

CentreNET® X Vision
ユーザーマニュアル
詳細編

アライドテレシス株式会社

ご注意

本書の中に含まれる情報は、弊社（アライドテレシス株式会社）が保有しています。弊社の同意なく本書の全体もしくは一部をコピーまたは転載しないでください。

弊社は、予告なく本書の全体もしくは一部を修正または改訂することがあります。あらかじめご了承ください。

弊社は、改良のため予告なく製品の仕様を変更することがあります。あらかじめご了承ください。

本製品の内容またはその仕様に関連して発生した結果については、いかなる責任も負いかねますのであらかじめご了承ください。

Copyright 2001 アライドテレシス株式会社

商標について

CentreNETは、アライドテレシス株式会社の登録商標です。

UNIXは、X/Openカンパニーリミテッドがライセンスする米国ならびに他の国における登録商標です。

Microsoft、MS-DOS、Windows、Windows NTは、米国Microsoft Corporationの米国その他の国における登録商標です。

HPは、米国Hewlett-Packard Companyの商標または登録商標です。

Intel、Pentiumはインテルの登録商標です。Celeronはインテルの商標です。

その他、本書に記載されている会社名、製品名等は、各社の商標または登録商標です。

マニュアルバージョン

2001年2月 初版

目次

ようこそX Visionへ	9	ネットワーク接続の使用 (サポート対象外) ...	45
このマニュアルについて	10	シリアル接続またはモデム接続の使用 (サポート対象外)	46
参考マニュアル	11	シリアルプロトコル	47
その他の情報源	12	シリアル構成またはモデム構成の概要	48
このマニュアルで使用する表記法	12	TAPI デバイスとしてのモデムの構成	49
X Vision の PC へのセットアップ	15	シリアルトランスポートまたはモデムトランスポートの構成	50
X Vision の UNIX へのインストール ..	17	Serial-Com1 トランスポートの構成	51
UNIX ソフトウェアの概要	18	Modem トランスポートの構成	53
XRemote ホストソフトウェアについて	18	シリアルまたはモデム USP トランスポートの作成	54
ハードウェアとソフトウェアの前提条件	19	シリアル接続またはモデム接続に対するホストの構成	56
UNIX ソフトウェアのインストール	19	端末エミュレータからのモデム接続のオープン	58
インストール時の注意点	21	コネクションマネージャ (サポート対象外) ...	59
UNIX 版 Vision サービスの構成	22	コネクションモニタの使用 (サポート対象外)	60
UNIX ソフトウェアの削除	22	Vision コミュニケーションデータベース (サポート対象外)	61
Vision コミュニケーションアーキテクチャ	23	データベースサーバの構成	62
通信の概要	24	Vision USP サービス	63
Vision コミュニケーションアーキテクチャのモデル	26	Vision サービスプログラム	63
システム設定の変更	29	USP リモートプロシージャコールサービス ...	64
トランスポートの構成	30	USP ソケットサービス	65
トランスポートの追加	31	資源の検索 (サポート対象外)	66
通信オプションの指定	32	ネームスペース環境	66
トランスポート用セキュリティオプションの指定	34	ネームスペースマネージャ	67
トランスポート用スクリプトオプションの指定	35	ローカルネームエージェント	69
詳細オプションの指定	36	サーバの構成	70
ホストの構成	38	サーバプログラムの起動	71
ホストファインダを使用したホストの構成 ...	38	他のネットワーク上にある資源の検索	72
コントロールパネルを使用したホストの構成	40	認証 (サポート対象外)	73
Unix ネットワークコンピュータを使用したホストの構成	41	認証のタイプ	73
ホストマネージャを使用したホストの構成 ...	42	認証サーバ	74
ホストオプションの指定	43	クライアント認証	74
ホスト用セキュリティオプションの指定	44	パスワードの格納	75
ホスト用スクリプトオプションの指定	44	ライセンス管理 (サポート対象外)	76
		ユーザのドメインへの組織化	77
		ライセンス番号	77
		製品と構成要素	77

ライセンスサーバ	78
クライアントの認証	78
ライセンス要求の異常終了イベントの管理	79
イベントとエラーのログ収集	80
PCでのログ収集	80
UNIXでのログ収集	81
Xサーバの使い方	83
Xサーバの概要	84
制限事項	86
起動と終了	86
Xサーバの起動	86
Xサーバの手動クローズ	87
xdmのメインプログラムのクローズ	88
Xサーバの構成	88
Xサーバのプロファイル	89
ディスプレイモードの選択	90
ウィンドウの管理	92
アクティブなトランスポートの選択	92
パフォーマンスの最適化	93
グラフィックスピードオプティマイザ	93
グラフィックキャッシュ	95
セキュリティ	96
ホストアクセス制御の使い方	96
XDMCPセキュリティの使い方	96
ローカルな認証ファイルの使い方	98
キーボードの使い方	99
Xにおけるキーボード入力	99
Xサーバのキーボード入力	99
Xキーマップの構成	101
XサーバとXmodmap	101
ショートカットキー	101
修飾キー	102
Num Lockの管理	103
マウスの使い方	103
フォントの使い方	104
Xにおけるフォント	104
フォントとXサーバ	106
標準的なXフォント	106
フォントパス項目	107
PCへの追加フォントのインストール	109
フォントフォルダ項目	110

フォントサーバ	113
XクライアントでのWindowsフォントの使い方	114
他のWindowsアプリケーションでのXフォントの使い方	115
フォント置換の使い方	115
カラーの使い方	118
Xにおけるカラー	118
カラーとXサーバ	119
カラーマップの管理	121
カラー名データベース	122
コピーアンドペースト	123
Xにおけるコピーアンドペースト	123
WindowsとX間でのコピー	124
グラフィックのコピー	124
サウンドの扱い方	125
Helpful Interactive Support	127
Xエラーのログ	128
Xサーバに関する情報の検索	129
Xキーマップの使用	133
キーマップファイルの形式	134
キーマップ記述	134
キーマップテーブル	134
NumLockテーブル	137
修飾子テーブル	138
キーマップの修正	138
ALTのMod2へのマッピング	139
ファンクションキーのマッピング	142
キーマップのコンパイル	144
新しいキーマップの作成	145
Windowsキーコードの表示	147
Xウィンドウの管理	149
ウィンドウ管理の概要	150
Visionウィンドウマネージャ	150
Visionウィンドウマネージャの概要	151
初期化	151
クライアントウィンドウのマッピング	152
フォーカスポリシー	154
カラーマップ管理	154
クライアントのクローズ	155
Motifとの互換性	156

ウィンドウプロパティ	156	Vision Resume の動作のしくみ	201
VWM 関数	161	X サーバと Vision Resume	201
Motif の使用	162	PC での Vision Resume の構成	202
Motif と X サーバ	163	UNIX での Vision Resume の構成	203
ローカル MWM について	163	Vision Resume セッションの起動または再開 ..	204
ローカル MWM の起動	164	Resume セッションでのクライアントの起動 ..	206
ホストをベースにした MWM の起動	164	Vision Resume セッションの一時停止	207
Motif セッションの終了	165	Vision Resume セッションの再配置	207
MWM の構成	165	Vision Resume セッションの終了	208
OPEN LOOK の使用	166	シリアル接続またはモデム接続での Vision	
OPEN LOOK と X サーバ	166	Resume の使用	209
OLWM を使用するための X サーバの構成 ..	167	端末エミュレーション	211
ホスト環境の構成	168	端末エミュレーションの概要	212
OPEN LOOK ウィンドウマネージャの起動 ..	168	端末接続の開始	213
OPEN LOOK セッションの終了	169	VCA 端末サービス	216
X プログラムの起動	171	VCA 端末サービスの概要	216
X プログラム起動の概要	172	端末接続方式の選択	216
Unix アプリケーションウィザードの使用	172	USP タスクサービス	217
リモートプログラムスタータの使用	176	端末エミュレータドキュメント	219
端末エミュレータを使用した X クライアントの起		エミュレータドキュメントの概要	219
動	178	Windows 2000/Me/98/95 または Windows NT	
UNIX システムプロンプトからのクライアント		4.0 環境でのドキュメントの作成	219
の起動	179	Windows NT 3.51 環境でのドキュメントの作成	
X Display Manager の使用	181	220	
X Display Manager の概要	181	エミュレータドキュメントの構成	220
X サーバと XDMCP	181	サマリ情報	221
X Display Manager 用 X サーバの構成	182	ウィンドウのサイズおよび位置の変更	221
UNIX での X Display Manager の構成	184	ヒストリビューの表示	221
X Display Manager セッションの起動	186	フォント	222
X Display Manager セッションの停止	189	カラー	223
その他の情報源	189	ツールバー	223
NCD XRemote の使用 (サポート対象外)	190	キーボード入力	223
XRemote の概要	190	コピーアンドペースト	225
X サーバと XRemote	191	イベント	225
PC での XRemote の構成	192	印刷できない文字	226
XRemote の UNIX へのインストール	193	スタイル	226
UNIX での XRemote の構成	194	記録および再生	227
XRemote セッションの起動	196	端末エミュレータからの印刷	227
XRemote セッションの終了	198	印刷画面	228
セッションの一時停止と再開	199	スレープ印刷	228
Vision Resume の概要	200	他のドキュメントへのリンク	229
		リンクの概要	229

エミュレータオブジェクトの埋め込み	229
別のアプリケーションへのエミュレータドキュメントのテキストのコピー	232
ホットリンクの作成	232
エミュレータドキュメントでのプログラミング 234	
VT420 エミュレータ	235
VT420 の仕様	235
サポートされていない機能	235
VT420 端末の設定	236
VT420 キャラクタセット	236
OLE オートメーション	239
OLE オートメーションの概要	240
Vision オブジェクト	240
タイプライブラリ	242
関数の概要	242
Connect	243
Disconnect	244
Load	245
Save	246
Scrape	247
SendKey	248
SendString	249
ShowWindow	250
エミュレータキーマップの編集	251
エミュレータキーマップ編集の概要	252
キーマップの作成	252
エミュレータでのキーマップの変更	254
リモートホストの参照	255
参照の概要	256
Windows 2000/Me/98/95 または Windows NT 4.0 上での参照	256
[パス] ボタンの使い方	259
Windows NT 3.51 での参照	259
VCA ファイルアクセスサービス	260
VCA ファイルアクセスサービスの概要	261
ファイルアクセス方式の選び方	261
USP ファイルアクセスサービス (サポート対象 外)	262
FTP ファイルアクセス	262
ファイルタイプ	264

ファイルタイプの概要	264
ローカルファイルタイプ	264
リモートファイルタイプ	265
パターンマッチング	268
アクション	269
属性	269
関連付け	270
ファイルのコピーと移動	271
ファイルのコピーと移動の概要	271
ファイル名	271
改行文字	272
バッチファイル (サポート対象外)	272
hostmgr コマンド (サポート対象外)	273
デスクトップショートカット	274
プログラムマネージャへのコピー	275
リモートプログラムの実行	277
リモートプログラムの実行の概要	278
プログラムの実行	278
VCA リモート実行サービス	280
VCA リモート実行サービスの概要	280
リモート実行方式の選び方	281
USP タスクサービス (リモート実行)	282
rexec コマンドプロセッサ	282
rsh コマンドプロセッサ	282
リモートプログラムスタータのドキュメント	283
印刷	285
印刷の概要	286
UNIX から PC へのローカル印刷	286
ローカルプリンタの追加	287
BSD UNIX で LPD を使用するためのリモートプ リントラの構成	288
BSD UNIX で RPC を使用するためのリモートプ リントラの構成 (サポート対象外)	289
UNIX System V で LPD を使用するためのリモ トプリンタの構成	290
UNIX System V で RPC を使用するためのリモ トプリンタの構成 (サポート対象外)	290
印刷要求の表示	292
PC から UNIX へのリモート印刷	292
リモートプリンタの追加	293
メッセージ処理	299

メッセージ処理の概要	300
メッセージパッド	300
テキストメッセージ	302
マルチメディアメッセージ	303
UNIX メッセージ	305
ユーザ情報の表示	307
ユーザ情報表示の概要	308
ユーザ情報の表示	308
ユーザプロパティの表示	309
Zone によるデスクトップの並べ替え 311	
Zone プログラムの概要	312
Zone の並べ替え	312
Zone プログラムの終了または Zone の削除	314
Zone コントロールの使い方	315
Zone プロパティの表示	316
プログラムの自動起動	316
用語集	319
UNIX コマンド	341
UNIX ファイル形式	377
ComScript	399
X Keysyms	443
X Vision を使用するためのヒントと秘訣 455	
端末エミュレータのキーマップ	471
索引	481

ようこそX Visionへ

X Vision は、Microsoft Windows コンピュータシステムで動作する透過的な X サーバです。X Visionには、ハードウェアから最高のパフォーマンス引き出すための自動構成機能、インストールされているネットワークソフトウェアの識別機能、その他簡単にクライアントを実行するための各種機能(フォント自動置換など)を提供します。

X Vision では次のことを行うことができます。

- リモートの UNIX システム上で実行したプログラムを Microsoft Windows PC 上で表示できます。X プログラムまたは文字端末 (VT420) アプリケーションを実行でき、しかもこれらのアプリケーションをデスクトップ上に構成する作業もウィザードによって簡単に行うことができます。
- Windows ファイルの参照と同じ方法で、UNIX システム上のファイルやフォルダを参照し、ファイル階層を表示できます。
- X クライアント、文字端末アプリケーション、Microsoft Windows アプリケーションの間で情報をコピーできます。
- 端末エミュレータドキュメントの埋め込みやホットリンクの作成によって、リモート情報を Windows ドキュメントに統合できます。
- OLE オートメーションを使用して、端末エミュレータを実行している UNIX の文字ベースアプリケーションに対して新しい Windows インタフェースを作成できます。
- PC と UNIX システム間でファイルの転送ができます。
- キーボード構成ファイルの作成と変更ができます。
- 仮想デスクトップスペースを Zone プログラムで作成することによって、複数のゾーンにウィンドウを編成できます。
- PC に接続したプリンタにリモートホストからファイルを印刷できます。
- UNIX システムに接続したプリンタに Windows アプリケーションから印刷できます。

このマニュアルについて

このマニュアルでは、X Vision パッケージの構成要素についてテクニカルな情報も提供します。「ユーザーマニュアル・詳細編」は、オンラインヘルプでも提供されます。

- 1 **X Vision の PC へのセットアップ** 別冊子「ユーザーマニュアル・インストール編」をご覧ください。
- 2 **X Vision の UNIX へのインストール (サポート対象外)** この章では、X Vision と共に提供される UNIX ソフトウェアのインストール方法を説明します。また、ソフトウェアの削除方法も説明します。
- 3 **Vision コミュニケーションアーキテクチャ (部分的にサポートされない機能があります)** この章では、X Vision プログラムが PC をリモートホストシステムに接続する際に使用する Vision コミュニケーションアーキテクチャを説明します。
- 4 **X サーバの使用** この章では、Vision X サーバとその設定方法を説明します。
- 5 **X キーマップでの作業** この章では、X サーバと共に使用する X キーマップの作成と変更の方法を説明します。
- 6 **X ウィンドウの管理** この章では、X サーバ使用時に利用できるウィンドウ管理オプションを説明します。
- 7 **X プログラムの起動** この章では、X アプリケーションのさまざまな起動方法を説明します。また、シリアルラインやモデムから X セッションを起動する方法の詳細も説明します。
- 8 **セッションの一時停止と再開 (サポート対象外)** この章では、X セッションの一時停止とその後の再開ができる Vision Resume について説明します。
- 9 **端末エミュレーション** この章では、端末接続について、また端末エミュレーション機能について説明します。
- 10 **OLE オートメーション** この章では、OLE オートメーション技術のサポートについて説明します。
- 11 **エミュレータキーマップの編集** この章では、端末エミュレータのキーボードマップを紹介し、それらの編集方法を説明します。
- 12 **リモートホストの参照 (部分的にサポートされない機能があります)** この章では、リモート UNIX システム上にあるファイルの参照方法を説明します。プログラムの実行やドキュメントのオープンを簡単に行うための UNIX ファイルタイプの登録について説明します。

- 13 **リモートプログラムの実行** この章では、端末接続をオープンせずにリモート UNIX システム上でプログラムを実行する方法を検討します。
 - 14 **印刷（部分的にサポートされない機能があります）** この章では、PC に接続されたプリンタでの UNIX アプリケーションの印刷方法と、Windows アプリケーションで UNIX プリンタを使用する方法を説明します。
 - 15 **メッセージ処理（サポート対象外）** この章では、X Vision で強化されたメッセージ交換機能を紹介します。
 - 16 **ユーザ情報の表示（サポート対象外）** この章では、現在、どの PC が X Vision ソフトウェアを実行しているのかを検索する方法を示します。
 - 17 **Zone によるデスクトップの並べ換え** この章では、仮想デスクトップスペースを作成することで、ウィンドウを複数のゾーンに組織化する Zone プログラムを説明します。
- A **用語集** この付録では、用語解説を行います。
 - B **UNIX コマンド（サポート対象外）** X Vision のホストソフトウェアで利用できる UNIX コマンドについて、UNIX のマニュアルページ形式で説明します。
 - C **UNIX ファイル形式（サポート対象外）** X Vision に関連する UNIX ファイルについて、UNIX のマニュアルページ形式で説明します。
 - D **ComScript** この付録では、自動接続の制御に使用する Vision 通信スクリプト言語を説明します。
 - E **Xキーシンボル** この付録は、X キーシンボルのリストです。
 - F **X Vision を使用するためのヒントと秘訣** この付録では、特定のトランスポートまたは特定のホストオペレーティングシステムで X Vision を使用する際の情報を示します。
 - G **キーボードマップ** この付録は、PC のキーボード上のキーと実際の端末のキーボードにあるキーとの間のマッピングのリストです。

参考マニュアル

X Vision 用のマニュアルは、コンピュータ画面で Microsoft Windows 標準のヘルプアプリケーションを使用してオンラインでアクセスできます。この情報のほとんどは、印刷媒体でも用意されています。

オンライン情報

ヘルプシステムにアクセスするには、[スタート] メニューをクリックし、[CentreNET X Vision] をポイントしてから [CentreNET X Vision ヘルプ] をクリックします。Windows NT 3.51 ではプログラムマネージャの [CentreNET X Vision] グループにある [CentreNET X Vision ヘルプ] アイコンをダブルクリックします。

[目次] タブを使用して X Vision のマニュアルを展開します。表示される情報の内容は、PC に X Vision のどのソフトウェアパッケージがインストールされているかによって異なります。[X Vision の紹介] を参照すると、インストールされている Vision ソフトウェアの機能についての基本情報が得られます。特定のタスクの命令については、[キーワード] を使用するか、[目次] から [方法] を選択してください。システム管理者用の詳細については、[目次] から利用できる [リファレンスブック] を参照してください。

X Vision プログラムを使用していると、[ヘルプ] メニューから [トピックの検索] を選択することで、そのプログラムに固有のヘルプを得ることができます。またウィンドウやダイアログボックスの特定のオプションについての情報を得ることもできます。タイトルバーで  をクリックしてからオプションをクリックします。Windows NT 3.51 の場合には、SHIFT+F1 を押してヘルプポインタを表示させてから、オプションをクリックします。

印刷情報

『CentreNET X Vision ユーザーマニュアル・かんたん編』 X クライアントと文字端末アプリケーションを実行するために必要な情報の要約です。

『CentreNET X Vision ユーザーマニュアル・詳細編』 X Vision の構成要素についての完全な参照情報を記載しています。

その他の情報源

Microsoft Windows の基本情報については、Windows パッケージのマニュアルを参照してください。技術的な情報については、Windows のバージョンに合った『Microsoft Windows リソースキット』を参照してください。

ホストのオペレーティングシステムについての情報は、使用している UNIX のバージョンのマニュアルを参照してください。

このマニュアルで使用する表記法

次に示す表記用語、テキストフォーマット、記号は、X Vision のマニュアル全体で使用しています。

表記	意味
[大括弧]	構文中のオプション項目を囲みます。たとえば、[password] はコマンドと共にパスワードも指定可能なことを示しています。括弧内の情報だけを入力し、大括弧自体は入力しないでください。
... (省略記号)	構文で、その前の項目を繰り返せることを示します。たとえば、filename [...] はファイル名の間にコマンドを入れると、複数のファイルを指定できることを示しています。

	“または”を示し、中括弧や大括弧内の項目を区切ります。たとえば、{0 1} は0または1のどちらかを必ず入力することを示しています。
太字	ダイアログボックスまたはコマンドプロンプトに入力する実際のコマンド、単語、文字を示します。
<i>斜体</i>	入力する必要のある情報やパラメータの可変部分を示します。たとえば、プロシージャから <i>filename</i> の入力が必要されると、ファイルの実際の名前を入力する必要があります。
C:¥Windows	Windows ファイル名は大文字と小文字の混在する太字で示されます。これは、Windows エクスプローラで表示されるファイル名と同じです。
unixprog	UNIX ファイル名は太字で示されます。UNIX では大文字と小文字が区別されるので、大文字と小文字が混在していることもあります。
Monospace	画面上のテキスト、またはユーザがコマンド行または初期化ファイルで入力するエントリの例を表します。
Windows	Windows 2000/Me/98/95 または Windows NT 4.0 以降のオペレーティングシステムを指します。
Windows NT 3.51	Windows NT 4.0 以前のバージョンを指します。これらのバージョンではユーザインタフェースが異なります。
Windows フォルダ	Windows システムフォルダツリーを指します。これは、Windows をインストールしたときにユーザが指定した C:¥Windows または他のいずれかのフォルダ名になります。
x86	32 ビットのコンピュータ、x86 をベースにしたマイクロプロセッサ (Intel® 80386 またはそれ以降など)、または Intel の Pentium® マイクロプロセッサを指します。

(空白のページです)

第1章

X Vision の PC へのセットアップ

CentreNET X Vision の PC へのセットアップは、別冊子「ユーザーマニュアル・インストール編」をご覧ください。

第 2 章

X Vision の UNIX へのインストール

この章では、X Vision で提供される UNIX ソフトウェアの概要と、インストール手順を説明します（この章の内容はサポート対象外です）。

この章では次の内容を説明します。

- UNIX ソフトウェアの概要
- XRemote ホストソフトウェアについて
- ハードウェアとソフトウェアの前提条件
- UNIX へのインストール
- UNIX からの削除

UNIX ソフトウェアの概要

X Vision で提供される UNIX ソフトウェアには、UNIX 版 Vision サービスと、追加の X サービスモジュールがあります。UNIX ソフトウェアがインストールされない場合、X Vision プログラムの中には正常に作動するものもありますが、作動しないプログラムや機能が制限されるプログラムもあります。

UNIX 版 Vision サービスは、端末接続、リモート実行、ファイルアクセス、印刷、強化されたメッセージ交換、ライセンス管理といった十分に安全なサービスを UNIX コンピュータ上で提供します。これらのサービスは、ネットワークを経由して直接使用できます。シリアルラインまたはモデムリンクを行うユーザに対しては、ネットワークホスト上のすべてのサービスへのアクセスをネットワーク上で可能にするために、PC に対してプロキシとして作動するヘルパープロセスが提供されます。多重化プロトコルによって、複数のプログラムが単一の接続上で実行できます。

X サービスモジュールは、X コンポーネント用のサポートを提供し、XRemote や Vision Resume のようなサービスがシステムで利用できる場合には、それらに対するサポートも提供します。これらの追加サービスをサポートするプラットフォームの情報については、`¥Unix¥Unixhost.txt` ファイルを参照してください。XRemote では、シリアルラインまたはモデム接続経由で X クライアントを実行できます。Vision Resume では、X セッションを一時停止し、同じクライアントで後で再開することができます。

X Vision ホストソフトウェアには、操作するためのライセンスが必要です。インストール中にライセンス番号を入力しない場合には、一定期間使用できるソフトウェアの評価版が使用できます。X Vision を購入した場合には、X Vision 標準コンポーネント用のライセンス番号が提供されます。Vision Resume は、別個に購入する必要があり、すでに購入した場合には、Vision Resume だけのライセンス番号が提供されます。Vision Resume 用のライセンス番号がない場合には、評価版としてインストールできます。Vision Resume を購入し、ライセンス番号を入力したときに、すべてのライセンスを持つバージョンにアップグレードできます。

XRemote ホストソフトウェアについて

XRemote ソフトウェアは NCD によって開発されたものであり、NCD X 端末や XRemote と互換性のある他のサーバがサイトで使用中の場合には、ホストにすでにインストールされている可能性があります。

X Vision ホストソフトウェアをインストールすると、プラットフォームで利用できるのであれば XRemote コンポーネントがインストールされます。X Vision でサポートされる UNIX システムの中には、XRemote が利用できないものもあります。これらは、`¥Unix¥Unixhost.txt` ファイルに記載されています。

また、X Vision の CD には、XRemote の標準 NCD 配布ソフトウェアも含まれており、これは、その他の UNIX や VMS もサポートしています。Xremote フォルダには MS-DOS 圧縮ファイルフォーマットのファイルが入っており、1つのファイルは1つのプラットフォームをサポートしています。サポートしているオペレーティングシステムの全リストと、インストール手順は `✕Xremote✕Readme.txt` ファイルに記載されています。通常、X Vision UNIX ソフトウェアをインストールしているのであれば、XRemote が X サービスモジュールの一部としてインストールされるので、ここに書かれているインストール方法を使用する必要はありません。

ハードウェアとソフトウェアの前提条件

UNIX ソフトウェアのインストールと使用には、次が必要です。

- X Vision ソフトウェアが利用できる UNIX のバージョンを稼動しているコンピュータ。サポートされるオペレーティングシステムの全リストは、CD 内の `/unix/unixhost.txt` ファイルに記載されています。
- TCP/IP ネットワークソフトウェアが UNIX ホスト上にインストールされている必要があります。
- 約 5 MB のディスクスペース (UNIX 版の Vision サービスがインストールされている場合には 2 MB)
- ISO9660 フォーマットの CD を読める CD-ROM ドライブが UNIX サーバ上にあること、あるいは、CD-ROM ドライブが搭載された PC にネットワークでアクセスできること

UNIX ソフトウェアのインストール

X Vision UNIX ソフトウェアは、次の方法を使用してインストールできます。

- 1 ISO9660 フォーマットをサポートする CD-ROM ドライブが UNIX サーバ上にあるならば、その CD から直接インストールできます。
- 2 CD を PC 上にロードして、UNIX サーバに tar フォーマットのファイルをコピーできます。

アップグレードの場合には、以前のバージョン上にインストールできます。構成ファイルを変更していて、それをインストール中に保存していない場合には、古い構成ファイルが先頭に O つきのファイル名で保存されます。たとえば、VISION_NS_GATEWAY 変数を設定するために `vision.conf` ファイルを変更した場合には、ファイルは `Ovision.conf` という名前で保存されます。インストール後に手でファイルを編集しなおし、Vision サービスが実行中の場合にはリスタートする必要があります。

インストール時に、エクスプレスインストールまたはカスタムインストールのいずれを実行するのが質問されます。一般にエクスプレスインストールオプションが適しており、インストール中の質問の数もあまりありません。インストールするソフトウェアを制御する場合にはカスタムインストールを選択してください。制御できるのは、X Vision ソフトウェア

のインストール先ディレクトリ、どのサーバプログラムを起動させるかです。他の X Vision ソフトウェアがすでにこのホスト上にインストールされている場合には、このソフトウェアを同じディレクトリにインストールする必要があります。

また、X Vision ソフトウェアパッケージにライセンス番号を入力することも要求されます。インストール中にライセンス番号を入力しない場合には、Vision Resume 以外のすべてのコンポーネントに対しては `licadmin` コマンドを使用して後で指定することもできます。Vision Resume の場合には、`xresume -licadd license_number` を使用してください。

注意 USP トランスポート経由でホストに接続されている端末エミュレータでは、X Vision UNIX ソフトウェアをインストールしないでください。USP 以外のトランスポートを選択してホストに接続するのであれば、X Vision エミュレータを使用できます。

CD から UNIX ソフトウェアをインストールするには

- 1 UNIX システムに `root` としてログインします。
- 2 CD をマウントします。CD のマウント方法を知らない場合には UNIX のマニュアルを参照してください。
- 3 CD 上の `unix` ディレクトリにディレクトリを変更します。たとえば、CD が `/cdrom` にマウントされている場合は、`cd /cdrom/unix` コマンドを入力します。
- 4 `./install` と入力します。

CD-ROM ファイルシステムの表記が UNIX システムによって違うので、`unix` ディレクトリとインストールスクリプト名は大文字にする場合もあります。インストールスクリプト名には、バージョン番号を付ける場合もあり、たとえば、HP-UX の場合には `./INSTALL;1` と入力します。

インストールスクリプトは UNIX のバージョンを検出し、インストールできる X Vision サービスのメニューを表示します。UNIX ソフトウェアのインストールは、画面に表示される手順に従ってください。

tar ファイルから UNIX ソフトウェアをインストールするには

- 1 UNIX システムに `root` としてログインします。
- 2 使用している UNIX のバージョンに合った tar フォーマットのファイルを `/tmp` などの UNIX サーバの一時ディレクトリにコピーし、そのディレクトリに `cd` します。UNIX 版 Vision サービスのファイルと X サービスファイルの両方をコピーする必要があります。

X Vision ソフトウェアをすでにインストールしている PC であれば、どの PC にも CD をロードでき、Unix ネットワークコンピュータまたはホストマネージャ (Windows NT の以前のバージョン用) を使用して UNIX サーバに tar ファイルをコピーできます。この tar ファイルは CD 上の

¥Unix¥Unixhost.txt ファイルにリストされています。ツールバーから [バイナリ転送モード] を選択してください。それ以外の方法として、別のファイル転送機構を使用することもできます。

使用している UNIX のバージョンに合ったファイルを見つける必要がある場合には、CD から UNIX システムにファイル ¥Unix¥Modules¥Scoarch をコピーできます。このスクリプトを実行すると、8 文字のコードが戻ります。最初の 4 文字はアーキテクチャを示し、後の 4 つの数字はバージョン番号を示します。4 文字のアーキテクチャコードは CD 上の ¥Unix フォルダ内にあるサブフォルダ名に対応し、4 つの数字のバージョン番号は tar ファイル名に使われます。たとえば、scoarch が Ågi3sc0322” を戻した場合には、i3sc ディレクトリでファイル 0322.vs と 0322.xv を探してください。

- 3 tar xvf filename と入力して tar ファイルからファイルを抽出します。filename はシステム固有の tar ファイルの名前です。
- 4 UNIX 版 Vision サービスをインストールするには、./vsinstall コマンドを入力します。
UNIX 版 Vision サービスのインストールは、画面に表示される手順に従ってください。X サービスをインストールする前に UNIX 版 Vision サービスをインストールする必要があります。
- 5 X サービスをインストールするには、./xvinstall コマンドを入力します。
X サービスのインストールは、画面に表示される手順に従ってください。

インストール時の注意点

次のオプションスイッチは、install コマンドで使用できます。

- o code 4 文字の code で指定されるシステムにインストールします。これは、UNIX システムのタイプを自動的に検出する機構を上書きします。システムコードは CD 上の ¥Unix フォルダ内のサブフォルダ名として使用され、¥Unix¥Unixhost.txt テキストファイル内にリストされています。たとえば、install -o i3sc は OpenServer へのインストールを指定します。
- f filename 指定された tar filename を使用してインストールします。インストールスクリプトが適切な tar ファイルを自動的に抽出できなかった場合にはこのオプションを使用します。抽出の失敗は、ファイル表記が UNIX システム間で異なるために発生します。たとえば、install -f “MYUX/9999.XV;1” のように指定します。

ほとんどのファイルは、/etc/vision.conf およびシンボリックリンク /usr/bin/usputppxy (/usr/local/vision/bin/usputppxy にリンクされる) を除く Vision メインディレクトリ (デフォルトで /usr/local/vision) のサブディレクトリにインストールされます。前にインストールされたファイルよりも古いものはコピーされません。

カスタムインストールでは、このホスト上でネームスペースマネージャ (NSM) を実行するかどうかを質問されます。画面上に表示される手順を参考にして、NSM を実行するかどうかを決定してください。また、マスタの NSM を指定するようにも要求されます。ネットワーク上のどの NSM をマスタにするかどうかの選択は、新しい NSM が起動するときにトリガされます。マスタの指定はこの時に考慮されます。

UNIX 版 Vision サービスの構成

Vision サービスは、インストール処理の最後に起動されます。システムのブート時にローカルネームエージェントプログラムが自動的にリスタートされるようにしてください。リスタートを保証するには、適切なシステムスタートアップファイル (UNIX システム V リリース4 の場合は `/etc/rc`) に次のコマンドを入れます。

```
nohup /usr/local/vision/bin/lina -b &
```

このソフトウェアをデフォルトディレクトリにインストールしなかった場合には、インストール先ディレクトリに合わせてパスを修正してください。

UNIX ソフトウェアの削除

注意 USP トランスポート経由でホストに接続された端末エミュレータでは、X Vision UNIX ソフトウェアを削除しないでください。USP 以外のトランスポートを選択してホストに接続するのであれば、X Vision エミュレータを使用できます。

UNIX ソフトウェアを削除するには

- 1 これから削除しようとするサービスを使用しているユーザがいらないことを確認します。
- 2 UNIX システムに `root` としてログインします。
- 3 ディレクトリを X Vision UNIX ソフトウェアがインストールされたディレクトリに変更します。たとえば、`cd /usr/local/vision` を行います。
- 4 X サービスを削除するには、コマンド `./xvremove` を入力します。
画面上に表示される手順に従って、X サービスを削除します。
- 5 UNIX 版 Vision サービスを削除するには、コマンド `./vsremove` を入力します。

画面上に表示される手順に従って、UNIX 版 Vision サービスを削除します。

注意 UNIX 版 Vision サービスは、他の X Vision 製品と共有されるコンポーネントなので、UNIX 版 Vision サービスを削除すると、稼動している他の製品を停止したり、利用できる機能を減らす可能性があります。

第3章

Vision コミュニケーションアーキテクチャ

このセクションの内容は、次のとおりです（トランスポートは、イーサネット上のTCP/IPのみをサポートしています）。

- 通信の概要
- Vision コミュニケーションアーキテクチャのモデル
- システム設定の変更
- トランスポートの構成
- ホストの構成
- ネットワーク接続の使用（サポート対象外）
- シリアル接続またはモデム接続の使用（サポート対象外）
- Vision コミュニケーションデータベース（サポート対象外）
- Vision USP サービス
- 資源の検索（サポート対象外）
- 認証（サポート対象外）
- ライセンス管理（サポート対象外）
- イベントとエラーのログ収集

通信の概要

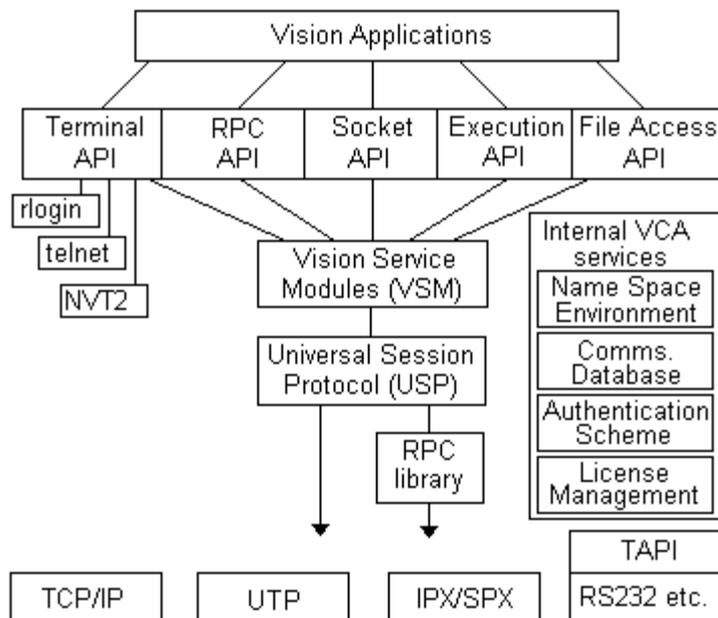
X Vision のソフトウェア製品は、クライアント/サーバ技術によってコンピュータ資源を最も効率よく活用します。ほとんどのアプリケーションには、PCの構成要素とUNIXの構成要素があります。Vision コミュニケーションアーキテクチャは、これらの構成要素間の最も重要なリンクを提供します。その設計のキーポイントは次のとおりです

- Vision コミュニケーションソフトウェアは、それを使用しているアプリケーションとは独立した別個の階層を形成します。
- PC 構成要素と UNIX 構成要素を持つ X Vision ソフトウェア製品は、通信リンクとして Vision コミュニケーションソフトウェアを使用します。
- 通信トランスポートと利用可能なホストは、利用可能なサービスと共に Vision コミュニケーション (ソフトウェア) に定義されています。
- Vision Toolkit は、一連の API (Application Programming Interfaces) によって細かく定義されたインタフェースを通信ソフトウェアに提供します。他の X Vision プログラムは、これらの API を使用して、必要な資源とサービスにアクセスします。
- Vision コミュニケーションアーキテクチャは、TCP/IP をサポートします。Vision ソフトウェアは、Windows 環境で構成された既存の通信トランスポートの最上部で実行されます。Vision コミュニケーションアーキテクチャは、プロトコルスタックを含みません (独立した TCP/IP スタックが提供されるパッケージもあります)。
- プログラムやユーザがアプリケーションとやり取りをする方法は、使用されるトランスポートプロトコルには関係なく類似しています。たとえば、サービスは、下位のプロトコルが認識するアドレスに自動的に翻訳される標準的な命名規則を使用して要求されます。
- Vision コミュニケーションアーキテクチャは、シリアルラインでは独自のプロトコルを提供します。このプロトコルは、ネットワーク接続と同様の接続性を提供します。SLIP や PPP などを使用するよりも効率的で簡単にインストールや管理ができます。
- PC と UNIX システムの両方を保護するセキュリティ対策は、Vision コミュニケーションアーキテクチャの必須部分です。
- Vision コミュニケーションアーキテクチャは、リモートシステム管理用に設計されています。たとえば、パスワードは通常、Vision コミュニケーションアーキテクチャが保護するオブジェクトと一緒に格納されません。

- Vision コミュニケーションソフトウェアは、高性能アーキテクチャを提供する完全な 32 ビットの実装です。
- Vision コミュニケーションアーキテクチャは、Windows 2000/Me/98/95 の機能を最大限に利用しており、Windows 2000/Me/98/95 の規則に従っています。必要な場合には、Windows NT 3.51をサポートするための代替方法が提供されます。

Vision コミュニケーションアーキテクチャのモデル

次の図は、Vision コミュニケーションアーキテクチャモデルを示しています
(トランスポートは、イーサネット上のTCP/IPのみをサポートしています)。



Vision アプリケーション

X Vision ソフトウェアパッケージで提供されるアプリケーションは、Vision Toolkit API に対して呼び出しを行うことで接続性を実装します。これによって、これらのアプリケーションは必要なサービスを提供する方式から独立することができます。

Vision Toolkit API

高度な API (Application Programming Interfaces) 群が、端末、RPC、ソケット、実行、ファイルアクセスの関数に対する標準のインタフェースを提供します。API の中には、たとえば端末 API のように、Vision コミュニケーションの USP サービスの他に Rlogin などの標準ネットワークサービスをサポートするものもあります。これらのサービスでは、ユーザまたはシステム管理者がサービスを提供する方式を構成できます。

Vision サービスモジュール (VSM)

USP を使用した機能別のサービスを実装します。これらのモジュールは、Vision ソフトウェアがインストールされているホストに対して標準のサービスをサポートする最適の方式を提供します。サービスの中には、USP とそれに関連する VSM を使用するときだけにサポートされるものもあります。

ユニバーサルセッションプロトコル (USP)

他のソフトウェア階層に対して標準化された通信インタフェースを提示する一方で、Vision コミュニケーションアーキテクチャに対しては複数トランスポートプロトコルのサポートを許可します。USP は、TCP/IP トランスポートが利用可能な場合には、TCP/IP 経由で直接動作します。PC 上で TCP/IP が利用できない場合には、USP プロキシプロセスが UNIX システム上で実行され、ネットワーク上の PC の代わりになります。このプロキシプロセスは、PC との通信のために、任意の有効なトランスポートを経由して動作する UTP を使用します。

RPC ライブラリ

USP の RPC サービスは Sun ONC RPC の最上部に階層化されており、この RPC 自体がトランスポートとして Vision UTP プロトコルを使用するために変更されています。また、ONC のトランスポートに依存しない RPC に提供される機能に似た非同期 RPC 呼び出し用の機能が追加されています。

トランスポートプロトコル

Vision コミュニケーションアーキテクチャでサポートされるネットワークは、TCP/IP です。

ユニバーサルトランスポートプロトコル (UTP) は、任意の通信トランスポート上で使用できる信頼性の高い多重プロトコルです。UTP によって、Vision PC と UNIX ソフトウェアは RS232C シリアルリンクなどの TCP/IP 以外のトランスポートを経由した通信ができます。たとえば、オフィスから離れたところで仕事をしているノートブック型 PC のユーザは、モデムを使用してダイヤルインを行い、オフィスネットワーク上にある Vision コミュニケーションサービスすべてにアクセスできます。

TAPI

Microsoft Windows の TAPI (Telephony Application Programming Interface) は、さまざまな電話線ネットワークサービスへのアクセスを提供します。これによって、Windows アプリケーションは、サービスに固有なプログラミングコードをまったく使わずに音声やデータのリンクを制御できます。Windows 2000/Me/98/95 と Windows NT では、TAPI はモデムや、ISDN などのデジタル電話サービスとのインタフェースを取る標準の方法です。

ネームスペース環境 (NSE)

Vision 資源にアクセスする際に、トランスポートに依存しない一貫したメカニズムを提供します。NSE は UNC (Uniform Naming Convention) フォーマットを使用して資源の特定を行います。これによって、クライアントは、特定のネットワークアドレス指定形式には依存しない文字列で、目的のホストシステムにあるサービスを要求できます。NSE は複写されたネームスペースマネージャ (NSM) と、それに関連するローカルネームエージェント (LNA) を使用して、クライアント/サーバアーキテクチャに最も適した安全でかつ柔軟性のあるフレームワークを実装しています。

Vision コミュニケーションデータベース

Vision コミュニケーションアーキテクチャで構成可能なすべての要素の詳細を保持するファイルの集合です。

認証スキーム

基本的な UNIX ユーザ認識メカニズムを改良し、それを PC をベースにしたサーバに展開する設計になっています。サーバプログラムが提供するサービスに必要な認証のタイプを指定するのは、各サーバプログラムの責任になります。クライアントは UNIX 認証サーバにコンタクトして、トークンを取得してから、対象のサービスにアクセスできます。

ライセンス管理

弊社から提供される X Vision の Windows ソフトウェア製品は、同時使用ユーザの数をもとにライセンスされます。ライセンス管理機能によって、システム管理者はライセンスを効率的に管理できます。

システム設定の変更

管理者は基本通信階層を構成できます。また、ホストに UNIX 版 Vision サービスがインストールされている場合には、より高いレベルのサービスも構成できます（**トランスポートは、イーサネット上のTCP/IPのみをサポートしていません**）。

TCP/IP ネットワークを経由して接続している場合には、Vision コミュニケーション設定の変更は行わないでください。他のトランスポートを使用している場合には、手動で行う構成がいくつか必要になることもあります。

通常、Vision サービス設定で必要な変更は、プリンタ設定に対するものだけです。

システム設定を変更するには

1 次のいずれか 1 つを行います。

- [スタート] ボタンをクリックし、[設定] をポイントしてからコントロールパネルをクリックします。
- Windows NT 3.51の場合は、プログラムマネージャにある [メイン] グループを開いてから [コントロールパネル] アイコンをダブルクリックします。



2 [Vision コミュニケーション] または [Vision サービス] のいずれかをダブルクリックします。

Windows 2000/Me/98/95 または Windows NT では、タイトルバーで **?** をクリックしてから項目をクリックすると、その項目の [ヘルプ] を得ることができます。Windows NT 3.51 では、SHIFT+F1 を押してヘルプポインタを表示させてから、項目をクリックします。

トランスポートの構成

Vision コミュニケーションデータベースには、X Vision プログラムで使用できるトランスポートの詳細が保持されています。トランスポートの構成に関するキーポイントは、次のとおりです（トランスポートは、イーサネット上のTCP/IPのみをサポートしています）。

- Local トランスポートはPC 上のプロセス間通信のメカニズムを定義します。これ以外のトランスポートは、リモートホストをアクセスするためのメカニズムを定義しています。この定義には、接続に使用されるプロトコル、ログインとコマンドの実行に使用するスクリプト、必要な認証、利用可能なサービスと提供方法を指定した詳細設定があります。
- トランスポートの選択は、Vision コミュニケーションアーキテクチャをPC にインストールするときに作成されます。TCP-Unix などのいくつかのトランスポートは常に作成されますが、他のものはその下位のトランスポートメカニズムがPC 上に存在するときにだけ作成されます。
- Vision コミュニケーションアーキテクチャサービスをシリアルラインやモデムを経由して接続する場合には、新しいトランスポートを構成する必要があります。理由は、USP プロキシプロセスが実行するホストの詳細を指定する必要があるためです。
- トランスポートは、有効または無効を明示的に指定できます。また、その利用可能状態は動的に検出できます。利用可能状態をチェックするために、Vision コミュニケーションは下位のトランスポートメカニズムをプローブし、トランスポートが [Available]、[unavailable]、[Disabled] のいずれの利用可能状態なのかを判断します。
- トランスポート定義は、リモートホストの接続方法を指定するためにホストオブジェクトで使用されます。デフォルトでは、トランスポートは接続がオープンされたときに自動的に選択されます。これが動作するように、トランスポートはPC から使用される可能性のある全種類の接続に対して構成されていることが必要です。たとえば、ネットワーク接続とモデム接続を構成しておく必要があります。
- Local 以外のすべてのトランスポートでは、ComScript スクリプト言語で書かれたスクリプトが、UNIX ホスト上で接続、ログイン、コマンドの実行処理のさまざまな部分を制御するために使用されます（付録「ComScript」参照）。

トランスポートの追加

トランスポートの追加、削除、変更には、Windows のコントロールパネルが使われます。

トランスポートを追加するには

- 1 Windows のコントロールパネルで、[Vision コミュニケーション] をダブルクリックしてから [トランスポート] タブをクリックします。構成されたトランスポートを示すダイアログボックスが表示されます。



- 2 新しいトランスポートを作成するには [追加]、変更するには既存のトランスポートを選択してから [プロパティ]、削除するには [削除] をそれぞれクリックします。

新しいトランスポートを追加するときには、説明を入力し、使用するプロトコルを選択する必要があります (この章で後述する「通信オプションの指定」を参照)。通常はデフォルト値が適切なので、他のオプションを変更する必要はほとんどありません。設定の情報を確認する場合はプロパティダイアログボックスのコンテキストヘルプを使用してください。また、概要についてはこの章を参照してください。

トランスポートを自動選択の対象として構成するには

- 1 各トランスポートに対するプロパティウィンドウで、トランスポートを自動選択の対象とするかどうかを指定します。
- 2 [トランスポート] タブで矢印ボタンを使用してトランスポートの順序を調整します。ホストがトランスポートを自動選択するように構成されている場合には、Vision コミュニケーションアーキテクチャは、リストの先頭から順に、利用可能なトランスポートを使用してホストへの接続を試行していきます。

通信オプションの指定

[通信] タブでは、トランスポートの説明と使用するプロトコルを指定できます。ボタンを使用してさらに詳細を指定することもできます。



[説明] には、トランスポートを説明する任意のテキストを入力できます。指定できる [プロトコル] を次の表に示します。

オプション	説明
None	プロトコルは選択されません。
Local	PC 上のプロセス間通信に専用プロトコルを使用することを指定します。[設定] ボタンと [プロキシ] ボタンは選択できなくなっています。
TCP/IP	TCP/IP プロトコルを使用することを指定します。[設定] ボタンと [プロキシ] ボタンは選択できなくなっています。

オプション	説明
IPX/SPX	IPX/SPX プロトコルを使用することを指定します。[設定] ボタンと [プロキシ] ボタンは選択できなくなっています (サポート対象外)。
Serial	シリアルラインまたは TAPI 以外のモデムを使用して UNIX ホストと通信することを指定します。[設定] ボタンで COM ポートを選択し、必要に応じて電話番号を入力することができます (サポート対象外)。
Modem	UNIX ホストとの通信に TAPI モデムを使用することを指定します。[設定] ボタンでモデムを TAPI デバイスとして構成することを選択できます (サポート対象外)。
USP/TCP	TCP をトランスポートとして使用し、ユニバーサルセッションプロトコル (USP) サービスを利用することを指定します。[設定] ボタンと [プロキシ] ボタンは選択できなくなっています。
USP Via Proxy	ユニバーサルセッションプロトコル (USP) サービスがリモートホスト上で動作するプロキシから提供されることを指定します。これは [プロキシ] ボタンを使用して構成できます。[設定] ボタンは選択できなくなっています。
Xtrace	弊社固有のプロトコルであり、Vision X サーバのインストール時に X ウィンドウシステムで発生する障害の問題解決に使用できます。

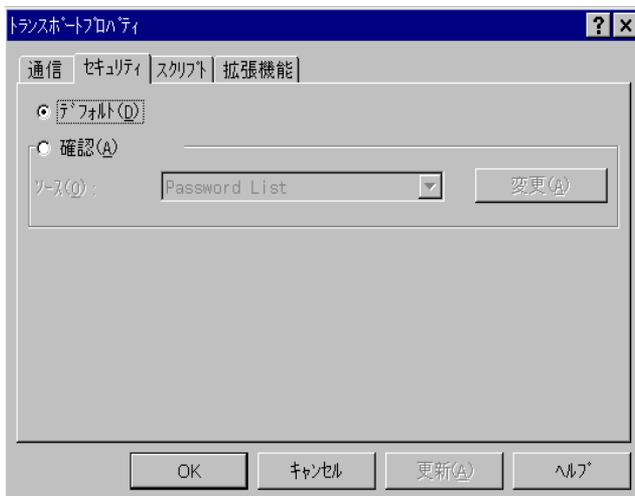
プロキシの設定

[プロキシ] ボタンを選択すると、ユニバーサルセッションプロトコルのヘルパープロセスが動作するリモートホストの詳細を指定できるダイアログボックスが表示されます。このヘルパープロセスは、PC をシリアルライン経由で接続する際にネットワーク上で PC 用のプロキシとして動作するので、ユーザはネットワーク上のすべてのホストにアクセスし、すべての Vision サービスを使用できるようになります。プロキシ用のホストは、ホストへのアクセスが可能になるトランスポート、たとえば [モデム] などを選択して、通常のホストを作成するのと同じ方法で構成します。

注意 プロキシホストの定義は、一般の Vision コミュニケーションアーキテクチャのホストオブジェクトと同じネームスペースには格納されません。一般のホストは、不必要な循環リンクを生じさせることになるので、プロキシホストとしては選択できません。

トランスポート用セキュリティオプションの指定

[セキュリティ] タブでは、パスワードの取得と格納の方法の指定や、パスワードの変更ができます。



[確認] を選択した場合には、次のオプションを選択できます。

オプション	説明
Password List	[Password List] ファイルにパスワードが格納されることを指定します。
Prompt	パスワードの入力を要求するプロンプトを常に表示することを指定します。
Permanent	保護するオブジェクトと共にパスワードが格納されることを指定します。

トランスポートを構成する際に、[デフォルト] 設定を指定すると [Password List] を使用する意味になります。ホストに対しては、[デフォルト] はトランスポート用に指定されたセキュリティ方式を使用することを指定します。

[Password List] または [Permanent] を選択して、[変更] を選択した場合には、ダイアログボックスでオブジェクト用に新しい [ユーザ名] と [パスワード] を入力できます。[Prompt] を選択した場合には、[変更] を選ぶと、プロンプトボックスが表示されたときにデフォルトで示される [ユーザ名] と [パスワード] を入力できます。

オプションの詳細については、この章で後述する「パスワードの格納」を参照してください。

トランスポート用スクリプトオプションの指定

[スクリプト] タブでは、ホストへの接続、ログイン、コマンドの実行について、そのいくつかまたはすべての処理の制御に使用する Vision コミュニケーションスクリプトファイルを指定できます。USP トランスポートでは、スクリプトのコマンド実行部分だけが使用されます。



[スクリプトオプション] を選択すると、スクリプトの[ファイル名]を選択するか[参照] ボタンを使用してスクリプトファイルを検索できます。スクリプトファイルは、通常は **C:\Program Files\Common Files\Vision\Scripts** に格納されています。

[モード] リストでは次のオプションを指定できます。

オプション	説明
Automatic	スクリプトが自動的に実行されることを指定します。
Manual	スクリプトをロードせずに [コネクションモニタ] ウィンドウが表示され、接続処理を手動で制御できることを指定します。
Trace	[コネクションモニタ] ウィンドウが指定したスクリプトをロードして表示されることを指定します。この指定で、接続処理の進行状況を表示できます。

[シェル] オプションでは、スクリプトファイルのコマンド実行部分を実行するために使用するシェルコマンドインタプリタを指定できます。インタプリタは、デフォルトでは Bourne シェルです。

トランスポートを構成する際、[デフォルト] の設定は、プロトコルに適したスクリプトファイルの使用、[Automatic] モード、Bourne シェルになります。ホストに対しては、[デフォルト] はトランスポート用に指定したスクリプトオプションを使用することを示しています。

スクリプトファイルの詳細については、付録の「ComScript」を参照してください。

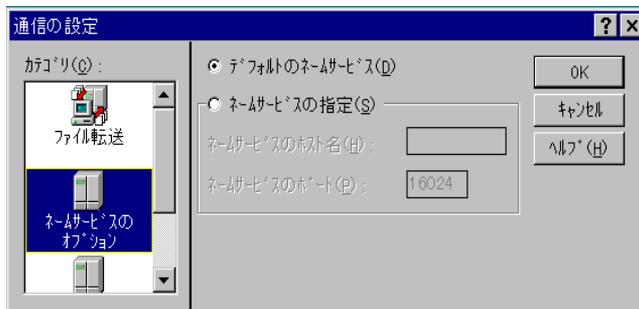
詳細オプションの指定

[拡張機能] タブでは、トランスポート用に指定可能な各サービスを提供する方式を指定できます。



上記のリストは、利用可能なサービスと、各サービスを提供するために使用する方式を示しています。方式を変更するには、リスト内のサービスをクリックして選択してから [変更] ボタンをクリックします。利用できるサービスと方式は [通信] タブにある [プロトコル] 設定によって異なります。ほとんどの場合、代替方法はありません。

[設定] ボタンを選択すると、現在選択されているサービスで利用できる構成オプションが表示されます。次の例は、USP/TCP プロトコル用の端末サービスの [通信の設定] を示しています。[カテゴリ] リストから選択すると、ダイアログボックスの右側にあるオプションが変更されます。図は、[Name Service Options] カテゴリを選択した状態を示しています。



コンテキストヘルプを使用すると、これらのダイアログボックスのオプションに関する情報が表示されます。Windows 2000/Me/98/95またはWindows NTでは、タイトルバーで **?** をクリックしてオプションをクリックします。Windows NT 3.51 ではSHIFT+F1 を押してヘルプポインタを表示させてからオプションをクリックします。

ホストの構成

Vision コミュニケーションデータベースには、リモートホストの詳細が保持されています。ホストの構成に関するキーポイントは次のとおりです。

- PC がネットワークに直結している場合には、通常、ホスト構成はホストファインダによって自動的に行われます。
- TAPI モデムを構成する場合は、Modem というホストがホストファインダによって自動的に作成されます。
- 直結のシリアルラインまたは TAPI 以外のモデムだけを使用する場合には、少なくとも 1 つのホストを手動で構成する必要があります。
- ホストは、Vision コミュニケーションデータベースにも格納されているトランスポート定義を使用します。トランスポートの選択は、PC に Vision コミュニケーションソフトウェアをインストールする際に作成されますが、構成の変更や新しいトランスポートの作成が必要な場合もあります (この章で前述した「トランスポートの構成」を参照)。
- ホストへの接続に使用されるトランスポートは、接続時に自動的に決定できません。オフィスでネットワークに接続し、家で仕事をするときにはモデム経由で接続するノートブック型 PC を持っている場合には、Vision コミュニケーションは適切なトランスポートを自動的に選択します。
- ユーザは、ユーザ名とパスワードをホストオブジェクトと共に格納するのかわ、独立したパスワードリストファイルに格納するのかわを選択できます。これによって、パスワードを提供せずに、ホストオブジェクトを他のユーザに配布できます。

ホストファインダを使用したホストの構成

通常、ホストはホストファインダプログラムによって自動的に構成されます。ホストファインダプログラムは、隠しバックグラウンドタスクとして実行されます。次に示すホストタイプを検出するように指定できます。

- Vision ホスト
- XDM ホスト
- エコーホスト
- Modem

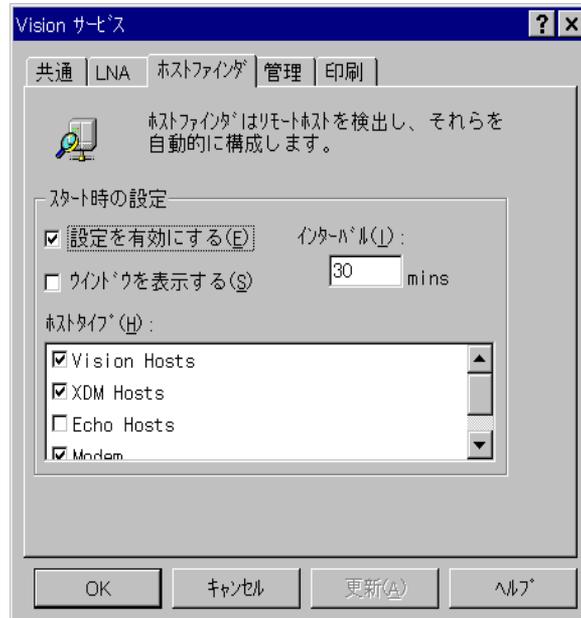
Vision ホストを検出するには、ホストファインダは RPC 経由でネームスペースマネージャに接続し、どの UNIX にローカルネームエージェントが登録されているのかわを検出します。XDM ホストは、X ウィンドウシステムのディスプレイマネージャデーモンである `xdm` を実行しているホストです。エコーホストは、TCP/IP を実行しているローカルネットワーク上のすべてのホストです。Modem ホストは Windows に TAPI モデムが構成されている場合に作成されます。TAPI 以外のモデム、シリアルライン、TCP/IP を実行していないホストは、ホストファインダでは検出されません。

ホストファインダが Vision コミュニケーションデータベースに登録したホストは、ホストファインダが検出したタイプでマークされます。ホストファインダは一定の間隔 (デフォルトは 30 分間隔) でネットワーク上のホストを走査し、それまで検出していたホストが存在しなくなったことを発見した場合には、利用不能なホストとしてマークします。利用不能なホストは、そのアイコンに x 記号がついて X Vision プログラムで表示されます。

PC が SuperVision によってリモートに管理される場合には、ホストファインダはどのホストが PC の管理ホストであるのかを検出します。

ホストファインダを構成するには

- 1 Windows のコントロールパネルで [Vision サービス] をダブルクリックしてから [ホストファインダ] タブをクリックします。
- 2 必要なオプションを選択します。



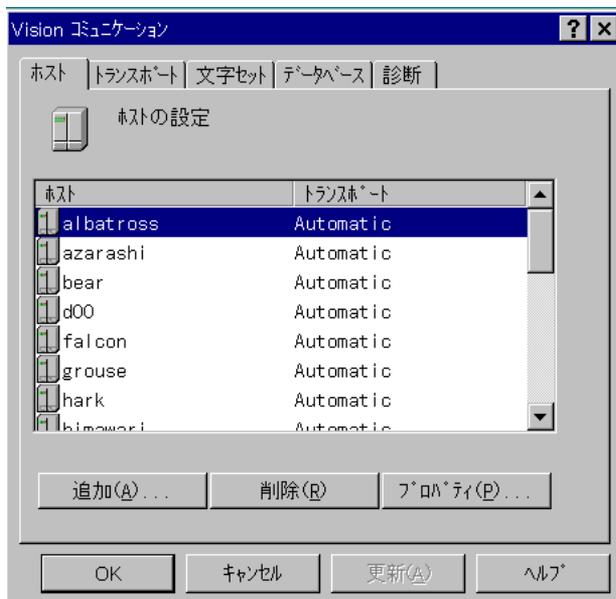
Windows 2000/Me/98/95 または Windows NT では、タイトルバーで **?** をクリックしてから項目をクリックすると、その項目の [ヘルプ] を得ることができます。Windows NT 3.51 では、SHIFT+F1 を押すとヘルプポインタが表示されるので、項目をクリックします。

コントロールパネルを使用したホストの構成

ホストの追加、削除、変更には、Windows の [コントロールパネル] を使用します。

コントロールパネルを使用してホストを追加するには

- 1 Windows のコントロールパネルで、[Vision コミュニケーション] をダブルクリックしてから [ホスト] タブをクリックします。構成されたホストを示すダイアログボックスが表示されます。



- 2 新しいホストを作成するには [追加]、変更するには既存のホストを選択してから [プロパティ]、削除するには [削除] をそれぞれクリックします。

新しいホストを追加するときには、ホスト名を入力するようにプロンプトが表示されます。通常はデフォルト値が適切なので、ホストプロパティを変更する必要はありません。トランスポートは自動的に選択されます。プロパティを変更するのであれば、リスト内のホストをクリックしてから [プロパティ] をクリックします。設定の情報を確認する場合はプロパティダイアログボックスのテキストヘルプを使用してください。また、概要についてはこの章を参照してください。

Unix ネットワークコンピュータを使用したホストの構成

Windows 2000/Me/98/95 または Windows NT 4.0 では、[Unix ネットワークコンピュータ]または[ホストエクスプローラ]を使用して手動でホストの追加、削除、変更を行うことができます。

Unix ネットワークコンピュータを使用してホストを追加するには

- 1 デスクトップ上にある [Unix ネットワークコンピュータ] アイコンをダブルクリックします。

ホストファインダが検出したホストが表示され、また、以前に手動で構成した他のホストも表示されます。



- 2 [ファイル] メニューをクリックしてから [新規ホスト] をクリックします。
新しいホストアイコンに一時的な名前がついて表示されます。
- 3 新しいホストに名前を入力し、ENTER を押します。

通常はデフォルト値が適切なので、ホストプロパティを変更する必要はありません。トランスポートは自動的に選択されます。プロパティを変更するのであれば、[ファイル]メニューをクリックしてから[プロパティ]をクリックします。設定の情報を確認する場合はプロパティダイアログボックスのコンテキストヘルプを使用してください。また、概要についてはこの章を参照してください。

新しいホスト項目をテストするには、そのホストのアイコンをダブルクリックして接続をオープンします。ユーザ名とパスワードを入力するプロンプトが表示されます。ここで入力したものはデフォルトでパスワードリストに格納されます。ログインができれば、2回目以降の接続でこの情報の入力が要求されることはありません。

ホストの変更や削除の詳細については、[ヘルプ]メニュー上の[トピックの検索]をクリックしてください。同じ情報がメインの[Vision ヘルプの目次]の[ホストエクスプローラの使い方]を選択しても表示されます。

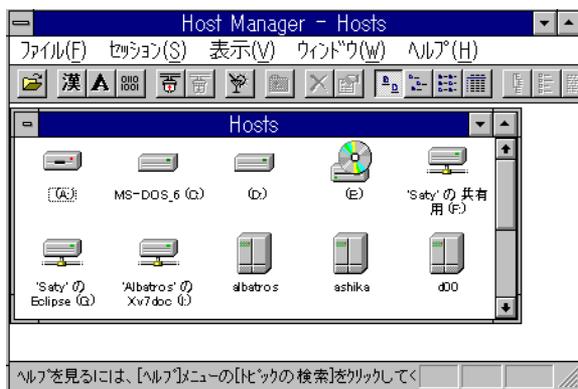
ホストマネージャを使用したホストの構成

Windows NT 3.51 以前では、[ホストマネージャ] を使用して手動でホストの追加、削除、変更を行うことができます。

ホストマネージャを使用してホストを追加するには

- 1 プログラムマネージャで [Vision グループ] を開き、[ホストマネージャ] アイコンをダブルクリックします。

PC ディスク、接続された Windows ドライブ、現在構成されている UNIX ホストを示すウィンドウが表示されます。



- 2 [ファイル] メニューをクリックしてから [新規作成] をクリックします。新しいホストアイコンに一時的な名前が表示されます。
- 3 新しいホストに名前を入力し、ENTER を押します。

通常はデフォルト値が適切なので、ホストプロパティを変更する必要はありません。トランスポートは自動的に選択されます。プロパティを変更するのであれば、[ファイル]メニューをクリックしてから [プロパティ] をクリックします。設定の情報を確認する場合はプロパティダイアログボックスのコンテキストヘルプを使用してください。また、概要についてはこの章を参照してください。

新しいホスト項目をテストするには、そのホストのアイコンをダブルクリックして接続をオープンします。ユーザ名とパスワードを入力するプロンプトが表示されます。ここで入力したものはデフォルトでパスワードリストに格納されます。ログインができれば、2 回目以降の接続でこの情報の入力が要求されることはありません。

ホストの変更や削除の詳細については、[ヘルプ]メニュー上の [トピックの検索] をクリックしてください。同じ情報がメインの [Vision ヘルプの目次] の [ホストマネージャの使い方] を選択しても表示されます。

ホストオプションの指定

[ホスト] タブでは、ホストに名前をつけ、ホストへの接続で使用するトランスポートを指定することができます。ホストを表すアイコンを選ぶこともできます。



[トランスポート] ボックスで [Automatic] を選択すると、トランスポートは接続時に自動的に選択されます。Vision コミュニケーションは、接続ができるまでトランスポートを順番に使用してリモートホストへのリンクをオープンしようとし、このとき、[Automatic Selection] オプションの [自動] が有効になっているトランスポートだけが対象になります。トランスポートの順序は [トランスポート] ダイアログボックスに指定できます (この章で前述した「トランスポートの追加」を参照)。

[ホスト名] ボックスでは、Vision プログラムに認識させたい名前をホスト名に指定します。通常、ここにはホストのネットワークノード名を指定しますが、どんな名前でも構いません。

[設定] ボタンは、接続に関する詳細な情報を提供するために使用できます。この情報はトランスポートによって異なります。ネットワークトランスポートでは、ネットワークノード名やアドレスなどの詳細を入力できます。[設定] ボタンを選択しなければ、[ホスト名] に入力された名前がネットワークノード名として使用されます。シリアルライトトランスポートでは、トランスポート用に指定されたものを上書きする [電話] 情報を指定できます。

ホスト用セキュリティオプションの指定

[セキュリティ] タブでは、トランスポートに指定された [セキュリティ] オプションを上書きできます。[デフォルト] を選択すると、トランスポート用の設定が使用されます。[確認] を選択すると、そのホストに対してユーザ名やパスワードをどのように管理するかを指定できます。

[セキュリティ] オプションの詳細については、この章で前述した「トランスポート用セキュリティオプションの指定」を参照してください。

ホスト用スクリプトオプションの指定

[スクリプト] タブでは、ホストへの接続、ログイン、コマンドの実行について、そのいくつかまたはすべての処理の制御に使用する Vision コミュニケーションスクリプトファイルを指定できます。USP トランスポートでは、スクリプトのコマンド実行部分だけが使用されます。

ホスト項目に対して、[スクリプト] タブを使用すると、トランスポート用に指定された [スクリプト] オプションを上書きできます。[デフォルト] を選択すると、トランスポート用の設定が使用されます。[スクリプトオプション] を選択すると、このホスト固有のオプションを指定できます。

[スクリプト] オプションの詳細については、この章で前述した「トランスポート用スクリプトオプションの指定」を参照してください。

ネットワーク接続の使用（サポート対象外）

Vision コミュニケーションアーキテクチャは、TCP/IP のネットワークだけをサポートしています。ネットワークソフトウェアのインストールについては、索引の「インストール」項目を調べてください。

TCP/IP スタックは Windows Sockets をサポートするものであれば任意のものを使用できますが、Windows で提供される Microsoft TCP/IP を使用することが多いでしょう。UDP/TCP プロトコルは、すべての Vision コミュニケーションサービスを TCP/IP ネットワーク経由で提供します。

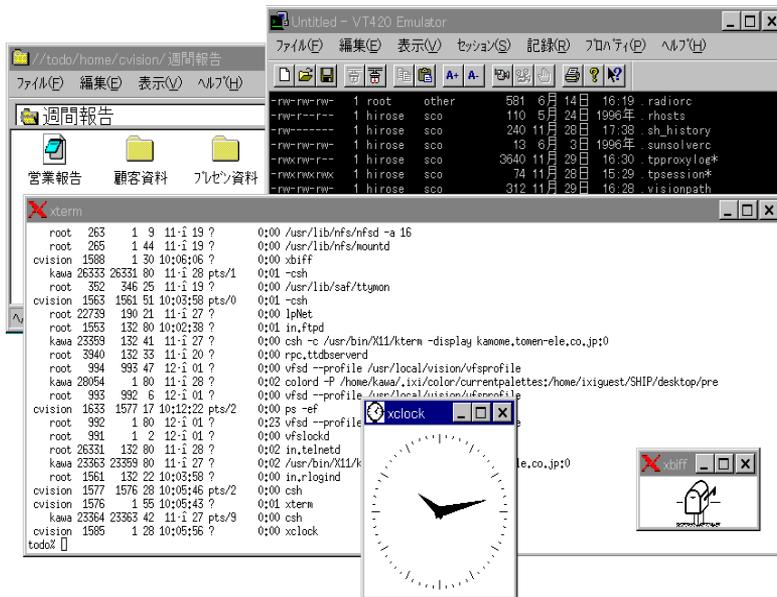
シリアル接続またはモデム接続の使用（サポート対象外）

Vision コミュニケーションアーキテクチャは、一定の範囲内のサービスを提供するトランスポートとして PC とリモートホストの間のシリアル接続に使用できます。正しいホストソフトウェアがインストールされると、シリアルライン経由でもネットワーク接続経由で利用可能なサービスが使用できます。シリアル接続はさまざまな方法で提供できますが、最も一般的な方法としては次のものがあります。

- PC と UNIX ホストとを直結する RS232C ケーブル
- Windows TAPI デバイスとして構成されるモデム (Windows 2000/Me/98/95 および Windows NT 用)

シリアルライン経由で提供されるサービスは、リモート UNIX システム上にホストソフトウェアがインストールされていない場合は限定されます。Vision USP サービスを使用するには、UNIX 版 Vision サービスがインストールされている必要があります。Vision X サーバがインストールされているときにシリアルライン経由で X ウィンドウシステムのクライアントを実行させるには、NCD の XRemote をインストールするようにしてください (現在使用中の X Vision 製品のリファレンスマニュアルにこれが含まれているのであれば、「X プログラムの起動」の章の「NCD XRemote の使用」を参照してください)。

次の図は、VT420 端末エミュレータ、Unix ネットワークコンピュータブラウザ、XRemote セッションを示しており、この 3 つの X クライアントは単一のモデム接続上ですべて動作しています。



シリアルプロトコル

リモートの UNIX システムにホストソフトウェアがインストールされていない場合には、**一般的な RS232C プロトコル** が使用されます。Vision コミュニケーションアーキテクチャは、このプロトコル上で端末接続とリモート実行サービスを提供します。

UNIX 版 Vision サービスがリモートホストにインストールされている場合には、**ユニバーサルセッションプロトコル (USP)** で、端末接続、リモート実行、ファイルアクセス、RPC、リモートソケットなど、すべての USP サービスが提供されます。ヘルパープロセスがネットワークに接続された UNIX システム上で実行され、PC の**プロキシ**として動作します。弊社固有の多重プロトコルである**ユニバーサルトランスポートプロトコル (UTP)** は、PC とプロキシプロセスとの間のシリアルライン経由の通信に使用されます。PC はプロキシへのリンクによってネットワークに接続された他のホストにアクセスできます。

Vision X サーバがインストールされているときに X ウィンドウシステムのクライアントを実行させるには、NCD の **XRemote プロトコル** を使用するようになっています。このプロトコルをサポートするためには、リモートホスト上に XRemote をインストールする必要があります。ホストシステム上では“プロキシサーバ”として動作するヘルパープロセスが機能します。このプロキシサーバは、X クライアントにとっては UNIX システム上で通常の X サーバが動作しているように見えます。ヘルパープロセスは PC との通信を行い、Vision X サーバとのメッセージの受け渡しを行います。XRemote プロトコルは、単純なシリアル接続や USP 端末接続など、下位の任意のトランスポートプロトコル上で使用できます。

ユニバーサルトランスポートプロトコル

ユニバーサルトランスポートプロトコル (UTP) は、任意の通信トランスポート上で使用できる信頼性の高い多重プロトコルです。UTP によって、X Vision の PC と UNIX ソフトウェアは、RS232C のシリアルリンクのように TCP/IP 以外のトランスポートで通信を行うことができます。たとえば、オフィスから離れた場所でノートブック型 PC で仕事をするユーザは、モデムを使用してダイヤルインを行い、オフィスネットワークのすべての Vision コミュニケーションサービスにアクセスすることができます。

UTP では、両方の通信リンクの終端でのサポートが必要になります。PC 側では、UTP のサポートは Vision コミュニケーションソフトウェアに組み込まれています。UNIX 側では、ヘルパープロセス `usputppxy` がネットワーク上の PC に対して USP プロキシのように働きます。

PC クライアントがサービスを要求すると、その要求はシリアルライン経由の伝送のためにパッケージ化されます。要求が UNIX システムに到達すると、メッセージはプロキシプロセスによって取り出されます。このプロキシプロセスは、次にネームスペースマネージャ `nsmsrv` を呼び出して USP アドレスを解析します。サーバプログラムへのアクセスを取得するために使用されるこのメカニズムは、UTP リンクからリモート PC にメッセージを転送するプロキシプロセスがあるので、これ以降はローカルに接続された PC と同じ動作になります。

`usputppxy` はネームスペースマネージャにリモート PC 上で動作するサーバを登録する責任があります。サービスは、プロキシプロセスが動作しているコンピュータのネットワークアドレスと、一意のポート番号を使用して登録されます。たとえば、ユーザ `steve` が `scobox` という UNIX コンピュータにリモート接続をしているとします。このとき、UNIX クライアントが USP アドレス `steve:prtsrv` の解釈をネームサーバに依頼するときには、`scobox` の IP アドレスと、`usputppxy` がサービス要求を受信しているポート番号が提供されます。

シリアル構成またはモデム構成の概要

シリアル接続でサービス全体にアクセスするためには、次を行う必要があります。

- モデムを TAPI デバイスとして構成します (Windows 2000/Me/98/95 および Windows NT の場合だけ)。
- Serial-Com1 または Modem トランスポート を構成し、オプションで新しい USP トランスポートを作成します。
- Vision コミュニケーションに 1 つ以上のホストエントリを構成します。
- Vision X サーバを使用している場合には、XRemote プロファイルを構成します (現在使用中の X Vision 製品用のリファレンスマニュアルにこれが含まれるのであれば、「X プログラムの起動」の章の「NCD XRemote の使用」を参照)

TAPI デバイスとしてのモデムの構成

Windows 2000/Me/98/95 と Windows NT では、コンピュータと電話ネットワークを統合する際には Microsoft 社の Telephony API (TAPI) を使用してください。TAPI は、相手先へのダイヤリング、着呼への応答などのオペレーション用に通信デバイスに依存しない一貫性のある制御を提供します。

モデムを TAPI デバイスとして構成するには

- 1 [スタート] をクリックして [設定] をポイントしてから [コントロールパネル] をクリックします。
- 2 [モデム] をダブルクリックします。モデムが現在インストールされていない場合には、モデムウィザードが起動します。



画面に表示されるモデムのセットアップ手順に従ってください。

モデムが正常にセットアップされると、次を行うことができます。

- どこから電話をかけるのか、どのように電話をかけるのかの指定
- 速度、フロー制御、エラー制御などの接続プロパティの指定
- モデムと COM ポートに対する診断テストの実行

シリアルトランスポートまたはモデムトランスポートの構成

Vision コミュニケーションアーキテクチャでは、PC からリモートコンピュータに接続できるようにトランスポートを構成する必要があります。X Vision ソフトウェアをインストールすると、Serial-Com1 トランスポートと Modem トランスポートが作成されます。Serial-Com1 トランスポートは直結のシリアルライン上で、または Windows TAPI デバイスとしては構成されていないモデム上で使用するためのものです。Modem トランスポートは TAPI デバイスとして構成されたモデムで使用してください。

Serial-Com1 と Modem トランスポートでは、端末接続サービスとリモート実行サービスが提供されます。Vision X サーバがあり、UNIX ホストに XRemote がインストールされているのであれば、どちらのトランスポートでも X セッションを実行することもできます。

UNIX 版 Vision サービスがリモートホスト上にインストールされている場合には、シリアルラインまたはモデム経由で USP (ユニバーサルセッションプロトコル) の全機能を使用できる新しいトランスポートを作成できます。このトランスポートは USP-Serial または USP-Modem と命名できます。これは、多重通信とエラー修正を行う UTP プロトコルを使用するので、単一の接続上で複数のプログラムを同時に実行できます。XRemote セッションもこのトランスポート上で実行でき、X クライアントと文字ベースのアプリケーションを同時に実行することができます。USP Terminal サービス上で XRemote を実行するもう 1 つの長所は、通常要求される 8 ビット接続の代わりに 7 ビット接続が可能であることです。

要約すると、手順は以下のようになります。

- 1 Modem トランスポート (TAPI モデムの場合) または Serial-Com1 トランスポート (直結のシリアルラインまたは TAPI 以外のモデムの場合) を構成します。
- 2 UNIX ホスト上で X Vision サービスを使用する場合には、USP-Serial または USP-Modem という新しいトランスポートを作成します。プロキシを実行しているホストを指定し、そのホストにアクセスするトランスポートとして Serial-Com1 または Modem を選択します。
- 3 適切なトランスポートを使用するホストを構成します。TAPI モデムがインストールされている場合には、ホストファインディングが Modem というホストを追加し、Modem トランスポートを指定します。

注意 プロキシホストを構成する際に使用するダイアログボックスは、Vision コミュニケーションデータベースでホストオブジェクトの構成に使用するものと似ています。ただし、プロキシホストの詳細はホストオブジェクトを作成しません。

使用する電話番号

モデムを使用するためにトランスポートとホストを構成すると、別々の場所にあるいくつかの電話番号や他の詳細を入力できます。電話番号を入力するかどうかの判断には、次の項目を考慮してください。

- USP-SerialまたはUSP-Modem トランスポートを使用して接続する場合には、プロキシホスト用に構成される電話番号および他の詳細が常に使用されます。
- Serial-Com1 または Modem トランスポートを使用して接続する場合、および、ホストのトランスポート設定が Automatic の場合には、トランスポート内に指定されている電話番号と他の詳細が使用されます。
- Serial-Com1 または Modem トランスポートを使用して接続する場合、および、ホストのトランスポートが明示的に Serial-Com1 または Modem に設定されている場合には、電話番号がホストの接続設定に指定されているのであれば、その電話番号が使用されます。ホストに電話番号が指定されていない場合には、トランスポート内に指定されている電話番号が使用されます。

注意 Vision コミュニケーションアーキテクチャは、Serial-Com1 で電話番号を見つけれない場合には、直結のシリアルラインが利用可能であると想定します。実際にはモデム経由で接続しようとしている場合には、“Waiting for login”メッセージが表示されて接続に失敗します。Modem トランスポートは電話番号を見つけれない場合には電話番号を要求するプロンプトを表示します。

Serial-Com1 トランスポートの構成

Serial-Com1 というトランスポートは、PC に X Vision ソフトウェアをインストールすると作成されます。このトランスポートは COM1 を直接使用するために構成され、COM ポートオプションはデフォルトの値 (9600 ボー、8 ビット、パリティなし、1 ストップビット、フロー制御なし) に設定されます。このトランスポートは、COM1 に直接接続された直結のシリアルラインやモデムがあり、かつ、UNIX ホストに接続するときに、Microsoft Windows の特徴を変更することなく動作します。ただし、デフォルト設定は最も単純で遅い接続に合うように選択されるので、Serial-Com1 トランスポートを再構成してください。

Serial-Com1 トランスポートを構成するには

- 1 Windows のコントロールパネルで [Vision コミュニケーション] をダブルクリックしてから [トランスポート] タブをクリックします。
- 2 [トランスポート] ボックスで [Serial-Com1] をダブルクリックします。
- 3 このトランスポートを自動トランスポート選択メカニズムの対象にするのであれば、[自動セクションにこのトランスポートを含める?] にチェックマークがあることを確認してください。

- 4 [設定] をクリックします。Windows 2000/Me/98/95 のダイアログボックスでは、[国コード] と [地域コード] ボックスは選択できなくなっています。モデムを使用して接続する場合には、[電話番号] を入力します。電話番号を入力しないと、直結のシリアルラインを使用すると解釈されます。[電話番号] を入力する場合には、地域コードや前につける番号を含めて番号を入力してください。前につける番号とは、たとえば、交換機を経由して接続する場合には、外線へのアクセス番号を入力してください。[接続] ボックスでは、[ポート設定] または他の COM ポートを選択します。
- 5 [設定] をクリックします。[ポート設定] ダイアログボックスで、[スピード] ボックスのドロップダウンリストから使用できる最も高速のボーレートを選択します (Vision コミュニケーションがボーレートをネゴシエートできるモデムを使用しているのであれば、リストの中で最も高速のものを選ぶことができます)。このダイアログボックスの他の設定は通常は変更できません。



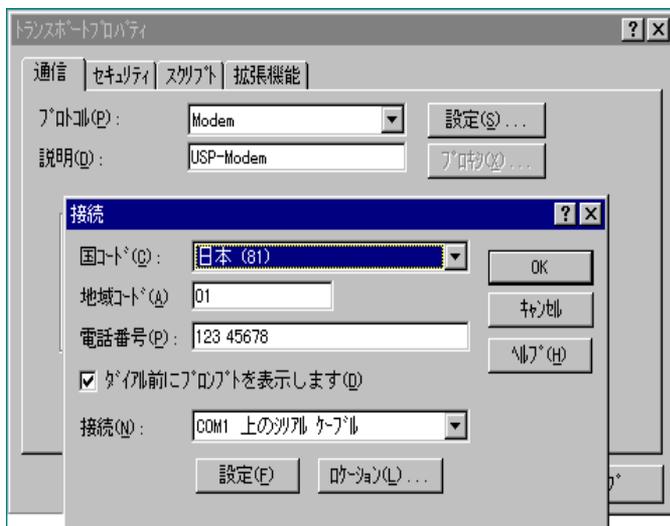
- 6 [OK] をクリックして [トランスポートプロパティ] ダイアログボックスに戻ります。ここで、リモートシステムへの接続やログインに使用するスクリプトを変更しても構いません。デフォルトのスクリプトは `Unixsl.scr` であり、このスクリプトはシリアルラインまたはモデムで UNIX ホストに接続するように設計されています。このスクリプトでは、トランスポートまたはトランスポートを使用するホストに電話番号が指定されているときにはモデム接続が必要であると認識します。別のスクリプトを使用する場合、たとえば、OpenServer ホストに `Unixslsc.scr` を使用する場合には、[スクリプト] タブをクリックして設定を変更してください。

Modem トランスポートの構成

Modem というトランスポートは、PC に X Vision ソフトウェアをインストールすると作成されます。このトランスポートは最初に利用できる TAPI モデムを使用するために構成されます。システムに複数の TAPI モデムが構成されている場合には、使用するモデムを選択するためにトランスポートを構成する必要があります。

Modem トランスポートを構成するには

- 1 Windows のコントロールパネルで [Vision コミュニケーション] をダブルクリックしてから [トランスポート] タブをクリックします。
- 2 [トランスポート] ボックスで [Modem] をダブルクリックします。
- 3 このトランスポートを自動トランスポート選択メカニズムの対象にするのであれば、[自動セレクションにこのトランスポートを含める?] にチェックマークがあることを確認してください。
- 4 [設定] をクリックします。ダイアログボックスで、[国コード] と [地域コード] にはすでにユーザがどこからダイアルするのかが表示されています。UNIX ホストに接続されているモデムの電話番号を入力してください。[接続] ボックスで、TAPI デバイスとして構成したモデムを選択してください。



- 5 [OK] をクリックして [トランスポートプロパティ] ダイアログボックスに戻ります。ここで、リモートシステムへの接続やログインに使用するスクリプトを変更しても構いません。デフォルトのスクリプトは `Unixsl.scr` であり、このスクリプトはシリアルラインまたはモデムで UNIX ホストに接続するように設計されています。このスクリプトでは、トランスポートまたはトランスポートを使用するホストに電話番号が指定されているときにはモデム接続が必要であると認識します。別のスクリプトを使用する場合、たとえば、OpenServer ホストに `Unixslsc.scr` を使用する場合には、[スクリプト] タブをクリックして設定を変更してください。

シリアルまたはモデム USP トランスポートの作成

X Vision プログラムのすべての機能を使用できるように、X Vision サービスを UNIX ホストにインストールしている場合には、プロキシ経由で USP サービスを実行させる新しいトランスポートを作成する必要があります。

注意 XRemote を使用する X クライアントを実行するだけならば、USP トランスポートを作成する必要はありません。ただし、USP 接続上で XRemote を実行すると、文字端末アプリケーションの実行や UNIX ファイルシステムの参照を同時に行うことができます。

USP のシリアルトランスポートまたはモデムトランスポートを作成するには

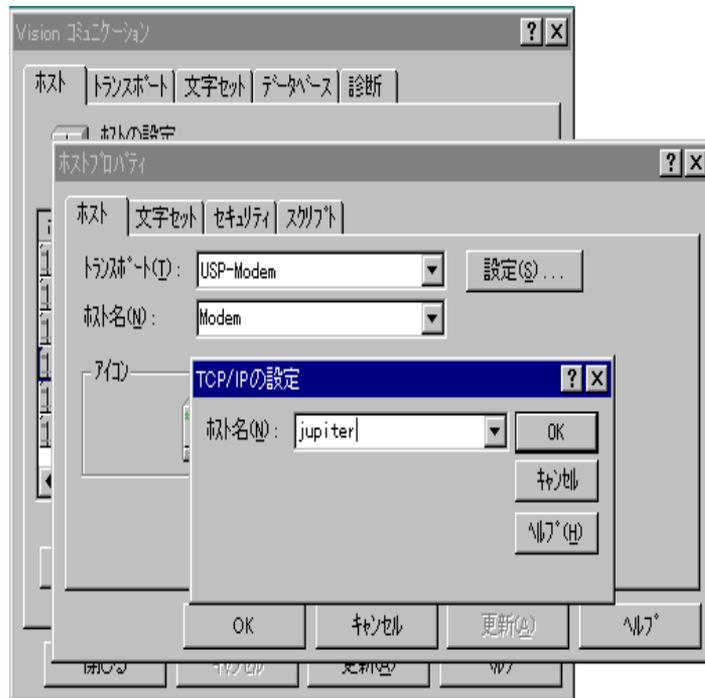
- 1 Windows のコントロールパネルで [Vision コミュニケーション] をダブルクリックしてから [トランスポート] タブをクリックします。
- 2 [トランスポート] ボックスで [追加] をクリックします。
- 3 [プロトコル] ボックスで、[USP via Proxy] を選択します。
- 4 [説明] ボックスで、下位のトランスポートとして [Modem] または [Serial-Com1] のいずれを使用するかによって、[USP-Modem] または [USP-Serial] を入力します。
- 5 このトランスポートを自動トランスポート選択の対象にするのであれば、[自動セレクションにこのトランスポートを含める?] にチェックマークがあることを確認してください。
- 6 [プロキシ] をクリックします。
- 7 [トランスポート] ボックスで、[Modem] (TAPI モデムの場合) または [Serial-Com1] (直結のシリアルラインまたは TAPI 以外のモデムの場合) を選択します。
- 8 [ホスト名] ボックスで、ネットワーク上でこの PC のプロキシとして動作する UNIX ホストのネットワークノード名を入力します。
- 9 [設定] をクリックします。
- 10 [電話番号] ボックスで、ネットワーク上でこの PC のプロキシとして動作する UNIX ホストに接続されているモデムの電話番号を入力します。[接続] ボックスで、[TAPI] モデムまたは [COM] ポートのいずれが適切なものを選択します。
- 11 [OK] をクリックして [トランスポート] ダイアログボックスに戻ります。
- 12 このトランスポートを自動トランスポート選択の対象に指定したのであれば、矢印ボタンを使用してトランスポートの順序を調整する必要があります。Vision コミュニケーションは、リストの先頭から順に各トランスポートを使用してホストとの接続を試行します。オフィスでネットワークに接続し、家で仕事をするときにはモデム経由で接続するノートブック型 PC を持っている場合には、適切な順序が次の図のように表示されます。



シリアル接続またはモデム接続に対するホストの構成

Windows で TAPI モデムが構成されている場合には、ホストファインダが Modem というホストを作成します。これは、Modem トランスポートを使用するために指定され、電話をかける前にプロンプトを表示してユーザ名とパスワードの入力を要求します。Modem ホストを使用すると、インストール後に構成を行うことなく、直接接続をオープンすることができます。TAPI モデムが存在しない場合には、最初のホストエントリを手動で作成する必要があります。

USP-Modem トランスポートを作成する場合には（「シリアルまたはモデム USP トランスポートの作成」参照）、このトランスポートを使用するために Modem ホストを変更できます。[TCP/IP 設定] ダイアログボックスの [ホスト名] を USP タスクサーバが実行されるホストの名前（通常はリモートモデムが接続されているホストのネットワークノード名）に変更する必要があります。たとえば、次の図では、ユーザは“jupiter”というホスト上でプロキシサーバを実行するために USP-Modem トランスポートを構成しています。ユーザはトランスポートを [Modem] ホストから [USP-Modem] に変更し、[設定] ボタンをクリックしています。続いて表示される [TCP/IP 設定] ダイアログボックスで、[ホスト名] として **jupiter** を入力しています。



USP トランスポートを使用している場合には、最初に Modem ホストに接続したときに ホストファインダが UNIX 版 Vision サービスを実行しているネットワーク上の他のすべてのホストを検出します。

PC がネットワークに接続されたときに X Vision ソフトウェアをインストールした場合には、UNIX 版 Vision サービスを実行しているすべてのホストがホストファインダによって自動的に構成されます。これらのホストは [Automatic] として指定されるトランスポートを持ち、ネットワークトランスポートを使用して接続を行います。たとえば、ノートブック型コンピュータを家に持ち帰ったときなどに USP-Modem トランスポートを使用して起動する場合には、[Unix ネットワークコンピュータ] に表示されたホストが最初は × 記号つきで、それらが利用できないことを示して表示されます。実際に、プロキシホストが USP-Modem トランスポートに構成されている場合には、[Unix ネットワークコンピュータ] にある任意のホストをダブルクリックし、接続をオープンすることができます。接続がオープンされると、次回ホストファインダが実行されたときに他のすべてのホストから × 記号が取り除かれます。

ホストの構成方法の詳細については、この章で前述した「ホストの構成」を参照してください。

端末エミュレータからのモデム接続のオープン

端末エミュレータを起動し、Modem ホストに接続すると、電話番号を確認し、必要があれば変更できるダイアログボックスが表示されてから、モデムが電話をかけます。モデムが TAPI デバイスとして構成されている場合には、次に示すダイアログボックスが表示されます。TAPI 以外のモデムの場合には、別のダイアログボックスが表示されます。

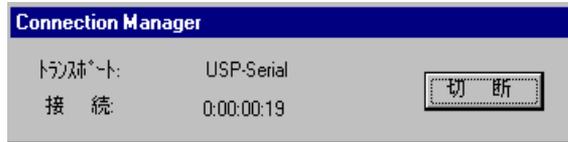


[OK] を選択すると、リモートモデムに電話がかけられ、接続がオープンされます。コマンドを指定すると、自動的にログインが行われ、そのコマンドが実行されます。コマンドが指定されていない場合には、UNIX ログインプロンプトが表示されるので、手動でログインする必要があります。

コネクションマネージャ (サポート対象外)

コネクションマネージャは、プロキシホストへの接続を提供するユニバーサルトランスポートプロトコル (UTP) を使用するとき、PC 上で実行され、多重リンクの PC 側を管理します。コネクションマネージャは、最初の接続が要求されたときに自動的に起動され、多重リンクの最後の接続がクローズされると終了します。

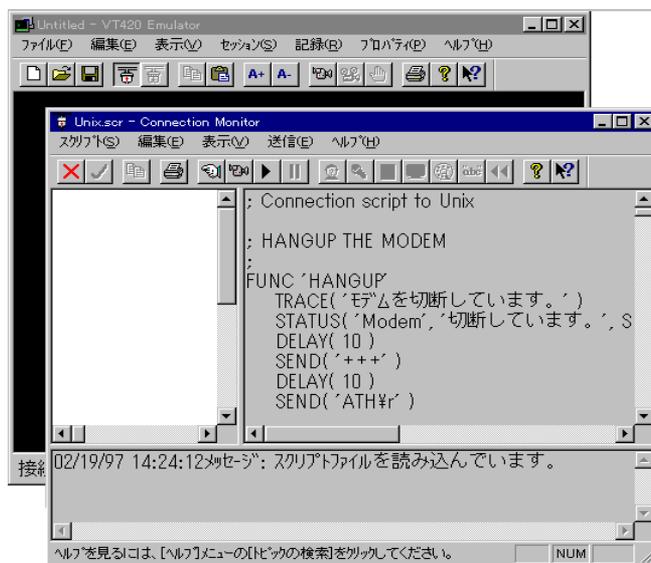
コネクションマネージャは、Windows 2000/Me/98/95 または Windows NT 4.0 ではタスクバー内に表示され、Windows NT 3.51 ではデスクトップ上のアイコンとして表示されます。[コネクションマネージャ] ウィンドウを開くと、接続時間の合計を表示できます。また、[切断] ボタンをクリックすることで、プロキシホストへの多重リンクを手動でクローズできます。このリンクを使用しているすべてのプログラムの接続を切断したい場合以外は、この操作は行わないでください。



コネクションマネージャは、[スタート] メニューまたはプログラムマネージャから手動で起動することもできます。これによって、UTP 接続がすでにオープンされているので、リンクを使用したプログラムをより迅速に起動できます。コネクションマネージャを手動で実行するには、[USP via Proxy] プロトコルを使用するトランスポートを事前に構成し、そのトランスポートが自動選択の対象になるように指定する必要があります。コネクションマネージャを起動すると、コネクションマネージャはトランスポートのリストにある最初の該当するトランスポートにプロキシホストへの接続をオープンします。代替方法として、複数のプロキシトランスポートがある場合には、コマンド行にトランスポート名を指定できます。この場合、トランスポートは自動選択用に構成する必要はありません。たとえば、Windows の [ファイル名を指定して実行] ダイアログボックスでは、USP-Modem トランスポートを作成した場合には **conman USP-Modem** と入力できます。コネクションマネージャがプロキシトランスポートを見つけられないときにコネクションマネージャを手動で起動すると、エラーメッセージが表示されます。

コネクションモニタの使用 (サポート対象外)

コネクションモニタを使用すると、接続処理の監視と制御を行うことができます。SHIFT キーを押しながら接続をオープンすると、コネクションモニタが表示されます。



詳しい操作方法については、[ヘルプ]メニューの[トピックの検索]をクリックしてください。同じ情報は、メインの[Vision ヘルプの目次]から[コネクションモニタの使い方]を選択しても表示できます。次の項目に関して[ヘルプ]が用意されています。

- ウィンドウの表示形態の変更
- 接続の監視
- 印刷
- スクリプトファイルの生成

Vision コミュニケーションデータベース (サポート対象外)

Vision コミュニケーションデータベースは、Vision コミュニケーションアーキテクチャで構成可能なすべての要素の詳細を保持しているファイルの集合です。このファイルには、次のものがあります。

- ホスト定義 (*.hst)
- トランスポート定義 (*.vct)
- サービス構成 (*.vcf)
- 認証パスワードリスト (*.vca)
- 通信定義 (*.vcc)
- ComScript スクリプトファイル (*.scr)
- アイコンライブラリ (*.dll)

これらのファイルは、C:\Program files\Common files\Vision フォルダのサブフォルダに格納されています。ファイルの間にくらかの依存関係はありますが、SuperVision で他の PC ユーザに分散できるように、依存は最低限になっています。たとえば、ホスト定義が特定のトランスポートを参照していても、必ずしもユーザ名とパスワードを格納してはいません。

ホスト定義 ホストオブジェクト (*.hst) は、ホスト名、ホストとの通信に使用するトランスポート、トランスポートで指定されている設定を上書きする設定を定義しています。“Automatic” クラスは、特定のホスト用のトランスポートの動的選択を指定するために使用され、標準構成は実際に利用可能なトランスポートメカニズムを考慮せずに使用できます。このスキームによって、“プラグアンドプレイ” オペレーションが許可され、ノートブック型 PC がオフィス内の LAN 接続や家からのシリアル接続やモデム接続経由でも使用できます。

トランスポート定義 トランスポートオブジェクト (*.vct) は、通信(トランスポート) プロトコル、セキュリティ、サービス、スクリプトの詳細を提供します。利用可能なサービスごとに、選択された利用可能な方式の詳細も保持されています。

サービス構成 サービス構成オブジェクト (*.vcf) は、Vision USP サービス用の設定を格納しています。サービスは、Windows のコントロールパネルにある [Vision サービス] オプションを使用して構成され独立した *.vcf ファイルでダイアログボックスの各タブ用の設定を格納しています。たとえば、Hostfind.vcf にはホストファインダ用の設定が格納されています。

この中には、すべてのサービスに対する [設定を有効にする] および [ウィンドウを表示する] 設定を保持している Launch.vcf という特別なインスタンスがあり、これが Vision サービスプログラムによって読み取られ、起動するサービスの決定に使用されます (サービスにこれらのオプションだけがある場合には、そのサービスには独立した *.vcf ファイルは存在しません)。

認証パスワードリスト パスワードリストファイル(*.vca)は、Windows ユーザが初めて Vision ホストにアクセスするときに作成されます。パスワードリストには、ホスト名、ユーザ名、パスワード詳細の相互関係が格納されています。Windows にログインするユーザごとに独立したファイルがあります。たとえば、stevel.vca のようなファイルがあります。

ComScript スクリプトファイル ComScript 言語によって作成されたスクリプトファイルは、ホストへの接続、ログイン、コマンドの実行の処理のすべてまたは一部を制御するために使われます。詳細については、付録の「ComScript」を参照してください。

アイコンライブラリ アイコンの選択は、Windows ライブラリ (*.dll) ファイルの形式で利用できます。さまざまなダイアログボックスから [アイコンの変更] ボタンを使用して、アイコンを選択することができます。

データベースサーバの構成

データベースサーバは、Vision コミュニケーションデータベースをアクセスする手段となります。データベースサーバを起動時に開始するか、ウィンドウを表示させるかを指定できます。

データベースサーバを構成するには

- 1 Windows のコントロールパネルで [Vision コミュニケーション] をダブルクリックしてから [データベース] タブをクリックします。
- 2 必要なオプションを選択します。



Windows 2000/Me/98/95 または Windows NT では、タイトルバーで **?** をクリックしてから項目をクリックすると、項目の [ヘルプ] を得ることができます。Windows NT 3.51 では、SHIFT+F1を押してヘルプポインタを表示させ、項目をクリックします。

Vision USP サービス

Vision コミュニケーションアーキテクチャは、X Vision プログラムやサードパーティのアプリケーションで使用できる一定の範囲のサービスを提供します。このサービスは、豊富な機能を持ち、特定のトランスポートやホストオペレーティングシステムに依存しません。サービスは、ユニバーサルセッションプロトコル (USP) によって実行され、Vision サービスモジュールによって PC 上で実装されます。これらのサービスは、Vision Toolkit API 経由でプログラムに組み込むことができます。サービスには、次のものがあります。

- **タスクサービス** 端末接続とリモート実行機能を提供します。詳細は、第9章「端末エミュレーション」の「USP タスクサービス」を参照してください。
- **ファイルアクセスサービス** リモートホスト上に格納されたファイルの参照と操作の機能を提供します。また、ホスト間でのファイル転送もできます。詳細は、第12章「リモートホストの参照」の「USP ファイルアクセスサービス」を参照してください。
- **リモートプロシージャコールサービス** クライアントプログラムがサーバプログラム内のプロシージャを呼び出す機能を提供します。詳細は、この章で後述する「USP リモートプロシージャコールサービス」を参照してください。
- **ソケットサービス** クライアントプログラムがサーバプログラム内にあるプロシージャを呼び出すための低いレベルの方式を提供します。詳細は、この章で後述する「USP ソケットサービス」を参照してください。

Vision サービスプログラム

Vision サービスプログラム (**Vservice.exe**) は、ローカルネームエージェント (LNA) などの PC への多くのサービスを直接提供します。それ以外のサービスは、LNA からの要求によって起動される独立したプログラムによって提供されます。[Vision サービス] のショートカットは、Windows が起動したときにプログラムが実行されるように、Windows の [セットアップ] フォルダや [プログラムマネージャ] グループ (Windows NT 3.51の場合) に置かれます。また、Windows のコントロールパネルからも手動でプログラムを起動できます。

Vision サービスプログラムは **Launch.vcf** ファイル (この章で前述した「Vision コミュニケーションデータベース」を参照) を読み取り、どのサービスを利用可能にするのかを決定します。

Vision サービスを手動で起動するには

- 1 Windows のコントロールパネルで [Vision サービス] をダブルクリックしてから [共通] タブをクリックします。
- 2 [スタート] をクリックします。



他のダイアログボックスオプションの情報については、コンテキストヘルプを使用してください。Windows 2000/Me/98/95 または Windows NT では、タイトルバーで **?** をクリックしてからオプションをクリックします。Windows NT 3.51 では、SHIFT+F1 を押すとヘルプポインタが表示されるので、オプションをクリックします。

Vision サービスをシャットダウンするには

- 1 Windows のコントロールパネルで [Vision サービス] をダブルクリックしてから [共通] タブをクリックします。
- 2 [シャットダウン] をクリックします。

警告メッセージが表示され、Vision サービスプログラムから起動されたサーバプログラムはすべてシャットダウンされます。

USP リモートプロシージャコールサービス

リモートプロシージャコール (RPC) システムでは、クライアントプログラムがサーバプログラム内のプロシージャを呼び出すことができます。このメカニズムは、多くの場合、特にクライアントプログラムとサーバプログラムがネットワーク内の異なるコンピュータ上で実行されている場合は代替方法である下位レベルのソケットインタフェースより便利です。Vision RPC メカニズムの設計のキーポイントは次のとおりです。

- Vision RPC は Sun ONC RPC の最上部階層に置かれており、Sun ONC RPC は Vision USP プロトコルを使用するために変更されています。さらに、ONC トランスポートに依存しない RPC に提供される機能に似た 非同期 RPC 呼び出し用の機能が追加されています。
- 起動中のサーバ内の ONC ポートマッパーの機能は、Vision ローカルネームエージェントに置き換えられています。
- Sun ONC RPC を使用する際の不都合な点の 1 つは、Sun 以外のベンダーのソフトウェアがクラッシュを起こさないように、すべての RPC サーバプログラムの番号を Sun に登録する必要があるということです。Vision サーバプログラムの番号は Sun に登録する必要はありません。
- **RPCwiz** を含みます。これは、PC 上で実行され、ONC の **rpcgen** の役割を担います。PC 用または UNIX 用に、クライアントとサーバのスタブをさまざまなプログラム言語で生成できます。通常、プログラマは、クライアントプログラムへのリンク以外はクライアントスタブに対して何も行う必要はありません。サーバスタブには、個別のプロシージャを追加可能なコードへのテンプレートが用意されています。
- **RPCwiz** にサブミットされる **.rpc** ファイルは、ONC の **.x** ファイルに似た機能を持ち、Visual Basic に似た構文を使用します。
- XDR の符号化 / 復号化のメカニズムは、**.rpc** ファイルにあるデータ型を変換するために使用されます。これは Visual Basic の言語固有のデータ型に似ています。

USP ソケットサービス

Vision コミュニケーションアーキテクチャは、パークレイ版 UNIX ソケットに似たプロセス間通信用のソケットメカニズムを提供します。Vision ソケットメカニズムの設計のキーポイントは次のとおりです。

- ソケットは、IP ポートではなく USP アドレスにバインドされます。
- リモートソケット API は、標準ソケット呼び出しすべてをサポートし、ソケットを USP アドレスに関連づける方法を提供します。
- ソケットは、Vision ソフトウェア構成要素だけに使用されます。
- UTP を経由して接続すると、PC 上で受信しているソケットは UNIX 上の **usputppxy** というプロキシプロセスにエクスポートされます。その後、このプロキシプロセスは対応するアドレスに対して受信を行います。

資源の検索（サポート対象外）

Vision コミュニケーションアーキテクチャは、クライアントプログラムがサービスを検索し、サーバプログラムに接続できるようにするメカニズムを提供します。設計のキーポイントは次のとおりです。

- 資源は、汎用命名規則 (UNC) 名を使用して特定されます。UNC によって、下位のトランスポートプロトコルへの依存がなくなります。また、プログラム名を通知せずにサービスへの要求ができます。このプログラム名はホストシステムによって異なります。
- ネームスペース環境 (NSE) は、Vision コミュニケーションアーキテクチャ構成要素によって使用され、UNC を使用したネームスペースマネージャ (NSM) とローカルネームエージェント (LNA) を経由して実装されるネームスペースです。NSE は UNC 名を下位のトランスポートプロトコルの本来のアドレスに解釈します。たとえば、NSE 名 `sun4prtServ` は TCP/IP の 4 バイトの IP (Internet Protocol) アドレスとプリントサーバのポート番号に翻訳されます。
- ネームスペースマネージャは UNIX ホストシステム上で動作し、Vision コミュニケーションアーキテクチャサービスを検索せよというクライアントからの要求に応答します。サービスがローカルに登録されていない場合には、他のネットワークセグメントにあるローカルネームエージェントや NSM に対して要求の処理を依頼できるように、分散型の命名方式を使用しています。複数の NSM によってネットワークの障害許容性が強化されます。
- ローカルネームエージェントは、PC、UNIX ホストなど、X Vision ソフトウェアを実行しているすべてのコンピュータ上で動作させる必要があります。ローカルネームエージェントには、サーバプログラムを起動する責任があります。UNIX では、すべての Vision サーバは LNA によって起動されます。PC では、いくつかのサーバは Windows のスタートアップ時に起動されますが、LNA は要求に応じてサーバを起動させます。

ネームスペース環境

ネームスペース環境 (NSE) は、トランスポートに依存しない一貫したメカニズムによる Vision 資源へのアクセスを提供します。NSE は Uniform Naming Convention (UNC) フォーマットを使用して資源を特定します。UNC によって、クライアントは、独自のネットワークアドレス指定形式には依存しない文字列を使用して、特定のホストシステム上にサービスを要求できます。

NSE はネームスペースマネージャ (NSM) やローカルネームエージェント (LNA) 経由で実装され、これらは下位のトランスポートプロトコル本来のアドレスに UNC 名を解釈します。

ユニフォームネーミングコンベンション

ユニフォームネーミングコンベンションは、ユニバーサルセッションプロトコル (USP) が資源を特定するときに使用します。そのフォーマットは次のとおりです。

`*/hostname*/service`

ここで、*hostname* は Vision コミュニケーションアーキテクチャのホスト名であり、*service* はサービス名、*parameters* はサーバ特有のパラメータです。UNC フォーマット名は、ネームスペース環境によって、ネットワークアドレスとポート番号に解釈されます。

ネームスペースマネージャ

ネームスペースマネージャ (NSM) は、フォルトトレラントであり、複製されたネーミングサービスを提供します。1 つ以上のホスト (UNIX) システムが NSM ホストに指名され、そのホストで `nsmsrv` という NSM プロセスが自動的に起動されます。NSM プロセスはマスターモードまたはバックアップモードのいずれかで実行されます。一度にサブネットワーク上でマスター NSM プロセスになれるのは 1 つだけですが、バックアップ NSM プロセスは多数が同時に動作できます。バックアップ NSM は更新のために時おりマスターのポーリングを行います。マスター NSM はいわゆる「選挙」方式で選択されます。この「選挙」は新しい NSM がオンラインになったとき、またはクライアントが NSM マネージャを検索するときに行われます。NSM の選出は、NSM プロセスの生存期間に基づいて、投票メカニズムの結果によって判断されます。

Vision サーバプロセスは、起動時に NSM への登録を行います。登録は、PC 上のサーバも UNIX ホスト上のサーバも行います。サーバは自身の USP アドレスを登録し、サーバが受信している実際のネットワークアドレスとポート番号も登録します。また、必要とする認証のタイプなど、他の情報も登録します。

ネームスペースマネージャへの RPC インタフェースになるのは、`nsmrpcsrv` という RPC サーバです。

ユーザは、マスター NSM が動作するホストはどれなのか、サーバプロセスが受信するポートはどれなのかを指定できます。

UNIX でのマスター NSM の指定 UNIX ホストに Vision ソフトウェアをインストールすると、そのホストで NSM を実行するのが、(別のホスト上にある) マスター NSM に指定するのが質問されます。マスターとして指定したホストは、`vision.conf` ファイルの `VISION_NAMESVC` エントリに記録されます。エントリのフォーマットは次のようになります。

`VISION_NAMESVC=host.port_number`

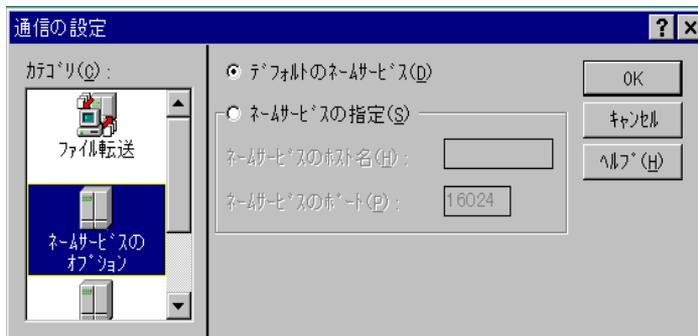
host はマスター NSM が動作している UNIX ホスト名、*port_number* はサーバが受信している IP ポート番号です。次の例を参照してください。

`VISION_NAMESVC=jupiter.16024`

PCでのマスター NSMの指定 PCクライアントは通常はLAN上でブロードキャストすることによってNSMをアクセスしますが、USPトランスポートを構成するときにNSMを指定することもできます。

PCにNSMを指定するには

- 1 Windowsのコントロールパネルで[Vision コミュニケーション]をダブルクリックしてから[トランスポート]タブをクリックします。
- 2 トランスポートリストにある[トランスポート]をダブルクリックします。
- 3 [拡張機能]タブをクリックしてから[設定]をクリックします。
現在選択されている[サービス]が何であるかは関係ありません。
- 4 [カテゴリ]リストで[ネームサービスのオプション]をクリックします。



- 5 [ネームサービスの指定]をクリックします。
- 6 マスターネームサーバが動作しているUNIXホスト名を入力します。
[デフォルトのネームサービス]をクリックすると、PCがNSMに対してブロードキャストを行います。

NSMは指定されても、クライアントがそのNSMにアクセスしようとしても利用できない状態になっているときには、クライアントはマスターNSMにブロードキャストを依頼します。クライアントがマスターNSMからの応答を受け取った場合には、続行することができます。応答が返ってこない場合には、選挙が実行されて新しいマスターNSMが選出され、その新しいマスターNSMがクライアントに応答します。

注意 ホストは変更しないで、[ネームサービスのポート]を変更するだけの場合には、[ネームサービスのホスト名]ボックスにアスタリスク(*)を入力してください。

管理された PC の登録

サービスの登録に加えて、NSM は SuperVision が使用中の場合には PC も登録します。plist というプログラムがセッションのスタート時に管理ホスト上で実行され、NSM でそのホストが管理するすべての PC を登録します。すると、この後変更があるとこのリストは自動的に更新されます。これによって、NSM はどのホストが特定の PC の管理ホストになっているのかをクライアントに通知することができます。

ローカルネームエージェント

ローカルネームエージェント (LNA) は、すべての UNIX システムと PC システム上に存在します。LNA は、ネームスペースマネージャ (NSM) に代わって、ローカルドメイン内部のタスクを遂行します。

LNA の主なタスクは、サーバプログラムを起動することです (「サーバプログラムの起動」参照)。この点では、LNA の機能は Sun ONC RPC システムのポートマップに似ています。サーバの中には LNA が実行された直後に起動するものもありますが、それ以外のサーバはクライアントの要求によって起動されます。

LNA の起動と構成

ローカルネームエージェント (LNA) は他のすべての Vision サーバの起動に責任があります。PC でも UNIX システムでも、Vision ソフトウェアを実行しているコンピュータならどれでもセッションの開始時に LNA を起動させる必要があります。

UNIX で LNA を起動するには

- LNA はシステムのブート時に、システムスタートアップファイルに次のコマンドを含めることで自動的に起動されます。

```
nohup/usr/local/vision/bin/lna-ix
```

-b オプションによって、servers データベースに構成されたブート時サーバを LNA が起動します。

PC で LNA を起動するには

- ローカルネームエージェントは、PC 上の Vision サービスプログラム (Vservice.exe) に組み込まれています。通常、このプログラムは Windows が起動すると自動的に起動されます。

PC で LNA を構成するには

- 1 Windows のコントロールパネルで [Vision サービス] をダブルクリックしてから [LNA] タブをクリックします。
- 2 必要なオプションを選択します。



Windows 2000/Me/98/95 または Windows NT では、タイトルバーで **?** をクリックしてから項目をクリックすると、その項目の [ヘルプ] を得ることができます。Windows NT 3.51 では、SHIFT+F1 を押すとヘルプポインタが表示されるので、項目をクリックします。

サーバの構成

UNIX では、サーバは **servers** データベース内で構成されます。詳細は、付録の「UNIX ファイル形式」を参照してください。

PC では、ローカルネームエージェント (LNA) は Windows レジストリの **HKEY_LOCAL_MACHINE\Software\SCO\Lna\Servers** セクションにあるエントリを参照します。そのフォーマットは次のとおりです。

multiple | single server

ここで、**server** はフルパス名または部分パス名付きのサーバプログラム名であり、**multiple** は複数クライアントを、**single** は単一クライアントを意味します。次の例を参照してください。

```
single,C:\PROGRAM~1\COMMON~1\SYSTEM\VISION\infosrv.exe
```

multiple オプションは、1つのインスタンスが複数のクライアントを扱えることを意味します。このタイプのサーバは、起動されると、セッションの残りのために実行を続けます。

single オプションは、各クライアント用に独立したインスタンスを実行し、そのクライアントとの対話が終了したときにはプロセスが終了することを意味します。

サーバプログラムの起動

サーバプログラムは、クライアントアプリケーションからの要求を扱うために実行されている必要があります。タスクサーバのように頻繁に使用されるサーバは、通常はセッションの起動時に開始され、動作を続けます。あまり頻繁に使用されないサーバは、クライアントがサービスを要求したときにだけ起動されます。UNIX では、すべてのサーバは、セッションの開始時または要求時にローカルネームエージェント (LNA) によって起動されます。PC では、Vision サービスプログラム (**Vservice.exe**) が Windows 起動時にいくつかのサーバを起動し、他のサーバは LNA からの要請があれば起動されます (「Vision サービスプログラム」を参照)。

クライアントがサービスを使用しようとする、ネームスペースマネージャ (NSM) にコンタクトして、ホスト名とサービス名を含む USP アドレスを NSM に渡します。サービスが登録されている場合には、現在サーバが実行されており、NSM がサーバの IP アドレスと、サーバから要求される認証の詳細を戻すことを示します。クライアントはサーバにアクセスする権限を得るために必要な手順を実行します (この章で後述する「認証」を参照)。また、その後、TCP オペレーションを使用してサーバへの接続を試行します。接続できた場合には、クライアントとサーバが互いの対話を行います。

サービスが NSM に登録されていない場合には、NSM は要求のホスト名部分から要求されたサービスのローカルネームエージェント (LNA) を判断します。たとえば、要求されたサービスが ~~✖~~jupiter✖prtServ の場合には、~~✖~~jupiter✖として登録された LNA を検索します。LNA が NSM に登録されていない場合には、クライアントにエラーが戻されます。LNA が登録されている場合には、NSM がサービス要求を LNA に渡します。

LNA は、NSM からのサーバ起動要求を受け取ると、UNIX 上の **servers** データベースまたは PC 上の Registry を検索します。サービスが構成されていない場合には、NSM にエラーを戻し、NSM はクライアントにエラーを戻します。サービスが構成されている場合には、関連するサーバプログラムを LNA が起動します。また、LNA はサーバプロセスの名前を NSM に通知しますが、これはサーバが単一のクライアントまたは複数のクライアントを扱うかどうかによって異なります。サーバプログラムは、起動すると自身を NSM に登録し、NSM はクライアントにアドレス詳細を戻します。

クライアントが NSM から提供される IP アドレスを使用してサーバに接続しようとする、サーバプロセスが強制終了などの理由で、接続に失敗することがあります。この場合には、発生した内容を示すフラグを含めて、NSM に対する別の要求が送信されます。すると、NSM は LNA に対してサーバの再起動を依頼します。

他のネットワーク上にある資源の検索

クライアントはローカルサブネットワーク上だけでしかブロードキャストを行いませんが、ネームスペースマネージャ (NSM) は他の LAN へのゲートウェイになることができます。**vision.conf** 構成ファイルに **VISION_NSNGATEWAY** エントリが存在する場合には (付録の「UNIX ファイル形式」を参照)、このエントリが他のサブネットワーク上の NSM アドレスを示します。ローカルな NSM がこれらのリモート NSM に対して、現在サービスが登録されているホストを定期的に通知します。ローカルの NSM が他のサブネットワーク上にあることを知っているホストに対するサービス要求を受け取った場合には、NSM はその要求をリモート NSM にリダイレクトします。

VISION_NSNGATEWAY エントリのフォーマットは次のとおりです。

hostname.port-number [: ...]

hostname は、リモートのサブネットワークが動作するためにマスター NSM があるホストであり、多くの場合、フルドメイン名として指定されます。*port-number* は NSM が要求を受信するポートの番号です。ポート番号は、たとえば **jupiter.sco.com.16025** のようになります。

ホストには、リモートサブネットワークでマスター NSM を通常実行しているものを指定します。実行時に別のホストがマスター NSM の肩代わりをする可能性もあり、この場合には、ゲートウェイメッセージは、ユーザが指定したホスト上で NSM (バックアップ) が動作していれば、マスターに転送されます。

ゲートウェイメッセージはゲートウェイの連鎖を通して伝達されます。たとえば、サブネットワーク A からサブネットワーク B に、次にサブネットワーク B からサブネットワーク C へと伝達されます。

認証 (サポート対象外)

Vision コミュニケーションセキュリティスキームは、基本的な UNIX のユーザ識別メカニズムを改良するため、また、それを PC をベースにしたサーバに拡張するために設計されています。設計のキーポイントは次のとおりです。

- サービスはそれぞれ認証のタイプを指定し、クライアントは指定された認証を実行してから初めてアクセスを許可されます。
- サービスを使用する際に認可が必要な場合には、クライアントは権限を得るために UNIX ベースの認証サーバにアクセスする必要があります。
- 認証サーバとそのクライアントとの間に渡されるメッセージは、すべて暗号化されます。パスワードがプレーンテキストのまま通信リンク上を送信されることはありません。
- 認証サーバはスーパーユーザ特権があれば暗号化されたパスワードを含むファイルにアクセスできますが、それ以外のサービスはこのファイルにアクセスする必要がないのでスーパーユーザ特権を必要としません。
- UNIX RPC クライアントは認証されたユーザによって PC から起動させることができ、このとき、ユーザ名やパスワードの入力は要求されません。
- PC をベースにしたサーバは、クライアントが管理ホストで認証されてから、アクセスを許可されるように指定できます。
- パーミッション群は、いくつかのサービスへのアクセスだけを許可して各ユーザに割り当てられます。たとえば、サービスの中にはシステム管理者だけが使用できるものもあります。

認証のタイプ

サービスが要求できる認証には、3つのタイプがあります。

- 認証なし
- 直接認証
- 間接認証

プリントサービスのように、任意のユーザがサービスを使用できるのであれば、サーバは認証を指定しません。タスクサービスのようにアクセス制限のある UNIX サービスでは、直接認証が指定されます。PC にはローカルな認証サーバが存在しないので、PC サーバが直接認証を要求することはできません。このため、PC サーバは間接認証を要求します。つまり、クライアントは権限を得るために UNIX 認証ホスト上の認証サーバにアクセスしてから、PC ベースのサーバにアクセスする必要があります。

サーバがネームスペースマネージャ (NSM) にサービスを登録すると、必要な認証のタイプが格納され、クライアントは NSM にこのサービスの検索を依頼したときにこの情報を受け取ります。

認証サーバ

`authsrv`という認証サーバは、UNIX版 Vision サービスを実行するすべてのホスト上で実行する必要があります。このプログラムは、起動すると、ユーザ名とパーミッションのデータベースである `visionpriv` を読み取り、この情報を内部キャッシュに格納します。その後、クライアントプログラムからの要求を待機します。

クライアント認証

認証が動作するしくみは、PC をベースにしたクライアントと UNIX をベースにしたクライアントとは少し違います。PC をベースにしたクライアントは、ホストへのアクセスに必要なユーザ名とパスワードを取得すると、クライアントは常にその情報へのアクセス権を持ち、必要があればサーバにその情報を提供できます。これに対して、UNIX をベースにしたクライアントはユーザ名とパスワードには同じアクセス権を持ちません。

注意 Vision セキュリティは、少なくともインストールされたパーミッションと同じ程度の制限で、残りのシステムファイルへのアクセス権を想定します。

PC クライアントの認証

UNIX をベースにしたサーバは、ネームスペースマネージャ (NSM) でのサービス登録時に直接認証を指定しています。PC クライアントは同じホスト上にある認証サーバから権限を取得してからサーバにアクセスする必要があります。クライアントは、ユーザ名、パスワード、ホスト名、サービス名を含む要求を渡します。認証サーバはユーザ名とパスワードを `/etc/passwd` (または `/etc/shadow`) の内容と照合します。有効である場合には、認証サーバはトークンを作成し、保存して、それをクライアントに戻します。有効ではない場合には、エラーステータスを戻します。トークンにはクライアントが使用を認可されたサービスが指定され、ユーザのアクセス権の設定が含まれています。

PC をベースにしたサーバは NSM によるサービス登録時に間接認証を指定します。すると、NSM はその PC 用の管理ホストを指定します。クライアントが NSM にサービスを依頼すると、クライアント自体を管理ホストで認証してからサーバにアクセスするように命令されます。

認可されたクライアントがサーバとやり取りをするときに、クライアントは認証サーバから取得したトークンを渡します。サーバは認証サーバに呼び出しを行って、そのトークンの妥当性検査を行います。トークンが有効であり、ユーザが正しいパーミッションを持っている場合には、クライアントにサービスへのアクセス権が与えられます。トークンの生存期間は限られているので、クライアント/サーバの会話が一定の時間よりも長く続く場合には、クライアントはもう一度認証サーバにアクセスする必要があります。ただし、トークンは8時間までは“リフレッシュ”されて使用されるので、毎回ユーザ名とパスワードを送信する必要はありません。8時間経過すると、トークンは無効になり、クライアントは新しいトークンを取得する必要があります。

認証サーバとそのクライアントの間でやり取りされるすべてのメッセージは、固有のアルゴリズムによって暗号化されています。このためパスワードは通信リンク間をプレーンテキストのまま送信されることはありません。

UNIX クライアントの認証

UNIX クライアントは、パスワードにアクセスする必要はありません。パスワードはユーザがログインしたときには UNIX システムのどこにも格納されていないためです。クライアントが起動するときにユーザに対してプロンプトを表示することは適切ではないので、別の方式によって認証サーバでクライアントの妥当性検査を行うことが必要になります。

UNIX のクライアントは、まず最初にキーを含む一意の特性を持つファイルを作成します。その後、ファイル名、作成したキー、クライアントの UID、ホスト名、サービス名を含む要求を認証サーバに渡します。認証サーバは複数のチェックを行って、これらの詳細が認証ファイルの内容と一致することを確認し、一致した場合には、クライアントにトークンを発行します。一致しない場合には、エラーステータスを戻します。認証ファイルは認証サーバによって削除されるか、サーバが削除できなかった場合には、クライアントによって削除されます。

これ以降のクライアントとサーバの間のやり取りは、PC をベースにしたクライアントのものと似ています。

パスワードの格納

Vision コミュニケーションアーキテクチャは、Windows 2000/Me/98/95 と類似した方法でパスワードの格納を扱います。ホストなどの特定のオブジェクトはパスワードによって保護されます。パスワードは、次のように指定することができます。

- オブジェクトへアクセス時にプロンプト表示
- オブジェクトと共に永続的に格納
- パスワードリストファイルに格納 (デフォルト)

パスワードリストファイルには、オブジェクトのリストとそれに関連するユーザ名およびパスワードが含まれています。パスワードリストファイルの使用を選択した場合、その PC でユーザ用のパスワードが存在しない場合には、ファイルが作成されます。

パスワードリストにエントリを持たない、保護されたオブジェクトを使用する場合には、ユーザ名とパスワードの入力を要求するプロンプトが表示され、その後、入力されたユーザ名とパスワードは保存されます。将来、そのオブジェクトにアクセスしようとする、Vision コミュニケーションは格納されたパスワードを使用してオペレーションを試行します。パスワードが有効ではなくなった場合には、新しいパスワードでパスワードリストファイルを更新するようにプロンプトが表示されます。

ライセンス管理（サポート対象外）

分散ネットワークで多数の PC 上にソフトウェア製品が使用されていると、それぞれの PC にどんなソフトウェアがインストールされているのかを常に把握するのは難しいことです。このため、Microsoft Windows 上で動作する X Vision 製品は同時使用ユーザ単位でライセンスされ、ソフトウェアがインストールされているコンピュータの数を単位にはしません。たとえば、X Vision の 100 ユーザライセンスを購入すると、同時に 100 人までのユーザが X Vision ソフトウェアを使用できることとなります。ソフトウェアが 100 台を超える PC にインストールされている場合には、ソフトウェアを一度に使用するユーザが実際に 100 人以内であることを確認するのは、購入者の責任となります。

通常、ライセンス契約の方針に責任を持つのはシステム管理者であり、ライセンス管理機能は X Vision ソフトウェアのライセンスを管理するタスクを支援するように設計されています。設計のキープポイントは次のとおりです

- ライセンス管理機能は、システム管理者が使用するツールです。
- ライセンス番号は、通常は PC に X Vision ソフトウェアをインストールするときに入力します。インストール時にライセンス番号を入力しなかった場合には、ソフトウェアの評価版がインストールされ、一定期間だけの使用が許可されます。
- UNIX システムにまだ正しいライセンスが存在していない場合は、UNIX サーバに X Vision ソフトウェアをインストールするときライセンス番号を入力します。
- UNIX をベースにしたライセンスサーバは PC をベースにしたクライアントプログラムが実行できるようにトークンを発行します。
- ライセンスサーバがサブネットワーク上で動作していない場合には、ソフトウェアからライセンス制限を強制されることはありません。システム管理者は、X Vision ソフトウェアが同時使用ユーザライセンスで指定されているユーザ数よりも多くの PC にインストールされていないことを確認するなどの他の手段を使用して、ライセンス契約が破られていないことを確認する必要があります。
- サブネットワークは、ライセンス管理のためにドメインに分割されます。
- X Vision ソフトウェア製品は 1 つ以上のライセンスされた構成要素を含むものとして定義されます。各構成要素は、ライセンス契約で指定されているユーザ数まで使用できます。
- シェルスクリプトは、X Vision プログラムがライセンストークンの取得に失敗したときに UNIX ホスト上で実行されます。ライセンストークンは、システム管理者がライセンスの異常終了イベントを管理するために調整できます。

- UNIX ソフトウェアは異なる単位でライセンスされます。
- ソフトウェアの盗用は、重大な違反であり、ライセンス契約は注意深く監視し、完全に履行される必要があります。

ユーザのドメインへの組織化

サブネットワーク上で利用可能なライセンスの総数は、論理的なドメインに分けられます。たとえば、企業全体のサブネットワークは、営業、製造、経理などに分けられます。システム管理者がドメイン構造のセットアップを選択する理由には、少なくとも次の2つの理由を挙げることができます。

- 部署が独立したビジネス単位として扱われ、部署内のコンピュータ資源の購買と管理の責任を持っている場合。この場合には、システム管理者は部署内のすべてのユーザのユーザ名（つまり PC のノード名）を取得し、ドメインデータベースファイル **domain.dat**（付録「UNIX ファイル形式」参照）を編集して、新しいドメインを作成する必要があります。
- 一般的なライセンスの“プール”の使用にかかわらず、常に一定のユーザがライセンスにアクセスすることを保証したほうがよい場合。これは、これらのユーザの名前をはっきりと指定したドメインを作成することで保証されます。

ライセンス番号

ライセンス番号は、X Vision 製品を取得するときにユーザに提供されます。このライセンス番号は、通常は PC や UNIX システム上に X Vision ソフトウェアをインストールするときに入力する必要があります。

PC では、ライセンス番号はセットアッププログラムによって、製品を構成するすべての X Vision 構成要素用の使用権を判断するために使われます。構成要素は完全なライセンスを受けるか、一定の期間の評価目的で使用するかどちらかです。この情報は Windows レジストリに格納されます。

UNIX システムでは、ライセンス番号は製品に含まれる各構成要素の同時実行ユーザの最大数を判断するために使われます。

製品と構成要素

Microsoft Windows 用のほとんどの X Vision ソフトウェア製品には複数の構成要素があり、これらの構成要素の使用方法がライセンス管理メカニズムによって実際に監視されます。製品とそこにある“ライセンスに対応する”構成要素は、製品情報データベース **prodinfo.dat** に定義されます。たとえば、プログラムの中で、X Vision には、X サーバや VT420 エミュレータがあり、双方ともライセンスに対応しています。ライセンスに対応したプログラムの使用方法は、UNIX をベースにしたライセンスサーバによって発行されるトークンの発行によって制御されます。

ライセンスサーバ

licsrv というライセンスサーバは、マスターのネームスペースマネージャ (NSM) が動作しているホスト上で自動的に起動されます。ライセンスサーバは、クライアントプログラムからの最初の要求を受け取ったときに起動されます。1つのライセンスサーバだけが1つのサブネットワーク上で実行でき、また、他のホスト上のNSMがマスターの役割をするのであれば、ライセンスサーバはそのホスト上で起動され、元のホスト上のライセンスサーバはシャットダウンされます。

licsrv は、起動されたときに次のファイルを読み取ることでライセンスキャッシュを初期化します。

- | | |
|---------------------|--|
| licsrv.dat | このファイルは、ライセンスの使用可能性と現在発行されたトークンの詳細を保持します。 |
| cola.dat | これはライセンスデータベースであり、COLAライセンス番号を保持します。 |
| prodinfo.dat | 製品情報データベースであり、X Vision ソフトウェア製品と、これらの製品を形成する構成要素を定義しています。 |
| domain.dat | ライセンスドメインデータベースであり、ドメインのライセンス指定を定義し、各ドメインに対して、各製品を使用するライセンスされたユーザの最大数を指定しています。 |

これらのファイルは、バックアップ NSM を実行している他のすべてのホストに定期的にコピーされます。

次に、ライセンスサーバは、PC 上で動作しているローカルネームエージェントからの要求を受信します。このローカルネームエージェントは、起動時にライセンスサーバに登録されます。ライセンスサーバは、共に登録されているすべてのPCのLNAリストを保持しています。LNAは、PC上で起動するプログラムからの要求に回答する新しいトークン要求を発行します。また、LNAは、ライセンスサーバによってタイムスタンプが押されたときに一定の間隔でトークンリストを再送信します。ライセンスサーバは定期的に(デフォルトで10分間隔)これらのトークンリストの記録をチェックします。古くなったものがある場合には、ライセンスサーバはそのトークンを解放します。

クライアントの認証

ローカルネームエージェント (LNA) は、PC のライセンス管理を実装する責任があります。LNA だけが UNIX システム上のライセンスサーバに直接コンタクトします。他のプログラムは、LNA を経由して要求を作成します。

LNA は、起動時に通常の USP アドレス指定方式を使用して、ライセンスサーバへの接続を試行します。接続すると、ライセンスサーバに(名前と一意のセッションIDで)PCを登録します。接続に失敗した場合には、PC上のプログラムは“オープン”モードで実行され、ライセンス制限に従わないことになります。LNA がオープンモードで実行されていない場合には、終了時にもう一度ライセンスサーバに接続して、PCの登録を解除します(そして、そのPCに発行されているすべてのトークンを解放します)。

“ライセンスに対応した” PC プログラムは、その構成要素および Vision サービス構成要素も使用できるパーミッションをローカル LNA に依頼するために LNA に要求を送ります。通常は、この要求はプログラムが起動すると送信されますが、X サーバの場合には、最初の X クライアントの起動時に送信されます。LNA はまず最初にこの構成要素に対して使用権についての Windows レジストリ情報をチェックします。構成要素が評価用の権利だけで使用されている場合には、これ以上のライセンスチェックは発生せず、プログラムは評価期間が終了していなければ、制限なく実行できます。構成要素を完全に使用できるパーミッションがある場合には、LNA はプログラム用のライセンストークンを取得する手順に進みます。

LNA は、PC にすでにこの構成要素用のライセンストークンが付与されていることを確認するために LNA のローカルキャッシュをチェックします (同じ構成要素のインスタンスがいくつあっても必要なライセンストークンはひとつだけです)。ローカルでトークンを検出した場合には、リモートライセンスサーバに関係なくプログラムに直接応答します。トークンが見つからなかった場合には、ライセンスサーバにトークンの依頼を行い、トークンを受け取ったときに要求側のプログラムに応答します。

ライセンスに対応したプログラムは、終了すると、LNA に対してそのトークンを解放するように要求します。

LNA は発行された各トークンがどの程度の数のプログラムで“共有”されているのかを保持しています。その数がゼロになった場合には、LNA はライセンスサーバに対して、トークンを開放するように命令します。

LNA はトークンを要求するプログラムのすべてのプロセス ID のリストを保持しています。10 秒おきに、これらのプロセス ID がまだアクティブであるかどうかをチェックします。すでに実行されていないものがあれば、LNA のリストから削除します。これは、完全に終了しないプログラムを扱うための処理です。

LNA は定期的にライセンスサーバに接続して、自身のトークンリストをライセンスサーバに送信します。これによって、ライセンスサーバは PC が活動中かどうかをチェックし、また、ライセンスサーバが再起動された場合にはそのトークンリストを活動中の PC から再配置することが保証されます。

ライセンス要求の異常終了イベントの管理

クライアントがライセンストークンを要求し、それが拒否された場合には、シェルスクリプト `$VISION_ETC/DIR/license_event` がライセンスサーバを動作している UNIX ホスト上で実行されます。初期インストール時には、このスクリプトは何も行いません。システム管理者は、このスクリプトを変更して、ライセンス要求の異常終了イベントで何らかの動作を行うこともできます。たとえば、コンソールにメッセージを表示したり、管理者自身に電子メールメッセージを送ることなどが考えられます。

イベントとエラーのログ収集

PC 上と UNIX 上にある X Vision 製品は、ログファイルにデバッグメッセージを書き込み、問題解決やネットワークパフォーマンスの監視に使用することができます。これらのメッセージは、弊社のテクニカルサポートだけにとって意味があることが多く、通常はサポートエンジニアからデバッグのログ収集の依頼を受けたときにだけ収集を行います。UNIX では、より高いレベルのイベントログ収集機能が提供されており、これはデフォルトで使用可能になっており、システム管理者が使用できます。

注意 デバッグログファイルは、非常に大きくなり、ディスクスペースを使い果たしてしまう可能性があります。ログ収集も X Vision ソフトウェアのパフォーマンスにかなり影響を与えるので、特定の問題を解決する時に短期間使用する以外は、デバッグログの収集は有効にはしないでください。

PC でのログ収集

Windows のコントロールパネルを使用して、デバッグログ収集の有効または無効を指定できます。

デバッグを有効にするには

- 1 Windows のコントロールパネルで、[Vision コミュニケーション] をダブルクリックしてから [診断] タブをクリックします。



- 2 [診断を行う] ボックスをクリックします。チェックマークが表示されます。

- 3 ボックスをクリックして、デバッグ情報のタイプを選択します。または [全て設定] をクリックします。
- 4 [更新] または [OK] をクリックします。

デバッグ情報は `C:\Vwcomms.txt` というファイルに書き込まれます。エントリのフォーマットは次のようになります。

tick_count : component : description

次にサンプルを示します。

```
18144850 : NMUSPT32 : findNS (bcast = 0)
18144850 : NMUSPT32 : presetNS (10252592, 0)
18144904 : NMUSPT32 : usptcp_GetNSCache (192.86.169.164, 16024)
18144904 : NMUSPT32 : presetNS: calling GetInetAddr (192.86.169.164)
18144904 : NMUSPT32 : presetNS: returning 192.86.169.164 (16024)
18144957 : NMUSPT32 : UspMapIO: nl_connect (192.86.169.164 (16024)) errno 35
18144957 : NMUSPT32 : UspMapIO: awaiting connection completion
18145149 : VWC32 : NlTerminate : Called for protocol 1
18145149 : VWC32 : Free stack for 1 ref 61
```

UNIX でのログ収集

UNIX でのログ収集サービスは、`logsrv` サーバによって提供されます。このサーバは、別々のファイルに次の 2 種類の情報を書き込むことができます。

- システムイベント
- デバッグメッセージ

イベントのログ収集

重要なシステムイベントは、通常は `vweventlog` というファイルに収集されます (付録の「UNIX ファイル形式」を参照)。また、システム管理者は、テキストエディタを使用してこのファイルの内容を確認することができます。SuperVision が使用中ではないかぎり、エラーだけがログに書き込まれます。

イベントログ収集は、`/etc/vision.conf` ファイル内にある `VISION_EVENTLEVEL` 変数を使用して有効や無効を指定できます。このとき、0 は無効、1 は有効の指定です。

デバッグ情報のログ収集

デバッグ情報は、製品サポートエンジニアにとって有用なので、ユーザは依頼されない限り、通常はログ収集を行わないでください。デバッグは、多くの Vison Family プログラムでは、プログラム起動時に `-d mask` オプションを指定することで有効になります。この `mask` は、ファイルに書き込む情報の量とタイプを制御するプログラムに固有な数値マスクです。

デバッグメッセージのログ収集には、`vwdebuglog` ファイルが使用されます (付録の「UNIX ファイル形式」を参照)。

(空白のページです)

第4章

Xサーバの使い方

この章の内容は次のとおりです。

- Xサーバの概要
- 起動と終了
- サーバの構成
- サーバのプロファイル
- ディスプレイモードの選択
- ウィンドウの管理
- アクティブなトランスポートの選択
- パフォーマンスの最適化
- セキュリティ
- キーボードの使い方
- マウスの使い方
- フォントの使い方
- カラーの使い方
- コピーアンドペースト
- サウンドの扱い方
- HelpfulInteractiveSupport
- Xエラーのログ
- Xサーバに関する情報の検索
- Windowsの日本語入力を使用するには

X サーバの概要

X Vision の X サーバは、Microsoft Windows 環境で動作する X ウィンドウシステムのサーバです。主な機能の概要は次のとおりです。

- X バージョン 11、リリース 6 のサンプルサーバに基づきます。ただし、Microsoft Windows 環境において最高のパフォーマンスを実現するために一部変更を加えています。
- X サーバは、Windows 2000/Me/98/95 または Windows NT で動作するように設計された 32 ビットアプリケーションです。
- X クライアントと Microsoft Windows のアプリケーションを同時に同じ画面に表示させることができます。
- X サーバは**スマートサーバ**です。たとえば、使用するシステムを自動認識、自動設定します。クライアントを起動するための最適な方法を選択します。また、フォントを置換したり、アクティブでない X クライアントでも正しいカラーで表示できます。さらに、クライアントで障害が発生した場合、ヘルプを提示します。
- X サーバの構成および実行に関する情報が格納されたプロファイルを作成できます。プロファイルの数はいくつでも好きなだけ設定でき、一度に複数のプロファイルを実行できます。このため、複数の X セッションを同時に実行できます。
- X サーバはマルチウィンドウモードまたはシングルウィンドウモードで使用できます。マルチウィンドウモードでは、X クライアントはそれぞれ専用用意された Microsoft ウィンドウ内で実行され、X ルートウィンドウを Microsoft Windows 背景として表示することもできます。このモードでは、X アプリケーションに Microsoft Windows のルックアンドフィールを提供します。シングルウィンドウモードでは、X サーバウィンドウに X ルートウィンドウが常に表示されます。X クライアントはすべて、X サーバウィンドウ内の子ウィンドウに表示されます。このモードでは、標準的な X ディスプレイのルックアンドフィールを提供します。サーバウィンドウは Microsoft Windows の他のアプリケーションと画面を共有でき、通常の移動、サイズ変更、アイコン化が可能です。
- マルチウィンドウモードでは、X クライアントのウィンドウ管理は、組み込まれている VWM ウィンドウマネージャ (ICCCM に準拠) を使用してローカルに行われます。サーバがシングルウィンドウモードの場合は、所定のローカルな Motif ウィンドウマネージャまたはホストに基づいた任意の X ウィンドウマネージャを使用できます。ウィンドウ管理をローカルで行うと、ネットワークトラフィックとホストプロセッサに対する負荷が減り、パフォーマンスが向上します。
- マルチウィンドウモードでは、X サーバは**透過的**、すなわち非表示で実行できます。透過的に実行している場合、Windows のセッションが開始されると X サーバが自動的に起動されます。
- パニングおよびスクロールメカニズムによって、実画面よりも大きな表示領域を必要とするクライアントでも実行できます。

- XクライアントとMicrosoft Windowsアプリケーション間で、テキストやグラフィックをコピーアンドペーストできます。
- Xサーバの拡張機能 SHAPE、XTEST、XIE によって、矩形以外のウィンドウ、X Test Suite、および X Image Extension もサポートされます。
- Xサーバ独自の拡張機能によって、Vision Resume がサポートされているため、Xセッションを一度中断し、後で再開することが可能です。
- Xサーバでは、Microsoft Windowsの .fon 形式にコンパイルされているXフォント、およびXの .pcf 形式のフォントを表示できます。Xウィンドウシステムの標準的なフォントはコンパイルされた形で提供されています。また、Font Compiler が用意されているため、サードパーティのフォントもコンパイルできます。さらに、サーバはMicrosoft Windowsで一般的に使用されるフォントをXアプリケーションに提供できます。ただし、日本語フォントのコンバートは行えません。
- X Font Server Protocol がサポートされているため、個々のPC上ではなくネットワーク上のフォントサーバにフォントを格納しておくことができます。
- フォントサーバのフォントに対してグリフロード遅延機能がサポートされているため、使用頻度の低いサイズの大きなアジアフォントにおけるパフォーマンスが向上します。
- クライアントで要求されたフォントが使用できない場合、フォント置換メカニズムで代替フォントを提供できます。
- Xサーバでは、2ボタンマウスで3ボタンマウスの機能を提供します。
- ローカルなバッキングストアによって、画面の高速更新が実現されます。また、バッキングストアが有効な場合、高速にテキストをスクロールするための特別なオプションが提供されます。
- 独自のグラフィックススピードオプティマイザ機能によって、使用するディスプレイハードウェアの最高の速度に合わせてXサーバを調整します。
- ホストアクセス制御およびユーザアクセス制御のサポートによって、セキュリティが保証されます。
- MIT-MAGIC-COOKIE およびXDM-AUTHORIZATION-1 セキュリティモードなどのXDMCPをサポートします。
- Xサーバでは、24ビットのカラーディスプレイなどWindowsでサポートされているすべてのグラフィックディスプレイを使用できます。
- 多くの種類の言語およびキーボードをサポートします。
- クライアントからネットワークまたはシリアル回線を介してXサーバにアクセスできます。NCDのXRemoteプロトコルを使用すれば、シリアル回線をサポートできます。
- Microsoft Windowsの標準的なメカニズムを使用したオンラインヘルプが提供されています。
- HINTS (Helpful Interactive Support) によって、クライアントの起動後エラーが発生した場合にヘルプを提示します。

制限事項

X サーバは、X コンソーシアムによって配布されている X11R6 サンプルサーバを実装したものです。ただし、制限事項がいくつかあるため、注意してください。この制限事項の多くについては、以降で説明します。

次のような制限事項があります。

- PEX はサポートしません。
- X Input Device Extension はサポートしません。
- 認証サービスの SUN-DES-1 はサポートしません。
- カーソルのサイズは Microsoft Windows の制限値である 32x32 に制限されます。
- キーボードクリック、オートリピート、LED やベルのボリュームや間隔は設定できません。
- マルチウィンドウモードでは、ルートウィンドウへのキーボード入力はできません。

起動と終了

このセクションの内容は次のとおりです。

- X サーバの起動
- サーバの手動クローズ
- Xdm のメインプログラムのクローズ

X サーバの起動

X サーバは通常 Windows が起動されると自動的に起動されます。X サーバを自動的に起動しないようにするには、Windows のコントロールパネルで、[X Vision プロファイル] をダブルクリックし、[共通] タブをクリックして、[Windows 起動時に X Vision をロードする] オプションを無効にします。また、プログラムスタータを使用して X クライアントを実行するときに X サーバが実行中でない場合にも、X サーバは自動起動されます。

X サーバを手動で起動するには

以下のいずれかを行います。

- [スタート] ボタンをクリックし、[プログラム]、[CentreNET X Vision] を順にポイントして、[X Vision サーバ] をクリックします。
- Windows NT 3.51 の場合は、プログラムマネージャで [CentreNET X Vision] グループを開き、[X Vision サーバ] アイコンをダブルクリックします。

サーバが起動し、非表示で実行するように構成されている場合は、Windows 2000/Me/98/95 または Windows NT 4.0 のタスクバーの通知領域にインジケータが表示されます。



構成を変更してある場合、Xサーバはタスクバー上のアイコンまたはウィンドウとして表示されることがあります。

プロファイルを設定し実行すれば、一度に複数のXサーバのインスタンスを実行できます。プロファイルの詳細については、本章の「Xサーバのプロファイル」を参照してください。ただし、マルチウィンドウモードでは1つのXサーバしか実行できず、実行中の各Xサーバには一意のXディスプレイ番号が必要です。ディスプレイ番号は、[サーバプロパティ] ダイアログボックスで変更できます。

Xサーバの手動クローズ

Xサーバを手動で閉じるには、以下の手順に従います。

Xサーバを閉じるには

- 1 各アプリケーションで推奨されている終了手順に従って、Xクライアントをすべて閉じます。
- 2 以下のいずれかを行います。
 - Xサーバが非表示で実行中であれば、マウスの右ボタンでタスクバーの通知領域にあるXサーバのアイコンをクリックしてから、[閉じる]をクリックします (Windows NT 3.51 の場合は、ルートメニューを表示してから、[閉じる]をクリックします)。
 - Xサーバが非表示でない場合は、マウスの右ボタンでタスクバーのXサーバのアイコンをクリックしてから、[閉じる]をクリックします。
 - Xサーバがシングルウィンドウモードで動作している場合は、[コントロール]メニューをクリックしてから、[閉じる]をクリックします。Xサーバがフルスクリーンモードの場合は、ALT+SPACE を押して、[コントロール]メニューを表示します。
- 3 セッションを終了することを確認します。

起動中または動作中のXクライアントは自動的に終了され、警告は表示されません。

プログラムが終了すると、Xサーバウィンドウが画面から削除されます。

xdm のメインプログラムのクローズ

xdm ディスプレイマネージャを使用している場合、“メインプログラム”を閉じると、それ以外のすべてのクライアントが終了されます。ログアウトし、xdm との接続は閉じられます。XDMCP を使用し、[最初のマネージャを選択] オプションが有効な場合は、X サーバが終了します。これ以外の場合は、現在の XDMCP モードの設定に応じて、X サーバは新しいセッションを開始します。

X サーバの構成

X ウィンドウシステムは、サイズやアーキテクチャが異なる多くのコンピュータ上で使用できるように、可能な限り柔軟な設計になっています。このため、X サーバは X のさまざまな使い方に対応できなければならず、さらに Microsoft Windows で提供される機能ができる限り使用できるようになっていなければなりません。

X サーバを構成するためのオプションは多く用意されており、インストール時にセットアッププログラムが使用する PC に合った設定を選択します。このため、デフォルトの設定を変更する必要はほとんどありませんが、オプションを変更する場合は、実行中の X サーバから行うか、または Windows のコントロールパネルでプロファイルを構成します。詳細については、本章の「サーバのプロファイル」を参照してください。

アクティブ X サーバを構成するには

以下のいずれかを行います。

- X サーバが非表示で実行中であれば、マウスの右ボタンでタスクバーの通知領域にある X サーバのアイコンをクリックしてから、[プロパティ]をクリックします (Windows NT 3.51 の場合は、ルートメニューを表示してから、[プロパティ]をクリックします)。
- X サーバが非表示でない場合は、マウスの右ボタンでタスクバーの X サーバのアイコンをクリックしてから、[プロパティ]をクリックします。
- X サーバがシングルウィンドウモードで動作している場合は、[コントロール]メニューをクリックしてから、[プロパティ]をクリックします。X サーバがフルスクリーンモードの場合は、ALT+SPACE を押して、[コントロール]メニューを表示します。

ダイアログボックスが表示され、タイトルバーに現在のプロファイル名が示されます。タブを使用すれば、次のサーバの属性に移動できます。変更内容を保存し、[プロパティ]ダイアログボックスを閉じるには、[OK] をクリックします。[プロパティ]ダイアログボックスを閉じる前に変更を有効にする場合は、[変更] をクリックしてください。

オプションによっては、別のオプションを選択した場合に限り、使用できるものがあります。使用できないオプションは、ダイアログボックス中にグレー表示されます。グレー表示されているオプションが再度使用できるようになると、あらかじめ指定されているデフォルトの設定が表示されます。

ほとんどの変更は直ちに有効になりますが、ディスプレイモードの変更やトランスポートの変更など一部の変更ではXサーバを再起動する必要があります。Xサーバの再起動が必要な場合は、Xサーバを直ちに再起動するかどうか確認を求めるダイアログボックスが表示されます。Xサーバを直ちに再起動する場合は、[はい] ボタンを選択します。アクティブなクライアントがある場合には、[いいえ] を選択して、クライアントを閉じてから、サーバを再起動します。

Xサーバを再起動するには

以下のいずれかを行います。

- Xサーバが非表示で実行中であれば、マウスの右ボタンでタスクバーにあるXサーバのアイコンをクリックしてから、[再起動] をクリックします (Windows NT 3.51 の場合は、メニューを表示してから、[再起動] をクリックします)。
- Xサーバが非表示でない場合は、マウスの右ボタンでタスクバーのXサーバのアイコンをクリックしてから、[再起動] をクリックします。
- Xサーバがシングルウィンドウモードで動作している場合は、[コントロール] メニューをクリックしてから、[再起動] をクリックします。サーバがフルスクリーンモードの場合は、ALT+SPACE を押して、[コントロール] メニューを表示します。

Xサーバのプロファイル

プロファイルとはXサーバで使用される構成オプションの集まりで、Xクライアントの表示方法やXサーバの実行および動作方法に関する情報が格納されています。サーバを起動すると、自動的またはユーザが明示的に指定することによって、プロファイルが実行されます。ネットワークXサーバとシリアルラインXサーバのどちらをインストールしているかによって、X Vision か XRemote のいずれかがデフォルトのプロファイルになります。

Xサーバのプロファイルによって、さまざまなXサーバの構成を保存しておくことができる上、同時に複数の構成を実行できます。プロファイルは非表示ファイルとして拡張子 .xfg を付け、**¥Vision¥User¥Profiles** フォルダに保存します。プロファイルの追加、削除、構成は、[X Vision プロファイル] ダイアログボックスの [メイン] タブを使用します。

プロファイルを作成するには

- 1 以下のいずれかを行います。
 - [スタート] ボタンをクリックし、[設定] をポイントしてから、コントロールパネルをクリックします。
 - または-
 - Windows NT 3.51 の場合は、プログラムマネージャでメイングループを開き、コントロールパネルのアイコンをダブルクリックします。

- 2 [X Vision プロファイル] をダブルクリックします。
- 3 [追加] をクリックし、所定の位置にプロファイル名を入力します。
- 4 プロファイルを構成するには、リストから目的のプロファイルを選択し、[プロパティ] をクリックします。

クライアントを Microsoft のマルチウィンドウとシングルウィンドウのどちらに表示するのか、およびサーバの拡張機能、リソースの最適化、トランスポート、セキュリティメカニズムなどの X クライアントのパフォーマンスや表示を左右するオプション、クリップボードのオプションなどの情報を指定できます。必要なオプションをすべて設定したら、[OK] をクリックします。

X サーバは、起動時に常に特定のプロファイルを実行するように構成できます。[X Vision プロファイル] ダイアログボックスの [一般] タブで、[常に使用] をクリックし、ドロップダウンリストから目的のプロファイル名を選択します。X サーバの起動時にどのプロファイルを実行するかユーザに指定を求めるようにする場合は、[使用するプロファイルを尋ねる] をクリックします。

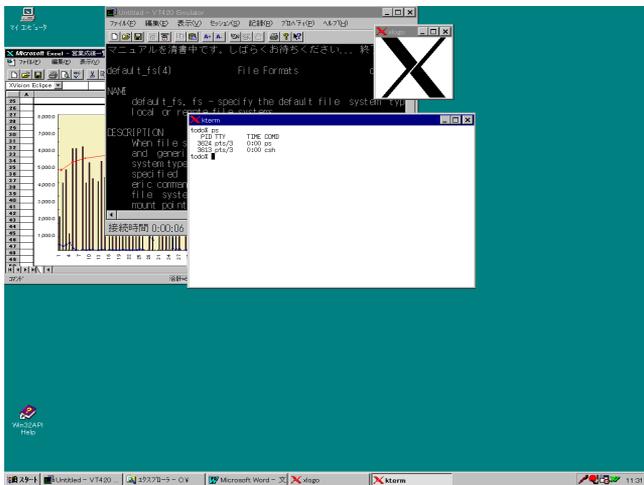
注意 同時に 2 つのサーバセッションを実行するには、X ディスプレイ番号の異なる 2 種類のプロファイルを作成する必要があります。マルチウィンドウモードでは 1 つの X サーバしか実行できませんが、シングルウィンドウモードではいくつでも X サーバを実行できます。

共通設定

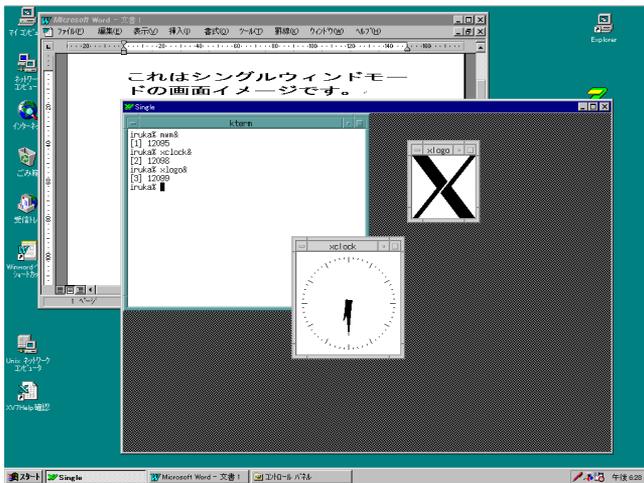
プロファイルの一部として設定可能な属性だけでなく、すべてのプロファイルに適用されるグローバル設定があります。このグローバル設定としては、Windows の起動時に自動的に X サーバを起動するかどうかの指定、カラーデータベースファイル名の変更、XDMCP セキュリティやホストアクセス制御セキュリティの基本的なオプションの設定などがあります。

ディスプレイモードの選択

X サーバは、マルチウィンドウモードとシングルウィンドウモードの 2 つのモードで動作が可能です。デフォルトのマルチウィンドウモードでは、各 X クライアントがそれぞれ専用に用意された Microsoft のウィンドウに表示され、X サーバに組み込まれている Vision ウィンドウマネージャ でウィンドウが管理されます。これによって、X クライアントで Microsoft Windows のロックアンドフィールが実現されます。



OPEN LOOK や DECwindows などの別のウィンドウマネージャで定義されているスタイルが必要な場合は、シングルウィンドウモードで X サーバを使用することも可能です。シングルウィンドウモードでは、PC は標準的な X ディスプレイのようになり、X クライアントはすべてメインの X サーバウィンドウ内に表示されます。ウィンドウマネージャは、通常の X ウィンドウシステムと同じように独立した X クライアントとして機能します。ディスプレイモードの切り換えは、[ディスプレイプロパティ] ダイアログボックスで行います。



ウィンドウの管理

Xクライアントのルックアンドフィールを Microsoft Windows と同様にしたい場合は、Xサーバに組み込まれている Vision ウィンドウマネージャ (VWM) を使用します。このウィンドウマネージャは、サーバがマルチウィンドウモードの場合に動作します。また、VWMはXクライアントでMicrosoft Windowsのルックアンドフィールを実現するように設計されている Motif のウィンドウマネージャ MWM と互換性があります。

MWM ウィンドウマネージャ自体は、Xサーバがシングルウィンドウモードであれば、ローカルな Motif ウィンドウマネージャを使用できます。このウィンドウマネージャは、特別なローカルのトランスポートを使用して、ウィンドウマネージャとサーバ間でメッセージの受け渡しを行うため、TCP/IPをPCにインストールする必要はありません。さらに、OPEN LOOK のウィンドウマネージャである OLWM などのホストベースのウィンドウマネージャも使用できます。

詳細については、「Xウィンドウの管理」の章を参照してください。

アクティブなトランスポートの選択

Xサーバは、TCP/IPまたはIPX/SPXプロトコルを使用してネットワークを介してリモートホストシステムと交信するだけでなく、NCDのXRemoteを使用してシリアル回線を介してもリモートホストシステムと交信できます。ネットワークとシリアルトランスポートでは異なるサーバプログラムが使用されるため、Xサーバのプロファイルの構成時にどちらのサーバを使用するか指定しなければなりません。

注意 シリアル回線Xサーバをインストールするには、[カスタムセットアップ]を使用します。どちらか一方のXサーバだけがインストールされている場合は、トランスポートを選択するためのオプションは使用できません。

アクティブなトランスポートを選択するには

- 1 Xサーバの[プロパティ]ダイアログボックスで、[トランスポート]タブをクリックします。
- 2 Xサーバをネットワークを介して使用する場合は、[ネットワークでX Visionを使用]がチェックされているか確かめます。

- または -

Xサーバをシリアル回線またはモデムを介して使用する場合は、[シリアルラインでX Visionを使用]をクリックします。XRemoteプロキシサーバを実行しているUNIXホスト名を所定の位置に指定します。

パフォーマンスの最適化

Xサーバは、正しく構成された Microsoft Windows のコンピュータ上で最速のパフォーマンスを実現することを目的としています。このため、XサーバのベースとなるXコンソーシアムのサンプルサーバの一部が、Windowsで最速の動作を実現し、PCグラフィックディスプレイハードウェアを利用できるように変更されています。グラフィックの最適なパフォーマンスを実現するために、以下の機能を使用できます。

- グラフィックスピードオプティマイザ
- グラフィックキャッシュ

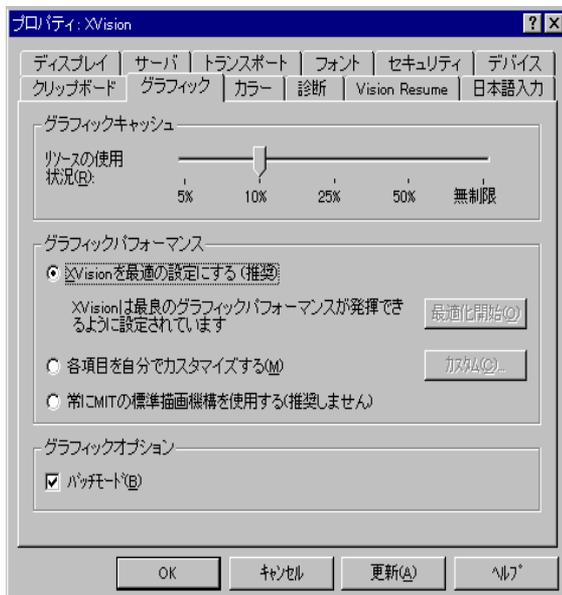
グラフィックスピードオプティマイザ

XサーバはX描画操作を同じようなMicrosoft Windows機能に変換します。XとMicrosoft Windows間では全く同じ機能もありますが、操作によっては、X機能に直接対応する機能がMicrosoft Windowsになかったり、Microsoft Windowsの対応機能がXの仕様に合わないことがあります。このような場合、Xサーバを最も近いMicrosoft Windowsの対応機能を使用するように設定したり(Windowsモード)、Xルーチンを使用するように設定したり(コレクトモード)、またはカスタムモードを使用して最速のパフォーマンスを実現するように設定できます。ただし、カスタムモードの場合、実際には特定のディスプレイハードウェア上で最速の操作を実現できないこともあります。

常にピクセル上完全なイメージを生成する必要がある場合は、コレクトモードを使用するようにサーバを構成してください。最速のパフォーマンスを実現したい場合は、グラフィックスピードオプティマイザ(GSO)を実行して、個々のディスプレイで最速のパフォーマンスを実現するようにサーバを調整できます。この場合、一連のテストを行い、数種類の描画操作に対して使用しているディスプレイで3つの描画モードのうちどれが最も高速かが判断されます。GSOは通常設定の一部として実行されますが、必要に応じて手動で実行することも可能です。PCの構成を変更してある場合は、GSOを実行して、Xサーバが最適なパフォーマンスを実現しているか確かめてください。GSOは、個々の描画操作に対して実行することも、またはすべての操作に対して実行することもできます。ただし、すべての操作に対して実行する場合は、実行が終了するまでに数分かかります。

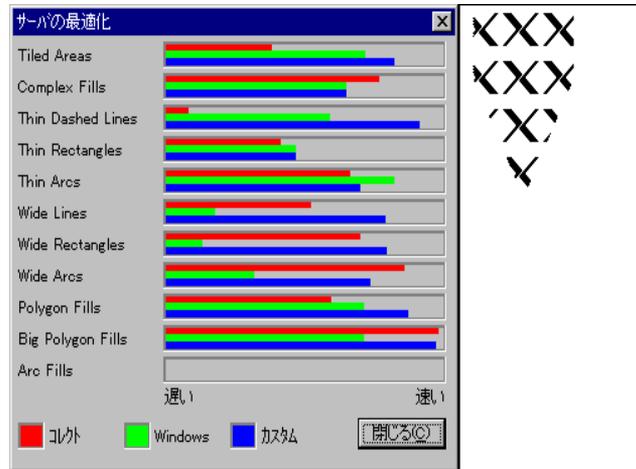
グラフィックスピードオプティマイザを実行するには

- 1 Xサーバの[プロパティ]ダイアログボックスで、[グラフィックス]タブをクリックします。



- 2 すべての描画操作を最適化する場合は、[最適化開始] をクリックします。
- 3 個々の描画操作を最適化する場合は、[各項目を自分でカスタマイズする] をクリックした後、[カスタマイズ] をクリックします。
- 4 最適化したい操作をドロップダウンリストから選択して、[最適化] をクリックします。選択した操作に対して、X サーバが最も高速なモードを選択します。

ダイアログボックスが拡大され、ベンチマークディスプレイに3つのそれぞれのモードで最適化対象の操作がテストされていることが示されます。[最適化開始] を選択した場合、[サーバ最適化] ダイアログボックスが表示され、描画操作の最適化結果が示されます。バーが長いほど、操作は高速です。最適化プロセスを中止する場合は、ESCAPE キーを押します。



結果を参照したら、[閉じる]をクリックすると、ダイアログボックスが通常の表示に戻ります。

グラフィックキャッシュ

Xサーバではグラフィックキャッシュを使用して、ブラシ、ペン、ビットマップなどのリソースを格納します。これによって、リソースを作成および削除する回数を減らすことができます。また、リソースの共有化を図ることができます。リソースの共有化は、Microsoft Windows においては、リソースの数がシステム全体で制限されているため重要になります。

グラフィックキャッシュに使用される Windows のリソースの数を変更するには

- 1 Xサーバの[プロパティ]ダイアログボックスで、[グラフィック]タブをクリックします。
- 2 [グラフィックキャッシュ]で、スライダをドラッグします。

大きな値を選択すると、Windowsのパフォーマンスが結果的に落ちることがあるため、Xサーバのパフォーマンスが向上しないことがあります。

[X Vision について]ダイアログボックスに、現在Xサーバで使用されているシステムリソースの割合が表示されます。

セキュリティ

X モデルでは、クライアントとサーバ間の接続はクライアントが開始します。場合によっては、サーバへのアクセスを制限したり、不要なクライアントが画面に表示されないようにする必要があります。このため、X ウィンドウシステムでは、ディスプレイサーバへのアクセスを制御するための方法がいくつか提供されています。ホストアクセス制御方式では、指定されたホストで実行中のクライアントへのアクセスを制限します。ホストアクセス制御が無効の場合は、どのクライアントでも PC 上で表示させることができます。これに対し、ユーザアクセス制御は、クライアントを実行する前にサーバとクライアントの両方が認識していなければならない認証コードを使用するため、よりセキュリティが保護されます。ユーザアクセス制御は一般的に XDMCP の使用時に一般的に使用されますが、ローカルな認証ファイルも使用できます。

ホストアクセス制御の使い方

ホストアクセス制御方式では、指定されたホストで実行中のクライアントに対するディスプレイのアクセスが制限されます。認証ホストのリストは、デフォルトでは `Xhosts.txt` と呼ばれる PC 上のファイルに格納されます。X サーバが起動すると、ホスト名が内部テーブルにロードされます。

さらに XDMCP または Local Authorization File が有効の場合、そのクライアントが実行中のホストが認証ホストのリストに含まれているかどうかに関係なく、正しい認証コードを提示すれば、どのクライアントでも実行できます。

ホストアクセス制御を有効にするには

- 1 X サーバの [プロパティ] ダイアログボックスで、[セキュリティ] タブをクリックします。
- 2 [ホストアクセスコントロール] をクリックします。
[X Vision プロファイル] ダイアログボックスの [共通] タブで指定した認証ホストが一覧表示されます。
- 3 特別なホストを追加するには、所定の位置にホスト名を入力するか、またはドロップダウンリストから目的のホストを選択します。

XDMCP セキュリティの使い方

X サーバでは、サーバとディスプレイマネージャ間の対話を規定した XDMCP (X Display Management Control Protocol) をサポートします。XDMCP が有効であれば、`xdm` などのディスプレイマネージャを使用して、X セッションを開始できます。詳細については、ヘルプのキーワードの「X ディスプレイマネージャの使い方」を参照してください。

XDMCP の使用中は、ユーザアクセス制御方式を使用して、クライアントからサーバへのアクセスを制限できます。ログインすると、サーバに認証コードが付けられます。そのディスプレイ管理セッションに属するクライアントは、それ以降サーバと通信をとる際同じコードを渡すため、実行が可能です。この認

証コードを提示しないクライアントは実行できません。認証されたクライアントが実行中であるホストは対象外です。なお、認証コードを使用するには、アプリケーションをコンパイルしておく必要があります(詳細については、アプリケーションのサプライヤに問い合わせてください)。X11R4以降の正しいライブラリでコンパイルしていないアプリケーションは、実行されません。

MIT-MAGIC-COOKIE-1とXDM-AUTHORIZATION-1の2つのレベルのユーザアクセス制御がサポートされています。MIT-MAGIC-COOKIE-1セキュリティを使用すると、認証コード(“magic cookie”)がそのままの文字列としてサーバとクライアントの間で受け渡されます。このため、スヌーププログラムで認証コードを読み取ることができます。これに対して、XDM-AUTHORIZATION-1メカニズムはよりセキュリティレベルが高く、認証コードを一度暗号化してから転送します。暗号化および復号化は、[X Vision プロファイル] ダイアログボックスの[共通] タブの[ディスプレイ ID]と[キー]の値に基づきます。この2つの値はまた通常 /usr/lib/X11/xdm に格納されているホストファイル **xdm-keys** にも登録されています。XDM-AUTHORIZATION-1メカニズムが使用できるのは、X11R5以降です。X11R5以前のクライアントでこのセキュリティメカニズムを使用すると、XDM-AUTHORIZATION-1メカニズムの代わりに MIT-MAGIC-COOKIE-1セキュリティが暗黙に使用されます。

ユーザアクセス制御が機能するには、ホストアクセス制御も有効でなければなりません。[ホスト] リストに含まれるホストはサーバですべてのクライアントを起動できるため、認証コードを示すクライアントへのアクセスを制限するために、ホストアクセス制御を有効にし、使用しているPC以外のホストをリストに入れないようにします。ホストアクセス制御が無効であれば、すべてのホストでクライアントを表示できます。なお、XDMCPが有効である場合にX11R3クライアントを実行するには、[ホスト] リストにそのクライアントが実行しているホストを入れる必要があります。

XDMCPが有効でなくても、次のセクションに示すように、ローカルな認証ファイルを作成すれば、ユーザアクセス制御を使用できます。ただし、どちらの方式においてもホスト上の **.Xauthority** ファイルを使用するため、両方の方式を同時に使用することはできません。詳細については、本章の「ローカルな認証ファイルの使い方」を参照してください。

XDMCP ベースのセキュリティを有効にするには

- 1 Xサーバの[プロパティ]ダイアログボックスで、[セキュリティ]タブをクリックします。
- 2 XDMCPがチェックボックスで有効になっているかどうか確かめます。
- 3 XDMCPで、必要なモードをクリックします。マウスの右ボタンでクリックした後、[ヘルプ]をクリックすると、そのモードの詳細情報が表示されません。

[ダイレクト]モードまたは[インダイレクト]モードを使用している場合は、[ホスト]ボックスに使用するホスト名を入力するか、またはドロップダウンリストから目的のホストを選択します。

- 4 XDM-AUTHORIZATION-1 セキュリティメカニズムを使用する場合は、[X Vision プロファイル] ダイアログボックスの [共通] タブの [キー] ボックスにキーを指定します。

キーオプションの値は、システム管理者から入手してください。システム管理者は、キーオプションの値の他、[ディスプレイ ID] の値も必要です。一度登録された値は変更できません。[キー] ボックスが空のままの場合は、MIT-MAGIC-COOKIE-1 セキュリティメカニズムが使用されます。

- 5 他にも必要なオプションがある場合は、[拡張設定] をクリックして指定します。

ローカルな認証ファイルの使い方

XDMCP が無効であっても、ローカルな認証ファイルをホストシステムと PC に設定すれば、ユーザアクセス制御を使用できます。このファイルには、X11R4 以降のクライアントをサーバで実行するために必要な認証コードが格納されます。

ホストシステムで認証ファイルを作成するには、X11R4 以降で配布される **xauth** プログラムを使用します。デフォルトでは、このファイルは **.Xauthority** という名前で、ユーザの **/home** ディレクトリに格納されています。Unix ネットワークコンピュータを使用して、このファイルを PC に転送し、必要に応じて簡易ファイル名の取り決めに従い、ファイル名を変更します。クライアントはこのファイルに格納されているコードを転送すると、現在のディスプレイマネージャのセッションに属していなくても、常にサーバにアクセスできます。

正しい認証コードを提示するクライアントへのアクセスを制限するには、ホストアクセス制御を有効にして、該当するホストを明示的に除外します。

XDMCP セキュリティとローカルな認証ファイルは、同時に使用できません。XDMCP はホスト上の一時的な **.Xauthority** ファイルを使用します。このファイルは、XDMCP を使用しているクライアントが実行されると、動的に作成されます。また、このファイルは手動で作成した **.Xauthority** ファイルに上書きするため、PC とホストファイルの値は一致しなくなり、クライアントは実行されません。

ローカルな認証ファイルを有効にするには

- 1 X サーバの [プロパティ] ダイアログボックスで、[セキュリティ] タブをクリックします。
- 2 [Local Authorization File] をクリックします。
- 3 [ファイル] テキストボックスに認証ファイル名を入力します。
ファイルが MS-DOS のパス内にはないフォルダに入っている場合は、フォルダもすべて指定する必要があります。

認証ファイルの位置がわからない場合は、[参照] をクリックして、[Local Authorization File] ダイアログボックスを表示すると、ファイルを検索できます。

キーボードの使い方

このセクションの内容は次のとおりです。

- Xにおけるキーボード入力
- Xサーバのキーボード入力
- Xキーマップの構成
- XサーバとXmodmap
- ショートカットキー
- 修飾キー
- Num Lockの管理

Xにおけるキーボード入力

X Window Systemでは、キーボード上のキーは、キーコードと呼ばれる8-254までの数字で識別されます。キーコードをどのように物理的なキーに割り当てるかは任意で、キーボードのタイプによって異なります。Xでは、キーコードをキーシンボルと呼ばれる論理的なキーの値に変換するためのメカニズムを提供します。キーシンボルとはキーボード上のキーに記された記号に対応付けられる値です。原則的には、キーに記されたすべての記号にキーシンボルがあります。特定の言語を対象とした特別なキーボードには、適切なキーの集まりとそれに対応付けられたキーシンボルがあります。

キーの多くは、SHIFTキーなどの1つまたは複数の修飾キーの状態に応じて、異なる値を生成する必要があります。このため、Xでは1つのキーコードを索引リストの複数のキーシンボルに対応付けることができます。最初のキーシンボルはシフトキーを押さないときのキーに対応し、2番目のキーシンボルはシフトキーを押したときのキーに、3番目のキーシンボルはMod2修飾キー(モードスイッチ)を押したときのキーに、そして4番目のキーシンボルはSHIFTキーとMod2キーを押したときのキーにそれぞれ対応します。

キーを押したり離したりすると、クライアントは通常キーのキーコードと修飾キーの状態を示すキーボードイベントを受け取ります。この情報を対応付けられているXキーシンボルに変換できるようにするため、クライアントは接続起動時にXサーバで提供されるキーボードマップを使用します。

Xサーバのキーボード入力

Xサーバでは、キーボード上の各キーに対応付けられるキーシンボルを定義した、コンパイルされたキーボードマップを使用します。キーボードマップは、標準的なWindowsのすべてのキー配列に規定されています。

106. HP HP-9000 シリーズ用	106. IBM RS/6000 シリーズ用
106. NEC EWS4800 シリーズ用	106. SONY NEWS シリーズ用
106. SUN SPARC Station シリーズ用	106. その他(基本設定)
106. 富士通 DS/90 7000 シリーズ用	
親指シフト - HP HP-9000 シリーズ用	親指シフト - IBM RS/6000 シリーズ用
親指シフト - NEC EWS4800 シリーズ用	親指シフト - SONY NEWS シリーズ用
親指シフト - SUN SPARC Station シリーズ用	親指シフト - その他(基本設定)
親指シフト - 富士通 DS/90 7000 シリーズ用	
98. HP HP-9000 シリーズ用	98. IBM RS/6000 シリーズ用
98. NEC EWS4800 シリーズ用	98. SONY NEWS シリーズ用
98. SUN SPARC Station シリーズ用	98. その他(基本設定)
98. 富士通 DS/90 7000 シリーズ用	

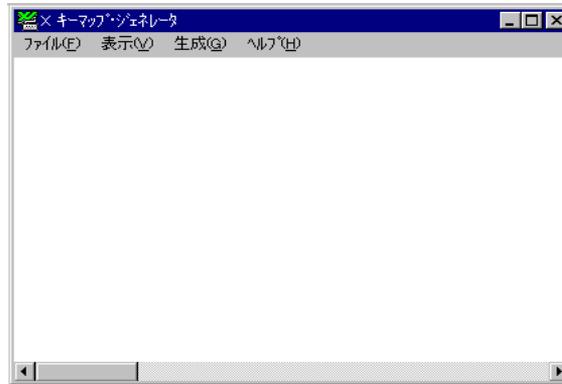
現在のキーボードマップ名は、[デバイスプロパティ]ダイアログボックスに指定されています。ご使用の Windows システムに合ったキーボードマップは、設定時に自動的に選択されません。キーボードマップは、X キーマップエディタを使用して、ASCII テキスト形式ファイル(.txt) からコンパイルされます(「X キーマップの使用」の章の「キーマップのコンパイル」を参照してください)。

PC キーボード上でキーが押されると、アクティブなアプリケーションに Microsoft Windows メッセージが送信されます。このメッセージには、仮想キーコード、キーが標準的な IBM キーボードにおけるキーパッドの ENTER キーや右側 CONTROL キーなどの拡張キーであるのかといった情報があります。X サーバのキーボードマップで、仮想キーコード/拡張キーの値と X キーシンボル間の対応が定義されます。

注意 キーボードマッピングが正しく機能するには、Microsoft Windows のインストール時に正しいキー配列と国が指定されていなければなりません。特定のキーが押されたときに生成される走査コードは、使用しているキーボードによって異なります。また、走査コードから仮想キーコードへの変換は、Windows の国設定によります。標準でないキーボードを使用している場合は、変更されたキーボードマップを使用する必要があることがあります。

X キーマップの構成

X キーマップエディタを使用すれば、X サーバで使用するためのキーボードマップの作成や変更を行うことができます。



X キーマップエディタの詳しい使い方については、[ヘルプ]メニューの[ヘルプトピックス]をクリックしてください。また、メインの Vision のヘルプ目次から[X キーマップエディタの使い方]を選択しても同じ情報を検索できます。以下に関するヘルプを表示できます。

- 起動と終了
- キーマップの扱い方
- キーボードシステム情報の表示
- Windows のキーコードの表示

X サーバと Xmodmap

X サーバは仮想キーコード/拡張キー情報を使用して、各キーに対して一意の X キーコードを生成します。この値が、クライアントに送信されます。キーコードはコンパイル時に任意の方法でキーに割り当てられ、特定の物理的なキーには対応付けられません。このため、**xmodmap** クライアントを使用して、キーボードマップのキーコードとキーシンボル間の対応付けを直接変更しないでください。ただし、式リスト内のキーシンボルだけを使用していれば、**xmodmap** を使用して、キーボードマップを変更できます。

ショートカットキー

Microsoft Windows のアプリケーションでは通常迅速な処理を行うために、ALT+key キーボードシーケンス(“ショートカットキー”)を使用できます。このようなキーは押されると、通常 Microsoft Windows で処理され、アプリケーションプログラムでは認識できません。

ショートカットキーは Microsoft Windows でなく X ウィンドウシステムにも転送できます。たとえば、OSF/Motif ウィンドウマネージャは Microsoft Windows と似たような方法で ALT+key ショートカットキーを使用します。[Devices プロパティ] ダイアログボックスに [ショートカットキー] チェックボックスがあります。このボックスにチェックマークを付けると、ショートカットキーは Microsoft Windows で受け付けられますが、X ウィンドウシステムでは受け付けられません。[ショートカットキー] ボックスにチェックマークが付けられていない場合は、ショートカットキーはその他のすべてのキーボード入力と同じように X ウィンドウシステムに渡されます。

ショートカットキーとして扱われるキーは、キーボードマップに定義されています。このキーには通常次のようなものがあります。

ALT+SPACEBAR	ALT+F8
ALT+F4	ALT+F9
ALT+F5	ALT+F10

修飾キー

修飾キーは、他のキーのアクションを変更するので、このように呼ばれます。たとえば、ほとんどのキーボードユーザは SHIFT キーと CONTROL キーのアクションをよく知っています。X を特定のキー配列から独立させるには、キーボード上の物理キーに対応可能な “論理的な” 修飾キーの概念を使用します。論理的な修飾キーは、以下のとおりです。

Shift	Mod2
Lock	Mod3
Control	Mod4
Mod1	Mod5

X サーバでは、最高 3 つの物理キーを各修飾キーに対応付けることができます。

修飾キーの場合、オートリピート機能は自動的に無効になります。

Mod2 修飾キーにはさらにメインキーマップのモードスイッチとしての機能もあります。標準的な IBM PC 拡張キーボードの場合、X サーバはモードスイッチとして ALT GR キーを使用します。Microsoft Windows では、このキーによって VK_CONTROL キーイベントとそれに続いて VK_MENU イベントが生成されます。サーバはこのイベントの発生を検出し、代わりに異なる 1 つの仮想キーコードを生成します。これは、X サーバのキーボードマップで使用される新しい仮想キーコードです。[キーボードマップ生成] または [デフォルトテーブルの追加] を選択すると、X キーマップエディタは Mode_switch キーシンボルに既存のマッピングがあるかどうか探し、見つからない場合にはこのマッピングを挿入します。

Num Lock の管理

テンキー上のキーを押した際に生成される値は、通常 Num Lock トグルスイッチの状態によって異なります。Num Lock が ON であれば、数字キーでキーボード数字が生成されます。Num Lock が OFF であれば、同じキーで HOME、END などが生成されます。

[デバイスプロパティ] ダイアログボックスに、ローカルな Num Lock の管理を無効にするためのオプションがあります。[NumLock キー制御] チェックボックスにチェックマークを付けないと、キーボード数字キーによって常に数字が生成され、NUM LOCK キーが押された場合キーボードイベントがクライアントに送信されます。[NumLock キー制御] が有効であると、NUM LOCK キーボードイベントは送信されません。

マウスの使い方

X ウィンドウシステムは、3 ボタンマウスで使用するよう設計されています。ほとんどの PC のマウスは 2 つのボタンから構成されていますが、さいわい Microsoft Windows では中央のボタンを認識でき、X サーバでこの機能をエミュレートできます。

マウスの中央ボタンのクリックをエミュレートするには

- 1 マウスの左ボタンと右ボタンを一緒に押します。
- 2 この両方のボタンを離します。

X サーバが左右両方のボタンを押して中央ボタンとすることを認識するには、素早くこの両方のボタンを押す必要があります。この間隔は、[デバイスプロパティ] ダイアログボックスで変更できます(ヘルプのキーワードの「マウスの中央ボタンのエミュレーション」を参照してください)。一方のボタンを押してからもう一方のボタンを押すまでの間隔がこの指定時間を超えると、たとえば左ボタンをクリックし続いて右ボタンをクリックしたなど、2 つの別のマウスアクションとして解釈されます。

中央ボタンを使用してドラッグするには

- 1 マウスの左ボタンと右ボタンを一緒に押します。
- 2 一方のボタンを離すことができます。
アクションを開始するために使用した一方または両方のボタンが押した状態であれば、“中央ボタン”は押したままとみなされます。
- 3 ボタンを押したままマウスを移動します。
- 4 ドラッグが終了したら、中央ボタンのアクションを保持しているボタンを離します。

“中央ボタン”と他のマウスボタンやキーと一緒に押すことができます。たとえば、Xアプリケーションで左ボタンと中央ボタンを押しながらドラッグする場合、両方のボタンを一緒に押し、右ボタンを押したまま左ボタンを離します。そして、左ボタンをもう一度押して、マウスを新しい位置にドラッグしてから、マウスの両方のボタンを離します。

3ボタンマウスがPCに接続されている場合は、中央のボタンのエミュレーション機能を無効にすると、マウスの左ボタンと右ボタンを一緒に押した場合に、この操作で別のマウスアクションが行われます。

フォントの使い方

このセクションの内容は次のとおりです。

- Xにおけるフォント
- フォントとXサーバ
- 標準的なXフォント
- フォントパス項目
- PCへの追加フォントのインストール
- フォントフォルダ項目
- フォントサーバ
- XクライアントでのWindowsフォントの使い方
- 他のWindowsアプリケーションでのXフォントの使い方
- フォント置換の使い方

Xにおけるフォント

Xプロトコルでは、ワイルドカード文字も含め文字列を指定することによって、フォントを指定できます。ただし、プロトコルでは文字列の内容は定義していません。X11R2では、通常この文字列にフォントファイル名が格納されていました。

X11R3では、この文字列を使用してフォントのあらゆる特性の詳細な情報を保持するための取り決めが導入されています。この文字列はX論理フォント記述(XLFD)と呼ばれ、X11R3以降と互換性のあるXクライアントはこのような文字列を使用して、Xサーバに特定のフォントを要求します。X論理フォント記述の要素は左から順に、制作/開発者名、フォントファミリ、重み、傾斜状態、文字間隔の詰まり具合、補助的なスタイル、ピクセル値、ポイント数(10分の1ポイント単位)、水平方向の分解能(ピクセル/インチ単位)、垂直方向の分解能(ピクセル/インチ単位)、ピッチの種類、文字の平均幅(10分の1ピクセル単位)、文字セットを設定した団体名や標準化番号などの識別名、文字セットの識別名です。これらの要素をハイフンでつないで表します。たとえば、Courier 10ptフォントは、次のように指定します。

-Adobe-Courier-Medium-R-Normal-10-100-75-75-M-60-ISO8859-1

どの要素でも、アスタリスク文字 (*) をワイルドカードとして使用できます。たとえば、次のように指定しても、Courier 10pt フォントを指定できます。

-Courier--R-*-100-*

X11R5 では、X において拡張可能なフォントの概念を採用しました。X サーバが拡大縮小可能なフォントを提供できることをクライアントに報告すると、クライアントはどのポイントサイズのフォントでも要求できます。このスケラブルフォントは、XLFD においてポイントサイズおよびピッチの値がゼロになります。

X11R6 では、フォント技術が改良されました。新機能のひとつはマトリックス XLFD 拡張機能です。このため、変換マトリックスを使用して、既存のビットマップと拡大縮小可能なフォントの変換を生成できるようになりました。これによって、アプリケーションでフォント変換 (拡大や縮小) を作成したり、特殊な効果 (ミラー化や回転) を生成できます。マトリックスは、次の例に示すように、ポイントサイズとピクセルサイズのフィールドに指定できます。

-Adobe-New Century Schoolbook-Medium-R-Normal-0-¢12 0 0
12¢-100-100-P-0-ISO8859-1

マトリックス XLFD 拡張機能が適用されるのは、フォントが開かれている際に、クライアントで特殊な効果を生成するために、各文字で異なるフォントを開く必要がある場合だけです。多くのフォントを開くプロセスをスピードアップするために、文字サブセットがサポートされています。これは、フォント名により、どの文字が該当するかのヒントを示す機能です。文字サブセットの指定は、XLFD 名の後に =value [:value] [+value] の形式で追加できます。この場合の value は ASCII 文字の値です。範囲を指定するには 2 つの値をコロン (:) で区切り、範囲を複数指定するにはプラス (+) で区切ります。たとえば、次のようになります。

-Adobe-Utopia-Medium-R-Normal-0-120-0-0-P-0-ISO8859-
1=65:67+118:120+32

フォント別名化システムも使用できます。別名とは、X の論理フォント記述を使用する代わりに、フォントを識別するために使用可能なショートネームです。X サーバでは、フォント名の別名は **Fonts.ali** ファイルで定義します。詳細については、本章の「**Fonts.ali** ファイル」を参照してください。

フォントと X サーバ

X のアプリケーションと Windows のアプリケーションでは、フォントファイルの形式が異なります。X のアプリケーションでは多様なフォント形式を使用します。その代表的なものには Bitmap Distribution Format (**.bdf**) と Portable Compiled Format (**.pcf**) です。これに対し、Microsoft Windows のアプリケーションでは Windows の **.fon** 形式を使用します。X サーバは、**.fon** 形式か **.pcf** 形式のいずれかのフォントを使用できます。

X ウィンドウシステムの標準的なフォントはすべて X サーバと一緒に配布され、Microsoft Windows で使用できるようにコンパイルされます。また、フォントコンパイラが提供され、他の X フォントを X サーバで使用できるようにするために、**.bdf** フォントファイルと **.pcf** フォントファイルを **.fon** ファイルに変換します。フォントコンパイラはさらに **.bdf** ファイルを **.pcf** フォントに変換することも可能です。これは、大きくて **.fon** 形式に変換できないフォントの場合に便利です。一度 **.fon** 形式にすれば、必要に応じて、X フォントを他の Windows アプリケーションで使用できます。

X サーバにフォントを提供するために、リモートフォントサーバも使用できます。X クライアントで使用されるフォントは、平均的な PC で大量のディスク容量を使用する恐れがあります。フォントサーバを使用すると、ほとんどのフォントをリモートに格納できます。X サーバでのフォントサーバの使い方の詳細については、本章の「フォントサーバ」を参照してください。

フォント変換機能によっては、拡大縮小可能なフォントやマトリックス XLFD 拡張機能などの **.pcf** フォントの場合にしか使用できないものがあります。

また、X サーバには、X クライアントで要求されたフォントが使用できない場合に、代替フォントを提供するフォント置換メカニズムが用意されています。対話形式でユーザが使用するフォントを選択することも、自動的に代替フォントを選択することも可能です。詳細については、本章の「フォント置換」を参照してください。

X サーバではまた、Microsoft Windows のフォントを X クライアントで使用できます。詳細については、本章の「X クライアントでの Windows フォントの使い方」を参照してください。

標準的な X フォント

X ウィンドウシステムの標準的なフォントは、各種の UNIX システムで必要な追加フォントと一緒にコンパイル形式で配布されます。これらのフォントは次のように、**C:¥Program Files¥Vision¥Fonts** のサブフォルダに格納されます。

フォルダ	フォント
Misc	固定ピッチのフォント; カーソルフォント; 旧バージョンの X と互換性のあるフォント。このフォントセットは必ずインストールされます。
75dpi	可変ピッチの 75 d.p.i. フォント。

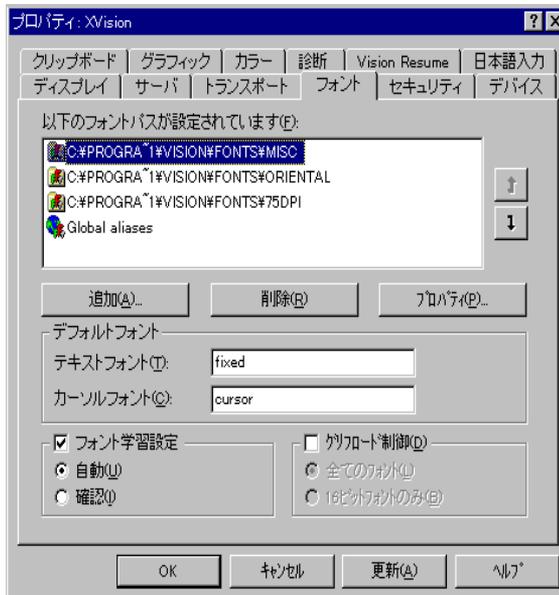
フォルダ	フォント
100dpi	可変ピッチの 100 d.p.i. フォント。
Oriental	漢字などの東洋のフォント。このフォントファイルはサイズが大きいため、.fon ファイルではなく、.pcf ファイルとして提供されます。
Andrew	一部の IBM アプリケーションに必要な Andrew ツールキットのフォント。
ICL	一部の ICL アプリケーションに必要なフォント。旧バージョンの OPEN LOOK のフォント、および ICL VT 端末エミュレータのフォントなどです。
HP	一部の Hewlett-Packard のアプリケーションに必要なフォント。

75 d.p.i. フォントと 100 d.p.i. フォントの集まりには、さまざまなサイズの多様な書体、つまりフォントごとに通常書体、ボールド書体、イタリック書体などが含まれます。これは、Adobe Systems, Inc. や Bitstream, Inc のフォントなど、高品質のフォントです。

設定で [標準] オプションを使用すると、ご使用のディスプレイハードウェアに合わせて、Oriental フォントと Misc フォントと 75DPI フォントまたは 100DPI フォントのいずれかがインストールされます。Andrew、ICL、HP のフォントはインストールされません。[カスタムセットアップ] を使用すると、どのフォントセットをインストールするかを選択できます。

フォントパス項目

フォントパス項目のリストを [フォントプロパティ] ダイアログボックスで指定して、X フォントがどこに格納されているかをサーバに指定しなければなりません。初期フォントパスは、設定時に選択したオプションによって異なります。このパスには通常 X フォントが格納されたいいくつかのフォルダが含まれ、さらにフォントサーバも含まれることがあります。フォントパス項目として、Windows のフォントも入れることができます。さらに、“Global aliases” というフォントパス項目もあります。このパス項目は、使用する固定フォント置換のリストに登録されていなければなりません。項目のタイプはすべて、[フォントプロパティ] ダイアログボックスのコマンドボタンを使用して構成できます。



アプリケーションに付属しているフォントなどの新しいフォントを追加する場合、適切な項目をフォントパスに追加しなければなりません。追加したフォントは、フォントパスを変更すると直ちに、Xクライアントで使用できるようになります。

フォントパス項目の順序の変更

特定のフォントを検索する際、Xサーバはフォントパス項目のリストボックス内の順序でフォントパス項目を検索します。このため、矢印ボタンを使用して項目の順序を変更して、効率化を図ることができます。できる限り、要求されたフォントが別名の前になるように、グローバル別名項目をリストの一番下にしてください。

フォントパス項目の順序を変更するには

- 1 Xサーバの[プロパティ]ダイアログボックスで、[フォント]タブをクリックします。
- 2 [フォントパス]リストボックスの項目を選択します。
- 3 矢印を使用して、リスト内の新しい位置に移動します。
項目がリスト内で上下に移動します。

PC への追加フォントのインストール

ご使用のXアプリケーションに特殊なフォントが付属している場合、配布メディアからPCにコピーする必要があります。

追加フォントをインストールするには

- 1 マイコンピュータまたはファイルマネージャを使用して、Xサーバのフォントフォルダ内に新しいフォルダを作成します。Windows 2000/Me/98/95の場合、このフォントフォルダはデフォルトでは **C:¥Program Files¥Vision ¥Fonts** です。
- 2 [Unix ネットワークコンピュータ]を使用して、ホストシステムのフォントファイルをPCの新しいフォルダにコピーします。
- 3 以下のいずれかにより、Xフォントコンパイラを起動します。

[スタート] ボタンをクリックして、[プログラム]、[CentreNET X Vision]、[アクセサリ]を順にポイントした後、[フォントコンパイラ]をクリックします。

-または-

Windows NT 3.51 の場合は、プログラムマネージャで Vision グループを開いてから、[フォントコンパイラ] アイコンをダブルクリックします。

フォントのコンパイル

.bdf形式のフォントは、Xサーバで使用できるようにコンパイルしなければなりません。.pcf形式のフォントは、コンパイルする必要はありませんが、コンパイルを行えばパフォーマンスが向上する可能性があります。ただし、漢字フォントなどの一部のフォントは、大きすぎてコンパイルできないことがあります。いずれの場合にも、**Fonts.dir** 索引ファイルを作成して、フォントパスにフォルダを追加しなければなりません。

フォントをコンパイルするには

- 1 X Vision フォントコンパイラで、[ファイル] メニューをクリックしてから、[プリファレンス] をクリックします。[フォントインデックスを自動作成] オプションと [X Vision フォントパスを自動アップデート] オプションが必ず有効でなければならず、有効でないと、この手順を個々に行う必要があります。
- 2 [ファイル] メニューで、[フォントファイルの選択] をクリックして、新しいフォントフォルダを選択します。
- 3 [全てをコンパイル] をクリックします。

これで、新しいフォントの追加プロセスは終わりです。

フォント索引の作成

新しいフォントをコンパイルしていない場合は、各フォントファイルとそれに対応付けられたX論理フォント記述がリストされた索引ファイル (**Fonts.dir**) を作成する必要があります。[フォントインデックスの作成] コマンドを使用すれ

ば、Xの属性が格納されたすべての.fonファイルに対して新しい**Fonts.dir**ファイルを作成できます。Xサーバで直接読み取り可能な.pcfフォントも**Fonts.dir**ファイルに含まれます。**Fonts.dir**ファイルは手動では編集できません。

フォント索引ファイルを作成するには

- 1 X Vision フォントコンパイラで、[ファイル]メニューをクリックしてから、[フォントインデックスの作成]をクリックします。
- 2 .fon形式または.pcf形式のフォントファイルが格納されているドライブとフォルダに切り換え、[OK]をクリックします。

ダイアログボックスが表示され、索引付けの進捗状況が示されます。

フォント索引が作成されると、メッセージが表示されます。[フォントパスの自動更新]オプションが有効であると、フォントパスが更新され、新しいフォルダがフォントパスに含まれます。

フォントパスへのフォルダの追加

新しいフォントをXサーバで利用できるようにするには、フォントをインストールしたフォルダがXサーバのフォントパス内になければなりません。[プリファレンス]ダイアログボックスでこのオプションが選択されていれば、X Vision フォントコンパイラが自動的にこの処理を行います。ただし、このオプションが選択されていないと、Xサーバの[フォントプロパティ]ダイアログボックスを表示して、フォントパスに新しいフォントフォルダを追加する必要があります。フォルダを手動で追加する方法については、「フォントフォルダ項目」を参照してください。

フォント別名の作成

フォント別名の設定については、Xアプリケーションのサブライヤの指示に従ってください。[フォントフォルダエディタ]ダイアログボックスを使用して、フォルダ内の**Fonts.ali**ファイルの作成および変更を行うことができます。サーバの[フォントプロパティ]ダイアログボックスで、目的の[フォントフォルダ]項目を選択してから、[プロパティ]ボタンを選択します。

フォントフォルダ項目

Fonts.dirファイルが格納されているフォルダは、フォントフォルダ項目として、Xサーバのフォントパスに追加できます。フォントフォルダにはまた**Fonts.ali**ファイルが格納されることもあります。この**Fonts.ali**ファイルには、**Fonts.dir**ファイルの論理フォント記述の別名が格納されます。

フォルダをフォントパスに追加するには

- 1 Xサーバの[プロパティ]ダイアログボックスで、[フォント]タブをクリックします。
- 2 [追加]をクリックします。
- 3 [フォントフォルダ]をクリックしてから、[追加]をクリックします。
- 4 ドライブとファイルを検索して、必要な**Fonts.dir**ファイルを探します。

フォルダを [フォントパス] 項目リストに追加するには、あらかじめ **Fonts.dir** ファイルがフォルダ内になければなりません。正しい **Fonts.dir** ファイルがないフォルダを追加すると、**Fonts.dir** ファイルが見つからないという旨のエラーメッセージが表示されます。X Vision フォントコンパイラを使用して X フォントを **.fon** 形式または **.pcf** 形式にコンパイルした場合には、**Fonts.dir** ファイルは自動的に作成されます。

新しい項目が、[フォントプロパティ] ダイアログボックスのフォントパス項目のリストにパス名として表示されます。フォントフォルダ項目が作成されている場合は、フォルダのパス名は変更できません。

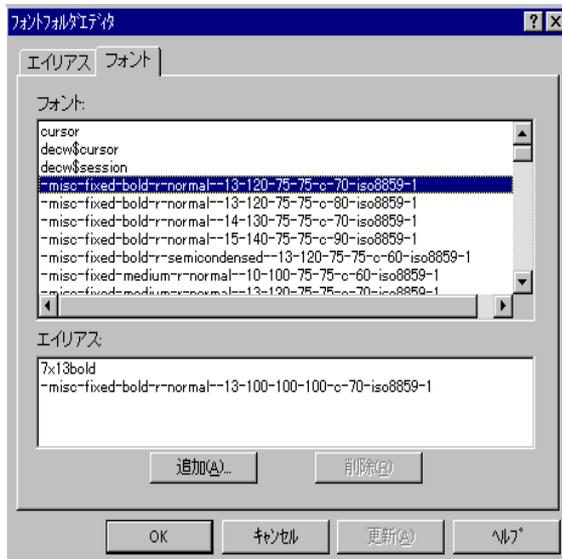
フォントフォルダ項目の構成

既存のフォントフォルダ項目に別名を設定するように構成するには、フォントパス項目のリストで目的の項目をクリックします。[フォントフォルダエディタ] ダイアログボックスが表示され、[エイリアス] タブにすでに設定されている別名が表示されます。**Fonts.ali** ファイルがフォルダ内にない場合は、作成されます。



別名はそれぞれ **Fonts.ali** ファイルで X 論理フォント記述と対応付けられ、X 論理フォント記述は **Fonts.dir** ファイルのフォントファイルと対応付けられます。リスト内の別名をクリックすると、対応付けられている論理フォント記述とフォントファイルが表示されます。別名は新しく追加したり、削除することができます。

[フォント] タブには **Fonts.dir** ファイル内のフォントがリストされているため、別の表示に切り換えることができます。



フォントをクリックして、対応付けられている別名を表示したり、別名の追加や削除を行うことができます。

Fonts.dir ファイルでは常に論理フォント記述とフォントファイルが1対1で対応しますが、**Fonts.ali** ファイルでは1つの論理フォント記述に複数の別名を定義できます。また、**Fonts.ali** ファイルで、論理フォント記述を別の論理フォント記述の別名にできます。これは、フォントを置換するための1つのメカニズムです。フォントの置換は、本章で説明するXサーバのフォント置換メカニズムを使用すれば、より柔軟に対処できます。

Fonts.dir ファイル

Fonts.dir ファイルにはフォルダ内のすべてのフォントがリストされます。各行に次の形式でそれぞれのフォントがリストされます。

```
filename logical-name
```

この場合、*filename*はそのフォントが格納されているファイル名で、*logical-name*はX論理フォント記述です。次に例を示します。

```
COURB010.fon -Adobe-Courier-Bold-O-Normal-10-100-75-75-M-60-ISO8859-1
```

Fonts.ali ファイル

Fonts.ali と呼ばれるファイルを各フォントフォルダに格納し、フォントの別名を指定できます。このファイルは、次の形式で指定します。

```
alias logical-name
```

この場合、*alias* はフォントに使用する代替名で、*logical-name* は X 論理フォント記述です。次に例を示します。

```
variable *-helvetica-bold-r-normal-***-140-*
```

フォント別名ファイルは、**Fonts.ali** 内の独立した行に別の別名ファイル名を指定するだけで、簡単に入れ子にできます。また、**Fonts.ali** ファイルの任意の場所に次の 1 行を書き込むこともできます。

```
FILE_NAMES_ALIASES
```

この指定によって、論理名の別名としてファイル名を使用して、**Fonts.dir** ファイルにすべてのフォントの別名が作成されます。この方式は、ファイル名を使用してフォントを要求する X11R2 クライアントと互換性を保つためのものです。

フォントサーバ

フォントサーバは、リモートの UNIX ホストに格納されているフォントを X サーバで使用できるようにするためにリモート UNIX ホストで実行されるプログラムです。これによって、X サーバが使用される各 PC にフォントをインストールする必要はなく、別の PC を使用していても必ずフォントにアクセスできるので、フォント管理作業は簡易化されます。また、サーバを実行するために必要なディスク容量を節約することにもなります。

フォントサーバによってアクセスされるフォントは、*カタログ*と呼ばれる名前付きのセットに分類できます。カタログはフォントファイルが格納された各ディレクトリに相当します。

UNIX システム上のフォントサーバプログラムは通常 **fs** と呼ばれ、**/usr/bin/X11** にインストールされます。システムで使用できるフォントサーバの詳細については、システム管理者にお問い合わせください。クライアントアプリケーション **fsinfo** を使用すると、使用可能なカタログなど、フォントサーバに関する詳細情報を入手できます。

X クライアント **fsfonts** を使用すれば、サーバフォントをリストできます。

フォントサーバで提供されるフォントは、そのフォントを格納するために使用されるファイル形式と関係なく、一貫したバイナリ形式です。

X サーバはフォントサーバのフォントの保留グリフローディングをサポートします。これによって、不要な文字の生成が保留され、特に大きなアジアのフォントなどのフォントの生成におけるパフォーマンスが向上します。

フォントサーバは [フォントプロパティ] ダイアログボックスを使用して、フォントパス項目として作成できます。

フォントサーバを使用するように X サーバを構成するには

- 1 X サーバの [プロパティ] ダイアログボックスで、[フォント] タブをクリックします。
- 2 [追加] をクリックします。
- 3 [Font Server Fonts] をクリックしてから、[追加] をクリックします。
- 4 必要なオプションを設定します。
- 5 フォントサーバのフォントの保留グリフローディングを有効にするために、[グリフロード制御] をクリックしてから、[16 ビットフォントのみ] か [全てのフォント] のいずれかをクリックします。

X サーバはフォントサーバとの接続が失われたことを検出すると、再度接続を行うかどうか求めるメッセージを表示します。ここで [はい] を選択すると再度接続が開かれ、[いいえ] を選択するとサーバが再起動するまではフォントサーバには接続しません。フォントサーバがリモートシステムで動作していないと、再接続は失敗し、タイムアウト遅延になります。

X クライアントでの Windows フォントの使い方

フォントバスに Windows のフォント項目を追加すれば、PC にインストールされている Microsoft Windows のフォントを X クライアントで使用できます。X サーバは Windows フォントの X 論理フォント記述を内部フォントテーブルに追加して、X フォントフォルダ内の **Fonts.dir** ファイルから読み込まれたフォント記述を補足します。

X クライアントで Windows フォントを使用するには

- 1 X サーバの [プロパティ] ダイアログボックスで、[フォント] タブをクリックします。
- 2 [追加] をクリックします。
- 3 [Windows フォント] をクリックしてから、[追加] をクリックします。
- 4 必要なオプションを設定します。

X サーバは X 論理フォント記述を追加する際、“microsoft windows” の制作 / 開発者名を使用します。次に例を示します。

```
-microsoft windows-Courier-Bold-O-Normal  
10-100-75-75-M-60-ISO8859-1
```

Microsoft Windows フォントの別名を設定できます (本章の「**Fonts.ali** ファイル」を参照してください)。フォントの代替名を **Fonts.ali** ファイルに追加できますが、**Fonts.ali** ファイルは **Fonts.dir** ファイルが格納されたフォルダ内になければなりません。このため、***Misc** フォントフォルダ内の **Fonts.ali** ファイルの使用をお勧めします。

TrueType フォントの使い方

TrueType フォントを Microsoft Windows にインストールして有効にすると、拡大縮小可能なフォントとして X で使用できます。

TrueType フォントを使用するには

- 1 [Windows フォント] ダイアログボックスを開きます。
- 2 [Plotter] ボックスと [Scale] ボックスをクリックして、チェックマークを付けます。
- 3 必要に応じて、[ボールド] ボックスと [イタリック] ボックスの一方または両方をクリックして、チェックマークを付けます。
- 4 [最大サイズ] 値と [最小サイズ] 値を指定します。この2つの値により、論理フォント記述を追加するポイントサイズの範囲が設定されます。さらに、論理フォント記述を追加し、フォントを拡大縮小可能フォントとして使用するようになります。

TrueType フォントをインストールすると、X サーバの内部フォントテーブルに追加されます。各フォントに対して、論理フォント記述が拡大縮小可能なフォントとしてフォントに、またダイアログボックスの [最小サイズ] 値と [最大サイズ] 値で設定された特定のポイントサイズの範囲に追加されます。

他の Windows アプリケーションでの X フォントの使い方

フォントコンパイラで .fon 形式に変換した X フォントは、Microsoft Windows の他のアプリケーションで使用できます。フォントを他のアプリケーションで使用できるようにするには、Windows のコントロールパネルを使用して使用可能な Windows のフォントのリストに追加します。コントロールパネルを使用したフォントの追加方法の詳細については、Microsoft Windows のマニュアルを参照してください。

注意 X フォントを X クライアントでのみ使用する場合には、X フォントを Microsoft Windows に追加する必要はありません。

フォント置換の使い方

ほとんどの X アプリケーションでは特定のフォントを要求し、そのフォントが使用可能でないと、クライアントは実行できません。X サーバはフォント置換を使用することでこの問題に対処します。フォント置換は、自動的にも対話形式でも行うことができます。ただし、一部のアプリケーションでは独自の置換機能があり、X サーバよりも良いフォントを選択できるため、フォント置換メカニズムは注意して使用してください。フォント置換は、HINTS (「Helpful Interactive Support」を参照してください) を参照しながら使用し、フォントが見つからないためにクライアントを終了するというメッセージが HINTS から提示された場合に限り有効にしてください。

フォント置換を使用可能にするには

- 1 X サーバの [プロパティ] ダイアログボックスで、[フォント] タブをクリックします。

- 2 [フォント学習設定] をクリックしてから、[確認] と [自動] のいずれかををクリックします。
- 3 固定フォント置換を使用する場合は、[グローバルエイリアス]項目を[フォントパス] リストに追加します。

ダイアログボックスが削除され、新しいフォント置換モードが直ちに有効になります。

自動別名化

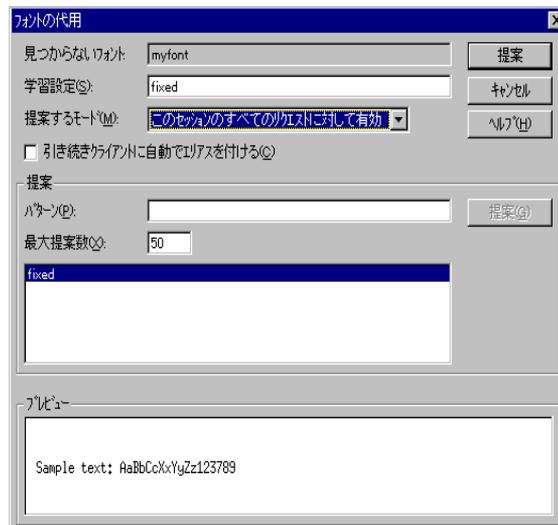
自動フォント置換が有効な場合、クライアントが要求するフォントが見つからないと、X サーバは暗黙のうちに適切な代替フォントを提示します。自動フォント置換に使用されるフォント別名は、現在のセッションに限り有効です。[自動]モードは、フォントがごくわずかしかない最小システムの場合に限り使用してください。

対話形式の別名化

対話形式のフォント置換を使用しているときに正しいフォントが見つからない場合は、ユーザが代替フォントを選択するためのダイアログボックスが表示されます。対話形式のフォント置換は、1回の要求、セッション内の全要求、または永久的に使用できます。固定フォント置換を使用するためには、[グローバルエイリアス]項目が[フォントパス]リスト内になければなりません。

代替フォントを選択するには

- 1 [フォント学習設定] ダイアログボックスが自動的に表示され、[学習設定]ボックスに X サーバが提示する代替フォントが示されます。



- 2 ドロップダウンリストボックスから [学習] モードをクリックします。
- 3 Xサーバの初期提案が適切でない場合は、[提案] リストボックスから代替フォントをクリックするか、または必要なフォントの別名が論理フォント記述を [学習設定] ボックスに入力します。
- 4 [学習] をクリックします。
- 5 各フォントに対してこの手順を繰り返すか、[引き継ぎクライアントに自動でエリアスを付ける] オプションを有効にします。

初期フォント提案リストに適切なフォントが含まれていない場合は、使用可能なすべてのフォントを手動で検索できます。

使用可能なフォントを検索するには

- 1 [フォント学習設定] ダイアログボックスを表示します。
- 2 [パターン] ボックスに検索パターンを入力します。

アスタリスク文字 (*) をワイルドカードとして使用して、論理フォント記述の要素を置換できます。* だけを使用すると、提案の最大数以内で使用可能なすべてのフォントが表示されます。

標準的な Windows のコピーアンドペースト手法を使用すれば、[存在しないフォント] ボックスから [パターン] ボックスに論理フォント記述の一部をコピーできます。コピーしたいテキストを選択し、CONTROL+INSERT を押します。コピー先を選択して、SHIFT+INSERT を押します。
- 3 表示したい提案の最大数を [最大提案数] ボックスに入力します。

特に Microsoft Windows のフォントが使用可能である場合など、使用可能なフォントの数は非常に大きいので、この値は適切な値に設定することをお勧めします。
- 4 [提案] をクリックします。

このパターンに合ったフォントがリストボックスに表示されます。

カラーの使い方

このセクションの内容は次のとおりです。

- Xにおけるカラー
- カラーとXサーバ
- カラーマップの管理
- カラー名データベース

Xにおけるカラー

カラーディスプレイのほとんどはRGBモデルになっていて、画面は多くのピクセル(画素)から構成され、それぞれのピクセルには赤、緑、青の蛍光ドットが含まれます。異なるカラーの蛍光体が独立した電子ビームに反応し、各カラーの蛍光体の強度を変えることで、ピクセルの表示色を変えることができます。

各ピクセルには、カラー情報を格納するために、ビデオメモリで使用可能なビット数があります。このビット数はまたプレーン数とも呼ばれます。通常、これらのビットは“カラーマップ”の索引となる値を格納するために使用され、カラーマップにはRGB値を指定する“カラーセル”が格納されています。たとえば、ピクセル値が12であれば、カラーマップの12番目のエントリにアクセスし、そのカラーセルに格納されているRGB値を使用して画面上の該当するピクセルに赤、緑、青の蛍光体の強度を設定します。ピクセル当たりで使用できるビット数によって、カラーマップで使用可能なエントリ数、および同時に表示できるカラーの数が決まります。

一部の高性能のカラーディスプレイでは、カラーマップがRGB値が格納されたカラーセルから構成される代わりに、原色ごとに1つずつの3種類の索引が格納されます。ピクセル当たりで使用できるビットは、各カラー索引それぞれにアクセスするために分割されます。たとえば、24ビットのカラーディスプレイでは、最初の8ビットが赤の値をインデックスするために使用され、次の8ビットが緑の値を、最後の8ビットが青の値をインデックスするために使用されます。このため、使用可能なカラーの数が多い場合にも簡単に管理できます。

Xでは“ビジュアル”と呼ばれる構造を使用して、ディスプレイのカラー機能を表します。たとえば、カラーマップのカラーセルの数(またはカラーマップに原色に対して独立した索引がある場合に、最大のサブマップのサイズ)に関する情報などです。また、ビジュアルにはビジュアル“クラス”が含まれます。ディスプレイハードウェアのタイプに応じて、グレー、グレースケール、カラー、疑似カラー、トゥルーカラー、ダイレクトカラーの6つのビジュアルクラスがあります。

ビジュアルは実際には特定のディスプレイの使用方法を表すため、1つの画面で数種類のビジュアルを使用できます。たとえば、通常カラーディスプレイはモノクロまたはグレースケールのデバイスとして扱うことができます。Xサーバはどのビジュアルクラスが現在使用できるかを指定し、デフォルトのクラスを定義します。

グレー、カラー、トゥルーカラーのビジュアルのカラーマップは変更できませんが、グレースケール、疑似カラー、ダイレクトカラーのビジュアルのカラーマップは変更可能で、アプリケーションでカラーを追加したり、独自のカラーマップをインストールできます。なお、変更可能なカラーマップのカラーセルは読み取り専用または読み取り/書き込みのいずれかです。

ディスプレイのほとんどにはハードウェアカラーマップが1つしかないため、Xでは複数の仮想カラーマップを実装します。これによって、ハードウェアカラーマップの内と外で仮想カラーマップをスワップできます。サーバは初期化時に、“デフォルトのカラーマップ”と呼ばれる1つの仮想カラーマップを作成し、ハードウェアカラーマップにインストールします。

カラーとXサーバ

PCのディスプレイのタイプに応じて、Xサーバはできるだけ多くのカラービジュアルをXクライアントで使用できるようにします。Xサーバが自動的に適切なデフォルトのビジュアルを選択することも、ユーザが[カラー]ダイアログボックスを使用して異なるデフォルトのビジュアルを指定することもできます。

XサーバではPCディスプレイを次の4つのカテゴリに分類します。

カテゴリ	PCディスプレイ
モノクロ	Hercules、EGA mono、VGA mono など
変更不可カラー	CGA など
変更可能カラー	EGA、VGA、Super-VGA、XGA、8514/A など
トゥルーカラー	256色以上のディスプレイ

Xサーバがこの各カテゴリでどのように動作するかについては、以降のセクションで説明します。

モノクロディスプレイ

モノクロディスプレイでは、Xサーバは白と黒の2色のカラーマップのグレービジュアルを提供します。EGAまたはVGAディスプレイをモノクロデバイスとして扱うためには、Windowsの設定手順を使用して、モノクロデバイスとして構成する必要があります。

変更不可カラーディスプレイ

Xサーバで変更できないハードウェアカラーマップを持つカラーディスプレイでは、カラービジュアルが提供されます。このカラーマップには、標準的なWindowsのカラーが格納されます。

変更可能カラーディスプレイ

X サーバがハードウェアカラーマップのカラーを変更できるカラーディスプレイの場合、サーバは全部で6つのビジュアルを提供できます。EGA または VGA ディスプレイでは、サーバはカラーマップを直接変更します。他のディスプレイでは、サーバはMicrosoft Windows のパレットマネージャを使用して、カラーマップを変更します。

工場出荷時のデフォルトのビジュアルは疑似カラー [Pseudo Color] ですが、これは [カラー] ダイアログボックスで変更できます。

グレイビジュアルを選択すると、カラーマップには白と黒が格納され、さらに各カラーセルの RGB 値の赤、緑、青のコンポーネントを同じに設定することによって生成されたグレイ値の範囲が含まれます。

グレースケールビジュアルでは、デフォルトのカラーマップに白と黒だけが格納され、残りのカラーセルは割り当てられません。[空きプライベートカラーマップ] オプションが有効であると、白と黒が格納されたカラーセルは変更できませんが、カラーマップはクライアントにとっては空に見えます。クライアントでその他に割り当てられたカラーはグレイになり、各 RGB 値の赤、緑、青のコンポーネントは等しくなります。

カラービジュアルを選択し、[カラー] 属性ダイアログボックスの [Windows システムカラー] オプションが無効であると、カラーマップに白と黒が格納され、その他のカラーも均等に配分されます。[Windows システムカラー] オプションが有効の場合には、カラーマップには Microsoft Windows システムの標準的なカラーが格納され、他のカラーは別のカラーセルに均等に配分されます。

疑似カラービジュアルでは、[Windows システムカラー] オプションが無効の場合、白と黒だけが格納され、残りのカラーセルは割り当てられていないデフォルトのカラーマップが提供されます。[空きプライベートカラーマップ] オプションが有効の場合は、白と黒が格納されたカラーセルは変更できませんが、カラーマップはクライアントにとっては空に見えます。また、[Windows システムカラー] が有効の場合は、デフォルトのカラーマップには読み取り専用のカラーセルに Windows の標準的なカラーが格納され、残りのカラーセルは割り当てられません。クライアントは、割り当てられていないカラーセルをデフォルトのカラーマップにカラーを追加したり、独自の代替ビジュアルカラーマップを作成できます。

トゥルーカラーおよびダイレクトカラービジュアルは、プレーンが8個以下のディスプレイではほとんどカラーが提供されないため効果がありません。

なお、4ビットの EGA または VGA ディスプレイは16個のカラーセルから構成されるカラーマップだけしかサポートできません。このため、[Windows システムカラー] オプションを使用しても、カラーまたは疑似カラーのカラーマップに16色しか提供できません。ただし、これよりも多くのカラーセルが使用可能であれば20色を提供します。また、疑似カラービジュアルで [Windows システムカラー] オプションを有効にすると、すべてのカラーセルが割り当てられます。MWM ウィンドウマネージャなどの一部のクライアントでは、カラーマップにカラーを追加できないと、エラーを表示します。

トゥルーカラーディスプレイ

8個以上のプレーン(256色以上)を持つカラーディスプレイでは、サーバはトゥルーカラービジュアルと疑似カラービジュアルを提供します。トゥルーカラービジュアルを使用すると、Xクライアントで指定されたRGB値がMicrosoft Windowsに直接渡されます。疑似カラービジュアルが使用可能であるため、疑似カラービジュアルを受け付けるクライアントであればトゥルーカラーディスプレイにも使用できます。

カラーマップの管理

ハードウェアカラーマップはすべてのクライアントで共有されるため、クライアントでデフォルトのカラーマップを変更したり、デフォルトのカラーマップをスワップアウトし独自のカラーマップをインストールすると、Xモデルで問題が発生する可能性があります。新しいカラーマップがインストールされると、古いカラーマップを使用するクライアントがすべて間違ったカラーで表示されます。このため、Xカラーマップのスワッピングはウィンドウマネージャにまかせてください。ウィンドウごとにウィンドウの属性として指定されたカラーマップを持つことができるため、ウィンドウマネージャでは通常、現在入力フォーカスを持っているウィンドウに対応付けられているカラーマップをインストールします。このため、Visionウィンドウマネージャ(VWM)は、Xクライアントがアクティブウィンドウになると、このウィンドウに対応付けられているカラーマップをインストールします。VWMの詳細については、「Xウィンドウの管理」の章の「Visionウィンドウマネージャ」を参照してください。

Xサーバの場合には、Microsoft Windowsの他のアプリケーションが同時に画面に表示されるため、複雑になります。入力フォーカスがサーバまたはXクライアントに属していない場合は、Xカラーマップはインストールされません。この場合の画面のカラーは、Windowsの標準的なカラー、または別のアクティブなWindowsアプリケーションで決められたカラーになります。XカラーマップのインストールでWindowsの標準的なカラーが使用できないようになっている必要がある場合は、[高速カラー変更]オプションが無効であると、サーバはウィンドウのフレーム、スクロールバー、タイトルバーなどに使用されるカラーを白と黒に変更します。この2つのカラーは常に使用可能であるため、これらの項目は常に表示されます。

なお、特定のディスプレイタイプやビジュアルクラスでは、デフォルトのカラーマップには白と黒しか格納されません。このため、Xサーバがシングルウィンドウモードで起動されると、画面のカラーは(Xサーバウィンドウにフォーカスがあるため)最初白と黒に変更されます。同じように、マルチウィンドウモードでは、最初のXクライアントが起動され、そのクライアントのウィンドウにフォーカスが当たると、画面のカラーが変更されます。Microsoft Windowsのアプリケーションをアクティブにすると、Windowsのカラーに復元されます。

デフォルトのXカラーマップにカラーを追加するXクライアントを起動すると、そのカラーをディスプレイ全体で使用できます。ただし、Microsoft Windowsの標準的なカラーがデフォルトのカラーマップに格納されていないと、他のWindowsアプリケーションが間違ったカラーで表示されます。このXクライ

アントが終了すると、カラーマップに追加したカラーは使用可能なままですが、カラーセルには未割り当てとマークされます。このため、デフォルトのカラーマップにカラーを追加する別のクライアントが起動されると、このカラーは以前のカラーに置き換わります。

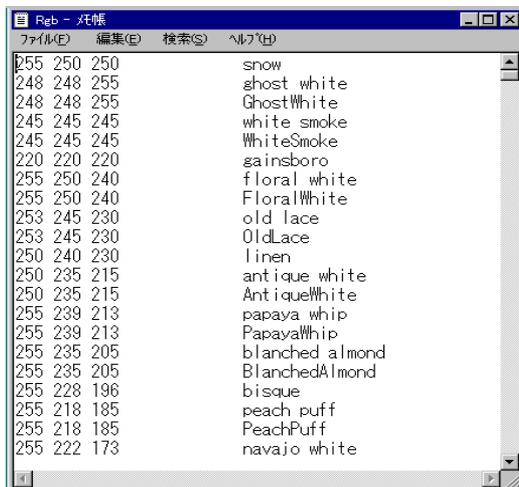
[カラー] 属性ダイアログボックスの [インテリジェントカラーマッピング] オプションを有効にすると、Microsoft Windows のアプリケーションに入力フォーカスが当たっている場合に、X クライアントを独自のカラーに近いカラーで表示できます。これによって、間違ったカラーで X クライアントが表示されるのを防ぐことができますが、Microsoft Windows アプリケーションと X クライアントを頻繁に切り換える場合には、このオプションを無効にして処理速度を上げることも可能です。

カラー名データベース

X クライアントから名前でカラーを指定できるようにするために、X では RGB 値を基準としたカラー名の検索テーブルが格納されたカラー名データベースを提供します。

X サーバはこのカラー名データベースを使用して、クライアントのカラー名による要求を RGB 値に変換します。現在のカラーマップの中で最も近い RGB 値を検索し、Microsoft Windows にそのカラーを指定します。X サーバが実行中の間、カラー名データベースは迅速にアクセスできるようにメモリ内に格納されます。

カラーデータベースは、通常 **Rgb.txt** という ASCII テキストファイルに格納されます。このファイルは、標準的な X ウィンドウシステムの配布物の 1 つで、設定プログラムによって X サーバをインストールしたフォルダに入れられ、弊社で変更されていません。カラーデータベースファイルは、[カラープロパティ] ダイアログボックスの [編集] ボタンを選択することで、表示できます。現在のカラーデータベースファイルは Windows のメモ帳に開かれます。



ファイルの各行でカラーを指定します。表示される数字は0から255までの範囲の値で、左から順にそのカラーの赤、緑、青のコンポーネントの強度を示します。また、表示される名前はカラー名で、Xアプリケーションでカラーを表すために使用されます。

コピーアンドペースト

Xサーバでは、Xアプリケーション間での情報の標準的なコピーアンドペースト方式をサポートします。さらに、Xクライアント、キャラクタ端末アプリケーション、Microsoft Windows のアプリケーション間でデータのコピーが可能です。

このセクションの内容は次のとおりです。

- Xにおけるコピーアンドペースト
- WindowsとX間でのコピー
- グラフィックのコピー

Xにおけるコピーアンドペースト

Xウィンドウシステムでは、クライアント間で情報を受け渡す方式が提供されており、ウィンドウ間でのコピーアンドペーストが可能です。主な方式はセレクションメカニズムと呼ばれ、データをグローバルな名前付きのメモリーオブジェクトつまり“アトム”に関連付けるものです。セレクションには“所有者”があり、情報の受け取り側のクライアントは“要求者”と呼ばれます。これは対話形式のメカニズムで、所有者と要求者に責任を付与することで、情報をスムーズに受け渡すことができます。

この他にも、もっと簡単なカットバッファと呼ばれる情報の受け渡し方式があります。これは、クライアントがアトムとデータを関連付け、後で別のクライアントがこれを検索する受動動作です。あらかじめ定義されているカットバッファアトムは8個あり、ルートウィンドウの属性として実装されています。

Xサーバは、セレクションメカニズムとカットバッファの両方をXクライアント間での情報の受け渡し方法としてサポートします。また、Xセレクションアトムまたはカットバッファアトムに関連付けられているデータをMicrosoft Windowsのクリップボードでミラー化できます。このため、XクライアントとMicrosoft Windowsのアプリケーション間でデータをコピーできます。[クリップボードプロパティ]ダイアログボックスを使用して、セレクションアトムとカットバッファのどちらをWindowsのクリップボードと関連付けるかを選択できます。

注意 ほとんどのXクライアントはデフォルトでは一次セレクションを使用します。このバッファはXサーバを最初にインストールしたときにWindowsのクリップボードに関連付けられているものです。なお、`cmdtool`、`shelltool`、`textedit`、`mailtool`などの一部のOPENLOOKアプリケーションではクリップボードセレクションを使用します。

Windows と X 間でのコピー

X セレクションと Windows のクリップボード間でのデータの受け渡しは、自動的に行うこともできますが、[コントロール]メニューの[編集]サブメニューのコマンドを使用して手動で行うこともできます。初期状態の設定プログラムでは自動モードが選択され、メニューコマンドはグレー表示されます。手動モードに切り換えるには、[X サーバプロパティ]ダイアログボックスの[クリップボード]タブを選択し、[カットアンドペースト]領域で[手動]をクリックします。

自動モードでのセレクション

セレクションが X サーバで所有されていない場合は、必要に応じて、セレクションの所有者からデータが要求され、Windows のクリップボードにコピーされます。

Windows のクリップボードが変更されると、X サーバは Windows のクリップボードに代わってセレクションを所有します。これ以降の X クライアントからのセレクションの要求は X サーバによって処理され、Windows のクリップボードの現在の値が返されます。

自動モードでのカットバッファ

指定されたカットバッファが X クライアントで新しい値に設定されると、その新しい値に合わせて Windows のクリップボードが変更されます。

必要であれば、カットバッファが Windows のクリップボードの値で更新されます。

手動モードでのセレクション

[X セレクションからコピー]メニューコマンドを選択した場合、セレクションが X サーバで所有されていないと、セレクションの所有者からデータが要求され、Windows のクリップボードにコピーされます。

[X セレクションへペースト]メニューコマンドを選択した場合、X サーバがセレクションの所有者になります。これ以降の X クライアントからのセレクションの要求は X サーバによって処理され、Windows のクリップボードの現在の値が返されます。

手動モードでのカットバッファ

Windows のクリップボードと X セレクション間でデータをコピーするためのメニューコマンドを選択すると、指定した操作が実行されます。

グラフィックのコピー

セレクションメカニズム(カットバッファは除く)を使用すれば、X アプリケーション間でグラフィックをコピーアンドペーストできます。X クライアントと Microsoft Windows のアプリケーション間でグラフィックをコピーするには、X セレクションを Windows のクリップボードでミラー化するためのメカニズムを使用できます。[クリップボードプロパティ]ダイアログボックスで、適切なグラフィックのクリップボードのターゲット形式(MULTIPLE)を指定する必要があります。

あります。なお、デフォルトのターゲット形式はCOMPOUND-TXTで、この形式ではグラフィックはコピーできません。

この他にも、XウィンドウのグラフィックをMicrosoft Windowsアプリケーションにコピーするための簡単な方式があります。サーバの[コントロール]メニューおよびクライアントウィンドウに用意されている[コピー領域]コマンドを使用できます。

コピー領域を使用してコピーするには

- 1 XウィンドウまたはXサーバウィンドウの[コントロール]メニューで、[編集]をポイントして、[コピー領域]をクリックします。
マウスポインタの形が変わります。
- 2 コピーする画面領域の左上隅をポイントします。
- 3 マウスのボタンを押しながら、外枠にコピーしたいグラフィック情報がすべて収まるまでドラッグします。
- 4 マウスのボタンを離します。

外枠内の領域がクリップボードにコピーされ、Windowsのペイントブラシなどのグラフィック情報を使用できるWindowsアプリケーションにコピーできます。

サウンドの扱い方

オーディオサーバは、Microsoft Windows またはそれと互換性のある環境で動作するNetwork Audio System (NetworkAudio) サーバで、Network Audio System Release 12 P1 サンプルサーバに基づいています。

オーディオサーバをインストールするには、PCにVisionソフトウェアをインストールする際に[カスタムセットアップ]を選択し、[Xコンポーネントの選択]ダイアログボックスから[Audioサーバ]オプションを選択します。オーディオサーバを既存のインストールに追加する場合には、[Maintenance Setup]を使用します。

NetworkAudioサービスによって、音声、音響、音楽などのオーディオデータをネットワーク環境で送受信するための機能がアプリケーションに提供されます。NetworkAudioサービスでは、クライアント/サーバモデルを使用して、オーディオの入出力デバイスを制御する特定のドライバからアプリケーションを切り離します。オーディオサーバは、他のホストで実行中のクライアントプログラムから送信されたメッセージに応じて、PC上の各種の入出力デバイスを制御します。

オーディオサーバでは、オーディオデータをプレイバック/レコードするために、次の形式をサポートします。

11025 kHz, Mono, 8-bit	11025 kHz, Stereo, 8-bit
11025 kHz, Mono, 16-bit	11025 kHz, Stereo, 16-bit

22050 kHz, Mono, 8-bit	22050 kHz, Stereo, 8-bit
22050 kHz, Mono, 16-bit	22050 kHz, Stereo, 16-bit
44100 kHz, Mono, 8-bit	44100 kHz, Stereo, 8-bit
44100 kHz, Mono, 16-bit	44100 kHz, Stereo, 16-bit

注意 NetworkAudio では、オーディオサーバが入力と出力の両方をサポートする場合に、両方の操作を同時に行うことを前提としています。ただし、ほとんどの PC のサウンドカードがこれに対応していないため、Vision オーディオサーバはこの機能を実装していません。オーディオの入力と出力は順番に行うことができます。

オーディオサーバを起動するには

以下のいずれかを行います。

- [スタート] ボタンをクリックして、[プログラム]、[CentreNET X Vision]、[アクセサリ] を順にポイントして、[Audio サーバ] をクリ

bクします。

- Windows NT 3.51 の場合は、プログラムマネージャで [CentreNET X Vision] グループを開き、Audio サーバのアイコンをダブルクリックします。

オーディオサーバを起動すると、タスクバーのアイコンとして表示されます。このアイコンはウィンドウに拡大できませんが、マウスの右ボタンでアイコンをクリックするか (Windows NT 3.51 ではマウスの左ボタン)、ALT+TAB を使用してアクティブなアプリケーションを選択後 ALT+SPACEBAR を使用すると、[コントロール] メニューを表示できます。

オーディオサーバは起動すると、入力または出力デバイスドライバがサポートできるオーディオ形式を検出し、デフォルトでは最初に検出されたデバイスドライバに対して該当する形式のうち最下位の形式を使用します。異なるオーディオ形式またはデバイスドライバを選択するには、オーディオサーバを構成する必要があります。

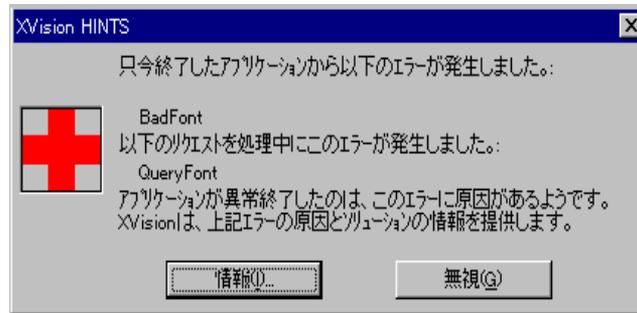
オーディオサーバを構成するには

- 1 [コントロール] メニューで、[設定] をクリックします。
- 2 オプションに対して必要な変更を行い、[OK] をクリックします。

アクティブなクライアントがあり、変更を行った結果サーバを再起動する必要がある場合、メッセージボックスが表示されます。ここで [OK] を選択すると、サーバが再起動され、アクティブなクライアントが切り離されます。

Helpful Interactive Support

HINTS (Helpful Interactive Support) メカニズムではXクライアントが起動後、正しいフォントを検出できないなどのために、エラーになったときに、助言を提供します。ダイアログボックスが表示され、エラーの種類と、そのエラーが発生したときに処理されていたプロトコル要求が示されます。



[情報] をクリックすると、考えられるエラー原因とその対処方法が表示されます。

HINTS メカニズムはXサーバで使用可能になります。

HINTS を使用可能にするには

- 1 Xサーバの [プロパティ] ダイアログボックスで、[診断] タブをクリックします。



- 2 [オプション] で、[HINTS を有効にする] がチェックされているか確かめます。

X エラーのログ

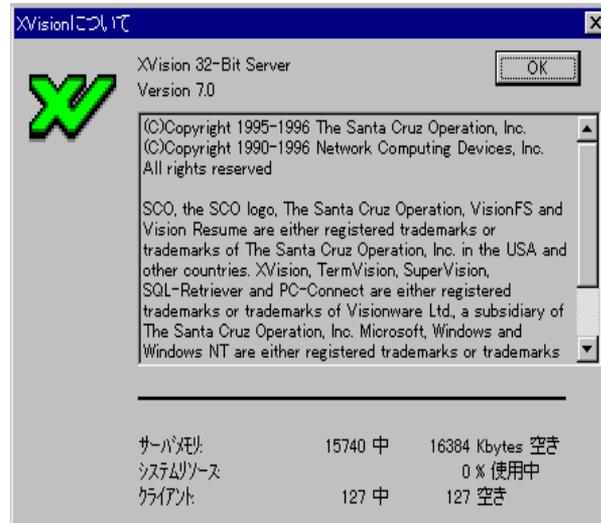
X サーバが起動すると、デフォルトでは **Xlog.txt** という名前のログファイルが作成されます。X サーバの [プロパティ] ダイアログボックスの [診断] タブで特定のオプションを有効にしていらない限り、最低限の情報がこのファイルに書き込まれます。デフォルトでは、このファイルは **C:\Program Files\Vision\User** または Windows NT 3.51 では **C:\Vision\User** に作成されますが、これはダイアログボックスで変更できます。ダイアログボックスの [表示] ボタンを使用すれば、この ASCII テキストファイルの内容を表示できます。

Xサーバに関する情報の検索

Xサーバのバージョン番号、メモリの使用率の詳細、およびサーバを使用しているXクライアントの数などの情報を検索できます。

Xサーバに関する情報を検索するには

- 以下のいずれかを行います。
 - マウスの右ボタンを使用してタスクバーの[Xサーバ]アイコンをクリックし、[ヘルプ] [X Vision について]をクリックします。
 - または-
 - Windows NT 3.51 では、サーバが透過的に実行されている場合、ルートメニューから[X Vision について]をクリックします。サーバの[コントロール]メニューにアクセスできる場合は、[コントロール]メニューから[X Vision について]をクリックします。



- 情報を参照し終わったら、[OK]をクリックします。

Windowsの日本語入力を使用するには

Windowsの日本語入力機構を利用してXクライアントに対して日本語入力を行うことができます。ここでは、次に示す項目についての説明をします。

- サポートするXプロトコル
- サポートするWindowsの日本語入力機構
- 動作
- 設定

サポートするXプロトコル

Xウィンドウの国際化テキストの入力のためのプロトコルは、いくつか種類が存在します。UNIXベンダー各社が日本語版で標準として用意しているプロトコルを中心にサポートします。具体的なサポートプロトコルはリリースノートを参照下さい。

サポートするWindowsの日本語入力機構

Windowsの国際化テキスト入力機構(IME)として標準で用意されているものは、現在以下のものがあります。

- MS - IME95
- MS - IME97
- MS - IME98
- MS - IME2000

本製品では、この標準の日本語入力をサポートします。

動作

- 入力モードの切り替え

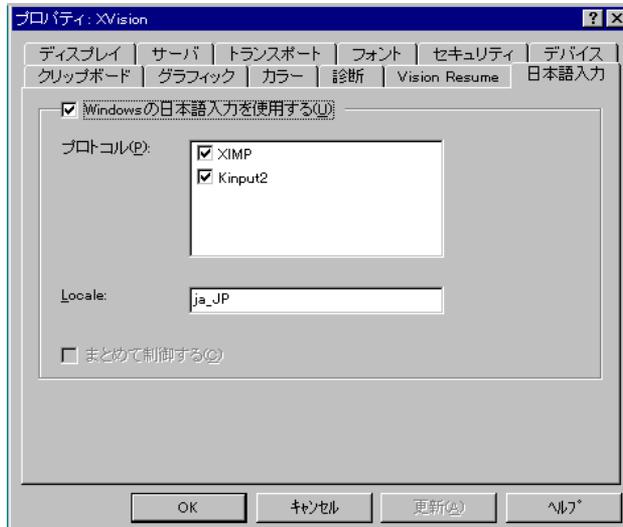
X Visionの標準の状態で、“漢字キー”(デフォルトでは[Alt]+[半角/全角])を使用して、日本語入力モードのON/OFFの切り替えを行います。この設定は、Windowsをお使いの方であれば、通常お使いの操作方法と同じように、Xウィンドウのアプリケーションでも日本語入力をお使いいただけます。

- 漢字入力中の動作

漢字を入力そのものは、Windowsの日本語入力機構(IME)が行います。したがって、入力の方法、変換の方法、確定の方法などの操作方は、Microsoft社の提供するマニュアルを参照して下さい。

- 設定について

日本語入力の動作については、[X Visionプロファイル] の [プロパティ] の [日本語入力] タブで設定画面が表示されます。



[Windowsの日本語入力を使用する] を選択することで、Windowsの日本語入力を使用してUNIXのXクライアントに日本語入力が可能になります。アプリケーションによってはUNIX側に設定が必要になる場合もあります。また、この項目を選択しない場合、この画面の他の設定項目は無効になり、シャドウがかかります。

[プロトコル] リスト中に使用するXプロトコルの一覧が表示されます。そのなかから、使用するプロトコルをチェックします。

[Locale] には、使用するアプリケーションのXlibが使用するロケールを入力して下さい。

[まとめて制御する] を選択すると、X VisionのXサーバ全体で共通の日本語入力の状態となります。デフォルトでは、ここは選択されません。選択しない場合には、各クライアントウィンドウ毎に日本語入力状態を管理します。

(空白のページです)

第5章

X キーマップの使用

このセクションの内容は次のとおりです。

- キーマップファイルの形式
- キーマップの修正
- キーマップのコンパイル
- 新しいキーマップの作成
- キーボードシステムの情報
- Windows キーコードの表示

キーマップファイルの形式

Vision X サーバのキーマップファイルは、次の4つのセクションに分かれます。

- キーマップ記述
- キーマップテーブル
- NumLock テーブル
- 修飾子テーブル

キーマップ記述

キーワード DESC をキーマップ記述の前に指定し、キーマップ記述は 128 文字以内で指定します。次の例を参照してください。

```
DESC IBM 101/102 keymap - USA
```

キーマップ記述は [Vision X サーバ] 構成の [デバイスプロパティ] ダイアログボックスにあるキーマップリストにあります。

キーマップテーブル

ここで説明するエントリの形式は次のとおりです。

```
virtual-key extended-bit flags keysym1 keysym2 keysym3 keysym4
```

ここで、*virtual-key* は Microsoft Windows の仮想キーコードであり、*extended-bit* はキーが拡張キーであるかどうかを示します。*flags* は特殊キー特性を示し、*keysym1* から *keysym4* までは X キーシンボルを示します。最初のキーシンボルは、修飾キーが押されずにそのキーだけが押されたときに使用されます。2 番目のキーシンボルは SHIFT キーが押されているとき、3 番目のキーシンボルは Mod2 (Mode Switch) が押されているとき、4 番目のキーシンボルは SHIFT キーと Mod2 キーが同時に押されているときに使用されます。

仮想キーとキーシンボルには、両方ともキーコードと名前が示されています。括弧内の名前は実際にはコンパイラは対象にせず、ファイルをわかりやすくするためのものです。

extended-bit は、常に 0 または 1 のいずれかであり、キーボード上で同じ名前を持つキーを見分けるために使われます。たとえば、キーボードには 2 つの CONTROL キーと、2 つの ENTER キー (1 つは数字キーパッド上にある) があります。

フラグは、キーが次の特殊タイプのいずれのキーであるのかを示します。

- 0x01 (S) ショートカットキー (「X サーバの使い方」の章の「ショートカットキー」を参照)。
- 0x02 (T) トグルキー。トグルキーは CAPS LOCK、NUM LOCK、SCROLL LOCK のように、押したときに 2 つの状態を切り替えるキーです。
- 0x04 (M) 修飾キー
- 0x08 (N) NUM LOCK キーに影響されるキー
- 0x10 (U) キーのトグル状態を UP、つまり、トグルされたままにしておくことが必要なキーです。たとえば、LK250 キーボード上の NUM LOCK キーがあります。これはキーパッドで番号の打ち込みを強制するために必要です。
- 0x20 (D) キーのトグル状態を DOWN のトグル状態にしておくことが必要なキーです。LK250 キーボード上の SCROLL LOCK キーなどがあります。

このファイルを編集するときには、数値を変更する必要があります。括弧内のテキストはコンパイラによって自動的に更新されます。もちろん、括弧内のテキストは手動で変更できますが、コンパイル時にコンパイラによって上書きされます。

キーマップテーブルの例を次に示します。

```

; X Vision Keymap
;
; Fields: VKcode  (VKcode name)  Extended Key  Flags (Flag letter)
;          KeySym  (KeySym name)  ...
; Flags field: 0x01=(S)ShortcutKey,    0x02=(T)ToggleKey,
;              0x04=(M)ModifierKey,    0x08=(N)NumlockableKey,
;              0x10=(U)ForceToggleUp,  0x20=(D)ForceToggleDown
; KeySym fields: Unshifted  Shifted Modifier2  Shift+Modifier2

0x08 (VK_BACK)      0 0x00      0xff08 (BackSpace)  0xff08 (BackSpace)
0x09 (VK_TAB)      0 0x01 (S)    0xff09 (Tab)        0xff09 (Tab)
0x0c (VK_CLEAR)    0 0x08 (N)    0xff0b (Clear)
0x0d (VK_RETURN)   0 0x00      0xff0d (Return)     0xff0d (Return)
0x0d (VK_RETURN)   1 0x00      0xff8d (KP_Enter)   0xff8d (KP_Enter)
0x10 (VK_SHIFT)    0 0x04 (M)    0xfe1 (Shift_L)
0x11 (VK_CONTROL)  0 0x04 (M)    0xfe3 (Control_L)
0x11 (VK_CONTROL)  1 0x04 (M)    0xfe4 (Control_R)
.
.
.
0x30 (VK_0)        0 0x00      0x0030 (0)          0x0029 (parenright)
0x31 (VK_1)        0 0x00      0x0031 (1)          0x0021 (exclam)
0x32 (VK_2)        0 0x00      0x0032 (2)          0x0040 (at)
0x33 (VK_3)        0 0x00      0x0033 (3)          0x0023 (numbersign)
.
.
.
0x41 (VK_A)        0 0x00      0x0061 (a)          0x0041 (A)
0x42 (VK_B)        0 0x00      0x0062 (b)          0x0042 (B)
0x43 (VK_C)        0 0x00      0x0063 (c)          0x0043 (C)
0x44 (VK_D)        0 0x00      0x0064 (d)          0x0044 (D)

```

NumLock テーブル

NumLock テーブルは、ローカル NumLock 管理が無効になっているときに使用されます。この場合には、数字キーボード上のキーは常にキーボードの番号をクライアントに送る必要があります。NUMLOCK キーの状態が数字以外の値を生成している場合には、X サーバはこのテーブルを使用して受信する仮想キーを数字(生の)仮想キーに変換します。この数字仮想キーはその後通常通りキーマップテーブルにあるマッピングを使用してキーシンボルに翻訳されます。

NumLock テーブルのエントリは、キーワード NUMLOCK で示され、仮想キーコードと拡張ビット設定の2組の組み合わせで構成されます。左側の組は、右側の組に翻訳されます。

次に NumLock テーブルの例を示します。

```
; NumLockTable
;
; NumlockID      MainVKcode      MainEx      RawVKcode      RawEx
NUMLOCK 0x000c   (VK_CLEAR)    0           0x0065 (VK_NUMPAD5)   0
NUMLOCK 0x0021   (VK_PRIOR)    0           0x0069 (VK_NUMPAD9)   0
NUMLOCK 0x0022   (VK_NEXT)     0           0x0063 (VK_NUMPAD3)   0
NUMLOCK 0x0023   (VK_END)      0           0x0061 (VK_NUMPAD1)   0
NUMLOCK 0x0024   (VK_HOME)     0           0x0067 (VK_NUMPAD7)   0
NUMLOCK 0x0025   (VK_LEFT)     0           0x0064 (VK_NUMPAD4)   0
NUMLOCK 0x0026   (VK_UP)       0           0x0068 (VK_NUMPAD8)   0
NUMLOCK 0x0027   (VK_RIGHT)    0           0x0066 (VK_NUMPAD6)   0
NUMLOCK 0x0028   (VK_DOWN)     0           0x0062 (VK_NUMPAD2)   0
NUMLOCK 0x002d   (VK_INSERT)   0           0x0060 (VK_NUMPAD0)   0
NUMLOCK 0x002e   (VK_DELETE)   0           0x006e (VK_DECIMAL)   0
```

注意 このテーブルで NumLock 可能なキーとしてキーを定義すると、キーマップテーブルにあるそのキーのエントリにはコンパイル時に NumLockable フラグが自動的に設定されます。

修飾子テーブル

このテーブルによって、X 修飾キーそれぞれを最大 3 つの物理キーまで指定できます (物理キーコードと拡張ビットで表示されます)。たとえば、次に示す修飾子テーブルでは、左右の CONTROL キーは Control 修飾子に割り当てられています。

次に修飾子テーブルの例を示します。

```
;ModifierTable
;
; ModifierMask          VKcode      Ex ...

MOD 0x01 (SHIFT)       0x10 (VK_SHIFT)      0
MOD 0x02 (LOCK)        0x14 (VK_CAPITAL)    0
MOD 0x04 (CONTROL)     0x11 (VK_CONTROL)    0 0x11 (VK_CONTROL)  1
MOD 0x08 (MOD1)        0x12 (VK_MENU)       0 0x12 (VK_MENU)     1
MOD 0x10 (MOD2)        0xfe (VK_????)       0
```

注意 このテーブルでキーを修飾キーとして指定すると、キーマップテーブルにあるそのキーのエントリにはコンパイル時に ModifierKey フラグが自動的に設定されます。

キーマップの修正

キーマップを修正することによって、キーに対応するキーシンボルを変更したり、非標準キーボードを使用している場合にその他のキーのキーシンボルを追加できます。また、修飾キーとして使用されるキーも変更できます。

キーマップを修正するには、メモ帳を使用してテキスト形式のマップファイルを編集し、それを X キーマップジェネレータでコンパイルする必要があります (「キーマップのコンパイル」参照)。自分の要件に最も近いキーマップを使用して開始してください。変更の対象は、Vision X サーバと共に配布される標準キーマップ、または X キーマップジェネレータで作成済の新しいマップです (「新しいキーマップの作成」を参照してください)。

標準キーマップファイルを修正する場合は、ファイルのコピーを違う名前で作成し、それを新しいキーマップのベースとして使用してください。標準キーマップは上書きしないでください。[デバイス] サーバ構成ダイアログボックスの [キーマップ] リストに反映されるので、.txt ファイルの記述行を忘れずに変更してください。

注意 キーマップを変更するときには、対応する名前よりも数値を変更するようにしてください。名前はテキストファイルが保存されたときに X キーマップジェネレータによって追加されます。テキストファイルにコメント行は追加しないでください。

X キーマップジェネレータの機能を使用すると、PC のキーボードでキーを押したときに Windows のどの仮想キーコードが生成されるのかを確認できます (「Windows キーコードの表示」参照)。

ヘルプのキーワードの「`keysyms`」を参照すると、標準 X キーシンボルのリストが得られます。コンパイラはこのリストにはない値も受け付けますが、警告メッセージを表示します。

テキストエディタとして使用するメモ帳の詳細については、メモ帳のヘルプを参照してください。

ALT の Mod2 へのマッピング

Mod2 は、X の論理修飾キーの 1 つであり、他のキーの動作を修正するために使用できます。これはメインキーマップ用の Mode Switch として使用できるので重要です（「キーマップテーブル」参照）。キーマップテーブルは、修飾キーが押されていないとき、SHIFT キーが押されているとき、Mod2 キーが押されているとき、SHIFT キーと Mod2 キーの両方が押されているときについて、対象のキーが押されるときのキーの動作を定義します。

全範囲のキーマッピングを使用するためには、Mode Switch および Mod2 と、PC キーボード上のキーを関連づける必要があります。キーボードに ALT GR キーがある場合には、X サーバが標準キーマップと共に使用される際に、**ALT GR は Mode Switch を生成し、Mod2 として扱われます**。ALT GR キーがない場合には、次で説明するように、左側の ALT キーを Mode Switch や Mod2 にマップできます。

ALT を Mode Switch および Mod2 にマップするには

- 1 メモ帳を使用して、現在使用しているキーマップファイルのテキストバージョンを開きます。

キーマップファイルは、**Vision*System** フォルダ内の **Keymap** フォルダに、**C:*Program Files*Vision*System*Keymap** のような名前が存在します。各キーボードレイアウトに対して、標準 IBM PC 拡張キーボード用のキーマップや、DEC の LK250 などのベンダー固有のキーボード用の追加キーマップが用意されています。ファイル名には意味があり、**102us.txt** は US (アメリカ合衆国) の IBM キーボード用のファイルになります。

- 2 「Mykeys.txt」などの別の名前でファイルを保存します。

標準キーマップを上書きすることは避けてください。

- 3 ファイルの先頭にある DESC 行を変更して、標準キーマップと区別してください。

この記述はこの後、[デバイスプロパティ] ダイアログボックスに表示されるので、変更したキーマップを確認することができます。

- 4 ファイルの最後に移動して、修飾子テーブルを表示します。

修飾子テーブルは、修飾キーと Windows の仮想キーコードとのバインディングを示しています。US 101/102 キーのキーマップでの修飾子テーブルを次に示します。

```

; Modifier Table
;
;   ModifierMask          VKcode          Ex ...

MOD 0x01 (SHIFT)         0x10 (VK_SHIFT)          0
MOD 0x02 (LOCK)          0x14 (VK_CAPITAL)        0
MOD 0x04 (CONTROL)       0x11 (VK_CONTROL)        0 0x11 (VK_CONTROL)    1
MOD 0x08 (MOD1)          0x12 (VK_MENU)           0 0x12 (VK_MENU)      1
MOD 0x10 (MOD2)          0xfe (VK_????)           0

```

修飾子テーブルでは、最大3つまでの仮想キーコードを各修飾キーに関連づけることができます。たとえば、左右の CONTROL キーを両方ともコントロール修飾キーに関連づけます。右側と左側の CONTROL キーを区別するために拡張ビットを使用します。

最後の行では、Mod2 が 0xfe にマップされています。これは、そのようなキーのあるキーボードで ALT GR が押されたときに X サーバが使用する仮想キーコードを示しています。この例では、ALT GR は利用できないと想定しているので、Mod2 が左側の ALT キーにマップされるようにこのバインディングを変更します。修飾子テーブルを編集すると、次のようになります。

```

;ModifierTable
;
;   ModifierMask          VKcode          Ex ...

MOD 0x01 (SHIFT)         0x10 (VK_SHIFT)          0
MOD 0x02 (LOCK)          0x14 (VK_CAPITAL)        0
MOD 0x04 (CONTROL)       0x11 (VK_CONTROL)        0 0x11 (VK_CONTROL)    1
MOD 0x08 (MOD1)          0x12 (VK_MENU)           1
MOD 0x10 (MOD2)          0x12 (VK_MENU)           0

```

ここで、左側の ALT キーの割り当て (仮想キーコード 0x12、拡張ビット 0) を Mod1 から Mod2 に変更しました。右側の ALT キーは Mod1 に割り当てられたままです。

次に、左側の ALT キーで Mode Switch キーシンボルを生成させます。

- 5 [検索] メニューで [検索] をクリックして、ALT キーを押したときに生成される Windows 仮想キーコードである `iVK_MENUi` を検索します。

この検索で、キーマップのキーマップテーブルセクションに次の行が見つかります。

```
0x12 (VK_MENU)  0 0x04 (M)  0xffe9 (Alt_L)
```

この行の形式の詳細は「キーマップテーブル」セクションを参照してください。

- 6 この行を次のように編集します。

```
0x12 (VK_MENU)      0 0x04 (M)    0xff7e (Mode_switch)
```

これで、左側の ALT キーを押すと、Mode_switch キーシンボルが送信されます。そして、このキーは修飾子テーブルで Mod2 として定義されているので、Mod2 との使用のためにキーマップテーブルで指定された任意のバインディングを活性化させるために使用できます。Mod2 修飾キーを F1 ファンクションキーと共に使用する例を示している「ファンクションキーのマッピング」を参照してください。

- 7 変更したキーマップファイルを保存します。

次に、このテキストファイルをコンパイルして、バイナリのキーマップファイルを作成します。

- 8 次の操作のいずれかを行って、X キーマップジェネレータを起動します。

[スタート] ボタンをクリックして、[プログラム]、[CentreNET X Vision]、[アクセサリ]の順にポイントしてから [X キーマップジェネレータ] をクリックします。

-あるいは-

Windows NT 3.51では、プログラムマネージャの [CentreNET X Vision] グループを開いてから [X キーマップジェネレータ] アイコンをダブルクリックします。

- 9 新しいキーマップファイルを開きます。

ファイルはすぐにコンパイルされ、エラーがあればメインウィンドウ上にメッセージが表示されます。エラーを修正するには、もう一度テキストファイルを編集する必要があります。

最初は、タイトルバーに表示されるファイル名の後ろにアスタリスク (*) がついて、そのファイルがコンパイル後に保存されていないことが示されています。

- 10 [ファイル] メニューで [保存] をクリックしてテキストとバイナリの両方のキーマップを保存してから、X キーマップジェネレータを終了させます。

- 11 [X サーバプロパティ] ダイアログボックスで [デバイス] タブをクリックします。[キーボード] の下で、ドロップダウンリストから新しいキーマップをクリックします。また、ALT キーを押したことを X サーバが Windows のショートカットキーとして Windows に送らないように、[ショートカットキー] チェックボックスをクリアします。

- 12 [OK] をクリックします。サーバが再起動するまでこの変更が有効にならないことを示すメッセージが表示されます。このときに再起動してもよいのであれば、[はい] ボタンを選択してサーバを再起動します。

新しいキーマッピングは、サーバが再起動すると有効になります。

ファンクションキーのマッピング

キーマップ修正の例として次に追加の X キーシンボルを F1 ファンクションキーに割り当てる次の手順を示します。これを行うと、PC のキーボードには存在しない他のキーボード上のキーに関連づけられたキーシンボルを生成できます。たとえば、Sun ワークステーションのキーボードにはマップ対象になるキーが複数あります。キーシンボルのリストについては、ヘルプのキーワードの「keysyms」を参照してください(Sun キーシンボルは XKeysymDBrfkeysyms_xkeysymbd 文字セットにあります)。

キーシンボルを F1 キーにバインドするには

- 1 メモ帳を使用して、現在使用しているキーマップファイルのテキストバージョンを開きます。

キーマップファイルは、**Vision*System** フォルダ内の **Keymap** フォルダに、**C:*Program Files*Vision*System*Keymap** のような名前が存在します。各キーボードレイアウトに対して、標準 IBM PC 拡張キーボード用のキーマップや、DEC の LK250 などのベンダー固有のキーボード用の追加キーマップが用意されています。ファイル名には意味があり、**102us.txt** は US (アメリカ合衆国) の IBM キーボード用のファイルになります。

- 2 **iMykeys.txti** などの別の名前で作成したファイルを保存します。

標準キーマップを上書きすることは避けてください。

- 3 ファイルの先頭にある **DESC** 行を変更して、標準キーマップと区別してください。

この記述はその後、[デバイスオプション] ダイアログボックスに表示されるので、変更したキーマップを確認することができます。

- 4 [検索] メニューで [検索] をクリックして、F1 キーを押したときに生成される Windows 仮想キーコードである **iVK_F1i** を検索します。

この検索で、キーマップのキーマップテーブルセクションに次の行が見つかります。

```
0x70 (VK_F1) 0 0x00 0xffbe (F1)
```

この行の形式の詳細は「キーマップテーブル」セクションを参照してください。左側のカラムにある各 Windows の仮想キーコードは、最大 4 つのキーシンボルにマップできます。上記の例では、右側のカラムに指定されているのは 1 つのキーシンボル (F1) だけです。これは、修飾キーの状態にかかわらず、F1 を押すと F1 キーシンボルが常に生成されることを意味します。この例では、修飾キーの状態に従って F1 キーが異なるキーシンボルを生成するように、いくつかのキーシンボルを追加します。

- 5 次のようにこの行を編集します。

```
0x70 (VK_F1)          0 0x00          0xffbe(F1)          0x0041 (A)          0x0042 (B)
                                                                0x0043 (C)
```

これで、修飾キーを押していないときに、F1 キーを押すと、通常の F1 が送信されます。SHIFT キーが押されているときに F1 を押すと iAi が生成されます。Mod2 を押しているときに F1 を押すと iBi が生成され、SHIFT キーと Mod2 キーを両方とも押しているときに F1 を押すと iCi が生成されます。

注意 キーボードに ALT GR キーがある場合には、そのキーは Mod2 として扱われます。キーボードに ALT GR がない場合には、「ALT の Mod2 へのマッピング」を参照して別のキーを Mod2 にマップしてください。

- 6 変更したキーマップファイルを保存します。

次に、このテキストファイルをコンパイルして、バイナリのキーマップファイルを作成します。

- 7 次の操作のいずれかを行って、X キーマップジェネレータを起動します。

[スタート] ボタンをクリックして、[プログラム]、[CentreNET X Vision]、[アクセサリ]の順にポイントしてから [X キーマップジェネレータ] をクリックします。

-あるいは-

Windows NT 3.51では、プログラムマネージャの [CentreNET X Vision] グループを開いてから [X キーマップジェネレータ] アイコンをダブルクリックします。

- 8 新しいキーマップファイルを開きます。

ファイルはすぐにコンパイルされ、エラーがあればメインウィンドウ上にメッセージが表示されます。エラーを修正するには、もう一度テキストファイルを編集する必要があります。

最初は、タイトルバーに表示されるファイル名の後ろにアスタリスク (*) がついて、そのファイルがコンパイル後に保存されていないことが示されています。

- 9 [ファイル] メニューで [保存] をクリックしてテキストとバイナリの両方のキーマップを保存してから、X キーマップジェネレータを終了させます。

- 10 [X サーバ プロパティ] ダイアログボックスで [デバイス] タブをクリックします。[キーボード] の下で、ドロップダウンリストから新しいキーマップをクリックします。

- 11 [OK] をクリックします。サーバが再起動するまでこの変更が有効にならないことを示すメッセージが表示されます。このときに再起動してもよいのであれば、[はい] ボタンを選択してサーバを再起動します。

新しいキーマッピングは、サーバが再起動すると有効になります。

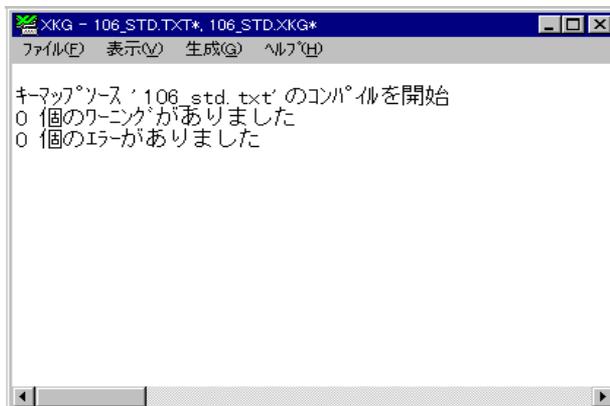
キーマップのコンパイル

テキスト形式のキーマップ(.txt)は、Xサーバが使用するバイナリ形式(.xkg)にコンパイルできます。

キーマップをコンパイルするには

- 1 次のいずれかを行います。
 - [スタート] ボタンをクリックして、[プログラム]、[CentreNET X Vision]、[アクセサリ]の順にポイントしてから[X キーマップジェネレータ]をクリックします。
 - Windows NT 3.51では、プログラムマネージャの[CentreNET X Vision] グループを開き、[X キーマップジェネレータ] アイコンをダブルクリックします。
- 2 [ファイル] メニューで[開く]をクリックします。
- 3 [ファイルの種類] ボックスで[キーマップソース]をクリックします。
- 4 [ファイル名] ボックスにあるファイルを選択します。

選択したファイルがロードされ、コンパイルされます。この時にエラーメッセージや警告メッセージが表示されることもあります。エラーがなくても次のようなメッセージが表示されます。



ウィンドウのタイトルは変更され、現在ロードされたキーマップのテキストとバイナリ形式のファイル名が示されます。ファイル名の後ろにあるアスタリスクは、このファイルには未保存の変更があることを示しています。テキスト形式ファイルを保存するときには、Windows 仮想キーコードとXキーシンボルが名前に拡張されるので、テキストファイルには常に保存用マークが付きま

テキスト形式キーマップの作成

バイナリ形式のキーマップ (.xkg) もロードできます。また、X キーマップジェネレータは、対応するテキスト形式マップ (.txt) を作成します。ここで、テキスト形式ファイルをメモ帳で変更し、これを再コンパイルすることができます。「キーマップの修正」参照してください。

テキスト形式キーマップを作成するには

- 1 [X キーマップジェネレータ] で [ファイル] メニューをクリックしてから [開く] をクリックします。
 - 2 [リストファイルタイプ] ボックスの [バイナリファイル] をクリックします。
 - 3 [ファイル名] ボックスにあるファイルを選択し、[OK] をクリックします。
- ファイルがロードされ、X キーマップジェネレータは対応するテキスト形式マップを作成します。[ファイル] メニューから [名前を付けて保存] コマンドを使用して、ファイルを保存できます。

新しいキーマップの作成

X キーマップジェネレータは、現在の Windows キーボードドライバに問い合わせることで新しいキーマップを作成できます。キーマップを作成する前に、次を指定する必要があります。

- 新しいキーマップがベースとするテンプレート
- 必要なユーザ検証の量

新しいキーマップを作成するときには、空のファイルで作成を始めることも ([ファイル] メニューから [新規] コマンドを選択)、既存のキーマップをロードして ([ファイル] メニューから [開く] コマンドを選択)、現在のキーボードドライバに必要な変更を行って上書きすることもできます。

テンプレートには、事前に設定された仮想キーコードと X キーシンボルとの間の翻訳がいくつか含まれています。これらのキーシンボルは、通常、0xFF よりも上の値になります。また、テンプレートにはデフォルトの NumLock テーブルと修飾子テーブルがあり、また、デフォルトのショートカットキーとトグルキーのフラグ集合もあります。自分のキーボードのタイプに最も近いと思われるテンプレートを選択してください。

[確認しない] を選択すると、ユーザが何も指定しなくても X キーマップジェネレータが新しいキーマップを作成します。この場合は、キーマップ内の既存のエントリが、キーボードドライバまたはテンプレートから取得した値に対応して変更されます。[全ての変更を確認する] を選択すると、キーマップに変更があるか、または追加があるかを確認するメッセージが表示されます。[ANSI 以外を確認する] を選択すると、マップの中で 0xFF よりも大きなキーシンボル値 (ANSI 以外) でキーマップエントリに変更を行った場合に確認のメッセージが表示されます。

新しいキーマップを作成するには

- 1 [X キーマップジェネレータ] で [ファイル] メニューをクリックしてから [新規] をクリックするか、[開く] をクリックしてファイルを選択し、既存のキーマップで開始するか、いずれかを行います。
- 2 [生成] メニューで、自分が持っているキーボードのタイプに似ているテンプレートをクリックします。
たとえば、標準 IBM PC 拡張キーボードを持っている場合には、[IBM 101/102 Template] を選択します。
- 3 [生成] メニューで必要な検証をクリックします。
- 4 [生成] メニューで [キーボードマップの生成] をクリックします。
新しいキーマップが生成され、ユーザが指定した検証レベルに従って、検証メッセージが表示されたり、されなかったりします。
- 5 [ファイル] メニューで [名前を付けて保存] を選択して新しいキーマップを保存します。

標準キーマップファイルは上書きしないように注意してください。

Windows キーコードの表示

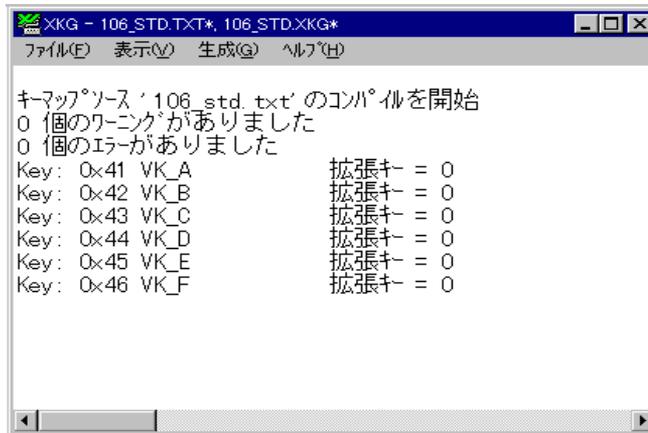
X キーマップジェネレータは、仮想キーコードに対応するキーを Microsoft Windows で表示するために使用できます。また、必要があれば、実際の Windows イベントメッセージも表示できます。

Windows 仮想キーコードを表示するには

- 1 [X キーマップジェネレータ] で [表示] メニューをクリックしてから [標準] または [詳細] をクリックします。

[標準] メッセージでは、それぞれのキーを押すと、仮想キーコードとそのキーに対応する拡張ビット設定が表示されます。長いメッセージを要求した場合には、実際の Windows のイベントメッセージが Windows WinView プログラムで使用された形式で表示されます。

- 2 任意のキーを押します。



- 3 メッセージの表示が終わったら、[表示] メニューをクリックしてから [なし] をクリックします。

この機能は、キーマップを修正するときに、変更するキーに対応する仮想キーコードと拡張ビット設定を確認するときに使用してください。

(空白のページです)

第6章

X ウィンドウの管理

このセクションの内容は次のとおりです。

ウィンドウ管理の概要

Vision ウィンドウマネージャ

Motif ウィンドウマネージャ

OPEN LOOK ウィンドウマネージャ

ウィンドウ管理の概要

X ウィンドウシステムモデルでは、X クライアントのトップレベルウィンドウのルックアンドフィールはウィンドウマネージャによって定義すべきであると規定されています。通常、ウィンドウマネージャは、サーバと他のクライアントとのやり取りを行う独立したプログラムとして実行されます。

Vision X サーバは、ウィンドウ管理方針の選択肢を提供します。サーバには、サーバが“マルチウィンドウモード”のときに動作する組み込みの Vision ウィンドウマネージャ (VWM) があります(「X サーバの使い方」章の「ディスプレイモードの選択」を参照してください)。各 X クライアントは、独立した Microsoft ウィンドウに表示され、そのウィンドウ装飾は他の Microsoft Windows アプリケーションと同じです。

サーバが“シングルウィンドウモード”で構成される場合には VWM は使用されず、独立したウィンドウマネージャが実行されます。単一の Microsoft ウィンドウ内のウィンドウ群として表示される X クライアント、および X ウィンドウの装飾とそれに対するユーザの入力は、ウィンドウマネージャによって定義されます。ウィンドウマネージャは PC 上でも UNIX ホスト上でも実行できます。ウィンドウマネージャの例としては、TWM (標準 X11R6 ディストリビューションの一部)、Motif ウィンドウマネージャの MWM、UI OpenLook ウィンドウマネージャの OLWM などがあります。

Motif ウィンドウマネージャは、サーバがシングルウィンドウモードのときに使用できる Vision X サーバパッケージで提供されます。これによって、ホストをベースにしたウィンドウマネージャを実行するようなオーバーヘッドなしに、Motif を実行する標準 X ディスプレイのルックアンドフィールを作成できます。

Vision ウィンドウマネージャ

このセクションの内容は次のとおりです。

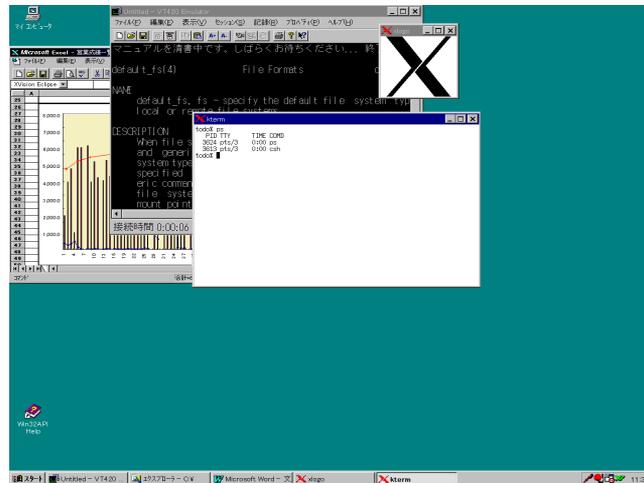
- 概要
- 初期化
- クライアントウィンドウのマッピング
- フォーカスポリシー
- カラーマップ管理
- クライアントのクローズ
- Motif との互換性
- ウィンドウプロパティ
- VWM 関数

Vision ウィンドウマネージャの概要

Vision ウィンドウマネージャ (VWM) は、X サーバに組み込まれており、サーバがマルチウィンドウモードで使用されているときに有効になります。

VWM は、X コンソーシアムが発行した『*Inter-Client Communications Conventions Manual*』(ICCCM) に準拠しています。これは、クライアントとウィンドウマネージャのやり取りのしくみを定義したものです。また、Motif と互換性を持つように設計され、Motif の機能にいくつかの追加機能もサポートしています。

VWM には、[マルチウィンドウモード] ダイアログボックスを使用して設定できる多くのオプションがあります。



注意 Microsoft Windows の使用方法を知っている場合には、VWM でほとんどのウィンドウ管理作業を行う方法をすでに知っていることになります。たとえば、ウィンドウの移動、サイズ変更、クローズは同じであり、[コントロール]メニュー上にあるメニューもほとんど同じです。詳細は、Microsoft Windows のマニュアルを参照してください。

初期化

X サーバをマルチウィンドウモードで起動すると、サーバの初期化に続いて、VWM が初期化されます。ウィンドウマネージャは効率的に最初のクライアントになります。VWM は、実際にはクライアント資源を使用せず、そのため [X Vision について] ダイアログボックス内のクライアント数にはカウントされません。

VWM用の初期化ルーチンは、ホストベースのウィンドウマネージャが起動するときに実行されるものと似たオペレーションを実行します。たとえば、ルートウィンドウ上には、WM_ICON_SIZE プロパティと_MOTIF_WM_INFO プロパティを設定します。また、トップレベルウィンドウに関連する特定のリクエストのリダイレクションを取得することで、クライアントウィンドウの管理を開始します。VWMが起動すると、他のどんなウィンドウマネージャも起動できません。

クライアントウィンドウのマッピング

クライアントは、複数のトップレベルウィンドウを作成できます。クライアントがトップレベルウィンドウを“マップ”するように依頼するまで、トップレベルウィンドウは表示されず、ウィンドウマネージャはトップレベルウィンドウを含みません。

クライアントがトップレベルウィンドウのマッピングを要求すると、目的のタイトルやアイコンなどを指定できる任意の ICCCM や Motif の “hints” を VWM が調べます。次に、VWM は、クライアントが提供するできるだけ多くのヒントを使用して、X ウィンドウを含む Microsoft ウィンドウを作成します。

このセクションの内容は次のとおりです。

- ウィンドウ装飾
- サイズと位置
- タイトルとアイコン
- 初期状態
- Motif のヒント
- Motif メニュー

ウィンドウ装飾

通常のトップレベルウィンドウでは、サイズ変更ボータ、タイトルバー、サイズボタン、[コントロール]メニューのある完全な Windows 装飾が提供され、それに相当する X ボータはオフになっています。

トップレベルウィンドウに WM_TRANSIENT_FOR プロパティセットがある場合には、VWM がサイズ変更ボータ、タイトルバー、[コントロール]メニュー付きのウィンドウを作成します。対応する X ボータはオフになります。VWM は、画面上のウィンドウのスタックの中で、作成されたウィンドウが常にその親ウィンドウの上に残ることを保証します。このタイプのウィンドウは、多くの場合ダイアログボックスに使われます。

`override_redirect` 属性セットを持つトップレベルウィンドウは、装飾を全く VWM で追加されずになしで作成されます。X ボータは変更されません。

サイズと位置

VWM は、クライアントから指定されたサイズと位置を使用して、PC 画面上に新しい Microsoft のウィンドウを表示しようとします。X ウィンドウシステムは、起点を画面の左上端に置いてピクセル座標システムを使用します。

クライアントが画面領域から外れた位置を指定すると、新しいウィンドウの起点はできるだけそのウィンドウを多く表示できるように調整されます。VWM も Microsoft ウィンドウのタイトルバーの部分が見えるように、必要があれば X クライアントの位置を変更して保証します。

[マルチウィンドウモード] ダイアログボックスの [自動設置] オプションが有効になっているときに、クライアントが初期の位置を指定しないと、VWM はデフォルトの初期配置を行います。

要求されたサイズが PC の画面領域よりも大きく、[マルチウィンドウモード] ダイアログボックスの [Windows がスクリーンより大きい] オプションが有効になっているときには、ウィンドウは目的のサイズで作成されるので、表示可能領域を移動するためにパニング機能を使用する必要があります。[Windows がスクリーンより大きい] オプションが無効になっているときには、ウィンドウは画面内におさまる程度の大きさになり、クライアントにサイズ変更通知が送信されます。この通知にどのように応答するかはクライアントの責任であり、指定された最小サイズよりもウィンドウが小さい場合には拒絶する場合があります。

クライアントは、ウィンドウに許可されている最大値や最小値のサイズを指定できます。ユーザは、この制限値を越えたウィンドウのサイズ変更はできません。VWM はウィンドウのサイズ変更がクライアントから指定されたときに、サイズ変更用に指定された増分を認識します。

タイトルとアイコン

クライアントはウィンドウの名前を指定できます。この名前は、Microsoft ウィンドウの見出しタイトルとして使用されます。また、ウィンドウが最小化されたときに使用される独立したアイコン名も指定できます。アイコン名が指定されない場合には、ウィンドウ名がアイコンタイトルとして使用されます。何も指定されていないときには VWM がデフォルトアイコンを提供しますが、使用予定のアイコンのピクセルマップもクライアントで指定できます。アイコンウィンドウは Windows NT 3.51 でもサポートされています。

初期状態

クライアントは、初期状態ではウィンドウの最小化 (アイコンとして表示) を要求することもあります。

Motif のヒント

VWM が Motif ヒントを理解するには、[マルチウィンドウモード] ボックスの [OSF/Motif ヒント] オプションが有効になっている必要があります。Motif ヒントは、ウィンドウのフレームに適用される装飾は何であるかを指定します。このフレームは Motif ウィンドウマネージャの機能がウィンドウに適用され、入力フォーカス条件がクライアントウィンドウから強制されます。詳細については、この章で後述する「Motif との互換性」を参照してください。

Motif メニュー

[Motif ヒント] が有効になっている場合には、VWM は、Motif クライアントがトップレベルウィンドウの [コントロール] メニューのカスタマイズに使用する `_MOTIF_WM_MENU` プロパティを認識します。

フォーカスポリシー

キーボードからの入力を受け付けるウィンドウは、入力フォーカスがあることになっています。X ウィンドウマネージャはウィンドウにフォーカスを提供する 2 つの手段を使用できます。明示フォーカスポリシーでは、ユーザがウィンドウ内をマウスでクリックしたときにフォーカスがウィンドウに与えられます。ポインタフォーカスポリシーでは、マウスポインタがウィンドウの内部に移動するとすぐに、ユーザがクリックしなくても、ウィンドウはフォーカスを取得します。VWM はどちらのポリシーも提供できます。

通常、Microsoft Windows は明示フォーカスポリシーを使用するので、フォーカスが与えられる前にウィンドウの中でクリックする必要があります。[マルチウィンドウモード] ダイアログボックスの [ライズした時にフォーカスする] オプションが無効になっている場合は、X クライアントウィンドウ上でクリックすると一般の Windows 方式でフォーカスが与えられます。ただし、このオプションが有効になっている場合には、マウスポインタがウィンドウ内に移動するとすぐにフォーカスが与えられます。これは、サーバが稼働している間は、X クライアントだけでなく、すべてのウィンドウに適用されます。

クライアントのトップレベルウィンドウ内では、子ウィンドウにフォーカスを与えるポリシーは、アプリケーションで使用される X ツールキットによって決められます。これは、ウィンドウマネージャによって指定された原則とは別の原則を使用して正常に機能します。

ユーザは、ウィンドウがマップされたときに自動的にフォーカスを与えることも、ウィンドウがスタックの最上部に上げられたときにも与えることも指定できます。ポインタフォーカスポリシーを使用する場合には、これらのオプションを無効にすることもあります。

カラーマップ管理

X クライアントがフォーカスを取得すると、通常、VWM はクライアントのトップレベルウィンドウに対応するカラーマップをインストールします。ただし、トップレベルウィンドウでは、親ウィンドウとは異なるカラーマップを子ウィ

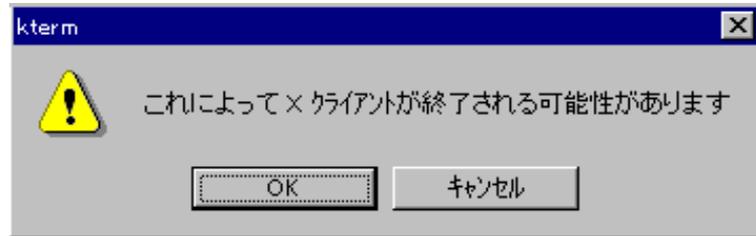
ンドウに指定するために WM_COLORMAP_WINDOWS プロパティセットを備えていることもあります。[マルチウィンドウモード] ダイアログボックスの [カラーマップの変更] オプションが有効になっている場合には、独自のカラーマップを持つ子ウィンドウがフォーカスを取得したときに、そのカラーマップがインストールされます。このオプションが無効になっているときには、VWM は子ウィンドウ用のカラーマップをインストールしません。

クライアントのクローズ

VWM がクライアントのトップレベルウィンドウに [コントロール] メニューを提供すると、ユーザは [閉じる] コマンドを選択するか、通常の Microsoft Windows 方式の [コントロール] メニューボックスをダブルクリックできます。

注意 トップレベルウィンドウがメインクライアントウィンドウであり、クライアントが適切な終了機能を持っている場合には、その方法でアプリケーションを終了させてください。

[マルチウィンドウモード] ダイアログボックスの [警告を閉じる] オプションが有効になっている場合には、ユーザが [コントロールで閉じる] を選択したときに、動作を確認するメッセージボックスが表示されます。



ウィンドウを閉じることを確認するときに、ウィンドウで WM_DELETE_WINDOW プロトコルが有効になっている場合には、VWM は実際にはウィンドウを閉じません。その代わりにクライアントにメッセージを送信して、ウィンドウを削除するのかが確認します。

WM_SAVE_YOURSELF プロトコルが有効になっている場合は、VWM はウィンドウが終了処理中であることを通知するメッセージをクライアントに送信します。それに対して、クライアントは準備を行い、準備が完了したらウィンドウマネージャにメッセージを送り返します。メッセージを受けて、VWM はクライアント接続をクローズします。クライアントが [マルチウィンドウモード] ダイアログボックスで指定されている時間内で応答しない場合には、VWM はそのままクライアントをクローズします。

クライアント接続がクローズされると、サーバによってクライアントに割り当てられていたすべての資源が解放されます。

Motif との互換性

Motif ウィンドウマネージャ (MWM) は Microsoft Windows と同じルックアンドフィールなので、Vision ウィンドウマネージャ VWM の良いモデルになります。MWM と同様に、VWM ウィンドウマネージャは ICCCM 規則に従い、そのため、Motif クライアントでの使用に制限を受けません。ただし、Motif 環境専用で作成されたクライアントによりよいサポートを提供できるように VWM には Motif 固有の機能が組み込まれています。[Motif ヒント] へのサポートは、[マルチウィンドウモード] ダイアログボックスで有効にも無効にもできます。

VWM でサポートされる MWM 関数と Motif ウィンドウプロパティについては、この章で後述します。

VWM が資源ファイルまたは資源マネージャプロパティを処理しないことに注意してください。

“奥へ” の追加コマンド

[マルチウィンドウモード] ダイアログボックスには、クライアントウィンドウの [コントロール] メニューに“奥へ”を追加するオプションがあります。[奥へ] を選択すると、ウィンドウスタックの最下部にあるウィンドウが画面の前面に表示されます。

ウィンドウプロパティ

VWM ウィンドウマネージャが使用するウィンドウプロパティは次の 2 種類に分けられます。

- クライアントが設定するプロパティ
- ウィンドウマネージャが設定するプロパティ

プロパティの中には、Motif ウィンドウマネージャ MWM との互換性を保つためにサポートされているものもあります。[マルチウィンドウモード] ダイアログボックスの [Motif ヒント] オプションを無効にすると、VWM がこれらのプロパティを認識しなくなりますが、Motif 以外のクライアントを実行するときには通常はこの設定は必要ありません。

xprop ユーティリティは、X ウィンドウのプロパティセットの詳細を戻します。このプログラムは **xterm** ウィンドウから実行されるので、ユーザはどのプロパティを戻すのかを指定するために X ウィンドウ内をクリックする必要があります。X サーバでは、ユーザが X クライアントウィンドウの外側をクリックすると、X ルートウィンドウのプロパティが戻ります。複数ウィンドウ表示モードでは、Microsoft ウィンドウの装飾(タイトルバー、ボーダなど)はクライアントのトップレベルウィンドウの一部とはみなされません。

クライアントによるプロパティ設定

次に示すウィンドウプロパティは、クライアントによって設定され、VWM によって認識されます。

WM_NAME

このウィンドウ名は Microsoft ウィンドウの見出しタイトルに使用されます。

WM_ICON_NAME

このアイコン名は、ウィンドウが最小化されたときにアイコンタイトルに使用されます。

WM_NORMAL_HINTS

このプロパティは、トップレベルクライアントウィンドウに適切な範囲のサイズをウィンドウマネージャに通知するサイズヒントを提供します。X11R2 またはそれ以前のクライアントでは、VWM はウィンドウの希望の初期位置を示すヒントも認識します。X11R3 とそれ以降のクライアントでは、これらの位置ヒントは無視され、ウィンドウの作成要求や構成要求で与えられた情報が代わりに使用されます。

認識されるフィールドは次のとおりです。

フィールド 意味

x	x 座標の初期位置 (X11R2 クライアントのみ)
y	y 座標の初期位置 (X11R2 クライアントのみ)
width	初期幅 (X11R2 クライアントのみ)
height	初期の高さ (X11R2 クライアントのみ)
min_width	許可される最小幅
min_height	許可される最小の高さ
max_width	許可される最大幅
max_height	許可される最大の高さ
width_inc	幅の増分
height_inc	高さの増分
base_width	基底幅
base_height	基底の高さ

WM_HINTS

このプロパティは、オプションであるウィンドウマネージャの追加ヒントを格納します。クライアントがアイコンピクセルマップを提供しない場合には、VWM は組み込みのデフォルトアイコンを使用します。

認識されるフィールドは次のとおりです。

フィールド	意味
input	アプリケーションが使用するキーボードフォーカスモデルを指定するブール型の値
initial_state	ウィンドウがマップされるときに希望するウィンドウの初期状態を指定します
icon_pixmap	ウィンドウが最小化されるときに使用するアイコンピクセルマップを指定します
icon_window	ウィンドウが最小化されるときにアイコンとして使用するウィンドウを指定します

_MOTIF_WM_HINTS

このプロパティは、Motif クライアントがウィンドウマネージャの特定の要望を通信するために使用します。

認識されるフィールドは次のとおりです。

フィールド	意味
flags	このフィールドは、プロパティ内の他のフィールドがデータを含んでいるかどうかを示します
decorations	ウィンドウフレームにどのような装飾がつけられるかを指定します
functions	どのウィンドウマネージャ機能がクライアントウィンドウに適用されるかを指定します
input_mode	クライアントウィンドウによって強制される入力フォーカス条件を示します

クライアントが MWM_DECOR_RESIZEH 装飾と MWM_FUNC_RESIZE 関数をオフにしていると、クライアントのトップレベルウィンドウにある Microsoft Windows ボーダはオフになります。MWM_DECOR_BORDER 装飾がオンになっていると、X ボーダが Microsoft Windows ボーダに追加される形で表示されません。

ウィンドウマネージャは、装飾に対する関数に優先度を与えます。このため、たとえば、最大化関数または最小化関数がオンになっていると、対応する装飾も強制的にオンになります。

WM_TRANSIENT_FOR

クライアントがこのプロパティを設定する場合は、ウィンドウマネージャはサイズ変更ボタ、見出しつきタイトルバー、[コントロール]メニューのあるウィンドウを作成します。

override_redirect 属性セット付きのウィンドウは、VWM では装飾がなにも追加されずに作成されます。

_MOTIF_WM_MESSAGES

このプロパティは、Motif クライアントが使用し、ウィンドウマネージャから送信されたメッセージが、現在のクライアントで処理中であることをウィンドウマネージャに示します。_MOTIF_WM_MESSAGES アトムは、ウィンドウマネージャが _MOTIF_WM_MESSAGES プロパティを追跡するための WM_PROTOCOLS プロパティに含まれている必要があります。

WM_PROTOCOLS

このプロパティは、アトムのリストであり、そのそれぞれは、クライアントが積極的に参加するクライアントとウィンドウマネージャとの通信プロトコルを特定しています。これらのプロトコルは、次の表に示されており、プロトコルが有効になったときにそれぞれが組み合わされてクライアントに対して有効になります。

プロトコル	効果
WM_TAKE_FOCUS	クライアントはセットフォーカス時にメッセージを受け取ります。
WM_SAVE_YOURSELF	ウィンドウマネージャがウィンドウを閉じようとするときにクライアントがメッセージを受け取ります。クライアントは準備していることを想定されており、準備が完了したらウィンドウマネージャにメッセージを送信します。クライアントが一定の時間内に応答しない場合でも、クライアントは終了します。
WM_DELETE_WINDOW	ウィンドウはウィンドウマネージャによっては削除されません。その代わりに、ユーザがウィンドウを閉じようとするとき、メッセージがクライアントに送られます。
_MOTIF_WM_MESSAGES	ウィンドウマネージャはクライアントウィンドウ上で _MOTIF_WM_MESSAGES プロパティを監視します。このプロパティはウィンドウマネージャから送信されたメッセージのうち、クライアントが現在処理中のものは何かを示します。

_MOTIF_WM_MENU

このプロパティは、トップレベルウィンドウの [コントロール] メニューをカスタマイズするために Motif クライアントが使用します。新しいメニュー項目がメニューの最後に追加されます。

WM_COLORMAP_WINDOWS

このプロパティは、クライアントのトップレベルウィンドウのカラーマップとは異なるカラーマップをインストールする必要があるウィンドウの ID リストです。ウィンドウマネージャはこのウィンドウリストを見て、カラーマップ属性の変更を行います。

_SUN_DRAGDROP_INTEREST

VWM は、任意のトップレベルウィンドウ上でこのプロパティを探し、任意の関連アイコンウィンドウ上にある対応する SITE_FORWARD プロパティを設定して Open Windows のドラッグ&ドロップを VWM アイコンに許可します。

ウィンドウマネージャによって設定されるプロパティ

次に示すプロパティは、VWM が設定し、保守するものであり、クライアントからの参照ができます。

WM_ICON_SIZE

[Motif ヒント] オプションが有効になっている場合は、VWM はこのプロパティを X ルートウィンドウに設定します。クライアントは、このプロパティを読み取り、前述の WM_HINTS プロパティの一部として適切なサイズのアイコンピクセルマップを提供します。大きなサイズのアイコンは切り捨てられます。

_MOTIF_WM_INFO

[Motif ヒント] オプションが有効になっている場合は、VWM はこのプロパティを X ルートウィンドウ上に設定します。このプロパティは Motif クライアントに MWM 固有の情報を提供します。このプロパティにはフラグフィールドとウィンドウ ID が含まれます。Vision ウィンドウマネージャは、フラグ内で MWM_INFO_STARTUP_STANDARD 値を設定し、ルートウィンドウの ID を指定します。

WM_STATE

このプロパティは、トップレベルクライアントウィンドウ用にウィンドウマネージャによって保守されます。このプロパティは、ウィンドウの現在の状態を指定し、ウィンドウマネージャとセッションマネージャとの通信のために使用されます。

VWM 関数

VWM ウィンドウマネージャがサポートする関数を次の表に示します。これらの関数は、Motif ウィンドウマネージャ MWMによって提供される関数の一部です。詳細は『*Motif Programmer's Guide - MWM*』を参照してください。

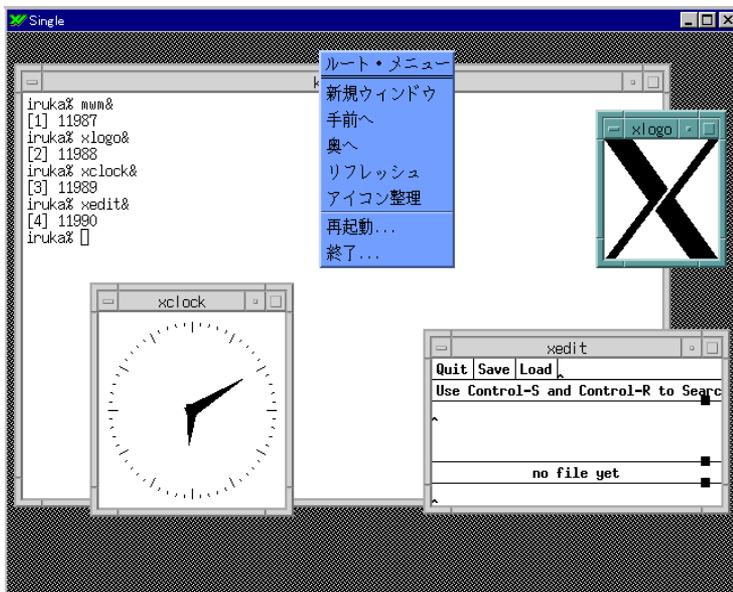
関数	説明
f.beep	警告音を鳴らします。
f.circle_down	ウィンドウをスタックの最下部に置きます。
f.circle_up	ウィンドウをスタックの最上部に置きます。
f.kill	クライアントを終了させます。
f.lower	ウィンドウをスタックの下側に下げます。
f.maximize	ウィンドウをその最大サイズまで広げます。
f.minimize	ウィンドウをアイコンに変えます。
f.move	ウィンドウを対話形式で移動できるようにします。
f.next_cmap	カラーマップフォーカスでウィンドウ内の次のカラーマップをインストールします。
f.nop	オペレーションは実行されません。
f.normalize	ウィンドウを通常のサイズで表示します。
f.prev_cmap	カラーマップフォーカスでウィンドウの1つ前のカラーマップをインストールします。
f.raise	ウィンドウをスタックの上部へ上げます。
f.raise_lower	部分的に隠されたウィンドウを上げ、隠されていないウィンドウを下げます。
f.refresh	すべてのウィンドウを再描画します。
f.refresh_win	クライアントウィンドウを再描画します。
f.resize	ウィンドウを対話形式でサイズ変更できるようにします。
f.send_msg	クライアントメッセージを送信します。
f.separator	メニューコマンド間に線を引きます。
f.title	メニューの指定位置にコマンドを挿入します。

Motif の使用

このセクションの内容は次のとおりです。

- Motif の概要
- Motif と X サーバ
- Motif ウィンドウマネージャの起動
- Motif セッションの終了
- MWM を使用するための X サーバの構成
- ホスト環境の構成

Motif GUI は、Open Software Foundation (OSF) によって開発されたものであり、ハードウェアやオペレーティングシステムの分野で数多くの企業が実装しています。PC 上で稼動する Motif ウィンドウマネージャ (MWM) は、Vision X サーバと共に提供されます。



Motif と X サーバ

Motif アプリケーションは、マルチウィンドウモードまたはシングルウィンドウモードのいずれかで X サーバを使用して PC 上に表示できます。

マルチウィンドウモードでは、クライアントウィンドウは Vision ウィンドウマネージャ (VWM) によって管理され、Microsoft Windows のようなルックアンドフィールを備えています (この章で前述した「Vision ウィンドウマネージャ」を参照)。VWM と MWM のルックアンドフィールは非常に似ており、VWM には奥へ コマンドや Motif ヒントの認識など、Motif との互換性を保つための多くの機能があります。同時に、Motif ウィンドウマネージャを使用しているユーザにとって VWM を使用しても、アプリケーションの表示の違いを意識することはありません。主な違いは、VWM が資源ファイルや RESOURCE_MANAGER プロパティを処理しないということです。

ワークステーションと同じ Motif 環境が必要な場合には、シングルウィンドウモードで X サーバを使用して、Motif ウィンドウマネージャ (MWM) を実行してください。PC 上で稼動するローカルの MWM は、Vision X サーバと共に提供され、通常、ユーザはこれを使用します。ただし、代わりにホストをベースにした MWM を使用することもあります。

ローカル MWM について

ローカル MWM は、標準セットアップを選択した場合にはインストールされません。インストール時にカスタムセットアップを選択するか、または後でローカル MWM を追加するために Maintenance Setup を使用してインストールします。

パフォーマンスを改良し、TCP/IP をインストールせずにローカル MWM を使用するには、MWM と X サーバとの通信に特別な Local トランスポートを使用します。

次に示す標準構成からの変更は、`mwmrc.txt` 構成ファイルに対して行われています (「MWM の構成」参照)。Microsoft Windows がデフォルトで Motif ショートカットの多くから入力を得るので、キーボードショートカットに変更が加えられています。

デフォルトウィンドウメニュー

CTRL 修飾がアクセラレータに追加され、以前は ALT+“キー” だったものが CTRL+ALT+“キー” になります。

資源の変更

パフォーマンスを改良するために、オペーク移動がデフォルトでオンになります。

デフォルトのキーバインディング

ALT+SPACEBAR が CTRL+ALT+SPACEBAR になりました

ALT+TAB が CTRL+ALT+TAB になりました

ALT+SHIFT+TAB が CTRL+ALT+SHIFT+TAB になりました

ALT+ESC が CTRL+ALT+n になりました

ALT+SHIFT+ESC が CTRL+ALT+p になりました

ALT+F6 が CTRL+ALT+F6 になりました

ALT+SHIFT+F6 が CTRL+ALT+SHIFT+F6 になりました

ALT+SHIFT+Delete が CTRL+ALT+SHIFT+d になりました

ローカル MWM の起動

ローカル MWM を使用する場合は、シングルウィンドウモードで X サーバを構成してください。[シングルウィンドウモード] ダイアログボックスで、X サーバの起動時にローカル MWM が起動するように [起動時にローカル Motif ウィンドウマネージャを実行する] を有効にします。また、ローカル MWM を手動で起動することもできます。

ローカル MWM を手動で起動するには

- サーバがシングルウィンドウモードで稼働しているときに、[コントロール] メニューをクリックしてから [ローカルの MWM を起動] をクリックします。サーバがフルスクリーンモードの場合には、ALT+SPACE を押して [コントロール] メニューを表示します。

ウィンドウマネージャがすでに起動している場合には、[ローカルの MWM を起動] コマンドは選択できません。

ホストをベースにした MWM の起動

ホストをベースにした MWM を使用する場合には、シングルウィンドウモードで X サーバを構成してください。

MWM を起動する最も簡単な方法は `xterm` ウィンドウから起動する方法です。`xterm` はリモートプログラムスターターで起動できます。次に、`xterm` ウィンドウが X サーバウィンドウの内部に表示されたら、システムプロンプトでウィンドウマネージャを起動する次のコマンドを入力します。

```
mwm &
```

X サーバがマルチウィンドウモードの場合には、ウィンドウマネージャがすでに動作しているので、MWM は起動しないでください。

Motif セッションの終了

通常、MWMはXサーバがクローズされると終了します。このウィンドウマネージャはルートメニューから[終了]を選択することで手動で終了できます(ルートメニューの内容は変更可能なので、サイトによっては[終了]が利用できない場合もあります)。MWMの終了は、他のアプリケーションを閉じることにはならないので、他のアプリケーションを閉じてからウィンドウマネージャを終了させてください。

MWM の構成

MWMは、資源データベースから構成されます。資源データベースはさまざまなソースから構築されています。ソースは、優先順位の低いものから高いものへ、ローカルMWMファイルを先頭にUNIXの等価名を大括弧で囲んでリストされます(末尾にUNIXの等価名が大括弧で囲んで示されます)。

RESOURCE_MANAGER ルートウィンドウプロパティ

¥Program files¥Vision¥User¥Xdefault.txt (\$HOME/.Xdefaults)

¥Program files¥Vision¥User¥Xdef-lcl.txt

(\$XENVIRONMENT または \$HOME/.Xdefaults-*host*)

mwm コマンド行オプション

表示されているパス名は、これらのファイルがWindows 2000/Me/98/95またはWindows NT 4.0上で通常置かれる位置を表しています。

資源データベースのエントリは、他の資源ファイルで特定のタイプの資源を参照することもあります。ローカル MWM では、ファイル **¥Program files¥Vision¥User¥Mwmrc.txt** (UNIX では \$HOME/.mwmrc) にはメニュー定義とキーバインディングが含まれています。

OPEN LOOK の使用

このセクションの内容は次のとおりです。

- OPEN LOOK の概要
- OPEN LOOK と X サーバ
- OLWM を使用した X サーバの構成
- ホスト環境の構成
- OPEN LOOK ウィンドウマネージャの起動
- OPEN LOOK セッションの終了

OPEN LOOK UI はソフトウェアの一種ではありません。というよりも、マルチタスクコンピュータシステムにおけるウィンドウ環境のルックアンドフィールを規定する仕様といえます。OPEN LOOK は、ハードウェアやオペレーティングシステムとして多くの企業によって実装されています。たとえば、SunSoft Inc の OpenWindows や AT&T の OPEN LOOK Graphical User Interface があります。ここで説明する図と例は、OpenWindows Version 3.4 の機能を示しています。

OPEN LOOK と X サーバ

OPEN LOOK アプリケーションは、マルチウィンドウモードでもシングルウィンドウモードでも X サーバを使用する PC 上に表示できます。

マルチウィンドウモードでは、クライアントウィンドウは Vision ウィンドウマネージャ (VWM) によって管理され、Microsoft Windows のルックアンドフィールを備えています。ローカルウィンドウ管理を使用するとパフォーマンスが向上するので、多くのユーザはマルチウィンドウモードを使用する傾向にあります。

OPEN LOOK 環境をワークステーションと同じにする場合には、シングルウィンドウモードで X サーバを使用して OPEN LOOK ウィンドウマネージャ (OLWM) を実行してください。

OPEN LOOK ファイルマネージャアプリケーションは、UNIX ファイルシステムをグラフィックに表示できます。ファイルやディレクトリはアイコンで表されます。ファイルマネージャは、ファイルのオープン、移動、コピー、削除など、標準ファイル操作手順すべてに使用できます。アプリケーションはファイルマネージャから起動できます。

制限事項

OpenWindows を使用するときには、PostScript の表示ができないので、[Help] ハンドブックにはアクセスできません。代替方法として、マニュアルページにアクセスできます。マニュアルページにアクセスするには、**SOPENWINHOME/man** ディレクトリを **MANPATH** 環境変数に含めて指定する必要があります。

OLWM を使用するための X サーバの構成

X サーバで OPEN LOOK ウィンドウマネージャを使用するには、デフォルトのサーバ構成にいくつかの変更を加える必要があります。

OLWM を使用するために X Vision を構成するには

- 1 X サーバをシングルウィンドウモードに設定します。
- 2 VGA ディスプレイを使用している場合には、[Windows システムカラー] をオフにします。
- 3 ほとんどの OPEN LOOK アプリケーションは、3 つボタンのマウスを想定していますが、ほとんどの PC では 2 つボタンのマウスで対応できます。ADJUST マウスボタンを使用する必要がある場合には、[マウスの中央ボタンのエミュレーション] がオンにする必要があります。

マウスボタンが OPEN LOOK システム内で再マップされた場合には、中央ボタンのエミュレーションはこのマッピングを反映します。

- 4 OPEN LOOK アプリケーションがサーバウィンドウ内で完全に表示されない場合には、[仮想ルート] のサイズをワークステーションモニタに一致するように設定して、ワークスペースの表示可能領域を移動できるように [スクロールバー] を有効にしてください。
- 5 特定のワークステーションキーボードの特殊キーたとえば、Sun キーボード上にある Front キーや Open キーなどを模擬する場合には、新しいキーマップを作成する必要があります。

新しいキーマップファイルの作成方法の詳細については、ヘルプのキーワードの「X キーマップ、作成」を参照してください。

- 6 OLWM を実行させる予定のホストがホストエクスプローラ内に表示されていることを確認します。

OPEN LOOK ウィンドウマネージャの起動コマンドは、通常は `olwm` ですが、使用している実装によって異なることがあります。詳細については、使用しているバージョンの OPEN LOOK のマニュアルを参照してください。

ホスト構成の詳細については、ヘルプのキーワードの「ホスト、構成」を参照してください。

ホスト環境の構成

OPEN LOOK の実行に必要なホスト構成は、サイトで使用されているバージョンによって異なります。一般に、1つ以上の環境変数を設定して、パスに OPEN LOOK ファイルのあるディレクトリを含める必要があります。たとえば、OpenWindows を使用するときには、環境変数 OPENWINHOME を設定して、主な OpenWindows ディレクトリをポイントし、このディレクトリと対応する `/bin` サブディレクトリを `PATH` 変数に入れる必要があります。

OPEN LOOK の特定のバージョンの詳細については、システム管理者に問い合わせてください。

SUN日本語OpenWindows (2, x, 3, x) を使用する場合は以下の環境変数も必要です。

```
LD-LIBRARY-PATH    /usr/uponwin/lib
```

```
OPENWINHOME       /usr/uponwin
```

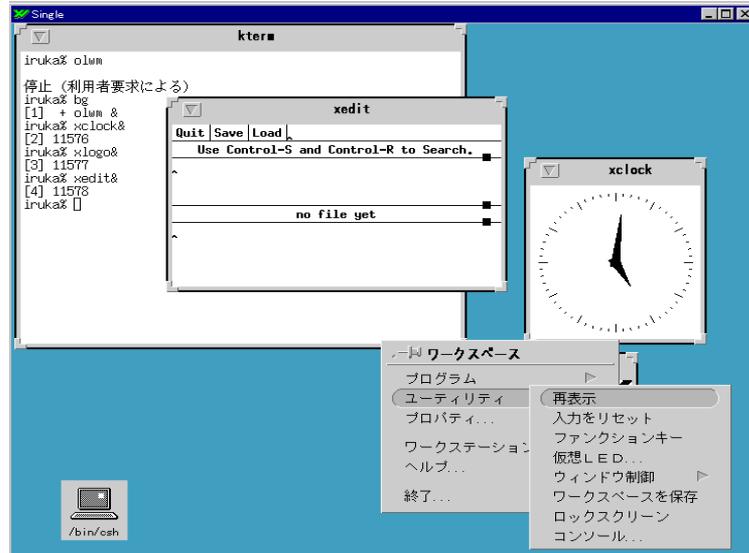
OPEN LOOK ウィンドウマネージャの起動

OPEN LOOK ウィンドウマネージャを使用する場合には、「OLWM を使用するための X サーバの構成」で説明されているように、シングルウィンドウモードで X サーバを構成してください。

OLWM を起動する最も簡単な方法は、`xterm` ウィンドウから起動する方法です。`xterm` はリモートプログラムスターターで起動できます。次に、`xterm` ウィンドウがサーバウィンドウ内に表示されら、システムプロンプトでウィンドウマネージャを起動する次のコマンドを入力します。

```
olwm &
```

X サーバがマルチウィンドウモードの場合には、ウィンドウマネージャがすでに動作しているので、OLWM は起動しないでください。



注意 Sun OpenWindows を使用するには、**openwin** コマンドは**使用しない** てください。このコマンドはOpenWindows環境をシステムコンソール上で起動するものだからです。OpenWindows がすでにコンソール上で動作している状態でPCからこのコマンドを実行すると、ワークステーションシステムが過負荷になる可能性があります。

OPEN LOOK セッションの終了

OLWM で管理されているセッションを終了するには、[ワークスペース] メニュー上のコマンドを使用します。

OPEN LOOK セッションを終了するには

- 1 MENU マウスボタンでルートウィンドウをクリックして[ワークスペース]メニューを表示します。
MENU マウスボタンは、通常は右ボタンです。
- 2 [ワークスペース]メニューで[終了]をクリックします。
- 3 [終了]をクリックしてOPEN LOOK を停止します。
[キャンセル]をクリックすると、OPEN LOOK に戻ります。
ここで必要ならXサーバを終了させることもできます。

(空白のページです)

第 7 章

X プログラムの起動

この章では次の内容を説明します。

- 概要
- Unix アプリケーションウィザードの使用
- リモートプログラムスタータの使用
- 端末エミュレータを使用した X クライアントの起動
- X Display Manager の使用
- NCD XRemote の使用 (サポート対象外)

X プログラム起動の概要

UNIX システム上で実行され、PC 上に表示される X アプリケーションの起動には、複数の方法があります。ネットワークトランスポートを使用している場合には、Unix アプリケーションウィザードを使用してアプリケーションを開くのに必要なすべての設定を含むドキュメントをセットアップすることができます。それ以外の方法としては、リモートプログラムスタータを使用して X クライアントを実行したり、端末エミュレータからクライアントを起動することもできます。

X アプリケーションで事前に構成された選択を実行する X セッションの起動には、いくつかの方法があります。リモートプログラムスタータの設定ファイルは Windows のスタートアップフォルダ、または、Zone のスタートアップフォルダに置くことができます。ネットワーク接続では、標準 `xdm` ディスプレイマネージャを使用してセッションを開始することができます。シリアルトランスポートを使用している場合には、最初の X クライアントを実行する唯一の方法は XRemote セッションを起動することです。Vision Resume がサイトで使用されている場合には、X セッションを一時停止すると、クライアントはセッションが再開されたときに自動的にリスタートされます。

Unix アプリケーションウィザードの使用

Unix アプリケーションウィザードは、UNIX アプリケーションを起動できる設定ドキュメントの作成に使用できます。次のドキュメントのタイプは、必要な X Vision コンポーネントがインストールされている場合に作成できます。

- X アプリケーション用のリモートプログラムスタータ
- 文字ベースのアプリケーション用の端末エミュレータドキュメント (VT420)
- サーバプログラムやネットワークデーモンのようにユーザインタフェースを必要としない UNIX アプリケーション用のリモートプログラムスタータドキュメント

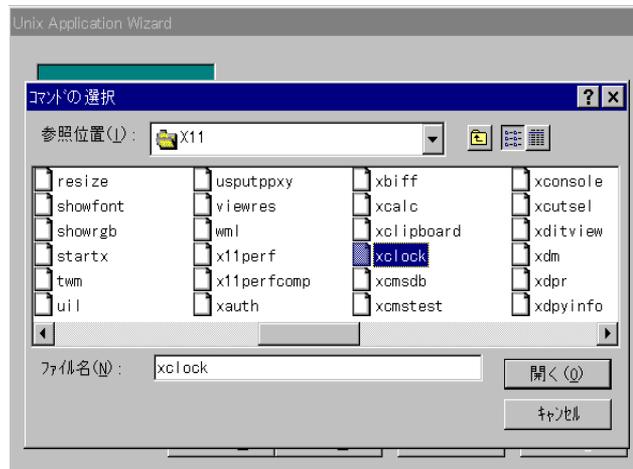
Windows 2000/Me/98/95 と Windows NT 4.0 では、ユーザがウィザードを起動する方法に従って、[スタート] メニューにある [Unix プログラムの一覧] メニューにアプリケーションが追加されるか、デスクトップやフォルダに新しい項目が作成されます。

新しい設定ドキュメントを作成するには

- 1 次のいずれか 1 つを行います。
 - [スタート] メニューでアプリケーションを表示させる場合には、[スタート] ボタンをクリックしてから [Unix プログラムの登録] をクリックします。
 - デスクトップ上またはフォルダ内にアプリケーションを表示させる場合には、その項目を表示させる場所でマウスの右ボタンをクリックしてから [新規作成] をクリックし、次に [Unix プログラム] をクリックします。
 - Windows NT 3.51 では、プログラムマネージャ内の [CentreNET X Vision] グループを開き、[Unix プログラムの登録] アイコンをダブルクリックします。



- 2 ウィザードが示す手順に従って、実行するホストとコマンドを選択します。
[参照] ボタンを使用すると必要な実行可能プログラムを見つけることができます。



- 3 プログラムのタイプを選択します。“Other”とは、ユーザインタフェースを必要としないUNIXプログラムのことです。



- 4 アプリケーションのテストを行うかどうかを決定します。ユーザが実行する前に、コマンドがホストシステム上で不必要な変更を行わないことを確認してください。

アプリケーションがXクライアントまたは“Other”の場合には、[ホストリプライ]ウィンドウにホストのオペレーティングシステムが生成したエラーメッセージがあればそれも含めたステータスメッセージが表示されます。Xクライアントが次に表示されます。



- 5 コマンドでエラーが発生した場合には、ウィザードによって表示されます。最も一般的なエラーの場合は、リストボックス内のエラーを選択し、特定の障害についての情報を示す [ヘルプ] ボタンをクリックします。[戻る] ボタンを押すと前の設定に戻るので、もう一度プログラムを実行することができます。



- 6 アプリケーション名を入力するプロンプトが表示されます。入力したアプリケーション名は設定ファイル名として使用されます。

Unix アプリケーションウィザードはリモートプログラムスタートドキュメント (.rps) または端末エミュレートドキュメント (.v42) のいずれかを作成します。[スタート] メニューからウィザードを実行した場合には、アプリケーションは [スタート] メニュー上の UNIX プログラムのリストに追加されます。デスクトップからウィザードを実行した場合には、新しいデスクトップアイコンが表示されます。フォルダの内部からウィザードを実行した場合には、設定用ドキュメントはフォルダ内に入ります。

リモートプログラムスタータの使用

リモートプログラムスタータは、X アプリケーションの起動に使用できます(「リモートプログラムの実行」を参照)。リモートプログラムスタータは、[スタート] メニューまたはプログラムマネージャから起動でき、また、設定用ドキュメントは Windows デスクトップまたは [プログラムマネージャ] グループ内のアイコンとして作成し、配置できます。

リモートプログラムスタータから X アプリケーションを起動するには

- 次のいずれか 1 つを行ってください。
 - [スタート] ボタンをクリックしてから [Unix プログラムの実行] をクリックします。
 - [スタート] ボタンをクリックしてから [プログラム]、[CentreNET X Vision]、[アクセサリ] の順でポイントし、[プログラムスタータ] をクリックします。
 - Windows NT 3.51 の場合、X サーバがシングルウィンドウモードのときには、コントロールメニューから [実行] をクリックします。
 - Windows NT 3.51 の場合、プログラムマネージャにある [CentreNET X Vision] グループを開き、[プログラムスタータ] アイコンをダブルクリックします。

新しい [リモートプログラムスタータ] ウィンドウが表示されます。



- [ホスト] ボックスには、ホスト名を入力するか、ドロップダウンリストからホストを選択します。

- 3 [コマンド] ボックスには、ホストシステムで実行するコマンドを入力します。ドロップダウンリストを使用すると、直前に使用したコマンド5つの中から選択できます。

UNIX システムのコマンド構文は、端末から切り離されてプロセスが実行されることを示すアンパーサント記号(&)を入力しないことを除いて、リモートシステム上のシステムプロンプトに入力する構文と同じです。X アプリケーションでは、デフォルトで **DISPLAY** 環境変数が自動的に設定されるので、**-display** を入力する必要はありません。

その他の方法として、[参照] ボタンをクリックします。[ホスト] ボックスに何も指定されていない場合には、構成されたすべてのホストが表示されます。[ホスト] ボックスでホストを指定した場合には、そのホスト上のユーザのホームディレクトリの内容が表示されます。実行するプログラムまたはシェルスクリプトが見つかるまで、ホストを参照してください。

- 4 その他の設定を変更する場合には、[プロパティ] メニューからコマンドを選択します。X アプリケーションを使用していないのであれば、[ディスプレイ]、[サイズ]、[カラー] の各プロパティに指定されている内容に意味はありません。
- 5 [実行] をクリックします。

数秒して X アプリケーションが起動します。[X サーバを起動] オプションを有効にしている場合には、必要に応じて X サーバが自動的に起動され、指定されたプロファイル内で実行するか、最初に検出したものを起動します。プロファイルの詳細については、ヘルプのキーワードの「サーバプロファイル」を参照してください。

リモートプログラムスタータの使用手順の詳細については、[ヘルプ] メニューの [トピックの検索] をクリックしてください。メインの [Vision ヘルプの目次] から [リモートプログラムスタータの使い方] を選択すると、同じ情報が表示されます。次の項目についてのヘルプが用意されています。

- 起動と停止
- ウィンドウ表示形態の変更
- プログラムの実行
- 設定用ファイルの使用

端末エミュレータを使用した X クライアントの起動

サーバ上で X クライアントを起動するときの最も信頼性の高い方法は、普通の端末または端末エミュレータからホストのオペレーティングシステムにログインして、ホストのシステムプロンプトからクライアントを実行することです。

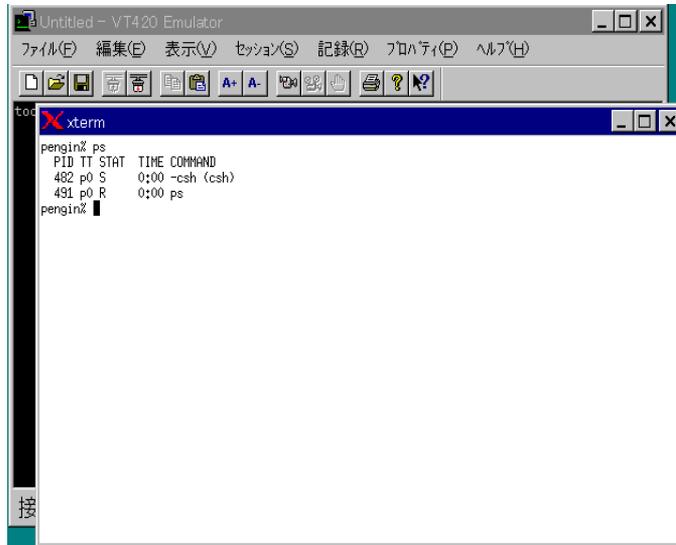
端末エミュレータから X クライアントを起動するには

- 1 次のいずれかを行います。
 - [スタート] ボタンをクリックして [プログラム]、[CentreNET X Vision] を順にポイントし、[VT420 エミュレータ] をクリックします。
 - Windows NT 3.51 ではプログラムマネージャの [CentreNET X Vision] グループを開き、[VT420 エミュレータ] アイコンをダブルクリックします。
- 2 [セッション] メニューで [接続] をクリックします。
- 3 [ホスト] ボックスでホスト名を入力するか、ドロップダウンリストからホストを選択します。
- 4 [コマンドライン] ボックスで、`ssh` など UNIX シェルの起動に必要なコマンドを入力します。

接続処理が始まり、この接続がホストへの初めての接続の場合は、ユーザ名とパスワードを入力するプロンプトが表示されます。接続処理中にはステータスメッセージを示すダイアログボックスが表示され、ウィンドウ内に UNIX システムプロンプトが表示されます。自動接続機構が作動しない場合には、[コマンドライン] ボックスに何も入力しないでもう一度試してください。これによって、自動ログインが抑止され、接続がオープンされた後に UNIX の `login` プログラムが実行されます。このプログラムはユーザのデフォルトログインシェルを起動する前にユーザ名とパスワードの入力を要求するプロンプトを表示します。

- 5 システムプロンプトが表示されたら、使用しているディスプレイ上で X クライアントを実行するために必要なコマンドを入力します（「UNIX システムプロンプトからのクライアントの起動」参照）。`DISPLAY` 環境変数がホスト上で構成されていない場合には、必ず `-display` スイッチを追加してください。たとえば、`nohup xterm -display stevel:0 &` のように指定します。

X クライアントは、使用中の PC 画面上に次のように表示されます。



その他の方法として、接続をオープンするときにXクライアントの起動に必要なコマンドを端末エミュレータの[コマンドライン]ボックスに直接入力する方法があります。

端末エミュレータを使用した接続ができない場合には、ヘルプのキーワードの「トラブルシューティング」を参照してください。

注意 PC上のXサーバは、Xクライアントの起動前に実行させておく必要があります。サーバが実行されていない場合には、[スタート]メニューをクリックしてから[プログラム]、[CentreNET X Vision]を順にポイントし、[X Vision サーバ]をクリックします。Windows NT 3.51では、プログラムマネージャの[CentreNET X Vision]グループにある[X Vision サーバ]アイコンをダブルクリックします。

UNIX システムプロンプトからのクライアントの起動

Xクライアントを含むディレクトリがPATH環境変数に含まれていない場合には、コマンドを入力するときにフルパス名を指定する必要があります。UNIXのバージョンによっては、端末接続をクローズするときにクライアントプロセスが強制終了されるのを防ぐためにnohupコマンドが必要なものもあります。

使用中のPCをXディスプレイデバイスとして指定する方法には2種類あります。DISPLAY環境変数を設定する方法と、-displayオプションを使用する方法です。

DISPLAY 環境変数を設定することで、指定したデバイス上にすべてのクライアントを表示することを指定します。TCP/IP を使用し、ログインシェルとして C シェルを使用する場合には、これを行うコマンドは次のようになります。

```
setenv DISPLAY nodename:0
```

nodename はホスト名ファイル (通常は `/etc/hosts`) に自分の PC として定義したネットワークノード名です。たとえば、次のようになります。

```
setenv DISPLAY stevel:0
```

ボーンシェルを使用する場合には、コマンドは次のようになります。

```
DISPLAY=nodename:0  
export DISPLAY
```

これらのコマンドは、ログインスタートアップファイルに含めることも、次の例のように端末エミュレータウィンドウ内に入力することもできます。

```
DISPLAY=stevel:0  
export DISPLAY
```

DISPLAY 変数が設定されていると、**nohup** とその後にプログラム名、バックグラウンドでの実行を強制するアンパーサンド記号を入力するだけで X クライアントを実行できます。次の例を参照してください。

```
nohup xclock &
```

注意 **PATH** 環境変数でクライアントを含むディレクトリのパス名が指定されていない場合は、ここでパス名を指定する必要があります。

DISPLAY 変数を設定するのは別の方法として、X クライアントを実行するときにコマンド行に **-display** オプションを使用する方法があります。ほとんどのクライアントはこのパラメータを受け付けるので、次のようにコマンドを入力することができます。

```
nohup xterm -display nodename:0 &
```

nodename は使用中の PC のネットワークノード名です。**-display nodename:0** パラメータは、*nodename* ディスプレイのサーバゼロ上にクライアントを表示することを指定します。アンパーサンド記号によって、プログラムを強制的にバックグラウンドで実行させます。たとえば、ネットワークノード名が “*stevel*” の PC 上に表示する場合は、次のように指定します。

```
nohup xterm -display stevel:0 &
```

xterm ウィンドウが使用中の PC 上に表示されると、それを別のクライアントの起動に使用できます。

X Display Manager の使用

ここでは次の内容を説明します。

- X Display Manager の概要
- X サーバと XDMCP
- X Display Manager 用 X サーバの構成
- UNIX での X Display Manager の構成
- X Display Manager セッションの起動
- X Display Manager セッションの停止
- その他の情報源

X Display Manager の概要

xdm というディスプレイマネージャは、UNIX システムにログインし、X セッションを起動する標準方法をユーザに提供するために X ウィンドウシステムと共に提供されています。メーカー固有のディスプレイマネージャを使用している場合には、オペレーティングシステムと共に提供されているマニュアルを参照してください。

xdm は、通常、UNIX システム上でデーモンとして実行され、X ディスプレイ管理制御プロトコル (XDMCP) を使用してリモートサーバとのやり取りを行います。その目的は、ユーザに対してログインを要求し、ユーザ名とパスワードの妥当性を検査してから、X「セッション」を起動することです。このセッションで起動されたクライアントの1つはセッションマネージャとして扱われ、このプロセスが終了するまでそのセッションは存続します。次に **xdm** は残りの X クライアントをクローズし、サーバのリセットまたは終了を行います。

X サーバと XDMCP

X サーバは XDMCP をサポートしますが、ユーザが利用できるかどうかは使用しているネットワークの種類に依存します。このネットワークは、ユーザデータグラムプロトコル (UDP) とブロードキャストサービスをサポートする必要があります。これによって、ネットワークは TCP/IP ネットワークに限定されません。

XDMCP を使用する場合には、サーバ内で有効にする必要があります (「X Display Manager 用 X サーバの構成」を参照)。

サーバとディスプレイマネージャとのやり取りの定義に加えて、XDMCP はどのクライアントにディスプレイへのアクセスを許可するかを制御するユーザアクセス制御方法を実装しています。詳細は、「X サーバの使い方」の「XDMCP セキュリティの使い方」を参照してください。

X Display Manager 用 X サーバの構成

カスタムセットアップを使用して X サーバをインストールすると、XDMCP ブロードキャストを有効にするかどうかのプロンプトが表示されます。このプロンプトは一般セットアップでは表示されず、デフォルトは XDMCP を無効にする設定です。X Display Manager を使用するためには、サーバプロファイルで XDMCP を有効にする必要があります。

XDMCP を有効にするには

- 1 マウスの右ボタンで [X サーバ] アイコンをクリックし、次に [プロパティ] をクリックします。
- 2 [セキュリティ] タブをクリックします。



- 3 チェックマークが表示されるように [XDMCP] をクリックします。
- 4 [XDMCP] の下で、必要なモードをクリックします。

[ダイレクト] モードまたは [インダイレクト] モードを使用している場合には、使用する予定のホストの名前を [ホスト] ボックスに入力するか、ドロップダウンリストから 1 つ選択します。

- 5 XDM-AUTHORIZATION-1 セキュリティ機構を使用するのであれば、[X Vision プロファイル] ダイアログボックスの [共通] タブにある [キー] ボックスにキーを入力しておいてください。



キーオプションの値は、システム管理者に尋ねてください。システム管理者は [ディスプレイ ID] の値も知っています。これらの値は一度登録されると変更できません。[キー] ボックスに何も指定されていない場合には MIT-MAGIC-COOKIE-1 セキュリティ機構が使用されます。詳細については「X サーバの使い方」の「XDMCP セキュリティの使い方」を参照してください。

- 6 オプションをさらに指定する必要がある場合は [拡張設定] をクリックします。

UNIX での X Display Manager の構成

`xdm` がサイトで通常の使用状態にある場合には、UNIX システムのシステムマネージャは、通常、このプログラムがシステムのブート時に起動されることを保証しています。デフォルトでは、`xdm` はバックグラウンドでデーモンとして実行されます。

`xdm` のオペレーションは `/usr/lib/X11/xdm/xdm-config` ファイルに資源を指定することで構成できます。これらの資源の中には、「X Display Manager セッションの起動」で説明されているアクセス制御ファイルや資源データベースなどの他の制御ファイルを指定しているものもあります。UNIX システムのシステムマネージャがこれらのファイルを変更することもあります。

デフォルトセッションプログラムである `/usr/lib/X11/xdm/Xsession` ファイルを参照する場合もあります。このスクリプトは、正常なログインが行われた後、各ユーザ用に実行されます。このファイルを変更できるのは、システムマネージャだけです。各ユーザに別々のセッションを許可するために、`Xsession` スクリプトは `/home/.xsession` ファイルが存在する場合には通常これを実行します。このファイルが存在しない場合には、`Xsession` は一般に `twm` ウィンドウマネージャと `xterm` で構成されるデフォルトセッションを起動します。

.xsession の作成

`.xsession` ファイルはテキストエディタを使用してホームディレクトリに作成します。このファイルには、シェルからクライアントを起動するとき使用するのと同じ構文のクライアント起動用コマンドを指定します。最後のクライアント以外のすべてのクライアントはバックグラウンドで実行してください。リストされている最後のクライアントはセッションマネージャとして使用するプログラムであり、これはフォアグラウンドで実行します。このクライアントを終了すると、`xdm` セッションは終了します。

マルチウィンドウモードで X サーバを使用している場合には、セッションファイルではウィンドウマネージャをセッションマネージャとしては起動しないでください。これを行うと、ウィンドウマネージャは実行できなくなり、セッションは終了してしまいます。ホストベースのウィンドウマネージャを実行する場合には、X サーバをシングルウィンドウモードに設定してください。

次に示すサンプルセッションファイルは、3つのクライアントを起動します。ここで、**DISPLAY** 環境変数は **xdm** によって *pc-name:0.0* (*pc-name* は使用中の PC のネットワークアドレス) に自動的に設定されます。

```
#!/bin/csh
# SCO X Vision
# X Display Manager または NCD XRemote で使用する
# サンプルセッションファイル.
# 作成者 : Steve Lewis 日付 : 1-Jul-1993
#
# このファイルを使用して xdm または Xremote セッションの起動時に
# クライアントを起動します.
# 最後のクライアント以外のすべてのクライアントはバックグラウンドで
# 実行させます. 起動された最後のクライアントは "セッションマネージャ"
# であり、フォアグラウンドで実行する必要があります.
# DISPLAY 変数は xdm または Xremote で設定されます.
#
xclock -g =80x80-0+0 -display $DISPLAY &
xbiff -g 40x40-100+150 -display $DISPLAY &
# この xterm はフォアグラウンドで実行され、セッションは、ユーザが
# クローズすると終了します.
xterm -display $DISPLAY -ls
```

ヒント このファイルを使用するには、[ヘルプ] でこのページを表示して、[ヘルプ] ウィンドウから [Edit Copy] を選択し、このテキストをメモ帳ファイルにコピーしてから、UNIX システムにファイルを [ファイル転送] します。

ログインと X セッションの起動にも XRemote を使用する場合は (「NCD XRemote の使用」を参照)、**.xsession** ファイルを **.xinitremoterc** (XRemote 用のスタートアップファイル) にコピーするか、**XINITRC** 変数を **.xsession** ファイルに指定することで、XRemote 環境を **xdm** 環境と同じにすることができます。

注意 **.xsession** ファイルには実行パーミッションを与える必要があります。次の例を参照してください。

```
chmod +x .xsession
```

Xdm ping の間隔

xdm は、通常の間隔でサーバに ping を行うことで、サーバがまだアクセス可能なことを確認します。サーバが応答しなければ、**xdm** はセッションが終了したと宣言します。この方法は、ユーザが MS-DOS 互換ウィンドウを使用しているときや、サーバが非常にビジーな状態の場合には問題になる可能性があります。UNIX システムマネージャは、**DisplayManager.DISPLAY.pingInterval** 資源を 0 に設定することで、**xdm** が ping を行わないように設定できます。

xdm ログファイル

デフォルトでは、セッションがシステムコンソールにエラーメッセージを書き込んでいる間に、**xdm** プログラムおよびさまざまなスクリプトやプログラムが実行されています。ただし、**DisplayManager.errorLogFile** 資源を設定するために **xdm-config** ファイルにエントリがあるはずで、通常、**/usr/lib/X11/xdm/xdm-errors** ファイルにエラーが書き込まれるようにこれが設定されます。

Xsession スクリプトは、**/home/.xsession-errors** などのユーザのホームディレクトリにあるファイルに、セッションを起動している間に発生したエラーをリダイレクトすることができます。

X Display Manager セッションの起動

XDMCP が利用可能なときは、X サーバは起動時やリセット時にネットワーク上にメッセージを送ります。このメッセージのタイプは、[セキュリティプロパティ] ダイアログボックスの指定に合わせて、ダイレクト、インダイレクト、またはブロードキャストのいずれかになります。**xdm** の応答の仕方は、これ以降で説明するように、メッセージのタイプによって異なります。

アクセス権限のチェック

X サーバが XDMCP 要求を行うと、応答は、通常 **/usr/lib/X11/xdm/Xaccess** というホスト上の XDMCP アクセス制御ファイルによって判別されます。直接クエリーやブロードキャストクエリーでは、**xdm** は要求を行った PC にこのコンピュータへのアクセスが許可されているかどうかをチェックします。**xdm** は **Xaccess** ファイル (パターンマッチング用のワイルドカードがある) にある直接 / ブロードキャストタイプのエントリをチェックしてから、ユーザの PC のネットワークノード名に一致する最初のエントリが応答を判断します。一致するエントリには、アクセスの許可または禁止が定義されており、一致するものが存在しなければアクセスは許可されません。

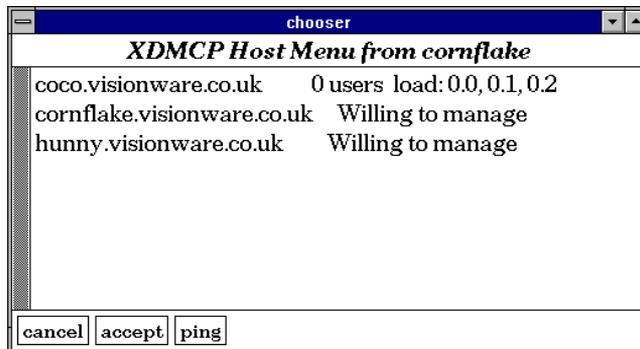
間接クエリーでは、ユーザの PC のネットワークノード名が **Xaccess** ファイル (パターンを含む) の間接タイプエントリに一致する必要があります。間接タイプのエントリは、単一ホスト、または管理サービスが要求されたホストのリストを指定しているはずで、代替方法として、間接エントリで **xdm** が **chooser** を実行して、接続するホストのメニューを提供する方法もあります (「自発的なディスプレイマネージャの選択」参照)。**chooser** を指定するエントリには、ホストのリストが含まれているか、**chooser** に対して自発的なディスプレイマネージャを見つけるようにブロードキャストを命令していることもあります。

自発的なディスプレイマネージャの選択

ブロードキャストクエリーでは、**xdm** はディスプレイを自発的に管理するというメッセージで応答します。[XDMCP 拡張設定] ダイアログボックスの [最初のマネージャを選択] オプションが有効になっていると、最初に応答するマネージャに対してディスプレイの管理を依頼します。オプションが有効になっていない場合には、ユーザが自発的なディスプレイマネージャのリストから選択できるように、X サーバが [ディスプレイマネージャセレクション] ダイアログボックスを表示します。



間接クエリーにおいて、**chooser** が **Xaccess** ファイル（「アクセス権限のチェック」参照）で指定されている場合には、[XDMCP 拡張設定] ダイアログボックスの [最初のマネージャを選択] オプションを有効にして、[ディスプレイマネージャセレクション] ダイアログボックスが表示されないようにします。**chooser** が自身で持っている自発的なディスプレイマネージャのリストを表示するので、そのうちの1つを選択し、[了解] ボタンを押してください。X サーバの [ディスプレイマネージャセレクション] ダイアログボックスと **chooser** のウィンドウの両方を表示させておくのはかえって混乱します。

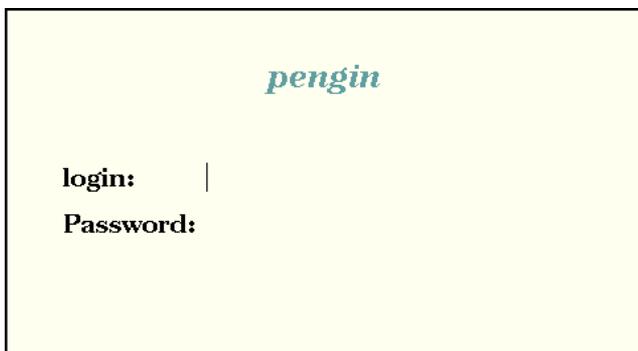


chooser が間接クエリーに使用されない場合には、X サーバの [ディスプレイマネージャセレクション] ダイアログボックスが表示されてユーザがディスプレイマネージャを選択できるように、[最初のマネージャを選択] オプションは無効にしてください。

chooser ウィンドウの表示形式は、慣例上 `/usr/lib/X11/xdm/Xresources` という名前になっているファイルにある資源によって制御できます。たとえば、VGA ディスプレイを使用している場合には小さめの初期ウィンドウとフォントを指定する必要があるでしょう。このファイルを変更できるのは、UNIX システムマネージャだけです。

ログインの方法

xdm でユーザのログイン準備が整うと、ユーザ名とパスワードを入力するダイアログボックスが表示されます。



注意 **x**dm ログインボックスはXクライアントであり、Windowsダイアログボックスではありません。このため、フィールド間を移動するには、Microsoft Windowsアプリケーション用のウィンドウのようにTABを使うのではなく、ENTERを使う必要があります。

ログインダイアログボックスの表示形式は、慣例上 `/usr/lib/X11/xdm/Xresources` という名前になっているファイルにある資源によって制御できます。このファイルを変更できるのは、UNIXシステムマネージャだけです。

セッションの起動

xdm はユーザのログインを認証すると、通常はシェルスクリプトであるスタートアッププログラムを実行して、リモートファイルシステムのマウントや日ごとのメッセージを表示するなどの処理を行います。デフォルトではスタートアッププログラムは実行されません。

次に、Xセッションを起動するプログラムが実行されます。デフォルトでは、これは `/usr/lib/X11/xdm/Xsession` というスクリプトです。通常、このスクリプトはユーザのホームディレクトリでセッション用に実行するプログラムを含む `.xsession` というファイルを検索します(「UNIXでのX Display Managerの構成」を参照)。`.xsession` ファイルがない場合には、`.xsession` スクリプトはシステムのデフォルトセッションを起動します。これには、たとえば、ウィンドウマネージャと単一の `xterm` が含まれます。

ウィンドウの管理

マルチウィンドウモードでは、Xクライアントは独自のMicrosoft ウィンドウ内に表示され、ローカルウィンドウマネージャ VWMによって管理されます。

シングルウィンドウモードでX Vision サーバを使用している場合には、クライアントはX Vision サーバウィンドウ内に表示されます。ユーザはウィンドウマネージャを実行して、ウィンドウの移動とサイズ変更ができるようにする必要があります。ローカルのMotif ウィンドウマネージャを使用するか、UNIX ホストシステム上のウィンドウマネージャを実行できます。いつもホストをベースにしたウィンドウマネージャを実行するつもりであれば、実行コマンドを `.xsession` ファイルに入れておきます。

X Display Manager セッションの停止

X Display Manager セッションは、セッションマネージャプログラムが閉じられると終了します(「X Display Manager の概要」を参照)。ディスプレイ上でまだ実行中のXクライアントがあれば終了されます。`.xsession` ファイルでは任意のプログラムをセッションマネージャとして指定できるので、どのプログラムにセッションマネージャの機能があるのかを覚えておいて、本当にセッションを終了させるときまで閉じないように注意してください。

[XDMCP 拡張設定] ダイアログボックスにある [1 度だけ] オプションが有効になっていると、Xサーバは終了します。このオプションが無効になっている場合には、サーバはXDMCPを使用して新しいセッションを起動しようとします。

その他の情報源

Valerie Quercia、Tim O'Reilly 著 『X Window System User's Guide for X11 Release 5』 1993年、O'Reilly & Associates, Inc. 発行

NCD XRemote の使用 (サポート対象外)

ここでは次の内容を説明します。

- XRemote の概要
- X サーバと XRemote
- PC での XRemote の構成
- XRemote の UNIX へのインストール
- UNIX での XRemote の構成
- XRemote セッションの起動
- XRemote セッションの終了

XRemote の概要

NCD XRemote では、シリアルライン接続またはモデム接続された PC で X クライアントを実行できます。

XRemote には、PC の部分と UNIX の部分が必要です。PC の部分は、シリアルライン X サーバという特殊な X サーバに含まれており、Vision コミュニケーションアーキテクチャの要素として提供されます。UNIX 部分は X Vision UNIX ソフトウェアの X サービスモジュールの一部として提供されるか、UNIX ホスト上の NCD XRemote の独立したインストールによって提供されます。

UNIX ソフトウェアには、X クライアントには UNIX システムで稼動している通常の X サーバに見える **Xremote** という“プロキシサーバ”があります。実際に Xremote は PC 上でソフトウェアと通信して、X サーバとのメッセージのやり取りを行います。

また、UNIX ソフトウェアには XRemote セッションの起動に使われる **xinitremote** というプログラムもあります (「XRemote セッションの起動」参照)。このプログラムは XRemote プロキシサーバを起動してからセッションファイルにリストされているクライアントを実行します。

セッション用に起動されたクライアントのうちの 1 つはセッションマネージャとして扱われ、このプロセスが終了するまでセッションは存続します。終了時に XRemote は他の X クライアントが残っていればそれを終了させ、接続をクローズします。

Xサーバと XRemote

XRemote へのサポートを含むシリアルライン Xサーバは、PC に Vision ソフトウェアをインストールするときにカスタムセットアップを選択し、[Select X Components] ダイアログボックスで[シリアルライン Xサーバ (XRemote)] オプションを選択するとインストールできます (標準セットアップではネットワーク Xサーバがインストールされます)。既存のインストールにシリアルライン Xサーバを追加する場合には、[Maintenance] セットアップを使用してください。

シリアルライン Xサーバをインストールすると、XRemote というプロファイルが作成されます。このファイルには、[トランスポート] タブの [シリアルラインで X Vision を使用] 設定が含まれています。必要ならば、シリアルライン Xサーバとネットワーク Xサーバを同時に実行させることが可能です。

XRemoteは PC とリモートホストとの間に直結された RS232C シリアルライン、またはモデムリンクを経由して使用するためのものです。Vision では、実際には Vision コミュニケーションアーキテクチャでサポートされる任意のトランスポートを経由して使用できるので、ネットワークを経由しても使用できます。XRemote の使用に関しては、次の項目を考慮してください。

- 接続は、通常 8 ビットすべてを通す接続にします。8 ビットパスを提供できない場合には、ホスト上に UNIX 版 Vision サービスをインストールするときに 7 ビット接続の使用を許可します。
- telnet サービスは適用されません。
- XRemote は **DISPLAY** 環境変数を設定します。`.cshrc` ファイルでは **DISPLAY** を明示的には設定しないでください。

XRemote でのモデムの選択やセットアップについては、PC 上への X Vision 製品の設定に関する項を参照してください。

Vision Resume がサイトで使用中であり、XRemote 上で Resume セッションを実行する場合には、「セッションの一時停止と再開」を参照してください。

PC での XRemote の構成

シリアルライン X サーバをインストールした後、XRemote プロファイルを構成して XRemote プロキシサーバが実行されるホストを選択できます。このホストは Vision コミュニケーションデータベースで事前に構成されている必要があります（「Vision コミュニケーションアーキテクチャ」の「ホスト設定」を参照）。ホスト名を入力しないと、XRemote セッションの起動時にプロンプトが表示されます。



通常、ユーザはシリアルラインやモデムを経由して XRemote を使用することでホストにアクセスするので、シリアルトランスポートとホストは、「Vision コミュニケーションアーキテクチャ」の「シリアルあるいはモデム接続を使用」に従って構成する必要があります。

ホストで [Automatic] トランスポート選択を使用すると構成すると、モバイルユーザは、オフィスでネットワーク経由でホストに接続する場合でも、社外でシリアルラインを使用する場合でも、一貫した X 環境で作業を行うことができます。どちらの場合でも、XRemote は X セッションを提供するために使われ、下位のトランスポートから提供される端末接続上で実行される XRemote プロトコルを使用します。[Modem] または [Serial-Com1] トランスポートのどちらを使用する場合でもプロパティダイアログボックスの [自動セレクションにこのトランスポートを含める] オプションを有効にする必要があります。

XRemote の UNIX へのインストール

XRemote ホストソフトウェアは、広範囲の UNIX システムをサポートしています。このソフトウェアが対象のプラットフォームで利用可能な場合には、UNIX 用 X サービスモジュールの一部としてインストールされます。XRemote が X サービスの一部としてインストールされるかどうかは、CD にある `/unix/unixhost.txt` ファイルを参照してください。XRemote をインストールしたくない場合、たとえば、NCDX 端末を使用し、XRemote がすでにインストールされている場合などには、X サービスのカスタムインストールを実行してください。

使用中のプラットフォームでは X サービスモジュールが利用できない場合でも、X Vision の CD から XRemote をインストールすることができます。CD 上の **Xremote** ディレクトリには、MS-DOS の圧縮されたファイル形式のファイルが、1 プラットフォーム 1 ファイルとして格納されています。***Xremote*Readme.txt** にはサポートするホストコンピュータとオペレーティングシステムの対応とファイル名が掲載されています。

X サービスとは別に XRemote をインストールするには

- 1 マイコンピュータまたはファイルマネージャを使用して、CD 上の **Xremote** フォルダから PC 上の空のフォルダにホスト用の XRemote ファイルをコピーします。
- 2 ファイル名をダブルクリックします。
- 3 Unix ネットワークコンピュータまたは他のファイルコピー方法を使用して、抽出したファイルをホストコンピュータのディレクトリに移動します。コピーはバイナリ形式で行ってください。
- 4 次に示す UNIX コマンドで tar ファイルからファイルを削除します。

```
tar -xvf filename
```
- 5 tar ファイル (**Xremote**、**xinitremote**) から抽出したファイルを `/usr/local/bin` などのようにエンドユーザのアクセスに適したディレクトリに置きます。操作にはスーパーユーザ特権が必要です。
- 6 **Xremote** ファイルと **xinitremote** ファイルのパーミッションを変更します。必要に応じて、グループまたはその他に実行パーミッションを与えます。たとえば、システム上の全員が XRemote を使用できるようにするには、次のように指定します。

```
chmod 0555 Xremote xinitremote
```

これで XRemote のインストールが終了しました。**xinitremote** プログラムや **Xremote** プログラムは起動する必要はありません。これらのプログラムは XRemote 接続がオープンされると、Vision コミュニケーションによって自動的に起動されます。

UNIX での XRemote の構成

UNIX 上の XRemote ユーザは、各自の環境で XRemote ファイルがアクセスできるかをチェックし、`.xinitmoterc` スタートアップファイルをセットアップしてください。

検索パス

`xinitremote` を含むディレクトリは、`PATH` 環境変数内に指定されている必要があります。UNIX システムマネージャは、通常、`PATH` 内に指定されるディレクトリにファイルを置いているはずですが、チェックするのであれば、UNIX システムプロンプトで “`env | more`” を入力し、`PATH` 変数を探してください。必要があれば、`PATH` の設定をするように UNIX システムマネージャに依頼してください。

`.xinitmoterc` の作成

XRemote セッションが開始すると、`xinitremote` はクライアントプログラムを起動するシェルスクリプトとして実行するために `.xinitmoterc` (ドットに注意) ファイルを検索します。そのようなファイルが存在しない場合には、`xinitremote` は次をデフォルトとして使用します。

```
xterm -geometry +1+1 -n login
```

UNIX システムマネージャは、XRemote ユーザ全員が使用できるようにグローバルセッションファイルを作成することもあります (次の例を参照)。このファイルは `.xinitmoterc` という名前にして、`xinitremote` と同じディレクトリに置く必要があります。このファイルの使用を要求されるユーザは、各自で `XINITRC` 環境変数をこのグローバルファイルにポイントする必要があります。デフォルトでは、`XINITRC` がユーザ環境で指定されていない場合には、ホームディレクトリにある `.xinitmoterc` が使用されます。

ホームディレクトリで `.xinitmoterc` ファイルを使用する場合には、ASCII テキストエディタを使用してください。このファイルには、シェルからクライアントを起動するとき使用するのと同じ構文のクライアント起動用コマンドを指定します。起動処理が速やかに行われるよう、クライアントを指定し過ぎないようにしてください。最後のクライアント以外はバックグラウンドで実行します。リスト内の最後のクライアントは、セッションマネージャとして使用するプログラムなので、これがフォアグラウンドで実行されます。このクライアントが終了すると、XRemote セッションが終了します。

マルチウィンドウモードで X サーバを使用している場合には、セッションファイルでウィンドウマネージャをセッションマネージャとしては起動しないでください。これを行うと、ウィンドウマネージャは実行できなくなり、セッションは終わってしまいます。ホストをベースにするウィンドウマネージャを実行する場合には、サーバをシングルウィンドウモードに設定してください。

セッションマネージャを起動し、セッションマネージャ自体が多数のクライアントを起動するのであれば、XRemote セッションが起動するまでに長い時間がかかります。

次に示すサンプルセッションファイルは、3つのクライアントを起動します。**DISPLAY**環境変数は **xinitremote** によって自動的に **net:1** に設定されます。ここで **net** はヘルパープロセスが実行されるホストのネットワークアドレスです。**DISPLAY** 変数は **.cshrc** ファイルには明示的には指定しないでください。

```
#!/bin/csh
# SCO X Vision
# Sample session file for use with X Display Manager
# or with NCD XRemote.
# Author : Steve Lewis Date : 1-Jul-1993
#
# Use this file to start clients at the start of an xdm
# or Xremote session.
# All clients except the last should be run in the background.
# The last client started is the "session manager" and must run
# in the foreground.
# Note that the DISPLAY variables is set by xdm or Xremote.
#
xclock -g =80x80-0+0 -display $DISPLAY &
xbiff -g 40x40-100+150 -display $DISPLAY &
# This xterm runs in the foreground and the session
# Terminates when you close it.
xterm -display $DISPLAY -ls
```

ヒント このファイルを使用するには、[ヘルプ] でこのページを表示して、[ヘルプ] ウィンドウから [コピー] コマンド ([編集] メニュー) を選択し、このテキストをメモ帳ファイルにコピーしてから、Unix ネットワークコンピュータを使用してUNIXシステムにこのファイルをコピーします。

Ethernet を使用して同じホストにローカルでいつものようにログインし、**xdm** ディスプレイマネージャ (「X Display Manager の概要」を参照) を使用している場合には、**.xsession** ファイルを XRemote のスタートアップファイルとしてコピーするか、**XINITRC** 変数で **.xsession** ファイルをポイントすることで、リモート環境をローカル環境と同一にすることができます。

注意 **.xinitmoterc** ファイルには、実行パーミッションを与える必要があります。次の例を参照してください。

```
chmod +x .xinitmoterc
```

UNIX 上の XRemote ログファイル

XREMOTETMPDIR 変数は、XRemote ヘルパーや **.xinitmoterc** スクリプトで実行されるすべてのクライアントからのすべての **stdio** ライブラリ出力用にログファイルのディレクトリとして設定できます。ログファイルは、**Xremote<n>** であり、<n> は表示番号を示します。

XREMOTETMPDIR 変数が設定されていない場合には、**TMPDIR** 変数からログファイルを作成するディレクトリ名が決定されます。どちらの変数も設定されていない場合には、ログファイルは **/usr/tmp** ディレクトリに作成されます。

バージョン番号

ホストに存在するヘルパープログラムとスタートアッププログラムのバージョン番号を表示するには、**what** コマンドを使用します。たとえば、SunOS では次のようになります。

```
%what Xremote xinitremote
Xremote:
    Copyright 1993 Network Computing Devices, Inc. All
rights reserved.
    NCDware version 4.0.115 Xremote
    main.c,v 15.20.8.2.6.1 1994/12/22 02:17:18 dct Exp $
    compress_lzw.c,v 14.13.12.1 1994/08/03 00:38:28 mse Exp $
    bytestream.c,v 15.22.8.1.6.1.2.1 1995/01/20 22:29:54 mse
Exp $
xinitremote:
    Copyright 1993 Network Computing Devices, Inc. All
rights reserved.
    NCDware version 4.0.115 xinitremote
    xinitremote.c,v 15.13.16.1 1995/02/10 18:57:35 jag Exp $
%
```

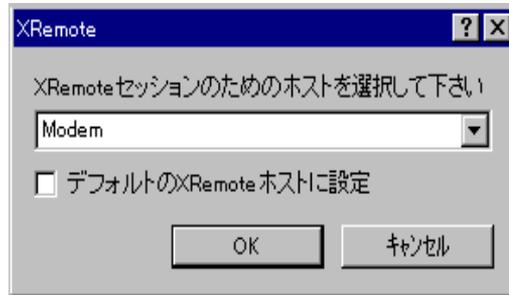
XRemote セッションの起動

XRemote セッションを起動するには

- 1 X サーバを実行し、XRemote プロファイルを指定します。
シリアルライン X サーバだけをインストールした場合には、ユーザが X サーバを起動すると選択されるデフォルトプロファイルは XRemote プロファイルになります。シリアルライン X サーバをネットワーク X サーバと同じシステムにインストールした場合には、[X Vision プロファイル] ダイアログボックスの [一般] タブにある [使用するプロファイルを尋ねる] オプションを有効にすることができます。X サーバを起動するときに、利用可能なプロファイルのリストから XRemote を選択してください。



- 2 X サーバプロファイルで [XRemote ホスト] を指定していなかった場合には、ホストを選択するプロンプトが表示されます。プロファイルで XRemote ホストを指定している場合には、このダイアログボックスは表示されません。



Xサーバプロファイルでこの XRemote ホスト名を設定する場合には、同じダイアログボックス内にあるチェックボックスをクリックします。

- 3 [Modem] トランスポートを使用して接続を行っている場合には、ユーザ名とパスワードの入力が常に要求されます。[Modem] トランスポートでは、ユーザ名とパスワードの格納は許可されていません。
- 4 [Modem] トランスポートを使用している場合には、電話番号を入力するプロンプトが表示されます。トランスポートまたはホストに対して指定された電話番号は、デフォルトとして表示されます。



[Vision コミュニケーションデータベース] で UNIX ホスト用に指定されたトランスポートを使用して、ホストへの接続がオープンされます。ログインの後、**xinitremote** という XRemote スタートアッププログラムが実行されます。これは **Xremote** ヘルパープロセスを起動し、実行する X クライアントを指定しているセッションファイルを実行しようとしています。まず最初に **XINITRC** 環境変数で参照されているファイルを検索します。これが見つからない場合には、ユーザのホームディレクトリで **.xinitremoterc** ファイルを検索し、これも見つからなければ、**.xinitrc** (XRemoteの旧バージョンと互換性あり) というファイルを検索します。どれも見つからない場合には、次のコマンドを使用して単一の **xterm** が起動されます。

```
xterm -geometry +1 +1 -n login
```

XRemote セッションの起動が失敗した場合には、シリアル接続のために[コネクションモニタ]ウィンドウを使用して接続の確立、ログイン、スタートアッププログラムの実行の過程を表示します。詳細については、ヘルプのキーワードの「トラブルシューティング」を参照してください。

XRemote セッションの終了

「.xinitremotercの作成」で説明したようにセッションファイルをセットアップしている場合には、セッションファイルの最後にリストされているクライアントを終了するか強制終了することによって、XRemote を終了させることができます。セッションファイルをセットアップしなかった場合には、XRemote で表示された最初の `xterm` を閉じることで、他のすべてのクライアントを強制終了してセッションを終らせることができます。

第 8 章

セッションの一時停止と再開

この章では次の内容を説明します（この章の内容はサポート対象外です）。

- Vision Resume の概要
- Vision Resume の動作のしくみ
- X サーバと Vision Resume
- PC での Vision Resume の構成
- UNIX での Vision Resume の構成
- Vision Resume セッションの起動または再開
- Vision Resume セッションの一時停止
- Vision Resume セッションの再配置
- Vision Resume セッションの終了
- シリアル接続またはモデム接続での Vision Resume の使用

Vision Resume の概要

Vision Resume は、X セッションを一時停止し、後で再開できるようにする機能です。この機能の利点には、主に次のものがあります。

- Resume セッションの一部である X クライアントは、実際には UNIX ホスト上の “ プロキシサーバ ” を経由して実行されます。
- プロキシサーバは、PC 上で作動している Vision X サーバと通信して、出力を表示し、ユーザからの入力を受け付けます。
- セッションは、ユーザからの動作でも、通信異常などのようにスケジューリングされていないイベントによっても、いつでも一時停止できます。セッションの一部である X クライアントは、プロキシサーバ上で実行し続けます。
- セッションが再開されると、前に実行されていた X クライアントは PC 上に再表示されるので、その X クライアントで作業を続けることができます。
- 1 人の UNIX ユーザにあるのは 1 つの Resume セッションだけなので、セッションは同一の PC 上でも別の PC 上でも再開可能です。
- X セッションはいつでも別のディスプレイに再配置できます。
- Vision Resume は、ネットワーク接続でもシリアルラインやモデムによる接続でも使用できます。シリアル接続やモデム接続の場合には、Vision Resume は XRemote トランスポート経由で実行されます。
- Vision Resume は独立した製品としてライセンスされますが、X サーバを含む X Vision 製品を購入したときには、UNIX サーバに Vision Resume の評価版をインストールできます。この評価版に対してインストール時または後でライセンス番号を入力すると、正式な製品にアップグレードできます。X Vision 製品を UNIX にインストールすることに関する詳細については、リファレンスマニュアルのインストールに関する章を参照してください。

Vision Resume の動作のしくみ

Vision Resume には、PC 部分と UNIX 部分が必要です。PC 部分は所有権を主張できる X 拡張機能として Vision X サーバ内で提供されます。UNIX 部分は、X Vision UNIX ソフトウェアの X サービスモジュール内で提供されます。

この UNIX ソフトウェアには、**tpproxy**と **tpwatch** という 2 つのプログラムから構成される “プロキシサーバ” が含まれています。プロキシサーバは、X クライアントにとっては UNIX システム上で作動する通常の X サーバに見えます。実際には、プロキシサーバは PC 上で Vision X サーバと通信して、出力の表示やユーザ入力の受け付けを行っています。

また、この UNIX ソフトウェアには、**xresume**というプログラムも含まれ、これは、Vision Resume セッションの起動や終了に使われます (「Vision Resume セッションの起動と再開」参照)。このプログラムは、実行中のプロキシサーバと Vision X サーバとの接続を再度オープンするか、プロキシサーバを起動してからセッションファイルにリストされているクライアントを実行するかのいずれかを行います。

Resume セッションは、ユーザが停止するまで存続します。

X サーバと Vision Resume

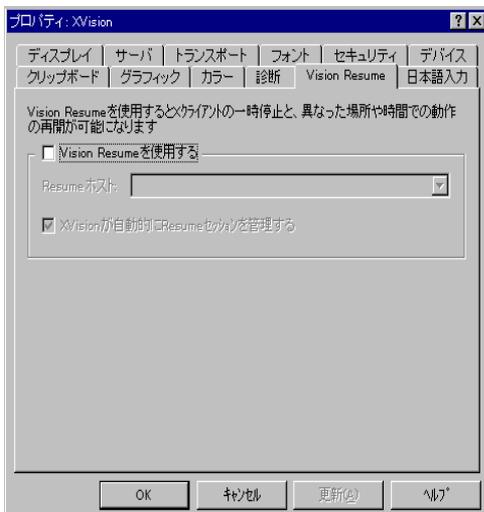
Vision X サーバには、Vision Resume へのサポートを提供する SCO©1996-VisionResumeExtensという弊社固有の X 拡張機能が含まれています。他の X サーバにはこの拡張機能は含まれていないので、他社の X サーバからは X セッションは制御できません。ただし、既存の X セッションを任意の X ディスプレイに再配置することはできます。

PC での Vision Resume の構成

Vision Resume は、X サーバプロファイル用のプロパティを変更することで構成されます。Windows のコントロールパネルを使用してプロパティを表示するか、または現在の X セッション用に X サーバの [コントロール] メニューを使用することで表示できます。デフォルトでは、Vision Resume を使用するオプションは無効になっています。

PC で Vision Resume を構成するには

- 1 [X サーバプロパティ] ダイアログボックスで [Vision Resume] タブをクリックします。



- 2 [Vision Resume を使用する] オプションを選択します。
- 3 [Resume ホスト] ボックスで、Vision Resume のプロキシサーバが作動するホストを選択します。このボックスに何も指定しないと、Resume セッションの起動時にホスト名を入力するプロンプトが表示されます。
- 4 X サーバの起動時に Resume セッションを自動的に起動させる場合には、[X Vision が自動的に Resume セッションを管理する] オプションを選択します。
- 5 [OK] または [更新] をクリックします。[X Vision が自動的に Resume セッションを管理する] が有効になっていると、Resume セッションが起動されます。

UNIX での Vision Resume の構成

Vision Resume ファイルを UNIX でアクセス可能にするために、環境変数を設定する必要があります。Vision Resume は新しいセッションの始めにクライアントを自動的に起動させるためにセッションファイルを使用します。また、システム管理者はグローバルセッションファイルのセットアップを行うか、ユーザがそれぞれのファイルを作成することを許可します。

環境の設定

`xresume` を含むディレクトリは、`PATH` 環境変数内に指定されている必要があります。インストレーション処理では、`/usr/bin` などの位置にあるこのプログラムにリンクを指定することによって、すでにこれを保証しています。別の方法として、`xresume` を含むディレクトリ、たとえば `/usr/local/vision/bin` が、ユーザの `PATH` に追加されます。Vision Resume の他の実行可能プログラムは `xresume` によって起動されます。`xresume` は `/etc/vision.conf` で構成された `VISION_BINDIR` に指定されたディレクトリで他の実行可能プログラムを検索します。

C シェルがログインシェルとして使用される場合には、環境変数は各ユーザ用に `$HOME/.cshrc` 内にセットアップされます。

Vision Resume は `DISPLAY` 環境変数を設定します。`DISPLAY` を明示的に `.cshrc` ファイルには設定しないでください。

Vision Resume セッションファイル

Vision Resume セッションが起動すると、`xresume` は `$HOME/.tpsession` (ドットに注意) というファイルを検索して、クライアントプログラムをスタートアップさせるためにシェルスクリプトとして実行します。このファイルがない場合には、`$VISION_ETCDIR/default.tpsession` というグローバルセッションファイルを検索します (`VISION_ETCDIR` の値は、`/etc/vision.conf` ファイルに、通常、`/usr/local/vision/etc` として定義されます)。セッションファイルが見つからない場合には、クライアントは自動的に起動されません。クライアントの起動は、リモートプログラムスターターを使用してユーザが行います (「Resume セッションでのクライアントの起動」参照)。

X Vision ソフトウェアと共にインストールされる `default.tpsession` ファイルには、`xterm` を起動する手順が含まれており、この `xterm` にはそこから起動されるすべてのクライアントもプロキシサーバ上でデフォルトで実行されるように、プロキシサーバ表示に `DISPLAY` 変数を設定するログインシェルが含まれています。システム管理者は、UNIX ホストのすべてのユーザに同一のデフォルト X セッションを与えて、他のクライアントを起動するために `default.tpsession` を編集できます。クライアントの起動に使用するコマンドは、シェルからクライアントを起動するときと同じ構文を使用します。起動処理が速やかに行われるよう、クライアントを指定し過ぎないようにしてください。すべてのクライアントはバックグラウンドで実行します。このため、各コマンド行の最後にはアンバーサンド記号 (&) を指定する必要があります。

ホームディレクトリに `.tpsession` ファイルを作成する場合は、`default.tpsession` をコピーして名称を変更し、ASCII テキストエディタでそのファイルを編集します。

注意 default.tpssession ファイルと .tpssession ファイルには実行パーミッションが許可されている必要があります。たとえば、ファイルに実行パーミッションを与える場合には、`chmod a+x $HOME/.tpssession` と入力します。

いくつかの UNIX のバージョンには、たとえば OpenServer 5 では、**PATH** 変数は X クライアントディレクトリを含むようにはセットアップされません。このような場合には、`/usr/bin/X11/xterm` などのセッションファイルにフルパス名を指定する必要があります。

一時プロキシセッションファイル

`$HOME/.tpinfo` ファイルは、現在のプロキシセッションについて、プロセス間で情報交換を行うために使われます。このファイルは通常のセッションの終了時に削除されますが、システムクラッシュなどの異常終了が発生した場合には、そのまま残されていることもあります。このような場合には、`.tpinfo` を削除してから新しい Vision Resume セッションを開始する必要があります。

`.tpinfo` ファイルには X クライアントがプロキシサーバとの通信に使用するための X ディスプレイ名とサーバ番号が含まれています。この値は、`.tpssession` ファイルによって起動される `xterm` とその子プロセスの `DISPLAY` 変数に設定されます。

Vision Resume ログファイル

プロキシサーバプログラムである `tpproxy` と `tpwatch` は、それぞれ `$HOME/.tpproxylog` ファイルと `$HOME/.tpwatchlog` ファイルというログファイルを生成します。`xresume` コマンドが実行している環境の中で `RESUME_DEBUG` 環境変数が 1 に設定されている場合には、適切な量の情報がこれらのログファイルに書き込まれます。設定されていない (0) 場合は、これらのログファイルにはわずかな情報が書き込まれます。この変数はシェルスタートアップファイル (`.profile`)、`.cshrc`、`vision.conf` のいずれかに設定できます。この変数を変更しても、すでに実行中のプロキシサーバは影響を受けないので、Vision Resume セッションを起動し直す必要があります。

Vision Resume セッションの起動または再開

[X Vision が自動的に Resume セッションを管理する] オプションが有効になっている場合には、Vision X サーバが起動すると、既存のセッションがあるときには再開され、セッションがないときには、新しいセッションが開始されます。

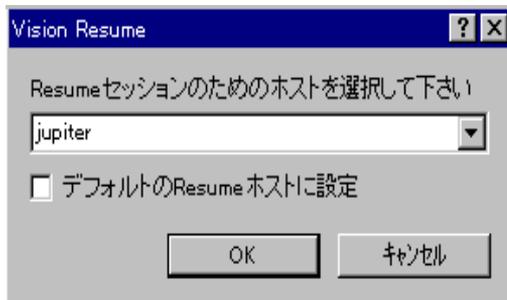
セッションの開始や再開は手動で行うこともできます。

Vision Resume セッションを手動で開始・再開するには

1 次のいずれか 1 つを行います。

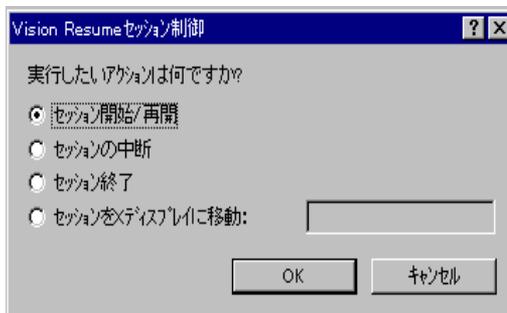
- サーバが隠された状態で実行されている場合には、タスクバーの通知領域にある [X サーバ] アイコンをマウスの右ボタンでクリックしてから、[Vision Resume] をクリックします (Windows NT 3.51 では、ルートメニューを表示してから [Vision Resume] をクリックします)。

- サーバが隠された状態ではない場合には、タスクバーの [X サーバ] アイコンをマウスの右ボタンでクリックしてから、[Vision Resume] をクリックします。
 - サーバがシングルウィンドウモードで実行されている場合には、[コントロール] メニューをクリックしてから [Vision Resume] をクリックします。
- 2 Xサーバのプロファイルで [Resume ホスト] が指定されていない場合には、ここでホストを選択するプロンプトが表示されます。プロファイルに Resume ホストが指定されている場合には、このダイアログボックスは表示されません。



Xサーバのプロファイルにここで入力した [Resume ホスト] を設定する場合には、チェックボックスをクリックします。

- 3 [Resume セッション制御] ダイアログボックスが表示されたら、[セッション開始/再開] をクリックします。



[Vision コミュニケーションデータベース]にあるそのホスト用に指定されたトランスポートを使用して、Resume ホストへの接続がオープンされます。**xresume** という Vision Resume スタートアッププログラムがそのトランスポート用に指定されたリモート実行サービスを使用して実行されます。このユーザ用の Resume セッションがすでに存在している場合には、プロキシサーバと PC 上の X サーバとの接続は再確立され、その後、任意のクライアントウィンドウが再描画されます。ホストベースのウィンドウマネージャがセッションの一部ではないと、X クライアント用の Window ジオメトリ情報の中には (特に、x 軸・y 軸の位置や幅と高さ)、クライアントが最初に起動された後に変更されるとなっているものもあります。

Resume セッションが存在していない場合には、**xresume** はプロキシサーバを起動し、X クライアントの実行を指定するセッションファイルを実行しようとします (「UNIX での Vision Resume の構成」参照)。ファイルが見つからないときには、クライアントは起動されません。

注意 Resume セッションは X クライアントが実行されないときには一時停止されていることがあります。この場合には、セッションを再開してもクライアントは表示されません。セッションファイル内のクライアントを表示する場合には、現在のセッションを終了してから新しいセッションを起動してください。

Resume セッションでのクライアントの起動

Resume セッションの起動時にセッションファイルが見つかると、このファイルに指定されたクライアントがプロキシサーバ上で起動され、ユーザのディスプレイに表示されます。これらのクライアントのいずれかが **xterm** の場合には、これを使用して他のクライアントを起動できます。この **xterm** 内の **DISPLAY** 環境変数は、プロキシサーバディスプレイに設定されるので、この **xterm** から起動されるすべてのクライアントもデフォルトで同じプロキシサーバ上で実行されます。

端末エミュレータからクライアントを手動で起動する場合には、**.tpinfo** ファイルに X クライアントがプロキシサーバとやり取りをするときに使用できる X ディスプレイ名とサーバ番号が含まれます。

Resume セッションが実行中のときに、リモートプログラムスターターを使用してセッションの一部になるクライアントを起動できます。Resume セッションが起動すると、プロキシサーバディスプレイの詳細は PC 上に格納され、リモートプログラムスターターはこの情報を使用してプロキシサーバ上でクライアントを起動します。この方法ですべてのクライアントを起動する方が良いのであれば、セッションのスタート時にクライアントが表示されないように空の **.tpsession** ファイルをセットアップします。

Vision Resume セッションの一時停止

Xセッションが一時停止されると、Xセッションの一部であるすべてのウィンドウは、PCの画面から削除されます。クライアントは、終了するまでプロキシサーバ上で実行を続行するか、プロキシサーバがクライアントからの要求を処理するのを待ちながら一時停止します。

Vision Resume セッションを一時停止するには

- 1 次のいずれか1つを行います。
 - サーバが隠された状態で実行されている場合には、タスクバーの通知領域にある [X サーバ] アイコンをマウスの右ボタンでクリックしてから、[Vision Resume] をクリックします (Windows NT 3.51では、ルートメニューを表示してから [Vision Resume] をクリックします)。
 - サーバが隠された状態ではない場合には、タスクバーの [X サーバ] アイコンをマウスの右ボタンでクリックしてから、[Vision Resume] をクリックします。
 - サーバがシングルウィンドウモードで実行されている場合には、[コントロール] メニューをクリックしてから [Vision Resume] をクリックします。
- 2 [サスペンドセッション] をクリックします。

この Resume セッションの X クライアントは PC 画面から削除されます。

Vision Resume セッションの再配置

Vision Resume セッションは、ユーザが現在使用している PC からアクセス権を持っている任意の X ディスプレイに再配置することができます。X クライアントによってディスプレイへのアクセスを制御する通常のセキュリティ機構はここでも適用されます。クライアントは任意のディスプレイに再配置できますが、Vision Resume セッションは Vision X サーバからだけ制御できます。

Vision Resume セッションを再配置するには

- 1 次のいずれか1つを行います。
 - サーバが隠された状態で実行されている場合には、タスクバーの通知領域にある [X サーバ] アイコンをマウスの右ボタンでクリックしてから、[Vision Resume] をクリックします (Windows NT 3.51では、ルートメニューを表示してから [Vision Resume] をクリックします)。
 - サーバが隠された状態ではない場合には、タスクバーの [X サーバ] アイコンをマウスの右ボタンでクリックしてから、[Vision Resume] をクリックします。
 - サーバがシングルウィンドウモードで実行されている場合には、[コントロール] メニューをクリックしてから [Vision Resume] をクリックします。

- 2 [リロケートセッション] をクリックします。
- 3 セッションを再配置するディスプレイの名前を入力します。たとえば、**chrisa:0** のように入力します。

この Resume セッションの一部であるすべての X クライアントは、ユーザの PC 画面から削除され、指定されたディスプレイ上に再表示されます。

元の PC にセッションを戻すには、元の PC で [Start/Resume セッション] オプションを選択します。

注意 セッションを再配置する先の PC には、Vision Resume プロキシサーバが実行されているホストからアクセスが許可されている必要があります。

Vision Resume セッションの終了

ユーザが Vision Resume セッションを停止すると、そのプロキシサーバ上で実行されているすべてのクライアントが終了されます。

Vision Resume セッションを終了するには

- 1 各アプリケーションで指示された終了手順を使用して、この Resume セッションの一部である任意の X クライアントを閉じます。
- 2 次のいずれか 1 つを行います。
 - サーバが隠された状態で実行されている場合には、タスクバーの通知領域にある [X サーバ] アイコンをマウスの右ボタンでクリックしてから、[Vision Resume] をクリックします (Windows NT 3.51 では、ルートメニューを表示してから [Vision Resume] をクリックします)。
 - サーバが隠された状態ではない場合には、タスクバーの [X サーバ] アイコンをマウスの右ボタンでクリックしてから、[Vision Resume] をクリックします。
 - サーバがシングルウィンドウモードで実行されている場合には、[コントロール] メニューをクリックしてから [Vision Resume] をクリックします。
- 3 [セッション終了] をクリックします。

この Resume セッションの一部でまだ実行中の X クライアントはすべて終了されます。

シリアル接続またはモデム接続での Vision Resume の使用

シリアルラインまたはモデムを経由して X クライアントを実行させるには、NCD の XRemote プロトコルを使用します (「X プログラムの起動」 の章にある 「NCD XRemote の使用」 を参照)。XRemote セッションファイルから Resume セッションを開始することで XRemote 接続上で Vision Resume を使用できます。単純化するために、XRemote セッションファイル `.xinitremoterc` にある任意のクライアント起動コマンドを Resume セッションファイル `.tpsession` または `default.tpsession` に移動します。次に `.xinitremoterc` ファイルに次の行を指定します。

```
REMOTE_DISPLAY=$DISPLAY
export REMOTE_DISPLAY
xresume -xremotestart
```

`xresume -xremotestart` を `.xinitremoterc` の最後の行に指定し、また、`xresume` をユーザの検索パス上に置く必要があります。

XRemote セッションを起動すると、既存の Resume セッションがあればリストアされ、なければ、新しい Resume セッションが起動されて Resume セッションファイル内のコマンドが実行されます。

Resume セッションの一時停止または停止を行うには、セッションから起動された `xterm` 内から `-xremotesuspend` オプションまたは `-xremotestop` オプションを使用します。たとえば、`xresume -xremotestop` は Vision Resume セッションを停止し、このセッションに関連する X クライアントがあればそれを終了させます。X サーバは自動的に終了されないので、ユーザが手動でクローズする必要があります。

(空白のページです)

第9章

端末エミュレーション

この章の内容は次のとおりです。

- 端末エミュレーションの概要
- 端末接続の開始
- VCA 端末サービス
- 端末エミュレータドキュメント
- ウィンドウのサイズおよび位置の変更
- 履歴ビューの表示
- フォント
- カラー
- ツールバー
- キーボード入力
- コピーアンドペースト
- イベント
- スタイル
- 記録および再生
- 端末エミュレータからの印刷
- 他のドキュメントへのリンク
- VT420 エミュレータ

端末エミュレーションの概要

X Vision 端末エミュレータを使用すると、キャラクタ端末アプリケーションを、リモート UNIX コンピュータ上で実行し、さらに PC の Microsoft Windows 環境で表示できます。使用できるエミュレータは VT420 です。

端末エミュレータを使用する場合は、ネットワークまたはシリアル回線を介してリモートホストに接続します。こうしてリモートホストに接続すると、このタイプの端末用に開発されたホストアプリケーションを実行できるようになります。包括的な端末エミュレーションを実現できる他に、次のような利点があります。

- 接続の開始、ホストのオペレーティングシステムへのログイン、およびコマンドの実行を自動的に行えます。広範囲のネットワークだけでなく、シリアル回線やモデムによる接続もサポートします。UNIX 対応の Vision サービスソフトウェアがホストにインストールされている場合は、単一のシリアル回線を介して複数のエミュレータセッションを実行できます。
- Windows アプリケーションとリモートアプリケーション間でデータを共有します。
- Windows 環境で新規エミュレータセッションドキュメントを作成して、現在のセッションからの設定を保存できます。
- Windows の属性設定ダイアログボックスを使用した構成が可能です。マウスの右ボタンで構成できる属性もあります。構成の詳細は、スタイルとして保存できるので、別のアプリケーションを実行するときに構成を簡単に変更できます。
- 画面の背景およびテキストのすべてのスタイルに対してカラーを指定できます。この指定は、リモートアプリケーションで指定されたカラーを上書きします。
- PC キーボード上のキーは、実際の端末のキーボードに配置されているキーにマッピングし、ほとんどの International PC キーボードに対応しています。さらに、ユーザは、単純なポイントおよびクリックインタフェースで、任意のキーまたはキー組み合わせをマッピングできます。
- ツールバーを使用すると、一般的な機能に簡単にアクセスできます。また、標準ツールバーを使用して、カスタムツールバーを作成できます。ツールバーは任意の場所に表示したり、任意の場所でウィンドウとドッキングできます。
- ウィンドウのサイズを変更するときに、拡大縮小がスムーズに行えるように TrueType フォントを使用できます。
- Windows プリンタに画面の内容を印刷できます。必要があれば、印刷前にプレビューを表示できます。

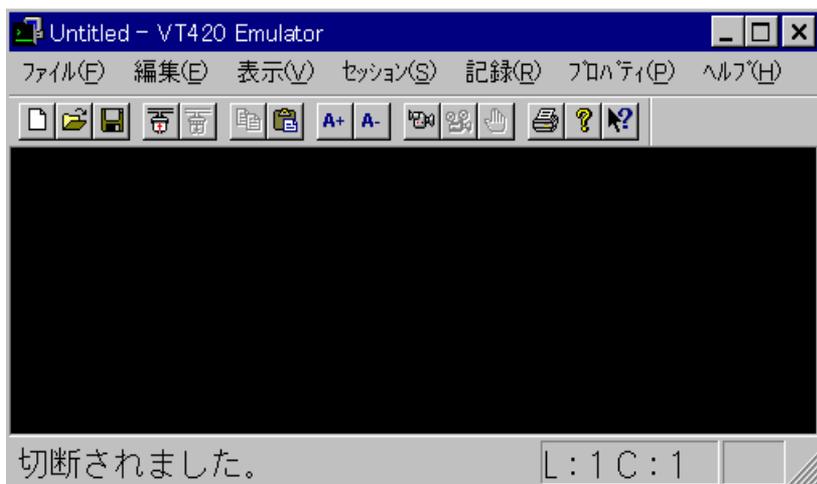
- 端末セッションを記録して、その場で再生したり、後で使用するために保管できます。
- データの着信または接続の切断のようなイベントに対応できます。
- ドラッグ&ドロップのような、Windowsスタイルのコピーアンドペーストが可能です。
- 通常のWindowsの手法を使用して、端末エミュレータウィンドウの移動、サイズ変更、およびアイコン化が可能です。
- 分割画面機能を使用して、これまでの出力の履歴を表示できます。この“履歴ビュー”は、通常の仮想画面とは別に構成できます。
- 他のWindowsアプリケーションにホットリンクし、別のドキュメントにコピーアンドペーストして、エミュレータウィンドウの変更に合わせて、データを更新できます。
- 端末エミュレータセッションを他のWindowsアプリケーションに埋め込むことができます。
- OLEオートメーションをサポートしています。このため、Windowsのプログラムは、隠しエミュレータを制御し、従来のキャラクタベースのアプリケーションに新しいグラフィカルインタフェースを提供できます。

端末接続の開始

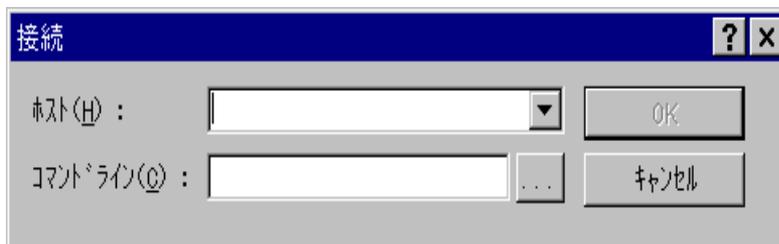
端末エミュレータを使用して、接続を開始できます。

端末エミュレータを使用して接続するには

- 1 以下のいずれかを行います。
 - [スタート] ボタンをクリックし、[プログラム] と [CentreNET X Vision] を順にポイントした後、[VT420 エミュレータ] をクリックします。
 - Windows NT 3.51 の場合は、プログラムマネージャで Vision グループを開き、[VT420 エミュレータ] のアイコンをダブルクリックします。



- 2 [セッション]メニュー上で、[接続]をクリックします。



- 3 [ホスト]ボックスにホスト名を入力するか、ドロップダウンリストから目的のホストを選択します。
- 4 [コマンドライン]ボックスに、アプリケーションを起動するために必要なコマンド名を入力します。

アプリケーションを起動するためのコマンドは、リモートホストでシステムプロンプトに対して入力するコマンドに似ています。たとえば、テキストエディタである **vi** を起動する場合には、**vi** と入力します。UNIX シェル (コマンドインタプリタ) を実行する場合は、**sh** (Bourne シェル)、**csh** (C シェル)、または **ksh** (Korn シェル) などのシェルの名前を入力します。なお、デフォルトのログインシェルでない場合は、そのシェルの非ログインインスタンスを取得すれば、起動ファイル (**.profile** または **.login**) 内のコマンドは実行されません。

また、参照ボタンをクリックして、Unix ネットワークコンピュータでホストおよびコマンドを参照することも可能です。ホストを指定しなければ、すべてのホストが表示されます。[ホスト] エントリですでにホストが指定されている場合は、そのホストのホームディレクトリが表示されます。実行するプログラムまたはシェルスクリプトが見つかるまで、参照を行います。

コマンドを入力しなければ、UNIX **login** プログラムが実行され、デフォルトのログインシェルを起動する前に、ユーザ名とパスワードの入力が求められます。

接続プロセス中はダイアログボックスにステータスメッセージが表示され、接続が完了すると、指定したプログラムがウィンドウに表示されます。

リモートホストとの接続がうまくいかない場合は、[コネクションモニタ] を使用して、接続プロセスをモニタしながら制御できます。[コネクションモニタ] を表示するには、SHIFT キーを押した状態で接続を開始します。詳細については、「Vision コミュニケーションアーキテクチャ」の「コネクションモニタの使用」を参照してください。

端末エミュレータの使用方法について詳細を知りたい場合は、[ヘルプ] メニューの [トピックの検索] をクリックします。また、メインの [Vision ヘルプの目次] から [ターミナルエミュレータの使い方] を選択しても、同じ情報が表示されません。次のようなヘルプ項目があります。

- ウィンドウの外観の変更
- リモートホストとの接続
- UNIX アプリケーションの外観の変更
- ツールバーの使い方
- VT420 端末の構成
- キーボードの構成
- 印刷
- エミュレータセッションの記録
- エミュレータドキュメントの使い方

VCA 端末サービス

このセクションの内容は次のとおりです。

- 概要
- 端末方式の選択
- USP タスクサービス

VCA 端末サービスの概要

VCA (Vision Communications Architecture) には、UNIX システムへのログインやコマンドを実行するための端末サービスが組み込まれています。

VCA 端末サービスには、次のような方式があります。

- UNIX 対応の Vision サービスがホストにインストールされている場合は、USP タスクサービスを使用する。
- UNIX オペレーティングシステムまたはネットワークソフトウェアの一部として提供されているリモートログインサービス (たとえば `rlogin` など) を使用する。

どちらの方式を使用しても似たようなサービスが提供されます。これらのサービスは端末 API を介してアプリケーションで使用できます。

端末接続方式の選択

Windows のコントロールパネルの中の [Vision コミュニケーション] オプションを使用して、トランスポートを構成できます。トランスポートを追加または修正するときは、[トランスポートプロパティ] ダイアログボックス内の [拡張機能] タブを使用して、端末接続サービスの方式を選択できます (「Vision コミュニケーションアーキテクチャ」の「トランスポートの構成」を参照してください)。



トランスポートで USP プロトコルを使用する場合は、USP タスクサービスで提供される USP 端末しか使用できません。また、トランスポートで TCP/IP プロトコルを使用する場合は、デフォルトは **rlogin** ですが、**telnet** も使用できます。

USP タスクサービス

このセクションの内容は次のとおりです。

- 概要
- 端末コマンドライン
- utmp ファイル

USP タスクサービスの概要

Vision USP タスクサービスを使用すると、リモートホストシステムにログインしてプログラムを実行できます。このサービスは、**rlogin** および **rexec** などの標準の UNIX サービスに取って替わるものです。USP タスクサービスは、トランスポートの端末接続サービスの構成時にオプションとして選択できます。端末接続に関するタスクサービスの設計における重要なポイントは次のとおりです。

- タスクサービスを使用すれば、UNIX のバージョンが異なると、それぞれのバージョン固有の **login** サービスを使用するログイン要求に対して応答の違いが生じるという問題を克服できます。これにより、PC から簡単かつ迅速に接続を開始できます。
- USP タスクサービスでは VCA 認証サービスを使用します。このため、暗号化されていないユーザ名およびパスワードはネットワークを介して送信されません。これに対して、**rlogin** ではユーザ名とパスワードを明確なテキストとして送信する必要があります。
- タスクサービスでは、他の UNIX ユーザから送られてくるメッセージを必要に応じて阻止して、端末エミュレータ画面ではなくメッセージパッドに表示します。
- タスクサーバは **tsrv** と呼ばれ、多くのクライアントからの複数のログインセッションを処理できます。これに対して、UNIX の標準的な **rlogin** プログラムは、独立したプロセスをフォークして各セッションを処理します。
- タスクサーバは、最大接続数までの複数の接続を処理できますが、接続数が最大数に到達すると、別の **tsrv** をフォークします。

- 端末タイプが呼び出し時に指定されている場合、タスクサーバはそのメッセージを端末接続の要求として扱い、プロセスをフォークしてシェルを実行します。このプロセスでは、クライアントのUIDが必要であり、ユーザに対するデフォルトシェルが実行されます。タスクサーバはシェルにとっては通常のttyと変わらない疑似ttyを使用して、シェルプロセスと通信します。実行するプログラムがクライアントで指定されている場合には、シェルはそのstdinおよびstdoutを継承するプロセスをフォークし、疑似ttyを使用します。そして、タスクサーバは、疑似ttyからPCにデータを返します。

端末コマンドライン

USP タスクサービスを使用して端末接続が行われる場合、コマンドラインは次のように解釈されます。

コマンドライン	動作
(なし)	UNIX の login プログラムを実行します。ユーザ名とパスワードの入力が求められ、これらの情報に基づいてログインが行われ、デフォルトのログインシェルが起動されます。
<i>program</i>	<i>program</i> を実行します。 <i>program</i> がデフォルトのログインシェル名であれば、ログインシェルとして実行されます。他のシェルの場合は、シェルの非ログインインスタンスとして実行されます。たとえば、ログインシェルとして実行されるCシェルは、 .cshrc および .login ファイルの両方を読み込みますが、非ログインシェルとして実行されるCシェルは、 .cshrc だけを読み込みます。
newterm program	プログラム名だけを指定した場合と同じように <i>program</i> を実行しますが、UNIX utmp ファイル内に端末のエントリを作成します。 <i>program</i> を指定しないで newterm キーワードを入力すると、コマンドを指定しなかった場合と同じ結果になります (utmp ファイルは login で処理されます)。

newtenm については、付録 B-19 を参照してください。

utmp ファイル

UNIX メッセージプログラムでは、**utmp** ファイルを使用して、ユーザがどの端末でログインしているかを検索します。タスクサーバは、通常、ユーザが最初にログインしたときに **utmp** 内に1つのエントリを作成して、そのユーザ名に送信されるメッセージを待機します。タスクサーバは、RPC クライアントとして機能し、これらのメッセージを該当するPC上のメッセージパッドサーバに送信します。この場合、タスクサーバは、端末エミュレータセッションに実際に関連付けられている疑似ttyデバイスのエントリを **utmp** 内に作成することはありません。最終接続が終了したユーザに対しては、**utmp** ファイルからそのユーザのエントリを削除します。

タスクサーバがどのように **utmp** ファイルを扱うかは変更できます。たとえば、特定の端末エミュレータセッション固有のエントリを **utmp** ファイル内に入れるには (この操作が必要なアプリケーションもあります)、コマンドライン上で **newterm** パラメータを使用します。また、特定のユーザ名で開始されたすべての端末接続の固有のエントリを **utmp** に入れるには、次のエントリを **tsrv.rc** ファイルに指定します。

```
#!UTMP-USERS
username
```

この場合、*username* は UNIX ユーザ名です。

メッセージ処理の詳細は、「メッセージ処理」を参照してください。

端末エミュレータドキュメント

このセクションの内容は次のとおりです。

- エミュレータドキュメントの概要
- Windows 2000/Me/98/95 または Windows NT 4.0 環境でのドキュメントの作成
- Windows NT 3.51 環境でのドキュメントの作成
- エミュレータドキュメントの構成
- サマリ情報

エミュレータドキュメントの概要

エミュレータドキュメントは、特定のタイプの端末エミュレータのための設定ファイル (*.v42) です。このファイルには、エミュレータセッションに指定できる構成可能な設定値がすべて入っています。Windows 2000/Me/98/95 または Windows NT 4.0 の場合、[スタート]メニュー、デスクトップ、または別のフォルダにドキュメントを作成できます。Windows NT 3.51 では、ドキュメントを任意のフォルダに保存できます。

エミュレータセッションで設定値を変更すると、そのセッションの終了時に変更内容を保存するかどうか求められます。

Windows 2000/Me/98/95 または Windows NT 4.0 環境でのドキュメントの作成

このセクションでは、[スタート]メニューまたはデスクトップへのエミュレータドキュメントの作成方法について説明します。また、別のフォルダにドキュメントを作成することも可能です。

[スタート]メニューでエミュレータドキュメントを作成するには

- 1 [スタート]をクリックしてから、[Unix プログラムの登録]をクリックします。
- 2 Unix アプリケーションウィザードの指示に従って、[スタート]メニュー上の [Unix プログラムの一覧] フォルダ内にアプリケーションのエントリを作成します。

デスクトップでエミュレータドキュメントを作成するには

- 1 マウスの右ボタンを使用して、デスクトップの空の部分をクリックします。
- 2 [新規作成] をクリックしてから、[Unix プログラム] をクリックします。
- 3 Unix アプリケーション ウィザードの指示に従って、新しい設定ドキュメントを示すアイコンをデスクトップ上に置きます。

このドキュメントアイコンをダブルクリックすると、ホストに対する端末接続が開始され、コマンドラインで指定したプログラムが、端末エミュレータウィンドウに表示されます。

Windows NT 3.51 環境でのドキュメントの作成

Windows NT 3.51 の場合、エミュレータを実行し、現在の設定値をファイルに保存することによって、エミュレータドキュメントを作成できます。

エミュレータドキュメントを作成するには

- 1 端末エミュレータで、まず [プロパティ] メニューをクリックして、次に [プログラム] をクリックします。
- 2 [タイトル] および [コマンドライン] に入力して、ドロップダウンリストから目的のホストを選択します。
who など、出力を表示して終了するプログラムの場合、[切断したら閉じる] オプションを無効にして、ウィンドウが閉じる前に出力を読み込みできるようにします。
- 3 必要があれば、他の属性を変更します。項目の [ヘルプ] を呼び出すには、SHIFT+F1 を押してヘルプポイントを表示させ、該当する項目をクリックします。ダイアログボックスを閉じるには、[OK] をクリックします。
- 4 [ファイル] メニューをクリックしてから、[名前を付けて保存] をクリックします。ドキュメントの名前を入力し、それを保存するフォルダを選択します。

エミュレータドキュメントの構成

ドキュメントを開かなくても、エミュレータドキュメントの属性を構成できます。

デスクトップまたはプログラムマネージャグループからドキュメントを構成するには

- Windows 2000/Me/98/95 または Windows NT 4.0 デスクトップ上にエミュレータドキュメントオブジェクトが存在する場合、マウスの右ボタンでそのオブジェクトをクリックしてから、ショートカットメニューの [プロパティ] をクリックします。
- プログラムマネージャグループにエミュレータドキュメントオブジェクトが存在する場合は、CTRL キーを押しながらドキュメントアイコンをダブルクリックします。

タブを使用して、エミュレータの各種属性を切り替えます。変更内容を保存して属性設定ダイアログボックスを閉じるには、OK をクリックします。また、属性設定ダイアログボックスを閉じる前に変更内容を有効にするには、[更新] をクリックします。

サマリ情報

エミュレータドキュメントのサマリ情報は [サマリ] タブに表示されます。このタブはエミュレータドキュメントの [プロパティ] ダイアログボックスに追加され、ダイアログボックスに情報を入力しないと、[サマリ] タブは表示されません。また、[サマリ] タブに [プログラム] タブの一部の情報も表示されますが、[サマリ] タブからはこの情報を変更できません。[サマリ] タブは、たとえば Windows 2000/Me/98/95 デスクトップまたはフォルダ内のドキュメントの属性を表示する場合など、ドキュメントを開く前にエミュレータドキュメントの属性を表示する場合に限り表示されます。

ウィンドウのサイズおよび位置の変更

端末エミュレータウィンドウは、通常 Windows で使用される方式、つまりウィンドウのリサイズボーダーおよびタイトルバーをドラッグすることによって、移動やサイズ変更を行うことができます。新しい位置とサイズは、[サイズと位置]、[表示] および [フォント] の属性設定ダイアログボックスのプレビュー画面、[サイズと位置] 属性設定ダイアログボックスの [高さと幅]、[上] および [左] ボックスに反映されます。

エミュレータドキュメントの場合、ドキュメントを開いたときのウィンドウの初期のサイズと位置は、[サイズと位置] 属性設定ダイアログボックスを使用して指定できます。ダイアログボックスの [プレビュー] セクションで縮小型のウィンドウのサイズを調整したり、ピクセル値を指定できます。また、エミュレータをデフォルトのサイズおよび位置で表示するように設定することもできます。通常、タイトルが付いていない新規のエミュレータは、デフォルトのサイズおよび位置に表示されます。この場合、入力した値はグレイ表示となり、代わりにデフォルト値が使用されます。

ヒストリビューの表示

端末エミュレータは、エミュレートする端末に合った行数と列数 (たとえば、24 行 80 列など) を保持する “仮想画面” を備えています。エミュレータウィンドウにはこの仮想画面のビューが表示され、さらにヒストリビューを表示するように画面を分割できます。このビューには、スクロールされて標準の端末ディスプレイに表示されない以前のテキストのヒストリが表示されます。

```

Untitled - VT420 Emulator
ファイル(F) 編集(E) 表示(V) セッション(S) 記録(R) フロアティ(P) ヘルプ(H)
[Icons] [A+] [A-] [Ctrl] [Alt] [F1] [F2] [F3] [F4] [F5] [F6] [F7] [F8] [F9] [F10] [F11] [F12] [Print] [Home] [End] [Esc] [Tab] [Enter] [Backspace] [Delete] [Insert] [Shift] [Caps] [Num] [Mouse]
root 28626 1 0 12月 13 ? 0:00 /usr/local/vision2/bin/logsrv -p5
root 29479 192 0 12月 16 ? 0:00 lpNet
root 2875 2869 0 19:16:18 pts/2 0:01 csh
root 28628 1 0 12月 13 ? 4:18 /usr/local/vision2/bin/nsmsrv -p5
root 28643 1 0 12月 13 ? 0:00 /usr/local/vision2/bin/tsrv -p5
-----
hirose 14338 14246 0 19:01:07 pts/4 0:00 man tar
root 14320 128 0 19:00:54 ? 0:00 in.rlogind
root 28937 1 0 12月 13 ? 0:45 /usr/local/vision2/bin/licsrv -p7
root 28621 1 0 12月 13 ? 1:01 /usr/local/vision2/bin/lna -b
root 12737 12736 0 14:10:12 ? 0:00 vfsd --profile /usr/local/visione
hirose 10979 1 0 12:00:26 ? 0:00 rsh -n syachi kterm -name syachi0
hirose 14246 14244 0 18:51:27 pts/4 0:01 -csh
hirose 2089 1 0 12月 16 ? 0:08 kterm
hirose 14340 14339 0 19:01:08 pts/4 0:00 sh -c more -s /usr/man/ja/cat1/tl
root 14347 14328 3 19:01:55 pts/8 0:00 ps -ef
hirose 14341 14340 1 19:01:08 pts/4 0:01 more -s /usr/man/ja/cat1/tar.1
hirose 12088 1 0 13:31:51 ? 0:00 rsh -n syachi kterm -name syachi0
cvision 14328 14320 0 19:00:55 pts/6 0:01 -csh
hirose 14050 14049 0 18:31:12 pts/3 0:01 csh
hirose 14049 1 0 18:31:10 ? 0:02 kterm
hirose 12483 1 0 13:54:40 ? 0:02 kterm
root 12735 1 0 14:10:11 ? 0:00 vfslockd
hirose 12751 12737 0 14:11:28 ? 0:03 vfsd --profile /usr/local/visione
iruka%
接続時間 0:00:01 L : 24 C : 8

```

画面出力の履歴を表示するには

- 1 マウスの右ボタンでエミュレータウィンドウをクリックしてから、分割された画面をクリックします。
- 2 分割線を希望の位置までドラッグします。

履歴ビューに表示可能な最大行数は、[表示] 属性設定ダイアログボックスで指定します。

フォント

端末エミュレータでは、3つのTrueTypeフォントを組み合わせで使用します。大部分の文字は一次フォントで表示されます。そのフォントのデフォルトはMSゴシックですが、[フォント] 属性設定ダイアログボックスを使用すれば、任意のTrueTypeフォントに変更できます。Microsoft Line Draw は、線やボックスを作成する際に使用します。Vision と呼ばれる専用フォントは、エミュレータにおいて追加の文字が必要な場合に使用します。Microsoft LineDraw および Vision は両方とも設定時にインストールされます。

[フォント] 属性設定ダイアログボックスでは、フォントサイズを指定できます。MSゴシックのデフォルトサイズは10x20です。[ウィンドサイズに合わせてフォントを変更] オプションを無効にすると、フォントはこのサイズで固定されます。また、このオプションを有効にすると、指定したフォントサイズがドキュメントが開いたときの初期サイズとして使用されます。ただし、フォントサイズはウィンドウのサイズを変更すると変わります。

カラー

端末エミュレータを使用すると、画面の背景色を指定したり、異なるテキスト項目を示すためにカラーやボールドまたは下線などの属性を設定できます。[カラー] 属性設定ダイアログボックスで指定したオプションによって、UNIX アプリケーションで指定された属性が上書きされます。たとえば、実際の端末で下線表示されるテキストを、エミュレータ画面では赤色表示にするように指定できます。

ウィンドウの [端末] および [ヒストリ] セクションの両方に設定値を指定できます。属性スキーマを保存したり、デフォルトのスキーマをはじめ既存の属性スキーマをカスタマイズできます。スキーマをドキュメント内で追加、削除、保存できます。また、スタイルの一部としてスキーマを保存することも可能です。

エミュレータには、デフォルトのスキーマおよび Black on White の 2 種類の定義済み属性スキーマが添付されています。デフォルトのスキーマは、黒色の背景に白色のテキストとなります。Black on White は、白色の背景に黒色のテキストで、このスキーマは、印刷機能を使用したり、[スタイル] 属性設定ダイアログボックスから [印刷] スタイルを選択すると、自動的に使用されます。

ツールバー

端末エミュレータには、以下のツールが組み込まれています。

ツールバー	目的
スタンダード	一般的なタスクへのアクセスを簡易化します。
ファンクションキー	ファンクションキー制御コードを送信します。
端末オプション	端末設定の変更を簡易化します。

上記のツールバーをカスタマイズして機能を追加したり、作業環境に合わせてツールバーを新しく作成できます。ツールバーボタンでは、コマンドの実行、再生セッションの実行、スタイルのアクティブ化、テキスト文字列の入力、およびファンクションキー制御コードの送信を行うことができます。また、印刷できない文字を取り込むこともできます(本章の「印刷できない文字」を参照してください)。ツールバーは、表示または非表示にしたり、任意の位置に表示させたり、任意の位置でウィンドウとドッキングすることができます。ツールバーをドキュメント内でツールバーを追加、削除、保存できます。また、スタイルの一部としてツールバーを保存することも可能です。

キーボード入力

端末エミュレータは、PC キーボード上のキーを、実際の端末に表示されるキーと関連付けます。エミュレータのタイプごとに、キーボードのタイプに応じたデフォルトマッピングが存在します。このデフォルトマッピングは、エミュレータドキュメントの一部として保存された新規マッピングによって上書きできます。デフォルトマッピングについては、付録内の「端末エミュレータのキーマップ」に記載されています。

PC キーボードでキーが押されると、エミュレータではそのキーに関連付けられているマッピングがあるかどうかを判定します。該当するマッピングが存在する場合は、そのマッピングが評価されます。該当するマッピングが存在しない場合は、Windows で提供されるデフォルトの変換機能が使用されます。ただし、英数字キーはマッピングされません。

デフォルトのキーボードマップ (*.dft) は、¥Program Files¥Vision¥System フォルダに保管され、次の表のように表示されます。

キーマップ エミュレータ

Vt420.dft VT420

各ファイルには、標準の IBM PC 拡張キーボードに対応するキーボードマップ (“ キーマップ ”) が入っています。また、ベンダ固有のキーボードに対応するキーマップが追加されている場合もあります。ご使用のキーボードのタイプに合ったマッピングは、新規エミュレータドキュメントで使用されます。

エミュレータドキュメントでは、[キーボード] 属性設定ダイアログボックスを使用して新規キーマップを作成できます。新しいキーマップには名前が付けられます。なお、初期状態ではデフォルトのキーマップと同じマッピングになっています。新しいキーマップの編集を選択すると、[キーマップエディタ] が表示されます。



このプログラムを使用すれば、英数字以外のキーに対するマッピングを変更できます。変更した内容を保存すると、このマッピングがエミュレータドキュメントの一部として保存されます。詳細については、[ヘルプ]メニューの[トピックの検索]をクリックします。なお、メインの [Vision ヘルプの目次] から [キーマップエディタの使い方] を選択しても同じ内容を表示できます。

キーマップエディタはスタンドアロンプログラムとしても実行できます(「エミュレータキーマップの編集」を参照してください)。この場合、スタンドアロンキーマップファイル(*.kmp)が生成されます。これらのファイルはX Vision 端末エミュレータでは使用できません。ただし、システム管理者の場合は、このようにキーマップエディタを使用して、デフォルトのキーマップを変更することがあります。そのような場合には、[スタート]メニューまたはプログラムマネージャからキーマップエディタを実行し、[オプション]メニューから該当する[エミュレータタイプ]および[キーボードタイプ]を選択します。必要な変更を行ったら、変更内容を保存して、このセクションの前の表に示したように、該当する*.dft ファイルを上書きします。

コピーアンドペースト

端末エミュレータウィンドウで表示される情報は、Windows のクリップボードにコピーしてから、エミュレータまたは他の Windows ドキュメントにペーストできます。

まず、クリップボードにコピーする情報を選択します。選択した情報は、コピーされるまでまたは他の選択が行われるまで維持されます。画面をリフレッシュする動作(ウィンドウへの書き込み、またはアクティブウィンドウの変更など)を行うと、選択はキャンセルされます。

テキストの行や列は、マウスをドラッグして選択します。カラムは、SHIFT キーを押した状態でドラッグして選択します。選択したテキストは、標準ツールバーの[コピー] ボタンをクリックすれば、クリップボードにコピーできます。そして、[貼り付け] ボタンを使用して、端末エミュレータの現在のカーソル位置にテキストを挿入したり、別の Windows アプリケーションに切り替えてからテキストをペーストすることができます。また、テキストを選択して、マウスで新しい場所までドラッグすることもできます。

注意 テキストをドラッグしたり、[貼り付け] コマンドを使用して、テキストを OLE 対応のドキュメントにコピーすると、テキストはエミュレータドキュメントとして埋め込まれます。

情報の選択およびコピー方法については、[ヘルプ]メニューの[トピックの検索]をクリックして、ヘルプのキーワードの「選択」または「コピー、エミュレータウィンドウからのデータ」を参照してください。

イベント

端末エミュレータを使用すると、エミュレータイベントをアクションにマッピングできます。イベントが発生すると、そのイベントに関連付けられているアクションがトリガされます。イベントには、ウィンドウのアクティブ化、接続の開始、およびデータ着信などがあります。また、アクションには、キーの送信、記録セッションの再生、現在のスタイルの変更、メニューコマンドの実行、テキスト文字列の送信などがあります。印刷できない文字を取り込むことができます(「印刷できない文字」を参照してください)。

イベントをドキュメント内でマッピング、マッピング解除、保存できます。また、スタイルの一部としてイベントを保存することも可能です。

イベントをアクションにマッピングする方法については、[ヘルプ]メニューの[トピックスの検索]をクリックして、ヘルプのキーワードの「イベント」を参照してください。

印刷できない文字

ツールバーまたはイベントを構成する際に印刷できない文字を入力するには、次の表のコードを使用します。

コード	意味
¥n	復帰改行
¥r	キャリッジリターン
¥t	タブ
¥b	バックスペース
¥ <i>nnn</i>	<i>nnn</i> は文字の 8 進数値

スタイル

端末エミュレータを使用すると、それぞれのエミュレータドキュメント内で、各種属性からなるスタイル(設定値)を作成できます。一意の構成を再呼び出し可能なスタイルとして保存できます。設定可能な属性としては、サイズと位置、表示、フォント、カラー、キーボード、端末、ツールバー、イベントなどです。

これにより、特定の UNIX アプリケーション用にエミュレータスタイルをカスタマイズできます。たとえば、Vi テキストエディタを実行する場合、固有のツールバー、Vi コマンドを実行するボタン、異なるキーマップ、端末設定、表示属性などを備えたスタイルを設定できます。さらに、電子メールを実行するために、ツールバーなどの固有の属性を備えたスタイルを設定することができます。この2つのアプリケーションのどちらを実行しても、カスタマイズしたスタイルを使用できます。

スタイルを新しく作成したら、そのスタイルを有効にするには、デフォルトとして設定しておくことが簡単な方法ですが、ツールバーに専用のボタンを作成することもできます。通常のスタイルをはじめとするどのスタイルでも、不要であれば、[スタイル]リストから削除できます。

エミュレータには、次の表に示すようなスタイルがあらかじめ定義されています。

スタイル	目的
通常	タイトルが付けられていないエミュレータを起動した場合のデフォルトのスタイルです。画面には黒色の背景にテキストが白色で表示されます。
印刷	このスタイルは、[印刷] および [印刷プレビュー] コマンドの実行時に自動的にアクティブになります。ディスプレイのカラーは、白色の背景に黒色のテキストに変換されます。印刷を開始したり、印刷プレビュー画面を閉じると、以前のスタイルに戻ります。
132 桁	現在のスタイルをベースにして、表示カラム数が 132 に設定されています。
80 桁	現在のスタイルをベースにして、表示カラム数が 80 に設定されています。
シンプル	現在のスタイルをベースにしていますが、ツールバー、メニューバーなどのウィンドウ装飾のすべてのフォームが削除されます。これによって、標準的な端末と同じようになります。別のスタイルに変更するには、マウスの右ボタンでエミュレータウィンドウをクリックした後、[フォント] または [カラー] のいずれかをクリックします。エミュレータの属性設定ダイアログボックスが表示されます。タブを使用して、[スタイル] 属性設定ダイアログボックスを表示させます。

スタイルの適用、作成、または削除の方法の詳細については、[ヘルプ]メニューの[トピックの検索]をクリックして、ヘルプのキーワードの「スタイル」を参照してください。

記録および再生

端末エミュレータを使用すると、コマンド、キー入力、およびマウスクリックを記録し、記録セッションとして保存することによって、タスクの自動化を実現できます。これらのタスクで実行するには、該当するセッションを再生します。記録セッションは、エミュレータドキュメントで追加、削除、保存できます。また、スタイルの一部として記録セッションを保存し、ツールバーボタンから再生したり、イベントやキーにマッピングできます。

記録セッションの記録、再生、削除の方法についての詳細は、[ヘルプ]メニューの[トピックの検索]をクリックして、ヘルプのキーワードの「スタイル」を参照してください。

端末エミュレータからの印刷

端末エミュレータは、データを印刷するためのメカニズムを2つ備えています。

- 印刷画面メカニズムでは、Microsoft Windows のプリンタドライバを使用して、ウィンドウの現在の内容を印刷します。
- スレーブ印刷メカニズムでは、アプリケーションから PC に接続されているプリンタにすべての出力を送信します。

印刷画面

実際の VT420 端末は“印刷画面”キーを備えており、これを使用して画面の現在の内容をプリンタに送信します。X Vision エミュレータの場合は、[ファイル]メニューから[プリント]コマンドを選択して、この機能を実現します。この場合、エミュレータウィンドウの現在の内容が印刷されます。ウィンドウに仮想端末画面全体が表示されない場合は、[プリント]を選択したときに表示されたデータだけが印刷されます。

必要に応じて、[キャンセル]ボタンを選択して、印刷を中止できます。

[プリント]コマンドでは、Microsoft Windows のプリンタドライバを使用します。したがって、テキストは自動的に PostScript などの正しいフォーマットに変換されます。スレーブ印刷の場合のように、フィルタを使用する必要はありません(詳細については、次のセクションを参照してください)。

[プリント]コマンドの出力は、現在選択されている Windows プリンタに送信されます。

ドキュメントを印刷する前に、正しく印刷されるかどうかを確認することをお勧めします。[ファイル]メニューから[印刷プレビュー]コマンドを使用して印刷されるページを確認し、ドキュメントに対して最終的な変更を行ってください。

スレーブ印刷

X Vision エミュレータでは、“スレーブ印刷”機能を提供しています。この機能を使用すると、出力を画面ではなくプリンタに(または画面とプリンタの両方に)に送信できます。この機能を PC でアクティブにするには、[端末]属性設定ダイアログボックスを開き、[印刷]をクリックし、[Transparent]または[Auto]モードを指定します。また、UNIX アプリケーションの場合、印刷モードを変更するために制御コードが送信されることがあり、このコードによって設定は上書きされます。

出力は、ファイル(デフォルト)、指定の COM か LPT ポート、Windows のデフォルトプリンタを示す“Pseudo Printen(疑似プリンタ)”のいずれかに送信できます。なお、Windows のプリンタドライバは使用されないため、プリンタで制御情報(たとえば、PostScript または Hewlett-Packard PCL など)をテキストに埋め込む必要がある場合は、プリンタがマルチユーザシステムに接続されているものとして出力を生成するように、マルチユーザアプリケーション(リモート印刷スプーラ)を設定する必要があります。疑似プリンタの場合は、端末エミュレータはポールド体および下線の制御コードを認識できます。

スレーブ印刷をオンにする方法の詳細は、[ヘルプ]メニューの[トピックの検索]をクリックして、ヘルプのキーワードの「スレーブ印刷」を参照してください。

他のドキュメントへのリンク

このセクションの内容は次のとおりです。

- 概要
- エミュレータオブジェクトの埋め込み
- 別のアプリケーションへのエミュレータドキュメントのテキストのコピー
- ホットリンクの作成
- エミュレータドキュメントでのプログラミング

リンクの概要

端末エミュレータでは、OLE (オブジェクトのリンクと埋め込み) をサポートすることによって、Microsoftのコンポーネント統合技術の大部分を実現しています。これにより、別の Windows アプリケーションで作成されたドキュメントにエミュレータオブジェクトを埋め込むことができます。たとえば、在庫レベルを表すレポートを Microsoft Word で作成し、在庫照会プログラムを開くエミュレータドキュメントを UNIX サーバ上に埋め込むことができます。さらに、OLE を使用すると、エミュレータウィンドウで表示されているテキストと別のドキュメント間でリンクを作成できます。たとえば、Excel スプレッドシートと、UNIX の在庫照会プログラムを表示するエミュレータウィンドウ間にリンクを作成できます。さらに、キャラクタベースの UNIX アプリケーションとのグラフィカルインタフェースを作成するなどのために、エミュレータドキュメントと交信するプログラムやマクロを作成できます。

エミュレータではまた、OLE クリップボードおよび OLE ドラッグ&ドロップの標準的なインプリメンテーションも提供します。これにより、すべての OLE アプリケーションに共通した基本機能がサポートされ、Windows アプリケーションからテキストをドラッグして、エミュレータウィンドウにドロップしたり、その逆にエミュレータアプリケーションからテキストをドラッグして、Windows ウィンドウにドロップできます。

エミュレータオブジェクトの埋め込み

端末エミュレータのデータファイルは標準的な OLE 構造になっているので、エミュレータオブジェクトを、別の Windows アプリケーションで作成されたドキュメントに埋め込むことができます。これには次の2つの方法があります。

- Windows ドキュメントに完全に属している新規エミュレータオブジェクトを埋め込む。
- 外部ファイルに保存されている既存のエミュレータドキュメントを埋め込む。Windows ドキュメント内で [ファイルへのリンク] を指定すると、ソースファイルとのリンクが作成され、エミュレータドキュメントでの変更を、埋め込まれたオブジェクトに反映できます。

エミュレータオブジェクトを Windows ドキュメントに埋め込むには

- 1 アプリケーションを開き、[挿入]メニューをクリックしてから、[オブジェクト]をクリックします。
- 2 新規エミュレータオブジェクトタイプをクリックするか、ファイルに保存されたドキュメントを検索します。

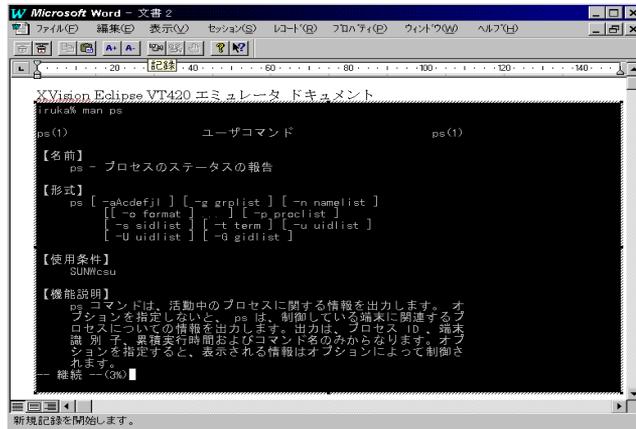
新規エミュレータドキュメントの埋め込みを選択すると、アプリケーションウィンドウ内でエミュレータが開から、アプリケーションのメニューバーがエミュレータ専用のものと置き換えられます。新規ドキュメントの構成を行う場合は、[プロパティ]メニューをクリックしてから [プログラム] をクリックします。所定のスペースに、接続先のホストおよびそのホストで実行するコマンドなどの詳細情報を入力します。ファイルに保存されたエミュレータドキュメントの埋め込みを選択すると、ドキュメントはアプリケーションのウィンドウ内にオブジェクトとして表示されますが、開くことはできません。

埋め込んだオブジェクトはその後 Windows 内で開くことができ、in-place または out-of-place モードのいずれかで実行されます。In-place モードで起動すると、OLE 複合ドキュメント技術が使用され、埋め込まれたアクティブオブジェクトを複合ドキュメント内で処理できます。Windows ドキュメント (コンテナ) およびエミュレータドキュメントは、それぞれのユーザインタフェースの要素をコンテナのウィンドウスペースにマージし、オブジェクトが編集ツールをコンテナに組み込みます。メニュー、ツールバー、および小さな子ウィンドウなど、すべてコンテナ内部に配置されます。このように、In-place モードで起動すると、ドキュメント中心の作業環境がユーザに提供され、該当するタスクで必要なすべてのものが 1 つのドキュメント内に用意されます。

in-place モードでエミュレータを開くには

- 1 エミュレータオブジェクトをクリックして、選択します。
- 2 [編集]メニューにおいて、[ドキュメントオブジェクト] をポイントしてから、[編集] をクリックします。

エミュレータが、アプリケーションウィンドウ内で開かれ、アプリケーションのメニューバーがエミュレータ専用のものと置き換えられます (オブジェクトが外部のエミュレータドキュメントとリンクされていない場合、またはオブジェクトがエミュレータウィンドウからコピーされた選択でない場合など、エミュレータが独立したウィンドウで実行していない場合に限りです)。すべてのメニューコマンドがエミュレータに設定されますが、[ファイル]メニューはコンテナに残ったままです。エミュレータは、コンテナウィンドウ内で in-place モードで実行しています。通常のエミュレータ処理を実行できます。たとえば、UNIX サーバに接続するには、[セッション]メニューをクリックしてから、[接続] をクリックします。



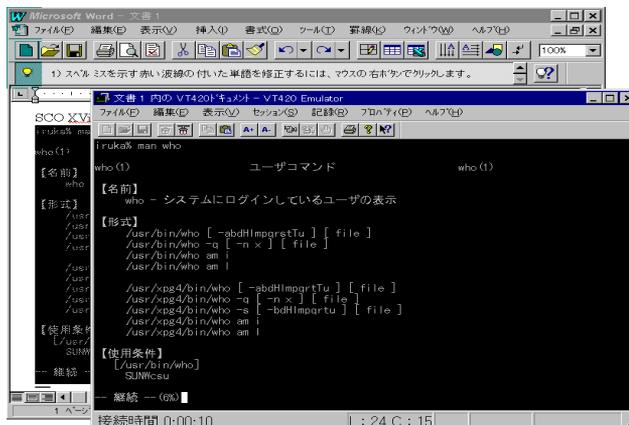
- 3 制御を Windows ドキュメントに戻すには、エミュレータオブジェクトフレームの外をクリックします。これでエミュレータが終了します。

エミュレータオブジェクトを out-of-place モードで起動することもできます。その場合、オブジェクトはコンテナフレームの外部で実行するので、独立したウィンドウでオブジェクトを操作できます。

エミュレータを out-of-place モードで開くには

- 1 エミュレータオブジェクトを、クリックして選択します。
- 2 [編集] メニューの [ドキュメントオブジェクト] をポイントし、[開く] をクリックします。

オブジェクトは、コンテナフレームの外の独立したウィンドウで動作します。次の図では、VT420 エミュレータが Word ドキュメントに埋め込まれ、コンテナの外で動作しています。



開いているエミュレータドキュメントで行った変更は、Windows コンテナのオブジェクト内で反映されます。これらの変更内容は、Windows ドキュメントに戻して保存することも、外部エミュレータファイルとして保存することも可能です。[ファイル]メニューをクリックしてから、[上書き保存]または[名前を付けて保存]のいずれかをクリックします。コンテナフレームの外で埋め込んだエミュレータを実行した場合には、[ファイル]メニューはエミュレータに設定されます。

Windows ドキュメント内のエミュレータオブジェクトは、エミュレータドキュメントが開いている間はロックされ、オブジェクトのインスタンスが複数開かれないようにしています。

オブジェクトの埋め込みについては、Windows アプリケーションのドキュメントを参照してください。

別のアプリケーションへのエミュレータドキュメントのテキストのコピー

エミュレータウィンドウからコピーしたテキストを、別の Windows アプリケーションにペーストできます。

エミュレータドキュメントから別のアプリケーションにテキストをコピーするには

- 1 エミュレータウィンドウ内のテキストを選択し、[編集]メニューをクリックしてから、[コピー]をクリックします。
- 2 情報のペースト先である Windows アプリケーションをアクティブにして、挿入ポイントを選択します。
- 3 アプリケーションの [形式を指定して貼り付け] コマンドをクリックして、[テキスト]を選択します。テキストをエミュレータドキュメントとして埋め込むこともできますが、通常は使用できません。また、テキストをピクチャとしてペーストすることもできます。

選択したテキストがドキュメント内に表示されます。

注意 [貼り付け] コマンドまたはドラッグ&ドロップ機能を使用して、エミュレータドキュメントから別の OLE 対応のドキュメントにコピーしたテキストは、埋め込まれます。

ホットリンクの作成

端末エミュレータではペーストリンク機能がサポートされ、エミュレータドキュメントと Windows アプリケーション間でホットリンクを作成できます。エミュレータドキュメント内で変更を行うと、Windows アプリケーションに反映されます。ペーストリンク操作では、エミュレータドキュメントをデータソース (つまり、サーバ) として扱い、Windows アプリケーションをオブジェクトのユーザ (つまり、クライアント) として扱います。

ホットリンクを設定するには

- 1 端末エミュレータで[プロパティ]メニューをクリックした後、[プログラム]をクリックします。
- 2 [拡張機能]をクリックします。



- 3 [リンク貼り付け]をクリックして、オプションを使用できるようにします。
- 4 [更新貼り付け]ボックスで、エミュレータドキュメントからリンクされているデータを更新する間隔を指定します。
- 5 エミュレータウィンドウでデータを選択し、[編集]メニューをクリックした後、[コピー]をクリックします。
- 6 情報のペースト先である Windows アプリケーションをアクティブにして、挿入ポイントを選択します。
- 7 アプリケーションの [貼り付け] コマンドをクリックしてから、[リンク貼り付け] をクリックします。データを、ドキュメントオブジェクト、ピクチャ、またはフォーマットされていないテキストのいずれとしてリンクするのかを選択し、[OK] をクリックします。通常は、[テキスト] を選択します。

選択したテキストがドキュメントに表示されます。クリップボードから直接テキストがペーストされるわけではありませんが、リンクがドキュメント内に挿入され、このリンクを使用してデータが検索されます。このリンクを使用すると、Windows アプリケーションから、直接エミュレータドキュメントを開いて構成できます。

リンクおよび [貼り付け] オプションの詳細については、Windows アプリケーションのドキュメントを参照してください。

注意 ペーストリンク操作を実行する場合は、エミュレータドキュメントをファイルに保存しておく必要があります。これは、タイトルが付けられていない状態のドキュメントではペーストリンク機能が動作しないためです。さらに、ペーストリンク操作中はドキュメントはロックされるので、エミュレータドキュメントのインスタンスは1つだけ開いてください。

エミュレータドキュメントでのプログラミング

OLEオートメーションを使用すると、端末エミュレータ内で実行するUNIXキャラクタベースのアプリケーションとのWindowsインタフェースを新しく作成できます。また、エミュレーションウィンドウを非表示にして、Windowsプログラムだけを認識できるようにすることも可能です。さらに、Microsoft Visual BasicまたはVisual C++で作成されたプログラムから、またはVisual Basic for Applicationsを使用するMicrosoft Excelのようなアプリケーションから、エミュレータを制御できます。

たとえば、VT420で表示するように作成されたUNIXの在庫管理プログラムは、新しいWindowsプログラムで制御して、従来のアプリケーションとのグラフィカルユーザインタフェースのすべての機能を実現できます。

詳細については、「OLEオートメーション」を参照してください。

VT420 エミュレータ

このセクションの内容は次のとおりです。

- 仕様
- サポートされていない機能
- 端末の設定
- 文字セット

VT420 の仕様

VT420 エミュレータは、Digital VT420 ビデオ端末をベースにしています。このエミュレータは、キーボードおよび通信機能の柔軟性が非常に優れたシングルセッションのテキストビデオ端末であり、ANSI および ASCII の 2 つのディスプレイおよびテキスト処理機能を備えています。

このエミュレーションを使用することによって、DEC ビデオ端末用に設計されているアプリケーションで PC の機能を利用できます。

詳細については、Digital Equipment Corporation から提供される『*VT420 Video Terminal Programmers Information*』マニュアルを参照してください。

サポートされていない機能

VT420 では、次の機能はサポートされていません。

モニタモード

リモート周辺機器インタフェース (バーコードリーダなど)

マクロ

CGI グラフィックス

ロード可能なカラーパレット

優先マルチインターナショナルキャラクタセットのレポート

VT420 端末の設定

[端末] 属性設定ダイアログボックスを使用すると、VT420 端末のディスプレイおよび機能を管理する ANSI 制御関数を設定できます。

端末識別情報 (端末がアプリケーションでどのように認識されるか) および仮想ページの大きさを設定できます。つまり、ディスプレイ内のデータ行およびデータ列の数、およびページメモリ内のページごとの行数などです。VT420 は、オフスクリーンメモリの 3 つのページを次の構成でサポートします。

3 ページ x 24 行 (デフォルト)

2 ページ x 25 行

2 ページ x 36 行

1 ページ x 41 行

1 ページ x 42 行

1 ページ x 48 行

1 ページ x 52 行

1 ページ x 53 行

1 ページ x 72 行

また、すべての仮想ページに対してタブストップを設定したり、端末印刷オプションおよびキャラクタセットを指定できます。

[高度な設定] ダイアログボックスを使用すると、端末管理制御機能やディスプレイの拡張機能を追加できます。

オプションについては、Windows 2000/Me/98/95 または Windows NT の場合は、ダイアログボックスのタイトルバーの **?** をクリックし、目的の項目をクリックしてください。Windows NT 3.51 の場合は、SHIFT+F1 を押しながら、ヘルプポインタを表示させ、項目をクリックします。

VT420 キャラクタセット

VT420 エミュレータでは、出力を表示するために、VT 8 ビットマルチナショナルセットおよび 7 ビットナショナル置換 (National Replacement) キャラクタセットの両方を提供しています。

VT420 は、VT 8 ビットマルチナショナルキャラクタセットの全範囲をサポートし、ISO Latin-1 および DEC Multinational を User Preferred Supplemental セットとしてサポートします。これらは、特殊な予備セットとして頻繁に使用されるセットです。DEC および ISO キャラクタセットは、ISO 2022 の構造と規則に準拠しています。

VT8ビットマルチナショナルキャラクタセットは、7ビットキャラクタ“ハーフ”2つから構成されています。ASCIIは左半分に使用され、これに対応して指定された補助的なセットが右半分として使用されています。これらは別々に呼び出されますが、実際には常に一緒に使用されます。8ビットキャラクタセットには、127を超えるキャラクタが含まれています。

ナショナル置換 (National Replacement) キャラクタセットを選択すると、選択されているキーボード言語によって、対応する7ビットセットが使用されます。7ビットの文字セットには、最大127のキャラクタが含まれています。

オプションについては、Windows 2000/Me/98/95 または Windows NT の場合は、ダイアログ

ボックスのタイトルバーの **?** をクリックし、目的の項目をクリックしてください。Windows NT 3.51の場合は、SHIFT + F1を押しながら、ヘルプポインタを表示させ、項目をクリックします。

(空白のページです)

第 10 章

OLE オートメーション

このセクションの内容は次のとおりです。

- OLE オートメーションの概要
- Vision オブジェクト
- タイプライブラリ
- 関数

OLE オートメーションの概要

OLE オートメーションは、2つのアプリケーションの間で情報を共有する主な手段です。これによって、プログラマは一連のオペレーションを定義し、他のアプリケーションにアクセスできます。最終的な結果として、スクリプトまたはプログラムが一度に複数のアプリケーションに影響を与えることができるようになります。

OLE オートメーションは、オートメーションオブジェクトとオートメーションコントローラという2つのコンポーネントで構成されています。オートメーションオブジェクトはプログラム可能なコンポーネントであり、オブジェクトが提供できるすべての情報を提供します。オートメーションコントローラは開発者がオブジェクトを使用するスクリプトやアプリケーションを作成できるプログラミング環境です。

X Vision 端末エミュレータはオートメーションオブジェクトとして機能します。オートメーションオブジェクトは、Microsoft Visual Basic や Visual C++ などのプログラミング言語、あるいは Visual Basic for Applications をサポートする Microsoft Excel や Word などのアプリケーションで扱えます。Vision オブジェクトタイプと追加機能を使用することで、Windows アプリケーションはエミュレータ内で実行されている UNIX アプリケーションから情報を取り出すことができ、エミュレータがユーザから見えない状態で自身のウィンドウで表示できます。

OLE オートメーションの使用の詳細については、プログラミング言語のマニュアルを参照してください。

Vision オブジェクト

X Vision は、他のアプリケーションからアクセスできるエミュレータオブジェクトタイプを自身のプログラム ID を使用して提供します。プログラム ID は OLE が OLE オートメーションオブジェクトをシステムにインストールされている他のアプリケーションと区別するために使用する識別子です。次の表では、プログラム ID を持つ Vision オブジェクトのタイプと、それらを Visual Basic の `CreateObject` 呼び出しを使用して起動する方法の例を示しています。

オブジェクト	プログラム ID	例
VT420 エミュレータ	VT420.document	Set VT420 = CreateObject (ÅgVT420.documentÅh)

`CreateObject` をいずれかのエミュレータタイプと共に使用すると、デスクトップやプログラムマネージャグループで新しいエミュレータドキュメントを作成するのと同じ方法で、デフォルト構成で初期化されたエミュレータインスタンスを起動します。その後、オブジェクトタイプに対応したプロパティを設定することでこの構成を変更できます。次の表では、これらのプロパティとその使用例を示します。

プロパティ	説明	例
Command	ホストコンピュータ上で実行されるコマンドの設定または取り出しを行います。	<pre>VT420.Command = `csh` MsgBox `Command = ` + VT420.Command</pre>
ConnectOnOpen	エミュレータドキュメントが開かれたときにホストへの接続がオープンされるように指定します。	<pre>VT420.ConnectOnOpen = 0 VT420.ConnectOnOpen = 1</pre>
CurColumn	現在のカーソル位置のカラム番号を取り出します。	<pre>Label6.Caption=VT420.CurColumn</pre>
CurLine	現在のカーソル位置の行番号を取り出します。	<pre>Labe15.Caption=VT420.CurLine</pre>
Host	ホスト名の設定または取り出しを行います。	<pre>VT420.Host = `jupiter` MsgBox `Host = ` + VT420.Host</pre>
NumLines	仮想画面上の行数の設定または取り出しを行います。	<pre>VT420.NumLines = 50 MsgBox `NumLines = ` + VT420.NumLines</pre>
NumColumns	エミュレータ画面上のカラム数の設定または取り出しを行います	<pre>VT420.NumColumns = 132 MsgBox `NumCols = ` + VT420.NumColumns</pre>
Prompt	プロンプトをオンまたはオフにします。これによって、接続がオープンされるときにユーザに確認用のプロンプトが表示されるかどうかが決まります。	<pre>VT420.Prompt = 0 VT420.Prompt = 1</pre>
TermType	端末の ID の設定または取り出しを行います。これによって、ホストソフトウェアに端末自身をどのように認識させるのが決まります。	<pre>VT420.TermType = `vt220` MsgBox `TermType = ` + VT420.TermType</pre>
Title	ウィンドウのタイトルバーに表示される記述の設定または取り出しを行います。	<pre>VT420.Title = `Unixshell` MsgBox `Title = ` + VT420.Title</pre>

注意 **Command**、**Host**、**Prompt**、**TermType** の設定は、UNIX サーバに接続する前に行う必要があります。その他のプロパティは、接続後に変更できます。

タイプライブラリ

次のタイプライブラリが Microsoft Visual C++ で使用するために提供されています。このライブラリは、X Vision 端末エミュレータを CD からインストールしたときの Devkit ディレクトリにあります。

V420term.tlb

関数の概要

次の関数は、プログラミング言語からエミュレータオブジェクトを操作するために提供されています。

Connect	UNIX サーバに接続する。
Disconnect	UNIX サーバへの接続を解除する。
Load	指定した構成ファイルをロードする。
Save	指定した構成ファイルを保存する。
Scrape	UNIX アプリケーションからテキストを探り出す。
SendKey	UNIX アプリケーションにキー値を送信する。
SendString	UNIX アプリケーションに文字列を送信する。
ShowWindow	エミュレータウィンドウを表示または隠す。

Connect

説明

UNIX サーバに接続し、コマンドを実行します。

構文

Connect (*strHost*, *strCommand*)

解説

この関数は、UNIX サーバへの接続をオープンし、コマンドを実行します。使用するパラメータは次のとおりです。

strHost ホスト名。Vision コミュニケーションがユーザ名とパスワードを持たないホストに指定すると、[確認] ダイアログボックスが表示されます。

strCommand 指定したホスト上で実行されるリモートコマンド。

例

次は、Pluto というホストへの接続をオープンし、VT420 エミュレータでC シェルを実行するコードです。

```
SubConnect_Click()  
    i = VT420.Connect (~pluto~, ~csh~)  
End Sub
```

Disconnect

説明

UNIX サーバへの接続を解除します。

構文

Disconnect

解説

UNIX サーバに接続し、追加関数を実行した後、**Disconnect** 関数は接続をクローズします。

例

次は、現在の VT420 エミュレータ接続をクローズするコードです。

```
SubDisconnect_Click()  
    VT420.Disconnect  
End Sub
```

Load

説明

指定した構成ファイルをロードします。

構文

`Load (strString)`

解説

指定した構成ファイルをロードします。使用するパラメータは次のとおりです。

`strString` 構成ファイルのパス

例

次は、構成ファイル `C:\Geoffs\Pluto.v42` をロードするコードです。

```
Sub Load_Click ()  
    VT420.Load ("c:\geoffs\pluto.v42")  
End Sub
```

Save

説明

指定したファイルに対する構成設定を保存します。

構文

Save (*strString*)

解説

指定したファイルに構成設定を保存します。使用するパラメータは次のとおりです。

strString 設定を保存するファイルのパス

例

次は、ファイル **C:\Geoffs\Pluto.v42** に構成設定を保存するコードです。

```
Sub Save_Click ()  
    VT420.Save ("c:\geoffs\pluto.v42")  
End Sub
```

Scrape

説明

エミュレータ画面からテキストを取り出します。

構文

`Scrape (intLine , intColumn , intLength)`

解説

この関数は、ユーザが選択した位置に置かれるエミュレータ画面からテキストを取り出します。使用するパラメータは次のとおりです。

intLine テキストを取り出す行の開始位置。これは 1 から始める必要があります。

intColumn テキストを取り出すカラムの開始位置。これは 1 から始める必要があります。

intLength 開始位置から取り出す文字数。

例

次は、エミュレータ画面の左上端から 80 文字分のテキストを取り出すコードです。

```
Sub Command2_Click ()  
    Label14.Caption = VT420.Scrape(1, 1, 80)  
End Sub
```

SendKey

説明

UNIX アプリケーションにファンクションキーの値を送信します。

構文

`SendKey (intKey)`

解説

この関数は、リモートアプリケーションにエスケープシーケンスを送信するときに便利です。使用するパラメータは次のとおりです。

`intKey` キーにマップする値

これらの値は、ファイル拡張子 `.bas` や `.h` がつくエミュレータ固有ファイルにリストされます。たとえば、`vt420def.bas` と `vt420def.h` には、VT420エミュレータのキーに対する値が含まれています。エントリは次のようなタイプになります。

```
Global Const VT420_F1 = 500
```

ファイルは、X Vision 端末エミュレータをインストールした CD 上の **Devkit** ディレクトリに置かれます。

例

次は、UNIX アプリケーションに VT420 F1 キーを送信するコードです。

```
Sub Command3_Click ()  
    VT420.SendKey (VT420_F1)  
End Sub
```

SendString

説明

UNIX アプリケーションに文字列を送信します。

構文

SendString (*strString*)

解説

リモートアプリケーションに文字列を送信します。使用するパラメータは次のとおりです。

strString 送信する文字列

印字不能な文字を入力する場合は、次の表のコードを使用してください。

コード	意味
¥n	改行
¥r	キャリッジリターン
¥t	タブ
¥b	バックスペース
¥ <i>nnn</i>	<i>nnn</i> は文字の 8 進値

例

次は、コマンド **ls -a** を UNIX アプリケーションに送信するコードです。

```
Sub Command1_Click ()
    VT420.SendString (~ls -a ¥r~)
End Sub
```

ShowWindow

説明

エミュレータウィンドウを表示したり、隠したりします。

構文

`ShowWindow(intShow)`

解説

この関数は、エミュレータをバックグラウンドに隠すか、ウィンドウを表示させてフォアグラウンドにするかを示します。使用するパラメータは次のとおりです。

intShow 表示する場合は 1、隠す場合は 0

例

次は、VT420 エミュレータウィンドウを隠すコードです。

```
SubHide_Click()  
    VT420.ShowWindow (0)  
End Sub
```

第 11 章

エミュレータキーマップの編集

このセクションの内容は次のとおりです。

- エミュレータキーマップ編集の概要
- キーマップの作成
- エミュレータでのキーマップの変更

エミュレータキーマップ編集の概要

キーマップエディタでは、X Vision 端末エミュレータで使用されるキーボード構成ファイル (“ Keymaps ”) の作成と変更ができます。

キーマップエディタは、端末エミュレータ内の [キーボードプロパティ] ダイアログボックスから、またはスタンドアロンのアプリケーションとして実行できます。端末エミュレータから実行した場合は、キーマップはキーマップエディタ内の一時ファイルになります。キーマップエディタには次の機能があります。

- ポイントとクリックによってキーを定義できる標準 PC キーボードの表示。キーボードレイアウトを選択することもできます。
- 文字列、組み込みコマンド、エミュレータ記録、スタイルなどの広範囲の動作やイベントをキーにマップする機能。
- キーマップファイル間でキー定義のコピーやペーストを行う機能。
- キー定義すべての表示とデフォルトマッピングへのリセットを行う機能。

キーマップの作成

キーマップを作成するために、キーマップエディタをスタンドアロンアプリケーションとして使用できます。

キーマップを作成するには

- 1 [スタート] ボタンをクリックし、[プログラム]、[CentreNET X Vision]、[アクセサリ] を順にポイントしてから [キーマップエディタ] をクリックします。

キーマップエディタが画面に表示されます。



- 2 必要なキー定義を設定します。たとえば、F9 を文字列 iEndf にマップする場合は、キーボードの F9 をクリックします。[マッピング] メニューで [文字列] をクリックした後、[マッピング] ボックスで “End” と入力します。ツールバー上の緑色マークをクリックして定義を確認します。



- 3 定義し終わったら、[ファイル] メニュー上の [上書き保存] をクリックして変更を保存します。キーマップファイルは拡張子 *.kmp つきで保存されます。

現時点では、この方法で作成したキーマップは X Vision エミュレータでは使用できません。ただし、システム管理者はこの方法を使用して、ファイル拡張子 *.dft つきのデフォルトのエミュレータキーマップを変更できます(「端末エミュレーション」章の「キーボード入力」を参照)。

注意 文字列をマップするときに印字不能な文字を入力する場合は、次の表のコードを使用してください。

コード	意味
-----	----

¥n	改行
¥r	キャリッジリターン
¥t	タブ
¥b	バックスペース
¥nnn	nnn は文字の 8 進値

キーマップエディタを使用する全命令については、[ヘルプ] メニューの [トピックの検索] をクリックしてください。同じ内容の情報がメインの [Vision ヘルプの目次] の [キーマップエディタの使い方] でも表示できます。ここには次の内容のヘルプが用意されています。

- ウィンドウの表示形態の変更
- キーマップの使用

エミュレータでのキーマップの変更

特定のエミュレータドキュメント内で使用するキーマップは変更できます。ユーザが行う変更は、そのドキュメント内だけに格納されます。

エミュレータでキーマップを変更するには

- 1 端末エミュレータで [プロパティ] メニューをクリックしてから [キーボード] をクリックします。
- 2 編集するキーマップ名をクリックしてから [編集] をクリックします。
キーマップエディタ内に選択したキーマップが一時ファイルとして表示されます。



[キーボードプロパティ] ダイアログボックスで [新規作成] をクリックすると、このエミュレータドキュメントで使用する新しいキーマップを作成することもできます。

- 3 必要なキー定義を設定します。たとえば、F9 を文字列 “End” にマップする場合には、キーボードの F9 をクリックします。[マッピング] メニューで [文字列] をクリックした後、[マッピング] ボックスで “End” と入力します。ツールバー上の緑色マークをクリックして定義を確認します。
- 4 定義し終わったら、[ファイル] メニュー上の [上書き保存] をクリックして変更を保存します。
変更は、エミュレータドキュメントの一部として保存されます。
- 5 [ファイル] メニューで [アプリケーションの終了] をクリックすると [キーボードプロパティ] ダイアログボックスに戻ります。

手順の詳細については、端末エミュレータのヘルプを参照してください。同様の情報が Vision ヘルプの「端末エミュレータの使い方」の「キーボードの構成」にもあります。ここには次の内容のヘルプが用意されています。

- ウィンドウの表示形態の変更
- キーマップの使用

第 12 章

リモートホストの参照

このセクションでは、次の内容について解説します。

- 参照の概要
- Windows 2000/Me/98/95 または Windows NT 4.0 での参照
- Windows NT 3.51 での参照
- VCA ファイルアクセスサービス
- ファイルタイプ
- ファイルのコピーと移動
- デスクトップショートカット
- プログラムマネージャへのコピー

参照の概要

X Vision ソフトウェアは、リモートホストを参照するためのいくつかの機能を備えており、次のような利点があります。

- ローカルハードディスクを参照するのと同じようにリモート UNIX ホストを簡単に参照できます。
- ファイルの移動、コピー、削除、名前変更が可能です。
- PC システムと UNIX システム間でファイルをコピーできます。
- ファイルタイプを登録しておいて、プログラムやドキュメントのアイコンをダブルクリックすることによって、プログラムを起動したり、ドキュメントを開いたりできます。
- デスクトップ上で UNIX プログラムやドキュメントへのショートカットを作成できます。
- Windows 2000/Me/98/95 や Windows NT 4.0 ユーザにとって、Unix ネットワークコンピュータブラウザはネットワークコンピュータと同じような見た目と使い心地を備えており、ホストエクスプローラはエクスプローラに似ています。実際、マイコンピュータ、ネットワークコンピュータ、エクスプローラがすべて Windows 上で同じプログラムを実行するように、Unix ネットワークコンピュータとホストエクスプローラは1つのプログラムを異なるモードで使用しています。
- Windows の以前のバージョンのユーザの場合、ファイルマネージャに似たホストマネージャアプリケーションを使用できます。
- ドラッグアンドドロップを使用して、ファイルを移動またはコピーしたり、ドキュメントを開いたり印刷するなどのアクションを実行できます。

注意 本書の内容は Unix ネットワークコンピュータとホストエクスプローラの両方に適用されますが、ホストエクスプローラのプログラム名を使用しています。

Windows 2000/Me/98/95 または Windows NT 4.0 上での参照

Unix ネットワークコンピュータとホストエクスプローラを使用すると、Windows 2000/Me/98/95 や Windows NT 4.0 ユーザは UNIX ホストをいくつかの方法で表示できます。

Unix ネットワークコンピュータを使用するには

- 1 デスクトップ上の [Unix ネットワークコンピュータ] をダブルクリックします。
ホストファインダが検出したホストと、手作業で設定したホストが表示されます。



ホストファインダによって検出されたが、現在使用可能でないホストには、アイコンマークに×印が付いています。

- 2 使用可能なホストをダブルクリックして、そのホスト上のホームフォルダの内容を表示します。



ファイルは認識されるファイルタイプ別に固有のアイコンを持っています。右マウスボタンでアイコンをクリックすると、実行可能な操作を示すショートカットメニューが表示されます。

ホストエクスプローラを使用するには

- 1 [スタート]メニューから [プログラム]、[CentreNET X Vision] の順に選択し、[ホストエクスプローラ] をクリックします。



- 2 ウィンドウの左側のホストをクリックします。このホストに初めて接続する場合には、ユーザ名とパスワードの入力を求めるプロンプトが表示されます。ツリーが展開されて、ホスト上のホームフォルダの内容が表示されます。



詳細については、[ヘルプ]メニューの[トピックの検索]をクリックします。メインメニューの[Vision ヘルプの目次]から[ホストエクスプローラの使い方]を選んでも、同じ情報が表示されます。次の内容についてのヘルプが表示されます。

- ウィンドウの見た目の変更方法
- ホストの操作方法
- ファイルとフォルダの操作方法
- ファイルタイプの操作方法
- 関連付けの使い方

[パス] ボタンの使い方

ホストエクスプローラを使用して UNIX 版 Vision サービスを実行しているホストに接続すると、[パス] ボタンを使用して、UNIX パスの中にあるディレクトリに迅速にアクセスできます。ホストエクスプローラは PATH 環境変数を直接に読み取れないので、PATH の内容をファイルに書き込むには `dumppath` ユーティリティを実行する必要があります。これが自動的に行われるようにするには、スタートアップファイル (`.login` または `.profile`) にこのプログラムを実行するコマンドを挿入するか、すべてのユーザがこの機能を必要とする場合には、このようなコマンドをグローバルスタートアップファイルに追加するようにシステム管理者に要求します。

`dumppath` によって作成されたファイルがない場合には、ファイルアクセスサービスによって提供されるデフォルトの設定が表示されます。

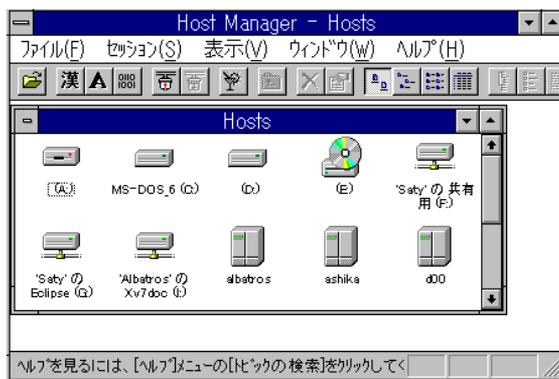
Windows NT 3.51 での参照

ホストマネージャを使用して、Windows NT 3.51 のユーザは UNIX ホストを参照できます。

ホストマネージャを使用するには

- 1 プログラムマネージャの [Vision グループ] を開き、[ホストマネージャ] アイコンをダブルクリックします。

ウィンドウが表示され、使用している PC ディスク、接続されている Windows ドライブ、現在設定されている UNIX ホストが表示されます。



- 2 これらの項目の内容を表示するには、該当する項目をダブルクリックします。



ファイルは認識されるファイルタイプ別に固有のアイコンを持っています。右マウスボタンでアイコンをクリックすると、実行可能な操作を示すショートカットメニューが表示されます。

詳細については、[ヘルプ]メニューの[トピックの検索]をクリックします。メインメニューの[Vision ヘルプの目次]から[ホストマネージャの使い方]を選んでも、同じ情報が表示されます。次の内容についてのヘルプが表示されます。

- ウィンドウの見た目の変更方法
- ホストの操作方法
- ファイルとフォルダの操作方法
- ホストマネージャ内での移動方法
- オプションの設定方法
- ファイルタイプの操作方法
- 関連付けの使い方

VCA ファイルアクセスサービス

このセクションでは、次の内容について説明します。

- 概要
- ファイルアクセス方式の選び方
- USP ファイルアクセスサービス
- FTP ファイルアクセス

VCA ファイルアクセスサービスの概要

Vision コミュニケーション アーキテクチャー (VCA) には、UNIX システム上でファイルやフォルダを参照したり操作するための機能を提供するファイルアクセスサービスが含まれています。

VCA ファイルアクセスサービスは、次の方法によって提供されます。

- UNIX ホスト上に UNIX 版 Vision サービスがインストールされている場合は、USP ファイルアクセスサービスを使用します (X Vision ではご使用にできません)。
- TCP/IP ネットワーク上では FTP ファイルアクセスを使用します。

アプリケーションは、ファイルアクセス API を介してサービスを使用します。

FTP 方式を使用するときには、いくつかの制限事項があります。

ファイルアクセス方式の選び方

Windows のコントロールパネルで Vision コミュニケーションオプションを使うと、トランスポートを設定できます。トランスポートを追加したり変更すると、[トランスポート] ダイアログボックスの [拡張機能] タブを使って、ファイルアクセスサービスに採用する方式を選択できます (「Vision コミュニケーション アーキテクチャ」の章の「トランスポートの構成」を参照してください)。



トランスポートが USP プロトコルを使用している場合は、USP ファイルアクセスサービスを使用します。トランスポートが TCP/IP プロトコルを使用している場合は、FTP ファイルアクセスを使用します。

USP ファイルアクセスサービス (サポート対象外)

Vision USP ファイルアクセスサービスを使うと、リモートホスト上に格納されているファイルを参照したり操作したりできます。また、ホスト間でファイルを転送することもできます。ファイルアクセスサービスは、設計上次の点が重要です。

- ファイルを開く、閉じる、読み取る、書き込む、移動する、削除する機能。
- フォルダを作成する、開く、閉じる、変更する、削除する機能。
- ファイルやフォルダについての情報を提供できます。
- `dumpath` コマンドを実行して `.visionpath` ファイルを作成している場合は、ホストエクスプローラを使ってユーザの UNIX パスにあるフォルダを表示できます。このファイルが存在しない場合は、ファイルアクセスサービスによって提供されるデフォルトの設定が表示されます。
- PC システムと UNIX システム間でファイルをコピーできます。
- UNIX システム間でファイルを直接にコピーできます。
- 転送しようとするファイルにテキストデータを入れるのか、バイナリデータを入れるのかを指定できるとともに、テキストファイル上の改行文字を適切に変換できます。
- ファイルのアクセス権を変更できます。
- ファイル転送中に状態情報を提供したり、転送を終了したりできます。
- ファイルアクセスサービスは Vision 遠隔手続き呼び出し (RPC) サービス上で動作します。

FTP ファイルアクセス

(UNIX 版 Vision サービスがインストールされていないために) USP サポートなしに TCP/IP トランスポートを使ってホストにアクセスすると、ファイルアクセスに FTP プロトコルが使用されます。ファイルを処理する機能は、USP ファイルアクセスサービスについて説明した機能に似ていますが、次のような例外があります。

- ファイルのアクセス権を変更できません。
- ホストエクスプローラに [パス] フォルダを表示できません。
- UNIX システム上のフォルダ間や、リモートホスト間でファイルをコピーする場合、ファイルは PC に転送されてから、宛先フォルダに戻されます。USP ファイルアクセス機能があれば、ファイルはコピー元から直接宛先にコピーされます。

FTP ホストにアクセスすると、FTP 変換のテキストを表示できます。Unix ネットワークコンピュータでは、ホストを開いて、[表示] メニューの [ホストからの応答] をクリックすると、[ホストからの応答表示] ウィンドウが表示されます。



ホストマネージャでは、[ウィンド]メニューの[テキスト表示領域]をクリックすると、ウィンドウの一番下に[テキスト表示領域]が表示されます。



ファイルタイプ

このセクションでは、次の内容について解説します。

- ファイルタイプの概要
- ローカルファイルタイプ
- リモートファイルタイプ
- パターンマッチング
- アクション
- 属性
- 関連付け

ファイルタイプの概要

ホストエクスプローラとホストマネージャは特定のファイルタイプを認識するので、ユーザはアイコンをダブルクリックすることによってプログラムやドキュメントを開くことができます。認識されたファイルを表すのに固有のアイコンが使用されます。

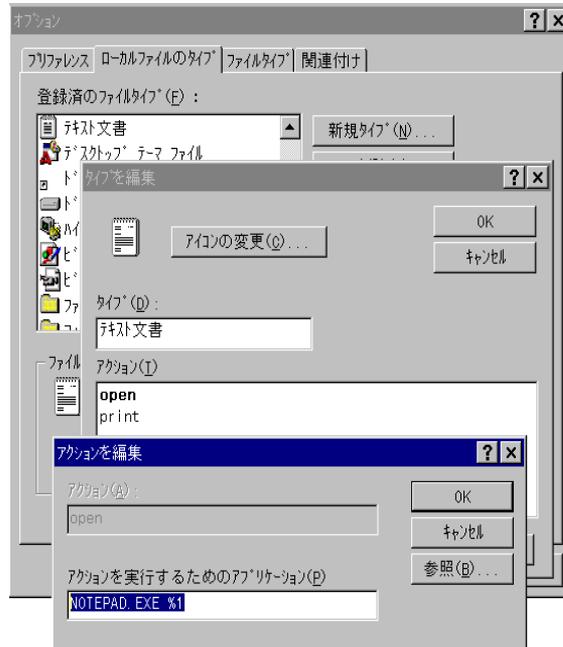
ホストエクスプローラを使うと、リモート UNIX ファイルのファイルタイプを設定できます。ホストマネージャを使うと、リモートファイルタイプとともに、ローカル Windows ファイルタイプも設定できます。

ローカルファイルタイプ

ホストマネージャを使うと、ファイルマネージャと同じようにしてローカルファイルタイプを設定できます。ホストマネージャは、Windows レジストリの **Hkey_Local_Machine\Software\Classes** セクションを変更します。ホストマネージャで設定されたファイルタイプはファイルマネージャに表示され、ファイルマネージャで設定されたファイルタイプもホストマネージャに表示されません。

Windows ファイルタイプは、ファイル名の中の3文字の拡張子とそれに続くドット文字 (.) によって識別できます。Windows 2000/Me/98/95や Windows NT で長いファイル名が使用されているときでも、すべてのファイルが拡張子を1つずつ持っています。

それぞれのファイルタイプには、そのファイル上で実行される1つまたは複数のアクションが関連付けられています。アクションごとに、そのアクションを実行するのに使用されるコマンドを指定できます。たとえば、.TXT 拡張子に関連付けられているテキストファイルタイプには、Windows をインストールするときに Open アクションと Print アクションが設定されます。Open アクションに指定されるコマンドは **notepad %1**、Print アクションに指定されるコマンドは **notepad /p %1** です。このタイプのファイルが選択されると、アクションをメニューの中で使用できます。



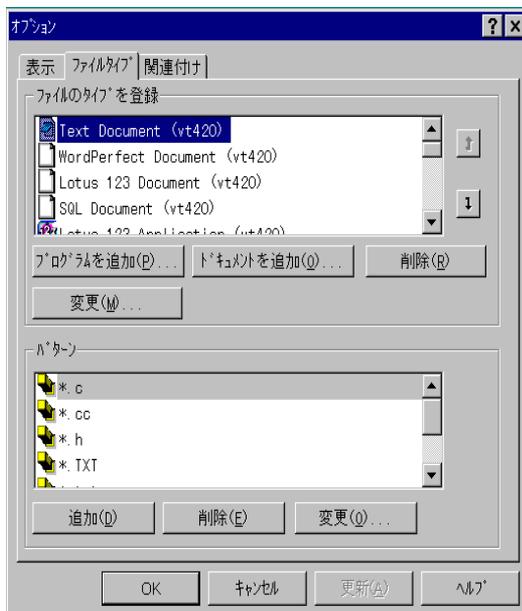
リモートファイルタイプ

リモートファイルタイプは、ホストエクスプローラまたはホストマネージャのいずれかから設定でき、その情報は **¥ Program Files¥Vision¥User¥Browser¥Classes** フォルダの中のサブフォルダにファイルとして格納されます。リモートファイルタイプには、プログラムとドキュメントの2つのカテゴリがあります。プログラムタイプはさらに、X Vision 端末エミュレータやリモートプログラムスタータに対応する基本プログラムタイプと、基本プログラムタイプのうちの1つに「もとづいて」決定される UNIX アプリケーションタイプの2つのカテゴリに分類されます。

購入した X Vision プロダクトに応じて、適切なリモートファイルタイプが選択され、インストールされます。新しいプログラムタイプを追加するときは、既存のタイプにもとづいて行います。アプリケーションをプログラムファイルタイプとして登録してからでないと、関連するドキュメントタイプを登録できません。

プログラムタイプを追加するには

- 1 ホストエクスプローラかホストマネージャで、[表示]メニューの[オプション]をクリックします。
- 2 [ファイルタイプ]タブをクリックします。



- 3 [プログラムの追加] をクリックします。



- 4 スペースの中に説明を入力します。[基本パターン] ボックスのドロップダウンリストから既存のプログラムタイプを選びます。必要であれば、run 引数と drop 引数を追加します。普通は、[高度な設定] オプションは変更する必要がありません。これらのダイアログボックスのオプションについての情報を得るには、コンテキストヘルプを使用します。Windows 2000/Me/98/95 や Windows NT の場合は、タイトルバーの **?** をクリックしてから、オプションをクリックします。Windows NT 3.51 の場合は、SHIFT+F1 を押してヘルプポインタを表示してから、オプションをクリックします。

[OK] ボタンは、[基本パターン] で何かを選択するまで、または [高度な設定] セクションでドキュメントタイプを選択するまでグレー表示されます。

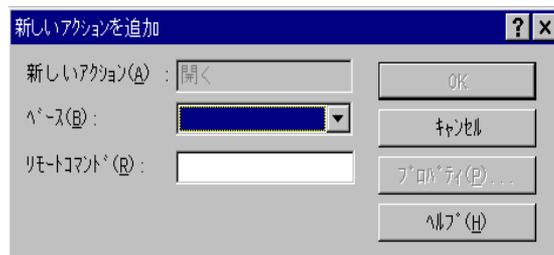
新しいプログラムタイプは[ファイルのタイプを登録]ボックスに表示されます。ここで1つまたは複数のパターンを追加すると、ブラウザプログラムがこのタイプのファイルを認識できるようになります。

ドキュメントタイプを追加するには

- 1 [ホストエクスプローラ]か[ホストマネージャ]で、[表示]メニューの[オプション]をクリックします。
- 2 [ファイルタイプ]タブをクリックします。
- 3 [ドキュメントを追加]をクリックします。



- 4 スペースに説明を入力し、テキストファイルの場合は[ASCIIドキュメント]をクリックします。
- 5 [アイコンを変更]をクリックして、ドキュメントタイプを表わすアイコンを選びます。[現在のアイコン]ボックスのアイコンをクリックするか、[参照]を使用してアイコンを含む別のWindowsファイルを選択します。
- 6 [動作]領域で[追加]をクリックします。



- 7 Open などの名前を入力します。このタイプのドキュメントのアクションを処理する PC 上で実行したい、VT420 エミュレータなどのプログラムのタイプを選びます。次に、ビューワプログラムを起動したりファイルを読み込むなどのアクションを実行するために UNIX システムで実行するリモートコマンドを指定します。選択したプログラムの属性を変更するには、[プロパティ] をクリックします。

アクションがリストに表示されます。別のアクションを追加するには、ステップ 6 を繰り返します。

ファイルタイプの変更や削除については、[ヘルプ] メニューの [トピックの検索] をクリックします。メインメニューの [Vision ヘルプの目次] から [ホストエクスプローラの使い方] や [ホストマネージャの使い方] を選んでも、同じ情報が表示されます。

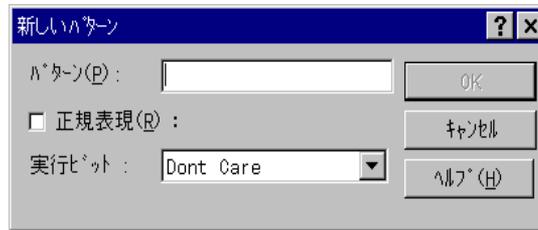
パターンマッチング

リモートファイルタイプには、1 つまたは複数のパス名パターンが関連付けられています。ファイルのパス名がいずれかのパターンと一致する場合は、ブラウザプログラムはこのタイプのファイルとして認識します。VT420 エミュレータファイルタイプには `bin/*` というパターンが設定されているので、`/usr/local/bin` などの “bin” を含むパス名を持つディレクトリの中のファイルは、VT420 アプリケーションとして認識されます。Uniplex などのアプリケーションの場合は、唯一のパターン設定は実行可能ファイル名である **uniplex** です。

ドキュメントファイルタイプの場合は、パターンの中にドット文字と識別文字からなるワイルドカード文字列が含まれるのが普通です。たとえば、Word Perfect ドキュメントの場合には、`*.doc` というワイルドカード文字列が含まれています。UNIX ではこれらはファイル名拡張子ではないため、パターンマッチング技術を使用する必要があります。この場合、ファイルを拡張子で認識するという制限がないため、1 つのファイルタイプに対してパターンをいくつでも設定できるという利点があります。たとえば、テキストファイルタイプには当初、`*.c`、`*.cc`、`*.h`、`*.txt`、`docs/*`、`README*` のパターンが関連付けられているので、ファイル `README.TV` はテキストドキュメントとして認識されます。既存のファイルタイプに新しいパターンを簡単に追加できます。

新しいパターンを追加するには

- 1 ホストエクスプローラがホストマネージャで、[表示] メニューの [オプション] をクリックします。
- 2 [ファイルタイプ] タブをクリックします。
- 3 ファイルタイプをクリックします。
- 4 [パターン] 領域で [追加] をクリックします。



- 5 スペースの中にパターンを入力し、使用している式のタイプを指定します。このようなダイアログボックスのオプションについての情報を得るには、コンテキストヘルプを使用します。
- 6 実行ビットの設定が関連しているかどうかを指定します。

アクション

ドキュメントファイルタイプの場合は、「開く」や「編集」などの1つまたは複数のアクションを指定します。これらのアクションはいずれかのプログラムタイプにもとづいており、UNIXホストで実行するコマンドも指定しています。たとえば、Word Perfect ドキュメントタイプの「開く」アクションは Word Perfect Application ファイルタイプにもとづいており、このファイルタイプは VT220 モードで動作するように設定されている VT420 エミュレータファイルタイプにもとづいています。「開く」アクションに指定されるコマンドは **wp(%1)** です。このタイプのファイルが選択されると、ブラウザプログラムのメニューでアクションを使用できます。

属性

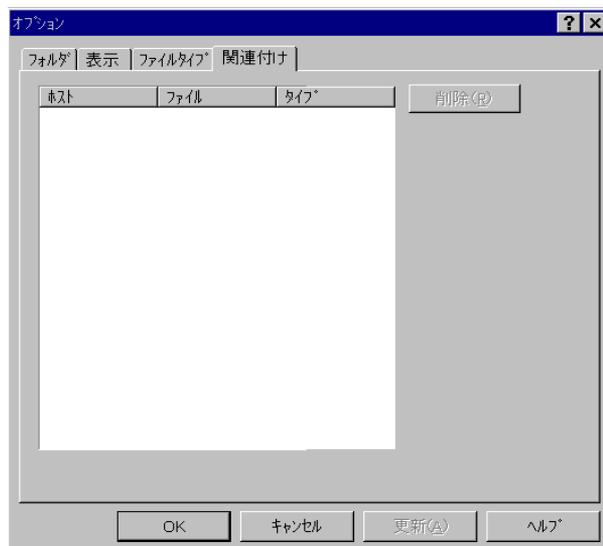
ファイルタイプの属性は、当初は元になるファイルタイプから継承されますが、変更可能です。事前に設定済みのエミュレータファイルタイプやリモートプログラムスタータファイルタイプの属性を変更できます。このような基本ファイルタイプの1つにもとづくアプリケーションプログラムファイルタイプは、基本ファイルタイプの属性を継承します。同様に、ドキュメントファイルタイプはアプリケーションプログラムファイルタイプの属性を継承します。

たとえば、Lotus 123 アプリケーション ファイルタイプは VT420 エミュレータにもとづいているので、[プログラムタイプ] ダイアログボックスの [プロパティ] ボタンをクリックすると、[VT420 エミュレータプロパティ] ダイアログボックスが表示されます。VT420 エミュレータファイルタイプから継承した属性は、たとえば [Lotus 123] アイコンを選択するように変更されています。Lotus 123 ドキュメントファイルタイプでは、別の属性を指定することもできますが、普通は変更しません。

関連付け

パターンマッチングによってUNIXファイルタイプを識別する方法は、Windowsファイルタイプで有効な拡張子方式ほど明確に識別できないという欠点があります。UNIXファイルシステムでは、設定されているパターンのうちの1つに一致する一部のファイルだけが自動的に認識されます。認識されないファイルはブラウザアプリケーションで汎用ドキュメントアイコンで表示され、ファイルタイプアクションを実行するコマンドを含むメニューではなく、[条件付きで開く]コマンドが入っています。このコマンドを使用すると、設定済みファイルタイプのリストからファイルを開くのに使用するプログラムを選ぶことができます。[関連付け]コマンドを使用すると、ファイルを開かずに、ファイルのタイプを手作業で指定できます。

[オプション]ダイアログボックスの[関連付け]タブには、設定した手動によるファイル関連付けが表示されます。パターンマッチング方式によって自動的に作成された関連付けは示されません。



関連付けの使い方については、[ヘルプ]メニューの[トピックの検索]をクリックしてください。メインメニューの[Vision ヘルプの目次]から[ホストエクスプローラの使い方]または[ホストマネージャの使い方]を選んでも、同じ情報が表示されます。

ファイルのコピーと移動

このセクションでは、次の内容について解説します。

- ファイルのコピーと移動の概要
- ファイル名
- 改行文字
- バッチファイル
- hostmgr コマンド

ファイルのコピーと移動の概要

ホストエクスプローラ (Windows 2000/Me/98/95 または Windows NT 4.0 の場合) やホストマネージャ (Windows NT 3.51 の場合) を使用して、次の場所でファイルを移動したりコピーしたりできます。

- UNIX ファイルシステム内
- UNIX ファイルシステム間
- PC と UNIX 間

ファイルやフォルダの移動やコピーについての基本的な説明が必要なときは、メインメニューの [Vision ヘルプの目次] から [ホストエクスプローラの使い方] または [ホストマネージャの使い方] を選んでください。

ファイル名

大文字/小文字を区別するファイル名

Windows 2000/Me/98/95 や Windows NT のファイル名は大文字/小文字を区別し、ファイル名は UNIX の場合と同じになります。マイコンピュータやファイルマネージャなどの Windows プログラムでファイルを表示するときは、ファイル名の大文字/小文字表示は実際と異なることがあるので注意してください。UNIX ファイルタイプ定義のパターンを指定するときに大文字バリエーションを指定しなかった場合には、問題が発生することがあります。たとえば、テキストファイルをパターン*.txtによって認識するように指定しておいて、ファイルを*.TXTとしてコピーすると、テキストファイルとして認識されません。UNIX にコピーしようとするファイルが登録済みのファイルタイプとして認識されない場合は、ファイルブラウザプログラム (ホストエクスプローラやホストマネージャ) に固有のアイコンで表示されず、関連付けられたアクションを使用できません。

無効なファイル名

ホストエクスプローラでは、PC ファイルを UNIX にコピーしたときにファイル名が無効であると、UNIX システムはエラーを出すか、通常の訂正アクションを実行します。たとえば、長いファイル名は切り捨てられます。UNIX ファイルを PC にコピーしようとして、Windows でファイル名が無効であると、ホストエクスプローラは有効なファイル名を作成しようと試みます。

ホストマネージャでは、宛先システムでファイル名が無効なときにアクションを制御するオプションを指定できます。指定できるのは、有効なファイル名を作成するオプションか、ユーザが新しい名前を確認したり新しい名前を入力するためのプロンプトを表示するオプションです。これらのオプションが無効な場合は、宛先オペレーティングシステムがエラーを出すか、通常の訂正アクションを実行します。

改行文字

ファイルをコピーしたり移動するときは、普通は、宛先システムにまったく同じバイナリコピーを作成したいでしょう。ただし、ASCII テキストファイルで異なる行末文字を使用するシステム間でテキストファイルをコピーするときは例外です。たとえば、Windows テキストファイルは行末でキャリッジリターンと改行文字を使用しますが、UNIX ファイルは改行文字しか使用しません。

デフォルトでは、ファイルタイプ指定の中の ASCII ドキュメント設定によって、改行文字を変換するかどうか自動的に決定されます。すなわち、認識されたファイルタイプを持つファイルには正しい変換が適用されますが、認識されないファイルの場合は、デフォルトではバイナリコピーに変換されます。ホストエクスプローラまたはホストマネージャでは、[転送モード]として[テキスト]または[バイナリ転送モード]を明示的に指定できます。[転送モード]が[自動設定]に設定されているときは、ツールバーの[テキスト転送モード]ボタンと[バイナリ転送モード]ボタンは両方とも押下されていません。

バッチファイル (サポート対象外)

ホストマネージャはコピー操作や移動操作のリストを含むバッチファイルを受け入れることができます。このファイルの形式は次のとおりです。

コピー元 宛先 オプション

ここで、コピー元と宛先は、それぞれ、コピーするファイルとその宛先を表わしています。オプションは「hostmgr コマンド」で説明したコマンド行スイッチです。次はバッチファイルエントリの例です。

```
local:c:¥steve¥accounts¥assets¥rvass1.xls vision:assets/* /d
local:d:¥work¥contract¥*.doc vision:contract/* /d
vision:¥usr2/stevel/prices.sbd local:c:¥steve¥*. * /d
```

バッチファイルを実行するには、ファイルマネージャの[ファイル名を指定して実行]ダイアログボックスにコマンドを次の書式で入力します。

hostmgr@ バッチファイル名

hostmgr コマンド (サポート対象外)

hostmgr コマンドは、ホストマネージャを起動するのに使用されます。コマンド形式は次のとおりです。

hostmgr [コピー元 宛先] [/d] [/e] [/m] [/n] [/N] [/r] [/R] [/t] [/x] [/k]

または

hostmgr [@バッチファイル]

注意 ホストマネージャを含むフォルダは普通は MS-DOS パスに含まれていないので、hostmgr コマンドを入力するときにフルパス名を指定する必要があります。

hostmgr キーワードの後にファイルが指定されていない場合は、Hostmgr.exe を起動するときに、転送用のファイルを選択しなければなりません。コマンド行にオプションが指定されている場合は、オプションの初期状態が設定されています。

コピー元と宛先が指定されている場合は、それぞれ、コピーするファイルと宛先を表わしています。両方にホスト名とファイル指定を指定できます。ホスト名とファイル指定はコロンで区切ります。たとえば、次のように入力します。

```
hostmgr local:c:%steve%myfile.txt vision /N /m /e
```

バッチファイル(前に@ シンボルが付いています)が指定されている場合は、ホストマネージャはそのファイルに転送リストがないかどうかを調べます。バッチファイルは、次の形式の行を含む ASCII テキストファイルです。

コピー元 宛先 オプション

これは、コマンド行で直接に入力するときと同じです。

オプションスイッチには、次のものがあります。

- /d ファイルをバイナリデータとして転送します。このオプションは、行末テキストファイルを除くすべてのファイルに使用します(デフォルト)。
- /e 完了後に終了します。
- /m 最小化されているホストマネージャを起動します。
- /n 完了時に通知します(デフォルト)。
- /N 完了時に通知しません。
- /r 宛先システム上の既存のファイルを置換するかどうかを確認するプロンプトを表示します。

- /R 確認のためのプロンプトを表示しません (デフォルト)。
- /t テキストファイルを転送します。
MS-DOS ファイルを UNIX に移動するときには、各行末の <CR><LF> (ASCII 13, 10) を <LF> (ASCII 10) と置換します。ファイルを UNIX から MS-DOS に移動するときは、逆に置換します。
- /x ソースファイルがローカルホストにある場合、完了時に削除します。
- /X ソースファイルを削除しません (デフォルト)。
- /k 漢字を含むテキストファイルを転送します。

バッチファイル転送を使用するとき、/t などの一部のスイッチはバッチファイルの中の個々の行に適用され、/e などのスイッチはバッチファイルを起動するコマンド行に指定されます。これは、スイッチがファイルを参照するのか、転送アプリケーション自体を参照するのかによって決まります。

デスクトップショートカット

Windows 2000/Me/98/95 と Windows NT 4.0 では、画面の背景をデスクトップといいます。UNIX フォルダ、プログラム、ドキュメントを表わすアイコンを使ってデスクトップ上にショートカットを作成できます。Unix ネットワークコンピュータに含めたい項目を選択して、そのアイコンをデスクトップにドラッグします。



フォルダショートカットをダブルクリックすると、Unix ネットワークコンピュータが起動されてフォルダの内容が表示されます。プログラムショートカットをダブルクリックすると、そのプログラムが起動され、ドキュメントショートカットをダブルクリックすると、読み込まれたドキュメントに関連付けられているプログラムが起動されます。

UNIX ファイルやフォルダをデスクトップにドラッグすると、**¥Windows¥Desktop** フォルダにリモート情報ファイル(.rif) が作成されます。このファイルの中には、関連する UNIX ファイルのホスト、パス、アイコンについての情報が入っています。プログラムやドキュメントの場合は、Unix ネットワークコンピュータと同じメカニズムが使用されて、正しいプログラムが起動されます。フォルダの場合は、Unix ネットワークコンピュータが起動されて、フォルダの内容が表示されます。

プログラムマネージャへのコピー

ホストマネージャを使用している場合は、プログラムやドキュメント項目を Windows プログラムマネージャにコピーできます。項目を選択し、[ファイル]メニューから [プログラムマネージャへコピー] を選びます。プロンプトが表示されるので、エクスポートしたいアクション (たとえば、"開く" や "編集") と中に項目を作成したいプログラムマネージャグループを選びます。作成されるプログラムマネージャ項目には、Windows ビューワプログラムを実行したり、アクションテンプレートを読み込んだり、ホストに接続したり、UNIX プログラムを実行したり、ドキュメントを読み込んだりするためのコマンド文字列が入りません。

(空白のページです)

第 13 章

リモートプログラムの実行

このセクションでは、次の内容について解説します。

- リモートプログラムの実行の概要
- リモートプログラムの実行
- VCA リモート実行サービス
- リモートプログラムスタータのドキュメント

リモートプログラムの実行の概要

X Vision ソフトウェアには、端末エミュレータを使ってログインしなくても UNIX プログラムを起動できる機能があります。主な利点は次のとおりです。

- Vision X サーバがインストールされている場合、X クライアントを簡単に起動できます。
- デモンプロセスのようなバックグラウンドタスクなど、ユーザとの対話を必要としないプログラムを起動するのに便利な方法です。
- コマンドとそのコマンドを実行したいホストを指定します。ホストへの接続とユーザの認証のプロセスは自動的に実行されます。
- ホストや実行可能プログラムの UNIX プログラムを参照できます。
- リモートプログラムスタータ設定をデスクトップ上やフォルダの中のドキュメントに保存できます。ドキュメントアイコンをダブルクリックすれば、UNIX プログラムを起動できます。必要であれば、X サーバを自動的に読み込みます。
- UNIX システムメッセージやプログラムによるテキスト出力などの出力内容を表示するかどうかを指定できます。

プログラムの実行

PC 上でリモートプログラムスタータアプリケーションを使って、リモートホストでプログラムを実行できます。

プログラムを実行するには

- 1 次のいずれかを行います。
 - [スタート] ボタンをクリックし、次に [Unix プログラムの実行] をクリックします。
 - [スタート] ボタンをクリックし、[プログラム]、[CentreNET X Vision]、[アクセスリ] の順に選択してから、[プログラムスタータ] をクリックします。
 - Windows の以前のバージョンの場合、X サーバがシングルウィンドウモードにあるときは、コントロールメニューの [ファイル名を指定して実行] をクリックします。
 - Windows の以前のバージョンの場合、プログラムマネージャの [CentreNET X Vision] グループを開き、[プログラムスタータ] アイコンをダブルクリックします。

新しい [リモートプログラムスタータ] ウィンドウが表示されます。



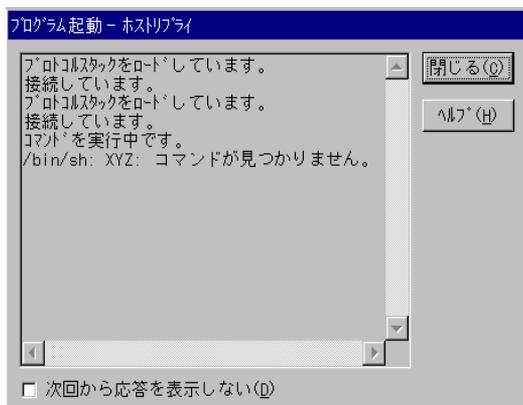
- 2 [ホスト] で、ホスト名を入力するか、ドロップダウンリストからホストを選びます。
- 3 [コマンド] ボックスにホストシステムで実行するコマンドを入力します。ドロップダウンリストを使って、最後に使用された 5 つのコマンドの中から選べます。

UNIX システムの場合は、コマンド構文がリモートシステムのシステムプロンプトに対して入力する構文と似ています。ただし、プロセスを端末から切り離して実行することを示すアンパサンド (&) は入力しません。X アプリケーションの場合は、デフォルトで **DISPLAY** 環境変数が自動的に設定されるので、**-display** を入力する必要がありません。

もう 1 つの方法は、[参照] ボタンをクリックします。[ホスト] ボックスが空の場合は、設定されているすべてのホストが表示されます。[ホスト] ボックスにホストを指定した場合は、そのホスト上のホームディレクトリの内容が表示されます。実行したいプログラムやシェルスクリプトが見つかるまでホストを参照します。

- 4 他の設定を変更したい場合は、[プロパティ] メニューからコマンドを選びます。X アプリケーションを実行している場合を除いて、ディスプレイ、サイズ、カラー属性は関連ありません。
- 5 [実行] をクリックします。

これで、ホストに接続され、指定したコマンドが実行されます。デフォルトでは、状態メッセージやエラーメッセージは表示されず、UNIX プログラムからの出力も表示されません。この出力内容を見るには、[プログラム] 属性を編集して、[常に表示] または [データがある時] を変更する必要があります。こうすると、次の例に示すように、[ホストリプライ] ウィンドウに出力が表示されます。



詳細については、[ヘルプ] メニューの [トピックの検索] をクリックします。メインメニューの [Vision ヘルプの目次] から [リモートプログラムスタータの使い方] を選んでも同じ情報を表示できます。次についてのヘルプが表示されません。

- プログラムの実行方法
- 設定ファイルの操作方法

VCA リモート実行サービス

このセクションでは、次の内容について解説します。

- 概要
- リモート実行方式の選び方
- USP タスクサービス (リモート実行)
- `rexec` コマンドプロセッサ
- `rsh` コマンドプロセッサ

VCA リモート実行サービスの概要

Vision コミュニケーションアーキテクチャ (VCA) には、UNIX システム上でコマンドを実行するためのリモート実行サービスが含まれています。一旦コマンドを実行すると、端末サービスを使ってログインしてコマンドを実行した場合と異なり、PCとホスト間に常駐論理接続はなくなるのが普通です。

VCA リモート実行サービスは、次のいずれかの方法を使って提供できます。

- UNIX ホストに UNIX 版 Vision サービスがインストールされている場合は、USP タスクサービス。

- UNIX オペレーティングシステムや `rexec` などのネットワークソフトウェアの一部として提供されるリモート実行サービス。
- `telnet` などの端末サービス。

どの方法を使っても似たようなサービスが提供され、実行 API を介してアプリケーションで使用できます。

リモート実行方式の選び方

Windows のコントロールパネルでは、[Vision コミュニケーション] オプションを使ってトランスポートを設定できます。トランスポートを追加したり変更したりするときは、[トランスポートプロパティ] ダイアログボックスの [拡張機能] タブを使ってリモート実行サービスの使い方を選択できます。（「Vision コミュニケーションアーキテクチャ」の章の「トランスポートの構成」を参照してください）。



リモート実行方式を変更するには

- 1 変更したいサービスをクリックし、次に [変更] をクリックします。
- 2 [変更] ボックスで、ドロップダウンリストから方式を選びます。

トランスポートが USP プロトコルを使用している場合は、USP タスクサービスによって提供される USP 実行を使用します。トランスポートが TCP/IP プロトコルを使用している場合は、デフォルトは `rexec` ですが、代替方式として `rsh`、`rlogin`、`telnet` も使用できます。

USP タスクサービス(リモート実行)

Vision USP タスクサービスは、リモートホストシステムにログインして、プログラムを実行する機能を提供します。標準 UNIX サービスの代替として **rlogin** や **rexec** などのサービスも提供します。この機能は、トランスポートのリモート実行サービスを設定するときにオプションとして選べます。リモート実行に関連するタスクサービスは、設計上次の点が重要です。

- USP タスクサービスは、ネットワークで暗号化されていないユーザ名やパスワードを送信しない VCA 認証サービスを使用します。これに対して、**rexec** はクリアテキストとして送信するユーザ名やパスワードを必要とします。
- タスクサーバは **tsrv** と呼ばれ、端末実行サービスとリモート実行サービスの両方を提供します。端末タイプを指定しない呼び出しを受信すると、メッセージを切り離された実行要求として処理します。プロセスをフォークしてシェルを実行し、シェルはネットワークソケット接続を **tsrv** から継承します。**tsrv** はこれ以上シェルとやりとりしません。

rexec コマンドプロセッサ

リモート実行プロセッサ (**rexec**) は TCP/IP を使用するほとんどの UNIX システムで使用できます。

UNIX システム上では、**rexecd** は **rexec** ルーチンのサーバです。**rexecd** は、ユーザ名とパスワードの検証後にリモート実行機能を提供します。**rexecd** は事前定義されたポート (512) でリモート接続要求を受け取ります。

コマンドを実行しようとするアプリケーションは、**rexecd** ポートに向けて接続要求を発信します。接続が成功すると、アプリケーションはコマンド文字列を **rexecd** に渡します。**rexecd** は、ログイン時のようにユーザ情報の妥当性を検査し、認証が成功した場合は、ユーザのホームディレクトリに変えて、ユーザのユーザおよびグループ保護を確立します。いずれかのステップが失敗すると、接続は終了されます。

次に、コマンドがユーザの標準ログインシェルに渡されます。シェルは **rexecd** によって確立されたネットワーク接続を継承します。ここで、ユーザのログインによって指定されるシェルがコマンドを実行します。

rsh コマンドプロセッサ

リモートシェルプロセッサ (**rsh**) は TCP/IP を使用するほとんどの UNIX システムで使用できます (ただし、**remsh** と呼ばれることもあります)。**rsh** は **rexec** に比べて安全性が低いとみなされているので、可能であれば、**rexec** を使用する方がよいでしょう。

UNIX システムでは、**rshd** (または **remshd**) は **rsh** ルーチンのサーバです。**rshd** は、特権ポート番号の検証後にリモート実行機能を提供します。**rshd** は事前に定義されたポート (514) でリモート接続要求を受け取ります。

コマンドを実行しようとするアプリケーションは、`rshd` ポートに向けて接続要求を発信します。接続が成功すると、`rshd` はアプリケーションの発信元ポート番号を調べます。発信元ポート番号が 0~1023 の範囲内にないと、`rshd` は接続を終了させます。次に、`rshd` は、アプリケーションを実行している PC のアドレスを調べます。UNIX ホスト上のホスト名データベースに対応するエントリが存在しないホストとそのアドレスが関連付けられている場合は、`rshd` は接続を終了させます。`rshd` は PC ノードアドレスを使用して、`/etc/hosts` ファイルから PC ノード名を検索します。

アプリケーションはユーザ名を指定しなければなりません。ほとんどの UNIX システム上では、ユーザ名が UNIX ホストマシン上でのユーザ識別情報と解釈され、指定されるパスワードがあれば、そのパスワードがアプリケーションを実行している PC 上でのユーザ識別情報と解釈されます。

注意 ユーザ名とパスワードが逆になっている UNIX マシンもあります。すなわち、アプリケーションが指定するユーザ名がアプリケーションを実行している PC 上でのユーザ識別情報と解釈され、指定されるパスワードがあれば、そのパスワードが UNIX ホストマシン上でのユーザ識別情報と解釈されます。

ここで、`rshd` は次のようにしてユーザアカウントの妥当性を検査します。すなわち、`rshd` はパスワードファイルの中のユーザ名を検索し、ユーザのホームディレクトリへ `chdir` を実行します。検索または `chdir` のいずれかが失敗すると、接続は終了します。スーパーユーザとして設定されていないユーザアカウントの場合は、`rshd` は `/etc/hosts.equiv` (または `hosts.lpd`) ファイルに「同等」とみなされるホストのリストがないかどうかを探します。PC のノード名がこのファイルにあると、認証は成功したとみなされます。検索が失敗したり、ユーザがスーパーユーザの場合は、`rshd` は UNIX ホスト上のユーザ名アカウントのホームディレクトリにある `.rhosts` ファイルで、PC のノード名と PC ユーザ識別情報を探します。この検索が失敗すると、接続は終了します。

PC アプリケーションが指定するコマンドはユーザの標準ログインシェルに渡されます。シェルは `rshd` によって確立されたネットワーク接続を継承します。ここで、ユーザのログインによって指定されるシェルがコマンドを実行します。

リモートプログラムスタートのドキュメント

リモートプログラムを実行するのに必要なすべての設定をリモートプログラムスタート設定ドキュメントに保存できます。リモートプログラムスタートを実行し、オプションを設定してから、[ファイル]メニューから [上書き保存] を選びます。

使用している X Vision プロダクトによって UNIX アプリケーションウィザードがインストールされている場合は、これを使用して `.rps` 設定ドキュメントを作成できます。ウィザードはプロセスの実行手順を案内し、実行したい UNIX プログラムのタイプに応じてオプションを設定します。ウィザードを実行するには、[スタート]メニューの [Unix プログラムの登録] をクリックします。

(空白のページです)

第 14 章

印刷

このセクションの内容は次のとおりです。

- 印刷の概要
- UNIX から PC へのローカル印刷 (LPD による方法のみをサポート)
- PC から UNIX へのリモート印刷 (LPD による方法のみをサポート)

印刷の概要

UNIX のバージョンの中には、リモートプリンタをユーザが構成できる機能を提供しているものもあります。ユーザは、ローカルコンピュータに直結したプリンタに印刷するのと同じ方法で、ネットワーク経由でリモートプリンタにも印刷できます。X Vision ソフトウェアは、UNIX アプリケーションが PC に接続されたプリンタに印刷できるように、このリモート印刷機能を拡張しています。また、PC アプリケーションから UNIX プリンタへの印刷が可能です。

プリントサーバは PC 上で動作し、印刷要求を待機します。プリントサーバは、リモート UNIX システムに印刷要求を送信するクライアントにもなります。TCP/IP ネットワークを使用しており、UNIX ホストが標準 LPD サービスをサポートしているのであれば、プリントサーバは LPD サービスを使用できます。UNIX ホストが標準 LPD サービスをサポートしていない場合には、プリントサーバは Vision USP RPC サービスを使用します。

UNIX から PC へのローカル印刷

プリントサーバプログラムは PC 上で動作し、UNIX システムからの印刷要求を受け付けます。ユーザは 1 つ以上の“ローカルプリンタ”を構成して、論理プリンタ名を実際の Windows プリンタに対応づける必要があります。また、論理プリンタが標準の LPD プロトコルを使用して印刷要求を待機するか、あるいは Vision USP RPC プロトコルを使用して印刷要求を待機するかを選択できます。Default というローカルプリンタが X Vision ソフトウェアのインストール時に LPD プロトコルを使用するために構成され、セットアップされています。

LPD プロトコルを使用する場合は、UNIX 側でリモートプリンタを構成する必要があります。構成の手順は、この章で後述するように UNIX のバージョンによって異なります。

RPC プロトコルを使用する場合（サポート対象外）には、印刷データを `pcprtclnt` プログラムに渡すシェルスクリプトで印刷要求が処理できるように、UNIX プリンタを構成する必要があります。この UNIX プログラムは PC 上でプリントサーバと通信する RPC クライアントです。プリンタは `$VISION_ETCDIR/printers` という UNIX ファイルで構成する必要があります。このファイルの詳細については、付録の「UNIX ファイル形式」を参照してください。

`pcprtclnt` を呼び出すシェルスクリプトは、UNIX System V システムの印刷インタフェーススクリプトまたは BSD UNIX 印刷システム上の出力フィルタになります。スクリプトファイルは `VISION_ETCDIR` ディレクトリにあります。リモートプリンタの構成手順の詳細については、「UNIX System V で RPC を使用するためのリモートプリンタの構成」または「BSD UNIX で RPC を使用するためのリモートプリンタの構成」を参照してください。

UNIX から印刷するときには、印刷出力が Windows プリンタで解釈できることを確認する必要があります。UNIX アプリケーションで適切なプリンタが選択されます。PostScript 出力には、テキストのままの出力を PostScript コードにパッケージ化するフィルタを使用することもできます。代替方法として、[ローカルプリンタ] 設定の [ラインプリンタモード] オプションを有効にすると、出力はテキストとして解釈されます。このオプションを有効にしたままで PostScript 出力をプリンタに送信すると、PostScript コードが印刷されることに注意してください。

ローカルプリンタの追加

ローカルプリンタを追加するには

- 1 Windows のコントロールパネルで [Vision サービス] をダブルクリックしてから [印刷] タブをクリックします。



現在構成されているローカルとリモートのプリンタが [プリンタ] リストに表示されます。Default というローカルプリンタは、X Vision ソフトウェアのインストール時にデフォルトの Windows プリンタとして構成されるようにセットアップされ、LPD プロトコルを使用します。このプリンタを編集して別のプリンタを指定するか、プロトコルを変更することができます。

- 2 [ローカルプリンタ追加] をクリックして、新しいローカルプリンタを追加します。



- 3 [論理プリンタ名] ボックスに新しい名前を入力します。
- 4 [物理プリンタ名] ボックスにドロップダウンリストから Windows のプリンタを選択します。
- 5 必要があれば他のオプションも変更します。Windows 2000/Me/98/95 または Windows NT では、タイトルバーで **?** をクリックしてから項目をクリックすると、項目の [ヘルプ] を得ることができます。Windows NT 3.51 では、SHIFT+F1 を押すとヘルプポイントが表示されるので、項目をクリックします。

PC のプリンタは、UNIX ホスト上のリモートプリンタとしても構成する必要があります。

BSD UNIX で LPD を使用するためのリモートプリンタの構成

LPD を使用するために BSD UNIX 上でリモートプリンタを構成するには

- 1 テキストエディタでファイル `/etc/printcap` を編集します。編集にはスーパーユーザ (root) 特権が必要です。
- 2 ファイルの最後に新しいプリンタ定義を追加します。次の例を参照してください。

```
pcprinter|My PC printer:¥
      :sd=/usr/spool/lpd/pcprinter:lf=/usr/adm/lpd-errs:¥
      :sh:rm=mypc:rp=default:
```

プリンタは UNIX 上では `pcprinter` という名前になります。sd フラグで、必ず存在しているスプールディレクトリを指定します。lf オプション名はエラー用のログファイルです。sh フラグはヘッダ抑制の意味であり、PC プリンタが PostScript デバイスの場合には必ず指定します。rm フラグはリモートコンピュータを指し、この例では `mypc` という PC を指します。rp フラグは PC 上の [ローカルプリンタ] サービス設定で指定された論理プリンタ名を指定する必要があり、この例では `default` になります。

ヒント

- `/etc/printcap` を編集する際には、行の最後に余計なスペースは入れないでください。
- 次のコマンドを使用すると、プリンタキューを表示できます。

```
lpqPpcprinter
```

BSD UNIX で RPC を使用するためのリモートプリンタの構成 (サポート対象外)

RPC を使用するために BSD UNIX 上でリモートプリンタを構成するには

- 1 テキストエディタでファイル `/usr/local/vision/etc/printers` を編集します。編集にはスーパーユーザ (root) 特権が必要です。
- 2 ファイルの最後に新しいプリンタ定義を追加します。次は、`mypc` という PC に接続され、UNIX 上では `pcprinter`、PC 上では [ローカルプリンタ] サービスで `default` と呼んでいるプリンタの例です。

```
pcprinter:mypc:default
```

詳細については、付録の「UNIX ファイル形式」を参照してください。

- 3 ファイル `/usr/local/vision/etc/lpr_rpc` をローカルプリンタ名 (UNIX 用のプリンタ名) と同じファイルにコピーまたはリンクします。たとえば、プリンタが `pcprinter` という名前の場合には、`/usr/local/vision/etc/lpr_rpc` を `/usr/local/vision/etc/pcprinter` にリンクします。このシェルスクリプトは実行可能である必要があります。
- 4 テキストエディタでファイル `/etc/printcap` を編集します。編集にはスーパーユーザ (root) 特権が必要です。
- 5 ファイルの最後に新しいプリンタ定義を追加します。次の例を参照してください。

```
pcprinter|My PC printer:¥
:sd=/usr/spool/lpd/pcprinter:lf=/usr/adm/lpd-errs:¥
:sh:lp=/dev/null:of=/usr/local/vision/etc/pcprinter:
```

プリンタは UNIX 上では `pcprinter` という名前になります。sd フラグで、必ず存在しているスプールディレクトリを指定します。lf オプション名はエラー用のログファイルです。sh フラグはヘッダ抑制の意味であり、PC プリンタが PostScript デバイスの場合には必ず指定します。lp フラグは、出力用にオープンするデバイス名であり、`/dev/null` に設定します。of フラグは出力フィルタを指定します。この出力フィルタは手順 3 で作成されたシェルスクリプトです。

新しいプリンタは、ユーザが何もしなくても有効になります。

ヒント

- `/etc/printcap` を編集する際には、行の最後に余計なスペースは入れないでください。
- 次のコマンドを使用すると、プリンタキューを表示できます。

```
lpq -p pcprinter
```

UNIX System V で LPD を使用するためのリモートプリンタの構成

LPD を使用するために UNIX System V 上でリモートプリンタを構成するには

- UNIX System V をベースにした UNIX システムにリモートプリンタを構成する場合には、さまざまな方法が用意されています。詳細は UNIX のマニュアルを参照してください。

UNIX System V で RPC を使用するためのリモートプリンタの構成 (サポート対象外)

RPC を使用するために UNIX System V 上でリモートプリンタを構成するには

- 1 テキストエディタでファイル `/usr/local/vision/etc/printers` を編集します。編集にはスーパーユーザ (root) 特権が必要です。
- 2 ファイルの最後に新しいプリンタ定義を追加します。次は、`mypc` という PC に接続され、UNIX 上では `pcprinter`、PC 上では [ローカルプリンタ] サービスで `default` と呼んでいるプリンタの例です。

```
pcprinter:mypc:default
```

詳細については、付録の「UNIX ファイル形式」を参照してください。

- 3 プリントキューが空であることを確認してから、次のコマンドを入力してプリントスケジューラをシャットダウンします。

```
/usr/lib/lpsht
```

- 4 次のようなコマンドを入力して、新しいプリンタインタフェースを作成します。

```
/usr/lib/lpadmin-pcprinter-v/dev/null-m dumb
```

`-p` スイッチで新しいプリンタが `pcprinter` という名前であることを指定します。`-m` スイッチではプリンタのモデルを指定し、ここでは PC に接続されている適切なプリンタのモデルを選択します。システムに最適なモデルがない場合には、`dumb` のように一般的なプリンタを選択するか、`dumb` に相当するシステムの名前を指定してください。

このコマンドによって、ユーザがプリンタ用に指定した名前でプリンタインタフェーススクリプトファイルが作成されます。格納位置は、UNIX のバージョンによって異なりますが、`/usr/spool/lp/interface` や `/usr/spool/lp/admins/lp/interfaces` などのディレクトリに格納されます。たとえば、`/usr/spool/lp/interface/pcprinter` のようなファイル名になります。

- 5 ここで、このプリンタインタフェースファイルを X Vision ソフトウェアが提供するプリンタインタフェースファイルに置き換えるか、この2つのファイルをマージする必要があります。スクリプト `lp_rpc` は PC 上のプリントサーバとやり取りをする RPC クライアントプログラム `pcprtclnt` を実行します。

プリンタ指定に `dumb` を使用している場合には、`lpadmin` で作成したプリンタインタフェースファイルの名前を (`old_pcprinter` などに変更して)、`/usr/local/vision/etc/lp_rpc` を `interface` ディレクトリにコピーし、その名前をオリジナルのプリンタ名にします。次の例を参照してください。

```
cp/usr/local/vision/etc/lp_rpc/usr/spool/lp/interface/pcprinter
```

このスクリプトは次の例のように実行可能な状態にしてください。

```
chmod +x pcprinter
```

`dumb` 以外のプリンタを使用している場合には、`lpadmin` で作成したプリンタインタフェースファイルを `lp_rpc` の相当する部分にマージする必要があります。

- 6 次のコマンドを入力して、新しいプリンタを使用可能な状態にします。

```
enablepcprinter
```

- 7 次のコマンドを入力して、もう一度スケジューラを起動します。

```
/usr/lib/lpsched
```

- 8 スプーラに対して次のコマンドを入力して新しいプリンタを受け付けるように命令します。

```
/usr/lib/acceptpcprinter
```

これで、プリントデバイス `pcprinter` が UNIX ユーザのプリントジョブのキューイング用に使用できるようになりました。

ヒント

- 次のコマンドを使用するとプリントキューを表示できます。

```
lpstatt
```

印刷要求の表示

PCで[プリントサーバ]ウィンドウを表示できるようにすると、リモートUNIXシステムから受信した印刷要求を確認することができます。



[プリントサーバ]ウィンドウを表示させるには

- [ウィンドウを表示する]ボックスをクリックするとチェックマークが表示されるので、[更新]または[OK]をクリックします。

PC から UNIX へのリモート印刷

PC上のプリントサーバは、実際にはWindowsプログラムでUNIXプリンタへ印刷要求を送信できるクライアントとしても働きます。1つ以上のリモートプリンタを構成して、WindowsプリンタとリモートUNIXプリンタの両方の論理プリンタ名を対応づける必要があります。ユーザは、この論理プリンタが印刷要求を送信するときにLPDプロトコルを使用するのか、Vision USP RPCプロトコルを使用するのかを選択できます。

論理プリンタを構成するときを選択する Windows のプリンタは、実際のプリンタポートに出力する代わりに、ファイルに出力する必要があります。このファイルは PC 上の Vision サービスが監視しており、その印刷要求は UNIX のプリントスプーラに送信されます。Windows アプリケーションからリモートプリンタに印刷するときは、適切な Windows プリンタを選択するだけです。

RPC プロトコルを使用して UNIX システムに送信される印刷要求は、**printsrv** プログラムで処理されます。UNIX 側には特別な構成は必要ありません。

リモートプリンタの追加

リモートの UNIX プリンタの構成作業は、次の 3 つの部分からなります。

- 1 UNIX ホスト上で、PC から印刷要求を受け取る予定のプリンタをセットアップします。たとえば、そのプリンタは **room123** という名前になります。
- 2 PC 上のファイルに出力する新しい Windows プリンタを追加します。
- 3 Vision サービスでリモートプリンタを構成します。

リモートプリンタを構成しているのであれば、ローカルプリンタを選択するときと同じように、Windows アプリケーションからそのプリンタを選択して印刷できます。

Windows 2000/Me/98/95 または Windows NT 4.0 でプリンタを追加するには

- 1 PC で [スタート] ボタンをクリックしてから [設定] をポイントして [プリンタ] をクリックします。
- 2 [プリンタの追加] アイコンをダブルクリックします。



[プリンタ] ウィザードが起動するので、新しいプリンタをセットアップする手順に従ってください。[ローカルプリンタ] を選択し、UNIX システムに接続されているプリンタのタイプを選択します。ポートの選択を質問されたときには、リモート印刷には既存ポートはどれも適していないので、一時的に LPT1: を選択します。プリンタには QMS-PS810 on UNIX のように分かりやすい名前をつけます。テストページの印刷は選択しないでください。

ウィザードが終了したら、新しいプリンタが [プリンタ] ウィンドウに表示されます。



- 3 マウスの右ボタンで新しいプリンタアイコンをクリックしてから [プロパティ] をクリックします。
- 4 [詳細] タブで [ポートの追加] をクリックします。
- 5 [その他] をクリックし、リストボックスにある [ローカルポート] を選択します。[OK] をクリックします。
- 6 新しいポート名としてファイル名を `c:\tmp\unixprt.tmp` のように入力します。[OK] をクリックします。



- 7 新しいポートは [印刷先のポート] ボックスに表示されます。[OK] をクリックして [プリンタのプロパティ] ダイアログボックスを終了します。

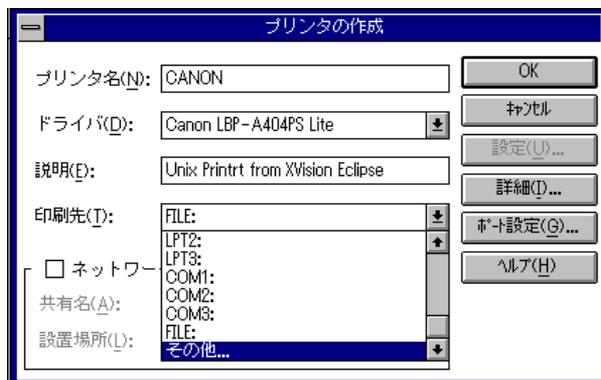
これで、Vision サービスにリモートプリンタを追加する手順に進むことができます。この手順はこの章で後述します。

Windows NT 3.51 でプリンタを追加するには

- 1 Windows のコントロールパネルで「プリンタ」をダブルクリックします。
- 2 プリントマネージャの [プリンタ] メニューから [プリンタの作成] を選択します。



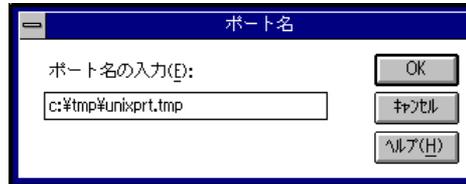
- 3 [プリンタの作成] ダイアログボックスにプリント名、ドライバを正しく設定し、印刷先の [その他] を選択します。



- 4 [印刷先] ダイアログボックスの利用可能な印刷モニタの中から [Local Port] を選択します。



- 5 [ポート名] ダイアログボックスに[c:\tmp\unixprt.tmp]などのファイル名をポートとして追加します。



- 6 これらの設定でWindowsに印刷ポートが作成されます。
これで、Vision サービスにリモートプリンタを追加する手順に進むことができます。

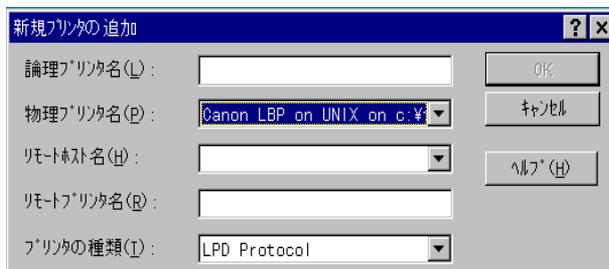
Vision サービスにリモートプリンタを構成するには

- 1 Windows のコントロールパネルで [Vision サービス] をダブルクリックしてから [印刷] タブをクリックします。



現在構成されているローカルとリモートのプリンタが [プリンタ] リストに表示されます。

- 2 [リモートプリンタ追加] をクリックします。



- 3 [論理プリンタ名] ボックスに新しい名前を入力します。
- 4 [物理プリンタ名] ボックスに今設定したプリンタを選択します (例 :ÅgQMS-PS810 on UNIX on c:¥tmp¥unixprt.tmpÅ)。)
- 5 リモートホストとリモートプリンタ (例 : room123) の名前を指定します。
- 6 PC から選択した UNIX ホストに印刷データが送信される通信プロトコル (LPD または RPC) を指定します。UNIX ホストが受信した LPD リモート印刷要求を処理し、PC がその UNIX ホストへの TCP/IP 接続を確立する場合には、[LPD Protocol] を選択します。それ以外の場合には [RPC Protocol] を選択します。

注意 LPD プロトコルを使用している場合には、ホスト上にある `/etc/hosts.equiv` ファイルまたは `/etc/hosts.lpd` ファイルに PC のネットワーク名を追加する必要があります (ホストシステムの LPD に関するマニュアルを参照してください)。

第 15 章

メッセージ処理

このセクションの内容は、次のとおりです（この章の内容はサポート対象外です）。

- メッセージ処理の概要
- メッセージパッド
- テキストメッセージ
- マルチメディアメッセージ
- UNIX メッセージ

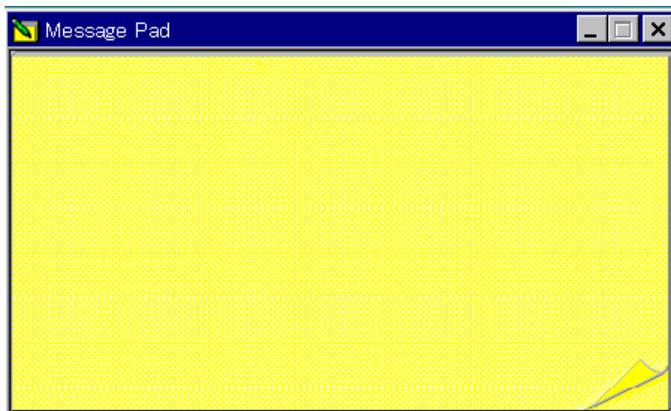
メッセージ処理の概要

X Vision ソフトウェアには、PC ユーザに送信されたメッセージを処理するためのいくつかの機能があり、次のような利点があります。

- 使用している PC から端末接続を開くと、UNIX から送信されたメッセージが端末エミュレータ画面ではなく、[メッセージパッド] ウィンドウに表示されます。
- ユーザビューワプログラムを使用すると、他の PC ユーザにメッセージを送信できます。これらのメッセージは宛先 PC の [メッセージパッド] ウィンドウに表示されます。
- サウンドやビデオクリップなどのマルチメディアメッセージを送受信できます。

メッセージパッド

メッセージパッドは、PC 上で実行する RPC サーバプログラムの 1 つです。Post-it メモに似たウィンドウにユーザに送信されたメッセージが表示されます。



PC が USP トランスポートを使って UNIX サーバに接続されている場合は、UNIX からユーザに送信されるメッセージは [メッセージパッド] ウィンドウに表示されます。ユーザビューワプログラムによってテキストメッセージとマルチメディアメッセージも送信できます。(本章の「テキストメッセージと「マルチメディアメッセージ」を参照してください。)

何を実行できるかを示すショートカットメニューを見るには、右マウスボタンでウィンドウ内をクリックします。詳しい説明については、[ヘルプ] メニューの [トピックの検索] をクリックします。メインメニューの [Vision ヘルプの目次] から [メッセージパッドの使い方] を選んでも、同じ情報が表示されます。

メッセージパッドを起動するには

メッセージパッドはRPC サーバプログラムなので、ユーザに初めてメッセージを送信するときにローカルネームエージェントによって自動的に起動されます。

メッセージを受信した場合

[メッセージパッド] ウィンドウをアイコン化しているときにメッセージを受信すると、サウンドが鳴ります。Windows 2000/Me/98/95 や Windows NT 4.0 の場合は、タスクバーの右側にある通知領域に小型のアイコンが表示されます。Windows のこれ以外のバージョンの場合は、[メッセージパッド] アイコンが点滅します。



[メッセージパッド] を表示するには

Windows 2000/Me/98/95 や Windows NT 4.0 の場合は、通知領域のアイコンをクリックします。

Windows のこれ以前のバージョンの場合は、アイコンをダブルクリックしてウィンドウを復元します。

[メッセージパッド] ウィンドウが表示されます。複数のメッセージを受信した場合は、2 の 2 のような注が表示され、ページの右下の i 背景 i 部分をクリックすれば、メッセージを順に表示できます。あるいは、右マウスボタンでメインウィンドウをクリックし、[次のメッセージ] または [前のメッセージ] を選びます。

マルチメディアメッセージを受信すると、ページ上にアイコンとして表示され、該当するビュープログラムが開きます。ビューを閉じて、メッセージは [メッセージパッド] ウィンドウに残ります。ビューを実行してもう一度メッセージを表示するには、メッセージアイコンをダブルクリックします。[メッセージパッド] ウィンドウを閉じると、すべてのメッセージが削除されます。

[メッセージパッド] からメッセージを保存する

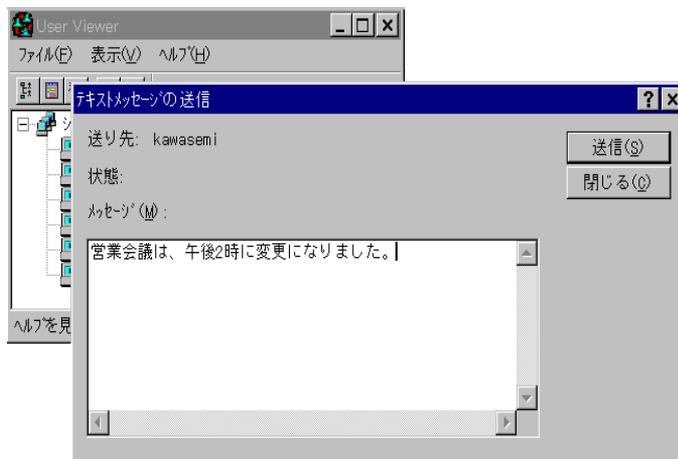
テキストをマウスで選択し、CTRL+INS を押せば、[メッセージパッド] からテキストをコピーできます。テキストを Windows ワードパッドやメモ帳にペーストして、ドキュメントとして保存することもできます。テキスト以外の形式のメッセージを保存するには、ビューワプログラムでメッセージを開き、[ファイル] メニューの [上書き保存] コマンドを使ってファイルに名前を付けて保存します。

テキストメッセージ

ユーザビューワプログラムを使用してテキストメッセージを別の Vision ユーザに送信できます。送信されたメッセージは宛先 PC の [メッセージパッド] ウィンドウに表示されます。

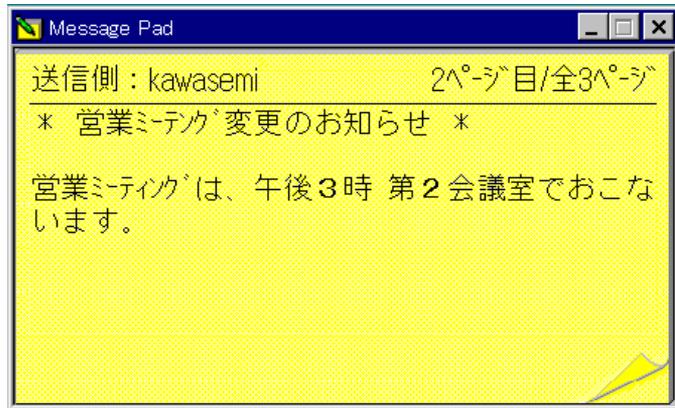
テキストメッセージを送信するには

- 1 [ユーザビューワ] ウィンドウで、メッセージの送信先のユーザの名前をクリックします。
- 2 [ファイル] メニューの [テキストメッセージの送信] をクリックします。
- 3 スペース内にメッセージを入力します。



- 4 [送信] をクリックします。

ユーザビューワプログラムがリモートユーザに接続し、メッセージがリモートユーザのコンピュータの [メッセージパッド] に表示されます。



マルチメディアメッセージ

ユーザビューワプログラムを使って次のようなマルチメディアメッセージを別のユーザに送信できます。送信されたメッセージはメッセージパッドによって認識されます。

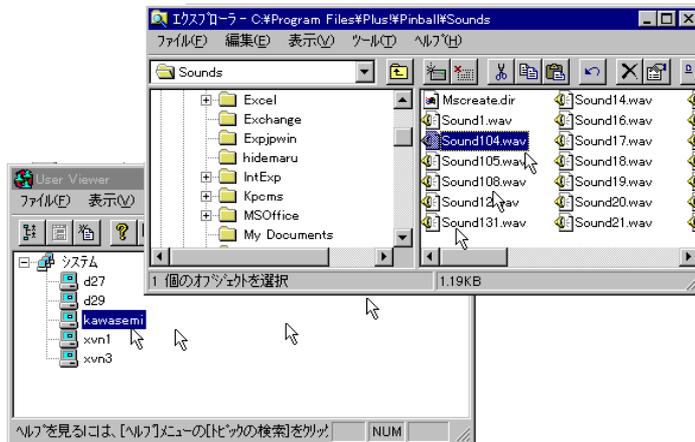
- テキストファイル (*.txt)
- Word ドキュメント (*.doc)
- アニメーションファイル (*.avi)
- ビットマップファイル (*.bmp)
- MIDIシーケンスファイル (*.mid)
- サウンドファイル (*.wav)

ユーザビューワがRPCメカニズムを使用してファイルをメッセージパッドに送信すると、メッセージパッドはファイルをWindows **Temp** フォルダに書き込んで、該当するビューワプログラムを起動します。[メッセージパッド]を閉じると、一時ファイルは削除されます。

メッセージパッドによって認識されるのは上記のファイルタイプだけで、これ以外のファイルタイプは追加できません。これらのファイルタイプは、通常は、WindowsをインストールするときにWindowsレジストリに登録されます。ただし、ファイルタイプを登録する必要がある場合は、Windowsのエクスプローラがファイルマネージャを使用します。

マルチメディアメッセージを送信するには

- 1 [エクスプローラ] (Windows 2000/Me/98/95 または Windows NT 4.0)か [ファイルマネージャ] (Windows NT 3.51) で、メッセージとして送信したいファイルをクリックします。
- 2 [ユーザビューワ] ウィンドウが表示されていることを確認します。
- 3 [ユーザビューワ] ウィンドウで目的のPC名までファイルをドラッグします。



- 4 [設定] ダイアログボックスの [マルチメディアメッセージの送信を確認する] を使用可能にした場合は、アクションを確認してから処理を続けます。

PC 上で該当するプログラムが起動され、ファイルが自動的に開きます。メッセージパッドも起動され、開かれているファイルの名前が表示されます。サウンドメッセージやビデオメッセージの場合は、メッセージを正常に受信したら、[再生] をクリックしてファイルを実行する必要があります。次の例は、サウンドメッセージ0を示しています。



これ以外のメッセージタイプの例については、オンラインヘルプの目次から [X Vision の紹介] を選び、次に [メッセージ送信] を選びます。

UNIX メッセージ

UNIX システムにログインすると、**write** や **wall** などのプログラムを使ってユーザ間でメッセージを送信できます。電子メールプログラムによって送信された“新規メール”メッセージや各種のシステムメッセージを受信することもあるでしょう。実際の文字ベース端末を使用している場合は、このようなメッセージが端末画面に表示されます。

X Vision 端末エミュレータを使用するとき、メッセージの処理方法は、端末サービスを提供するのに使用された方法によって決まります。たとえば、標準 UNIX **rlogin** サービスを使用するように設定されているトランスポートを使用している場合は、メッセージは実際の端末と同じように端末エミュレータ画面に表示されます。**rlogin** プログラムは、どの端末装置にログインしたのかを示すエントリを UNIX **utmp** ファイルに作成します。ユーザが指定するユーザ名にメッセージを送信しようとするプログラムは、**utmp** ファイルを見て、どの端末装置に書き込んだらよいかを調べます。

USP トランスポートを使用している場合は、端末接続要求は USP タスクサーバ **tsrv** によって処理されます。デフォルトでは、タスクサーバはログインした端末のエントリを **utmp** に作成せず、代わりに、別の擬似 tty デバイスに関連付けられているエントリを **utmp** に作成します。タスクサーバは、**utmp** にエントリが作成されているユーザに送信されたメッセージを受け取ります。メッセージが受信されると、タスクサーバは RPC クライアントプログラムとして動作して、正しい PC 上のメッセージパッド RPC サーバにメッセージを渡します。

必要であれば、タスクサーバにこのようにメッセージを処理させないこともできます。この場合は、**utmp** エントリを **rlogin** と同じように作成します。詳細については、「端末エミュレーション」の章の「**utmp** ファイル」を参照してください。

(空白のページです)

第 16 章

ユーザ情報の表示

このセクションの内容は、次のとおりです（この章の内容はサポート対象外です）。

- ユーザ情報表示の概要
- ユーザ情報の表示
- ユーザプロパティの表示

ユーザ情報表示の概要

ユーザビューワプログラムを使うと、X Vision ソフトウェアを実行するすべての PC ユーザのリストを得られます。ユーザビューワプログラムは RPC コールを介してネームスペースマネージャ (NSM) とやりとりし、ローカルネームエージェント (LNA) を登録したすべての PC のリストを得ます。

SuperVision を使用中の場合は、ユーザビューワはユーザグループ情報も表示します。plist プログラムが管理ホスト上で実行されていた場合は、この情報は NSM によっても保持されています。SuperVision は、セッションの開始時に管理ホスト上で LNA によって実行されるのが普通です。

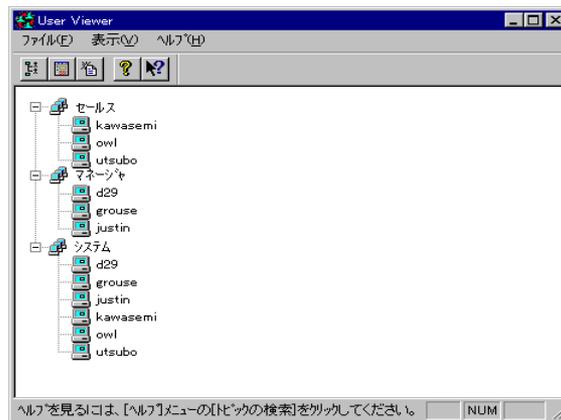
「メッセージ処理」の章で説明したように、ユーザビューワプログラムを使用して、テキストメッセージやマルチメディアメッセージをユーザに送信できます。

ユーザ情報の表示

ユーザビューワプログラムを使用して、現在どの PC が X Vision ソフトウェアを実行中かを調べられます。

現在のユーザを表示するには

- 1 [スタート] ボタンをクリックし、[プログラム]、[CentreNET X Vision]、[アクセサリ]の順に選択します。[ユーザビューワ] をクリックします。
[ユーザビューワ] が画面に表示されます。



各ユーザは、階層ツリーの中にグループ別にアイコン表示されます。[システム] グループにすべてのユーザが含まれています。SuperVision を使用中の場合は、表示されている追加グループは管理されているファイルストアに定義されています。[SuperVision] グループのユーザは [システム] グループにも表示されていることに注意してください。

- 2 詳細については、右マウスボタンでユーザをクリックし、[プロパティ] をクリックします。

詳細については、ヘルプメニューの [トピックの検索] をクリックします。メインメニューの [Vision ヘルプの目次] から [ユーザビューワの使い方] を選んでも、同じ情報が表示されます。次についてのヘルプが表示されます。

- ウィンドウの見た目の変更方法
- 現在のユーザの表示方法
- メッセージの送信方法

ユーザプロパティの表示

[ユーザビューワ] の中で右マウスボタンでユーザのアイコンをクリックし、[プロパティ] をクリックすると、[オプション] ダイアログボックスが表示され、ユーザについての個人情報が示されます。



使用している PC でこの操作を実行すると、ユーザビューワは Windows レジストリから現在の情報を読み取ります。設定を変更し、変更内容を保存できます。

別のユーザのプロパティを要求すると、ユーザビューワはネームスペースマネージャ (NSM) とやりとりし、宛先 PC 上で **infosrv** サービスを要求します。NSM が情報サーバ **Infosrv.exe** を起動するように宛先 PC のローカルネームエージェント (LNA) に要請すると、使用している PC 上のユーザビューワにユーザプロパティが提供されます。別のユーザのプロパティは編集できません。

第 17 章

Zone によるデスクトップの並べ替え

この節では、次の内容について解説します。

- Zone の概要
- ゾーンの並べ替え
- Zone プログラムの終了または Zone の削除
- Zone コントロールの使い方
- ゾーンプロパティの表示
- プログラムの自動起動

Zone プログラムの概要

Zone プログラムを使うと、仮想デスクトップ領域を作成することによってウィンドウを複数のゾーンに並べ替えることができます。この方法には、次のような利点があります。

- 作業するすべてのアプリケーションで画面領域を共有する代わりに、アプリケーションを「ゾーン」に並べ替えれば、ゾーンバーを使って簡単に切り替えることができます。
- アプリケーションのウィンドウを 1 つのゾーン、複数のゾーン、またはすべてのゾーンに表示できます。
- 使用可能なプログラムやドキュメントを表わすアイコンなどのデスクトップバックグラウンドがすべてのゾーンに表示されます。
- Windows のタスクバーがすべてのゾーンに表示されます。タスクバーボタンは、現在のゾーンにあるウィンドウを表わしています。通知領域の項目はすべてのゾーンで同じです。
- 各ゾーンには [スタートアップ] フォルダがあり、これを使ってアプリケーションを自動的に起動できます。
- Windows 2000/Me/98/95 と Windows NT 4.0 以降で使用できます。

注意 本書では、プライマリゾーンを Zone 1 として表わしています。プライマリゾーンの名前は変更されることがあるので注意してください。

Zone の並べ替え

Zone プログラムを使用して、現在実行中のアプリケーションをいくつかのグループ、すなわちゾーンに並べ替えておけば、簡単に切り替えることができます。

Zone を使用するには

- 1 [スタート] ボタンをクリックし、[プログラム]、[CentreNET X Vision]、[アクセサリ] の順に選択して、[Zone] をクリックします。

ゾーンバーが普通は画面の一番上に表示されます。

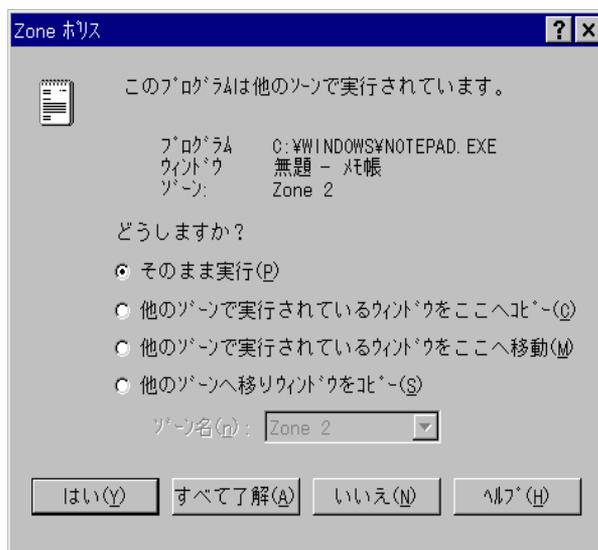


左側の [Zone] ボタンを押すと、メニューが表示されます。これ以外のボタンは、別のゾーンを表わしています。すでに Zone プログラムをカスタマイズ済みの場合は、ゾーンの名前を変更しており、ゾーンバーが画面上の別の場所に表示されたり、表示されていないかもしれません。

Zone を起動したときに実行しているすべてのプログラムは Zone 1 に入ります。初めに現在のゾーンになるゾーンは、[Zone プロパティ] ダイアログボックスの設定によって決まります。デフォルトは Zone 1 ですが、別のゾーンや最後に使用したゾーンを指定することもできます。

- 2 別のゾーンに切り替えたい場合は、ゾーンボタンをクリックします。
- 3 新しいゾーンに表示したいアプリケーションを起動します。

別のゾーンですでに実行しているアプリケーションを起動しようとする、普通は、[Zone ポリス] ダイアログボックスにプログラムの新しいインスタンスを起動するか、代替アクションをとるかを尋ねるメッセージが表示されます。アプリケーションで複数のインスタンスが許容されない場合、別のアプリケーションを起動しようとして生じる結果はアプリケーションによって異なります。たとえば、ウィンドウが変更後のゾーンのフォアグラウンドになったり (現在のゾーンに影響はありません)、ウィンドウが現在のゾーンに移動したりします。



[Zone ポリス] は、[Zone プロパティ] ダイアログボックスの [メイン] タブから使用不可にできます。

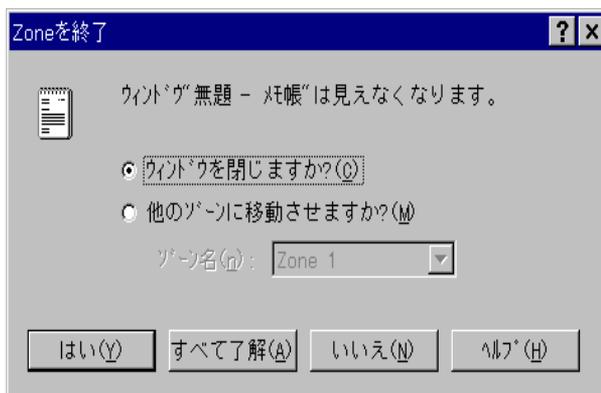
詳細については、[Zone] メニューの [トピックの検索] をクリックしてください。メインメニューの [Vision ヘルプの目次] から [Zone の使い方] を選んでも同じ情報が表示されます。次の内容のヘルプが表示されます。

- ゾーンの作成、削除、名前変更
- ゾーンのホットキーの指定

- ウィンドウのコピー、移動、削除
- [スタートアップ]フォルダでのショートカットの追加と削除
- ゾーンバーの非表示、表示、移動

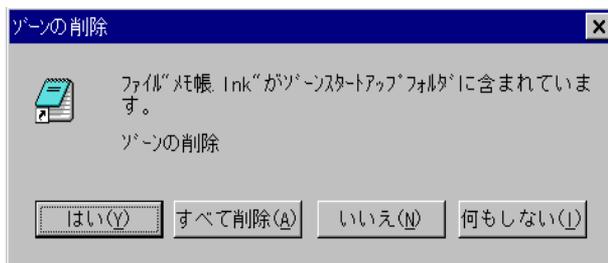
Zone プログラムの終了または Zone の削除

Zone プログラムを終了すると、ウィンドウが Zone 1 に表示されているアプリケーションしか実行を続けられません。したがって、Zone 1 以外のゾーンで実行しているアプリケーションは終了するか、Zone 1 に移動しなければなりません。



ウィンドウが当該ゾーンだけに表示されるアプリケーションを含むゾーンを削除しても、似たようなダイアログボックスが表示されます。また、ゾーンコントロールを使ってゾーンからウィンドウを削除し、アプリケーションのウィンドウが他のゾーンに表示されなくなった場合も、ウィンドウを閉じてもいいかの確認を求める似たようなダイアログボックスが表示されます。

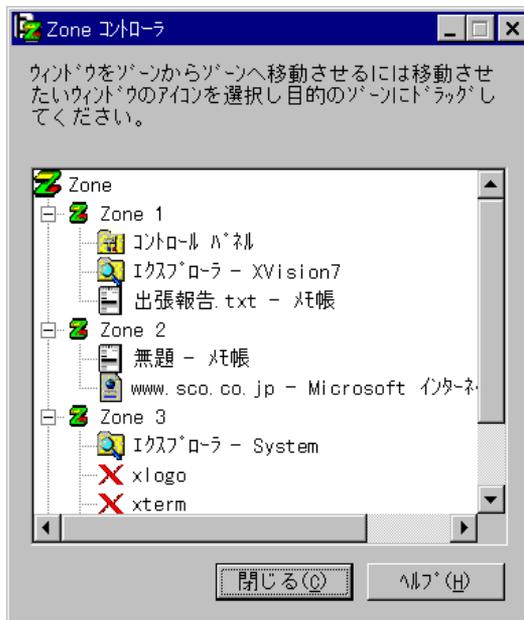
ショートカットを含む [スタートアップ] フォルダのあるゾーンを削除すると、次のダイアログボックスが表示されます。



すべてのショートカットを削除すると選択しても、[スタートアップ] フォルダしか削除されません。

Zone コントロールの使い方

[Zone コントローラ] ウィンドウを使うと、ゾーンを作成、削除、名前変更できます。単一ゾーン、またはすべてのゾーンの属性も変更できます。現在実行中のアプリケーションをゾーン間で移動したり、ゾーンごとに自動的に起動するアプリケーションを指定したりできます。



[Zone コントローラ] ウィンドウを表示するには

- [Zone] メニューで [コントローラ] をクリックします。

マウスを使用して、ゾーン間でウィンドウのアイコンをドラッグします。右マウスボタンを使えば、ウィンドウをコピーすることもできるので、アプリケーションのウィンドウを複数のゾーンに表示させることもできます。この場合、アプリケーションの1つのインスタンスだけが実行されているので、アプリケーションを閉じると、そのウィンドウはすべてのゾーンから削除されます。ゾーンやウィンドウでの操作方法の詳細については、ウィンドウの [ヘルプ] ボタンをクリックし、[ヘルプ] ウィンドウの [トピックの検索] ボタンを選びます。

Zone プロパティの表示

Zone バーの Zone のアイコンを右マウスボタンでクリックし、[プロパティ] をクリックすると、[プロパティ] ダイアログボックスが表示されます。



キーボードショートカットを使って、Zone の名前や、その Zone に速やかに切り替えるためのホットキーを変更できます。また、Zone のアイコンも変更できます。[スタートアップ] タブについては、「プログラムの自動起動を参照してください」。

Zone トップレベルの属性を表示すると、Zone バーのオプションを指定したり、[スタートアップ] プログラムを指定したりもできます。

プログラムの自動起動

各ゾーンには[スタートアップ] フォルダがあり、その中にプログラムを自動的に起動するためのショートカットを収めることができます。また、[スタートアップ] フォルダがトップレベルにあり、すべてのゾーンに表示されるプログラムを起動できます。

すべてのゾーンでプログラムを自動的に起動する

ゾーントップレベルで[スタートアップ] フォルダへのショートカットを追加すると、Zone プログラムが起動するとショートカットを作成したプログラムが起動され、プログラムのウィンドウがすべてのゾーンに表示されます。

すべての Zone でプログラムを自動的に起動するには

- 1 [Zone] メニューの [プロパティ] をクリックし、次に [スタートアップ] タブをクリックします。
- 2 [追加] をクリックします。画面の指示に従ってショートカットを作成します。

プログラムのウィンドウは複数の Zone に表示されることがありますが、プログラムの 1 つのインスタンスしか実行されていません。アプリケーションを閉じると、すべてのゾーンからそのウィンドウが削除されます。

プログラムを起動するかどうかの確認を求めるプロンプトを表示したくない場合は、[Zone プロパティ] ダイアログボックスの [メイン] タブの [プログラムを自動起動するか確認する] を無効にします。

1 つのゾーンでプログラムを自動的に起動する

アプリケーションを特定のゾーンに表示したい場合は、その Zone の [スタートアップ] フォルダにショートカットを挿入します。初期 Zone の [スタートアップ] フォルダにショートカットがあると、Zone プログラムを起動すると、ショートカットのあるプログラムが起動されます。初期 Zone 以外の Zone の場合は、初めて当該 Zone に切り替えたときに [スタートアップ] フォルダの中のプログラムが起動されます。

1 つの Zone でプログラムを自動的に起動するには

- 1 Zone バーで、右マウスボタンで Zone ボタンをクリックします。次に、[プロパティ] をクリックしてから、[スタートアップ] タブをクリックします。
- 2 [追加] をクリックします。画面の指示に従ってショートカットを作成します。

プログラムを起動するかどうかの確認を求めるプロンプトを表示したくない場合は、[Zone プロパティ] ダイアログボックスの [メイン] タブの [プログラムを自動起動するか確認する] を無効にします。

(空白のページです)

付録 A

用語集

ANSI

米国規格協会。コンピュータ業界の標準を確立する民間団体。

API

アプリケーションプログラムインタフェース。プログラムは、標準関数を呼び出すことによってソフトウェアルーチンを利用できます。

ASCII コード

American Standard Code for Information Interchange。256文字を表わす標準8ビットコード。最初の128文字の意味はANSIによって定義されており、残りの文字は個々のコンピュータメーカーやユーザが定義や解釈用に使用できます。

BDFフォント

ビットマップ配布形式。

COLA (Certificate of License Authority)

ライセンス権限の認証。

ComScript

ホストへの接続、ログイン、コマンドの実行のプロセス全体またはその一部を制御するスクリプトファイルを作成するためのVision通信スクリプト言語。

DBMS

データベース管理システムの省略形。データベースの中に保管されているデータへのアクセスを制御するソフトウェア。

DECnet

DECの専用通信ネットワーク。

DECwindows

X Window SystemのDEC版。DECwindowsはVMSシステムとULTRIXシステムの両方で使用できます。

HINTS

Helpful Interactive Support (HINTS) メカニズムは、クライアントを起動したものの正しいフォントが見つからないなどのエラーが生じてクライアントが終了するときに役立ちます。

ICCCM

X Consortium 発行のクライアント間通信規則マニュアル (Inter-Client Communications Conventions Manual)。クライアントとウィンドウマネージャの対話方式を定義します。

IP アドレス

インターネットプロトコルアドレス。ネットワーク上の特定 TCP/IP ホストの一意の 32 ビット識別子。IP アドレスは、150.126.27.92 のように、0 ~ 254 の 4 つの 10 進数で指定され、4 つの数字はピリオドで区切ります。

IPX

Novell のインターネットワークパケット交換プロトコル。

ISO

国際標準化機構。性能を規定する標準を調整し発行する国連の機関。

keysyms

キーイベントの意味に対応する定義済み定数。

LAN

ローカルエリアネットワーク。高性能ケーブルによって一定範囲内の PC などのコンピュータがリンクされているため、ユーザ間で情報を交換したり、周辺装置を共有したり、二次記憶装置の資源に描画したりできます。

LPD プロトコル

ラインプリンタデーモンプロトコル。LPD プロトコルは、リモート印刷のための標準 TCP/IP サービスを提供します。

NCP

ネットワーク制御プログラム。ネットワークノードを検索、管理、設定するのに使用される DECnet ユーティリティ。

NDIS

ネットワークドライバインタフェース仕様。Microsoft と 3Com によって開発された仕様で、メディアアクセス制御ドライバとトランスポートドライバのための共通プログラミングインタフェースを提供します。

NetBEUI

NetBIOS 拡張ユーザインタフェース。Windows 2000/Me/98/95、Windows NT で使用される基本ローカルエリアネットワークトランスポートプロトコルです。

NetBIOS

ネットワーク基本入出力システム。PC 上で動作するネットワークアプリケーションと元になるネットワークプロトコル間のインタフェースを提供します。

ODBC

Microsoft によって開発されたオープンデータベースコネクティビティ (ODBC) は、Windows プログラムからデータベース管理システムにアクセスするための標準方式です。Microsoft Windows Open Services Architecture (WOSA) の構成要素の1つです。

ODBC ドライバ

ODBC ドライバは、ドライバマネージャが特定のデータベースにアクセスできるようにするソフトウェアモジュールです。ドライバは単一階層ドライバまたは複数階層ドライバのどちらでもかまいません。単一階層ドライバは、ODBC コールとSQL 文を処理するのに対して、複数階層ドライバは、ODBC 文を処理して、SQL 文をデータソースに渡します。

OLE

オブジェクトのリンクと埋め込み。OLE は、Windows とアプリケーション間で情報を転送したり共有したりするための基本的技法です。

OLE オートメーション

OLE オートメーションを使うと、プログラムは一連の操作を定義し、それを他のアプリケーションからアクセス可能にできます。このとき、端末エミュレータを別の Windows アプリケーションの中でプログラマブルオブジェクトとして使用できます。

OPEN LOOK

OPEN LOOK ユーザインタフェースはソフトウェアではなく、マルチタスクコンピュータシステム向けのウィンドウ環境の見た目と使い心地についての仕様です。OPEN LOOK UI は、Sun Soft Inc. の OpenWindows や AT&T OPEN LOOK グラフィカルユーザインタフェースなど、多数の企業によって広範囲のハードウェアやオペレーティングシステム上に実装されています。

OSF/Motif

OSF/Motif は X Window System を実装しています。OSF/Motif GUI は Open Software Foundation によって開発され、多数の企業で採用されています。OSF/Motif の基本機能は一定ですが、販売会社によってさまざまな拡張機能が実装されています。

PCF フォント

ポータブルコンパイル形式。

PostScript

中程度解像度から高解像度の印刷装置向けの Adobe Systems, Inc. によって開発された上級ページ記述言語。ほとんどの Windows アプリケーションは、PostScript 出力を生成し、PostScript インタープリタを使ってその出力をプリンタに印刷できます。印刷可能なプリンタには、標準的なオフィス用レーザープリンタから、Linotronic 製の高価なタイプセットマシンに至る広範囲のマシンが含まれます。

PPP

ポイントツーポイントプロトコル。非同期通信向けの業界標準プロトコルで、サードパーティ製のリモートアクセスソフトウェアとの相互操作性を得られるようにします。PPP は SLIP の後継プロトコルです。

QWERTY キーボード

コンピュータキーボードにも使用されている標準タイプライタキーボードレイアウト。QWERTY キーボードでは、数値キーのすぐ下の列の左端から数えて6つのキーに QWERTY が採用されています。

Remote Sockets サービス

Vision コミュニケーションアーキテクチャによって提供される USP サービス。クライアントプログラムがサーバプログラムでプロシージャを呼び出すための低レベル手法です。

reuseaddr

Windows Sockets オプション。ソケットをすでに使用中のアドレスにバインドできます。

RS232C

RS232C は、コンピュータとモデムなどのデータ通信装置との間のシリアルバイナリデータ交換のための業界標準です。

SLIP

シリアル回線インターネットプロトコル。非同期通信の業界標準プロトコルで、サードパーティ製のリモートアクセスソフトウェアとの相互操作性が得られません。

SPX

Novell NetWare プロトコル。コネクションレスパケット伝送を行います。

SQLGold

SQLGold は、Visual Basic を使って作成された Windows プログラムの 1 つです。どの DBMS からでも独立した SQL データベースへの Windows フロントエンドになります。

SQL-Retrieve

SQL-Retrieve は、ソフトウェア構成要素の集まりで、Microsoft Windows アプリケーションから SQL データベースにアクセスするのに必要なすべてのコネクティビティを提供します。パッケージには、ODBC ドライバ、通信およびネットワークソフトウェア、SQL-Exchange プログラムが含まれており、非 ODBC アプリケーションからデータベースにアクセスできるようになっています。

SuperVision

X Vision ソフトウェア 1 つで、これを使うと、システム管理者は大きな Windows PC グループをリモート管理、設定、制御できます。

TAPI

Microsoft Windows Telephony Application Programming Interface (TAPI)。各種の電話網サービスにアクセスできます。Windows アプリケーションは、TAPI を使って、サービス固有のプログラミングコードを指定しなくても、音声およびデータリンクを制御できます。Windows 2000/Me/98/95 と Windows NT では、TAPI はモデムや ISDN などのデジタル電話サービスへのインタフェースを提供する標準方式になっています。

TCP

伝送制御プロトコルは、TCP/IP プロトコル群の中の主要部分です。TCP は、ローカルホストとリモートホストとの間で接続指向の信頼性の高い順次データ転送を行います。

TCP/IP

伝送制御プロトコル/インターネットプロトコル。TCP/IP は、大学、研究機関、各種団体、企業、軍事施設などから成る世界規模のネットワークを接続するのに使用されるプロトコル群です。TCP/IP には、コンピュータ間の通信標準と、ネットワークとルーティングトラフィックを接続する取り決めが含まれています。

Unix ネットワークコンピュータ

Unix ネットワークコンピュータを使うと、Windows 2000/Me/98/95 や Windows NT 4.0 ユーザは、Windows のネットワークコンピュータに似た画面を使ってネットワーク上のホストにアクセスできます。ユーザは、リモートホストコンピュータを設定したり、フォルダやファイルを参照したり、プログラムやドキュメントを開いたりできます。ファイルやフォルダ作成または削除すると同時に、リモートコンピュータ上での移動やコピー、または PC へのコピーができます。

USP サービス

USP サービスは Vision コミュニケーションアーキテクチャによって提供される一連のサービスで、Vision プログラムやサードパーティアプリケーションによって使用されます。USP サービスは豊富な機能を持ち、どのトランスポートやホストオペレーティングシステムからも独立しています。USP サービスはユニバーサルセッションプロトコル (USP) によって実行され、Vision サービスモジュールによって PC 上に実装されています。Vision Toolkit API を介して各種プログラムで使用できます。

Vision Resume

Vision Resume 機能を使うと、X セッションを延期して、後で X セッションを再開できます。

Vision ウィンドウマネージャ

Vision ウィンドウマネージャ (VWM) は X サーバに組み込まれており、サーバがマルチウィンドウモードで使用されているときに使用可能です。VWM は、X クライアントをローカルウィンドウ管理することができ、OSF/Motif と互換性があります。

Vision サービス

Vision サービスは、バックグラウンドで実行されるすべての USP サービスに付けられた汎用名です。Vision サービスプログラムである **Vservice.exe** によって起動したりシャットダウンしたりできます。

Vision コミュニケーションアーキテクチャ

Vision プロダクトと元になる通信トランスポートをリンクするソフトウェアモジュールの集まり。Vision コミュニケーションソフトウェアは、これを使用するアプリケーションから独立した別個の階層を形成しています。

Vision コミュニケーションデータベース

Vision コミュニケーションデータベースは、Vision コミュニケーションアーキテクチャの中の構成可能なすべての要素の詳細が保存されているファイルの集まりです。これらのファイルは、それぞれの Vision ソフトウェア PC上の **C:\Program files\Common files\Vision** フォルダの中のサブフォルダに保存されています。

VMS

マルチユーザ、マルチタスクのオペレーティングシステム。DEC の VAX コンピュータで動作します。

VT420 エミュレータ

VT420 エミュレータを使うと、文字ベースアプリケーションをリモートコンピュータで実行し、それを使用している PC 上のウィンドウに表示することができます。VT420 エミュレータはDECのVT420 端末にもとづいています。DEC ビデオ端末向けに設計されたアプリケーションは、このエミュレーション方式を使って、PC の機能を利用できます。

WAN

広域ネットワーク。ローカルエリアネットワークより広い範囲の領域でデータ通信機能を提供します。

X キーマップジェネレータ

X サーバで使用するキーボードマップを作成したり変更したりできます。

X クライアント

X Window System モデルでは、クライアントプログラムは普通はアプリケーションです。ウィンドウマネージャもクライアントです。

X サーバ

X モデルでは、サーバプログラムがディスプレイハードウェアを制御し、ディスプレイハードウェアとクライアント間のインタフェースを提供します。

X ディスプレイ

X と対話するためのユーザ向けハードウェア。X ディスプレイは、1 つまたは複数のビットマップ画面、1 台のキーボード、マウスなどのポインティングデバイスから構成されます。

xdm

X ディスプレイマネージャ。X11R5 で指定されるクライアントで、ユーザが UNIX システムにログインして X セッションを開始するときの標準的な方法です。xdm は XDMCP と関連付けて使用されます。

XDMCP

X ディスプレイ管理制御プロトコル。サーバとディスプレイハードウェアとの間の対話を定義します。

XRemote

Network Computing Devices Inc. (NCD) によって開発されたシリアル回線プロトコル。X Window System を非同期シリアル接続を介して実行できます。X Vision X サーバは XRemote をサポートしています。

xterm

端末ウィンドウがホストオペレーティングシステムにアクセスできるようにする標準 X クライアント。X Window System によっては、このクライアントに他の名前を使用することもあります。たとえば、DECwindows では DECterm が使用されます。

X Vision フォントコンパイラ

PCF または BDF 形式の X フォントを Microsoft Windows の .FON 形式フォントファイルにコンパイルします。.FON 形式フォントファイルは、X クライアントを PC 上に表示するときで使用できます。

X Vision

X Window System アプリケーションを Microsoft Windows プログラムと同時に PC 画面に表示できるようにする X Vision メンバ。

X 論理フォント

フォントの特性を記述するフォント記述。記述の各要素はハイフンで区切りません。たとえば、次の記述は Courier 10pt フォントを指定しています。

```
-Adobe-Courier-Medium-R-Normal-10-100-75-75-M-60-ISO8859-1
```

要素の中では、アスタリスク文字 (*) をワイルドカードとして使用できます。たとえば、次の記述も Courier 10pt を指定しています。

```
*-Courier-*-R-*-100-*
```

アドミンアップデート

アドミンアップデートは管理ホストとやりとりして、変更済みのファイルなどのオブジェクトの更新を要求します。Admupdat.exe と呼ばれる PC プログラムです。

アドミンサーバ

アドミンサーバ **Adminsrv.exe** は PC 上で動作し、PC を SuperVision によってリモート管理できるようにします。項目を管理ホストから PC にダウンロードするとき、または管理ワークベンチによってタスクリストや画面ショット要求を作成するときに、ローカルネームエージェントによって起動されます。

アドミンデータベース

Super Vision によって管理されている項目の集まり (コンピュータグループ、PC、ドライバ、フォルダ、ファイル、別名、プログラムマネージャのオブジェクト、レジストリファイル)。

アドミンワークベンチ

アドミンワークベンチを使うと、システム管理者はネットワーク上で X Vision ソフトウェアを実行しているすべての PC を中央から管理できます。PC の Windows のもとで動作する管理ワークベンチは、中央の UNIX サーバで管理データベースを保守するのに使用されます。外見は Windows のエクスプローラに似ています。

アドミンワークベンチサーバ

アドミンワークベンチサーバは管理ホスト上で動作し、アドミンデータベース上でのすべての操作に責任を負います。この UNIX プログラムは **workbsrv** と呼ばれ、PC 上でクライアントプログラムから要求が出されると、ローカルネームエージェントによって起動されます。プロセスが終了するまで動作し続けます。

アトム

文字列名に関連付けられている一意の数値 ID。X Window System では、プロパティ、型、選択を識別するのに使用されます。

アプリケーション

ワープロなどの特定の作業に使用されるコンピュータプログラム。“プログラム”と呼ぶこともあります。

インターネット

6大陸の数千箇所を結ぶ世界規模の主要 TCP/IP ネットワーク。政府機関、教育機関、軍隊、一般企業がユーザになっています。

インフォメーションサーバ

情報サーバは PC 上で動作し、ユーザビューワからユーザプロパティを表示します。**Infosrv.exe** と呼ばれるこのプログラムは、ユーザプロパティが要求されると LNA によって起動されます。

ウィンドウマネージャ

画面上で他のクラाइントのウィンドウの位置とサイズを制御する特殊な X Window System クライアント。ユーザ入力をウィンドウに送るときのポリシーも決定します。ウィンドウマネージャは、ウィンドウの外見やユーザインタフェースに影響を及ぼすクライアントによって提供される“ ヒント ”を認識することもあります。

遠隔手続き呼び出しサービス

Sun Microsystems の公衆ドメイン遠隔手続き呼び出し (RPC) メカニズムにもとづく、Vision コミュニケーションアーキテクチャによって提供される USP サービス。クライアントアプリケーションは、このサービスを使って、クライアントが動作しているマシンとは異なるサーバマシン上でも関数やプロシージャを実行できます。したがって、RPC を使用して、ネットワーク全体で関数やプロシージャを実行できます。

画像

X ディスプレイのカラー機能を説明する構造体。画像“ クラス ”は、*StaticGray*、*GrayScale*、*StaticColor*、*PseudoColor*、*TrueColor*、*DirectColor* といった各種のディスプレイハードウェアに対応します。画像には、カラーマップの中のカラーセル数についての情報も含まれています。

画像は、実際には、特定のディスプレイの使い方を説明しているもので、1つの画面に対して複数の画像を使用できます。たとえば、カラーディスプレイは、通常は、モノクロデバイスやグレースケールデバイスとして処理することが可能です。X サーバは、現在どの画像が使用可能かを指定し、デフォルトの画像を定義しています。

カットバッファ

X Window System ルートウィンドウ上のカットアンドペースト操作に使用されるプロパティ。8つのカットバッファを使用でき、`CUT_BUFFER0` から `CUT_BUFFER7` までの事前に定義済みのアトムによって名前が付けられています。アプリケーションによってカットバッファの中にテキストが挿入されることがあり、後で同一プログラムまたは別のプログラムによってここから検索できます。

カラーマップ

一連のカラーセル。各カラーセルが1つのRGB値を定義しています。ピクセル値は、ピクセルの赤、緑、青の各要素の明度を決定するカラーマップのインデックスとして使用されます。ほとんどの普及PCディスプレイでは、カラーマップを1つずつインストールできます。カラーマップの変更が可能なハードウェアの場合は、“仮想カラーマップ”をいくつでも作成でき、必要に応じてインストールしたりインストール解除したりできます。

管理ホスト

Super Vision 管理システムの中で中央サーバとして使用されている UNIX システム。

キーコード

PC キーボード上のキーを押すと生成される値。

キーボードマップ

PC キーボード上のキーと値を関連付ける表。

キーマップ

PC キーボード上のキーとエミュレートする端末のキーボード上のキー、または X Windows System の場合は keysym を関連付ける表。

キーマップエディタ

キーマップエディタを使うと、端末エミュレータによって使用されるキーボード設定ファイル(“キーマップ”)を作成したり変更したりできます。

クリップボード

Microsoft Windows アプリケーションがテキストやグラフィックを一時的に格納するための領域。

コネクションマネージャ

コネクションマネージャは、ユニバーサルトランスポートプロトコル (UTP) などの多重プロトコルを使用してプロキシホストに接続するときに PC 上で実行します。多重リンクの PC 側を制御します。コネクションマネージャは、最初の接続が要求されると自動的に起動し、多重リンクを使用する最後の接続が切断されると終了します。

コネクションモニタ

コネクションモニタを使うと、接続プロセスの監視や制御ができます。

[コントロールメニュー] アイコン

ウィンドウの左上角にあるアイコンで、これをクリックすると、[コントロール]メニューが表示されます。

システム管理者

企業内のすべての PC と UNIX サーバを管理する責任を負う人またはチーム。

修飾キー

別のキーの動作を変更するキー。修飾キーには、SHIFT、ALT、CTRL の各キーがあります。

ショートカット

ショートカットは、UNIX フォルダ、プログラム、ドキュメントを表すアイコンによって表示される Windows 2000/Me/98/95 デスクトップ上に置かれる項目です。アイコンをダブルクリックするだけで、フォルダまたはドキュメントを開いたり、プログラムを起動できます。

スクロールバー

ウィンドウの内容が完全に見えないときに、ウィンドウの右端または下端、あるいはその両方に表示されるバー。それぞれのスクロールバーには、2本のスクロール矢印と1つのスクロールボックスが表示されており、それを使ってウィンドウ内をスクロールできます。

ステータスバー

ウィンドウの一番下にあるメッセージを表示するバー。メニュー項目やツールバーのボタンを選択すると、ステータスバーの左側にメニュー項目やツールバーのボタンのアクションについての説明が表示されます。ステータスバーの右側には、Caps Lock (CAP)、Num Lock (NUM)、Scroll Lock (SCRL) の各キーがロックされているかどうかが表示されます。この領域には、アプリケーション固有の情報が表示されることもあります。

スレーブ印刷

VT420 端末エミュレータ上で使用可能な機能で、出力を画面ではなく (または画面とともに) プリンタに送信します。Windows プリンタドライバは使用されません。

タイトルバー

ウィンドウの最上部にあるバーで、タイトルバー上のボタンを使えば、[コントロール] メニュー上の多数のコマンドにマウスを使って簡単にアクセスできます。現在のプログラムやドキュメントについての情報が表示されることもあります。

ダイナミックリンクライブラリ

ダイナミックリンクライブラリ (DLL) は、実行時にプログラムにリンクされているルーチンやデータのバイナリコードを含むファイルで、普通は .DLL 拡張子が付いています。

多重プロトコル

複数のアプリケーションを単一のシリアル回線上で実行したり、1つのネットワーク接続を介して実行できるようにするユニバーサルトランスポートプロトコル (UTP) のようなプロトコル。

タスクサービス

Vision コミュニケーションアーキテクチャによって提供される USP サービス。このサービスを使うと、リモートホストシステムにログインしてプログラムを実行できます。

タスクバー

Windows 2000/Me/98/95 タスクバーは画面の一番下にあり、実行中のすべてのアプリケーションが表示されており、アプリケーションを切り替えることができます。[スタート] ボタンも表示されています。

端末

1 台のキーボードとビデオディスプレイから構成される入出力装置。普通は、マルチユーザシステムで使用されます。

端末エミュレータ

PC 画面上のウィンドウで、標準端末のエミュレーション機能を含みます。

端末サービス

Vision コミュニケーションアーキテクチャによって提供される、リモートホストシステムにログインしてコマンドを実行できるようにするサービス。端末サービスは、タスクサービスと、UNIX オペレーティングシステムまたは rlogin などのネットワークソフトウェアの一部として提供されるリモートログインサービスの 2 つの方法で提供できます。

中央処理装置

中央処理装置 (CPU) はコンピュータの制御部分で、命令を解釈して実行する装置です。マイクロコンピュータの場合、CPU やマイクロプロセッサは、命令を取り出して、復号化、実行し、コンピュータの主要データ転送パスであるバスとの間で情報を転送できます。

ツールコマンド言語

ツールコマンド言語 (Tcl) は、アプリケーションが共通コマンド言語として使用するよう設計されたスクリプト言語です。アプリケーションに追加可能な C ルーチンのライブラリから構成され、新規 C ルーチンを追加するか、新規プロシージャを Tcl 言語で作成すれば、簡単に拡張できます。

ツールバー

ツールバーを使うと、アプリケーションで使用される多数のコマンドにマウスを使って簡単にアクセスできます。デフォルトでは、ツールバーは、メニューバーの下の、アプリケーションウィンドウの一番上に表示されますが、ウィンドウ内の好きな場所に移動することもできます。

データ発信元

ODBC データ発信元は、DBMS プロダクトと、リモートオペレーティングシステムやそれにアクセスするのに必要なネットワークの組み合わせからなる特殊インスタンスです。

データベースウィンドウ

Microsoft Access データベースを開くと表示されるウィンドウ。[テーブル]、[クエリー]、[フォーム]、[レポート]、[マクロ]、[モジュール]の各ボタンがあり、各ボタンをクリックすれば、データベースの中の該当するタイプのすべてのオブジェクトの一覧が表示されます。

データベースサーバ

データベースサーバは Vision コミュニケーションデータベースにアクセスします。Dbserve.exe と呼ばれる PC プログラムの 1 つです。データベースサーバを起動時に起動して、バックグラウンドで連続して実行するかどうかを指定できます。

デスクトップ

Windows デスクトップはコンピュータ画面の背景で、すべての Windows アプリケーションによって表示されるすべてのウィンドウのベースウィンドウになります。

デフォルト

ユーザが値を指定しないときにプログラムが提供する推定値。

電子メール

電子通信媒体を使って、手紙、メモ、レポートなどのテキストメッセージを送信すること。

ドキュメント

ワープロによって作成されたレポートなど、プログラムが情報を保持するために作成するファイル。または、セッション中の端末エミュレータなどのプログラムの動作を制御する設定ファイル。

匿名 FTP

TCP/IP ネットワーク上で、ホストにログインしないでホストからファイルをダウンロードする手法の 1 つ。“匿名”でログインし、電子メールアドレスをパスワードとして指定します。

ドッキングバー

ツールバーを配置できる場所。

ドメインネームシステム

ドメインネームシステム (DNS) は、TCP/IP ホスト向けの階層および静的命名サービスを提供します。DNS サーバはネットワーク上のそれぞれのコンピュータのネットワークアドレスを保持しています。PC がネームサーバを呼び出してアドレス情報を取得すると、DNS サーバはシンボリック名を IP アドレスに変換します。

ドラッグ

マウスでポイントして、マウスボタンを押しながらマウスを移動すること。

トランスポート

トランスポートは、リモートホストにアクセスするためのメカニズムを定義します。トランスポート定義には、接続に使用するプロトコル、ログインやコマンドの実行に使用されるスクリプト、必要な認証、使用可能なサービスとその指定方法を指定する詳細設定が含まれています。

認証サーバ

認証サーバは、Vision コミュニケーションアーキテクチャセキュリティモデルの主要構成要素です。**authsrv** と呼ばれるプログラムで、UNIX 版 Vision サービスがインストールされている UNIX ホスト上で動作します。サービスを使用するのに認証が必要な場合は、クライアントは認証サーバにアクセスして、権限を得る必要があります。

ネームスペース環境

ネームスペース環境 (NSE) は、Vision 資源にアクセスするためのトランスポートとは独立した一貫性のあるメカニズムです。NSE は、資源を識別するのにユニフォームネーミングコンベンション (UNC) 形式を使用します。クライアントは、どのネットワークアドレス指定形式からも独立した文字列を使って、特定のホストシステム上でサービスを要求できます。

ネームスペースマネージャ

ネームスペースマネージャ (NSM) は、フォルトトレランスのある反復ネーミングサービスです。1 つまたは複数の UNIX 版 Vision サービスシステムが NSM ホストとして指定されると、これらのホスト上で NSM プロセスが自動的に起動されます。NSM は、ネームスペース環境 (NSE) を実装して管理します。

ネットワークチェックアウト

ネットワークチェックアウトは、TCP/IP ソフトウェア上で一連のネットワークテストを実行する診断プログラムの 1 つで、Vision ソフトウェアがシステム上で正しく動作するようにします。

パターンマッチング

ホストエクスプローラなどのブラウザプログラムが、リモートファイルタイプに関連付けられているパスネームパターンを一致させることによってUNIX ファイルを識別できるようにする手法。この結果、ファイルを拡張子以外でも認識できるようになり、1つのファイルタイプに対していくつでもパターンを設定できます。たとえば、Text Document ファイルタイプは、X Vision ソフトウェアがインストールされているときには、*.c、*.cc、*.h、*.txt、docs/*、README* のパターンと関連付けられています。したがって、README.TV ファイルはテキストドキュメントとして認識されます。

バッキングストア

X Window System サーバによって割り当てられるメモリで、クライアントウィンドウの内容のコピーをPC上でローカルに保持しています。他のウィンドウの裏側に隠れているウィンドウが前面に表示できる状態になったときに、素早くこれを再描画できるようにするために使われます。サーバは、クライアントのウィンドウの新しく識別できた部分の再描画を要求するメッセージをクライアントに送信しなくても、ローカルコピーからウィンドウを再描画できます。

パニング

ウィンドウの別の部分が見えるように、部分的に画面から隠れているウィンドウを移動する機能。Xサーバでは、マウスポインタを当該ウィンドウ内の画面の端に移動すると、ウィンドウの位置が再決定されます。

非ブロックモード

Windows Sockets オプションの1つで、関連する操作が不完全でも、関数が返されるようにします。必須オプションなので、このオプションが指定されていないと、`recv()` などの Winsock コールは、データが受信されるまで Windows をブロックし、Windows は正しく動作しません。

表示モード

Xサーバをマルチウィンドウモードで使用すると、各Xクライアントを固有のMicrosoft ウィンドウに表示して、組み込みVWMウィンドウマネージャによって管理できます。これに対して、シングルウィンドウモードでは、Xワールド全体が1つのMicrosoft Windows の中に表示され、Xクライアントの管理には別個のウィンドウマネージャが使用されます。

ファイルアクセスサービス

Vision コミュニケーションアーキテクチャによって提供される USP サービス。リモートホスト上でファイルストアを参照したり処理したりする機能を提供します。ホスト間でファイルを転送することもできます。

ファイル転送プロトコル

コンピュータ間でファイルを転送するためのインターネットのアプリケーションレベルの標準プロトコル。

フォント

一定のサイズとデザインを持つ一連の文字、数字、句読点、記号。

フォントサーバ

フォントサーバはリモート UNIX ホストや VMS ホストで動作するプログラムで、当該ホストに保存されているフォントを X サーバで使用可能にします。フォントサーバがあると、フォント管理が簡単になり、X サーバが使用されている各 PC 上にフォントをインストールする必要がなく、ユーザは別の PC を使用するときでもいつでもフォントにアクセスできます。

フォントサーバを介してアクセスするフォントは、カタログという名前付きグループにまとめることができます。カタログはフォントファイルが収められているディレクトリに相当します。

フォントパス

X Window System フォントファイルを含むディレクトリの一覧です。それぞれのディレクトリには、ディレクトリの中で使用可能なフォントを記述する FONTS.DIR ファイルを含める必要があります。

プリントサーバ

プリントサーバはリモート UNIX システムからの印刷要求を受け取り、ローカル Windows プリント上で印刷します。Windows アプリケーションからの印刷要求をリモート UNIX システムに送信するためのクライアントとしても動作します。プリントサーバを起動時に起動するかどうかを指定できます。

ブロードキャスト

Windows Sockets オプションの 1 つで、ソケット上でブロードキャストメッセージを伝送できるようにします。

プロキシ

プロキシは、ネットワークに接続された UNIX システムで動作する支援プロセスで、ネットワーク上で使用している PC の代理として動作します。

プロトコル

プロトコルは、コンピュータが他のデバイスとやりとりするのに使用する言語です。2 台のコンピュータが相互にやりとりするためには同じプロトコルを使用する必要があります。

プロパティ

プロパティとは、オブジェクトの状態、体裁、値を定義するのに使用されるオブジェクトの特性をいいます。

変換 DLL

Microsoft Access などのアプリケーションとデータソースとの間でやりとりされるすべてのデータを変換するダイナミックリンクライブラリ (DLL)。トランスレータの最も一般的な使い方は、異なる文字セット間で文字データを変換する場合です。トランスレータは、暗号化と暗号解除や圧縮と展開などのタスクも実行できます。

ホスト

アプリケーションを実行できる、アクセス可能なリモートマルチユーザコンピュータ。

ホストエクスプローラ

Windows 2000/Me/98/95 でホストエクスプローラを使うと、リモートホストコンピュータを設定したり、フォルダやファイルを参照したり、プログラムやドキュメントを開いたりできます。ファイルやフォルダの作成や削除を行うだけでなく、リモートコンピュータ上で、または PC にファイルやフォルダを移動したりコピーしたりできます。

ホストファインダ

ホストファインダは、ローカルネットワーク上で UNIX ホストを検出し、Vision コミュニケーションデータベースの中でエントリを作成したり変更したりします。検索するホストのタイプと起動時にホストを実行するかどうかを指定できます。

ホストマネージャ

Windows NT でホストマネージャを使うと、リモートホストコンピュータを設定したり、フォルダやファイルを参照したり、プログラムやドキュメントを開いたりできます。ファイルやフォルダの作成や削除を行うだけでなく、リモートコンピュータ上で、または PC にファイルやフォルダを移動したりコピーしたりできます。ホストマネージャを使用して、インターネット FTP サーバに接続できます。

マクロ

2 つ以上のアプリケーションプログラムコマンドが保存されているリスト。検出されると、コマンドを再実行してタスクを達成します。マクロを使うと、いくつかのオプションを選んだり、複数のキーボードコマンドを使用しなければならない冗長で繰り返しの多いタスクを自動化することができます。

マクロ言語

アプリケーション固有のプログラミング言語。マクロを記録する代替手段として、マクロを作成するのに使用されます。上級マクロ言語には、完全なプログラムコマンド構造セットが含まれ、アプリケーションを高度に自動化かつカスタマイズして使用できます。

メッセージパッド

メッセージパッドプログラムは、ユーザに送信されたテキストメッセージをポストイットメモのようなウィンドウに表示します。マルチメディアメッセージの場合は、該当するビューアプログラムを実行します。

メニューバー

ウィンドウの最上部にあるバーで、使用可能なメニューが表示されています。

文字

キーを押すと画面に生成される文字、数字、句読点、記号。

文字端末アプリケーション

VT420 などの文字ベース端末を使用するように設計されている UNIX アプリケーション。

モデム

コンピュータ装置で使用されるデジタル信号を一般に電話回線で使われるアナログ信号に変換したり、この逆を行う装置。モデムという用語は、*i*モジュレータ/*i*デモジュレータという名前からとったものです。

ユーザデータグラムプロトコル

ユーザデータグラムプロトコル (UDP) は標準 TCP/IP プロトコルの 1 つで、アプリケーションはデータグラムを別のマシンのアプリケーションに送信できます。UDP は、インターネットプロトコル (IP) を使用してデータグラムを配布します。UDP データグラムにはプロトコルポート番号が含まれている点が IP データグラムと異なり、UDP データグラムでは、送信側からリモートマシン上の複数のアプリケーションを区別できます。

ユーザビューワ

ユーザビューワプログラムを使うと、X Vision ソフトウェアを実行しているすべての PC ユーザのリストを得ることができ、テキストメッセージやマルチメディアメッセージを別の Vision ユーザに送信できます。

ユニバーサルセッションプロトコル

ユニバーサルセッションプロトコル (USP) を使うと、Vision コミュニケーションアーキテクチャは複数のトランスポートプロトコルをサポートしながら、他のソフトウェア階層への標準通信インタフェースを提供できます。USP は、当該トランスポートが使用可能であれば、TCP/IP を介して直接に動作します。TCP/IP を PC 上で使用できない場合は、ネットワーク上の PC を表わす USP プロキシプロセスが UNIX システムで動作します。プロキシプロセスは、使用可能なトランスポート上で動作している UTP を使用します。

ユニバーサルトランスポートプロトコル

ユニバーサルトランスポートプロトコル (UTP) は、通信環境でエラーが訂正されたトランスポートを提供する信頼性の高い多重プロトコルです。これを使うと、Vision PC や UNIX ソフトウェアは RS232C シリアルリンクなどの非 TCP/IP トランスポートを介して通信できます。

ユニフォームネーミングコンベンション

ユニフォームネーミングコンベンション (UNC) は、ネットワークファイルサーバを含むパスの標準形式です。UNC 名とは、他のユーザやアプリケーションが明示的な接続を確立し、ネットワークを介して資源にアクセスできるようにデバイス、コンピュータ、資源に付けられる名前をいいます。UNC は、ユニバーサルセッションプロトコル (USP) が次の形式の資源を識別するのに使用されます。

`¥¥hostname¥service¥parameters`

ここで、*hostname* は Vision コミュニケーションアーキテクチャのホスト名を、*service* はサービス名を、*parameters* はサーバ固有のパラメータを表わしています。ユニバーサルネーミングコンベンションともいいます。

ライセンスサーバ

ライセンスサーバは `licsrv` と呼ばれるプログラムで、Vision コミュニケーションアーキテクチャによって提供されるライセンス管理メカニズムの構成要素の 1 つです。サブネットワーク上の UNIX 版 Vision サービスホストのうちの 1 つで動作します。

リモート印刷

リモート印刷は、使用している PC 上の Windows アプリケーションからリモート UNIX プリンタに印刷する機能です。

リモートコマンドプロセッサ

PC などのリモートコンピュータから UNIX システムに送信されたコマンドを実行する、UNIX システム上で動作するプログラム。

リモート実行サービス

Vision コミュニケーションアーキテクチャによって提供されるサービスで、リモートホストシステム上でコマンドを実行する機能を提供します。

リモートプログラムスタータ

リモートプログラムスタータを使うと、端末エミュレータを使ってログインしなくても、使用している PC からリモートシステム上のプログラムを実行できます。リモートプログラムを実行するのに必要なすべての設定をリモートプログラムスタータ (RPS) ドキュメントに保存できます。

リンガー

Windows Sockets オプションで未送信データがあってもソケットが閉じないようにします。

ルートウィンドウ

X のウィンドウは、ルートウィンドウをベースとしてツリーに似た階層に並べられています。ルートの子ウィンドウはトップレベルウィンドウといわれ、普通はクライアントごとに1つずつのトップレベルウィンドウがあります。トップレベルウィンドウが子ウィンドウを持つこともあり、その中にさらに子ウィンドウがあることもあります。X サーバがシングルウィンドウモードのときは、ルートはメイン X サーバウィンドウの中に見えます。マルチウィンドウモードのときは、ルートを Microsoft Windows 画面背景として表示できます。

レジストリ

Windows 2000/Me/98/95 レジストリは、コンピュータを設定したり構成したりするのに必要な情報の中央保管場所です。Autoexec.bat、Config.sys、.ini ファイルが入っています。

ローカル印刷

ローカル印刷は、UNIX アプリケーションから使用している PC に接続されているプリンタに印刷する機能です。

ローカルネームエージェント

X Vision PC と UNIX 版 Vision サービスシステムはどれも、サーバプログラムの起動を行うローカルネームエージェント (LNA) を持っています。LNA の機能は Sun ONC RPC システムのポートマッパーに似ています。LNA の実行後直ちに起動されるサーバもありますし、クライアント要求に答えて起動されるサーバもあります。

ログサーバ

ログサーバは **logsrv** と呼ばれるプログラムで、UNIX 上でのデバッグおよびイベントログ機能を提供します。ローカルネームエージェントセッションの開始時に起動され、そのまま実行し続けます。

ワークベンチサーバ

ワークベンチサーバ **Workbsrv.exe** は PC 上で動作し、システム管理者がアドミンワークベンチから直接に PC を参照する操作を行います。参照が要求されると、アドミンワークベンチによって起動されます。

付録 B

UNIX コマンド

このセクションの内容は、次のとおりです（この章の内容はサポート対象外です）。

- UNIX コマンドの概要
- UNIX コマンド

UNIX コマンドの概要

admcopy	PC と UNIX ホスト間でファイルとディレクトリをコピーします。
admexec	PC 上でプログラムを実行します。
admindex	アドミンデータベースの索引を構築します。
admls	アドミンデータベースの内容を一覧表示します。
admupdate	アドミンされている PC の更新を開始します。
authsrv	認証サーバ
browsesrv	ファイルアクセスサーバ
dumppath	パス環境ファイルを作成します。
licadmin	ライセンス管理統計を表示します。
licsrv	ライセンスサーバ
lna	ローカルネームエージェント
logsrv	ログサーバ
newterm	新しいプロセスに対して utmp エントリを作成または削除します。
nsmadmin	現在実行中のネームスペースマネージャを一覧表示します。
nsmrpcsrv	ネームスペースマネージャに対する RPC インタフェース
nsmsrv	ネームスペースマネージャ
odbc_server	Vision ODBC ドライバのデータベース固有のサーバモジュール
pclist	管理されている PC を登録します。
pcprtclnt	PC プリンタに UNIX ファイルを出力します。
printsrv	プリントサーバ
termdump	USP タスクサーバ構成ファイルを作成します。
tsrv	タスクサーバ
usputppxy	UTP ベースの USP プロキシ
visionver	プログラムのバージョン番号を表示します。
workbsrv	アドミンワークベンチサーバ
xresume	Vision Resume プログラム

admcopy

構文

`admcopy source destination`

説明

`admcopy` を使用できるのは、SuperVision がインストールされている場合だけです。このコマンドは、UNIX ホストから PC にファイルとディレクトリをコピーしたり、PC から UNIX にファイルだけをコピーするために使用されます。

二重スラッシュ (//) の後に PC 名を指定し、それに続いてドライブ名、パス、ファイル名を順に指定します。また、UNIX から PC へコピーする場合は、コピー先ファイルを指定する必要があります。UNIX パスには、相対パス名でもフルパス名でも使用できます。

UNIX のファイルが存在していない PC フォルダにコピーした場合、`admcopy` がそのフォルダを作成します。

注

`admcopy` を使用しないでアドミンデータベースから PC にファイルをコピーした場合、コピーしたファイルは実行できません。これは、コピーしたファイルに長いファイル名コンテナが添付されているためです。このファイルを使用するには、別のデータベースにコピーしなければなりません。長いファイル名コンテナの詳細については、`admls` を参照してください。

UNIX のファイルおよびディレクトリにワイルドカードを使用できるのは、UNIX から PC にコピーする場合だけです。

`admcopy` では、行終端文字の処理は行いません。

`admcopy` では短いファイル名を使用しますが、コピー元であるかコピー先であるかにかかわらず、ファイル名を引用符で囲むことで PC 側に長いファイル名を指定できます。

ファイル

`vision.conf` - Vision 構成ファイル

例

PC から現在の UNIX ディレクトリに `Chimes.wav` ファイルをコピーします。

```
admcopy/geoffs/c:/windows/media/chimes.wav
```

アドミンデータベースから PC に同じく `Chimes.wav` ファイルをコピーします。

```
admcopyPCs/geoffs/Files/chimes.wav/geoffs/c:/windows/media
```

admexec

構文

```
admexec [-m] [-M] [-h] [-w] [-i] //PCname/{path|alias}/command
```

説明

admexec を使用できるのは、SuperVision がインストールされている場合だけです。このコマンドは、指定された PC でアプリケーションを実行するために使用されます。絶対パス、Windows レジストリで定義されているフォルダパスを示す別名、またはフォルダが Windows パス内にある場合は実行可能名を指定できます。

次のオプションを使用できます。

- m** アプリケーションをアイコン化して起動します。
- M** アプリケーションを最大表示して起動します。
- h** アプリケーションをバックグラウンドに隠します。
- w** アプリケーションが閉じるのを待ってから、UNIX コマンドプロンプトに戻ります。ホストは 1 秒ごとに アドミンサーバをポーリングして、PC 上で処理が終了したかどうかを確認します。
- i** 1 秒ごとにアドミンサーバをポーリングする間隔を 1 秒から変更できるようにします。-w が指定されている場合に限り有効です。

注

自分の PC 以外の PC でアプリケーションを実行するには、アドミニストレータの特権が必要です。

ファイル

visionpriv - 認証データベース

関連項目

「Managing the Administration Database」の章の「Aliases」

例

フルパスを指定し、リモートプログラムスタータを最大表示して起動します。

```
admexec-M//geofffs/progra-1/vision-1/rps
```

既知の別名を使用します。

```
admexec-M//geofffs/DirectionApp/rps
```

アドミンサーバのポーリング間隔として、10 秒を指定します。

```
admexec-Mwi.10//geofffs/DirectionApp/rps
```

admindex

構文

```
admindex
```

説明

admindex を使用できるのは、SuperVision がインストールされている場合だけです。このコマンドは、アドミンデータベースの索引を構築するために使用されます。索引にはファイルごとまたはオブジェクトごとのバージョン番号が含まれ、これによって更新時にアドミンデータベースから PC にどのファイルが送信されたのかを識別できます。

注

SuperVision のインストール時に、**svision** という管理グループが作成されます。UNIX ユーザがコマンド行から **admindex** を実行するには、このグループのメンバーでなければなりません。このグループのメンバーでない場合には、アクセス権は却下されます。

このコマンドは、アドミンワークベンチを使用して索引を構築できるため、通常コマンド行から実行されません。また、アドミンワークベンチが実行されている間は、データベースがロックされるため、索引を構築できません。

ファイル

```
vision.conf          - Vision 構成ファイル
```

関連項目

```
admupdate
```

例

```
admindex
```

admls

構文

```
admls [-l] [-a] [-R] filename
```

説明

admls を使用できるのは、SuperVision がインストールされている場合だけです。このコマンドは、アドミンデータベースの内容を一覧表示するために使用されます。**admls** は UNIX ディレクトリで使用できますが、アドミンデータベースのディレクトリで使用した場合は、Windows の長いファイル名 (最大 255 文字) がこのデータベースの標準的な 8.3 形式のファイル名の隣の括弧内に表示されます。無効な文字はすべて、次のように省略されます。

```
a\largefo.bat[a\largefolder.bat]
```

アドミンワークベンチで作成されたファイルはすべて、長いファイル名と短いファイル名が同じであっても、長いファイル名コンテナが設定されます。次に示すようにファイル名の隣にアスタリスクが付けられているファイルは、アドミンワークベンチで作成されたファイルではなくコンテナは設定されません。

```
BOOTLOG.TXT [BOOTLOG.TXT]  
SCANDISK.LOG [SCANDISK.LOG]  
*wevent.log
```

次のオプションを使用できます。

- | | |
|-----------------|--|
| -l | 短いファイル名と長いファイル名を長形式 (ファイルタイプとバイト数を表示) で一覧表示します。 |
| -a | すべてのエントリを一覧表示します。通常、名前の先頭がピリオド (.) のエントリは除外されます。ファイル記憶域から削除するようにマーキングされたファイルは、名前の先頭にピリオドが付けられます。 |
| -R | 検出されたサブディレクトリをすべて繰り返し一覧表示します。 |
| <i>filename</i> | 条件を満たすファイル名を一覧表示します。ディレクトリ名とワイルドカードを使用できます。ワイルドカードは短いファイル名に基づきます。 |

注

長いファイル名が重複している場合は、Windows 2000/Me/98/95 の場合と同じように処理されます。たとえば、長いファイル名 `geoff a b c` は `geoffabc` という短いファイル名になり、長いファイル名 `geoff a b c d` は `geoffa~1` という短いファイル名になります。

ファイル

`vision.conf` - Vision 構成ファイル

関連項目

`admcopy`

例

```
admls -lR
```

```
admls -l *.TXT
```

admupdate

構文

```
admupdate [-p] [-g] [-l] [-a] [-f] [-n] [-s] [-t] [-v]
```

説明

admupdate を使用できるのは、SuperVision がインストールされている場合だけです。このコマンドは、アドミンデータベースの変更を PC またはグループにダウンロードするために使用されます。

次のオプションを使用できます。

- p** *pc1,pc2,...* 更新を行う PC 名。このオプションは、**-g** オプションと一緒に使用することはできません。
- g** *group1,group2,...* 更新を行うグループ名。このオプションは、**-p** オプションと一緒に使用することはできません。
- l** 更新する必要があるファイルをすべて一覧表示します。
- a** これまでにダウンロードしているかどうかにかかわらず、索引内の内容をすべてダウンロードします。
- f** アドミンサーバ ダイアログボックスをフォアグラウンドにします。このダイアログボックスでは、更新情報の概要が示され、ユーザに確認を求めます。
- n** [アドミンサーバ] ダイアログボックスを表示しないようにします。このため、ユーザの確認がバイパスされます。
- s** サイレント更新。このオプションを設定すると、アクションが失敗しても、**admupdate** ではコマンド行に出力が表示されません。
- t** [アドミンサーバ] ダイアログボックスが閉じられるまでに応答を待機する時間を指定します。デフォルトの値は 1 分です。応答がない場合は、更新が自動的に行われます。
- v** 各 PC またはグループ、およびダウンロードされているすべてのファイルとオブジェクトを一覧表示します。

注

自分の PC 以外の PC に変更内容をダウンロードするには、アドミニストレータの特権が必要です。

admupdate は索引の構築に基づき、データベースに目立った変更がない場合は実行されません。索引の詳細については、**adminindex** を参照してください。また、**admupdate** はアドミンワークベンチの実行中は実行されません。

ファイル

vision.conf	- Vision 構成ファイル
visionpriv	- 認証データベース

関連項目

adminindex

例

```
admupdate-pgeoffs, jim, stevel-fw
```

authsrv

構文

```
authsrv [-r] [-e]
```

説明

authsrv は認証サーバで、USP **vwauth** サービスを提供します。このコマンドはローカルネームエージェントセッション (**lna** を参照) の開始時に起動され、一度起動されると実行は継続されます。**authsrv** は起動されると、ユーザ名とアクセス権が格納された **visionpriv** というデータベースを読み取り、クライアントプログラムから要求が出されるのを待機します。

次のオプションを使用できます。

- r 認証データベースを読み取ります。デフォルトでは、この処理は認証サーバの起動時に行われます。プログラムの実行中に、このスイッチを指定して再度コマンドを実行した場合、プログラムの別のインスタンスが起動されるわけではなく、既存のプロセスが再度データベースファイルを読み取ります。
- e 期限切れであっても、正しいパスワードであれば受け付けます。

注

authsrv の実行中に **visionpriv** ファイルを編集した場合には、**authsrv+r** コマンドを使用して、データベースを再度読み取る必要があります。これには、スーパーユーザ特権が必要です。

ファイル

- vision.conf** - Vision 構成ファイル
- visionpriv** - 認証データベース

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「認証の概要」

例

```
authsrv
```

browsesrv

構文

```
browsesrv
```

説明

browsesrv はファイルアクセスサーバで、USP **Ftrans** サービスを提供します。このコマンドは必要に応じてローカルネームエージェント (**lna** を参照) で起動され、一度起動されるとブラウザセッション中は有効です。

関連項目

「リモートホストの参照」の章の「参照の概要」

例

```
browsesrv
```

dumppath

構文

```
dumppath
```

説明

dumppath では、現在のユーザのホームディレクトリに **.visionpath** ファイルを作成して、ファイルアクセスサーバで使用できるように、PATH 環境変数の現在の設定を格納します。これによって、ホストエクスプローラの [パス] フォルダに現在の [パス] 内のすべてのフォルダが格納されます。

注

dumppath を実行しないでこのファイルを作成すると、パスフォルダの内容はファイルアクセスサーバで提供されるデフォルトの設定になります。

.visionpath ファイルに常に最新情報が格納されるように、**.login** または **.profile** 起動ファイルに **dumppath** コマンドを取り込むことをお勧めします。

ファイル

.visionpath - パスデータファイル

関連項目

「リモートホストの参照」の章の「参照の概要」

browsesrv

例

```
dumppath
```

licadmin

構文

```
licadmin [-a] [-f] [-i] [-l] [-r]
```

説明

licadmin は、ライセンスサーバ (**licsrv** を参照) にアクセスしてライセンス管理タスクを行うクライアントプログラムです。

次のオプションを使用できます。

-a *license_number*

ライセンスデータベース **cola.dat** にライセンス番号を追加します。ライセンス番号を重複して追加しようとすると、この操作は無視されます。

-f

ライセンス要求が却下された回数を表示します。

-i

使用中のライセンスを表示します。

-l

cola.dat 内の現在のライセンス番号を一覧表示します。

-r *license_number*

cola.dat からライセンス番号を削除します。

ファイル

vision.conf

- Vision 構成ファイル

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ライセンス管理の概要」

例

最初の例では、ライセンスの使用状況を示します。

```
[3] % licadmin -i
Product code   License domain   Total licenses   Available licenses
-----
Browser        0                 100              32
VT420 emulator 0                 100              12
Infrastructure 0                 101              10
```

次に、失敗した要求に関する情報を示します。この例では、却下された要求はありません。

```
[4] % licadmin -f
Product code   License domain   Failed requests
-----
Browser        0                 0
VT420 emulator 0                 0
Infrastructure 0                 0
```

licsrv

構文

```
licsrv [-a] [-h] [-k]
```

説明

licsrv はライセンスサーバで、USP **lcnsesvc** サービスを提供します。このコマンドは、マスタネームスペースマネージャ (**nsmsrv** を参照) が実行されているホスト上で、ローカルネームエージェント (**lna** を参照) によって自動的に起動されます。このサーバは、クライアントプログラムから最初にサービス要求が出された際に起動され、一度起動されると実行は継続されます。現在のトークンに関する情報が、**licsrv.dat** ファイルに書き込まれます。

ライセンスサーバは、NSM で作成され削除されたロックファイルが存在するかどうかを確認することによって、アクティブであるかどうかを定期的に確認します。さらに、名前が NSM に登録されているかも確認されます。これらの条件が満たされていない場合は、異なるホスト上の別の NSM がマスタになっているとみなされ、終了されます。

licsrv は起動されると、**licsrv.dat** ファイル、**cola.dat** ファイル、**prodinfo.dat** ファイル、**domain.dat** ファイルを読み取り、ライセンスキャッシュを初期化します。これらのファイルはバックアップ NSM を実行しているホストに定期的にコピーされるため、バックアップ NSM がマスタになり新しいライセンスサーバを起動した場合でも、それまでのライセンス情報は保持されます。

次のオプションを使用できます。

- a age** ライセンスサーバの起動時にライセンスキャッシュを初期化するために使用する **licsrv.dat** ファイルの最大 *age* を秒単位で設定します (デフォルトの値は 3600)。
- h interval** ライセンスサーバがライセンスキャッシュを調べて、期限切れのライセンストークンを見つける *interval* を秒単位で設定します (デフォルトの値は 600)。
- k interval** ライセンスサーバがロックファイルがあるかどうか確認するとともに、名前が NSM に登録されているかを確認するキープアライブ *interval* を秒単位で設定します (デフォルトの値は 180)。

ファイル

vision.conf	- Vision 構成ファイル
prodinfo.dat	- プロダクト情報データベース
domain.dat	- ライセンスドメインデータベース
cola.dat	- ライセンスデータベース
licsrv.dat	- ライセンスの利用度および使用状況ファイル

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ライセンス管理の概要」

例

licsrv

lna

構文

```
lna [-b]
```

説明

lna はローカルネームエージェント (LNA) で、サーバプログラムを起動します。Sun ONC RPC システムのポートマッパーと似たような働きをします。サーバによって、LNA の実行後直ちに起動されるものと、クライアントの要求に応じて起動されるものがあります。

LNA は通常、システム起動ファイル内のエントリによって、システムのブート時に起動されます。このエントリでは **-b** オプションを指定して、LNA が **servers** ファイルを読み取り、ブート起動セクションで指定されているプログラムを実行するように設定してください。これによって、LNA がネームスペースマネージャ (NSM) に登録されるとともに、LNA で起動されるサーバプロセス名も NSM に設定されます。こうして設定されたサーバは起動されると、NSM に登録されます。その後登録されていないサービスが NSM に要求された場合には、NSM から LNA に適切なサーバを起動するように求められます。

次のオプションを使用できます。

-b ブートタイムサーバを起動します。

ファイル

vision.conf	- Vision 構成ファイル
servers	- サーバファイル .VCA

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「サーバプログラムの起動」

例

```
lna-b
```

logsrv

構文

```
logsrv
```

説明

logsrv はログサーバで、デバッグログ機能とイベントログ機能を果たします。このコマンドは、ローカルネームエージェントセッション (**lna** を参照) の開始時に起動され、一度起動されると実行は継続されます。

デバッグログ機能としては、**vwdebuglog** ファイルを作成し、トラブルシューティングに役立つ情報を格納します。

イベントログ機能としては、**vweventlog** ファイルを作成して、システム管理者にとって重要なイベントをすべて記録します。

注

ログサーバは **VISION_TMPDIR** 変数で指定されているディレクトリに **vwpipe** という FIFO を作成します。この FIFO は、クライアントがログファイルにエントリを追加するために使用します。

logsrv はブート時に起動され、LNA からは再起動できません。再起動する必要がある場合は、手動で行ってください。

ファイル

vision.conf	- Vision 構成ファイル
vwdebuglog	- デバッグログファイル
vweventlog	- イベントログファイル
vwpipe	- ログパイプ

例

```
logsrv
```

newterm

構文

```
newterm command  
newterm -r utmp-entry
```

説明

上記の最初の書式の場合は、**newterm** は動作している tty の **utmp** ファイルにエントリを作成してから、*command* をフォークします。**newterm** コマンドが USP 端末セッションで使用されるのは、動作している tty の **utmp** ファイルにエントリがない場合だけです。

-r オプションは、タスクサーバ **tsrv** で障害が発生した場合など特殊な場合に、**utmp** ファイルからエントリを削除するために使用します。パラメータ *utmp-entry* は tty の識別子です (例を参照)。

注

newterm を使用するにはスーパーユーザ特権が必要で、setuid ルート権限 (モード 4711) を使用してインストールします。

newterm を使用しない場合は、タスクサーバは通常ユーザの **utmp** エントリを 1 つ作成し、このエントリを監視してメッセージを待機します。メッセージはメッセージパッドにリダイレクトされます。

who コマンドを使用すれば、**utmp** エントリを参照できます。

関連項目

- 「端末エミュレーション」の章の「utmpファイル」
- 「端末エミュレーション」の章の「端末コマンド行」

例

専用の **utmp** エントリを使用して端末接続を開くには、[接続] ダイアログボックスに次のコマンドを指定します。

```
newtermcsh
```

utmp エントリを削除します。

```
newterm-ipty/ttyza
```

nsmadmin

構文

```
nsmadmin-n
```

説明

nsmadmin は、現在実行されているネームスペースマネージャを一覧表示します。

次のオプションを使用できます。

-n ネームスペースマネージャを一覧表示します。

注

nsmadmin はすべてのネームスペースマネージャに対する応答要求を伝送して、5 秒間待ってから終了します。

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ネームスペースマネージャ」

nsmsrv

例

次の例は **nsmadmin** の出力例で、マスタ NSM と 3 つのバックアップが表示されています。

```
cornflake%nsmadmin-n  
poptart.vision.co.uk(MASTER)  
cornflake (BACKUP)  
salt.vision.co.uk(BACKUP)  
poohbear.vision.co.uk(BACKUP)
```

nsmrpcsrv

構文

```
nsmrpcsrv
```

説明

nsmrpcsrv は USP **NameSvcRpc** サービスを提供するサーバです。このコマンドはローカルネームエージェント (**lna** を参照) で必要に応じて起動され、クライアントセッションの間は有効です。このコマンドは、ユーザビューワなどのさまざまなクライアントに対して、ネームスペースマネージャとのインタフェースを提供します。

関連項目

nsmsrv

例

```
nsmrpcsrv
```

nsmsrv

構文

```
nsmsrv [-b] [-g] [-h] [-i]
```

説明

nsmsrv はネームスペースマネージャ (NSM) で、USP **nsmsrv** サービスを提供します。このコマンドはローカルネームエージェント (LNA) セッション (**lna** を参照) の開始時に起動され、一度起動されると実行は続きます。このコマンドの主要な目的は、クライアントのネットワークアドレスに対する UNC 形式の要求を解釈処理することです。要求されたサーバが実行していない場合は、NSM は LNA にサーバを起動するように要求します。NSM の 2 番目の機能は、PC とそれに関連した管理ホストのリストを保持することです。

NSM ではすべてのローカルネームエージェントと、UNIX システムおよび PC 上で動作しているものなどローカルネットワークで動作しているすべての USP サーバのレジスタを保持します。

サブネットワーク上で複数の NSM が実行されている場合があります。このような場合、その中の 1 つがマスタ NSM で、その他の NSM は二次 NSM になります。二次 NSM はマスタに更新情報があるかどうかポーリングします。

次のオプションを使用できます。

- b interval** 二次ネームサーバがマスタに更新情報を要求する *interval* を秒単位で設定します (デフォルトは 600)。
- g interval** データをリモート NSM ゲートウェイに送信する *interval* を秒単位で設定します (デフォルトは 3600)。
- h interval** マスタ NSM が登録されているすべての LNA と接続して、アライブ状態になっているか確認する *interval* を秒単位で設定します (デフォルトは 120)。
- i ip-address** ブロードキャストに続いて NSM からの応答を受信する際に、指定された *ip-address* を無視します。このオプションは、マスタ NSM が実行されているシステムに複数の IP アドレスが割り当てられている場合に使用できます。

注

`vision.conf` の `VISION_NAMESVC` 設定によって、目的のマスタネームスペースマネージャがどこで実行されているかを調べることができます。

ファイル

`vision.conf` - Vision 構成ファイル

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ネームスペースマネージャ」

`nsmadmin`; `nsmrpcsrv`; `pclist`

例

```
nsmrpcsrv
```

odbc_server

構文

odbc_server [-i] [-r]

odbc_server はデータベースに固有のサーバプログラムです。詳細については、「Vision ODBC ドライバ」の章の個々のサーバモジュールに関するセクションを参照してください。

説明

odbc_server を使用できるのは、SQL-Retriever がインストールされている場合だけです。このコマンドは、Vision ODBC ドライバのデータベース固有のサーバモジュールです。ローカルネームエージェント (LNA) セッション (Ina を参照) の開始時に起動され、一度起動されると実行は継続します。

次のオプションを使用できます。

- i セキュリティ違反として扱われる固有のデータベース SQL を Security Manager で使用できるようにします。
- r Security Manager を有効にします。

ファイル

sqlrsec.conf -Security Manager 構成ファイル

サーバにはそれぞれ固有の構成ファイルがあります。詳細については、「Vision ODBC ドライバ」の章の個々のサーバモジュールに関するセクションを参照してください。

関連項目

「Vision ODBC ドライバ」の章の個々のサーバモジュールに関するセクション

例

Ingres 用のサーバプログラムです。

```
sqlr.ing40.dn-r-i
```

pclist

構文

pclist

説明

pclist が使用できるのは、SuperVision がインストールされている場合だけです。管理されているすべての PC をネームスペースマネージャ (NSM) に登録します。ローカルネームエージェント (LNA) セッション (lna を参照) の開始時に実行され、実行後終了します。

注

アドミンワークベンチを使用して PC を追加または削除すると、NSM キャッシュが自動的に更新されます。LNA セッションの開始時以外は、pclist を実行する必要はありません。

関連項目

nsmsrv

例

pclist

pcprtclnt

構文

```
pcprtclnt -h pc -p printer
```

説明

pcprtclnt コマンドは、UNIX 印刷ジョブをリモート PC 上のプリントサーバに引き渡します。

次のオプションを使用できます。

-h <i>pc</i>	印刷ジョブの送信先の PC 名
-p <i>printer</i>	リモート PC 上のプリントサーバで構成されているプリンタの論理名

注

pcprtclnt コマンドは、常に **lp** 印刷インタフェーススクリプトや **lpr** 出力ファイルなどのシェルスクリプトから呼び出されます。

ファイル

printers - プリンタデータベース

関連項目

「印刷」の章の「UNIX から PC へのローカル印刷」

例

```
pcprtclnt -hstevcl -pqms
```

printsrv

構文

```
printsrv
```

説明

`printsrv` はプリントサーバで、USP `prtServ` サービスを提供します。このコマンドは、必要に応じてローカルネームエージェント (`lna` を参照) によって起動され、一度起動されると実行が継続します。

関連項目

「印刷」の章の「UNIX から PC へのローカル印刷」

「印刷」の章の「PC から UNIX へのリモート印刷」

例

```
printsrv
```

termdump

構文

```
termdump
```

説明

termdump は、タスクサーバ **tsrv** で読み取られる 1 つまたは複数の端末構成ファイルを作成します。**termio** または **termios** というファイルが作成されますが、UNIX のバージョンによっては **winsize** というファイルも作成されることがあります。

このファイルには、UNIX の **stty** コマンドで構成される **tty** の現在の設定値が格納されます。**termio** ファイルによって、タスクサーバで提供されているデフォルトの値は無効になります。**termio** ファイルが提供されるのは、標準的な UNIX 端末セッションで提供されるような標準的な **tty** 設定を持つ端末エミュレータがインストールされている場合です。

注

新しい **termio** ファイルは、現在のファイルを上書きします。

USP タスクサーバが再起動されるまで、新しい設定は有効になりません。

termio ファイルを作成する場合、スーパーユーザ特権が必要になることがあります。

ファイル

vision.conf - Vision 構成ファイル

termio - 端末構成ファイル

関連項目

tsrv

例

```
termdump
```

tsrv

構文

```
tsrv [-i][ -k ][ -t ][ -u ]
```

説明

tsrv はタスクサーバで、USP **login** サービスを提供します。このコマンドはローカルネームエージェントセッション (**lna** を参照) の開始時に起動され、一度起動されると実行が継続します。このコマンドを使用すると、UNIX システムにログインした後、プログラムを実行できます。プログラムは端末接続に接続したままにしておくことも、端末接続と切り離しバックグラウンドで実行することも可能です。

タスクサーバでは、他の UNIX ユーザから送信されるメッセージを処理することもできます。デフォルトでは、ユーザが最初にログインした際に、**utmp** ファイルにエントリを 1 つ作成して、そのユーザ名に送信されるメッセージを待機します。そして、このメッセージを適切な PC 上のメッセージパッドに送信します。あるいは、一部のアプリケーションで必要とされるように、端末接続ごとに **utmp** エントリを作成することもできます。また、**-u** フラグを指定すると、すべてのユーザに対して **utmp** エントリが作成されます。個々のユーザを指定する場合は、**tsrv.rc** ファイルを編集してください。接続の開始時に **newterm** コマンドを使用して、特定の端末接続に対して **utmp** エントリを作成するように指定できます。

次のオプションを使用できます。

-i timeout イントレイの *timeout* を指定された秒数に設定します (デフォルトは 10)。タスクサーバは管理対象の端末接続のうち専用の **utmp** エントリが設定されていないいずれかの端末接続にメッセージが送信されるのを待機します。ただし、メッセージの終わり条件を検出できないため、時間切れを使用してメッセージの終了を確認します。

-k shared-memory 複数のタスクサーバが交信に使用できる共用メモリー領域を指定します (デフォルトは *itsrv1*)。これが必要になるのは、サードパーティのプログラムでも同じ名前の共用メモリー領域が使用されている場合だけです。*shared-memory* パラメータには、4 バイトの文字列を指定できます。

-t **termio** ファイルを読み取りません。**termio** ファイルの代わりに、内部のデフォルトの設定を使用します。

-u 常に、端末接続に対して **utmp** エントリを作成します。

注

`tsrv` は異なるクライアントからの複数のログインセッションを処理できません。さらに、複数の接続も処理できません。

ファイル

<code>vision.conf</code>	- Vision 構成ファイル
<code>termio</code>	- 端末構成ファイル
<code>tsrv.rc</code>	- タスクサーバ構成ファイル
<code>utmp</code>	- UNIX utmp ファイル

関連項目

「端末エミュレーション」の章の「USP タスクサービスの概要」

「端末エミュレーション」の章の「端末コマンド行」

例

```
tsrv
```

usputppxy

構文

```
usputppxy
```

説明

usputppxy は、UTP プロトコルを介して USP サービスを提供するヘルパープロセスです。このコマンドは、UTP トランスポートを使用してログインした際に、Vision コミュニケーションスクリプトで実行されます。このコマンドは、TCP/IP ネットワーク上で PC のプロキシの機能を果たします。PC クライアントからサービスが要求されると、**usputppxy** はネームスペースマネージャ (NSM) に通常の方法に従って USP アドレスを解釈処理するように要求します。これによって、ネットワーク上のいずれかのコンピュータにあるサーバと接続し、クライアントとサーバ間の通信をパッケージ化して、シリアル回線を介して伝送を行います。また、PC で複数のサーバが動作している場合は、**usputppxy** は動作しているサーバを NSM に登録します。

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ユニバーサルトランスポートプロトコル」

例

```
usputppxy
```

visionver

構文

`visionver program-file`

説明

`visionver` では、`Vision program-file` のバージョン番号を表示します。

例

```
%visionverIna  
Ina:Version=1
```

workbsrv

構文

`workbsrv`

説明

`workbsrv` を使用できるのは、SuperVision がインストールされている場合だけです。このプログラムはアドミンワークベンチサーバ で、USP `AdmWbSvc` サービスを提供します。このコマンドは必要に応じてローカルネームエージェント (`lna` を参照) で起動され、アドミンワークベンチセッション中は有効です。このコマンドは、アドミンホスト上でデータベースに対するあらゆる操作を行います。

ファイル

`vision.conf`

- Vision 構成ファイル

xresume、tpproxy、tpwatch

構文

```
xresume [ -licadd ] [ -version ] [ -xremotestart ] [ -xremotesuspend ]  
[ -xremotestop ]
```

```
tpproxy [ -version ]
```

```
tpwatch [ -version ]
```

説明

これらのプログラムを使用できるのは、Vision Resume がインストールされている場合だけです。上記の3つのバイナリではすべて Vision Resume プロキシサービスを提供します。これらのプログラムは、X セッションを中断し、しばらくした後再開し、同じ X クライアントを実行するために、Vision X サーバで使用できます。

次のオプションを使用できます。

-licadd *license_number*

ライセンス番号を入力します。

-version

バージョン番号を表示します。

-xremotestart

XRemote シリアル接続を介して Resume セッションを開始します。

-xremotesuspend

XRemote を使用して接続されている Resume セッションを中断します。

-xremotestop

XRemote を使用して接続されている Resume セッションを停止します。

xresume は Resume セッションを制御するために使用され、通常リモート実行によって Vision X サーバで起動されます。トランスポートに指定されるリモート実行方式によっては、**xresume** が格納されたディレクトリがユーザの PATH 環境変数に含まれていなければならないことがあります。これ以外のバイナリの位置は、これらのバイナリを起動する **xresume** バイナリが `/etc/vision.conf` ファイルの内容 (`$VISION_BINDIR` エントリ) から推測されます。

各ホストのユーザごとにプロキシサーバが1つずつ割り当てられています。それぞれのプロキシサーバインスタンスは `$HOME/tpinfo` ファイルに基づいているため、ネットワークアカウントが1つだけの場合は、プロキシサーバを一度に1つしか実行できません (ただし、ホストごとにアカウントが設定されている場合は、これは当てはまりません)。本来の使用モードはこのように1つのプロキシサーバを実行することで、これが Vision X サーバ内からプロキシサーバを制御するための本来の方法です。プロキシサービスを提供するホストの構成方法、および Vision X サーバを使用したプロキシサーバの制御方法については、「セッションの一時停止と再開」の章を参照してください。さらに、この章では、中断や再開といった用語の説明など、提供されて

いる機能についても説明されています。ユーザが制御しようとしたときにすでにプロキシサーバが実行されている場合は、そのインスタンスが制御されます (インスタンスの情報は `$HOME/.tpinfo` ファイルから取得します)。これに対し、プロキシサーバが実行されていない場合は、プロキシサーバが起動され、この新しいインスタンスが制御されます。また、プロキシサーバを起動すると、`tpproxy` プロセスと `tpwatch` プロセスも起動されます。

UNIX ベースの X システムの `.xsession` ファイルとほとんど同じ方法で、プロキシサーバの起動とともに起動する X クライアントを制御できます。セッションは `$HOME/.tpsession` で指定され、このファイルが存在しない場合には、代わりに `$VISION_BINDIR/default.tpsession` のシステム共通の定義が使用されます。このファイルも存在しない場合は、X クライアントは自動的に起動されませんが、プロキシサーバは正しく起動されます。`$HOME/.tpsession` ファイルと `default.tpsession` ファイルは両方とも一連のコマンド行から構成されます。このコマンド行は、すべてバックグラウンドで実行しなければならないので、最後に `&` が付けられます。また、先頭行は `#!/bin/sh` など、適切なコメント規則を使用できます (この場合は、行の先頭に `#!` が付けられています)。

このようなセッションファイルを使用して自動的に起動された X クライアントはすべてプロキシサーバと接続し、適宜 `DISPLAY` 環境変数が設定されます。プロキシサーバは、プロキシサーバと接続されている本来の X サーバにクライアントの出力を生成します。プロキシサーバ自身は正当な X プロトコルディスプレイであるため、プロキシサーバのインスタンスに固有の UNIX ホスト名に基づいてディスプレイ名とディスプレイ番号が割り振られます。このプロキシサーバのディスプレイ名が Vision X サーバでも使用できるので、Vision X サーバのディスプレイに直接接続されていなくても、プロキシサーバを介して接続されている X クライアントを起動できます。

プロキシサーバプログラム `tpproxy` および `tpwatch` では、それぞれ `$HOME/.tpproxylog` および `$HOME/.tpwatchlog` というログファイルを生成します。`xresume` コマンドが実行されている環境で `RESUME_DEBUG` 環境変数が設定されている (=1) 場合は、適量の情報がこれらのログファイルに書き込まれます。`RESUME_DEBUG` 環境変数が設定されていない (=0) 場合には、ごく一部の情報しかファイルに書き込まれません。この環境変数はシェル起動ファイル (`.profile` など) や `.cshrc`、あるいは `vision.conf` で設定できます。ただし、この変数を変更してもすでに実行中のプロキシサーバのログには影響を与えないため、Vision Resume セッションを一度終了してから起動し直す必要があります。

シリアル回線またはモデムを介して XRemote と接続している際に Vision Resume を使用するには、`.xinitremoterc` ファイルの最後に次の行を追加してください。

```
REMOTE_DISPLAY=$DISPLAY
export REMOTE_DISPLAY
xresume-xremotestart
```

Resume セッションを中断または停止するには、セッションで起動されている `xterm` 内から `-xremotesuspend` オプションまたは `-xremotestop` オプションを使用します。

注

xresume は起動されると、制御 PC が Vision Resume に対応したバージョンの Vision X サーバを実行しているかどうかを確認します。

Vision Resume は、独立した X Vision プロダクトとしてライセンスされています。X Vision の UNIX ソフトウェアのインストール時に Vision Resume ライセンス番号を入力しないと、試供用コピーが提供されます。後日プロダクトを購入した場合に、**xresume -licadd license_number** を使用して Vision Resume ライセンス番号を入力すれば、フルバージョンにアップグレードできます。このライセンスの取得は現在実行されている Vision Resume のプロキシサーバすべてに影響を与え、これらのプロキシサーバにライセンスが付与されます。

ファイル

/etc/vision.conf	Vision 構成ファイル。 tpproxy および tpwatch の位置を検出するために xresume で使用されます。
\$HOME/.tpinfo	プロキシサーバの X ディスプレイ名など、特殊な状態情報が格納されたファイル
\$HOME/.tpproxylog	ユーザ単位の tpproxy のログファイル
\$HOME/.tpwatchlog	ユーザ単位の tpwatch のログファイル
\$HOME/.tpsession	ユーザ単位のセッションファイル
\$VISION_ETCDIR/default.tpsession	広域のデフォルトのセッションファイル

関連項目

「セッションの一時停止と再開」の章の「Vision Resume の動作としくみ」

例

```
xresume-version
```

バグ

UNIX システムのシャットダウンなどにより、プロキシサーバが異常終了した場合でも、**\$HOME/.tpinfo** ファイルは保持されます。このため、ファイルに古い情報が格納され、ユーザがプロキシサーバを再起動できなくなります。システムのブート時に削除されるファイルのリストにこのファイルを追加するように、システム管理者に依頼してください。ファイルの内容は、UNIX システムをリブートすると、無効になります。

付録 C

UNIX ファイル形式

このセクションの内容は、次のとおりです（この章の内容はサポート対象外です）。

- UNIX ファイルの概要
- UNIX ファイル形式

UNIX ファイルの概要

.visionpath	パスデータファイル
cola.dat	ライセンスデータベース
domain.dat	ライセンスドメインデータベース
licsrv.dat	現在のライセンス利用度
printers	プリンタデータベース
prodinfo.dat	プロダクトデータベース
servers	サーバデータベース
sqlrsec.conf	Security Manager 構成ファイル
termio	端末構成ファイル
tsrv.rc	タスクサーバ構成ファイル
vision.conf	Vision 構成ファイル
visionpriv	認証データベース
vwdebuglog	デバッグログファイル
vweventlog	イベントログファイル
winsize	端末構成ファイル

注意 Vision 構成変数名の前にドル記号 (\$) が付いている場合は、その変数の値を使用します。

.visionpath

位置

`$HOME/.visionpath`

説明

`.visionpath` ファイルにはユーザの UNIX パスの詳細事項が記録されます。このファイルは、`dumppath` プログラムで作成され、ホストエクスプローラで読み取られます。

注

`.login` または `.profile` スクリプトファイルに最新の情報が記録されるように、`dumppath` コマンドを実行するよう指定することをお勧めします。

関連項目

`dumppath`

cola.dat

位置

`$VISION_ETCDIR/cola.dat`

説明

cola.dat ファイルはライセンスデータベースで、このサブネットワークで使用できるようにライセンスを与える X Vision ソフトウェアプロダクトと、ユーザの最大人数が定義されます。このファイルは、編集不可能なバイナリファイルです。

X Vision のソフトウェアを UNIX システムにインストールした場合、インストールスクリプトでは **licadmin** プログラムを使用して、ライセンスサーバを実行しているホスト上の **cola.dat** ファイルにプロダクトの COLA ライセンス番号を追加します。ライセンスサーバは、マスタ NSM を実行しているホストで実行されますが、これは新しいプロダクトをインストールしたばかりのホストの場合とサブネットワーク上の別のホストの可能性もあります。**cola.dat** ファイルは、バックアップ NSM を実行しているすべてのホストに定期的にコピーされます。

注

このファイルを変更した場合は、ライセンスサーバを再起動してください。

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ライセンス管理の概要」

licsrv

domain.dat

prodinfo.dat

licsrv.dat

domain.dat

位置

`$VISION_ETCDIR/domain.dat`

説明

domain.dat ファイルはライセンスドメインデータベースで、ライセンスドメインを定義し、ドメインごとに各プロダクトを使用できるようにライセンスを与えるユーザの最大人数を指定します。ドメインを指定するだけで、サブネットワーク上の特定のユーザグループにライセンスを割り当てることができます。デフォルトでは、ユーザ定義のドメインで指定されていない PC はドメイン 000 に属します。ドメイン 000 は、別のドメインに割り当てられていないすべてのライセンスが割り当てられます。**domain.dat** ファイルは、ASCII テキストファイルで、スーパーユーザ (ルート) 特権が付与されていれば、テキストエディタを使用して変更できます。ただし、このファイルのアクセス権は変更できません。

このファイルには、2 つのセクションがあり、それぞれ次のセクションヘッダが付けられています。ライセンスドメインを定義するセクションの形式は、次のとおりです。

[DOMAINS]

domain=domain-code pc-name [pc-name...]

セクションヘッダ [DOMAINS] の左右の角括弧は、必ず付けなければなりません。*domain-code* は、固有の任意の 3 けたの数字です。ドメイン 000 は予約されており、他のドメインで指定されていないすべての PC が含まれます。ドメインごとに、PC ノード名のリストをスペースで区切り、ドメインのメンバを定義する必要があります。この場合、*pc-name* の左右の角括弧はオプション項目であることを示し、角括弧を入力する必要はありません。

また、それぞれのドメインで各プロダクトを使用できるユーザの最大数を定義するセクションの形式は、次のとおりです。

[LICENSES]

domain=domain-code product=product-code users=number

セクションヘッダ [LICENSES] の左右の角括弧は、必ず付けなければなりません。*domain-code* は、[DOMAINS] セクションで指定されているドメインです。*product-code* に指定するプロダクトは、プロダクトデータベース **prodinfo.dat** で定義されていなければならず、ライセンス番号の先頭の 3 けたでプロダクトを識別できます。

ファイルを処理する場合、無効なプロダクトコードは無視されます。このドメインでこのプロダクトを使用できるようにライセンスを与えるユーザの最大 *number* を指定します。

X Vision のソフトウェアを UNIX システムにインストールする場合、エントリが格納されていない **domain.dat** ファイルがインストールされます。ライセンスサーバがこのホストで実行されている場合は (マスタ NSM がこのホストで実行されているため)、ファイルを編集してドメインを追加できます。ライセンスサーバが別のホストで実行されている場合は、このホストの **domain.dat** ファイルは無視されます。また、このホストでバックアップ NSM が実行されている場合は、**domain.dat** ファイルはマスタ NSM が実行されているホストからコピーされた **domain.dat** ファイルによって定期的を上書きされます。

ドメインにプロダクトの最大ユーザ数を割り当てる場合、プロダクトに指定するユーザの合計人数はライセンスが付与されたユーザの合計以内でなければなりません。

注

ファイルを変更した時点でライセンスサーバがすでに実行されている場合は、ライセンスサーバを再起動する必要があります。

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ライセンス管理の概要」

licsrv

prodinfo.dat

cola.dat

licsrv.dat

例

次の例では、ドメインを2つ定義して、それぞれのドメインにユーザを指定します。また、各ドメインで使用できるようにライセンスを与えるプロダクトも示します。

```
[DOMAINS]
domain=123 jim billw
domain=456 ron
[LICENSES]
domain=123 product=023 users=200
domain=123 product=024 users=200
domain=456 product=023 users=300
domain=456 product=024 users=300
```

licsrv.dat

位置

`$VISION_ETCDIR/licsrv.dat`

説明

`licsrv.dat` ファイルには、現在のライセンスの利用度と使用状況に関する情報が格納されます。このファイルは、編集不可能なバイナリファイルです。

`licsrv.dat` ファイルは、マスタ NSM と同じホスト上で実行されているライセンスサーバで作成および更新されます。このファイルはバックアップ NSM が実行しているホストに定期的にコピーされるため、バックアップ NSM がマスタ NSM になり新しいライセンスサーバを起動した場合でも、ライセンスの使用状況に関するそれまでの情報が保持されます。ただし、実際には、ライセンスの使用状況に関する情報は多少古い場合があり、ライセンスサーバの起動時に `licsrv.dat` ファイルを使用する場合に、`licsrv.dat` ファイルの最大期限 (デフォルトでは 1 時間) を指定するためのオプションがライセンスサーバで提供されています。

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ライセンス管理の概要」

`licsrv`

`domain.dat`

`prodinfo.dat`

`cola.dat`

printers

位置

`$VISION_ETCDIR/printers`

説明

`printers` ファイルはプリンタデータベースで、プリンタを構成するために使用され、RPCクライアントプログラム `pcprtclnt` で読み取られます。このファイルは、ASCII テキストファイルで、スーパーユーザ (ルート) 特権が付与されていれば、テキストエディタを使用して変更できます。このファイルのエントリの形式は次のとおりです。

local-printer-name:pc-name:remote-printer-name

local-printer-name はプリンタの UNIX 名で、*pc-name* は PC 名、*remote-printer-name* は [ローカルプリンタ] 設定で構成されているプリンタの論理名です。

関連項目

「印刷」の章の「UNIX から PC へのローカル印刷」

prodinfo.dat

位置

`$VISION_ETCDIR/prodinfo.dat`

説明

prodinfo.dat ファイルはプロダクト情報データベースで、X Vision のソフトウェアプロダクトとそのプロダクトを構成するコンポーネントを定義します。このファイルは、編集不可能なバイナリファイルです。

X Vision のソフトウェアを UNIX システムにインストールする場合、別の X Vision プロダクトで最新ファイルがインストールされていなければ、**prodinfo.dat** ファイルがインストールされます。最新の **prodinfo.dat** ファイルは、ライセンスサーバが実行されているホスト上になければならず、このホストでは同じくマスタ NSM が実行されます。また、この **prodinfo.dat** ファイルは、バックアップ NSM を実行しているすべてのホストに定期的にコピーされます。マスタ NSM を実行しているホスト以外のホストに最新の **prodinfo.dat** ファイルがインストールされている場合は、このファイルを手動でマスタ NSM ホストにコピーしなければなりません。

注

このファイルを変更した場合は、ライセンスサーバを再起動してください。

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「ライセンス管理の概要」

licsrv

domain.dat

cola.dat

licsrv.dat

servers

位置

\$VISION_ETCDIR/servers

説明

servers ファイルには、ローカルネームエージェント (LNA) から起動できるサーバのリストが格納されます。このファイルは、ASCII テキストファイルで、スーパーユーザ (ルート) 特権が付与されていれば、テキストエディタを使用して変更できます。**servers** ファイルのエントリの形式は次のとおりです。

```
service server m | s [options] [env:variable=value,...]
```

service はサービス名で、**server** はサーバプログラム名です。複数のクライアントをまとめて扱う場合は **m** オプションを指定し、1 つのクライアントを扱う場合には **s** オプションを指定します。また、**options** はコマンド行オプションスイッチで、**variable** はシェル形式の環境変数です。環境変数を複数指定する場合は、次のようにカンマで区切ります。

```
login tsrv m -u env: LINES=24,COLUMNS=132
```

server にパスが指定されていない場合、LNA では **\$VISION_BINDIR** 内のプログラムが検索されます。また、エントリにフルパスを指定することもできます。

1 つのサーバプログラムで複数のサービスを提供することもでき、この場合にはサービスごとにエントリがファイルに含まれます。

m オプションを指定すると、1 つのプロセスで複数のクライアントを処理できます。このタイプのサーバは、一度起動すると、1 つのクライアントの処理を終了しても実行は継続されます。

s オプションを指定すると、クライアントごとにプロセスを実行し、そのクライアントとの会話が終了した時点でプロセスが終了します。

servers ファイルには、**BOOT_START** と **BOOT_END** で示されるセクションがあります。この 2 つのコメントには含まれたサービスエントリは、コマンドで **-b** フラグが指定されていれば、LNA が起動されると同時に起動されます。

ブート時に起動されるサーバのうちいずれかが失敗すると、そのサービスが要求された際に LNA で通常の手順に従って再起動されますが、**logsrv** と **nsmsrv** は例外です。

BOOT_DEBUG_OFF コメントと **BOOT_DEBUG_ON** コメントを使用して、ログを行わないようにすることができます。この 2 つのコメントは通常、ログサーバ **logsrv** の起動中にログを無効にするために使用されます。

サポートされているサーバおよびサービスは、次の表のとおりです。なお、**servers** ファイルに設定されるサービスは、インストールされている X Vision プロダクトに応じて異なります。

サービス	サーバ	s:m	ブート起動
AdmWbSvc	workbsrv	s	なし
FTrans	browesrv	s	なし
lcnsesvc	licsrv	m	なし
login	tsrv	m	あり
logsrv	logsrv	m	あり
NameSvcRpc	nsmrpcsrv	s	なし
nsmsrv	nsmsrv	m	あり
<i>odbc_service</i>	<i>odbc_server</i> (データベースに依存)	m	あり
pclist	pclist	m	あり
prtServ	printsrv	m	なし
vwauth	authsrv	m	あり

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「認証の概要」

Ina

例

```
#Vision control file for UNIX INA.
##BOOT_START
#BOOT_DEBUG_OFF
logsrv logsrv m
#BOOT_DEBUG_ON
nsmsrv nsmsrv m
vwauth authsrv m
login tsrv m -l
pclist pclist m
#BOOT_END
FTrans browesrv s
prtServ printsrv m
NameSvcRpc nsmrpcsrv s
lcnsesvc licensrv m
AdmWbSvc workbsrv s
```

sqlrsec.conf

位置

`$VISION_ETCDIR/sqlrsec.conf`

説明

`sqlrsec.conf` ファイルは、SQL-Retriever がインストールされている場合に限り使用できます。このファイルは、Security Manager 用の構成ファイルです。また、このファイルは ASCII テキストファイルで、スーパーユーザ (ルート) 特権が付与されていれば、テキストエディタを使用して変更できます。ただし、このファイルのアクセス権は変更できません。このファイルには、権限を付与したり、取り消すために使用するエントリ、さらにこのようなエントリで使用されるユーザグループ、データベース、アプリケーション、SQL ステートメントなどの定義が格納されます。グループを定義するために使用するエントリの形式は、次のとおりです。

keyword groupname = groupmember,...

keyword は、次に示すように、作成するグループに対するキーワードです。

キーワード	グループ
USERGROUP	ユーザまたはユーザグループの UNIX ユーザ名
DBGROUP	データベースまたはデータベースグループ
TABLEGROUP	データベーステーブルまたはテーブルグループ
APPGROUP	アプリケーション (アプリケーションを指定する場合の規約については、下記参照) またはアプリケーショングループ
SQLGROUP	SQL ステートメントまたは SQL ステートメントグループ

groupmembers を複数指定する場合は、次のようにカンマで区切ります。

`APPGROUP ALLAPPS = msaccess, excel, msqry32, SALESAPPS`

上記の例では、`APPGROUP` を使用して、Microsoft Access、Excel、Query、さらに `SALESAPPS` グループで定義されているアプリケーションが含まれている `ALLAPPS` という名前のアプリケーショングループを作成します。

特権を付与および取り消すためのエントリの形式は、次のとおりです。

GRANT | REVOKE *user database:reserved:owner:table:application = privilege,...*

user は、特権を付与または取り消す個々のユーザあるいはユーザグループのメンバーの UNIX ユーザ名です。*database* は UNIX データベース (データベース名は、必ずしも ODBC データソース名と同じである必要はありません) またはデータベースグループです。*owner* はデータベーステーブルの所有者です。*table* はテーブルまたはテーブルグループです。*application* は、内部名で

識別されるアプリケーションまたはアプリケーショングループです。アプリケーションの内部名を検索するには、Windowsのエクスプローラで実行可能ファイルを右クリックした後、ショートカットメニューで[プロパティ]をクリックします。[プロパティ]ダイアログボックスで、[Version]タブをクリックした後、[項目]リストの[内部名]をクリックします。Windows NT 3.51の場合は、ファイルマネージャで実行可能ファイルをクリックした後、ALT+ENTERを押します。その後、[プロパティ]ダイアログボックスで、[バージョン情報]リストの[内部名]をクリックします。また、アプリケーションに内部名が付けられていない場合は、アプリケーションの実行可能ファイル(拡張子は除く)を指定できます。この方式では、別のアプリケーションがファイル名を変更することで、この方法で指定されるアプリケーションを装う恐れがあるため、安全性は低くなります。*privilege*は特権を付与または取り消すSQLステートメントあるいはSQLステートメントグループです。特権を複数指定する場合は、カンマで区切ります。なお、これらのエントリのフィールドは、`sqlrsec.conf`で定義されているグループまたはグループメンバーである必要はありません。

どのフィールドでも、ワイルドカードとしてALLまたはアスタリスク文字(*)を使用できます。[*privilege*]フィールドにALLを指定した場合、Vision ODBCドライバでサポートされるすべてのSQLステートメントのあらかじめ定義されている次のグループが使用されます。

SELECT	CREATE INDEX
INSERT	CREATE VIEW
UPDATE	DROP TABLE
DELETE	DROP VIEW
GRANT	DROP INDEX
REVOKE	EXECUTE PROCEDURE
CREATE TABLE	ALTER TABLE

注

デモンストレーションデータベースでの Security Manager の使い方を示すサンプルファイル `sqlrsec.sample` は、`sqlrsec.conf` という名前で保存し、必要に合わせて変更してください。ただし、これにはスーパーユーザ (ルート) 特権が必要です。

関連項目

「Vision ODBC Driver」の章の「Security Manager」

例

次の例では、SALES グループのメンバーにデモンストレーションデータベースのすべてのテーブルに対する照会特権を付与します。次に、この特権を特定のテーブルに対して取り消します。

```
GRANT SALES scodemo:ALL:ALL:ALL:gold32 = SELECT
REVOKE lee scodemo:ALL:ALL:sales:gold32 = SELECT
REVOKE SALES scodemo:ALL:ALL:nominal:gold32 = SELECT
```

termio

位置

`$VISION_ETCDIR/termio`

`$VISION_ETCDIR/termios`

説明

termio (または一部の UNIX システムの場合は **termios**) ファイルはバイナリ形式の端末構成ファイルで、タスクサーバ **tsrv** で読み取られます。このファイルには、UNIX の **stty** コマンドで構成される **tty** の現在の設定値が格納されます。**termio** ファイルによって、タスクサーバで提供されているデフォルトの値は無効になります。**termio** ファイルが提供されるのは、標準的な UNIX 端末セッションで提供されるような標準的な **tty** 設定を持つ端末エミュレータがインストールされている場合です。

一部の UNIX システムでは、さらに **winsize** ファイルが作成され、ウィンドウサイズの指定が格納されます。

関連項目

tsrv

tsrv.rc

位置

`$VISION_ETCDIR/tsrv.rc`

説明

`tsrv.rc` ファイルは、タスクサーバ `tsrv` 用の構成ファイルです。このファイルは、ASCII テキストファイルで、スーパーユーザ (ルート) 特権が付与されていれば、テキストエディタを使用して変更できます。ファイル内のセクションの先頭には `#!` が付けられます。ただし、現在提供されているセクションは `#!UTMP-USERS` だけで、このセクションにはデフォルトのアクションとして各端末接続に対して UNIX の `utmp` ファイルにエントリを作成するように設定されているユーザのリストが格納されます。エントリの形式は次のとおりです。

```
#!UTMP-USERS
username
```

`username` は、UNIX ユーザ名です。

関連項目

`tsrv`

例

```
#Tsrvconfigurationfile.
# Sections in this file are introduced by `#!'
# Other lines beginning with # are comments.
# Users whose names appear (one name per line)
# in the UTMP-USERS section
# will have a utmp entry created for every login session.
#!UTMP-USERS
root
stevel
```

vision.conf

位置

/etc/vision.conf

説明

vision.conf ファイルは Vision のマスタ構成ファイルです。このファイルは、ASCII テキストファイルで、スーパーユーザ (ルート) 特権が付与されていれば、テキストエディタを使用して変更できます。エントリの形式は次のとおりです。

variable=value

variable は、次に示す変数名です。

変数	説明
VISION_ADMINDIR	SuperVision ファイルの位置
VISION_BINDIR	実行可能ファイルの位置
VISION_DEBUGDIR	デバッグログファイルの位置
VISION_DIR	Vision のベースディレクトリ
VISION_ETCDIR	各種ファイルの位置
VISION_EVENTDIR	イベントログファイルの位置
VISION_EVENTLEVEL	必要なイベントログのレベル
VISION_LOGLIMIT	ログファイル内の最大行数
VISION_MAXSESSIONS	USP 端末接続の最大数
VISION_NAMESVC	マスタネームサーバのアドレス
VISION_NS_GATEWAY	他のネットワーク上の NSM のアドレス
VISION_TCLDIR	Tcl ファイルの位置
VISION_TCLSCRIPTDIR	Tcl スクリプトファイルの位置
VISION_TCLINTERP	Tcl インタプリタプログラムのパス名
VISION_TMPDIR	一時ファイルの位置

注

ユーザはホームディレクトリに **.vision.conf** ファイルを作成できます。この場合、このファイルの設定値によって、広域 **vision.conf** ファイルの設定値が無効になります。

例

```
#Visionmasterconfigurationfile.
VISION_DIR=/usr/local/vision
VISION_BINDIR=$VISION_DIR/bin
VISION_ETCDIR=$VISION_DIR/etc
VISION_ADMINDIR=$VISION_DIR/admin
VISION_TMPDIR=$VISION_DIR/log
VISION_DEBUGDIR=$VISION_TMPDIR
VISION_EVENTDIR=$VISION_TMPDIR
VISION_EVENTLEVEL=1
VISION_MAXSESSIONS=8
VISION_LOGLIMIT=5000
VISION_TCLDIR=$VISION_DIR/tcl
VISION_TCLSCRIPTDIR=$VISION_TCLDIR/admtasks
VISION_TCLINTERP=$VISION_BINDIR/admsh
VISION_NAMESVC=poptart.16024
VISION_NS_GATEWAY=pooohbear.16025
```

visionpriv

位置

`$VISION_ETCDIR/visionpriv`

説明

`visionpriv` ファイルは認証データベースで、ユーザ名とアクセス権が格納されます。このファイルは、ASCII テキストファイルで、スーパーユーザ(ルート) 特権が付与されていれば、テキストエディタを使用して変更できます。ただし、このファイルのアクセス権は変更できません。認証データベースのユーザエントリの形式は次のとおりです。

`user:username:permissions`

`permissions` は、次に示すようにアクセス権を指定する 10 進数です。

数字	意味
0	通常ユーザ (デフォルト)
1	管理者

次のエントリ例では、ユーザ "stevel" に管理者のアクセス権を付与します。

```
user:stevel:1
```

Vision ソフトウェアを最初に UNIX システムにインストールしたときには、認証データベースにはユーザエントリは含まれず、サーバ特権情報が格納されます。サーバ特権情報は変更できず、特殊なアクセス権を必要とするユーザに対してエントリを追加するだけです。ファイル内にはユーザの場合は、デフォルトの値 (0) が使用されます。

注

ファイルを変更した時点で認証サーバがすでに実行されている場合は、UNIX システムプロンプトから次のコマンドを入力して、再度情報を読み取らせる必要があります。

```
authsrv-r
```

このコマンドを実行するには、スーパーバイザ特権が必要です。

関連項目

「Vision コミュニケーションアーキテクチャ」の章の「認証の概要」

`authsrv`

vwdebuglog

位置

`SVISION_DEBUGDIR/vwdebuglog`

説明

vwdebuglog ファイルはログサーバ **logsrv** によって作成されるログファイルで、トラブルシューティングに有効な情報が格納されます。このファイルは、ASCII テキストファイルで、テキストエディタを使用すれば参照できます。エントリの形式は次のとおりです。

date time process_id program_name: function: description

注

デフォルトでは、ログファイル内のエントリの最大数は 5000 個です。この限度は **VISION_LOGLIMIT** 変数で指定されます。この限度を超えると、現在のログファイル名が変更され、新しいファイルが作成されます。

関連項目

logsrv

例

デバッグログの抜粋を次に示します。

```
Aug22 13:47:54: [12668] nsmsrv: findNS (bcast=0)
Aug22 13:47:54: [12668] nsmsrv: presetNS (1073760656, 0)
Aug22 13:47:54: [12668] nsmsrv: presetNS: callingGetInetAddr (poptart)
Aug22 13:47:54: [12668] nsmsrv: presetNS: returning 192.86.169.164 (16024)
Aug 22 13:47:54: [27169] nsmsrv: DoLookup: NS_LOOKUP ok
'¥¥jamesw¥adminsvc2' 19'
Aug 22 13:47:54: [12668] nsmsrv: usptcp_LookupUspName:
(¥¥jamesw¥adminsvc2, 192.)
Aug 22 13:47:54: [12668] nsmsrv: ForwardToAgent: notifying client
'¥¥jamesw¥admk
Aug 22 13:47:55: [27169] nsmsrv: DoLookup: NS_LOOKUP ok '¥¥poptart¥'
192.86.169
```

vweventlog

位置

`$VISION_EVENTDIR/vweventlog`

説明

`vweventlog` ファイルは、ログサーバ `logsrv` コマンドによって作成されるログファイルで、システム管理者にとって重要なイベントがすべて記録されます。このファイルは ASCII テキストファイルで、テキストエディタを使用すれば参照できます。エントリの形式は次のとおりです。

date time process_id program_name: function: description

注

デフォルトでは、ログファイル内のエントリの最大数は 5000 個です。この限度は `VISION_LOGLIMIT` 変数で指定されます。この限度を超えると、現在のログファイル名が変更され、新しいファイルが作成されます。

関連項目

`logsrv`

例

イベントログの抜粋を次に示します。

```
Aug 14 12:23:52: [4133] workbsrv: CreateItem: CreatedPCRickyd
Aug 14 12:24:05: [4133] workbsrv: SetGroupMembership: CreatedGroupSales
Aug 14 12:24:05: [4133] workbsrv: SetGroupMembership: Added PCRickyd to group Sales
Aug 14 12:24:22: [4133] workbsrv: CreateItem: CreatedPCSigmundd
Aug 14 12:24:34: [4133] workbsrv: BuildIndex: Indexesbuilt, newversionnumber is 808398979
```

winsize

位置

`$VISION_ETCDIR/winsize`

説明

`winsize` ファイルはバイナリ形式の端末構成ファイルで、タスクサーバ `tsrv` で読み取られます。このファイルは一部の UNIX システムで使用され、UNIX の `stty` コマンドで構成される端末の行と列の設定値が格納されます (一部の UNIX システムでは、この情報は `termio` ファイルに含まれるため、`winsize` ファイルが不要になります)。

関連項目

`tsrv`

`termio`

付録 D

ComScript

このセクションの内容は、次のとおりです。

- ComScript の概要
- ComScript ファイル
- 提供されている ComScript ファイル
- スクリプト言語の概要
- コマンド構造
- ユーザ定義の関数
- パラメータ
- 照合文字列
- コメント
- ComScript 関数の概要
- ComScript ファイル例の概要
- ComScript ファイルの開発
- コネクションモニタを使用したテスト
- ComScript ファイルの生成
- ダイアルバック

ComScript の概要

リモートホストシステムとリンクを確立するためには、Vision コミュニケーションアーキテクチャで物理的な接続を開き、ホストのオペレーティングシステムにログインする必要があります。さらに、場合によっては、USP プロキシプロセスである **usputppxy** などのリモートプログラムを実行する必要があります。次に示すこのプロセスの部分の一部または全部は、“ComScript” と呼ばれる言語で作成された Vision コミュニケーションスクリプトファイルで制御されます。

- ホストとの接続
- ログイン
- コマンドの実行

USP 接続では、このスクリプトはコマンドの実行に使用されます。USP 以外の接続では、ログインとコマンドの実行に使用されます。また、TAPI 以外のモデム接続では、スクリプトはホストとの接続、ログイン、コマンドの実行に使用されます。スクリプトファイルでは、必要に応じて、一連の文字列をリモートホストまたは端末サーバに送信します。そして、指定された文字列と照合することで、リモート装置からの応答を監視して、接続手順の次のフェーズに進んでも問題ないかどうか判断します。ステップが失敗した場合には、該当するエラーメッセージを発行できます。

スクリプトファイルはまた、接続が終了した際に切断操作を行うためにも使用されます。

ComScript ファイル

Vision コミュニケーションスクリプトファイルはテキストファイルで、Windows のメモ帳などの PC ベースのエディタで作成できます。使用する前に、ファイルをコンパイルする必要はありません。

スクリプトファイル名は *filename.scr* 形式で設定し、通常 **C:¥Program Files¥Common Files¥Vision¥Scripts** に格納されます。いくつかのスクリプトファイルがあらかじめ提供されているので、これをベースにして固有のスクリプトファイルを作成できます。また、コネクションモニタを使用して、まったく新しくスクリプトファイルを作成することも可能です (本章の「ComScript ファイルの生成」を参照)。

接続プロセスを制御するために使用される特殊なスクリプトファイルは、ホストまたはホストで使用されるトランスポートの設定によって異なります。

トランスポートを構成する場合、デフォルトの設定ではそのトランスポートで使用されるプロトコルに合ったスクリプトファイルが使用されます。(本章の「提供されている ComScript ファイル」を参照)。Vision コミュニケーションアーキテクチャの標準的なトランスポートで使用されるスクリプトの詳細については、下表を参照してください。

トランスポート	スクリプト
Local	なし
Modem	Unixsl.scr
Serial-com1	Unixsl.scr
SPX-Unix	Unixipx.scr
TCP-Unix	Unix.scr
USP-TCP	Unixusp.scr
Xtrace	なし

注意 プロキシプロセスを介して USP サービスを実行するように新しいトランスポートが作成されている場合、元になっているトランスポート用に設定されたスクリプトを使用してプロキシホストにアクセスし、**Unixusp.scr** はコマンドを実行したいホストに使用されます。

デフォルトの設定では、ホストはトランスポートに指定されたスクリプトオプションを使用します。ただし、ホストに対してスクリプトオプションを指定すると、ホストで使用されるトランスポートに対して指定されているスクリプトオプションは無効になります。たとえば、トランスポートで使用されるスクリプトと異なるスクリプトを使用するには、該当するホストのプロパティ設定ダイアログボックスで [スクリプト] タブをクリックし、[スクリプトオプション] をクリックした後、フォルダを参照して目的のファイルを設定します。

提供されている ComScript ファイル

セットアッププログラムにより、次の ComScript ファイルが **¥Program Files¥Common Files¥Vision¥Scripts** フォルダに設定されます。

ファイル	説明
Unix.scr	TCP/IP ネットワークを介したりリモート UNIX ホストへの接続プロセスを制御します。
Unixipx.scr	IPX/SPX ネットワークを介したりリモート UNIX ホストへの接続プロセスを制御します。
Unixusp.scr	USP 接続に対して TCP/IP ネットワークを介したりリモート UNIX ホストへの接続プロセスを制御します。
Unixsl.scr	シリアル回線を介したりリモート UNIX ホストへの接続プロセスを制御し、TCP/IP ネットワーク用に X 環境を設定します。
Unixslsc.scr	シリアル回線を介したりリモート SCO UNIX ホストへの接続プロセスを制御し、TCP/IP ネットワーク用に X 環境を設定します。

スクリプト言語の概要

Vision コミュニケーションのスクリプト言語は、システムで定義されているいくつかの関数と簡単なコマンド構造から構成されます。演算子は使用できないため、ほとんどの関数では成功したかどうかではなく失敗したかどうかをテストするための補助関数が設定されています。

コマンド構造

このセクションの内容は、次のとおりです。

- IF...THEN...ELSE ステートメント
- LOOP ステートメント

IF...THEN...ELSE ステートメント

この構造は、条件付きで実行するためのものです。IF...THEN...ELSE ステートメントを使用するための構文は、次のとおりです。

```
IF...THEN  
    [ELSE]  
ENDIF
```

IF ステートメントの後には、ユーザ定義の関数呼び出しまたは照合操作が続きます。ComScript では演算子がサポートされていないため、正の応答と負の応答のどちらをテストするのかに応じて、システムで定義されている適切な関数を使用して、条件を指定する必要があります。たとえば、ユーザ定義の関数を呼び出して、成功した場合に THEN 操作を行うには、**CALL** 関数を使用して関数を呼び出します。また、関数が失敗した場合に THEN 操作を行うには、**NOTCALL** 関数を使用して関数を呼び出します。同じような関係が **MATCH** と **NOMATCH**、および **REMATCH** と **NOREMATCH** にも存在します。

LOOP ステートメント

LOOP 構造は、処理を繰り返し行う場合に使用します。LOOP ステートメントを使用するための構文は、次のとおりです。

```
LOOP  
    [EXITLOOP]  
ENDLOOP
```

理論上では EXITLOOP ステートメントは指定しても指定しなくてもかまいませんが、無限ループにならないように、EXITLOOP ステートメントを 1 つは指定してください。

次の関数は、この 2 つの構造の使用例を示したものです。この関数によって、接続プロセス時にユーザは新しいパスワードを入力できます。

```

FUNC `NEWPASSWORD`

TRACE( `Entering new password` )
PASSWORD( `Enter new password`, 8 )
STATUS( `Script`, `Entering New Password`, SL_CANCEL )
SEND( `¥p08¥r` )
MATCH( 10, `New Password`, `ialup`, `erm`, `\$`, `%`, `#`,
`ncorrect` )

LOOP
  IF REMATCH( `New Password` ) THEN
    PASSWORD( `Re-enter new password`, 8 )
    SEND( `¥p08¥r` )
    MATCH( 10, `New Password`, `ialup`, `erm`, `\$`, `%`, `#`,
`ncorrecti` )
  ELSE
    IF REMATCH( `ncorrect` ) THEN
      TRACE( `Failed to enter new password correctly` )
      IF NOTCALL( `SETPASSWORDERROR` ) THEN
        RETURN( `FALSE` )
      ENDIF
    ENDIF
  EXITLOOP
ENDIF
ENDLOOP

ENDFUNC

```

ユーザ定義の関数

各スクリプトは、一連のユーザ定義の関数から構成されます。それぞれの関数は **FUNC** “*function-name*” 形式 (*function-name* は、スクリプト全体でこの関数を参照するために使用される名前) で始まり、**ENDFUNC** で終わります。スクリプトにどのユーザ定義の関数が含まれるかは、そのスクリプトを呼び出してどのタスクを行うかによって異なります。コマンドを実行する場合は、**DOEXEC** および **CLOSEEXEC** という名前の関数が必要です。また、ログインの場合は、**DOLOGIN** 関数と **CLOSELOGIN** 関数が必要になります。さらに、ホストに接続する場合は、**DODIAL** 関数と **CLOSEDIAL** 関数が必要です。それぞれの前者の関数は、その関数に関連する接続プロセスの特定の部分を制御するスクリプトが実行されると、自動的に呼び出されます。また、後者の関数は切断操作時に呼び出されます。これ以外にもユーザ定義の関数を使用して、スクリプトの構造を単純化したり、コードを繰り返す必要を解消できます。

パラメータ

Vision コミュニケーションのスク립ト内では、p00 から p19 までの最大 20 個までのパラメータを使用できます。パラメータの値は、接続プロセス時に **DIALOG** 関数または **PASSWORD** 関数を呼び出して、ユーザに必要な情報を指定するよう求めることで設定されます。ただし、パラメータ p00 から p09 には、次に示すようにデフォルトの値が自動的に割り当てられます。これらのパラメータ値は誤って変更しないように、注意してください。

パラメータ	デフォルトの値
p00	ホスト名
p01	ホスト定義に基づくユーザ名
p02	ホスト定義に基づくパスワード
p03	ホスト定義に基づく電話番号
p04	リモートホストに送信するコマンド
p05	端末タイプ
p06	リセット文字列
p07	PC のネットワークノード名
p08	PC の X ディスプレイ名
p09	PC の X ディスプレイ番号

照合文字列

Vision コミュニケーションのスク립トでは、文字列をモデム、端末サーバ、ホストのいずれかに送信し、返された文字列を評価して、該当する応答を決定します。

リモート出力を評価するために、4 つの関数を使用できます。各関数ではパラメータとして最大 128 文字の文字列リストが設定され、リモート出力された文字列と照合されます。**MATCH** 関数と **NOMATCH** 関数では、指定されたタイムアウト時間内に出力が届くと、直ちにその出力をテストします。**MATCH** では、着信データが関数に対して指定されている文字列のいずれかと一致した場合に **TRUE** を返し、**NOMATCH** では一致する文字列が見つからなかった場合に **TRUE** を返します。

リモート出力はまたバッファにも格納されるため、**REMATCH** 関数および **NOREMATCH** 関数を使用して再評価できます。これらの関数は **MATCH** および **NOMATCH** と同じように使用して、一致する文字列が見つからなかったかどうかをテストし、該当する応答を生成できます。

注意 DELAY 関数または SEND 関数を呼び出すと、バッファがフラッシュされます。

次の例では、これらの関数を使用して、モデム接続が行われたかどうかを判断し、接続が失敗した場合その原因に応じて該当するエラーメッセージを生成します。

```
; Dial the phone no
TRACE( `Dialing the phone number` )
SEND( `ATDT%p03#r` )
IF MATCH( 30, `CONNECT`, `BUSY`, `NO CARRIER`, `NO DIALTONE` )
THEN
    IF NOREMATCH( `CONNECT` ) THEN
        IF REMATCH( `BUSY` ) THEN
            TRACE( `BUSY` )
            ERROR( `Line is busy` )
        ELSE IF REMATCH( `NO DIALTONE` ) THEN
            TRACE( `No Dialtone` )
            ERROR( `No Dialtone` )
        ELSE
            TRACE( `No Carrier` )
            ERROR( `No Carrier` )
        ENDIF
    ENDIF
    RETURN( `FALSE` )
ENDIF
ENDIF
```

コメント

Vision コミュニケーションスクリプト内にコメント行を指定できます。コメントテキストはセミコロン (;) で示され、この文字から行末までのテキストはスクリプトの実行時には無視されます。

ComScript 関数の概要

ComScript では、次のシステム定義関数を使用できます。各関数の使い方については、以降のページで詳しく説明します。

BREAK	“ブレーク”シグナルを送信します。
CALL	関数を呼び出し、その呼び出しが成功したかどうかテストします。
DELAY	遅延を設定します。
DIALOG	ダイアログボックスを表示します。
ERROR	エラーメッセージを設定します。
EXECUTE	Windows または DOS プログラムを実行します。
ISSET	パラメータまたは文字列に値が割り当てられているかどうかを調べます。
LONGBREAK	“ロングブレーク”シグナルを送信します。
MATCH	指定された文字列と受信データを照合します。
NOMATCH	受信データに指定された文字列の一致がないかどうかを調べます。
NOREMATCH	バッファ内のデータに指定された文字列の一致がないかどうかを調べます。
NOTCALL	関数を呼び出し、その呼び出しが失敗したかどうかテストします。
NOTSET	パラメータまたは文字列に値が割り当てられていないかどうかを調べます。
PASSWORD	ダイアログボックスを表示します。テキストボックスに表示されるテキストは、パスワードと同じように、マスクされます。
REMATCH	指定された文字列とバッファ内のデータを照合します。
RETURN	該当する戻り値を設定して、関数呼び出しから戻ります。
SAVEPASSWORD	ホスト定義内のパスワードを更新します。
SEND	文字列を送信します。
STATUS	ステータスアイコンとメッセージを設定します。
TRACE	トレースメッセージを設定します。

BREAK 関数

構文

`BREAK()`

説明

`BREAK` 関数は、通信 “ブレーク” シグナルをホストに送信します。

関連項目

`LONGBREAK` (“ロングブレーク” シグナルを送信する)

CALL 関数

構文

```
CALL(string)
```

説明

CALL 関数は、スクリプトのいずれかの位置に定義されている関数を呼び出します。

string は、呼び出す関数名です。

注

デフォルトでは、関数が呼び出されると、値 TRUE が返され、呼び出しが正常終了したことが示されます。

関連項目

NOTCALL (関数を呼び出して、呼び出しが失敗したかどうかをテストする)

RETURN (関数が失敗した場合に、FALSE を返す)

例

```
IF NOTCALL( `LOGIN` ) THEN
    STATUS( `Script`, `Failed to login, resetting`, SL_NOCANCEL )
    CALL( `HANGUP` )
    CALL( `RESET` )
    RETURN( `FALSE` )
ENDIF
```

DELAY 関数

構文

`DELAY(time)`

説明

time に指定された期間遅延します。

time には、10 分の 1 秒単位で指定します。

注

DELAY 関数を呼び出すと、バッファ内のデータはフラッシュされます。

例

```
TRACE( `Send CR to get Login Prompt` )  
DELAY( 10 )  
SEND( `¥r¥r` )  
TRACE( `Connected to remote modem` )
```

DIALOG 関数

構文

```
DIALOG('string', parameter)
```

解説

DIALOG 関数は、ダイアログボックスを表示して、ユーザに情報を指定するように求めます。ここでユーザが指定した情報は、指定されているパラメータの値として割り当てられます。

string はダイアログボックスにプロンプトとして表示するテキスト文字列で、*parameter* はユーザが入力した値が割り当てられるパラメータの数値識別子です。

注

DIALOG 関数で生成されるダイアログボックスのタイトルは、接続しているホスト名になります。

String には、最大 128 文字まで指定できます。

現在パラメータに値が割り当てられている場合には、この値が自動的にダイアログボックスに表示されます。パラメータ p00-p09 では、代替値が割り当てられていなければ、デフォルトの値が表示されます。

DIALOG 関数を使用して作成したダイアログボックスは、ホストに対して [プロンプト] ボックスがチェックされているかどうかに関係なく、接続プロセス時に表示されます。

関連項目

PASSWORD (ユーザが入力したテキストがアスタリスクでマスクされるダイアログボックスを生成する)

例

次のコマンドでは、ユーザにリモートホスト名または IP アドレスの指定を求めるダイアログボックスを生成します。入力した文字列はパラメータ p08 に割り当てられます。

```
DIALOG( `Enter machine name or IP address`, 8 )
```

ERROR 関数

構文

```
Error('string')
```

解説

ERROR 関数は、ある特定の時点で接続プロセスが失敗した場合に表示されるエラーメッセージのテキストを設定します。

string はエラーメッセージのテキストです。

注

接続プロセス時にエラーが発生した場合、メッセージボックスが表示され、システム定義の該当するメッセージが示されます。スクリプト内の接続が失敗した箇所で **ERROR** 関数が呼び出されている場合、システムメッセージに続いて *string* 値として指定されているテキストが括弧内に表示されます。**ERROR** 関数が呼び出されていない場合は、メッセージボックスにシステムメッセージだけが表示されます。

String には、最大 128 文字まで指定できます。

例

```
ERROR( `Login incorrect` )
```

EXECUTE 関数

構文

```
EXECUTE('command', 'show-parameter')
```

説明

EXECUTE 関数は、Vision コミュニケーションのスクリプト内から DOS または Windows のプログラムを実行します。

Command は実行するコマンドです。show-parameter は次に示す定数値のいずれかで、プログラムをどのように表示するのかを決定するために使用されます。コマンドを指定する際、プログラムが DOS PATH 内にないフォルダに格納されている場合は、フルパスを指定してください。この際、パス内に ¥ 文字がある場合は、¥ をもう 1 つ付けて保護してください。こうしないと、特殊文字のインジケータとして扱われ、コマンドは失敗します。

パラメータ	値
SL_HIDE	ウィンドウが非表示のままプログラムが実行されます。フォーカスは変わりません。
SL_NORMAL	プログラムが通常のサイズと位置で実行されます。プログラムウィンドウにフォーカスが移ります。
SL_MAXIMIZE	ウィンドウが最大表示され、プログラムが実行されます。プログラムウィンドウにフォーカスが移ります。
SL_MINIMIZE	プログラムがアイコンとして実行されます。アイコンにフォーカスが移ります。
SL_RESTORE	プログラムが通常のサイズと位置で実行されます。プログラムウィンドウにフォーカスが移ります。
SL_SHOW	プログラムが通常のサイズと位置で実行されます。プログラムウィンドウにフォーカスが移ります。
SL_SHOWMAXIMIZED	ウィンドウが最大表示され、プログラムが実行されます。プログラムウィンドウにフォーカスが移ります。
SL_SHOWMINIMIZED	プログラムがアイコンとして実行されます。アイコンにフォーカスが移ります。
SL_SHOWMINNOACTIVATE	プログラムがアイコンとして実行されます。フォーカスは変わりません。

パラメータ	値
SL_SHOWNA	プログラムが通常のサイズと位置で実行されます。フォーカスは変わりません。
SL_SHOWNOACTIVATE	プログラムが通常のサイズと位置で実行されます。フォーカスは変わりません。
SL_SHOWNORMAL	プログラムが通常のサイズと位置で実行されます。プログラムウィンドウにフォーカスが移ります。

関連項目

SEND (リモートホストでコマンドを実行する)

例

次の関数では、ホストエクスプローラを実行します。

```
EXECUTE (`C:¥¥PROGRAMS¥¥VISION¥¥hostexp.exe`, `SL_SHOW`)
```

ISSET 関数

構文

```
ISSET('string', parameter)
```

説明

指定された文字列またはパラメータに値が割り当てられている場合に、TRUE を返します。

string は、DETACH または TTY のいずれかのテキスト文字列です。DETACH を指定すると、Vision コミュニケーションアーキテクチャではコマンドの実行後ホストから生成された出力が必要かどうかを確認されます。TTY を指定した場合には、端末接続なのか端末以外の接続なのかが確認されます。これ以外に *string* に指定されている値は無視されます。

parameter はパラメータの数値識別子です。

関連項目

NOTSET (指定された文字列またはパラメータに値が割り当てられていないかどうかを調べる)

例

次の例では、ISSET 関数を使用して、文字列 TTY に値が割り当てられているかどうかを確認します。この文字列に値が割り当てられている場合は、端末接続であり、コマンドの実行により生成されるエコーをすべて処理する必要があります。

```
IF ISSET( `TTY` ) THEN
    ; consume the echo
    MATCH( 5, `¥n` )
ENDIF
```

LONGBREAK 関数

構文

LONGBREAK()

説明

LONGBREAK 関数は、通信“ロングブレーク”シグナルをホストに送信します。

関連項目

BREAK (“ブレーク”シグナルを送信する)

MATCH 関数

構文

`MATCH (timeout, 'string' [, 'string'])`

説明

MATCH 関数では、*string* のいずれかの値と一致する文字列を *timeout* で指定された時間内にリモートホストから受信した場合に、TRUE を返します。

timeout には秒単位で指定します。

注

string パラメータは最大 9 つ指定でき、1 つのパラメータには最大 128 文字指定できます。

String には、次の特殊文字も指定できます。

文字	値
¥r	キャリッジリターン
¥n	復帰改行
¥t	タブ
¥ <i>nnn</i>	<i>nnn</i> は文字の 8 進数値
¥ <i>pnn</i>	<i>nn</i> は接続パラメータの数値識別子です。パラメータを 1 桁の数字から構成する場合は、¥p03 というように、先行ゼロを付けます。

注意 ¥¥ と指定すれば、¥文字も指定できます。

関連項目

NOMATCH (一致が見つからないかどうかを調べる)

REMATCH (バッファ内のデータに一致があるかどうかを調べる)

NOREMATCH (バッファ内のデータに一致する文字列がないかどうかを調べる)

例

次の例では、**MATCH** 関数を使用して、新しいパスワードを入力した際に返されるデータを評価します。

```
FUNC `NEWPASSWORD`

TRACE( `Entering new password` )
PASSWORD( `Enter new password`, 8 )
STATUS( `Script`, `Entering New Password`, SL_CANCEL )
SEND( `p08` )
MATCH( 10, `New Password`, `ialup`, `erm`, `$`, `%`, `#`, `ncorrect` )

LOOP
  IF REMATCH( `New Password` ) THEN
    PASSWORD( `Re-enter new password`, 8 )
    SEND( `p08` )
    MATCH( 10, `New Password`, `ialup`, `erm`, `$`, `%`, `#`,
`ncorrect` )
  ELSE
    IF REMATCH( `ncorrect` ) THEN
      TRACE( `Failed to enter new password correctly` )
      IF NOTCALL( `SETPASSWORDERROR` ) THEN
        RETURN( `FALSE` )
      ENDIF
    ENDIF
  EXITLOOP
ENDIF
ENDLOOP

ENDFUNC
```

NOMATCH 関数

構文

`NOMATCH(timeout, 'string' [, 'string'])`

説明

NOMATCH 関数では、*string* のいずれかの値と一致する文字列が *timeout* で指定された時間内にリモートホストから受信されない場合に、TRUE を返します。

timeout には、秒単位で指定します。

注

string パラメータは最大 9 つ指定でき、1 つのパラメータには最大 128 文字指定できます。

String には、次の特殊文字も指定できます。

文字	値
¥r	キャリッジリターン
¥n	復帰改行
¥t	タブ
¥ <i>nnn</i>	<i>nnn</i> は文字の 8 進数値
¥ <i>pnn</i>	<i>nn</i> は接続パラメータの数値識別子です。パラメータを 1 桁の数字から構成する場合は、¥p03 というように、先行ゼロを付けます。

注意 ¥¥ と指定すれば、¥ 文字も指定できます。

関連項目

MATCH (一致文字列が見つかるかどうかを調べる)

REMATCH (バッファ内のデータに一致文字列があるかどうかを調べる)

NOREMATCH (バッファ内のデータに一致文字列がないかどうかを調べる)

例

次の例では、**NOMATCH** 関数を使用して、モデムが応答しないことを確認します。

```
IF NOMATCH( 5, `OK` ) THEN
    TRACE( `No response from modem` )
    ERROR( `No response from modem` )
    RETURN( `FALSE` )
ENDIF
```

NOREMATCH 関数

構文

NOREMATCH (*string* [, *string*])

説明

NOREMATCH 関数では、*string* のいずれかの値と一致する文字列がリモートホストから受信したデータが格納されているバッファに存在しない場合に、TRUE を返します。

注

string パラメータは最大 10 個指定でき、1 つのパラメータには最大 128 文字指定できます。

string には、次の特殊文字も指定できます。

文字	値
¥r	キャリッジリターン
¥n	復帰改行
¥t	タブ
¥ <i>nnn</i>	<i>nnn</i> は文字の 8 進数値
¥ <i>pnn</i>	<i>nn</i> は接続パラメータの数値識別子です。パラメータを 1 桁の数字から構成する場合は、¥p03 というように、先行ゼロを付けます。

注意 ¥* と指定すれば、¥ 文字も指定できます。

DELAY 関数または SEND 関数を呼び出すと、バッファ内のデータはフラッシュされます。

関連項目

MATCH (一致する文字列が見つかるかどうかを調べる)

NOMATCH (一致する文字列が見つからないかどうかを調べる)

REMATCH (バッファ内のデータに一致する文字列があるかどうかを調べる)

例

次の例では、**NOREMATCH** 関数を使用して、ログインプロセスの開始時に返されたデータにおいて一致を評価します。

```
; Dial the phone no
TRACE( `Dialing the phone number` )
SEND( `ATDT¥p03¥r` )
IF MATCH( 30, `CONNECT`, `BUSY`, `NO CARRIER`, `NO DIALTONE` ) THEN
  IF NOREMATCH( `CONNECT` ) THEN
    IF REMATCH( `BUSY` ) THEN
      TRACE( `BUSY` )
      ERROR( `Line is busy` )
    ELSE IF REMATCH( `NO DIALTONE` ) THEN
      TRACE( `No Dialtone` )
      ERROR( `No Dialtone` )
    ELSE
      TRACE( `No Carrier` )
      ERROR( `No Carrier` )
    ENDIF
  ENDIF
  RETURN( `FALSE` )
ENDIF
ENDIF
```

NOTCALL 関数

構文

```
NOTCALL (string)
```

説明

NOTCALL 関数では、スクリプトのいずれかの位置に定義されている関数呼び出しが失敗した場合に、TRUE を返します。

string は、呼び出される関数名です。

注

デフォルトでは、呼び出された関数は値 TRUE を返します。関数が失敗した場合に値 FALSE が返すようにするには、RETURN 関数を使用します。

関連項目

CALL (関数を呼び出し、呼び出しが成功したかどうかをテストする)

RETURN (関数が失敗した場合に、FALSE を返す)

例

次の例では、LOGIN 関数の呼び出しが失敗したかどうかをテストし、それに応じて適切な応答を生成します。

```
IF NOTCALL( `LOGIN` ) THEN
    STATUS( `Script`, `Failed to log in, resetting`, SL_NOCANCEL )
    CALL( `RESET` )
    RETURN( `FALSE` )
ENDIF
```

NOTSET 関数

構文

```
NOTSET('string', parameter)
```

説明

指定された文字列またはパラメータに値が割り当てられていない場合に、TRUE を返します。

string は、DETACH または TTY のいずれかのテキスト文字列です。DETACH を指定すると、Vision コミュニケーションアーキテクチャではコマンドの実行後ホストから生成された出力が必要かどうかを確認されます。TTY を指定した場合には、端末接続なのか端末以外の接続なのかを確認されます。これ以外に *string* に指定されている値は無視されます。

parameter は、パラメータの数値識別子です。

関連項目

ISSET (指定された文字列またはパラメータに値が割り当てられているかどうかを調べる)

例

次の例では、NOTSET 関数を使用して、パラメータ p02 に値が割り当てられているかどうかを確認します。パラメータ p02 には、デフォルトでパスワードが指定されます。

```
IF NOTSET( 2 ) THEN
    PASSWORD( `Enter your password:', 2 )
ENDIF
```

PASSWORD 関数

構文

```
PASSWORD('string', parameter)
```

説明

PASSWORD 関数は、ダイアログボックスを表示して、ユーザに情報を指定するように求めます。ここでユーザが指定した情報は、指定されているパラメータの値として割り当てられます。また、ユーザが入力したテキストは表示されず、各文字の代わりにアスタリスクが表示されます。

string はダイアログボックスにプロンプトとして表示するテキスト文字列で、*parameter* はユーザが入力した値が割り当てられるパラメータの数値識別子です。

注

PASSWORD 関数で生成されるダイアログボックスのタイトルは、接続しているホスト名になります。

string には、最大 128 文字まで指定できます。

現在 *parameter* に値が割り当てられている場合には、この値が自動的にダイアログボックスに表示されます。パラメータ p00-p09 では、代替値が割り当てられていなければ、デフォルトの値が表示されます。

PASSWORD 関数を使用して作成したダイアログボックスは、ホストに対して [プロンプト] ボックスがチェックされているかどうかに関係なく、接続プロセス時に表示されます。

関連項目

DIALOG (通常のダイアログボックスを生成する)

例

次のコマンドでは、ユーザに新しいパスワードの指定を求めるダイアログボックスを生成します。入力した文字列はパラメータ p10 に割り当てられません。

```
TRACE( `Entering new password` )
PASSWORD( `Enter new password`, 10 )
STATUS( `Script`, `Entering New Password`, SL_CANCEL )
SEND( `¥p10¥r` )
```

REMATCH 関数

構文

REMATCH('string' [, 'string'])

説明

REMATCH 関数では、*string* のいずれかの値と一致する文字列がリモートホストから受信したデータが格納されているバッファに存在する場合に、TRUE を返します。

注

string パラメータは最大 10 個指定でき、1 つのパラメータには最大 128 文字指定できます。

string には、次の特殊文字も指定できます。

文字	値
¥r	キャリッジリターン
¥n	復帰改行
¥t	タブ
¥nnn	nnn は文字の 8 進数値
¥pnn	nn は接続パラメータの数値識別子です。パラメータを 1 桁の数字から構成する場合は、¥p03 というように、先行ゼロを付けます。

注意 ¥* と指定すれば、¥ 文字も指定できます。

DELAY 関数または SEND 関数を呼び出すと、バッファ内のデータはフラッシュされます。

関連項目

MATCH (一致する文字列が見つかるかどうかを調べる)

NOMATCH (一致する文字列が見つからないかどうかを調べる)

NOREMATCH (バッファ内のデータに一致する文字列がないかどうかを調べる)

例

次の例では、**REMATCH** 関数を使用して、ログインプロセスの開始時に返されたデータにおいて一致を評価します。

```
TRACE( `Logging in to remote system` )
STATUS( `Script`, `Logging in to remote system`, SL_CANCEL )
IF MATCH( 10, `ogin`, `assword:`, `Old Password`, `New Password`, `Dialup`, `erm`, `$`, `%`, `#` )
THEN

; Send user name to login prompt, then match again on possible prompts.
IF REMATCH( `ogin` ) THEN
    TRACE( `Sending user name to login prompt` )
    SEND( `¥p01¥r` )
    MATCH( 10, `assword:`, `Choose`, `expired`, `Dialup`, `erm`, `$`, `%`, `#`, `ogin` )
    ; A login prompt here means login has failed.
    IF REMATCH( `ogin` ) THEN
        TRACE( `Login failed` )
        ERROR( `Login failed` )
        RETURN( `FALSE` )
    ENDIF
ENDIF
ENDIF
ENDIF
```

RETURN 関数

構文

```
RETURN ('TRUE'|'FALSE')
```

説明

RETURN 関数では、指定されている戻り値を設定して、関数呼び出しから戻ります。

注

関数にデフォルトで設定されている戻り値は TRUE です。戻り値を明示的に FALSE に設定する場合に、RETURN 関数を使用します。

関連項目

CALL (関数を呼び出し、呼び出しが成功したかどうかをテストする)

NOTCALL (関数を呼び出して、呼び出しが失敗したかどうかをテストする)

例

次の例では、RETURN 関数を使用して、戻り値を FALSE に設定します。

```
;
; SEND A DIALUP PASSWORD
;
FUNC `DIALUPPASSWORD`

TRACE( `Entering Dialup password` )
PASSWORD( `Enter Dialup password`, 9 )
SEND( `p09r` )
MATCH( 10, `erm`, `$`, `%`, `#`, `ncorrect` )

IF REMATCH( `ncorrect` ) THEN
    TRACE( `Failed to enter Dialup password correctly` )
    IF NOTCALL( `SETPASSWORDERROR` ) THEN
        RETURN( `FALSE` )
    ENDIF
ENDIF

ENDFUNC
```

SAVEPASSWORD 関数

構文

SAVEPASSWORD (*'string1'*, *'string2'*)

説明

SAVEPASSWORD 関数は、ホスト定義でパスワードを更新します。
string1 は現在のパスワードで、*string2* は新しいパスワードです。

注

SAVEPASSWORD 関数は、接続プロセス時に [プロンプト] ダイアログボックスを表示するようにホストが構成されている場合は使用できません。[プロンプト] ダイアログボックスを表示するようにホストが構成されていると、*string1* はホスト定義に現在格納されているパスワードと一致しません。

例

次の例では、SAVEPASSWORD 関数を使用して、パラメータ p02 (現在のパスワード) に割り当てられているテキスト文字列をパラメータ p09 (新しいパスワード) に割り当てられている文字列に更新します。ユーザが次にリモートホストにログインすると、新しいパスワードを求めるプロンプトに対して入力された文字列がパラメータ p02 にロードされ、現在のパスワードになります。

```
FUNC `NEWPASSWORD`

TRACE( `Entering new password` )
PASSWORD( `Enter new password`, 8 )
STATUS( `Script`, `Entering New Password`, SL_CANCEL )
SEND( `¥p08¥r` )
MATCH( 10, `New Password`, `ialup`, `erm`, `$`, `%`, `esktop`, `ncorrect`, `¥033` )

LOOP
  IF REMATCH( `New Password` ) THEN
    IF NOTSET( 8 ) THEN
      PASSWORD( `Re-enter new password`, 8 )
      SEND( `¥p08¥r` )
    ELSE
      PASSWORD( `Re-enter new password`, 9 )
      SEND( `¥p09¥r` )
    END IF
    MATCH( 10, `New Password`, `ialup`, `erm`, `$`, `%`, `esktop`, `ncorrect` )
  ELSE
    IF REMATCH( `ncorrect` ) THEN
      TRACE( `Failed to enter new password correctly` )
      IF NOTCALL( `SETPASSWORDERROR` ) THEN
        RETURN( `FALSE` )
      ENDIF
    ELSE
      SAVEPASSWORD( `¥p02`, `¥p09` )
    ENDIF
  EXITLOOP
  ENDIF
ENDLOOP

ENDFUNC
```

SEND 関数

構文

```
SEND('string')
```

説明

SEND 関数は、モデム、端末サーバ、リモートホストのいずれかに文字列を送信します。

注

string には、次の特殊文字も指定できます。

文字	値
¥r	キャリッジリターン
¥n	復帰改行
¥t	タブ
¥nnn	nnn は文字の 8 進数値
¥pnn	nn は接続パラメータの数値識別子です。パラメータを 1 桁の数字から構成する場合は、¥p03 というように、先行ゼロを付けます。

注意 ¥# と指定すれば、¥ 文字も指定できます。

SEND 関数を呼び出すと、バッファ内のデータはフラッシュされます。

関連項目

EXECUTE (DOS または Windows のプログラムを実行する)

例

次の例では、キャリッジリターンを 2 つリモートホストに送信して、ログインプロンプトを取り出します。

```
TRACE( `Send CR to get Login Prompt` )
DELAY( 10 )
SEND( `¥r¥r` )
TRACE( `Connected to remote host` )
```

STATUS 関数

構文

STATUS ('icon', 'string', SL_CANCEL|SL_NOCANCEL)

説明

STATUS 関数は、接続時にステータスボックスに表示するアイコンとテキストを指定します。

icon は Vision コミュニケーションアーキテクチャで提供されているアイコンを示す識別子で、*string* はステータスボックスに表示するテキストです。

SL_CANCEL パラメータまたは SL_NOCANCEL パラメータによって、[キャンセル] ボタンをアクティブにするかグレー表示にするのかが決まります。

注

次のアイコンを使用できます。

アイコン	識別子
------	-----



Fax



Go



Modem



Script

string には、次の特殊文字も含めて、最大 128 文字まで指定できます。

文字	値
¥r	キャリッジリターン
¥n	復帰改行
¥t	タブ
¥ <i>nnn</i>	<i>nnn</i> は文字の 8 進数値
¥ <i>pnn</i>	<i>nn</i> は接続パラメータの数値識別子です。パラメータを 1 桁の数字から構成する場合は、¥p03 というように、先行ゼロを付けます。

注意 ¥¥ と指定すれば、¥ 文字も指定できます。

取り消しパラメータとして、次の設定を使用できます。

パラメータ	値
SL_CANCEL	[キャンセル] ボタンをアクティブにする
SL_NOCANCEL	[キャンセル] ボタンをグレー表示にする

関連項目

TRACE ([Trace] オプションが有効な場合に表示するメッセージを指定する)

例

```
STATUS( `Script`, `Connection Opened`, SL_CANCEL )
```

TRACE 関数

構文

TRACE('string')

説明

TRACE 関数では、[スクリプト] タブで [Trace] モードが選択されている場合に、接続時に [コネクションモニタ] ウィンドウに表示するメッセージを設定します。

string は、表示するテキストです。

注

string には、次の特殊文字も含めて、最大 128 文字まで指定できます。

文字	値
¥r	キャリッジリターン
¥n	復帰改行
¥t	タブ
¥nnn	nnn は文字の 8 進数値
¥pnn	nn は接続パラメータの数値識別子です。パラメータを 1 桁の数字から構成する場合は、¥p03 というように、先行ゼロを付けます。

注意 ¥* と指定すれば、¥ 文字も指定できます。

関連項目

STATUS (接続時にステータスボックスに表示するアイコンおよびステータスメッセージを設定する)

例

```
TRACE( `Opening connection` )
```

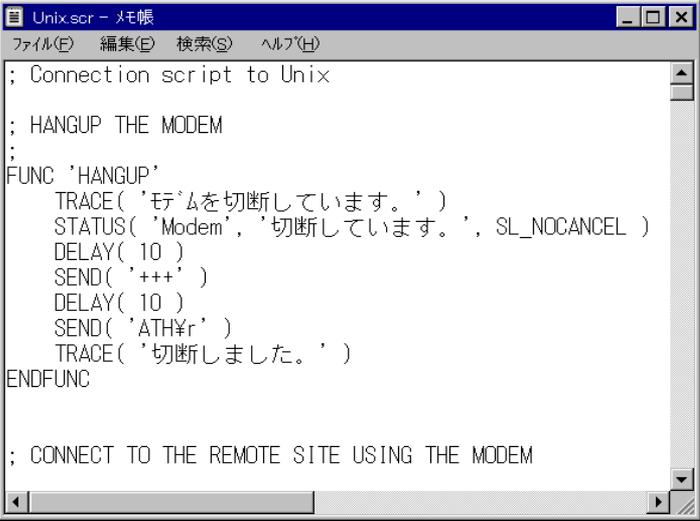
ComScript ファイルの例

このセクションの内容は、次のとおりです。

- サンプルファイルの概要
- ComScript ファイルの開発
- Connection Monitor を使用したテスト
- ComScript ファイルの生成
- ダイアルバック

サンプルファイルの概要

Vision コミュニケーションのスクリプトファイルは ASCII テキストファイルで、Windows のメモ帳などの PC ベースのエディタを使用して作成や変更を行うことができます。固有のスクリプトを作成する場合は、既存のスクリプトファイルのコピーを変更すると簡単に作成できます。ダイアルバック例ではこの方式を使用しています (ホストでどのスクリプトを使用しているかについては、本章の「ComScript ファイル」を参照してください)。また、コネクションモタを使用して、まったく新しいスクリプトを生成することもできます。詳細については、本章の「スクリプトファイルの生成」を参照してください。



```
Unix.scr - メモ帳
ファイル(F) 編集(E) 検索(S) ヘルプ(H)
; Connection script to Unix

; HANGUP THE MODEM
;
FUNC 'HANGUP'
    TRACE( 'モデムを切断しています。' )
    STATUS( 'Modem', '切断しています。', SL_NOCANCEL )
    DELAY( 10 )
    SEND( '+++ ' )
    DELAY( 10 )
    SEND( 'ATH#r ' )
    TRACE( '切断しました。' )
ENDFUNC

; CONNECT TO THE REMOTE SITE USING THE MODEM
```

通常、ネットワークを使用している場合は、スクリプトを変更する必要はありません。このため、この例ではシリアル回線接続の場合です。

ComScript ファイルの開発

スクリプトを作成する場合、まずそのスクリプトで制御される接続プロセス部分で必要となるステップを決定する必要があります。このステップは、[コネクションモニタ] ウィンドウを使用して決定できます。

手動接続を開くには

- 1 次のいずれかを行います。
 - [スタート] ボタンをクリックして、[プログラム] と [CentreNET X Vision] を順にポイントしてから、[VT420 エミュレータ] をクリックします。
 - または-
 - Windows NT 3.51 の場合は、プログラムマネージャで [CentreNET X Vision] グループを開いて、[VT420 エミュレータ] アイコンをダブルクリックします。

エミュレータウィンドウが画面に表示されます。
- 2 [セッション] メニューで、[接続] をクリックします。
- 3 [ホスト] ボックスで、ホスト名を入力するか、ドロップダウンリストから目的のホストを選択します。
- 4 [コマンドライン] ボックスで、ホストシステムで実行したいコマンドを入力します。
- 5 SHIFT キーを押したまま、[OK] をクリックします。
[コネクションモニタ] ウィンドウが表示され、接続プロセスが停止します。
- 6 [スクリプト] メニューで、[手動] をクリックします。
ここで、リンクを開きリモートホストにログインするために必要な情報を入力します。
- 7 [スクリプト] メニューで、[印脱] をクリックします。
この情報に基づいて、スクリプトの内容を作成します。
- 8 [スクリプト] メニューで、[続行] をクリックします。
[コネクションモニタ] ウィンドウが非表示になり、接続が起動されたプログラムである VT420 エミュレータにコントロールが返されます。

コネクションモニタを使用したテスト

また、[コネクションモニタ] ウィンドウを使用して、スクリプトファイルの実行によって導き出されるホストシステムと PC の間の対話を監視することもできます。このため、接続プロセスに異常が発生した場合に、異常箇所を究明して、異常が発生しないように ComScript ファイルを変更できます。

自動接続するには

- 1 次のいずれかを行います。
 - [スタート] ボタンをクリックして、[プログラム] と [CentreNET X Vision] を順にポイントしてから、[VT420 エミュレータ] をクリックします。
 - または-
 - Windows NT 3.51 の場合は、プログラムマネージャで [CentreNET X Vision] グループを開いて、[VT420 エミュレータ] アイコンをダブルクリックします。

エミュレータウィンドウが画面に表示されます。
- 2 [セッション] メニューで、[接続] をクリックします。
- 3 [ホスト] ボックスで、ホスト名を入力するか、ドロップダウンリストから目的のホストを選択します。
- 4 [コマンドライン] ボックスで、ホストシステムで実行したいコマンドを入力します。
- 5 SHIFT キーを押したまま、[OK] をクリックします。

[コネクションモニタ] ウィンドウが表示され、接続プロセスが停止します。

- 6 [スクリプト] メニューで、[再生] をクリックします。

[コネクションモニタ] ウィンドウの左側に出力が表示されるので、対話を監視できます。このため、スクリプトの異常箇所を簡単に究明することが可能です。

ComScript ファイルの生成

Connection Monitor を使用して、各接続に合った ComScript ファイルを生成することによって、さまざまな接続に対応できます。たとえば、次に示す手順に従えば、UNIX ホストに対する非 TAPI モデム接続用のスクリプトを生成できます。この接続では、接続プロセスのすべての部分 (接続、ログイン、コマンドの実行) を制御するためのスクリプトファイルが必要であるため、接続プロセスの 1 つの部分に対して詳細情報を指定したら、[続行] コマンドを使用して次のプロセス部分を指定します。

非 TAPI モデム接続用のスクリプトファイルを生成するには

- 1 [コネクションモニタ] で、[スクリプト] メニューをクリックしてから、[記録] をクリックします。
- 2 [接続領域] に `atz` と入力します。
モデム should は、このコマンドおよびステップ 3-5 で入力したコマンドに対して “OK” と応答するはずです。
- 3 `atv1` と入力します。
- 4 `atq0e1s0=0` と入力します。
- 5 `at` と入力します。
- 6 `atdt phone-number` を入力します。この場合、`phone-number` は [送信] メニューをクリックし、[電話番号] をクリック後、RETURN キーを押すことで指定されます。
- 7 CONNECT 応答が表示されたら、RETURN キーを押します。
- 8 [スクリプト] メニューで、[続行] をクリックします。
- 9 ホストに手動でログインします。[送信] メニューのコマンドを使用して、ユーザ名、パスワード、必要な端末タイプを指定します。
- 10 シェルコマンドインタプリタを実行するために、次のように指定します。

`execcommand-interpretter`

この場合、`command-interpretter` は [送信] メニューをクリックしてから [コマンド] をクリックすることによってホストに送信するコマンドインタプリタ (デフォルトでは、Bourne シェル `sh`) です。

- 11 [スクリプト] メニューで、[続行] をクリックします。

- 12 システムプロンプトで、これまでに端末タイプ情報を指定していない場合は、端末タイプをホストに知らせます。Bourne シェルの場合は、次のコマンドを入力します。

```
TERM=terminal type;export TERM
```

この場合、*terminal type* は [送信] メニューをクリックしてから、[Terminal] をクリックすることによって、ホストに送信する端末タイプです。

- 13 ホストでコマンドを実行するには、次のように指定します。

```
exec command
```

この場合、*command* は [送信] メニューをクリックしてから、[コマンド] をクリックして、ホストに送信するコマンドです。

- 14 [スクリプト] メニューで、[続行] をクリックします。
- 15 新しく作成したスクリプトファイルを保存するように求められたら、[はい] をクリックします。このファイルを、**¥Program Files¥Common Files¥Vision¥Scripts** フォルダに拡張子 **.scr** を付けて保存します。
- 16 [VT420 エミュレータ] で、[セッション] メニューをクリックしてから、必要に応じて [切断] をクリックします。[コネクションモニタ] ウィンドウが表示されます。
- 17 [スクリプト] メニューで、[続行] をクリックします。このスクリプトファイルが制御する接続プロセスの 3 つの部分に対して正しい切断操作を行う関数を生成するために、この操作を 3 回行います。
- 18 指示に応じて、スクリプトファイルに対する変更を保存します。
- 19 この新しいスクリプトファイルをホストで使用するかどうか求められたら、[はい] をクリックします。

ダイヤルバック

次に示す例では、ダイヤルバックシステムで使用できるようにするには、どのように **Unixsl.scr** を変更するかを示したものです。この場合、接続後ローカルモデムはハングアップし、正しい番号であれば、コールバックを発したモデムをダイヤルします。この例では、ローカルモデムが TAPI デバイスとして構成されていないものとします。また、特殊なダイヤルバックシステムを使用するには、これ以外にも ComScript ファイルを変更する必要がある場合があります。たとえば、ダイヤルバックシステムでプロンプトとして “login:” ではなく **idialback:i** を表示するには、これに応じて **DIALBACK** 関数を変更します。

Unixsl.scr の **DODIAL** 関数に新しくユーザが定義した関数 **DIALBACK** に対する呼び出しが設定されています。**NOTCALL** 関数を使用して、**DIALBACK** が呼び出され、呼び出しが失敗したかどうかテストされます。

```

FUNC `DODIAL`
    TRACE( `DODIAL` )
    IF NOTCALL( `MODEM` ) THEN
        RETURN( `FALSE` )
    IF NOTCALL( `DIALBACK` ) THEN
        RETURN( `FALSE` )
    ENDF
ENDFUNC

```

DIALBACK が失敗した場合 (いずれかの段階で `FALSE` 値が返された場合)、**RETURN** によって、呼び出された関数 (**DIALBACK**) の戻り値が **FALSE** に設定され、接続プロセスが停止されます。

MODEM によってリモートモデムにはじめて接続した場合、ログインプロンプトを引き出すためにいくつかのキャリッジリターンが送信されます。

```

FUNC `DIALBACK`

; Send some carriage returns to elicit a dialback login prompt.
; If it is not encountered assume that we have failed to get a
; dialback, and close the connection.
STATUS( `Script`, `Wait for login prompt`, SL_CANCEL )
SEND( `¥r¥r` )
IF NOMATCH( 5, `ogin:` )
THEN
    SEND( `¥r¥r` )
    IF NOMATCH( 5, `ogin:` )
    THEN
        SEND( `¥r¥r` )
        IF NOMATCH( 5, `ogin:` )
        THEN
            ERROR( `No login prompt` )
            RETURN( `FALSE` )
        ENDF
    ENDF
ENDIF

TRACE( `Dealing with the dialback system` )
STATUS( `Script`, `Dealing with the dialback system`, SL_CANCEL )

```

3組目のキャリッジリターンが送信されても、ダイヤルバックログインプロンプトが示されない場合は、該当するエラーメッセージが生成され、**FALSE** 値が返されます。

ダイヤルバックユーザ名とダイヤルバックパスワードを指定するように求められます。ダイヤルバックユーザ名とダイヤルバックパスワードはそれぞれのプロンプトに送信されます。

```
; Send dialback username to dialback prompt, then match again on
; possible prompts
IF REMATCH( `ogin` ) THEN
    DIALOG( `Enter dialback username`, 13 )
    TRACE( `Sending dialback username to login prompt` )
    SEND( `*p13*r` )
    MATCH( 10, `assword:`, `ogin` )
    ; A login prompt here means login has failed
    IF REMATCH( `ogin` ) THEN
        TRACE( `Initial connection failed` )
        ERROR( `Initial connection failed` )
        RETURN( `FALSE` )
    ENDIF
ENDIF

; Send dialback password to password prompt
IF REMATCH( `assword:` ) THEN
    DIALOG( `Enter dialback password`, 14 )
    TRACE( `Sending password to password prompt` )
    SEND( `*p14*r` )
```

ローカルモデムでリモートモデムが呼び出していることが検出された場合、RING 応答が PC にエコーされます。この場合、A コマンド (先頭に AT が付けられ、以降の文字がデータでなくコマンドであることを示します) を使用して、着呼に応答します。

```
; Wait for the modem to dial back, and answer it
MATCH( 30, `RING` )
IF REMATCH( `RING` ) THEN
    SEND( `ATA` )
```

最後に、2つのモデム間でデータ接続が確立されると、CONNECT 応答がPCにエコーされます。この結果コードが受け付けられない場合は、該当するエラーメッセージが表示されます。

```
        ; If we don't get a `CONNECT` prompt, the modems have
failed to
        ; synchronize on dialback
MATCH( 10, `ONNECT` )
IF NOREMATCH ( `ONNECT` ) THEN
    TRACE( `PC failed to answer callback` )
    ERROR ( `PC failed to answer callback` )
    RETURN( `FALSE` )
ENDIF
ELSE
    TRACE( `Unix system failed to call PC` )
    ERROR( `Unix system failed to call PC` )
    RETURN( `FALSE` )
ENDIF
ELSE
    TRACE( `No password prompt` )
    ERROR( `No password prompt` )
    RETURN( `FALSE` )
ENDIF

TRACE( `Dialback succeeded` )

ENDFUNC
```

付録 E

X Keysyms

標準的な X Keysyms と文字セットの対応は次のとおりです。

Miscellany

Latin 1

Latin 2

Latin 3

Latin 4

Katakana

Arabic

Cyrillic

Greek

Technical

Special

Publishing

APL

Hebrew

XKeysymDB

Miscellany

BackSpace	0xFF08 /*backspace,back char*/	KP_Enter	0xFF8D /*enter*/
Tab	0xFF09	KP_F1	0xFF91 /*PF1,KP_A,...*/
Linefeed	0xFF0A /*Linefeed,LF*/	KP_F2	0xFF92
Clear	0xFF0B	KP_F3	0xFF93
Return	0xFF0D /*Return,enter*/	KP_F4	0xFF94
Pause	0xFF13 /*Pause,hold*/	KP_Equal	0xFFBD /*equals*/
Scroll_Lock	0xFF14	KP_Multiply	0xFFAA
Escape	0xFF1B	KP_Add	0xFFAB
Delete	0xFFFF /*Delete,rubout*/	KP_Separator	0xFFAC /*separator,often comma*/
Multi_key	0xFF20 /*Multi-keycharactercompose*/	KP_Subtract	0xFFAD
Kanji	0xFF21 /*Kanji,Kanji convert*/	KP_Decimal	0xFFAE
Muhenkan	0xFF22 /*CancelConversion*/	KP_Divide	0xFFAF
Henkan_Mode	0xFF23 /*Start/StopConversion*/	KP_0	0xFFB0
Henkan	0xFF23 /*AliasforHenkan_Mode*/	KP_1	0xFFB1
Romaji	0xFF24 /*toRomaji*/	KP_2	0xFFB2
Hiragana	0xFF25 /*toHiragana*/	KP_3	0xFFB3
Katakana	0xFF26 /*toKatakana*/	KP_4	0xFFB4
Hiragana_Katakana	0xFF27	KP_5	0xFFB5
Zenkaku	0xFF28 /*toZenkaku*/	KP_6	0xFFB6
Hankaku	0xFF29 /*toHankaku*/	KP_7	0xFFB7
Zenkaku_Hankaku	0xFF2A	KP_8	0xFFB8
Touroku	0xFF2B /*AddtoDictionary*/	KP_9	0xFFB9
Massyo	0xFF2C /*DeletefromDictionary*/	F1	0xFFBE
Kana_Lock	0xFF2D /*KanaLock*/	F2	0xFFBF
Kana_Shift	0xFF2E /*KanaShift*/	F3	0xFFC0
Eisu_Shift	0xFF2F /*AlphanumericShift*/	F4	0xFFC1
Eisu_toggle	0xFF30 /*Alphanumerictoggle*/	F5	0xFFC2
Home	0xFF50	F6	0xFFC3
Left	0xFF51 /*Moveleft,leftarrow*/	F7	0xFFC4
Up	0xFF52 /*Moveup,uparrow*/	F8	0xFFC5
Right	0xFF53 /*Moveright,rightarrow*/	F9	0xFFC6
Down	0xFF54 /*Movedown,downarrow*/	F10	0xFFC7
Prior	0xFF55 /*Prior,previous*/	F11	0xFFC8
Next	0xFF56 /*Next*/	L1	0xFFC9
End	0xFF57 /*EOL*/	F12	0xFFCA
Begin	0xFF58 /*BOL*/	L2	0xFFCB
Select	0xFF60 /*Select,mark*/	F13	0xFFCA
Print	0xFF61	L3	0xFFCB
Execute	0xFF62 /*Execute,run,do*/	F14	0xFFCB
Insert	0xFF63 /*Insert,inserthere*/	L4	0xFFCB
Undo	0xFF65 /*Undo,oops*/	F15	0xFFCC
Redo	0xFF66 /*redo,again*/	L5	0xFFCC
Menu	0xFF67	F16	0xFFCD
Find	0xFF68 /*Find,search*/	L6	0xFFCD
Cancel	0xFF69 /*Cancel,stop,abort,exit*/	F17	0xFFCE
Help	0xFF6A /*Help,?*/	L7	0xFFCE
Break	0xFF6B	F18	0xFFCF
Mode_switch	0xFF7E /*Charactersetswitch*/	L8	0xFFCF
script_switch	0xFF7E /*Aliasformode_switch*/	F19	0xFFD0
Num_Lock	0xFF7F	L9	0xFFD0
KP_Space	0xFF80 /*space*/	F20	0xFFD1
KP_Tab	0xFF89	L10	0xFFD1
		F21	0xFFD2
		R1	0xFFD2

F22	0xFFD3	quoteright	0x027 /*deprecated*/
R2	0xFFD3	parenleft	0x028
F23	0xFFD4	parenright	0x029
R3	0xFFD4	asterisk	0x02a
F24	0xFFD5	plus	0x02b
R4	0xFFD5	comma	0x02c
F25	0xFFD6	minus	0x02d
R5	0xFFD6	period	0x02e
F26	0xFFD7	slash	0x02f
R6	0xFFD7	0	0x030
F27	0xFFD8	1	0x031
R7	0xFFD8	2	0x032
F28	0xFFD9	3	0x033
R8	0xFFD9	4	0x034
F29	0xFFDA	5	0x035
R9	0xFFDA	6	0x036
F30	0xFFDB	7	0x037
R10	0xFFDB	8	0x038
F31	0xFFDC	9	0x039
R11	0xFFDC	colon	0x03a
F32	0xFFDD	semicolon	0x03b
R12	0xFFDD	less	0x03c
R13	0xFFDE	equal	0x03d
F33	0xFFDE	greater	0x03e
F34	0xFFDF	question	0x03f
R14	0xFFDF	at	0x040
F35	0xFFE0	A	0x041
R15	0xFFE0	B	0x042
Shift_L	0xFFE1 /*Left shift*/	C	0x043
Shift_R	0xFFE2 /*Right shift*/	D	0x044
Control_L	0xFFE3 /*Left control*/	E	0x045
Control_R	0xFFE4 /*Right control*/	F	0x046
Caps_Lock	0xFFE5 /*Caps lock*/	G	0x047
Shift_Lock	0xFFE6 /*Shift lock*/	H	0x048
Meta_L	0xFFE7 /*Left meta*/	I	0x049
Meta_R	0xFFE8 /*Right meta*/	J	0x04a
Alt_L	0xFFE9 /*Left alt*/	K	0x04b
Alt_R	0xFFEA /*Right alt*/	L	0x04c
Super_L	0xFFEB /*Left super*/	M	0x04d
Super_R	0xFFEC /*Right super*/	N	0x04e
Hyper_L	0xFFED /*Left hyper*/	O	0x04f
Hyper_R	0xFFEE /*Right hyper*/	P	0x050
		Q	0x051
		R	0x052
		S	0x053
		T	0x054
		U	0x055
		V	0x056
		W	0x057
		X	0x058
		Y	0x059
		Z	0x05a
		bracketleft	0x05b
		backslash	0x05c
			0x05d
			0x05e
			0x05f
			0x060
			0x061
			0x062
			0x063
			0x064
			0x065
			0x066
			0x067
			0x068
			0x069
			0x06a
			0x06b
			0x06c
			0x06d
			0x06e
			0x06f
			0x070
			0x071
			0x072
			0x073
			0x074
			0x075
			0x076
			0x077
			0x078
			0x079
			0x07a
			0x07b
			0x07c
			0x07d
			0x07e
			0x07f
			0x080
			0x081
			0x082
			0x083
			0x084
			0x085
			0x086
			0x087
			0x088
			0x089
			0x08a
			0x08b
			0x08c
			0x08d
			0x08e
			0x08f
			0x090
			0x091
			0x092
			0x093
			0x094
			0x095
			0x096
			0x097
			0x098
			0x099
			0x09a
			0x09b
			0x09c
			0x09d
			0x09e
			0x09f
			0x0a0
			0x0a1
			0x0a2
			0x0a3
			0x0a4
			0x0a5
			0x0a6
			0x0a7
			0x0a8
			0x0a9
			0x0aa
			0x0ab
			0x0ac
			0x0ad
			0x0ae
			0x0af
			0x0b0
			0x0b1
			0x0b2
			0x0b3
			0x0b4
			0x0b5
			0x0b6
			0x0b7
			0x0b8
			0x0b9
			0x0ba
			0x0bb
			0x0bc
			0x0bd
			0x0be
			0x0bf
			0x0c0
			0x0c1
			0x0c2
			0x0c3
			0x0c4
			0x0c5
			0x0c6
			0x0c7
			0x0c8
			0x0c9
			0x0ca
			0x0cb
			0x0cc
			0x0cd
			0x0ce
			0x0cf
			0x0d0
			0x0d1
			0x0d2
			0x0d3
			0x0d4
			0x0d5
			0x0d6
			0x0d7
			0x0d8
			0x0d9
			0x0da
			0x0db
			0x0dc
			0x0dd
			0x0de
			0x0df
			0x0e0
			0x0e1
			0x0e2
			0x0e3
			0x0e4
			0x0e5
			0x0e6
			0x0e7
			0x0e8
			0x0e9
			0x0ea
			0x0eb
			0x0ec
			0x0ed
			0x0ee
			0x0ef
			0x0f0
			0x0f1
			0x0f2
			0x0f3
			0x0f4
			0x0f5
			0x0f6
			0x0f7
			0x0f8
			0x0f9
			0x0fa
			0x0fb
			0x0fc
			0x0fd
			0x0fe
			0x0ff
			0x100
			0x101
			0x102
			0x103
			0x104
			0x105
			0x106
			0x107
			0x108
			0x109
			0x10a
			0x10b
			0x10c
			0x10d
			0x10e
			0x10f
			0x110
			0x111
			0x112
			0x113
			0x114
			0x115
			0x116
			0x117
			0x118
			0x119
			0x11a
			0x11b
			0x11c
			0x11d
			0x11e
			0x11f
			0x120
			0x121
			0x122
			0x123
			0x124
			0x125
			0x126
			0x127
			0x128
			0x129
			0x12a
			0x12b
			0x12c
			0x12d
			0x12e
			0x12f
			0x130
			0x131
			0x132
			0x133
			0x134
			0x135
			0x136
			0x137
			0x138
			0x139
			0x13a
			0x13b
			0x13c
			0x13d
			0x13e
			0x13f
			0x140
			0x141
			0x142
			0x143
			0x144
			0x145
			0x146
			0x147
			0x148
			0x149
			0x14a
			0x14b
			0x14c
			0x14d
			0x14e
			0x14f
			0x150
			0x151
			0x152
			0x153
			0x154
			0x155
			0x156
			0x157
			0x158
			0x159
			0x15a
			0x15b
			0x15c
			0x15d
			0x15e
			0x15f
			0x160
			0x161
			0x162
			0x163
			0x164
			0x165
			0x166
			0x167
			0x168
			0x169
			0x16a
			0x16b
			0x16c
			0x16d
			0x16e
			0x16f
			0x170
			0x171
			0x172
			0x173
			0x174
			0x175
			0x176
			0x177
			0x178
			0x179
			0x17a
			0x17b
			0x17c
			0x17d
			0x17e
			0x17f
			0x180
			0x181
			0x182
			0x183
			0x184
			0x185
			0x186
			0x187
			0x188
			0x189
			0x18a
			0x18b
			0x18c
			0x18d
			0x18e
			0x18f
			0x190
			0x191
			0x192
			0x193
			0x194
			0x195
			0x196
			0x197
			0x198
			0x199
			0x19a
			0x19b
			0x19c
			0x19d
			0x19e
			0x19f
			0x1a0
			0x1a1
			0x1a2
			0x1a3
			0x1a4
			0x1a5
			0x1a6
			0x1a7
			0x1a8
			0x1a9
			0x1aa
			0x1ab
			0x1ac
			0x1ad
			0x1ae
			0x1af
			0x1b0
			0x1b1
			0x1b2
			0x1b3
			0x1b4
			0x1b5
			0x1b6
			0x1b7
			0x1b8
			0x1b9
			0x1ba
			0x1bb
			0x1bc
			0x1bd
			0x1be
			0x1bf
			0x1c0
			0x1c1
			0x1c2
			0x1c3
			0x1c4
			0x1c5
			0x1c6
			0x1c7
			0x1c8
			0x1c9
			0x1ca
			0x1cb
			0x1cc
			0x1cd
			0x1ce
			0x1cf
			0x1d0
			0x1d1
			0x1d2
			0x1d3
			0x1d4
			0x1d5
			0x1d6
			0x1d7
			0x1d8
			0x1d9
			0x1da
			0x1db
			0x1dc
			0x1dd

bracketright	0x05d	twosuperior	0x0b2
asciicircum	0x05e	threesuperior	0x0b3
underscore	0x05f	acute	0x0b4
grave	0x060	mu	0x0b5
quoteleft	0x060 /*deprecated*/	paragraph	0x0b6
a	0x061	periodcentered	0x0b7
b	0x062	cedilla	0x0b8
c	0x063	onesuperior	0x0b9
d	0x064	masculine	0x0ba
e	0x065	guillemotright	0x0bc
f	0x066	onehalf	0x0bd
g	0x067	threequarters	0x0be
h	0x068	questiondown	0x0bf
i	0x069	Agrave	0x0c0
j	0x06a	Aacute	0x0c1
k	0x06b	Acircumflex	0x0c2
l	0x06c	Atilde	0x0c3
m	0x06d	Adiaeresis	0x0c4
n	0x06e	Aring	0x0c5
o	0x06f	AE	0x0c6
p	0x070	Ccedilla	0x0c7
q	0x071	Egrave	0x0c8
r	0x072	Eacute	0x0c9
s	0x073	Ecircumflex	0x0ca
t	0x074	Ediaeresis	0x0cb
u	0x075	Igrave	0x0cc
v	0x076	iacute	0x0cd
w	0x077	Icircumflex	0x0ce
x	0x078	Idiaeresis	0x0cf
y	0x079	ETH	0x0d0
z	0x07a	Eth	0x0d0 /*deprecated*/
braceleft	0x07b	Ntilde	0x0d1
bar	0x07c	Ograve	0x0d2
braceright	0x07d	Oacute	0x0d3
asciitilde	0x07e	Ocircumflex	0x0d4
nobreakspace	0x0a0	Otilde	0x0d5
exclamdown	0x0a1	Odiaeresis	0x0d6
cent	0x0a2	multiply	0x0d7
sterling	0x0a3	Coblique	0x0d8
currency	0x0a4	Ugrave	0x0d9
yen	0x0a5	Uacute	0x0da
brokenbar	0x0a6	Ucircumflex	0x0db
section	0x0a7	Udiaeresis	0x0dc
diaeresis	0x0a8	Yacute	0x0dd
copyright	0x0a9	THORN	0x0de
ordfeminine	0x0aa	Thorn	0x0de /*deprecated*/
guillemotleft	0x0ab /*left angle quotation	ssharp	0x0df
mark*/		agrave	0x0e0
notsign	0x0ac	aacute	0x0e1
hyphen	0x0ad	acircumflex	0x0e2
registered	0x0ae	atilde	0x0e3
macron	0x0af	adiaeresis	0x0e4
degree	0x0b0	aring	0x0e5
plusminus	0x0b1	ae	0x0e6

ccedilla	0x0e7
egrave	0x0e8
eacute	0x0e9
ecircumflex	0x0ea
ediaeresis	0x0eb
igrave	0x0ec
iacute	0x0ed
icircumflex	0x0ee
idiaeresis	0x0ef
eth	0x0f0
ntilde	0x0f1
ograve	0x0f2
oacute	0x0f3
ocircumflex	0x0f4
otilde	0x0f5
odiaeresis	0x0f6
division	0x0f7
oslash	0x0f8
ugrave	0x0f9
uacute	0x0fa
ucircumflex	0x0fb
udiaeresis	0x0fc
yacute	0x0fd
thorn	0x0fe
ydiaeresis	0x0ff

Latin 2

Aogonek	0x1a1
breve	0x1a2
Lstroke	0x1a3
Lcaron	0x1a5
Sacute	0x1a6
Scaron	0x1a9
Scedilla	0x1aa
Tcaron	0x1ab
Zacute	0x1ac
Zcaron	0x1ae
Zabovedot	0x1af
aogonek	0x1b1
ogonek	0x1b2
lstroke	0x1b3
lcaron	0x1b5
sacute	0x1b6
caron	0x1b7
scaron	0x1b9
scedilla	0x1ba
tcaron	0x1bb
zacute	0x1bc
doubleacute	0x1bd
zcaron	0x1be
zabovedot	0x1bf
Racute	0x1c0

Abreve	0x1c3
Lacute	0x1c5
Cacute	0x1c6
Ccaron	0x1c8
Eogonek	0x1ca
Ecaron	0x1cc
Dcaron	0x1cf
Dstroke	0x1d0
Nacute	0x1d1
Ncaron	0x1d2
Odoubleacute	0x1d5
Rcaron	0x1d8
Uring	0x1d9
Udoubleacute	0x1db
Tcedilla	0x1de
racute	0x1e0
abreve	0x1e3
lacute	0x1e5
cacute	0x1e6
ccaron	0x1e8
eogonek	0x1ea
ecaron	0x1ec
dcaron	0x1ef
dstroke	0x1f0
racute	0x1f1
ncaron	0x1f2
odoubleacute	0x1f5
udoubleacute	0x1fb
rcaron	0x1f8
uring	0x1f9
tcedilla	0x1fe
abovedot	0x1ff

Latin 3

Hstroke	0x2a1
Hcircumflex	0x2a6
Iabovedot	0x2a9
Gbreve	0x2ab
Jcircumflex	0x2ac
hstroke	0x2b1
hcircumflex	0x2b6
idotless	0x2b9
gbreve	0x2bb
jcircumflex	0x2bc
Cabovedot	0x2c5
Ccircumflex	0x2c6
Gabovedot	0x2d5
Gcircumflex	0x2d8
Ubreve	0x2dd
Scircumflex	0x2de
cabovedot	0x2e5
ccircumflex	0x2e6

gabovedot	0x2f5	kana_WO	0x4a6
gcircumflex	0x2f8	kana_a	0x4a7
ubreve	0x2fd	kana_i	0x4a8
scircumflex	0x2fe	kana_u	0x4a9
		kana_e	0x4aa
		kana_o	0x4ab
		kana_ya	0x4ac
		kana_yu	0x4ad
		kana_yo	0x4ae
		kana_tsu	0x4af
		kana_tu	0x4af /* deprecated */
		prolongedsound	0x4b0
		kana_A	0x4b1
		kana_I	0x4b2
		kana_U	0x4b3
		kana_E	0x4b4
		kana_O	0x4b5
		kana_KA	0x4b6
		kana_KI	0x4b7
		kana_KU	0x4b8
		kana_KE	0x4b9
		kana_KO	0x4ba
		kana_SA	0x4bb
		kana_SHI	0x4bc
		kana_SU	0x4bd
		kana_SE	0x4be
		kana_SO	0x4bf
		kana_TA	0x4c0
		kana_CHI	0x4c1
		kana_TI	0x4c1 /* deprecated */
		kana_TSU	0x4c2
		kana_TU	0x4c2 /* deprecated */
		kana_TE	0x4c3
		kana_TO	0x4c4
		kana_NA	0x4c5
		kana_NI	0x4c6
		kana_NU	0x4c7
		kana_NE	0x4c8
		kana_NO	0x4c9
		kana_HA	0x4ca
		kana_HI	0x4cb
		kana_FU	0x4cc
		kana_HU	0x4cc /* deprecated */
		kana_HE	0x4cd
		kana_HO	0x4ce
		kana_MA	0x4cf
		kana_MI	0x4d0
		kana_MU	0x4d1
		kana_ME	0x4d2
		kana_MO	0x4d3
		kana_YA	0x4d4
		kana_YU	0x4d5
		kana_YO	0x4d6
		kana_RA	0x4d7

Latin 4

kra	0x3a2
kappa	0x3a2 /* deprecated */
Rcedilla	0x3a3
Itilde	0x3a5
Lcedilla	0x3a6
Emacron	0x3aa
Gcedilla	0x3ab
Tslash	0x3ac
rcedilla	0x3b3
itilde	0x3b5
lcedilla	0x3b6
emacron	0x3ba
gcedilla	0x3bb
tslash	0x3bc
ENG	0x3bd
eng	0x3bf
Amacron	0x3c0
Iogonek	0x3c7
Eabovedot	0x3cc
Imacron	0x3cf
Ncedilla	0x3d1
Omacron	0x3d2
Kcedilla	0x3d3
Uogonek	0x3d9
Utilde	0x3dd
Umacron	0x3de
amacron	0x3e0
iogonek	0x3e7
eabovedot	0x3ec
imacron	0x3ef
ncedilla	0x3f1
omacron	0x3f2
kcedilla	0x3f3
uogonek	0x3f9
utilde	0x3fd
umacron	0x3fe

Katakana

overline	0x47e
kana_fullstop	0x4a1
kana_openingbracket	0x4a2
kana_closingbracket	0x4a3
kana_comma	0x4a4
kana_conjunctive	0x4a5
kana_middledot	0x4a5 /* deprecated */

kana_RI	0x4d8	Arabic_yeh	0x5ea
kana_RU	0x4d9	Arabic_fathatan	0x5eb
kana_RE	0x4da	Arabic_dammatan	0x5ec
kana_RO	0x4db	Arabic_kasratan	0x5ed
kana_WA	0x4dc	Arabic_fatha	0x5ee
kana_N	0x4dd	Arabic_damma	0x5ef
voicedsound	0x4de	Arabic_kasra	0x5f0
semivoicedsound	0x4df	Arabic_shadda	0x5f1
kana_switch	0xFF7E /* Alias for	Arabic_sukun	0x5f2
mode_switch*/		Arabic_switch	0xFF7E /* Alias for
		mode_switch*/	

Arabic

Arabic_comma	0x5ac
Arabic_semicolon	0x5bb
Arabic_question_mark	0x5bf
Arabic_hamza	0x5c1
Arabic_maddaonalef	0x5c2
Arabic_hamzaonalef	0x5c3
Arabic_hamzaonwaw	0x5c4
Arabic_hamzaunderalef	0x5c5
Arabic_hamzaonyeh	0x5c6
Arabic_alef	0x5c7
Arabic_beh	0x5c8
Arabic_tehmarbuta	0x5c9
Arabic_teh	0x5ca
Arabic_theh	0x5cb
Arabic_jeem	0x5cc
Arabic_hah	0x5cd
Arabic_khah	0x5ce
Arabic_dal	0x5cf
Arabic_thal	0x5d0
Arabic_ra	0x5d1
Arabic_zain	0x5d2
Arabic_seen	0x5d3
Arabic_sheen	0x5d4
Arabic_sad	0x5d5
Arabic_dad	0x5d6
Arabic_tah	0x5d7
Arabic_zah	0x5d8
Arabic_ain	0x5d9
Arabic_ghain	0x5da
Arabic_tatweel	0x5e0
Arabic_feh	0x5e1
Arabic_qaf	0x5e2
Arabic_kaf	0x5e3
Arabic_lam	0x5e4
Arabic_meem	0x5e5
Arabic_noon	0x5e6
Arabic_ha	0x5e7
Arabic_heh	0x5e7 /*deprecated*/
Arabic_waw	0x5e8
Arabic_alefmaksura	0x5e9

Cyrillic

Serbian_dje	0x6a1
Macedonia_gje	0x6a2
Cyrillic_io	0x6a3
Ukrainian_ie	0x6a4
Ukrainian_je	0x6a4 /*deprecated*/
Macedonia_dse	0x6a5
Ukrainian_i	0x6a6
Ukrainian_i	0x6a6 /*deprecated*/
Ukrainian_yi	0x6a7
Ukrainian_yi	0x6a7 /*deprecated*/
Cyrillic_je	0x6a8
Serbian_je	0x6a8 /*deprecated*/
Cyrillic_lje	0x6a9
Serbian_lje	0x6a9 /*deprecated*/
Cyrillic_nje	0x6aa
Serbian_nje	0x6aa /*deprecated*/
Serbian_tshe	0x6ab
Macedonia_kje	0x6ac
Byelorussian_shortu	0x6ae
Cyrillic_dzhe	0x6af
Serbian_dze	0x6af /*deprecated*/
numerosign	0x6b0
Serbian_DJE	0x6b1
Macedonia_GJE	0x6b2
Cyrillic_IO	0x6b3
Ukrainian_IE	0x6b4
Ukrainian_IE	0x6b4 /*deprecated*/
Macedonia_DSE	0x6b5
Ukrainian_I	0x6b6
Ukrainian_I	0x6b6 /*deprecated*/
Ukrainian_YI	0x6b7
Ukrainian_YI	0x6b7 /*deprecated*/
Cyrillic_JE	0x6b8
Serbian_JE	0x6b8 /*deprecated*/
Cyrillic_LJE	0x6b9
Serbian_LJE	0x6b9 /*deprecated*/
Cyrillic_NJE	0x6ba
Serbian_NJE	0x6ba /*deprecated*/
Serbian_TSHE	0x6bb

Macedonia_KJE	0x6bc
Byelorussian_SHORTU	0x6be
Cyrillic_DZHE	0x6bf
Serbian_DZJE	0x6bf /*deprecated*/
Cyrillic_yu	0x6c0
Cyrillic_a	0x6c1
Cyrillic_be	0x6c2
Cyrillic_tse	0x6c3
Cyrillic_de	0x6c4
Cyrillic_ie	0x6c5
Cyrillic_ef	0x6c6
Cyrillic_ghe	0x6c7
Cyrillic_ha	0x6c8
Cyrillic_i	0x6c9
Cyrillic_shorti	0x6ca
Cyrillic_ka	0x6cb
Cyrillic_el	0x6cc
Cyrillic_em	0x6cd
Cyrillic_en	0x6ce
Cyrillic_o	0x6cf
Cyrillic_pe	0x6d0
Cyrillic_ya	0x6d1
Cyrillic_er	0x6d2
Cyrillic_es	0x6d3
Cyrillic_te	0x6d4
Cyrillic_u	0x6d5
Cyrillic_zhe	0x6d6
Cyrillic_ve	0x6d7
Cyrillic_softsign	0x6d8
Cyrillic_yeru	0x6d9
Cyrillic_ze	0x6da
Cyrillic_sha	0x6db
Cyrillic_e	0x6dc
Cyrillic_shcha	0x6dd
Cyrillic_che	0x6de
Cyrillic_hardsign	0x6df
Cyrillic_YU	0x6e0
Cyrillic_A	0x6e1
Cyrillic_BE	0x6e2
Cyrillic_TSE	0x6e3
Cyrillic_DE	0x6e4
Cyrillic_IE	0x6e5
Cyrillic_EF	0x6e6
Cyrillic_GHE	0x6e7
Cyrillic_HA	0x6e8
Cyrillic_I	0x6e9
Cyrillic_SHORTU	0x6ea
Cyrillic_KA	0x6eb
Cyrillic_EL	0x6ec
Cyrillic_EM	0x6ed
Cyrillic_EN	0x6ee
Cyrillic_O	0x6ef
Cyrillic_PE	0x6f0
Cyrillic_YA	0x6f1

Cyrillic_ER	0x6f2
Cyrillic_ES	0x6f3
Cyrillic_TE	0x6f4
Cyrillic_U	0x6f5
Cyrillic_ZHE	0x6f6
Cyrillic_VE	0x6f7
Cyrillic_SOFTSIGN	0x6f8
Cyrillic_YERU	0x6f9
Cyrillic_ZE	0x6fa
Cyrillic_SHA	0x6fb
Cyrillic_E	0x6fc
Cyrillic_SHCHA	0x6fd
Cyrillic_CHE	0x6fe
Cyrillic_HARDSIGN	0x6ff

Greek

Greek_ALPHAaccent	0x7a1
Greek_EPSILONaccent	0x7a2
Greek_ETAaccent	0x7a3
Greek_IOTAaccent	0x7a4
Greek_IOTAdiaeresis	0x7a5
Greek_OMICRONaccent	0x7a7
Greek_UPSILONaccent	0x7a8
Greek_UPSILONdieresis	0x7a9
Greek_OMEGAaccent	0x7ab
Greek_accentdieresis	0x7ae
Greek_horizbar	0x7af
Greek_alphaaccent	0x7b1
Greek_epsilonaccent	0x7b2
Greek_etaaccent	0x7b3
Greek_iotaaccent	0x7b4
Greek_jotadieresis	0x7b5
Greek_iotaaccentdieresis	0x7b6
Greek_omicronaccent	0x7b7
Greek_upsilonaccent	0x7b8
Greek_upsilondieresis	0x7b9
Greek_upsilonaccentdieresis	0x7ba
Greek_omegaaccent	0x7bb
Greek_ALPHA	0x7c1
Greek_BETA	0x7c2
Greek_GAMMA	0x7c3
Greek_DELTA	0x7c4
Greek_EPSILON	0x7c5
Greek_ZETA	0x7c6
Greek_ETA	0x7c7
Greek_THETA	0x7c8
Greek_IOTA	0x7c9
Greek_KAPPA	0x7ca
Greek_LAMDA	0x7cb
Greek_LAMBDA	0x7cb
Greek_MU	0x7cc
Greek_NU	0x7cd

Greek_XI	0x7ce	botleftparens	0x8ac
Greek_OMICRON	0x7cf	toprightparens	0x8ad
Greek_PI	0x7d0	botrightparens	0x8ae
Greek_RHO	0x7d1	leftmiddlecurlybrace	0x8af
Greek_SIGMA	0x7d2	rightmiddlecurlybrace	0x8b0
Greek_TAU	0x7d4	toleftsummation	0x8b1
Greek_UPSILON	0x7d5	botleftsummation	0x8b2
Greek_PHI	0x7d6	topvertsummationconnector	0x8b3
Greek_CHI	0x7d7	botvertsummationconnector	0x8b4
Greek_PSI	0x7d8	toprightsummation	0x8b5
Greek_OMEGA	0x7d9	botrightsummation	0x8b6
Greek_alpha	0x7e1	rightmiddlesummation	0x8b7
Greek_beta	0x7e2	lessthanequal	0x8bc
Greek_gamma	0x7e3	notequal	0x8bd
Greek_delta	0x7e4	greaterthanequal	0x8be
Greek_epsilon	0x7e5	integral	0x8bf
Greek_zeta	0x7e6	therefore	0x8c0
Greek_eta	0x7e7	variation	0x8c1
Greek_theta	0x7e8	infinity	0x8c2
Greek_iota	0x7e9	nabla	0x8c5
Greek_kappa	0x7ea	approximate	0x8c8
Greek_lambda	0x7eb	similarequal	0x8c9
Greek_lambda	0x7eb	ifonlyif	0x8cd
Greek_mu	0x7ec	implies	0x8ce
Greek_nu	0x7ed	identical	0x8cf
Greek_xi	0x7ee	radical	0x8d6
Greek_omicron	0x7ef	includedin	0x8da
Greek_pi	0x7f0	includes	0x8db
Greek_tho	0x7f1	intersection	0x8dc
Greek_sigma	0x7f2	union	0x8dd
Greek_finalsmallsigma	0x7f3	logicaland	0x8de
Greek_tau	0x7f4	logicalor	0x8df
Greek_upsilon	0x7f5	partialderivative	0x8ef
Greek_phi	0x7f6	function	0x8f6
Greek_chi	0x7f7	leftarrow	0x8fb
Greek_psi	0x7f8	uparrow	0x8fc
Greek_omega	0x7f9	rightarrow	0x8fd
Greek_switch	0xFF7E /* Alias for	downarrow	0x8fe
mode_switch*/			

Technical

leftradical	0x8a1
toplefttradical	0x8a2
horizconnector	0x8a3
topintegral	0x8a4
botintegral	0x8a5
vertconnector	0x8a6
topleftsqbracket	0x8a7
botleftsqbracket	0x8a8
toprightsqbracket	0x8a9
botrightsqbracket	0x8aa
topleftparens	0x8ab

Special

blank	0x9df
soliddiamond	0x9e0
checkerboard	0x9e1
ht	0x9e2
ff	0x9e3
cr	0x9e4
lf	0x9e5
nl	0x9e8
vt	0x9e9
lowrightcorner	0x9ea
uprightcorner	0x9eb
upleftcorner	0x9ec

lowleftcorner	0x9ed
crossinglines	0x9ee
horizlinescan1	0x9ef
horizlinescan3	0x9f0
horizlinescan5	0x9f1
horizlinescan7	0x9f2
horizlinescan9	0x9f3
leftt	0x9f4
rightt	0x9f5
bott	0x9f6
topt	0x9f7
vertbar	0x9f8

Publishing

emspace	0xaa1
enspace	0xaa2
em3space	0xaa3
em4space	0xaa4
digitsspace	0xaa5
punctspace	0xaa6
thinspace	0xaa7
hairspace	0xaa8
emdash	0xaa9
endash	0xaaa
signifblank	0xaac
ellipsis	0xaae
doubbaselinedot	0xaaf
onethird	0xab0
twothirds	0xab1
onefifth	0xab2
twofifths	0xab3
threefifths	0xab4
fourfifths	0xab5
onesixth	0xab6
fivesixths	0xab7
careof	0xab8
figdash	0xab9
leftanglebracket	0xabc
decimalpoint	0xabd
rightanglebracket	0xabe
marker	0xabf
oneeighth	0xac3
threeeighths	0xac4
fiveeighths	0xac5
seveneighths	0xac6
trademark	0xac9
signaturemark	0xaca
trademarkincircle	0xacb
leftpentrangle	0xacc
rightpentrangle	0xacd
emopencircle	0xace
emopenrectangle	0xacf

leftsinglequotemark	0xad0
rightsinglequotemark	0xad1
leftdoublequotemark	0xad2
rightdoublequotemark	0xad3
prescription	0xad4
minutes	0xad6
seconds	0xad7
latincross	0xad9
hexagram	0xada
filledrectbullet	0xadb
filledlefttribullet	0xadc
filledrighttribullet	0xadd
emfilledcircle	0xade
emfilledrect	0xadf
enopencircbullet	0xae0
enopensquarebullet	0xae1
openrectbullet	0xae2
opentribulletup	0xae3
opentribulletdown	0xae4
openstar	0xae5
enfilledcircbullet	0xae6
enfilledsqbullet	0xae7
filledtribulletup	0xae8
filledtribulletdown	0xae9
leftpointer	0xaea
rightpointer	0xaeb
club	0xaec
diamond	0xaed
heart	0xaee
maltesecross	0xaf0
dagger	0xaf1
doubledagger	0xaf2
checkmark	0xaf3
ballotcross	0xaf4
musicalsharp	0xaf5
musicalflat	0xaf6
malesymbol	0xaf7
femalesymbol	0xaf8
telephone	0xaf9
telephonerecorder	0xafa
phonographcopyright	0xafb
caret	0xafc
singlelowquotemark	0xafd
doublelowquotemark	0xafe
cursor	0xaff

APL

leftcaret	0xba3
rightcaret	0xba6
downcaret	0xba8
upcaret	0xba9
overbar	0xbc0

downtack	0xbc2
upshoe	0xbc3
downstile	0xbc4
underbar	0xbc6
jot	0xbca
quad	0xbcc
uptack	0xbce
circle	0xbcf
upstile	0xbd3
downshoe	0xbd6
rightshoe	0xbd8
leftshoe	0xbda
lefttack	0xbdc
righttack	0xbfc

Hebrew

hebrew_doublelowline	0xcdf
hebrew_aleph	0xce0
hebrew_bet	0xce1
hebrew_beth	0xce1 /* deprecated */
hebrew_gimel	0xce2
hebrew_gimmel	0xce2 /* deprecated */
hebrew_dalet	0xce3
hebrew_daleth	0xce3 /* deprecated */
hebrew_he	0xce4
hebrew_waw	0xce5
hebrew_zain	0xce6
hebrew_zayin	0xce6 /* deprecated */
hebrew_chet	0xce7
hebrew_het	0xce7 /* deprecated */
hebrew_tet	0xce8
hebrew_teth	0xce8 /* deprecated */
hebrew_yod	0xce9
hebrew_finalkaph	0xcea
hebrew_kaph	0xceb
hebrew_lamed	0xcec
hebrew_finalmem	0xced
hebrew_mem	0xcee
hebrew_finalnun	0xcef
hebrew_nun	0xcf0
hebrew_samech	0xcf1
hebrew_samekh	0xcf1 /* deprecated */
hebrew_ayin	0xcf2
hebrew_finalpe	0xcf3
hebrew_pe	0xcf4
hebrew_finalzade	0xcf5
hebrew_finalzadi	0xcf5 /* deprecated */
hebrew_zade	0xcf6
hebrew_zadi	0xcf6 /* deprecated */
hebrew_qoph	0xcf7
hebrew_kuf	0xcf7 /* deprecated */
hebrew_resh	0xcf8

hebrew_shin	0xcf9
hebrew_taw	0xcfa
hebrew_taf	0xcfa /* deprecated */
Hebrew_switch	0xFF7E /* Alias for
mode_switch*/	

XKeysymDB

hpmute_grave	0x10000A9
hpmute_asciicircum	0x10000AA
hpmute_diaeresis	0x10000AB
hpmute_asciitilde	0x10000AC
hplira	0x10000AF
hpguilder	0x10000BE
hpYdiaeresis	0x10000EE
hpIO	0x10000EE
hplongminus	0x10000F6
hpblock	0x10000FC
apLineDel	0x1000FF00
apCharDel	0x1000FF01
apCopy	0x1000FF02
apCut	0x1000FF03
apPaste	0x1000FF04
apMove	0x1000FF05
apGrow	0x1000FF06
apCmd	0x1000FF07
apShell	0x1000FF08
apLeftBar	0x1000FF09
apRightBar	0x1000FF0A
apLeftBox	0x1000FF0B
apRightBox	0x1000FF0C
apUpBox	0x1000FF0D
apDownBox	0x1000FF0E
apPop	0x1000FF0F
apRead	0x1000FF10
apEdit	0x1000FF11
apSave	0x1000FF12
apExit	0x1000FF13
apRepeat	0x1000FF14
hpModelock1	0x1000FF48
hpModelock2	0x1000FF49
hpReset	0x1000FF6C
hpSystem	0x1000FF6D
hpUser	0x1000FF6E
hpClearLine	0x1000FF6F
hpInsertLine	0x1000FF70
hpDeleteLine	0x1000FF71
hpInsertChar	0x1000FF72
hpDeleteChar	0x1000FF73
hpBackTab	0x1000FF74
hpKP_BackTab	0x1000FF75
apKP_parenleft	0x1000FFA8
apKP_parenright	0x1000FFA9

I2ND_FUNC_L	0x10004001	osfPageUp	0x1004FF41
I2ND_FUNC_R	0x10004002	osfPageDown	0x1004FF42
IREMOVE	0x10004003	osfPageRight	0x1004FF43
IREPEAT	0x10004004	osfActivate	0x1004FF44
IA1	0x10004101	osfMenuBar	0x1004FF45
IA2	0x10004102	osfLeft	0x1004FF51
IA3	0x10004103	osfUp	0x1004FF52
IA4	0x10004104	osfRight	0x1004FF53
IA5	0x10004105	osfDown	0x1004FF54
IA6	0x10004106	osfEndLine	0x1004FF57
IA7	0x10004107	osfBeginLine	0x1004FF58
IA8	0x10004108	osfEndData	0x1004FF59
IA9	0x10004109	osfBeginData	0x1004FF5A
IA10	0x1000410A	osfPrevMenu	0x1004FF5B
IA11	0x1000410B	osfNextMenu	0x1004FF5C
IA12	0x1000410C	osfPrevField	0x1004FF5D
IA13	0x1000410D	osfNextField	0x1004FF5E
IA14	0x1000410E	osfSelect	0x1004FF60
IA15	0x1000410F	osfInsert	0x1004FF63
IB1	0x10004201	osfUndo	0x1004FF65
IB2	0x10004202	osfMenu	0x1004FF67
IB3	0x10004203	osfCancel	0x1004FF69
IB4	0x10004204	osfHelp	0x1004FF6A
IB5	0x10004205	osfSelectAll	0x1004FF71
IB6	0x10004206	osfDeselectAll	0x1004FF72
IB7	0x10004207	osfReselect	0x1004FF73
IB8	0x10004208	osfExtend	0x1004FF74
IB9	0x10004209	osfRestore	0x1004FF78
IB10	0x1000420B	osfDelete	0x1004FFFF
IB11	0x1000420B	SunFA_Grave	0x1005FF00
IB12	0x1000420C	SunFA_Circum	0x1005FF01
IB13	0x1000420D	SunFA_Tilde	0x1005FF02
IB14	0x1000420E	SunF36	0x1005FF10
IB15	0x1000420F	SunF37	0x1005FF11
IB16	0x10004210	SunSys_Req	0x1005FF60
DRemove	0x1000FF00	SunProps	0x1005FF70
Dring_accent	0x1000FEB0	SunFront	0x1005FF71
Dcircumflex_accent	0x1000FE5E	SunCopy	0x1005FF72
Dcedilla_accent	0x1000FE2C	SunOpen	0x1005FF73
Dacute_accent	0x1000FE27	SunPaste	0x1005FF74
Dgrave_accent	0x1000FE60	SunCut	0x1005FF75
Dtilde	0x1000FE7E	SunCompose	0xFF20
Ddiaeresis	0x1000FE22	SunPageUp	0xFF55
osfCopy	0x1004FF02	SunPageDown	0xFF56
osfCut	0x1004FF03	SunPrint_Screen	0xFF61
osfPaste	0x1004FF04	SunUndo	0xFF65
osfBackTab	0x1004FF07	SunAgain	0xFF66
osfBackSpace	0x1004FF08	SunFind	0xFF68
osfClear	0x1004FF0B	SunStop	0xFF69
osfEscape	0x1004FF1B	SunAltGraph	0xFF7E
osfAddMode	0x1004FF31	WYSetup	0x1006FF00
osfPrimaryPaste	0x1004FF32		
osfQuickPaste	0x1004FF33		
osfPageLeft	0x1004FF40		

付録 F

X Vision を使用するためのヒントと秘訣

このノートでは、X Vision を使用して、さまざまなオペレーティングシステムでアプリケーションを実行するための詳しい情報を示します。

このセクションの内容は、次のとおりです。

- UNIX 全般
- AIX
- Data General
- DEC OSF/1
- DEC Ultrix
- HP-UX
- SCO-ODT
- Solaris
- SunOS
- UnixWare
- XRemote

UNIX 全般

DISPLAY 環境変数を設定して、すべてのクライアントを PC に表示させます。シェルに応じて、次のいずれかの方式を使用します。

cshtcsh

```
setenv DISPLAY pcname:0
```

sh/bash

```
DISPLAY=pcname:0
```

```
export DISPLAY
```

ksh

```
DISPLAY=pcname:0.0
```

```
export DISPLAY
```

AIX

このセクションの内容は、次のとおりです。

- AIX フォント
- キーパッドのキー
- AIX での印刷サービスの使い方

AIX フォント

アプリケーションで PC に設定されていないフォントを使用する場合は、目的のフォントを AIX システムから PC にコピーして、X Vision フォントコンパイラを使用し X サーバで使用できるフォントを生成してください。フォントコンパイラでは PCF 形式または BDF 形式の X フォントをコンパイルでき、Microsoft Windows の .fon 形式のファイルを作成します。

フォントの詳細については、「X サーバの使い方」の章の「フォントの使い方」を参照してください。

キーパッドのキー

キーパッドのキーで **aixterm** に正しい出力を生成できない場合は、**aixterm -v** を実行してみてください。

AIX での印刷サービスの使い方

X Vision のソフトウェアで提供される印刷サービスは、AIX に対してネットワークプリンタサーバとしての機能を果たすことができます。UNIX システム上のプリンタを、次のように構成してください。

- 1 ルートとしてログインします。
- 2 コマンド **smit** (System Management Interface Tool) を入力します。
PC から X バージョンの *smit* を実行している場合は、X サーバが動作していません。文字端末バージョンも使用できますが、このノートでは X バージョンを使用するものとします。
- 3 [デバイス] を選択します。
- 4 [プリンタ/プロッタ] を選択します。
- 5 [印刷スプーリング] を選択します。
- 6 [印刷待ち行列の追加] を選択します。
- 7 ダイアログから [remote] を選択します。
- 8 [Add a Remote Queue] を選択します。
- 9 ダイアログボックスに次のように指定します。

Name of queue to add	pcprinter
Activate the queue?	yes
Will this become the default queue?	no
Queueing discipline	first come first served
Accounting file pathname	/tmp/account_pcprinter
Destination host for remote jobs	<i>your_pcname</i> (stevel など)
Pathname of the short form filter for queue status output	/usr/lpd/aixshort
Pathname of the long form filter for queue status output	/usr/lpd/aixlong
Name of queue on remote printer	<i>pc_port</i> (lpt1 など)
Name of device to add	lp1
Backend program pathname	/usr/lpd/sembak

各キューは一意でなければならないため、複数の PC ユーザが各自の PC に対応したキューを設定する場合、異なるキュー名を使用しなければなりません。また、必要に応じて、キューを AIX システムのデフォルトとすることもできます。

- 10 [Do] を選択するか、ENTER を押します。
要求が処理されるまで待ち、OK メッセージが表示されたら、*smit* を終了します。

- 11 ルートシステムプロンプトで、次のように入力して AIX マシンで *lpd* が実行されていることを確認します。

```
lpd
```

- 12 次のように入力して、新しいプリンタを有効にします。

```
enablepprinter
```

これで、UNIX プリンタの準備が整いました。ファイルを印刷するには、次のコマンドを使用します。

lpr -Ppprinter file_name

プリントサーバは、PC で実行されていない場合には自動的に起動されます。ファイルは Windows のデフォルトのプリンタに出力されます (なお、デフォルトのプリンタをテスト用のファイルとして設定できます)。

なお、プリントサーバでは Windows のプリンタドライバを使用しないので、プリントファイルはプリンタで必要とする形式でなければなりません。たとえば、PostScript プリンタの場合は、UNIX フィルタを使用してテキストファイルを処理してから、プリンタに送送する必要があります。

DEC OSF/1

DECwindows クライアントには、他の UNIX バージョンと同じ命名規約は適用されません。通常クライアント名の前に `ÅgdÅh` が付けられます。たとえば、UNIX System V コンピュータ上のクライアント `xterm` は、DEC Ultrix システムでは `dxterm` になります。

DECwindows クライアント名をチェックするには、Ultrix システムプロンプトで次のコマンドを入力します。

```
ccj/usr/bin/X11
```

```
♫
```

目的のクライアントがディレクトリリストにない場合は、そのクライアント名の先頭に `ÅgdÅh` を付けて検索してください。

DEC Ultrix

DECwindows クライアントには、他の UNIX バージョンと同じ命名規約は適用されません。通常クライアント名の前に `ÅgdÅh` が付けられます。たとえば、UNIX System V コンピュータ上のクライアント `xterm` は、DEC Ultrix システムでは `dxterm` になります。

DECwindows クライアント名をチェックするには、Ultrix システムプロンプトで次のコマンドを入力します。

```
ccj/usr/bin/X11
```

```
♫
```

目的のクライアントがディレクトリリストにない場合は、そのクライアント名の先頭に “d” を付けて検索してください。

HP-UX

このセクションの内容は、次のとおりです。

- XDMCP の使い方
- HP VUE で使用するフォント
- VUE ファイルマネージャが起動しない場合
- HP-UX での印刷サービスの使い方

HP-UX での XDMCP の使い方

Xサーバがマルチウィンドウモードで実行している場合、XDMCP から X セッションを開始できないことがあります。これは、Vision ウィンドウマネージャ (VWM) がアクティブであるため、ログインおよびセッションを開始するためのプログラムと競合するからです。HINTS が使用可能であれば、次のメッセージが表示されます。

```
“The application that has just terminated generated the following error: BadAccess
```

```
This error occurred while processing the following request: ChangeWindowAttributes”
```

この問題を解消するには、シングルウィンドウモードで実行できるように X サーバを構成してください。

HP VUE で使用するフォント

Xサーバに添付されている HP フォントを使用すれば、ほとんどの HP アプリケーションを実行できます。ただし、標準セットアップを使用する場合は、このフォントはインストールされません。カスタムセットアップを使用してフォントを選択してから Xサーバをインストールするか、[スタート] メニューフォルダまたはプログラムマネージャグループから Maintenance Setup を実行して既存のインストールにフォントを追加してください。

HP VUE で使用されるフォントがすべて Xサーバに添付されているわけではありません。Xサーバに存在しないフォントを使用するには、UNIX システムで HP フォントサーバを実行し、このフォントサーバを使用するように Xサーバを設定します。

VUE ファイルマネージャが起動されない場合

VUE 内から VUE ファイルマネージャを起動すると、エラーになり、次のメッセージが表示されることがあります。

```
VUE-WM:Message server refused connection due to inetd.sec
```

これは、PC 上の TCP/IP スタックには 5 つ以上の TCP 接続を構成できるものの、一度に開くことができる X ウィンドウは 5 つしかないためです。しかし、VUE ファイルマネージャを正しく実行するには、これ以上の接続が必要です。

これは、**vuefile** アプリケーションで使用されるネットワークソフトウェアメッセージサーバのバグです。メッセージサーバのパッチは今後提供される予定です。最新情報については、HP のテクニカルサポートにお問い合わせください。

X サーバを HP 9000/847 ホストのパッチが済んでいない HP-UX 9.04 で使用するためには、メッセージサーバのセキュリティ機能を無効にしてください。/usr/adm/inetd.sec ファイルに次の行を追加します。

```
mesgdallow
```

HP-UX での印刷サービスの使い方

X Vision のソフトウェアで提供される印刷サービスは、HP-UX に対してネットワークプリンタサーバとしての機能を果たすことができます。UNIX システム上のプリンタを、次のように構成してください。

- 1 ルートとしてログインします。
- 2 コマンド **sam** (System Administration Manager) を入力します。
- 3 [Printers and Plotters] に該当するものを選択して、[開く] を選択します。
- 4 [Printers/Plotters] に該当するものを選択して、[開く] を選択します。
- 5 [Actions] メニューから、[Add Remote Printer/Plotter] を選択します。
- 6 [プリンタ名] ボックスに、リモートプリンタ名を入力します。プリンタを接続する PC 名を指定できます。[Remote System Name] ボックスに、プリンタを接続する PC のネットワークノード名を入力します。次に、[Remote Printer Name] ボックスに、Windows のデフォルトのプリンタを意味する **[Default]** と入力します。この際、大文字と小文字を区別し、正しいスペルで入力してください。

さらに、[Make This The System Default Printer] オプションと [Allow Anyone to Cancel a Request] オプションの両方または一方も選択できます。ただし、[Remote Printer is on a BSD System] オプションは選択できません。

[OK] を選択して UNIX システムでプリンタを作成すると、/usr/spool/lp/interface ディレクトリに新しいファイルが作成され、[プリンタ名] に指定した名前と同じ名前が付けられます。

- 7 テキストエディタでファイルを開き、次の行を検索します。

```
shift;shift;shift;shift;shift
```

この行の後に、次の 3 行を追加します。

```
/usr/bin/ux2dos$1>/tmp/tmp$$
mv /tmp/tmp$$ $1
echo `^L` >> $1
```

`ux2dos` コマンドは `/usr/bin` ディレクトリに常駐する HP-UX の標準コマンドで、キャリッジリターン文字を行終端に追加します (これは、UNIX では LF を使用し、DOS では CR LF を使用するためです)。

エコーコマンドはファイルの終わりに用紙送り文字を追加します。これは、プリンタにファイルを送信するために必要です。この行には制御コードが含まれているため、通常のテキストとしては追加できません。次のように指定してください。

```
echo `ctrl-Vctrl-L` >> $1
```

これで、UNIX プリンタの準備が整いました。ファイルを印刷するには、次のコマンドを使用します。

```
lp -d printer_name file_name
```

プリントサーバは、PC で実行されていない場合には自動的に起動されます。ファイルは Windows のデフォルトのプリンタに出力されます (なお、デフォルトのプリンタをテスト用のファイルとして設定できます)。

なお、プリントサーバでは Windows のプリンタドライバを使用しないので、プリントファイルはプリンタで必要とする形式でなければなりません。たとえば、PostScript プリンタの場合は、UNIX フィルタを使用してテキストファイルを処理してから、プリンタにサブミットする必要があります。

SCO-ODT

このセクションの内容は、次のとおりです。

- SCO-ODT との接続
- XDMCP の使い方
- SCO-ODT での印刷サービスの使い方

SCO-ODT との接続

SCO-ODT では他の UNIX システムと多少異なる方法でログインプロシージャを処理するため、標準的な ComScript ファイルを使用すると問題が発生することがあります。シリアル接続の場合は、`Unixslsc.scr` ファイルを使用してください。

XDMCP の使い方

X サーバがマルチウィンドウモードで実行している場合、XDMCP から X セッションを開始できないことがあります。これは、Vision ウィンドウマネージャ (VWM) がアクティブであるため、ログインおよびセッションを開始するためのプログラムと競合するからです。HINTS が使用可能であれば、次のメッセージが表示されます。

```
“The application that has just terminated generated the following error: BadAccess  
This error occurred while processing the following request:
```

```
ChangeWindowAttributes.”
```

この問題を解決するには、次の 2 つの方法があります。

- シングルウィンドウモードで動作するように X サーバを設定します。
- または -
- ホームディレクトリに `.startxrc` ファイルを作成し、次の行を指定します。

```
exec scoterm 2> /dev/null
```

SCO-ODT での印刷サービスの使い方

デフォルトの SCO UNIX オペレーティングシステムは、BSD システムではなく UNIX System V Release 3 の印刷システムを使用します。ただし、SCO にはごく一部の BSD `lpd` サービスをインストールするためのメカニズムが提供されているため、このメカニズムを使用すれば、接続されている PC 上のプリントサーバにアクセスできます。UNIX システム上のプリンタを、次のように構成してください。

- 1 ルートとしてログインします。
- 2 コマンド `mkdev rlp` を入力します。
- 3 リモートプリンタをインストールするのか、それとも削除するのかが求められます。ここではインストールするため、“i”と入力します。`/usr/lpd/remote` ディレクトリ、`/usr/spool/lpd` ディレクトリ、および `/etc/printcap` ファイルが作成されます。また、`lp` コマンド、`cancel` コマンド、`lpstat` コマンドもインストールされます。
- 4 次に、“Do you wish to to change the printer description file /etc/printcap?” というメッセージに対して“y”と入力して、リモートプリンタを構成できるようにします。
- 5 次のように指定して、リモートプリンタを構成します。

Please enter the printer name (q to quit):

your_printername (Tims_Printer など)

Is Tims_Printer a remote printer or a local printer (r/l)? r

Please enter the name of the remote host that Tims_Printer is attached to:

your_pcname (timp など)

指定したプリンタが対象のホストに正しく接続されているか確認が求められます。このままで良い場合は“y”を入力し、リモートホスト名を再度入力する場合は“n”を入力します。

- 6 構成したプリンタをシステムのデフォルトにするかどうか選択します。必要に応じて“y”または“n”を指定します。設定したプリンタ情報が `/etc/printcap` ファイルに追加され、プリンタのスプーリングディレクトリが作成されます。

他にもプリンタを構成する場合はステップ 5 を再度行い、処理を終了する場合は“q”と指定します。

- 7 “Do you want to start remote daemon now?” に対して“y”と入力します。

- 8 テキストエディタで `/etc/printcap` ファイルを開きます。このファイルには、`rlpconf` コマンドを使用して構成した各プリンタに対するエントリが格納されます。`rp` 機能でリモートプリンタが **Default** (Windows のデフォルトのプリンタ) として指定されるように、各エントリを変更する必要があります。たとえば、`Tims_Printer` と `Steves_Printer` の 2 つのリモートラインプリンタを構成し、この 2 つのプリンタが `timp` と `stevel` という名前の 2 つの PC に接続する場合、`/etc/printcap` ファイルの内容は次のようになります。

```
# Remote Line Printer (BSD format)
Tims_Printer:¥
:lp=:rm=timp:rp=Tims_Printer:sd=/usr/spool/lpd/Tims_Printer:
Steves_Printer:¥
:lp=:rm=stevel:rp=Steves_Printer:sd=/usr/spool/lpd/Steves_Printer:
```

このエントリを次のように変更します。

```
# Remote Line Printer (BSD format)
Tims_Printer:¥
:lp=:rm=timp:rp=Default:sd=/usr/spool/lpd/Tims_Printer:
Steves_Printer:¥
:lp=:rm=stevel:rp=Default:sd=/usr/spool/lpd/Steves_Printer:
```

大文字と小文字を区別し、正しいスペルで `ÅgDefaultÅh` と入力してください。必ずこのようにプリンタエントリを変更してから、印刷要求をプリンタに送信してください。この変更を行っていない場合は、`/etc/printcap` ファイルを開き、該当するエントリを修正した後、ルートプロンプトから次のように指定してください。

```
lpdstop
lpd start
```

LPD サーバが PC で実行されていれば、これでファイルが出力されます。

- 9 後でリモートプリンタを追加する場合は、ルートとしてログインし、次のように入力します。

```
/etc/rlpconf
```

次に行う処理については、ステップ 5、6、8 を参照してください。

これで、UNIX プリンタの準備が整いました。ファイルを印刷するには、次のコマンドを使用します。

lp -d printer_name file_name

プリントサーバは、PC で実行されていない場合には自動的に起動されます。ファイルは Windows のデフォルトのプリンタに出力されます (なお、デフォルトのプリンタをテスト用のファイルとして設定できます)。

なお、プリントサーバでは Windows のプリンタドライバを使用しないので、プリントファイルはプリンタで必要とする形式でなければなりません。たとえば、PostScript プリンタの場合は、UNIX フィルタを使用してテキストファイルを処理してから、プリンタにサブミットする必要があります。

Solaris

このノートでは、AnswerBook の使い方について説明します。AnswerBook はオンラインで Sun のマニュアルにアクセスできます。AnswerBook では、まずナビゲータウィンドウを使用して、使用可能なブックとその内容を一覧表示します。そして、ビューアウィンドウを使用して、選択されたページを実際に表示します。Sun のワークステーションで AnswerBook を実行した場合に使用されるビューアは **docviewer** で、Display PostScript を使用して、ページのイメージが画面に描画されます。PC では Display PostScript が提供されていないため、別の方法によって PostScript ドキュメントページを表示する必要があります。

このノートでは、2 つの方式について説明します。どちらの場合でも、UNIX システムに GhostScript をインストールする必要があります。GhostScript がない場合は、匿名 FTP を使用してインターネットからダウンロードできます。

FTP サーバ: src.doc.ic.ac.uk

位置: /gnu

ファイル名: ghostscript-2.6.2.tar.gz

拡張子 **.gz** が付けられているファイルは、GNU zip を使用して圧縮されています。このため、UNIX システムに **gzip** または **gunzip** がインストールされていなければなりません。このファイルを解凍したら、tar ファイルを復元し、**README** ファイルを参照してください。

どちらの方式の場合でも、次に示すように、**DISPLAY** 環境変数を PC を指すように設定し、**BOOKINFO** 変数は AnswerBook の **bookinfo** ファイルの位置に設定しなければなりません。

```
setenv DISPLAY mypc:0
```

```
setenv BOOKINFO answerbook_directory/SysSoft1_4/bookinfo
```

方式 1: xnav

この方式ではハイパーテキストリンクを提供しますが、異なる PostScript ファイル間では使用できません。また、検索機能もありません (方式 2 では検索機能を使用できます)。この方式では、**xnav** プログラムを使用します。このプログラムは次の位置からダウンロードできます。

FTP サーバ: src.doc.ic.ac.uk

位置: /computing/systems/sun/sun-faq/Source

ファイル名: ab.tar.gz

ファイルを解凍し、復元したら、**README** ファイルを参照してください。

プログラムをコンパイルする前に、Imakefile ファイルを次のように 3ヶ所変更する必要があります。

`srcdir/Imakefile`

次の行を追加します。

```
INCLUDES=-I/usr/openwin/include
```

`srcdir/gscanvas/Imakefile`

`srcdir/dc/Imakefile`

INCLUDES 行を次のように変更します。

```
INCLUDES=-I/usr/openwin/include
```

次のように入力して、プログラムをコンパイルします。

```
xmkmf;makeMakefiles;make
```

ps.c の 786 行目に警告メッセージが 1 行表示されますが、これは無視してかまいません。

xnav を起動するために、次のコマンドを入力します。

```
xnav SUNWab_1_4
```

ナビゲータウィンドウが表示され、AnswerBooks の内容を表示できます。[目次] のエントリをダブルクリックすると、ビューアウィンドウに該当するページが表示されます (場合によっては、ビューアウィンドウにフォーカスを当てる必要があります)。ビューアウィンドウで [Next Page] ボタンおよび [Previous Page] ボタンを使用できます。また、ハイパーテキストリンクが提供され、該当するテキストが矩形で囲まれます。リンクをダブルクリックすると、新しいトピックにジャンプします。

方式 2: xdocviewer

この方式ではサーチ機能は提供されませんが、ハイパーテキストリンクは使用できません。この方式では、`xdocviewer` スクリプトを使用します。このスクリプトは次の位置からダウンロードできます。

FTP サーバ: `src.doc.ic.ac.uk`

位置: `computing/graphics/systems/X11/R5contrib`

ファイル名: `xdocviewer-2.0.1.README`

ファイル名: `xdocviewer-2.0.1.z`

README ファイルを参照して、スクリプトのベースとなるオペレーティングシステムバージョンにおける制約事項を確認してください。

スクリプトファイル名に `.z` という拡張子が付いている場合は、UNIX の標準的な **compress** プログラムを使用してファイルが圧縮されています。このファイルを解凍するには、**compress** に続きファイル名を入力します。

xdocviewer スクリプトは Perl で作成されるため、このスクリプトを解釈するには、UNIX システムに Perl がインストールされていなければなりません。Perl がインストールされていない場合は、以下からダウンロードできます。

FTP サーバ: src.doc.ic.ac.uk

位置: /gnu

ファイル名: perl5.002.tar.gz

場合によっては、**xdocviewer** スクリプトを編集して、Perl の位置 (デフォルトは `/usr/local/bin/perl`) を指すように変数を設定する必要があります。デフォルトの **xdocviewer** スクリプトでは、GhostScript ビューア *ghostview* をドキュメントビューアとして実行します。**gspreview** も使用でき、さらにスクリプトを変更するか、**GSVIEWER** 環境変数を設定すれば、ビューアを変更することも可能です。

Perl スクリプトを実行可能にし、名前を **docviewer** に変更した後、元の **docviewer** プログラムよりも前のパス内の位置に移動します。

次に、**answerbook** スクリプトをコピーし、これに **xanswerbook** という名前を付けます。Perl スクリプトのコメントに従い、パスが間違っても変更されないように、新しいスクリプトを編集します。このような変更の詳細については、**xdocviewer-2.0.1.README** ファイルを参照してください。

次のコマンドを入力して、スクリプトを実行します。

```
xanswerbook
```

ナビゲータウィンドウが表示され、AnswerBooks の内容を表示できます。[目次] のエントリをダブルクリックすると、GhostScript ビューアに該当するページが表示されます。ナビゲータウィンドウで適切なボタンを選択することで、検索が可能です。

SunOS

このセクションの内容は、次のとおりです。

- OPEN LOOK ボタンが表示されない場合
- Interrupted System Call エラー

OPEN LOOK ボタンが表示されない場合

OPEN LOOK のウィンドウマネージャである OLWM を使用すると、ボタンのフォントが画面に表示されないことがあります。特に、メニュー内のオプションの回りにサムタック (画鋏) と楕円形が表示されないことがあります。16 色の VGA ディスプレイアダプタの使用時にこの現象が起きた場合は、ウィンドウマネージャが色を割り当てることができない可能性があります。この場合、ウィンドウマネージャは使用可能な色を使用して、ボタンフォントとメニューの背景を同じ色で描画します。このため、オブジェクトを描画しても画面に表示されません。

この問題を解消するには、[Colors Properties] ダイアログボックスの [Windows System Colors] オプションを無効にしてください。このオプションが有効であると、カラーマップに使用可能な色がなくなります。

Interrupted System Call エラー

OpenWindows 3 環境で、**shelltool** または **cmdtool** の実行時に `^GInterrupted System Call` エラーが表示される場合は、`.cshrc` ファイルに `stty` ステートメントが設定されている可能性があります。このステートメントを `.login` ファイルに移動してください。

OpenWindows 2 環境で、**xterm** を実行すると表示されないのに、**shelltool** または **cmdtool** を実行すると `^GInterrupted System Call` エラーが表示される場合があります。この問題を解消するには、次のように、**shelltool** または **cmdtool** の実行時に `setuid` を実行してください。

```
/usr/etc/setuidusr/openwin/bin/cmdtool-displayname:0
```

この原因は制御端末に関係します (詳細については、`man setuid` を参照してください)。

XRemote

このセクションの内容は、次のとおりです。

- 端末サーバを介した XRemote の使い方
- シリーズ 800 HP-UX 9.0 での XRemote の使い方
- BSD UNIX での Xremote の使い方

端末サーバを介した XRemote の使い方

端末サーバを介して XRemote を使用する場合は、特別な ComScript ファイルについてテクニカルサポートにおたずねください。

シリーズ 800 HP-UX 9.0 での XRemote の使い方

HP-UX 9.0 では、デフォルトのシリーズ 800 X サーバはサポートされていません。XRemote を HP-UX 9.0 を実行している HP 9000 シリーズ 800 マシンで使用するには、次のディレクトリを作成する必要があります。

```
/usr/spool/sockets/X11
```

なお、このディレクトリの内容は不要で、ディレクトリ自体がただ存在すればよいだけです。

BSD UNIX での XRemote の使い方

XRemote では PC とリモートホスト間を結ぶ 8 ビットデータバスが必要となります。このために、ホストポートは手動で構成できます。`/etc/gettytab` ファイルに新しいエントリを作成し、`/etc/ttytab` ファイルでこのエントリを参照する必要があります。

- 1 ルートとしてログインします。
- 2 テキストエディタで `/etc/ttytab` ファイルを開きます。 `ttytab` ファイルでは、UNIX システムでどの `ttys` (端末デバイス) が使用できるかを設定します。`/etc/ttytab` ファイルには、通常次のような行が指定されます。

```
ttya  `~/usr/etc/getty std.9600~ unknown on local secure
```

この例では、`ttya` (シリアルポート A) で、`/usr/etc/getty` プログラム (端末回線を開き初期化して、ログイン名を読み取り、`login` を起動します) が引数 `is td.9600i` で実行されています。この引数 (使用するボーレートコード名) は、次のように `/etc/gettytab` ファイルに設定されています。

```
2|std.9600|9600-baud:¥
      :sp#9600:
```

このエントリでは、`sp` フィールドを使用して、回線速度を 9600 ボーに設定します。

`/etc/ttytab` ファイルの中で使用したいシリアルポートに関連する行をコピーして、その行の直前にハッシュ記号 (#) を入力してコメント行にします。

- 3 次に、コピーしたテキストをオリジナルのすぐ下にペーストします。この行の 2 番目のフィールドを、`getty` プログラムが `i8bit.9600i` という新しい引数で動作するように変更する必要があります。たとえば、シリアルポート A でリモートホストと接続するには、`ttytab` ファイルの行を次のように変更します。

```
ttya  `~/usr/etc/getty 8bit.9600~ unknown on local secure
```

新しい引数を作成したので、次に `/etc/gettytab` ファイルにこのためのエントリを作成する必要があります。

- 4 テキストエディタで `/etc/gettytab` ファイルを開きます。変更した引数のオリジナルのエントリを探しコピーした後、コピーしたテキストをオリジナルの下にペーストします。なお、コピー名を `i8bit.9600i` に変更する必要があります (各エントリの最初のフィールドは引数名です。引数名が複数ある場合は、パイプ (|) 文字で区切ります)。

- 5 8ビット接続を有効にするために、次のように、新しいエントリに `p8` フィールドとそれに続きコロン (:) を追加します。

```
8bit.9600:¥
      :sp#9600:p8:
```

- 6 ルートプロンプトから、次のように入力します。

```
kill-11
```

これによって、`/etc/ttytab` ファイルが変更されたために、ファイル内の新しい情報を読み取るように `init` に指定されます。その結果、`Init` は新しい引数で `/usr/etc/getty` コマンドを実行します。

モデム接続の場合は、次の `gettytab` ファイルエントリ例も使用できます。

```
Modem.9600:¥
      :sp#9600:p8:hc:to#30:
```

この場合にも、`sp` フィールドと `p8` フィールドを使用して、9600 ボーでの 8 ビット接続が設定されています。さらに、`hc` フィールドと `to` フィールドが指定されているため、最後に終了してから 30 秒以内に新しいユーザ名が指定されない場合、ダイヤルイン回線がハングアップします。

注意 一部の BSD UNIX システムでは、`/etc/ttytab` ファイルではなく `/etc/ttys` ファイルを使用します。

付録 G

端末エミュレータのキーマップ

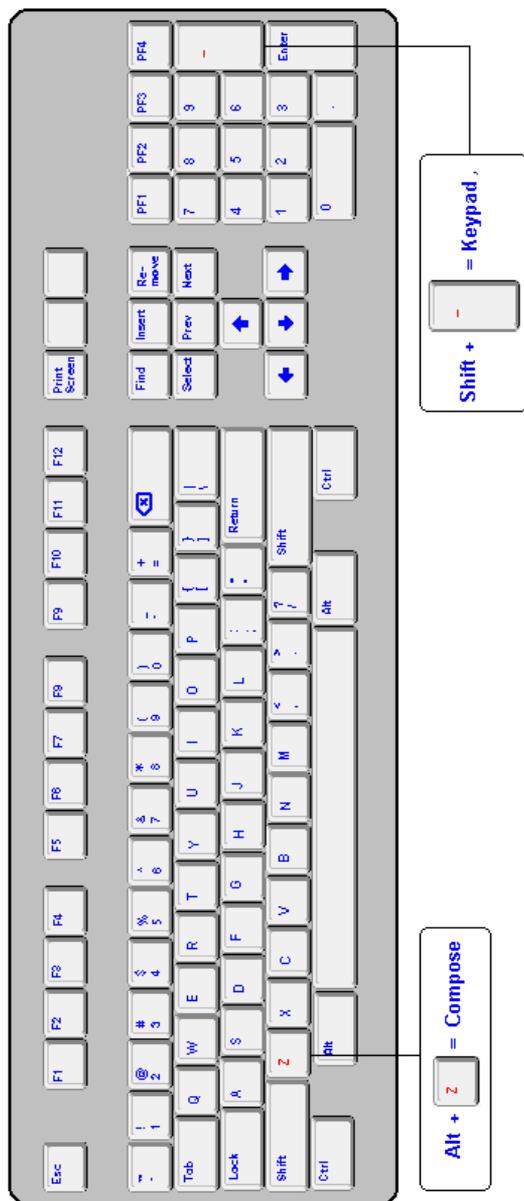
このセクションの内容は次のとおりです。

- IBM 拡張 101 キーボード上の VT420 端末キー
- IBM 拡張 102 キーボード上の VT420 端末キー
- 日本語 106 キーボード上の VT420 端末キー
- PC98 キーボード上の VT420 端末キー

IBM 拡張 101 キーボード上の VT420 端末キー

ハードウェアキーは割り当てられていません。

F13 ~ F20 のファンクションキーにアクセスするには、VT420 エミュレータの中で [ファンクションキー] ツールバー を使用する必要があります。



[ファンクションキー] ツールバー

[ファンクションキー] ツールバーにアクセスするには、VT420 エミュレータを起動し、右マウスボタンで標準ツールバーをクリックします。ショートカットメニューの [ファンクションキー] をクリックすると、項目のとなりチェックマークが表示されます。

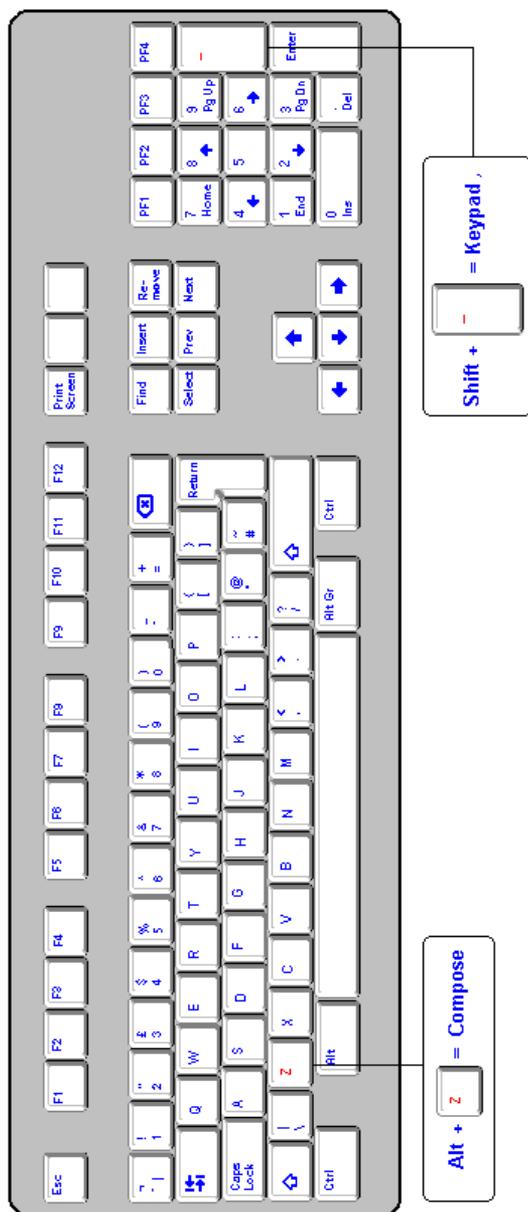


注意 キーボードから F13 ~ F20 のキーにアクセスしたい場合は、キーマップエディタで必要なマップを作成します。キーマップエディタでこのマップを作成しないと、キーと修飾子を組み合わせ使用できません。

IBM 拡張 102 キーボード上の VT420 端末キー

ハードウェアキーは割り当てられていません。

F13 ~ F20 のファンクションキーにアクセスするには、VT420 エミュレータの中で [ファンクションキー] ツールバー を使用する必要があります。



[ファンクションキー] ツールバー

[ファンクションキー] ツールバーにアクセスするには、VT420 エミュレータを起動し、右マウスボタンで標準ツールバーをクリックします。ショートカットメニューの [ファンクションキー] をクリックすると、項目のとなりチェックマークが表示されます。

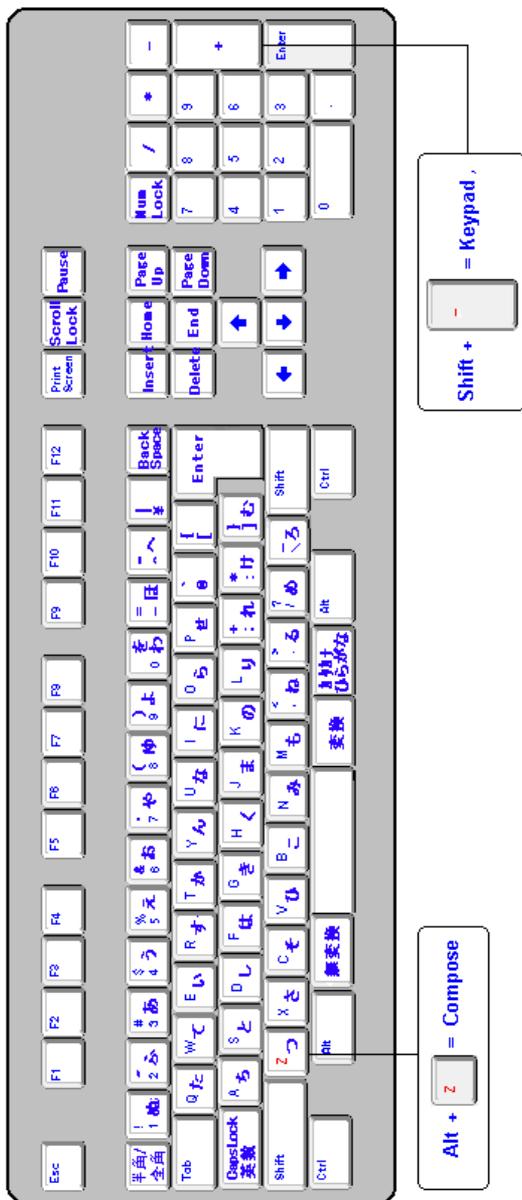


注意 キーボードから F13 ~ F20 のキーにアクセスしたい場合は、キーマップエディタで必要なマップを作成します。キーマップエディタでこのマップを作成しないと、キーと修飾子を組み合わせて使用できません。

日本語 106 キーボード上の VT420 端末キー

ハードウェアキーは割り当てられていません。

F13 ~ F20 のファンクションキーにアクセスするには、VT420 エミュレータの中で [ファンクションキー] ツールバーを使用する必要があります。



[ファンクションキー] ツールバー

[ファンクションキー] ツールバーにアクセスするには、VT420 エミュレータを起動し、右マウスボタンで標準ツールバーをクリックします。ショートカットメニューの [ファンクションキー] をクリックすると、項目のとなりチェックマークが表示されます。



注意 キーボードから F13 ~ F20 のキーにアクセスしたい場合は、キーマップエディタで必要なマップを作成します。キーマップエディタでこのマップを作成しないと、キーと修飾子を組み合わせ使用できません。

[ファンクションキー] ツールバー

[ファンクションキー] ツールバーにアクセスするには、VT420 エミュレータを起動し、右マウスボタンで標準ツールバーをクリックします。ショートカットメニューの [ファンクションキー] をクリックすると、項目のとなりチェックマークが表示されます。



注意 キーボードから F11 ~ F20 のキーにアクセスしたい場合は、キーマップエディタで必要なマップを作成します。キーマップエディタでこのマップを作成しないと、キーと修飾子を組み合わせて使用できません。

(空白のページです)

索引

記号

.bdf フォント 106
 .fon フォント 106
 .mwmrc ファイル 165
 .pcf フォント 106
 .tpinfo ファイル 204
 .tpsession ファイル 203
 .visionpath ファイル 262, 379
 .Xauthority ファイル 98, 97
 .Xdefaults-host ファイル 165
 .Xdefaults ファイル 165
 .xinitrc ファイル 197
 .xinitremoterc ファイル 185, 194, 197
 .xsession-errors ファイル 186
 .xsession ファイル 184, 185, 188
 _MOTIF_WM_HINTS プロパティ 158
 _MOTIF_WM_INFO プロパティ 160
 _MOTIF_WM_MENU プロパティ 154, 160
 _MOTIF_WM_MESSAGES プロトコル 159
 _MOTIF_WM_MESSAGES プロパティ 159
 _SUN_DRAGDROP_INTEREST プロパティ 160
 -display オプション 180
 100DPI フォント 107
 132 カラムスタイル 227
 75DPI フォント 106
 80 カラムスタイル 227

A

admcopu 343
 admexec 344
 admindex 345
 admls 346
 admupdate 348
 AIX
 印刷サービスの使い方 457
 キーボードのキー 456
 フォント 456
 ALT
 +key キーボードシーケンス 101
 Mod2 へのマッピング 139
 Andrew フォント 107
 ANSI 319

AnswerBook 464
 API 319
 ASCII コード 319
 authsrv 350

B

BDF フォント 319
 Bitmap Distribution Format フォント 106
 BREAK 関数
 ComScript 408
 browsesrv 351
 BSD UNIX
 での XRemote の使い方 468

C

CALL 関数
 ComScript 409
 chooser
 xdm 186, 187
 CLIPBOARD セレクションアトム 123
 COLA 319
 cola.dat ファイル 78, 380
 ComScript 319, 400
 BREAK 関数 408
 CALL 関数 409
 ComScript ファイルの開発 436
 ComScript ファイルの生成 438
 ComScript ファイルのテスト 437
 DELAY 関数 410
 DIALOG 関数 411
 ENDFUNC ステートメント 404
 ERROR 関数 412
 EXECUTE 関数 413
 FUNC ステートメント 404
 IF...THEN...ELSE ステートメント 403
 ISSET 関数 415
 LONGBREAK 関数 416
 LOOP ステートメント 403
 MATCH 関数 405, 417
 NOMATCH 関数 405, 419
 NOREMATCH 関数 405, 421
 NOTCALL 関数 423
 NOTSET 関数 424
 PASSWORD 関数 425
 REMATCH 関数 426, 405

RETURN 関数 428
 SAVEPASSWORD 関数 429
 SEND 関数 431
 STATUS 関数 432
 TRACE 関数 434
 関数の概要 407
 繰り返し処理 403
 コメントテキスト 406
 サンプルファイル 435
 条件付きで実行 403
 照合文字列 405
 ダイアルバックの例 439
 パラメータ 405
 ファイル 400
 ユーザ定義の関数 404
 ComScript ファイルの例 435
 ダイアルバック 439
 Connect 関数
 OLE オートメーション 243
 Copy Area コマンド 125

D

DBMS 319
 DEC OSF/1 458
 DEC Ultrix 458
 DECnet 319
 DECwindows 319
 DELAY 関数
 ComScript 410
 DESC キーワード 134
 DIALOG 関数
 ComScript 411
 Disconnect 関数
 OLE オートメーション 244
 Display Id
 XDM-AUTHORIZATION-1 セキュリティのための
 98, 183
 DISPLAY 変数 180
 DISPLAY変数
 Vision Resumeセッションにおける 206
 domain.dat ファイル 77, 78, 381
 dumpspath 259, 262, 352

E

EGA ディスプレイ 120
 ENDFUNC ステートメント
 ComScript 404
 ERROR 関数

ComScript 412
 EXECUTE 関数
 ComScript 413

F

FILE_NAMES_ALIASES 113
 Fonts.ali ファイル 105, 113, 114
 別名の作成 110
 Fonts.dir ファイル 109, 112, 114
 作成 109
 FTP ファイルアクセス 262
 FUNC ステートメント
 ComScript 404

G

Global aliases 107
 GSO. グラフィックススピードオブティマイザを参照

H

Helpful Interactive Suport 127
 HINTS 320. *Helpful Interactive Suport* も参照
 hints
 Motif 152
 hostmgr コマンド 273
 HP VUE フォント 459
 HP-UX
 XDMCP の使い方 459
 印刷サービスの使い方 460
 での XRemote の使い方 467
 フォント 459
 HP フォント 107

I

ICCCM 320. Inter-Client Communications Conventions
 Manual も参照
 ICL フォント 107
 IF...THEN...ELSE ステートメント
 ComScript 403
 in-place モードでの起動 230
 infosrv サービス 310
 Inter-Client Communications Conventions Manual 151
 IPX 320
 IP アドレス 320
 ISO 320
 ISSET 関数

ComScript 415

K

keysyms 99, 320
 APL 452
 Arabic 449
 Cyrillic 449
 Greek 450
 Hebrew 453
 Katakana 448
 Latin1 445
 Latin2 447
 Latin3 447
 Latin4 448
 miscellany 444
 Publishing 452
 Special 451
 Technical 451
 XKeysymDB 453

L

LAN 320
 licadmin 353
 license_event スクリプト 79
 licsrv 355
 licsrv.dat ファイル 78, 383
 LNA. ローカルネームエージェントを参照
 lna 357
 Load 関数
 OLE オートメーション 245
 logsrv 358
 logsrv サーバ 81
 LONGBREAK 関数
 ComScript 416
 LOOP ステートメント
 ComScript 403
 Lower コマンド (コントロールメニュー)
 X サーバと Motif の互換性 156
 LPD プロトコル 286, 292, 320

M

起動
 リモートプログラムスタータで X アプリケーションを 181
 MATCH 関数
 ComScript 405, 417
 MISC フォント 106
 MIT-MAGIC-COOKIE-1 97, 183

Mod2
 ALT キーへのマッピング 139
 Mode Switch
 キーマップテーブル 134
 Motif ウィンドウマネージャ
 起動 164
 Motif 162
 hints 152
 Vision ウィンドウマネージャの互換性 156
 ウィンドウのマッピング 152
 ウィンドウマネージャの関数 161
 終了 165
 と X サーバ 163
 ヒント 154
 メニュー 154
 Motif ウィンドウマネージャ 92, 162, 163
 ローカル 163
 MWM. Motif ウィンドウマネージャを参照
 Mwmrc.txt ファイル 165

N

NCP 320
 NDIS 320
 NetBEUI 321
 NetBIOS 321
 newterm 359
 NOMATCH 関数
 ComScript 405, 419
 NOREMATCH 関数
 ComScript 405, 421
 NOTCALL 関数
 ComScript 423
 NOTSET 関数
 ComScript 424
 NSE. ネームスペース環境を参照
 NSM. ネームスペースマネージャを参照
 nsmadmin 360
 nsmrpcsrv 67, 361
 nsmsrv 67, 362
 Num Lock 管理 103
 Local NumLock 管理 137
 NUMLOCK キーワード 137
 NumLock テーブル 137
 O
 ODBC 321
 odbc_server 364

ODBC ドライバ 321
OLE 229, 321
 in-place モードでの起動 230
 エミュレータオブジェクトを埋め込む 229
 ベーストリンク 232
OLE オートメーション 240, 321
 Connect 関数 243
 Disconnect 関数 244
 Load 関数 245
 Save 関数 246
 Scrape 関数 247
 SendKey 関数 248
 SendString 関数 249
 ShowWindow 関数 250
 Vision オブジェクト 240
 エミュレータドキュメントでのプログラミング 234
 関数の概要 242
 タイプライブラリ 242
OLWM. *OPEN LOOK* ウィンドウマネージャ を参照
OPEN LOOK ウィンドウマネージャ
 のための X サーバの構成 167
 のためのホスト構成 168
 ファイルマネージャ 166
OPEN LOOK 166, 321
 ボタンが表示されない 466
OPEN LOOK ウィンドウマネージャ 168
 X サーバと 166
 起動 168
 終了 169
OpenWindows 166
 制限事項 166
OSF/Motif 322
override_redirect 属性 159
 ウィンドウ装飾 152

P

PASSWORD 関数
 ComScript 425
PATH
 XRemote ホストソフトウェア 194
PC
 ログ収集 80
PCF フォント 322
pclist 69, 365
pclist プログラム 308
pcprtclnt 286, 366
PC クライアントの認証 74
ping の間隔 185
Portable Compiled Format フォント 106

PostScript 287, 322
PPP 322
PRIMARY セレクションアトム 123
printers ファイル 384
printsrv 293, 367
prodinfo.dat ファイル 77, 78, 385

Q

QWERTY キーボード 322

R

REMATCH 関数
 ComScript 405, 426
Remote Sockets サービス 322
Resume 200
RETURN 関数
 ComScript 428
reuseaddr 322
rexec 217, 282
Rgb.txt ファイル 122
RGB 値 118
 カラー名データベース 122
RGB モデル 118
rlogin 217
RPC. リモートプロシジャコール を参照
RPC プロトコル 286, 292
RS232C 322
RS232C プロトコル 47
rsh 282

S

SAVEPASSWORD 関数
 ComScript 429
Save 関数
 OLE オートメーション 246
SCO-ODT
 XDMCP の使い方 461
 印刷サービスの使い方 462
 接続 461
scoarch 21
Scrape 関数
 OLE オートメーション 247
SendKey 関数
 OLE オートメーション 248
SendString 関数
 OLE オートメーション 249

SEND 関数
 ComScript 431
 servers ファイル 386
 SITE_FORWARD プロパティ 160
 SLIP 323
 Solaris 464
 SPX 323
 SQL-Retriever 323
 SQLGold 323
 sqlrsec.conf ファイル 388
 STATUS 関数
 ComScript 432
 SunOS 466
 Interrupted System Call エラー 467
 SuperVision 323

T

TAPI 49, 323
 TCP 323
 TCP/IP 323
 XDMCP サポート 181
 termdump 368
 termios ファイル 391
 termio ファイル 391
 tpproxy 201, 374
 tpwatch 201, 374
 TRACE 関数
 ComScript 434
 TrueType フォント 222
 X アプリケーションで使用 115
 tsrv 217, 282, 369
 tsrv.rc ファイル 219, 392

U

UNIX
 UNIX版Visionサービスの構成 22
 xdm の構成 184
 XRemote のインストール 193
 XRemote の構成 194
 クライアント認証 75
 上へのインストール 19
 上へのインストールのための要求条件 19
 ソフトウェアのアップグレード 19
 ソフトウェアの削除 22
 ヒントと秘訣 456
 メッセージ 305
 ログ収集 81

Unix アプリケーションウィザード 172
 UNIX ファイル
 termios 391
 Unixslsc.scr ファイル 461
 UNIX コマンド 342
 admcopy 343
 admexec 344
 admindex 345
 admls 346
 admupdate 348
 authsrv 350
 browsesrv 351
 dumppath 352
 licadmin 353
 licsrv 355
 lna 357
 logsrv 358
 newterm 359
 nsmadmin 360
 nsmrpsrv 361
 nsmsrv 362
 odbc_server 364
 pclist 365
 pcprtclnt 366
 printsrv 367
 termdump 368
 tpproxy 374
 tpwatch 374
 tsrv 369
 usputppxy 371
 visionver 372
 workbsrv 373
 xresume 374
 X アプリケーションを起動する 179
 システムの要求条件
 UNIX 19
 Unix ネットワークコンピュータ 324
 ブラウジング 256
 ホストの構成 41
 UNIX/パス内のディレクトリ
 ホストエクスプローラからアクセス 259
 UNIX 版 Vision サービス 18
 構成 22
 UNIX ファイル 378
 .visionpath 379
 cola.dat 380
 domain.dat 381
 licsrv.dat 383
 printers 384
 prodinfo.dat 385
 servers 386

sqlrsec.conf 388
 termio 391
 tsrv.rc 392
 vision.conf 393
 visionpriv 395
 vwdebuglog 396
 vweventlog 397
 winsize 398
 usputppxy 48, 65, 371
 USP サービス 63, 324
 ソケットサービス 65
 タスクサービス 217, 282
 ファイルアクセスサービス 262
 リモートプロシージャコール 64
 utmp ファイル 218
 UTP. ユニバーサルトランスポートプロトコル を参照

V

VGA ディスプレイ 120
 Vision Resume 200, 324
 PC上での構成 202
 UNIX上での構成 203
 Xサーバと 201
 クライアントの起動 206
 セッションの一時停止 207
 セッションの起動 204
 セッションの再開 204
 セッションの再配置 207
 セッションの終了 208, 209
 動作のしくみ 201
 評価 18
 ログファイル 204
 Vision Resume セッションを一時停止する 207
 Vision Resume セッションを再開する 204
 Vision Resume セッションを再配置する 207
 Vision ウィンドウマネージャ
 カラーマップ 121
 Vision ウィンドウマネージャ
 初期化 151
 vision.conf ファイル 67, 72, 393
 VISION_ADMINDIR 393
 VISION_BINDIR 393
 VISION_DEBUGDIR 393
 VISION_DIR 393
 VISION_ETCDIR 393
 VISION_EVENTDIR 393
 VISION_EVENTLEVEL 393
 VISION_LOGLIMIT 393
 VISION_MAXSESSIONS 393
 VISION_NAMESVC 393
 VISION_NSOGATEWAY 19, 393
 VISION_TCLDIR 393
 VISION_TCLINTERP 393
 VISION_TCLSCRIPTDIR 393
 VISION_TMPDIR 393
 visionpriv ファイル 395
 visionver 372
 Vision ウィンドウマネージャ 92, 151, 324
 Lower コマンド (コントロールメニュー) 156
 Motif との互換性 156
 X クライアントのクローズ 155
 アイコン 153
 ウィンドウ装飾 152
 ウィンドウの位置 153
 ウィンドウのサイズ 153
 ウィンドウの初期状態 153
 ウィンドウのタイトル 153
 ウィンドウのマッピング 152
 カラーマップ管理 154
 関数 161
 によるウィンドウプロパティ設定 160
 フォーカスポリシー 154
 Vision コミュニケーション
 XRemote と 191
 Vision コミュニケーションアーキテクチャ 24, 324
 ComScript 400
 USP サービス 63
 資源の検索 66
 スクリプト言語 402
 セキュリティ 73
 端末サービス 216
 トランスポートの構成 30
 トランスポートの追加 31
 ネームスペース環境 66
 パスワードの格納 75
 ファイルアクセスサービス 261
 ホストの構成 38
 モデル 26
 ユニバーサルトランスポートプロトコル 47
 ライセンス管理 76
 リモート実行サービス 280
 Vision コミュニケーションデータベース 61, 325
 Vision コミュニケーションのプロープ 30
 Vision サービス 63, 324
 起動 63
 シャットダウン 64
 VMS 325
 vsinstall 21

vsremove 22
 VT420 エミュレータ 325
 キャラクタセット 236
 仕様 235, 237
 端末設定 236
 非サポート機能 235
 VUE ファイルマネージャ 459
 vwdebuglog ファイル 81, 396
 vweventlog ファイル 81, 397
 VWM. *Vision* ウィンドウマネージャ を参照

W

WAN 325
 Windows
 X フォントを使用する 115
 仮想キーコード 147
 クリップボード 123
 フォントと X クライアント 114
 フォントの別名 114
 リソース 95
 [Windows がスクリーンよりも大きい] オプション 153
 Windows レジストリ 70
 winsize ファイル 398
 WM_COLORMAP_WINDOWS プロパティ 155, 160
 WM_DELETE_WINDOW プロトコル 155, 159
 WM_HINTS プロパティ 158
 WM_ICON_NAME プロパティ 157
 WM_ICON_SIZE プロパティ 160
 WM_NAME プロパティ 157
 WM_NORMAL_HINTS プロパティ 157
 WM_PROTOCOLS プロパティ 159
 WM_SAVE_YOURSELF プロトコル 155, 159
 WM_STATE プロパティ 160
 WM_TAKE_FOCUS プロトコル 159
 WM_TRANSIENT_FOR プロパティ 159
 ウィンドウ装飾 152
 workbsrv 373

X

X
 PC にフォントをコピー 109
 Windows アプリケーションで使用するフォント 115
 アプリケーションのクローズ 155
 カラー 118
 キーボード入力 99
 キーボードマップ 99
 コピー 123

セキュリティ 96
 によるウィンドウプロパティ設定 157
 フォント 104
 ベースト 123

X Display Management Control Protocol. *XDMCP* を参照

X Display Manager 181
 chooser 186, 187
 ping の間隔 185
 UNIX 上での構成 184
 xdm 181
 ユーザアクセス制御 186
 ウィンドウ管理 189
 起動 186
 自発的選択 186
 終了 189
 セッションの起動 188
 用に X サーバを構成 182
 ログイン 188
 ログファイル 186
 Xaccess ファイル 186
 xauth 98
 Xdef-icl.txt ファイル 165
 Xdefault.txt ファイル 165
 xdm 181, 326
 chooser 186, 187
 ping の間隔 185
 UNIX 上での構成 184
 XRemote のセッションファイルを使用 185
 起動 186
 自発的選択 186
 終了 189
 セッションの起動 188
 閉じる 88
 ユーザアクセス制御 186
 用に X サーバを構成 182
 ログイン 188
 ログファイル 186
 XDM-AUTHORIZATION-1 97, 183
 xdm-config 184
 xdm-errors ファイル 186
 xdm-keys ファイル 97
 XDMCP 96, 181, 326
 HP-UX での使い方 459
 SCO-ODT での使い方 461
 X11R3 クライアントと 97
 xdm を起動 186
 間接クエリー 186, 187
 サポートされるネットワーク 181
 セキュリティ 186
 直接クエリー 186
 使い方 96

- ブロードキャストクエリー 186
 - 有効にする 97, 182
- xdmでのログイン 188
- Xhosts.txt ファイル 96
- xinitremote 190
 - XRemote ホストソフトウェア 194
- XLFD 104
- Xlog.txt ファイル 128
- xmodmap 101
- xprop 156
- XRemote 190, 326
 - PC 上での構成 192
 - UNIX上のログファイル 195
 - UNIXでの構成 194
 - USP 接続上での実行 54
 - Vision コミュニケーションと 191
 - xdmのセッションファイルの使用 195
 - Xサーバと 191
 - コネクションモニタと 198
 - 終了 198
 - シリーズ 800 HP-UX 9.0 での使い方 467
 - セッションマネージャ 190
 - 端末サーバを経由 467
 - でのBSD UNIXの使い方 468
 - バージョン番号 196
 - プロトコル 47
 - ホストソフトウェア 193
 - ホストソフトウェアの検索パス 194
 - ホストモジュール 18
- Xremote
 - ヘルパープロセス 197
- XREMOTETMPDIR 195
- Xremote プロキシサーバ 190
- xresume 201, 206, 374
 - 検索パス 203
- Xsession 188
- xterm 326
- Xtrace プロトコル 33
- xvinstall 21
- X Vision 326
 - 評価 18
- X Vision フォントコンパイラ 326
- xvremove 22
- Xアプリケーション
 - 起動 172, 188
 - 端末エミュレータによる起動 178
 - リモートプログラムスタータによる起動 176
 - を起動するUNIX コマンド 179
- X ウィンドウシステム
 - Xにおけるフォント 104
 - カラー 118
 - キーボード入力 99
 - 標準Xフォント 106
- X キーマップ 134
 - コンパイル 144
 - 作成 145
 - 修正 138
- X キーマップジェネレータ 101, 325
 - Windows 仮想キーコードの表示 147
 - 新しいキーマップの作成 145
 - キーマップのコンパイル 144
 - 検証 145
 - テキスト形式キーマップの作成 145
- X キーマップの修正例
 - ALTのMod2へのマッピング 139
 - ファンクションキーのマッピング 142
- Xクライアント 325
- Xクライアントのクローズ
 - WM_SAVE_YOURSELF プロトコル 155
 - 155
 - WM_DELETE_WINDOW プロトコル 155
- Xサーバ 84, 325
 - Motifと 163
 - OLWMのための構成 167
 - OPEN LOOKと 166
 - Vision Resumeと 201
 - X Display Manager用に構成 182
 - XDMCP 181
 - XRemoteと 191
 - XRemote プロトコルを有効に 196
 - カラー 119
 - キーボード入力 99
 - 起動 86
 - グラフィックキャッシュ 95
 - グラフィックのコピー 124
 - 構成 88
 - コピー 123
 - 再起動 89
 - 手動クローズ 87
 - 診断 128
 - 制限 86
 - セキュリティ 96
 - ディスプレイモード 90
 - トランスポート 92
 - に関する情報の入手 129
 - パフォーマンスの最適化 93
 - フォント 106
 - フォントバス順 108
 - プロファイル 89
 - ペースト 123
 - ログファイル 128

X サービス 18
 X ディスプレイ 325
 X ディスプレイデバイス 179
 X におけるスケーラブルフォント 105
 X 論理フォント 326
 X 論理フォント記述 104
 Windows フォントの 114

Z

Zone
 Zone コントロールの使い方 315
 Zone 属性の表示 316
 デスクトップの並べ替え 312
 並べ替え 312
 プログラムの自動起動 316
 Zone コントロール 315
 Zone プログラム
 終了 314
 Zone ポリス 313

ア

アイコン
 Vision ウィンドウマネージャ 153
 アクション 269
 アクセス制御
 ホスト 96
 ユーザ 96, 186
 ローカル認証ファイル 98
 アップグレード
 UNIX ソフトウェア 19
 アドミンアップデート 326
 アドミンデータベース 327
 項目 327
 アドミンワークベンチ 327
 アドミンワークベンチサーバ 327
 アトム 123, 327
 CLIPBOARD 123
 PRIMARY 123
 アプリケーション 327
 X のクローズ 155
 暗号化
 XDM-AUTHORIZATION-1 セキュリティのための 97

イ

以前の出力 221
 位置
 端末エミュレータ 221

移動 271
 改行文字の変換 272
 イベント 225
 印刷不能文字 226
 ログ収集 81
 印刷 286
 PC から UNIX へ 292
 UNIX から PC へ 286
 エミュレータ画面 228
 エミュレータからの 227
 エミュレータスレーブ印刷 228
 構成 292
 リモート 292
 ローカル 286
 印刷サービス
 AIX での 457
 SCO-ODT での 462
 印刷スタイル 227
 印刷サービス
 HP-UX での 460
 印字不能な文字
 OLE オートメーションの SendString 関数 249
 キーマップ 253
 インストール
 UNIX ソフトウェア 19
 XRemote ホストソフトウェア 193
 インストールスクリプト
 オプションスイッチ 21
 インターネット 327
 インフォメーションサーバ 327

ウ

ウィンドウ
 Motif のマッピング 152
 override_redirect 属性 152
 Vision ウィンドウマネージャによるプロパティ設定 160
 WM_TRANSIENT_FOR プロパティ 152
 X クライアントによるプロパティ設定 157
 位置 153
 サイズ 153
 初期状態 153
 装飾 152
 タイトル 153
 プロパティ 156
 ウィンドウ管理 92, 150
 OPEN LOOK 166
 Vision ウィンドウマネージャ 150
 X Display Manager 189
 ウィンドウサイズの最小値 153

ウィンドウサイズの最大値 153

ウィンドウズレジストリ 339

ウィンドウの管理

Motif 163

ウィンドウの初期状態

Vision ウィンドウマネージャ 153

ウィンドウのマッピング

Vision ウィンドウマネージャ 152

ウィンドウマネージャ 328

エ

エミュレータオブジェクトをドキュメントに埋め込む 229

エミュレータでのキーマップの変更 254

エミュレータドキュメント

Unix アプリケーションウィザードと 172

Windows95またはNT4.0で作成 219

Windows デスクトップから構成 220

オブジェクトとして埋め込む 229

サマリ情報 221

プログラミング 234

プログラムマネージャグループから構成 220

別のアプリケーションにテキストをコピー 232

ホットリンク 232

リンクと埋め込み 229

エミュレータドキュメントでのプログラミング 234, 240

オブジェクトタイプ 240

遠隔手続き呼び出しサービス 328

オ

オーディオサーバ 125

起動 126

構成 126

オートメーションオブジェクト 240

オープン

端末エミュレータからのモデム接続の 58

オブジェクトのリンクと埋め込み. OLE を参照

オリエンタルフォント 107

カ

改行文字 272

拡張キー 100

画像 328

仮想キーコード 100

表示 147

カタログ

フォントサーバ 113, 335

カットバッファ 123, 328

画面出力の履歴 221

カラー 118, 223

RGB モデル 118

Vision ウィンドウマネージャのポリシー 121

X サーバにおける 119

X における 118

カラーマップの管理 121

ツールカラーディスプレイ 121

ビジュアル 118

変更可能なカラーディスプレイ 120

変更不可のカラーディスプレイ 119

モノクロディスプレイ 119

カラーセル 118

カラーディスプレイ

ツールカラー 121

変更可能 120

変更不可 119

カラーマップ 118, 328

スワッピング 121

デフォルト 119

ハードウェア 119

ビジュアル 119

カラーマップ管理

Vision ウィンドウマネージャ 154

カラーマップの管理 121

カラー名データベース 122

漢字フォント 107

関数

ComScript 407

Vision ウィンドウマネージャ 161

間接クエリー

XDMCP 186

ディスプレイマネージャの選択 187

管理サーバ 327

管理された PC の登録

ネームスペースマネージャ 69

管理ホスト 329

管理ワークベンチサーバ 373

関連付け 270

関連マニユアル 12

キ

キー

XDM-AUTHORIZATION-1 セキュリティのための
98, 183

キーコード 99, 329

キーコードマップ 329

キーパッドのキー

AIX 456

キーボード入力 99
 Num Lock 管理 103
 X サーバにおける 99
 X における 99
 修飾キー 102
 ショートカットキー 101
 キーボードマップ 223
 .txt ファイル 100
 VT420 472, 474, 476, 478
 X 99
 xmodmap による変更 101
 Xサーバとともに提供される 99
 キーマップ 134, 329
 NumLock テーブル 137
 X キーマップのコンパイル 144
 新しい X キーマップの作成 145
 エミュレータでの変更 254
 エミュレータにおける 224
 キーマップテーブル 134
 作成 252
 修飾子テーブル 138
 修正 138
 修正例 139, 142
 テキスト形式の作成 145
 テンプレート 145
 キーマップエディタ 224, 252, 329
 キーマップジェネレータ 101
 キーマップテーブル 134
 キーマップの編集 252
 起動
 Motif ウィンドウマネージャ 164
 OPEN LOOK ウィンドウマネージャ 168
 UNIX コマンドによる X アプリケーションの 179
 UNIX コマンドによるクライアントの 179
 Vision Resume セッション 204
 Vision Resume セッションでのクライアントの 206
 xdm 186
 xdm による X アプリケーションの 188
 X アプリケーション 172
 X サーバ 86
 オーディオサーバ 126
 サーバプログラム 63, 71
 端末エミュレータによる X アプリケーションの 178
 端末エミュレータによるクライアントの 178
 プログラムを自動的に 316
 リモートプログラムスタータで X アプリケーションを
 176, 181, 184, 186, 188, 189
 ローカル Motif ウィンドウマネージャ 164
 ローカルネームエージェント 69
 キャラクタセット
 VT420 エミュレータ 236

記録 227

ク

クライアント
 閉じる 155
 クライアント認証
 セキュリティ 74
 クライアントの認証
 ライセンス管理 78
 グラフィック
 コピー 124
 グラフィックキャッシュ 95
 グラフィックスピードオブティマイザ 93
 [プロパティ]ダイアログから実行 93
 ベンチマークディスプレイ 94
 グラフィックのクリップボード形式 124
 クリップボード 329
 Windows 123
 グラフィックの形式 124
 クローズ
 xdm 88
 X サーバ 87
 グローバルエイリアス 116

ケ

検索
 資源 66
 検索する資源 66
 検索パス
 XRemote ホストソフトウェア 194

コ

構成
 LPD を使用するための BSD UNIX プリンタの 288
 LPD を使用するための UNIX System V 上のプリンタの
 290
 Modem トランスポート 53
 OLWM のための X サーバの 167
 OPEN LOOK のためのホスト 168
 PC 上での Vision Resume の 202
 PC 上での XRemote の 192
 RPC を使用するための BSD UNIX 上のプリンタの 289
 RPC を使用するための UNIX System V 上のプリンタの
 290
 Serial-Com1 トランスポート 51
 UNIX 上での Vision Resume の 203
 UNIX 上での XRemote の 194
 UNIX 版 Vision サービス 22

Vision サービスでリモートプリンタを 297
 Windows デスクトップからエミュレータドキュメントを
 220
 X Display Manager 184
 X Display Manager 用にサーバを 182
 X サーバ 88
 印刷 292
 オーディオサーバ 126
 サーバプログラム 70
 シリアル接続 48
 シリアル接続に対するホストの 56
 シリアルトランスポート 50
 データベースサーバ 62
 トランスポート 30
 プリントサーバ 292
 プログラムマネージャグループから
 エミュレータドキュメントを 220
 ホスト 38
 モデム 49
 モデム接続 48
 モデム接続に対するホストの 56
 モデムトランスポート 50

構成要素
 ライセンス管理 77

互換性
 Vision ウィンドウマネージャの Motif との 156

コネクションマネージャ 329

コネクションモニタ 59
 ComScript ファイルの生成 438
 ComScript ファイルのテスト 437
 XRemote と 198
 シリアル接続のオープン 60

コネクションモニター 329

コピー 123, 225, 271
 Windows と X との間での 124
 X サーバにおける 123
 X における 123
 X フォントを PC に 109
 改行文字の変換 272
 グラフィック 124
 ファイル名 271
 プログラムマネージャに 275
 別のアプリケーションにテキストを 232

コマンド
 hostmgr 273
 UNIX 342

コマンドライン 218

コメントテキスト
 ComScript 406

コントロールパネル
 ホストの構成 40

[コントロールメニュー] アイコン 329
 コンパイル
 X キーマップ 144

サ

サーバ
 XDMCP 181
 XRemote プロトコルを有効に 196
 X ウィンドウシステム 84
 起動 63, 71
 構成 70

サービス 282

再起動
 X サーバ 89

サイズ
 端末エミュレータ 221

最適化
 サーバのパフォーマンス 93

削除
 UNIX ソフトウェア 22

作成
 Windows NT 3.51でエミュレータドキュメント 223, 229
 Windows95またはNT4.0でのエミュレータドキュメント
 219
 X キーマップ 145
 キーマップ 252
 テキスト形式の X キーマップ 145

サマリ情報
 エミュレータドキュメント 221

シ

システム管理者 329

システム設定 29

システム定義関数
 ComScript 407

自動フォント置換 116

修飾キー 99, 102, 329
 修飾子テーブル 138

修飾子テーブル 138

修正
 X キーマップ 138

終了
 Motif 165
 OPEN LOOK ウィンドウマネージャ 169
 Vision Resumeセッション 208, 209
 xdm 189
 XRemote 198

出力

- 以前の 221
- 仕様
 - VT420 エミュレータ 235, 237
- ショートカット 330
- ショートカットキー 101
- 照合文字列
 - ComScript 405
- 詳細オプション
 - トランスポート 36
- 情報
 - X サーバに関する 129
- 情報サーバ 310
- 初期化
 - Vision ウィンドウマネージャ 151
- シリアル接続 46
 - Serial-Com1 トランスポートの構成 51
 - USP のシリアルトランスポートの作成 54
 - 構成 48
 - コネクションモニタを使用してオープンする 60
 - 電話番号 51
 - トランスポートの構成 50
 - プロトコル 47
 - ホストの構成 56
- シリアル接続のオープン
 - コネクションモニタの使用 60
- シングルウィンドウモード 84, 91
- シンプルスタイル 227

ス

- スクリプトオプション
 - トランスポート 35
 - ホスト 44
- スクリプト言語 402
 - IF...THEN...ELSE ステートメント 403
 - LOOP ステートメント 403
 - コメント 406
 - 照合文字列 405
 - パラメータ 405
 - ユーザ定義の関数 404
- スクロールバー 330
- [スタート]メニュー
 - でエミュレータドキュメントを作成 219
- スタイル 226
 - 規定の 226
- ステータスバー 330
- スレーブ印刷 228, 330

セ

- 制限
 - X サーバ 86
- 制限事項
 - OpenWindows 166
- セキュリティ 73, 96
 - XDMCP 96
 - クライアント認証 74
 - 認証サーバ 74
 - 認証のタイプ 73
 - ホストアクセス制御 96
 - ユーザアクセス制御 96
 - ローカル認証ファイル 98
- セキュリティオプション
 - トランスポート 34
 - ホスト 44
- セッション
 - xdm の起動 188
- セッションマネージャ
 - XRemote 190
- セレクションアトム 123
 - CLIPBOARD 123
 - PRIMARY 123
- セレクションメカニズム 123
- 前提条件
 - UNIXソフトウェアのインストール 19

ソ

- ゾーンバー 312
- 走査コード 100
- ソケットサービス 65

タ

- 代替フォント
 - 選択 116
- タイトルバー 330
- ダイナミックリンクライブラリ 330
- ダイヤルバック
 - ComScript ファイルの例 439
- 対話形式のフォント置換 116
- 多重プロトコル 330
- タスクサーバ 369
- タスクサービス 217, 331
 - utmp ファイル 218
 - 端末コマンドライン 218
- タスクバー 331

端末 331
 端末エミュレーション 212
 OLE オートメーションの関数 242
 タイプライブラリ 242
 端末エミュレータ 331
 OLE 229
 OLE オブジェクトタイプ 240
 UNIX メッセージ 305
 VT420 キャラクタセット 236
 VT420 仕様 235, 237
 VT420 端末設定 236
 VT420 非サポート機能 235
 イベント 225
 印刷 227
 エミュレータオブジェクトの埋め込み 229
 エミュレータドキュメントでのプログラミング 234
 仮想画面 221
 画面の印刷 228
 カラー 223
 キーボード入力 223
 記録と再生 227
 コピーアンドペースト 225
 サイズおよび位置の変更 221
 スタイル 226
 スレーブ印刷 228
 ツールバー 223
 ドキュメント 219
 による X アプリケーションの起動 178
 フォント 222
 ホットリンクを作成 232
 モデム接続のオープン 58
 履歴ビューの表示 221
 端末コマンドライン 218
 端末サーバ
 を介した XRemote の使い方 467
 端末サービス 216, 331
 USP 217
 端末設定
 VT420 エミュレータ 236
 端末接続サービス
 方法の選択 216

チ

置換
 フォント 115
 中央処理装置 331
 直接クエリー
 XDMCP 186

ツ

ツールコマンド言語 331
 ツールバー 223, 331
 印刷不能文字 223
 追加
 X フォント 109
 ドキュメントファイルタイプ 267
 トランスポート 31
 パターン 268
 フォントパスへのフォルダの 110
 プログラムファイルタイプ 265
 リモートプリンタ 293
 ローカル印刷を Vision サービスで 287
 通常スタイル 227
 通信オプション
 トランスポート 32

テ

データ発信元 332
 データベース
 Vision コミュニケーション 61
 データベースウィンドウ 332
 データベースサーバ 332
 構成 62
 ディスプレイマネージャ
 ping の間隔 185
 起動 186
 用に X サーバを構成 182
 ログファイル 186
 ディスプレイモード
 選択 90
 テキストメッセージ 302
 デスクトップ 332
 からエミュレータドキュメントを構成 220
 ショートカット 274
 でエミュレータドキュメントを作成 219
 デスクトップの並べ替え 312, 315
 デバッグ
 メッセージのログ収集 81
 有効にする 80
 デフォルト 332
 デフォルトカラーマップ 119
 デフォルトの.tpsessionファイル 203
 電子メール 332
 テンプレート
 キーマップ 145
 電話番号 51

ト

- ツールカラーディスプレイ 121
- ドキュメント 332
 - タイプを追加 267
 - 端末エミュレータ 219
 - リモートプログラムスタータ 283
- 匿名 FTP 332
- ドッキングバー 332
- ドメインエームシステム 333
- ドラッグ 333
- トランスポート 333
 - Modem の構成 53
 - Serial-Com1 の構成 51
 - X サーバが使用する 92
 - 構成 30
 - 詳細オプションの指定 36
 - シリアル USP の作成 54
 - スクリプトオプションの指定 35
 - セキュリティオプションの指定 34
 - 調整 32
 - 追加 31
 - 通信オプションの指定 32
 - モデム USP の作成 54

二

- 認証
 - PC クライアント 74
 - UNIX クライアント 75
- 認証コード 98
 - XDMCP セキュリティ 96
- 認証サーバ 333, 350
 - セキュリティ 74
- 認証のタイプ
 - セキュリティ 73
- 認証ファイル
 - ローカル 98

ネ

- ネームスペース環境 66, 333
- ネームスペースマネージャ 67, 333
 - インストレーション 22
 - 管理された PC の登録 69
 - 他のネットワーク上にある資源の検索 72
- ネットワーク
 - XDMCP サポート 181
- ネットワーク接続 45
- ネットワークチェックアウト 333

ハ

- バージョン番号
 - XRemote 196
- ハードウェアカラーマップ 119
 - カラーマップの管理 121
- パス
 - ホストエクスプローラ 259
- パスワードの格納 34, 75
- パターン
 - 追加 268
- パターンマッチング 268, 334
- バックングストア 334
- パフォーマンスの最適化 93
- パラメータ
 - ComScript 405
- パレットマネージャ 120
- パン 334

ヒ

- 非サポート機能
 - VT420 エミュレータ 235
- ビジュアル 118
 - クラス 118
- ビジュアルクラス
 - カラー 118
 - 擬似カラー 118
 - グレー 118
 - グレースケール 118
 - ダイレクトカラー 118
 - ツールカラー 118
- ビデオプレーン 118
- 評価
 - Vision Resume 18
 - X Vision 18
- 表記法 12
- 表示
 - エミュレータ履歴ビュー 221
 - 仮想キーコード 147
 - 現在のユーザを 308
 - メッセージパッド 301
 - ユーザプロパティ 309
- 表示モード 334
- 標準ツールバー 223
- ヒント
 - Motif 154

フ

- ファイル
 - ComScript 400
 - ComScript ファイルの開発 436
 - ComScript ファイルの生成 438
 - ComScript ファイルのテスト 437
 - UNIX 378
 - ブラウジングファイル
 - ブラウジング 256
- ファイルアクセス
 - FTP 262
 - USP 262
- ファイルアクセスサーバ 351
- ファイルアクセスサービス 334
 - 方式の選び方 261
- ファイルアクセスサービスファイルアクセスサービス 261, 262
- ファイルタイプ 264
 - アクション 269
 - 関連付け 270
 - 属性 269
 - パターンマッチング 268
 - リモート 265
 - ローカル 264
- ファイルタイプの属性 269
- ファイル転送プロトコル 334
- ファイルマネージャ
 - OPEN LOOK 166
- ファイル名
 - コピー 271
- ファンクションキー
 - マッピング 142
- ファンクションキーツールバー 223
- フォーカスポリシー
 - Vision ウィンドウマネージャ 154
- フォント 104, 222, 335
 - AIX 456
 - HP-UX 459
 - TrueType と X アプリケーション 115
 - Windows と X クライアント 114
 - X ウィンドウシステム 106
 - X サーバにおける 106
 - X における 104
 - X フォントと Windows アプリケーション 115
 - X フォントを PC にコピー 109
 - サーバ 113
 - サポートされる形式 106
 - 置換 115
 - フォルダ 110
 - 別名 105, 110, 114
 - マトリックス XLFD 拡張機能 105
- フォントサーバ 335
- フォント置換 115
 - 自動 116
 - 使用可能なフォントの検索 117
 - 代替フォントの選択 116
 - 対話形式 116
 - 有効にする 115
- フォントパス 107, 335
 - グローバルエイリアス 116
 - 順序の変更 108
 - へのフォルダの追加 110
- 復号化 97
- ブラウジング
 - Windows 2000/Me/98/95またはNT 4.0上での 256
 - Windows NT 3.51における 259
- ブラウズブラウジング 256
- プリンタ
 - LPD を使用するための BSD UNIX 上での構成 288
 - LPD を使用するための UNIX System V 上の構成 290
 - RPC を使用するための BSD UNIX 上の構成 289
 - RPC を使用するための UNIX System V 上の構成 290
 - Vision サービスでのローカルの追加 287
 - Vision サービスでのリモートの構成 297
 - リモートの追加 293
- プリンタファイル 286
- プリントサーバ 335, 367
 - ウィンドウの表示 292
 - 構成 292
- プレーン
 - ビデオ 118
- ブロードキャスト 334, 335
- ブロードキャストクエリー
 - XDMCP 186
 - ディスプレイマネージャの選択 186
- プロキシサーバ
 - tpproxy 201
 - tpwatch 201
 - Vision Resume 201
 - Xremote 190
- プロキシプロセス 47, 48, 335, 371
- プロキシ用のホスト 33
- プログラム
 - タイプを追加 265
- プログラムマネージャグループ
 - からエミュレートドキュメントを構成 220
- プロトコル 335
- プロパティ 335
- プロファイル
 - X サーバ 84, 89

へ

ペースト 123, 225

- X サーバにおける 123
- X における 123
- ペーストリンク 232
- 別名
 - Windows フォントの 114
 - X フォントの 105
 - 自動 116
 - 対話形式 116
- 変換 DLL 336
- 変更
 - エミュレータウィンドウのサイズおよび位置 221
- 変更可能なカラーディスプレイ 120
- 変更不可のカラーディスプレイ 119
- ベンチマークディスプレイ
 - グラフィックススピードオブティマイザ 94

ホ

- [ポート設定] ダイアログボックス 52
- 方式
 - ファイルアクセス 261
- 方法
 - 端末接続 216
 - リモート実行 281
- 他のネットワーク上にある資源の検索
 - ネームスペースマネージャ 72
- ホスト 336
 - OPEN LOOK のための構成 168
 - 構成 38
 - シリアル接続に対する構成 56
 - スクリプトオプションの指定 44
 - セキュリティオプションの指定 44
 - の参照 256
 - ホストオプションの指定 43
 - モデム接続に対する構成 56
- ホストアクセス制御 96
 - 有効にする 96
- ホストエクスプローラ 336
 - 参照 256
 - パス 259
- ホストオプション
 - ホスト 43
- ホストファインダ 336
 - ホストの構成 38
- ホストマネージャ 336
 - hostmgr コマンド 273
 - 参照 259
 - パッチファイル 272
 - プログラムマネージャにコピー 275
 - ホストの構成 42

- ホストマネージャのパッチファイル 272
- ボタンエミュレーション 103
- ホットリンク 232

マ

- マウス
 - の中央ボタンのエミュレーション 103
- マウスの中央ボタン
 - との組み合わせ 104
 - ドラッグ 103
 - のエミュレーション 103
- マクロ 336
- マクロ言語 336
- マジッククッキー 97
- マッピング
 - ALT の Mod2 への 139
 - ファンクションキー 142
- マトリックス XLFD 拡張機能 105
 - 文字のサブセット化 105
- マニュアル 10
 - オンラインヘルプ 11
 - 関連情報 12
 - 表記法 12
- マニュアルについて 11
- マルチウィンドウモード 84, 90
- マルチメディアメッセージ 303

メ

- メインプログラム
 - クローズ 88
- メッセージ処理 300
 - UNIX メッセージ 305
 - テキストメッセージの送信 302
 - マルチメディアメッセージの送信 303
 - メッセージパッド 300
- メッセージパッド 337
 - 表示 301
 - メッセージの処理 300
- メニュー
 - Motif 154
- メニューバー 337

モ

- モードスイッチ 99, 102
- 文字 337
- 文字端末アプリケーション 337
- モデム 337

構成 49
 モデム接続 46
 Modem トランスポートの構成 53
 TAPI デバイスとしてのモデムのコウセイ 49
 USP のモデムトランスポートの作成 54
 構成 48
 端末エミュレータからのオープン 58
 電話番号 51
 トランスポートの構成 50
 ホストの構成 56
 モノクロディスプレイ 119

ユ

ユーザアクセス制御 96
 MIT-MAGIC-COOKIE-1 97
 XDM-AUTHORIZATION-1 97
 ローカル認証ファイル 98
 ユーザデータグラムプロトコル 337
 ユーザ定義の関数
 ComScript 404
 ユーザのドメインへの組織化
 ライセンス管理 77
 ユーザビューワ 337
 テキストメッセージの送信 302
 マルチメディアメッセージの送信 303
 ユーザビューワプログラム
 現在のユーザを表示 308
 ユーザプロパティの表示 309
 ユニバーサルセッションプロトコル 47, 337
 サービス 63
 ユニフォームネーミングコンベンション 67
 ユニバーサルトランスポートプロトコル 47, 338
 ユニフォームネーミングコンベンション 338

ヨ

用語集 319-339
 要求条件
 UNIXソフトウェア実行のための 19

ラ

ライセンス管理 76
 クライアントの認証 78
 製品と構成要素 77
 ユーザのドメインへの組織化 77
 ライセンスイベント 79
 ライセンスサーバ 78
 ライセンス番号 77
 ライセンスサーバ 78, 338, 355

ライセンス番号 77

リ

リソース
 グラフィックキャッシュ 95
 リモート印刷 292, 338
 リモートコマンドプロセッサ 338
 リモート実行
 rexec 282
 rsh 282
 USP 282
 方法を選択する 281
 リモート実行サービス 280, 338
 方法を選択する 281
 リモートファイルタイプ 265
 リモートプリンタ
 LPD を使用するための BSD UNIX 上の構成 288
 LPD を使用するための UNIX System V 上の構成 290
 RPC を使用するための BSD UNIX 上の構成 289
 RPC を使用するための UNIX System V 上の構成 290
 Vision サービスでの構成 297
 追加 293
 リモートプログラム
 実行 278
 リモートプログラムスタータ 278, 338
 Unix アプリケーションウィザードと 172
 ドキュメント 283
 による X アプリケーションの起動リモートプログラム 176
 リモートプログラムを実行 278
 リモートプロシージャコール 64
 リモートプロシージャコールサービス 64
 リンガー 339

ル

ルートウィンドウ 339

ロ

ローカル Motif ウィンドウマネージャ 163
 起動 164
 ローカル印刷 286, 339
 ローカル認証ファイル 98
 有効にする 98
 ローカルネームエージェント 69, 339, 357
 UNIX上で自動的に起動 22
 起動 69
 クライアントの認証 78
 サーバの構成 70
 ローカルファイルタイプ 264

- ローカルプリンタ
 - Vision サービスでの追加 287
- ログイン
 - 手動 178
- ログサーバ 339, 358
- ログ収集 80
 - PC での 80
 - UNIX での 81
 - イベントの 81
 - デバッグ情報 81
- ログファイル
 - UNIX上の XRemote 195
 - Vision Resume 204
 - xdm 186
 - X サーバ 128
- 論理フォント記述 104
 - Windows フォント 114

ワ

- ワークベンチサーバ 339
- ワイルドカード
 - X 論理フォント記述 105

