

Tru64 UNIX

インストレーション・ガイド — 上級ユーザ編

Part Number: AA-RK3BE-TE

2002 年 11 月

ソフトウェア・バージョン: Tru64 UNIX Version 5.1B

HP Tru64 UNIX オペレーティング・システムのインストレーションに関する上級ユーザ向けのリファレンス情報を提供します。

日本ヒューレット・パッカード株式会社

© 2002 日本ヒューレット・パッカート株式会社

本書の著作権は日本ヒューレット・パッカート株式会社が保有しており、本書中の解説および図、表は日本ヒューレット・パッカートの文書による許可なしに、その全体または一部を、いかなる場合にも再版あるいは複製することを禁じます。

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

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

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

Microsoft[®]、Windows[®]、および Windows NT[®] は、米国ならびに他の国における Microsoft Corporation の商標です。Motif[®]、OSF/1[®]、UNIX[®]、および X/Open[®] は、米国ならびに他の国における The Open Group の商標です。

原典: Installation Guide — Advanced Topics (AA-RH8TE-TE)
© 2002 Hewlett-Packard Company

目次

まえがき

1 RIS サーバからのインストール

1.1	RIS サーバからのインストールの前提条件	1-1
1.2	システムがネットワークに接続されているか	1-2
1.3	RIS サーバからのアップデート・インストールの開始 ..	1-3
1.4	RIS サーバからのフル・インストールの開始	1-5
1.4.1	ネットワーク・ブート・オプション: FDDI (Fiber Distributed Data Interface)	1-8
1.4.2	ネットワーク・ブートの失敗のトラブルシューティング ..	1-9
1.4.3	ISA バスにグラフィックス・デバイスがあるシステムのネットワーク・リブートに関する考慮事項	1-9

2 ワールドワイド言語サポートのインストール手順

2.1	ワールドワイド言語サポートの概要	2-1
2.2	[ステップ 1]: WLS インストールの準備	2-2
2.3	[オプション]: WLS ソフトウェア用に利用可能なディスク・スペースの拡大	2-3
2.3.1	UFS ファイル・システムのディスク・スペースを大きくする	2-3
2.3.2	AdvFS ファイル・システムのディスク・スペースを大きくする	2-4
2.4	[ステップ 2]: WLS インストールの開始	2-4
2.4.1	CD-ROM からのインストール	2-5
2.4.2	RIS サーバからのインストール	2-5
2.5	[ステップ 3]: WLS インストール処理中のユーザ応答 ..	2-6

2.5.1	ワールドワイド・ソフトウェア・サブセットの選択	2-9
2.5.2	ワールドワイド・ソフトウェア・サブセットのロード	2-11
2.5.3	ワールドワイド・ソフトウェア・サブセットの構成	2-12
2.5.4	カーネルの構築	2-13
2.5.5	国際化機能構成アプリケーションの使用	2-14
2.6	[ステップ 4]: インストール後のアジア言語用カーネルの構築 (オプション)	2-15
2.6.1	アジア言語用のターミナル・ドライバおよびデーモンをサ ポートするためのカーネルの再構成	2-15
2.7	WLS インストレーションで作成されるバックアップ・ファイ ル	2-18
2.8	[ステップ 5]: 環境変数の設定	2-19
2.9	関連ドキュメント	2-20

3 UNIX シェル

3.1	概要	3-1
3.2	UNIX シェルの起動	3-2
3.3	UNIX シェルの機能	3-3
3.4	スワップ領域の作成	3-4
3.5	ファイル・システムのマウント	3-5
3.6	バックアップからのファイル・システムのリストア	3-6
3.6.1	テープ・バックアップからの UNIX ファイル・システム (UFS) のリストア	3-7
3.6.2	テープ・バックアップからの AdvFS のリストア	3-9
3.7	破損したディスクから新しいディスクへのルート・ファイル・ システム・イメージのリストア	3-13
3.8	disklabel コマンドによるディスク・パーティション・サイズ の変更	3-15
3.9	UNIX シェルからインストレーション・プロシージャに戻る .	3-18
3.9.1	テキスト・ベースのインタフェース	3-18

3.9.2	グラフィカル・インタフェース	3-18
4	ディスク・スペースの計画	
4.1	概要	4-2
4.2	クラスタ環境におけるディスク計画上の考慮事項	4-2
4.3	ファイル・システム・タイプの概要	4-2
4.3.1	AdvFS (Advanced File System)	4-3
4.3.2	UNIX ファイル・システム (UFS)	4-4
4.4	フル・インストールの自動ディスク・スペース計画機能	4-5
4.5	手作業でディスク・スペースを計画しなければならない場合	4-7
4.5.1	リモート・インストール・サービス (RIS) に関する考慮事項	4-8
4.5.2	データレス管理サービス (DMS) に関する考慮事項	4-9
4.6	ディスク・パーティションおよびファイル・システム・レイアウトのカスタマイズに関する考慮事項	4-10
4.7	既存のディスクおよびパーティションのサイズの確認	4-10
4.8	ソフトウェアが必要とするディスク容量	4-12
4.8.1	ファイル・システムのオーバヘッド	4-13
4.9	ファイル・システムの内容	4-13
4.10	/usr ファイル・システムの内容	4-14
4.10.1	オプションのソフトウェア・サブセットおよび関連プロダクトのためのスペース	4-15
4.10.2	ユーザ・アカウントおよびユーザ・ファイルのためのスペース	4-16
4.11	var ファイル・システムの内容	4-17
4.11.1	var ファイル・システム内のクラッシュ・ダンプ領域	4-17
4.11.2	エラー・ロガーおよび syslog ファイルのスペース	4-19
4.11.3	システム課金ファイルのスペース	4-20
4.12	スワップ領域の概要	4-20

5 インストレーション処理のカスタマイズ

5.1	概要	5-1
5.2	構成記述ファイルの概要	5-2
5.3	ユーザ提供ファイルの概要	5-3
5.4	CDF とユーザ提供ファイルの関係	5-5
5.5	管理者の作業の概要	5-5
5.6	動作原理	5-7
5.7	ユーザ提供ファイルの作成	5-13
5.7.1	preinstall ファイルの作成	5-14
5.7.2	postload ファイルの作成	5-16
5.7.3	postreboot ファイルの作成	5-18
5.8	ユーザ提供ファイルと CDF を正しい位置にコピーする	5-19
5.8.1	ファイルをディスクにコピーする	5-20
5.8.2	RIS サーバのプロファイル・セット・ディレクトリにファ イルをコピーする	5-21
5.8.3	/var/tmp ディレクトリにファイルをコピーする	5-22
5.8.4	CD-ROM にファイルをコピーする	5-23

6 インストレーションのクローニング

6.1	概要	6-1
6.2	実行方法	6-3
6.3	インストレーション CDF の概要	6-4
6.3.1	CDF のフォーマットと内容	6-4
6.3.2	インストレーション CDF の例	6-5
6.4	インストレーション・クローニング手順の要約	6-8
6.5	[ステップ 1]: 適切な CDF の作成と選択	6-9
6.5.1	許容されるディスクの相違点	6-10
6.5.2	グラフィックス・アダプタの違い	6-11

6.5.3	フォント・サイズの違い	6-13
6.5.4	キーボード・タイプの違い	6-14
6.6	[ステップ 2]: CDF の変更	6-15
6.6.1	ユーザの操作を不要にするための CDF 確認属性の設定 ..	6-16
6.6.2	ユーザの操作を不要にするためのカーネル・オプション属性の設定	6-17
6.6.3	ホストおよびサイトに固有の属性の設定	6-19
6.6.4	よくある誤り: 行末の空白	6-21
6.7	[ステップ 3]: その他のユーザ提供ファイルの作成と配置 (オプション)	6-22
6.8	[ステップ 4]: CDF を正しい位置にコピーする	6-22
6.9	[ステップ 5]: ターゲット・システム上でのフル・インストールの開始	6-23

7 構成のクローニング

7.1	概要	7-1
7.2	実行方法	7-2
7.3	構成 CDF のフォーマットと内容	7-4
7.3.1	構成 CDF の例	7-5
7.4	構成のクローニングに関する制限事項	7-6
7.5	構成のクローニング手順の要約	7-8
7.6	[ステップ 1]: 構成情報を構成 CDF に保存する	7-9
7.7	[ステップ 2]: 構成 CDF 内のホストおよびサイトに固有の属性を変更する	7-10
7.7.1	[オプション]: 構成 CDF 内のサイト固有情報を変更する .	7-12
7.7.2	複数のレコードを持つコンポーネント・グループに対して CDFMODE 属性を使用する	7-12
7.8	[ステップ 3]: 変更した CDF の検証	7-14
7.9	[ステップ 4]: フル・インストールの際に複数のシステムを複製するスクリプトを作成する (オプション)	7-15

7.10	[ステップ 5]: 構成 CDF を正しい位置にコピーする	7-15
7.10.1	構成 CDF をディスクセットにコピーする	7-16
7.10.2	構成 CDF をネットワークに接続されていないシステムに コピーする	7-16
7.10.3	構成 CDF を配布メディアにコピーする	7-18
7.11	[ステップ 6]: 構成 CDF をターゲット・システムに適用する .	7-19
7.11.1	フル・インストールの際に構成 CDF を適用する ..	7-19
7.11.2	稼働中のシステムに構成 CDF を適用する	7-19
7.11.3	構成 CDF を使用してシステム構成を復元する	7-20

A インストール CDF の属性と値

A.1	Inst_islinfo 項目の属性	A-2
A.2	Inst_filesystem 項目の属性	A-4
A.3	Inst_disklabel 項目の属性	A-6
A.4	Inst_lsm_global 項目の属性	A-7
A.5	Inst_lsm_disks 項目の属性	A-8
A.6	Inst_subsets 項目の属性	A-9
A.7	Inst_cinstall 項目の属性	A-10

B ユーザ提供ファイルの例

B.1	preinstall ファイルの例	B-1
B.2	update_preinstall ファイルの例	B-2
B.3	postload ファイルの例	B-3
B.4	update_postload ファイルの例	B-5
B.5	postreboot ファイルの例	B-6

索引

例

1-1	/sbin/ping コマンドの出力	1-2
-----	--------------------------	-----

5-1	他のファイル呼び出す preinstall スクリプトの例	5-15
5-2	preinstall スクリプト例から呼び出される DiskLabelSave ファイル	5-16
5-3	postload スクリプトの例	5-17
6-1	インストレーション CDF の例	6-6
6-2	install.cdf ファイルへの CDF 確認属性の追加	6-17
6-3	install.cdf ファイル内でのカーネル構築タイプの設定	6-18
7-1	config.cdf ファイルの一部	7-6
7-2	複数のレコードがあるコンポーネント・グループの例	7-12
7-3	CDF への CDFMODE 属性の挿入	7-14
B-1	preinstall スクリプトの例	B-1
B-2	update_preinstall ファイルの例	B-2
B-3	postload スクリプトの例	B-3
B-4	update_postload ファイルの例	B-6
B-5	postreboot ファイルの例	B-6

図

2-1	国際化機能構成アプリケーション	2-14
4-1	省略時の「File System Layout : Defaults」ウィンドウ	4-7
4-2	「Disk Configuration」アプリケーション: ディスク・パーティション情報	4-11
4-3	「Disk Configuration」アプリケーション: ディスク・パーティション・テーブル	4-12
5-1	管理者の作業の概要	5-6
5-2	動作原理: ユーザ提供ファイルと CDF	5-8

表

1-1	プロセッサ固有のネットワーク・ブート・コマンド	1-7
1-2	バスのタイプによる FDDI ブート・デバイス	1-8

2-1	WLS インストレーションで作成されるバックアップ・ファイル	2-19
4-1	ディスク容量ごとの推奨パーティション・テーブル	4-6
5-1	構成記述ファイルを読み出すタイミング	5-3
5-2	ユーザ提供ファイルを読み出すタイミング	5-4
5-3	ユーザ提供ファイルの一般的な使用目的	5-4
5-4	ユーザ提供ファイルと CDF の格納位置	5-19
6-1	install.cdf ファイルの検索順序	6-3
6-2	install.cdf ファイルの項目	6-5
6-3	モデル・システムとターゲット・システムの間で許容される ディスクの相違点	6-11
6-4	モデル・システムとターゲット・システムのグラフィックス・ アダプタ	6-12
6-5	モデル・システムとターゲット・システムのフォント・サイ ズ	6-14
6-6	モデル・システムとターゲット・システムのキーボード・タイ プ	6-15
6-7	prompt= 属性として有効な値	6-17
6-8	kernel_option= 属性として有効な値	6-18
6-9	timeset 属性として有効な値	6-21
6-10	install.cdf ファイルを置く位置	6-23
7-1	config.cdf ファイルの検索順序	7-3
7-2	config.cdf ファイルのコンポーネント	7-4
7-3	config.cdf ファイル内のホスト固有の属性	7-11
7-4	CDFMODE 属性の値	7-13
7-5	config.cdf ファイルを置く位置	7-18
A-1	イニシャル・サブセット・ロード (Inst_islinfo) 項目の属性定 義	A-2
A-2	ファイル・システム (Inst_filesystem) 項目の属性定義	A-5
A-3	ディスクラベル (Inst_disklabel) 項目の属性定義	A-7

A-4	LSM グローバル (Inst_lsm_global) 項目の属性定義	A-8
A-5	LSM ディスク (Inst_lsm_disks) 項目の属性定義	A-9
A-6	ソフトウェア・サブセット・ロード (Inst_subsets) 項目の属性 定義	A-9
A-7	インストレーション構成 (Inst_cinstall) 項目の属性定義	A-11



まえがき

本書では、HP Tru64 UNIX オペレーティング・システム・ソフトウェアの上級ユーザ向けのインストール手順について説明します。RIS サーバからのフル・インストールおよびアップデート・インストール、インストール・クローニング、構成のクローニング、ディスク・スペースの計画、フル・インストール後のワールドワイド言語サポート (WLS) のインストールなどについて説明します。

本書の対象読者

本書は、種々のインストール機能 (インストールや構成のクローニングなど) を利用しようとしている上級ユーザを対象としています。

新しい機能および変更された機能

Version 5.1A リリースと比べて、以下の点が変更されています。

- 国際化ソフトウェアの作成方法を説明しているドキュメントの紹介が、2.9 節に追加されました。
- UFS ファイル・システムと AdvFS ファイル・システムの違いが明確になりました (4.3.2 項を参照)。
- 構成をクローニングする際には、パッチがすでに適用されているシステムを使って `config.cdf` ファイルを作成しないでください。この制限事項と回避策の説明が、7.4 節に追加されました。

本書の以前のバージョンは、下記の URL で参照できます。

<http://tru64unix.compaq.co.jp/document/index.html>

本書の構成

本書の構成は次のとおりです。

第 1 章	RIS (Remote Installation Services) ネットワーク・サーバからフル・インストールおよびアップデート・インストールを実行するための準備作業および起動方法について説明します。
第 2 章	フル・インストール後に <code>wwinstall</code> スクリプトあるいは <code>setld</code> ユーティリティを使用して、ワールドワイド言語サポート (WLS) ソフトウェアをインストールする方法について説明します。
第 3 章	UNIX シェル環境における、ファイル・システムのリストア方法、ディスク・ラベルの変更方法、システム保守の実行方法について説明します。
第 4 章	推奨ディスク・パーティションを使用しないで手動でディスクのパーティショニングを行なう場合の、ディスク・スペースの計画手順について説明します。
第 5 章	ユーザ提供ファイルおよび構成記述ファイル (CDF) を使用してインストール・プロセスをカスタマイズおよび拡張する方法について説明します。
第 6 章	フル・インストールの際に、モデル・システムから 1 つあるいは複数のターゲット・システムへインストールを複製する、インストール・クローニング処理について説明します。
第 7 章	すでに構成されたモデル・システムから 1 つあるいは複数のターゲット・システムへ構成を複製する、構成クローニング処理について説明します。
付録 A	インストール構成記述ファイル (CDF) の属性と値をリストして定義しています。
付録 B	フル・インストールまたはアップデート・インストールの際に起動されるユーザ提供スクリプトのサンプルを提供します。

関連資料

高度なインストール作業を実行する際には、以下の関連資料が役に立ちます。

- ハードウェア・ドキュメント

プロセッサおよび追加デバイスのセッティング方法、コンソール環境変数の設定、トラブルシューティングのガイドラインについては、ハードウェアに付属のドキュメントを参照してください。

- 『ネットワーク管理ガイド：サービス編』

ネットワークの設定および管理については『ネットワーク管理ガイド：サービス編』を参照してください。

- 『*Sharing Software on a Local Area Network*』

RIS サーバの設定方法，ソフトウェア環境の作成および提供方法，クライアント・システムの登録方法については『*Sharing Software on a Local Area Network*』を参照してください。

- 『*AdvFS 管理ガイド*』

Advanced File System (AdvFS) の管理については『*AdvFS 管理ガイド*』を参照してください。

- 『システム管理ガイド』

インストール後のシステムの管理については『システム管理ガイド』を参照してください。

- Logical Storage Manager (LSM) の構成方法，管理方法については，『*Logical Storage Manager*』を参照してください。

- ドキュメント・セット内のすべてのマニュアルについての情報は，『ドキュメント概要』を参照してください。

Tru64 UNIX のドキュメントは，次の URL でも参照できます。

<http://tru64unix.compaq.co.jp/document/index.html>

本書の表記法

％

＄

パーセント記号は，C シェルのシステム・プロンプトを表します。ドル記号は，Bourne シェル，Korn シェル，および POSIX シェルの場合のシステム・プロンプトを表します。

#

番号記号は root としてログインした場合のシステム・プロンプトを表します。

％ cat

対話式の例における太字(ボールド体)は，ユーザが入力する文字を示します。

>>>	コンソール・サブシステム・プロンプトは 3 つの山カッコで表記されます。
<i>file</i>	イタリック体 (斜体) は、変数値、プレースホルダ、および関数の引数名を示します。
cat(1)	リファレンス・ページの参照には、該当するセクション番号をカッコ内に示します。たとえば、cat(1) は、cat コマンドについての情報が、リファレンス・ページのセクション 1 に記載されていることを示します。
Ctrl/x	この記号は、スラッシュの前に指定されているキーを押しながら、スラッシュの後のキーまたはマウス・ボタンを押すことを示します。例中では、このようなキーの組み合わせは、四角あるいは大カッコで囲まれて示されます(たとえば、 Ctrl/C)。

RIS サーバからのインストール

この章では、次の項目について説明します。

- RIS サーバからのインストールの準備作業 (1.1 節)
- 開始前に、ネットワークが構成されているか調べる簡単なテスト (1.2 節)
- RIS サーバからのアップデート・インストールの開始方法 (1.3 節)
- RIS サーバからのフル・インストールの開始方法 (1.4 節)

1.1 RIS サーバからのインストールの前提条件

ご使用のシステムがネットワーク環境にある場合は、次の条件を満たせば、オペレーティング・システムをネットワーク上のサーバからインストールすることができます。

- サイトに、リモート・インストール・サービス (RIS) サーバとして構成されているシステムがある。
- ご使用のシステムは、Version 5.1B のオペレーティング・システムをサービスする RIS サーバに、クライアントとして登録されている。

RIS 管理者への注意

Version 5.1B をクライアントに提供するためには、RIS サーバがバージョン 4.0 以降のオペレーティング・システムで稼働していなければなりません。

RIS サーバのセットアップ、サービスを行うためのソフトウェア環境の作成、正しいソフトウェア環境へのクライアント・システムの登録についての詳細が必要な場合は、『*Sharing Software on a Local Area Network*』を参照してください。

1.2 システムがネットワークに接続されているか

ご使用のシステムですでにオペレーティング・システムが稼働しており、アップデート・インストレーションを行いたい場合は、`/sbin/ping` コマンドを実行してネットワーク接続を確かめることにより、システムが RIS サーバと通信できることを確認します。

ネットワーク接続をテストするには、次のコマンドを入力します。

`ris_server_name` には、ローカルの RIS サーバ名を指定します。

```
# /sbin/ping -c2 ris_server_name
```

正常に動作した場合には、ネットワーク上の RIS サーバと通信できたことを示す、パケット紛失 0 % が表示されます。例 1-1 では、RIS サーバ名は `server1` です。

例 1-1: `/sbin/ping` コマンドの出力

```
# ping -c2 server1
PING server1 (16.59.124.96): 56 data bytes
64 bytes from 16.59.124.96: icmp_seq=0 ttl=255 time=1 ms
64 bytes from 16.59.124.96: icmp_seq=1 ttl=255 time=0 ms

----server1 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/1 ms
```

`/sbin/ping` コマンドの結果が次のようになった場合、システムは RIS サーバと通信できません。

- ネーム・サーバはシステム名を認識しているが、ネットワーク接続が設定されていない場合、システム名と IP アドレスのみがコマンドの出力として表示されます。
- システムが RIS サーバ名を認識していない場合、`Unknown host` というメッセージが表示されます。

このような結果になった場合は、ネットワーク管理者にトラブルシューティングを依頼してください。ネットワークのトラブルシューティングについての詳細は、『ネットワーク管理ガイド：接続編』および『ネットワーク管理ガイド：サービス編』を参照してください。

1.3 RIS サーバからのアップデート・インストールの開始

RIS サーバからアップデート・インストールを開始するには、次の手順を実行します。

1. 1.1 節で説明した、ネットワーク経由のインストールの前提条件をシステムが満たしていることを確認します。また、1.2 節で説明しているように、システムがネットワーク上で構成されていることも確認します。問題が発生した場合は、ネットワーク管理者に連絡するか、『*Sharing Software on a Local Area Network*』を参照してください。
2. アップデート・インストールの前提条件となっているすべての作業（『インストール・ガイド』を参照）を完了します。この作業には、現在のオペレーティング・システムのバックアップ、『リリース・ノート』の通読、システム・ファームウェアのアップデートが含まれます。
3. スーパユーザ (root) として `shutdown` コマンドを実行して、システムをシングルユーザ・モードにします。
 - 次の例では、スーパユーザになり、システムをシャットダウンして、シングルユーザ・モードに移行する方法を示します。

```
% su -  
password:  
# shutdown +10 Please log out
```

この例では、+10 は、10 分後にシステムをシャットダウンすることを示し、Please log out というメッセージをすべてのログイン・ユーザに送信します。

システムがシングルユーザ・モードになると、画面は次のようになります。

```
⋮  
Halting processes ...  
  
INIT: SINGLE-USER MODE  
#
```

4. ローカル・ファイル・システムをマウントします。

```
# /sbin/bcheckrc
```

bcheckrc コマンドは `mount -a` コマンドを呼び出し、標準の UNIX ファイル・システム (`/`, `usr`, `var`) だけでなく、`/etc/fstab` ファイル内のすべてのファイル・システムをマウントします。bcheckrc コマンドは、UNIX ファイル・システム (UFS) に対してファイル・システム・チェック・コマンド `fsck` も実行し、必要であれば LSM (Logical Storage Manager) を起動します。fsck が `/` (ルート) パーティションの問題を検出した場合、システムはシャットダウンします。この場合、システムをリブートして、ファイル・システムを修復しなければなりません。

5. インターネット・アドレスのテーブルを削除し、アップデート・インストール中に `routed` および `gated` デーモンが起動されないようにします。

```
# route flush
```

6. 次の形式で、`/sbin/installupdate` コマンドを入力します。

```
/sbin/installupdate [-u] [-nogui] {ris_server_name:}
```

各オプションについて、次に説明します。

- オプションの `-u` フラグを指定すると、ユーザの介入なしでアップデート・インストールが実行されます。このオプションを指定すると、障害となっているレイヤード・プロダクトの削除や旧ファイルの削除が自動で行われ、すべてのカーネル構成要素がインストールされます。したがって、アップデート中に質問に答える必要はありません。
- オプションの `-nogui` (非グラフィカル・ユーザ・インタフェース) フラグを指定すると、グラフィック機能があるシステムで、テキスト・ベース・インタフェースが起動されます。
- RIS サーバ名にはコロン (:) を付ける必要があります。

たとえば、`server1` という名前の RIS サーバからアップデート・インストールを実行するには、次のコマンドを入力します。

```
# /sbin/installupdate server1:
```

別の例として、テキスト・ベース・インタフェースを使用して `server2` という名前の RIS サーバから自動モードでアップデート・インストールを実行するには、次のコマンドを入力します。

```
# /sbin/installupdate -u -nogui server2:
```

注意

RIS サーバからアップデートを開始できない場合は、
『*Sharing Software on a Local Area Network*』のトラブル
シューティング情報を調べてください。

7. アップデートが開始されたら、『インストレーション・ガイド』のアップデート・インストレーションの残りの手順に進みます。

1.4 RIS サーバからのフル・インストレーションの開始

システムをネットワーク経由でブートして、RIS サーバからフル・インストレーションを開始するには、次の手順を実行します。

1. 1.1 節で説明した、ネットワーク経由でのインストールの前提条件を、システムが満たしていることを確認します。また、1.2 節で説明しているように、システムがネットワークに接続されていることも確認します。問題が発生した場合は、ネットワーク管理者に連絡するか、『*Sharing Software on a Local Area Network*』を参照してください。
2. アップデート・インストレーションの前提条件となっているすべての作業（『インストレーション・ガイド』を参照）を完了します。この作業は独立した章で説明されており、現在のオペレーティング・システムのバックアップ、『リリース・ノート』の通読、システム・ファームウェアのアップデートが含まれます。
3. システムをコンソール・モード (>>> プロンプト) にします。システムの現在の状態に応じて、次のいずれかの処理を実行します。
 - システムが稼働していて、任意のバージョンのオペレーティング・システムをすでに実行中の場合、次のようなコマンドを使用して、シャットダウンしプロセッサを停止させます。

```
# shutdown -h +10 Please log out
```

この例では、`Please log out` というメッセージがすべてのログイン・ユーザに送信され、システムは 10 分後にシャットダウンされて停止します。システムをシャットダウンしてコンソール・モードにする方法についての詳細は、『システム管理ガイド』または `shutdown(8)` を参照してください。

- システムの電源が入っていない場合は、プロセッサの電源を入れます。コンソール・サブシステムはさまざまなスタートアップ・メッセージと診断メッセージを出力し、コンソール・モード・プロンプト(>>>)を表示して入力待ちになります。ただし、システムに自動リブートが構成されている場合は、コンソール・モードでは停止せず、ブートが実行されてマルチユーザ・モードになります。システムをコンソール・モードにするには、前述のように shutdown -h コマンドを使用します。

4. 一部のプロセッサでは、すべてのプロセッサで設定される標準のコンソール変数の他に、1つ以上のコンソール環境変数を設定する必要があります。ハードウェアのフロント・パネルの社名ロゴを探して、プロセッサの種類を確認します。次に、『インストレーション・ガイド』の「フル・インストレーションの実行」の章に進み、ネットワーク経由でシステムをブートする前に、特殊なコンソール変数の設定が必要か調べます。そして、本手順のステップ5に進みます。

5. boot_osflags コンソール変数をクリアします。

```
>>> set boot_osflags ""
```

6. インストレーション時にシステムがクラッシュしたり、電源障害が発生した場合にシステムがコンソール・モードに戻るようするには、auto_action コンソール変数を次のように設定します。

```
>>> set auto_action halt1
```

7. ネットワーク・アダプタ・デバイス名を調べます。

```
>>> show device
```

次のようなデバイス情報テーブルが表示されます。

dka400.4.0.6.0	DKA400	RRD43	2893
dva0.0.0.0.1	DVA0		
ewa0.0.0.13.0	EWA0	08-00-2B-3E-B6-C8	
pka0.7.0.6.0	PKA0	SCSI Bus ID 7	

ネットワーク・ブート・デバイスは、3番目の欄にハードウェア・イーサネット・アドレスがある行の中央の欄に表示されます。この例では、ハードウェア・イーサネット・アドレスは 08-00-2B-3E-B6-C8 で、ブート・デバイスは EWA0 です。

¹ インストレーション完了後に、auto_action を restart に再設定し、電源やプロセッサの障害が発生した後でシステムが自動的にリブートするようにできます。

8. 次のコマンド構文を使用して、bootp ブート・リクエスト・プロトコルを初期化します。

set network_device_protocols bootp

set network_device_inet_init bootp

ネットワーク・ブート・デバイス名には大文字小文字の区別があり、小文字または大文字の英字が使用できます。たとえば、ステップ 7 で調べたネットワーク・アダプタ・デバイス名を使用して、次の bootp 初期化コマンドを入力します。

```
>>> set ewa0_protocols bootp
>>> set ewa0_inet_init bootp
```

9. 次のステップでシステムをブートする前に、表 1-1 を参照して、ご使用のプロセッサ・タイプに固有の boot コマンドがないか確認してください。

注意

表 1-1 の情報が、サポートされている各システムで、完全に正確なものになるように努めましたが、ご使用のプロセッサのハードウェア・ドキュメントもチェックして、他のコンソール変数や boot フラグの設定が必要でないことを確認してください。ハードウェア・ドキュメントは、システム・タイプごとにカスタマイズされており、必要となるコンソール変数および boot フラグの最終的な情報となります。

表 1-1: プロセッサ固有のネットワーク・ブート・コマンド

プロセッサまたはオプション	ブート・コマンド
AlphaStation 500 , 600 , 600A AlphaServer 800 , 1000 , 1000A Personal Workstation au シリーズ Ultimate Workstation au シリーズ Professional Workstation XP1000	>>> boot -f1 "" ewa
AlphaServer 4000 , 4100 Alphaserver 8200 , 8400 ES40 , GS60 , および GS140 サーバ	>>> boot -f1 "" ewa0
FDDI デバイスがあるシステム	1.4.1 項 を参照

10. ブート・ファイルをリセットします。

```
>>> set boot_file ""
```

11. 次の構文で、boot コマンドを入力します。

```
boot network_device
```

たとえば、ステップ 7 で取得した情報を使用するときには、boot コマンドは次のようになります。

```
>>> boot ewa0
```

これで、フル・インストールを起動するための、ネットワーク経由でのシステムのブートは完了です。フル・インストールの残りの手順は、『インストール・ガイド』で説明しています。

ネットワーク経由でのシステムのブートで問題が発生した場合は、『*Sharing Software on a Local Area Network*』を参照してください。

1.4.1 ネットワーク・ブート・オプション: FDDI (Fiber Distributed Data Interface)

DEFTA デバイスでは、FDDI ネットワーク・インタフェースを経由したブートがサポートされています。

FDDI ネットワーク・インタフェース・デバイスからブートするには、システムがコンソール・レベルのときに show config コマンドを入力して、デバイス名を調べます。たとえば、show config コマンドを実行すると、DEFTA デバイスは PMAF-FA として表示されます。スロット番号を確認し、表 1-2 を参照してブート・コマンドを入力します。

表 1-2 に、FDDI を使用しているシステムで、ネットワークを経由してブートするために必要なブート・デバイスを示します。ネットワーク経由でブートする前に、ハードウェアのマニュアルの指示に従って、FDDI のファームウェアをアップデートしてください。

表 1-2: バスのタイプによる FDDI ブート・デバイス

バスのタイプ	ブート・デバイス
EISA (Extended Integrated System Architecture)	fra0 ^a

表 1-2: バスのタイプによる FDDI ブート・デバイス (続き)

バスのタイプ	ブート・デバイス
PCI (Peripheral Component Interconnect)	fwa0 ^a
XMI (Extended Memory Interface)	fxa0 ^a

^aネットワーク経由でブートする前に、ハードウェア・アドレスを調べて RIS サーバにシステムを登録しておかなければなりません。コンソール・モード・プロンプト (>>>) から `show dev` コマンドを入力すると、ハードウェア・アドレスを調べることができます。

1.4.2 ネットワーク・ブートの失敗のトラブルシューティング

最初のネットワーク・ブートに失敗した場合は、コンソール・プロンプトに `init` と入力して、ネットワーク経由でのブートを再試行します。リモート・サーバ・インストールの際に別の問題が発生した場合は、『*Sharing Software on a Local Area Network*』を参照してください。このガイドには、ネットワーク・ブートの失敗についてのトラブルシューティング情報が記載されています。

1.4.3 ISA バスにグラフィックス・デバイスがあるシステムのネットワーク・リブートに関する考慮事項

ソフトウェア・サブセットがロードされた後にシステムが自動的にリブートしないとき、またカーネル・デバイス・ドライバを必要とする ISA バスのグラフィックス・デバイスがシステムにある場合には、システムをリブートする前に、`isacfg` エントリを変更して、カーネル・デバイス・ドライバに一致させなければなりません。

次のコマンドを入力する際には、行の途中で Return キーを押さないで、入力行を自然に改行させてください。バックスラッシュ (\) 文字は、行の継続を示しています。コマンド行ではバックスラッシュを入力しないでください。

```
>>> isacfg -mod -slot slot_number -dev device_number \
      -handle vendor_handle -etyp 1 -enadev 1
```

この例では、`vendor_handle` に、ベンダのインストール・ドキュメントに記載されているハンドルを指定してください。

RIS サーバ上の汎用カーネルにカーネル・グラフィックス・デバイス・ドライバがすでに組み込まれている RIS 領域から RIS インストールを実行し、ベンダのインストール・ドキュメントに記載されているハンドルをすでに設定している場合は、このコマンドを実行する必要はありません。

システムが自動リブート機能をサポートしていない場合は、ブート用のコマンドが画面上に表示されます。

ワールドワイド言語サポートのインストール手順

この章では、次の項目について説明します。

- ワールドワイド言語サポート (WLS) の概要 (2.1 節)
- WLS インストールに備えてシステムを準備する方法 (2.2 節)
- WLS ソフトウェア用に、利用可能なディスク・スペースを増加させる方法 (2.3 節)
- `wwinstall` スクリプトを使用して、CD-ROM から WLS ソフトウェア・サブセットをインストールする方法 (2.4.1 項)
- `setld` コマンドを使用して、RIS サーバからネットワーク経由で WLS ソフトウェア・サブセットをインストールする方法 (2.4.2 項)
- WLS インストールの際のプロンプトへの応答方法 (2.5 節)
- `/usr/sbin/wwconfig` を起動して、アジア系端末オプションをカーネルに組み込む方法 (2.6.1 項)
- ワールドワイド・インストールで作成されるバックアップ・ファイル (2.7 節)
- 言語固有の環境変数の設定方法 (2.8 節)
- 関連ドキュメントの紹介 (2.9 節)

2.1 ワールドワイド言語サポートの概要

WLS ソフトウェア・サブセットは、標準でインストールされるアメリカ英語以外の、さまざまな言語および国々に対するサポートを提供します。WLS ソフトウェア・サブセットをインストールすると、ソフトウェア開発者は、さまざまな国々で利用できる国際化されたソフトウェアを開発することができます。ユーザは母国語の環境で作業することができます。

『インストール・ガイド』で説明しているように、フル・インストール中に使用言語を選択することにより、1 つ以上の言語をサポートするためのソフトウェアをインストールすることができます。WLS ソフトウェアのインストールでは、この方法をお勧めします。ただし、この章で説明する手順を使用すると、ベース・オペレーティング・システムのインストール後に、言語サポートを追加することもできます。

WLS ソフトウェアは、`/usr/i18n` ディレクトリにインストールされます。このディレクトリは、独立したファイル・システムとして作成することも、別のファイル・システムへのシンボリック・リンクとして作成することもできます。 `wwinstall` コマンドを使用して WLS をインストールしているときに、`/usr/i18n` ディレクトリ (ファイル・システム) が存在していない場合は、このディレクトリを作成するように求められます。なお、**i18n** は国際化 (*internationalization*) の略語です。

各言語でサポートされている端末およびプリンタについての詳細は、『QuickSpecs』を参照してください。

2.2 [ステップ 1]: WLS インストールの準備

WLS のインストールを開始する前に、次の作業を実行してください。

1. インストール前に行う一般的な準備作業について説明している、『インストール・ガイド』のインストールの準備作業の節を読みます。
2. インストールしようとしている WLS と同じバージョンのベース・オペレーティング・システムがすでにシステムにインストールされていることを確認します。
3. 現行のバージョンの *Operating System Volume 1* を用意します。システムにインストールされていないオペレーティング・システム・サブセットに WLS サブセットが依存している場合は、`setld` コマンドを使用して、そのオペレーティング・システム・サブセットをインストールします。WLS のインストール処理の際に、ベース・オペレーティング・システム・ソフトウェア・サブセットへの依存性が調べられます。WLS のインストールを中断して、依存しているサブセットをインストールしてから作業を続行します。

4. RIS サーバから WLS ソフトウェア・サブセットをインストールしている場合は、ご使用のシステムが RIS サーバ上でクライアントとして登録されていることを確認します。ご使用のシステムが登録されている RIS 領域には、WLS プロダクトが含まれていなければなりません。

RIS サーバおよび RIS クライアントのセットアップ方法の詳細については、『*Sharing Software on a Local Area Network*』を参照してください。

2.3 [オプション]: WLS ソフトウェア用に利用可能なディスク・スペースの拡大

WLS インストール・プロシージャは、ほとんどの WLS ファイルを、`/usr/i18n` ディレクトリ下のサブディレクトリにロードします。`/usr/i18n` ディレクトリが存在しない場合は、インストール・プロシージャがそのディレクトリを作成します。ソフトウェアの選択処理では、すべての WLS ソフトウェア・サブセットは、`/usr` ファイル・システムのサイズに含めて計算されます。`/usr/i18n` ディレクトリが存在する場合、インストール・プロシージャはこのディレクトリを使用します。

WLS ソフトウェア・サブセット用に十分なディスク・スペースがなく、システム上の別のディスクやディスク・パーティションに空きスペースがある場合は、ファイル・システムのタイプによって 2.3.1 項または 2.3.2 項の手順に従い、ディスク・スペースを大きくしてください。

2.3.1 UFS ファイル・システムのディスク・スペースを大きくする

インストール手順を開始する前に、次の手順に従い、UNIX ファイル・システム (UFS) の `/usr/i18n` ディレクトリのディスク・スペースを大きくします。

1. `root` としてログインするか、`su` コマンドを使用してスーパーユーザになります。
2. `/usr/i18n` ディレクトリが存在しない場合は、このディレクトリを作成します。

```
# mkdir /usr/i18n
```
3. `/etc/fstab` ファイルに次のような行を追加して、新しく作成したディレクトリが追加スペースのあるディスク・パーティションのマウント・ポイントになるように指定します。

```
/dev/disk/dsk2c /usr/i18n ufs,rw 0 0
```

4. 新しいマウント・ポイント /usr/i18n へのマウントを実行します。

```
# mount -a
```

2.3.2 AdvFS ファイル・システムのディスク・スペースを大きくする

インストール手順を開始する前に、次の手順に従い、/usr/i18n ディレクトリ用の AdvFS ファイル・ドメインを作成します。

1. root としてログインするか、su コマンドを使用してスーパーユーザになります。
2. /usr/i18n ディレクトリが存在しない場合は、このディレクトリを作成します。

```
# mkdir /usr/i18n
```

3. 次のコマンド構文を使用して、AdvFS ドメインを作成します。

```
mkfdmn /dev/disk/dsk <disk_number>c domain_name
```

たとえば、dsk2 上に i18n_domain ドメインを作成するには、次のようにします。

```
# mkfdmn /dev/disk/dsk2c i18n_domain
```

4. AdvFS ファイルセットを作成します。

```
# mkfset i18n_domain i18n
```

5. テキスト・エディタを使用して次の行を /etc/fstab ファイルに追加し、新しく作成したドメインをマウントできるようにします。

```
i18n_domain#i18n /usr/i18n advfs rw, 0 0
```

6. 新しいファイル・ドメインをマウントします。

```
# mount -t advfs i18n_domain#i18n /usr/i18n
```

2.4 [ステップ 2]: WLS インストールの開始

WLS インストールの開始方法は、配布メディアのソースによって少し異なります。

- 「Associated Products Volume 1」CD-ROM を使用する場合は、2.4.1 項に進みます。

- リモート・インストール・サービス (RIS) を使用する場合は、2.4.2 項に進みます。

インストール・プロシージャの途中で、システムの構成に関するいくつかの質問があります。説明が必要な場合は、疑問符 (?) を入力するとオンライン・ヘルプが表示されます。

2.4.1 CD-ROM からのインストール

CD-ROM から `wwinstall` スクリプトを起動するには、次の手順を実行します。

1. 「*Associated Products Volume 1*」CD-ROM を、CD-ROM ドライブにロードします。
2. この CD-ROM のマウント・ポイント・ディレクトリを作成して、CD-ROM をマウントします。

```
# mkdir /cdrom
# mount /dev/disk/cdrom0c /cdrom
# cd /cdrom/Worldwide_Language_Support/kit
```

この例では、`/dev/disk/cdrom0c` が CD-ROM のデバイス名です。CD-ROM デバイスのデバイス名が分からない場合には、次のコマンドを入力します。

```
# ls /dev/disk/cdrom*c
```

3. `wwinstall` スクリプトがあるディレクトリに移動します。

```
# cd /cdrom/Worldwide_Language_Support/kit
```
4. `wwinstall` スクリプトを起動します。

```
# ./wwinstall
```

2.5 節に進んで、WLS インストール・プロシージャを続行します。

2.4.2 RIS サーバからのインストール

この項では、`setld` ユーティリティを使用して RIS サーバから WLS ソフトウェア・サブセットをインストールする方法について説明します。`wwinstall` コマンドを使用したい場合は、ネットワーク・ファイル・システム (NFS) マウント機能を使用して、エクスポートされた RIS 領域をマウントします。この後、マウントされた RIS 領域から `wwinstall` を実行できます (ここでは、この手順は説明していません)。

RIS サーバから WLS インストールを開始するには、次の手順を実行します。

1. インストールするシステム上でネットワークが構成されていない場合は、クイック・セットアップ・アプリケーションを使用して、基本ネットワーク・サービスをセットアップします。クイック・セットアップは、System セットアップ・アプリケーションから利用できます。

```
# /usr/sbin/checklist
```

2. RIS サーバ管理者に、ご使用のシステムが、現在のバージョンの WLS ソフトウェアをサービスするサーバのクライアントとして登録されているか確認します。

3. WLS インストール処理を開始します。

```
# setld -l ris_server_name:
```

指定する RIS サーバの名前にコロン (:) を付けて指定します。 2.5 節に進んで、WLS インストール・プロシージャを続行します。

2.5 [ステップ 3]: WLS インストール処理中のユーザ 応答

以下の例では、必要なベース・オペレーティング・システム・ソフトウェア・サブセットがすべてインストールされているものとして説明します。CD-ROM と RIS のどちらを使用する場合も、インストールの画面はほぼ同じです。ただし、setld ユーティリティを使用して RIS サーバからインストールする場合は、別のパーティションへのシンボリック・リンクを作成して WLS ソフトウェアを格納することはできません。

wwinstall スクリプトを使用する場合、/usr/i18n 領域で利用可能なディスク・スペースを計算し、/usr/i18n 領域をどのように作成するかをユーザが選択できます。/usr/i18n に必要なディスク容量は、選択した国または言語により異なります。たとえば、日本語のサポートをインストールする場合は、イタリア語のサポートをインストールする場合に必要なディスク・スペース (約 20 MB) よりも大きなディスク・スペース (約 200 MB) を必要とします。各言語で必要とするディスク・スペースについては、『リリース・ノート』を参照してください。

CD-ROM からインストールしている場合、あるいは `wwinstall` コマンドを実行して RIS サーバから WLS をインストールしている場合は、次のようなメッセージが表示されます。

```
Most of the subsets will be installed under the /usr/i18n directory.
Depending on the number of subsets you choose to install, you may
need more than 200 MB of free disk space for installation.
```

```
You have the following amount of free disk space
available in /usr:
```

```
$ df -k /usr/i18n
Filesystem      1024-blocks      Used   Available Capacity  Mounted on
usr_domain#usr      716800      238120    456512      35%    /usr
```

```
Two ways to set up the /usr/i18n directory :
```

- [1] Create the /usr/i18n directory
- [2] Set up a symbolic link to another partition that has enough free disk space for installation

```
Which way do you want ? [1] :
```

次のいずれかの処理を実行します。

- Return を押すか 1 を入力して、`/usr/i18n` ディレクトリを作成する。次のメッセージが表示されます。

```
Directory /usr/i18n has been created
```

- 2 を入力して、別のディスク・パーティションへのシンボリック・リンクを作成する。次のようなプロンプトが表示されるので、シンボリック・リンクのリンク先にしたいパスを入力します。

```
You have chosen to make a symbolic link to another partition.
Please enter the installation path: /var/i18n
```

```
You have the following amount of free disk space available in
/usr/i18n :
```

```
$ df -k /usr/i18n
Filesystem      1024-blocks      Used   Available Capacity  Mounted on
usr_domain#usr      716800      238120    456512      35%    /usr
Do you want to continue this installation procedure? (y/n) [y]:y
```

インストール・スクリプトにより、インストールできる国と言語の一覧が表示されます。

```
*****
*
*   Tru64 UNIX WORLDWIDE LANGUAGE SUPPORT INSTALLATION PROCEDURE   *
*
*****
```

```
Please select the countries for which you want to install support subsets:
```

- | | |
|-----------------------------|-------------------------------|
| 1) Belgium (French) | 2) Canada (French) |
| 3) China (Hong Kong) | 4) China (Simplified Chinese) |
| 5) China (Taiwan) | 6) Czech Republic |
| 7) Euro (Latin-9 & Unicode) | 8) France |

9) Germany	10) Greece
11) Hungary	12) Israel
13) Italy	14) Japan
15) Korea	16) Lithuania
17) Poland	18) Russia
19) Slovakia	20) Slovenia
21) Spain (Catalan)	22) Spain (Spanish)
23) Sweden	24) Switzerland (French)
25) Switzerland (German)	26) Thailand
27) Turkey	
28) All of the above	
29) None of the above	

Choices (for example, 1 2 3) :

プロンプトに対して複数の番号を指定する場合は、番号をスペースで区切ります。選択を行うと、次のようなメッセージが表示されます。

You are installing localized software for the following countries:

<list of countries>

Is this correct? [n]

この質問に対して次のように応答します。

- ソフトウェア・サブセット選択メニューを再表示して国をもう一度選択するには、**n** を入力します。
- この選択でよい場合は、**y** を入力します。

選択した国によっては、次の質問が表示されることがあります。

- アウトライン・フォントのインストール

Would you like to install outline fonts for printing and display? [y] :

アウトライン・フォントをインストールしたい場合は、**y** を入力します。アウトライン・フォントをインストールすると印刷および表示出力が見やすくなりますが、かなりのディスク・スペースを消費します。

- パブリック・ドメイン・ソース・ファイルのインストール

Would you like to install program sources of the public domain software packaged in the Worldwide Language Support kit? [n] :

ワールドワイド言語サポート・ソフトウェアに含まれているパブリック・ドメイン・ソフトウェアのソース・ファイルをインストールしたい場合は、**y** を入力します。GNU のライセンス・ガイドラインに従って、パブリック・ドメインの MULE、および Wnn のソースが提供されています。MULE は、GNU Emacs Version 19 をベースにマルチリンガ

ル機能を追加した GNU Emacs です。 Wnn は、日本語を入力するための日本語入力システムです。

2.5.1 項では、WLS ソフトウェアの選択処理について説明します。 2.5.2 項では、WLS ソフトウェアのロード処理について説明します。 2.5.3 項では、WLS ソフトウェアの構成処理について説明します。

2.5.1 ワールドワイド・ソフトウェア・サブセットの選択

利用可能なソフトウェア・サブセットのメニューが表示されます。メニューにはまず、選択した国または言語で必須のソフトウェア・サブセットがリストされます。これらのソフトウェア・サブセットは、自動的にロードされます。その後このメニューには、選択した国に応じて、インストール可能なオプションのソフトウェア・サブセットがリストされます。プロンプトに対して複数の番号を指定する場合は、番号をスペースかコンマで区切ります。範囲を指定する場合は、ハイフン (-) を使用します。

次の例では、Japan を選択した場合に利用できる、オプションのソフトウェアを示します。ソフトウェア・サブセット・リストは、次のように表示されます。

```
*** Enter subset selections ***
```

```
The following subsets are mandatory and will be installed automatically
unless you choose to exit without installing any subsets:
```

```
* Japanese Standard Kernel Modules
* Japanese CDE Mail Interface
* Japanese Base System
* Japanese Base System Management Applications and Utilities
* Japanese Graphical Base System Management Utilities
* Japanese Graphical System Administration Utilities
* Japanese Basic X Environment
* Japanese CDE Desktop Environment
* Japanese CDE Minimum Runtime Environment
* Japanese DECwindows 100dpi Fonts
* Japanese DECwindows 75dpi Fonts
* Japanese DECwindows Fonts
* Japanese Netscape Communicator V4.7
```

```
The subsets listed below are optional:
```

```
There may be more optional subsets than can be presented on a single
screen. If this is the case, you can choose subsets screen by screen
or all at once on the last screen. All of the choices you make will
be collected for your confirmation before any subsets are installed.
```

```
- Japanese Support - General Applications :
  1) Additional Japanese Software
```

```
--- MORE TO FOLLOW ---
```

```
Enter your choices or press RETURN to display the next screen.
```

Estimated free disk space(MB) in root:53.1 usr:346.7

Choices (for example, 1 2 4-6):

- 2) Japanese DOS tools
- 3) Wnn Input Method
- Japanese Support - Reference Pages :
 - 4) Japanese Ref. Pages for Additional Software
 - 5) Japanese Reference Pages
 - 6) Japanese Windows Reference Pages
- Japanese Support - Software Development :
 - 7) Japanese CDE Software Development
 - 8) Japanese Ladebug Debugger Version 4.0
 - 9) Japanese Software Development
 - 10) Japanese Software Development Desktop Environment
 - 11) Japanese X Window Software Development
 - 12) Wnn Software Development
- Japanese Support - System Administration :
 - 13) Japanese Advanced File System Graphical User Interface
 - 14) Japanese Logical Storage Manager GUI
- Japanese Support - Windowing Environment :
 - 15) Japanese (SJIS) CDE Online Help
 - 16) Japanese CDE Online Help
 - 17) Japanese DECwindows Additional 100dpi Fonts
 - 18) Japanese DECwindows Additional 75dpi Fonts
- Japanese Support - Windows Applications :
 - 19) Japanese Additional DECwindows Applications
 - 20) Japanese CDE Additional Applications
- Worldwide Language Support - General Applications :
 - 21) Worldwide MULE
- Worldwide Language Support - Obsolete Components :
 - 22) Worldwide Obsolete Commands and Utilities
- Worldwide Language Support - Operating System :
 - 23) Worldwide European Unicode Locales
 - 24) Worldwide Phrase Input Support
 - 25) Worldwide User Defined Character Support
- Worldwide Language Support - Software Development :
 - 26) Worldwide SVE MNLs Migration Tools
 - 27) Worldwide Software Development
 - 28) Worldwide X Window Software Development
- Worldwide Language Support - Windowing Environment :
 - 29) Worldwide Composite Unicode Fonts
 - 30) Worldwide DECwindows Additional Fonts
 - 31) Worldwide Two-Byte Outline Font Renderer
 - 32) Worldwide User Defined Character Workstation Service
 - 33) Worldwide X/Motif 1.1

--- MORE TO FOLLOW ---

Add to your choices or press RETURN to display the next screen.

Estimated free disk space(MB) in root:53.1 usr:337.0

2-10 ワールドワイド言語サポートのインストール手順

Choices (for example, 1 2 4-6): 3

The following choices override your previous selections:

- 34) ALL mandatory and all optional subsets
- 35) MANDATORY subsets only
- 36) CANCEL selections and redisplay menus
- 37) EXIT without installing any subsets

Estimated free disk space(MB) in root:53.1 usr:337.0

Add to your choices, choose an overriding action or
press RETURN to confirm previous selections.

Choices (for example, 1 2 4-6): 3

次の例に示すように、選択内容を確認することができます。

You are installing the following mandatory subsets:

- Japanese Standard Kernel Modules
- Japanese CDE Mail Interface
- Japanese Base System
- Japanese Base System Management Applications and Utilities
- Japanese Graphical Base System Management Utilities
- Japanese Graphical System Administration Utilities
- Japanese Basic X Environment
- Japanese CDE Desktop Environment
- Japanese CDE Minimum Runtime Environment
- Japanese DECwindows 100dpi Fonts
- Japanese DECwindows 75dpi Fonts
- Japanese DECwindows Fonts
- Japanese Netscape Communicator V4.7

You are installing the following optional subsets:

- Japanese Support - General Applications :
 - Wnn Input Method

Estimated free disk space(MB) in root:53.1 usr:337.0

Is this correct? (y/n): y

ソフトウェア・サブセットを選択し直したい場合は、**n** を入力します。リストが正しい場合は、**y** を入力します。

2.5.2 項では、WLS ソフトウェアのロード処理について説明します。

2.5.2 ワールドワイド・ソフトウェア・サブセットのロード

インストール・プロセスは、選択されたサブセットをロードできるだけのディスク・スペースがあるか確認します。次のようなメッセージが表示されます。

Checking file system space required to install selected subsets:
Working...Fri Aug 24 13:47:50 EST 2001

File system space checked OK.

選択したすべてのソフトウェア・サブセットを格納できるだけのディスク・スペースがない場合は、前に戻って、選択するオプション・ソフトウェア・サブセットの数を減らします。

次に、インストール・プロセスはシステム上にソフトウェア・サブセットをインストールします。次のようなメッセージが表示されます。

```
14 subsets will be installed.

Loading subset 1 of 14 ...

Japanese Base System
  Copying from system9 (inet)
    Working....Fri Aug 24 13:49:58 EST 2001
  Verifying

Loading subset 2 of 14 ...

Wnn Input Method
  Copying from system9 (inet)
    Working....Fri Aug 24 13:50:30 EST 2001
  Verifying

Loading subset 3 of 14 ...

Japanese Basic X Environment
  Copying from system9 (inet)
  Verifying
:
:

Loading subset 12 of 14 ...

Japanese Base System Management Applications and Utilities
  Copying from system9 (inet)
  Verifying

Loading subset 13 of 14 ...

Japanese CDE Mail Interface
  Copying from system9 (inet)
  Verifying

Loading subset 14 of 14 ...

Japanese Standard Kernel Modules
  Copying from system9 (inet)
  Verifying

14 of 14 subsets installed successfully.
```

2.5.3 項では、WLS ソフトウェアの構成処理について説明します。

2.5.3 ワールドワイド・ソフトウェア・サブセットの構成

次に、サブセットの構成 (オペレーティング・システム・ソフトウェアの調整) が行われます。画面出力を十分にチェックしてください。インストールしたソフトウェア・サブセットによっては、セットアップ・ス

クリプト実行の指示が表示される場合があります。CD-ROM からインストールを実行した場合には、インストール完了後に、
/var/adm/smlogs/wwinstall.log ログ・ファイルでインストールの記録を見直してください。RIS によるインストールでは、ログ・ファイルは作成されません。

```
Configuring "Japanese Base System" (IOSJPBASE540)
Configuring "Wnn Input Method" (IOSJPWNN540)
Configuring "Japanese Basic X Environment" (IOSJPIX11540)
:
:
Configuring "Japanese CDE Mail Interface" (IOSJPCDEMAIL540)
Configuring "Japanese Standard Kernel Modules" (IOSJPBIN540)
```

2.5.4 項では、カーネル構築処理について説明します。

2.5.4 カーネルの構築

必要であれば、ソフトウェア・サブセットの構成後、カーネルの構築が自動的に開始されます。

注意

データレス・インストールを行った場合、カーネルの構築は自動的には行われません。この場合は、doconfig コマンドを使用してカーネルを構築してください。詳細は、doconfig(8) を参照してください。

カーネルの構築時には、次のようなメッセージが表示されます。最初のログイン画面に言語オプションを含めるためには、X サーバの再スタートが必要になることがありますので、この情報を注意深く読んでください。

```
*****

Reconfiguring kernel to incorporate Asian/Thai tty drivers
*****

**** Adding Worldwide Support tty Features into Kernel Configuration File ****
Loading I18N tty kernel modules ... done

The installation software has completed the installation process.
The log file /var/adm/smlogs/wwinstall.log contains a record of
your installation.

Total installation time = 23 minutes 35 seconds.

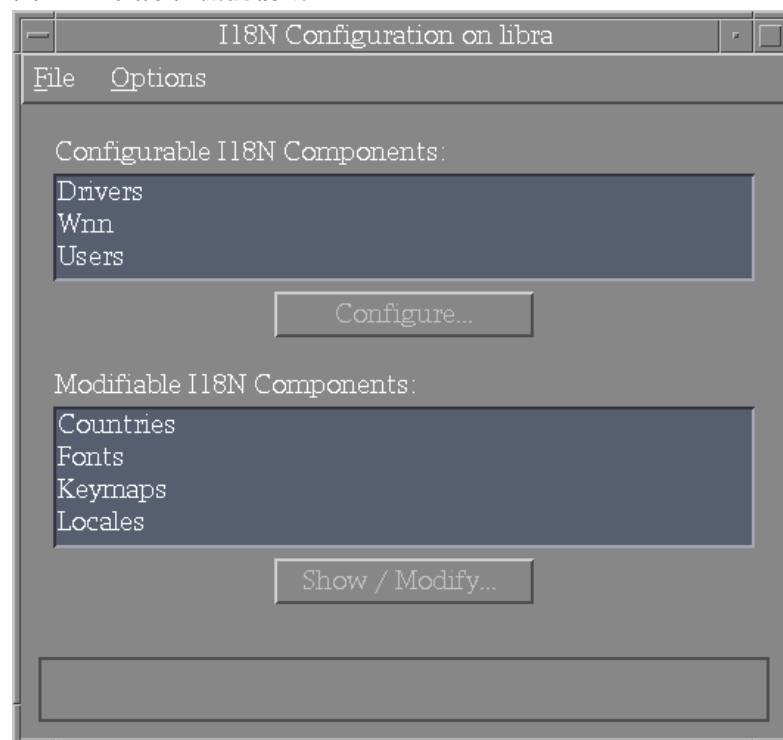
Do you want to start il8nconfig at this time? (y/n) [y] : y
```

カーネルの構築に失敗した場合は、`/var/adm/smlogs/it.log` のログ・ファイルを参照して、原因を調べてください。2.5.5 項では、この時点で使用可能な国際化機能構成ユーティリティについて説明します。

2.5.5 国際化機能構成アプリケーションの使用

カーネルの構築が完了したら、`i18nconfig` アプリケーションを使用できます。`i18nconfig` アプリケーションを使用すると、システムにインストールされているワールドワイド・サポート・オプション (ドライバ、フォント、Wnn 日本語入力システム、ユーザ・アカウントなど) をセットアップおよびメンテナンスしたり、インストールされているキーマップやロケールを参照することができます。これらのオプションを構成すると、国際化アプリケーションの作成や使用のためのオペレーティング・システム環境が確立されます。図 2-1 に、このアプリケーションのメイン・ウィンドウを示します。

図 2-1: 国際化機能構成アプリケーション



2.6 [ステップ 4]: インストール後のアジア言語用カーネルの構築 (オプション)

日本、中国、香港、韓国、台湾、またはタイの言語をサポートするソフトウェア・サブセットをインストールした場合、ワールドワイド・インストレーション・プロセスは、インストールされたアジア言語またはタイ語の端末をサポートするカーネルを構築します。カーネル内のアジア言語またはタイ語の端末サポートを使用可能にするには、その後、新しいカーネルでシステムをリブートする必要があります。

カーネルでのアジア言語またはタイ語端末のサポートを有効または無効にしたい場合は、2.6.1 項のアジア言語用カーネルの再構築手順を参照してください。

2.6.1 アジア言語用のターミナル・ドライバおよびデーモンをサポートするためのカーネルの再構成

アジア言語用ターミナル・ドライバおよびデーモンをサポートするためにカーネルを再構成するには、`-a` オプションを指定して `wwconfig` スクリプトを実行します。

```
# /usr/sbin/wwconfig -a
```

詳細については、`wwconfig(8)` を参照してください。

`IOSWWBIN540` をインストールしており、`IOSWWUDCOS540` (オン・デマンド・フォント・ローディング)、`IOSWWPHRASE540` (ソフトウェア入力メソッド)、または `IOSJPBASE540` (かな漢字変換) の少なくとも 1 つのサブセットをインストールしている場合は、`UTX` の構成に関する次のようなメッセージが表示されます。`UTX` は、アジア言語用のターミナル・ドライバとデーモンの間の通信のためのメカニズムです。

```
*** UTX CONFIGURATION SELECTION ***
```

Selection	Asian service Daemon
1	On Demand Font Loading (odld)
2	Software Input Method (simd)
3	Kana-Kanji Conversion (kkcd)
4	All of the above
5	None of the above

Enter the selection number for each daemon you want.

For example, 1 2 :

番号を選択すると、選択したデーモンが確認のために表示されます。「4 (All of the above)」を選択した場合は、次の確認メッセージが表示されます。

You specified the following daemons:

```
On Demand Font Loading (odld)
Software Input Method  (simd)
Kana-Kanji Conversion  (kkcd)
```

Is this correct? (y/n) [n]:

表示された内容が正しければ、y を入力します。

インストレーション・プロシージャは次に、作成する UTX デバイスの数を聞いてきます。

How many UTX devices do you want to create? [default: 32] :

入力した数値は、/var/i18n/sys/stanza.loadable ファイルに保存されます。UTX デバイスの作成は、システムのリブート時に行われます。

UTX デバイスを 1 つ使用する utxd マスタ・デーモンが 1 つあります。odld、simd、および kkcd のいずれかのデーモンを起動するたびに、UTX デバイスが 1 つ使用されます。ユーザが、他の odld でまだ処理されていないデータベース上で odld をオンにすると、新しい odld プロセスが開始されます。詳細については、リファレンス・ページの stty(1) および cedit(1) を参照してください。ソフトウェア語句入力メソッド (Software Phrase Input Method) をオンにしている各ユーザ・セッションでは、simd が 1 つ必要になります。ユーザが、別の kkcd でまだ処理されていないデータベース上のかな漢字変換をオンにすると (詳細については、stty(1) を参照)、新しい kkcd プロセスが開始されます。たとえば、10 人のユーザに対してそれぞれ 3 つのサービスをサポートするには、UTX デバイスが 31 個必要になります。

IOSZHBIG5540、IOSZHTELEX540、および IOSZHCONV540 ソフトウェア・サブセットをインストールしている場合は、次のメニューが表示されます。

*** ADDITIONAL TERMINAL CODESETS SELECTION ***

Selection	Terminal Codeset
1	BIG-5
2	Telecode
3	Traditional & Simplified Chinese Conversion

- 4 All of the above
- 5 None of the above

Enter the selection number for each codeset you want.
For example, 1 2 :

このプロンプトで端末コードセットを選択すると、ターミナル・ドライバに端末コードセットのサポートが組み込まれます。コードセット・サポートがターミナル・ドライバに組み込まれると、`/usr/lib8n/bin/stty` コマンドを使用することにより、そのコードセットを端末コードとして選択できます。

端末コードに簡体字中国語コードセット、アプリケーション・コードに繁体字中国語コードセットが設定されているとき (または逆のとき) に、適切なコードセット変換をサポートしたい場合は、3 を選択します。

3つのソフトウェア・サブセットのうち2つだけがインストールされている場合は、選択メニューは表示されますが、インストールされていない構成要素はリストされません。

サブセットのうち1つだけをインストールしている場合は、代りに質問が表示されます。次の例に示すように、`IOSTHBIN540` ソフトウェア・サブセットをインストールしている場合は、タイ語用ターミナル・ドライバをカーネルに追加するかどうかの質問が表示されます。

Do you want to install the Thai tty driver? (y/n) [y] :

タイ語用ターミナル・ドライバは、タイ語端末入力/出力 (I/O) をサポートします。その他のアジア言語は、アジア言語用ターミナル・ドライバによってサポートされます。`IOSTHBIN540` ソフトウェア・サブセットだけをインストールして、`IOSWWBIN540` ソフトウェア・サブセットをインストールしていない場合には、この質問だけが表示されます。

インストール・プロシージャは次に、カーネルを再構築するかどうかを質問します。

If you wish, you may use an automated kernel build procedure by answering 'y' to the next question.

You will need about 10 MB available in the /sys file system for the kernel build. If you do not have this much space, do not choose an automated build.

You have the following amount of free disk space available:

```
df -k /sys
Filesystem      kbytes    used    avail capacity  Mounted on
/dev/disk/dsk3g  825507    670890    72066     90%    /ufs/dsk3g
```

Do you want this procedure to rebuild your kernel? (y/n):

y を入力するとカーネルの構築が開始され、次のようなメッセージが表示されます。

Starting kernel rebuild...

*** KERNEL CONFIGURATION AND BUILD PROCEDURE ***

Saving /sys/conf/SYSTEM9 as /sys/conf/SYSTEM9.bck

Do you want to edit the configuration file? (y/n) [n]: n

*** PERFORMING KERNEL BUILD ***

Working...Fri Aug 24 13:54:25 EST 2001

Working...Fri Aug 24 13:56:25 EST 2001

Working...Fri Aug 24 13:58:25 EST 2001

The new kernel is /sys/SYSTEM9/vmunix

Saving /vmunix as /vmunix.IOS540.3

Copying /sys/SYSTEM9/vmunix to /vmunix

この例の SYSTEM9 は、システム名です。

注意

-s フラグを指定して wwconfig を起動すると、静的にリンクされたカーネルが構築されます。この場合、wwconfig コマンドの出力は、この項で示した内容とは異なります。

有効または無効にしたい端末オプションがある場合には、次のコマンドを使用して、カーネルの再構成および再構築が必要です。

```
# /usr/sbin/wwconfig -a
```

2.7 WLS インストレーションで作成されるバックアップ・ファイル

ワールドワイド・サポート・ソフトウェア・サブセットのインストレーション中に、インストレーション・プロシージャが置き換えるファイルのオリジナルの内容を保存するために、バックアップ・ファイルが作成されます。表 2-1 に、インストレーション・プロシージャが置き換えるファイルをリストします。

バックアップ・ファイルには、.IOS540_sav.* (* は整数) または .IOS540_sav (整数は付かない) という拡張子が付けられます。

表 2-1: WLS インストレーションで作成されるバックアップ・ファイル

拡張子付きで保存される ファイル	ファイル名
.<file>.IOS540_sav	/usr/bin/X11/dxkeycaps

IOS540_sav という拡張子のバックアップ・ファイルは、システム管理者が変更することがないデータ・ファイルまたはバイナリ・ファイルのバックアップです。これらのファイルは、ワールドワイド・サポート・ソフトウェア・サブセットを削除する際にリストアされます。

注意

IOS540_sav という拡張子の付いたファイルは削除しないでください。これらのファイルを削除すると、WLS ソフトウェアを削除する際に、対応するデータ・ファイルおよびバイナリ・ファイルをリストアすることができなくなります。

.IOS540_sav.* という拡張子の付いたファイルは、システムの構成によってはシステム管理者が変更することがあるデータ・ファイルのバックアップです。空きスペースを確保するために .IOS540_sav.* を削除しても構いません。これらのファイルは、サブセットを削除する際にオリジナルをリストアするために使用されることはありません。

2.8 [ステップ 5]: 環境変数の設定

ある言語 (Euro Latin-9 および Unicode を除く) を WLS キットから初めてインストールしたときには、省略時の設定では CDE デスクトップがその言語で開始されます。ただし、初めてのインストレーションで複数の言語を選択した場合は、CDE デスクトップは英語のままです。CDE デスクトップが別の言語に切り替えられている場合は、後で他の言語をインストールしても、省略時の CDE 言語には影響しません。

次のコマンドを使用すると、省略時の CDE 言語を英語に戻すことができます。

```
# rm -f /etc/dt/config/Xconfig
```

複数の言語のサポートをインストールした場合は、LANG 環境変数または LC_ALL 環境変数を定義して、ロケールを設定します。

CDE ログイン・ウィンドウでログインするときに共通デスクトップ環境 (CDE) の言語を設定するには、[オプション]メニューから、[言語] をクリックして、使用したい言語を選択します。

2.9 関連ドキュメント

Tru64 UNIX の国際化機能についての詳細は、次のドキュメントを参照してください。

- 『国際化機能ユーザズ・ガイド』。国際化ソフトウェアの使い方と、多言語環境での作業方法について説明しています。
- 『国際化ソフトウェア・プログラミング・ガイド』。オペレーティング・システムに付属しているツールで国際化ソフトウェアを作成する方法について説明しています。
- リファレンス・ページ `locale(1)`、`i18n_intro(5)`、`l10n_intro(5)` も利用できます。個々の言語のリファレンス・ページ (`spanish(5)`、`italian(5)`、`japanese(5)`、`hebrew(5)` など) も利用できます。

この章では、次の項目について説明します。

- UNIX シェルの説明 (3.1 節)
- フル・インストレーション処理での UNIX シェルの起動方法 (3.2 節)
- UNIX シェル環境から実行できるタスクの種類 (3.3 節)
- 以下の目的のための UNIX シェルの使い方
 - スワップ領域の作成 (3.4 節)
 - ファイル・システムのマウント (3.5 節)
 - テープ・バックアップからの UNIX ファイル・システム (UFS) あるいは AdvFS (Advanced File System) のリストア (3.6 節)
 - 破損したディスクから新しいディスクへのシステム・イメージのリストアと、古いデバイス名の新しいデバイス名への割り当て (3.7 節)
 - `disklabel` コマンドによるディスク・パーティション・サイズの変更 (3.8 節)
- UNIX シェル環境からフル・インストレーション・プロシージャへ戻る方法 (3.9 節)

3.1 概要

UNIX シェル・オプションは、フル・インストレーション中に標準の UNIX コマンドを使用するための手段です。UNIX シェル・オプションの主な目的は、フル・インストレーションの前および最中に、ディスクおよびファイル・システムの保守のための手段を提供することです。UNIX シェルでは、/ (root) ファイル・システムの破損などの重大な問題からの回復を支援し、一般的なファイル・システムおよびディスクの保守作業を実行できるように、すべての UNIX コマンドにアクセスすることができます。一般には、インストレーション中に UNIX シェルにアクセスする必要はありませんが、必要な場合にアクセスが可能になるように、このオプションが用意されています。

ベース・オペレーティング・システムの配布メディア (CD-ROM あるいは RIS) には、システムにソフトウェアをインストールしたときと同じ状態にレイアウトされたファイル・システムが含まれており、直接アクセスできる `/`、`/usr`、および `/var` が含まれています。この配布メディアには、圧縮されたソフトウェア・サブセットおよび圧縮されていないソフトウェア・サブセットが混在して含まれています。このフォーマットにより、オペレーティング・システムがまだ完全に機能しない状態であっても、すべての Tru64 UNIX コマンドおよびユーティリティを使用することができます。マウントされた配布メディアは、実質的には完全なオペレーティング・システム環境と考えることができます。UNIX シェル・オプションは、インストレーション・メディアを障害回復のためのツールとして使用することを可能にしています。

UNIX シェルにおけるシステム管理作業は、UNIX オペレーティング・システムに精通している場合に限り、行うようにしてください。

3.2 UNIX シェルの起動

インストレーション・プロシージャからの UNIX シェルの起動は、グラフィカル・インタフェースまたはテキスト・ベース・インタフェースのいずれを使用しているかによって異なります。UNIX シェル・オプションを起動する際、システムは、スーパーユーザ特権のあるシングルユーザ・モードで Bourne シェルを使用します。シェルおよび特権についての詳細は、『*Tru64 UNIX ユーザーズ・ガイド*』を参照してください。

フル・インストレーションを開始するために配布メディアからシステムをブートすると、グラフィカル表示機能を備えたシステムでは、言語選択メニューで言語を選択した後に「Installation Welcome」ウィンドウが表示されます。UNIX シェルを起動するには、ダイアログ・ボックス上部の [ファイル] メニューから [端末ウィンドウ] メニューを選択します。

グラフィカル表示機能を持たないシステムの場合、テキスト・ベース・インストレーション・インタフェースから UNIX シェルを起動する方法は 2 通りあります。1 つ目の方法は、テキスト・ベース・インタフェースの最初のスクリーンからメニュー・オプション 3 を選択する方法です。2 つ目の方法は、`Ctrl/c` を押す方法です (ただし、ソフトウェア・サブセットのロード中は除く)。

注意

UNIX シェルを起動するとテキスト・ベース・インタフェースで選択したインストレーション項目は失われます。テキスト・ベース・インストレーションを再起動して再び選択操作を行なう手順は以下のとおりです。

```
# cd /  
# restart
```

3.3 UNIX シェルの機能

インストレーション環境における UNIX シェルでは、オペレーティング・システムのほとんどの機能を使用できます。インストレーション環境における UNIX シェルの使用と通常のオペレーティング・システム環境の違いは、UNIX シェルはスワップ・デバイスなしで動作し、メモリ・ファイル・システム (MFS) 内に非常に限られた空きディスク容量があれば動作するという点です。この 2 つの要因により、スワップの作成が必要となるような大量のメモリを必要とするタスクや、大量のディスク容量 (/tmp 領域など) を必要とするタスクは、UNIX シェル環境では障害を起こす可能性が高くなります。

注意

/ (ルート) ファイル・システムは CD-ROM あるいは RIS サーバに置かれており、インストレーション配布メディアがブートされた後、読み取り専用モードでマウントされます。/var, /dev, および /devices のメモリ・ファイル・システム (MFS) は、読み書きモードでマウントされます。/var, /dev, および /devices 内のファイルに対して行われた変更は、インストレーションを halt してコンソール・プロンプト (>>>) に戻った時点で失われ保管されません。

UNIX シェルで実行できる作業の例としては、以下のようなものがあります。このうちの一部の作業について、この章で説明します。

- UNIX ファイル・システム (UFS) に対しては `newfs` コマンドを使用することにより、また、Advanced File Systems (AdvFS) に対しては `mkfdmn` および `mkfset` コマンドを使用することにより、新しいファイル・システムを作成する。

- `restore` コマンド (UFS の場合) または `vrestore` コマンド (AdvFS の場合) を使用して、ファイル・システムをリストアする。
- グラフィック機能のないシステムで、`disklabel` コマンドを使用してディスク・パーティション・テーブルを変更する。

グラフィック機能を備えたシステムでは、「File System Layout」ウィンドウの [パーティションの編集...] ボタンをクリックするか、またはシェル・プロンプトで `/usr/sbin/diskconfig` ユーティリティを起動して、グラフィカル・ディスク構成ユーティリティを使用します。

- `mount` コマンドでディスクおよびファイル・システムをマウントする。
- `fsck` コマンドで UFS ファイル・システムを修正する。

AdvFS ファイル・システムでは `fsck` コマンドは必要ありません。

- `ed` テキスト・エディタでファイルの内容を表示する。

省略時の設定では、`EDITOR` 環境変数が `ed` に設定されています。`vi` テキスト・エディタを使用できるようにするためには、次のようにします。

- `curses` 機能をサポートするコンソール・ファームウェアがインストールされている古いシステムの場合

```
# TERM=vt1001
# export TERM
```

- PC のようなグラフィック機能を使用している新しいシステムの場合

```
# TERM=pccons
# export TERM
```

3.4 スワップ領域の作成

スワップ領域を必要とするような作業を実行する場合は、UNIX シェルでスワッピングをオンにすることができます。UNIX シェル環境でスワップ領域を使用できるようにする場合、上書きしてしまわないように、保護しておきたいデータを含むパーティションを使用しないように注意してください。このような問題を避けるためには、以前スワップ領域として使用していたディスク・パーティションを使用します。

¹ 別のタイプのビデオ端末タイプを使用している場合は、`vt100` をたとえば `vt302` に置き換えます。

UNIX シェルでスワッピングをオンにするには次のようにします。

1. スワップ領域に使用するデバイス (デバイス名およびパーティション) を決めます。保護しておきたいデータの領域を選択しないように注意してください。

注意

以前スワップ領域として使われていたディスク・パーティションは、ディスクラベルに `swap` というラベルがついています。このパーティションは問題なく使用できます。

2. `/dev/disk` ディレクトリに移動します。

```
# cd /dev/disk
```
3. スワップ・デバイスをオンにします。この例では、スワップ・デバイスは `dsk0b` です。

```
# swapon dsk0b
```
4. 次のコマンドで、スワップ領域がオンになっていることを確認します。

```
# swapon -s
```

注意

スワップ領域が有効になった後は、UNIX シェルからインストレーションを再スタートするための `restart` コマンドは使用できません。インストレーションを再スタートするためには、システムを停止させ、配布メディアからブートし直します。

3.5 ファイル・システムのマウント

UNIX シェルは、既存のファイル・システムに関する保守操作を実行するために使用されます。たとえば、`/` ファイル・システム上のカーネル (`vmunix`) が破損したとき、カーネルのバックアップ・コピーがある場合には、`/` ファイル・システムをマウントして、破損したカーネルを置換することができます。

/ ファイル・システムに対してLSM ボリュームを使用している場合は、LSM の開始方法について『インストール・ガイド』を参照してください。

/dev/disk/dsk0a にある既存のルート・ファイル・システムを別のマウント・ポイントにマウントするには、次のコマンドを実行してください。

1. /var にマウント・ポイントを作成します。

```
# mkdir /var/mnt
```

2. ファイル・システムをマウントします。

- a. UNIX ファイル・システム (UFS) の場合は、次のように入力します。

```
# fsck -y /dev/disk/dsk0a
# mount /dev/disk/dsk0a /var/mnt
```

- b. Advanced File Systems (AdvFS) の場合は、次のように入力します。

```
# mkdir -p /etc/fdmns/root_domain
# cd /etc/fdmns/root_domain
# ln -s /dev/disk/dsk0a dsk0a
# mount root_domain#root /var/mnt
```

既存の / ファイル・システムは、/var/mnt でアクセスおよび変更することができます。

3.6 バックアップからのファイル・システムのリストア

UNIX シェルは破損した / ファイル・システムのリストアには最適ですが、/ 以外のファイル・システムのリストアにも使用できます。

ファイル・システムのリストアは、完全なオペレーティング・システム環境から行うことをお勧めします。/var あるいは /usr のリストアが必要なために、そのような環境が使用できない場合は、既存のあるいはリストアした / ファイル・システムを使用して、システムをシングルユーザ・モードにブートします。これにより、インストール環境で利用できる書き込み可能ディスク・スペースよりも多くのスペースを確保することができます。スワップ領域は、3.4 節 に示す手順で UNIX シェルで有効にできます。

3.6.1 項 に UFS ファイル・システムのリストア方法を、3.6.2 項 に AdvFS ファイル・システムのリストア方法を示します。どちらの手順も、同じディスクにファイル・システムをリストアすることを前提としています。

3.6.1 テープ・バックアップからの UNIX ファイル・システム (UFS) のリストア

次の手順に従って、UNIX ファイル・システムを、元と同じディスクにリストアします。リストアを行おうとしているパーティションのサイズは、ダンプ・ファイルのサイズより大きくなければなりません。

注意

以下の手順では、システムが配布メディアから起動されていることと、[ファイル] メニューからインストレーション処理を終了して UNIX シェルへ戻っていることを前提としています。

1. 破損したファイル・システムが置かれているディスクのラベルを読み取り、ディスクのタイプを判断します。

```
# disklabel -r /dev/rdisk/dskna | more
```

ディスク・タイプは「disk:」フィールドの隣に表示されます。ディスク・タイプをここに記録してください: _____

ディスクにディスク・ラベルがない場合は、3.8 節に進んでディスク・ラベルを作成します。

2. 次のコマンド構文を使用して、新しい / (ルート) ファイル・システムを作成します。

```
newfs device_name disk_type
```

device_name パラメータには、影響を受けるディスクの完全なパス名を指定します。 / (ルート) ファイル・システムは、a パーティションになければなりません。たとえば、RZ58 タイプのディスクの a パーティションに新しいファイル・システムを作成するには、次のようなコマンドを入力します。

```
# newfs /dev/disk/dsk0a rz58
```

3. /var にマウント・ポイントを作成して、ファイル・システムをマウントします。

```
# mkdir /var/mnt
```

UNIX シェルから書き込み可能なディレクトリは `/var` または `/tmp` に限られるので、このうちいずれかにマウント・ポイントを作成します。

4. 次のコマンド構文で、破損したファイル・システムをマウントします。

mount *device_name mount_point*

device_name パラメータにはディスク・デバイスの完全なデバイス・パス名を指定します。たとえば、前の手順で作成したファイル・システムをマウントするには、次のコマンドを入力します。

```
# mount /dev/disk/dsk0a /var/mnt
```

5. テープ・デバイスの電源がオンで、システムに接続されていることを確認します。

注意

システムの初期化時にテープ・デバイスの電源がオンでなかった場合、システムからそのデバイスが見えないことがあります。このような場合は、システムをシャットダウンして、テープ・デバイスの電源をオンにし、システムをリブートしてください。

6. テープ・デバイス特殊ファイルが存在するかどうかを調べます。次のコマンドを使用して、システム上のすべてのデバイスについての情報を取得します。

```
# hwmgr -view device
```

このコマンドで `/dev/tape/tape0` のようなデバイス名が出力されなかった場合は、手順 7 の説明に従って、テープ・デバイス特殊ファイルを作成します。テープ・デバイスが存在する場合は、手順 9 に進みます。

7. テープ・デバイス用のデバイス特殊ファイルを作成します。

```
# /sbin/dn_setup -install_tape
```

+tape0 のようなメッセージで、この操作で作成されたデバイス名が表示されます。プラス記号は、デバイスが追加されたことを示します。

8. `hwmgr` コマンドを再度使用して、テープ・デバイスの完全パス名を表示します。このデバイス名は手順 9 で使用します。

```
# hwmgr -view device
```

9. ルート・ファイル・システムをリストアします。 ローカル・ファイル・システムからダンプ・ファイルをリストアしている場合、`/var/mnt` ディレクトリに移動し、ダンプ・ファイルを含むメディアを挿入し、以下の構文で `restore` コマンドを入力します。

```
restore -rf tape_device
```

`tape_device` パラメータには、ダンプ・データを含むテープ・デバイスの完全パス名を指定します。たとえば、次のコマンドを入力します。

```
# cd /var/mnt
# restore -rf /dev/tape/tape_device
```

10. システムをシャットダウンして停止させます。

```
# shutdown -h now
```

11. リストアしたディスクからシステムをリブートします (リストアしたディスクに対応するコンソール・デバイス名が分からない場合は、`show dev` コマンドを使用します)。

```
# show dev
# boot console_device_name
```

3.6.2 テープ・バックアップからの AdvFS のリストア

次の手順に従って、AdvFSファイル・システムを、元と同じディスクにリストアします。

注意

以下の手順では、システムが配布メディアから起動されていることと、インストール処理を終了して UNIX シェルへ戻っていることを前提としています。

1. 破損したファイル・システムが置かれているディスクのディスク・ラベルを読み取り、ディスク・ラベルが有効であることを確認します。

```
# /usr/sbin/disklabel -r /dev/rdisk/dskna | more
```

ディスクにディスク・ラベルがない場合は、3.8 節に進んでディスク・ラベルを作成します。

2. 次のコマンド構文を使用して、新しい / ファイル・ドメインを作成します。

```
mkfdmn -orF device_name domain
```

device_name パラメータには、使用しているシステムにあるディスク・デバイスの完全なデバイス・パス名を指定します。たとえば、*dsk0* の *a* パーティションに新しいファイル・システムを作成するには、次のコマンドを入力します。

```
# mkfdmn -orF /dev/disk/dsk0a root_domain
```

3. *root_domain* ファイルに *root* ファイルセットを作成します。次のコマンドを使用してください。

```
# mkfset domain fileset
```

domain パラメータには、ファイルセットを作成するドメインの名前を指定します。たとえば、*root_domain* ファイル・ドメインに *root* ファイルセットを作成するには、次のコマンドを入力します。

```
# mkfset root_domain root
```

4. */var* または */tmp* にマウント・ポイントを作成し、ファイルセットをマウントする準備をします。

```
# mkdir /var/mnt
```

マウント・ポイントは、UNIX シェルから書き込み可能なディレクトリである */var* または */tmp* ディレクトリの下に作成することをお勧めします。

5. *root* ファイルセットをマウントします。次のコマンド構文を使用してください。

```
mount domain#fileset mount_point
```

domain#fileset パラメータには、*root* ファイル・ドメインおよび *root* ファイルセットを指定します。たとえば、前のステップで作成したファイルセットをマウントするには、次のコマンドを入力します。

```
# mount root_domain#root /var/mnt
```

6. ファイル・ドメインとファイルセットがマウントされていることを確認します。

```
# showfdmn root_domain  
# showfsets root_domain
```


7. テープ・デバイスの電源がオンで、システムに接続されていることを確認します。

注意

システムの初期化時にテープ・デバイスの電源がオンでなかった場合、システムからそのデバイスが見えないことがあります。このような場合は、システムをシャットダウンして、テープ・デバイスの電源をオンにし、システムをリブートしてください。

8. テープ・デバイス特殊ファイルが存在するかどうかを調べます。次のコマンドを使用して、システム上のすべてのデバイスについての情報を取得します。

```
# hwmgr -view device
```

このコマンドで `/dev/tape/tape0_d0` のようなデバイスが出力されなかった場合は、手順 9 の説明に従って、テープ・デバイス特殊ファイルを作成します。このようなデバイスが出力された場合は、手順 11 に進みます。

9. テープ・デバイス用のデバイス特殊ファイルを作成します。

```
# /sbin/dn_setup -install_tape
```

+tape0 のようなメッセージで、この操作で作成されたデバイス名がリストされます。プラス記号は、デバイスが追加されたことを示します。すべてのテープ・デバイス特殊ファイルが作成されたら、手順 11 に進んで、破損した / ファイル・システムの回復作業を続けます。

10. `hwmgr` コマンドを再度使用して、テープ・デバイスの完全パス名を表示します。このデバイス名はステップ 11 で使用します。

```
# hwmgr -view device
```

11. `vrestore` コマンドを使用して、ファイルセットをリストアします。ローカル・ファイル・システムからファイルをリストアするには、`/var/mnt` ディレクトリに変更し、ダンプ・ファイルを格納しているメディアを挿入して、`vrestore` コマンドを次の構文で入力します。

```
vrestore -vxf tape_device_name
```

`tape_device_name` パラメータには、10 で取得した、ダンプ・データが入っているテープ・デバイスのパス名を指定します。

```
# cd /var/mnt
# vrestore -vxf /dev/tape/tape_device_name
```

注意

UFS フォーマットのダンプ・テープを AdvFS にリストアしたり (たとえば、UFS の / ファイル・システムを AdvFS に変換する場合)、UFS 上に `vdump` テープを作成することができます。使用するリストア・コマンドは、テープのフォーマット (`dump` または `vdump`) によって異なります。`vdump` コマンドで行われた AdvFS ダンプのリストアには `vrestore` を使用し、`dump` コマンドで行われた UFS ダンプのリストアには `restore` を使用します。ターゲット・ファイル・システムのタイプにかかわらず、対応するリストア・コマンドを使用します。

12. `/etc/fstab` ファイルおよび `/etc/fdmns` ディレクトリの内容を確認します。UNIX シェルで `mkfdmn` コマンドを使用して / ファイル・システムに追加した `/etc/fdmns/root_domain` は、システムをリブートすると削除されます。

13. システムをシャットダウンして停止させます。

```
# shutdown -h now
```

14. リストアしたディスクからシステムをリブートします (リストアしたディスクに対応するコンソール・デバイス名が分からない場合は、`show dev` コマンドを使用します)。

```
# show dev
# boot console_device_name
```

UNIX シェルを使用して、他のファイル・システムをリストアすることができますが、ファイル・システムのリストアは、できるだけ完全なオペレーティング・システム環境から実行するようにしてください。`/var` または `/usr` のいずれかのリストアが必要なために、そのような環境が利用できない場合は、既存の / ファイル・システムまたはリストアした / ファイ

ル・システムを使用して、システムをシングルユーザ・モードにブートする必要があります。シングルユーザ・モードでは、より多くのディスク・スペースが利用可能になり、次のコマンドを発行すると、スワップ領域を利用可能にすることができます。

```
# swapon -a
```

3.7 破損したディスクから新しいディスクへのルート・ファイル・システム・イメージのリストア

この手順は、保存されているルート・ファイル・システム・イメージを不良または破損したディスクから新しいディスクへリストアし、次に新しいディスクの名前を古いシステム・ディスクの名前に変更する方法について説明します。

新しいシステム・ディスクを導入する処理は、`/etc/fstab` ファイルのアップデートだけでは済みません。これは、インストレーション・カーネルが、保存されているシステム・イメージ上で割り当てられているデバイス名とは異なるデバイス名を、システム上のデバイスに割り当てることがあるためです。この結果、リストアされたシステム・イメージが、イメージのリストアに使用されたカーネルと合わなくなり、両者の間の対応がわからなくなります。

注意

次の手順は、3.6 節の説明に従って、新しいディスクへのルート・ファイルのシステム・イメージをリストアしていることを前提としています。

1. システムを停止します。

```
# shutdown -h now
```

2. システム上のコンソール・ディスク・デバイス名を調べます。通常、コンソール・デバイス名はコマンド出力の中央の欄に表示されます。コンソール・デバイス名は `DKA100`、`DKC0` などのような名前になっています。

```
>>> show dev
```

破損したディスクのコンソール・デバイス名をここに記録してください: _____

新しいディスクのコンソール・デバイス名をここに記録してください: _____

3. リストアしたイメージが格納されている新しいディスクのコンソール・デバイス名 (ステップ 2 で取得した名前) を使用して、システムをシングルユーザ・モードでブートします。

```
>>> boot -fl s console_device_name
```

4. シングルユーザ・モードで、/ ファイル・システムを書き込み可能でマウントします。

```
# /sbin/mountroot
```

このプロセスは、新しく見つかったディスクのデバイス・エントリを作成します。次のようなメッセージで終わる処理メッセージが表示されます。

```
dskNa dskNb dskNc ...
```

上記の例の N は、初めて認識された新しいディスクに割り当てられた番号です。このディスクは、リストアされたディスクです。ただし、複数のディスクが出力された場合は、システム・ディスクの保存を最後に行った後に複数のディスクを追加しています。

新しいディスクの名前をここに記録してください: _____

手順 5 に進んで、損傷のある古いディスクのディスク名を調べます。

5. 古いブート・ディスクの名前を調べます。

```
# /sbin/consvar -g bootdef_dev
```

このコマンドの出力は、次のようになります。

```
bootdef_dev = dskN
```

上記のコマンド出力で、 $dskN$ は `old_disk_name` です。

古いディスクの名前をここに記録してください: _____

6. 古いディスク名と新しいディスク名を交換します。

```
# dsfmgr -e new_disk_name old_disk_name
```

このコマンド行では、`new_disk_name` には 4 で得たディスク名を指定し、`old_disk_name` には手順 5 で古いディスク名を指定します。

注意

破損した古いディスクが削除され、アクティブでない場合は、メッセージが表示されます。この場合は、移動フラグ (-m) を代りに使用します。

```
# dsfmgr -m new_disk_name old_disk_name
```

7. ブート・デバイスを設定し直します。

```
# /sbin/consvar -s bootdef_dev new_disk_name
```

8. 必要に応じて他の保守作業を行うか、作業が完了していれば次のコマンドでシステムをマルチユーザ・モードにします。

```
# init 3
```

3.8 disklabel コマンドによるディスク・パーティション・サイズの変更

テキスト・ベース・インストレーション・インタフェースを使用している場合、disklabel コマンドを使用してドライブ ID あるいはディスク・パーティションを変更し、破損したラベルやブートストラップを置き換えることができます。詳細については、disklabel(8) を参照してください。

既存のディスク・パーティションのレイアウトを確認するには、disklabel コマンドを次のフォーマットで入力します。変数 *n* には、ディスクのユニット番号を指定してください。たとえば、ディスクの既存のディスク・パーティション・レイアウトを確認するには、次のコマンドを入力します。

```
# disklabel -r /dev/rdisk/dskn
```

次の例では、rz261 ディスクを使用します。この例では、b パーティションのサイズを小さくし、a パーティションのサイズを / (ルート) ファイル・システムを保持する最低の大きさである 128 MB まで大きくしています。ディスク・パーティションのサイズ変更は以下の手順で行います。

1. dsk0 のディスクラベルを調べます。

```
# disklabel -r dsk0
```

2. EDITOR 環境変数に ed エディタを設定します。

```
# EDITOR=ed  
# export EDITOR
```

注意

VGA モニタを使用しており、vi エディタを使用したい場合は、まず、次の変数を設定する必要があります。

```
# TERM=vt100
# export TERM
# EDITOR=vi
# export EDITOR
```

この後、例に示すコマンドの代わりに、対応する vi コマンドを使用してディスク・ラベルの変更を行います。以下の例では ed を使用します。

3. dsk0 のディスク・ラベルをテンポラリ・ファイルにコピーします。

```
# disklabel -r dsk0 > /tmp/old_disklabel
```

4. 実際のディスク・ラベルにエラーが含まれるのを防ぐために、ディスク・ラベルのコピーの方を編集します。

```
# ed /tmp/old_disklabel
```

5. ディスク・ラベルを再度表示して、変更を確認します。

```
1,$p
```

6. b パーティションを探します。

```
/b:
```

次のような情報が表示されます。

```
b: 262144 131072 unused 1024 8192 # (Cyl. 164*- 492*)
```

7. b パーティションのサイズを 262144 セクタから 131072 セクタに変更します。

```
s/262144/131072/p
```

これにより、b パーティションの大きさが 128 MB から 64 MB に変更されます。変更された情報が表示されます。

```
b: 131072 131072 unused 1024 8192 # (Cyl. 164*- 402)
```

シリンダ情報を変更する必要はありません。ファイルを保存して終了すると、自動的にシリンダ情報が変更されます。

8. b パーティションのオフセットを 131072 セクタから 262144 セクタへ変更します。

```
s/131073/262144/2p
```

これにより、262144 の位置から始まるよう b パーティションのオフセットが変更されます。

```
b: 131072 262144 unused 1024 8192 # (Cyl. 164*- 402)
```

シリンダ情報を変更する必要はありません。ファイルを保存して終了すると、自動的にシリンダ情報が変更されます。

9. a パーティションを探します。

```
/a:
```

次のような情報が表示されます。

```
a: 131072 0 unused 0 0 # (Cyl. 0 -164*)
```

10. b パーティションのサイズは 131072 セクタ減っているので、a パーティションのサイズは 131072 セクタ増えるはずです。a パーティションのサイズを 131072 セクタから 262144 セクタ変更します。

```
s/131072/262144/p
```

これにより、a パーティションのサイズが 64 MB から 128 MB に増えます。

11. 変更内容を確認するために、ディスク・ラベルを再度表示します。

```
l,$p
```

12. 編集内容を保管しエディタを終了します。

```
wq
```

13. 新しいディスク・ラベルをディスクに適用します。

```
# disklabel -R dsk0 /tmp/old_label
```

14. 新たにカスタマイズされたディスク・ラベルを表示します。

```
# disklabel -r dsk0
```

15. シェルを終了しフル・インストールを再スタートします。

3.9 UNIX シェルからインストール・プロシージャに戻る

UNIX シェルからフル・インストールに戻る手順は、テキスト・ベースのインタフェースを使用しているか、またはグラフィカル・インタフェースを使用しているかによって異なります。

3.9.1 テキスト・ベースのインタフェース

テキスト・ベースのインタフェース、または「Summary」ウィンドウの [Quit] ボタンで UNIX シェルに入った場合、UNIX シェルからフル・インストール・インタフェースを再起動するには、次のコマンドを入力します。

```
# cd /  
# restart
```

グラフィックス機能を持つシステム・コンソールを使用しているとき、グラフィカル・ユーザ・インタフェースの代わりにテキスト・ベースのインタフェースでインストール・プロシージャを再起動する場合は、次のコマンドを入力します。

```
# cd /  
# restart nogui
```

UNIX シェルへ移った時点で、それまでに選択したインストール項目は失われます。フル・インストールを再スタートする場合は、選択をはじめからやり直す必要があります。

3.9.2 グラフィカル・インタフェース

グラフィカル・フル・インストール・インタフェースの [ファイル] メニューから UNIX シェルに入った場合、フル・インストール・ウィンドウをクリックすることにより、いつでもフル・インストールを再開することができます。シェル・ウィンドウを終了する必要はありませんが、シェル・ウィンドウを終了したい場合には、シェル・ウィンドウ内の # プロンプトに `exit` と入力します。

サマリ・ウィンドウの [中止] ボタンで UNIX シェルに入った場合は、3.9.1 項の手順に従ってフル・インストールを再スタートしてください。

ディスク・スペースの計画

この章では、標準の UNIX ファイル・システムについて説明し、手作業でディスク・スペースの計画を行う際に必要となるディスク・スペース計画のガイドラインを示します。フル・インストールを行う場合は、自動ディスク・スペース計画機能が組み込まれているため、手作業でディスク・スペースを計画する必要はまずありません。

この章では、次の項目について説明します。

- ファイル・システムとディスク・スペースの概要 (4.1 節)
- サポートされている 2 つのファイル・システム・タイプ (AdvFS と UFS) についての説明 (4.3 節)
- クラスタのディスク・スペース計画に関する情報の参照先 (4.2 節)
- フル・インストール処理で自動的に行われるディスク・スペース計画 (4.4 節)
- 手作業でディスク・スペースを計画しなければならない状況について (4.5 節)
- ディスク・パーティションやファイル・システム・レイアウトをカスタマイズする場合の考慮事項 (4.6 節)
- 既存のディスクおよびパーティションのサイズの確認方法 (4.7 節)
- オペレーティング・システムをインストールするために必要なディスク・スペースの確認方法 (4.8 節)
- 標準の UNIX ファイル・システムの内容 (4.9 節)
- スワップ領域割り当てのガイドライン (4.12 節)

注意

この章で使用する用語の定義については、『インストール・ガイド』の用語集を参照してください。

4.1 概要

オペレーティング・システムのフル・インストールでは、基本的な UNIX ファイル・システム (`/`、`/usr`、および 1 つのスワップ領域) が作成されます。 `var` 領域は、`/usr` ファイル・システム下のディレクトリとすることも、独立したファイル・システムとして作成することもできます。フル・インストール処理は、これらのファイル・システムをディスク上のどこに置き、オペレーティング・システムをインストールするためにどの程度の大きさにしなければならないかという判断をユーザが簡単にできるようにします。この章では、フル・インストールの組み込みディスク計画機能について説明してから、ユーザのコンピュータ環境でカスタマイズが必要かどうかを判断するための情報について説明します。

4.2 クラスタ環境におけるディスク計画上の考慮事項

この章の情報は、シングル・システム・インストールのディスク計画に適用されます。クラスタ・ファイル・システムのディスク・スペース計画、クォーラム・ディスク、システムをクラスタ・メンバとして構成するための一般的な情報については、TruCluster Server の『クラスタ・インストール・ガイド』および『クラスタ管理ガイド』を参照してください。

4.3 ファイル・システム・タイプの概要

このオペレーティング・システムは、AdvFS (Advanced File System) と、UFS (UNIX ファイル・システム) の、2 種類のファイル・システムをサポートしています。

`/`、`/usr`、`/var`、および `/usr/i18n` ファイル・システムには、ファイル・システム・タイプとして AdvFS または UFS を選択できます。ファイル・システム・レイアウトをカスタマイズする場合は、必ずしもすべてのファイル・システムに同じファイル・システム・タイプを使用する必要はありません。省略時のファイル・システム・レイアウトを使用する場合は、すべてのファイル・システムに対してファイル・システム・タイプを 1 つだけ選択できます。

どちらのファイル・システム・タイプも、LSM (Logical Storage Manager) をインストールして構成することができます。

この節では、ファイル・システム・タイプを選択する上で役立つ情報を説明します。

4.3.1 AdvFS (Advanced File System)

AdvFS は、省略時のファイル・システム・タイプです。クラスタ構成で稼働させる場合は、AdvFS ファイル・システムを選択する必要があります。

AdvFS は、ログ・ベースのファイル・システムであり、柔軟で、互換性があり、データの可用性が高く、高性能で、システム管理が簡単です。AdvFS は 64 ビットのコンピュータ環境を利用し、16 テラバイト (TB) 程度の大きさのファイルやファイルセットを扱えるように設計されています。

AdvFS の構成は、従来の UNIX ファイル・システムの構成とは異なります。AdvFS では、物理記憶レイヤが、ディレクトリ・レイヤとは独立して管理されます。システム管理者は、ファイル・システムをアンマウントしたり、オペレーティング・システムを停止することなく、ストレージを追加したり削除したりできます。このため、構成の計画はそれほど複雑ではなく、柔軟性もあります。

ユーザから見ると、AdvFS は他の UNIX ファイル・システムと同様に動作します。mkdir コマンドを使用して新しいディレクトリを作成したり、cd コマンドを使用してディレクトリ間を移動したり、ls コマンドを使用してディレクトリの内容をリストしたりできます。AdvFS の論理構造、クォータ制御、およびバックアップ機能は、従来のファイル・システム設計を基にしています。

AdvFS システムをオフラインにすることなく、システム管理者はバックアップを実行したり、ファイル・システムを再構成したり、ファイル・システムのチューニングを行ったりできます。エンド・ユーザは、システム管理者の助けを借りることなく、誤って削除してしまった自分のファイルを、事前に定義されたごみ箱ディレクトリやクローン・ファイルセットから取り出すことができます。

AdvFS は、マルチボリューム・ファイル・システムをサポートしています。これにより、ファイル・レベルのストライピング (複数のボリュームヘデータを分散する) が可能となり、I/O を多用するアプリケーションのファイル転送速度を改善できます。ボリューム・レベルのストライピングが可能な LSM (Logical Storage Manager) を、AdvFS 構成に組み込むこともできます。

オペレーティング・システムとは別のライセンスを必要する AdvFS Utilities を使用すると、追加のファイル管理機能や、システム管理を簡単にするためのグラフィカル・ユーザ・インタフェース (GUI) を提供します。AdvFS

の GUI は、共通デスクトップ環境 (CDE) 下で動作し、AdvFS の操作を簡単に実行できる、メニュー、グラフィカル表示、わかりやすいオンライン・ヘルプを備えています。

/ ファイル・システムを除き、AdvFS ファイル・システムのサイズは、いつでも変更することができます (addvol コマンドを使用)。ファイル・システム・サイズの増加または減少は、ユーザからは見えません。

AdvFS をファイル・システム・タイプとして使用し、オプションの AdvFS Utilities (OS とは別のライセンスが必要) をインストールすると、ファイル・システム・スペースの変更が簡単になります。インストレーション後、AdvFS Utilities を使用すると、ディレクトリ構造を変更したり、ユーザの作業を中断することなく、AdvFS ファイル・システムでのボリュームの追加や削除が可能です。AdvFS ファイル・システムに対し、ファイル・システム・スペースを余分に割り当てる必要はありません。

AdvFS についての詳細は、『*AdvFS 管理ガイド*』を参照してください。

4.3.2 UNIX ファイル・システム (UFS)

UFS は、UNIX オペレーティング・システム用の、古くからある従来のファイル・システム・タイプです。AdvFS はログ・ベースのファイル・システムで、UFS よりも強力で安定しており、万一の場合でもデータ破壊が発生しにくくなっています。AdvFS では、1 つのファイル・システムのファイルを、複数のディスクまたはパーティション上に持つことができますが、UFS の階層構造はもっと厳格です。UFS ファイル・システムでは、各ディスクや各パーティションには、独立したファイル・システムが格納されます。つまり、ファイル・システム内のすべてのファイルは、単一のディスク上の、割り当てられたディスク・パーティション内に存在します。UFS ファイル・システムの特長には、階層構造、ファイルの作成および削除機能、ファイルの動的拡張、ファイル・データの保護があります。

UFS は、Berkeley 4.3 Tahoe リリースと互換性があります。UFS では 255 バイトまでのパス名構成要素が許されており、完全修飾パス名の長さは、1023 バイトまで許されています。この実装の UFS では、最大のファイル・システム (128 GB) と同じ大きさの最大ファイル・サイズがサポートされています。

注意

LSM (Logical Storage Manager) は、UFS ファイル・システム・タイプでも使用することができます。クラスタでは、UFS ファイル・システム・タイプは使用できません。

UFS についての詳細は、『システム管理ガイド』を参照してください。

4.4 フル・インストールの自動ディスク・スペース計画機能

フル・インストール・プロセスでは、次のような、事前定義の省略時ファイル・システム・レイアウトを使用できます。

- 基本的な UNIX ファイル・システムとスワップ領域は、選択した 1 台のディスクにインストールされます。このディスクのサイズは、1 ギガバイト (GB) 以上でなければなりません。
- ファイル・システムは、次のとおりにインストールされます。
 - / (ルート) ファイル・システムは、a パーティションにインストールされます。
 - /usr ファイル・システムは g パーティションにインストールされます。var は、/usr 内のディレクトリになります。ファイル・システム・タイプとして AdvFS を選択した場合は、var は usr_domain に置かれます。
 - swap 領域が 1 つ、b パーティションに置かれます。
- すべてのファイル・システムに対するファイル・システム・タイプとして、AdvFS または UFS のどちらか 1 つが選択されます。省略時のファイル・システム・タイプは、AdvFS です。
- 推奨パーティション・テーブルでは、すべてのファイル・システムおよびベース・オペレーティング・システム・ソフトウェア・サブセットに十分な大きさのディスク・パーティションが作成されます。パーティション・サイズは、ディスク・サイズに基づいて表 4-1 のように算出されます。

表 4-1: ディスク容量ごとの推奨パーティション・テーブル

ファイル・システムの名前と位置	ディスク容量ごとのファイル・システム・サイズ		
	1 GB ディスク	2 GB ディスク	3 GB 以上のディスク
/ (ルート) パーティション ^a	128 MB	256 MB	384 MB
/usr パーティション ^g	745 MB	1490 MB	2235 MB
スワップ領域パーティション ^b	128 MB	256 MB ^a	384 MB ^a

^aスワップ領域は、省略時のディスク・パーティション・テーブルの ^b パーティションのサイズより大きくすることができます。

推奨ディスク・パーティション・テーブルを使用する省略時ファイル・システム・レイアウトの設計では、オペレーティング・システム全体が、1 GB 以上のサイズの 1 つのディスクに収まるようになっています。

注意

1 台のディスクにインストールすることは可能ですが、ディスクが複数ある場合は、1 台の 1 GB ディスクへのフル・インストールはお勧めできません。このようにすると、性能が低下するだけでなく、クラッシュ・ダンプを保存できず、ディスクがクラッシュした際のルート・ディスクの回復が難しくなります。

/ (ルート) ファイル・システムを 1 台のディスクに置き、/usr ファイル・システムとスワップ領域を別のディスクに置くことをお勧めします。ユーザ・データはこれ以外のディスクに置くようにしてください。

この省略時のレイアウトで満足でき、システムの構成に特別な考慮が必要でない場合は、これ以上ディスク計画を行う必要はありません。省略時のファイル・システム・レイアウトについての詳細は、図 4-1 を参照してください。この例では、オペレーティング・システムをインストールする単一のディスクとして、`dsk1` を使用しています。

図 4-1: 省略時の「File System Layout : Defaults」ウィンドウ



4.5 節では、ファイル・システム・レイアウトとディスク・パーティション・サイズを手作業で計画しなければならない場合について説明します。

4.5 手作業でディスク・スペースを計画しなければならない場合

省略時の設定や推奨設定を受け入れるのではなく、ディスク・スペースを手作業で計画しなければならない場合もあります。

ディスク・スペースを計画しなければならないのは、次のいずれかの条件に該当する場合です。

- システムをリモート・インストレーション・サービス (RIS) サーバまたはデータレス管理サービス (DMS) サーバとして使用する場合。これらの 2 つのユーティリティは重要なデータを /var ファイル・システムに置くため、/var を独立したパーティション (可能であれば独立したディスク) とすることをお勧めします。RIS ユーティリティと DMS ユーティリティについては、『*Sharing Software on a Local Area Network*』を参照してください。4.5.1 項と 4.5.2 項では、これらの 2 つのユーティリティでのディスク計画の考慮事項を、簡単に説明します。
- / (ルート) ファイル・システム・スペースを大量に必要とする関連プロダクトやレイヤード・プロダクトをインストールしたい場合。『リリース・ノート』に、すべての関連プロダクトおよびレイヤード・プロダク

トのソフトウェア・サブセット・サイズの情報があります。『リリース・ノート』を読んで、追加ソフトウェア・プロダクトが大量のディスク・スペースを必要としないことを確認してください。

- フル・インストール後に、別のインストール作業として、WLS (ワールドワイド言語サポート) ソフトウェア・サブセットをインストールする場合。 / , /usr , /var ファイル・システムのサイズをそれぞれ指定し、 /usr/i18n ディレクトリのサイズを考慮しなければならないことがあります。 ソフトウェア・サブセット名、サイズ、および各言語グループについての説明は、『インストール・ガイド』および『リリース・ノート』を参照してください。 本書の第 2 章では、WLS インストール手順について説明しています。
- ファイル・システムを複数のディスクにインストールする場合。
- スワップ領域を 2 つ作成したい場合 (省略時は 1 つ)。 または、省略時のサイズよりも大きなスワップ領域を作成したい場合。

4.5.1 リモート・インストール・サービス (RIS) に関する考慮事項

RIS は、登録されたクライアント・システムに対して、ネットワーク経由でソフトウェアを提供するサーバをセットアップするために使用されるユーティリティです。 システムを RIS サーバとしてセットアップする場合は、ローカルにマウントされたメディアを RIS 領域のソースとして使用する代りに、ソフトウェア・サブセットを配布メディアから var 領域の /var/adm/ris ディレクトリに展開することができます。 ソフトウェアを展開すると、オペレーティング・システム CD-ROM の内容がディスクにコピーされるため、大量のディスク・スペースが専有されます。 複数のバージョンのオペレーティング・システムを提供しなければならない場合は、さらに多くのソフトウェア・サブセットを展開して格納するためのストレージが必要になります。 ベース・オペレーティング・システムを展開する RIS 領域は、1 つあたり約 600 MB のディスク・スペースが必要です。 各レイヤード・プロダクトに必要なディスク・スペースは、『リリース・ノート』に記載されています。

RIS を使用する際のオプションとして、シンボリック・リンクされた RIS 領域を作成する方法もあります。 このタイプの領域には、CD-ROM の内容は全くコピーされず、RIS 領域にリンクされます。 シンボリック・リンクされた領域の唯一の欠点は、配布メディアをセットしたデバイスが、この目的のために占有されてしまうことです。 ガイドラインとして、シンボリック・リン

クされた RIS 領域はネットワーク・ブート可能なカーネルとその他のサポート・ファイル用に 10 MB 程度のスペースを専有すると想定してください。

各 RIS 領域に格納するソフトウェア用に十分なスペースを、var ファイル・システム内の /var/adm/ris ディレクトリに確保しなければなりません。確保するスペースの量は、作成する RIS 環境の数により異なります。各ソフトウェア・サブセットの説明や他のサブセットの名前、または RIS 操作に関連するカーネル構成ファイル・オプションについては、『インストール・ガイド』を参照してください。

4.5.2 データレス管理サービス (DMS) に関する考慮事項

データレス管理サービス (DMS) 環境では、サーバ・システムは、すべてのクライアント・システムの / ファイル・システムおよび /usr ファイル・システムを管理します。サーバは、クライアントごとに / ファイル・システムを 1 コピー管理します。/usr ファイル・システムは読み取り専用でエクスポートされ、その環境に登録されているすべてのクライアントで共有されます。各クライアント・システムは、自分専用の /var ファイル・システムを持ちます。

各 DMS 環境は、/var/adm/dms/dmsn.alpha ディレクトリに置かれます。サイトのサーバとクライアントの関係に応じて、n.alpha 領域をいくつか持つことができます。各領域には、オプション・ソフトウェア・サブセットの他、インストールされるベース・オペレーティング・システムの必須サブセットが少なくとも必要です。関連プロダクトまたはレイヤード・プロダクト用のスペースと、システム管理作業およびシステム・ファイル情報用に 10 % の追加スペースを確保しなければなりません。データレス環境のサイズ要件についての詳細は、『*Sharing Software on a Local Area Network*』を参照してください。

データレス環境を提供するシステムにする場合は、/var を独立したファイル・システムにするかどうか、/var/adm/dms ディレクトリ下にマウントするパーティションを確保するかどうかを決定しなければなりません。

また、各データレス・クライアントの / ファイル・システムを格納する /clients ディレクトリのスペース要件も考慮しなければなりません。/clients ディレクトリは、独立したファイル・システムにすることも、サーバの / ファイル・システム内に保持することもできます。この場合は、1 クライアントにつき 64 MB のスペースを割り当てなければなりません。

4.6 ディスク・パーティションおよびファイル・システム・レイアウトのカスタマイズに関する考慮事項

以降の節では、自分でディスク・パーティション・サイズを決めたい場合や、独自のファイル・システム・レイアウトを定義したい場合に必要になる情報について説明します。次のトピックについて説明します。

- ファイル・システム・タイプとして AdvFS と UFS のどちらを選択すべきか (4.3 節)
- 既存のディスクのサイズおよびパーティションを確認する方法 (4.7 節)
- ソフトウェア自身に必要なディスク容量を確認する方法 (4.8 節)
- 計画する上で考慮すべき、各ファイル・システムの基本的な内容 (4.9 節)

4.7 既存のディスクおよびパーティションのサイズの確認

以前のバージョンのオペレーティング・システムがすでに稼働しているシステムには、カスタマイズされたディスク・パーティション・テーブルがすでに存在する可能性があります。ディスク・レイアウトおよびパーティション・サイズをチェックするには、既存のディスク・ラベルを調べます。ディスク・ラベルには、ディスク・タイプ、物理パラメータ、パーティション・サイズなどの、ディスクに関する情報が含まれています。ディスク・ラベルなしでは、ディスクはブートできません。

テキスト・ベースのフル・インストレーション・インタフェースからディスク・ラベルを参照するには、`disklabel` コマンドを使用します (第 3 章を参照)。詳細については、`disklabel(8)` を参照してください。

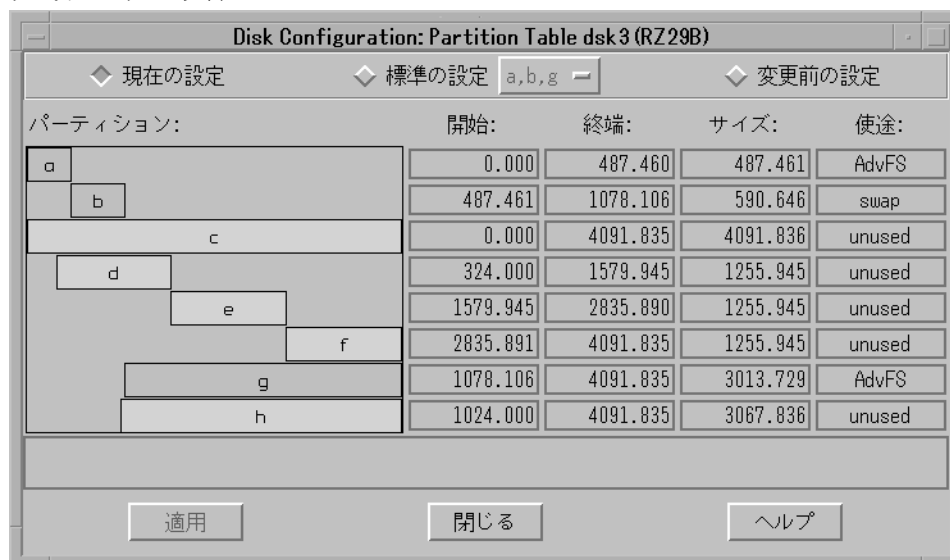
フル・インストレーションをグラフィカル・インタフェースで開始した場合は、「Custom File System Layout」ダイアログ・ボックスの [パーティションの編集...] ボタンをクリックし「Disk Configuration」アプリケーションをオープンして、現在のディスク・パーティション情報を参照します。図 4-2 に、「Disk Configuration」アプリケーションのディスク・パーティション画面の例を示します。

図 4-2: 「Disk Configuration」アプリケーション: ディスク・パーティション情報



図 4-3 に、「Disk Configuration」アプリケーションのディスク・テーブル画面の例を示します。

図 4-3: 「Disk Configuration」アプリケーション: ディスク・パーティション・テーブル



4.8 ソフトウェアが必要とするディスク容量

ソフトウェアがどれくらいのディスク・スペースを専有するかは、知っていないければならない重要な事項です。フル・インストール中は、次のいずれかの方法で、`/`、`/usr`、および `/var` ファイル・システムに必要なディスク容量を知ることができます。

- グラフィカル・ユーザ・インタフェースを使用している場合は、「Software Selection」ウィンドウで、インストールするソフトウェアのタイプ (必須、オプション、またはすべて) を決定し、対応する [サブセット一覧...] または [リストの編集...] ボタンをクリックします。画面の下部に、選択したソフトウェアに基づいて、必要なディスク容量が表示されます。この数値を参照して、ファイル・システムに必要な大きさを判断してください。WLS ソフトウェアをインストールする場合は、インストールする言語をこの時点で選択してください。これは、言語ソフトウェア・サブセットのサイズも、ディスク・スペースの必要量として計算されるためです。
- テキスト・ベース・インタフェースを使用している場合は、オプションのソフトウェア・サブセットを選択すると、フル・インストール処理が、使用されるディスク・スペースを計算します。アメリカ合衆国

の英語以外の言語を選択した場合は、ディスク・スペースの計算に、言語ソフトウェア・サブセットも含められます。この数値を参照して、ファイル・システムに必要な大きさを判断してください。

この情報は、選択したディスク・パーティションが、インストールするソフトウェア・サブセットを十分格納できる大きさかどうかを判断するのに役立ちます。パーティションのスペースが不足している場合、いつでもディスクおよびパーティションの選択を変更することができます。4.8.1 項で説明するように、ファイル・システムのオーバヘッドは、ファイル・システム・スペースの計算に含まれます。

注意

フル・インストレーションの際に、/ ファイル・システムに選択されたディスク上のパーティション・テーブルが推奨テーブルと異なる場合は、推奨ディスク・パーティションを使用するか、カスタマイズされたパーティションを使用するかを選択できます。

この他、ソフトウェア・サブセットのサイズを手作業で調べる方法には、『リリース・ノート』に記載されているソフトウェア・サブセット・サイズの表を参照する方法があります。

4.8.1 ファイル・システムのオーバヘッド

/, /usr, および /var ファイル・システムで利用可能なディスク・スペースを計算する際に、フル・インストレーション・プロシージャは、各ファイル・システムに対して選択されたファイル・システム・タイプに基づき、ファイル・システム・オーバヘッドの概算値を使用します。フル・インストレーション中、ソフトウェア・サブセットを選択した際に計算されるディスク・スペースの量には、これらのオーバヘッドに必要な容量が含まれています。

4.9 ファイル・システムの内容

以降の節では、基本的な UNIX ファイル・システムの内容と、そのファイル・システムのスペースに関する特記事項について説明します。

- 4.10 節では、/usr ファイル・システムの内容と、フル・インストレーション後にオプション・ソフトウェアをインストールするために必要な

スペース、およびユーザ・アカウントやユーザ作成ファイルに十分なスペースを割り当てるために必要なスペースについて説明します。

- 4.11 節では、`/var` ファイル・システムの内容と、クラッシュ・ダンプを格納するためのスペースや、エラー・ロガーや `syslog` ファイルのスペースについて説明します。
- 4.12 節では、オペレーティング・システムでのスワップ領域の実装方法と、スワップ領域に十分なスペースを割り当てるためのガイドラインについて説明します。

4.10 `/usr` ファイル・システムの内容

`/usr` ディレクトリには、ライブラリ、実行可能プログラム、およびドキュメントなど、オペレーティング・システムのファイルのほとんどが格納されます。このディレクトリ構造には、`/usr/sys`、`/usr/adm`、`/usr/bin` などのディレクトリが含まれます。これらのディレクトリには、`/usr` ファイル・システム中でかなりのスペースを占めるシステム・ファイルや UNIX コマンドのバイナリ・ファイルが置かれています。一般的には、専用のディスク上に可能な限り大きな `/usr` ファイル・システムを作成します。

`/usr` ファイル・システムの大きさを決定する際には、次の点を考慮します。

- インストールするソフトウェア・サブセットの数。大半のソフトウェア・サブセットは、`/usr` ファイル・システムに多数のシステム・ファイルやコマンドをロードします。各ソフトウェア・サブセットが各ファイル・システム内で消費する容量については、『リリース・ノート』を参照してください。
- ユーザ・アカウント数および、ユーザのホーム・ディレクトリが `/usr` にある場合には、各ユーザが必要とするディスク容量

注意

将来、フル・インストールを行う際にユーザ情報が上書きされないように、ユーザ・アカウント用に独立したファイル・システムをマウントすることをお勧めします。独立したファイル・システムをマウントすると、ユーザ・アカウントを失うことなくオペレーティング・システムを再インストールできるため、障害回復も簡単になります。

- /var 領域のサイズ (/usr と同じパーティション上にある場合)
- /usr/i18n のサイズ (/usr ファイル・システム内にある場合)

/usr ファイル・システムには、いくつかのファイルが追加され、オペレーティング・システムの将来のリリースでは必要なスペースが増加することがあるため、このファイル・システムのスペースは不足しがちです。将来に備えて、/usr ファイル・システムには十分な空き容量を確保してください。一般的には、/usr ファイル・システムはできるだけ大きくします。

ファイル・システム・タイプとして AdvFS (Advanced File System) を使用し、AdvFS Utilities (オペレーティング・システムとは別のライセンスが必要) をインストールする場合は、/usr ファイル・システムのスペースを過度に多く割り当てる必要はありません。AdvFS ファイル・システムでは、ディレクトリ構造の変更やシステムの中断を行うことなく、動的にスペースを増やすことができます。AdvFS ファイル・システムについての詳細は、『AdvFS 管理ガイド』を参照してください。

4.10.1 項と 4.10.2 項では、さまざまな事項が /usr ファイル・システムのサイズに及ぼす影響、およびディスク・スペースの計画時に考慮する必要がある事項について説明します。

4.10.1 オプションのソフトウェア・サブセットおよび関連プロダクトのためのスペース

/usr ファイル・システムには、インストール予定のソフトウェア・サブセットを格納するのに十分な大きさが必要です。ソフトウェア・サブセットは、特定の機能を実行したり、特定クラスのサービスを提供するために必要な実行可能ファイルとデータ・ファイルの集合です。たとえば、システム課金を実行するには、System Accounting Utilities ソフトウェア・サブセットが必要です。

『インストレーション・ガイド』では、各ソフトウェア・サブセットとともに、依存ソフトウェア・サブセットとソフトウェア・サブセットに関連するカーネル構成ファイル・オプションについて説明しています。ソフトウェア・サブセットのサイズの一覧表については、『リリース・ノート』を参照してください。

必須ソフトウェア・サブセットは必ずインストールされます。オプション・ソフトウェア・サブセットはオペレーティング・システムが機能するために必須のサブセットではありません。このため、ユーザの必要性和ディスク・

スペースに応じて、オプションのソフトウェア・サブセットを全く選択しないことも、一部だけを選択することも、すべてを選択することもできます。

WLS インストールでは、`/usr/i18n` が独立したファイル・システムとして作成されていない場合、`/usr/i18n` ディレクトリのサイズについて考慮しなければなりません。

また、このバージョンのオペレーティング・システムと互換性のある関連プロダクトやレイヤード・プロダクトを将来インストールするためのスペースを割り当てておくこともできます。将来プロダクトを追加する場合は、`/usr` のディスク・スペースの必要容量の計画時に、余分なスペースを確保してください。ソフトウェアの正確なサイズについては、それぞれのプロダクトの『リリース・ノート』を参照してください。

4.10.2 ユーザ・アカウントおよびユーザ・ファイルのためのスペース

フル・インストールでは、ユーザ・アカウントおよびユーザ・ファイルのための領域は用意されません。このため、インストール後にこの領域をセットアップしなければなりません。ただし、システムの計画時にユーザ・ファイル用に必要となるスペースを考慮にいれなければなりません。ユーザのホーム・ディレクトリを `/usr` に置く場合は、システムの各ユーザに対して、最低でも 10 ~ 20 MB のディスク・スペースを確保しなければなりません。ユーザのホーム・ディレクトリ用に必要なサイズを概算するには、この値にユーザ数を掛けてください。

注意

ユーザのホーム・ディレクトリを別のファイル・システム (可能であれば別のディスク) に作成し、`/usr` ファイル・システムの下にマウントすることをお勧めします。ユーザのホーム・ディレクトリを別のファイル・システムに置くことにより、将来、フル・インストールを実行する場合に、そのディレクトリが上書きされるのを防ぐことができます。別のファイル・システムにすると、ユーザ・アカウントを失うことなくオペレーティング・システムを再インストールできるため、障害回復も簡単になります。

ユーザ領域にクォータを設定する場合は、ユーザのクォータにユーザ数を掛けて、ユーザ・スペースの容量を決定してください。ディスク・クォータについては、『システム管理ガイド』を参照してください。

4.11 var ファイル・システムの内容

/var 領域には、一時的なマシン固有のディレクトリ、および tmp や adm などのディレクトリが置かれます。

/var 領域は、独立したパーティションのファイル・システムとして割り当てることも、/usr ファイル・システム下のディレクトリとして割り当てることもできます。/var が頻繁に使用されるシステムの場合は、独立した /var ファイル・システムを作成することをお勧めします。

/var 領域のサイズを決定するときには、次の容量上の要件を考慮してください。

- クラッシュ・ダンプ (4.11.1 項)
- エラー・ロガー・ファイル (4.11.2 項)
- システム課金ファイル (4.11.3 項)
- システムをリモート・インストレーション・サービス (RIS) サーバとして使用する場合、/var/adm/ris ディレクトリのサイズ (4.5.1 項)
- システムをデータレス管理サービス (DMS) サーバとして使用する場合、/var/adm/dms ディレクトリのサイズ (4.5.2 項)
- メール、印刷、および uucp のスプーリング

/var のファイル・システム・タイプとして AdvFS を使用し、AdvFS Utilities (別のライセンスが必要) を使用する場合は、/var ファイル・システムのスペースを過度に多く割り当てる必要はありません。AdvFS ファイル・システムでは、ディレクトリ構造の変更やシステムの中断を行うことなく、動的にスペースを増やすことができます。AdvFS についての詳細は、『AdvFS 管理ガイド』を参照してください。

4.11.1 var ファイル・システム内のクラッシュ・ダンプ領域

システムがクラッシュ・ダンプを生成するときに、2 つのディスク領域が使用されます。4.12 節で説明するように、1 番目の領域は swap パーティションに置かれ、システムがリブートされるまでの間、クラッシュ・ダンプの保

管に使用されます。この領域は、クラッシュ・ダンプを1つ格納するのに十分な大きさでなければなりません。

2番目の領域は、システムがリブートされたときに、`savecore`ユーティリティがクラッシュ・ダンプとカーネル `/vmunix` をコピーする場所です。この領域は、`/var/adm/crash` ディレクトリに置かれます。`/var/adm/crash` が置かれるディスク・パーティションは、少なくとも1つのクラッシュ・ダンプと1つの `/vmunix` のコピーを格納できる十分な大きさ (10 ~ 13 MB) でなければなりません。クラッシュ・ダンプを複数保管したい場合は、ディスク・リソースが許す範囲で、このパーティションを大きくすることができます。

クラッシュ・ダンプ・パーティションのサイズは、フル・ダンプとして構成されたシステムでは物理メモリのサイズと同じで、部分ダンプとして構成されたシステムでは少し小さくなります。

複数のクラッシュ・ダンプを保管したい場合は、1つのクラッシュ・ダンプと1つの `/vmunix` のコピーに必要な合計サイズに n (n は保管しておくクラッシュ・ダンプの数) を掛けて、このパーティションに必要なサイズを見積もります。

『システム管理ガイド』には、クラッシュ・ダンプおよびクラッシュ・ダンプ・ファイルの管理について説明している章があります。この章では、クラッシュ・ダンプがどのように書き込まれるか、部分ダンプあるいはフル・ダンプの選択、クラッシュ・ダンプおよびクラッシュ・ダンプ・ファイルの両方を保管するのに必要な容量の決定などについて説明しています。

クラッシュ・ダンプ領域のサイズを決定し、位置を記録するには、次の情報が必要です。

1. システムのメモリ容量 (MB 単位) は _____ です。

システムのメモリ容量は、次の方法で調べます。

- スーパユーザ (root) として、次のコマンドを入力します。

```
# uerf | grep -i memory
```

_____ 注意 _____

`uerf` コマンドは、新しい EV6 プロセッサではサポートされません。代わりに、Compaq Analyze または

DECevent を使用してください。これらのユーティリティは、Associated Products CD-ROM からインストールできます。詳細は、dia(8) を参照してください。

- コンソール・モード・プロンプト (>>>) が表示されている場合は、次のコマンドを入力します。

```
>>> show mem
```

2. クラッシュ・ダンプ・パーティションのために _____ メモリが必要です。

4.11.2 エラー・ロガーおよび syslog ファイルのスペース

/var 領域には syslog およびバイナリ・エラー・ロガーの両方で作成されるログ・ファイルを格納できるだけのスペースが必要です。これらのログ・ファイルには、システム・イベントとエラーが ASCII テキスト (syslog) とバイナリ・フォーマットで記録されます。

syslog ユーティリティは、次のようなシステム・アクティビティに関する情報を収集します。

- メール
- システム・スタートアップ
- システム・シャットダウン
- リブート
- ルート・アカウントでのログイン
- タイム・デーモン
- プリンタ・サブシステム
- syslog 自身

ハードウェア・エラーに関する要約情報も記録されます。記録されるデータ量は、システム・アクティビティとユーザ数に関係します。

バイナリ・エラー・ロガーは、ハードウェア・エラーおよびシステム・スタートアップに関する情報を記録します。

新しいシステムを作成する場合には、1 週間につき約 500 KB を見積もっておいてください。 /var/adm/binary.errlog ファイルおよび

/var/adm/syslog ファイルの大きさには制限がないため、これらのファイルが大きくなって、パーティションの空きスペースを使い果たしてしまうことがあります。

4.11.3 システム課金ファイルのスペース

/var/adm ディレクトリには、acct や wtmp などの管理プログラムで生成されるデータ・ファイルが格納されます。これらのプログラムが生成するデータは、システムによっても、時期によっても大きく異なります。たとえば、/var/adm/acct ファイルを作成する場合、大規模なシステムではこのファイルのサイズは 1 日に 50 KB 増加することがありますが、ワークステーションでは 5 KB 程度です。

システム課金の一般的なガイドラインとして、ワークステーションでは 1 日に 10 KB を割り当て、大規模なシステムでは 1 日に 100 KB を割り当てるようにします。システム課金に必要なスペースについての詳細は、『システム管理ガイド』を参照してください。

4.12 スワップ領域の概要

オペレーティング・システムの仮想メモリは、物理メモリとスワップ領域の間でメモリ・ページを透過的に移動させることにより実現されます。作成可能な仮想アドレス空間の大きさは、スワップ領域の容量によってのみ制限されます。この節では、システムにスワップ領域を構成する際に考慮しなければならない要因について説明します。スワップ領域使用の最適化についての詳細は、『システムの構成とチューニング』を参照してください。

フル・インストールの際には、swap1 という名前の一次スワップ・パーティションと、swap2 という名前のオプション・スワップ・パーティションを構成することができます。インストールの終了後に『システム管理ガイド』で説明されている手順に従って、追加のスワップ・パーティションを構成することができます。

フル・インストールでは、swap1 に使用するディスク・パーティションの選択についての問い合わせがあります。省略時の選択は、システム・ディスクのパーティション b です。

注意

スワップ領域は、物理メモリの 2 倍のサイズで作成することをお勧めします。

スワップ領域の使用を最適化するには、スワップ領域を複数のデバイスに分散するとともに、高速のディスクをスワップ・デバイスに使用します。最大限の性能を得るには、同じディスク上に複数のスワップ領域を置くのではなく、異なるディスクにスワップ領域を置きます。スワップ領域として割り当てる容量は、インストールするアプリケーションに必要な仮想メモリによっても異なります。

インストレーション中にはスワップ・ストラテジ・モードを選択できませんが、スワップ割り当てには immediate と over-commitment の 2 つのストラテジがあります。省略時の設定では、オペレーティング・システムのスワップ・ストラテジ・モードは、変更可能な仮想アドレス空間が作成されたときにスワップ領域を割り当てる immediate モードです。このモードでは、変更可能なすべての仮想ページが変更されたとしても十分なスワップ領域があることを保証するため、over-commitment モードよりも多くのスワップ領域を必要とします。スワップ割り当てストラテジや、インストレーション後にスワップ割り当てモードを切り替える方法についての詳細は、『システム管理ガイド』を参照してください。

省略時の設定では、クラッシュ・ダンプは一時的にスワップ・パーティションに保存されます。この領域は、クラッシュ・ダンプをシステム・リブート時まで保存しておくために使用され、クラッシュ・ダンプを 1 つ保持するのに十分なサイズが必要です。この領域は、クラッシュ・ダンプ・パーティションと呼ばれます。システム・クラッシュが発生すると、カーネルは物理メモリの内容を、スワップ・パーティションに書き込みます。書き込まれる情報の量とクラッシュ・ダンプのサイズは、いくつかの要因に依存します。

- 『システム管理ガイド』で説明しているように、フル・ダンプを生成するようにシステムが構成されている場合は、クラッシュ・ダンプのサイズはシステムの物理メモリのサイズと同じです。
- 部分ダンプを生成するようにシステムが構成されている場合は、クラッシュ・ダンプのサイズはフル・ダンプよりも小さくなります。

部分クラッシュ・ダンプのサイズを決定する要因は、クラッシュ発生時に、システムの状態を定義するいくつかのカーネル・データ構造体で使用されていた物理メモリの容量です。アクティブなタスクやスレッドが多いほど使用中のカーネルデータ構造体が多くなり、生成される部分クラッシュ・ダンプが大きくなります。

スワップ領域が残り少ないことを示す警告メッセージが表示された場合は、後でスワップ領域を追加してください。

インストレーション処理のカスタマイズ

この章では、インストレーション処理を自動化したりカスタマイズしたりするための、高度な機能について説明します。この章で述べる項目は、次のとおりです。

- インストレーションやシステム構成処理の自動化とカスタマイズのために使用できる機能の概要 (5.1 節)
- 構成記述ファイル (CDF) の紹介と、システムのクローンを作成するために構成記述ファイルを使用する方法 (5.2 節)
- フル・インストレーションまたはアップデート・インストレーションの実行中に、ターゲット・システム上でカスタマイズを行うためにユーザ提供ファイルを起動するタイミング (5.3 節)
- ユーザ提供ファイルと CDF の関係 (5.4 節)
- ユーザ提供ファイルと CDF をセットアップするために管理者が行う高度な作業 (5.5 節)
- インストレーション・プロセスがユーザ提供ファイルと CDF を呼び出す方法の背景にある動作の原理 (5.6 節)
- フル・インストレーションおよびアップデート・インストレーションで使用するための、ユーザ提供ファイルと CDF の作成および配置方法 (5.7 節)

5.1 概要

フル・インストレーションおよびアップデート・インストレーション・プロセスには、あらかじめ定義された時点で特定のファイルを検索する機能が組み込まれています。これらのファイルを正しいディレクトリ位置にコピーすると、次のような機能が実現できます。

- インストレーションのクローニング

`install.cdf` ファイルには、インストール済みのシステムのインストレーション特性が保存されています。このファイルは 1 つまたは複

数のターゲット・システムへインストールを複製するために使用されます。

- 構成のクローニング

`config.cdf` ファイルには、構成済みのシステムの構成特性が保存されています。このファイルは、1 つまたは複数のターゲット・システムへ構成を複製するために使用されます。構成は、ネットワーク、メール・システム、インターネット・アクセス、およびプリンタを設定して、システムが他のシステムおよびユーザと通信できるようにするための処理です。

- ユーザ提供ファイルによるその他のカスタマイズ

ユーザ提供ファイルを作成して、フル・インストールまたはアップデート・インストールの際に、ターゲット・システム上で特定の時点に特定の作業を実行することができます。このファイルには、`preinstall`、`update_preinstall`、`postload`、`update_postload`、および `postreboot` という名前をつける必要があります。

この章では、構成記述ファイルの概念について簡単に説明します。また、インストールの際に、カスタマイズした作業を実行するためのファイルの作成について説明します。

5.2 構成記述ファイルの概要

構成記述ファイル (CDF) は、システムを複製するために使用されます。インストールして構成されたシステムのインストール特性と構成特性は、2 つの CDF ファイルに保存されます。インストール情報は `install.cdf` ファイルに保存され、構成特性は `config.cdf` ファイルに保存されます。この 2 つのファイルについて、以下に簡単に説明します。

- `install.cdf` ファイルは、フル・インストール処理によって、オペレーティング・システムの現行バージョンをシステムにインストールした場合に生成されます。このファイルは `/var/adm/smlogs` ディレクトリに置かれます。このファイルにはファイル・システム・レイアウト、ホストおよびサイトに特有の情報、フル・インストールの際にインストールされたソフトウェアの記録が保存されます。このファイル内の情報は、ハードウェア構成が類似した他のシステムに、同じインストールを複製するために使用されます。クローン・

インストールには多くのオプションがあり，広範囲にわたる内容なので，手順については第 6 章で説明します。 5.6 節で述べる動作原理では，フル・インストールの処理のどこで `install.cdf` ファイルが呼び出されるかを説明しています。

- `config.cdf` ファイルには，完全にインストールされ構成されたシステムからの，ネットワーク，インターネット，プリンタ，およびメールの構成情報が収められています。 `config.cdf` ファイルは，構成情報を保存したいと思った時点で `sysman -clone -save` コマンドを使用して手動で作成します。 `config.cdf` ファイルは，フル・インストールの際にターゲット・システムに適用することも，運用中のシステムに手動で適用することもできます。 構成のクローニングには多くのオプションがあり，広範囲にわたる内容なので，手順については第 7 章で説明します。 5.6 節で述べる動作原理では，フル・インストール処理のどこで `config.cdf` ファイルが呼び出されるかを説明しています。

表 5-1 には，フル・インストール・プロセスが CDF を検索するタイミングと検索するファイル名を示します。

表 5-1: 構成記述ファイルを読み出すタイミング

フル・インストールでの呼び出しポイント	フル・インストール・プロセスが検索するファイル名	見つかった場合の動作
インストールのユーザ・インタフェースが表示される前	<code>install.cdf</code>	<code>install.cdf</code> ファイルに格納されている情報を使って，同じインストールをターゲット・システムに複製する。
システムがリブートされてから，カーネルが構築されるまでの間	<code>config.cdf</code>	<code>config.cdf</code> ファイルに格納されている情報を使って，ターゲット・システムを構成する。

5.4 節で説明しているように，CDF をユーザ提供ファイルとともに使用して，ターゲット・システム上で追加の作業を実行することもできます。

5.3 ユーザ提供ファイルの概要

ユーザ提供ファイルは，スクリプト，実行可能ファイル，またはプログラムを実行してインストール処理を拡張したりカスタマイズしたりするために使用します。表 5-2 には，インストール処理での呼び出しポイント，検索するファイル名，それらを検索するインストールのタイプを

示します。フル・インストールおよびアップデート・インストール・プロセスは、これらのファイルを検索し、見つかると実行します。`postreboot` ファイルを除き、ユーザ提供ファイルが実行されてゼロ以外のステータス (障害を示す) を返した場合、インストールは停止します。

表 5-2: ユーザ提供ファイル呼び出すタイミング

呼び出しポイント	インストール・プロセスが検索するファイル名	どのタイプのインストールで検索されるか
フル・インストールのユーザ・インタフェースが表示される前	<code>preinstall</code>	フルおよびクローン
アップデート・インストールのユーザ・インタフェースが表示される前	<code>update_preinstall</code>	アップデート
ソフトウェアがインストールされてから、汎用カーネルの最初のリブートまでの間	<code>postload</code>	フルおよびクローン
ソフトウェアがアップデートされてから、システムがリブートされるまでの間	<code>update_postload</code>	アップデート
最初のリブートから、カーネルの構築までの間	<code>postreboot</code>	フルおよびクローン

表 5-3 に、ユーザ提供ファイルの一般的な使用目的を示します。

表 5-3: ユーザ提供ファイルの一般的な使用目的

ファイル名	使用目的
<code>preinstall</code>	カスタマイズしたディスク・ラベルを定義し、フル・インストールの際に定義しなくても良いようにする。
<code>postload</code>	<code>config.cdf</code> ファイル内のホスト固有の情報を動的に修正して、クローニング処理が完了すると同時に、ターゲット・システムがネットワーク上で一意に定義されるようにする。この機能を実行するスクリプトの例を、付録 B に記載しています。
<code>postreboot</code>	オプションのソフトウェアを追加インストールして、ソフトウェアの選択処理を簡略化する。

表 5-3: ユーザ提供ファイルの一般的な使用目的 (続き)

ファイル名	使用目的
update_preinstall	実際のアップデート処理が開始される前に、オペレーティング・システムのバックアップを自動的に実行する。
update_postload	アップデート・インストレーションの継続を妨げないように削除したレイヤード・プロダクトを、再インストールする。

この章ではこの後、CDF とユーザ提供ファイルの関係、インストレーション・プロセスがユーザ提供ファイルを検索する方法、ファイルの作成方法と配置方法について説明します。

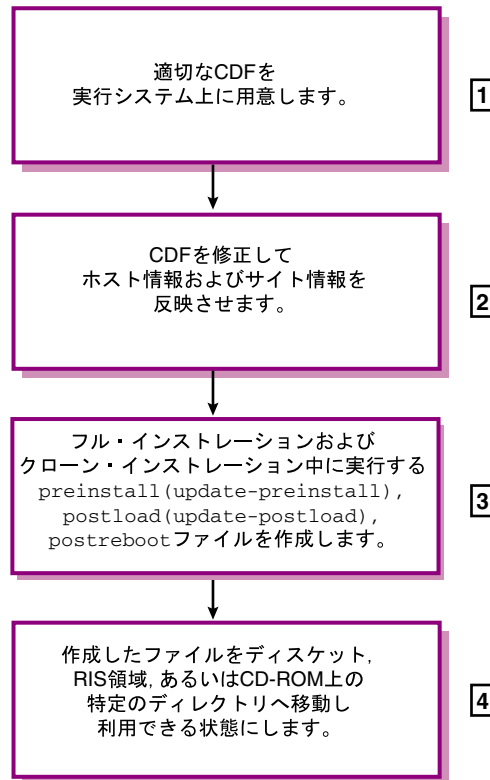
5.4 CDF とユーザ提供ファイルの関係

CDF とユーザ提供ファイルは、単独で使用することも併用することもできます。CDF とユーザ提供ファイルは、異なるソース上に置いてもかまいません。たとえば、`install.cdf` と `config.cdf` のどちらか、または両方の CDF をディスクセットにいれ、`preinstall` ファイルを RIS サーバに置き、`postload` ファイルを RIS サーバの `/isl` ディレクトリに置くことができます。ただし、`postload` ファイルで `config.cdf` ファイルを操作する場合は、両方のファイルを同じ位置に置いてください。ユーザ提供ファイルでは、CDF を動的に変更することができます。

5.5 管理者の作業の概要

図 5-1 では、CDF とユーザ提供ファイルをセットアップするために必要な、システム管理者の作業を示します。フル・インストレーションまたはアップデート・インストレーションの際にユーザ提供ファイルを実行する場合、システム管理者は作業の 3 と 4 のみを実行します。

図 5-1: 管理者の作業の概要



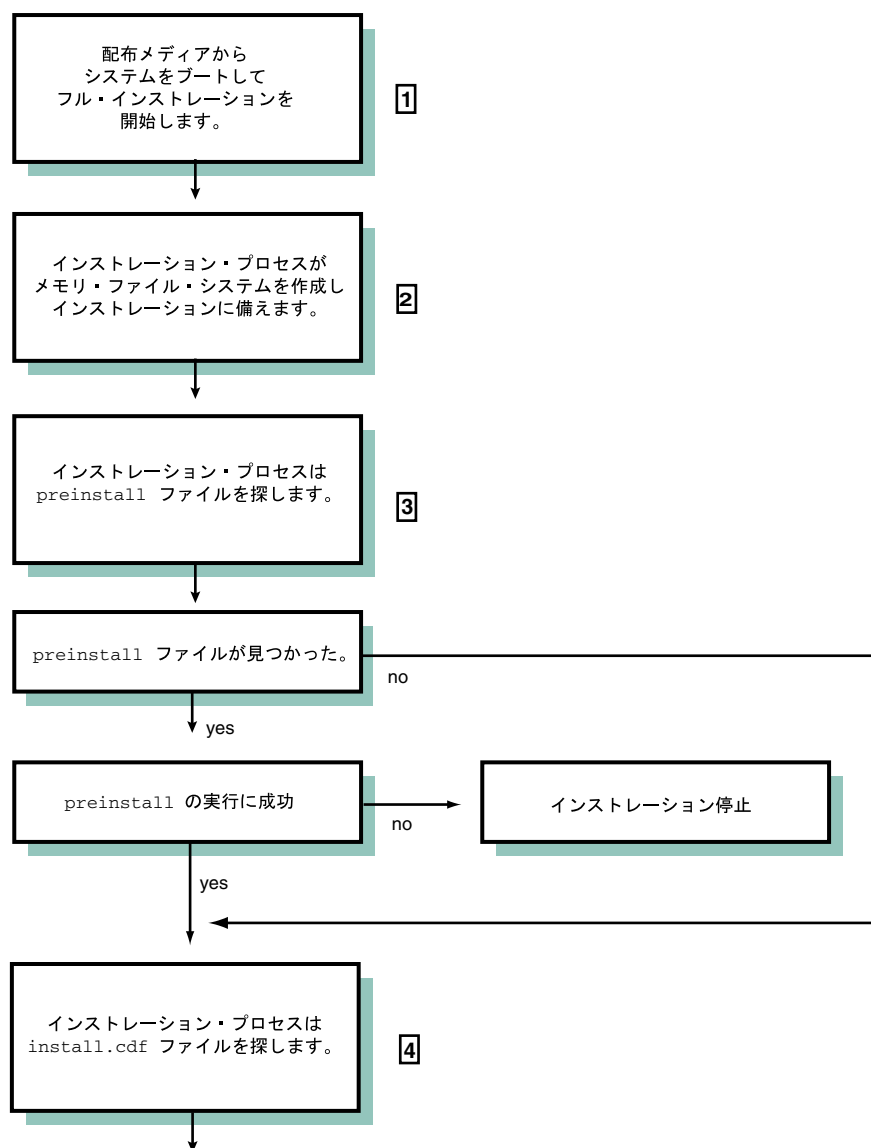
- ① 管理者は、クローニングに適した CDF を生成するかまたは探します。オペレーティング・システムの現行バージョンがインストールされているシステムでは、`install.cdf` は `/var/adm/smlogs` ディレクトリにあります。`config.cdf` は `sysman -clone -save` コマンドを使用してどのディレクトリにも保存できますが、省略時のディレクトリは `/var/adm/smlogs` です。インストールおよび構成のクローニング手順は、それぞれ 第 6 章 と 第 7 章 に記載しています。
- ② 管理者はオリジナルの CDF を作業エリアにコピーして変更します。少なくとも、ホスト固有の情報を変更して、複製したシステムが一意に識別されるようにしなければなりません。オリジナルの CDF には初期のシステム・インストールまたは構成に関する情報が含まれており、将来のトラブルシューティングで必要になる可能性があるため、オリジナルの CDF は、`/var/adm/smlogs` ディレクトリに残しておく必要があります。

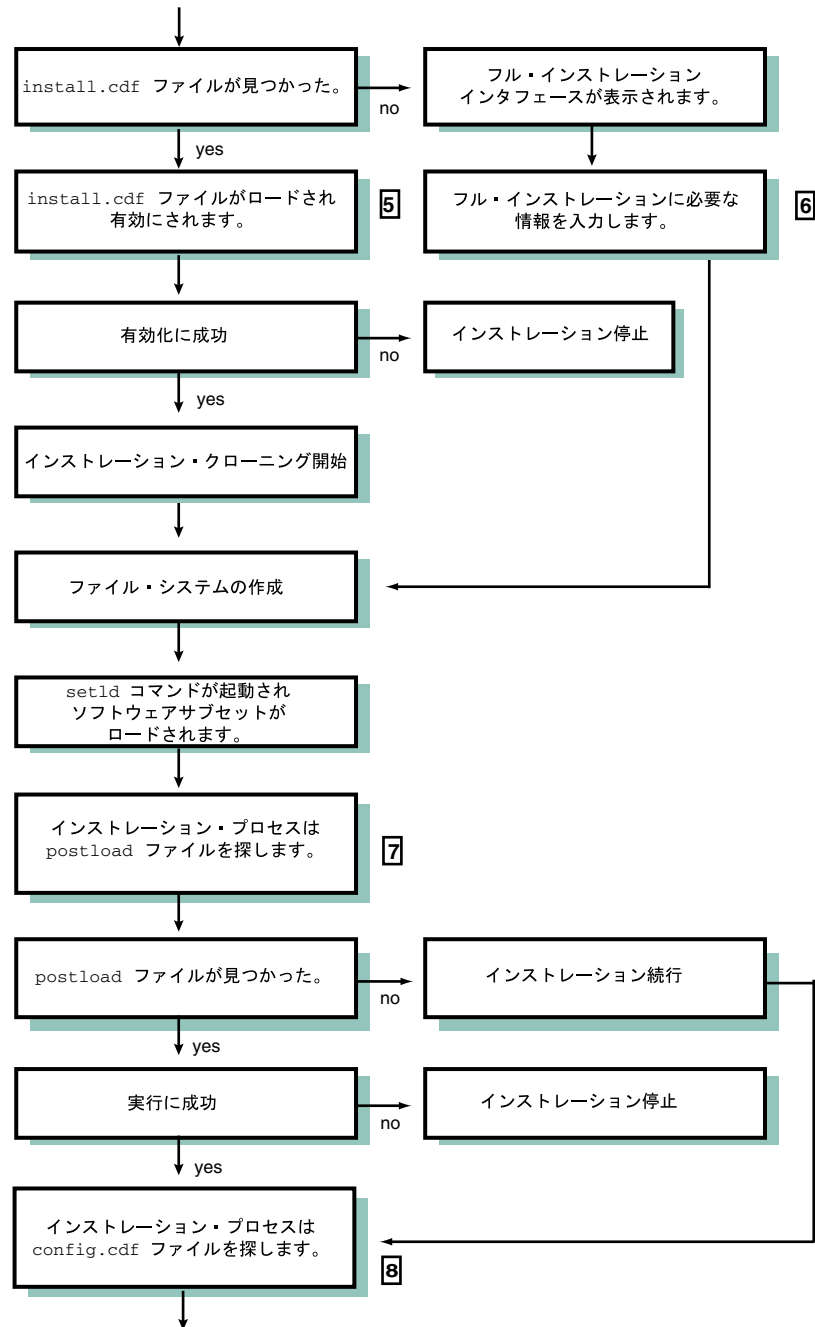
- ③ 管理者は、オプションとして、フル・インストールまたはクローン・インストールのあらかじめ定義された3箇所(アップデート・インストールの処理では2箇所)で実行されるスクリプトまたはプログラムを作成します。管理者は、これらのファイルで実行する動作を定義します。これらのファイルの作成についての詳細は、この章の5.7節を参照してください。
- ④ 管理者は、修正したCDFとユーザ提供ファイルを、ディスクットの/(root)ディレクトリ、RISサーバの/var/adm/ris/clients/sets/profile_setディレクトリ、CD-ROMの/islディレクトリ(配布メディアを再パッケージする場合)のいずれかに移動します。これらのファイルは、展開したRIS領域の内部にある/islディレクトリにもコピーできます。正しい位置へのファイルの移動についての詳細は、この章の5.8節を参照してください。

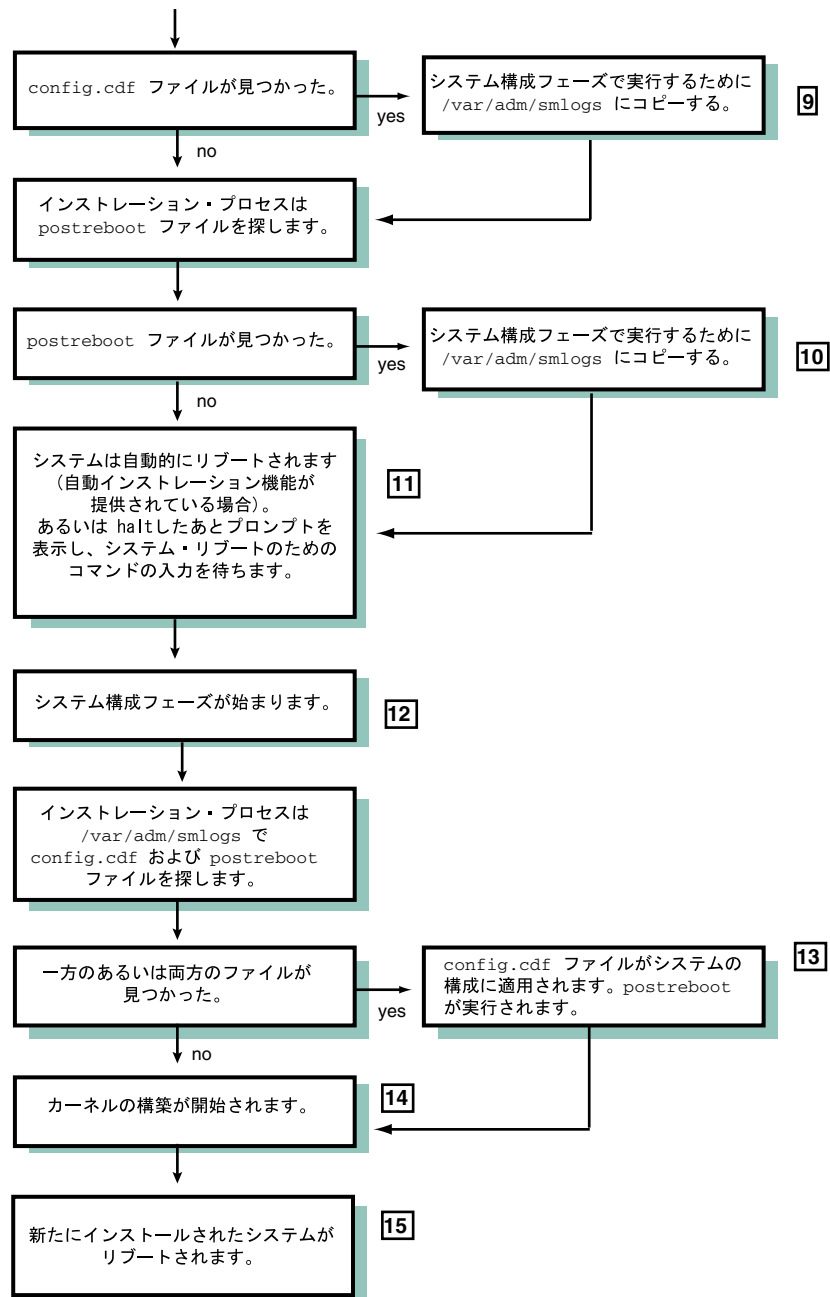
5.6 動作原理

図5-2では、ユーザ提供ファイルとCDFがフル・インストールの際に呼び出される様子を示します。

図 5-2: 動作原理: ユーザ提供ファイルと CDF







1 配布メディアからシステムがブートされ、フル・インストールが開始されます。

5-10 インストール処理のカスタマイズ

- ② メモリ・ファイル・システム (MFS) は、インストール処理に必要な書き込み可能なスペースとして使用されます。
- ③ フル・インストレーション・プロセスは、`preinstall` という名前のファイルを検索します。検索順序とファイルの位置を、表 5-4 に示します。このファイルは、ユーザ提供のスクリプト、プログラム、実行可能ファイルのいずれかであり、インストレーション処理が始まる前に実行するアクションが含まれています。このファイルが見つかると、ファイルが実行されます。実行が正常に終了すると、インストレーション処理が開始されます。実行が正常に終了しなかった場合 (すなわちゼロ以外の終了状態が返された場合)、インストレーション処理は停止します。`preinstall` ファイルが見つからなかった場合、フル・インストレーション・プロセスは `install.cdf` ファイルの検索を開始します。
- ④ 表 5-4 に示す位置のいずれかに `install.cdf` ファイルがあれば、このファイルに基づいてインストレーションの残り部分が処理され、ターゲット・システム上でクローン・インストレーション処理が開始されます。`install.cdf` ファイルが見つからなかった場合、通常のフル・インストレーション処理が開始されます。
- ⑤ インストレーション・プロセスは、ターゲット・システムを複製する前に、`install.cdf` ファイルの妥当性をチェックします。このチェックには、CDF ファイルで指定されたディスク名とディスク・タイプが複製されるシステム上にあることの確認などがあります。RIS インストレーションの場合、このチェックには、CDF に含まれるソフトウェア・サブセットのバージョンと、RIS 環境内にインストールされているソフトウェア・サブセットのバージョンの比較などがあります。CDF のチェックに失敗すると、インストレーションは停止します。チェックに失敗した原因を示す診断メッセージが表示されます。CDF のチェックが正常に終了すると、クローン・インストレーション処理が続行されます。
- ⑥ `install.cdf` ファイルが見つからない場合、通常のフル・インストレーションの際に表示される質問に、インストールしているユーザが応えます。
- ⑦ ソフトウェア・サブセットがロードされた後、フル・インストレーション・プロセスは `postload` という名前のファイルを検索します。検索順序と位置を、表 5-4 に示します。このファイルは、ユーザ提供のスクリプト、プログラム、実行可能ファイルのいずれかであり、ソフトウェア・サブセットがロードされた後に実行するアクションが含まれていま

す。このファイルが見つかり、ファイルが実行されます。実行が正常に終了しなかった場合、インストール処理は停止します。

- ❶ フル・インストール・プロセスは、表 5-4 に示した検索順序と位置で、`config.cdf` ファイルを検索します。
- ❷ `config.cdf` ファイルが見つかり、`/var/adm/smlogs` ディレクトリにコピーされます。このファイルは、インストール処理の後半の構成フェーズで、ターゲット・システムの構成を複製する際に適用されます。
- ❸ フル・インストール・プロセスは、`postreboot` という名前のファイルを検索します。このファイルは、ユーザ提供のスクリプト、プログラム、実行可能ファイルのいずれかであり、インストール・プロセスが、インストールされたシステム・ディスクから初めてブートした後に実行するアクションが含まれています。このファイルが見つかり、システム構成フェーズで実行するために、このファイルは `/var/adm/smlogs` ディレクトリにコピーされます。
- ❹ ソフトウェア・サブセットがロードされた後、システムは自動的にリブートされます。システムに自動リブート機能がない場合、またはユーザの介入が不要になるように `install.cdf` ファイルが変更されていない場合、インストール処理は停止し、新しくインストールしたディスクからシステムをリブートするためのコマンドを入力するようにプロンプトを表示します。システムが自動リブートするようにセットアップされていない場合、システムをリブートするために入力しなければならないブートコマンドが画面に表示されます。
- ❺ システム構成フェーズは、システムのリブート後に自動的に開始されます。構成とは、ソフトウェア・サブセットの調整、ホスト名、`root` パスワード、日付、時刻、ロケーションおよび地域の設定、システムのチューニング、カーネルの構築などの処理です。これらの属性の値が定義されていない場合や、フル・インストールの際にユーザが応答を入力しなかった場合、インストール処理は対話形式の動作になり入力を要求します。
- ❻ フル・インストール・プロセスは、`/var/adm/smlogs` ディレクトリを参照して、`config.cdf` ファイルまたは `postreboot` ファイルがそこに移動されているかどうかを調べます。このディレクトリは、イン

ストレーション・プロセスがファイルを移動する位置です。ユーザがファイルを置くディレクトリ位置ではありません。

`config.cdf` ファイルが見つかると、そのファイルはターゲット・システムを構成するために使用されます。構成のクローニングでは、7.8 節で説明しているような妥当性チェックが実行されます。`config.cdf` ファイルの妥当性チェックに失敗した場合、構成のクローニングは実行されませんが、インストレーション処理は続けられます。`config.cdf` ファイルを他のシステムの複製に適用するには、CDF が妥当性チェックにパスしなければなりません。

`postreboot` ファイルが見つかると、そのファイルが実行されます。実行が正常に終了しなかった場合、フル・インストレーションが続行されます。`postreboot` ファイルの実行が異常終了しても、インストレーション処理の次のステップ (カーネルの構築) は中止されません。

- [14] クローン・インストレーションの場合、カーネル構築のタイプは `install.cdf` ファイルの `kernel_options=` 属性で定義されます。通常のフル・インストレーションの場合、カーネル構築のタイプは、カーネルへ組み込むカーネル構成要素のタイプ (すべてのカーネル構成要素、必須のみ、あるいは必須およびオプションの組み合わせ) によって決まります。
- [15] インストレーションの最終フェーズは、システムの最終リブートです。この後、ユーザは初めてログインできるようになります。フル・インストレーションまたはクローン・インストレーションの場合、この時点でログインできるユーザは `root` だけです。

5.7 ユーザ提供ファイルの作成

ユーザ提供ファイルの内容は、実行する作業によって異なりますが、ユーザ提供ファイルには、読み取り許可と実行許可が必要です。ユーザ提供ファイルを作成する際は、実行する環境に注意してください。たとえば、`preinstall` ファイルおよび `postload` ファイルでは、配布メディア上で使用可能なコマンドとユーティリティだけを呼び出すことができます。これは、オペレーティング・システムの環境がまだ完全にインストールされていないためです。`postreboot` ファイルでは、インストールされたオペレーティング・システムで使用可能なすべてのコマンドとユーティリティを呼び出すことができます。

配布メディアは、ソフトウェアをインストールしたときのレイアウトと同じようにレイアウトされたファイル・システムで構成されています。ただし、性能上の理由およびスペースを考慮して、配布メディアには圧縮されていないソフトウェア・サブセットと圧縮されたソフトウェア・サブセットが組み合わせて収められています。インストールの際には、圧縮されていないソフトウェア・サブセットのコマンドとユーティリティのみが使用できます。配布メディアで利用できるコマンドを確かめるには、そのメディアをマウントして、`cd` コマンドと `ls` コマンドで、内容を表示してください。

5.7.1 項では `preinstall` ファイルの作成についての詳細、5.7.2 項では `postload` ファイルの作成についての詳細、5.7.3 項では `postreboot` ファイルの作成についての詳細を説明します。

5.7.1 `preinstall` ファイルの作成

フル・インストールおよびアップデート・インストールの際に、ユーザ提供ファイルを最初に呼び出すタイミングは、ユーザ・インタフェースを表示する前です。フル・インストールの場合は、`install.cdf` ファイルを検索する前になります。

この時点で、フル・インストール・プロセスは `preinstall` という名前のファイルを検索します。このファイルはユーザ提供のスクリプト、プログラム、実行可能ファイルのいずれかです。このファイルには、インストールのユーザ・インタフェースが表示される前に実行するアクションが含まれています。アップデート・インストール・プロセスは、`update_preinstall` という名前のファイルを検索します。

ファイル・システムの作成とソフトウェア・サブセットのロードの前に実行するアクションには、1つまたは複数のディスクへのカスタマイズしたディスク・ラベルの書き込みがあります。他の利用方法として、汎用の `install.cdf` ファイルにホスト固有の属性を動的に書き込み、そのファイルを `/var/tmp` に置いて、フル・インストール・プロセスが見つけることができるようにするという処理も考えられます。

`preinstall` ファイルでは、インストール済のファイル・システムおよびソフトウェアを必要とする処理を実行しないでください。これは、これらをインストールするフェーズがまだ完了していないためです。ただし、アップデート・インストールの場合、このようなファイル・システムとソフトウェアが使用できます。

preinstall および update_preinstall ファイルと、これらが呼び出すすべてのファイルには、読み取り許可と実行許可が必要です。

preinstall ファイルは、他のユーザ提供ファイルおよび CDF と同じ位置になくてもかまいません (スクリプトから変更する場合を除く)。

インストール・プロセスは、preinstall および update_preinstall ファイルの実行結果の戻り値を調べ、ゼロ以外の戻り値を受け取った場合はインストール処理を停止します。preinstall および update_preinstall ファイルは、独自のステータスまたはエラー・メッセージを出力する必要があります。インストール処理は、ユーザ提供のスクリプトやプログラムの実行結果を保証しませんが、正常に完了した場合にインストール処理が継続されることは保証します。

以下の例に示す preinstall スクリプトの例では、カスタマイズされたディスク・ラベルを RZ26 ディスクに適用します。

例 5-1: 他のファイルを読み出す preinstall スクリプトの例

```
#!/sbin/sh
#
# Write a custom disk label to the
# system disk before starting the installation.
#

# NOTE:  THIS FILE ASSUMES A DISK NAME OF dsk0 AND DISK TYPE OF RZ26

#
# First, zero the label
#
2>/dev/null disklabel -z dsk0

#
# Next, restore the label
#
disklabel -Rr dsk0 ./DiskLabelSave RZ26 || 1
{
    echo "\nError restoring disklabel on dsk0\n"
    exit 1
}

echo "\nThe disklabel that has been applied is:\n"
disklabel -r dsk0 | tail -10
exit 0
```

- 1 preinstall スクリプトから呼び出される DiskLabelSave ファイルは、preinstall スクリプトと同じディレクトリに置き、読み取り許可を設定する必要があります。DiskLabelSave ファイルのサンプルを、例 5-2 に示します。

preinstall スクリプトから呼び出される DiskLabelSave ファイルには、dsk0 のディスク・ラベルを読み取り、その出力をファイルにリダイレクトすることで作成したディスク・ラベルが入っています。このファイルを作成するには、次のようなコマンドを入力します。

```
# disklabel -r dsk0 > DiskLabelSave
```

例 5-2: preinstall スクリプト例から呼び出される DiskLabelSave ファイル

```
# /dev/rdisk/dsk0a:
type: SCSI
disk: rz26
label:
flags:
bytes/sector: 512
sectors/track: 57
tracks/cylinder: 14
sectors/cylinder: 798
cylinders: 2570
sectors/unit: 2050860
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0 # milliseconds
track-to-track seek: 0 # milliseconds
drivedata: 0

8 partitions:
#      size  offset  fstype [fsize bsize  cpg]
a: 131072      0  4.2BSD  1024  8192   16 # (Cyl.      0 - 164*)
b: 262144 131072  unused  1024  8192    # (Cyl. 164*- 492*)
c: 2050860      0  unused  1024  8192    # (Cyl.      0 - 2569)
d: 552548 393216  unused  1024  8192    # (Cyl. 492*- 1185*)
e: 552548 945764  unused  1024  8192    # (Cyl. 1185*- 1877*)
f: 552548 1498312  unused  1024  8192    # (Cyl. 1877*- 2569*)
g: 1210000 393216  4.2BSD  1024  8192   16 # (Cyl. 492*- 2009*)
h: 447644 1603216  4.2BSD  1024  8192   16 # (Cyl. 2009*- 2569*)
```

DiskLabelSave ファイルを作成した後でこのファイルを編集すると、必要に応じてパーティションのサイズをカスタマイズできます。

5.7.2 postload ファイルの作成

ファイル・システムの作成、ソフトウェア・サブセットのロード、保留中のソフトウェア構成フェーズの構成環境の準備が完了すると、フル・インストールおよびアップデート・インストール・プロセスは、システムが初めてリブートされる前に、postload および update_postload という名前のファイルを探します。

ソフトウェア・サブセットがロードされた後に実行するアクションには、他のファイル・システムの作成や、複数のターゲット・システムに適用される config.cdf ファイルの動的な変更があります。付録 Bには postload スクリプトの例を記載し、ホスト固有の属性を config.cdf ファイルに設定する方法を示します。

postload ファイルと postload が呼び出すすべてのファイルには読み取り許可と実行許可が必要です。postload ファイルは、ユーザ提供スクリプトおよび install.cdf ファイルと同じ位置になくてもかまいません。ただし、postload ファイルで config.cdf ファイルを操作する場合は、これらのファイルは同じ位置に置く必要があります。

インストレーション・プロセスは postload ファイルの実行結果を調べ、戻り値がゼロ以外の場合、インストレーション処理を停止します。postload ファイルは、独自のステータスまたはエラー・メッセージを出力する必要があります。インストレーション処理は、ユーザ提供のスクリプトやプログラムの実行結果を保証しませんが、正常に完了した場合にインストレーション処理が続行されることは保証します。

フル・インストレーションのこの時点では、磁気メディア上に新しく作成された / , /usr , /var ファイル・システムは、新しくインストールされたシステム・ディスクからリブートするまでは /mnt ディレクトリの下に相対的にマウントされているという点に注意してください。すなわち、/ ファイル・システムは /mnt , usr ファイル・システムは /mnt/usr , のようになります。

例 5-3に示す postload スクリプトの例では、users という新しいファイル・システムを作成し、次に /etc/fstab ファイルにエントリを追加して、リブートするたびにこの新しいファイル・システムがマウントされるようにします。

例 5-3: postload スクリプトの例

```
#!/sbin/sh
#
# postload - script which is invoked after the subset load of a full
# installation. The script creates a new file system and
# adds an entry in the fstab file. Doing this will make the
# file system available as soon as the installation completes.
#
# Create a new file system on dsk2c which is to be mounted at /usr/users
#
echo "postload:  creating new file system on dsk2c\n"
```

例 5-3: postload スクリプトの例 (続き)

```
# Create the UFS file system on dsk2c, an RZ26L disk.

/usr/sbin/newfs -F /dev/rdisk/dsk2c RZ26L ||
{
    echo "postload: failed to create a new file system on dsk2c\n"

    # We consider this a nonfatal error and allow the install to
    # continue. This is done by returning 0. Otherwise, exit with a
    # non-zero value.

    exit 0
}
# Next, add an entry to fstab so that this new file system is
# automatically mounted when the system boots.

# NOTE: the actual installed file systems are mounted at /mnt.
# Therefore, we want to add the entry to /mnt/etc/fstab and
# not /etc/fstab.

echo "/dev/disk/dsk2c /usr/users ufs rw 1 2" >> /mnt/etc/fstab

# Finally, make sure the mount point is created. Again, create it
# relative to /mnt.

/bin/mkdir /mnt/usr/users

# Process complete!

exit 0
```

5.7.3 postreboot ファイルの作成

すでにスクリプトを記述してシステムを構成しているユーザがそのスクリプトを利用できるようにするためと、構成されたシステムからサポートされているスクリプト機能を利用できるようにするために、3 番目の起動ポイントがあります。

このユーザ提供ファイルは `postreboot` と呼ばれます。このファイルは、フル・インストールのソフトウェア構成フェーズで検索されて起動されます。ソフトウェア構成フェーズは、サブセットがロードされてシステムが汎用カーネルでリブートされた後に行われます。より厳密には、`postreboot` スクリプトは `config.cdf` ファイルをチェックした後に呼び出されるため、`postreboot` スクリプトではネットワーク構成済のシステムを利用できます。

また、`postreboot` スクリプトは、最適化カーネルの構築より前に呼び出されるため、他のレイヤード・ソフトウェア (カーネルが依存するものな

ど) をインストールして、必須のカーネル構成要素の要件を最適化カーネルの構築前に組み込むことができます。フル・インストールの際に `postreboot` スクリプトが検索され、見つかると実行されます。アップデート・インストール・プロセスはこのファイルを検索しません。

インストール・プロセスは、`postreboot` ファイルの実行結果を調べますが、戻り値がゼロ以外でも、インストール処理を停止しません。`postreboot` ファイルは、独自のステータスまたはエラー・メッセージを出力する必要があります。インストール・プロセスは、ユーザ提供のスクリプトやプログラムの実行結果を保証しませんが、スクリプトの終了状態に関係なく、インストール処理を続行することを保証します。

B.5 節に、`postreboot` スクリプトの例を示します。

5.8 ユーザ提供ファイルと CDF を正しい位置にコピーする

CDF およびユーザ提供ファイルと、これらが必要とするファイルはすべて正しいディレクトリに配置し、インストール・プロセスが検索できるようにする必要があります。

フル・インストール・プロセスとアップデート・インストール・プロセスは、表 5-4 に示す順序でユーザ提供ファイルと CDF を検索します。ファイルが見つかると、インストール・プロセスは残りの場所の検索を中止します。たとえば、インストール・プロセスがディスク上で `preinstall` ファイルを見つけると、RIS サーバの検索は行いません。

表 5-4: ユーザ提供ファイルと CDF の格納位置

検索順序	位置	コピー手順の参照先
1	ディスク・ドライブ <code>floppy0</code> または <code>floppy1</code> の / (ルート) ディレクトリ	5.8.1 項
2	クライアント・システムが登録されている RIS サーバ上の <code>/var/adm/ris/clients/sets/</code> ディレクトリにある <code>profile_set</code> サブディレクトリ	5.8.2 項
3	複製されるシステムの <code>/var/tmp</code> メモリ・ファイル・システム (MFS)	5.8.3 項
4	配布メディア (ローカル CD-ROM または展開された RIS 領域) の <code>/isl</code> ディレクトリ	5.8.4 項

5.8.1 ファイルをディスクにコピーする

ユーザ提供ファイルと CDF をディスクにコピーする前に、次のようなコマンド構文を用いて、ディスクをフォーマットし、新しいディスク・ラベルを書き込み、新しいファイル・システムを作成しなければなりません。

```
fddisk -fmt raw_diskette_device
```

```
disklabel -wr diskette_drive disk_type
```

```
newfs raw_diskette_device_partition
```

以下の手順に従って、ディスク・ドライブ `floppy0` のディスクをフォーマットし、ディスクのタイプ `rx23` (標準の 3.5 インチ・ディスク) を指定してディスク・ラベルを書き込み、ディスク全体 (パーティション `c`) に新しいファイル・システムを作成します。

1. ドライブ `floppy0` のディスクをフォーマットします。

```
# fddisk -fmt /dev/rdisk/floppy0c
```

2. 標準の 3.5 インチ・ディスクに新しいディスク・ラベルを書き込みます。

```
# disklabel -wr floppy0 rx23
```

3. ディスク全体、すなわち `c` パーティションに新しいファイル・システムを作成します。

```
# newfs /dev/rdisk/floppy0c
```

`preinstall`, `postload`, `postreboot` ファイルのいずれかがディスク上にある場合、`preinstall`, `postload`, `postreboot` ファイルから呼び出されるファイルもすべて、ディスク上に置かなければなりません。

以下の手順に従ってディスク・ドライブをマウントし、ファイルをディスクにコピーします。

1. マウント・ポイント `/mnt` にディスク・ドライブをマウントします。

```
# mount /dev/disk/floppy0c /mnt
```

2. ファイルが置かれているディレクトリにいと仮定して、以下のコマンドを入力してファイルをディスクにコピーします。

```
# cp ./install.cdf /mnt/install.cdf
```

```
# cp ./preinstall /mnt/preinstall
```

```
# cp ./postload /mnt/postload
```

```
# cp ./config.cdf /mnt/config.cdf
# cp ./postreboot /mnt/postreboot
# cp ./file_name /mnt/file_name
```

3. 次の `chmod` コマンドを実行して、すべてのファイルに実行許可を設定します。

```
# chmod 755 /mnt/*
```

4. ディスケット・ドライブをアンマウントします。

```
# umount /mnt
```

5. ドライブからディスクを取り出し、ターゲット・システムのディスク・ドライブに挿入します。
6. ターゲット・システム上でフル・インストールを開始します (『インストール・ガイド』を参照)。

5.8.2 RIS サーバのプロファイル・セット・ディレクトリにファイルをコピーする

リモート・インストール・サービス (RIS) サーバでは、RIS 管理者が作成した論理的に編成されたサブディレクトリに、CDF とユーザ提供ファイルが格納されます。これらのサブディレクトリはプロファイル・セットと呼ばれ、`/var/adm/ris/clients/sets` ディレクトリに置かれます。システムを RIS クライアントとして登録すると、CDF やインストール中に実行するユーザ提供ファイルが収められているプロファイル・セットにもシステムを登録できます。

プロファイル・セットおよび RIS 管理についての詳細は、『*Sharing Software on a Local Area Network*』を参照してください。

ユーザまたは RIS 管理者が命名規則と RIS サーバ上のプロファイル・セット・ディレクトリの構造を決定した後、次の手順に従って、CDF、ユーザ提供ファイル、および関連するファイルをプロファイル・セット・ディレクトリにコピーします。

1. RIS サーバに `root` としてログインします。
2. `/var/adm/ris/clients/sets` ディレクトリに移動します。

```
# cd /var/adm/ris/clients/sets
```

3. 決定した命名規則に従って、意味のある名前でプロファイル・セット・ディレクトリを作成します。この例では技術部門のプロファイル・セット・ディレクトリを作成しています。

```
# mkdir engineering
```

4. 新しいプロファイル・セット・ディレクトリに移動し、ファイルが正しいディレクトリにコピーされるようにします。

```
# cd engineering
```

5. 正しいCDF、ユーザ提供ファイル、その他の関連ファイルを、作業エリアから新しいプロファイル・セット・ディレクトリ `engineering` に、使い慣れたコピー・ツール (`ftp`、`dcp`、または `rcp`) でコピーします。

6. 次の `chmod` コマンドを実行して、すべてのファイルに読み取り許可と実行許可を設定します。

```
# chmod 755 *
```

7. `ris` ユーティリティを起動して、ターゲット・システムを正しいRISソフトウェア環境とプロファイル・セット・ディレクトリに登録します。

```
# /usr/sbin/ris
```

8. ターゲット・システム上でフル・インストールを開始します (『インストール・ガイド』を参照)。

5.8.3 /var/tmp ディレクトリにファイルをコピーする

`/var/tmp` ディレクトリは、インストール処理で作成される書き込み可能なディレクトリなので、CDF とユーザ提供ファイルの受け渡しには使用できません。ただし、`preinstall` スクリプトを使用すると、CDF、`postload`、`postreboot`、および `postload` と `postreboot` に必要な任意のファイルを、インストール処理の途中で `/var/tmp` にコピーすることができます。これは、ファイルを `/var/tmp` に移動するために使用できる唯一のメカニズムなので、`preinstall` ファイル自体を `/var/tmp` から起動することはできません。

オペレーティング・システムをパッケージし直し、CDF とユーザ提供ファイルを CD-ROM に収めて提供する場合に、この機能が役立ちます。また、多数のクライアント・システムが登録されている RIS 領域にある汎用の CDF を動的に変更する場合にも、この機能は役立ちます。ユーザ提供ファイルを使用すると、インストールして構成するクライアントに応じて CDF を変更

し、結果のファイルを `/var/tmp` ディレクトリに置いて、インストール・プロセスが検索できるようにすることができます。

`/var/tmp` は、ユーザ提供ファイルの 3 番目の検索位置です。この位置を書き込み可能なディレクトリとして使用している場合、最初の 2 つの位置 (ディスク・ドライブと RIS 領域) には、同じユーザ提供ファイルを置かないようにしてください。これは、ファイルが見つかったと検索が終了するためです。

インストール処理の一部として `install.cdf`、`config.cdf`、`postload`、または `postreboot` ファイルを変更したり選択する必要がある場合、CD-ROM には書き込めないため、書き込み可能なディレクトリ位置が必要になります。たとえば、1 つの配布メディアで複数のハードウェアまたは構成をサポートするために、CD-ROM に複数の CDF を格納して出荷するとします。この場合、インストールを実行するシステムを検査する `preinstall` ファイルを作成し、検査結果に基づいて、出荷された CDF ファイルの中から正しいものを選択します。`preinstall` ファイルはこの CDF を `/var/tmp/install.cdf` にコピーします。このようにすると、インストール・プロセスが後でこのファイルを読み取ることができます。同じように、`preinstall` ファイルは複数の `postload` ファイルの中から正しいものを選択して `/var/tmp/postload` にコピーすることができます。

`preinstall` スクリプトでは、`/var/tmp` にコピーされたファイルに正しい許可を設定する必要があります。確実に正しい許可を設定するには、`chmod 777 *` コマンドを実行するのが最も安全な方法です。

5.8.4 CD-ROM にファイルをコピーする

オペレーティング・システムの CD-ROM をパッケージし直して、CDF とユーザ提供ファイルを `/isl` ディレクトリに含めることができます。

注意

ソフトウェアのコピーは、オペレーティング・システムをライセンスに従って使用するときに限って実行できます。コピーしたオペレーティング・システムのすべてに有効なライセンス契約がなければなりません。

install.cdf, config.cdf, preinstall, postload, postreboot
ファイルを CD-ROM 上の /isl ディレクトリに収める場合は、通常の方法
で CD-ROM を作成します (すなわち、CD-ROM に書き込みます)。使用する
方法は、CD-ROM 書き込みソフトウェアによって異なります。

以下の基本手順に従って操作し、CD-ROM 上にイメージを作成します。

1. Tru64 UNIX Version 5.1B の CD-ROM をマウントして、CD-ROM の
内容をコピーする磁気ディスク上にどれだけのディスク・スペース
が必要か調べます。たとえば、/dev/disk/cdrom0c ドライブにあ
る CD-ROM を /mnt ディレクトリにマウントするには、次のような
コマンドを入力します。

```
# mkdir /mnt  
# mount /dev/disk/cdrom0c /mnt  
# cd /mnt
```

2. 次のコマンドを入力して、ディスク容量を K バイト単位で調べます。

```
# df -k
```

表示される数値を記録して、要求を満たすだけのスペースがディスク上
にあることを確認してください。

記録: _____

3. イメージを作成する前に、磁気ディスクのディスク・ラベルを消去し
ます。

```
# disklabel -z /dev/disk/dsk2
```

4. このディスクに省略時のディスク・ラベルを設定します。

```
# disklabel -rw /dev/disk/dsk2
```

5. CD-ROM のマウント・ポイントに移動します。

```
# cd /mnt
```

6. ステップ 2 で調べた値を使用して、次のステップで tar ファイルを作成
するファイル・システムにファイルを保存するための十分な容量があ
るか確認します。別のディスクをマウントして新しく一時的なファ
イル・システムを作成しなければならないこともあります。必要に応じ
て、例で使用している /spare ディレクトリとして、/var/spare や
/usr/spare のような、他の既存ファイル・システムのディレクトリを
使用することもできます (十分な空き領域がある場合)。この後のステッ
プは、ディスク容量をどのように確保するかに応じて変更してください。

7. CD-ROM の内容を含む tar ファイルを作成します。

```
# tar cf /spare/OS1.tar .
```

8. tar ファイルを作成した後、CD-ROM がマウントされたディレクトリの外に移動して CD-ROM をアンマウントします。

```
# cd /  
# umount /mnt
```

9. dsk2 に新しくファイル・システムを作成してマウントします。

```
# newfs /dev/disk/dsk2c  
# mkdir /cdimage  
# mount /dev/disk/dsk2c /cdimage
```

10. OS1.tar ファイルを新しいディスクにコピーします。

```
# cd /cdimage  
# tar xf /spare/OS1.tar
```

11. ユーザのファイルが存在するディレクトリに移動して、cp コマンドを使用して install.cdf、preinstall、postload、config.cdf、postreboot ファイル、およびこれらのファイルから呼び出されるファイルを、イメージの /cdimage/isl ディレクトリにコピーします。

フル・インストールの場合、次のようにファイルをコピーします。

```
# cd file_location  
# cp ./preinstall /cdimage/isl/preinstall  
# cp ./install.cdf /cdimage/isl/install.cdf  
# cp ./postload /cdimage/isl/postload  
# cp ./config.cdf /cdimage/isl/config.cdf  
# cp ./postreboot /cdimage/isl/postreboot  
# cp ./filename /cdimage/isl/filename
```

アップデート・インストールの場合、次のようにファイルをコピーします。

```
# cd file_location  
# cp ./update_preinstall /cdimage/isl/update_preinstall  
# cp ./update_postload /cdimage/isl/update_postload  
# cp ./filename /cdimage/isl/filename
```

12. 使用する CD-ROM 書き込みソフトウェアのタイプに応じた方法で、ディスク上の変更済みイメージを CD-ROM に書き込みます。CD-ROM に書き込むラベルはオペレーティング・システム CD-ROM のラベルと一致している必要があります。一致していないと、書き込み処理は失敗します。disklabel -r コマンドを使用して、label: *string* を探し、オペレーティング・システム CD-ROM のラベルを調べてください。

注意

オペレーティング・システムのイメージが有効でブート可能であることを確認するために、CD-ROM に書き込む前にディスク上のイメージからブートできるか試してみることをお勧めします。

インストレーションのクローニング

この章では、次のトピックについて説明します。

- インストレーションのクローニング処理の概要 (6.1 節)
- インストレーションのクローニングはどのように行われるか (6.2 節)
- インストレーションの構成記述ファイル (CDF) , `install.cdf` の概要 (6.3 節)
- `install.cdf` ファイルのフォーマットと内容 (6.3.1 項)
- `install.cdf` ファイルの例 (6.3.2 項)
- 適切な CDF の生成 , CDF の編集 , クローン・インストレーションの開始についての手順 (6.4 節)

フル・インストレーション・プロセスで行えるその他のカスタマイズについての詳細は、第 5 章を参照してください。

6.1 概要

インストレーションのクローニングでは、インストレーションの特性 (ファイル・システムやインストールされたソフトウェア) を稼働中のシステムから、同一または類似したハードウェア構成の 1 つまたは複数のシステムに複製します。

オペレーティング・システムの現行バージョンをマシンにインストールする際に、インストレーション・プロセスは、ユーザが指定したインストレーション・セットアップ・データのレコードが格納された構成記述ファイル (CDF) を自動的に生成します。このファイルは、`/var/adm/smllogs` ディレクトリに、`install.cdf` というファイル名で作成されます。`install.cdf` ファイルには、ターゲット・システム上で同じインストレーションを行うために必要なインストレーション情報がすべて含まれています。

注意

オペレーティング・システムの Version 5.1B をターゲット・システムに複製する場合、Version 5.1B のフル・インストールで CDF を作成しなければなりません。インストールのクローニングはオペレーティング・システムの異なるリリース間ではサポートされていません。これは、オペレーティング・システムのお他バージョンで作成された CDF が現行バージョンと互換性がないためです。

クローニング処理でインストールするシステムは、CDF が生成されたシステムと同じディスク構成でなければなりません。すなわち、ファイル・システム / (root) , /usr , /var , /usr/i18n および swap 領域用に使用するディスクは、ディスク・タイプとデバイス名が両方のシステムで同じでなければなりません。ただし、構成の多少の違いは許容されています。許容される違いは、6.5.1 項に記載しています。

install.cdf ファイル内の属性を変更して、ターゲット・システムでのユーザの操作を不要にすることもできます。install.cdf ファイルにはホストおよびサイトに固有の属性も含まれていますが、複製されるシステムが一意に識別されるように、これらの属性を変更する必要があります。通常、複製するシステムに対して、ホスト名エントリを変更しなければなりません。

インストールのクローニングと組み合わせて構成のクローニング (説明は第 7 章) やユーザ提供スクリプト (説明は第 5 章) を使用すると、インストールと構成を行った 1 つのシステムから、1 つまたは複数のターゲット・システムに対して、完全な複製とカスタマイズが行えます。

インストールのクローニングを用いて多数のシステムにインストールを行う方法には、次のような利点があります。

- 最小限の労力で同じインストールが実行できる。
- ユーザの操作が最小限になるように、インストールのクローニング処理をセットアップできる。
- インストールのクローニングを使用すると、すべてのシステムで同じインストールを手動で実行する必要がないので、時間を短縮し、環境の違いによるエラーを少なくすることができます。

- ローカルにマウントしたリムーバブル・メディア (CD-ROM など) で同時にインストールを行うのではなく、ソフトウェアを集中管理できる。

6.2 実行方法

システムを複製するには、オリジナルのモデル・システムの `install.cdf` ファイルを、表 6-1 に示す 4 つの位置のいずれかにコピーします。クローニングを行うシステム上でフル・インストールを開始する際に、インストール・プロセスはこの表に記載した順番で `install.cdf` ファイルを検索します。

表 6-1: `install.cdf` ファイルの検索順序

検索順序	位置
1	ディスク・ドライブ <code>floppy0</code> または <code>floppy1</code> のディスク
2	RIS サーバ上の <code>/var/adm/ris/clients/sets/profile_set</code> サブディレクトリ。RIS クライアントを登録する際には、使用する <code>install.cdf</code> ファイルが存在する <code>profile_set</code> ディレクトリにターゲット・システムを登録する必要があります。
3	クローンを作成するシステムの <code>/var/tmp</code> メモリ・ファイル・システム (MFS)
4	配布メディア (ローカル CD-ROM または展開された RIS 領域) の <code>/isl</code> ディレクトリ

インストール・プロセスが表 6-1 に示した場所のいずれかで `install.cdf` ファイルを見つけると、ターゲット・システムでインストールのクローニングが開始されます。ファイルが見つかり、インストール・プロセスは残りの場所の検索を中止します。たとえば、インストール・プロセスがディスク上で `install.cdf` ファイルを見つけると、RIS サーバ上の検索は行いません。`install.cdf` ファイルが上記の 4 つの場所のどこにもなかった場合は、ターゲット・システムで通常のフル・インストールが開始されます。

この章ではこの後、主に `install.cdf` ファイルの編集方法について説明します。CD-ROM からフル・インストールを行う場合は (CDF が置かれている場所には関係なく)、6.6.3 項を参照してください。ここでは、ホストおよびサイトに固有の属性を変更することの重要性を説明しています。

CDF のフォーマットと内容についての詳細は、6.3 節、6.3.1 項、6.3.2 項を参照してください。インストール・クローニングを実行するための手順の詳細については、直接 6.4 節に進んでください。

6.3 インストール CDF の概要

`install.cdf` ファイルには、インストールに関する次のような情報が格納されています。

- ファイル・システムが作成された位置: / (ルート), /usr, /var, /usr/il8n
- スワップ領域が作成された位置
- 各デバイスに対するディスク・ラベル定義
- ファイル・システムが存在するディスクのタイプとディスク名
- ファイル・システムのレイアウト (ファイル・システムが存在する個別のパーティション)
- ファイル・システムのタイプ: UNIX ファイル・システム (UFS) または Advanced File System (AdvFS)
- Logical Storage Manager (LSM) の構成
- ホスト名や暗号化された `root` パスワードなど、ホストに固有の情報、および場所や地域 (タイムゾーン) など、サイトに固有の情報
- インストールに使用した配布メディアのタイプ (CD-ROM または RIS)
- インストールされたソフトウェア・サブセット

6.3.1 項ではインストール CDF のフォーマットと内容を示し、6.3.2 項ではインストール CDF の例を示します。

6.3.1 CDF のフォーマットと内容

`install.cdf` ファイルは、属性と値のペアのグループから構成されています。属性と値は、等号 (=) で区切られています。_item= は、属性と値のペアの論理グループを定義します。スタンザ・ファイルのフォーマットについての詳細は、`stanza(4)` を参照してください。

表 6-2 に、`install.cdf` ファイルの _item= を示します。各項目内の属性と値のペアについての詳細は、付録 A に記載しています。

表 6-2: install.cdf ファイルの項目

_item=	内容
Inst_islinfo	インストールに使用するメディア (CD-ROM , RIS , またはクローン) に関する情報と、インストール処理が開始される前のシステムの状態を伝えるその他のシステム情報
Inst_disklabel	パーティションのサイズやオフセットなど、ディスクの構成情報
Inst_filesystem	システム上で作成されたファイル・システムの数やタイプなど、ファイル・システムに関する情報。作成されたファイル・システムとスワップ領域には、それぞれ Inst_filesystem 項目が 1 つあります。CDF には、/ (root) , /usr , /var , およびスワップ・デバイスを記述するために少なくとも 4 つの Inst_filesystem 項目があります。
Inst_subsets	インストールされたベース・ソフトウェア・サブセットのリスト。ワールドワイド言語サポート (WLS) サブセットをインストールした場合、最大 2 つの Inst_subsets 項目が追加で作成されています。ハードウェア・リリースがインストールされたシステムを複製する場合は、他の Inst_subsets 項目にハードウェアに固有のベース・サブセットが含まれています。
Inst_cinstall	インストール処理に伝達されるターゲット・システムの構成情報。Inst_cinstall 項目に指定された属性はすべてオプションです。これらの属性に値が指定されていない場合、インストール処理は、システムの構成フェーズの中で対話形式でこの情報を要求します。
Inst_lsm_disks	Logical Storage Manager (LSM) のプライベート・リージョンのパーティション情報
Inst_lsm_global	プライベート・リージョンのサイズや LSM ホスト名などの、グローバル LSM 情報

6.3.2 インストール CDF の例

例 6-1 に、install.cdf ファイルの内容を示します。各属性の説明と有効な値は、付録 A に記載します。各_item 内での属性の順序には意味はありません。

例 6-1: インストレーション CDF の例

```
install:
  _item=Inst_islinfo
  _action=create
  media_type=CDROM
  srcloc=/ALPHA/BASE

install:
  _item=Inst_disklabel
  g_size=1433600
  c_offset=0
  e_offset=1812528
  b_size=401408
  g_offset=663552
  d_size=1148976
  b_offset=262144
  f_size=1148976
  name=dsk1
  h_size=2013328
  d_offset=663552
  a_size=262144
  f_offset=2961504
  c_size=4110480
  _action=create
  h_offset=2097152
  e_size=1148976
  a_offset=0

install:
  _item=Inst_filesystem
  disk_number=1
  disk_name=dsk1
  controller_type=SCSI
  name=root
  partition=a
  controller_number=0
  disk_type=RZ28M
  file_system_type=AdvFS
  _action=create

install:
  _item=Inst_filesystem
  disk_number=1
  disk_name=dsk1
  controller_type=SCSI
  name=usr
  partition=g
  controller_number=0
  disk_type=RZ28M
```

例 6-1: インストール CDF の例 (続き)

```
file_system_type=AdvFS
_action=create
```

```
install:
_item=Inst_filesystem
disk_number=1
disk_name="in usr_domain"
controller_type=SCSI
name=var
partition=g
controller_number=0
disk_type=RZ28M
file_system_type=AdvFS
_action=create
```

```
install:
_item=Inst_filesystem
disk_number=1
disk_name=dsk1
controller_type=SCSI
name=swap1
partition=b
controller_number=0
disk_type=RZ28M
file_system_type=swap
_action=create
```

```
install:
_item=Inst_filesystem
disk_number=1
disk_name="in usr_domain"
controller_type=SCSI
name=i18n
partition=g
controller_number=0
disk_type=RZ28M
file_system_type=AdvFS
_action=create
```

```
install:
_item=Inst_subsets
volume_name=DISC1
name=BASE
ss_names=OSFADVFS540,OSFADVFSBIN540,OSFBASE540,OSFBIN540,
OSFBINCOM540,OSFCDEDT540,OSFCDEMAIL540,OSFCDEMIN540,
OSFCLINET540,OSFCMPLRS540,OSFFONT15540,OSFHWBASE540,
OSFHWBIN540,OSFHWBINCOM540,OSFJAVA540,OSFKBDPCXAL540,
```

例 6-1: インストレーション CDF の例 (続き)

```
OSFMITFONT540,OSFNETCONF540,OSFNETSCAPE540,OSFNFS540,
OSFNFSCONF540,OSFOLDX11540,OSFPRINT540,OSFSER540,
OSFSERPC540,OSFSYSMAN540,OSFTCLBASE540,OSFTKBASE540,
OSFX11540,OSFXADMIN540,OSFXPRINT540,OSFXSYSMAN540
advflag=1
_action=create

install:
_item=Inst_subsets
volume_name=DISC2
name=Worldwide_Language_Support
ss_names=IOSPLCDEDT540,IOSPLCDEMAIL540,IOSPLCEMIN540,
IOSPLOLDX11540,IOSPLUCSBASE540,IOSPLX11540,IOSWWBASE540,
IOSWWLAT2FONT100M540,IOSWWLAT9FONT100M540,IOSWWPRINT540,
IOSWWSYSMAN540,IOSWWUCSBASE540,IOSWWX11540
advflag=1
_action=create

install:
_item=Inst_cinstall
kernel_option=interactive
timeset=yes
lang_env=C
password=Bp2xAe46zVpUo
timezone=New_York
locality=America
_action=create
hostname=taurus
```

6.4 インストレーション・クローニング手順の要約

この節では、ターゲット・システムでインストレーションのクローニングをセットアップして実行するための作業の要約を説明します。

1. 適切なインストレーション CDF を作成または選択します (6.5 節)。
2. インストレーション CDF の変更。複製されたシステムでのユーザの操作が最小限になるように、ホストおよびサイトに固有の属性を設定し、特定の属性を組み込みます (6.6 節)。
3. フル・インストレーションでも実行されるユーザ提供スクリプトまたは config.cdf ファイルを作成します (オプション) (6.7 節)。

6-8 インストレーションのクローニング

4. 配布のニーズに応じた適切な場所へ、インストール CDF をコピーします (6.8 節)。
5. ターゲット・システムでのフル・インストールの開始 (6.9 節)。

インストールのクローニングの各ステップの詳細を、以下の節で説明します。

6.5 [ステップ 1]: 適切な CDF の作成と選択

CDF を生成するためにモデル・システムをインストールする場合も、既存の CDF の中から選択してターゲット・システムを複製する場合も、複製するシステムのディスク構成、グラフィックス・アダプタ、フォント・サイズ、キーボード・タイプを考慮する必要があります。ハードウェア構成が同じシステムを複製するのが理想的です。

モデル・システムで標準のフル・インストールを行う際 (クローニングではない)、インストール・プロセスは、システムに備わっているグラフィックス・アダプタ、フォント・サイズ、キーボード・タイプに必要な必須ソフトウェア・サブセットを自動的に判断します。その他のソフトウェア・サブセットはすべてオプションと見なされるため、個別に選択しない限りインストールされません。

システムのクローンを作成する際に、CDF はターゲット・システムにインストールされるソフトウェア・サブセットを定義します。したがって、ターゲット・システムのグラフィックス・アダプタ、フォント・サイズ、キーボード・タイプが、CDF を作成したモデル・システムと異なっている場合、正しいソフトウェア・サブセットがインストールされないため、クローニングで作成したシステムが動作しないことがあります。

異なるシステムでも使用できるような汎用的なインストール CDF を生成するために、モデル・システムでフル・インストールを実行して、そのインストールで生成された CDF が異なるグラフィックス・アダプタ、フォント・サイズ、キーボードを備えたシステムでも使用できるように変更することができます。これは、モデル・システムでは必要なくても、複製されるシステムに必要なグラフィックス・アダプタ、フォント・サイズ、キーボード・タイプをすべてサポートするソフトウェア・サブセットをインストールすることで実現できます。このようなオプションのソフトウェア・サブセットをインストールすると、使用しないソフトウェアが各シ

システムにロードされることになりますが、異なる構成のターゲット・システムのクローニングに使用できる汎用の CDF が作成できます。

ディスク構成、グラフィックス・アダプタ、フォント・サイズ、キーボード・タイプに関して、モデル・システムとターゲット・システム間で許容される相違点を、以下の項で説明します。

6.5.1 許容されるディスクの相違点

ターゲット・システムのハードウェア構成は、CDF の生成に使用したモデル・システムと同じである必要があります。ただし、多少の違いは許されています。

`/`、`/usr`、`swap1`、`/var`、`/usr/i18n` (`/usr` の下のディレクトリでない場合)、`swap2` (割り当てられている場合) が存在するディスクは、ターゲット・システムと CDF を生成したモデル・システムの間でディスク構成が同じでなければなりません。ディスク構成が同じとは、ディスク・タイプ (RZ1BB など) とデバイス名 (`dsk0` など) が一致しているということです。これらのディスクのパーティション・テーブルが両方のシステムで同じでない場合、クローニング・プロセスは、ターゲット・システムのディスクを CDF に記述されているとおりに再構成します。クローニング処理の際に使用されないディスクが、ターゲット・システム上で異なっても問題はありません。

注意

クローニング・プロセスがターゲット・システムのディスクを再構成する場合、再構成されるディスクのユーザ・データが壊れる恐れがあります。

`install.cdf` ファイルには、モデル・システムのディスク・タイプとディスク名が入っています。ターゲット・システムのディスク・タイプとディスク名を調べるには、ターゲット・システムがコンソール・モード (`>>>`) のときに `show dev` コマンドを実行します。

モデル・システムで生成された CDF とターゲット・システムの間で許容される相違点の例を、表 6-3 に示します。どちらのシステムにもディスクが 2 つありますが、ターゲット・システムの 2 番目のディスクが、RZ1CB ディスクではなく RZ1BB ディスクになっています。クローニングは、デバイス名

が dsk0 である最初の RZ1BB ディスクで実行され、このディスクは両システムで共通であるため、この違いは問題になりません。

表 6-3: モデル・システムとターゲット・システムの間で許容されるディスクの相違点

システム	ディスク・タイプ	デバイス名
モデル・システム	RZ1BB	dsk0 ^a
	RZ1CB	dsk1
ターゲット・システム	RZ1BB	dsk0
	RZ1BB	dsk1

^a/ (ルート) および /usr ファイル・システムと swap1 領域は、モデル・システムの dsk0 デバイスにあります。

ディスク構成の違いが他になければ、ターゲット・システムはモデル・システムで生成された CDF を使用できます。 / , /usr , /var ファイル・システムとスワップ領域が dsk1 になれば、デバイス名 dsk1 のディスク・タイプは異なってもかまいません。ただし dsk0 のディスク・デバイスが異なっていると、クローン・インストールは失敗します。

6.5.2 グラフィックス・アダプタの違い

ターゲット・システムのいずれかに、モデル・システムと異なるグラフィックス・アダプタがある場合、ターゲット・システムでクローニング処理を開始する前に、そのシステムに必要なグラフィックス・オプションをサポートするソフトウェア・サブセットをモデル・システムにインストールするか、install.cdf ファイルに手作業で追加する必要があります。

モデル・システムのインストール中にソフトウェア・サブセットを選択する際、「Windowing Environment」カテゴリの中で、「X Servers for <name>」という単語列で始まるソフトウェア・サブセットを探してください。 <name> の部分は、ソフトウェア・サブセットがサポートするグラフィックス・オプションを示す名前に置き換えます。使用できるグラフィックス・ソフトウェア・サブセットは、次のとおりです。

- X Servers Base – デバイス独立の X サーバのサポート (常にインストールされる)
- X Servers for PCbus – EISA バスと PCI バスのグラフィックス・アダプタをサポート

注意

本リリースのオペレーティング・システムでサポートされるグラフィックス・アダプタはすべて、『QuickSpecs』にリストされています。ターゲット・システムに他社製のグラフィックス・アダプタを装着している場合、インストール・クローニング後にこのアダプタのサポートをインストールする必要があります。

モデル・システムとターゲット・システムのグラフィックス・アダプタを表 6-4 に示します。

表 6-4: モデル・システムとターゲット・システムのグラフィックス・アダプタ

システム	グラフィックス・アダプタ
モデル・システム	デバイスに依存しない x サーバ
ターゲット・システム	QVision (PCbus)

モデル・システムの構成に基づいて、インストール・プロセスは、ソフトウェア・サブセット X Servers for Base をモデル・システムの必須サブセットとして自動的にインストールします。したがって、ソフトウェア・サブセット X Servers for PCbus は、モデル・システムではオプションです。クローン・システムで正しいソフトウェアが確実に使用できるようにするには、モデル・システムに X Servers for PCbus ソフトウェア・サブセットもインストールしなければなりません。インストールしないと、ターゲット・システムでグラフィックス機能が使用できなくなる恐れがあります。

注意

グラフィックス機能を備えていないシステムの CDF は、グラフィックス・ハードウェアを備えているシステムのクローニングには使用しないでください。ソフトウェア・サブセットの中には、特に共通デスクトップ環境 (CDE) に関するものに、グラフィックス機能を備えているシステムには必須でも、グラフィックス機能を備えていないシステムにはロードされないものがあります。

モデル・システムにすべてのグラフィックス・ソフトウェアをインストールすると、グラフィックスを確実に動作させることができます。ただし、すべてのソフトウェア・サブセットをインストールすると、選択したグラフィックス・ソフトウェア・サブセットのみをロードする場合よりも多くのディスク容量が必要になることに注意してください。

6.5.3 フォント・サイズの違い

モデル・システムのフル・インストールによって CDF を生成する場合は、ターゲット・システムに必要なフォント・サイズを考慮しなければなりません。CDF で定義されたサイズとは異なるフォントがターゲット・システムに必要な場合、モデル・システムをインストールする際に、正しいフォント・ソフトウェア・サブセットをロードしてください。

DECwindows 75dpi Fonts と DECwindows 100dpi Fonts のどちらが必要かは、使用するグラフィックス・アダプタの解像度によって異なります。すでにオペレーティング・システムがインストールされているシステムでは、この値は次のコマンドを入力して調べることができます。

```
# /usr/sbin/sizer -gr
```

解像度が 1024×768 以下の場合は、DECwindows 75dpi Fonts が必要です。解像度がそれより高い場合は、DECwindows 100dpi Fonts が必要です。複製するシステムの解像度がはっきりしない場合は、両方のフォント・ソフトウェア・サブセットを選択しておけば、正しいフォントが確実に使用できます。

複数のグラフィックス・アダプタを備えたシステムで、1024×768 以下の解像度のアダプタとそれより高い解像度のアダプタがある場合、DECwindows 75dpi Fonts と DECwindows 100dpi Fonts の両方が必要になります。

他にもフォントを含むソフトウェア・サブセットがありますが、フォント・サイズによってパッケージが異なるのは DECwindows フォントだけです。

表 6-5 に、モデル・システムとターゲット・システムに必要なフォント・サイズが異なる例を示します。

表 6-5: モデル・システムとターゲット・システムのフォント・サイズ

システム	グラフィックス解像度	必要なフォント・サイズ
モデル・システム	1024x768	DECwindows 75dpi Fonts
ターゲット・システム	1280x1024	DECwindows 100dpi Fonts

モデル・システムをインストールする際は DECwindows 75dpi Fonts ソフトウェア・サブセットが必須であり、自動的にインストールされます。DECwindows 100dpi Fonts ソフトウェア・サブセットはオプションです。ターゲット・システムのインストールのクローニングに必要なフォントを用意するためには、このオプションのソフトウェア・サブセットもインストールする必要があります。

モデル・システムにすべてのフォント・ソフトウェア・サブセットをインストールすると、正しいフォントを確実に用意できます。ただし、すべてのフォント・ソフトウェア・サブセットをインストールすると、選択したフォントのみをロードする場合よりも多くのディスク容量が必要になります。

6.5.4 キーボード・タイプの違い

モデル・システムのインストールによって CDF を生成する場合は、CDF を用いてクローニングを行うシステムのキーボード・タイプを考慮しなければなりません。クローニングで作成されるシステムのキーボード・タイプが異なる場合は、モデル・システムをインストールする際に、正しいキーボード用のソフトウェア・サブセットをロードしてください。

オペレーティング・システムの現行リリースがすでにインストールされているシステムでキーボード・タイプを調べるには、次のコマンドを使用します。

```
# /usr/sbin/sizer -wk
```

表 6-6 に、モデル・システムとターゲット・システムのキーボード・タイプの例を示します。

表 6-6: モデル・システムとターゲット・システムのキーボード・タイプ

システム	キーボード・タイプ
モデル・システム	PCXAL
ターゲット・システム	LK411

モデル・システムの構成に基づいて、インストール・プロセスは、ソフトウェア・サブセット PCXAL Keyboard Support をモデル・システムの必須サブセットとして自動的にインストールします。したがって、LK411 Keyboard Support のソフトウェア・サブセットはオプションです。このオプションのソフトウェア・サブセットをインストールすると、モデル・システムには不要なソフトウェアもロードされることとなりますが、CDF はターゲット・システムのクローニングに適したものになります。

モデル・システムにすべてのキーボード・ソフトウェア・サブセットをインストールすると、正しいキーボード・タイプを確実に用意できます。すべてのキーボード・ソフトウェア・サブセットをインストールすると、選択したキーボード・ソフトウェア・サブセットのみをロードする場合よりも多くのディスク容量がモデル・システムに必要なことに注意してください。

6.6 [ステップ 2]: CDF の変更

`install.cdf` ファイルを変更することで、フル・インストールで通常は必要とされるユーザからの応答をすべて省略することができます。ホストおよびサイトに固有の属性を変更して、クローニング処理が完了したときにターゲット・システムが一意に識別されるようにしてください。

注意

入力ミスがある場合や、属性と値のペアが間違った項目に挿入された場合は、インストールのクローニング処理が失敗し、クローン・システムが動作しないことがあります。

属性と値のペアには、空白を含めることはできません。空白によりデータ確認エラーが発生するためです。空白はよく注意してすべて削除してください(特に行末の空白)。属性の値をヌルにした場合は、等号(=)の後に何も入力しないように注意してください。

先頭に下線(_)が付いた属性は変更したり削除してはなりません。このような属性(たとえば `_action=create`)は、フル・

インストールとクローン・インストールに必要な内部変数です。

`/var/adm/smlogs` ディレクトリにあるオリジナルの `install.cdf` ファイルは、変更しないでください。代わりに、`install.cdf` をコピーして、そのコピーを変更してください。オリジナルの CDF には初期のシステム・インストールに関する情報が格納されており、将来のトラブルシューティングに役立つことがあるので、この CDF は `/var/adm/smlogs` ディレクトリに残しておいてください。

以下の 3 つの項では、クローニングを自動で実行するための確認属性およびカーネル・オプション属性の設定方法と、クローン・システムが一意になるようにするためのホストおよびサイト固有属性の変更方法について説明します。6.6.4 項では、CDF を変更したときによく発生するエラーを防ぐ方法について説明します。

6.6.1 ユーザの操作を不要にするための CDF 確認属性の設定

特に指定しなければ、インストール・プロセスはユーザに対してプロンプトを表示し、`install.cdf` ファイルを適用するかどうか確認を求めます。この確認をオフにしてユーザの操作を不要にするには、CDF 確認属性をオフにします。この属性は、CDF を用いたインストールのクローニング・プロセスを開始する前に、ユーザの確認が必要かどうかを決定します。この機能は、CDF の `Inst_islinfo` 項目の中の属性と値のペア、`prompt=` を用いて設定します。インストールのユーザ・インタフェースにはこの値を設定する機能がないので、この属性は必ず手作業で CDF に追加する必要があります。有効な値は、表 6-7 のとおりです。

表 6-7: prompt= 属性として有効な値

値	説明
prompt=yes	install.cdf ファイルを用いてインストール・プロセスを実行することを、ユーザに確認します。この値は省略時の値であり、この属性と値のペアが未定義またはヌルの場合に使用されます。
prompt=no	CDF 確認のための質問を省略します。フル・インストール・プロセスが install.cdf ファイルを検出したときには、インストール・クローニング・プロセスが自動的に開始されます。これにより、複製されるシステムでのユーザの操作は不要になります。

例 6-2 の install.cdf ファイルの抜粋では、Inst_islinfo 項目のどこに属性と値のペア prompt= を入れるかを示します。

例 6-2: install.cdf ファイルへの CDF 確認属性の追加

```
install:

_item=Inst_islinfo
prompt=no
media_type=CDROM
server=cosmos
_action=create
srcloc=/ALPHA/BASE
```

6.6.2 ユーザの操作を不要にするためのカーネル・オプション属性の設定

Inst_cinstall 項目の kernel_option 属性は、最適化カーネルに組み込まれるカーネル構成要素のタイプを制御し、ユーザの操作が必要かどうかを制御します。

注意

ターゲット・システム上で構築される最適化カーネルには、モデル・システムのオプション・カーネル構成要素が自動的に組み込まれるわけではありません。ただし、カーネル構築タイプが

interactive に設定され、ユーザが意識的にそれらの構成要素を選択すれば、組み込まれます。

kernel_option 属性として有効な値は、表 6-8 のとおりです。

表 6-8: kernel_option= 属性として有効な値

値	説明
kernel_option=interactive	対話形式でカーネルを構築し、ユーザがオプション・カーネル構成要素をメニューから選択できるようにクローニング処理を停止します。モデル・システムと同じオプション・カーネル構成要素を選択するには、この値に interactive を設定します。
kernel_option=mandatory	すべての必須カーネル構成要素でカーネルを構築します。ユーザの操作は不要になります。
kernel_option=all	すべての必須カーネル構成要素とすべてのオプション・カーネル構成要素でカーネルを構築します。ユーザの操作は不要になります。

例 6-3 の install.cdf ファイルの抜粋では、Inst_cinstall 項目のどこに属性と値のペア kernel_option=を入れるかを示します。この例では値を mandatory に設定し、必須コンポーネントでカーネルを構築し、ユーザの操作を不要にしています。

例 6-3: install.cdf ファイル内でのカーネル構築タイプの設定

```
install:
  _item=Inst_cinstall
  kernel_option=mandatory
  timeset=yes
  lang_env=C
  password=Bp2xAe46zVpUo
  timezone=New_York
  locality=America
  _action=create
  hostname=taurus
```

6.6.3 ホストおよびサイトに固有の属性の設定

CD-ROM からフル・インストールを行う場合は、必ずこの項をお読みください。RIS サーバからフル・インストールを行う場合も、この項を読むことをお勧めします。

ホスト名、地理上のロケーションおよび地域、日付、時刻など、ホストおよびサイト固有情報の設定は、RIS インストールの場合には必要ありません。これらの値は、CDF で定義されている場合でも、RIS サーバから自動的に取得されるためです。これは、RIS からのフル・インストールと RIS からのクローン・インストールの場合に該当します。

ただし、CD-ROM からスタンドアロン・システムをインストールする場合、これらの値はインストールのクローニングに用いる CDF によって決まります。CDF でこれらの属性が定義されていない場合、ソフトウェアがロードされた後に行われる、インストールのクローニング処理のソフトウェア構成フェーズで、値を対話形式で入力する必要があります。クローニング処理でのユーザの操作を不要にしたい場合は、これらの属性の値を定義する必要があります。

考慮すべきホスト固有属性としては、以下のものがあります。

- ホスト名

システムのホスト名は、`Inst_cinstall` 項目の属性と値のペア `hostname=` に設定されています。同じネットワーク上にあるモデル・システムおよびターゲット・システムのホスト名は一意でなければならないので、この値は変更が必要です。`hostname=` 属性が CDF に記述されていない場合、またはこの属性の値がヌルの場合、クローン・インストール処理のソフトウェア構成フェーズで、インストール・プロセスはこの情報を要求するために対話形式の動作になります。LSM 構成データが CDF に記述されている場合は、`Inst_lsm_global` 項目の `lsm_hostname=` 属性には、`hostname=` 属性に指定されているホスト名と同じ値を設定してください。適切なホスト名の選択についてのガイドラインが必要な場合は、『インストール・ガイド』を参照してください。

- パスワード

`password=` 属性に暗号化された値が設定されている場合、複製されるシステムはすべてモデル・システムと同じ `root` パスワードになるので注

意してください。この値をヌルにしておくことにより、インストール・プロセスが対話形式で root パスワードの入力を求めるように設定することも可能です。セキュリティ上の理由から、システム間でパスワードを共有することはお勧めできません。CDF 内に暗号化パスワードを残しておく場合は、そのパスワードがモデル・システムのものであることを念頭に置き、モデル・システムが不正にアクセスされるのを防ぐために、パスワードを変更してください。password= 属性の値は暗号化されていないため、この値を手作業で設定することはできません。モデル・システム上のパスワードを変更する必要がある場合、『インストール・ガイド』に記載されている正しいパスワードの選び方に関するガイドラインを参照してください。

考慮すべきサイト固有属性としては、以下のものがあります。

- 地理上のロケーションと地域

システムの地理上のロケーションとタイム・ゾーンは、Inst_cinstall 項目の属性と値のペア locality= および timezone= に設定されています。オペレーティング・システムの現行バージョンがすでにインストールされているシステムでは、これらの属性に対する有効な値が /etc/zoneinfo ディレクトリに置かれています。

ロケーションおよびタイム・ゾーンを業界標準に準拠した形式に変更したため、このファイルも変更されています。たとえば、locality=US および timezone=Eastern は、現在では locality=America および timezone=New_York と表現されます。どちらの表現も使用できますが、新しいスタイルの方をお勧めします。タイム・ゾーンで分類された locality の値を指定する際は、そのロケーションに対して有効なタイム・ゾーンを選択する必要があります。ロケーションと地域についての詳細は、『インストール・ガイド』に記載しています。

locality= および timezone= 属性が CDF に存在しない場合、またはこれらの属性に対する値がヌルの場合、インストール・プロセスはソフトウェア構成フェーズで対話形式になり、この情報の入力を要求します。timezone= 属性なしで locality= 属性のみが存在することもあります。これは、地理上のロケーションがすべてタイム・ゾーンで分類されているとは限らないためです。たとえば、地理上のロケーション Japan にはタイム・ゾーンがありません。この場合、インストール・プロセスは日本にタイム・ゾーンがないことを認識し、タイム・ゾーンの要求を省略します。

• 日付と時刻

install.cdf には日付や時刻のような動的な値は指定できませんが、複製されたシステムでも正確さを保つことはできます。この機能は、以前にインストール・インタフェースまたは RIS クローン・インストール・プロセスのいずれかにより日付と時刻が設定されたことを install.cdf に示すことで実現されます。日付と時刻が設定されたかどうかを示すために使用される方法は、Inst_cinstall 項目にある timeset= 属性と値のペアです。

timeset 属性として有効な値を、表 6-9 に示します。

表 6-9: timeset 属性として有効な値

値	説明
timeset=no	システムの日付と時刻がまだ設定されていないことを示します。クローン・インストール・プロセスは、対話形式になり、この情報を要求します。
timeset=yes	システムの日付と時刻がすでに設定されていることを示します。したがって、システムの時刻が設定されていないなくても、timeset= 属性に yes を設定して、インストールを自動モードで続行することができます。ターゲット・システムの日付と時刻は、最初のユーザがログインして、date コマンドで正しい値を設定するまでは不定になります。

6.6.4 よくある誤り：行末の空白

CDF を変更する際に、属性と値のペアの後に空白を付けるという誤りを犯すことがよくあります。確認処理で、CDF 内の行末に空白が検出されると、次のようなメッセージが表示されます。

```
-----
Some errors occurred:
SetItmAttr: invalid attribute value kernel_option=all
-----
```

このエラーが発生すると、インストール処理は停止します。上記の例では、確認処理は、属性と値のペア kernel_option=all の all という単語の後に空白を検出しています。訂正するには、CDF を編集して空白を削除

し、CDF を元の位置に戻します。そして、ターゲット・システム上でインストール・プロセスを再起動します。

6.7 [ステップ 3]: その他のユーザ提供ファイルの作成と配置 (オプション)

さらに進んで、モデル・システムでカスタマイズした機能をクローン・システム上に再現したいときには、`preinstall`、`postload`、`postreboot` ファイルを作成します。フル・インストールには、ユーザ提供ファイル呼び出してフル・インストール中にカスタマイズを行う機能があります。フル・インストールでのこれらのファイルの呼び出しについての詳細は、第 5 章 を参照してください。

モデル・システムからターゲット・システムへ構成を複製することもできます。ターゲット・システムにインストールすると同時に完全な構成を行いたい場合は、第 7 章 を参照して、フル・インストールの際に `config.cdf` ファイルを作成して適用する方法を検討してください。

ユーザ提供ファイルと `config.cdf` ファイルは、`install.cdf` ファイルと同じ位置と順番で検索されるため、これらの機能は簡単に組み合わせることができます。構成のクローニングの利点を利用したり、ユーザ提供スクリプトを呼び出してクローン・システムをさらにカスタマイズするには、それぞれ第 5 章 と第 7 章 を参照してください。

クローン・インストール・プロシージャを続行する場合は、6.8 節に進んでください。

6.8 [ステップ 4]: CDF を正しい位置にコピーする

次の処理ステップは、変更した CDF を正しい位置にコピーすることです。`install.cdf` ファイルは、ユーザ提供ファイルや構成のクローニングでの `config.cdf` ファイルと同じ位置に置く必要があり、同じ位置で検索されるため、コピー処理の実際のステップは、ここでは繰り返しません。

メディアの要件に基づいて、`install.cdf` ファイルをどこに移動するかを決定し、表 6-10 に示す項に進んで、ファイルをその場所にコピーする手順を参照してください。

表 6-10: install.cdf ファイルを置く位置

検索順序	位置	コピー手順 の参照先
1	ディスク・ドライブ floppy0 または floppy1 のディスク	5.8.1 項
2	RIS サーバ上の /var/adm/ris/clients/sets ディレクトリ内にある profile_set サブディレクトリ	5.8.2 項
3	複製されるシステム上の /var/tmp メモリ・ファイル・システム (MFS)	5.8.3 項
4	配布メディア (ローカル CD-ROM または抽出された RIS 領域) の /isl ディレクトリ	5.8.4 項

6.9 [ステップ 5]: ターゲット・システム上でのフル・インストールの開始

メディアの要件に基づいて、install.cdf ファイルの内容を検証して正しい位置にコピーしたら、『インストール・ガイド』の説明に沿ってターゲット・システム上でフル・インストールを開始します。インストール・プロセスが install.cdf ファイルを見つけると、ターゲット・システム上でインストールのクローニング処理が開始されます。フル・インストール・プロセスが、正しい位置と名前のユーザ提供ファイルまたは config.cdf を見つけると、これらのファイルも実行されます。

ターゲット・システム上でフル・インストールを開始し、フル・インストールのグラフィカルまたはテキストベースのインタフェースが代わりに表示された場合は、サポートされている 4 箇所のどこにも install.cdf ファイルがなかったということです。この場合は、install.cdf ファイルを表 6-1 に示した場所のいずれかにコピーして、ターゲット・システム上でフル・インストールを再起動してください。



構成のクローニング

この章では、次の項目について説明します。

- 構成のクローニングの概要 (7.1 節)
- 構成のクローニングの実行方法 (7.2 節)
- config.cdf ファイルのフォーマットと内容についての説明 (7.3 節)
- config.cdf ファイルの例 (7.3.1 項)
- config.cdf ファイルを生成するシステムに関する制限と、クローンの作成に適したシステムのタイプ (7.4 節)
- CDF への構成情報の保存、CDF の編集、およびクローニング処理開始の手順 (7.5 節)

7.1 概要

新しくインストールしたシステムを業務に使用し、他のシステムおよびユーザと通信できるようにするには、ネットワーク・サービス、ネーム・サービス、プリンタ、インターネット・サービス、電子メールの配信を構成する必要があります。新しくインストールしたシステムでは、システム構成作業はクイック・セットアップ・アプリケーション、カスタム・セットアップ・チェックリスト、SysMan Menu のいずれかを用いて行います。構成のクローニングが使用できるようになる前は、各システムで個別に構成作業を実行する必要がありました。

構成のクローニングを使用すると、すでに構成されたシステムから 1 つまたは複数のターゲット・システムに構成を複製できるため、個別の構成作業を実行しなくて済みます。

構成のクローニングは、ハードウェアや機能が類似したシステムに対して使用することをお勧めします。構成のクローニングは、クラスやタイプが同じマシンのクローニングに適しています。クローニングに適した状況であるかどうかの判断基準は、システムが同じタイプで、同じタイプのネットワーク・アダプタを同じ数だけ備えているかどうかです。各システムを

個別に構成する代わりに、1つのシステムを構成し、クラスまたはタイプが同じ他のシステムにその構成を複製します。

また、現在の構成が壊れた場合や、以前の構成に戻したい場合など、システムの構成を復元するときにも、構成のクローニングが利用できます。

他のシステムのインストレーションおよび構成を完全に自動化するために、構成のクローニングをインストレーションのクローニングと併用して、ターゲット・システム自身でこれらの作業を実行しなくてもすむようにすることもできます。インストレーションのクローニングには、システムをインストールしたときに自動的に作成される、別のファイル `install.cdf` が必要です。インストレーションのクローニングについての詳細は、第6章を参照してください。

7.2 実行方法

モデル・システムとして使用するシステムの構成が完了したら、`sysman -clone -save` コマンドを使用して、システム構成データのスナップショットを構成記述ファイル (CDF) に保存します。特に指定しなければ、このファイルは `config.cdf` という名前で `/var/adm/smlogs` ディレクトリに保存されます。`config.cdf` ファイルに保存された情報は、類似したシステムに同じ構成を複製するために使用されます。`config.cdf` ファイルを編集して任意の値 (チェックサム以外) を変更することができますが、ホスト固有の属性 (ホスト名や IP アドレス) の中には、複製したシステムがネットワーク上で一意に認識されるようにするために必ず編集しなければならないものもあります。

`config.cdf` ファイルは、フル・インストレーション処理の際に自動的に他のシステムに適用することも、複製するシステムをインストールしてから構成するまでの間に手動で適用することもできます。以下のリストにオプションを要約します。

- システムを1つずつ手動で複製するには、`config.cdf` ファイルを編集してホストおよびサイトに固有の属性を設定します。そして、CDF をターゲット・システムへコピーし、`sysman -clone -apply` コマンドを使用して、インストール済のターゲット・システムにCDFを手動で適用します。

- フル・インストールで1つのシステムを複製するには、`config.cdf` ファイルを編集してホストおよびサイト固有属性を設定し、表 7-1 に示すいずれかの位置に `config.cdf` ファイルを置きます。複製したいシステムでフル・インストールを開始する際に、インストール・プロセスは表 7-1 に示した順序で `config.cdf` ファイルを検索します。

表 7-1: `config.cdf` ファイルの検索順序

検索順序	位置
1	ディスク・ドライブ <code>floppy0</code> または <code>floppy1</code> のディスク
2	RIS サーバ上の <code>/var/adm/ris/clients/sets/profile_set</code> サブディレクトリ。RIS クライアントを登録するには、使用する <code>install.cdf</code> ファイルが存在する <code>profile_set</code> ディレクトリにターゲット・システムを登録する必要があります。
3	クローンを作成するシステムの <code>/var/tmp</code> メモリ・ファイル・システム (MFS)
4	配布メディア (ローカル CD-ROM または展開された RIS 領域) の <code>/isl</code> ディレクトリ

- フル・インストールで複数のシステムを複製するには、`postload` スクリプトを記述し、そこで複製される各システムを定義し、ホストおよびサイトに固有の属性を `config.cdf` に設定します。`config.cdf` ファイルと `postload` スクリプトは、正しい位置に置くとフル・インストールで自動的に実行されます。

注意

構成のクローニングは、インストール時にのみ有効というわけではありません。すでに稼働しているシステムに構成を適用する方法を、7.11.2 項で説明しています。

詳細については、フル・インストール中にどのユーザ提供ファイルと CDF を呼び出すかについて示した、第 5 章の図 5-2 を参照してください。

7.3 構成 CDF のフォーマットと内容

`config.cdf` は、各構成コンポーネントの情報から構成されています。各コンポーネントには、いくつかのグループがあり、その中の 1 つ以上の属性と値のペアで構成データが定義されます。属性とそれに対応する値は、等号 (=) で区切られています。

表 7-2 に、`config.cdf` ファイルのコンポーネントを示します。

表 7-2: `config.cdf` ファイルのコンポーネント

コンポーネント名	説明
<code>bindconfig</code>	ドメイン・ネーム・サービス (DNS) の構成データが含まれています。DNS クライアントのクローニングのみがサポートされます。DNS サーバは複製できません。
<code>internetServices</code>	インターネットへの接続を管理するプロセスの開始と停止を行います。
<code>mail</code>	メール・サーバの構成に関する情報が含まれています。メール・クライアントのクローニングのみがサポートされます。
<code>networkAdapters</code>	構成されているネットワーク・アダプタのタイプを定義します。
<code>networkServices</code>	システムがクラスタのメンバかどうかを指定します。
<code>networkedSystems</code>	主に <code>/etc/hosts</code> データベースのホストを定義します。
<code>networks</code>	DARPA を含む既知のネットワークに関する情報からなる <code>/etc/networks</code> ファイルのデータを定義します。
<code>nfs_export</code>	システムがネットワークを介してローカル・ファイル・システムをエクスポートするように構成されているかどうかを定義します。
<code>nfsconfig</code>	NFS (Networked File Systems) のクライアントおよびサーバの設定を定義します。
<code>nisconfig</code>	NIS (Network Information Services) の構成情報が含まれます。

表 7-2: config.cdf ファイルのコンポーネント (続き)

コンポーネント名	説明
ntpconfig	NTP (Network Time Protocol) のサーバ情報が含まれます。
printcap	/etc/printcap ファイルの情報と類似した、定義済みプリンタに関する情報が含まれます。
remoteWhoServices	rwhod デーモン (rwho および ruptime デーモンが使用するデータベースを保守するサーバ) を管理します。
routing	ルーティングの構成を定義します。

7.3.1 構成 CDF の例

例 7-1に、config.cdf ファイルの bindconfig コンポーネントを示します。シャープ記号 (#) で始まる行はコメント行です。config.cdf ファイルのコンポーネントの最後にある情報は Group: componentid です。ここには、検証のためにクローニング・プロセスが必要とする内部情報が含まれています。

注意

紙面の都合上、例には config.cdf ファイルの一部しか記載していません。config.cdf ファイルの全体を参照するには、構成済みのシステムで sysman -clone -save コマンドを用いて、構成情報を省略時の位置 /var/adm/smlogs/config.cdf に保存し、次にテキスト・エディタなどの使い慣れたツールでファイルを表示してください。

例 7-1: config.cdf ファイルの一部

```
CHECKSUM=56823

#
# CDF Created: Fri Aug 23 11:34:30 EST 2002
#
#
# Component: bindconfig
#
# Group: bind
#
/bindconfig/bind:
    configured=YES
    bindtype=CLIENT
#
# Group: resolver
#
/bindconfig/resolver:
    change_hostname=NO
    domain=mydomain.com
    precedence=first
#
# Group: ns
#
/bindconfig/ns:
    cdf_record=00000001
    ipaddress=16.29.221.1
    hostname=libra.mydomain.com
#
# Group: search
#
/bindconfig/search:
#
# Group: componentid
#
/bindconfig/componentid:
    manufacturer=Hewlett-Packard Company
    product=Domain Name Service Configuration
    version=DNS-1.1.4.17
    serialnumber=1.1.4.17
    installation=19990203184929.000000-300
    verify=7
#
# CDF Created: Fri Aug 23 11:34:47 EST 2002
#
```

7.4 構成のクローニングに関する制限事項

他のシステムへ複製するために、config.cdf ファイルを生成したり選択する際には、以下の制限に従うことをお勧めします。

- 構成のクローニングはオペレーティング・システムの異なるリリース間ではサポートされていません。Version 5.1Bを実行しているシステムに構

7-6 構成のクローニング

成を複製する場合、使用する config.cdf ファイルは Version 5.1B をインストールして構成したモデル・システムで生成する必要があります。

- ソフトウェア・パッチはクローニングされません。パッチをインストールするとシリアル番号が変わるため、config.cdf ファイルがターゲット・システムに適用された際に検証エラーになります。このため、この制限事項があります。config.cdf ファイルを作成するときには、パッチされていないモデル・システムから作成しなければなりません。

モデル・システムにすでにパッチが適用されている場合は、代替手段として次の 2 つの方法があります (ただし、お勧めしません)。

- パッチが適用されているシステムから config.cdf を作成します。ターゲット・システムへ config.cdf ファイルを適用するときに -f オプションを使って、config.cdf ファイルをターゲット・システムに強制的に適用します。たとえば、ターゲット・システムで次のコマンドを入力します。

```
# sysman -clone -apply -f config.cdf
```

その後、他のシステムへのクローニングに使うために、クローニングで新しく作成されたターゲット・システムで別の config.cdf を作成します。新しい config.cdf のチェックサムは正しいため、このファイルを使うと、残りのシステムへのクローニングが正常にできます。

注意

ターゲット・システムに CDF を強制的に適用する -f オプションは、主に内部デバッグのために使われます。この config.cdf にはパッチされたモデル・システムの問題以外にも問題がある可能性があるため、CDF を強制的に適用すると、ターゲット・システム上で回復不可能な問題が発生することがあります。

- 次のバージョンがリリースされている場合は、アップデート・インストール手順を使って、パッチが適用されているモデル・システムを次のバージョンのオペレーティング・システムにアップデートします。これにより、パッチの問題を回避します。

- ソフトウェア・ライセンスは複製されません。オリジナルのシステムにインストールして登録した製品の認証コード (PAK) は、ターゲット・システムには複製されません。ソフトウェア・ライセンスをインストールして登録するには、クローン・システム上で License Manager アプリケーションまたは `lmfsetup` スクリプトを実行する必要があります。
- 複数のネットワーク・インタフェース (アダプタ) を持つシステムで作成された `config.cdf` ファイルは、ネットワーク・アダプタが 1 つしかないシステムの構成のクローニングには使用できません。このようなクローニングを行う場合は、CDF から余分なアダプタの情報を注意深く削除してください。そうしないと、CDF は検証に失敗します。
- 構成のクローニングは、類似したクラスのマシンのクローニングに最も適しています。サーバ・クラス・システムから作成した CDF はサーバを複製するために使用し、ワークステーション・クラス・システムから作成した CDF は他のワークステーションを複製するために使用することをお勧めします。

ワークステーション・クラス・システムは、どのタイプであれサーバ用に構成されたマシンの CDF を用いて複製しないでください。この CDF には、ワークステーションでは必要としないサーバのセットアップ情報がたくさん含まれているため、ネットワークに悪影響を与えることがあるためです。たとえば、システムがサイトのプライマリ IP ルータとして構成されている場合は、他のシステムに複製しないでください。

- 構成のクローニングはクラスタ環境ではサポートされていません。個々のクラスタ・メンバはディスクを共有しており、CDF には個々のメンバの情報ではなくクラスタ全体に関する情報が保存されるので、クラスタ・メンバの構成は保存しないでください。

7.5 構成のクローニング手順の要約

以下のリストは、モデル・システムから 1 つまたは複数のターゲット・システムに構成を複製する際に必要な手順を要約したものです。

1. `sysman -clone -save` コマンドを使用して、構成済みのモデル・システムから `config.cdf` ファイルにシステム構成情報を保存します (7.6 節)。
2. `config.cdf` ファイルを変更して、ホスト固有属性を設定し、必要の場合はサイト固有属性も設定します (7.7 節)。

3. CDFを使って構成のクローニングを行う前に、CDF を検証します (7.8 節)。
4. フル・インストール時に複数システムの構成のクローニングをセットアップするには、オプションの `postload` スクリプトを作成し、ホスト固有情報を動的に設定します (7.9 節)。
5. ターゲット・システムへの適用方法または配布メディアの要件に応じて、`config.cdf` ファイルを正しい位置にコピーします (7.10 節)。
6. フル・インストール・プロセスまたは `sysman -clone -apply` コマンドを使用して、ターゲット・システムに `config.cdf` ファイルを適用します (7.11 節)。

7.6 [ステップ 1]: 構成情報を構成 CDF に保存する

インストールのクローニングでは `install.cdf` ファイルは自動的に作成されますが、構成のクローニングでは、意識的に構成データを `config.cdf` ファイルに保存する必要があります。この処理が自動化されていない理由は、人が介入しなければ、いつシステム構成が完了したか分からないためです。構成データは、システムがインストールされた後ならいつでも、何度でも CDF ファイルに保存できます。構成データを保存する場合、値が設定されていない属性は `config.cdf` ファイル内に定義されません。

1 つの `config.cdf` ファイルを使用して多数のシステムに構成を複製する場合は、1 つのシステムに対してインストールと構成を行ってモデル・システムとして使用します。このようにすると、ホスト固有属性以外は、CDF に必要な編集はほとんどなくなります。

必要な構成を行ったシステム上で、次のようにフラグを指定した `sysman` コマンドを用いて、構成データを CDF に保存します。

`sysman -clone -save [filename]`

このコマンドでは、別のパスおよびファイル名を指定しない限り、現在のシステム構成が `/var/adm/smlogs/config.cdf` ファイルに保存されます。フル・インストール処理の際にシステム構成を複製する場合、インストール・プロセスは `config.cdf` という名前のファイルを検索することに注意してください。他の名前の構成 CDF は認識されません。

また、`sysman -clone -save` コマンドは、CDF を有効な状態で作成します。CDF が有効かどうかは、`config.cdf` ファイルの先頭にあるチェック

サムによって確認されます。CDF をターゲット・システムの複製に適用する際、不正な変更によってシステムが使用できない状態になるのを防ぐために、チェックサムによって config.cdf ファイルの正当性が判断されます。このチェックサムは変更しないでください。

構成されているコンポーネントをすべてリストするには、次のように sysman コマンドを使用します。

```
sysman -clone -list
```

7.7 [ステップ 2]: 構成 CDF 内のホストおよびサイトに固有の属性を変更する

config.cdf ファイルには、ホストおよびサイトに固有の属性が含まれています。この属性は、各クローン・システムがネットワーク上で一意に識別されるようにするために、編集が必要です。/var/adm/smlogs ディレクトリにあるオリジナルの config.cdf ファイルは変更しないでください。代わりに、このファイルをコピーして、コピーの方を変更してください。オリジナルの CDF にはモデル・システムの構成に関する情報が含まれており、これが将来のトラブルシューティングやオリジナルの構成の復元に役立つことがあるので、オリジナルの CDF は /var/adm/smlogs ディレクトリに残しておいてください。変更した config.cdf ファイルは、変更の記録としてバックアップを保存してください。

注意

ファイルの先頭にはチェックサムがあり、チェックのために使用されます。この番号は変更しないでください。変更すると、クローニング・プロセスが CDF の正当性を確認できなくなり、その後他のシステムのクローニングに使用できなくなる恐れがあるためです。

このファイルを編集する場合は、意味がわからない属性や値は変更しないでください。ホストまたはサイトに固有の情報があり、それがターゲット・システムと互換性のないことがわかっている場合（たとえばモデル・システムのホスト名など）は、変更を加えてください。この章で説明している属性と値だけを変更することをお勧めします。

表 7-3には、クローン・システムがネットワーク上で一意に識別されるようにするために、構成を適用する前に変更する必要がある、ホスト固有の情報をリストしています。複製されるシステムのハードウェアまたはプロセッサのタイプが、config.cdf ファイルが生成されたシステムと正確に一致しない場合には、3 番目の属性を考慮してください。

表 7-3: config.cdf ファイル内のホスト固有の属性

ホスト固有の属性	説明
systemName=	ネットワーク上で認識されるシステム名を設定します。システムにはそれぞれ一意の名前を付けなければならないので、この属性は変更が必要です。この属性は interface グループの networkAdapters コンポーネントにあり、他にどのようなコンポーネントが構成されているかに応じて、名前が他の属性にも指定されることがあります。システム名は必ず検索してすべて変更するようにしてください。
networkAddress=	ネットワーク上での認識に利用されるクライアント・システムに対する一意のインターネット・プロトコル (IP) アドレスを設定します。この属性は interface グループの networkAdapters コンポーネントおよび hostMappings グループの networkedSystems コンポーネントにあります。
devName= と type=	システムに実装されているネットワーク・アダプタを定義します。この値は、複製されるシステム上のネットワーク・アダプタが config.cdf ファイル内に定義されているものと異なっている場合のみ変更します。この属性に対して有効な値には、Tulip デバイスの tu0、Lance デバイスの ln0、FDDI デバイスの fddi0、および ee0 があります。システムがまだオペレーティング・システムを実行していない場合に、複製されるシステムで利用できるネットワーク・アダプタを調べるには、コンソール・モードのプロンプト (>>>) で show dev および show config コマンドを実行します。複製されるシステムでオペレーティング・システムが実行されている場合は、rcmgr mget grep NET コマンドを使用して、ネットワーク・アダプタ・デバイス (NETDEV_0) を調べます。

7.7.1 項 では、変更が必要なサイト固有の属性について説明します。
7.7.2 項 では、複数のレコードを持つコンポーネント・グループを変更するための CDFMODE 属性の使い方について説明します。

7.7.1 [オプション]: 構成 CDF 内のサイト固有情報を変更する

`config.cdf` ファイルには多くのネットワーク関連の属性が含まれています。この属性は、複製するシステムごとに要件が異なる場合に、編集が必要になります。たとえば、特殊なネットワーク構成が必要な特定の部署またはサイトのシステムをすべて複製することがあります。このような場合、変更の検討が必要になる属性は、DNS サーバ、ネットマスク、ドメイン名、ネットワーク・ルーティングのタイプ、ネットワーク・タイム・プロトコル (NTP) のサーバ名、省略時のプリンタ、プリンタおよびメール・クライアント構成などです。

7.7.2 複数のレコードを持つコンポーネント・グループに対して CDFMODE 属性を使用する

CDFMODE 属性は、複数のレコードを持つコンポーネント・グループのみに適用されます。CDFMODE は、CDF 全体のコンポーネント・グループ内にあるすべての `cdf_record` ラベルに適用されるグローバルな属性です。複数のレコードを含むコンポーネント・グループの例には、`networkedSystems` コンポーネント内の `hostMappings` グループがあります。このグループは、`/etc/hosts` ファイル内のデータを表します。

構成 CDF では複数レコード・グループ内の各レコードを `cdf_record=` というラベルで開始し、グループ内に複数のレコードがあることを示します。例 7-2 は、複数のレコードがあるホスト・グループの例です。

例 7-2: 複数のレコードがあるコンポーネント・グループの例

```
# Component: networkedSystems
#
# Group: hostEquivalencies
#
/networkedSystems/hostEquivalencies:
#
# Group: hostMappings
#
/networkedSystems/hostMappings:
    cdf_record=00000001
    networkAddress=127.0.0.1
    systemName=localhost
    cdf_record=00000002
    aliases=aries
    networkAddress=16.29.221.1
```

例 7-2: 複数のレコードがあるコンポーネント・グループの例 (続き)

```
systemName=aries.mydomain.com
cdf_record=00000003
networkAddress=16.29.221.15
systemName=pluto
cdf_record=00000004
aliases=virgo
networkAddress=16.29.221.27
systemName=virgo.mydomain.com
#
#
```

表 7-4 に、CDFMODE 属性の有効な値を示します。

表 7-4: CDFMODE 属性の値

値	説明
MERGE	コンポーネント/グループのデータを、ターゲット・システム上の既存のデータとマージします。重複するデータは無視されます。CDFMODE が指定されていない場合、これが構成 CDF 全体に対する省略時の動作になります。
APPEND	コンポーネントおよびグループのデータを、ターゲット・システム上の既存の構成データに追加します。重複するデータは無視されません。
REPLACE	ターゲット・システム上のコンポーネントおよびグループのデータを、構成 CDF 内のデータで置き換えます。ターゲット・システムには存在するが CDF には存在しないデータは削除されます。構成 CDF には存在するがターゲット・システムには存在しないデータは適用されます。CDF とターゲット・システムの両方に存在するデータは、適切に変更されます。

CDFMODE 属性は、config.cdf ファイル内のコンポーネントの外にも置くことができます。

注意

例 7-3 に示すように、CDFMODE= 属性は行の最初のカラムから始まっていなければなりません。スペースやタブを挿入してインデントしないでください。

CDFMODE の値は、別の CDFMODE= 文に到達するまで有効です。例 7-3 に、CDFMODE 属性の使い方を示します。この例では、CDFMODE=REPLACE が見つかるまでは、省略時の MERGE モードが有効です。この指定により、/etc/hosts ファイル全体が CDF の内容で置き換えられます。その後、CDFMODE=MERGE (省略時のモード) に到達し、残りの属性は複製されるシステムにマージされます。

例 7-3: CDF への CDFMODE 属性の挿入

```
CDFMODE=REPLACE
#
# Component: networkedSystems
#
# Group: hostEquivalencies
#
/networkedSystems/hostEquivalencies:
#
# Group: hostMappings
#
/networkedSystems/hostMappings:
    cdf_record=00000001
    networkAddress=127.0.0.1
    systemName=localhost
    cdf_record=00000002
    aliases=host1
    networkAddress=16.29.221.2
    systemName=host1.mydomain.com
    cdf_record=00000003
    networkAddress=16.29.221.16
    systemName=host2
    cdf_record=00000004
    aliases=host3
    networkAddress=16.29.221.28
    systemName=host3.mydomain.com
#
CDFMODE=MERGE
```

7.8 [ステップ 3]: 変更した CDF の検証

config.cdf ファイルを変更したら、次のコマンドで内容を検証してください。

7-14 構成のクローニング

sysman -clone -validate [*filename*]

ファイル名を省略すると、ファイル名は `/var/adm/smlogs/config.cdf` と見なされます。このコマンドは、ファイルの先頭にあるチェックサムを用いて、変更によってファイルが他のシステムの複製に使用できなくなっていないか調べます。検証処理では、エラーが発生したコンポーネント・グループの切り分けが行われ、メッセージが表示されます。検証エラーが発生した場合は、未変更のオリジナルの `config.cdf` を使ってやり直してください。

7.9 [ステップ 4]: フル・インストールの際に複数のシステムを複製するスクリプトを作成する (オプション)

フル・インストールの際に `config.cdf` ファイルを用いて多数のシステムを一度に複製するには、ホスト固有の値に変数を設定した `config.cdf` ファイルを代表で 1 つ作成してください。システムに適用する前に変数に値を手動で入力するか、スクリプトを作成してフル・インストールの際に呼び出し、`config.cdf` ファイルがターゲット・システムに適用される前にそのファイル内の変数を動的に変更するようにします。

フル・インストール・プロセスは、フル・インストールの途中で、`config.cdf` ファイルが置かれている場所で特定のスクリプトを検索します。インストール・プロセスが適切なディレクトリ位置で `postload` という名前のスクリプトを見つけると、ソフトウェア・サブセットがロードされてからカーネルの構築が行われるまでの間に `postload` スクリプトが実行されます。これは、`config.cdf` ファイルが適用されるフェーズでもあります。付録 B にはこのような `postload` スクリプトの例を記載し、ホスト固有の値に対する変数の作り方を示します。`postload` ファイルと `config.cdf` ファイルは同じ場所になければなりません。

7.10.3 項では、インストール・プロセスが `postload` ファイルと `config.cdf` ファイルを検索する 4 つの場所を説明します。

7.10 [ステップ 5]: 構成 CDF を正しい位置にコピーする

`config.cdf` ファイルの位置は、フル・インストールでシステム構成を複製する場合と、すでにインストールしてまだ構成していないシステムの構成を複製する場合で異なります。

- `config.cdf` ファイルを手動で単一のシステムに適用するか、システムに 1 つずつ適用する場合は、7.10.1 項に示すように `config.cdf` ファ

イルをディスクにコピーするか、7.10.2 項 に示すように一時的なネットワークをセットアップして、クローニングを行うシステムに config.cdf ファイルを移します。

- フル・インストールの際にシステム構成を複製する場合は、7.10.3 項を参照して、検索対象としてサポートされている位置を調べてください。これらの位置は、インストール・プロセスがユーザ提供スクリプトと install.cdf ファイルを検索する位置と同じです。

7.10.1 構成 CDF をディスクにコピーする

以下の手順で、config.cdf ファイルをディスクにコピーします。

1. ディスクをフォーマットし、新しいディスク・ラベルを書き込み、新しいファイル・システムを作成します。

```
# fdisk -fmt /dev/rdisk/floppy0a
# disklabel -wr floppy0 rx23
# newfs /dev/rdisk/floppy0c
```

2. マウント・ポイント /mnt にディスクをマウントします。

```
# mount /dev/disk/floppy0 /mnt
```

3. 構成データを config.cdf という名前でディスクに保存します。

```
# sysman -clone -save /mnt/config.cdf
```

config.cdf ファイルをディスクにコピーしたら、複製するシステムのディスク・ドライブにディスクを挿入してマウントします。config.cdf ファイルを手動で適用してシステムを自動的に構成する方法については、7.11.2 項を参照してください。

7.10.2 構成 CDF をネットワークに接続されていないシステムにコピーする

構成を複製するシステムがインストールされていてまだ構成されておらずディスク・ドライブがない場合、一時的にネットワークに接続する以外に、config.cdf ファイルをコピーする方法はありません。以下の手順で一時的なネットワークをセットアップし、次にファイル転送プロトコル (FTP) を用いて、複製するシステムに config.cdf ファイルをコピーします。CDF をコピーする元のシステムの IP (Internet Protocol) アドレスを知っている必要があります。

すでにインストールされていてまだ構成されていないシステムで、次の手順を実行します。両方のマシンで root の権限を持っているものとします。

1. クローニングを行うシステムで、次のコマンドを実行してネットワーク・アダプタと IP アドレスを調べます。

```
# ifconfig -a
```

マシンの IP アドレスを含む行を探して、どのネットワーク・インタフェースを構成すればよいか調べます。

2. 一時的にネットワーク・アダプタをネットワークに接続します。

```
# ifconfig adapter_name IP_address
```

3. セットアップしたネットワーク接続をチェックして、システムが CDF の置かれているモデル・システムを認識することを確認します。

```
# /usr/sbin/ping -c2 model_system_name
```

4. config.cdf ファイルの省略時の位置である次のディレクトリに移動します。

```
# cd /var/adm/smlogs
```

5. ファイル転送プロトコル (FTP) を用いて、config.cdf ファイルが生成されたモデル・システムに接続します。ドメイン・ネーム・サービス (DNS) は実行されていないので、モデル・システムのシステム名ではなく IP アドレスを指定してください。

```
# ftp model_system_IP_address
```

6. ユーザ root として FTP にログインし、root のパスワードを入力します。

7. config.cdf ファイルのあるディレクトリに移動します。この例では、ファイルは省略時の位置にあるものと仮定しています。

```
ftp> cd /var/adm/smlogs
```

8. config.cdf ファイルをコピー (get) して、クローニングを行うシステムに転送します。

```
ftp> get config.cdf
```

9. 接続をクローズして FTP を終了します。

```
ftp> bye
```

これで、config.cdf ファイルは、クローニングを行うシステムの /var/adm/smlogs ディレクトリに置かれました。 7.11.2 項に進んで、config.cdf ファイルを手動で適用してシステムを構成する方法を参照してください。

7.10.3 構成 CDF を配布メディアにコピーする

フル・インストールの際に構成を複製するには、オリジナルのモデル・システムの config.cdf ファイルを表 7-5 に示す 4 箇所のいずれかにコピーします。クローニングを行うシステム上でフル・インストールを開始すると、インストール・プロセスはこの表に示した順番で config.cdf ファイルを検索します。

config.cdf ファイルはインストールのクローニングで使用するユーザ提供ファイルおよび install.cdf ファイルと同じ場所に配置され検索されるので、コピー処理の実際のステップは、ここでは繰り返しません。メディアの要件に基づいて、config.cdf ファイルをどこに移動するかを決定し、下記のテーブルの 3 番目の欄に示す項に進んで、ファイルをコピーする手順を参照してください。

表 7-5: config.cdf ファイルを置く位置

検索順序	位置	コピー手順 の参照先
1	ディスク・ドライブ floppy0 または floppy1 のディスク	5.8.1 項
2	RIS サーバ上の /var/adm/ris/clients/sets ディレクトリ内にある profile_set サブディレクトリ	5.8.2 項
3	クローンを作成するシステムの /var/tmp メモリ・ファイル・システム (MFS)	5.8.3 項
4	配布メディア (ローカル CD-ROM または展開された RIS 領域) の /isl ディレクトリ	5.8.4 項

config.cdf ファイルを正しい位置にコピーしたら、7.11.1 項に進んで CDF をターゲット・システムに適用してください。

7.11 [ステップ 6]: 構成 CDF をターゲット・システムに適用する

`config.cdf` ファイルをターゲット・システムに適用する際、CDF に定義されていない属性はすべて、ターゲット・システムでは変更されないか未構成になります。

必要に応じて、`config.cdf` ファイルを次のようにシステムに適用します。

- 構成クローニングを開始する正しい場所に CDF があると、フル・インストレーションでその CDF が適用されます (7.11.1 項)。
- CDF をターゲット・システムに手動で適用して、構成クローニングを開始することができます (7.11.2 項)。
- CDF を手動で適用して、構成が壊れたシステム上に構成を復元できます (7.11.3 項)。

7.11.1 フル・インストレーションの際に構成 CDF を適用する

フル・インストレーション・プロセスが 7.10.3 項 に示した場所のいずれかで `config.cdf` ファイルを見つけると、ファイル・システムが作成され、ソフトウェアがロードされ、新しくインストールされたシステム・ディスクからシステムがリブートされた後に、CDF が適用されます。新しくインストールされたシステムは CDF に定義されたとおりに正確に構成されます。

7.11.2 稼働中のシステムに構成 CDF を適用する

ディスク・ドライブに挿入されマウントされたディスクに `config.cdf` ファイルをコピーするか、またはクローニングを行うシステムの `/var/adm/smlogs` ディレクトリにコピーした後、次の構文で `sysman` コマンドを使用して構成をシステムに適用します。

sysman -clone -apply [*filename*]

フル・パスおよびファイル名を指定しない場合、`sysman` コマンドは省略時の位置およびファイル名 `/var/adm/smlogs/config.cdf` が指定されたものと見なします。`config.cdf` ファイルを `/var/adm/smlogs` 以外に置くか、またはファイル名を `config.cdf` 以外にした場合は、コマンド行でパスと正確なファイル名を指定する必要があります。

7.11.3 構成 CDF を使用してシステム構成を復元する

壊れたシステム構成を復元したり，単に config.cdf ファイルに保存された以前の構成に戻したい場合は，正しい config.cdf ファイルが /var/adm/smlogs ディレクトリにあることを確認して，次のコマンドを実行してください。

sysman -clone -apply [*filename*]

構成 CDF が /var/adm/smlogs/config.cdf とは異なる位置に保存されている場合は，コマンド行でフル・パスを指定する必要があります。

A

インストール CDF の属性と値

個々の項目内に含まれる属性とその値は、モデル・システムの最初のフル・インストールを実行する際に使用された配布メディア (CD-ROM あるいは RIS) によって異なります。ここでは、`install.cdf` ファイルの以下の項目の内容について説明します。

- `Inst_islinfo` (A.1 節)
- `Inst_filesystem` (A.2 節)
- `Inst_disklabel` (A.3 節)
- `Inst_lsm_global` (A.4 節)
- `Inst_lsm_disks` (A.5 節)
- `Inst_subsets` (A.6 節)
- `Inst_cinstall` (A.7 節)

注意

`install.cdf` ファイル内の属性と値の変更は、経験のあるシステム管理者だけが行うようにしてください。入力ミスや間違った項目への属性と値の挿入は、クローン・システムで重大な誤りが発生する原因となり、システムが使用できない状態になることがあります。

また、属性とその値にはブランク・スペースを含めることはできません。ブランク・スペースはデータ確認エラーの原因となります。ブランク・スペースはすべて削除するようにしてください (特に、行末のブランク・スペース)。属性に値を指定しない場合は、等号 (=) の後に何も入力しないように注意してください。

先頭に下線 (`_`) が付いた属性を変更あるいは削除してはなりません。このような属性 (たとえば `_action=create`) は、フル・

インストールあるいはクローン・インストール
で必要とする内部変数です。

A.1 Inst_islinfo 項目の属性

表 A-1 に示すのは、CDF の `Inst_islinfo` 項目に含まれる属性です。
`Inst_islinfo` 項目は、インストールを開始する前のシステムの状
態を伝えるために使用されます。

表 A-1: イニシャル・サブセット・ロード (`Inst_islinfo`) 項目の属性定義

属性	有効な値	説明
<code>client=</code>		変更しないでください。この属性は RIS フル・インストール (クローン・ インストールではなく) に対して のみ有効で、複製されたシステムの RIS クライアント名を指定します。RIS クラ イアント名は、サーバに対する <code>bootp</code> 要求の結果、自動的に決定されます。
<code>clone=</code>		変更しないでください。この属性 は、クローン・インストールの 結果、CDF に自動的に挿入されま す。この属性は、クローン・インス トレーションでのみ有効です。
<code>force_ccii=</code>	<code>yes</code> <code>no</code>	この属性を <code>yes</code> に設定すると、グラ フィックス機能を持つシステムでテ キスト・ベースのインストール・ユ ーザ・インタフェースが起動 されます。省略時の値は <code>no</code> で、 これは <code>no</code> に相当します。
<code>media_type=</code>	<code>REMOTE</code> <code>CDROM</code>	この属性は、現在実行しようとしてい るインストールの配布メディア のタイプを指定するために、フル・イ ンストールおよびクローン・イ ンストールで使用されます。こ の属性は、 <code>Inst_islinfo</code> 項目で欠か せない唯一の属性です。イニシャル・ インストールで使った配布メ ディアのタイプと実行しようとしてい るクローン・インストールの配 布メディアのタイプが異なる場合は、 この属性を編集してください。

表 A-1: イニシャル・サブセット・ロード (Inst_islinfo) 項目の属性定義 (続き)

属性	有効な値	説明
prompt=	yes no	<p>この属性は、クローン・インストールの開始をユーザに確認するかどうかを指定するために使用されます。</p> <p>インストレーション・インタフェースはこの属性を CDF に挿入する機能を提供しないので、この属性は手作業で CDF に追加しなければなりません。</p> <p>yes の属性値は、CDF の使用に関する確認プロンプトを表示することを指定します。no の属性値は、確認プロンプトの表示を省略して自動的に CDF を使用することを指定します。</p> <p>この属性が CDF に含まれていない場合は、省略時の値として prompt=yes が使用されます。no を設定する場合は、インストレーション・プロセスが CDF を検出するとすぐにクローン・インストールが開始されることに注意してください。配布メディアからシステムをブートしてシステム管理作業あるいはディスク・メンテナンスを実行したい場合などは、クローン・インストールがすぐに開始されると困ります。</p>
risdir=		<p>変更しないでください。この属性は RIS フル・インストールに固有の属性で、クライアント・システムが登録されているプロダクト環境のベース RIS ディレクトリが自動的に設定されます。</p>
server=		<p>変更しないでください。この属性は RIS フル・インストールおよび RIS クローン・インストールに固有の属性で、クライアント・システムが現在登録されている RIS サーバを指定します。</p>
server_locality=		<p>変更しないでください。この属性は RIS フル・インストールに固有の属性で、現在の地理的位置を指定します。この属性は、Inst_cinstall 項目の locality= 属性で制御されます。</p>

表 A-1: イニシャル・サブセット・ロード (Inst_islinfo) 項目の属性定義 (続き)

属性	有効な値	説明
server_timezone=		変更しないでください。この属性は RIS フル・インストレーションに固有の属性で、地理的なタイムゾーンを指定します。この値は RIS フル・インストレーション中に自動的に設定されます。
srcloc=	/ALPHA/BASE RISserver:	この属性は、オペレーティング・システムが内部処理のために使用し、フル・インストレーション・プロセスあるいはクローン・インストレーション・プロセスによっては使用されません。この属性はロードするソフトウェアの位置を指定します。RIS インストレーションの場合、この値はサーバ名を指定します。サーバ名の後ろにはコロン(:)を付けます。CD-ROM インストレーションの場合、この値は、ディレクトリ・パス /ALPHA/BASE になります。この属性の値は media_type 属性の値と矛盾がないようにしなければならないので、media_type 属性を変更していなければこの属性は変更しないでください。
timeset=	0 1	変更しないでください。この属性はフル・インストレーションに対して適用され、クライアント・システムの日付と時刻が正しく設定されていてインストレーション中にそれを表示できるかどうかを示しています。日付と時刻が設定されておらず、インストレーション中にそれを表示できない場合、この属性は 0 になります。日付と時刻が正しく設定されており、インストレーション中にそれを表示できる場合、この属性は 1 になります。この属性は内部インストレーション・プロセスによって制御されているため、この属性を変更するとオリジナルの設定に戻されます。

A.2 Inst_filesystem 項目の属性

表 A-2 に示すのは、CDF の Inst_filesystem 項目で使用される属性です。Inst_filesystem 項目は、複製されたシステムで作成されるファイル・システムの数とタイプに関する情報を伝えます。少なくとも、/, /usr, および /var ファイル・システムと 1 つの swap 領域の合計 4 つのファイル・システムに関する情報が記述されていなければなりません。お

勧めませんが、注意事項として記述されている状況が該当する場合を除いて、この項目の属性はすべて変更可能です。

表 A-2: ファイル・システム(`Inst_filesystem`) 項目の属性定義

属性	有効な値	説明
<code>name=</code>	<code>root</code> <code>usr</code> <code>var</code> <code>il8n</code> <code>swap1</code> <code>swap2</code>	必須。作成するファイル・システムの名前を指定します。 <code>/</code> 、 <code>usr</code> 、 <code>var</code> 、 <code>il8n</code> 、 <code>swap1</code> 、および <code>swap2</code> ごとに 1 つの項目が記述できます。
<code>file_sys- tem_type=</code>	<code>ufs</code> <code>advfs</code> <code>swap</code>	<p>必須。作成するファイル・システムのタイプを指定します。<code>name=</code> 属性の値が <code>swap1</code> あるいは <code>swap2</code> の場合、この属性の値は <code>swap</code> でなければなりません。値 <code>swap</code> は、他の <code>name=</code> 属性の値としては許可されません。</p> <p>注意: AdvFS のサポートに必要なサブセットが CDF に定義されていない場合、複製されるシステムにこのサブセットがインストールされないため、この値を <code>ufs</code> から <code>advfs</code> へ変更すると、複製されるシステムでエラーが生じることがあります。このような状況が生じると、ファイル・システムは読み取り不能な状態になります。</p> <p>インストール・プロセスによって他のファイル・システムが <code>advfs</code> に設定されていない場合、あるいは <code>Inst_subsets</code> 項目の <code>ss_names=</code> 属性に必要な AdvFS サブセットが記述されていない場合は、この値を <code>advfs</code> に変更しないでください。サブセットの内容については『インストール・ガイド』を参照してください。</p>
<code>disk_name=</code>	<code>dskn</code>	<p>必須。ファイル・システムのディスク名を指定するのに必要な属性で、<code>dsk0</code> などオペレーティング・システムが認識できる名前を指定します。この属性の値は、<code>disk_type=</code> 属性の値と矛盾してはなりません。この属性を変更する場合は、<code>disk_type=</code> 属性の値を確認する必要があります。ディスク・タイプを確認するには、次のコマンドを入力します。</p> <pre>disklabel dskn grep disk</pre>

表 A-2: ファイル・システム(**Inst_filesystem**) 項目の属性定義 (続き)

属性	有効な値	説明
		たとえば、この属性の値を <code>disk_name=dsk1</code> と変更する場合は、 <code>dsk1</code> のディスクのタイプを調べる必要があります。ディスクのタイプが <code>RZ58</code> の場合は、 <code>disk_type=</code> 属性の値を <code>RZ58</code> にします。
<code>disk_number=</code>	<code>n</code>	この属性は、フル・インストール・プロセスによって設定され、クローニング・プロセスでは使用されません。
<code>disk_type=</code>	<code>RZnn[n] [A-Z]</code> <code>RAnn</code> <code>HSZnn</code>	必須。 <code>disk_name</code> で指定したディスクのディスク・タイプ (たとえば、 <code>RZ26</code> など) を示します。この属性の値は、 <code>disk_name=</code> 属性と矛盾してはなりません。詳細については、 <code>disk_name=</code> 属性の説明を参照してください。
<code>partition=</code>	<code>[a-h]</code>	必須。ファイル・システムを作成するディスク・パーティションを指定します。有効な値は、 <code>a ~ h</code> の英字です。/ ファイル・システムは、常に <code>a</code> パーティションになければなりません。
		注意: / 以外のファイル・システムに関してこの属性の値を変更する場合は、選択するパーティションが、アクティブなファイル・システムを含んでいる他のパーティションと重ならないように注意してください。
<code>con- troller_type=</code>		この属性は、指定したディスクが接続されているコントローラのタイプを指定するための属性です。フル・インストール中、情報提供のためにこの値が自動的に提供されます。クローン・インストールの場合はこの属性は使用されず、CDF で省略することも可能です。
<code>con- troller_num- ber=</code>		この属性は、ディスクが接続されているコントローラの番号を指定します。フル・インストール中、情報提供のためにこの値が自動的に提供されます。クローン・インストールの場合はこの属性は使用されず、CDF で省略することも可能です。

A.3 Inst_disklabel 項目の属性

表 A-3 は、CDF の `Inst_disklabel` 項目の属性定義を示しています。
`Inst_disklabel` 項目は、クローン・システム上に作成される各パーティションのサイズとオフセットを伝えます。`Inst_disklabel` 項目は、ファイ

ル・システムあるいはスワップ領域をインストールする各ディスクに対して作成されます。たとえば、dsk0 上に /、dsk1 上に /usr をインストールすると、dsk0 および dsk1 に対してInst_disklabel 項目が作成されます。クローン・システム内の他のディスクはすべて、複製中に影響を受けません。

表 A-3: ディスクラベル (Inst_disklabel) 項目の属性定義

属性	有効な値	説明
name=	dskn in usr_domain in usr	必須。変更しないでください。この属性は、Inst_disklabel 項目内のサイズおよびオフセット定義に関連するディスクを指定します。Inst_disklabel 属性は、モデル・システム内で構成されている各ディスクに対して作成されます。値 in usr_domain は、ファイル・システム・タイプが AdvFS の var および i18n ディレクトリが usr_domain に含まれることを意味します。値 in usr は、ファイル・システム・タイプが UFS の var および i18n ディレクトリが /usr ファイル・システムに含まれることを意味します。
[a-h]_size=	n (ブロック)	必須。変更しないでください。この属性は、構成されているディスクの各パーティションのサイズを指定します。ファイル・システム用に使用されているかどうかにかかわらず、すべてのパーティションが定義されています。Inst_filesystem 項目は、ファイル・システムを定義します。
[a-h]_offset=	n (ブロック)	必須。変更しないでください。この属性は、構成済みのディスク上の各パーティションのオフセットを指定します。ファイル・システム用に使用されているかどうかにかかわらず、すべてのパーティションが定義されています。Inst_filesystem 項目は、ファイル・システムを定義します。

A.4 Inst_lsm_global 項目の属性

表 A-4 は、CDF の Inst_lsm_global 項目の属性定義を示しています。Inst_lsm_global 項目は、構成済みのすべてのディスク上に作成される LSM プライベート・リージョンのサイズを伝えて、システムの LSM ホスト名を定義します。Inst_lsm_global 項目は、LSM ボリュームにインストールすることを選択した場合のみ作成されます。

表 A-4: LSM グローバル (Inst_lsm_global) 項目の属性定義

属性	有効な値	説明
priv_reg_size=	n (ブロック)	変更しないでください。LSM を使用することを選択した場合にこの属性が作成されて、クライアント・システムで構成されているすべてのディスク上で作成されるプライベート・リージョンのサイズを伝えます。
lsm_hostname=	nnnnn:hh:mm:ss 既存のホスト名	この属性は、rootdg ディスク・グループを定義する際に LSM が使用するホスト名を定義します。LSM ホスト名は、同じネットワーク上に存在するクライアントに対して一意でなければなりません。既存の LSM 構成からホスト名が判断できない場合、あるいはシステム上に LSM が存在しない場合は、フル・インストレーション・プロセスは、5 桁の数の後にタイムスタンプが付加された形式で LSM ホスト名を生成します。Inst_cinstall 項目の hostname= 属性を変更する場合、システムがネットワーク上でユニークな ID を維持する一方でホスト名と LSM ホスト名が同じになるように、ここで同じ変更を行うことをお勧めします。

A.5 Inst_lsm_disks 項目の属性

表 A-5 は、CDF の Inst_lsm_disks 項目の属性定義を示しています。Inst_lsm_disks 項目は、プライベート・リージョンのパーティションおよびディスク情報を伝達します。Inst_lsm_disks 属性は、構成済みの各ディスクに対して作成されます。Inst_lsm_disks 項目は、LSM を使用することを選択した場合のみ作成されます。

表 A-5: LSM ディスク (Inst_lsm_disks) 項目の属性定義

属性	有効な値	説明
name=	dskn	変更しないでください。この属性は、LSM プライベート・リージョン・パーティション情報に関連するディスクを指定します。Inst_lsm_disks 属性は、モデル・システムで構成されている各ディスクに対して作成されます。
priv_reg_part= [a-h]		変更しないでください。この属性は、name= 属性によって定義されたディスクに対して作成される LSM プライベート・リージョンのパーティションを示します。

A.6 Inst_subsets 項目の属性

表 A-6 に示すのは、CDF の Inst_subsets 項目で使用される属性です。Inst_subsets 項目は、複製されるシステムにインストールされるベース・ソフトウェア・サブセットについての情報をクローン・インストール・プロセスに知らせるために使用されます。Inst_subsets 項目は、ベース・オペレーティング・システム・ソフトウェア用に常に 1 つ存在します。WLS ソフトウェア用に Inst_subsets 項目をさらに 1 つ用意することも可能です。

表 A-6: ソフトウェア・サブセット・ロード(Inst_subsets) 項目の属性定義

属性	有効な値	説明
advflag=	1	変更しないでください。この属性は、サブセット・ロード・プロシージャ setld ユーティリティに (アップデート・インストールではなく) フル・インストールであることを知らせます。この属性は、フル・インストール・プロセスによって常に 1 に設定されます。
name=	BASE WLS a	必須。変更しないでください。この属性は、ss_names 属性にリストされているサブセットに関連するソフトウェアの名前を指定します。インストールされている各ソフトウェアごとに name 属性が定義されています。BASE はベース・オペレーティング・システムを意味します。WLS はワールドワイド言語サポート・ソフトウェアを意味します。

表 A-6: ソフトウェア・サブセット・ロード(*Inst_subsets*) 項目の属性定義 (続き)

属性	有効な値	説明
<i>ss_names=</i>	サブセット名の一覧	必須。この属性は、インストールするベース・オペレーティング・システムおよび WLS ソフトウェア・サブセットのリストを指定するための属性です。各ソフトウェア・サブセット名はコンマ (,) で区切り、リターン・キーで改行せずに 1 行で記述しなければなりません。この属性にソフトウェア・サブセット名を追加する場合は、ディスク容量と他のソフトウェア・サブセットへの依存性を考慮する必要があります。ソフトウェア・サブセットの依存関係については『インストール・ガイド』を、サブセットのサイズ情報については『リリース・ノート』を参照してください。
<i>volume_name=</i>	DISC1 DISC2	必須。変更しないでください。この属性は、 <i>name=</i> 属性によって定義されているソフトウェアが格納されているディストリビューションを指定します。たとえば、ベース・オペレーティング・システム CD-ROM の <i>volume_name</i> の値は DISC1 になります。また、WLS ソフトウェアが含まれている CD-ROM の <i>volume_name</i> の値は DISC2 になります。

^aこの属性の実際の値は *Worldwide_Language_Support* です。スペースを節約するために WLS と記載しています。

A.7 *Inst_cinstall* 項目の属性

表 A-7 に示すのは、CDF の *Inst_cinstall* 項目で使用する属性です。*Inst_cinstall* 項目は、ターゲット・システムの構成情報をクローン・インストール・プロセスに伝えるために使用されます。インストール構成項目で指定される属性は、すべてオプションです。これらの属性の値が指定されていない場合、インストール構成のフェーズでインストール・プロセスは対話モードになり、情報の入力を要求します。

複数のクローン・システムに対して単一の CDF を使用する場合は、ホスト名、パスワードなどのシステム固有の情報は指定しないで、カーネル・オプション、地理的位置、日付、時刻などのサイト固有の情報をのみを指定するようにします。

表 A-7: インストール構成 (Inst_cinstall) 項目の属性定義

属性	有効な値	説明
hostname=		この属性は、クライアント・システムのホスト名を指定します。 同一ネットワーク上に存在する各クライアント・システムのホスト名は、一意でなければなりません。 ホスト名の選択のガイドラインについては、『インストール・ガイド』を参照してください。 RIS クローン・インストールの場合、この値は自動的にクライアント・システムのホスト名に設定されます。 CD-ROM インストールの場合、この値が正しく設定されているか、あるいは何も設定されていないことを確認してください。 値が何も設定されていない場合、インストール構成のフェーズでインストール・プロセスが対話モードになり、ホスト名の入力を要求します。
kernel_option=	mandatory all interactive	<p>この属性は、カスタム・カーネル構築を対話モードで行うか非対話モードで行うかを指定します。 ユーザの介入なしでインストール・クローニング・プロセスを実行したい場合は、この属性を interactive に設定しないでください。</p> <p>interactive カーネル構築セッションでは、カーネル・オプション・メニューが表示され、カーネルへ組み込むカーネル・オプションを選択することができます。 ターゲット・システムでのカーネルの構築では、カーネル・オプション・タイプが interactive に設定されていてオプションのカーネル構成要素をユーザが意図的に選択しない限り、モデル・システムで組み込まれていたオプションのカーネル構成要素は自動的に組み込まれません。</p> <p>mandatory を指定すると、選択したソフトウェアにとって必須のカーネル・オプションでカーネルの構築を行います。 all を指定すると、すべての必須およびオプション・カーネル構成要素を使用してカーネルの構築を行います。 この属性を mandatory あるいは all に設定すると、ユーザの介入なしにカーネルが構築されます。</p>
lang_env	i18n_intro(5) 参照	この属性は言語環境を設定します。 この属性で有効な値は LANG 環境変数の値と同じです。 省略時の設定はアメリカ英語です。

表 A-7: インストール構成 (Inst_cinstall) 項目の属性定義 (続き)

属性	有効な値	説明
locality=	/etc/zoneinfo ディレクトリ を参照	この属性は、クライアント・システムの地理的位置を指定します。RIS クローン・インストール中、この値はRIS サーバの地理的位置に自動的に設定されます。値が指定されていない場合、システム構成のフェーズでインストール・プロセスは対話モードになり、地理的位置の入力を要求します。
password=		<p>変更しないでください。この属性は、クライアント・システムの暗号化された root パスワードを指定します。この属性に値が指定されている場合は、すべてのクローン・システムとモデル・システムが root パスワードを共用することになります。値が指定されていない場合、システム構成のフェーズでインストール・プロセスは対話モードになり、パスワードの入力を要求します。このエントリは暗号化されるため、そのまま残すか、ヌルにします。値を入力する場合は、解読できないパスワードに設定します。</p> <p>ターゲット・システムのクローニングが完了したら、セキュリティ上の理由から、パスワードは変更してください。</p> <p>password= の値は暗号化されていない必要がないため、手作業でこの属性に新しい値を指定することはできません。</p>
timeset=	yes no	<p>この属性は、インストール・プロセスに対して、ターゲット・システム上の日付と時刻の状態を指定します。RIS フル・インストールあるいは RIS クローン・インストールの場合、この値は常に yes に設定されます。これは、RIS サーバ・システムの日付と時刻が設定されていることを意味します。</p> <p>この値が no に設定されている場合は、システムの日付と時刻が設定されていないことを示します。インストール・プロセスは対話モードになり、日付と時刻の入力を要求します。</p> <p>CD-ROM インストールの場合、インストール・プロセスは対話モードになり、日付と時刻の入力を要求します。</p>

表 A-7: インストール構成 (Inst_cinstall) 項目の属性定義 (続き)

属性	有効な値	説明
timezone=	/etc/zoneinfo ディレクトリ を参照	<p>この属性は、特定の地理的位置内のタイムゾーンを指定します(該当する場合)。RIS クローン・インストール中、この値は自動的に RIS サーバのタイムゾーンに設定されます。timezone の値は、locality= 属性に定義されている地理的位置に対するタイムゾーンでなければなりません。たとえば、locality=US が設定されている場合は、合衆国のタイムゾーンだけが有効です。地理的位置に対応するタイムゾーンがない場合は、この属性には値を指定しません。地理的位置に対応するタイムゾーンがないことをインストール・プロセスが検知すると、構成のフェーズでタイムゾーンの入力が必要です。</p> <p>地理的位置に対応するタイムゾーンが存在しそれが指定されていない場合は、システム構成のフェーズで対話モードになり、タイムゾーンの入力が必要です。</p>



B

ユーザ提供ファイルの例

ここでは、フル・インストレーションおよびアップデート・インストレーションの際に起動される、以下のユーザ提供ファイルのサンプルを示します。

- preinstall ファイル (B.1 節)
- update_preinstall ファイル (B.2 節)
- postload ファイル (B.3 節)
- update_postload ファイル (B.4 節)
- postreboot ファイル (B.5 節)

B.1 preinstall ファイルの例

例 B-1 に示すのは、preinstall スクリプトの例です。このスクリプトは、CD-ROM からのインストレーション・クローニングのための install.cdf ファイルにサイト固有属性を設定します。preinstall スクリプトは、フル・インストレーションでファイル・システムが作成されソフトウェア・サブセットがロードされる直前に起動されます。

例 B-1: preinstall スクリプトの例

```
#!/sbin/sh
#
# This script takes a generic install.cdf file which has
# variables defined for the attributes "hostname",
# "locality", and "timezone", and substitutes real values
# for the system being installed.
#
# This script assumes the name of the generic CDF is
# install.cdf.generic. By using this namespace, the
# installation process will not find, nor attempt to use
# this generic version. The resulting install.cdf file is saved to the
# file /var/tmp/install.cdf so that the installation process
# will use the modified version.
#
# The relevant portion of the file install.cdf.generic is:
#
#     install:
#       _item=Inst_cinstall
#       kernel_option=mandatory
#       timeset=yes
#       password=gyq\Qy7xgNJZqXk
```

例 B-1: preinstall スクリプトの例 (続き)

```
#      timezone=TIMEZONE
#      locality=LOCALITY
#      _action=create
#      hostname=HOSTNAME
#
# Set the real values for the variables

HOSTNAME=aries
LOCALITY=America
TIMEZONE=New_York

#
# Substitute for the variables
#
sed -e "s/HOSTNAME/$HOSTNAME/" -e "s/LOCALITY/$LOCALITY/" \
    -e "s/TIMEZONE/$TIMEZONE/" ./install.cdf.generic > /var/tmp/install.cdf

if [ "$?" = "0" ]
then
    #
    # If the CDF was properly created, display a success
    # message, and exit with good status.
    #
    echo "/var/tmp/install.cdf successfully created"
    exit 0
else
    #
    # The CDF could not be created successfully.
    # Cause the installation process to stop.
    #
    echo " /var/tmp/install.cdf could not be created"
    exit 1
fi
```

B.2 update_preinstall ファイルの例

例 B-2 に示すのは、update_preinstall スクリプトの例です。このスクリプトは、アップデート・インストールの分析フェーズで、初期画面が表示される直前に実行されます。

例 B-2: update_preinstall ファイルの例

```
#!/bin/sh

#
# This is a sample script that demonstrates
# the type of operations that can be performed using
# the update_preinstall script.  Creating
# backup files of any file shipped with the operating
# system product is not strictly necessary because any
# user customizations are either merged automatically or
# saved to a .PreUPD extension.
#
```

B-2 ユーザ提供ファイルの例

例 B-2: update_preinstall ファイルの例 (続き)

```
BACKUP_LIST="/etc/passwd \  
/etc/fstab \  
/etc/group"  
  
for FILE in $BACKUP_LIST  
do  
    #  
    # Save each file to a .BACKUP extension if the  
    # file exists (-f $FILE) and a backup file does  
    # not already exist (! -f $FILE.BACKUP)  
    #  
    [ -f $FILE -a ! -f $FILE.BACKUP ] &&  
    {  
        cp $FILE $FILE.BACKUP  
    }  
done
```

B.3 postload ファイルの例

例 B-3 に示すのは、postload スクリプトの例です。このスクリプトは、複数システムの構成クローニングを実行するための複数のホスト固有属性を汎用の config.cdf ファイルに設定します。このスクリプトは、ネットワーク上で各システムを認識するための各ホスト名とその IP アドレスを設定します。postload スクリプトは、フル・インストールでファイル・システムが作成されソフトウェア・サブセットがロードされた後に起動されます。

例 B-3: postload スクリプトの例

```
#!/usr/bin/posix/sh  
#  
# This is a generic postload script that expects  
# to operate on a generic configuration cdf file named "config.cdf.generic"  
# also found in the same directory. This script will  
# query the host name of the system being installed and dynamically  
# modify the hostname and IP address in the config.cdf.generic file,  
# and place the file in the /var/tmp directory as config.cdf. The  
# installation process will then find the resulting CDF file in  
# /var/tmp.  
#  
# The relevant portion of the config.cdf.generic file is:  
#  
#  
#       debug=false  
#       devName=tul  
#       hasDynamicNetAddr=false  
#       hasRCInfo=true  
#       hopCountMetric=0  
#       ifaceNum=1  
#       maxTransUnit=1500  
#       networkAddress=IPADDRESS
```

例 B-3: postload スクリプトの例 (続き)

```
#      netMask=255.255.252.0
#      operational=true
#      receiveAll=false
#      receiveMulticast=false
#      speed=not_applicable
#      systemName=HOSTNAME
#      type=ETHERNET
#      useArp=true
#
# All other host name and IP address references FOR THE CLIENT BEING
# INSTALLED should be replaced by variables in the file
# config.cdf.generic.
#

MOUNT=/mnt
RCMGR=/usr/sbin/rcmgr
#
# Must use the mount-relative version of ./etc/rc.config, because
# file systems are still mounted relative to /mnt.
#
RC_CONFIG=$MOUNT/etc/rc.config; export RC_CONFIG

#
# Define a table of hostname to IPaddress to network adapter mapping
#
    aries=0; IP[aries]=16.69.56.100; DEV[aries]=tu0
    taurus=1; IP[taurus]=16.69.56.111; DEV[taurus]=tu0
    gemini=2; IP[gemini]=16.69.56.222; DEV[gemini]=ln0
    cancer=3; IP[cancer]=16.69.56.105; DEV[cancer]=tu0
    leo=4; IP[leo]=16.69.56.123; DEV[leo]=tu0
    virgo=5; IP[virgo]=16.69.56.75; DEV[virgo]=tu0
    libra=6; IP[libra]=16.69.56.50; DEV[libra]=ln0
    scorpio=7; IP[scorpio]=16.69.56.55; DEV[scorpio]=tu0
    sagittarius=8; IP[sagittarius]=16.69.56.60; DEV[sagittarius]=tu0
    capricorn=9; IP[capricorn]=16.69.56.66; DEV[capricorn]=ln0
    aquarius=10; IP[aquarius]=16.69.56.70; DEV[aquarius]=tu0
    pisces=11; IP[pisces]=16.69.56.77; DEV[pisces]=tu0

GetIPAddress()
{
    eval host=\${$1}

    echo ${IP[host]}
}

GetDevName()
{
    eval host=\${$1}

    echo ${DEV[host]}
}

Main()
{
    #
```

B-4 ユーザ提供ファイルの例

例 B-3: postload スクリプトの例 (続き)

```
# Get the host name of the system being installed. Use
# the host name to index into a table of IP addresses, and pull
# the correct IP address for this system. Then, dynamically
# update the host name and IP address in the config.cdf file.
#
HOSTNAME=`$RCMGR get HOSTNAME`

IPADDRESS=`GetIPAddress $HOSTNAME`

echo "Host is $HOSTNAME; IP is $IPADDRESS"

#
# Now modify the version of config.cdf.generic that exists in the
# current working directory, and copy it to the /var/tmp
# directory so that it is found by the installation process.
#
sed -e "s/$HOSTNAME/$HOSTNAME/g" -e "s/$IPADDRESS/$IPADDRESS/g" \
    ./config.cdf.generic > /var/tmp/config.cdf

[ -s /var/tmp/config.cdf ] &&
    echo "/var/tmp/config.cdf successfully created"

#
# Always exit with good status; the process is too far
# to exit the installation at this point.
#
exit 0
}

Main "$@"
```

B.4 update_postload ファイルの例

例 B-4 に示すのは、update_postload スクリプトの例です。このスクリプトは、アップデート・インストール中に残されたすべての *.PreUPD および *.PreMRG ファイルをアーカイブします。この処理は、アップデート処理中に行われることの例の 1 つです。update_postload スクリプトを使用する代りに、アップデート・インストール・クリーンアップ・アプリケーション (/usr/sbin/updadmin) を使用して、この種のファイルをアーカイブすることをお勧めします。update_postload スクリプトは、オペレーティング・システム・ソフトウェアがロードされた後、最初のリブートの前に実行されます。

例 B-4: update_postload ファイルの例

```
#!/bin/ksh
#
LOGDIR=/var/adm/smlogs
BACKUPDIR=/mybackups
PREMRG_FILE=$LOGDIR/upd_PremRG_files
PREUPD_FILE=$LOGDIR/upd_custom_files

if [ ! -d $BACKUPDIR ]
then
    mkdir -p $BACKUPDIR
fi

cd /
for FILE in `cat $PREMRG_FILE $PREUPD_FILE`
do
    cp $FILE $BACKUPDIR
done
```

B.5 postreboot ファイルの例

例 B-5 に示すのは、postreboot スクリプトの例です。このスクリプトは、RIS サーバからの追加のソフトウェア・サブセットのロード、/etc/fstab ファイルへのエントリ追加、.rhosts ファイルへのユーザの追加、リモートからの root ログインの許可などを行います。postreboot スクリプトは、フル・インストールの後のシステム・リブート後に起動されます。

例 B-5: postreboot ファイルの例

```
#!/usr/bin/posix/sh
#
# This script is executed during the c-install phase of the
# full installation process. At the time of execution, all network
# services will be available assuming that the system was configured
# using the 'sysman -clone' capability.
#

echo "Executing postreboot script"
#
# Load reference page software subsets from the
# RIS server.
#
SERVER=`rcmgr get INST_SERVER`

2>&1 /usr/sbin/setld -l $SERVER: OSFMANOS540 OSFCDEMANOS540 \
OSFMANWOS540 OSFCDEMANOP540 OSFMANWOP540 OSFDCMTEXT540

#
# Add an entry to the /etc/fstab file to provide an
# NFS-mount from an exporting NFS-server
#
cp /etc/fstab /etc/fstab.ORIG
echo "/mnt1@giants /nfs-mount nfs ro,bg 0 0" >> /etc/fstab
```

B-6 ユーザ提供ファイルの例

例 B-5: postreboot ファイルの例 (続き)

```
#
# Make a local mount-relative directory on which to mount
# the NFS-mounted directory
#
mkdir -p /nfs-mount

#
# Add the Engineering team members to the .rhosts file on this system.
#

cat <> /.rhosts
aries.company.com jsmith
aries jsmith
libra.company.com mwang
libra mwang
virgo.company.com rhurley
virgo rhurley
leo.company.com jcruz
leo jcruz
taurus.company.com gwilliams
taurus gwilliams
$(hostname)
EOF

#
# Allow root logins from remote systems.
#
echo ptys >> /etc/securettys

#
# Use the Korn shell as the default shell for root.
#
sed "s@/bin/sh@/bin/ksh@" /etc/passwd > /tmp/passwd
[[ -s /tmp/passwd ]] && mv /tmp/passwd /etc/passwd
chmod 644 /etc/passwd

#
# Make changes to .profile to change default editor to vi
#
echo "EDITOR=vi; export EDITOR" >> /.profile

#
# Additions to sysconfigtab that help our testing.
#
cp /etc/sysconfigtab /etc/sysconfigtab.ORIG
cat <> /etc/sysconfigtab
streams:
    nstrpush=15

advfs:
    AdvfsPanicLevel=1

proc:
    maxusers=1024
    max-proc-per-user=256
    max-threads-per-user=512

vfs:
    revoke_tty_only=0
```

例 B-5: postreboot ファイルの例 (続き)

```
kdebug:
    kdebug_escape = iseeme
    kdebug_stop_on_panic = 0
EOF

    exit 0
```

A

- action=create** 属性..... A-1
- Advanced File System**
(AdvFS を参照)
- AdvFS**
 - var および usr ファイル・システム
のリストア 3-12
 - 定義 4-3
 - ファイル・システム・タイプの選
択 4-3
 - ファイル・システムのリストア 3-9

B

- bootp** プロトコル
 - ネットワーク経由 1-6
- boot** コマンド・シーケンス
 - ネットワーク経由でのフル・インス
トレーションの開始 1-5

C

- CD-ROM**
 - CDF とユーザ提供ファイルのコ
ピー 5-23
 - WLS インストレーション・プロ
シージャ 2-5
 - イメージの書き込み..... 5-24

- ユーザ提供ファイルと CDF があ
る 5-22, 5-23
- CD-ROM** イメージの書き込み . 5-24
- CDF**..... 5-3, 6-4
(config.cdf ファイル; install.cdf
ファイル も参照)
 - CD-ROM へのコピー 5-23
 - RIS サーバへのコピー 5-21
 - /var/tmp ディレクトリへのコ
ピー 5-22
 - 概要 5-2
 - 検索順序..... 5-3
 - 正しい位置への移動..... 5-19
 - ディスクットへのコピー 5-20
 - 動作原理..... 5-7
 - 複製時に動的に変更..... 5-22
 - 命名規則..... 5-6
- CDFMODE** 属性..... 7-12
 - 検証エラー..... 7-13
- componentid** グループ..... 7-5
- config.cdf** ファイル 7-1
(構成のクローニング も参照)
 - CDFMODE 属性..... 7-12
 - componentid グループ 7-5
 - 検証エラー..... 7-14
 - 作成 7-9
 - システムに適用する..... 7-19
 - 使用可能な位置 7-3

正しい位置にコピー..... 7-15
チェックサム 7-9
フォーマットと内容..... 7-4
変更 7-10
ホストおよびサイトに固有の属性 7-11
例..... 7-5
config.cdf ファイルの検証..... 7-14

D

disklabel コマンド
UNIX シェルの使用..... 3-7
ディスク・パーティション・サイズ
の変更..... 3-15
パーティション・サイズの確認 4-10
DMS
ディスク・スペースの要件.... 4-7
ディスク・スペース要件 4-9
DNS サーバ
クローニング 7-4t

E

EISA バス
FDDI からのシステムのブート 1-8
Extended Memory Interface
(XMI バス を参照)

F

FDDI
ネットワーク・インタフェース 1-8
ブート手順..... 1-8
Fiber Distributed Data Interface

(FDDI を参照)

ftp

による config.cdf ファイルのコピー 7-16

H

halt コマンド 3-18

I

i18n
定義 2-2
i18n インストール
(WLS を参照)
i18n ディレクトリ
作成 2-7
init コマンド..... 1-9
Inst_cinstall 項目 A-10
Inst_disklabel 項目 A-6
Inst_filesystem 項目 A-5
Inst_global 項目..... A-7
Inst_islinfo 項目 A-2
Inst_subsets 項目 A-9
install.cdf ファイル 6-1
(インストールのクローニング も参照)
action=create 属性 A-1
Ins_islinfo 項目 A-2
Inst_cinstall 項目 A-10
Inst_disklabel 項目 A-6
Inst_filesystem 項目 A-5
Inst_global 項目..... A-7
Inst_subsets 項目 A-9
確認エラー..... 6-15
カーネル構築のタイプ設定... 6-17

クローン・システムで許容される相
違点 6-10
作成 6-9
説明 6-4
属性と値のペア 6-4
属性と値のペアの定義 A-1
正しい位置にコピーする 6-22
地域およびロケーションの属
性 6-19
ヌル値 A-1
パスワード属性 6-19
日付および時刻の属性 6-19
ファイル内の空白 6-21
フォーマットと内容 6-4
変更 6-15
編集時の注意 A-1
ホストおよびサイトに固有の属
性 6-19
モデル・システムからの生成 6-9
許される位置 6-3
例 6-5
installupdate コマンド
ネットワーク・サーバからの開
始 1-3
ISA バス
グラフィック・デバイスがあるシス
テムのリブート 1-9
/isl ディレクトリ 5-23

M

MULE
WLS インストール 2-8

N

NEMACS
WLS インストール 2-8

P

PCI バス
FDDI からのシステムのブート 1-8
Peripheral Component
Interconnect
PCI 1-8
ping コマンド 1-2
postload ファイル
作成 5-16
サンプル・スクリプト B-1
正しい位置への移動 5-19
postreboot ファイル
作成 5-18
サンプル・スクリプト B-1
正しい位置への移動 5-19
preinstall ファイル
作成 5-14
サンプル・スクリプト B-1
正しい位置への移動 5-19

R

restart コマンド 3-18
nogui 3-18
RIS
スペース要件 4-8
必要なディスク・スペース 4-7
RIS サーバ

CDF とユーザ提供ファイルのコ
ピー 5-21
WLS インストレーションの起
動 2-5
アップデート・インストレーシ
ョンの開始 1-3
フル・インストレーションの開
始 1-5
プロファイル・セット・ディレク
ト 5-21
root ファイル・システム
UNIX シェルでの構築 3-7
破損時のリストア 3-6

S

setld コマンド
WLS インストレーション 2-5
syslog ファイル
ディスク・スペースの要件... 4-19
sysman コマンド
構成を複製する 7-8

T

TERM 環境変数 3-4
turbochannel バス
FDDI からのシステムのブート 1-8

U

UFS
定義 4-4
ファイル・システム・タイプとして
選択 4-4

ファイル・システムのリストア 3-7

UNIX シェル

AdvFS ファイル・システムのリス
トア 3-9
UFS ファイル・システムのリス
ト 3-7
インストレーションの再起動 3-18
起動 3-1
終了 3-18
スワップ領域の作成 3-4
タスクの例 3-3
定義 3-1
ディスク・パーティションの変
更 3-15
テキスト・エディタの設定 3-4
テキスト・ベース・インタフェース
の再起動 3-18
ファイル・システムのマウント 3-5
ファイル・システムのリストア 3-6
メモリ・ファイル・システム . 3-3

UNIX ファイル・システム

(UFS を参照)

unknown host メッセージ 1-2

update_postload ファイル

サンプル・スクリプト B-1

update_preinstall ファイル

サンプル・スクリプト B-1

/usr ファイル・システム

AdvFS のリストア 3-12
定義 4-14
内容 4-14

UTX デバイス

WLS インストレーション 2-15

V

/var/adm/ris ディレクトリ.....	4-8
/var/adm ディレクトリ	
システム課金	4-20
/var/tmp ディレクトリ	
CDF とユーザ提供ファイルのコ	
ピー	5-22
/var ファイル・システム	
クラッシュ・ダンプ領域	4-17
var 領域	
AdvFS のリストア	3-12
スペースの計画	4-17
vi テキスト・エディタ	3-4
vmunix	
サイズ	4-17

W

WLS インストール	
CD-ROM から実行	2-5
i18n ディレクトリの作成	2-7
MULE	2-8
NEMACS	2-8
RIS サーバから	2-6
RIS サーバからの開始	2-5
setld コマンドの使用	2-5
UTX デバイス	2-15
Wnn	2-8
wwconfig コマンド	2-15
アウトライン・フォント	2-8
アジア言語のターミナル・ドライバ	
およびデーモン	2-15
開始	2-4

カーネル構築の失敗	2-14
カーネル構築のログ・ファイ	
ル	2-14
カーネルの構築	2-13
サポートされている端末およびプリ	
ンタ	2-2
サポートされる国	2-7
システムの準備	2-2
ソフトウェア・サブセットの構	
成	2-12
ソフトウェア・サブセットの選	
択	2-9
ソフトウェア・サブセットのロー	
ド	2-11
端末コードセット	2-17
ディスク・スペースの拡大	2-3
ディスク・スペース不足	2-6
バックアップ・ファイル	2-18
パブリック・ドメイン・ソース・	
ファイル	2-8
ログイン画面の言語オプション	2-13

Wnn

WLS インストール	2-8
wwconfig コマンド	2-15
wwinstall スクリプト	
CD-ROM からの起動	2-5

X

XMI	1-8
XMI バス	
FDDI を使用した場合のブート	1-8
X サーバ	

WLS インストール後の再スタート.....	2-13
------------------------	------

あ

アウトライン・フォント	
WLS インストール.....	2-8
アジア言語のターミナル・ドライバおよびデーモン	
WLS インストール.....	2-15
アップデート・インストール	
CDF とユーザ提供ファイルの関係	
係.....	5-5
RIS サーバからの起動.....	1-3
update_postload スクリプトの例.....	B-1
update_preinstall スクリプトの例.....	B-1
ネットワーク経由の起動.....	1-3
ユーザ提供ファイル名.....	5-3

い

インストール	
UNIX シェルからの再起動...	3-18
ワールドワイド言語サポート...	2-1
インストール CDF	
項目と属性.....	6-4
インストールのクローニング	
グ.....	6-1
(install.cdf ファイル も参照)	
CD-ROM から.....	6-19
CD-ROM へのファイルのコピー.....	5-23
CDF 変更時の注意.....	A-1
install.cdf ファイルの位置.....	6-3

install.cdf ファイルの変更 ...	6-15
install.cdf ファイルを正しい位置にコピーする	6-22
postload ファイルの作成	5-16
postreboot ファイルの作成 ...	5-18
preinstall ファイルの作成....	5-14
RIS サーバへのファイルのコピー.....	5-21
/var/tmp ディレクトリへのファイルのコピー	5-22
開始	6-23
開始時の失敗	6-23
確認エラー.....	6-15
確認属性の設定	6-16
管理者の作業	5-5
キーボード・タイプの違い...	6-14
グラフィックス・アダプタの違い.....	6-11
自動	6-16
自動カーネル構築.....	6-17
属性と値のペアの定義	A-1
正しい CDF の選択	6-9
ターゲット・システム	6-1
定義	6-1
ディスク構成の相違点	6-10
ディスケットへのファイルのコピー.....	5-20
手順	6-8
動作原理.....	5-7
ハードウェアの制限.....	6-2
フォント・サイズの違い.....	6-13
ホストおよびサイトに固有の属性	6-19
無効な属性値のエラー	6-21
モデル・システム.....	6-1

モデル・システムからの CDF の生
成 6-9
ユーザ提供ファイルの呼び出
し 6-22
ユーザの介入を不要にする... 6-16

え

エラー
インストールのクローニング
中 6-21
エラー・ロガー・ファイル
ディスク・スペースの要件... 4-19

お

オーバヘッド
ディスク・パーティション... 4-13

か

カーネル
vmunix 4-17
アジア言語のターミナル・ドライバ
およびデーモン 2-15
ターゲット・システムでのカーネル
の構築 6-17
カーネル・オプション
install.cdf ファイル内 6-17
カーネルの構築
install.cdf ファイルでの属性設
定 6-17
WLS インストール 2-13

WLS インストール後の失
敗 2-14
アジア言語用カーネル 2-15

き

キーボード・タイプ
CDF とクローン・システムの違
い 6-14

く

国
WLS インストールでサポー
トされる 2-7
クラスタ
クローニングの制限 7-6
ディスク・スペースの計画... 4-2
ファイル・システム・タイプ . 4-3
ファイル・システム・タイプの選
択 4-3
クラッシュ・ダンプ 4-21
ディスク・スペースの計画... 4-17
グラフィカル・ユーザ・インタフェー
ス
UNIX シェルからの再起動 ... 3-18
グラフィックス・アダプタ
CDF とクローン・システムの違
い 6-11
クローニング
DNS サーバ 7-4t
クローン・インストール.. 6-1

け

言語

- WLS インストールでサポートされる 2-7
- インストールする 2-1
- デスクトップ用の設定 2-19

言語オプション

- ログイン画面 2-13

検証エラー

- config.cdf ファイル 7-13

こ

構成 CDF

- (config.cdf ファイル を参照)

構成記述ファイル

- (CDF を参照)

- 構成のクローニング 7-1

- (config.cdf ファイル も参照)

CD-ROM へのファイルのコ

- ピー 5-23

- CDF の変更 7-10

- CDF をターゲット・システムに適
用する 7-19

- config.cdf ファイル内のチェックサ
ム 7-9

- config.cdf ファイルの位置 7-3

- config.cdf を稼働中のシステムに適
用する 7-19

- config.cdf を正しい位置にコ
ピー 7-15

- postload ファイルの作成 5-16

- postreboot ファイルの作成 ... 5-18

RIS サーバへのファイルのコ

- ピー 5-21

/var/tmp ディレクトリへのファイル

- のコピー 5-22

- インストールのクローニング
との併用 7-2

- 管理者の作業 5-5

- 検証 7-14

- 検証エラー 7-13

- 処理 7-2

- 制限 7-6

- ターゲット・システム 7-1

- 定義 7-1

ディスキットへのファイルのコ

- ピー 5-20

- データを CDF に保存 7-9

- 動作原理 5-7

- ネットワーク・アダプタ 7-11

- ネットワーク上での一意性を維
持 7-11

- ネットワークに接続されていない

- ターゲット・システム 7-16

- 複数の ID を持つレコード 7-12

- プロシージャの要約 7-8

- ホストおよびサイトに固有の属性の
変更 7-11

- モデル・システム 7-1

構成の復元

- config.cdf ファイルを使用 7-1

構築

- UNIX シェルでの新しい root ファイ
ル・システム 3-7

コマンド

- installupdate 1-3

- sysman -clone -save 7-8

さ

サイズ

- /usr ファイル・システム 4-14
- var 領域 4-17
- クラッシュ・ダンプ・パーティション 4-21

し

システム・アクティビティ

- ログ 4-19
- システム・アクティビティのログ 4-19
- システム課金
 - スペース 4-20
- システム・クラッシュ 4-21
- システム構成
 - config.cdf ファイルで復元 7-1
- 終了
 - UNIX シェル 3-18
- 省略時のパーティション・テーブル
 - 変更 3-15
- シンボリック・リンクされた **RIS** 領域 4-8

す

スクリプト

- ユーザ提供スクリプトの例 B-1
- スワップ領域
 - UNIX シェルでの作成 3-4
 - 計画 4-20
 - 最小サイズ 4-20

スーパーユーザ

- になる方法 1-3

そ

属性と値のペア

- install.cdf ファイル A-1
- 変更時の注意 A-1

ソフトウェア・サブセット

- /usr ファイル・システムに必要なスペース 4-15
- WLS インストレーションでの構成 2-12
- WLS インストレーションでの選択 2-9
- WLS インストレーションでのロード 2-11
- 定義 4-15
- 必要なディスク・スペース... 4-12

た

ダンプ

- フルまたは部分 4-21

端末

- WLS のサポート 2-2

端末コードセット

- WLS インストレーション 2-17

ターゲット・システム

- インストレーションのクローニング
 - グ 6-1
 - 構成のクローニング 7-1

ち

- チェックサム
- config.cdf ファイル..... 7-9

て

- ディスク
 - /usr ファイル・システム用... 4-14
 - ディスク・パーティション・サイズの変更..... 3-15
 - ユーザ・ファイルのスペース割り当て..... 4-16
- ディスク構成アプリケーション. 4-10
- ディスク・スペース
 - DMS サーバ 4-9
 - RIS サーバと DMS サーバ 4-7
 - /usr ファイル・システム 4-14
 - WLS 4-8
 - WLS インストレーションで不足 2-6
 - WLS ソフトウェア用..... 2-3
 - クラスタ..... 4-2
 - 計画 4-1
 - シンボリック・リンクされた RIS 領域 4-8
 - ソフトウェアが必要とする... 4-12
 - 手作業での計画 4-7
 - ファイル・システムのオーバーヘッド 4-13
 - フル・インストレーションの計画機能 4-5
 - ユーザ作成のファイル 4-16
 - レイヤード・プロダクト 4-7
- ディスク・スペースの計画
 - RIS サーバ..... 4-7

- WLS 4-8
- クラッシュ・ダンプ..... 4-17
- 考慮事項..... 4-10
- システム課金ファイル 4-20
- レイヤード・プロダクト 4-7
- ログ・ファイル 4-19
- ディスク・パーティション
 - 既存のサイズ 4-10
 - サイズのカスタマイズ 4-10
- ディスケット
 - CDF とユーザ提供ファイルのコピー 5-20
- ディレクトリ
 - /isl 5-23
 - /usr ファイル・システム 4-14
 - /var/adm..... 4-20
 - /var/adm/ris 4-8
 - /var/tmp 5-22
 - プロファイル・セット 5-21
- テキスト・エディタ
 - UNIX シェルでの設定 3-4
 - vi..... 3-4
- テキスト・ベース・インタフェース
 - UNIX シェルからの再起動 ... 3-18
- テキスト・ベース・コマンド
 - UNIX シェルからの再起動 ... 3-18
- データレス管理サービス
 - (DMS を参照)

ね

- ネットワーク
 - ISA バスのグラフィックス・デバイスがあるシステムのリポート 1-9
 - WLS インストレーションの起動 2-5

構成のクローニング.....	7-16
接続の確認.....	1-2
フル・インストールの開	
始	1-5
ブート・デバイス.....	1-6
ブートの失敗	1-9
ネットワーク・アダプタ	
確認	1-6, 1-8
調査	7-11t
ネットワーク・サーバ	
アップデート・インストール	
の開始.....	1-3
フル・インストールの開	
始	1-5
ネットワーク接続	
確認	1-2

は

配布メディア	
config.cdf ファイルをコピーす	
る	7-18
フォーマット	5-13
パッチ	
クローニング用のモデル・システム	
にインストールされた	7-6
パブリック・ドメイン・ソース・ファ	
イル	
WLS インストール	2-8
パーティション	
/usr ファイル・システム	4-14
クラッシュ・ダンプ.....	4-21
省略時のスワップ領域	4-20

ふ

ファイル	
postload ファイルの作成	5-16
postreboot ファイルの作成 ...	5-18
preinstall ファイルの作成....	5-14
/var/adm/smlogs/it.log	2-14
vmunix	4-17
WLS インストールでのパッ	
クアップ	2-18
インストール処理のカスタマ	
イズ	5-1
ユーザ作成.....	4-16
ユーザ提供スクリプトの例....	B-1
ファイル・システム	
UNIX シェルでのマウント	3-5
UNIX シェルでのリストア	3-6
/usr に必要なディスク・スパー	
ス	4-14
/var の内容.....	4-17
オーバーヘッド	4-13
内容	4-10
ユーザ作成ファイル用	4-16
ファイル・システム・タイプ	
AdvFS	4-3
UFS.....	4-4
省略時	4-5
定義	4-2
ファイル・システムの破損	
回復	3-1
ファイル・システム・レイアウト	
カスタマイズ	4-10
省略時	4-5
フォント・サイズ	

CDF とクローン・システムの違 い	6-13	ネットワーク・ブート・コマン ド	1-5
複製したシステム構成	7-1	ファイル・システム・タイプの選 択	4-2
部分ダンプ スペース	4-21	ユーザ提供ファイルと CDF の位 置	5-19
プリンタ WLS のサポート	2-2	ユーザ提供ファイルのガイドライ ン	5-13
フル・インストレーション CDF とユーザ提供ファイルの関 係	5-5	ユーザ提供ファイル名	5-3
CDF の検索順序	5-3	フル・ダンプ スペース	4-21
config.cdf ファイルの位置	7-3	プロファイル・セット・ディレクト リ	5-21
config.cdf ファイルをターゲット・ システムに適用する	7-19	ブート FDDI を使用するシステム	1-8
disklabel コマンドの起動	4-10	ブートの失敗 ネットワーク経由	1-9
install.cdf ファイルの位置	6-3		
postload ファイルの作成	5-16		
postreboot ファイルの作成	5-18		
preinstall ファイルの作成	5-14		
/usr ファイル・システムのスペース の計画	4-14		
インストレーションのクローニング の起動	6-1		
構成のクローニングの起動	7-1		
自動ディスク計画機能	4-5		
ソフトウェアのサイズ	4-12		
ターゲット・システム上でのクロー ン・インストレーションの開 始	6-23		
ディスク構成アプリケーションの起 動	4-10		
ディスク・パーティションの変 更	4-10		
ネットワーク経由のブート	1-5		

ま

マウント UNIX シェルからのファイル・シス テム	3-5
/usr 下のユーザ・ファイル・システ ム	4-16

む

無効な属性値のエラー	6-21
------------------	------

め

メモリ 容量の調査	4-18
--------------------	------

も

- モデル・システム
 - インストールのクローニング 6-1, 6-9
 - 構成のクローニング 7-1

ゆ

- ユーザ・アカウント
 - ディスク・スペース 4-16
- ユーザ提供ファイル
 - CD-ROM へのコピー 5-23
 - CDF との関係 5-5
 - postload ファイルの作成 5-16
 - postreboot ファイルの作成 ... 5-18
 - preinstall ファイルの作成 5-14
 - RIS サーバへのコピー 5-21
 - /var/tmp ディレクトリへのコピー 5-22
 - 一般的な使用目的 5-4
 - インストールのクローニングの際に使用 6-22
 - 概要 5-3
 - 管理者の作業 5-5
 - サンプル・スクリプト B-1
 - 正しい位置への移動 5-19
 - ディスクットへのコピー 5-20
 - 動作原理 5-7
 - 内容と制限 5-13

ら

- ランダム・アクセス・メモリ

- 容量の調査 4-18

り

- リストア
 - AdvFS var 領域 3-12
 - AdvFS ファイル・システム ... 3-9
 - UFS ファイル・システム 3-7
 - UNIX シェルにおけるファイル・システム 3-6
 - 破損した root ファイル・システム 3-6
- リブート
 - ISA バスのグラフィックス・デバイスがあるシステム 1-9

る

- ルート・ファイル・システム
 - 破損のリカバリ 3-1

れ

- レイヤード・プロダクト
 - ディスク・スペースの要件 4-7

ろ

- ログイン画面
 - 言語オプションの設定 2-13
- ログ・ファイル
 - syslog 4-19

わ

割り当て

/usr ファイル・システムのスペース 4-14
/var ファイル・システム 4-17
スワップ領域 4-20

ユーザ作成ファイルのスペース

..... 4-16

ワールドワイド・インストレーション

(WLS を参照)

ワールドワイド言語サポート

(WLS を参照)

Tru64 UNIX ドキュメントの購入方法

Tru64 UNIX ドキュメントのご購入については、弊社担当営業または日本ヒューレット・パッカートの各営業所/代理店にお問い合わせください。

各ドキュメント・キットの注文番号は以下のとおりです。ドキュメント・キットに含まれるマニュアルの内容については『ドキュメント概要』を参照してください。

キット名	注文番号
Tru64 UNIX Documentation CD-ROM	QA-6ADAA-G8
Tru64 UNIX Documentation Kit	QA-6ADAA-GZ
End User Documentation Kit	QA-6ADAB-GZ
- Startup Documentation Kit	QA-6ADAC-GZ
- General User Documentation Kit	QA-6ADAD-GZ
- System and Network Management Documentation Kit	QA-6ADAE-GZ
Developer's Documentation Kit	QA-6ADAF-GZ
Reference Pages Documentation Kit	QA-6ADAG-GZ
TruCluster Server Documentation Kit	QA-6BRAA-GZ
Tru64 UNIX 日本語ドキュメント・キット	QA-6ADJB-GZ
スタートアップ・ドキュメント・キット	QA-6ADJC-GZ
一般ユーザ・ドキュメント・キット	QA-6ADJD-GZ
システム/ネットワーク管理ドキュメント・キット	QA-6ADJE-GZ
プログラミング・ドキュメント・キット	QA-6ADJF-GZ
CDE 翻訳ドキュメント・キット	QA-6ADJG-GZ
TruCluster Server 日本語ドキュメント・キット	QA-05SJA-GZ
Advanced Server for UNIX 日本語ドキュメント・キット	QA-5U2JA-GZ



マニュアルに対するご意見

Tru64 UNIX

インストレーション・ガイド ― 上級ユーザ編

AA-RK3BE-TE

弊社のマニュアルに関して、ご意見、ご要望、または内容の不明確な部分など、お気づきの点がございましたら、下記にご記入の上、弊社社員にお渡しくださるようお願い申し上げます。

マニュアルの採点：

	大変良い	良い	普通	良くない
正確さ (説明どおりに動作するか)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
情報量 (十分か)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
分かり易さ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
マニュアルの構成	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
図 (役立つか)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
例 (役立つか)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
索引 (項目の検索性)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ページ・レイアウト (情報の検索性)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

内容の不明確な部分がありましたら、以下にご記入ください：

ペ ー ジ

その他お気づきの点がございましたら、以下にご記入ください：

ご使用のソフトウェアのバージョン： _____

貴社名/部課名 _____

御名前 _____

記入日 _____

(注) 当用紙を受け取った弊社社員は、すみやかに下記にお送りください。

ビジネスクリティカルシステム統括本部 **BCS** 技術本部 **Alpha** ソフトウェア技術部