏返し、モデル番号が記載されたラベルを探します。モデル番号は LK401-xx のような形式になっています。

モデル番号は、単に LK401 のようになっていることもあります。その場合は、必要な情報はラベルの別の部分にあり、nn-nnnnn-xx の形式の番号が記載されています。

- この番号の xx の部分を使用して、付録 B からキーマップ名を選択します。表は、キーボードの言語順に並んでいます。

以下の文字列で終わるキーマップを選択します。

_DP (データ処理キーボード・レイアウトの場合)

_TW (タイプライタ・レイアウトの場合)

_LK (LK シリーズのレイアウトの場合)

_PC (PC ベースのレイアウトの場合)

- キーマップ名を選択したら、DECW\$PRIVATE_SERVER_SETUP.COM ファイルを変更します。たとえば、キーボード・レイアウトをオランダ語のタイプライタ・レイアウトに変更するには、次の行を DECW\$PRIVATE_SERVER_SETUP.COM に追加します。

```
$ DECW$DEFAULT_KEYBOARD_MAP == "DUTCH_LK201LH_TW"
```

- 同様の行を、北米のキーボード・レイアウトでないワークステーションのそれぞれで追加するか、全ワークステーション共通のセクションにこの行を追加します (すべてのワークステーションが同じキーボード・レイアウトの場合)。

例 3-2 に、オランダ語キーボードと 100 dpi のモニタを使用した 2 台のワークステーション用の設定例を示します。

例 3-2 オランダ語キーボードと 100 dpi モニタの設定例

```
$do_DUTCH:
$do_DUTCH2:
$ decw$server_density == 100
$ decw$default_keyboard_map == "DUTCH_LK201LH_DP"
$ exit
```

3.5.2 X Keyboard キーマップ・ファイルの利用

X Keyboard キーマップ・ファイルは、DECwindows Motif でのキーマップに相当し、標準の X ウィンドウ・システムでの代替方法です。これは、DECwindows Motif のキーマップ・ファイルを置き換えるものではなく、補完するものとして設計されています。

以下の項で説明するように、X Keyboard Compiler コーティリティ (xkbcomp) を使用して、X Keyboard レイアウト・ファイルをコンパイルし、ロード可能なキーマップを作成することも、サーバが必要に応じてコンパイルすることもできます。

また、ベンダに依存しない、標準のロード可能なキーボードとして X Keyboard キーマップ形式 (.XKM) があるため、他の X11R6 ベース・システムおよび X ウィンドウ・システム・ソフトウェアのプロバイダが提供している XKM ファイルをロードすることもできます。

3.5.2.1 X Keyboard のコンポーネント・データベース

Xディスプレイ・サーバは、Xキーボード・コンポーネントおよび共通キーボード・マッピングのデータベースを保持しています。これらのコンポーネントを組み合わせることで、キーボードとその動作の完全な記述が得られます。

サーバは、DECW\$SERVER_XKEYBOARD_MAP パラメータで指定された、コンパイル済みのキーマップ・ファイルからデータベースをロードします。このファイルは、DECW\$SERVER_XKEYBOARD_COMPILED_DIR パラメータで定義されるディレクトリにあります。コンパイル済みのキーマップ・ファイルが存在しない場合、サーバは xkbcomp コーティリティを実行してコンポーネント・ソースをコンパイルしてコンパイル済みのキーマップ・ファイルを生成します。

データベースは以下のキーボード・コンポーネント・ソース・ファイルで構成され、これらのソース・ファイルを使用してロード可能なキーマップ・ファイルが生成されます。

- キーマップ・ソース・ファイル - これは、xkbcomp コーティリティのコマンド行で入力ファイルとして指定される、上位のソース・ファイルです。キーマップ・ソース・ファイルは、コンパイル時に他のコンポーネント・ソース・ファイルを参照し、完全なロード可能キーマップ (.XKM) ファイルが生成されます。

これらのファイルは、DECW\$SERVER_XKEYBOARD_DIRECTORY パラメータで指定されたルート・ディレクトリの KEYMAP.DIGITAL サブディレクトリに格納されます。サポートされる言語バリエーションごとに1つのキーマップ・ファイルがあります。次に例を示します。

```
DECW$SYSCOMMON:[SYSS$KEYMAP.XKB.KEYMAP.DIGITAL]US  
DECW$SYSCOMMON:[SYSS$KEYMAP.XKB.KEYMAP.DIGITAL]JAPANESE
```

- キーコード・コンポーネント・ソース・ファイル - このファイルは、入力デバイスから通知されるロー・キーコードの範囲と解釈を指定します。このファイルにはキーコードのシンボリック名、キーボードの最小と最大の有効なキーコード、各キーのシンボリック名が設定されます。

キーコード・ファイルには、キーの別名、インディケータのシンボリック名、どのインディケータが物理的に存在するかの記述も含まれています。

キーコード・コンポーネント・ソース・ファイルは、DECW\$SERVER_XKEYBOARD_DIRECTORY パラメータで指定されるルート・ディレクトリの KEYCODES.DIGITAL サブディレクトリに格納されます。次に例を示します。

```
DECW$SYSCOMMON:[SYSS$KEYMAP.XKB.KEYCODES.DIGITAL]LK  
DECW$SYSCOMMON:[SYSS$KEYMAP.XKB.KEYCODES.DIGITAL]PC
```

- タイプ・ソース・ファイル – このファイルは、さまざまなキーボード・キーに関連付けることができるレイアウト・タイプを指定します。このファイルは、タイプ・シンボリック名と、キーボードに関連付けられたレイアウト・タイプの一覧に影響を与えます。

タイプ・コンポーネントには、実際の修飾キーのバインディングと、1つ以上の仮想修飾キーのシンボリック名も格納されます。

これらのファイルは、DECW\$SERVER_XKEYBOARD_DIRECTORY パラメータによって指定されるルート・ディレクトリの TYPES サブディレクトリに格納されます。

```
DECW$SYSCOMMON:[SY$KEYMAP.XKB.TYPES]BASIC
DECW$SYSCOMMON:[SY$KEYMAP.XKB.TYPES]DEFAULT
```

- 互換性マップ・ソース・ファイル – このファイルは、クライアントとサーバの間の XKB 機能 (対応または非対応) に基づいて、キーボード・シンボル (keysyms) に対して動作を割り当てるために使用する規則を指定します。XKB 機能は、以下の互換性変換によって決まります。

XKB 拡張状態からコア状態への変換

コア・キーボード・マッピングから XKB キーボード・マッピングへの変換

XKB キーボード・マッピングからコア・キーボード・マッピングへの変換

互換性マップ・コンポーネントは、互換性シンボリック名、シンボル互換性マップ、およびグループ互換性マップに影響を与えます。このコンポーネントは、実際の修飾キー・バインディングおよびいくつかの仮想修飾キーのシンボリック名に加え、インディケータに対するマップを指定することもできます。

互換性マップのソース・ファイルは、DECW\$SERVER_XKEYBOARD_DIRECTORY パラメータで指定されるルート・ディレクトリの COMPAT サブディレクトリに格納されます。次に例を示します。

```
DECW$SYSCOMMON:[SY$KEYMAP.XKB.COMPAT]BASIC
DECW$SYSCOMMON:[SY$KEYMAP.XKB.COMPAT]DEFAULT
```

- シンボル・ソース・ファイル – このファイルは、キーボードの各キーにバインドするシンボルを指定します。このファイルは、各キーのキー・シンボル・マッピング、キーボード修飾キー・マッピング、およびキーボード・シンボル・グループのシンボリック名に影響を与えます。シンボル・コンポーネントには、いくつかのキーや実際の修飾キー・バインディングに対する明示的な動作、およびいくつかの仮想修飾キーのシンボリック名を格納することもできます。

シンボル・ソース・ファイルは、DECW\$SERVER_XKEYBOARD_DIRECTORY パラメータで指定されるルート・ディレクトリの SYMBOLS サブディレクトリと SYMBOLS.DIGITAL サブディレクトリに格納されます。次に例を示します。

```
DECW$SYSCOMMON:[SY$KEYMAP.XKB.SYMBOLS]US
```

```
DECW$SYSCOMMON:[SY$KEYMAP.XKB.SYMBOLS.DIGITAL]SWISS_
FRENCH
```

- ジオメトリ・ソース・ファイル – このファイルは、キーボードのジオメトリを定義します。このファイルは、ジオメトリのシンボリック名と、キーボード・ジオメトリ記述を定義します。ジオメトリ・コンポーネントには、キーの別名やインディケータのシンボリック名を格納することもでき、物理的に存在するインディケータのセットに影響を与えることもあります。キーボード・マッピングのジオメトリ・コンポーネント中で定義されたキーの別名は、キーコード・コンポーネントで定義されたキーの別名より優先されます。

これらのファイルは、DECW\$SERVER_XKEYBOARD_DIRECTORY パラメータで指定されたルート・ディレクトリの GEOMETRY サブディレクトリに格納されます。次に例を示します。

```
DECW$SYSCOMMON: [SY$KEYMAP.XKB.GEOMETRY.DIGITAL]LK
DECW$SYSCOMMON: [SY$KEYMAP.XKB.GEOMETRY.DIGITAL]PC
```

- その他のディレクトリ – ベース・ディレクトリ DECW\$SERVER_XKEYBOARD_DIRECTORY の SEMANTICS サブディレクトリには、キーボード・タイプのデフォルトのセマンティックと互換性コンポーネントを定義した一連のファイルが格納されています。

ベース・ディレクトリ DECW\$SERVER_XKEYBOARD_DIRECTORY と、サブ・ディレクトリ TMP および RULES 内のファイルはサーバで使用されるため、変更してはなりません。

3.5.2.2 X Keyboard のキーマップ・ファイルの作成

X Keyboard のキーマップ・ファイルを作成するには、以下の手順を実行します。

1. 第 3.5.2.1 項で説明した 1 つ以上のコンポーネント・ソース・ファイルを編集し、必要な変更を行います。たとえば、すべての US キーマップで左括弧と右括弧を入れ替えるには、SY\$COMMON:[SY\$KEYMAP.XKB.SYMBOLS.DIGITAL]US を次のように編集します。

```
.
.
.
19   key <AE09> {      [          9,   parenleft ] };
20   key <AE10> {      [          0,   parenright ] };
.
.
.
```

2. xkbcomp ユーティリティを使用して、コンポーネント・ソースファイルをコンパイルし、変更済みのキーマップ・ファイルを作成します。たとえば、DIGITAL_US_LK401 用の変更済みのキーマップ・ファイルを作成するには、次のようにしてソースをコンパイルします。

ディスプレイ・サーバの構成

3.5 デフォルトのキーボード・レイアウトの変更

```
$ xkbcomp -RDECW$SYSCOMMON:[SYS$KEYMAP.XKB] -xkm -m lk401 -  
_ $ DECW$SYSCOMMON:[SYS$KEYMAP.XKB.KEYMAP.DIGITAL]us -  
_ $ -o SYS$COMMON:[SYS$KEYMAP.XKB.COMPILED]digital_us_lk401.xkm
```

次に、第 3.5.2.3 項の説明に従って、変更済みのコンパイル済みキーマップ・ファイルをロードします。

3.5.2.3 コンパイル済みキーマップ・ファイルをロードする

コンパイル済みの X Keyboard キーマップ・ファイルをロードするには、以下の手順を実行します。

1. DECW\$PRIVATE_SERVER_SETUP.COM ファイルを変更し、X Keyboard (XKB) 拡張の使用を有効にするように DECW\$SERVER_EXTENSIONS パラメータの値を定義します。たとえば次のようにします。

```
$ DECW$SERVER_EXTENSIONS == "XKB,XINERAMA"
```

2. パラメータ DECW\$SERVER_XKEYBOARD_LOAD_MAP の値を、X Keyboard キーマップの使用を有効にするように定義します。

```
$ DECW$SERVER_XKEYBOARD_LOAD_MAP=="1"
```

3. DECW\$SERVER_XKEYBOARD_COMPILED_DIR パラメータの値を、キーマップ・ファイルがある場所を指すように定義します。このディレクトリには、サーバが必要に応じてコンパイルしたキーマップ・ファイルも格納されます。
4. DECW\$SERVER_XKEYBOARD_MAP パラメータの値を、サーバのスタートアップ時にロードするデフォルトの X Keyboard キーマップを指すように定義します。
5. ファイルを保存してサーバを再起動します。

3.5.2.4 AccessX キー機能を有効にする

AccessX キー機能を有効にするには、以下の手順を実行します。

1. DECW\$PRIVATE_SERVER_SETUP.COM ファイルを編集します。
2. DECW\$SERVER_EXTENSIONS パラメータの値を、X Keyboard (XKB) 拡張の使用を有効にするように定義します。たとえば次のようになります。

```
$ DECW$SERVER_EXTENSIONS == "XKB,XINERAMA"
```

3. DECW\$SERVER_ENABLE_ACCESSX パラメータに 1 (有効) を設定します。
4. ファイルを保存してサーバを再起動します。

次に、accessx ユーティリティを使用するか、スロー・キー機能やスティッキー・キー機能を使用して、AccessX の機能をさらに構成することができます。以下のようになります。

構成内容	実施方法
スロー・キーを切り替える	Shift キーを単独で 8 秒間押し続ける。
スティッキー・キーを切り替える	左または右の Shift キーを、間で他のキーを押さず、30 秒以上の間隔を開けずに 5 回連続で押す。
スティッキー・キーをオフにする	2 つ以上の修飾キーを同時に押す。

3.6 新しいフォントの指定

新しいフォントのサポートを追加するには、以下の手順を実行します。

1. DECWSFONT ファイルを DECWSSYSCOMMON:
[SYSFONT.DECW.USER_XX]ディレクトリに置きます。
2. 特権を持ったアカウントで次のコマンドを実行し、サーバのフォント・データベースを更新します。

```
$ @SYS$UPDATE:DECW$MKFONTDIR
```

このコマンドは、すべてのレイヤード・プロダクトのフォント用の新しいフォント・ディレクトリを作成します。DECwindows のインストール時に自動的に実行されます。

3.6.1 他社製のフォントの使用

他社製のフォントを使用する場合は、以下の方法のいずれかを選択します。

フォント・ファイルが BDF 形式の場合

1. BDF (Bitmap Distribution Format) 形式のフォント・ソース・ファイルを入手します。
2. 次のように、FONT コマンドでフォントをコンパイルします。

```
$ FONT file-name
```

FONT コマンドの詳細については、『VMS DECwindows Guide to Xlib (Release 4) Programming: MIT C Binding』を参照してください。

3. 得られた*.PCF ファイルを、以下のディレクトリのいずれかにコピーします。

```
SYS$COMMON:[SYSFONT.DECW.USER_75DPI]  
SYS$COMMON:[SYSFONT.DECW.USER_100DPI]  
SYS$COMMON:[SYSFONT.DECW.COMMON_USER]
```

4. 特権アカウントで、次のコマンドを実行して新しいフォント・ディレクトリを作成します。

```
$ @SYS$UPDATE:DECW$MKFONTDIR
```

ディスプレイ・サーバの構成

3.6 新しいフォントの指定

現在のセッションを終了させ、新しいセッションを開始して、新しいフォントを使用可能にします。

その他の他社製フォントの場合

その他の他社製フォントを利用するには、適切なフォント・ディレクトリに DECW\$FONT_ALIAS.DAT ファイルを作成します。

フォント別名ファイルの形式は次のとおりです。

alias-font-name actual-font-name

フォント名にスペースが含まれている場合は、名前を二重引用符(" ")で囲みます。

*alias-font-name*の値は他社製のフォント名で、*actual-font-name*の値は DECwindows のフォント名です。

サポートされている DECwindows フォント名については、『VMS DECwindows Guide to Xlib (Release 4) Programming: MIT C Binding』および『HP DECwindows Motif for OpenVMS New Features』を参照してください。

3.6.2 ユーロ通貨記号のサポートを有効にする

DECwindows Motif では、ユーロ通貨記号がサポートされています。ユーロ・フォント・セットは、基本 OpenVMS オペレーティング・システムのインストール時にインストールされます。インストール時にユーロ・オプションを選択することで、DECwindows Motif Version 1.3 以降が動作しているシステムでユーロ記号を有効にして表示できるようになります。

ユーロ・サポートを有効にするには、フォント定義をロードする必要があります。そのためには、一連のユーロセットアップ・コマンド・ファイルを作成します。以下に例を示します。

```
$ COPY SYS$COMMON:[SYSMGR]DECW$EURO_APPS_SETUP.TEMPLATE -
_ $ SYS$COMMON:[SYSMGR]DECW$EURO_APPS_SETUP.COM
$ COPY SYS$COMMON:[SYSMGR]DECW$EURO_SERVER_SETUP.TEMPLATE -
_ $ SYS$COMMON:[SYSMGR]DECW$EURO_SERVER_SETUP.COM
```

DECwindows Motif システムを再起動します。コマンドファイルは、DECwindows Motif のスタートアップ・プロシージャの一部として自動的に実行されます。

OpenVMS C ランタイム・ライブラリによるユーロ・ロケールのサポートは、基本的な DECwindows Motif のユーロ・サポートのためには必要ありません。ただし、ユーロ・ロケールで Motif アプリケーションを動作させる場合は、OpenVMS Alpha および OpenVMS I64 のメディア・キットに含まれているユーロ・ロケール・サポートをインストールする必要があります。

3.6.2.1 DECwindows Motif アプリケーションでのユーロ記号の表示

システムでユーロ・サポートを有効にすると、特に設定を行わなくても、ワークステーション上の任意の ISO8859-1 ビットマップ・フォントを使用してユーロ記号を表示することができます。標準の ISO Latin-1 フォントを使用して文字を表示する DECwindows Motif アプリケーションでは、文字 0xA4 でユーロ記号が表示されます。これらのフォントの XLFD (X Logical Font Description) 名の文字集合部分は、ISO8859-1 です。

DECterm ウィンドウにユーロ記号を表示するためには、「DECterm General Options」ダイアログ・ボックスで以下の 2 つの項目が選択されていることを確認してください。

UPSS ISO Latin-1
8-Bit Multinational Characters

3.6.2.2 キーボードを使ったユーロ記号の入力 (Alpha のみ)

ユーロ・サポートを有効にすると、システムに接続されているキーボードの種類に応じて、以下のキー・シーケンスのいずれかを使用してユーロ記号を入力することができます。

キーボード・タイプ	キーマップ	キー・シーケンス
LK- スタイル	*LK201*キーマップ	世界通貨記号と同じキー・シーケンスを使用。例: Compose+Space o x または Compose+Space O X Compose+Space x o または Compose+Space X O Compose+Space 0 x または Compose+Space 0 X Compose+Space x 0 または Compose+Space X 0
LK- スタイル	*LK401*キーマップ ¹	LeftCompose + E
PC- スタイル ²	*LK44*キーマップ	RightAlt + E

¹RUSSIAN_LK401_BT キーマップは、LeftCompose + E キーのシーケンスをサポートしていません。POLISH_LK401_BT キーマップは、LeftCompose + U キーのシーケンスを使用したユーロ入力をサポートしています。

²LK44*、PCXA*、または LK97W* など

3.6.3 フォント・サーバのサポートを有効にする

ファイル SYSSMANAGER:DECW\$PRIVATE_SERVER_SETUP.COM のサイト固有のサーバ・セクションでシンボル DECW\$FONT_SERVERS を定義することで、フォント・パスにフォント・サーバを追加することができます。トランスポートの種類に応じて、以下の方法のいずれかを使用します。

- TCP/IP トランスポートでのサポートを追加するには、フォント・サーバと通信するためのネットワーク接続タイプとして TCP を指定します。*node* をインターネット・ノード名に置き換えます。*port_number* に TCP/IP ポート番号を指定します。

```
$ DECW$FONT_SERVERS == "TCP/node::port_number"
```

- DECnet トランスポートでのサポートを追加するには、フォント・サーバと通信するためのネットワーク接続タイプとして DECNET を指定します。*node* を DECnet ノード名に置き換えます。*network_object* に DECnet オブジェクト名を指定します。

```
$ DECW$FONT_SERVERS == "DECNET/node::network_object"
```

コマンドで区切ったリストとしてシンボルを定義することで、複数のフォント・サーバを追加することができます。このシンボルでは大文字と小文字は区別されません。

3.7 LBX プロキシ・サーバの設定

図 1-3 に示したように、DECwindows は LBX プロキシ・サーバの使用をサポートしています。LBX プロキシ・サーバの使用は、以下の 3 種類の方法で構成できます。

- 管理対象 – プロキシ・サーバはプロキシ・マネージャで管理されます。このサーバは、複数のクライアントから、複数の X ディスプレイ・サーバにアクセスするために利用することができます。クライアントは、プロキシ・サーバのサーバ番号を知る必要はなく、単に要求された X ディスプレイ・サーバをプロキシ・マネージャに渡します。マネージャは、適切な既存のプロキシ・サーバを見つけるか、プロキシ・サーバの新しいインスタンスを自動的に起動します。
- 非管理対象 – プロキシ・サーバは手動で起動されます。プロキシ・マネージャはサーバを認識します。プロキシ・サーバは複数のクライアントから複数の X ディスプレイ・サーバにアクセスするために使用することができます。
- スタンドアロン – プロキシ・サーバは手動で起動されます。プロキシ・マネージャはプロキシ・サーバを認識しません。サーバは複数のクライアントから単一の X ディスプレイ・サーバにアクセスするために使用することができます。クライアントは、プロキシ・サーバの番号を知っている必要があります。

DECwindows Motif では、現在、管理対象構成およびスタンドアロン構成をサポートしています。

プロキシ・サーバとプロキシ・マネージャ・コンポーネントを使用するためには、DECwindows のインストール時に LBX オプションを選択しておく必要があります。LBX サポートのインストールについては、『日本語 HP DECwindows Motif for OpenVMS インストレーション・ガイド』を参照してください。

以降の項では、LBX 拡張を有効にする方法と、プロキシ・サーバの使用を構成する方法について説明します。

注意

クライアントとプロキシ・サーバ間の通信では、最適化されていない X プロトコルを使用するため、クライアントとプロキシ・サーバは常に同じノードまたは同じ LAN 上にある必要があります。

LBX によってシステム間のデータ・フローは減りますが、LAN のみの環境ではお勧めしません。全体的なトラフィック・フローは減りますが、その分プロセッサの負荷は増加します。通常、LAN のみの環境では、性能が少し低下します。

3.7.1 LBX 拡張を有効にする

LBX を使用するためには、X ディスプレイ・サーバが LBX プロトコルを解釈できる必要があります。DECwindows Motif システムで、DECW\$SERVER_EXTENSIONS パラメータを使用して、LBX プロトコルの使用を有効にしなければなりません。

LBX 拡張を有効にするには、SYSS\$MANAGER:DECW\$PRIVATE_SERVER_SETUP.COM ファイルを修正して、値 "LBX" を含むように DECW\$SERVER_EXTENSIONS パラメータを再定義します。次に例を示します。

```
$ DECW$SERVER_EXTENSIONS == "LBX"
```

ファイルを保存してサーバを再起動します。

3.7.2 プロキシ・サーバの起動

LBX プロキシ・サーバを起動する方法によって、プロキシ・サーバのタイプとクライアントがプロキシ・サーバにアクセスする方法が決まります。いずれの方法でも LBXPROXY コマンドを使用します。指定可能な修飾子とその値についての詳細は、コマンドのオンライン・ヘルプを参照してください。

LBX プロキシ・サーバを起動する前に、第 3.7.2.1 項の説明に従って、プロキシ・サーバから X ディスプレイ・サーバへの接続を適切に許可します。

3.7.2.1 プロキシ・サーバ環境での認証

プロキシ・サーバが X サーバに接続する際に、プロキシ・サーバはクライアントと同じようにして認証されます。プロキシ・サーバが認証情報を入手する方法は、プロキシ・サーバのタイプによって異なります。

ディスプレイ・サーバの構成

3.7 LBX プロキシ・サーバの設定

管理対象のプロキシ・サーバは、認証情報をプロキシ・マネージャから入手します。プロキシ・マネージャは、クライアントから認証情報を入手します。クライアントのデフォルトの認証情報は、クライアントの X authority ファイルに格納されています。クライアントでは、SET DISPLAY コマンドの/XAUTHORITY 修飾子を使用して、使用する X authority ファイルを制御することができます。クライアントは、SET DISPLAY コマンドの/LBXAUTHENTICATE 修飾子および/LBXDATA 修飾子を使用して、明示的に認証情報を指定することもできます。さらに、/NOLBXAUTHENTICATE 修飾子を使用して、プロキシ・サーバの現在の X authority ファイルの認証情報を使用するように指定することもできます。SET DISPLAY コマンドの LBX 用の修飾子についての詳細は、『HP OpenVMS DCL デイクショナリ: N-Z』を参照してください。

スタンドアロンのプロキシ・サーバは、現在の X authority ファイル中の情報から認証情報を入手します。

3.7.2.2 管理対象プロキシ・サーバの起動

管理対象 LBX プロキシ・サーバを起動するには、プロキシ・マネージャの構成ファイルに、次の LBX サービス・エントリを追加します (第 3.7.6 項を参照)。

```
LBX MANAGED COMMAND SYS$MANAGER:DECW$LBXPROXY_SUB ["qualifiers"]
```

プロキシ・マネージャの構成を終えたら、プロキシ・サーバを起動するために特別な操作を行う必要はありません。プロキシ・マネージャは、最初のクライアント要求を受け取ると、プロキシ・サーバを起動します。

3.7.2.3 スタンドアロン・プロキシ・サーバの起動

スタンドアロンの LBX プロキシ・サーバは、現在のプロセス内で起動するか、独立プロセスとして起動することができます。スタンドアロン・プロキシを現在のプロセス内で起動するには、LBXPROXY コマンドを使用します。

```
LBXPROXY [qualifiers]
```

たとえば、プロキシ・サーバを現在のプロセス内で起動し、サーバ番号 50 を割り当て、remote1.cmp.com 上の X ディスプレイ・サーバ用のプロキシとして動作させるには、次のコマンドを使用します。

```
$ LBXPROXY /DISPLAY="REMOTE1.CMP.COM:0"/SERVER=50/FIXED_SERVER
```

プロキシ・サーバを独立プロセスとして起動するには、コマンド・プロシージャ DECW\$LBXPROXY を使用します。

```
@SYS$MANAGER:DECW$LBXPROXY ["lbxproxy-qualifiers"] ["run-qualifiers"]
```

たとえば、プロキシ・サーバを独立プロセスとして起動し、サーバ番号 50 を割り当て、remote1.cmp.com 上の X ディスプレイ・サーバ用のプロキシとして動作させるには、次のコマンドを使用します。

```
$ @SYS$MANAGER:DECW$LBXPROXY "/DISPLAY=""REMOTE1.CMP.COM:0"" + -  
_$/SERVER=50/FIXED_SERVER"
```

LBXPROXY イメージを起動するために使用する RUN コマンドに修飾子を渡すには、*run-qualifiers*パラメータを使用します。このパラメータを使用する1つの例としては、デフォルトの LBXPROXY プロセス属性や論理名による値より優先させて、これらのデフォルトを変更することがあげられます。

指定可能な修飾子とその値についての詳細は、LBXPROXY コマンドのオンライン・ヘルプを参照してください。

注意

LBX プロキシ・サーバを独立プロセスとして起動するためには、DETACH 特権を持っているか、利用できる最大独立プロセスの制限に収まる必要があります。独立プロセスのプロセス制限を変更するには、DETACH 特権が必要です。

3.7.3 IPv6 環境でのプロキシ・サーバの利用

プロキシ・サーバは、起動すると、/TRANSPORT 修飾子を使用して構成されたすべてのトランスポートに対してリッスンを行います。TCPIP トランスポートが選択されている場合、プロキシ・サーバは、インタフェースが IPv4 互換であるか IPv6 互換であるかにかかわらず、構成されているすべてのインタフェースに対してリッスンを行います。

/DISPLAY 修飾子を使用すると、第4章で説明する追加のトランスポート名が使用できます。プロキシ・サーバ・プロセスによるトランスポート名 TCP および TCPIP の解釈は、DECW\$PRIVATE_APPS_SETUP.COM ファイルで定義されている DECW\$IPV6_SUPPORT 論理名で決まります。

3.7.4 デフォルトのプロキシ・サーバ・プロセス属性の変更

表 3-4 に、SYSSMANAGER:DECW\$LBXPROXY によって生成された RUN コマンド上で指定された、デフォルトの LBXPROXY プロセス属性を設定変更するための論理名の一覧を示します。

表 3-4 LBXPROXY プロセス属性論理名

論理名	RUN コマンドの修飾子
DECW\$LBX_AST_LIMIT	/AST_LIMIT
DECW\$LBX_BUFFER_LIMIT	/BUFFER_LIMIT
DECW\$LBX_DUMP	/DUMP
DECW\$LBX_ENQUEUE_LIMIT	/ENQUEUE_LIMIT

(次ページに続く)

表 3-4 (続き) LBXPROXY プロセス属性論理名

論理名	RUN コマンドの修飾子
DECWSLBX_EXTENT	/EXTENT
DECWSLBX_FILE_LIMIT	/FILE_LIMIT
DECWSLBX_IO_BUFFERED	/IO_BUFFERED
DECWSLBX_IO_DIRECT	/IO_DIRECT
DECWSLBX_LOG	/ERROR
DECWSLBX_MAXIMUM_WORKING_SET	/MAXIMUM_WORKING_SET
DECWSLBX_PAGE_FILE	/PAGE_FILE
DECWSLBX_PRIORITY	/PRIORITY
DECWSLBX_PROCESS_NAME	/PROCESS_NAME
DECWSLBX_QUEUE_LIMIT	/QUEUE_LIMIT
DECWSLBX_WORKING_SET	/WORKING_SET

3.7.5 プロキシ・サーバの停止

LBX プロキシ・サーバは自動または手動で停止できます。

3.7.5.1 自動停止

LBX プロキシ・サーバを自動的に停止するには、サーバを起動する際に /ONEXIT=TERMINATE 修飾子を使用します。スタンドアロン・プロキシ・サーバの場合、LBXPROXY コマンド行、またはコマンド・プロシージャ SYSS\$MANAGER:DECWSLBXPROXY の *lbxproxy-qualifiers* パラメータでこの修飾子を指定します。

管理対象サーバの場合は、プロキシ・マネージャの構成ファイルの LBX サービス定義中の *parameters* 引数でこの修飾子を指定します。

注意

プロキシ・マネージャは、停止する際に、まだ停止していない管理対象プロキシ・サーバがあると、すべて自動的に停止します。

3.7.5.2 手動停止

LBX プロキシ・サーバを手動で停止するには、DCL の STOP コマンドを使用します。

3.7.6 プロキシ・マネージャ構成ファイル

プロキシ・マネージャ構成ファイルには、プロキシ・マネージャがプロキシ・サービスを見つけるために必要となる情報が格納されます。構成ファイル内のそれぞれの行には、以下のいずれかを記述します。

- コメント – コメント行は、先頭を感嘆符 (!) で始めます。その行の他の文字はすべて無視されます。
- 管理対象サービス・エントリ – 管理対象サービス・エントリの形式は次のとおりです。

service-name MANAGED COMMAND *command-file* [*parameters*]

フィールドの意味は以下のとおりです。

<i>service-name</i>	管理対象サービスの名前を指定します。サービス名では大文字と小文字は区別されません。サービス名が同じエントリがファイル内に複数含まれている場合は、最初のエントリだけが有効です。LBX サービスでは、サービス名は LBX とする必要があります。サービス名では、スペース、タブ、改行文字を除き、X ポータブル文字集合の文字を使用する必要があります。
<i>command-file</i>	このサービスのプロキシ・サーバの新しいインスタンスを作成するために、プロキシ・マネージャが呼び出すコマンド・プロシージャの名前を指定します。LBX サービスでは、この引数は、通常 SYSSMANAGER:DECW\$LBXPROXY_SUB.COM になります。
<i>parameters</i>	<i>command-file</i> 引数で指定されたコマンド・プロシージャに渡すパラメータを指定します。 <i>command-file</i> 引数の後のスペースに続くすべての文字は、パラメータとしてコマンド・プロシージャに渡されます。LBX サービスでは、コマンド・プロシージャは、引用符で囲まれた 1 つのパラメータとして 1 つ以上のコマンド修飾子が渡されることを期待します。

現在、DECwindows Motif で提供されている唯一の管理対象サービスは LBX です。

- 非管理対象サービス・エントリ – 非管理対象サービス・エントリの形式は次のとおりです。

service-name UNMANAGED *address*

フィールドの意味は以下のとおりです。

<i>service-name</i>	非管理対象サービスの名前を指定します。サービス名では大文字と小文字は区別されません。サービス名が同じエントリがファイル内に複数含まれていると、マネージャは、アクティブで利用可能なプロキシ・サーバが見つかるまで各エントリを順に試します。サービス名では、スペース、タブ、改行文字を除き、X ポータブル文字集合の文字を使用する必要があります。
<i>address</i>	プロキシ・サーバのアドレスを、ICE (Inter-Client Exchange) 形式で指定します。

現在、DECwindows Motif でサポートされている非管理対象サービスはありません。

3.7.7 プロキシ・マネージャの起動

プロキシ・マネージャは、DECwindows Motif のスタートアップ時に自動的に起動するように構成することも、後から手動で起動するように構成することもできます。

3.7.7.1 DECwindows Motif のスタートアップ時に自動的に起動する

DECwindows Motif のスタートアップ時にプロキシ・マネージャを自動的に起動するには、SYSSMANAGER:DECWSPRIVATE_APPS_SETUP.COM ファイルを編集します。セッション・マネージャ環境のカスタマイズについての詳細は、第 4 章を参照してください。

表 3-5 では、このファイル内のプロキシ・マネージャを制御するシンボルについて説明します。

表 3-5 プロキシ・マネージャを制御するグローバル・シンボル

シンボル	説明
DECWSPROXY_MANAGER_CONFIG	プロキシ・マネージャの構成ファイルの名前を指定します。このシンボルを定義することにより、DECwindows のスタートアップ処理によってプロキシ・マネージャが起動されます。このシンボルで指定されたファイル名は、DECWSPROXY_MANAGER_OPTIONS シンボルで指定された構成ファイルより優先されます。デフォルトの構成ファイル SYSSMANAGER:DECW\$LBXPROXY.DECWSPMCFG がインストール時に提供されます。このファイルには、LBX サービス用のサービス・エントリが 1 つあります。
DECWSPROXY_MANAGER_LOG	プロキシ・マネージャがイベントを記録するために使用するログ・ファイルの名前を指定します。プロキシ・マネージャが起動してこのシンボルが未定義の場合、デフォルトでログ・ファイルは SYSSMANAGER:DECWSPROXYMANAGER.LOG になります。
DECWSPROXY_MANAGER_OPTIONS	プロキシ・マネージャの起動時に、XPROXYMANAGER コマンド行に追加する修飾子を指定します。DECWSPROXY_MANAGER_CONFIG シンボルで指定された構成ファイルは、このシンボルで指定された値よりも常に優先されます。
DECWSPROXY_MANAGER_QUOTAS	プロキシ・マネージャを起動するために使用する RUN コマンド行に追加する修飾子を指定します。

注意

プロキシ・マネージャ・プロセスが動作している最中に DECwindows Motif を再起動しても、プロキシ・マネージャは自動的に再起動されません。DECwindows のスタートアップ時にプロキシ・マネージャを必ず (関連するオプションとともに) 再起動させるには、DECwindows を再起動する前にプロキシ・マネージャ・プロセスを停止します。

DECwindows のスタートアップの一部としてプロキシ・マネージャを再起動する場合は、プロキシ・マネージャ・プロセスの所有者は DECwindows Motif のスタートアップ・コマンドを実行したユーザになります。DECwindows がシステムのスタートアップの一部として起動された場合は、所有者は SYSTEM アカウントです。DECwindows が別のアカウントから起動される場合は、そのアカウントの所有者に対して X ディスプレイ・サーバへのアクセスが許可されていることを確認してください。

3.7.7.2 手動起動

プロキシ・マネージャを手動で起動するには、XPROXYMANAGER コマンドを使用します。

XPROXYMANAGER [*qualifiers*]

たとえば、構成ファイル SYSSMANAGER:DECW\$LBXPROXY.DECW\$PMCFG とログ・ファイル SYSSMANAGER:DECW\$PM.LOG を使用してプロキシ・マネージャを起動するには、次のコマンドを使用します。

```
$ XPROXYMANAGER/CONFIGURATION=SYSSMANAGER:DECW$LBXPROXY.DECW$PMCFG -  
_ $ /LOG=SYSSMANAGER:DECW$PM.LOG
```

指定可能な修飾子とその値についての詳細は、XPROXYMANAGER コマンドのオンライン・ヘルプを参照してください。

DECwindows の使用方法

この章では、DECwindows が起動した後で使用する以下の内容について説明します。

- ディスプレイの設定
- New Desktop および Traditional DECwindows Desktop での DECwindows Motif のログイン・プロセスの説明
- ログイン環境のカスタマイズ
- スタートアップ環境のカスタマイズ
- セッション・マネージャのデフォルトの動作変更
- システム・リソース・ファイルの変更
- クライアント・アクセス制御の指定
- プリント・フォーマットのカスタマイズ

注意

この章の内容は、各デスクトップの基本的または高度なユーザ情報を総合的に説明することを目的としたものではありません。以下のマニュアルの補足となることを意図しています。

- 『HP OpenVMS DCL デictionary: N-Z』 (SET DISPLAY コマンドと SHOW DISPLAY コマンド)
 - 『日本語 DECwindows Motif New Desktop 使用概説書』
 - 『日本語 DECwindows Motif for OpenVMS 操作ガイド』
-

4.1 ディスプレイの設定

クライアント・アプリケーションが起動すると、ディスプレイ・サーバに対する接続を1つ以上オープンします。ディスプレイ・サーバは、ローカル・システムにあることもあれば、リモート・システムにあることもあります。アプリケーションは、基盤となる Xm, Xt, または Xlib ルーチンに対して、ディスプレイ・サーバのノード名とその他の接続情報を通知する必要があります。この情報は、ディスプレイ名と呼ばれる文字列に埋め込まれています。

X ウィンドウ・システムの用語では、ディスプレイ名は、「ハードウェア・ディスプレイ名を指定します。これは、使用するディスプレイと通信ドメインを決定します。[...]ディスプレイ名が NULL の場合、デフォルトでは環境変数 DISPLAY の値が使用されます。」と説明されています。

OpenVMS システムでは、ディスプレイ名は、必要なサーバ情報が格納された WSA デバイス、または必要なサーバ情報を明示的に指定するディスプレイ名文字列のどちらかの形式となります。どちらの形式のディスプレイ名も指定されないと、Xlib は論理名 DECW\$DISPLAY を展開してディスプレイ名を取得します。DECW\$DISPLAY 論理名が定義されていない場合には、Xlib は論理名 DISPLAY を展開します。論理名 DISPLAY も定義されていないときには、Xlib は論理名 SYSS\$OUTPUT を展開します。

事前に WSA デバイスを作成し、SET DISPLAY コマンドを使用してディスプレイ・サーバ情報を指定しておく必要があります。SET DISPLAY コマンドを実行していない場合には、論理名 DECW\$DISPLAY および DISPLAY は親プロセスから継承されます。なお、SET DISPLAY コマンドでは、/CREATE 修飾子を使用し論理名を指定していない場合には、論理名 DECW\$DISPLAY と DISPLAY が設定されます。

4.1.1 ディスプレイ名の形式

ディスプレイ名文字列の形式は次のとおりです。

```
[transport]/node[:]server[.screen]
```

transport

ディスプレイ・サーバに接続するために使用するトランスポートを指定します。指定できるトランスポートを以下に示します。表 4-1 に、DECwindows Motif がサポートしているすべてのトランスポート名とその意味を示します。一部のトランスポート名の実装は、X の仕様で定義されているものと少し異なる場合がありますので注意してください。

表 4-1 サポートされている DECwindows Motif のトランスポート名

トランスポート名	意味
DECNET	DECnet ネットワーク・プロトコルを使用します。
DNET	DECnet ネットワーク・プロトコルの X ウィンドウ・システムの名前です。DNET は、ほとんどの場合 DECNET と同等です。
INET	IPv4 プロトコル上の TCP を使用します。このオプションは、TCPIP に対する X の同義語です。IPv6 サポートの追加により、TCPIP が意味する 2 つのトランスポートのどちらかになります。
INET6	IPv6 プロトコル上の TCP が利用できればそれを使用します。IPv6 プロトコルが利用できない場合は、IPv4 プロトコルが使用されます。これは TCPIP が意味する 2 つのトランスポートのどちらかになります。
LAT	LAT プロトコルを使用します。このトランスポートに対する X の同義語はありません。
LOCAL	同じホスト内のサーバ・プロセスとクライアント・プロセスの間で共用メモリを使用します。
TCP	TCPIP プロトコル・オプションの X ウィンドウ・システム名です。TCP は TCPIP の同義語です。
TCPIP	あるバージョンの IP 上の TCP プロトコルを使用します。以前は、このオプションは INET を指す X の同義語でした。IPv6 のサポートが追加されたため、このオプションは、論理名 DECWSIPV6_SUPPORT の設定に応じて、INET または INET6 のどちらかを指すようになりました。この論理名の設定についての詳細は第 4.4.3 項を参照してください。

このパラメータはオプションです。デフォルトでは、コロン 1 つで終わるノード名では TCPIP トランスポートが使用されます。トランスポート名の代わりに 2 つ目のコロンを指定すると、DECNET トランスポートが使用されます。DECnet ノード名がコロンで終わる場合は、ノード名を二重引用符 ("") で囲みます。コロン 3 つ (:::) で終わるアドレスは、INET6 または DECnet のどちらかとして解釈されます。

node パラメータには、指定したトランスポートと互換性があるノード値を指定する必要があります。ディスプレイ名文字列を解析する際には、DECwindows Motif は、*node* パラメータと *transport* パラメータに互換性があるかどうかを検証しません。パラメータに互換性がないと、実際に接続しようとしたときに失敗します。ディスプレイ名パラメータのトランスポートとノードの両方を省略した場合は、LOCAL トランスポートを指定したものと見なされます。

node

ディスプレイ・サーバの DECnet ノード、TCP/IP ホスト、LAT アドレス、またはローカル・システム名を指定します。以下のいずれかを指定します。

DECnet ノード DECNET トランスポートまたは DNET トランスポートでは、DECnet フェーズ IV ノード名、DECnet フェーズ IV アドレス、DECnet フェーズ V ノード名または別名、DECnet フェーズ V アドレスのいずれかを指定します。ローカル・ノードを指定するには 0 を使用します。DECnet ノード名は引用符 (" ") で囲むことができます。

DECwindows の使用方法

4.1 ディスプレイの設定

TCP/IP ホスト	TCPIP, TCP, INET, INET6 のいずれかのトランスポートでは, TCP/IP ホスト名またはアドレスを指定します。詳細は, 表 4-2 および表 4-3 を参照してください。ローカルノードを指定するには 0 を使用します。
LAT サービス名	LAT トランスポートの場合は, ローカル・システムの名前を指定します。トランスポートは, この値をサービス名としてのみ使用します。
ローカル・システム名	LOCAL トランスポートの場合は, ローカル・システムの名前または 0 を指定します。

すべてのノード名またはアドレスの後には, 実際の *node* パラメータ内のコロンに加え, コロン 1 つを指定します。さらにもう 1 つコロンを追加して, DECNET トランスポートを指定することもできます。詳細は, *transport* パラメータの説明を参照してください。ディスプレイ名パラメータのトランスポートとノードの両方を省略した場合は, LOCAL トランスポートを指定したものと見なされます。

INET6 トランスポートの TCP/IP ホスト名またはアドレスを指定する場合 (または, DECWSIPV6_SUPPORT が TCP_IS_INET6 として定義されているときに TCPIP または TCP を使用する場合) には, 表 4-2 に示す IPv6 形式のいずれかを指定します。

表 4-2 サポートされている IPv6 ホスト名とアドレスの形式

形式	例
IPv6 ホスト名または完全修飾ドメイン名	ashfld.franklin.mass.us
IPv6 の通常のアドレス	FEDC:BA98:7694:3210:FEDC:BA98:7654:3210
IPv6 圧縮アドレス	::7654:3210
IPv4 互換 IPv6 アドレス	::978.765.432.1
IPv4 射影 IPv6 アドレス	::FFFF:978.765.432.1
RFC 2732 スタイル IPv6 リテラル・アドレス	[FEDC:BA98:7694:3210:FEDC:BA98:7654:3210]

INET, INET6, TCPIP, または TCP トランスポートの TCP/IP ホスト名またはアドレスを指定する場合には, 表 4-3 に示す IPv4 形式のいずれかを指定します。

表 4-3 サポートされる IPv4 ホスト名とアドレスの形式

形式	例
IPv4 ホスト名または完全修飾ドメイン名	ashfld.franklin.mass.us
IPv4 の通常のアドレス	978.765.432.10

server

X サーバ・ディスプレイ・システム上のディスプレイ・サーバのディスプレイ番号を 10 進数で指定します。ディスプレイ番号は通常は 0 から始まります。単一ディスプレイ・サーバ・システムでは, 通常は 0 番を使用します。

screen

指定された X ディスプレイ・サーバで使用する画面番号を 10 進数で指定します。画面番号は 0 から始まります。このパラメータはオプションです。デフォルトの画面番号は 0 です。

4.1.2 TCP/IP ホスト名の変換

ホスト名の変換方法は、IPv6 と IPv4 のどちらが選択されているかによって変わります。INET6 または TCPIP トランスポートが指定されていて、論理名 DECW\$IPV6_SUPPORT が "TCP_IS_INET6" と定義されている場合は、TCP/IP Services の getaddrinfo 関数を使用して、ローカル・データベースまたはドメイン・ネーム・サーバからアドレス情報が取得されます。複数のアドレスが利用可能な場合は、IPv4 アドレスの前にすべての IPv6 アドレスが試されます。各 IP バージョン内でアドレスを試す順序は不定です。

INET トランスポートが指定されていて、論理名 DECW\$IPV6_SUPPORT に "DISABLED" が定義されていない場合は、TCP/IP Services の getaddrinfo 関数を使用して、ローカル・データベースまたはドメイン・ネーム・サーバから IPv4 アドレス情報が取得されます。複数の IPv4 アドレスが利用可能な場合、アドレスが試される順序は不定です。

INET または TCPIP トランスポートが指定されていて、論理名 DECW\$IPV6_SUPPORT が "DISABLED" と定義されている場合は、TCP/IP Services の gethostbyaddr 関数を使用してローカル・データベースまたはドメイン・ネーム・サーバからアドレス情報が取得されます。この方法は、以前のバージョンの DECwindows Motif との互換性のためだけに提供されています。

4.2 ログイン・プロセスの説明

ここでは、DECwindows のログイン・プロセスについて説明します。ログイン・プロセスは、「Start Session」ダイアログ・ボックスが表示されてから、セッション・マネージャが起動するまでのプロセスです。

4.2.1 New Desktop のログイン・シーケンス

「Start Session」ダイアログ・ボックスに正しいユーザ名とパスワードを入力すると、以下の処理が実行されます。

1. LOGINOUT.EXE: DECW\$LOGINOUT.EXE 内のルーチンが呼び出されます。このプログラムは、以下の機能を実行します。
 - a. DW-MOTIF のライセンスを確認して、システムが、ディスプレイ・サーバとの接続をオープンするライセンスを持っていることを確認します。ライセンスがない場合、DECW\$LOGINOUT.EXE は警告メッセージを表示し、自身を再

- 起動して終了します。新しい「Start Session」ダイアログ・ボックスが表示されます。
- b. 入力ポインタが時計のカーソルに変わります。
 - c. 許可ユーザ・リストと信頼ユーザ・リストを変更して、ディスプレイ・サーバに接続するアプリケーションを実行できるようにし、セキュリティの設定を変更します。また、以前設定されたサーバのセキュリティ設定を削除します。
 - d. サーバ・プロセスの所有者を、ログイン中のユーザに変更するようにサーバに指示します。
 - e. サーバでランダムな magic cookie 値を設定し、X authority ファイル DECW\$ENDSESSION.DECW\$XAUTH に保存します。この cookie は、セッションを終了するためにプロセスがサーバに接続できることを保証します。
 - f. DTGREET プロセスに終了するように指示します。
 - g. DTHELLO.EXE を実行するプロセスを起動します。DTHELLO.EXE は、ウェルカム画面を表示し、スタートアップ・プロセスの間、サーバとの接続を保持します。
 - h. プロセス名として DTSESSION を設定します。
 - i. 「Start Session」ダイアログの表示に費やした時間がユーザ・プロセスに課金されないように、Start Session プロセスの累積 CPU 時間をゼロに初期化します。
 - j. DCL が実行するコマンド・プロシージャとして、CDE\$SYSTEM_DEFAULTS:[BIN]XSESSION.COM を渡します。
 - k. 終了して、プロセスの制御を DCL に渡します。
2. XSESSION.COM: セッション・マネージャのデフォルトの DCL コマンド・プロシージャとして動作します。このコマンド・プロシージャは、以下の機能を実行します。
- a. 論理名 DECW\$SM_WSQUOTA が定義されている場合、セッション・マネージャ・プロセスのワーキング・セット制限として論理名の値を設定します。
 - b. 論理名 SYSSYLOGIN が示すシステムのログイン・コマンド・プロシージャを実行します。デフォルトでは SYLOGIN.COM です。
 - c. 「Start Session」ダイアログ・ボックスで、ユーザ名の後に/NOCOMMAND を指定しなかった場合、ユーザごとの LOGIN.COM (または UAF レコードで指定されているその他のログイン・コマンド・プロシージャ) が実行されます。リソース・ファイルで使用するディレクトリを変更するには、LOGIN.COM 内で論理名 DECW\$USER_DEFAULTS を再定義します。
 - d. Kerberos 認証が必要な場合、Kerberos のログイン・ダイアログ・ボックスが表示されます。このダイアログはプリンシパルとパスワードを要求し、チケットを生成します。

- e. 必要に応じて、ユーザのデフォルトの X authority ファイルへのエントリの挿入や削除を行います。
- f. SYSSMANAGER:DECW\$SYLOGIN.COM があれば実行します。
- g. SYSSLOGIN:DECW\$LOGIN.COM があれば実行します。
- h. CDE\$SYSTEM_DEFAULTS:[BIN]DTSESSION.EXE を実行してセッション・マネージャを起動します。

4.2.2 Traditional DECwindows Desktop のログイン・シーケンス

「Start Session」ダイアログ・ボックスに正しいユーザ名とパスワードを入力すると、以下の処理が実行されます。

1. LOGINOUT.EXE: DECW\$LOGINOUT.EXE 内のルーチンが呼び出されます。このプログラムは、以下の機能を実行します。
 - a. DW-MOTIF のライセンスを確認して、システムが、ディスプレイ・サーバとの接続をオープンするライセンスを持っていることを確認します。ライセンスがない場合、DECW\$LOGINOUT.EXE は警告メッセージを表示して、自身を再起動して終了します。新しい「Start Session」ダイアログ・ボックスが表示されます。
 - b. 入力ポインタが時計のカーソルに変わります。
 - c. 許可ユーザ・リストと信頼ユーザ・リストを変更して、ディスプレイ・サーバに接続するアプリケーションを実行できるようにし、セキュリティの設定を変更します。また、以前設定されたサーバのセキュリティ設定を削除します。
 - d. サーバ・プロセスの所有者を、ログイン中のユーザに変更するようにサーバに指示します。
 - e. サーバでランダムな magic cookie 値を設定し、X authority ファイル DECW\$ENDSESSION.DECW\$XAUTH に保存します。この cookie は、セッションを終了するためにプロセスがサーバに接続できることを保証します。
 - f. スタートアップ・プロセスの間サーバとの接続を保持する DECW\$WAITFORSM.EXE を実行するプロセスを起動します。
 - g. プロセス名として DECW\$SESSION を設定します。
 - h. 「Start Session」ダイアログの表示に費やした時間がユーザ・プロセスに課金されないように、Start Session プロセスの累積 CPU 時間をゼロに初期化します。
 - i. DCL が実行するコマンド・プロシージャとして、SYSSMANAGER:DECW\$STARTSM.COM を渡します。
 - j. 終了して、プロセスの制御を DCL に渡します。

2. SYS\$MANAGER:DECW\$STARTSM.COM: セッション・マネージャのデフォルトの DCL コマンド・プロシージャとして動作します。このコマンド・プロシージャは、以下の機能を実行します。
 - a. 論理名 DECW\$SM_WSQUOTA が定義されている場合、セッション・マネージャ・プロセスのワーキング・セット制限として論理名の値を設定します。
 - b. 論理名 SYSSYLOGIN が示すシステムのログイン・コマンド・プロシージャを実行します。デフォルトでは SYLOGIN.COM です。
 - c. 「Start Session」ダイアログ・ボックスで、ユーザ名の後に/NOCOMMAND を指定しなかった場合、ユーザごとの LOGIN.COM (または UAF レコードで指定されているその他のログイン・コマンド・プロシージャ) が実行されます。リソース・ファイルで使用するディレクトリを変更するには、LOGIN.COM 内で論理名 DECW\$USER_DEFAULTS を再定義します。
 - d. SYSSYSTEM:DECW\$WSINIT.EXE を実行して、ユーザのワークステーション・カスタマイズ・ファイルを読み込みます。このプログラムは、ディスプレイ・サーバへの接続をオープンし、Xlib の呼び出しを行って、カスタマイズ内容を適用し、画面の背景、ポインタの形状と色、その他のワークステーションの設定を行います。また、ルート・ウィンドウ上で、後から起動した DECwindows アプリケーションにカスタマイズ内容を伝えるためのプロパティを作成します。

注意

XUI 版の DECwindows を使用した後で、DECwindows Motif システムに初めてログインすると、DECW\$WSINIT.EXE はユーザの XUI リソース・ファイルを読み込み、DECwindows Motif 形式に変換します。

- e. Kerberos 認証が必要な場合、Kerberos のログイン・ダイアログ・ボックスが表示されます。このダイアログはプリンシパルとパスワードを要求し、チケットを生成します
- f. 必要に応じて、ユーザのデフォルトの X authority ファイルへのエントリの挿入や削除を行います。
- g. SYS\$MANAGER:DECW\$SYLOGIN.COM があれば実行します。
- h. SYSSLOGIN:DECW\$LOGIN.COM があれば実行します。
- i. SYSSYSTEM:DECW\$SESSION.EXE を実行してセッション・マネージャを起動します。

4.3 ログイン環境のカスタマイズ

ここでは、DECwindows のデフォルトのログイン環境を改善またはカスタマイズする方法について説明します。

4.3.1 アプリケーションのスタートアップ性能の向上

コマンド・プロシージャ SYLOGIN.COM や LOGIN.COM が大きいと、アプリケーションのスタートアップに時間がかかります。SYLOGIN.COM や LOGIN.COM の中で行っている操作の多くは、DECwindows アプリケーションのスタートアップでは意味がありません。そこで、DECwindows アプリケーションのスタートアップ性能を向上させるために、SYLOGIN.COM ファイルや LOGIN.COM ファイルを条件付きにすることをお勧めします。

DECwindows アプリケーションの起動時には、最小限の SYLOGIN.COM コマンドおよび LOGIN.COM コマンドだけを実行します。通常、実行が必要なコマンドは、DECW\$USER_DEFAULTS の再定義 (存在する場合) と、DECwindows アプリケーションのコンテキストから参照するその他論理名の定義です。

次のコード・セグメントは、SYLOGIN.COM および LOGIN.COM 内で、DECwindows に必要なコマンドの直後に挿入することができます。

```
$ mode = f$mode()
$ tt_devname = f$trnlrm("TT")
$ session_mgr_login = (mode .eqs. "INTERACTIVE") .and. -
    (f$locate("WSA",tt_devname) .ne. f$len(tt_devname))
$ session_detached_process = (mode .eqs. "INTERACTIVE") .and. -
    (f$locate("MBA",tt_devname) .ne. f$len(tt_devname))
$ if session_mgr_login .or. session_detached_process then exit
```

4.3.2 ログイン画面のカスタマイズ (Traditional DECwindows Desktop のみ)

DECwindows Motif のログイン画面をカスタマイズして、別のロゴを表示したり、画面の色を変更することができます。ログイン画面をカスタマイズするには、リソース定義が格納されている SYSSMANAGER ディレクトリに、DECW\$LOGIN.DAT という名前のファイルを作成します。SYSSMANAGER:DECW\$LOGIN.DAT 内のカスタム・リソース定義は、SYSSCOMMON:[DECW\$DEFAULTS.SYSTEM]DECW\$LOGIN.DAT に格納されている弊社提供のリソース定義とマージされて、新しいログイン画面が構成されます。

カスタマイズしたリソース・ファイル DECW\$LOGIN.DAT は、新しいバージョンの DECwindows Motif ソフトウェアにアップグレードしたときに上書きされないように、DECW\$SYSTEM_DEFAULTS ディレクトリではなく SYSSMANAGER ディレクトリに格納してください。また、SYSSMANAGER ディレクトリにファイルを保存することで、弊社が提供しているファイルをカスタム・ファイルによって置き換えてしまうのを防ぐことができます。

4.3.2.1 ログおよびログイン画面の色のカスタマイズ

表 4-4 に示すリソースを定義することで、ロゴの位置と色、および「Start Session」画面の背景色を変更することができます。

表 4-4 ログの移動とログイン画面の色の変更

リソース	説明
rootColor	画面の背景色
logoColor	ロゴの色 (デフォルトは赤紫色)
logoX	ロゴの x 位置 (デフォルトは 0)
logoY	ロゴの y 位置 (デフォルトは 75)
centerLogoX	論理値。true (デフォルト) の場合、ロゴは画面の左右方向の中央に配置される。

たとえば、 $x=100$ 、 $y=600$ の位置にロゴを表示するには、次のリソース定義を SYSSMANAGER:DECWSLOGIN.DAT ファイルに追加します。

```
decw$login.logoX: 100  
decw$login.logoY: 600  
decw$login.centerLogoX: false
```

4.3.2.2 「Start Session」ダイアログ・ボックスと「Set Password」ダイアログ・ボックスの位置の変更

表 4-5 に示すリソースを定義することで、「Start Session」ダイアログ・ボックスと「Set Password」ダイアログ・ボックスの位置を変更することができます。

表 4-5 「Start Session」ダイアログ・ボックスと「Set Password」ダイアログ・ボックスの位置の変更

リソース	説明
centerStartSessionX	論理値。true (デフォルト) の場合、「Start Session」ダイアログ・ボックスは左右方向の中央に配置される。
centerStartSessionY	論理値。true (デフォルト) の場合、「Start Session」ダイアログ・ボックスは上下方向の中央に配置される。
centerSetPasswordX	論理値。true (デフォルト) の場合、パスワードの期限切れを示す「Set Password」ダイアログ・ボックスは、左右方向の中央に配置される。
centerSetPasswordY	論理値。true (デフォルト) の場合、「Set Password」ダイアログ・ボックスは上下方向の中央に配置される。

たとえば、「Set Password」ダイアログ・ボックスを $x=100$ 、 $y=600$ の位置に表示するには、次のリソース定義を SYSSMANAGER:DECWSLOGIN.DAT ファイルに追加します。

```
decw$login.centerStartSessionX: false  
decw$login.centerStartSessionY: false  
decw$login.HiddenShell.x: 100  
decw$login.HiddenShell.y: 600
```

「Set Password」ダイアログ・ボックスを $x=30$, $y=100$ の位置に表示するには、次のリソース定義を SYSSMANAGER:DECWSLOGIN.DAT ファイルに追加します。

```
decw$login.centerSetPasswordX: false
decw$login.centerSetPasswordY: false
decw$login.SetPasswordShell.x: 30
decw$login.SetPasswordShell.y: 100
```

4.3.2.3 「Start Session」ダイアログ・ボックスでのノード名表示を無効にする

「Start Session」ダイアログ・ボックスにノード名が表示されないようにするには、次のリソース定義を SYSSMANAGER:DECWSLOGIN.DAT ファイルに追加します。

```
decw$login.displayNodeName: false
```

4.3.3 ログイン前にカスタム・メッセージを表示する (New Desktop のみ)

システム管理者などの特権ユーザは、New Desktop でのセッション・ログインの前に、カスタム・メッセージ (挨拶文、セキュリティ・アップデート、システム・ブロードキャストなど) を表示することができます。

DECwindows Motif は、セッションのスタートアップ時に、メッセージ・ファイル SYSSMANAGER:DECWSGREET.TXT を探します。ファイルが見つかったら、メッセージ・ファイル中の文章が表示されたウィンドウが、ログイン・ダイアログ・ボックスの前に表示されます。ユーザは、Return キーを押すか [了解] をクリックして先に進み、New Desktop セッションにログインします。ファイルが見つからない場合は、このウィンドウは表示されず、ユーザは直接 New Desktop セッションにログインすることができます。

カスタム・メッセージを作成するには、以下の手順を実行します。

1. SYSTEM (または他の特権アカウント) でログインします。
2. 以下の場所のどちらかにファイル DECWSGREET.TXT を作成します。
 - スタンドアロン・システムの場合は、ファイルを SYSSSPECIFIC:[SYSMGR]ディレクトリに作成します。
 - クラスタ全体でメッセージを表示する場合は、ファイルを SYSSCOMMON:[SYSMGR]ディレクトリに作成します。
3. 表示するメッセージ・テキストを入力します。テキストは、現在 CDE\$SYSTEM_DEFAULTS:[CONFIG]XCONFIG.DAT および CDE\$SYSTEM_DEFAULTS:[CONFIG.%L]XRESOURCES.DAT で定義されているフォント・ファミリと言語に従って表示されます。現在のフォント・ファミリと言語でサポートされているプリント可能なすべての文字を表示することができます。

また、明確なサイズ要件がない点に注意してください。メッセージ・ウィンドウは動的にサイズが変わります。非常に長い行 (長すぎて画面自体に収まらない行) は切り捨てられます。

注意

テキストまたは書式制御文字を含まない行は無視されます。メッセージ・ファイルに空白行を挿入するには、行の先頭に1つ以上のスペース文字<sp>を入力します。

4. ファイルを保存してデスクトップ・セッションを再起動します。

4.3.4 推奨パスワード・リストを無効にする (New Desktop のみ)

New Desktop に期限切れのパスワードでログインしたときでも、推奨パスワード・リストが表示されないようにすることができます。

推奨パスワード・リストを抑制するには、システム論理名 CDE\$NOGENPWD にゼロ以外の値を定義します。次に例を示します。

```
$ DEFINE/SYSTEM CDE$NOGENPWD 1
```

4.3.5 UNIX スタイルのファイル名のサポートを有効にする (New Desktop のみ)

ファイル・セレクション widget およびファイル・マネージャで、ファイル名とデバイス名を UNIX スタイルで表示することができます。

この機能を有効にすると、ファイルとディレクトリは、ディレクトリ・ツリーを区切る際に大括弧の代わりにスラッシュを使用するなど、UNIX のパス名規約に従って表示されます。また、UNIX スタイルのパス名表示では、デバイス名は大文字に変換されず、大文字と小文字の違いが保持されます。

以降の項では、この機能を有効にする方法を簡単に説明します。

4.3.5.1 「File Selection」ダイアログ・ボックスで有効にする

「File Selection」ダイアログ・ボックスで UNIX スタイルのファイル名の表示を有効にするには、以下の論理名の1つ以上にゼロ以外の値を設定します。

```
DECC$FILENAME_UNIX_ONLY (CRTL モード)  
DECW$XM_FORCE_UNIX_NAMES_TO_VMS
```

これらの論理名を SYSS\$MANAGER:SYLOGICALS.COM ファイルに追加することで、システム全体で定義することができます。また、各ユーザの DECW\$LOGIN.COM ファイルまたは LOGIN.COM ファイルに追加することで、ユーザごとに定義することもできます。

「File Selection」ダイアログ・ボックスで、UNIX 形式で表示する場合でも、選択されたファイル名は OpenVMS 形式で返すようにするには、論理名 DECW\$XM_UNIX_NAMES_TO_VMS を定義します。これにより、UNIX 形式でファイル名を表

示する一方で、OpenVMS 形式のファイル名を必要とするその他のアプリケーションが、「File Selection」ダイアログ・ボックスと正常にやり取りすることができます。

4.3.5.2 ファイル・マネージャで有効にする

ファイル・マネージャで UNIX スタイルのファイル名の表示を有効にするには、論理名 CDE\$DTFILE_UNIX_NAMES にゼロ以外の値を設定します。

この論理名をシステム全体で定義するには、SYSSMANAGER:SYLOGICALS.COM ファイルに追加します。ユーザごとに定義するには、各ユーザの DECW\$LOGIN.COM ファイルまたは LOGIN.COM ファイルに追加します。

4.4 スタートアップ環境のカスタマイズ

ここでは、アプリケーション・カスタマイズ・パラメータを使用して、DECwindows Motif のデフォルトのスタートアップ環境を変更する方法について概要を説明します。

4.4.1 DECW\$PRIVATE_APPS_SETUP ファイルの利用

DECW\$PRIVATE_APPS_SETUP.TEMPLATE ファイルは、SYSSMANAGER ディレクトリにあります。このテンプレート・ファイルには、DECwindows Motif のスタートアップ環境をカスタマイズするために使用できる情報が格納されています。

このファイルを使用することで、表 4-6 に示すクライアント領域を変更することができます。デスクトップに依存するパラメータには注が付けられています。New Desktop 環境固有の追加パラメータについては、『日本語 DECwindows Motif New Desktop 使用概説書』を参照してください。

表 4-6 変更可能なクライアント領域

クライアント領域	グローバル・シンボル	デフォルト
アプリケーション		
Bookreader のオンライン・ブック・ディレクトリ	DECW\$BOOK	SYSSSYSROOT:[DECW\$BOOK]
FileView のコマンド・ファイル・ディレクトリ	VUE\$LIBRARY ¹	DECW\$SYSCOMMON: [VUE\$LIBRARY.USER], DECW\$SYSCOMMON: [VUE\$LIBRARY.SYSTEM]
FileView の公用プロファイル・ファイル・ディレクトリ	VUE\$LIBRARY_WRITE ¹	SYSSCOMMON: [VUE\$LIBRARY.USER]

¹Traditional DECwindows Desktop のみ。

(次ページに続く)

DECwindows の使用方法
4.4 スタートアップ環境のカスタマイズ

表 4-6 (続き) 変更可能なクライアント領域

クライアント領域	グローバル・シンボル	デフォルト
コンソール・ウィンドウ		
コンソール・ウィンドウへのシステム・メッセージの表示を有効または無効にする	DECW\$CONSOLE_SELECTION	無効
コンソール・ウィンドウの位置を指定する	DECW\$CONSOLE_GEOMETRY	-0,-0
デスクトップ		
セッションのスタートアップ時に実行するデスクトップ	DECW\$START_NEW_DESKTOP	True (New Desktop を実行)
ディスプレイ・デバイス		
SET DISPLAY コマンドのパラメータ	DECW\$APPSNODE DECW\$APPSSERVER DECW\$APPSSCREEN	ノード 0 サーバ 0 画面 0
TCP/IP トランスポート上の IPv6 のサポートを有効または無効にする	DECW\$IPV6_SUPPORT	なし
例と自習書		
プログラム例のディレクトリ	DECW\$EXAMPLES ²	SYSSYSROOT: [SYSHLP.EXAMPLES.DECW]
学習支援システムのファイルがあるディレクトリ	DECW\$CBI	SYSSYSROOT: [SYSCBI.DECW\$CBI]
キーマップ		
キーマップ・ファイルのディレクトリ	DECW\$KEYMAP	DECW\$SYSCOMMON: [SYSSKEYMAP.DECW.USER], DECW\$SYSCOMMON: [SYSSKEYMAP.DECW.SYSTEM]
ログイン		
LOGINOUT ログ・ファイル	DECW\$LOGINLOG	なし
loginout 中に複数の接続を使用	DECW\$LOGINMANY	False
ログイン・ロゴ	DECW\$LOGINLOGO ¹	なし
ログイン・ロゴをサブプロセスで実行	DECW\$LOGINLOGOSUB ¹	False (独立プロセス)

¹Traditional DECwindows Desktop のみ。

²別のディレクトリを指すようにこのシンボルを再定義する場合は、DECW\$UTILS シンボルの定義もそれに合わせて変更する必要があります。

(次ページに続く)

表 4-6 (続き) 変更可能なクライアント領域

クライアント領域	グローバル・シンボル	デフォルト
ログイン		
loginout 中に指定された X authority ファイルからデータを提供	DECW\$LOGIN_XAUTHORITY	SYSS\$LOGIN: DECW\$XAUTHORITY.DECW\$XAUTH
プロキシ・マネージャ		
プロキシ・マネージャ構成ファイル	DECW\$PROXY_MANAGER_CONFIG	なし
プロキシ・マネージャ・ログ・ファイル	DECW\$PROXY_MANAGER_LOG	なし
プロキシ・マネージャ・アプリケーションを起動する際のデフォルトのコマンド行オプション	DECW\$PROXY_MANAGER_OPTIONS	/PORT=6500/TRANSPORT="LOCAL,TCPIP,DECNET"
プロキシ・マネージャを独立プロセスとして起動する際のデフォルトのコマンド行オプション	DECW\$PROXY_MANAGER_QUOTAS	/PROCESS_NAME=DECW\$PROXY
セッション管理		
セッションを実行するコマンド	DECW\$SESSIONMAIN ¹	Run SYSS\$SYSTEM: DECW\$SESSION
セッションをリセットするコマンド	DECW\$SESSIONEND ¹	なし
セッション・ログの有効化または無効化	DECW\$SESSIONLOG ¹	True (ログ・ファイルはオン)
システム・スタートアップ		
DECwindows のスタートアップ完了時に実行する DCL コマンド	DECW\$MAINAPP ¹	Run SYSS\$SYSTEM: DECW\$STARTLOGIN
ユーザ認証完了後に実行するコマンド・プロシージャ	DECW\$SESSIONCOM	SYSS\$MANAGER: DECW\$STARTSM.COM
DECwindows Motif の SYLOGIN コマンド・プロシージャ	DECW\$SYLOGINCOM	なし
UNIX との互換性		
UNIX との互換性のための論理名を定義	DECW\$UNIXLOGICALS	True

¹Traditional DECwindows Desktop のみ。

(次ページに続く)

表 4-6 (続き) 変更可能なクライアント領域

クライアント領域	グローバル・シンボル	デフォルト
ユーザ・リソース		
正しいユーザ・リソース値でシステムを初期化するためのコマンド	DECW\$SESSIONINIT	Run SYSS\$SYSTEM: DECW\$WSINIT
ユーティリティ		
X ウィンドウ・システム・ユーティリティ・ディレクトリ	DECW\$UTILS	SYSS\$SYSROOT:[SYSHLP. EXAMPLES.DECW.UTILS]

上記の領域のいずれかをカスタマイズするには、SYSS\$MANAGER:DECW\$PRIVATE_APPS_SETUP.TEMPLATE ファイルを SYSS\$MANAGER:DECW\$PRIVATE_APPS_SETUP.COM にコピーします。次に例を示します。

```
$ COPY SYSS$COMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.TEMPLATE -
_$ SYSS$COMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.COM/LOG
```

次に、ファイルを編集して、該当するシンボルを変更します。

以降の項では、テンプレート・ファイルを使用して、DECwindows のクライアント環境をカスタマイズする例を示します。

このファイルには、Cluster Common セクションと Cluster Member Workstation-Specific セクションの 2 つのセクションがあります。適切なセクションに新しいシンボルを追加します。

4.4.2 デスクトップの切り替え

インストール時に選択したものと異なるデスクトップをロードするには、DECW\$PRIVATE_APPS_SETUP.COM 内の DECW\$START_NEW_DESKTOP シンボルに TRUE (New Desktop) または FALSE (Traditional DECwindows Desktop) を設定します。

ファイル SYSS\$MANAGER:DECW\$PRIVATE_APPS_SETUP.COM が存在しない場合には、次のようにしてテンプレートから作成します。

```
$ COPY SYSS$MANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE -
_$ SYSS$COMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.COM
```


4.4.3 IPv6 サポートを有効にする

IPv6 の利用を有効にするには、DECW\$PRIVATE_APPS_SETUP.COM ファイル内でグローバル・シンボル DECW\$IPV6_SUPPORT を定義します。ユーザは、必要に応じて、この結果得られるシステム全体の論理名を設定変更することができます。以下の値のいずれかを使用してシンボルを定義します。

- TCP_IS_INET6

ネットワークが IPv6 をサポートするように構成されている場合には、この値を使用します。トランスポート名 TCPIP および TCP は、INET6 トランスポートの同義語として解釈されます。INET6 トランスポートでは、IPv6 が利用可能な場合にはそれを使おうとしますが、IPv6 が使用される保証はありません。個々のプログラムでは、トランスポート・オプション INET6 または INET を明示的に指定することができるためです。

この値を設定したときに、ドメイン・ネーム・サーバが IPv6 を扱うように構成されていない場合は、ディスプレイ接続のオープン時に遅延が発生します。

- DISABLED

この値は、アプリケーションが、ユーザ・モードの AST ルーチンからディスプレイ接続をオープンする場合にだけ使用します。トランスポート名 TCPIP および TCP は、INET トランスポートの同義語として解釈されます。INET6 トランスポート・オプションは使用できません。

- その他の値

この値は、ネットワークが IPv6 をサポートしていないか、一部の IPv6-over-IPv4 トンネルのためだけに IPv6 を使用している場合に使用します。トランスポート名 TCPIP および TCP は INET トランスポートの同義語です。DISABLED オプションと異なり、個々のプログラムがトランスポート・オプション INET6 または INET を明示的に指定することができます。これは、論理名が定義されていない場合のデフォルト値です。

この論理名のデフォルトの動作により、以前のバージョンの DECwindows Motif と同じディスプレイ名とクライアント-サーバ接続機能が提供されます。

注意

TCP_IS_INET6 オプションでは TCP/IP Services の gethostbyaddr 関数の代わりに、getaddrinfo 関数を使用します。TCP/IP Services では、AST ルーチンからの getaddrinfo 関数の利用はサポートされていません。使用するアプリケーションが AST ルーチン内から DECwindows トランスポートをオープンする場合には、DISABLED オプションを使用することをお勧めします。DECwindows Motif アプリケーションは、すべて TCP_IS_INET6 オプションをサポートしています。

4.4.4 デフォルトのロゴの変更 (Traditional DECwindows Desktop のみ)

独自のロゴを表示する個別のアプリケーションを実行することで、デフォルトの HP ロゴを別のロゴで置き換えることができます。

それには、DECW\$PRIVATE_APPS_SETUP.COM ファイル内でグローバル・シンボルを定義し、ロゴを表示するコマンドが格納された DCL コマンド・ファイルを作成する必要があります。たとえば、ログイン画面に独自のロゴを表示するには、以下の手順を実行します。

- コマンド・ファイル (たとえば SYSS\$MANAGER:LOGO.COM) を作成します。
- SYSS\$MANAGER:DECW\$PRIVATE_APPS_SETUP.COM 内でグローバル・シンボル DECW\$LOGINLOGO を定義します。シンボルの値には、コマンド・ファイルの名前を設定します。このコマンド・ファイルはシステム・アカウントで実行される点に注意してください。

カスタム・ロゴを作成するには、以下の手順を実行します。

1. SYSS\$MANAGER ディレクトリにコマンド・ファイルがない場合は、テンプレート・ファイルをコピーしてコマンド・ファイルとします。

```
$ COPY SYS$COMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.TEMPLATE -  
_ $ SYS$SPECIFIC:[SYSMGR]DECW$PRIVATE_APPS_SETUP.COM/LOG
```

2. ファイルを編集し、カスタム・ロゴを表示するコマンド・ファイルを指すようにシンボル DECW\$LOGINLOGO を定義します。

次の行を追加します。

```
$ DECW$LOGINLOGO == "SYS$MANAGER:MYLOGO.COM"
```

3. ロゴを表示するコマンド・ファイルを作成します。ファイル名とディレクトリ名は、ステップ 2 で指定したものに合わせる必要があります。コマンド・ファイルは、SYSTEM アカウントで独立プロセスとして実行される点に注意してください。カスタム・ロゴを通じて、システム・アカウントへのアクセスをユーザに許可してしまわないように注意してください。カスタム・ロゴとして FileView や DECterm などのアプリケーションを実行しないようにしてください。

たとえば、ファイル SYSS\$MANAGER:MYLOGO.COM に、プログラム例として添付されている、二十面体を表示する次のコマンドを記述します。

```
$ RUN DECW$EXAMPLES:ICO
```

4. DECW\$STARTUP を次のように実行してワークステーションをリポートし、DECwindows を再起動します。

```
$ @SYS$STARTUP:DECW$STARTUP RESTART
```

HP ロゴを復元するには、DECW\$PRIVATE_APPS_SETUP.COM ファイルから DECW\$LOGINLOGO の定義を削除し、DECwindows を再起動します。

4.4.5 個人使用ライセンスを使用したシステムでのロゴの表示 (Traditional DECwindows Desktop のみ)

DECwindows Motif のライセンスが SYSTEM アカウントに対して登録されていない場合、DECwindows はカスタマイズされたログイン・ロゴを表示しません。これは、DECwindows Motif の個人使用ライセンスを使用したシステムで、許可された DECwindows ユーザの一覧に SYSTEM が含まれていない場合に起こります。

個人使用ライセンスを使用した DECwindows Motif でカスタマイズされたロゴを表示するには、次の定義を SYSSMANAGER:DECW\$PRIVATE_APPS_SETUP.COM ファイルに追加します。

```
$ DECW$LOGINLOGOSUB == "TRUE"
```

設定ファイルを編集したら、次のようにして DECwindows Motif を再起動します。

```
$ @SYSSMANAGER:DECW$STARTUP RESTART
```

DECwindows Motif ログインは、独立プロセスとしてではなく、サブプロセスとしてロゴ・プロセスを起動します。

4.4.6 コンソール・メッセージの表示

Console Window アプリケーションを使用して、オペレータ・メッセージ (OPCOM) をコンソール・ウィンドウに表示することができます。コンソール・ウィンドウは、画面上部の 6 行の表示領域です。

注意

コンソール通信用に代替ポート (OPA0: 以外) を選択した場合は、DECwindows Console Window は無効となり、コンソール・ブロードキャストが有効となります。代替コンソールポートの選択についての詳細は、使用しているワークステーションのオーナーズ・ガイドを参照してください。

メッセージの表示方法を指定するには、DECW\$PRIVATE_APPS_SETUP.COM ファイル内でグローバル・シンボル DECW\$CONSOLE_SELECTION を定義します。値には、WINDOW、DISABLE、ENABLE のいずれかを設定します。

- WINDOW

Console Window アプリケーション内にコンソール・メッセージを表示します。値 WINDOW を指定すると、デフォルトではログイン画面の右下に Console Window が表示され、ユーザがシステムにログインした後も表示されたままになります。

Console Window アプリケーションは、Message Window と同じ実行可能ファイルを共有し、表示も似ています。ただし、Console Window にはメニュー・バーが表示されません。また、リソースは DECW\$MESSAGEPANEL.DAT ファイルからではなく DECW\$CONSOLE.DAT ファイルから読み込まれます。内部的には、実行可能ファイル DECW\$MESSAGEPANEL.EXE にコマンド行オプション *-console* を指定することで Console Window が起動されます。

Console Window の初期位置と、有効になっている OPCOM 出力のクラスを制御するには、ファイル SYSS\$MANAGER:DECW\$PRIVATE_APPS_SETUP.COM 内でグローバル・シンボル DECW\$CONSOLE_GEOMETRY を定義します。

DECW\$CONSOLE_GEOMETRY シンボルは、DECW\$MESSAGEPANEL.EXE コマンド行の *-geometry* オプションの値を指定します。このコマンドは、Console Window アプリケーションを起動するために使用されます。デフォルト値は“-0-0”で、画面の右下にウィンドウを表示することを指定します。

たとえば、ウィンドウを画面の左下に表示するには、コマンド・ファイル SYSS\$MANAGER:DECW\$PRIVATE_APPS_SETUP.COM に次の行を追加します。

```
$ DECW$CONSOLE_GEOMETRY == "+0-0"
```

- DISABLE (デフォルト)

OPA0: デバイスへのブロードキャストを無効にします。コンソール・メッセージは表示されません。

- ENABLE

コンソール・ウィンドウにコンソール・メッセージを表示します。

注意

DECwindows Motif の以前のバージョンでは ENABLE がデフォルトでしたが、このオプションは使用しないことをお勧めします。コンソール・メッセージをデフォルトでコンソール・ウィンドウに表示すると、ワークステーションのディスプレイの表示内容が崩れることがあります。

4.4.7 専用アカウントの作成 (Traditional DECwindows Desktop のみ)

システム管理者は、特定のアプリケーションだけを実行するユーザ・アカウントを設定することができます。たとえば、Bookreader アプリケーションだけをユーザに使用させたい場合があります。

ここでは、専用の DECwindows アカウントを作成する以下の 3 つの方法について説明します。

- セッション・マネージャのコマンド・プロシージャを変更する方法
- セッション・マネージャの実行可能ファイルを変更する方法
- セッション・マネージャのプロファイル・ファイルを変更する方法

4.4.7.1 セッション・マネージャのコマンド・プロシージャを変更する

SYSSMANAGER:DECW\$PRIVATE_APPS_SETUP.COM ファイル内でグローバル・シンボル DECW\$SESSIONCOM を定義することで、ユーザがログインしたときに、LOGINOUT によって DECW\$STARTSM.COM 以外のコマンド・ファイルが実行されるようにすることができます。この動作によって、セッション・マネージャが完全に迂回されます。

例 4-1 では、DECW\$PRIVATE_APPS_SETUP.COM 内で、プライベート・コマンド・プロシージャ SYSSMANAGER:PRIVATE_SESSIONCOM.COM を指すように DECW\$SESSIONCOM が定義されています。デフォルトのコマンド・ファイルは SYSSMANAGER:DECW\$STARTSM.COM です。

DECW\$SESSIONCOM はシステム論理名テーブルで定義されているため、すべてのユーザ名のセッション・マネージャのスタートアップに影響を与えます。プライベート・コマンド・ファイルでは、実行元のユーザ名を確認し、専用アカウント以外のアカウントに対しては、通常の DECW\$STARTSM.COM を実行する必要があります。

注意

この方法では、スタートアップ・コマンド・プロシージャ SYLOGIN.COM, LOGIN.COM, DECW\$SYLOGIN.COM, および DECW\$LOGIN.COM は実行されません (通常は DECW\$STARTSM.COM から実行されます)。

DECwindows の使用方法

4.4 スタートアップ環境のカスタマイズ

セッション・マネージャのコマンド・プロシーダを変更するには、以下の手順を実行します。

1. 例 4-1 に示すコマンド・プロシーダを作成します。

例 4-1 プライベート・セッション・マネージャ・コマンド・ファイルの使用例

```
$ CREATE SYS$MANAGER:PRIVATE_SESSIONCOM.COM
$!
$! Check to see whether this is the dedicated DECwindows account.
$! If it isn't, run the normal procedure to start Session Manager.
$!
$ username = F$USER()
$ IF F$LOCATE("BOOK_READER", username) .NE. F$LENGTH(username) -
  THEN GOTO dedicated
$!
$! Normal, nondedicated DECwindows login
$!
$ @SYS$MANAGER:DECW$STARTSM
$ EXIT
$!
$! Dedicated DECwindows login. Run Bookreader and then exit.
$!
$ dedicated:
$!
$ RUN SYS$SYSTEM:DECW$SINIT
$ display = F$TRNLNM("DECW$DISPLAY")
$ RUN/DETACHED/OUTPUT='display' SYS$SYSTEM:DECW$MWM 1
$ RUN SYS$SYSTEM:DECW$BOOKREADER
$ endsession := $DECW$ENDSESSION 2
$ endsession -noprompt 3
$ STOP/ID=0 4
```

- 1 DECW\$MWM は Motif Window Manager です。独立プロセス内で DECW\$DISPLAY が正しく設定されるように、その SYS\$OUTPUT 定義には、DECW\$DISPLAY の展開結果を設定する必要があります。
 - 2 DECW\$ENDSESSION はサーバをリセットし、「Start Session」ダイアログ・ログイン・ボックスを表示します。
 - 3 -noprompt 修飾子を指定することで、DECW\$ENDSESSION は、セッションを終了するかどうかをユーザに確認しなくなります。(ユーザがセッション終了動作をキャンセルすると、Bookreader が終了してもログイン・ボックスが表示されないため、ワークステーションが使用できなくなります。)
 - 4 STOP/ID=0 は、エラー・メッセージを SYS\$OUTPUT に書き込まずにプロセスをログアウトさせます。
2. DECW\$PRIVATE_APPS_SETUP.COM 内で DECW\$SESSIONCOM シンボルを次のように定義します。

```
$ DECW$SESSIONCOM ::= SYS$MANAGER:PRIVATE_SESSIONCOM.COM
```

SYSSMANAGER:DECW\$PRIVATE_APPS_SETUP.COM がない場合は、次のようにしてテンプレート・ファイルから作成します。

```
$ COPY SYSSMANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE -  
_ $ SYSSCOMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.COM
```

3. 次のコマンドを入力して、DECwindows を再起動します。

```
$ @SYSSMANAGER:DECW$STARTUP RESTART
```

このステップは、DECW\$PRIVATE_APPS_SETUP.COM への変更を有効にするために必要です。

4. 専用アカウントでDECwindows システムにログインします。

専用アカウントのユーザ名とパスワードを入力すると、Bookreader が起動し、セッション・マネージャは表示されません。ユーザが Bookreader を終了すると、PRIVATE_SESSIONCOM.COM プロシージャがセッション操作を終了し、ログアウトします。End Session コマンドによって「Start Session」画面が表示されます。

4.4.7.2 セッション・マネージャの実行可能ファイルを変更する

SYSSMANAGER:DECW\$PRIVATE_APPS_SETUP.COM ファイル内で DECW\$SESSIONMAIN を定義することで、DECW\$STARTSM.COM が、DECW\$SESSION.EXE の代わりに、指定されたコマンド・ファイルを実行することができます。セッション・マネージャはリソース・ファイルを読み込み、ログイン・コマンド・ファイルを実行しますが、セッション・マネージャのメニュー・バーは表示されません。この方法では、End Session コマンドが確認を促さないようにしてください。

DECW\$STARTSM.COM が起動すると (すなわち、DECW\$SESSIONCOM がデフォルト値の場合)、論理名 DECW\$SESSIONMAIN に格納されている DCL コマンドを実行することでセッション・マネージャが起動されます。この論理名を定義するには、SYSSMANAGER:PRIVATE_APPS_SETUP.COM を編集して、グローバル・シンボル DECW\$SESSIONMAIN に、実行する DCL コマンドを定義します。

この方法は、第 4.4.7.1 項で説明した方法と似ていますが、この方法では DECW\$STARTSM.COM がコマンド・プロシージャ SYLOGIN.COM、LOGIN.COM、DECW\$SYLOGIN.COM、および DECW\$LOGIN.COM を実行します。また、DECW\$STARTSM.COM は、プライベート・コマンド・プロシージャで実行しなくてもすむように、DECW\$WSINIT.EXE を実行します。

DECwindows の使用方法

4.4 スタートアップ環境のカスタマイズ

セッション・マネージャの実行可能ファイルを変更するには、以下の手順を実行します。

1. 例 4-2 に示すコマンド・プロシージャを作成します。

例 4-2 セッション・マネージャの実行可能ファイルを変更する

```
$ CREATE SYS$MANAGER:PRIVATE_SESSIONMAIN.COM
$!
$! Check to see whether this is the dedicated DECwindows account.
$! If it isn't, run the normal procedure to start Session Manager.
$!
$ username = F$USER()
$ IF F$LOCATE("BOOK READER", USERNAME) .NE. F$LENGTH(USERNAME) -
  THEN GOTO dedicated
$!
$! Normal, nondedicated DECwindows login
$!
$ RUN SYS$SYSTEM:DECW$SESSION
$ EXIT
$!
$! Dedicated DECwindows login. Run Bookreader and then exit.
$!
$ dedicated:
$!
$ display = F$TRNLNM("DECW$DISPLAY")
$ RUN/DETACHED/OUTPUT='display' SYS$SYSTEM:DECW$MWM 1
$ RUN SYS$SYSTEM:DECW$BOOKREADER
$ endsession := $DECW$ENDSESSION 2
$ endsession -noprompt 3
$ STOP/ID=0 4
```

- 1 DECW\$MWM は、Motif Window Manager です。独立プロセス内で DECW\$DISPLAY が正しく設定されるように、その SYS\$OUTPUT 定義には、DECW\$DISPLAY の展開結果を設定する必要があります。
 - 2 DECW\$ENDSESSION はサーバをリセットし、新しい「Start Session」画面を表示します。
 - 3 -noprompt 修飾子を指定することで、DECW\$ENDSESSION は、「End Session」プロンプトを表示しなくなります。(ユーザが「End Session」プロンプトをキャンセルすると、Bookreader が終了しても Start Session 画面が表示されないため、ワークステーションが使用できなくなります。)
 - 4 STOP/ID=0 は、メッセージを SYS\$OUTPUT に書き込まずにプロセスをログアウトさせます。
2. DECW\$PRIVATE_APPS_SETUP.COM 内で DECW\$SESSIONMAIN シンボルを定義します。

このコマンド・プロシーダを編集し、DECW\$SESSIONMAIN を定義します。シンボルにはDCL コマンドを定義する必要がある点に注意してください(コマンド・プロシーダ名を定義する DECW\$SESSIONCOM とは異なります)。

```
$ DECW$SESSIONMAIN ::= @SYS$MANAGER:PRIVATE_SESSIONMAIN.COM
```

SYS\$MANAGER:DECW\$PRIVATE_APPS_SETUP.COM が存在しない場合は、次のようにしてテンプレート・ファイルから作成します。

```
$ COPY SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE -  
_ $ SYS$COMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.COM
```

3. 次のコマンドを入力して DECwindows を再起動します。

```
$ @SYS$MANAGER:DECW$STARTUP RESTART
```

このステップは、DECW\$PRIVATE_APPS_SETUP.COM に対する変更を有効にするために必要です。

4. 専用アカウントで DECwindows システムにログインします。

専用アカウントのユーザ名とパスワードを入力すると、Bookreader が起動し、セッション・マネージャのメニュー・バーは表示されません。ユーザが Bookreader を終了すると、PRIVATE_SESSIONMAIN.COM プロシーダがセッション終了処理を実行し、ログアウトします。End Session コマンドによって DECwindows の「Start Session」画面が表示されます。

4.4.7.3 セッション・マネージャのプロファイル・ファイルを変更する

この方法では、通常どおりセッション・マネージャが起動しますが、システム管理者がメニューをカスタマイズして、自動的に起動されないアプリケーションをすべて削除します。特定のアプリケーションを削除することで、ユーザのアプリケーションへのアクセスを制限することができます。

この方法では、セッション・マネージャのメニュー・バーが表示され、ユーザは、システム管理者が無効にしていないあらゆるダイアログ・ボックスと対話することができます。

セッション・マネージャのプロファイル・ファイルを変更するには、以下の手順を実行します。

1. 専用アカウントで DECwindows システムにログインします。

専用アカウントのユーザ名とパスワードを入力すると、セッション・マネージャが起動します。

2. ユーザがログインしたときに表示するアプリケーションを、自動スタートアップ・リストに追加します。

セッション・マネージャの[オプション]メニューから、[自動起動...]を選択します。セッション・マネージャによって、「自動起動」ダイアログ・ボックスが表示されます。

このダイアログ・ボックスを使用して、自動スタートアップ・リストから、FileView などの不要なアプリケーションを削除します（「自動起動」ダイアログ・ボックスの使用方法についての詳細は、『日本語 DECwindows Motif for OpenVMS 操作ガイド』を参照してください）。

注意

自動スタートアップ・リストからウィンドウ・マネージャを削除しないでください。

この例では、Bookreader をリストに追加して[了解]をクリックします。

3. セッション・マネージャのメニュー・バーからアプリケーションを削除します。
セッション・マネージャの[オプション]メニューから、[メニューバー...]を選択します。セッション・マネージャによって「メニュー・バー」ダイアログ・ボックスが表示されます。

このダイアログ・ボックスを使用して、表示しないアプリケーションを削除します。（「メニュー」ダイアログ・ボックスの使用方法についての詳細は、『日本語 DECwindows Motif for OpenVMS 操作ガイド』を参照してください。）

[了解]をクリックし、変更を保存してダイアログ・ボックスを閉じます。

4. セッション・マネージャの[オプション]メニューから[セッション・マネージャ保管]を選択して、メニュー・バーの設定を保存します。
5. 表示しないメニュー項目を[オプション]メニューから削除します。

セッション・マネージャの[オプション]メニューから[メニュー...]を選択し、セッション・マネージャの[オプション]メニューから以下のメニュー項目を削除します。

[自動設定...]
[メニュー...]
[メニューバー...]
[セッション・マネージャ保管]

また、専用アカウントからのアクセスを禁止する、[セキュリティ]などのメニュー項目も削除します。

項目を削除したら、[了解]をクリックしてダイアログ・ボックスを閉じ、メニューの設定を保存します。

6. セッション・マネージャの[セッション]メニューから[セッション終了]を選択します。

セッション・マネージャは専用アカウントをログアウトさせ、DECwindows のログイン・ボックスを表示します。

7. 専用アカウントで再度ログインします。

変更内容をテストするために、DECwindows のログイン・ボックスから専用アカウントでログインします。デフォルトで Bookreader とセッション・マネージャが起動し、他のアプリケーションは起動できなくなっています。ログアウトするには、セッション・マネージャの[セッション]メニューから[セッション終了]を選択します。

アカウントを通常のアカウントに戻す場合や、システム・メニュー・バーから削除したダイアログ・ボックスを変更する場合は、専用アカウントの SYS\$LOGIN ディレクトリにある VUES\$PROFILE.VUES\$DAT ファイルを削除するか名前を変更します。

4.4.8 カスタム Bookreader ディレクトリの作成

Bookreader を起動する際に、システムは論理名 DECW\$BOOK を使用してファイル LIBRARY.DECW\$BOOKSHELF を探します。デフォルトでは、論理名 DECW\$BOOK は SYS\$SYSROOT:[DECW\$BOOK]に展開されます。LIBRARY.DECW\$BOOKSHELF が格納されたプライベート・ディレクトリを作成することができます。

そのためには、ディレクトリを作成し、DECW\$PRIVATE_APPS_SETUP.COM ファイル内にグローバル・シンボルを定義する必要があります。

SYS\$COMMON:[DECW\$BOOK_LOCAL]というプライベート Bookreader ディレクトリを作成するには、以下の手順を実行します。

1. ディレクトリを作成し、world に読み取りアクセスを許可します。次に例を示します。

```
$ CREATE/DIRECTORY/PROTECTION=WORLD:R -  
_ $ SYS$COMMON:[DECW$BOOK_LOCAL]
```

2. Bookreader の LIBRARY.DECW\$BOOKSHELF ファイルを新しいディレクトリにコピーします。

```
$ COPY SYS$SYSROOT:[DECW$BOOK]LIBRARY.DECW$BOOKSHELF -  
_ $ SYS$COMMON:[DECW$BOOK_LOCAL]/LOG
```

3. SYSSMANAGER ディレクトリにコマンド・ファイル DECW\$PRIVATE_APPS_SETUP.COM が格納されていない場合は、テンプレート・ファイルをコマンド・ファイルとしてコピーします。

```
$ COPY SYSSMANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE -  
_ $ SYSSMANAGER:DECW$PRIVATE_APPS_SETUP.COM/LOG
```

4. ファイルを編集して、個人の Bookreader ディレクトリを指すようにグローバル・シンボル DECW\$BOOK を定義します。

次の行を追加します。

```
$ DECW$BOOK == "SYS$COMMON:[DECW$BOOK_LOCAL]"
```

5. 次のようにして DECwindows Motif を再起動します。

```
$ @SYS$MANAGER:DECW$STARTUP RESTART
```

4.5 セッション・マネージャの動作の変更 (Traditional DECwindows Desktop のみ)

いくつかの論理名を使用することで、DECwindows セッション・マネージャの動作を変更することができます。これらの論理名は、通常は使用しませんが、2つのセッション・マネージャを同じシステムで動作させるなど、特別な状況で役立ちます。表 4-7 に論理名とその意味を示します。

表 4-7 セッション・マネージャの論理名

論理名	意味
DECW\$VUENOAUTOSTART	セッション・マネージャの自動起動リスト中のアプリケーションの起動を抑制する。
DECW\$VUENORESET	セッション・マネージャの終了時のディスプレイ・サーバのリセットを抑制する。
DECW\$VUELOGINOUTPUT	セッション・マネージャからアプリケーションが起動されるに従って、ログイン・コマンド・ファイル (SYSLOGIN, LOGIN, DECW\$LOGIN) からの出力を保存する。この情報には、「セッション・マネージャ」ウィンドウの[セッション]メニューにある「処理中」ダイアログ・ボックスの[出力表示]ボタンをクリックすることでアクセスできる。

これらの論理名は、DECterm またはログイン・コマンド・ファイル (SYSLOGIN, LOGIN, DECW\$LOGIN) で定義します。これらの論理名を設定するには、論理名に任意の値を定義します。次に例を示します。

```
$ DEFINE DECW$VUENOAUTOSTART 1
```

4.6 システム・リソース・ファイルの変更

ほとんどの DECwindows アプリケーションには、DECwindows 環境をカスタマイズし、設定を保存するためのオプションが表示されたダイアログ・ボックスがあります。たとえば、セッション・マネージャの[オプション]メニューを使用すると、多くの事前に定義された設定を参照し、新しい設定を選択して保存することができます。

ただし、設定を変更して保存するためのダイアログ・ボックスがすべてのアプリケーションにあるわけではありません。たとえば、アプリケーションの中には、アプリケーションのメイン・ウィンドウのサイズと位置を変更して保存できるものもあれば、そうでないものもあります。ダイアログ・ボックスで変更できない設定を変更して保存するには、DECW\$XDEFAULTS.DAT ファイルにリソースを指定します。

警告

以下の理由から、リソースを変更する際には十分に注意してください。

- 「オプション」ダイアログ・ボックスを使用して、DECW\$XDEFAULTS.DAT ファイルで指定した設定を変更することはできません。
 - 将来の DECwindows のリリースでは、Xdefaults ファイルに追加したオプションが、アプリケーションのダイアログ・ボックスに含まれる場合があります。そのような変更を反映するためには、DECW\$XDEFAULTS.DAT ファイルを変更する必要があります。
 - このバージョンの DECwindows では、リソースのエラーは処理できません。リソース・ファイルに構文エラーがあると、DECwindows はエラー・メッセージを表示せずに失敗します。
 - 大きな DECW\$XDEFAULTS.DAT ファイルを使用すると、アプリケーションの起動性能が大幅に低下します。
-

リソースを使用してアプリケーションの設定を変更するには、論理名 DECW\$USER_DEFAULTS が指すディレクトリにファイル (DECW\$XDEFAULTS.DAT) を作成します。このファイルで、変更したいアプリケーション設定のリソースを指定します。また、変更が 1 つのアプリケーションに特有の場合は、アプリケーションのリソース・ファイル (たとえば DECW\$MAIL.DAT) を変更することもできます。

DECW\$XDEFAULTS.DAT ファイル内の各リソース指定は、明示的な構文規則に従う点に注意してください。リソース構文についての詳細は、『VMS DECwindows Guide to Xlib (Release 4) Programming: MIT C Binding』または『X Window System』を参照してください。

アプリケーション固有のリソースに関しては、『日本語 DECwindows Motif New Desktop 使用概説書』および『日本語 DECwindows Motif for OpenVMS 操作ガイド』を参照してください。

4.7 クライアント・アクセス制御の指定

クライアント・アプリケーションが X ディスプレイ・サーバに接続する際に、サーバは、どのアクセス制御方式を使用するかを、現在の X authority ファイルを参照して決定します。X authority ファイルでは、クライアントの接続先のワークステーションに基づいて、使用するプロトコルを識別します。X authority ファイルを変更するには、「Security Options」ダイアログ・ボックスを使用します。

4.7.1 セキュリティ・オプションの設定

このワークステーション上のクライアント・アプリケーションが X サーバに接続する際に使用するアクセス制御方式を指定するには、以下の手順を実行します。

1. デスクトップに応じて以下のどちらかを実行します。
 - Traditional Desktop の場合は、セッション・マネージャの[オプション]メニューから[セキュリティ...]を選択します。
 - New Desktop の場合は、スタイル・マネージャの[J-Security]コントロールをクリックします。「Security Options」ダイアログ・ボックスが表示されます。
2. 「Client Access Control」領域で、以下の項目のいずれかを選択します。
 - 「Authorized Users List」
 - 「Kerberos」
 - 「Magic Cookie」
3. [了解]をクリックし、変更を適用して「Security Options」ダイアログ・ボックスを閉じます。

以降、現在のユーザによってこのシステムから実行されるすべてのクライアント・アプリケーションでは、ローカルの X サーバに接続する際にこのアクセス制御方式が適用されます。

注意

クライアント・アクセス制御設定に対する変更は、現在のユーザのデフォルトの X authority ファイルのエントリ (ローカルおよび DECnet) の内容だけに影響を与えます。システム上の他のアクセス制御設定には影響しません。

4.7.2 セッションの最中にセキュリティ・オプションを更新する

場合によっては、現在のセッションに対して別の X authority ファイルを指定したいことがあります。しかし、セッションの最中にセキュリティ・オプションを変更すると、クライアント・アプリケーションが以降 X サーバにアクセスできなくなることがあります。この状況は、DECwindows Motif セッションの実行中に以下の順序で作業を行うと発生します。

1. クライアント・アクセス制御の方法をトークン・ベースのアクセス制御に変更またはリセットする。
2. セッションで使用している現在のディスプレイ・デバイスに対して、別の X authority ファイルを指定する。

別の X authority ファイルを指定すると、セッション中にアクセスを許可するために使用した元の設定が適用されなくなり、またクライアントは新しい設定を使用することはできません。

セキュリティ・オプションを更新してクライアントとサーバの許可エントリを同期させるには、以下の手順を実行します。

1. セッション・マネージャ (Traditional DECwindows Desktop の場合) またはスタイル・マネージャ (New Desktop の場合) の[オプション]メニューから[セキュリティ...]を選択します。
2. 「Security Options」ダイアログ・ボックスで、以下のどちらかを実行します。
 - Magic Cookie によるアクセス制御の場合は、[Create Cookie]をクリックします。
 - Kerberos によるアクセス制御の場合は、「Client Access Control」領域の「Kerberos」オプションの選択を解除し、[適用]をクリックします。次にオプションを再度選択して、[適用]をクリックします。

どちらの操作でも、新しい X authority エントリがサーバと代替 X authority ファイルの両方に作成されます。

セッション・マネージャやスタイル・マネージャにアクセスできない場合は、DECwindows Motif セッションを終了して再起動します。現在のセッションを終了することで、サーバがデフォルトの状態に復元されます。

4.7.3 ログイン時のアクセス制御を有効または無効にする

DECwindows Motif では、デフォルトではアクセス制御は有効になりません。代わりに、スタートアップ時にサーバによって設定されるアクセス制御を使用します。

DECwindows のセッション・マネージャに対して、ログイン時にアクセス制御を明示的に有効または無効にするよう指示するには、次の論理名のどちらかを定義します。

```
$ DEFINE/SYSTEM/EXECUTIVE DECW$LOGIN_ACCESS_CONTROL ENABLE  
$ DEFINE/SYSTEM/EXECUTIVE DECW$LOGIN_ACCESS_CONTROL DISABLE
```

論理名が定義されていない場合や、「SERVER」など他の値が定義されている場合は、DECwindows のログインではアクセス制御は有効にも無効にもされません。

ほとんどの場合、論理名を定義する必要はありません。

4.7.4 信頼ユーザによる一時停止状態のデスクトップ・セッションのアンロックを有効にする

DECwindows Motif ユーザに対し、Screen Lock 機能を使用して一時停止状態にした DECwindows Motif セッションのアンロックを許可することができます。

信頼ユーザを指定するには、次のようにシステム論理名 DECW\$TRUSTED_UNPAUSE を定義します。ここで、*username* は OpenVMS ユーザを表します。

```
$ DEFINE/SYSTEM DECW$TRUSTED_UNPAUSE "username"
```

画面のアンロックが正常に機能するように、セッション・ユーザと指定した信頼ユーザは、同じレベルのパスワード・アクセスを共有する必要がある点に注意してください。

4.8 プリント・フォーマットのカスタマイズ

ここでは、DECwindows Motif ソフトウェアのプリント・フォーマットを定義し、論理名および関連するプリント・フォーマットの一覧を表示する方法について説明します。構成によっては、DECwindows Motif のカスタマイズ作業の中で、プリント・フォーマットを指定するための論理名を定義します。

システムを起動する前に、OpenVMS スタートアップ・プロシージャを編集して、プリント・キューとプリント・フォーマットを関連付ける論理名を定義します。このスタートアップ・プロシージャは、その後 DECwindows Motif のスタートアップ・プロシージャを呼び出します。また、これらの論理名は、ログイン・コマンド・ファイルに追加することもできます。

4.8.1 プリント・フォーマットの定義

DECwindows Motif の多くのアプリケーションは、「Print」ダイアログ・ボックスを使用してファイルや画面をプリンタのキューに登録します。デフォルトでは、システム上のすべてのプリント・デバイスが「プリンタ」リスト・ボックスに表示されます。しかし、論理名を定義することで、プリント・キューにプリント・フォーマットを関連付けることができます。論理名は、OpenVMS のスタートアップ・コマンド・プロシージャ内で、DECwindows Motif のスタートアップ・プロシージャを呼び出す前に定義します。プリント・キューのリストはコンマまたはスペースで区切り、最初のキューがデフォルトの選択肢になります。

「Print」ダイアログ・ボックスに表示されるプリンタ・キューの一覧をカスタマイズするには、表 4-8 に示す論理名のいずれかを定義します。この方法は、システム上のすべてのキューの名前を「Print」ダイアログ・ボックスに表示するよりも高速です。これは、キューのほとんどが、想定しているプリント・フォーマットに該当しな

いたためです。次の例は、OpenVMS のスタートアップ・ファイル内でプリント・フォーマットの論理名を定義する方法を示します。

```
$ Define DECW$PRINTER_FORMAT_TEXT "CLUSTER_LN03,CLUSTER_PRINT, -
_$ ANSI_ARTWRK,ANSI_PROTON"
$ Define DECW$PRINTER_FORMAT_LINE "CLUSTER_PRINT"
$ Define DECW$PRINTER_FORMAT_ANSI2 "CLUSTER_LN03,ANSI_ARTWRK,ANSI2_PROTON"
$ Define DECW$PRINTER_FORMAT_ANSI "CLUSTER_LN03,ANSI_ARTWRK,ANSI_PROTON"
$ Define DECW$PRINTER_FORMAT_PS "PS_ARTWRK,PS_PROTON"
$ Define DECW$PRINTER_FORMAT_REGIS "SYS$NULL"
```

たとえば、DECW\$PRINTER_FORMAT_ANSI2 が CLUSTER_LN03、ANSI_ARTWRK、ANSI2_PROTON として定義されている場合、「プリント・フォーマット」リスト・ボックスから「ANSI2」を選択すると、CLUSTER_LN03、ANSI_ARTWRK、および ANSI2_PROTON だけが「プリンタ」リスト・ボックスに表示され、CLUSTER_LN03 がデフォルトの選択肢となります。

4.8.2 論理名とプリント・フォーマット

表 4-8 に、DECwindows Motif 製品の論理名とそれに関連付けられているプリント・フォーマットの一覧を示します。

表 4-8 「Print」ダイアログ・ボックスの論理名

論理名	プリント・フォーマット
DECW\$PRINTER_FORMAT_DEFAULT	Default
DECW\$PRINTER_FORMAT_TEXT	Text
DECW\$PRINTER_FORMAT_LINE	Line Printer
DECW\$PRINTER_FORMAT_TERM	Terminal
DECW\$PRINTER_FORMAT_ANSI2	ANSI2
DECW\$PRINTER_FORMAT_ANSI	ANSI
DECW\$PRINTER_FORMAT_PS	PostScript
DECW\$PRINTER_FORMAT_REGIS	ReGIS
DECW\$PRINTER_FORMAT_TEK	TEKTRONIX
DECW\$PRINTER_FORMAT_DDIF	DDIF

システム全体の論理名テーブルで論理名を定義し、ユーザが必要に応じて論理名を設定変更することができます。

DECwindows システムのチューニング

この付録では、DECwindows Motif システムの制限値とシステム・パラメータの推奨値の一覧を示します。

A.1 DECwindows アプリケーション向けの UAF パラメータの設定

DECwindows アプリケーションは、利用者登録ファイル (UAF) のリミットに敏感に反応することがあります。セッションの起動時やアプリケーションの起動時に問題が発生したり、エラー・メッセージを表示せずにアプリケーションが消えた場合は、セッションを開始した際のアカウントの UAF リミットを確認してください。

表 A-1 に、推奨される UAF リミットを示します。具体的な数値は、ガイドラインとして使用することを意図したものです。適切な UAF パラメータは、各自のアプリケーションとプロセスに依存します。詳細は、『HP OpenVMS システム管理者マニュアル』を参照してください。

表 A-1 推奨される UAF リミットの設定値

リミット	設定	使用目的
ASTLM	100	非同期システム・トラップ (AST) キュー・リミット
BIOLM	100	バッファード入出力回数リミット
BYTLM	100000	入出力バイト・リミット
DIOLM	100	直接入出力回数リミット
ENQLM	300	同時発行ロック・リミット
FILLM	100	オープン・ファイル・リミット
MAXDETACH	0	単一のユーザ名に対する最大独立プロセス数 (0 = 無制限)
MAXJOBS	0	単一のユーザ名に対する最大アクティブ・プロセス数 (0 = 無制限)
MAXACCTJOB	0	単一のアカウントに対する最大アクティブ・プロセス数 (0 = 無制限)
PGFLQUOTA	70000	ページング・ファイル・リミット
PRCLM	10	サブプロセス作成リミット
TQELM	10	タイマ・キュー・エントリ・リミット
WSDEFAULT	1024	デフォルトのワーキング・セット・サイズ

(次ページに続く)

表 A-1 (続き) 推奨される UAF リミットの設定値

リミット	設定	使用目的
WSEXTENT	8192	ワーキング・セット超過値
WSQUOTA	2048	ワーキング・セット制限

A.2 DECwindows アプリケーションおよびディスプレイ・サーバ用のシステム・パラメータの設定

ほとんどの DECwindows プロセス，特に DECterm ウィンドウは独立モードで動作するため，特定のシステム・パラメータの値を増やす必要があります。表 A-2 および表 A-3 に，そのようなパラメータとその最小設定値，加算値，Alpha システムと I64 システムでの使用方法を示します。加算値は，DECwindows Motif がリソースを利用できるように AUTOGEN がパラメータを増やす際の増分値です。

システム・パラメータの変更についての詳細は，『HP OpenVMS システム管理ユーティリティ・リファレンス・マニュアル』を参照してください。

表 A-2 に，OpenVMS Alpha システムでのパラメータ，最小設定値，加算値，使用目的を示します。

表 A-2 Alpha システムでのシステム・パラメータの推奨される設定

パラメータ	最小値	加算値	説明
GBLSECTIONS	600	280	システム・グローバル・セクション
GBLPAGES	150000	92000	システム・グローバル・ページ・テーブル・エントリ
GBLPAGFIL	4096	768	システム・グローバル・ページ・ファイル・セクション・リミット
NPAGEDYN	1348576	300000	システム非ページング動的プール
PAGEDYN	704288	180000	システム・ページング動的プール
SWPOUTPGCNT	512	-	スワッピング前の最小プロセス・ページ・サイズ
MAXBUF	8192	-	最大バッファ・サイズ
CHANNELCNT	255	-	システム永久入出力チャネル・リミット
PROCSECTCNT	128	-	プロセス・イメージ・セクション記述子リミット

(次ページに続く)

表 A-2 (続き) Alpha システムでのシステム・パラメータの推奨される設定

パラメータ	最小値	加算値	説明
WSMAX	12000	-	プロセス・ワーキング・セットの最大
CLISYMTBL	512	-	コマンド・インタプリタのシンボル・テーブルの最小サイズ
PQL_MPGFLQUOTA	32768	-	最小ページ・ファイル制限
PQL_MASTLM	100	-	最小 AST リミット
PQL_MBIOLM	100	-	最小バッファード入出力リミット
PQL_MDIOLM	100	-	最小直接入出力リミット
PQL_MFILLM	100	-	最小オープン・ファイル・リミット
PQL_MBYTLM	100000	-	最小バッファード入出力バイト・リミット
PQL_MPRCLM	10	-	最小サブプロセス・リミット
PQL_MENQLM	300	-	最小同時発行ロック・リミット
PQL_MWSDEFAULT	1024	-	最小ワーキング・セット・デフォルト
PQL_MWSQUOTA	2048	-	最小ワーキング・セット制限
PQL_MWSEXTENT	8192	-	最小ワーキング・セット超過値
GH_RES_CODE	1584	560	常駐イメージ・コード粒度ヒント領域リミット
IMGREG_PAGES	10000	3160	共用可能アドレス・データを持つイメージをインストールするための最小予約ページ数

表 A-3 に、OpenVMS I64 システムでのパラメータ、最小設定値、加算値、使用目的を示します。

表 A-3 I64 システムでのシステム・パラメータの推奨される設定

パラメータ	最小値	加算値	説明
GBLSECTIONS	1000	400	システム・グローバル・セクション
GBLPAGES	150000	92000	システム・グローバル・ページ・テーブル・エントリ
GBLPAGFIL	4096	768	システム・グローバル・ページ・ファイル・セクション・リミット
NPAGEDYN	4194304	300000	システム非ページング動的プール
PAGEDYN	4194304	180000	システム・ページング動的プール

(次ページに続く)

表 A-3 (続き) 164 システムでのシステム・パラメータの推奨される設定

パラメータ	最小値	加算値	説明
SWPOUTPGCNT	512	-	スワッピング前の最小プロセス・ページ・サイズ
MAXBUF	8192	-	最大バッファ・サイズ
CHANNELCNT	255	-	システム永久入出力チャンネル・リミット
PROCSECTCNT	64	-	プロセス・イメージ・セクション記述子リミット
WSMAX	131072	-	プロセス・ワーキング・セットの最大
CLISYMTBL	512	-	コマンド・インタプリタのシンボル・テーブルの最小サイズ
PQL_MPGFLQUOTA	32768	-	最小ページ・ファイル制限
PQL_MASTLM	100	-	最小 AST リミット
PQL_MBIOLM	100	-	最小バッファード入出力リミット
PQL_MDIOLM	100	-	最小直接入出力リミット
PQL_MFILLM	100	-	最小オープン・ファイル・リミット
PQL_MBYTLM	100000	-	最小バッファード入出力バイト・リミット
PQL_MPRCLM	10	-	最小サブプロセス・リミット
PQL_MENQLM	300	-	最小同時発行ロック・リミット
PQL_MWSDEFAULT	4096	-	最小ワーキング・セット・デフォルト
PQL_MWSQUOTA	8192	-	最小ワーキング・セット制限
PQL_MWSEXTENT	16384	-	最小ワーキング・セット超過値
GH_RES_CODE	3072	1450	常駐イメージ・コード粒度ヒント領域リミット
IMGREG_PAGES	10000	-	共用可能アドレス・データを持つイメージをインストールするための最小予約ページ数

A.3 非VGA デバイス向けのサーバ・パラメータの設定

グラフィックスを頻繁に使用するアプリケーションや3Dアプリケーションでは、システム・リソースへの要求が大きいため、DECwindowsのXディスプレイ・サーバを個別にチューニングする必要があります。3Dアクセラレータが搭載されたシステムでは、サーバ制限値の調整が必要な場合があります。これは、物理メモリが64MBしか搭載されていないシステムで複雑なクライアントを動作させる場合の推奨される最小値です。

表 A-4 に、非 VGA デバイス向けに推奨される最小制限値を示します。これらのサーバ制限値を設定するには、DECW\$PRIVATE_SERVER_SETUP.COM ファイルでグローバル・シンボルを定義します。このファイルが存在しない場合は、第 3.1 節で説明しているように、ファイル SYSS\$MANAGER:DECW\$PRIVATE_SERVER_SETUP.TEMPLATE を SYSS\$COMMON:[SYSMGR]DECW\$PRIVATE_SERVER_SETUP.COM にコピーします。

表 A-4 非 VGA デバイス向けの最小サーバ制限値

制限	値
DECW\$SERVER_FILE_LIMIT	400
DECW\$SERVER_ENQUEUE_LIMIT	2000
DECW\$SERVER_WSDEF	10240
DECW\$SERVER_WSQUOTA	16384

アニメーション・アプリケーション向けのチューニング

大規模なモデルや組み立て部品のアニメーションを長時間にわたって行うアプリケーションでは、以下のサーバ・ワーキング・セット制限と値を設定することで、性能が向上する場合があります。

制限	値
DECW\$SERVER_WSDEF	10240
DECW\$SERVER_WSQUOTA	20480

注意

これらのパラメータは、128 MB 以上の物理メモリがないかぎり、これ以上増やさないでください。

メモリを大量に使用するアプリケーション向けのチューニング

メモリを大量に使用するアプリケーション (HP Secure Web Browser など) を、かなりの量の物理メモリ (512 MB 以上) が搭載されたシステム上で動作させる場合は、以下のサーバ制限と値を設定することで、性能が向上することがあります。

制限	値
DECW\$SERVER_FILE_LIMIT	6000
DECW\$SERVER_WSDEF	32768
DECW\$SERVER_WSQUOTA	65536
DECW\$SERVER_PAGE_FILE_QUOTA	2000000

チューニングの必要性を判断する

パラメータを増やす必要があるかどうかを判断するには、ディスプレイの負荷が最も高いときにサーバ・プロセスを監視します。ワーキング・セットの使用量が最大値に近い場合は、サーバ・パラメータ DECW\$SERVER_WS* の値を調整する必要があります。

DECwindows システムのチューニング

A.3 非 VGA デバイス向けのサーバ・パラメータの設定

ただし、必要な場合以外は大きな値を設定しないでください。これらの値を大きくしすぎると、性能が低下するおそれがあります。DECwindows サーバの最善の性能は、アプリケーションが必要とするメモリ量と使用可能な物理メモリによって変わります。

次回サーバを再起動すると、新しい値が有効になります。最初にチューニングを行ってある程度使用した後で、サーバで障害が発生するか、必要以上に応答が悪い場合は、メモリを使い果たしているか、メモリが断片化している可能性があります。特に要求の激しいアプリケーションでは、サーバの DECW\$SERVER_PAGE_FILE_QUOTA の値をより大きくする必要があります。サーバのページ・ファイル制限は、システム・ページ・ファイルから決定されます。

サーバのエラー・ログ SYS\$MANAGER:DECW\$SERVER_0_ERROR.LOG に、xxx: Out of memory という文が含まれている場合は、サーバのページ・ファイル制限を増やします。この制限値を設定するには、ファイル DECW\$PRIVATE_SERVER_SETUP.COM 内の DECW\$SERVER_PAGE_FILE_QUOTA パラメータを変更します。マルチヘッド構成では、システム要件を満たすために、DECW\$SERVER_PAGE_FILE_QUOTA パラメータを増やす必要がある点に注意してください。

B

DECwindows Motif キーマップ名

表 B-1 に、DECwindows Motif のキーマップ名を示します。表の内容は、各キーボードの言語の順に並んでいます。

デフォルトのキーボード・レイアウトを変更する方法については、第 3.5 節を参照してください。

表 B-1 DECwindows Motif キーマップ

言語	モデル	DECwindows キーマップ名
オーストリア/ドイツ語	LK201-(AG,LG,BG,MG)	AUSTRIAN_GERMAN_LK201LG_DP AUSTRIAN_GERMAN_LK201LG_TW
	LK201-(NG,PG)	AUSTRIAN_GERMAN_LK201NG_DP AUSTRIAN_GERMAN_LK201NG_TW
	LK401-(AG)	AUSTRIAN_GERMAN_LK401AG_TW
	LK444-(AG)	AUSTRIAN_GERMAN_LK444AG_LK AUSTRIAN_GERMAN_LK444AG_PC
ベルギー/フランス語	LK201-(AP,LP,BP,MP)	BELGIAN_FRENCH_LK201LP_DP BELGIAN_FRENCH_LK201LP_TW
	LK401-(AP)	BELGIAN_FRENCH_LK401AP_DP BELGIAN_FRENCH_LK401AP_TW
	LK444-(AP)	BELGIAN_FRENCH_LK444AP_LK BELGIAN_FRENCH_LK444AP_PC
英語	LK201-(AE,LE,BE,ME)	BRITISH_LK201LE_DP BRITISH_LK201LE_TW
	LK401-(AA,PA)	BRITISH_LK401AA_DP BRITISH_LK401AA_TW
	LK444-(AE)	BRITISH_LK444AE_LK BRITISH_LK444AE_PC
カナダ/フランス語	LK201-(AC,LC,BC,MC)	CANADIAN_FRENCH_LK201LC_DP CANADIAN_FRENCH_LK201LC_TW
	LK401-(AC,LC,BC,MC)	CANADIAN_FRENCH_LK401AC_DP CANADIAN_FRENCH_LK401AC_TW
カナダ/フランス語	LK444-(AC,LC,BC,MC)	CANADIAN_FRENCH_LK444AC_LK CANADIAN_FRENCH_LK444AC_PC

(次ページに続く)

表 B-1 (続き) DECwindows Motif キーマップ

言語	モデル	DECwindows キーマップ名
チェコ語	LK401-(BV)	CZECH_LK401_BV
デンマーク語	LK201-(AD,LD,BD,MD)	DANISH_LK201LD_DP DANISH_LK201LD_TW
	LK201-(ED,RD,FD)	DANISH_LK201RD_DP DANISH_LK201RD_TW
	LK401-(AD,LD,BD,MD)	DANISH_LK401AD_DP DANISH_LK401AD_TW
	LK444-(AD,LD,BD,MD)	DANISH_LK401AD_LK DANISH_LK401AD_PC
オランダ語	LK201-(AH,LH,BH,MH)	DUTCH_LK201LH_DP DUTCH_LK201LH_TW
	LK201-(NH,PH)	DUTCH_LK201NH
	LK401-(NH,PH)	DUTCH_LK401AH
	LK444-(NH,PH)	DUTCH_LK444AH_LK DUTCH_LK444AH_PC
フィンランド語	LK201-(AF,LF,BF,MF)	FINNISH_LK201LF_DP FINNISH_LK201LF_TW
	LK201-(NX,PX)	FINNISH_LK201NX_DP FINNISH_LK201NX_TW
	LK401-(AF,LF,BF,MF)	FINNISH_LK401AF_DP FINNISH_LK401AF_TW
	LK444-(CA)	FINNISH_LK401CA_LK FINNISH_LK401CA_PC
フラマン語	LK201-(AB,LB,BB,MB)	FLEMISH_LK201LB_DP FLEMISH_LK201LB_TW
	LK401-(AB,LB,BB,MB)	FLEMISH_LK401AB_DP FLEMISH_LK401AB_TW
	LK444-(AB,LB,BB,MB)	FLEMISH_LK401AB_LK FLEMISH_LK401AB_PC
ハンガリー語	LK401-(BQ)	HUNGARIAN_LK401_BQ
アイスランド語	LK201-(AU,LU,BU,MU)	ICELANDIC_LK201LU_DP ICELANDIC_LK201LU_TW
イタリア語	LK201-(AI,LI,BI,MI)	ITALIAN_LK201LI_DP ITALIAN_LK201LI_TW
	LK401-(AI,LI,BI,MI)	ITALIAN_LK401AI_DP
イタリア語	LK444-(AI,LI,BI,MI)	ITALIAN_LK401AI_TW ITALIAN_LK444AI_LK

(次ページに続く)

表 B-1 (続き) DECwindows Motif キーマップ

言語	モデル	DECwindows キーマップ名
北米	LK201-(AA,LA,BA,MA)	ITALIAN_LK444AI_PC NORTH_AMERICAN_LK201LA
	LK401-(AA,LA,BA,MA)	NORTH_AMERICAN_LK401AA
	LK443-(AA,LA,BA,MA)	NORTH_AMERICAN_LK443AA_LK NORTH_AMERICAN_LK443AA_PC
ノルウェー語	LK201-(AN,LN,BN,MN)	NORWEGIAN_LK201LN_DP NORWEGIAN_LK201LN_TW
	LK201-(EN,RN,FN)	NORWEGIAN_LK201RN_DP NORWEGIAN_LK201RN_TW
	LK401-(AN,LN,BN,MN)	NORWEGIAN_LK401AN_DP NORWEGIAN_LK401AN_TW
	LK444-(AN,LN,BN,MN)	NORWEGIAN_LK444AN_LK NORWEGIAN_LK444AN_PC
ポーランド語	LK401-(BP)	POLISH_LK401_BP
ポルトガル語	LK201-(AV,LV,BV,MV)	PORTUGUESE_LK201LV
	LK401-(AV,LV,BV,MV)	PORTUGUESE_LK401AV
	LK444-(AV,LV,BV,MV)	PORTUGUESE_LK444AV_LK PORTUGUESE_LK444AV_PC
ロシア語	LK401-(BT)	RUSSIAN_LK401_BT
スロバキア語	LK401-(CZ)	SLOVAC_LK401_CZ
スペイン語	LK201-(AS,LS,BS,MS)	SPANISH_LK201LS_DP SPANISH_LK201LS_TW
	LK401-(AS,LS,BS,MS)	SPANISH_LK401AS_DP SPANISH_LK401AS_TW
	LK444-(AS,LS,BS,MS)	SPANISH_LK444AS_LK SPANISH_LK444AS_PC
	スウェーデン語	LK201-(AM,LM,BM,MM)
LK201-(NM,PM)		SWEDISH_LK201NM_DP SWEDISH_LK201NM_TW
LK401-(AM,LM,BM,MM)		SWEDISH_LK401AM_DP SWEDISH_LK401AM_TW
LK444-(CA)		SWEDISH_LK444CA_LK SWEDISH_LK444CA_PC
スイス/フランス語	LK201-(AK,LK,BK,MK)	SWISS_FRENCH_LK201LK_DP SWISS_FRENCH_LK201LK_TW
スイス/フランス語	LK401-(AK,LK,BK,MK)	SWISS_FRENCH_LK401AK_DP

(次ページに続く)

DECwindows Motif キーマップ名

表 B-1 (続き) DECwindows Motif キーマップ

言語	モデル	DECwindows キーマップ名
		SWISS_FRENCH_LK401AK_TW
	LK444-(AK,LK,BK,MK)	SWISS_FRENCH_LK444AK_LK
		SWISS_FRENCH_LK444AK_PC
	LK201-(AL,LL,BL,ML)	SWISS_GERMAN_LK201LL_DP
		SWISS_GERMAN_LK201LL_TW
	LK401-(AL,LL,BL,ML)	SWISS_GERMAN_LK401AL_DP
		SWISS_GERMAN_LK401AL_TW
	LK444-(CH)	SWISS_GERMAN_LK401CH_LK
		SWISS_GERMAN_LK401CH_PC
US/UK 共用	LK201-(EE,RE,PE)	UK_LK201RE
		US_LK201RE
	LK401-(EE,RE,PE)	UK_LK401AA
		US_LK401AA
	LK443-(EE,RE,PE)	UK_LK443AA_LK
		US_LK443AA_PC

A

AccessX 機能	
有効にする	3-64
AUTOGEN コーティリティ	2-7

B

BDF (Bitmap Distribution Format)	3-65
Bookreader ディレクトリ	4-13
変更	4-27

C

CDESDTFILE_UNIX_NAMES 論理名	4-13
CDE\$NOGENPWD 論理名	4-12
Console Window	
起動	4-20
グローバル・シンボルの定義	4-20
初期位置の制御	4-20

D

DECC\$FILENAME_UNIX_ONLY 論理名	4-12
DECWSBOOK シンボル	4-27
DECWSCLIENT_ERROR_THRESHOLD パラメータ	3-32
DECWS\$CONSOLE_GEOMETRY	
シンボル	4-20
DECWS\$CONSOLE_SELECTION	
シンボル	4-19
DECWSCURSOR_SIZE パラメータ	3-20
DECWSDEFAULT_KEYBOARD_MAP パラメータ	3-23
DECWSDEFAULT_VISUAL_CLASS パラメータ	3-13
DECWS\$DEVICE.COM コマンド・プロシージャ	2-3
DECWS\$DEVICE_XX.COM コマンド・プロシージャ	3-13
DECWS\$DEVICE パラメータ	3-13
DECWS\$DISPLAY 論理名	
名前の形式	4-2
DECWS\$FONT_SERVERS パラメータ	3-21, 3-67
DECWSINSTALL_IMAGES.COM コマンド・プロシージャ	2-3

DECWSIPV6_FONT_SUPPORT パラメータ	3-21
DECWSIPV6_SUPPORT シンボル	4-17
DECWSKEYMAP パラメータ	3-23
DECWSMONITOR_DENSITY パラメータ	3-14
DECWSMULTI_HEAD パラメータ	3-13
DECWSPRIMARY_DEVICE パラメータ	3-13
DECWSPRIVATE_APPS_SETUP.COM コマンド・プロシージャ	2-6
DECWSPRIVATE_APPS_SETUP.TEMPLATE	4-16
DECWSPRIVATE_SERVER_SETUP.COM コマンド・プロシージャ	2-5
DECWSPRIVATE_SERVER_SETUP.TEMPLATE	3-2
DECWS\$PROXY_MANAGER_CONFIG シンボル	3-74
DECWS\$PROXY_MANAGER_LOG シンボル	3-74
DECWS\$PROXY_MANAGER_OPTIONS シンボル	3-74
DECWS\$PROXY_MANAGER_QUOTAS シンボル	3-74
DECWS\$RGBPATH パラメータ	3-27
DECWS\$SECURITY_POLICY パラメータ	3-10
DECWS\$SERVER_ACCESS_ALLOWED パラメータ	3-12, 3-49
DECWS\$SERVER_ACCESS_TRUSTED パラメータ	3-11, 3-49
DECWS\$SERVER_AUDIT_LEVEL パラメータ	3-31
DECWS\$SERVER_BELL_BASE_VOLUME パラメータ	3-24
DECWS\$SERVER_BUG_COMPATIBILITY パラメータ	3-31
DECWS\$SERVER_DEFAULT_BACKING_STORE	3-29
DECWS\$SERVER_DENSITY パラメータ	3-20
DECWS\$SERVER_DISABLE_BACKING_STORE	3-29
DECWS\$SERVER_DISABLE_SAVE_UNDER	3-30
DECWS\$SERVER_DISABLE_TEST	3-10
DECWS\$SERVER_DISABLESCREEN	3-16

DECWSSERVER_DUMP パラメータ 3-31
 DECWSSERVER_EDGE_BOTTOM パラメータ 3-18, 3-59
 DECWSSERVER_EDGE_LEFT パラメータ 3-16, 3-59
 DECWSSERVER_EDGE_RIGHT パラメータ 3-17, 3-59
 DECWSSERVER_EDGE_TOP パラメータ 3-17, 3-59
 DECWSSERVER_ENABLE_ACCESSX パラメータ 3-24
 DECWSSERVER_ENABLE_KB_AUTOREPEAT パラメータ 3-24
 DECWSSERVER_ENABLESCREEN パラメータ 3-15
 DECWSSERVER_ENQUEUE_LIMIT パラメータ 3-7
 DECWSSERVER_ERROR_THRESHOLD パラメータ 3-32
 DECWSSERVER_EXTENSIONS パラメータ 3-8
 DECWSSERVER_FILE_LIMIT パラメータ 3-7
 DECWSSERVER_FONT_CACHE_SIZE パラメータ 3-22
 DECWSSERVER_KEY_REPEAT_DELAY パラメータ 3-25
 DECWSSERVER_KEY_REPEAT_INTERVAL パラメータ 3-25
 DECWSSERVER_KEYCLICK_VOLUME パラメータ 3-24
 DECWSSERVER_MOUSE_ACCELERATION パラメータ 3-26
 DECWSSERVER_MOUSE_THRESHOLD パラメータ 3-27
 DECWSSERVER_ONLYSCREEN パラメータ 3-16
 DECWSSERVER_PAGE_FILE パラメータ 3-7
 DECWSSERVER_PRIORITY パラメータ 3-5
 DECWSSERVER_SCALE_BITMAP_FONTS パラメータ 3-23
 DECWSSERVER_SCREEN_SAVER_INTERVAL パラメータ 3-28
 DECWSSERVER_SCREEN_SAVER_PREFER_BLANKING パラメータ 3-28
 DECWSSERVER_SCREEN_SAVER_TIMEOUT パラメータ 3-28
 DECWSSERVER_SCREENSAVERS パラメータ 3-15, 3-58
 DECWSSERVER_TRANSPORTS パラメータ 3-18
 DECWSSERVER_WSDEF パラメータ 3-6
 DECWSSERVER_WSEXTENT パラメータ 3-6
 DECWSSERVER_WSQUOTA パラメータ 3-6
 DECWSSERVER_XAUTHORITY パラメータ 3-11, 3-51
 DECWSSERVER_XKEYBOARD_COMPILED_DIR パラメータ 3-25, 3-64

DECWSSERVER_XKEYBOARD_DIRECTORY パラメータ 3-25
 DECWSSERVER_XKEYBOARD_LOAD_MAP パラメータ 3-26, 3-64
 DECWSSERVER_XKEYBOARD_MAP パラメータ 3-26, 3-64
 DECW\$SETPARAMS.COM コマンド・プロシージャ 2-3
 DECW\$STARTAPPS.COM コマンド・プロシージャ 2-6
 DECW\$STARTI18N.COM コマンド・プロシージャ 2-4
 DECW\$STARTLIBS.COM コマンド・プロシージャ 2-5
 DECW\$STARTSERVER.COM コマンド・プロシージャ 2-4
 DECW\$STARTUP.COM コマンド・プロシージャ 2-6
 DECW\$STARTUP_n.LOG ファイル 2-4
 DECW\$STARTXTERMINAL.COM コマンド・プロシージャ 2-4
 DECWSXM_FORCE_UNIX_NAMES_TO_VMS 論理名 4-12
 DECWSXPORT_SYNC_TIMEOUT パラメータ 3-19
 DECWSLOGICALS.COM コマンド・プロシージャ 2-2
 DECWSLOGIN.DAT ファイル 4-9
 DECwindows システム
 システムを参照
 DECwindows の起動
 SYLOGIN.COM 4-6, 4-8
 セキュリティ・リスト 4-6, 4-7
 ライセンスの確認 4-5, 4-7
 DECwindows の再起動 2-4
 DECwindows ロゴの復元 4-18

F

FileView
 ディレクトリ 4-13
 FS プロトコル 1-4

I

IPv4
 ホストとアドレスの形式 4-4
 IPv6
 LBX プロキシ・サーバでのサポート 3-71
 サポートされているトランスポート名 4-2
 サポートを有効にする 4-17
 認証への影響 3-52
 ホストとアドレスの形式 4-4
 ホスト名の変換 4-5

K

Kerberos によるアクセス制御	
IPv6 でのサポート	3-52
アクセスの無効化	3-55
概要	3-44
クライアントに対する指定	4-29
無効にする	3-54
有効にする	3-52

L

LAT トランスポート	3-41
LBX (Low-Bandwidth X)	
概要	1-5
構成要素	1-6, 3-68
認証	3-69
プロキシ・サーバ	
IPv6 でのサポート	3-71
管理対象プロキシ・サーバの起動	3-70
自動停止	3-72
手動停止	3-72
スタンドアロン・プロキシ・サーバの起 動	3-70
タイプ	3-68
プロセス属性の変更	3-71
プロセス論理名	3-71
プロキシ・マネージャ	
グローバル・シンボル	3-74
構成ファイル	3-72
自動起動	3-74
手動起動	3-75
LBXPROXY	
管理対象プロキシ・サーバの起動	3-70
構成タイプ	3-68
自動停止	3-72
手動停止	3-72
スタンドアロン・プロキシ・サーバの起 動	3-70
プロセス属性の変更	3-71
プロセス論理名	3-71
LBX プロトコル	1-5

M

Magic Cookie によるアクセス制御	
概要	3-43
クライアントに対する指定	4-29
無効にする	3-52
有効にする	3-51
MIT-KERBEROS-5 プロトコル	3-44
Kerberos によるアクセス制御も参照	
MIT-MAGIC-COOKIE-1 プロトコル	3-43
Magic Cookie によるアクセス制御も参照	

P

PCF (Portable compiled format)	3-22
「Print」ダイアログ・ボックス 論理名	4-33

R

RGB データベース	3-27
------------	------

S

「Set Password」ダイアログ・ボックス 位置の変更	4-10
「Start Session」ダイアログ・ボックス 位置の変更	4-10
SYSTARTUP_VMS.COM	2-7

T

TCP/IP トランスポート	3-41
----------------	------

U

UAF パラメータ	
推奨する設定	A-1
UNIX との互換性	4-12, 4-15

W

WSA デバイス	2-6
----------	-----

X

X authority ファイル	3-45
X Display Information ユーティリティ (xdpyinfo)	3-37
Xinerama 拡張 (XINERAMA)	
有効にする	3-58
X Keyboard Compiler ユーティリティ (xkbcomp)	
コンポーネント・データベース	3-61
X Keyboard 拡張 (XKB)	3-23
XPROXYMANAGER	
グローバル・シンボル	3-74
構成ファイル	3-72
自動起動	3-74
手動起動	3-75
X 設定ユーティリティ (xset)	3-39
X ディスプレイ・サーバ	
Magic Cookie によるアクセス制御	3-43
SECURITY 拡張の利用	3-55
X Keyboard キーマップ・ファイルの利 用	3-60
エラーしきい値	3-32
エラー条件ハンドラ	3-31
エラー報告	3-30
拡張	3-7
カスタマイズ	3-2

X ディスプレイ・サーバ (続き)	
カスタマイズ・パラメータ	3-2
起動のカスタマイズ	3-34
情報の表示	
論理名テーブル	3-35
セキュリティ	3-10
セキュリティ・ポリシー・ファイルの利 用	3-56
セッション外にアクセス制御を指定す る	3-48
セッション内にアクセス制御を指定す る	3-48
セーブ・アンドの利用	3-29
特殊な要件	3-34
トークン・ベースのアクセス制御	3-43
トランスポート・タイムアウトの定義	3-19
バッキング・ストアの使用	3-28
プロセス優先順位の設定	3-5
密度値	3-20
ユーザ・ベースのアクセス制御	3-42
要求の処理	1-4
X プロトコル	1-3

ア

アカウント	
専用	4-21
アーキテクチャ	
ディスプレイ・サーバ	1-2
フォント・サーバ	1-4
プロキシ・サーバ	1-6
アクセス許可ファイル	3-46
アクセス信頼ファイル	3-47
アクセス制御	
Kerberos	3-44, 3-52
Magic Cookie	3-43, 3-51
SECURITY 拡張の利用	3-55
概要	3-42
クライアント・アプリケーションに対する指 定	4-29
セキュリティ・ポリシー・ファイルの利 用	3-56
セッション外の接続に指定する	3-48
セッション内の接続に指定する	3-48
トークン・ベース	3-43
ユーザ・ベース	3-42, 3-48

イ

イメージ	
共有	1-7
入力ドライバ	1-8
ビデオ・ドライバ	1-8
インストール後の作業	
ソフトウェアのカスタマイズ	4-32
インターネット・プロトコル・バージョン 4 トラン スポート	
IPv4 を参照	

エ

エラーしきい値	3-32
エラー条件ハンドラ	3-30
エラー報告	3-30
エラー・ログ・ファイル	3-30

カ

学習支援システムのファイルがあるディレクト リ	4-14
拡張	
組み込み	3-8
動的にロード可能	3-8
隠れたウィンドウの復元	3-28
カーソル・サイズ	3-20
画面	
幅と高さの定義	3-14
密度	3-20
カラー・データベース	3-27

キ

キーボード	
AccessX キーを有効にする	3-64
X Keyboard のキーマップ・ファイルの作 成	3-63
カスタマイズ	3-59
コンパイル済み X Keyboard キーマップ・ファイ ルのロード	3-64
設定	3-23
デフォルト・レイアウトの変更	3-59
モデル番号	3-59
キーマップ	
設定	3-23
キーマップ・ファイルのディレクトリ	4-14
キーマップ名	
確認	3-59
共用メモリ	1-3

ク

クライアント	1-3
クライアント/サーバ・モデル	1-2
クライアント領域	
変更	4-13
クライアント領域の変更	4-13

コ

構成要素	1-7
オペレーティング・システムで提供	1-7
コマンド・プロシージャ	
スタートアップで使用	2-2
コンソール・メッセージ	
表示	4-19

サ

サーバ	
X ディスプレイ	1-3
共有イメージ	1-7
フォント	1-4
プロキシ・サーバ	1-5
サーバ・パラメータ	
非 VGA デバイス向けの推奨設定	A-4
サーバ・プロセス	
シンボル	3-4
サーバ論理名テーブル	3-35

シ

システム	
AUTOGEN の実行	2-7
概要	1-1
スタートアップで使用される論理名	2-7
スタートアップ・ファイル	2-7
システム・パラメータ	2-3, 2-7
Alpha システムで推奨される設定	A-2
I64 システムで推奨される設定	A-3
情報の表示	
xdpyinfo の利用	3-37
xset ユーティリティの利用	3-39
サーバ設定	3-35
サーバ論理名テーブル	3-35
処理	
スタートアップ	2-2
シンボルの定義	3-2

ス

スクリーン・サーバ属性	3-27
スタートアップ	
コマンド・プロシージャ	2-2
処理	2-2

セ

セキュリティ	
パラメータ	3-10
セキュリティ・オプション	
セッションの最中にクライアント・オプションを 更新する	4-30
セキュリティ拡張 (SECURITY)	
有効にする	3-56
利用	3-55
セキュリティ・ポリシー・ファイル	3-56
セッション・マネージャ	
実行可能ファイル	4-23
動作の変更	4-28
セッション・マネージャの動作の変更	4-28
セーブ・アンド・オプション	3-29

専用アカウント

セッション・マネージャのコマンド・プロシージャによる作成	4-21
セッション・マネージャの実行可能ファイルによる作成	4-23
セッション・マネージャのプロファイル・ファイルによる作成	4-25

タ

他社製のフォント	3-65
----------	------

チ

チューニング	
非 VGA デバイス向けの調整	A-4
大量のメモリの使用のための調整	3-5

テ

提供されているコンポーネント	1-7
ディスプレイ・サーバ	
X ディスプレイ・サーバを参照	
ディスプレイ・デバイス	4-2
サポートされているトランスポート名	4-2
ディスプレイ名の形式	4-2
ディスプレイの設定	4-2
デスクトップ・セッション	
一時停止状態のセッションのアンロック	4-32
データ・ファイル	1-8
デバイス	
WSA	2-6
グラフィックス	3-12
デバイス設定	
シンボル	3-12
デバイス・ドライバ	1-8, 2-3

ト

トランスポート	
DECnet	1-3, 3-41
LAT	1-3, 3-41
TCP/IP	1-3, 3-41
共有イメージ	1-7
シンボルを使った指定	3-18
設定例	3-42
タイムアウトの定義	3-19
ローカル	1-3, 3-40
トランスポート層	1-3

ノ

ノード名表示 無効にする	4-11
-----------------	------

ハ

バッキング・ストア・オプション パラメータ	3-28
UAF	A-1
サーバ	3-2, A-4
システム	A-2
ディスプレイ・サーバのカスタマイズ用	3-2

ヒ

ビデオ デバイス・ドライバ	1-8
ビットマップ・フォントのスケーリング	3-22

フ

ファイル・リミット	3-7
フォント	
BDF (Bitmap Distribution Format)	3-65
PCF (Portable Compiled Format)	3-22
新しいディレクトリの作成	3-65
カーソル・サイズ	3-20
キャッシュ	3-22
グリフ	3-22
サイズの決定	3-20
サーバ	3-21
指定	3-65
新規追加	3-65
他社製	3-65
パスで使用されるディレクトリ	3-19
ビットマップのスケーリング	3-22
ファイル	3-22
ベンダに依存しないサポート	3-22
ユーザ定義	3-20
ユーロ記号のサポート	3-66
フォント・サーバのサポート	1-4
フォントのキャッシュ	3-22
フォントの設定	3-19
フォント・パス	3-19
フォント別名ファイル	3-19
プリント・キュー	4-33
プリント・フォーマット	
カスタマイズ	4-32
定義	4-32
論理名	4-32, 4-33
プロキシ・サーバ	
LBXPROXY を参照	
プロキシ・マネージャ	
XPROXYMGR を参照	

プログラム例のディレクトリ	4-14
プロセス・クォータ	
ENQUEUE_LIMIT パラメータ	3-7
PAGE_FILE パラメータ	3-7
WSDEF パラメータ	3-6
WSEXTENT パラメータ	3-6
WSQUOTA パラメータ	3-6
非 VGA デバイス向けの	A-4
プロファイル・ファイル・ディレクトリ	4-13

マ

マウスの設定	3-26
マルチヘッド・サポート	3-12, 3-13, 3-56
マルチヘッド・システム	
XINERAMA での要件	3-58
マルチヘッドの設定	3-56

ミ

密度	
画面	3-20
モニタ	3-14

メ

メッセージ	
ログイン前に表示する	4-11
メモリ	
仮想	3-7
共用	1-3
セーブ・アンドアの要件	3-29
大量使用のための調整	3-5
定義	3-6
バッキング・ストアの要件	3-29
物理ページ	3-6

モ

モニタ	
スクリーン・セーバによる表示の消去	3-27
マルチヘッドのサポート	3-12, 3-13
密度の定義	3-14

ユ

ユーザ・ベースのアクセス制御	
IPv6 でのサポート	3-48
クライアントに対する指定	4-29
無効にする	3-50
有効にする	3-48
ユーティリティ・ディレクトリ	4-16
ユーロ記号	3-66

リ

リソース・ファイル	
変更	4-28
リソース・ファイルの変更	4-28
利用者登録ファイル (UAF)	
UAF パラメータを参照	

ロ

ログイン画面	
色の変更	4-10
カスタマイズ	4-9
ログイン・プロセス	

New Desktop	4-5
Traditional DECwindows Desktop	4-7
カスタマイズ	4-8

ロゴ

カスタマイズ	4-10, 4-18, 4-19
--------	------------------

論理名

設定変更	4-33
定義	4-33
プリント・フォーマット	4-32, 4-33
プリント・フォーマットの	4-33

ワ

ワーキング・セット

定義	3-6
----	-----

