

日本語 Advanced Printing Software

システム管理/操作ガイド

2001 年 9 月

ソフトウェア・バージョン: 日本語 Advanced Printing Software バージョン 1.1

オペレーティング・システム: Tru64 UNIX バージョン 4.0F 以降

本書は、Tru64 UNIX オペレーティング・システムが稼働しているシステムで、日本語 Advanced Printing Software システムを構成し、管理する方法について説明します。

© 2001 Compaq Computer K.K.

本書の著作権はコンパックコンピュータ株式会社が保有しており、本書中の解説および図、表はコンパックコンピュータ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。

コンパックコンピュータは、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。

COMPAQ, Compaq ロゴ, Digital ロゴは U.S. Patent and Trademark Office に登録されています。以下は、Digital Equipment Corporation の商標です: ALL-IN-1, Alpha AXP, AlphaGeneration, AlphaServer, AltaVista, ATMworks, AXP, Bookreader, CDA, DDIS, DEC, DEC Ada, DECEvent, DEC Fortran, DEC FUSE, DECnet, DECstation, DECsystem, DECterm, DECUS, DECwindows, DTIF, Massbus, MicroVAX, OpenVMS, POLYCENTER, PrintServer, Q-bus, StorageWorks, Tru64, TruCluster, TURBOchannel, ULTRIX, ULTRIX Mail Connection, ULTRIX Worksystem Software, UNIBUS, VAX, VAXstation, VMS, XUI。このドキュメントに記載されているその他の会社名および製品名は、各社の商標または登録商標です。

PrintXchange は米国 Xerox 社の商標です。

Advanced Printing Software は、Xerox 社の PrintXchange 技術に基づいています。

まえがき

1 Advanced Printing Software の概要

1.1	概念および用語	1-1
1.1.1	クライアントおよびサーバ	1-1
1.1.1.1	クライアント・プログラム	1-2
1.1.1.2	サーバ・プログラム	1-2
1.1.2	オブジェクトおよび属性	1-3
1.1.3	操作	1-4
1.1.4	論理プリンタ, 物理プリンタ, およびキュー	1-5
1.1.5	ジョブおよびドキュメント	1-5
1.1.6	初期値オブジェクト	1-6
1.1.7	supported 属性および ready 属性	1-6
1.1.8	ネーム・スペースおよびネーム・サービス	1-8
1.1.9	サーバ・データベース	1-8
1.1.10	ユーザ・タイプおよびアクセス制御	1-8
1.1.11	通知およびイベント	1-9
1.2	コンポーネント	1-9
1.2.1	クライアント・プログラム	1-9
1.2.1.1	エンド・ユーザ・プログラム	1-10
1.2.1.2	管理プログラム	1-10
1.2.2	サーバ・プログラム	1-10
1.2.2.1	スーパーラ	1-10
1.2.2.2	スーパーバイザ	1-11
1.2.2.3	LPD アウトバウンド・ゲートウェイ・スーパーバイザ	1-11

1.2.3	サーバ・デーモン	1-12
1.2.4	プリンタ属性ファイル	1-12
1.2.5	翻訳フィルタおよび変更フィルタ	1-13
2	インストレーションの計画	
2.1	論理構成	2-1
2.1.1	プリンタの構成	2-2
2.1.2	複数サーバ・ホスト	2-2
2.1.3	サーバの制限事項	2-2
2.1.4	論理プリンタから物理プリンタへの関連付け	2-2
2.1.5	緊密な関連付けと疎遠な関連付け	2-2
2.1.6	疎遠な関連付け, ファン・イン・キューとファン・アウト・キュー	2-3
2.1.7	単一ホスト上での複数のスーパバイザの使用	2-3
2.1.8	直接接続されたプリンタ	2-4
2.2	ネーミング・サービスの計画	2-4
2.3	通知	2-4
2.4	セキュリティ	2-5
2.5	プリント・システム・ソフトウェアの構成	2-5
3	ネーム・スペースの管理	
3.1	ローカル・ファイル・ネーム・サービス	3-1
3.2	ネットワーク情報サービス	3-3
3.3	LDAP ネーム・サービス	3-6
3.3.1	apx.conf ファイルの作成と編集	3-6
3.3.2	LDAP クライアントのユーザ名とパスワードの作成	3-7
3.4	プロトサーバ	3-8
3.5	TruCluster Server 環境における高可用性アプリケーションとしての Advanced Printing Software	3-8
3.5.1	リソース・プロファイル	3-9

3.5.2	処理スクリプト	3-9
3.5.3	リソースの登録	3-9
3.5.4	リソースの起動と停止	3-10
3.5.5	apx-default リソースへのサーバの追加	3-10
3.5.6	Advanced Printing クラスタ環境のカスタマイズ	3-10
3.5.6.1	リソースの再配置	3-10
3.5.7	1つのメンバ上で実行するプリンタの構成	3-11
3.5.8	クラスタにおけるLPD インバウンド・ゲートウェイ	3-11
4	セキュリティの管理	
4.1	システム・セキュリティ	4-1
4.2	許可	4-2
4.2.1	エンド・ユーザの操作	4-2
4.2.2	オペレータの操作	4-3
4.2.3	管理者	4-3
4.3	アクセス制御リスト (ACL) による認証	4-4
4.3.1	ACL の作成	4-5
5	サーバの作成と管理	
5.1	スーパーバイザとスプーラの作成	5-1
5.1.1	オブジェクト・データベースの作成	5-1
5.1.2	オブジェクト・データベース・データの表示	5-3
5.1.3	既存のオブジェクト・データベースの変更	5-4
5.1.4	オブジェクト・データベースのバックアップ	5-5
5.2	サーバ・オブジェクト属性の構成	5-6
5.2.1	サーバ属性の設定例	5-7
5.3	TruCluster 環境におけるサーバの構成	5-8
5.3.1	リソース・プロファイル	5-8
5.3.2	処理スクリプト	5-9

5.3.3	リソースの登録	5-9
5.3.4	リソースの起動と停止	5-9
5.3.5	apx-default リソースへのサーバの追加	5-10
5.3.6	Advanced Printing クラスタ環境のカスタマイズ	5-10
5.3.6.1	リソースの再配置	5-10
5.3.6.2	CAA リソースを管理するための apx_caa_setup スクリプトの使用	5-11
5.3.7	1つのメンバで実行するためのプリンタの構成	5-14
5.3.8	クラスタにおける LPD インバウンド・ゲートウェイ	5-15
5.4	サーバの管理	5-15
5.4.1	サーバの状態	5-15
5.4.2	サーバの起動	5-16
5.4.3	スプーラの起動	5-16
5.4.4	スーパーバイザの起動	5-17
5.4.5	サーバを使用可能にする	5-19
5.4.6	サーバを使用不能にする	5-19
5.4.7	サーバからのジョブの消去	5-20
5.4.7.1	サーバからのジョブの削除	5-20
5.4.7.2	スプーラからのすべてのジョブの消去	5-21
5.4.8	サーバの一時停止	5-21
5.4.9	サーバの再開	5-22
5.4.10	サーバのシャットダウン	5-23
5.4.11	サーバの削除	5-24

6 キューとプリンタの作成および管理

6.1	キューの作成と管理	6-1
6.1.1	キューを使用可能にする	6-2
6.1.2	キューからのジョブの削除	6-2
6.1.3	キューの状態	6-2

6.1.4	キュー属性のリスト	6-3
6.2	物理プリンタの作成と管理	6-3
6.2.1	物理プリンタの作成	6-4
6.2.1.1	プリンタ・アドレスの設定	6-4
6.2.1.2	プリンタ接続方法の設定	6-5
6.2.1.3	プリンタ接続レベルの設定	6-6
6.2.1.4	シリアル・ポート・パラメータ属性の設定	6-7
6.2.1.5	TCP/IP ポート番号の設定	6-8
6.2.2	物理プリンタの作成例	6-8
6.2.3	物理プリンタに対するオプション属性の設定	6-9
6.2.4	物理プリンタを使用可能にする	6-10
6.2.5	物理プリンタを使用不能にする	6-11
6.2.6	物理プリンタの一時停止	6-11
6.2.7	物理プリンタの再開	6-12
6.2.8	物理プリンタ・オブジェクトの削除	6-12
6.3	論理プリンタの作成と管理	6-13
6.3.1	論理プリンタの作成例	6-14
6.3.2	オプションの論理プリンタ属性の設定	6-14
6.3.3	論理プリンタに対する管理機能の実行	6-14
6.3.4	論理プリンタを使用可能にする	6-15
6.3.5	論理プリンタを使用不能にする	6-15
6.3.6	プリンタ属性のリスト表示	6-16
6.3.7	論理プリンタに対して実行されたジョブの表示	6-16
6.3.8	論理プリンタの削除	6-18
6.4	省略時のオブジェクトの作成	6-18
6.4.1	ジョブとドキュメントの省略時の値の指定	6-19
6.4.2	省略時のオブジェクトの作成	6-20

7 ジョブとドキュメントの管理

7.1	ジョブとドキュメントのオブジェクト処理の実行	7-1
7.1.1	ジョブの印刷	7-2
7.1.2	指定した時刻より後のジョブの印刷	7-2
7.1.3	ジョブの再実行	7-3
7.1.4	ジョブの昇格	7-5
7.1.5	ジョブの一時停止	7-5
7.1.6	ジョブの再開	7-6
7.1.7	ジョブの保留	7-6
7.1.8	ジョブの保持	7-7
7.1.9	ジョブの破棄	7-8

8 ドキュメント・データのフィルタリング

8.1	フィルタのタイプ	8-1
8.1.1	フィルタに関連する属性	8-2
8.1.2	コマンド・テキスト処理	8-5
8.1.3	フィルタの呼び出し	8-7
8.1.4	エラー処理	8-8
8.1.5	フィルタ・プログラムの作成	8-8
8.2	テキストから PostScript への翻訳フィルタ	8-9

9 イベント通知の設定

9.1	通知サーバ	9-1
9.2	通知デーモンの変更	9-2
9.3	通知方法	9-2
9.4	通知イベント	9-2
9.4.1	サーバ・イベント	9-2
9.4.2	キュー通知イベント	9-3

9.4.3	プリンタ通知イベント	9-4
9.4.4	ジョブ通知イベント	9-4
10	LPD インバウンド・ゲートウェイ	
10.1	LPD インバウンド・ゲートウェイの構成	10-1
10.1.1	/etc/printcap ファイルへの変更	10-2
10.2	ジョブ取り消しメッセージの説明	10-3
10.3	LPD インバウンド・ゲートウェイ・マッピング	10-3
11	LPD アウトバウンド・ゲートウェイ	
11.1	printer-address 属性の変更	11-1
11.2	プロトコル拡張	11-2
11.3	物理プリンタの作成	11-2
11.4	LPD スーパーバイザではサポートされない機能	11-3
11.5	アウトバウンド・ゲートウェイ・マッピング	11-4
12	トラブルシューティング	
12.1	サーバ問題の解決	12-1
12.1.1	どのサーバ・プロセスが実行中であることを調べる	12-1
12.1.2	サーバは実行中だが何も動かない	12-2
12.1.3	スーパーバイザがシャットダウンしない	12-2
12.1.4	スプーラがシャットダウンしない	12-3
12.2	ジョブとプリントの問題	12-3
12.2.1	PostScript ドキュメントで PostScript プログラム・コード が印刷される	12-3
12.2.2	物理プリンタが接続状態でハングする	12-4
12.2.3	処理待ち状態のままのジョブ	12-4
12.3	システム・エラーとエラー情報	12-5
12.3.1	コンソール通知が働かない	12-5

12.3.2	システム・エラー情報の検出	12-5
--------	---------------------	------

A 属性

A.1	サーバ属性	A-1
A.1.1	スプーラ属性	A-3
A.1.2	スーパーバイザのみ	A-3
A.2	プリンタ属性	A-4
A.2.1	論理プリンタ属性	A-8
A.2.2	物理プリンタのみ	A-8
A.3	キュー属性	A-12
A.4	ジョブ属性	A-13
A.5	ドキュメント属性	A-17

B プリンタ・データ・シート

C ONC バインディング・エントリ

C.1	ONC バインディング・エントリ	C-1
-----	------------------------	-----

D 日本語出力のための設定

D.1	日本語ファイル出力のための設定	D-1
D.1.1	スーパーバイザ (pdspvr) を使用する方 法	D-1
D.1.2	Lpd Outbound Gateway スーパーバイザ (pdspvlpr) を使用する 場合 (lpd 印刷)	D-2
D.2	プリンタの制御	D-3

用語集

索引

表

6-1	プリンタ・アドレス・フォーマット	6-5
-----	------------------------	-----

8-1	コマンド置換で使用するドキュメント属性	8-6
8-2	テキストから PostScript への翻訳プログラムのコマンド・オプション	8-10
10-1	ジョブおよびファイル・メッセージ	10-4
10-2	プリント・メッセージ	10-5
10-3	Sun プロトコル拡張	10-6
10-4	DEC プロトコル拡張	10-7
10-5	Xerox プロトコル拡張 - DocuSP および DocuPrint	10-8
10-6	Xerox プロトコル拡張 - DocuSP のみ	10-11
10-7	Xerox プロトコル拡張 - DocuPrint のみ	10-14
11-1	ジョブとファイル・メッセージ	11-4
11-2	Sun プロトコル拡張	11-4
11-3	DIGITAL プロトコル拡張	11-5
11-4	Xerox プロトコル拡張 - DocuSP および DocuPrint	11-5
11-5	Xerox プロトコル拡張 - DocuSP のみ	11-8
B-1	プリンタ・データ・シート: Digital PrintServer 20 Level 2 PS	B-2
B-2	プリンタ・データ・シート: Digital PrintServer 32 Level 2 PS	B-3
B-3	プリンタ・データ・シート: DIGITAL PrintServer LPS17 Level 2 PS	B-4
D-1	プリンタの制御 (CLI 使用時)	D-4
D-2	プリンタの制御 (GUI 使用時)	D-4
D-3	プリンタの制御 (CLI 使用時)	D-5

まえがき

本書は、Tru64 UNIX オペレーティング・システムが稼働しているシステムで、日本語 Advanced Printing Software を構成して管理するために実行するタスクについて説明しています。

対象読者

本書は、システム管理者またはプリント・システムのオペレータを対象としています。システム管理者は、このオペレーティング・システムの概念およびコマンド、さらに、対象となるシステムのハードウェアおよびソフトウェアの構成について理解している必要があります。

関連ソフトウェア

日本語 Advanced Printing Software は、Xerox 社の開発/販売による PrintXchange の技術を利用しています。

本書の構成

本書の構成は次のとおりです。

- | | |
|-------|--|
| 第 1 章 | プリント・システムの概要を示します。プリント・システムの特長および機能について説明し、プリント・システムを構成しているコンポーネントについて説明します。 |
| 第 2 章 | プリント・システムのインストールを計画する前に検討する必要がある項目について説明し、プリント・システムを構成するために必要なタスクをリストします。 |
| 第 3 章 | プリント・システムのネーム・サービスを構成し、管理する方法について説明します。 |
| 第 4 章 | プリント・システムのセキュリティ機能、およびセキュリティを設定し、構成する方法について説明します。 |
| 第 5 章 | スプーラおよびスーパーバイザを作成し、管理する方法について説明します。 |
| 第 6 章 | キュー、論理プリンタ、および物理プリンタを作成し、管理する方法について説明します。 |

第 7 章	プリント・ジョブを実行し、管理する方法について説明します。
第 8 章	データ・フィルタリングについて説明します。
第 9 章	プリント・システムのイベント通知機能について説明します。
第 10 章	LPD インバウンド・ゲートウェイを構成し、管理する方法について説明します。
第 11 章	LPD アウトバウンド・ゲートウェイを構成し、管理する方法について説明します。
第 12 章	プリント・システムで発生する可能性のある一般的な問題について説明します。
付録 A	設定可能なオブジェクト属性について簡単に説明します。
付録 B	サポートされているプリンタに関するプリンタ・データ・シートが記載されています。
付録 C	ネーム・サービスで使用する ONC バインディング・エントリの形式について説明します。
付録 D	日本語ファイルを出力するための方法と、サポートされる日本語プリンタで指定できるオプションについて説明します。
用語集	本書で使用される用語、頭字語、および略語について説明します。

関連資料

プリント・システムのユーザおよび管理者をサポートするために、次のドキュメント、リファレンス・ページ、およびオンライン・ヘルプ・ボリュームを利用できます。

日本語 <i>Advanced Printing Software</i> インストール・ガイド	このプリント・システムの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。
<i>Advanced Printing Software</i> コマンド・リファレンス・ガイド	ユーザおよび管理者が利用できるすべてのコマンド行インタフェースについて説明します。

<i>Advanced Printing Software</i> ユーザ・ガイド	プリント・ジョブを実行する方法について説明します。
GUI ヘルプ・ファイル	ヘルプ・ボリュームは、 <i>pdprintadmin</i> 、 <i>pdprint</i> 、および <i>pdprintinfo</i> GUI クライアント・プログラムからアクセスできます。

本書で使用する表記法

本書では、次の表記法を使用しています。

%

\$

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。

% **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。

file

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。

[|]

{ | }

構文定義では、大カッコはオプションの項目を示し、中カッコは必須項目を示します。大カッコまたは中カッコの中の項目を縦線で区切っている場合は、そこに併記されている項目の中から1つの項目を選択することを示します。

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。

Advanced Printing Software の概要

Advanced Printing Software は、UNIX オペレーティング・システムのための、柔軟な分散型のクライアント・サーバ型プリント・システムです。このプリント・システムの特長は、次のとおりです。

- 相互に動作して印刷に関連する機能を管理し実行する、クライアント・プログラムおよびサーバ・プログラムのセットから構成されます。
- 管理者およびオペレータには、ネットワーク環境における印刷作業を管理するためのツールのセットが提供されます。
- エンド・ユーザには、ドキュメントを印刷し、それらのプリント・ジョブを監視するユーティリティ・プログラムが提供されます。

このプリント・システムは、印刷に関する次の業界標準に準拠しています。

- ISO/IEC 10175 - Document Printing Application (DPA)
- POSIX 1387.4 - System Administration -- Part 4: Printing Interfaces

DPA 標準では、抽象的なプリント・オブジェクト、それらの状態、構文、プロトコル、および総合的な標準属性のセットが定義されています。POSIX ドキュメントでは、プリントおよびプリント・システムの管理に関連するコマンド操作、および引数のセットが指定されています。

1.1 概念および用語

Advanced Printing Software の概念を理解すると、システムの管理を簡略化するのに役立ちます。この節では、本書で使用される用語を定義し、説明します。

1.1.1 クライアントおよびサーバ

Advanced Printing Software では、クライアント・サーバ型プリント・アーキテクチャが採用されています。クライアントおよびサーバは、プロセス間通信メカニズムを使用して情報を交換します。クライアントはサーバに要求を送信し、サーバは応答をクライアントに戻します。このプリン

ト・システム・ソフトウェアには、プリント・システムとのインタフェースをとるいくつかのクライアント・プログラムが含まれ、また、クライアント・ユーティリティによって要求された操作を実行するサーバ・プログラムが提供されています。

1.1.1.1 クライアント・プログラム

クライアント・プログラムを使用すると、ユーザはプリント、リスト、およびシステム管理操作を開始することができます。要求を受信すると、サーバはその操作を実行して、結果をクライアントに送り返します。要求によって、結果は、単純な肯定応答、エラー・コード、またはプリント・ジョブのフォーマットされたリストなどのような、要求された情報のセットになります。

サポートされているクライアント・プログラムは、次のとおりです。

- コマンド行インタフェース (CLI) ユーティリティ — サーバ・プロセスによってサポートされている操作を実行するコマンドのセットを提供します。
- グラフィカル・ユーザ・インタフェース (GUI) ユーティリティ — サーバ・プロセスによってサポートされている操作を実行するグラフィカル・インタフェースを提供します。
- LPD インバウンド・ゲートウェイ・デーモン — lp プrint・システムおよび lpr プrint・システムからプリント・ジョブを受け付けて、Advanced Printing Software サーバにこれらのジョブを送信します。

1.1.1.2 サーバ・プログラム

サーバ・プロセスには 2 種類あります。

- スプーラ — プリンタ・ジョブを収集して、それらをプリント・キューに関連付け、スケジューリングして印刷します。
- スーパーバイザ — スプーラ・キューからジョブを受け取り、ジョブの必要条件およびプリンタの機能に従って印刷します。スーパーバイザは、プリンタと通信して、プリンタ状態をスプーラおよびクライアントに通知します。

どの構成にも、少なくとも 1 つのスプーラおよび 1 つのスーパーバイザが含まれます。それぞれのスプーラおよびスーパーバイザは、プリント・オブジェクトのデータベースを管理します。物理プリンタを除く、プリント・システム中のすべてのオブジェクトは、1 つのスプーラ・データベース内に作成さ

れて、保存されます。物理プリンタのオブジェクトは、1つのスーパーバイザ・データベースに保存されます。

スプーラはシステム中の1次サーバとして認識されます。スーパーバイザはエンド・ユーザにはほとんど認識されません。

1.1.2 オブジェクトおよび属性

このプリント・システムのアーキテクチャは、指定されたオブジェクトおよびオブジェクト属性によって表現されます。オブジェクトには、プリンタ、キュー、プリント・ジョブなどがあります。1つのオブジェクトの属性は、そのオブジェクトを表現します。属性には、プリンタ名、ドキュメント・フォーマット、ジョブ識別子などがあります。

プリント・システムで定義されているオブジェクトは、次のとおりです。

- server (サーバ)
- printer (プリンタ)
- queue (キュー)
- job (ジョブ)
- document (ドキュメント)
- initial-value-job (初期値ジョブ)
- initial-value-document (初期値ドキュメント)

オブジェクトを作成すると、システムによって、一部の属性に値が割り当てられます。これらの属性の多くは読み取り専用属性であるため、これらの値を設定することはできません。その他の属性については、管理者や他のプリント・システムのユーザが定義したり、変更したりすることができます。このような属性は、読み取り/書き込み属性と呼ばれます。

このプリント・システムは、多数の属性をサポートしていますが、これらすべてについて知る必要はありません。ただし、頻繁に目にする特定の属性については、時間をかけて知る価値があります。

属性には、1語またはハイフンで区切られた複数の語からなる標準名があります。一般的なプリンタ属性の例としては、printer-name、associated-queue、enabled、input-trays-ready、およびsides-supportedがあります。同様に、ジョブ属性の例としては、

job-name , job-id , および job-state-message があります。ドキュメント・オブジェクトは , document-format , sides , および number-up のような属性で表現できます。

1.1.3 操作

操作とは、オブジェクトに対して実行するアクションです。たとえば、ユーザはドキュメントを印刷したり、状態情報をリストしたり、ジョブを削除したり、プリンタを使用可能または使用不可能にしたり、サーバ属性を設定したりします。

サポートされている操作は、次のとおりです。

- 作成 (create)
- 削除 (delete)
- クリーン処理 (clean)
- 使用可能化 (enable)
- 使用不能化 (disable)
- 一時停止 (pause)
- 再開 (resume)
- 設定 (set)
- オブジェクト属性のリスト (list object attributes)
- サーバのシャットダウン (shutdown servers)
- ドキュメントの印刷 (print documents)
- ジョブの削除 (remove job)
- ジョブの再実行 (resubmit job)
- ジョブ・キューのリスト (list job queue)
- ジョブの変更 (modify job)
- ジョブの昇格 (promote job)

エンド・ユーザが使用することを想定している操作もありますが、管理目的に限って使用することを想定している操作もあります。ほとんどの操作は、特定のオブジェクトにだけ適用されますが、すべてのオブジェクトに適用される操作もあります。たとえば、管理者は「作成」操作を使用して、

新しいプリンタ、キュー、およびオブジェクトの初期値を設定することができます。ユーザは、「印刷」操作を使用してジョブやドキュメントを作成します。同様に、管理者はプリンタ属性を設定しますが、ユーザはジョブを変更できます。

1.1.4 論理プリンタ、物理プリンタ、およびキュー

このプリント・システムでは、論理プリンタおよび物理プリンタという2種類のプリンタ・オブジェクトが定義されています。これらとともに、プリント・デバイスのソフトウェア的表現を意味します。論理プリンタ・オブジェクトには、1つまたは複数の物理プリンタの特性および機能が含まれています。ユーザがジョブを印刷するときに論理プリンタを指定すると、その論理プリンタは、そのジョブおよびドキュメントの印刷を行うために必要な条件を満たす物理プリンタにジョブを送信します。物理プリンタは、実際の実出力デバイスのソフトウェア的表現です。物理プリンタ・オブジェクトには、そのプリンタの製造元とモデル、接続の詳細、およびすぐに使用できる機能のリストが含まれています。

論理プリンタと物理プリンタは、キューによって結合されています。論理プリンタと物理プリンタをキューで関連付けることによって、ユーザが印刷のために指定するプリンタと、出力デバイスの特性によって定義された物理プリンタの間に関係が確立されます。

論理プリンタと物理プリンタを区別することにより、ユーザは、アドレスやモデル番号によってではなく、機能によってプリンタを指定することができます。このようにすると、ユーザのデスクトップでのプリンタ設定をほとんど、またはまったく更新することなく、プリンタのハードウェア構成を拡張することができます。論理プリンタは、エンド・ユーザに対して、一定の、サイト固有の、省略時の属性値を持つ物理プリンタを提供するために使用されます。さらに、論理プリンタを使用すると、ファン・イン・キューおよびファン・アウト・キューを設定して、複数の類似の論理プリンタがジョブを1つのキューに送信したり、1つのキューが2つ以上の出力デバイスに対してジョブを実行したりすることも可能になります。

1.1.5 ジョブおよびドキュメント

ドキュメントとは、ユーザが印刷するファイルのことです。ドキュメントによっては、アプリケーションまたはプリンタ・ドライバによって、PCLまたはPostScriptのようなプリンタ言語にフォーマットされます。ドキュメ

ントは、ほとんど、またはまったくフォーマット情報を持たない、シンプル・テキスト・ファイルにすることもできます。

1つまたは複数のドキュメントを印刷するという要求によって、システムにジョブが作成されます。プリント・システムはこのジョブおよびそのドキュメントに、追跡と管理のために使用する固有の識別子を割り当てます。さらに、このジョブに名前を付けて、プリント・キューの末尾に挿入します。

1.1.6 初期値オブジェクト

システムでは、ジョブおよびドキュメントに対する属性の初期セットを提供する、次の2つのオブジェクトがサポートされています。

- `initial-value-job`
- `initial-value-document`

`initial-value-job` オブジェクトは、ジョブをシステムに対して実行するときに適用される、ジョブ属性のセットです。1つのスーパーラ上にある1つまたは複数の論理プリンタを `initial-value-job` オブジェクトに関連付けることにより、これらのプリンタに対して実行されるすべてのジョブが、ジョブ属性の初期セットをとるようにすることができます。ユーザは、`initial-value-job` オブジェクトをプリント・ジョブの実行の一部として指定することにより、ジョブがその属性をとるようにすることができます。

`initial-value-document` オブジェクトは、`initial-value-job` オブジェクトと似ていますが、含まれるのはドキュメント属性だけであり、ジョブを構成する個々のドキュメントに適用できるという点が異なります。

初期値オブジェクトを使用すると、ユーザは、よく使用するジョブおよびドキュメント属性のセットを、容易にプリント・ジョブに適用することができます。初期値オブジェクトは、必要に応じていくつでも作成することができます。また、その中に含まれる属性および値は、簡単に追加、削除、および変更することができます。

1.1.7 supported 属性および ready 属性

特定のオブジェクトまたは機能がサポート対象および使用可能になっている場合を宣言する、いくつかの属性があります。このような属性は、`xxx-supported` 属性および `xxx-ready` 属性と呼ばれます。これらの属性のほとんどはプリンタに適用され、管理者およびオペレータによって設定されま

す。これらの属性のごく一部はサーバに適用されますが、そのような属性はサーバによって設定され、読み取り専用です。

スプーラは、ジョブの必要条件および使用可能なプリント機能に基づいて、プリント・ジョブをスケジューリングします。多くのプリント機能は、ユーザがそれらを利用できるようにするためには、その前に、サポート対象および使用可能な属性として宣言する必要があります。管理者およびオペレータは、特定の xxx-supported 属性および xxx-ready 属性を設定して、ユーザが出力デバイスまたはプリント・ソフトウェアの機能を使用できるようにする必要があります。たとえば、物理プリンタの sides-supported 属性および sides-ready 属性を設定して、ユーザがドキュメントを両面印刷できるようにする必要があります。

物理プリンタ、論理プリンタ、スプーラ、およびスーパーバイザは、supported 属性および ready 属性を、次のように処理します。

- 物理プリンタに xxx-supported 属性および xxx-ready 属性の両方がある場合。

ジョブで input-tray=top が指定されている場合は、物理プリンタ属性 input-trays-supported および input-trays-ready の両方に、値の 1 つとして 'top' が含まれていなければなりません。

この機能を指定するジョブを受け付けるためには、この機能が supported かつ ready になっていなければなりません。この機能が supported であっても ready でない場合には、この機能が ready になるまで、ジョブがスケジューリングされません。一部の xxx-supported 属性には、対応する xxx-ready 属性がない場合があります。

- 論理プリンタに xxx-supported 属性だけがある場合。

機能が ready であるかどうかは、物理プリンタによってのみ決まります。要求されている機能が supported である場合に、ジョブを論理プリンタに対して実行することができます。

- スプーラに、たとえば論理プリンタやキューといった、スプーラのデータベースで定義されているオブジェクトおよび機能を列挙するいくつかの xxx-supported 属性および xxx-ready 属性がある場合。
- スーパーバイザに、スーパーバイザおよびそのデータベース上で定義されているオブジェクトおよび機能を列挙する、いくつかの xxx-supported 属性および xxx-ready 属性がある場合。

設定する必要がある supported 属性および ready 属性についての詳しい説明は、第 5 章を参照してください。

1.1.8 ネーム・スペースおよびネーム・サービス

プリント・システムでは、サーバ、プリンタ、およびキューは、名前で識別されます。ネーム・サービスを使用することによって、分散型のクライアントおよびサーバが、オブジェクトのネットワーク・アドレスをその名前から探し出すことができます。クライアントおよびサーバは、要求を処理するときに、ネーム・サービスを使用してお互いを探し出します。

1.1.9 サーバ・データベース

サーバは、プリンタ、キュー、およびジョブなどのオブジェクトを、永続的な、ディスク上のデータベースで保守します。このため、管理者は、システムがリブートされた場合でも、変更されるまでシステム上に存在するこれらのオブジェクトの属性を定義することができます。ユーザのニーズに合わせて、プリンタ、キュー、初期値オブジェクトを作成します。ジョブ・オブジェクトおよびドキュメント・オブジェクトは、プリント・クライアントがジョブを実行すると作成されます。これらのオブジェクトは、物理プリンタ・オブジェクトを除いてすべて、スプーラのデータベースに保存されます。物理プリンタ・オブジェクトは、スーパーバイザ・データベースに保存されます。

ここで説明したオブジェクトの他にも、それぞれのサーバ・データベースには、サーバ名、スーパーバイザなのかスプーラなのか、プリンタの名前とサポートするキューなど、そのサーバについて記述する属性を保存するために使用されるサーバ・オブジェクトが含まれています。

1.1.10 ユーザ・タイプおよびアクセス制御

プリント・システム・ソフトウェアを管理するユーザを制限するために、プリント・システムでは、明確なユーザのクラスおよびそれぞれのクラスが実行できる操作を定義します。

管理操作は、システム管理者によって実行されるものと、プリンタ・オペレータによって実行されるものに分類されます。エンド・ユーザ操作では、ファイル(ドキュメント)を印刷することや、プリント・ジョブがシステムで処理される途中経過を監視することができます。

- 管理者は、すべての操作を実行できます。

- オペレータが実行できる操作は、プリンタを使用可能または使用不能にしたり、supported 属性および ready 属性を設定したりするなど、特定の管理操作に限られます。
- エンド・ユーザは、特権が最も少ないユーザであり、実行できる操作は、プリント・ジョブの実行の他、プリント・ジョブおよびプリンタ状態の表示に限られます。

アクセス制御リストを使用して、スーパーおよびスーパーバイザへのアクセスを制限します。管理者は、どのユーザが、どの種類の操作を実行するかをサーバごとに指定する、アクセス制御リストを定義します。

1.1.11 通知およびイベント

通知プロファイルを定義することによって、プリント環境がどのように機能しているかについて、常に情報を得ることができます。システム内のサーバ、プリンタ、キュー、ジョブ、およびドキュメント・オブジェクトには、それぞれ関連付けられたさまざまなイベントがあります。たとえば、プリンタの状態が変わると通知されるイベント、ジョブが完了すると通知されるイベント、サーバがシャットダウンされると通知されるイベントなどがあります。個々のイベントを指定する他に、通知プロファイルによっても、特定のイベントが発生したときの通知方法を指定することができます。イベントの通知方法としては、電子メール、または X/Motif コンソールのどちらかを選ぶことができます。

1.2 コンポーネント

この節では、1次プリント・システム・ソフトウェアのコンポーネントについて詳しく説明します。対象となるのは、クライアント・ユーティリティ、サーバ・プログラム、デーモン、データベース、およびプリンタ属性ファイルです。

1.2.1 クライアント・プログラム

プリント・システム・ソフトウェアには、システムに対するコマンド行インタフェース (CLI)、および3つのグラフィカル・ユーザ・インタフェース (GUI)・プログラムが含まれています。

1.2.1.1 エンド・ユーザ・プログラム

CLI を使用すると、エンド・ユーザがアクセスできるすべての印刷、リスト、およびジョブ削除のプログラムにアクセスできます。CLI に加えて、エンド・ユーザは `pdprint` および `pdprintinfo` という GUI プログラムを使用して、これらの機能を実行することもできます。インストレーション・プロシージャでは、`pdprint` プログラムおよび `pdprintinfo` プログラムが CDE デスクトップに統合されるため、これらが省略時の印刷および印刷情報の動作となります。

1.2.1.2 管理プログラム

すべてのシステム機能について、プリント・システムの管理者が CLI を使用してアクセスできるようになっており、同等の機能の多数が、`pdprintadmin` GUI を使用して実行できるようになっています。

1.2.2 サーバ・プログラム

プリント・システム・ソフトウェアには、プリント・システム機能の大部分を提供する、次のような 3 つの 1 次サーバ・プログラムが含まれています。

- スプーラ (`pdsplr`)
- スーパーバイザ (`pdsplr`)
- LPD アウトバウンド・ゲートウェイ・スーパーバイザ (`pdsplpr`)

これらのサーバ・プログラムは、マルチスレッド・アプリケーションであり、多数のプリンタ、ジョブ、および要求を同時に処理することができます。中小規模のインストレーションでは、1 つのスプーラ・プロセスと 1 つのスーパーバイザ・プロセスだけが必要となります。大規模なインストレーション (プリンタが 50 台を超える) の場合には、2 つ以上のスーパーバイザ・プロセスを実行する必要があります。プリント・システムから既存の `lpr/lpd` システムまたはプリンタに対して、印刷操作をサポートする必要がある場合には、LPD アウトバウンド・ゲートウェイ・スーパーバイザを構成する必要があります。

1.2.2.1 スプーラ

スプーラ `pdsplr` は、このプリント・システムの中心となるコンポーネントです。`pdsplr` は、次のような機能を実行します。

- キューにプリント・ジョブを収集する

- 初期値属性をジョブおよびドキュメントに適用する
- ジョブの必要条件およびプリンタの機能に基づいて、プリント・ジョブを印刷のためにスケジューリングする
- プリンタで印刷の準備が完了すると、ジョブをスーパーバイザに転送する
- クライアントに対して利用可能状態にする
- 重大なイベントが発生した場合に、ユーザおよび管理者に通知する

クライアントは、要求を作成し、プリント・ジョブを指定されたプリンタに対して実行します。スプーラは、このジョブをプリンタに関連付けられたキューに入れ、initial-value-job 属性および initial-value-document 属性を、このジョブおよびそのドキュメントそれぞれに適用します。スプーラはこのジョブのドキュメント・ファイルをスプール・ディレクトリに保存し、スプール・ディレクトリでは、スーパーバイザによってアクセスされるか、またはスーパーバイザに引き渡されます。関連付けられた物理プリンタでジョブを受け付ける準備が完了すると、スプーラはジョブを印刷のためにこのスーパーバイザに転送します。

1.2.2.2 スーパーバイザ

スーパーバイザ `pdspvr` は、直接またはネットワーク接続を経由してホストに接続されている出力デバイスを制御し、これらと通信します。`pdspvr` は、ドキュメント・データを出力デバイスに送信し、`printing` (印刷中) 状態にある間のジョブの進捗状況に関する状態情報を返して、報告する役割を果たします。さらに `pdspvr` は、デバイスが通信に応答しない場合にも、これを報告することができます。

スーパーバイザ・プロセスは、最大で 50 のさまざまな種類と接続方式の出力デバイスを制御し、これらと通信することができます。プリンタ、ネイティブ言語、および通信機能に応じて、スーパーバイザは双方向通信、柔軟なデバイス制御、複数のプリンタ言語、ドキュメントのデータ・ストリームを翻訳または変更できるフィルタ、およびオプションのフォーマットされたセパレータ・ページを提供します。

1.2.2.3 LPD アウトバウンド・ゲートウェイ・スーパーバイザ

LPD アウトバウンド・ゲートウェイ・スーパーバイザ `pdspvlpr` を使用すると、ユーザは、プリント・ジョブを既存の `lp/lpr` プリンタに送信したり、使用している環境にあるプリント・システムに送信したりすることができます。

す。ジョブを LPD キューに登録したり、まだプリントしていないジョブをキューから削除したりできます。

1.2.3 サーバ・デーモン

Advanced Printing Software には、クライアント・ユーティリティおよびサーバ・プログラムの代わりに動作する、次のデーモン・プログラムが含まれています。

- プロトサーバ (rpc.pts)
- 通知サーバ (pdntfs)
- コンソール通知デーモン (pdconntf)

プロトサーバ・デーモンは、ホスト上で動作しているスプーラおよびスーパーバイザのために、サーバ登録および検索エージェントとしてサービスを提供します。クライアントはまず、ホストのプロトサーバと通信して、操作で指定されたサーバの RPC バインド情報を取得しなければなりません。プロトサーバは、固定登録プログラム番号を使用して自分自身をホスト上で登録します。

通知サーバは、スプーラおよびスーパーバイザによって発行された通知メッセージを配信します。通知サーバは、重大なイベントが発生した場合に、電子メールを送信したり、GUI メッセージ領域にメッセージを表示したりする役割を果たします。コンソール通知デーモン pdconntf は、通知メッセージを CDE システム・コンソール・ウィンドウに表示させる場合に必要です。

これらのデーモンのうち、それぞれ 1 つのコピーだけが、一度に 1 つのホストで実行されます。プロトサーバ・デーモンおよび通知サーバ・デーモンは、inetd プログラムの制御によって自動的に実行されます。

1.2.4 プリンタ属性ファイル

このプリント・システム・ソフトウェアは、さまざまなプリンタ・モデルをサポートしています。このソフトウェア・パッケージには、サポートされているプリンタ・モデルそれぞれについて、プリンタ属性ファイルが含まれています。それぞれのファイルは、対応するプリンタ・モデルの機能を記述し、オプションを指定せずに関連付ける属性のセットを指定します。プリンタ属性ファイルを使用すると、新しくサポートされるプリンタを、現在のプリント・システムに容易にインストールできるようになります。こ

これらのプリンタ属性ファイルには、対応するプリンタの製造元、およびモデルを容易に識別できる名前が付いています。

1.2.5 翻訳フィルタおよび変更フィルタ

スーパーバイザは、ドキュメント・データ・ストリームを、出力デバイスに引き渡している間、これを翻訳したり変更したりすることができます。管理者は、1つまたは複数の翻訳フィルタを設定して、あるフォーマットのドキュメントを、ターゲット・プリンタと互換性を持つ別のフォーマットに自動的に翻訳することができます。ドキュメントを印刷中に、ユーザは変更フィルタを指定することもできます。そのようなフィルタを使用すると、行番号およびページ番号を追加したり、ドキュメントの不要な部分を削除したり、プリンタの必要条件に合わせてコード・セットを変更したりできます。

ほとんどの標準 UNIX ユーティリティを始め、入力を `stdin` からとり、出力を `stdout` に送信するプログラムであれば、どのようなプログラムでも、翻訳フィルタまたは変更フィルタとして構成することができます。また、独自のフィルタ・プログラムを開発して、構成することもできます。

Advanced Printing Software には、テキスト・ドキュメントを PostScript に変更する翻訳フィルタ `trn_asciips` が含まれています。このフィルタはさらに、片面に1つ、2つ、または4つのページ・イメージが配置されるようにテキストをフォーマットすることもできる (`number-up`) ほか、横長または縦長の方向で印刷することもできます (`content-orientation`)。ユーザがテキスト・ファイルを PostScript プリンタで印刷する必要がある場合には、スーパーバイザの `filter-definition` 属性を設定してこの機能を使用する必要があります。

インストールの計画

この章では、プリント・システムをインストールする際の計画について説明します。

プリント・システムをインストールするには入念な計画が必要です。プリント・システムを作成する前に次の質問項目を検討してください。

- サーバ・コンポーネントのホストとして働くのはどのコンピュータか？
- クライアント・コンポーネントのホストとして働くのはどのコンピュータか？
- ユーザ、オペレータ、管理者は誰にするのか？
- どのネーム・サービスを使用するのか？
- 論理プリンタをどのように物理プリンタにマップするのか？
- ジョブとドキュメントでは、どのような省略時の値を必要とするのか？

この章では、これらの質問に答えるための説明を行います。この章では概要だけを説明し、ソフトウェアをインストールするコマンドや、オブジェクトを作成するコマンドの詳細については説明しません。オブジェクト作成の詳細情報については、本書のこれ以降の章で説明し、ソフトウェアのインストール手順については、『インストール・ガイド』を参照してください。

2.1 論理構成

プリント・システム・ソフトウェアのコンポーネントを設定するにはいくつかの方法があります。この節では、クライアント・コンポーネントおよびサーバ・コンポーネントのインストール方法とインストール場所、インストールするサーバの数を決める方法や、全コンポーネントが効率的に通信し合うことを確実にする方法について説明します。

2.1.1 プリンタの構成

サーバ、プリンタ、キューは、システム内で高い柔軟性を持って構成できます。1つのホスト上ですべてのプリント・システム・コンポーネントを実行することも、あるいはプリント・システム・コンポーネントを分散して、それぞれの機能に合ったホスト上で実行することもできます。たとえば、サーバ・ホスト上でサーバ・プロセスを実行し、ユーザ・ワークステーション上でプリント・システム・クライアントを実行することもできます。

2.1.2 複数サーバ・ホスト

プリント・システムのスプーラとスーパーバイザは、それぞれ異なるホスト上に存在していても相互に通信することができます。たとえば、配線が楽になるよう、直接接続された複数台のプリンタの近くにあるホストでスーパーバイザを実行し、一方、スプーラは、建物やサイト内の任意の場所にあるサーバ・ホスト上で実行することもできます。

プリント・システム・コンポーネントを実行する際に複数のホストを使用するときには、作成するオブジェクトの名前を、それらのオブジェクトを必要とするすべてのホストに配信する必要があります。

2.1.3 サーバの制限事項

プリント・システムのスプーラとスーパーバイザに対して、論理プリンタ、物理プリンタ、キューを構成する際には、いくつかの制限事項が適用されます。

- あるキューに関連付けられているすべての論理プリンタは、そのキューと同じスプーラ上に存在する必要があります。
- あるスーパーバイザ上のすべての物理プリンタに関連付けることができるのは、1つのスプーラ上のキューと論理プリンタだけです。

2.1.4 論理プリンタから物理プリンタへの関連付け

プリンタをシステム内に構成するときには、論理プリンタと物理プリンタの間で緊密な関連付けを確立するかどうかを選択できます。

2.1.5 緊密な関連付けと疎遠な関連付け

一部のユーザにとって、論理プリンタと物理プリンタの概念とその違いを理解するのは難しいことかもしれません。あるいは、論理プリンタと物理プリンタとの区別をする必要がないこともあります。そのような場合は、論理プ

リタ、物理プリンタ、キューの3つを、1つのプリンタとして共同で動作するように密接に結合させて作成します。つまり、すべての出力デバイスはこれらのオブジェクトそれぞれ1つによって表されます。

2.1.6 疎遠な関連付け、ファン・イン・キューとファン・アウト・キュー

ユーザの混乱が最小限あるとしても、プリンタ構成の変更が可能なプリント環境において柔軟性が必要となる場合には、論理プリンタと物理プリンタの抽象化を活用してもよいでしょう。

たとえば、ある「プリンタ・ルーム」に、すべて同じ機能を持つ複数の類似した出力デバイスがあるとします。いくつかの物理プリンタを1つのキューに関連付けることは、「負荷分散」構成を実装することになり、ここで、ユーザが1つの論理プリンタに対してジョブを実行すれば、システムは印刷のために次に利用可能なデバイスを選択します。このような構成は、ファン・アウト・キューと呼ばれることもあり、ユーザから見える部分には何ら影響を与えずに、個々のプリンタへのサービスを提供したり、追加のプリンタを構成したり、余分なプリンタを取り除いたりできます。

これに対して、ファン・イン・キュー構成とは、複数の論理プリンタのジョブが1つのキューに入るものです。初期値オブジェクトを使用して、論理プリンタを特定の省略時のジョブとドキュメントのオプションで設定することもできます。そのようなプリンタは、それぞれ独自のジョブ処理機能を持つと考えられます。たとえば、ある論理プリンタはテキスト・ドキュメントを縦長モードで片面に1ページを常に印刷するように設定し、別の論理プリンタは行番号付きの横長モードで片面に2ページのテキスト・ドキュメントを印刷するように設定できます。これらそれぞれの論理プリンタを1つのキューに関連付けると、システムはこれらの論理プリンタに対して実行されたドキュメントを、そのキューに関連付けられた1つ以上の物理プリンタに出力します。

ファン・イン・キューとファン・アウト・キューは、一緒に使用することで高い柔軟性と便宜性を提供できます。

2.1.7 単一ホスト上での複数のスーパーバイザの使用

サーバ属性 `maximum-number-of-printers-supported` で指定されているよりも多くの数の出力デバイスを1つのサーバでサポートしようとする場合、複数のスーパーバイザ・プロセスとオブジェクトを使用する必要があります。

出てきます。次の理由から、最大数に達する前でも複数のスーパーバイザをサポートすることもあります。

- 負荷が過剰にかかっているサーバ・プロセスが単一の障害点を構成すると、ダウンする可能性がある。
- 最大数よりも少なく作成する方が、ニーズが変更するたびにシステムを再構成する際の柔軟性が高くなる。
- 負荷が過剰なサーバ・プロセスは応答性が低くなり、また、大きくなるので、スワップするのが難しくなる。また、スワップはより多くのプリンタに影響する。

2.1.8 直接接続されたプリンタ

出力デバイスの中には、ホスト・システムのハードウェアに直接取り付けられているシリアル・ポートやパラレル・ポート経由で接続されたものがあります。プリント・システムでは、このように直接接続されたプリンタを備えたすべてのホスト上で、スーパーバイザ・プロセスを実行する必要があります。使用中の環境で、ユーザのワークステーションにそのようなプリンタが何台か接続されている場合、それぞれのワークステーションでスーパーバイザ・プロセスを実行する必要があります。それらのワークステーションの管理者は、中央の1つのスプーラ上のキュー、あるいは分散された複数のスプーラ上のキューに対して、物理プリンタを関連付けるかどうかを選択できます。

2.2 ネーミング・サービスの計画

サーバ、キュー、プリンタの数が多くなるにつれ、一意のオブジェクト名を割り当てて、管理し、配信することは、次第に複雑な作業になります。このような環境では、Network Information System (NIS) ソフトウェアを活用して、集中管理したプリンタ・オブジェクト名マップを配信するとよいでしょう。

プリンタ構成ファイルと Network Information System ソフトウェアの詳細については、第3章の「ネーム・スペースの管理」を参照してください。

2.3 通知

プリント・システム・アーキテクチャは、プリント・システムでイベントが発生したときに、エンド・ユーザ、オペレータ、管理者に通知するための方法を提供します。

詳細については第9章を参照してください。

2-4 インストレーションの計画

2.4 セキュリティ

セキュリティ機能によって、ジョブ・データや管理機能へのアクセスを制御することができます。これらの機能はすべての操作要求に対して認証と許可のチェックを行うことにより、許可されたユーザだけがプリント・システムへのアクセスを取得することを保証します。

- 認証とは、ユーザが、主張しているその人であることの正当性を検査する処理です。
- 許可とは、認証されたユーザが特定の操作を実行するために必要なアクセス許可を備えていることを確認する処理です。

ユーザのアクセス許可には、次の3つの特権レベルがあります。

- エンド・ユーザ
- オペレータ
- 管理者

特権レベルにはアクセス順序があります。管理者は、オブジェクトを構成でき、それに加えてオペレータが実行できるすべてのタスクを実行できます。オペレータはプリントの制御ができ、それに加えて、エンド・ユーザが実行できるすべてのタスクを実行できます。

プリント・システムは、`access-control-list` 属性を使用して、各ユーザに特権レベルを指定します。認証は、`access-control-list` 属性内のエントリに操作要求を開始したユーザに対して、ドメインのパスワード・ファイル内のユーザ識別子 (UID) との比較を行います。この比較が成功した場合、次に、ユーザの特権レベルについて `access-control-list` 属性をチェックする許可処理が行われます。

2.5 プリント・システム・ソフトウェアの構成

この節では、プリント・システムを最初に構成して、開始するために必要なタスクの概要を説明します。タスクは実行する順に説明しており、また、必須タスクの他にオプションのタスクも含めています。

それぞれのタスクでは、簡潔な説明を記載し、タスクの実行方法を詳細に説明している本書内の章を示しています。

1. サーバ、スプーラ、インバウンド・ゲートウェイ・サーバを作成して、構成する。
サーバの作成と構成に関しては、第 5 章を参照してください。
インバウンド・ゲートウェイの作成と構成に関しては、第 10 章を参照してください。
2. キューとプリンタを作成して、構成する。
論理プリンタ、物理プリンタ、キューの作成と構成に関しては、第 6 章を参照してください。
3. ネーミング・サービスを設定する。
システムをサポートするために使用するネーミング・サービスの設定に関しては、第 3 章を参照してください。
4. セキュリティを設定する。
プリント・システムのセキュリティに関しては、第 4 章を参照してください。
5. オプションとして、イベント通知を設定する。
イベント通知の設定に関しては、第 9 章を参照してください。

Advanced Printing Software キットには、管理者が作動中のプリント・システムを構成するために使用できる、文字セルでメニュー形式のスクリプトが含まれています。このスクリプトは `/usr/pd/scripts/pd_get_started` ディレクトリにあり、新しいプリント・オブジェクトを作成するとき `root` ユーザが実行する必要があります。

`pd_get_started` スクリプトによって、スプーラ・プロセスとスーパーバイザ・プロセス、論理プリンタと物理プリンタ、キュー、および関連付けられた初期値オブジェクトの作成はかなり簡単になります。また、このスクリプトは、サーバ・プロセスの起動とシャットダウンの手段、およびローカル・サーバ構成の表示のための手段も提供します。

このスクリプトは、日常的なプリント・システムの管理用ではありません。これは、システムをただちに起動して、稼働するために最も便利なもので

す。日常のプリント・システム管理には、CDE pdprintadmin GUI ツールの方が適切です。

ネーム・スペースの管理

ネーム・スペースには、作成したオブジェクト (スプーラ、スーパーバイザ、論理プリンタと物理プリンタ、キュー) の名前およびネットワーク・アドレスが含まれています。それぞれのオブジェクトに対するネーム・サービス・エントリには、オブジェクトをサポートするサーバのアドレスが入っている情報が含まれています。

クライアントとサーバは、ネーム・サービス機能を使用して、指定されたネットワーク・オブジェクトをサポートするサーバの場所を特定し、そのサーバをバインドします。クライアントまたはサーバは、プリンタ名またはサーバ名のバインドを必要とするときに、ネーム・サービスを使用してこの情報を取得します。ネーム・サービスは要求された名前を検索し、要求されたバインディングをクライアントに戻します。RPC メカニズムが、戻されたバインディングを使用してサーバに接続します。

プリント・システムは次のネーム・サービスをサポートします。

- ローカル・ファイル・ネーム・サービス (Local File Name Service)
- ネットワーク情報サービス (NIS)
- LDAP (Lightweight Directory Access Protocol)

3.1 ローカル・ファイル・ネーム・サービス

ローカル・ファイル・ネーム・サービスは、単一のスタンドアロン・システム構成でも、あるいは分散環境でも作動します。

スタンドアロン構成では、クライアントとサーバ (スプーラとスーパーバイザ) は同一のワークステーション上に存在します。管理者がプリント・オブジェクトを作成すると、ネーム・サービスはそのバインディング情報をただちに利用可能にします。プリント・オブジェクトを削除すると、ネーム・サービスはそのバインディング情報をただちに削除します。

ローカル・ファイル・ネーム・サービスは、`/etc/printers.conf` ファイルにプリンタ・バインディング情報を格納します。このファイルの別個のイ

インスタンスがそれぞれのホスト・システム上に存在します。プリント・システムは、複数のホストによる単一の /etc/printers.conf ファイルの共有をサポートしません。次に示すのは、printers.conf ファイルの例です。

```
#
# If you hand edit this file, comments and structure may change.
#
bulldog_sup:\
:saddr=bulldog.gandalf.xyz.com,105004,1,sys,sv,bulldog_sup,1:
bulldog_spl:\
:saddr=bulldog.gandalf.xyz.com,105004,1,sys,sl,bulldog_spl,1:
bulldogl:\
:paddr=bulldog.gandalf.xyz.com,105004,1,sys,pp,bulldog_sup,1:
bulldog_q:\
:qaddr=bulldog.gandalf.xyz.com,105004,1,sys,qu,bulldog_spl,1:
bulldog_log:\
:paddr=bulldog.gandalf.xyz.com,105004,1,sys,lp,bulldog_spl,1:\
:spooling-type=dpa:
cc3:\
:paddr=bulldog.gandalf.xyz.com,105004,1,sys,pp,bulldog_sup,1:
```

サーバのスタートアップ・プロセスは、サーバがエントリをローカル・ファイルに追加しようとしたときにファイルが存在しない場合には、printers.conf ファイルを自動的に生成します。また、サーバはオブジェクト・データベースの内容をローカル・ファイルと比較して、オブジェクトがオブジェクト・データベースにはあるのにローカル・ファイルにはないときには、そのオブジェクトを追加します。

オブジェクト作成処理は、操作を実行したホスト上のローカル・ファイルだけを更新するので、その情報は他のホスト上のクライアントやサーバでは利用できません。このため、分散環境でローカル・ファイル・ネーム・サービスを使用するときには、あらかじめ構成ファイルを作成して、クライアントやサーバを実行するすべてのホストにそれをコピーしてください。

printers.conf ファイルは、エディタで作成したり、単一のホストからすべてのプリント・オブジェクトを作成することで作成したりできます。ただし、後者の場合、作成操作でサーバを作成していないので、それぞれの異なるホストについてサーバ・エントリを手動でファイルに追加する必要があります。オブジェクトを削除する場合には、すべてのホスト上にあるファイルを更新しなければなりません。

ローカル・ファイルのネーミングには、以下が必要になります。

- printers.conf ファイルが /etc ディレクトリに存在し、rw-r--r-- のアクセス権を持つ。
- ファイル所有者は root である。

3-2 ネーム・スペースの管理

3.2 ネットワーク情報サービス

ネットワーク情報サービス (NIS) は、ローカル・ファイル・ネーム・サービス用のプリンタ構成エントリと同じフォーマットを使用します。ただし、NIS は同一のプリンタ構成データを 1 つの NIS ドメイン全体で管理し、配信する手段を提供します。

ローカル・ファイル・ネーム・サービスの使用と NIS の使用の最も重要な相違点は、プリント・システムが NIS エントリを変更できないということです。代わりに、管理者は、NIS ファイルにある NIS エントリを手動で更新する必要があります。NIS ファイルへの変更権限が、あるいは、変更を行う権限のあるプロキシ管理者が必要です。

NIS ネーム・サービスでは、構成変更に関連した更新が必要になります。つまり、NIS ファイルにオブジェクト名を追加してからプリント・システム・オブジェクトを作成する必要があります。ただし、複数のホストそれぞれにあるローカル・ファイルを更新する必要があるローカル・ファイル・ネーム・サービスの場合とは異なり、NIS ではデータの更新が必要になるのは 1 箇所だけです。

プリンタ、サーバ、キューの名前と場所を、NIS クライアントとして設定されたホストに配信するには、プリント・システム・サーバ・プロセスを実行するホストから 1 つ以上の `printers.conf` ファイルを集めて、それらを `/var/yp/src/printers.conf` のマスタ・マップ・ファイルに格納する必要があります。テキスト・エディタを使用して、これらのファイルをマージしてもかまいません。結果として生成したファイルでエントリが重複している場合には、テキスト・エディタを使用して重複部分を削除してください。

マスタの `printers.conf` ファイルを作成した後に、次の手順を使用して、NIS サーバ上の NIS マップを更新してください。

1. `root` としてログインします。
2. まだコピーを行っていない場合には、`/usr/pd/scripts/Makefile.printers` を `/var/yp/Makefile.printers` にコピーします。このファイルのコピーを編集して、`DOM` 変数で NIS ドメインを定義します。
3. 現在のディレクトリを `/var/yp/src` に設定します。

4. 関連するすべての `printers.conf` ファイルを、それぞれに一意的な名前を付けて、さまざまなホストから現在のディレクトリにコピーします。
5. テキスト・エディタを使用して、これらのファイルの内容を新しいマスタの `printers.conf` ファイルにマージします。
6. 現在のディレクトリを `/var/yp` に変更します。
7. `printers.conf` (または“すべての”) ターゲットを指定して `Makefile.printers` ファイルを実行することにより、プリンタ・マップを再作成して配信し直します。
8. 新しい `printers.conf` マップが利用可能であることを確認します。

次の例はこの手順を示しています。

```
# cp /usr/pd/scripts/Makefile.printers /var/yp/Makefile.printers
# cd /var/yp/src
# mv printers.conf printers.conf.<date>
# cp /etc/printers.conf ./
# cat printers.conf.host1 printers.conf.host2\
  printers.conf > printers.conf
# cd ..
# make -f Makefile.printers
updated printers.conf
pushed printers.conf
```

次のコマンドを使用して、新しい `printers.conf` エントリを検索できることを確認してください。

```
# ypcat printers.conf.byname
```

このコマンドを実行すると、次に示すような出力が生成されます。各行が、マスタ・マップにリストされているそれぞれのサーバ、プリンタ、キューの情報を示しています。

```
WS_sharie_PP:paddr=wstent.gandalf.xyz.com,105004,1,sys,pp,wstent_sup,1:
ws_lg_queue:qaddr=wstent.gandalf.xyz.com,105004,1,sys,qu,wstent_spl,1:
WS_cross_PP:paddr=wstent.gandalf.xyz.com,105004,1,sys,pp,wstent_sup,1:
WS_cress_PP:paddr=wstent.gandalf.xyz.com,105004,1,sys,pp,wstent_sup,1:
wstent_sup:saddr=wstent.gandalf.xyz.com,105004,1,sys,sv,wstent_sup,1:
ws_lg04_pp:paddr=wstent.gandalf.xyz.com,105004,1,sys,pp,wstent_sup,1:
glypha_spl:saddr=glypha.gandalf.xyz.com,105004,1,sys,sl,glypha_spl,1:
glypha_obg:saddr=glypha.gandalf.xyz.com,105004,1,sys,sv,glypha_obg,1:
WSQ1:qaddr=wstent.gandalf.xyz.com,105004,1,sys,qu,wstent_spl,1:
BigLinePrinter:paddr=wstent.gandalf.xyz.com,105004,1,sys,lp,wstent_spl,1: :spooling-type=dpa:
wstent_spl:saddr=wstent.gandalf.xyz.com,105004,1,sys,sl,wstent_spl,1:
ws_test_lp:paddr=wstent.gandalf.xyz.com,105004,1,sys,lp,wstent_spl,1: :spooling-type=dpa:
WS_lps:paddr=wstent.gandalf.xyz.com,105004,1,sys,lp,wstent_spl,1: :spooling-type=dpa:
```

一般に、オブジェクト(サーバ、プリンタ、キュー)を作成したり、削除したりしたときにはいつでも、プリント・システム・クライアントが使用される

ホスト上のネーム・スペースを更新する必要があります。NIS ベースのネーム・サービスでは、クライアントがサイトの NIS サーバから最新のマップを読み取るので、これは自動的に発生します。

また、他のサーバ・ホストと通信するサーバ・マシン上のネーム・スペースを更新する必要もあります。たとえば、ホスト A 上のスプーラがホスト B 上にあるプリンタに供給するキューを持つ場合、ホスト A とホスト B の両方にあるネーム・スペースは、互いのオブジェクトに対するエントリを含む必要があります。プリント・システム・サーバ・ホストがドメインの NIS サーバではない場合、その `printers.conf` ファイルのコピーを NIS サーバに転送し、そのプリンタ構成を変更したときにはいつでも `printers.conf` マップをプッシュする必要があります。

NIS マップが既にオブジェクトを含んでいるとき、(`pddelete` コマンドを使用して) そのオブジェクトを削除し、次に、同じオブジェクトを (`pdcreate` コマンドを使用して) もう一度作成した場合、そのオブジェクトはローカル・ファイル・ネーム・スペースでは削除されます。これは、その名前が既に NIS ネーム・スペース内に存在するためであり、サーバは既にエントリが存在する場合には、新しいネーム・スペース・エントリを作成しません。このため、次に収集とプッシュを行うときにネーム・スペース・エントリを失う可能性があるため、オブジェクトを削除し再作成するときには十分注意してください。このような状況を回避するには、次の 2 つの方法があります。

1. オブジェクトを削除した直後と、オブジェクトを再作成した直後に NIS マップの収集とプッシュを行う。
2. `printers.conf` ファイルと NIS ネームスペースの現在のスナップショットをマージしてから、NIS マップをプッシュする。

```
# cd /var/yp/src
#ypcat printers.conf.byname > printers.conf.NIS
#cat /etc/printers.conf printers.conf.NIS > printers.conf
```

結果として生成された `printers.conf` ファイルの重複部分や古くなったエントリを削除してエントリを調整してから、前の例で説明したように `Makefile.printers` ファイルを使用してマップを更新してください。

3.3 LDAP ネーム・サービス

LDAP は、プリント・オブジェクトが作成または削除される場合に、ネーム・スペースを動的に更新する機能を NIS に提供します。この節では、LDAP クライアントを設定するための必要事項について説明します。

LDAP クライアントを設定するには、次のことを行う必要があります。

- 使用するネーム・サービスとして LDAP を識別するため、および LDAP サーバ・ホストを識別するために `/var/pd/config/apx.conf` ファイルを作成または変更します。このファイルは、スプーラおよびスーパーバイザを含むホストおよび Advanced Printing クライアントで必要です。
- サーバが LDAP データベースを変更するために、スプーラおよびスーパーバイザが存在するホストにユーザ ID およびパスワードを作成します。

3.3.1 apx.conf ファイルの作成と編集

`/var/pd/config/apx.conf` ファイルは、ユーザが使用するネーム・サービスに関する情報および LDAP ディレクトリ・サービスを実行するホスト・システムに関する情報を含んでいます。次に、`/var/pd/config/apx.conf` ファイルの 1 例を示します。

```
name-services = file nis ldap
LDAP_hosts   = system.abc.xyz.com
LDAP_path    = ou=organizational unit,o=organization
```

- `name-services` は、プリント・システムが使用するネーム・サービスを示し、その他はネーム・サービスが使用する情報です。この例では、ローカル・ファイル、NIS および LDAP が使用されます。プリント・オブジェクトの検索は、ローカル・ファイル (`/etc/printers.conf`)、`printers.conf.byname` NIS マップ、および LDAP ディレクトリ・サーバで行われます。
- `LDAP_hosts` には、LDAP ディレクトリ・サービスが実行されているホストを 3 つまで指定できます。同じサービスを提供するホストが複数ある場合には、複数のホストを指定します。複数のホストを指定する場合は、コロンで区切らなければなりません。LDAP サーバが TruCluster のバージョン 5.0 以降のソフトウェアを実行している場合には、ホスト名としてクラスタ別名を使用できます。

標準の LDAP サービスのポート番号は、389 です。非標準ポート番号の指定も可能です。LDAP ポート番号は、次の例のようにホスト名に括弧で囲んだポート番号を追加して指定します。

```
LDAP_hosts = myhost.xyz.com(8182)
```

- LDAP_path では、ディレクトリ検索パスを指定します。LDAP ベースのデータベースにアクセスするには、ネーム・パスを指定しなくてはなりません。印刷クライアントおよびサーバは、このパスを使用してエントリの作成や、検索を行います。Netscape ディレクトリ・サーバでは、"Organization" (組織) と "organizational units" (部門) を指定します。Advanced Printing に対して指定するネーム・パスは、ネーム・パス構文中に組織および部門を含みます。例えば、トップ・レベルの組織名が "XYZ Corp" で、部門を "AdvPrint" と定義した場合、ネーム・パスは、次のようになります。

```
ou=AdvPrint,o=XYZ Corp
```

LDAP_path 式には、エントリを 3 つまで指定できます。

各々のエントリは、コロン (":") で区切ります。例えば、XYZ Corp 中に "Test Environment" という 2 番目の部門を設定する場合、LDAP_path を次のように指定します。

```
ou=AdvPrint,o=XYZ Corp:ou=Test Environment,o=XYZ Corp
```

LDAP_path 式のエントリの検索は、一致するエントリが見つかるまで続けられます。エントリを作成または削除するときには、最初のパス・エントリだけが使用されます。LDAP_path 式にリストされている 2 番目と 3 番目のエントリは検索時に使用されます。

3.3.2 LDAP クライアントのユーザ名とパスワードの作成

Advanced Printing サーバで LDAP サーバ上のネーム・エントリを変更するにはユーザ名とパスワードが必要です。ユーザ名とパスワードは、apx.conf ファイルで LDAP_hosts を定義してから作成する必要があります。

pdldappw コマンドでユーザ名とパスワード作成します。

```
# /usr/sbin/pdldappw
```

このコマンドは、/var/pd/config/apx.conf ファイル中の現在の設定を表示し、ユーザ ID およびパスワードの入力を促します。

```
Contents of configuration file /var/pd/config/apx.conf:
```

```
name-services = ldap
```

```
LDAP-path=ou=advprint,o=Organization
```

```
LDAP-hosts=toons.xyz.ayy.com
LDAP User ID: advprintid
LDAP password: *****
```

3.4 プロトサーバ

プロトサーバは、ネーム・サービスとともに動作して、サーバ・プロセスがプリント・システム・オブジェクトの名前やバインディング情報にアクセスできるようにするプリント・システム・デーモンです。

プロトサーバはプリント・システム内の 1 次 RPC サーバとして働きます。リモート・ホスト上のクライアントとサーバは、サーバ・ホスト上のプロトサーバとやりとりして、他のプリント・システム・サーバ・プロセス用の RPC バインディング情報を調べます。

プリント・システム・インストール・プロシージャは、ネーム・サービスが自動的にプロトサーバを実行できるように /etc/inetd.conf ファイルに次の行を追加します。

```
105004/1 dgram rpc/tcp wait nobody /usr/sbin/rpc.pts\
rpc.pts
```

3.5 TruCluster Server 環境における高可用性アプリケーションとしての Advanced Printing Software

Advanced Printing Software は、TruCluster Server 環境の高可用性シングル・インスタンス・アプリケーションとして実行するように構成できます。シングル・インスタンス・アプリケーションは、クラスタのただ 1 つのメンバ上でのみインストール、構成、および実行されますが、クラスタの全メンバで使用することができます。

高可用性シングル・インスタンス・アプリケーションは、TruCluster Server の CAA (Cluster Application Availability) サブシステムを使用するように構成されます。CAA は、クラスタ中の Advanced Printing Software に必要なリソースを監視し、これらのリソースの要求に合うクラスタ・メンバ上で実行するようにします。Advanced Printing Software の実行が失敗したり、リソースの要求が失敗した場合、CAA は Advanced Printing Software を要求されたリソースを持つ他のメンバに再割り当て、つまりフェイルオーバーします。

高可用性シングル・インスタンス・アプリケーションとして Advanced Printing Software を構成し、実行するには次の準備が必要です。

- リソース・プロファイル
- 処理スクリプト
- アプリケーションを CAA に登録
- アプリケーションの起動

これらの項目については、以降の各項で説明します。詳細については、『*TruCluster Server* 高可用性アプリケーション・ガイド』を参照してください。

3.5.1 リソース・プロファイル

リソース・プロファイルでは、CAA によるアプリケーションの起動、管理、および監視方法を定義します。

Advanced Printing Software サーバのサブセット (APXSERVERxxx) を構成するときには、システムに TruCluster ソフトウェアがインストールされているかどうかを判断します。TruCluster Server がインストールされている場合には、構成スクリプトは `/var/cluster/caa/profile` ディレクトリ中に CAA リソース・プロファイル `apx-default.cap` をインストールします。これは、プリント環境で動作するサーバおよびスーパーバイザを示すリソース・ファイルです。

3.5.2 処理スクリプト

処理スクリプトは、アプリケーションの起動方法、アプリケーションの停止およびフェイルオーバー前のクリーンアップ方法、およびアプリケーションが実行中であるかどうかの確認方法を指定します。

Advanced Printing Software のインストール中に、TruCluster Server ソフトウェアが検出された場合、構成スクリプトは処理スクリプト `/var/cluster/caa/script/apx-default.scr` をインストールします。

3.5.3 リソースの登録

リソース・プロファイルと処理スクリプトのインストール後、リソースを CAA に登録する必要があります。リソースを登録するには、`caa_register` コマンドを使用します。

```
# /usr/sbin/caa_register apx-default
```

3.5.4 リソースの起動と停止

apx-default を CAA に登録すると、caa_start コマンドを使用してアプリケーションを実行できます。

```
# /usr/sbin/caa_start apx-default
```

アプリケーションが起動されると次のようなメッセージが表示されます。

```
Attempting to start 'apx-default' on member 'membername'  
Start of 'apx-default' on member 'membername' succeeded.
```

アプリケーションを停止するには、caa_stop コマンドを使用します。

```
# /usr/sbin/caa_stop apx-default
```

3.5.5 apx-default リソースへのサーバの追加

Advanced Printing Software アプリケーションが最初にクラスタ環境で動作した後、別のスプーラおよびスーパーバイザが必要になる場合があります。追加するスプーラおよびスーパーバイザを作成および起動します。/usr/pd/scripts/pd_get_started スクリプトまたは pdmakedb、pdsplr および pdspvr コマンドを使用できます。サーバが作成され、起動されると、自動的に apx-default リソースに追加されます。

3.5.6 Advanced Printing クラスタ環境のカスタマイズ

3.5.6.1 リソースの再配置

アプリケーションのリソースを再配置するには、caa_relocate コマンドを使用します。リソースを再配置したいクラスタ・メンバを指定したり、CAA が利用可能なメンバを判断できるようにします。次の例は、caa_relocate コマンドを使用してリソースを再配置する方法を示します。

```
# caa_relocate apx-default -c daffy
```

次のメッセージがこのコマンドの応答として表示されます。

```
Attempting to stop 'apx-default' on member 'goofy'  
Stop of 'apx-default' on member 'goofy' succeeded.  
Attempting to start 'apx-default' on member 'daffy'  
Start of 'apx-default' on member 'daffy' succeeded.
```

アプリケーションのリソース・プロファイルで定義した配置ポリシーに従ってリソースを再配置するには、オプションを指定しないで caa_relocate コマンドを使用します。

```
# caa_relocate apx-default
```

3.5.7 1つのメンバ上で実行するプリンタの構成

クラスタの1つのメンバだけがアクセスできるようにプリンタを構成できます。このプリンタは、通常、特定のクラスタ・メンバに直接接続されています。

`printer-associated-host` 属性で指定されているものとは異なるホスト上で実行されているスーパーバイザからプリンタにジョブが送られてくる場合、スーパーバイザは物理プリンタを利用できないため、次の属性を設定します。

- `enabled=false`
- `availability=none`
- `printer-problem-message= " Printer port associated with a different host "`

3.5.8 クラスタにおける LPD インバウンド・ゲートウェイ

LPD インバウンド・ゲートウェイは CAA アプリケーションとして構成されませんが、クラスタの各メンバ上で実行できます。LPD インバウンド・ゲートウェイを構成するには、`/usr/pd/scripts/inbound_gw_config.sh` スクリプトを実行します。このスクリプトは、クラスタの各メンバに LPD インバウンド・ゲートウェイを構成し、`rc.config` ファイルを修正します。`rc.config` ファイルの修正は、システムが再起動するたびに LPD インバウンド・ゲートウェイを再起動するために必要です。

LPD インバウンド・ゲートウェイを手動で起動するには、`/sbin/init.d/apx` の `start` コマンドと `stop` コマンドを使用します。

セキュリティの管理

この章では、プリント・システム上にセキュリティ・ポリシを実装するために利用できるオプションについて説明します。セキュリティ・オプションは、管理機能に対する無許可のアクセスを防いだり、プリント・ジョブやプリント・データが無許可のユーザからアクセス、コピー、再ルート指定、あるいは削除されないように保護します。

この章では、次のセキュリティ機能を説明します。

- システム・セキュリティ — セキュリティ機能の概要とセキュリティの作動のしくみを説明します。
- 許可 — プrint・システムでサポートされるさまざまな特権レベルを説明し、それぞれの特権レベルを持つユーザによって実行できる機能を説明します。
- 認証 — プrint・システム・サービスに対して要求を行うユーザを識別して、許可するためにPrint・システムが使用する手法を説明します。

4.1 システム・セキュリティ

サーバに対して要求が行われるとき、その要求はリモート・プロシージャ・コール (RPC) の形式になります。この RPC 要求には、要求を作成したユーザの *UID* が含まれています。要求を作成したユーザを識別して、認証するために、Print・システムはローカル・パスワード・ファイルと NIS 分散パスワード・ファイルをチェックして、要求側の *UID* と有効な *UID* を照合します。この照合が一致する場合、スーパーバイザが、サーバ ACL にアクセスして、そのユーザが、要求したサービスの使用を許可されているかどうかを判断します。許可の際には、次の処理が行われます。

- 要求側のユーザ名またはグループ名と、ACL 内のユーザ名またはグループ名の照合が行われます。

- ユーザ名またはグループ名の照合が一致した場合は、要求したサービスと、ACL でユーザまたはグループに付与されている特権レベルとの照合が行われます。

ユーザ名またはグループ名と、要求者と操作の許可が、ACL 内で対応するものと一致した場合、操作が実行されます。特権を付与されていないサービスにアクセスしようとしたユーザは、要求したサービスを拒否されます。

4.2 許可

プリント・システムの許可は、ユーザと特権レベルとの関連付けに基づいています。プリント・システムのユーザに対しては、次のような 3 つのユーザ・レベルが存在します。

1. エンド・ユーザ — プリント・システムで印刷だけを行うユーザ。これは最小限の特権を持つユーザです。
2. オペレータ — 施設内のプリンタを制御するユーザ。オペレータには、プリンタやサーバの一時停止と再開、ジョブの再キューイング、サーバのシャットダウンを行うことができる特権があります。
3. 管理者 — オペレータ特権と、システムを構成するために必要な特権を持つユーザ。

プリント・システムは、ローカルの root アカウントにも特権を付与しています。これは、プリント・システム・サーバを実行しているシステム上の root アカウント (UID 0) です。ACL に何が定義されているかにかかわらず、ローカルの root ユーザには管理者特権があり、サーバの起動に必要な許可を持っているユーザはローカルの root だけです。

4.2.1 エンド・ユーザの操作

プリント・システム要求のクライアントがエンド・ユーザ許可を付与されている場合、クライアントは次の操作を行うことができます。

- そのユーザがアクセス権を持つサーバに関連付けられている論理プリンタに印刷する。
- 所有しているジョブを再実行する (クライアントがアクセス権を持っているサーバ経由で再度ジョブが実行される場合)。
- 所有しているジョブを取り消す。

- サーバ、キュー、プリンタに対する操作をリスト (pdls) する。
- 所有しているジョブやドキュメントに対する操作をリスト (pdls) する。
- 所有しているジョブに関する属性を設定または変更する。
- ジョブ・キュー操作をリスト (pdq) する。ただし、プリント・システムは、リストを行ったユーザに属していないジョブに関して戻された属性については、サーバの job-attributes-visible-to-all リストにある属性に限定して表示する。

4.2.2 オペレータの操作

プリント・システムのオペレータは、プリンタおよびジョブの日常的な管理機能を実行します。プリント・システム操作のクライアントにオペレータ許可が付与されていると、そのクライアントはエンド・ユーザが実行できるすべての操作の他に、次の操作を実行できます。

- サーバまたはキューのクリーン処理をする (pdclean)。
- サーバ、キュー、プリンタを使用可能または使用不能にする。
- サーバ、キュー、物理プリンタ、または任意のジョブを一時停止または再開する。
- ジョブを昇格(キューの先頭に移動) する。
- 物理プリンタの属性を設定または変更する。
- スプーラまたはスーパーバイザをシャットダウンする。

4.2.3 管理者

プリント・システムのインストール環境の中には、オペレータがおらず、エンド・ユーザと管理者だけの場合もあります。そのような場合、プリントおよびジョブの日常的な管理機能を実行するのは管理者になります。

プリント・システム要求のクライアントに管理者許可が付与されている場合、そのクライアントは次の操作を行うことができます。

- オペレータが実行を許可されているすべての操作。
- 全オブジェクトのすべての読み取り/書き込み属性を設定する (pdset)。
- サーバ、キュー、プリンタ、初期値ジョブ、初期値ドキュメントを作成する。

- 任意のオブジェクトを削除する (`pddelete`)。

4.3 アクセス制御リスト (ACL) による認証

エンド・ユーザ，オペレータ，管理者が実行を許可される操作は，ACL によって制御されています。これらの許可と ACL だけがサーバ・オブジェクトへのアクセスを制御します。ACL は，あるユーザのアクセスを特定のプリンタに限定しません。

ACL は，サーバに対する `access-control-list` 属性を定義することで作成されます。この属性は複数の値をとり，サーバが作成されたとき (`pdmakedb`)，または，`pdset` コマンドまたは GUI を使用していつでも，定義できます。ほとんどの場合，サーバを作成するときにはすべての ACL 属性は定義せず，サーバを作成した後にこれらの属性を追加します。次の表に，`access-control-list` 属性のフィールドについて説明します。

フィールド	説明
<code>name</code> <code>name@host</code>	ユーザ名およびホスト名 (省略可能)
<code>name-type</code>	ユーザのタイプ: <code>all-users</code> <code>user</code> <code>group</code> <code>netgroup</code>
<code>privilege-level</code>	アクセス・レベルのタイプ: <code>end-user</code> <code>operator</code> <code>administrator</code>

次の規則は，ACL にアクセスして，ユーザの許可を検証する際に適用されます。

- `name-type` が `all-users` の場合，サーバは `name` を無視する。つまり，サーバは指定された特権レベルに対するすべての要求を受け付ける。
- `name` を指定せず，また，`name-type` も指定しなければ，サーバは `name-type` を `all-users` に設定する。

- name-type が all-users 以外の場合は、name を指定する必要がある。また、要求を作成するユーザの名前は (パスワード・ファイル内の UID を検索することによって判断されるが)、次のいずれかである必要がある。
 - name-type=user の場合、name に正確に一致する。
 - name-type=group の場合、group のメンバ。
 - name-type=netgroup の場合、netgroup のメンバ。

name-type の netgroup 値は、特定のユーザを特定のホストに限定するので、これをオペレータと管理者の特権レベルに主に使用することは普通であると考えられます。

注意

ACL に ホスト名 (root@host_name) を指定しないで root を追加すると、ネットワーク環境内にあるすべてのリモート・システムの全 root ユーザがサーバに対して特権付きのアクセス権を持ってしまいますので、ACL に ホスト名 (root@host_name) を指定しないで root を追加してはなりません。

4.3.1 ACL の作成

コマンド行から、または pdprintadmin GUI を使用することによって、ACL の作成と変更を行うことができます。この項では、この両方のインタフェースを使用して ACL の作成と管理を行う方法について説明します。

- ユーザ smith を既存のスーパー ACL に追加するには、次のコマンドを使用します。

```
# pdset -c server \  
-x"access-control-list+={name=smith name-type=user privilege-level=end-user}" \  
red_spl
```

- スーパーバイザ ACL からユーザ jones を削除するには、次のコマンドを使用します。

```
# pdset -c server -x"access-control-list-={jones}" blue_sup
```

pdprintadmin GUI を使用して ACL の作成や変更を行うには、次の手順を使用します。

1. 「ツール」メニューから、「ユーザ管理」を選択します。

2. 「構成要素」ボックスから、スーパーラまたはスーパーバイザ・サーバを選択します。
3. 「ユーザ」ボックスで、「システム管理者」、「オペレータ」、「一般ユーザ」の中から、追加する「ユーザクラス」を選択します。
4. 選択したユーザ・クラスに対して、「全員」または「特定のユーザ」のいずれかを選択します。
5. 「特定のユーザ」を選択した場合、「追加」ボタンを押して新しいエントリを追加するか、または既存のエントリをクリックしてそのエントリを削除します。
6. 手順 5 で追加を選択した場合は、「新しいエントリの追加」ダイアログ・ボックスが表示されます。単一のユーザを追加する場合には「ユーザ」、グループを追加する場合には「グループ」、ネットグループの名前を追加する場合には「ネットグループ」をそれぞれ選択します。テキスト・フィールドにユーザ、グループ、ネットグループのいずれかの名前を入力して「OK」を選択します。

サーバの作成と管理

この章では、プリント・システムのスプーラとスーパーバイザの作成および管理方法について説明します。この章は次の節に分かれています。

- サーバの作成では、プリント・システム用のサーバを作成するために必要な手順を説明します。
- サーバの構成では、サーバに設定できる属性を説明します。
- TruCluster 環境におけるサーバの構成では、高可用性サーバの構成方法を説明します。
- サーバの管理では、サーバに対して実行する管理タスクを説明します。

5.1 スーパーバイザとスプーラの作成

プリント・システムを構成する最初の手順の1つは、スーパーバイザとスプーラの作成です。これらのサーバは、ユーザによって実行されたプリント・ジョブの流れを制御します。つまり、スーパーバイザとスプーラはユーザから実行されたジョブを受け取り、プリント・データをプリンタに渡し、プリンタとユーザ間で制御情報と状態情報を送受信します。

プリント・システム・サーバ・プロセスには、それぞれ、それに関連付けられたデータベースがあります。このデータベースにはサーバの構成属性が含まれています。

実行するサーバ・プロセスの作成は、次の2つの手順で行います。

1. サーバ用のオブジェクト・データベースを作成する。
2. サーバを起動する。

5.1.1 オブジェクト・データベースの作成

オブジェクト・データベースを作成するには、`pdmakedb` コマンドを使用します。オブジェクト・データベースはサーバ・プロセスを起動する前に作成する必要があります。

pdmakedb コマンドは、はじめは単一のオブジェクト、つまり、対応するサーバ・オブジェクトを含むオブジェクト・データベースを作成します。データベースの名前は、サーバ・オブジェクトの名前でもあります (-n オプションに注意)。このコマンドはサーバ・オブジェクトに対して、省略時の属性と、管理者がオプションを使用して指定した任意の属性を追加します。

指定するデータベースが既に存在する場合、pdmakedb コマンドはエラー・メッセージを表示して終了します。

サーバを作成するときには次の構文を使用してください。

```
pdmakedb [-A adminACL] [-h] [-n serverName] [-s spoolDir] [-t serverType]
```

-A *adminACL*

サーバに対するアクセス制御リスト (ACL) を指定します。省略時の値は {name group} です。

-h

pdmakedb で、サポートされるオプションのリストを表示します。

-n *serverName*

サーバの名前の他に、オブジェクト・データベースの名前を指定します。このオプションには次の規則が適用されます。

- 次の文字から構成される: A-Z, a-z, 0-9, -, _。

サーバ名を指定しないとき、サーバの省略時の名前は、スプーラの場合はホスト名_spl になり、スーパーバイザの場合はホスト名_sup になります。

-s *spoolDir*

-s オプションは、サーバが、印刷予定のドキュメントをスプールする場所を指定します。スプール・ディレクトリは任意の有効な UNIX ディレクトリ・パス名にできます。省略時の値は /var/spool/pd/*serverName* です (*serverName* は -n オプションで指定される)。

-t *serverType*

-t オプションは、オブジェクト・データベースのタイプ、つまり、スプーラまたはスーパーバイザを指定します。serverType として有効な値は、スプーラの場合は SPL、スーパーバイザの場合は SUP です。

5-2 サーバの作成と管理

このオプションを指定すると、`pdmakedb` はコマンド行で渡された属性の正当性を確認して、メッセージを表示します。

`pdmakedb` プログラムは、コマンド行で指定しない任意のオプションに対して省略時の値を提供します。次に示すのは、`pdmakedb` を使用してスーパーとスーパーバイザを作成する例です。オブジェクト・データベースは `root` アカウントから作成する必要があります。ACL にユーザを追加する必要があるときは、本書のセキュリティの章で説明されている手順を使用してください。

次の例は `pdmakedb` コマンドの使用方法を示しています。

- システム `blue` 上で、省略時の属性を使用するスーパーバイザに対してオブジェクト・データベースを作成するには、`root` アカウントから次のコマンドを実行します。

```
# pdmakedb -t SUP
```

- システム `blue` 上で、省略時の属性を使用するスーパーに対してオブジェクト・データベースを作成するには、`root` アカウントから次のコマンドを実行します。

```
# pdmakedb -t SPL
```

- アウトバウンド・ゲートウェイ・スーパーバイザを作成するには、次のコマンドを使用します。

```
# pdmakedb -t SUP -e /usr/pd/lib/pdspvlpr -n blue_obg
```

前述のコマンドを実行すると、結果として、サーバ・データベースの `blue_sup`、`blue_spl`、`blue_obg` が作成されます。

5.1.2 オブジェクト・データベース・データの表示

`pdshowdb` コマンドを使用すると、サーバ・オブジェクトに現在設定されているすべての属性の値が表示されます。表示される属性には、構成属性と非構成属性の両方、および、ユーザから見える属性に加えて、内部属性が含まれます。このコマンドは、どのように使用してもオブジェクト・データベースを変更しません。

`pdshowdb` がサーバ・オブジェクト属性を表示するために使用するフォーマットは、`-v` オプション・セットを指定した `pdmakedb` および `pdmddb` で使用するのと同じフォーマットです。

pdshowdb コマンドのコマンド行構文は次のとおりです。

pdshowdb [-e] [-h] [-n *serverName*]

-e

サーバを起動するために使用する実行可能ファイルの絶対パスを表示します。

-h

pdshowdb コマンドで使用可能なオプションを表示します。

-n *serverName*

オブジェクト・データベースの名前とそれに対応するサーバを指定します。

5.1.3 既存のオブジェクト・データベースの変更

pdmoddb コマンドを使用すると、既存のオブジェクト・データベースを変更できます。このコマンドは、属性が既にデータベースに存在している場合、コマンド行に指定された属性を変更します。指定された属性が存在しない場合、pdmoddb コマンドはその属性を作成して、指定された値を割り当てます。また、このコマンドは、まだ存在しておらず、コマンド行に指定されていない省略時の属性があれば、それを追加します。

pdmoddb コマンドは、処理中のサーバが使用しているデータベースを変更するために使用することはできません。サーバは、pdmoddb でデータベース・ファイルにアクセスする前にシャットダウンする必要があります。

指定されたデータベースが存在しないか、使用中である場合、pdmoddb コマンドはエラー・メッセージを表示して終了します。

pdmoddb コマンドのコマンド行構文は次のとおりです。

pdmoddb [-A *adminACL*] [-h] [-n *serverName*] [-s *spoolDIR*] [-t *serverType*]

-A *adminACL*

サーバに対するアクセス制御リスト (ACL) を指定します。省略時の値は {name group} です。

-h

pdmoddb で、サポートされているオプションのリストを表示します。

-n *serverName*

変更しようとしているサーバ名の他に、オブジェクト・データベースの名前も指定します。

-s *spoolDir*

-s オプションは、サーバが、印刷予定のドキュメントをスプールする場所を指定します。スプール・ディレクトリ *spoolDir* は、任意の有効な UNIX ディレクトリ・パス名にすることができます。省略時の値は `/var/spool/pd/serverName` です (*serverName* は -n オプションで指定される)。

-t *serverType*

-t オプションは、オブジェクト・データベースのタイプを、SPL または SUP のいずれかに指定します。

-v

このオプションを指定すると、pdmoddb は、コマンド行で渡される属性の正当性を確認して、メッセージを表示します。

5.1.4 オブジェクト・データベースのバックアップ

不注意によるデータ喪失やデータ破損に備えて、プリント・システムのオブジェクト・データベースを定期的にバックアップしておくといいでしよう。障害が発生して、オブジェクト・データベースにアクセスできなくなったり、オブジェクト・データベースが破損した場合は、最新のバックアップからシステムをリストアできるため、プリント・システム環境を最初から再作成する必要がありません。

次の領域のバックアップについて検討してください。

- `/var/pd/odb` ディレクトリには、サーバ・データベースが格納されています。この領域のバックアップをとっておくと、サーバ、プリンタ、およびキューの構成の他に、スプーラに格納されているジョブおよびドキュメント・オブジェクトが保存されます。

- ドキュメント・データ・ファイルは、`/var/spool/pd` ディレクトリに保存されています。ほとんどの場合、ジョブ・データは時間の影響を受けやすいため、システムのリストアにおいては、プリンタやキュー情報ほど重要ではありません。
- `/etc/printers.conf` ファイルには、サーバ・ホスト上のローカル・ネーム・ファイルが格納されています。`/var/pd/obj` 内のデータベースが完全な状態であれば、ローカル・ネーム・エントリは、サーバの起動時に自動的に作成されます。ただし、このファイルには、他のホスト上にあるサーバ、プリンタ、およびキューのエントリも格納されていることがあります。これらのエントリは起動時には追加されませんので、このファイルのバックアップをとっておくとよいでしょう。

サーバ・データベースは、サーバが稼働していないときにバックアップをとるようにしてください。これにより、バックアップ・コピーが内部的に一貫していることが保証されます。定期バックアップは、バックアップの間、プリント・システムを一時的にシャットダウンできる時間に、毎日または毎週実行します。

5.2 サーバ・オブジェクト属性の構成

設定や変更ができるサーバ・オブジェクト属性は数多くあります。次に、属性を示し、それらの属性がサーバに対してどのような効果があるのかを説明します。

`access-control-list`

スプーラまたはスーパーバイザの許可されたユーザを指定します。

`descriptor`

サーバを説明しているテキスト文字列です。

`job-attributes-visible-to-all`

`pdls` コマンド、および `pdq` コマンドを使用したときに、ジョブの所有者以外にも見えるジョブ属性とドキュメント属性のリストです。

`message`

サーバの状態を示すテキスト文字列です。

`notification-profile`

ユーザに通知されるイベントと、その通知方法を指定します。

hold-jobs-interrupted-by-printer-failure

再起動後にスプーラに戻されたジョブを、保留 (held) 状態または処理待ち (pending) 状態にするかどうかを指定します。

filter-definition

スーパーバイザでサポートされている 1 つ以上の翻訳フィルタまたは変更フィルタを定義して、そのフィルタの起動方法を定義します。PostScript プリンタにテキスト・ドキュメントを印刷したい場合は、8.2 節で説明しているように、text-to-PostScript 翻訳フィルタを設定する必要があります。

5.2.1 サーバ属性の設定例

次の例は、スプーラおよびスーパーバイザ用に共通なサーバ属性のいくつかを設定する方法を示します。

- `descriptor` 属性は、サーバの説明を提供するためのテキスト文字列です。この属性は、CLI および `pdprintadmin` GUI の両方を使用して、変更できます。次の例は、サーバに対して `descriptor` 属性を設定する方法を示しています。

```
# pdset -c server \  
-x "descriptor=Print spooler on node blue" blue_spl
```

- `job-attributes-visible-to-all` 属性は、すべてのユーザが `pdls` コマンドで、特定のジョブ属性を設定できるようにするために使用されます。

印刷処理の詳細説明を取得するために、`pdls` コマンドを実行するとき、さらに多くのジョブ属性をすべてのユーザに見えるように設定することもあります。次のコマンドは、`pdls` でジョブ所有者、ジョブ・サイズ、ジョブの実行時刻を表示するために属性を表示可能に設定します。

```
# pdset -c server \  
-x job-attributes-visible-to-all+= \  
{job-owner=yes submission-time=yes total-job-octets=yes} \  
blue_sup
```

- スーパーバイザ `blue_sup` に対して承認されたオペレータのリストに、ユーザ `smith` を追加するには、次のコマンドを使用します。

```
# pdset -c server -x "access-control-list+= \  
{name=smith name-type=user privilege-level=operator}" \  
blue_sup
```

5.3 TruCluster 環境におけるサーバの構成

スプーラおよびスーパーバイザ (サーバ) は、TruCluster Server 環境の高可用性シングル・インスタンス・アプリケーションとして構成することができます。シングル・インスタンス・アプリケーションは、クラスタのただ 1 つのメンバ上でインストールされ、構成され、そして実行されますが、クラスタの全メンバで使用することができます。

高可用性サーバは、TruCluster Server CAA (Cluster Application Availability) サブシステムを使用するように構成されます。CAA は、クラスタ内でサーバによって要求されたリソースを監視し、リソースの必要性に合ったクラスタ・メンバ上で実行するようにします。サーバを実行中のクラスタ・メンバや、要求されたリソースに障害が発生した場合には、CAA は再配置、つまり要求されたリソースを持つ他のメンバにサーバをフェイルオーバーします。

サーバを高可用性シングル・インスタンスとして構成および実行するには、次の準備が必要となります。

- リソース・プロファイル
- 処理スクリプト
- アプリケーションを CAA に登録
- CAA 環境でのアプリケーションの起動

これらの各項目については、以降の各項で説明します。詳細については、『*TruCluster Server* 高可用性アプリケーション・ガイド』を参照してください。

5.3.1 リソース・プロファイル

リソース・プロファイルは、CAA によるアプリケーションの起動、管理、および監視方法を定義します。

Advanced Printing Software サーバのサブセット (APXSVRxxx) を構成するときには、TruCluster ソフトウェアがシステムにインストールされているかどうかを判断します。TruCluster Server ソフトウェアがインストールされている場合には、構成スクリプトが CAA リソース・ファイル `apx-default.cap` を `/var/cluster/caa/profile` ディレクトリにインストールします。これは、プリント環境で動作するサーバおよびスーパーバイザを表すリソース・プロファイルです。ユーザが作成するスプーラおよびスーパーバイザは、`apx-default` リソースに格納されます。追加のリソースを

作成して、スプーラおよびスーパーバイザを `apx-default` リソース・プロファイルから作成したリソースへ移動することもできます。

5.3.2 処理スクリプト

処理スクリプトは、アプリケーションの起動方法、アプリケーションの停止およびフェイルオーバー前のクリーンアップ方法、およびアプリケーションが実行中であるかどうかの確認方法を指定します。

Advanced Printing Software を TruCluster Server ソフトウェアが存在するシステムにインストールする場合には、
`/var/cluster/caa/script/apx-default.scr` 処理スクリプトがインストールされます。

5.3.3 リソースの登録

リソース・プロファイルおよび処理スクリプトのインストール後、CAA にリソースを登録する必要があります。 `apx-default` リソースを登録するには、`caa_register` コマンドを使用してください。

```
# /usr/sbin/caa_register apx-default
```

この CAA リソースを一旦登録すると、サーバを起動およびシャットダウンするには CAA コマンドを使用しなくてはなりません。 `pdsplr`、`pdspvr`、または `pdshutdown` コマンドを使用しないでください。次の項で、CAA コマンドがどのようにサーバを起動およびシャットダウンするかを説明します。

5.3.4 リソースの起動と停止

CAA に `apx-default` を登録すると、`caa_start` コマンドを使用してリソースを起動できます。

```
# /usr/sbin/caa_start apx-default
```

リソースが起動されると次のようなメッセージが表示されます。

```
Attempting to start 'apx-default' on member 'membername'  
Start of 'apx-default' on member 'membername' succeeded.
```

アプリケーションを停止するには、`caa_stop` コマンドを使用します。

```
# /usr/sbin/caa_stop apx-default
```

5.3.5 apx-default リソースへのサーバの追加

クラスタ環境でサーバを構成後、リソースに追加のスプーラおよびスーパーバイザが必要になる場合があります。省略時のリソースにサーバを追加するには、まず、追加したいスプーラおよびスーパーバイザを起動します。/usr/pd/scripts/pd_get_started スクリプトまたは pdmakedb, pdsplr, および pdspvr コマンドを使用することができます。新しいサーバが作成され、apx-default リソースの一部となります。

5.3.6 Advanced Printing クラスタ環境のカスタマイズ

リソースを再配置したり、新しいリソースを作成したり、リソースを変更および削除したりして、CAA のプリント環境をカスタマイズすることができます。以降の各項で、これらのトピックについて説明します。

5.3.6.1 リソースの再配置

リソースを再配置する場合には、あるクラスタ・メンバから他のクラスタ・メンバへリソースを移動します。リソースを再配置するには、caa_relocate コマンドを使用してください。リソースの再配置先のクラスタ・メンバを指定したり、CAA に有効なメンバを判断させることができます。リソースを再配置する場合には、関連するサーバをシャットダウンしてから新しいメンバで再起動します。

次の例で、リソースを再配置するための caa_relocate コマンドの使い方を示します。

apx-default リソースをメンバ goofy からメンバ daffy に再配置するには、次のコマンドを使用します。

```
# caa_relocate apx-default -c daffy
```

このコマンドに対して次のようなメッセージが表示されます。

```
Attempting to stop 'apx-default' on member 'goofy'  
Stop of 'apx-default' on member 'goofy' succeeded.  
Attempting to start 'apx-default' on member 'daffy'  
Start of 'apx-default' on member 'daffy' succeeded.
```

アプリケーション・リソース・プロファイルに定義された配置ポリシーに従ってリソースを再配置するには、オプションを指定しないで caa_relocate コマンドを使用してください。

```
# caa_relocate apx-default
```

5.3.6.2 CAA リソースを管理するための `apx_caa_setup` スクリプトの使用

ユーザの CAA のプリント環境をカスタマイズするには `apx_caa_setup` スクリプトを使用してください。このスクリプトは、`/usr/pd/cluster` ディレクトリにあり、次のことを行うことができます。

- 現在の CAA プリント・リソースの表示
- 新しい CAA プリント・リソースの作成
- CAA プリント・リソースの変更
- CAA プリント・リソースの削除
- CAA プリント・リソースの初期化

`apx_caa_setup` スクリプトのメイン・メニューを次に示します。

```
Advanced Printing Software Cluster Setup
```

```
*** MAIN MENU ***
```

- 1 Display CAA printing resources
- 2 Create CAA printing resource
- 3 Modify CAA printing resource
- 4 Delete CAA printing resource
- 5 Initialize CAA resources for Advanced Printing
- 6 Exit

```
Enter the number that corresponds to your choice:[6]
```

CAA リソースの表示

Display CAA printing resources 機能により、Advanced Printing 環境のすべての CAA リソースを表示することができます。apx-default リソースは、ユーザが作成したリソースと同様に表示されます。各リソースには、リソースの一部であるサーバのリストが含まれます。

```
Enter the number that corresponds to your choice:[6] 1
```

```
Currently defined Advanced Printing CAA Resources
```

```
1 mikes: green_spl green_sup
2 robs: rob_spl rob_sup
3 orange: orange_spl orange_sup
4 BLExit: root_sup root_spl
5 blue: blue_sup blue_spl

apx-default: daffy_spl daffy_sup
```

CAA プリント・リソースの作成

apx_caa_setup スクリプトを使用して CAA リソースを作成する場合には、作成するリソースと関連付けるスプーラおよびスーパーバイザについての問い合わせ後、そのリソースが CAA に登録されます。次に、このプロセスの出力例を示します。

```
Enter the number that corresponds to your choice:[6] 2
```

```
Currently defined Advanced Printing CAA Resources
```

```
1 mikes: green_spl green_sup
2 robs: rob_spl rob_sup
3 orange: orange_spl orange_sup
4 BLExit: root_sup root_spl
5 blue: blue_sup blue_spl

apx-default: daffy_spl daffy_sup
```

```
Enter a unique name for the new CAA resource: resourceX
```

```
Enter the names of one or more Advanced Printing spooler
or supervisors to be associated with the resource "resourceX":
```

```
resourceX: orange_sup orange_spl
```

```
Creating /var/cluster/caa/script/resourceX.scr ...
Creating /var/cluster/caa/profile/resourceX.cap ...
Registering resourceX with CAA...
```

CAA プリント・リソースの変更

CAA プリント・リソースによって管理されているサーバを変更するには、apx_caa_setup スクリプトを使用してください。リソースを変更する場合には、サーバを追加するか、またはリソースからサーバを削除します。リソースからサーバを削除する場合、そのサーバは apx-default リソースに格納されます。

次の例では、2つのサーバ、`blue_sup` および `blue_spl` を CAA プリント・リソース `orange` に追加する方法を示しています。サーバ `orange_spl` および `orange_sup` は既にリソースに関連付けられていますが、CAA リソース `orange` に他のものを追加する場合には、再度入力しなくてはならないことに注意してください。このもともとのサーバを再入力しない場合には、サーバは `apx-default` リソースに戻されてしまいます。

```
Enter the number that corresponds to your choice:[6] 3
```

```
Currently defined Advanced Printing CAA Resources
```

```
1 mikes: green_spl green_sup
2 robs:  rob_spl  rob_sup
3 orange: orange_spl orange_sup
4 BLExit: root_sup root_spl
5 blue:  blue_sup blue_spl
6 apx_orange: orange_sup orange_spl
```

```
apx-default: daffy_spl daffy_sup
```

```
Modify which resource? 3
```

```
Modify CAA map entry "orange"? ([y]/n)
```

```
Enter the names of one or more Advanced Printing spooler
or supervisors to be associated with the resource "orange":
```

```
orange: orange_spl orange_sup blue_spl blue_sup
Modifying resource orange...
```

CAA プリント・リソースの削除

`apx_caa_setup` スクリプトを使用して CAA プリント・リソースを削除すると、そのリソースは停止して、登録が取り消され、処理スクリプトおよびリソース・プロファイルが削除されます。

CAA プリント・リソースの初期化

CAA プリント・リソースを初期化する場合には、次のタスクが実行されます。

- すべての Advanced Printing サーバは、`apx-default` リソース以外の CAA リソースとの関連付けが解除されます。
- すべてのプリント・リソースが停止して、登録が取り消されます。スプーラおよびスーパーバイザの実行は続きます。

- /var/pd/config/apx_caa_map.conf ファイルに定義されているプリント・リソースに関連付けられているプロファイルおよび処理スクリプトを削除します。
- 新しい、空の /var/pd/config/apx_caa_map.conf ファイルを作成します。
- すべてのサーバを apx-default リソースに戻し、リソースを CAA に登録します。

初期化中に表示される出力例を次に示します。

```
Enter the number that corresponds to your choice:[6] 5
```

```
The Advanced Printing CAA map file already exists.
```

```
This option performs the following tasks:
```

- Disassociates all Advanced Printing spoolers and supervisors from named CAA resources.
 - Stops and unregisters printing resources, but leaves spoolers and supervisors running.
 - Deletes resource profiles and action scripts associated with printing resources defined in the /var/pd/config/apx_caa_map.conf file.
 - Creates a new, empty /var/pd/config/apx_caa_map.conf file.
 - Registers the apx-default resource.
- Do you want to initialize anyway? (y/[n])

5.3.7 1つのメンバで実行するためのプリンタの構成

いくつかのプリンタは、ホストのシリアルまたはパラレル・ポートに直接接続されます。これらのプリンタは、そのホスト上で実行されているスーパーバイザによって制御されなくてはなりません。正しいクラスタ・メンバ上でスーパーバイザが実行されるようにするには、そのスーパーバイザに対して別個の CAA リソースを作成する必要があります。リソース・プロファイルには、次のように指定します。

- placement=restricted
- hosting_members=hostname

物理プリンタ・オブジェクトの作成後、printer-associated-host 属性にプリンタをアクセスするホストの名前を設定します。

```
# pdset -c printer -x printer-associated-host=hostname
printername
```

さらに、プリンタに関連付けられているスーパーバイザは、そのプリンタが直接接続されているホスト上で実行する必要があります。

`printer-associated-host` 属性で指定されている以外のホスト上で実行中のスーパーバイザからジョブをプリンタに送信すると、そのスーパーバイザは物理プリンタを使用不能にして、次の属性を設定します。

- `enabled=false`
- `availability=none`
- `printer-problem-message= " Printer port associated with a different host "`

5.3.8 クラスタにおける LPD インバウンド・ゲートウェイ

LPD インバウンド・ゲートウェイは CAA アプリケーションとして構成されませんが、クラスタの各メンバ上で実行できます。LPD インバウンド・ゲートウェイを構成するには、`/usr/pd/scripts/inbound_gw_config.sh` スクリプトを実行します。このスクリプトはクラスタの各メンバ上で LPD インバウンド・ゲートウェイを構成し、`rc.config` ファイルを修正して、システムが再起動するたびに LPD インバウンド・ゲートウェイを再起動できるようにします。

`/sbin/init.d/apx` の `start` コマンドおよび `stop` コマンドによって LPD インバウンド・ゲートウェイを起動および停止します。

5.4 サーバの管理

この節では、スプーラおよびスーパーバイザに対して実行できる管理タスクについて説明します。これらのタスクには、サーバの起動と停止、一時停止と再開、使用可能設定と使用不能設定、そして、削除も含まれます。

5.4.1 サーバの状態

サーバ属性の `server-state` と `enabled` は、サーバ・プロセスの状態を決定します。サーバ・プロセスは次の状態のいずれか 1 つになります。

- `ready` (処理可能) — サーバはプリント・ジョブを受け付け可能です。これはサーバの通常の処理状態です。

- **terminating** (後処理中) — サーバはシャットダウン処理の最中であり，終了中です。
- **paused** (一時停止) — サーバは一時停止され，再開されるまでジョブを受け付けません。

`server-state` は，`ready` (処理可能) に設定されていても，サーバが使用不能の場合には，プリント・ジョブを受け付けません。

次のコマンドを使用して，サーバの状態を調べてください。

```
# pdls -c server blue_sup
```

次に示すように，このコマンドの省略時の設定では，`server-state` 属性と `enabled` 属性の両方を返します。

```
server-name server-state enabled
-----
blue_sup    ready      yes
```

5.4.2 サーバの起動

サーバの起動には，それぞれのサーバ・タイプに対する実行可能ファイルを実行することが含まれます。スタートアップ構文は，シェル・プロンプトでタイプできる文字列です。ただし，一般に，サーバの起動はシェル・スクリプトの形式で行い，ホストはそのシェル・スクリプト内部からスタートアップ構文を実行します。シェル・スクリプトは，管理者が作成するスタートアップ・スクリプトの一部にすることも，あるいは，UNIX システムのブート時に通常実行されるスクリプトにすることもできます。Tru64 UNIX システムでは，このファイルは `/sbin/init.d/apx` になります。

プリント・システム・サーバは UNIX デーモンに似た動きをします。プリント・システム・サーバは，ブート時に，ユーザからの入力なしに，シェル・スクリプトによって起動します。典型的な UNIX デーモンと同様，サーバはバックグラウンドで起動し，ランタイム・ディレクトリに対して相対的に動作します。スプーラまたはスーパーバイザを起動するときには，`root` アカウントから起動する必要があります。

5.4.3 スプーラの起動

プリント・システムのスプーラを起動するコマンド構文は次のとおりです。

```
/usr/pd/lib/pdsplr [-a] [-c] [-d ODBpath] [-e emailAddress] [serverName]
```

-a

-a オプションは、クラッシュからの回復処理の一部として、オブジェクト・データベースをサーバが修復しなければならなかった後に行われるサーバのスタートアップ時だけに必要です。

-c

このオプションは、正常なシャットダウンの後のスタートアップ時に、データベースの完全性チェックを強制します (クラッシュ後の再起動時には、サーバはデータベースの完全性を自動的にチェックします)。

-d *ODBpath*

オブジェクト・データベース・ファイルの場所を指定します。サーバはスタートアップ時にサーバ名を必要とし、これによってオブジェクト・データベースの場所を特定できます。省略時の設定では、サーバは、オブジェクト・データベースがすべてのサーバ・オブジェクト・データベース用の標準位置 (`/var/pd/odb`) に存在すると想定します。

-e *emailAddress*

-e オプションは、サーバ・スタートアップ・エラーの通知を受け取る電子メール・アドレスを指定するために使用します。メール・アドレスを指定しない場合には、サーバ・スタートアップ・エラーは `standard error` および `syslog` に通知されます。

```
/usr/pd/lib/pdsplr -e someuser@somesystem.com myserver
```

次に示すのは、スプーラを起動する `pdsplr` コマンドの使用例です。

- オプションを付けずにスプーラ `red_spl` を起動するには、次のコマンドを使用します。

```
# /usr/pd/lib/pdsplr red_spl
```
- 次のコマンドを使用すると、`red_spl` が起動され、スタートアップ・エラーがあれば `root` アカウントに電子メールで通知されます。

```
# /usr/pd/lib/pdsplr -e root red_spl
```

5.4.4 スーパーバイザの起動

プリント・システムのスーパーバイザを起動するためのコマンド構文は次のとおりです。

`/usr/pd/lib/pdspvr [-a] [-c] [-dODBpath] [-emailAddress] [serverName]`

-a

このオプションは、クラッシュからの回復処理の一部としてオブジェクト・データベースをサーバが修復しなければならなかった後に行われるサーバ・スタートアップでの必須オプションです。

-c

-c オプションは、正常なシャットダウンの後のスタートアップ時に、データベースの完全性チェックを強制する、省略可能なオプションです。データベースが破損している可能性があるときには、-c オプションを使用するとよいでしょう。

-e *emailAddress*

-e オプションは、サーバ・スタートアップ・エラーの通知を受け取る電子メール・アドレスを指定するために使用します。電子メール・アドレスを指定しない場合には、サーバ・スタートアップ・エラーは standard error および syslog に通知されます。

プリント・システムのアウトバウンド・ゲートウェイ・スーパーバイザを起動するためのオプションは、(前述の) スーパーバイザ用のオプションと同じですが、コマンドが違います。アウトバウンド・ゲートウェイ・スーパーバイザを起動するには、次のコマンドを使用してください。

```
# /usr/pd/lib/pdspvlpr
```

次に示すのは、スーパーバイザを起動する `pdspvr` コマンドと `pdspvlpr` コマンドの使用例です。

- オプションを付けずにスーパーバイザを起動するには、次のコマンドを使用します。

```
# /usr/pd/lib/pdspvr red_sup
```

- 次のコマンドを使用すると、スーパーバイザが起動され、スタートアップ・エラーがあれば、root アカウントに電子メールで通知されます。

```
# /usr/pd/lib/pdspvr -e root red_sup
```

- 次のコマンドを使用すると、アウトバウンド・ゲートウェイ・サーバが起動します。

```
# /usr/pd/lib/pdspvlpr red_obg
```

5.4.5 サーバを使用可能にする

はじめてサーバを作成して、起動した後、他のユーザから利用できるようにする前に、サーバを使用可能にする必要があります。

pdenable [-c *server*] [-mmessage_txt] [server name]
-m *message text*

このオプションには、サーバが使用可能になったときのメッセージを含めることができます。このメッセージは `pdls` コマンドで取得することができます。

次に示すのは、`pdenable` コマンドを使用してスプーラを使用可能にする例です。

- スプーラ `red_spl` を使用可能にするには、次のコマンドを使用します。

```
# pdenable -c server red_spl
```

5.4.6 サーバを使用不能にする

`pddisable` コマンドを使用するとサーバが使用不能になります。スプーラが使用不能にされると、それ以前に実行されたジョブはすべてスケジューリングされて、物理プリンタに渡されますが、新しいジョブの要求はすべて拒否されます。また、スーパーバイザが使用不能にされると、スーパーバイザによって処理中のジョブは最後まで処理が行われて、新しいジョブはすべて拒否されます。

`pddisable` コマンドの構文は次のとおりです。

pddisable [-c *class_name*] [-m *message_txt*] [-x *extended_attribute_string*] [-X *attribute_filename*] [- server name]
-c *class_name*

使用不能にしようとしているオブジェクトのクラスとタイプを指定します (この場合、クラスはサーバ)。

-m *message text*

このオプションを使用すると、`pdls` コマンドで取得できるメッセージを含めることができます。

-x extended_attribute_string

このオプションを使用すると、`pddisable` コマンドを実行したときに設定される「属性タイプ = 値」のペアを1つ以上指定できます。

-X attribute_filename

このオプションを使用すると、`pddisable` コマンドを実行したときに設定される「属性タイプ = 値」の一連のペアを含むファイルの名前を指定できます。

- スプーラ `red_spl` を使用不能にするには、次のコマンドを使用します。

```
# pddisable -c server red_spl
```

5.4.7 サーバからのジョブの消去

サーバからジョブを消去するには、`pddelete` と `pdclean` の2つのコマンドが使用できます。`pdclean` コマンドは、1つのサーバにあるすべてのジョブを消去します。`pddelete` コマンドは、1つのサーバから名前を指定した1つ以上のジョブを消去するために使用します。

5.4.7.1 サーバからのジョブの削除

`pddelete` コマンドは、1つのサーバから名前を指定したジョブを1つ以上消去するときに使用します。

```
pddelete [-c job] [-x extended_attribute_string] [-X attribute_filename]  
[job_id]
```

-x extended_attribute_string

このオプションを使用すると、`pddelete` コマンドを実行したときに設定される「属性タイプ = 値」の一連のペアを指定できます。

-X attribute_filename

このオプションを使用すると、`pddelete` コマンドを実行したときに設定される「属性タイプ = 値」の一連のペアを含むファイルの名前を指定できます。

次に示すのは、`pddelete` コマンドを使用してサーバから特定のジョブを削除する例です。

- スプーラ *blue_spl* 上のジョブ 155 を削除するには、次のコマンドを使用します。

```
# pdelete -c job blue_spl:155
```

5.4.7.2 スプーラからのすべてのジョブの消去

`pdclean` コマンドは 1 つのスプーラにあるすべてのジョブを消去します。`pdclean` を使用してスプーラからジョブを消去する際には、次の事項を考慮してください。

- クリーン処理は非同期です。処理が完了しなくても新しいシェル・プロンプトが表示されます。
- 処理を実行する前に、スプーラを使用不能にする必要があります。
- 保持されたジョブ、完了したジョブ、現在処理中のジョブ、印刷中のジョブが削除されます。

`pdclean` コマンドのフォーマットは次のとおりです。

```
pdclean [-c server] [-mmessage_txt] [server name]  
-m message text
```

サーバ・メッセージ・テキストを設定します。`pdls` コマンドを使用すると、このメッセージを表示できます。

- `pdclean` コマンドを使用して *red_spl* にあるすべてのジョブを消去するには、次のコマンドを使用します。

```
# pdclean -c server red_spl
```

5.4.8 サーバの一時停止

スプーラを一時停止するには `pdpause` コマンドを使用します。スーパーバイザを一時停止することはできません。スプーラが一時停止されると、ジョブをスーパーバイザに引き渡しません。スプーラに関連付けられた論理プリンタは引き続き新しいジョブを受け付けます。

`pdpause` コマンドの構文は次のとおりです。

```
pdpause [-c server] [-m message_txt] [-x extended_attribute_string] [-X  
attribute_filename] [server_name]
```

`-m message text`

このオプションには、サーバが一時停止されたときのメッセージを含めることができます。このメッセージは `pdls` コマンドで取得することができます。

`-x extended_attribute_string`

このオプションを使用すると、コマンドを実行したときに設定される「属性タイプ = 値」の一連のペアを指定できます。

`-X attribute_filename`

このオプションを使用すると、コマンドを実行したときに設定される「属性タイプ = 値」の一連のペアを含むファイルの名前を指定できます。

次に示すのは、`pdpause` コマンドを使用してサーバを一時停止する例です。

- `red_spl` を一時停止するには、次のコマンドを使います。

```
# pdpause -c server red_spl
```

5.4.9 サーバの再開

`pdresume` コマンドを使用すると、`pdpause` コマンドによって一時停止されていたサーバの処理を再開できます。`pdresume` コマンドの構文は次のとおりです。

```
pdresume [-c server] [-m message_txt] [-x extended_attribute_string] [-X attribute_filename] [server_name]
```

`-m message text`

このオプションでは、サーバ処理が再開されたときのメッセージを含めることができます。このメッセージは `pdls` コマンドで取得することができます。

次に示すのは、`pdresume` コマンドを使用してサーバを再開する例です。

- スプーラを再開し、サーバ・メッセージを設定するには、次のコマンドを使用します。

```
# pdresume -c server /  
-m "Spooler operation resumed at 13:00" red_spl
```

5.4.10 サーバのシャットダウン

`pdshutdown` コマンドを使用すると、サーバ・プロセスをシャットダウンできます。このコマンドが実行されると、スプーラとスーパーバイザは自分自身を使用不能にしてから、`when_time` オプションの値で指定された方法によってシャットダウンします。

pdshutdown [-c *server*] [-w *when_time*] [-x *extended_attribute_string*] [-X *attribute_filename*] [*server_name*]

-w *when_time*

このオプションは、サーバがシャットダウンされる前にどの程度の処理ができるかを指定します。次の引数が指定可能です。

- **now**
 - サーバがスプーラの場合、ただちにシャットダウンされます。スーパーバイザに対して完全に実行されていないジョブは、完了できないことがあります。
 - サーバがスーパーバイザの場合、すべてのジョブが取り消され、後で処理するためにスプーラに戻されます。
- **after-current** (省略時の値)
 - サーバがスプーラの場合、スーパーバイザに対して新しいジョブの送信を停止して、既に送信したジョブの処理が完了した後でシャットダウンします。
 - サーバがスーパーバイザの場合、現在印刷中のジョブがすべて完了してからシャットダウンします。
- **after-all**
 - サーバがスプーラの場合、新しいジョブの受け付けを停止し、現在印刷中のジョブがすべて完了してからシャットダウンします。
 - サーバがスーパーバイザの場合、新しいジョブの受け付けを停止し、現在印刷中のジョブがすべて完了してからシャットダウンします (**after-current** と同じことに注意)。
いずれかのプリンタが一時停止され、プリント・ジョブが待機中である場合、サーバはシャットダウンしないことに注意してください。

次に示すのは、サーバを停止するために `pdshutdown` コマンドを使用する例です。

- 現在印刷中のジョブすべてが完了した後でスプーラをシャットダウンするには、次のコマンドを使用します。

```
# pdshutdown red_spl
```

5.4.11 サーバの削除

`pddelete` コマンドを使用すると、サーバを削除できます。サーバを削除するときには、次の項目を考慮してください。

サーバがスプーラの場合：

- サーバを使用不能にして、スプーラ上に現在存在するすべてのジョブを削除する必要があります。
- スーパーバイザは `associated-queue` 属性と `printer-associated-printers` 属性を設定して、スプーラに関連付けられているすべての物理プリンタを空にします。
- 関連付けられたすべてのサーバ・オブジェクト (キュー、論理プリンタ、初期値ジョブ、初期値ドキュメント) は削除されます。
- サーバは終了します。

削除する予定のサーバがスーパーバイザの場合：

- スーパーバイザ上にアクティブなジョブが存在しないこと、サーバを使用不能にすることが必要です。
- 関連付けられたスプーラは、すべてのマッピング属性から物理プリンタへの参照を消去して、影響を受けるキューと論理プリンタを更新します。
- 関連付けられたすべての物理プリンタ・オブジェクトとスーパーバイザ・データベースは削除されます。
- スーパーバイザは終了します。

`pddelete` コマンドを使用して、サーバを削除します。このコマンドの構文は次のとおりです。

```
pddelete [-c server] [server_name]
```

- 次の例のコマンドは、スプーラ blue_spl , 関連付けられたすべての論理プリンタ , 関連付けられたすべてのキュー , 残っているすべてのジョブを削除します。

```
# pddisable -c server blue_spl
# pdclean -c server blue_spl
# pddelete -c server blue_spl
```

- 次の例のコマンドでは、スーパーバイザ blue_sup とサーバの物理プリンタ・オブジェクトすべてを削除します。

```
# pddisable -c server blue_sup
# pddelete -c server blue_sup
```

キューとプリンタの作成および管理

この章では、キュー、論理プリンタ、物理プリンタの作成および管理の方法と、プリンタに対する省略時のオブジェクトの作成方法を説明します。この章は次の項に分けられています。

- キューの作成と管理
- 物理プリンタの作成と管理
- 論理プリンタの作成と管理
- 省略時のオブジェクトの作成

この章で使用されるコマンドの詳細説明については、『*Advanced Printing Software* コマンド・リファレンス・ガイド』を参照してください。

6.1 キューの作成と管理

論理プリンタと物理プリンタの抽象化は、キューによって結び付けられます。1つの論理プリンタと1つの物理プリンタを1つのキュー・オブジェクトに関連付けることによって、ユーザが印刷のために指定するプリンタと、出力デバイスの特性によって定義された物理プリンタとの関係を確立することになります。

キューを作成するときには、キューをスプーラに関連付けます。関連付けられたスプーラは、次の処理を行います。

- ローカル・ファイルの `/etc/printers.conf` ファイルにエントリを作成することで、キューをネーム・サービスに登録します。
- 新しいキューを使用不能として初期化します。
- 新しいキューの名前をスプーラ属性 `queues-supported` に追加します。

キューの作成は、`pdcreate` コマンドまたは `pdprintadmin` GUI を使用して行います。`pdprintadmin` GUI を使用してプリンタを作成すると、プリント・システムでキューを自動的に作成するかどうか尋ねられます。ま

た、`pdcreate` を使用してキューを作成するとき、`pdcreate` の構文は次のとおりです。

```
pdcreate [-c queue] [-x extended_attribute_string] [-X attribute_filename]  
[queue_name]
```

- `production_q` という名前のキューを作成し、それを `red_spl` という名前のスプーラに関連付けるには、次のコマンドを使用します。

```
# pdcreate -c queue red_spl:production_q
```

6.1.1 キューを使用可能にする

`pdenable` コマンドを使用するとキューを使用可能にできます。キューを使用可能にすると、キューはプリント・ジョブを受け付けます。キューを使用可能にする場合、`pdenable` コマンドの構文は次のようになります。

```
pdenable [-c queue] [-m message text] [-x extended_attribute_string] [-X  
attribute_filename] [queue_name]
```

6.1.2 キューからのジョブの削除

オペレータと管理者は `pdclean` コマンドを使用して、1 つのキューからすべてのジョブを削除することができます。

`pdclean` コマンドを使用してキューからすべてのジョブを削除すると、関連付けられた物理プリンタで、現在印刷中、保持中、または、処理中のジョブは削除されます。このコマンドを実行する前に、キューを使用不能にする必要があります。

- `production_q` という名前のキューからすべてのジョブを削除するには、次のコマンドを使用します。

```
# pdclean -c queue production_q
```

6.1.3 キューの状態

`state` 属性には、キューの現在の状態が入っています。キューは次の 2 つの状態のいずれかになります。

- `ready` (処理可能) — キューの通常の動作状態。
- `paused` (一時停止) — キューは一時停止され、`ready` 状態に戻るまで物理プリンタにジョブを送信しません。

6-2 キューとプリンタの作成および管理

キューは `pdpause` コマンドで一時停止され、`pdresume` コマンドで再開されます。

6.1.4 キュー属性のリスト

`pdls` コマンドを使用すると、キューの属性を表示できます。

- `production_q` という名前のキューの `logical-printers-ready` 属性と `physical-printers-ready` 属性を表示するには、次のコマンドを使用します。

```
# pdls -c queue production_q
```

このコマンドの出力は、次のとおりです。

```
queue-name logical-printers-ready physical-printers-ready
-----
production_q colorPS          hpcolorPS
```

- キュー `mail_q` のすべての属性をリストするには、次のコマンドを使用します。

```
# pdls -c queue -r all -s line mail_q
```

このコマンドの出力は、次のとおりです。

```
mail_q: object-class = queue
mail_q: queue-name = mail_q
mail_q: associated-server = red_spl
mail_q: enabled = no
mail_q: availability = none
mail_q: state = ready
```

6.2 物理プリンタの作成と管理

プリント・システムでは、物理プリンタとは出力デバイスを表すオブジェクトのことです。

サポートされる物理プリンタには、それぞれのプリンタのためのプリンタ属性ファイル (Printer Attribute File - PAF) があります。PAF ファイルには、プリンタとの通信とその制御に必要な属性の推奨設定が含まれています。プリンタ属性ファイルは名前のフォーマットが `printer_model.paf` なので、新しい物理プリンタ用の適切なプリンタ属性ファイルは、プリンタ・モデルによって決定できます。たとえば、`Digital_DEClaser5100_level2PS.paf` は Digital DEClaser5100 プリンタ用のプリンタ属性ファイルになります。

paf ファイルは `/usr/pd/share/cap` ディレクトリに格納されています。

6.2.1 物理プリンタの作成

pdcreate コマンドを使用すると物理プリンタを作成できます。pdcreate コマンドの構文は次のとおりです。

```
pdcreate [-c printer] [-x extended_attribute_string] [-X attribute_filename]  
[printer_name]
```

物理プリンタを作成する際に指定する必要がある属性は、次のとおりです。

- `printer-model` — プリンタのモデル名。
- `printer-address` — プリンタのデバイス・アドレスまたはネットワーク・アドレス。
- `associated-queue` — プリンタが関連付けられているキュー (1 つまたは複数) の名前。
- `printer-connection-method` — シリアル接続, パラレル接続, ip-socket 接続, LPD 接続, `digital-printserver` 接続がサポートされています。

プリンタ属性ファイルには, `printer-model`, `printer-connection-level`, `printer-tcpip-port-number` 属性の省略時の値が含まれています。他の属性の値については, 物理プリンタを作成するときに指定するか, 物理プリンタを作成した後でその値を設定または変更できます。

6.2.1.1 プリンタ・アドレスの設定

スーパーバイザでは, 接続されているすべてのプリンタについて, `printer-address` 属性に一意な値が指定されていることが必要です。この属性に値が割り当てられていない場合, 物理プリンタを使用可能にすることはできません。

次の表は, スーパーバイザが認識する `printer-address` 属性の一般的な形式です。この形式は, シリアル, パラレル, IP ネットワーク・ソケットの各接続を示しています。IP ネットワーク・ソケット・ポートは, シンプルなノード名, ドット付きまたはドットなしのドメイン構文のアドレス, ポート番号付きまたはポート番号なしのアドレスのいずれかになります。

表 6-1: プリンタ・アドレス・フォーマット

形式	説明	接続方法	スーパーバイザ
/dev/lp<n>	パラレル・ポート n	パラレル	pdspvr
/dev/tty<nn>	シリアル・ポート nn	シリアル	pdspvr
name[:port]	ホスト名 (ドメインなし) とポート	IP ソケット	pdspvr
name.dom[:port]	ドメインを指定したホスト名とポート	IP ソケット	pdspvr
address [:port]	数値の IP アドレスとポート	IP ソケット	pdspvr
ホスト, プリンタ, 拡張	アウトバウンド・ゲートウェイ・プリンタ	bsd	pdspvlpr

ポート番号を指定しなければ、スーパーバイザはその番号を `printer-tcpip-port-number` 属性から取り出すことに注意してください。 `printer-address` の一部としてポート番号を指定した場合は、 `printer-tcpip-port-number` に値が指定されていても、そのポート番号が優先されます。

6.2.1.2 プリンタ接続方法の設定

プリンタ・ハードウェアは、いくつかの方法でスーパーバイザ・ホストに接続されます。デスクトップ・プリンタやミッドレンジ・プリンタには、通常、背面に1つ以上のコネクタがあり、それによってホスト、ターミナル・サーバ、ネットワークなどのデータ・ソースに接続できます。そのようなコネクタと、コネクタが接続しているハードウェアをプリンタ・インタコネクタといいます。

プリンタの中には、インタコネクタが、顧客が別個に購入するハードウェア・オプションであり、1つのプリンタ・モデルに対して常に同じとは限らないものもあります。また、別のプリンタでは、接続可能なものは既知であって、変わりません。物理インタコネクタの例としては、次のものが挙げられます。

- RS-232 シリアル
- セントロニクス・パラレル
- イーサネット

物理インタコネクトの中には、単方向や双方向などのバリエーションがあります。また、中には、TCP/IP などのリンク・レイヤ・プロトコルや `bsd/lpd` などのセッション・レイヤ・プロトコルをサポートするものもあります。

プリント・システムでの接続方法とは、サーバ・プロセスがプリンタ出力デバイスと通信できる一連の方法のことです `printer-connection-method` 属性は、プリンタで使用できる適切な接続方法を指定します。サポートされている接続方法には、次のものがあります。

- `serial` (シリアル)
- `parallel` (パラレル)
- `ip-socket` (IP ソケット)
- `digital-printserver` (DIGITAL PrintServer)
- `bsd`

1 つのスーパーバイザは、複数の接続方法を同時に使用することによって、複数の出力デバイスをサポートできます。

`printer-connection-method` の値が指定されていない場合は、`pdspvr` スーパーバイザは次の基準で値を選択します。

1. `printer-address` の値の形式が `/dev/lp<n>` の場合、接続方法は `parallel` です。
2. `printer-address` の値の形式が `/dev/tty<nn>` の場合、接続方法は `serial` です。
3. `printer-address` の値が `"/dev/デバイス・ファイル名"` ではない場合、接続方法は `ip-socket` です。

6.2.1.3 プリンタ接続レベルの設定

プリンタの中には、他のプリンタが提供するよりも多くの接続機能を使用するものがあります。たとえば、シリアル・バック・チャンネルに状態メッセージを提供するプリンタもあります。広範囲の接続タイプをサポートするために、プリント・システムは `printer-connection-level` 属性を使用します。次の表に示すように、この属性に設定可能な値は 0 から 5 までです。

値	説明
0	何も指定していない - システムの省略時の値を使用する。
1	出力専用データ (単方向)。単方向のみのプリンタは、用紙切れや紙詰まり、オフラインなどの状態をスーパーバイザ・プロセスに報告できない。スーパーバイザは、フロー制御を利用 (利用可能な場合) して、プリンタの電源が入っていなかったり、利用可能でなかったりしたときに、ドキュメント・データがあふれるのを防ぐ。
2	出力専用データ、状態ビットが返される。状態ビットの形式で状態を報告するプリンタは、プリンタがオフラインであるときや、エンジン・エラーが発生したときなどの状態情報をスーパーバイザに提供できる。
3	双方向 (同期のとれたセッション制御のないもの)。双方向データ接続を利用する PostScript プリンタは、ホストに対してさらに詳細で、広範囲のプリンタ情報メッセージを送信する。プリント・システム・スーパーバイザは、このようなプリンタ・メッセージの多くをイベントや障害テキストに変換する。これは、クライアントが表示できる。
4	双方向 (同期のとれたセッション制御のあるもの)。前述の値 3 での PostScript プリンタに関する説明を参照。
5	DIGITAL PrintServer。

プリント・システム・ソフトウェアで提供されるプリンタ属性ファイルには、サポートされるそれぞれのプリンタの適切な接続レベル定義が含まれています。通常環境では、プリンタ属性ファイル設定の `printer-connection-level` の値を変更する必要はありません。これは、通常、プリンタとそのインタコネクトによってレベルが決まるためです。ただし、ターミナル・サーバや他の特殊な目的のインタフェースを使用すると、プリント・システムが提供できるサポートのレベルに影響するので、プリンタに対する (性能は劣るが) 信頼性の高いサポートを得るために、接続レベルを低くしなければならないこともあります。

6.2.1.4 シリアル・ポート・パラメータ属性の設定

プリンタをシリアル・ポートに接続するには、いくつかの通信パラメータを指定する必要があります。 `printer-baud-rate` 属性を指定しない場合、プリント・システムは、ポートがあらかじめ正しく設定されていると想定します。ただし、 `printer-baud-rate` を指定した場合、スーパーバイザは、他の属性が当初何も指定されていない場合は、それらの属性を管理者が指定した値、または次の表に示す値に設定します。

属性	有効な値	省略時の値
printer-baud-rate	300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	物理ポート設定を使用する。
printer-data-bits	5, 6, 7, 8	8
printer-stop-bits	0, 1, 2	1
printer-input-flow-control	none, xoff, dtr	xoff
printer-output-flow-control	none, xoff, dtr	xoff
printer-parity	none, even, odd, mark, space	none

6.2.1.5 TCP/IP ポート番号の設定

TCP/IP ソケット接続へ接続するには、プリンタが通信に使用するポート番号を知っている必要があります。printer-tcpip-port-number 属性により、スーパーバイザは ip-socket 接続のプリンタに接続するときに、指定されたポート番号を使用します。

printer-address 属性の一部としてポート番号を指定した場合、そのポート番号はこの属性値を無効にします。プリンタ属性ファイルは、HP JetDirect プリンタなどのように、主流となっているネットワーク接続オプションをプリンタが備えているときにはいつでも printer-tcpip-port-number 属性を含みます。

属性	有効な値	省略時の値
printer-tcpip-port-number	1 - 65535	9100

6.2.2 物理プリンタの作成例

次に示すのは、物理プリンタ作成の例です。

- 次の例は、blue_sup で制御され、キュー production_q に関連付けられた物理プリンタ dec5100 の作成方法を示しています。

```
# pdcreate -c printer \  
-X Digital_DEClaser5100_Level2PS.paf \  
-x printer-address=61.140.16.20 \  
-x associated-queue=production_q \  
blue_sup:dec5100
```

- 物理プリンタ `highcap` を作成するとき、Lexmark 4039 10plus Level 2 PS プリンタ用に、スーパーバイザ `red_supr` で、プリンタ・アドレスが `/dev/pts/1`、関連付けられたキューが `mail_q`、`input-trays-supported` 属性と `input-trays-ready` 属性の追加値として `large-capacity` を持つように設定するには、次のように入力します。

```
# pdcreate -c printer \
-x LXX_4039plus_Level2PS.paf \
-x printer-address=/dev/pts/1 \
-x associated-queue=mail_q \
-x input-trays-supported+=large-capacity \
-x input-trays-ready+=large-capacity \
red_sup:lxk4039
```

Lexmark 4039 10plus Level 2 PS プリンタ用のプリンタ・データ・シートは、`large-capacity` をオプションの給紙トレイとして指定していることに注意してください。これはオプションなので、プリンタ属性ファイルは `input-trays-supported` 属性と `input-trays-ready` 属性のための値を含みません。このトレイをサイトで利用可能にするには、オプションのトレイ用の値を属性に追加する必要があります。この例では、作成操作中にこの値を設定できることを示しています。また、プリンタの作成後に、`pdset` コマンドで属性値を追加することもできます。

6.2.3 物理プリンタに対するオプション属性の設定

プリンタやプリント・システムの機能を十分に活用するために、多くの属性を設定することができます。これらの属性によって、両面印刷、セパレータ・ページ、「N アップ (片面への複数ページ印刷)」などの機能を使用することができます。

次の例は、いくつかの一般的な属性を設定する方法を示しています。

- ジョブとジョブの間に、プリンタがセパレータ・ページを印刷するようにするには、次のコマンドを使用して、必要な属性を物理プリンタに設定します。

```
# pdset -c printer \
-x 'job-sheets-supported=none job-copy-start job-copy-wrap' \
<pp_name>
# pdset -c printer
-x 'job-sheets-ready=none job-copy-start job-copy-wrap' \
<pp_name>
# pdset -c printer \
-x 'document-sheets-ready=none document-set-start-copy-separate' \
<pp_name>
```

- プリンタが両面印刷をサポートしており、PAF ファイルにこの属性が設定されていない場合は、次の属性を追加または設定する必要があります。

```
# pdset -c printer -x 'sides-supported=1 2' \
-x 'sides-ready=1 2' <pp_name>
# pdset -c printer -x 'sides-ready=1 2' \
-x 'sides-ready=1 2' <pp_name>
# pdset -c printer -x 'plexes-supported=simplex \
duplex tumble' <pp_name>
```

- プリンタが PostScript プリンタの場合に、横長フォーマットや縦長フォーマットでテキスト・ファイルを印刷できるようにするには、次のように設定します。

```
# pdset -c printer \
-x 'content-orientations-supported=landscape portrait' \
<pp_name>
```

6.2.4 物理プリンタを使用可能にする

物理プリンタを使用可能にすると、プリンタはそれに関連付けられたキューからのプリント・ジョブを受け付けることができます。

`pdenable` コマンドを使用して物理プリンタを使用可能にします。物理プリンタを使用可能にする場合は、次のことに注意する必要があります。

- *associated-queue* 属性で指定したキューが既に存在している必要があります。このキューをサポートするスプーラは、操作が正常に行われるように実行中でなければなりません。
- スーパーバイザ、スプーラ、キューの *physical-printers-ready* 属性にそのプリンタが追加されます。

`pdenable` コマンドの構文は次のとおりです。

```
pdenable [-c class_name] [-mmessage_txt] [-x extended_attribute_string] [-X attribute_filename] [server_name: ] [printer_name]
```

`pdenable` コマンドの省略時クラスが `printer` なので、プリンタを使用可能にするときに `-c` オプションを指定する必要はありません。

- プリンタ *dec5100* を使用可能にするには、次のコマンドを使用します。

```
# pdenable dec5100
```

- プリンタ *ljk4039* を使用可能にして、プリンタとプリンタの場所を説明するメッセージを含めるには、次のコマンドを使用します。

```
# pdenable -m "High-speed PostScript - Lab 2B" lxx4039
```

6.2.5 物理プリンタを使用不能にする

`pddisable` コマンドは物理プリンタを使用不能にするために使用します。物理プリンタを使用不能にすると、プリンタは新しいプリント・ジョブを受け付けなくなります。ただし、現在印刷中または処理中のジョブは最後まで実行されます。

次の属性は、物理プリンタを使用不能にすると更新されます。

- スーパバイザ、キュー、スプーラの *physical-printers-ready* 属性からそのプリンタの名前が消去されます。

`pddisable` コマンドの構文は次のとおりです。

```
pddisable [-c printer] [-m message_txt] [-x extended_attribute_string] [-X attribute_filename] [server_name: ] [printer_name]
```

次に示すのは、`pddisable` コマンドを使用して物理プリンタを使用不能にする例です。

- 物理プリンタ *lxx4039* を使用不能にするには、次のコマンドを使用します。

```
# pddisable lxx4039
```

6.2.6 物理プリンタの一時停止

`pdpause` コマンドは、物理プリンタの出力を一時停止するために使用できます。物理プリンタを一時停止すると、次の処理が行われます。

- スーパバイザは出力デバイスへのプリント・データの送信を停止します。
- 物理プリンタは可能な限りすぐに印刷を停止します。
- 物理プリンタに関連付けられたスプーラは、プリンタが一時停止 (paused) 状態の間は新しいジョブをスケジューリングしません。

一時停止されたときにプリンタが印刷をしていない場合にも、印刷をもう一度開始するには、再開操作 (`pdresume`) が必要です。

`pdpause` コマンドの構文は次のとおりです。

```
pdpause [-c class_name] [-m message_txt] [-x extended_attribute_string] [-X attribute_filename] [printer_name]
```

`pdpause` を使用して物理プリンタを一時停止させる場合には、オブジェクト・クラスの省略時の値がプリンタなので、これを指定する必要はありません。

- プリンタ `dec5100` を一時停止させるには、次のコマンドを使用します。

```
# pdpause dec5100
```

6.2.7 物理プリンタの再開

`pdresume` コマンドは、`pdpause` コマンドによって一時停止されているプリンタの処理を再開します。プリンタを再開するとき、プリンタは、現在割り当てられているプリント・ジョブがある場合には、そのジョブが一時停止された時点から印刷を再開します。

`pdresume` コマンドの構文は次のとおりです。

```
pdresume [-c class_name] [-m message_txt] [-x extended_attribute_string] [-X attribute_filename] [printer_name]
```

このコマンドの省略時のクラスはプリンタです。

- プリンタ `hpcolorPS` を再開するには、次のコマンドを実行します。

```
# pdresume hpcolorPS
```

6.2.8 物理プリンタ・オブジェクトの削除

`pddelete` コマンドは、物理プリンタ・オブジェクトを削除するために使用します。

物理プリンタ・オブジェクトを削除する前に、プリンタを使用不能にして、現在アクティブなジョブをすべて削除する必要があります。

次の属性は、物理プリンタを削除するときに更新されます。

- スーパーバイザ、スプーラ、およびキューの *physical-printers-supported* 属性が更新されます。
- 論理プリンタの *printers-ready* 属性と *printer-associated-printers* 属性が更新されます。
- 指定されたプリンタがキューに関連付けられた唯一の物理プリンタである場合、スプーラは、ジョブが別の物理プリンタに関連付けられるまで、そのキューでのジョブのスケジューリングを停止します。

pddelete コマンドの構文は次のとおりです。

```
pddelete [-c class_name] [-x extended_attribute_string] [-X  
attribute_filename] [printer_name]
```

- プリンタ *hpcolorPS* を削除するには、次のコマンドを使用します。

```
# pddelete hpcolorPS
```

6.3 論理プリンタの作成と管理

論理プリンタ・オブジェクトは、特定の特性と1つ以上の物理プリンタの機能を示す抽象化です。ユーザはジョブを印刷するときに、論理プリンタを指定することによって、ジョブやドキュメントの必要条件を満たす一連のデバイスにプリント・データを渡します。

次の例のように、pdcreate コマンドを使用して、論理プリンタを作成し、オペランドの一部としてスプーラ・サーバを指定します。

```
pdcreate [-c printer] [-x extended_attribute_string] [-X attribute_filename]  
[spooler_name:printer_name]
```

論理プリンタを作成するとき、関連付けられたスプーラは次の処理を行います。

- 論理プリンタをネーム・サービスに登録します。
- 新しい論理プリンタを使用不能として初期化します。新しい論理プリンタを使用可能にするには、使用可能操作を実行する必要があります。
- 新しい論理プリンタの名前を、スプーラの `logical-printers-supported` 属性に追加します。

プリント・システムは、論理プリンタを使用可能にする前に、論理プリンタに関連付けられたキューの名前を必要とします。オブジェクトを作成した後でも、対応する `associated-queue` 属性を設定できますが、この属性は作成操作で指定するようにしてください。

作成操作で `associated-queue` 属性を指定した場合、関連付けられたスプーラはスプーラの `logical-printers-supported` 属性を更新します。`associated-queue` を指定した場合は、指定されたキューが存在する必要があります。キューが存在しない場合、処理は失敗します。

また、initial-value-job オブジェクトと initial-value-document オブジェクトを指定して、printer-initial-value-job 属性と printer-initial-value-document 属性を持つ論理プリンタのジョブとドキュメントの省略時の値を設定することもできます。これらの属性は、作成操作で設定することも、オブジェクト作成後に設定操作で設定することもできます。

6.3.1 論理プリンタの作成例

次は、論理プリンタの作成例を示しています。

- 論理プリンタ doc2 を、スプーラ red_spl 上のキュー production_q に関連付けて作成するには、次のコマンドを使用します。

```
# pdcreate -c printer \  
-x "associated-queue=production_q" red_spl:doc2
```

6.3.2 オプションの論理プリンタ属性の設定

プリンタの機能を最大限に実現するために、設定が必要となる属性は数多くあります。これらの属性によって、プリンタの機能の使用を可能にし、両面印刷、セパレータ・ページ、「N アップ (片面への複数ページ印刷)」(テキスト・ファイルの方向) などの機能を有効にできます。

これらの属性は、論理プリンタが使用可能になったときに、物理プリンタの属性から継承されることに注意してください。論理プリンタを使用可能にする前に、論理プリンタ属性の値を物理プリンタ属性とは異なる値に設定した場合、物理プリンタの属性は継承されません。

6.3.3 論理プリンタに対する管理機能の実行

以降の各項では、プリント環境で論理プリンタに対して実行できる管理機能を説明します。機能には次のものがあります。

- 論理プリンタを使用可能にする
- 論理プリンタを使用不能にする
- プリンタ属性をリスト表示する
- 論理プリンタに対して実行されたジョブを表示する

6.3.4 論理プリンタを使用可能にする

`pdenable` コマンドは、論理プリンタを使用可能にするために使用されます。論理プリンタを使用可能にするときには、次の点に注意してください。

- `associated-queue` 属性に名前を指定されたキューが既に存在している必要があります。
- プリンタは、関連付けられたキューとスプーラの `logical-printers-ready` 属性に追加されます。
- プリンタは、関連付けられたすべての物理プリンタの `printers-ready` 属性に追加されます。
- プリンタは、論理プリンタにまだ定義されていない物理プリンタの `xxx-supported` 属性を継承します。

`pdenable` コマンドの構文は次のとおりです。

```
pdenable [-c printer] [-m message_txt] [-x extended_attribute_string] [-X attribute_filename] [printer_name]
```

次に示すのは、論理プリンタを使用可能にする例です。

- 省略時のサーバ上の `log_printer_1` を使用可能にするには、次のコマンドを使用します。

```
# pdenable log_printer_1
```

6.3.5 論理プリンタを使用不能にする

`pddisable` コマンドは論理プリンタを使用不能にするために使用されます。論理プリンタを使用不能にすると、新しいプリント・ジョブ要求は受け付けられず、そのプリンタの名前は関連付けられたキューとスプーラの `logical-printers-ready` 属性から削除されます。

`pddisable` コマンドの構文は次のとおりです。

```
pddisable [-c printer] [-m message_txt] [-x extended_attribute_string] [-X attribute_filename] [printer_name]
```

次に示すのは、`pddisable` コマンドを使用して論理プリンタを使用不能にする例です。

- オプションを使用せずに論理プリンタを使用不能にするには、次のコマンドを使用します。

```
# pddisable logical_printer_1
```

6.3.6 プリンタ属性のリスト表示

pdls コマンドを使用すると、論理プリンタの属性をリスト表示できます。
pdls コマンドの構文は次のとおりです。

```
pdls [-c class_name] [-ffilter_txt] [-F] [-g] [-r requested_attribute]  
[s- style_name] [x-extended_attribute_string] [X-attribute_filename ]  
[printer_name]
```

次の例は、pdls コマンドの使用方法と、このコマンドで生成される出力を示しています。

- 論理プリンタ *lpx0001* 用のすべての属性を、1 行に 1 属性で表示するには、次のコマンドを実行します。

```
# pdls -c printer -r all -s line lpx0001  
lpx0001: object-class = printer  
lpx0001: printer-name = lpx0001  
lpx0001: availability = normal  
lpx0001: printer-realization = logical  
lpx0001: printer-state = idle  
lpx0001: enabled = yes  
lpx0001: associated-queue = px0001  
lpx0001: associated-server = amanda  
lpx0001: printer-associated-printers = px3  
pr1  
lpx0001: printers-ready = px3  
pr1
```

- 論理プリンタ *lpx0001* の *associated-queue* 属性だけを表示するには、次のコマンドを実行します。

```
# pdls -c printer -r associated-queue lpx0001  
associated-queue  
-----  
px0001
```

6.3.7 論理プリンタに対して実行されたジョブの表示

pdq コマンドを使用すると、論理プリンタに対して実行されたジョブのリストを作成することができます。このコマンドは、プリンタに関連付けられたキューの中に現在ある、1 つまたはすべてのジョブを表示するために使用できます。ジョブは、印刷のためにスケジューリングされた順序で表示されません。次に示すのは、pdq コマンドの使用例です。

- lpx0001 にキューイングされたすべてのジョブのリストを表示するには、次のコマンドを実行します。

```
# pdq -p lpx0001 -r all -s line
amanda:12: object-class = job
amanda:12: job-identifier = amanda:12
amanda:12: assigned-queue = px0001
amanda:12: printer-name-requested = lpx0001
amanda:12: job-originating-host = bulldog
amanda:12: job-owner = grace
amanda:12: print-complete = yes
amanda:12: user-name = grace
amanda:12: number-of-documents = 4
amanda:12: job-submission-complete = yes
amanda:12: submission-time = 26:06:97:11:10:42
amanda:12: total-job-octets = 464786
amanda:12: job-hold = no
amanda:12: results-profile = {job-copies=1}
amanda:12: job-name = cheatsheetCH2.ps
amanda:12: printers-assigned = px3
amanda:12: current-job-state = printing
amanda:12: started-printing-time = 26:06:97:11:10:44
amanda:12: previous-job-state = processing
amanda:12: intervening-jobs = 0
amanda:12: job-copies = 1
```

ここで提供される情報は1つのジョブのものです。複数のジョブがこのプリンタにキューイングされている場合、これと同じ情報がそれぞれのジョブについて提供されます。

- lpx0001 に対して実行されたジョブの簡潔な属性リストを表示するには、次のコマンドを実行します。

```
# pdq -p lpx0001 -r brief -s line
amanda:16: job-identifier = amanda:16
amanda:16: job-name = voicemail.txt
amanda:16: current-job-state = processing
amanda:16: intervening-jobs = 0
amanda:16: printer-name-requested = lpx0001
amanda:16: printers-assigned = pr1
amanda:19: job-identifier = amanda:19
amanda:19: job-name = net.Form.ps
amanda:19: current-job-state = printing
amanda:19: intervening-jobs = 0
amanda:19: printer-name-requested = lpx0001
amanda:19: printers-assigned = px3
```

6.3.8 論理プリンタの削除

`pddelete` コマンドは論理プリンタを削除するために使用します。論理プリンタを削除する際は、次のことを考慮してください。

- プリンタを使用不能にして、アクティブなジョブを置かないようにする必要があります。
- スーパーバイザ、スプーラ、キューの *physical-printers-supported* 属性が更新されます。
- 論理プリンタの *printers-ready* 属性と *printer-associated-printers* 属性が更新されます。
- 指定されたプリンタがキューに関連付けられている唯一の物理プリンタである場合、スプーラは、そのキューが別の物理プリンタに関連付けられるまで、そのキューでのジョブのスケジューリングを停止します。

`pddelete` コマンドの構文は次のとおりです。

```
pddelete [-c printer] [-m message_txf] [-x extended_attribute_string] [-X attribute_filename] [printer_name]
```

次のコマンドは、論理プリンタの削除方法を示しています。

- プリンタ *printer_1* を削除するには、次のコマンドを使用します。

```
# pddelete printer_1
```

6.4 省略時のオブジェクトの作成

印刷の省略時の値は、相当する属性がプリント要求の中に存在しない場合に、プリント・システムがジョブやドキュメントに自動的に適用する属性値です。プリント・システムはプリント・オブジェクトの2つのクラスを使用して、省略時の属性値のセットを特定します。これらのオブジェクト・クラスは、次のとおりです。

- `initial-value-job` — 省略時の値を持つジョブ属性を含む
- `initial-value-document` — 省略時の値を持つドキュメント属性を含む

省略時の値は、ジョブの本来の出力先である物理プリンタの機能を反映しています。このため、最初に物理プリンタを作成し、希望するプリンタ属性を

設定してから、`initial-value-job` オブジェクトと `initial-value-document` オブジェクトを作成してください。

`initial-value-job` オブジェクトと `initial-value-document` オブジェクトはスプーラ・オブジェクトなので、関連付けられたスプーラは、初期値オブジェクトを作成する前に存在している必要があります。

6.4.1 ジョブとドキュメントの省略時の値の指定

物理プリンタの機能を考慮して、1つの物理プリンタにさまざまな省略時の値を指定するために、複数の `initial-value-job` オブジェクトと `initial-value-document` オブジェクトを作成することができます。`initial-value-job` オブジェクトと `initial-value-document` オブジェクトを適用する方法によって、物理プリンタが出力に使用する省略時の値が決定されます。

`initial-value-job` オブジェクトまたは `initial-value-document` オブジェクトを適用するには、次のいずれか一方、または両方を行う必要があります。

- 論理プリンタに対して `printer-initial-value-job` 属性と `printer-initial-value-document` 属性を設定することで、論理プリンタにオブジェクト設定を適用します。
- プリント要求に `initial-value-job` 属性と `initial-value-document` 属性を指定することで、オブジェクト設定を適用します。

`initial-value-job` オブジェクトと `initial-value-document` オブジェクトを論理プリンタとジョブの両方に適用すると、ジョブに指定されたオブジェクトが、論理プリンタに指定されたオブジェクトを無効にします。

管理者は、`initial-value-job` オブジェクトと `initial-value-document` オブジェクトをどのように適用するかを決定してください。

- これらのオブジェクトを論理プリンタに適用するのであれば、適切な論理プリンタ属性 (`printer-initial-value-job` および `printer-initial-value-document`) を設定することによって適用します。
- これらのオブジェクトをジョブの実行時に適用するのであれば、適切なジョブ属性 (`initial-value-job` と `initial-value-document`) を設定することによって適用します。`initial-value-job` オブジェクトや

`initial-value-document` オブジェクトが利用できることをユーザに知らせるのは管理者の責任です。

次の例を参考にすると、サイトで適用可能な `initial-value-job` オブジェクトと `initial-value-document` オブジェクトを判断できるでしょう。

`initial-value-job` オブジェクト `JobDefaults1` には、ジョブ属性 `job-sheets=job-copy-start` が含まれていて、プリント・システムが `JobDefaults1` を適用するときに、印刷された出力にジョブの開始シートを含めるようにします。これはよく使用されるので、`JobDefaults1` を対応する論理プリンタに関連付けます。

```
# pdset -c printer \  
-x 'printer-initial-value-job=JobDefault1' \  
[spooler_name:]logical_printer_name
```

ただし、開始シートがいない場合もあるので、`initial-value-job` オブジェクト `JobDefaults2` を属性 `job-sheets=none` で作成します。これは特殊な場合だと考えられるので、`JobDefaults2` は論理プリンタには関連付けません。代わりに、`JobDefaults2` は次のようにジョブの実行時に適用できます。

```
# pdpr -x 'initial-value-job=JobDefaults2' file_name
```

`JobDefaults2` はジョブ実行プリント要求の一部なので、これは論理プリンタに関連付けられた `JobDefaults1` を無効にします。

この例は、1つの属性 (`job-sheets`) だけを説明していますが、`initial-value-job` オブジェクトの `JobDefaults1` と `JobDefaults2` は、属性が複数になると、異なる可能性があります。

6.4.2 省略時のオブジェクトの作成

サイトに必要なジョブとドキュメントの省略時の値を決定したので、`initial-value-job` オブジェクトと `initial-value-document` オブジェクトを作成する準備ができました。

`pdcreate` コマンドを使用すると、`initial-value-job` オブジェクトと `initial-value-document` オブジェクトを作成できます。これらのコマンドのフォーマットは次のとおりです。

```
pdcreate [-c initial-value-job] [[spooler_name]:iv-job-name]
```

```
pdcreate [-c initial-value-document] [[spooler_name]:iv-doc-name]
```

```
# pdcreate -c initial-value-job [spooler_name:]iv_job_name
# pdcreate -c initial-value-document
[spooler_name:]iv_document_name
```

省略時のサーバにオブジェクトが作成されない場合、スプーラ名を指定する必要があります。

`initial-value-job` オブジェクトと `initial-value-document` オブジェクトに含めたいジョブ属性とドキュメント属性が既にわかっている場合は、`pdcreate` コマンドに `-x 'attribute_name=attribute_value'` を指定することができます。

`pdprintadmin` GUI アプリケーションは、論理プリンタの省略時の設定へのアクセスを提供します。論理プリンタに `initial-value-job` オブジェクトと `initial-value-document` オブジェクトをまだ適用していない場合は、省略時の設定によって `initial-value-job` オブジェクトと `initial-value-document` オブジェクトを作成できます。

`initial-value-job` オブジェクトと `initial-value-document` オブジェクトには、次に示すように 2 セットの属性があります。

- オブジェクト自体に適用する属性には、次のようなものがあります。

```
initial-value-job-identifier
initial-value-document-identifier
associated-server
descriptor
message
object-class
```

- 対応するジョブやドキュメントの省略時の値を示す属性には、たとえば `initial-value-job` オブジェクトの場合は、次のものがあります。

```
job-copies
job-sheets
results-profile
```

- `initial-value-document` オブジェクトについては、次のものがあります。

```
copy-count
default-font
document-format
```

pdset コマンドを使用して、initial-value-job オブジェクトと initial-value-document オブジェクトの属性を設定します。

```
# pdset -c initial-value-job \  
-x attribute_name=attribute_value iv_job_name  
  
# pdset -c initial-value-document \  
-x attribute_name=attribute_value iv_document_name
```

ジョブとドキュメントの管理

ジョブとドキュメントをプリント・システムで適切に管理できるように、それらのさまざまな側面を考慮する必要があります。一般に、次の3種類の属性が、ジョブとドキュメントの管理の異なる側面を反映しています。

- 印刷された出力に現れるデータに関係なく、ジョブやドキュメントのオブジェクト自体に影響を与える属性。このような処理には、プリント・デバイスに使用される給紙トレイと排出トレイの設定があります。
- スプーラとスーパーバイザがプリント・データの処理方法を決定する属性。このような属性には翻訳フィルタがあります。
- 実際に印刷された出力の体裁と内容を決定する属性。このような属性には、それぞれの用紙にページがどのようにレイアウトされるかに影響を与えたり、セパレータ・ページがドキュメント間に出力されるかどうかを決定したりするものがあります。

7.1 ジョブとドキュメントのオブジェクト処理の実行

ユーザが通常実行するジョブとドキュメントの処理には、次のものが含まれます。

- ジョブの印刷
- 指定した時刻より後でのジョブの印刷
- ジョブとドキュメントの属性のリスト
- ジョブ・キューのリスト
- ジョブとドキュメントの属性の変更
- ジョブの消去、削除、取消し
- ジョブの再実行
- ジョブの昇格 (キューの先頭への移動)
- ジョブの一時停止と再開
- ジョブの保留
- ジョブの保持
- ジョブの破棄
- メッセージとイベント通知の設定

ジョブとドキュメントの違いを理解することは非常に重要です。1つのジョブには複数のドキュメントを含めることができ、それぞれのドキュメントに対して多くの異なる処理指示を含めることができます。

7.1.1 ジョブの印刷

印刷処理は、1つ以上のファイルを印刷する要求から、ジョブ・オブジェクトとドキュメント・オブジェクトを作成します。スプーラはそれぞれのプリント要求に対してジョブ・オブジェクトを作成するとともに、単一のプリント要求の一部として、ユーザが指定する各ファイルに対して1つのドキュメント・オブジェクトを作成します。このため、ジョブ・オブジェクトには、1つ以上のドキュメント・オブジェクトが含まれます。

それぞれのジョブに対して、スプーラは一意的ジョブ識別子 (`job-identifier`) を割り当て、これは、ジョブの一時停止などのジョブ処理が実行される時にジョブを特定するために使用されます。

プリント要求は `pdpr` コマンドを使用して作成されます。 `pdpr` コマンドの構文は次のとおりです。

```
pdpr [-f file_name] [-n copies] [-N notification_method]  
[-p logical_printer_name] [-t job_name] [files]
```

次に示すのは、`pdpr` コマンドを使用してプリント要求を実行する例です。

- 省略時の論理プリンタ (PDPRINTER) でファイル `mail-file.txt` を印刷するには、次のコマンドを使用します。

```
# pdpr mail-file.txt
```
- プリンタ `pr_doc1` でファイル `mail-file.txt` を印刷するには、次のコマンドを使用します。

```
# pdpr -p pr_doc1 mail-file.txt
```
- ファイル `'front-page.ps'` を片面で、ファイル `'ch1.ps'` と `'ch2.ps'` を両面で印刷するには、次のコマンドを使用します。

```
# pdpr -x sides=1 -f front-page.ps -x sides=2 ch1.ps ch2.ps
```

7.1.2 指定した時刻より後のジョブの印刷

プリント要求を実行し、そのジョブを指定した日付と時刻が来るまでは印刷しないように指定することができます。また、設定処理あるいは変更処理によって、ある時刻の後に印刷を指定することもできます。

印刷を大量のジョブの後の時刻に指定して、印刷量の少ない間に印刷できるようにすることもあります。

時刻と日付を指定するときには、次の構文を使用します。

```
dd:mm:yyyy:hh:mm:ss
```

ジョブの印刷時刻を設定すると、スプーラは次の処理を行います。

- 属性 `current-job-state` を保留 (held) に設定します。
- 属性 `job-state-reasons` に値 `job-print-after-specified` を追加します。

指定した印刷時刻が過ぎると、スプーラはジョブをスケジューリングし、`current-job-state` を処理待ち (pending) に変更します。

次の例は、ジョブに対して印刷時刻を指定する方法を示しています。

- ジョブを実行するときにジョブの印刷時刻を指定するには、次のコマンドを使用します。

```
# pdpr -x job-print-after=31:12:1998:12:59\  
book1.ps
```

- ジョブを実行し、その後で、1998年1月25日午前7:00以降に印刷するようにジョブを変更するには、次のコマンドを使用します。

```
# pdmod -x 'job-print-after=25:01:1998:07:00' \  
red_spl:1258
```

7.1.3 ジョブの再実行

同一のスプーラ上の異なるプリンタに対して、実行済みのプリント・ジョブを再実行することができます。

ジョブが次のいずれかの状態の場合にのみ、再実行することができます。

- pending (処理待ち)
- held (保留)
- paused (一時停止)
- retained (保持)
- terminating (後処理中) (この状態が `job-state-reasons` 属性の `documents-needed` を含まない場合)

ジョブの状態が `completed` (完了), `processing` (処理中), `printing` (印刷中) のいずれかの場合は, ジョブを再実行することはできません。

ジョブを再実行するには, `pdresubmit` コマンドを使用します。このコマンドの構文は次のとおりです。

```
pdresubmit [-c job] [logical_printer_name=name] [object_instance]
```

`object-instance` は, ジョブ識別子またはキュー名です。個々のジョブを再実行することも, キュー内のすべてのジョブを別の論理プリンタに対して再実行することもできます。

次に示すのは, `pdresubmit` コマンドを使用してジョブを再実行する例です。

- ジョブ 2000 をプリンタ `lpx0001` に対して再実行するには, 次のコマンドを使用します。

```
# pdresubmit -c job lpx0001 2000
```

- 指定したキュー内のすべてのジョブを再実行するには, まず, キューを使用不能にしてから, 再実行操作を行う必要があります。キュー `qpx0001` からすべてのジョブを論理プリンタ `lpx0001` に対して再実行するには, 次のコマンドを使用します。

```
# pddisable -c queue qpx0001
# pdresubmit -c queue lpx0001 qpx0001
```

この操作を実行すると, スプーラが次の処理を行うことに注意してください。

- 処理待ち (`pending`), 一時停止 (`paused`), 保留 (`held`) 状態のすべてのジョブを, 指定されたキューから消去し, それらを指定したターゲット論理プリンタに対して再実行します。
- 正常に再実行できないジョブに対しては警告を返します。これらのジョブは手動で再実行する必要があります。

この例では, 両方のプリンタが同一のスプーラに関連付けられている必要があります。

`pdresubmit` コマンドは非同期であることに注意してください。このコマンドは, サーバが処理を完了する前にプロンプトを返します。

7-4 ジョブとドキュメントの管理

7.1.4 ジョブの昇格

ジョブの昇格とは、ジョブをキューの先頭に移動することです。ジョブの昇格によって、そのジョブは、実行時刻に関係なく、昇格されていないジョブの前に印刷されます。1回に昇格できるのは1つのジョブですが、同一のキューで複数のジョブを昇格すると最後に昇格されたジョブが、最初に印刷されます。

現在印刷中のジョブは、昇格されたジョブを含むキューに関連付けられたそれぞれの物理プリンタで、正常に印刷を続行します。スプーラは、現在のジョブの印刷を完了し、別のジョブを扱うことのできる最初の物理プリンタに、最後に昇格されたジョブを割り当てます。

current-job-state が pending (処理待ち) または held (保留) のジョブは昇格することができます。キューの状態は、ready (処理可能) または paused (一時停止) のいずれかである必要があります。

管理者とオペレータは pdpromote コマンドを使用してジョブを昇格できます。エンド・ユーザはジョブを昇格できません。pdpromote コマンドのフォーマットは次のとおりです。

```
pdpromote [-m message_text] [server_name:job_id]
```

次に示すのは pdpromote コマンドの使用例です。

- スプーラ sx0001_spl でジョブ 2249 を昇格するには、次のコマンドを使用します。

```
# pdpromote sx0001_spl:2249
```

7.1.5 ジョブの一時停止

ジョブを一時停止すると、ジョブは印刷のために物理プリンタに渡されません。キュー内の他のジョブは一時停止されたジョブを残して渡されます。

処理待ち (pending) または保留 (held) 状態のジョブだけが一時停止できます。つまり、印刷が始まったジョブは一時停止できません。ジョブを一時停止するときには、次の点に注意してください。

- 1つのジョブ内で特定のドキュメントを一時停止することはできません。一時停止するのはジョブ全体です。
- ジョブは再開されるまで再スケジューリングすることはできません。

- ジョブが印刷のためにスーパーバイザに既に渡されていた場合、操作は失敗します。

pdpause コマンドを使用してジョブを一時停止します。このコマンドのフォーマットは次のとおりです。

```
pdpause [-c job] [-m message_text] [ server_name:job_id]
```

次に示すのは、pdpause コマンドを使用してジョブを一時停止する例です。

- 次のコマンドを使用して spooler1 でジョブ 11224 を一時停止し、ジョブが一時停止された理由を示すメッセージを含めます。

```
# pdpause -c job -m "Job will be printed later"\  
spooler1:11224
```

7.1.6 ジョブの再開

一時停止されたジョブは pdresume コマンドを使用して再開できます。一時停止されたジョブを再開すると、そのジョブのスケジューリングや印刷ができるようになります。

pdresume コマンドのフォーマットは次のとおりです。

```
pdresume [-c class_name] [-m message_text] [ job_id]
```

次に示すのは、pdresume コマンドを使用してジョブを再開する例です。

- 次のコマンドを使用すると、ジョブ 11224 が再開されます。

```
# pdresume -c job spooler1:11224
```

7.1.7 ジョブの保留

スプーラが印刷のためにジョブをスケジューリングしないように、ジョブを保留にすることができます。job-hold 属性を true に設定することで、ジョブを保留にできます。

ジョブを保留にすると、スプーラは、次の処理を行います。

- 属性 current-job-state を held に設定します。
- 属性 job-state-reasons に値 job-hold-set を追加します。

ジョブの保留は、次のイベントのいずれか 1 つが発生しない限り、ジョブが無期限に保留状態になることを除けば、ジョブの一時停止と似ています。

- `job-hold` 属性を `false` に設定したとき。スプーラはジョブをスケジューリングして、その `current-job-state` を `pending` (処理待ち) に設定できます。
- 以前に設定したジョブ破棄時刻が過ぎたとき。スプーラは保留にされたジョブを削除します。
- 関連付けられたキューまたはスプーラをクリーン処理したとき。スプーラは、`held` 状態のジョブを含めて、そのキューまたはスプーラにあるすべてのジョブを削除します。

次に示すのは、ジョブを保留状態にする例です。

- ジョブの所有者は、`pdmod` コマンドを使用して、スプーラ `sx0001_spl` にあるジョブ番号 `2002` を保留状態にすることができます。

```
$ pdmod -x 'job-hold=yes' sx0001_spl:2002
```

- 管理者またはオペレータは、`pdset` コマンドを使用して、前の例と同一のジョブを保留にできます。

```
# pdset -x 'job-hold=yes' sx0001_spl:2002
```

- ジョブの保留を解除するには、次の例のように `pdset` コマンドまたは `pdmod` コマンドを使用します。

```
# pdset -x 'job-hold=no' sx0001_spl:2002 # pdmod -x 'job-hold=no' sx0001_spl:2002
```

7.1.8 ジョブの保持

ジョブの印刷が完了した後に、指定した期間だけ、スプーラにジョブを保持させることができます。保持されたジョブは、再実行またはチェックのために利用することができます。

ジョブを保持すると、スプーラは次の処理を行います。

- ジョブが完了した後でも、ジョブ・オブジェクト、その属性、そのドキュメントを保持します。
- 属性 `current-job-state` を `retained` に設定します。

ジョブは、次のいずれかのイベントが起こるまで、保持 (`retained`) 状態のままになります。

- 保持期間が満了したとき。このとき、スプーラは保持されたジョブを削除します。

- 関連付けられたキューまたはスプーラをクリーン処理したとき。このとき、スプーラは、保持 (retained) 状態にあるジョブを含めて、キューまたはスプーラ内にあるすべてのジョブを削除します。

pdmod, pdset, pdrm コマンドのいずれかを使用して、ジョブを保持することができます。エンド・ユーザは pdmod コマンドと pdrm コマンドを使用して、所有するジョブに保持期間を設定できます。管理者とオペレータは、pdset コマンドを使用して任意のジョブの保持期間を設定できます。

次に示すのは、これらのコマンドを使用してジョブを保持する例です。

- スプーラ sx0001_spl にジョブ番号 3021 を 24 時間保持するには、次のいずれかのコマンドを実行します。

```
# pdrm -r '24:00' sx0001_spl:3021
# pdmod -x "job-retention-period=24:00" sx0001_spl:3021
# pdset -c job -x "job-retention-period=24:00"
sx0001_spl:3021
```

pdrm コマンドを使用すると、ジョブが再実行のために保持されていても、ジョブが取り消されて、キューから消去されることに注意してください。

7.1.9 ジョブの破棄

ジョブが特定の時刻までに印刷されない場合、スプーラがそのジョブを削除するように指定できます。属性 job-discard-time で時刻を指定します。

ジョブ破棄時刻になったときに、スプーラはジョブが印刷されたかどうかや、ジョブの状態に関わらず、ジョブを削除します。

ジョブ破棄時間を指定するために、エンド・ユーザは pdmod コマンドを使用することができ、また、管理者とオペレータは pdset コマンドを使用することができます。次に、ジョブ破棄時刻を設定する例を示します。

- 1998 年 1 月 2 日午後 5:00 にジョブ 3021 を破棄するには、次のコマンドのいずれかを入力します。

```
# pdmod -x "job-retention-period=02:01:1998:17:00:00" 3021
# pdset -c job \
-x "job-retention-period=02:01:1998:17:00:00" 3021
```

ドキュメント・データのフィルタリング

多くの場合、ドキュメント・データは印刷前に変更する必要があります。たとえば、シンプル・テキスト・ドキュメントは、PostScript に翻訳してから PostScript プリンタで印刷する必要があります。あるいは、EBCDIC 文字セットを使用するドキュメントは、ASCII 文字セットに変換してから一般的なデスクトップ・プリンタで印刷する必要があります。

ドキュメント・データの変更に対する必要性は、顧客や国によって異なるので、ドキュメントをプリンタに送信する前にドキュメント・データの変更を行うユーザ作成プログラム、またはプラットフォームで提供されるプログラムのためのメカニズムが、スーパーバイザに含まれています。これらのプログラムは、ジョブの一部として印刷されるそれぞれのドキュメントに対するフィルタ、またはデータ・タイプ翻訳プログラムと呼ばれます。

フィルタは子プロセスとして、スーパーバイザによって実行されます。ドキュメント・データはフィルタにパイプされ、スーパーバイザはデータをプリンタに送信する前にデータを読み戻します。スーパーバイザは通信とプリンタのコントロールを制御します。

この章の情報は、スーパーバイザ `pdspvr` によってサポートされる物理プリンタだけに適用されます。アウトバウンド・ゲートウェイのスーパーバイザはデータ・フィルタリングを実行しません。アウトバウンド・ゲートウェイ・スーパーバイザはリモート・ホストまたはプリンタに、フィルタリング・タスクの実行を依頼します。

8.1 フィルタのタイプ

プリント・システムは、翻訳フィルタと変更フィルタをサポートします。翻訳フィルタは次の処理を実行します。

- ドキュメントのドキュメント・フォーマット (PDL) をプリンタによって直接サポートされているドキュメント・フォーマット (プリンタの `native-document-formats-ready` 属性に含まれるドキュメント・フォーマット) に翻訳します。

- スーパーバイザが呼び出すと、必要なドキュメント・フォーマット翻訳を実行します。

変更フィルタは次の機能を実行します。

- 文字セット翻訳またはシンプルなテキスト・フォーマットなど、ドキュメント・フォーマットを変更しないドキュメント・データ変更を実行します。
- 特定のドキュメントに対して、ユーザによって呼び出されたときだけ、データ変更を実行します。

これら2つのタイプについてフィルタ・プログラムの書き方に違いはなく、また、スーパーバイザはそれらが適切に使用されているかどうかを検証しません。変更フィルタリングと翻訳フィルタリングは、両方ともドキュメントに適用できます。これが起こると、変更フィルタが元のドキュメント・データを受け取り、変更フィルタの出力が翻訳フィルタにパイプされ、翻訳フィルタからの出力がスーパーバイザによってプリンタに送信されます。

スーパーバイザはフィルタの動作を完全には制御できません。たとえば、フィルタはファイルへの書き込みやデバイスへの直接の書き込みを行うべきではありませんが、スーパーバイザがこれを防止することはできません。

8.1.1 フィルタに関連する属性

次の属性はフィルタについての情報を提供します。

- `filter-definition`
- `excluded-filter`
- `modification-filter`
- `translation-filter`
- `no-filtering`

filter-definition

サーバ属性 `filter-definition` は、フィルタのプログラムを定義し、そのプログラムの呼び出しに必要な情報を含みます。

`filter-definition` は、次の構文を持つ複雑な属性です。

```
filter-definition={name type input-format output-format  
command}
```

8-2 ドキュメント・データのフィルタリング

さらに、`filter-definition` 属性は複数の値を持ちます。名前を指定したフィルタをいくつでも定義することができます。

次の表で、属性値のそれぞれのコンポーネント・フィールドを説明します。

フィールド	値	説明
<code>name</code>	テキスト	フィルタの名前はサーバ内で一意である必要がある。プリント・システムは、新しいフィルタ定義の検索キーとして名前を使用する。
<code>type</code>	<code>translation</code> または <code>modification</code>	フィルタのタイプ。フィルタの呼び出しに使用するメカニズムを定義する。省略時の値は <code>translation</code> である。
<code>input-format</code>	ドキュメント・フォーマット	フィルタが入力としてサポートするドキュメント・フォーマット。省略した場合、フィルタは入力に任意のフォーマットを取ることができる。翻訳フィルタ呼び出しだけに使用される。
<code>output-format</code>	ドキュメント・フォーマット	フィルタが出力時に生成するドキュメント・フォーマット。
<code>command</code>	テキスト	サーバがフィルタを呼び出すために実行するコマンド。

フィルタを使用するためには、それをスーパーバイザで定義する必要があります。管理者は `pdset` コマンドで `filter-definition` 属性を設定することでフィルタを定義します。たとえば、次の例では、スーパーバイザに認識されているフィルタのリストに、PostScript から `simple-text` への翻訳フィルタを追加します。

```
# pdset -c server -x "filter-definition+={name=my-text-to-ps \  
type=translation \  
input-format=simple-text \  
output-format=PostScript \  
command='/usr/bin/ttp'}" \  
blue_sup
```

一度 `filter-definition` 属性に値が入ると、`+=` 構文を使用してさらにフィルタを追加できます。1 つのフィルタだけを削除して他のものを残すには、`-=` 構文を使用して 5 つすべてのフィールドを正確に指定します。すべてのフィルタ定義を削除するには、次のように `==` 構文を使用します。

```
# pdset -c server -x "filter-definition== blue_sup
```

セキュリティに関する重要な注意

コマンド実行可能ファイルとしては、`root` アカウントだけが置換や変更を行えるものを常に指定してください。非特権ユーザのディレクトリに存在するフィルタ・プログラムを指定すると、深刻なセキュリティ・リスクが生ずることになります。

excluded-filters

プリンタ属性 `excluded-filters` を使用すると、特定のプリンタに対して一定の翻訳フィルタの使用を不許可にします。`excluded-filters` の値はフィルタ名のリストです。スーパーバイザは、プリンタに渡されるドキュメントに対して翻訳フィルタを選択するとき、この属性にリストされているすべてのフィルタを除外します。スーパーバイザは `excluded-filters` リストにある名前が実際に定義済みのフィルタであるかどうかは確認しません。スーパーバイザは、フィルタが `filter-definition` リストから削除されても、`excluded-filters` 属性を更新しません。

modification-filter

ユーザはドキュメント属性 `modification-filter` を指定して、ジョブ内のドキュメントに変更フィルタを適用します。この属性の値は、他の翻訳フィルタリングよりも前にドキュメント・データに適用されるフィルタの名前です。プリント・システムは、指定されたフィルタが、ドキュメントが実行されたときにスーパーバイザに認識されているものかどうかは検証しません。

translation-filter

ユーザは、ドキュメント属性 `translation-filter` を指定して、指定された翻訳を実行できる複数のフィルタが利用可能なときに、1つの翻訳フィルタの自動起動を無効にすることができます。この属性が指定された場合、この属性の値は、`document-format` 属性や `native-document-formats-ready` 属性の値にかかわらず、ドキュメント・データに適用されるフィルタの名前になります。

no-filtering

ユーザは、ブール・ドキュメント属性 `no-filtering` を指定して、翻訳フィルタリングと変換フィルタリングの両方を使用不能にできます。`no-filtering`

が true の場合、サーバは翻訳フィルタを起動せず、modification-filter の値を無視します。

8.1.2 コマンド・テキスト処理

フィルタ定義のコマンド要素には、スーパーバイザがフィルタを呼び出すために実行するコマンドが含まれています。このコマンド要素は、属性値を指定する変数を含んでいることがあります。スーパーバイザはこの変数を対応する属性値に置き換えます。置換フィールドの構文は次のとおりです。
`${ attribute-name[,default-value[, substitution-expression]]}`

角カッコ内の項目はオプションです。default-value フィールドと substitution-expression フィールドは空文字列にできます。属性名は表 8-1 にリストされている任意のドキュメント属性にできます。スーパーバイザは次のいずれか 1 つで、置換フィールドを置き換えます。

- 指定された属性に定義済みの値がある場合、および置換フィールドが存在しない(コンマが 1 つ以下) 場合は、指定された属性の値。
- 属性が定義されておらず、default-value フィールドが存在する(属性の後に少なくとも 1 つのコンマがある) 場合は、default-value フィールドの文字。
- 評価された substitution-expression が存在し、属性が定義されている場合は、その評価された substitution-expression。

属性名と値は短縮することができません。属性に値がなく、default-value が指定された場合、スーパーバイザは default-value で置換フィールドを置き換えます。属性に値がなく、default-value が指定されていない場合、スーパーバイザは `${...}` を空文字列で置き換えます。

例：

number-up が値 "2" の場合、`"-N${number-up,0}"` は "-N2" と評価されます。

number-up が定義されていない場合、`"-N${number-up,0}"` は "-N0" と評価されます。

number-pages が定義されている場合、`"${number-pages,, -P}"` は "-P" と評価されますが、number-pages が定義されていない場合は空文字列 ("") と評価されます。

ネストした評価

場合によっては、置換文字列に 1 つ以上の属性の値を含めることが必要になります。これは、置換フィールド内に属性置換引数を含めることによつて行われます。

例：

" $\{\text{number-up}, , -N\{\text{number-up}\}$ " は、`number-up` が指されていない場合には空文字列に、また、`number-up` が値 "2" の場合には `-N2` と評価されます。

`top-margin=4`、`left-margin=0`、`right-margin=0`、`bottom-margin=4` のときに、" $\{\text{top-margin}, , -M\{\{\text{top-margin}\}, \{\text{left-margin}\}, \{\text{right-margin}\}, \{\text{bottom-margin}\}\}$ " は `-M4,0,0,4` と評価されます。

次の表は、コマンド置換フィールドで使用できる属性を示しています。プリント・システムは `simple-text` ドキュメントで主に使用される属性のいくつかをサポートします。これらの属性には、`bottom-margin`、`footer-text`、`header-text`、`left-margin`、`length`、`number-pages`、`repeated-tab-stops`、`right-margin`、`top-margin`、`width`、`content-orientation` があります。これらの属性の多くは、既存のフィルタに入力を提供するだけのためにサポートされています。

表 8-1: コマンド置換で使用されるドキュメント属性

属性名	構文	構文と説明
<code>bottom-margin</code>	整数	ページの下端とテキスト領域の下端との間隔 (文字数単位)。
<code>footer-text</code>	テキスト	各ページのフッタ行。
<code>header-text</code>	テキスト	各ページのヘッダ行。
<code>left-margin</code>	整数	論理ページの左端とテキスト領域の左端との間隔 (文字数単位)。
<code>document-length</code>	整数	テキスト領域の長さ (文字数単位)。
<code>number-pages</code>	ブール	ページに番号を付けるかどうかを示す。
<code>repeated-tab-stops</code>	整数	タブ・ストップ間の文字数。
<code>right-margin</code>	整数	ページの右端とテキスト領域の右端との間隔 (文字数単位)。

8-6 ドキュメント・データのフィルタリング

表 8-1: コマンド置換で 사용되는ドキュメント属性 (続き)

属性名	構文	構文と説明
top-margin	整数	ページの上端とテキスト領域の上端との間隔 (文字数単位)。
width	整数	行の最大幅 (文字数単位)。
content-orientation	Oid	縦長または横長。
default-character-set	名前または Oid	
default-font	テキスト	
default-medium	Oid 名またはテキスト	
default-font	テキスト	
document-format	Oid	document-format 要素だけが置き換えられる。
document-name	テキスト	
number-up	整数	OID はその整数値に変換される。
plex	Oid 名	Simplex , duplex , tumble のいずれか。

説明を簡単にするために、表 8-1 では、プリンタ (両面、片面など)、複雑な構文の属性 (page-select など)、複数の値を持つ属性 (explicit-tab-stops など) を制御するための属性は除外しています。

たとえば、翻訳フィルタのためのコマンドが `/usr/pd/my-filter -d${document-format}` で、変更フィルタのためのコマンドが `/usr/pd/your-filter -o${content-orientation} -n${number-up}` であるとき、ユーザが変更と翻訳を要求すると、子プロセスは次のようなコマンドで実行されます。

```
/usr/pd/my-filter -simple-text|/usr/pd/your-filter
-oportrait -n2
```

8.1.3 フィルタの呼び出し

スーパーバイザがフィルタの呼び出しに使用する規則は次のとおりです。

- `no-filtering` が `true` の場合、フィルタは呼び出されません。

- `modification-filter` に値がある場合、そのフィルタが選択され、そのコマンド行が処理されます。
- `translation-filter` に値がある場合、そのフィルタが選択され、そのコマンド行が処理されます。
- ドキュメント・フォーマットがプリンタの `native-document-formats-ready` 属性に含まれていない場合、定義されたフィルタのリストは、`input-format` がドキュメントの `document-format` であり、`output-format` が `native-document-formats-ready` に対する任意の `document-format` である翻訳フィルタを、スーパーバイザが見つかるまで検索されます。このフィルタのコマンド行が処理されます。
- スーパーバイザは選択したフィルタを呼び出すコマンドを構築し、このコマンドで子プロセスをフォークして、フィルタ・プロセスとの間でドキュメント・データを転送するためのパイプを構築します。

8.1.4 エラー処理

一般に、フィルタの設定、呼び出し、実行中に発生するエラーは(ドキュメントだけではなく)ジョブを打ち切らせることになります。ジョブの打ち切りになる条件には次のようなものがあります。

- 存在しないフィルタの呼び出し。
- 適切な翻訳フィルタを見つけられない。
- 子プロセス内のエラー。

スーパーバイザは、`job-state-message` 属性に格納されたメッセージを通して、または、対象の物理プリンタに印刷されたエラー・ページを通して、イベント通知 (`job-aborted-by-server`) により、これらの状態をユーザに通知します。エラーが発生すると、ジョブはスプーラ上で保持 (`retained`) 状態になります。

8.1.5 フィルタ・プログラムの作成

フィルタ・プログラムは次の規則に従う必要があります。

- プログラムは `stdin` から読み込みを行う。
- プログラムはドキュメント・データだけを `stdout` に書き出す。エラー・メッセージやその他のメッセージは書き出さない。

- プログラムはプリンタ制御文字列を送ってはいけない。
- コマンド実行可能ファイルは、非特権ユーザによって置き換えることはできない。

8.2 テキストから PostScript への翻訳フィルタ

プリント・システム・ソフトウェアには翻訳フィルタが含まれています。このプログラムは simple-text ドキュメントを PostScript に翻訳し、オプションとして N アップ (片面への複数ページ印刷) 処理を実行します。このフィルタはインストレーション・プロシージャ実行時に /usr/pd/bin/trn_texttps として格納され、日本語キットをインストールした場合には /usr/il8n/bin/trn_texttps_j にも格納されます。trn_texttps_j は、標準版テキスト・トランスレータ trn_texttps に、日本語 EUC のファイルのサポートを追加したもので、その他の機能は同じです。trn_texttps_j を使用する際には、オプションに -CeucJP を指定してください。

PostScript 言語だけを取り扱うプリンタに送られるシンプルなテキスト・フォーマットのドキュメントは、PostScript に翻訳する必要があります。この翻訳は、ドキュメントの document-format 属性値が simple-text であり、物理プリンタ属性 native-document-formats-ready 値が PostScript であるときに起こります。それ以外のフォーマット、特に PCL が、native-document-formats-ready 属性に指定されている場合、スーパーバイザはそのデータを直接プリンタに送ります。

プリント・システム・ソフトウェアはコマンド・スクリプト /usr/pd/scripts/pxc_get_started を含んでおり、このコマンド・スクリプトは、スーパーバイザを作成すると、自動的にテキストから PostScript への翻訳フィルタを構成します。

表 8-1 には、Text-to-PostScript 翻訳プログラムでサポートされるすべてのコマンド・オプションを示しています。管理者は filter-definition 属性をコマンド・オプション置換に設定して、プリント・システム属性を翻訳プログラム・オプションに関連させることができます。

次の例は、コマンド・オプションの使用方法を示しています。

```
# pdset -c server \
-x filter-definition=\
'{name=text-to-ps \
type=translation \
```

```

input-format=simple-text \
output-format=PostScript \
command="/usr/pd/bin/trn_textps -N${number-up,0} \
${content-orientation,,-O${content-orientation}} \
${top-margin,,-a${top-margin}} \
${bottom-margin,,-b${bottom-margin}} \
${left-margin,,-c${left-margin}} \
${right-margin,,-d${right-margin}} \
-l${length} -w${width} ${number-pages,,-P} \
${repeated-tab-stops,,-t${repeated-tab-stops}}" }' \
red_sup

```

表 8-2 で説明する置換の規則は、次の例に適用されます。このコマンドは number-up=2 と width=80 を必要とするドキュメントに対して使用されま

```
# /usr/pd/bin/trn_textps -N2 -w80
```

プリント要求で指定されていない属性が省略時の値で表されたり、置き換えられたりしないのに対して、プリント要求で指定された属性は、置換等価に変換されます。

さらに、ドキュメント属性 number-up が 1, 2, 4 のいずれかである場合、フィルタは用紙あたりそれぞれ 1 (マージンつき)、2, 4 ページを印刷します。number-up が 0 または none は有効であり、これは number-up 処理を抑制します。

ドキュメント属性 content-orientation は、number-up 処理で、用紙上での論理ページの配置方法に影響します。

表 8-2: テキストから PostScript への翻訳プログラムのコマンド・オプション

オプション	対応する属性	説明
-a	top-margin	ページの上端に省略時のマージンを追加するための行数。有効な値は 0 以上の整数。
-b	bottom-margin	ページの下端に省略時のマージンを追加するための行数。有効な値は 0 以上の整数。
-B	(属性なし)	代替のグレーのバーを 3 行幅で印刷する。
-c	left-margin	ページの左側に省略時のマージンを追加する文字数。有効な値は 0 以上の整数。

表 8-2: テキストから PostScript への翻訳プログラムのコマンド・オプション (続き)

オプション	対応する属性	説明
-CeucJP	(属性なし)	日本語 EUC のテキスト・ファイルを日本語 PostScript プリンタに出力する。このオプションは、日本語キットをインストールした場合にのみ指定可能。
-d	right-margin	ページの右側に省略時のマージンを追加する文字数。有効な値は 0 以上の整数。
-l	length	ページごとの行数。新しいページが始まる前にページ上に印刷される列の数。有効な値は 0 よりも大きな整数。
-L	(属性なし)	行番号を印刷する。
-N	number-up	物理紙面上に印刷するページ・スポット数を指定する値。有効な値は、0、1、2、4。
-O	content-orientation	ページを長辺または短辺に沿ってフォーマットするよう指定する方向値。有効な値は、landscape (横長)、portrait (縦長)。
-P	number-pages	ページ番号をページの上部に印刷するかどうかを指定する値。このオプションには引数はない。省略時の設定ではページ番号を印刷しない。
-Q	(属性なし)	ラップなし。ページ (明示的な -w 設定、または用紙サイズから導き出す) に対して許可された長さより長い行を切り捨てるかどうかを指定する。このオプションには引数はない。これがあると切り捨てが指定される。省略時の値は、行ラップあり。

表 8-2: テキストから PostScript への翻訳プログラムのコマンド・オプション (続き)

オプション	対応する属性	説明
-S	default-medium	翻訳されたページがフォーマットされる用紙サイズ。列とカラムの省略時の値は導き出されるが、-w オプションと -l オプションによって無効になることもある。有効な値は次のとおり。a, b, com10, legal, 7x9, a0, b4, d, letter, 9x12_envelope, a1, b5, d1_envelope, monarch, a2, b6, e, postcard, a3, business_envelope, executive, 10x13_envelope, a4, c, folio, 10x14, a5, c4_envelope, halfletter, 11x14, a6, c5_envelope, ledger, 7_envelope
-t	repeated-tab-stops	タブをバイト位置 number+1, 2*number+1, 3*number+1 などに拡張するタブ幅の値。number の省略時の値は 8。入力中のタブ文字は、次のタブ設定に対応する適切な数のスペースで拡張される。有効な値は 0 よりも大きな整数。
-T	header-text	ページ・ヘッダ・テキストを印刷する。
-w	width	1 行あたりの文字数。行ラップまたは行の切り捨てが行われるまでに、1 行に印刷するカラムの数。有効な値は 0 よりも大きな整数。
-p	page-select	1 ページ以上の選択範囲をコマンドで区切って指定する。範囲は整数のページ番号か、コロンで区切った 2 つの整数で指定できる。3~6 ページと 9 ページを印刷するには次のコマンドを使用する。3:6,9

イベント通知の設定

プリント・システムは、通知サービスによってシステムの障害やイベントを報告します。通知サービスは電子メールで、またはシステム・コンソールにメッセージを配信します。

`notification-profile` 属性は、報告されるイベントに加えて報告の配信方法も指定します。次のオブジェクトに対して `notification-profile` を指定できます。

- スプーラ
- スーパーバイザ
- キュー
- 物理プリンタ
- 論理プリンタ
- ジョブ

管理者だけが、サーバ、キュー、プリンタの各オブジェクトに対して `notification-profile` を指定できます。エンド・ユーザはジョブを実行するとき、そのジョブに対して通知プロファイルを指定できます。

9.1 通知サーバ

通知は、通知サーバによって制御されます。サーバはスプーラが存在するホスト上で実行するデーモンです。通知サーバ `/usr/pd/lib/pdntfs` は、スプーラおよびスーパーバイザが発行した通知メッセージを中継して、配信します。通知サーバは電子メール・メッセージの送信と、`pdprintadmin` と `pdprintinfo` の GUI メッセージ領域へのメッセージ表示の責任を持ちます。

通知サーバは、必要とされるときに `inetd` によって起動されます。`inetd.conf` ファイルには、プリント・システムのインストール時に、エントリが追加されます。

9.2 通知デーモンの変更

Advanced Printing Software スプーラをホストしているシステムと同様にクライアント・システム上でも `pdntfs` デーモン (通知デーモン) が起動されるようになりました。このデーモンは、コンソール通知デーモン `pdcomntf` と同様に、Advanced Printing Software の GUI (`pdprintadmin` および `pdprintinfo`) にメッセージを送信します。`pdntfs` は、これらのクライアントで使用されるとき、`inetd` によって自動的に起動されます。

9.3 通知方法

プリント・システムは、通知を電子メール・メッセージとして、または、コンソールに対する即時メッセージとして配信します。

9.4 通知イベント

イベントは、個別に指定したり、イベントのクラスによって指定したりすることができます。`notification-profile` でクラスを指定すると、通知サーバはそのクラスの全イベントを報告します。次のイベントのクラスがサポートされます。

- `aborted` — ジョブが停止された、取り消された、打ち切られた
- `error` — ジョブの処理中にエラーが発生した
- `report` — 発生したイベントの報告クラスにある多くのイベントの1つ
- `state-changed` — 変更されたオブジェクトの状態
- `warning` — 実行されたジョブやプリンタの警告

9.4.1 サーバ・イベント

次の表は、サーバ通知プロファイルで有効なイベントをリストしたものです。

クラス	イベント	メッセージ
	<code>error-no-document</code>	ドキュメントはサーバからアクセスまたは利用できません。
Error	<code>error-other</code>	サーバは未知のエラー状態を検出しました。
	<code>error-unrecognized-resource</code>	この必須リソースはサーバには未知のリソースです。

9-2 イベント通知の設定

クラス	イベント	メッセージ
Report	report-server-clean-complete	サーバはクリーン処理を完了しました。
	report-server-clean-aborted	サーバはクリーン処理を打ち切りました。
State change	state-changed-server	サーバの状態が変わりました。
Warning	warning-other	サーバは未知の警告状態を検出しました。

次の例は、サーバ・オブジェクトに対する通知プロファイルの作成方法を示しています。

- エラー・クラスと報告クラスのすべてのイベントを含むサーバ red_spl のために通知プロファイルを作成し、コンソール・メッセージとして配信するには、次のコマンドを使用します。

```
# pdset -c server \  
-x "notification-profile=notification-events=error report" \  
-x notification red_spl
```

9.4.2 キュー通知イベント

クラス	イベント	メッセージ
Report	report_queue-clean-complete	キューはクリーン処理を完了しました。
	report_queue-clean-aborted	キューはクリーン処理を打ち切りました。
	report-queue-resubmit-aborted	キューは再実行処理を打ち切りました。
State-change	state-changed-queue	キューの状態が変わりました。

9.4.3 プリンタ通知イベント

クラス	イベント	メッセージ
State change	state-changed-printer	プリンタの状態が 変わりました。
Warning	warning-resource-needs-attention	必須リソースはユーザ の操作待ちです。
	warning-resource-needs-operator	リソースはオペレータ の操作待ちです。

9.4.4 ジョブ通知イベント

クラス	イベント	メッセージ
Aborted	job-aborted-by-server	サーバはプリント・ジョブ を打ち切りました。
	job-cancelled-by-operator	オペレータはジョブを取 り消しました。
	job-cancelled-by-user	ユーザはプリント・ジョブ を取り消しました。
Error	error-job-submission-not-complete	サーバはジョブのための最終印 刷処理を受信していません。
	error-no-resources	必須リソースは準備ができていな いか、利用不能になりました。
	error-past-deadline	プリント・ジョブの最終期限 時刻が過ぎています。
	error-past-discard-time	プリント・ジョブの破棄時 刻が過ぎています。
Report	report-file-transferred	ジョブのファイル転送が 完了しました。
	report-job-completed	プリント・ジョブは正常に 完了しました。

9-4 イベント通知の設定

クラス	イベント	メッセージ
	report-job-discarded	プリント・ジョブは破棄されました。
	report-job-promoted	プリント・ジョブは昇格されました。
	report-job-resumed	プリント・ジョブは再開されました。
	report-processing-started	サーバはジョブの処理を開始しました。
State change	state-changed-job	ジョブの状態が変わりました。
Warning	warning-close-to-discard-time	保持期限が近づいています。
	warning-job-modified	ジョブが変更されました。
	warning-job-paused	ジョブが一時停止されました。

LPD インバウンド・ゲートウェイ

インバウンド・ゲートウェイ・クライアント・デーモンでは、ユーザは `lp` コマンドや `lpr` コマンドを使用して、プリント・システムのプリンタに対してプリント要求を実行することができます。また、LPD インバウンド・ゲートウェイでは、Berkeley Software Distribution (BSD) ベースのプリント・システムに印刷する機能だけを持つアプリケーションや PC 統合ソフトウェアからでも、プリント・システム・サービスにアクセスできます。

この章では、LPD インバウンド・ゲートウェイ・クライアント・デーモンのインストールと構成の方法、および、`lp` コマンドと `lpr` コマンドおよびメッセージをプリント・システムにマップする方法を説明します。

10.1 LPD インバウンド・ゲートウェイの構成

Advanced Printing Software の LPD インバウンド・ゲートウェイは、LPD プrint・システムのライン・プリンタ・デーモン (`lpd`) を置き換える、または共存させるために設計されました。`lpd` プrint・システムを Advanced Printing Software に置き換えることは、すべての印刷が、置き換えられた Print・システムで行われることを意味します。Print・システムと `lpd` が共存するようにシステムを構成した場合は、`lpr` コマンドを使用してローカルに印刷したり、`lpr` コマンドを使用してリモート・Print・システムのプリンタに印刷したりできます。

インバウンド・ゲートウェイを構成するには、次の手順を実行します。

- `lpd` ソフトウェアとともに、あるいは単独で実行する LPD インバウンド・ゲートウェイを構成するスクリプトを実行します。このスクリプトは `/usr/pd/scripts/inbound_gw_config.sh` にあります。スクリプトの実行中に、希望する構成についての質問に答える必要があります。スクリプトは、システムが再起動するたびに、または Print・システムが再起動するたびに、インバウンド・ゲートウェイ構成が保持されるように `rc.config` ファイルを編集します。
- インバウンド・ゲートウェイ・デーモンは次のコマンドで起動できます。

```
# /sbin/init.d/apx start
```

LPD インバウンド・ゲートウェイはシステムのブート時に自動的に起動されます。

10.1.1 /etc/printcap ファイルへの変更

インバウンド・ゲートウェイを構成する際には、ライン・プリンタ・デーモン (lpd) と Advanced Printing インバウンド・ゲートウェイをプリント・サーバ・ホスト上で同時に実行することを選択できます。これは、lpr/lpd クライアントまたは Advanced Printing の CLI と GUI を使用してユーザがプリント・ジョブを実行できるので、最も柔軟な設定です。

着信 lpd プリント・ジョブを Advanced Printing のプリンタ宛に送るには、BSD プリンタ (lpr 要求で指定) に Advanced Printing の論理プリンタを関連付けるプリント・サーバ・ホスト上に printcap エントリを作成する必要があります。そのような printcap エントリは、プリンタを「ゲートウェイ・プリンタ」として宣言します。ゲートウェイ・プリンタに対して実行されたすべてのジョブは、Advanced Printing ジョブに変換され、Advanced Printing スプーラ内の関連付けられている論理プリンタに送信されます。

プリント・サーバ・ホストの printcap エントリに "rm=@dpa" および "rp=<logical printer>" という式を含めておくことにより、BSD プリンタをゲートウェイ・プリンタとして識別します。また、1) プリンタがローカル・ポート上にないこと、2) スプーラ・ディレクトリ、および 3) 最大ジョブ・サイズを指定するには、それぞれ lp, sd, および mx オプションも含める必要があります。そのようなエントリにある他の printcap フラグはすべて無視されます。

次の例は、"archie" という名前の Advanced Printing 論理プリンタに関連付けられているゲートウェイ・プリンタとして BSD プリンタ "bunker" を指定する /etc/printcap エントリを示しています。

```
lp1|bunker|archie|Digital LN17ps:\
:lp=:\
:rm=@dpa:\
:rp=archie:\
:sd=/var/spool/printer/archie:\
:mx#0:
```

別のホスト上のライン・プリンタ・デーモンまたは lpr クライアントがプリント・ジョブをプリンタ bunker に渡すとき、lpd はそのジョブをインバ

ウンド・ゲートウェイを介して Advanced Printing 論理プリンタ archie に渡します。

10.2 ジョブ取り消しメッセージの説明

ゲートウェイ・プリンタのジョブを表示するために `lpq(1)` コマンドを使用すると、ゲートウェイを介して送信される場合も含めて、プリンタのキューに登録されたすべてのジョブを表示します。`lpq` コマンドの出力で、000 のジョブ・エントリ番号は `pdpr` コマンドまたは `pdprint` の GUI を使用して送信されたネイティブな Advanced Printing ジョブであり、BSD ジョブではないことを示しています。ネイティブな Advanced Printing ジョブは、`lprm` コマンドでは取り消すことができません。Advanced Printing ジョブを削除するには、`pdrn(1)` コマンドを使う必要があります。

`lprm` コマンドによる `lpd` インバウンド・ゲートウェイからのジョブ取り消しメッセージは、次のように表示されます。

```
#lprm -P gwp 73
fafner_spl:1652 (73) cancelled
```

このメッセージは、Advanced Printing Software ジョブ ID が `fafner_spl:1652` の LPD ジョブ 73 が削除されたことを示しています。

10.3 LPD インバウンド・ゲートウェイ・マッピング

LPD プリント・ジョブが LPD インバウンド・ゲートウェイに到着すると、ジョブ・オプションが、等価なプリント・システムのジョブ属性とドキュメント属性に変換されます。次の表は、プリント・システムと、LPD インバウンド・ゲートウェイ経由で実行された LPD コマンドとのマッピングを示しています。

いくつかの Xerox プロトコル拡張オプションについて、Advanced Printing Software では同等の属性を提供していません。これらの拡張機能は、「プリント・システムの属性」欄に「(マッピングなし)」として記述されています。これらのサポートされていない属性のいずれかを指定してジョブを実行すると、ジョブの印刷は失敗し、ユーザは、選択されたオプションは Advanced Printing Software でサポートされていないという旨の電子メール・メッセージを受けとります。

表 10-1: ジョブおよびファイル・メッセージ

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
C(ClassName)	クラス名: (バナー) (ジョブ)	-C	(なし)	job-comment
H(Hostname)	ジョブの実行元ホスト: 1(ジョブ)	(なし)	(なし)	job-originating-host
J(Jobname)	ジョブ名: (バナー) (ジョブ)	-J	-t	job-name
M(user)	完了時にメールの送信先ユーザ: (ジョブ)	-m	-m	notification-profile
N(name)	データ・ファイルのソース: (ファイル)	(なし)	(なし)	document-name
P(name)	要求元ユーザ: (ジョブ)	(なし)	(なし)	job-originator, job-owner
S(dev) (inode)	ファイル情報: (ファイル)	-s	(省略時設定)	(マッピングなし)
U(file)	ファイルのリンク解除: (ファイル)	^{-r} (なし)	(なし)	(マッピングなし)
1(file)	Times Roman フォント・ファイル: (roff)(ジョブ)	-1	(なし)	(マッピングなし)
2(file)	Times Italic フォント・ファイル: (roff)(ジョブ)	-2	(なし)	(マッピングなし)

表 10-1: ジョブおよびファイル・メッセージ (続き)

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
3(file)	Times Bold フォント・ ファイル: (roff)(ジョ ブ)	-3	(なし)	(マッピング なし)
4(file)	Times Special フォント・ ファイル: (roff)(ジョ ブ)	-4	(なし)	(マッピング なし)

表 10-2: プリント・メッセージ

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
c(file)	CIF データ として印刷 /プロット	-c	-T cif	document- format
d(file)	DVI データ として印刷	-d	-T dvi	document- format
f(file)	ASCII とし て印刷	(なし)	-T ascii	document- format
g(file)	プロット・ データとし て印刷	-g	-T plot	document- format
l(file)	印刷不能 ファイル変 換印刷	-l	-y catv_filter -o nofilebreak	document- format
n(file)	ditroff 出力 として印刷	-n	-T ditroff	document- format
o(file)	PostScript として印刷	-o	-T ps -T PostScript	document- format
p(file)	"pr" によ る印刷	-p	-T pr	document- format

表 10-2: プリント・メッセージ (続き)

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
r(file)	fortran として印刷	-f	-T fortran	document-format
t(file)	troff 出力として印刷	-t	-T troff	document-format
v(file)	ラスター・イメージとして印刷	-v	-T raster	document-format

表 10-3: Sun プロトコル拡張

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
O(option_list)	SVR4 LP -o オプション用	(なし)	-o option_list	(無視される)
5(opt)(value)	SVR4 LP 機能用	(なし)	(オプションでない)	
5f(form)	SVR4 Forms 用	(なし)	-f form_name	(無視される)
5H(handling)	SVR4 Handling 用	(なし)	-H-f form_name	(無視される)
5p(method:end)	SVR4 Notification 用	(なし)	-P	(無視される)
5P(pagelist)	SVR4 Pages 用	(なし)	-P page_list	page-select
5q(priority)	SVR4 Priority 用	(なし)	-q priority	job-priority
5S(char set)	SVR4 Charset 用	(なし)	-S char_set	(無視される)
5T(type)	SVR4 Type 用	(なし)	-T input_type	document-format
5y(mode)	SVR4 Mode 用	(なし)	-y filter_mode	(無視される)

表 10-4: DEC プロトコル拡張

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
<(tray)	給紙トレイ選択	-< (upper lower manual ...)	(なし)	default-input-tray
>(bin)	排出トレイ選択	-> (bin)	(なし)	output-bin
G(nup)	N アップ	-G (nup)	(なし)	number-up
K(sides/plex)	片面/両面	-K (sides/plex)	(なし)	sides, plex
O(orientation)	ページの印刷向き	-O (orientation)	(なし)	content-orientation

注意

Advanced Printing Software がサポートしていない Xerox プロトコル拡張は、次の表の「プリント・システムの属性」欄に（マッピングなし）として記述されています。これらのサポートされていない拡張を指定して実行したジョブは、印刷されません。サポートされていない拡張を使用してジョブを実行したユーザは、選択されたオプションはサポートされていないというメール・メッセージを受けとります。

表 10-5: Xerox プロトコル拡張 - DocuSP および DocuPrint

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
C" (doc-format)	ドキュメント・フォーマット、 doc-format= ps postscript ascii tiff pcl interpress ip lcds (サポートされていない)	-C	-o5	document-format= PostScript PostScript simple-text TIFF PCL Interpress Interpress LCDS
C" (orientation)	ドキュメントの向き、 orientation= portrait inverseportrait ipportrait landscape inverselandscape ilandscape	-C	-o	content-orientation= portrait reverse-portrait landscape reverse-landscape

表 10-5: Xerox プロトコル拡張 - DocuSP および DocuPrint (続き)

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
C" (staple) "	ドキュメントのホッチキスどめ, staple=staple nostaple	-C	-0	finishing= staple-top-left
C" (order)"	ページ順, order=1ton nto1	-C	-0	page-order-received= first-to-last last-to-first
C" (simplex)"	片面印刷	-C	-0	sides=1
C" (duplex)"	両面印刷	-C	-	sides=2
C" (tumble)"	短辺とじ	-C	-0	sides=2, plex=tumble

表 10-5: Xerox プロトコル拡張 - DocuSP および DocuPrint (続き)

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
C" (mediumsize)"	用紙サイズ, mediumsize= usletter uslegal a4 USLetter USLegal usledger a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 isob0 isob1 isob2 isob3 isob4 isob5 isob6 isob7 isob8 isob9 isob10 jisb0 jisb1 jisb2 jisb3 jisb4 jisb5 jisb6	-C	-0	default-medium= na-letter-white na-legal-white iso-a4-white iso-a4-white na-letter-white na-legal-white ledger-white iso-a0-white iso-a1-white iso-a2-white iso-a3-white iso-a4-white iso-a5-white iso-a6-white iso-a7-white iso-a8-white iso-a9-white iso-a10-white iso-b0-white iso-b1-white iso-b2-white iso-b3-white iso-b4-white iso-b5-white iso-b6-white iso-b7-white iso-b8-white iso-b9-white iso-b10-white jis-b0-white jis-b1-white jis-b2-white jis-b3-white jis-b4-white jis-b5-white
C" (xshift=x)"	x 方向への ページのシ フト	-C	-0	x-image-shift=x
C" (yshift=x)"	y 方向への ページのシ フト	-C	-0	y-image-shift=y
C" (font)"	フォント名	-C	-0	(マッピングなし)

Advanced Printing Software がサポートしていない Xerox プロトコル拡張は、次の表の「プリント・システムの属性」欄に（マッピングなし）として記述されています。これらのサポートされていない拡張を指定して実行したジョブは、印刷されません。サポートされていない拡張を使用してジョブを実行したユーザは、選択されたオプションはサポートされていないというメール・メッセージを受けとります。

表 10-6: Xerox プロトコル拡張 - DocuSP のみ

用途	説明	lp オプション	lpr オプション	プリント・システムの属性
C" (bind=edge)"	バインディング, edge= top bottom left right	-C	-0	バインディング, edge= top bottom left right
C" (stitch=how)"		-C	-0	finishing= staple-top-left staple-bottom-left edge-stitch
C" (uncollate) "	照合	-C		output=no-page-collate
C" (booklet) "	フィニッシング	-C		finishing=saddle-stitch
C" (signature)"	記号オプション	-C	-0	number-up=simple-2-up
C" (slipsheet)"	合紙オプション	-C	-0	job-sheets=job-copy-wrap
C" (prefinish=<option>)"	用紙プレフィニッシュ・オプション	-C	-0	(マッピングなし)
C" (pagesto-print =<CARDINAL><CARDINAL>)"	ページ・オプション	-C	-0	(マッピングなし)

表 10-6: Xerox プロトコル拡張 - DocuSP のみ (続き)

用途	説明	lp オプション	lpr オプション	プリント・システムの属性
C" (media=<size>:<type>:< color>:<weight>)"	用紙カスタム・オプション	-C	-0	(マッピングなし)
C" (mediaType=<type>)"	用紙カスタム・オプション	-C	-0	(マッピングなし)
C" (mediatSize=<size>)"	用紙カスタム・オプション	-C	-0	(マッピングなし)
C" (mediaColor=<color>)"	用紙カスタム・オプション	-C	-0	(マッピングなし)
C" (mediaWeight=<weight>)"	用紙カスタム・オプション	-C	-0	(マッピングなし)
C" (opacity=transparency opaque)"	その他のオプション	-C	-0	(マッピングなし)
C" (xshift2=x)"	ページの裏面を x 方向にシフト	-C	-0	(マッピングなし)
C" (yshift2=x)"	ページの裏面を y 方向にシフト	-C	-0	(マッピングなし)
C" (account=<text>)"	課金オプション	-C	-0	(マッピングなし)
C" (hipentry= <CAR-DINAL>)"	LCDS インデックス・オプション	-C	-0	(マッピングなし)
C" (recipient=<name>)"	受取人オプション	-C	-0	(マッピングなし)
C" (res= <CAR-DINAL>)"	解像度オプション	-C	-0	(マッピングなし)
C" (outputbin)"	その他のオプション	-C	-0	(マッピングなし)

表 10-6: Xerox プロトコル拡張 - DocuSP のみ (続き)

用途	説明	lp オプション	lpr オプション	プリント・システムの属性
C" (inputbin)"	その他のオプション	-C	-0	(マッピングなし)
C" (thicken)"	その他のオプション	-C	-0	(マッピングなし)
C" (offenhance)"	その他のオプション	-C	-0	(マッピングなし)
C" (onenhance)"	その他のオプション	-C	-0	(マッピングなし)
C" (bp)"	その他のオプション	-C	-0	(マッピングなし)
C" (ep)"	その他のオプション	-C	-0	(マッピングなし)
C" (diagnostic)"	その他のオプション	-C	-0	(マッピングなし)

注意

Advanced Printing Software がサポートしていない Xerox プロトコル拡張は、次の表の「プリント・システムの属性」欄に（マッピングなし）として記述されています。これらのサポートされていない拡張を指定して実行したジョブは、印刷されません。サポートされていない拡張を使用してジョブを実行したユーザは、選択されたオプションはサポートされていないというメール・メッセージを受けとります。

表 10-7: Xerox プロトコル拡張 - DocuPrint のみ

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
C" (hc=color) "	強調色,color=red blue green cyan magenta yellow cardinal royalblue ruby violet customName customName	-C	---	強調色,color=red blue green cyan magenta yellow cardinal royalblue ruby violet customName customName
C" (mc=color)"	強調マッピング色, 色の値は強調色と同じ,color	-C	---	highlight-mapping-colour, 値は highlight-colour と同じ
C" (hcr=rendering)"	強調色レンダリング・アルゴリズム, rendering= automatic colortohighlight colortables presentation pictorial	-C	---	強調色レンダリング・アルゴリズム, rendering= automatic colortohighlight colortables presentation pictorial

表 10-7: Xerox プロトコル拡張 - DocuPrint のみ (続き)

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
C" (hcm=action)"	強調色不一致動作, action= abort ignore operator	-C	---	強調色不一致動作, action= abort ignore operator
C" (stitch)"	ドキュメントのホッチキスどめ, stitch= stitch nostitch	-C	---	finishing= staple-top-left ---
C" (thick=x)"	濃度, x= 01 10	-C	-0	thickening-specification= thickening-entire-documents thickening-bit-map-images
C" (drilled) "	用紙ブレフィニッシュ・オプション	-C	---	(マッピングなし)
C" (media=<size>: <type>: <paper color>: <weight>)"	用紙カスタム・オプション	-C	-0	(マッピングなし)
C" (<n>x<m>)"	用紙カスタム・オプション	-C	---	(マッピングなし)
C" (size)"	フォント・サイズ	-C	---	(マッピングなし)
C" (fontsize)"	フォント・サイズ	-C	---	(マッピングなし)
C" (disposition=<>)"	分解サービス・オプション	-C	---	(マッピングなし)

表 10-7: Xerox プロトコル拡張 - DocuPrint のみ (続き)

用途	説明	lpr オプション	lp オプション	プリント・システムの属性
C" (back-ground=<>)"	分解サー ビス・オブ ション	-C	---	(マッピング なし)
C" (bf=<>)"	分解サー ビス・オブ ション	-C	---	(マッピング なし)

LPD アウトバウンド・ゲートウェイ

LPD アウトバウンド・ゲートウェイのスーパーバイザである `pdspvlpr` は、ライン・プリンタ・デーモン (LPD) プロトコルを使用して、リモート・サーバとプリンタにプリント・ジョブ要求を転送します。この処理を行うため、LPD スーパーバイザはプリント・システム・ジョブとその属性を RFC1179 に基づいた要求に変換し、指定した LPD サーバにその要求を転送します。

LPD スーパーバイザは、次のプリンタでジョブを印刷できます。

- リモート・ホストがライン・プリンタ・デーモン (`lpd`) ソフトウェアを実行しており、ホストがリモート・ジョブを受け付けるように構成されている場合に、そのホストに接続されたプリンタ
- `lpd` プロトコルを直接サポートするプリンタ

LPD スーパーバイザは、通常のプリント・システム・スーパーバイザ (`pdsprv`) によってサポートされないプリンタにアクセスする際に特に便利です。たとえば、使用中の環境に UNIX ホストに接続されている従来のプリンタがあり、それらのプリンタが特殊な UNIX フィルタを必要とする場合は、LPD スーパーバイザを使用して Advanced Printing Software ジョブをそのプリンタに渡すこともできます。同様に、Advanced Printing Software を実行していないホストに接続されているプリンタに対してプリント・ジョブを実行したい場合は、LPD スーパーバイザを使用してプリント・ジョブをそのプリンタに渡すことができます。

11.1 printer-address 属性の変更

アウトバウンド・ゲートウェイ・スーパーバイザ (`pdspvlpr`) によりサポートされている物理プリンタの `printer-address` 属性を変更する場合には、新しいアドレスを有効にするためにスーパーバイザをシャットダウンして再起動する必要があります。

次のコマンドを使って `lpd` アウトバウンド・ゲートウェイをシャットダウンし、再起動すると新しいプリンタ・アドレスが有効になります。サーバ

をシャットダウンし再起動するには root でなければならぬことに注意してください。

```
# pdshutdown -c server server_name  
# /usr/pd/lib/pdspvlpr server_name
```

11.2 プロトコル拡張

LPD スーパーバイザは、次に示すいくつかの一般的なプロトコル拡張を使用して、リモート・プリント・サーバと通信することができます。

- 1179 - 拡張のない RFC1179 (LPD) プロトコル
- Solaris - SunSoft の Solaris 拡張付きの LPD プロトコル
- DigitalUNIX - Tru64 UNIX 拡張付きの LPD プロトコル
- Xerox - Xerox DocuSP と DocuPrint 拡張付きの LPD プロトコル

物理プリンタ・オブジェクトを LPD スーパーバイザに作成するときには、最も適切なプロトコル指示子を指定します。これによって、ユーザは、LPD サーバやプリンタで利用可能なさまざまなプロトコル拡張を活用するジョブや、ドキュメントのオプションを指定できます。表 11-1 から表 11-4 には、プロトコルおよびサポートされる拡張で利用可能なコマンド・オプションと引数を示しています。

11.3 物理プリンタの作成

LPD スーパーバイザは、"bsd" プリンタ接続方法をサポートします。物理プリンタ・オブジェクトを作成するそれぞれのプリント・システムに対して、リモート・マシンと LPD プリンタ・キューのリモート・プリンタ名を指定する必要があります。

LPD スーパーバイザは、プリンタと正しく通信を行うために、プリンタの名前を知る必要があります。スーパーバイザは printer-address 属性を使用して LPD プリンタの名前を調べます。ゲートウェイはこの名前を使用して次のものを判断します。

- リモート・ホスト
- リモート・ホスト上のプリンタ名
- プロトコル適合性 (サポートする LPD 拡張)

これらの3つの項目は、次のフォーマットを使用して指定する必要があります。

```
printer-address=remotehost,printername,protocol_conformance
```

プロトコル適合性の有効なキーワードは次のとおりです。

```
1179
Solaris
DIGITAL UNIX
Xerox
```

リモート LPD プリンタ `big_red` を IP アドレス `fafner.xyz.com` のマシンに設定し、それを Tru64 UNIX プロトコル拡張を使用するように定義するには、次のように `printer-address` 属性を指定します。

```
printer-address="fafner.xyz.com,big_red,DigitalUNIX"
```

物理プリンタを作成した後は、第6章で概説した手順に従って、そのプリンタをキューに関連付けることができます。

11.4 LPD スーパーバイザではサポートされない機能

LPD スーパーバイザは、通常のスーパバイザ (`pdsprv`) と同じ機能群はサポートしません。次の機能はサポートされません。

- 翻訳フィルタまたは変更フィルタ — ドキュメント・データ・フィルタリングはサポートされません。
- セパレータ・ページ生成 — LPD スーパーバイザは、`job-sheets` 属性が `job-copy-start` や `job-copy-wrap` に設定されているときに、パナー・ページの要求を渡します。ただし、セパレータ・ページ・データ自体は生成しません。
- 用紙の選択、ただし Xerox DocuSP または DocuPrint プリンタに対して Xerox 拡張を使用する場合を除きます。
- `printer-baud-rate`、`printer-stop-bits`、`printer-data-bits`、`printer-parity`、`printer-input-flow-control`、`printer-output-flow-control`、`printer-connection-level` などの接続属性は、LPD スーパーバイザでは無視されます。`printer-connection-method` の唯一の有効な値は "bsd" です。

11.5 アウトバウンド・ゲートウェイ・マッピング

次の表は、プリント・システムの要求と属性が、LPD のコマンドと属性に変換される方法を示しています。

表 11-1: ジョブとファイル・メッセージ

プリント・システムの属性	キー	用途	説明	省略時の値	lpr オプション
job-comment	C	C (Class-Name)	クラス名: (バナー) (ジョブ)	(ホスト名)	-C
job-originating-host	H	H (Host-name)	ジョブの実行元ホスト: 1(ジョブ)	(ホスト名)	(なし)
job-name	J	J (Jobname)	ジョブ名: (バナー) (ジョブ)	(最初のファイル名)	-J
job-sheets	L	L (user)	バナー・ページ	なし	(なし)
notification-profile	M	M (user)	終了メール通知先ユーザ: (ジョブ)	なし	-m
document-name	N	N (name)	データ・ファイルのソース: (ファイル)	(ファイル名)	(なし)
job-originator, job-owner	P	P (name)	要求元ユーザ: (ジョブ)	(ユーザ)	(なし)

表 11-2: Sun プロトコル拡張

プリント・システムの属性	キー	用途	説明	省略時の値	lpr オプション
page-select	5P	5P(pagelist)	SVR4 Pages 用	なし	(なし)
job-priority	5q	5q(priority)	SVR4 Priority 用	なし	(なし)
document-format	5T	5T(type)	SVR4 Type 用	なし	(なし)

表 11-3: DIGITAL プロトコル拡張

プリント・システムの属性	キー	用途	説明	省略時の値	lpr オプション
default-input-tray	<	<(tray)	給紙トレイ選択	なし	-< (upper lower manual ...)
output-bin	>	<(bin)	排出トレイ選択	なし	-> (bin)
number-up	G	G(nup)	N アップ	なし	-G (nup)
sides, plex	K	K (sides/plex)	片面/両面	なし	-K (sides/plex)
content-orientation	O	O (orientation)	印刷の向き	なし	-O (orientation)

表 11-4: Xerox プロトコル拡張 - DocuSP および DocuPrint

プリント・システムの属性	キー	用途	説明	省略時の値	lpr オプション
document-format= PostScript simple-text TIFF PCL Interpress	C	C" (doc-format) " 1	ドキュメントのフォーマット, doc-format= ps postscript ascii tiff (DocuSP のみ) pcl interpress (DocuPrint のみ)	なし	-C
content-orientation= portrait reverse-portrait landscape reverse-landscape	C	C" (orientation) "	ドキュメントの向き, orientation= portrait inverseportrait landscape inverselandscape	なし	-C
finishing= staple-top-left	C	C" (staple) "	ドキュメントのホッチキスどめ, staple= staple	なし	-C
page-order-received= first-to-last last-to-first	C	C" (order) "	ページ順, order= 1ton nto1	なし	-C
sides=1	C	C" (simplex) "	片面印刷	なし	-C

表 11-4: Xerox プロトコル拡張 - DocuSP および DocuPrint (続き)

プリント・システム の属性	キー	用途	説明	省略 時の 値	lpr オプ ション
sides=2	C	C" (duplex)"	両面印刷	なし	-C
sides=2, plex=tumble	C	C" (tumble)"	短辺とじ	なし	-C

表 11-4: Xerox プロトコル拡張 - DocuSP および DocuPrint (続き)

プリント・システムの属性	キー	用途	説明	省略時の値	lpr オプション
default-medium= na-letter-white na-legal-white iso-a4-white na-letter-white na-legal-white ledger-white iso-a0-white iso-a1-white iso-a2-white iso-a3-white iso-a4-white iso-a5-white iso-a6-white iso-a7-white iso-a8-white iso-a9-white iso-a10-white iso-b0-white iso-b1-white iso-b2-white iso-b3-white iso-b4-white iso-b5-white iso-b6-white iso-b7-white iso-b8-white iso-b9-white iso-b10-white jis-b0-white jis-b1-white jis-b2-white jis-b3-white jis-b4-white jis-b5-white jis-b6-white jis-b7-white jis-b8-white jis-b9-white jis-b10-white	C	C" (mediumsize)"	用紙のサイズ, mediumsize= usletter (DocuPrint only) uslegal (DocuPrint only) a4 (DocuPrint only) The following values are for DocuSP only: USLetter USLegal usledger a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 isob0 isob1 isob2 isob3 isob4 isob5 isob6 isob7 isob8 isob9 isob10 jisb0 jisb1 jisb2 jisb3 jisb4 jisb5 jisb6 jisb7 jisb8 jisb9 jisb10	なし	-C

表 11-4: Xerox プロトコル拡張 - DocuSP および DocuPrint (続き)

プリント・システム の属性	キー	用途	説明	省略 時の 値	lpr オプ ショ ン
x-image- shift=x	C	C" (xshift=x)"	x 方向へのページ のシフト	なし	-C
y-image- shift=y	C	C" (yshift=x)"	y 方向へのページ のシフト	なし	-C

表 11-5: Xerox プロトコル拡張 - DocuSP のみ

プリント・シス テムの属性	キー	用途	説明	省略 時の 値	lpr オ プショ ン
binding-edge= top-edge bottom- edge left-edge right-edge	C	C" (bind=edge) "	バインディング 側, edge= top bottom left right	なし	-C
finishing= staple-bottom- left edge-stitch	C	C" (stitch=how)"	ドキュメントのホッ チキスどめ, how= singlelandscape duallandscape	なし	-C
output=no-page- collate	C	C" (uncollate) "	照合	なし	-C
finishing=sad- dle-stitch	C	C" (booklet) "	フィニッシング	なし	-C
number- up=simple-2-up	C	C" (signature)"	記号オプション	なし	-C
job-sheets=job- copy-wrap	C	C" (slipsheet)"	合紙オプション	なし	-C

トラブルシューティング

この章では、プリント・システムで検出される可能性のある一般的な問題のいくつかを説明するとともに、問題を修正するための対処方法について説明します。

12.1 サーバ問題の解決

この節では、通常のプリント・システム処理で起こる可能性のあるサーバ・エラーについて説明します。これらのエラーはサーバ・エラーとして記述されますが、必ずしもスプーラやスーパーバイザのオブジェクトだけが原因のエラーではなく、プリント・システムの他のオブジェクトが原因の場合もあります。

12.1.1 どのサーバ・プロセスが実行中であることを調べる

スーパーバイザまたはスプーラ・プロセスがクライアント要求に応答しない場合、次のコマンドを使用して、どのサーバ・プロセスが実行中であることを判断できます。

```
# ps -ef | grep pd
```

`pdsplr`、`pdspvr`、`pdsplpr` に対するプロセス・エントリが表示されない場合、正しいホストを見ているかどうかを確認してください。なくなっているサーバ・プロセスを再起動するには、次のコマンドの1つを使用してください。

```
# /usr/pd/lib/pdsplr  
# /usr/pd/lib/pdspvr  
# /usr/pd/lib/pdsplpr
```

スーパーバイザまたはスプーラのプロセスが実行中であっても、クライアントからの要求に応答しない場合は、`pdl`s コマンドを使用してオブジェクトの状態を調べることができます。

```
# pdls -c server -r all s line server_name
```

サーバから応答がない場合、プロトサーバ・デーモンが実行中であるかどうかを確認してください。

```
# rpcinfo -u 105004
```

プロトサーバが実行中の場合、次のように表示されます。

```
# program 105004 version 1 ready and waiting
```

12.1.2 サーバは実行中だが何も動かない

前述の手順を使用してどのコンポーネントが実行中であるのかと、それらのうちのどれが `rpcinfo -u` コマンドまたは `-t` コマンドに応答するのかを調べてください。プロトサーバが応答しない場合は、次の手順を使用してプリント・システムを再起動してください。

1. すべての `pdspvr` , `pdsplr` , `pdsplvr` プロセスについて `kill -9` を実行します。
2. フォーム `/var/pd/pts/.pts*` のすべてのファイルを消去します。
3. `inetd` プログラムの PID を調べて、それに "hangup" 信号を送信します。

```
# ps ax | grep inetd
462 ?? I 0:16.91 /usr/sbin/inetd
ttyp7 S+ 0:00.02 grep inetd
# kill -1 462
```

4. もう一度プロトサーバ・デーモンと通信してみます。

```
# rpcinfo -u host 105004
```

5. プリント・システム・スプーラ (`pdsplr`) を再起動し、今は通信できるかどうかをチェックします。

```
# /usr/pd/lib/pdsplr servername
# pdls -c server servername
```

スプーラが応答する場合は、他のサーバを同様に再起動してください。

12.1.3 スーパーバイザがシャットダウンしない

`pdshutdown` コマンドが、スーパーバイザ・サーバの停止に完全に有効ではない場合があります。そのような場合は、スーパーバイザ・サーバが終了するまで少なくとも 2 分間待ってください。物理プリンタが `printing` (印刷中) 状態でないことを検証することによって、サーバがプリンタに対してジョブを実際には印刷していないことを確認してください。プリンタに一時

停止されたジョブがあるかどうかを確認してください。スーパーバイザは、`pdshutdn` コマンドで `-w now` を指定しない限り、ジョブが一時停止されている場合にはシャットダウンしません。次に、スーパーバイザの PID を調べ、`"kill -9"` を使用してそれを終了させてください。

```
# pdls -c printer servername:
printer-name printer-realization printer-state enabled
-----
LN17ps_PP    physical      idle         yes
LGP          physical      idle         no
Richs_PP    physical      idle         no
Sharie_PP   physical      idle         no
LN03R       physical      idle         no
ln17bert    physical      idle         no
lps17_sue   physical      idle         no
Null_PP     physical      idle         no
# ps ax | grep pds
29874 ??    I    0:00.64 /usr/pd/lib/pdsplr merle_spl
30481 ??    S    0:34.69 /usr/pd/lib/pdspvr merle_sup
5008 ttyp7  S +  0:00.02 grep pds
# kill -9 30481
```

12.1.4 スプーラがシャットダウンしない

前述の「スーパーバイザがシャットダウンしない」と同じ手順を使用してスプーラ・プロセスを停止させます。

12.2 ジョブとプリントの問題

この節では、通常のプリント・システムの処理中に起こる可能性のあるジョブとプリントのエラーを説明します。これらのエラーはジョブやプリントのエラーとして記述されていますが、これは必ずしもプリントやジョブのオブジェクトだけが原因のエラーではなく、プリント・システムの他のオブジェクトが原因である可能性もあります。

12.2.1 PostScript ドキュメントで PostScript プログラム・コードが印刷される

PostScript コードを生成するアプリケーションの中には、PostScript のリードイン・シーケンス `"%!"` を誤って取り除くものがあります。ドキュメントが `"%!"` で始まらない場合、プリント・システムのスーパーバイザはそのファイルをテキスト・ファイルであると誤って判断し、それをプリンタに解釈させずに、テキスト・リストに翻訳します。

ジョブを再実行して、document-format を PostScript と指定します。

12.2.2 物理プリンタが接続状態でハングする

ジョブが割り当てられるときに、出力デバイスがアクセスできない場合、物理プリンタ・オブジェクトは "connecting-to-printer (プリンタに接続中)" 状態のままになります。これが起こるのには、次のような複数の理由があります。

- プリンタが他のジョブの印刷でビジー状態である。
- プリンタがオフラインである。
- プリンタの電源が入っていない。
- プリンタへのネットワーク・パスにアクセスできない。
- プリンタは双方向セッション・コントロール (printer-connection-level=4) 用に構成されているが、デバイスがこのレベルをサポートできない

プリンタ・オブジェクトを "idle (アイドル)" に戻す唯一の方法は、ジョブを消去 (取消し) することです。

12.2.3 処理待ち状態のままのジョブ

次の手順を使用して、ジョブが実行された後に処理待ち状態のままになる理由を調べてください。

- 物理プリンタが使用可能になっており、その状態がアイドルであることを確認します。
- ジョブを実行したときに明示的に指定された、あるいは初期値オブジェクトによって暗黙に指定された、すべてのジョブとドキュメントの要件が、キューに関連付けられている少なくとも1つの物理プリンタ上で、両方とも "supported" および "ready" であることを確認します。

たとえば、初期値オブジェクトで job-sheets, document-sheets, 両面印刷, N アップ (片面に複数ページ印刷) などを指定している論理プリンタに対してジョブを実行した場合、対応する xxx-supported 属性と xxx-ready 属性が物理プリンタに設定されていることをチェックしてください。

これを行うには、次のコマンドを使用します。

```
# pdls -c p -r job-sheets-ready pp_name
# pdls -c p -r document-sheets-ready pp_name
# pdls -c p -r plexes-supported pp_name
```

12-4 トラブルシューティング

```
# pdls -c p -r numbers-up-supported pp_name
```

12.3 システム・エラーとエラー情報

この節では、プリンタやサーバのオブジェクト以外のプリント・システムのサブシステムに関連するエラーとその解決方法について説明します。さらに、この節ではエラー情報がシステム内のどこにあるのかという情報も説明します。

12.3.1 コンソール通知が働かない

コンソールに通知メッセージが表示されない場合、あるいは、メール通知メッセージが送信されていない場合は、コンソール通知デーモン `pdconntf` が実行中かどうかを、次の手順を実行することによって確認してください。

- `ps` コマンドを使用して、デーモンが実行中であるかどうかを調べます。

```
# ps aux | grep pdconntf
```

- 実行されていない場合は、端末ウィンドウで次のように実行してください。

```
# /usr/pd/lib/pdconntf
```

12.3.2 システム・エラー情報の検出

ほとんどすべてのエラーについて、ユーザ・コンソールに書き込まれたクライアント・メッセージには、利用可能なすべてのエラー情報が含まれています。場合によっては、特に、ネットワーク、メモリの利用可能度、プロセス・スロットなどのシステム・リソースに関連する問題の場合は、追加のエラー情報がシステム・ログに書き込まれます。プリント・システムのクライアントとサーバは、特定のタイプのシステム・エラーが起こると、システム・ログにメッセージを書き込みます。Tru64 UNIX システムでは、システム・ログは `/var/adm/syslog.dated/` にあります。サーバはメッセージを `lpr.log` に書き込み、クライアントはメッセージを `user.log` に、ときには `lpr.log` に書き込みます。プリント・システム・プログラムがメッセージを `daemon.log` に書き込むこともあります。

コマンドの構文が正しく、オプション値がすべて正しいことを確認した後で、意味不明なコマンド・エラーを検出した場合は、システム・ログにさらに情報があるかどうかを確認してください。

この章では、プリント・システムのエンド・ユーザ、オペレータ、または管理者が読み出ししたり、設定したりできるオブジェクト属性について説明します。この章は、オブジェクトごとにまとめられています。それぞれの節では、特定のオブジェクトをサポートする属性について説明します。

A.1 サーバ属性

次の属性は、スーパーバイザおよびスプーラの両方に関連しています。

access-control-list

さまざまなドメインに所属する、特定の権限を持つプリント・システム・ユーザの特権レベルを指定します。

availability

サーバの一般的な可用性を示します。サーバが使用不能になっている場合には `none`、サーバが使用可能になっている場合には `normal` に設定されます。

descriptor

サーバのテキスト表現です。

enabled

サーバがクライアントからのプリント要求を受け付けているかどうかを示します。この属性は、`pdenable` コマンド、または `pddisable` コマンドを使用して設定されます。

events-supported

イベント通知のときにサーバでサポートされる、イベント・タイプおよびイベント・クラスを指定します。

job-attributes-visible-to-all

ジョブの所有者でないユーザが `pdls` コマンドを使用して表示できるジョブ属性のリストを指定します。省略時の設定では、このようにして表示できる属性は *intervening-job* だけです。

locale

エラー、通知、ログインなどのメッセージを構成するといった国際化タスクを実行するときに、サーバが使用するロケールの仕様が含まれています。この属性は、サーバによって locale 環境変数から設定されます。

locales-supported

サーバでサポートされるロケールのリストが含まれています。

message

オブジェクトに関連付けられ、オブジェクト状態に関する何らかの情報をユーザに示すための読み取り可能な文字列です。

modify-individual-document-supported

1つの複合ドキュメント・ジョブ内の個々のドキュメントを、サーバで変更できるかどうかを示します。

multiple-documents-supported

オブジェクトが複合ドキュメント・ジョブを処理できるかどうかを示します。

notification-delivery-methods-ready

イベント通知用に、このオブジェクトで利用できる配信方式を示します。

notification-delivery-methods-supported

イベント通知用に、このオブジェクトでサポートされる配信方式を示します。

object-class

オブジェクトのクラスを示します。

object-class-supported

サーバでサポートされているオブジェクト・クラスを示します。

physical-printers-ready

スプーラのキューすべてに関連付けられている物理プリンタを示すか、またはスーパーバイザのプリンタすべてを参照します。

physical-printers-supported

サーバでサポートされている物理プリンタを示します。

server-name

サーバの名前を示します。この属性は、サーバが作成されるときに設定されます。

server-state

サーバの状態を示します。サーバの状態には、ready (処理可能)、paused (一時停止)、および terminating (後処理中) があります。

server-type

サーバのタイプを指定します。サーバのタイプには、spooler、supervisor、notification server、および protoserver があります。

transfer-methods-supported

サーバでサポートされているプリント・ファイルの転送方式を示します。有効な方式は、with-request、socket、および file-reference です。

A.1.1 スプーラ属性**hold-jobs-interrupted-by-printer-failure**

再起動後にスプーラに戻されたジョブを、held (保留) または pending (処理待ち) 状態にするかどうかを指定します。

logical-printers-ready

論理プリンタを示します。

A.1.2 スーパーバイザのみ**filter-definition**

プログラムをフィルタとして定義し、このプログラムを起動するために必要な情報を含んでいます。

maximum-number-of-printers-supported

スーパーバイザが動作させる実プリンタの数を指定します。この上限に達した後は、スーパーバイザ上に物理プリンタを作成しようとしても、失敗します。

number-of-printers-supported

スーパーバイザが制御するように管理者が設定した、実プリンタの数を指定します。number-of-printers-supported の値を、maximum-number-of-printers-supported によって設定された上限を超えた値に設定しようとすると、失敗します。

printer-connection-methods-supported

プリンタがサーバ・ホストに接続される方法を示します。

A.2 プリンタ属性

associated-queue

プリンタに関連付けられているキューを示します。

associated-server

オブジェクトが関連付けられているサーバの名前です。この名前は、作成時にオブジェクトによって設定されます。

availability

オブジェクトの一般的な可用性を示します。オブジェクトが使用不能になっている場合には `none`、オブジェクトが使用可能になっている場合には `normal` に設定されます。

binding-edges-supported

`binding-edges-supported` 属性は、プリンタでサポートされている `binding-edge` の値を示します。

character-sets-supported

`character-sets-supported` 属性は、プリンタでサポートされている文字セット・エンコードを示します。

content-orientations-supported

`content-orientations-supported` 属性は、プリンタでサポートされているドキュメント・コンテンツの印刷向きを指定します。この属性値には、プリンタに送信されるドキュメント・コンテンツの印刷向きに関するすべての情報が含まれている必要があります。論理プリンタの `content-orientations-supported` に、ドキュメントの印刷向きに一致する値が含まれていない場合には、スプーラは、そのプリント要求を拒否します。物理プリンタの `content-orientations-supported` に、ドキュメントの印刷向きと一致する値が含まれていない場合には、スプーラはそのジョブを処理待ち (pending) の状態にします。

descriptor

`descriptor` 属性は、指定されたオブジェクトのテキスト表現を提供します。

document-formats-supported

プリンタでサポートされるドキュメント・フォーマットを指定します。

document-sheets-supported

プリンタでサポートされる補助シートを指定します。

enabled

指定されたオブジェクトが使用可能になっており、クライアントからのプリント要求を受け付けることができるかどうかを示します。オブジェクトは、`pdenable` コマンドまたは `pddisable` コマンドで、使用可能になります。

finishings-supported

プリンタでサポートされているドキュメントごとのフィニッシング処理を示します。

fonts-supported

プリンタでサポートされているフォント・リソースを示します。

highlight-colour-rendering-algorithms-supported

プリンタでサポートされている強調色のレンダリング・アルゴリズムを示します。

highlight-colours-supported

プリンタでサポートされている強調色を示します。

有効な色は、`red` (赤)、`blue` (青)、`green` (緑)、`cyan` (シアン)、`magenta` (マゼンタ)、`yellow` (黄色)、`cardinal` (深紅)、`royalblue` (ロイヤルブルー)、`ruby` (ルビー)、`violet` (バイオレット)、および `name` (指定色) です。

highlight-mapping-colours-supported

サポートされる強調マップ・カラーを示します。

input-trays-supported

プリンタでサポートされている給紙トレイを示します。

有効な値は、`top` (上段給紙トレイ)、`middle` (中段給紙トレイ)、`bottom` (下段給紙トレイ)、`envelope` (封筒専用トレイ)、`manual` (手差しトレイ)、`large-capacity` (大容量給紙トレイ)、`main` (標準給紙トレイ)、および `side` (サイド給紙トレイ) です。

job-sheets-supported**maximum-copies-supported**

プリンタでサポートされている補助シート・パッケージを指定します。

media-supported

プリンタでサポートされている用紙を示します。

サポートされている用紙のリストについては、この付録の最後にある表を参照してください。

message

オブジェクトに関連付けられるテキスト文字列です。この文字列は、ユーザに対して、オブジェクトの状態または位置に関する情報を示すためのものです。

notification-profile

このオブジェクトに関して通知されるイベントを指定し、通知方式を示します。

numbers-up-supported

プリンタでサポートされている N アップ (片面に印刷する複数ページ) の数を示します。

有効な値は、none または 0、simple-1-up または 1、simple-2-up または 2、simple-4-up または 4 です。

object-class

特定のオブジェクトのクラスを示します。

output-bins-supported

プリンタでサポートされている排出トレイを示します。

有効な値は、top (上段排出トレイ)、middle (中段排出トレイ)、bottom (下段排出トレイ)、side (サイド排出トレイ)、face-up (おもて面排出トレイ)、left (左側排出トレイ)、right (右側排出トレイ)、および large (大容量排出トレイ) です。

outputs-supported

プリンタでサポートされている出力方式を示します。

有効な値は、no-page-collate です。

page-select-supported

プリンタでサポートされているページ番号付けのタイプを示します。この属性は、アウトバウンド・ゲートウェイにのみ適用されます。

有効な値は、numeric (数字)、および alphabetic (英字) です。

plexes-supported

プリンタでサポートされる両面印刷方法を示します。

有効な値は、simplex (なし)、duplex (長辺とじ)、および tumble (短辺とじ) です。

printer-associated-host

直接接続した物理プリンタのホスト名を示します。

printer-associated-printers

この論理プリンタまたは物理プリンタに関連付けられる、物理プリンタまたは論理プリンタを示します。この属性は、関連するキュー属性が変更されるときに更新されます。この属性は、プリンタが使用可能になると、エンド・ツー・エンドの一貫性についてチェックされます。

printer-name

プリンタを識別する一意の名前です。この属性は、プリンタが作成されるときに設定されます。

printer-problem-message

問題を記述するテキスト文字列を作成できるプリンタについて、スーパーバイザはこの属性にテキスト文字列を設定します。

printer-realization

プリンタが論理プリンタであるか、または物理プリンタであるかを示します。

有効な値は、logical (論理プリンタ)、および physical (物理プリンタ) です。

printer-state

プリンタの現在の状態を示します。

有効な値は、unknown (不明)、idle (アイドル)、printing (印刷中)、needs-attention (操作待ち)、paused (一時停止)、shutdown (シャットダウン)、timed-out (タイムアウト)、および connecting-to-printer (プリンタに接続中) です。

printers-ready

現在、この物理プリンタ/論理プリンタで使用可能になっている、論理プリンタ/物理プリンタを示します。

sides-supported

このプリンタでサポートされている印刷面の数を示します。

有効な値は、1 (片面)、2 (両面) です。

thickening-supported

プリンタでサポートされている陰影指定アルゴリズムを示します。

有効な値は、entire-document (ドキュメント全体)、および bitmap-images (ビットマップ・イメージ) です。

x-image-shift-range-supported

プリンタでサポートされている x 方向へのイメージのシフト範囲を示します。

y-image-shift-range-supported

プリンタでサポートされている y 方向へのイメージのシフト範囲を示します。

A.2.1 論理プリンタ属性

hold-jobs-interrupted-by-printer-failure

再起動後にスプーラに戻されたジョブを、held (保留) 状態にするか、または pending (処理待ち) 状態にするかを指定します。

有効な値は、保留状態にするには true または yes、処理待ち状態にするには false または no です。

printer-initial-value-document

この論理プリンタで使用する、サーバでの初期値ドキュメントを示します。ドキュメントで初期値ドキュメントを指定していない場合には、プリンタの初期値ドキュメントが使用されます。

printer-initial-value-job

この論理プリンタで使用する、サーバでの初期値ジョブを示します。

A.2.2 物理プリンタのみ

LPD-printer-name

アウトバウンド・ゲートウェイでサポートされる LPD プリンタの名前を示すテキスト文字列です。

character-sets-ready

プリンタで使用できる文字セット・エンコードを示します。

document-formats-ready

プリンタおよび関連する翻訳プログラムで処理できるドキュメント・フォーマットを示します。

document-sheets-ready

このプリンタで使用できる補助ドキュメント・シートを指定します。

有効な値は、none、および doc-set-start-copies-separate です。

excluded-filters

特定のプリンタについて、特定の翻訳フィルタまたは変更フィルタの使用を禁止します。

有効な値はフィルタのリストです。

finishings-ready

プリンタでサポートされているドキュメントのフィニッシング処理を示します。

有効な値は、staple (ホチキスどめ)、staple-top-left (ホチキスどめ (左上隅))、staple-bottom-left (ホチキスどめ (左下隅))、staple-top-right (ホチキスどめ (右上隅))、staple-bottom-right (ホチキスどめ (右下隅))、saddle-stitch (中綴じ)、および edge-stitch (端綴じ) です。

fonts-ready

プリンタで使用できるフォント・リソースを示します。

highlight-colours-ready

プリンタで使用できる強調色を示します。

有効な値は、red (赤)、blue (青)、green (緑)、cyan (シアン)、magenta (マゼンタ)、yellow (黄色)、cardinal (深紅)、royalblue (ロイヤルブルー)、ruby (ルビー)、violet (バイオレット)、および name (指定色) です。

input-trays-medium

プリンタのそれぞれの給紙トレイの用紙を示します。

input-trays-ready

このプリンタで使用できる給紙トレイを示します。

有効な値は、top (上段給紙トレイ)、middle (中段給紙トレイ)、bottom (下段給紙トレイ)、envelope (封筒専用トレイ)、manual (手差しトレイ)、

large-capacity (大容量給紙トレイ) , main (標準給紙トレイ) , および side (サイド給紙トレイ) です。

job-sheets-ready

このプリンタで使用できる補助シートを示します。

有効な値は , none , job-copy-start , および job-copy-wrap です。

maximum-printer-speed

プリンタの最大速度を , 1 分あたりのページ数で指定します。

media-ready

現在このプリンタで使用できる用紙を示します。

native-document-formats-ready

プリンタのハードウェアでサポートされてるドキュメント・フォーマットを示します。

有効な値

output-bins-ready

物理プリンタで使用できる排出トレイを示します。

有効な値は , top (上段排出トレイ) , middle (中段排出トレイ) , bottom (下段排出トレイ) , side (サイド排出トレイ) , face-up (おもて面排出トレイ) , left (左側排出トレイ) , right (右側排出トレイ) , および large (大容量排出トレイ) です。

outputs-ready

このプリンタで現在使用できる出力方式を示します。

有効な値は , no-page-collate です。

printer-address

プリンタへの接続方法をスーパーバイザに指示する , スーパーバイザ固有の文字列です。

printer-baud-rate

接続されているプリンタのボー・レートです。

有効な値は , 300 , 1200 , 2400 , 4800 , 9600 , 19200 , および 38400 です。

printer-connection-level

プリンタでサポートされる接続機能のレベルを示します。

有効な値は、0 – 指定なし (システムの省略時の設定を使用)、1 – 出力のみ (単方向)、2 – 出力のみ (状態ビットが返される)、3 – 双方向 (同期セッションの制御なし)、4 – 双方向 (同期セッションの制御あり) です。

printer-connection-method

プリンタのサーバ・ホストへの接続方法を示します。

有効な値は、serial、parallel、ip-socket、data products、および LPD です。

printer-data-bits

接続されたプリンタのデータ・ビットを示します。

有効な値は、5、6、7、および 8 です。

printer-input-flow-control

プリンタで使用する入力フロー制御のタイプを示します。

有効な値は、none、xoff、cts、および dtr です。

printer-locations

プリンタの位置を示すために使用できるテキスト文字列です。

printer-model

プリンタの製造元とモデルを示します。

printer-output-flow-control

プリンタで使用する出力フロー制御のタイプを示します。

有効な値は、none、xoff、cts、および dtr です。

printer-parity

接続されたプリンタで使用されるパリティを示します。

有効な値は、none (なし)、even (偶数パリティ)、odd (奇数パリティ)、mark (マーク・パリティ)、および space (スペース・パリティ) です。

printer-stop-bits

プリンタで使用するストップ・ビットを示します。

有効な値は、0、1、および 2 です。

printer-tcpip-port-number

TCP/IP ソケット接続の場合に、接続されるプリンタが使用するポート番号を示します。

有効な値は、1024 ~ 65535 です。

printer-timeout-period

printer-state を *timed-out* に設定する前に、プリンタからの応答をサーバが待つ時間を、秒数で示します。

sides-ready

プリンタで使用できる印刷面の数を示します。

有効な値は、1 (片面) および 2 (両面) です。

A.3 キュー属性

associated-server

キューが関連付けられているサーバを示します。この属性は、キューが作成されるときに設定されます。

availability

キューの一般的な可用性を示します。キューが使用不能になっている場合には *none*、キューが使用可能になっている場合には *normal* に設定されます。

有効な値は、*high*、*normal*、*low*、*unknown*、および *none* です。

descriptor

キューのテキスト表現です。

enabled

キューが使用可能で、クライアントからのプリント要求を受け付けることができるかどうかを示します。この属性は、*pdenable* コマンドおよび *pddisable* コマンドを使用して設定されます。

有効な値は、*true/yes* (使用可能)、*false/no* (使用不能) です。

logical-printers-ready

キュー上で使用できる論理プリンタを示します。*associated-queue* 属性がこのキューを指している論理プリンタが、使用可能または使用不能になるたびに、この属性は更新されます。

logical-printers-supported

キューでサポートされている論理プリンタを示します。

message

キューの状態をユーザに示すためのテキスト文字列です。

notification-profile

キューに関してユーザに通知するイベントを指定し、その通知方式を指定します。

object-class

オブジェクトのクラスを示します。

physical-printers-ready

関連付けられたプリンタのうち、処理可能 (ready) 状態のプリンタを示します。

physical-printers-supported

キューでサポートされている物理プリンタを示します。

queue-name

キューの名前を示します。

state

キューの状態を示します。

有効な値は、ready (処理可能)、および paused (一時停止) です。

A.4 ジョブ属性

assigned-queue

ジョブが割り当てられたキューを示します。

completion-time

ジョブが印刷を完了した時刻を示します。

current-job-state

ジョブの現在の状態を示します。

有効な値は、unknown (不明)、pre-processing (前処理中)、pending (処理待ち)、processing (処理中)、retained (保持)、held (保留)、paused (一時停止)、terminating (後処理中)、completed (完了)、および printing (印刷中) です。

document-sheets

サーバがジョブに挿入する補助シートを指定します。

有効な値は、none および doc-set-start-copies-separate です。

initial-value-job

ジョブが、属性の省略時の設定に使用する初期値ジョブを識別できるようにします。ジョブの初期値ジョブは、プリンタの初期値ジョブよりも優先されます。

intervening-jobs

ジョブがスケジューリングされる前に印刷されるジョブの数を示します。ジョブが印刷中の場合には、0 に設定されます。

job-comment

ジョブに関連付けられ、セパレータ・ページに印刷されるテキストです。

job-copies

job-copies-completed

印刷されたジョブの部数を示します。

job-discard-time

印刷されたか、されていないかにかかわらず、ジョブを廃棄するカレンダー時刻を指定します。スプーラは、ジョブの状態に関係なく、ジョブをキャンセルまたは削除して、保持期間をゼロに設定します。

job-fault-count

スプーラまたはスーパーバイザのクラッシュにより、ジョブがスーパーバイザに戻った回数を示します。

job-hold

ジョブがスケジューリング可能かどうかを示します。保留が true である場合、ジョブはスケジューリングされず、*job-hold-set* が *job-state-reasons* に追加されて、*current-job-state* が *held* に設定されます。

job-identifier

スプーラによって生成され、ジョブを識別するスプーラ固有の値です。

job-identifier-on-client

以前のシステムから実行されたジョブのための、クライアント・ジョブ識別子です。

job-identifier-on-printer

プリンタによって提供されるジョブ識別子です。

job-message-from-administrator

ジョブに対して実行されたアクションの理由をユーザに示すためのメッセージを提供します。

job-name

開始シート，通知，ロギングに印刷するプリント・ジョブを表す読み取り可能な文字列です。CLI では，ユーザがジョブ名を指定しない場合には，最初のファイルの名前が使用されます。

job-originating-host

この属性は，LPD インバウンド・ゲートウェイによって使用され，ジョブの発信元のホスト名が含まれています。

job-originator**job-owner**

ジョブの所有者の名前を示します。

job-print-after

ジョブをスケジュールリングできるカレンダー日付/時刻を指定します。

job-promote-time

ジョブが昇格された時刻を指定します。

job-retention-period

ジョブが印刷された後，スプーラがジョブを保持する時間を指定します。

job-sheets

スーパーバイザがジョブで印刷する補助シートを指定します。

有効な値は，none，job-copy-start，および job-copy-wrap です。

job-state-message

ジョブの状態に関する情報を提供します。ジョブがエラーで終了するか，打ち切られると，スーパーバイザはこの属性にテキストを設定します。

job-state-reasons

ジョブが保留 (held)，後処理中 (terminating)，保持 (retained)，または完了 (completed) の状態になっている理由を示します。

有効な値は、documents-needed、job-hold-set、job-print-after-specified、required-resource-not-ready、successful-completion、completed-with-warnings、completed-with-errors、cancelled-by-user、cancelled-by-operator、aborted-by-system (cancelled-by-shutdown、printer-unavailable、wrong-printer、bad-job)、logfile-pending、およびlogfile-transferring です。

job-submission-complete

ジョブ内のすべてのドキュメントが実行されたことを示し、ジョブがクローズされたことを示します。

notification-profile

ユーザに通知されるジョブ・イベントを指定し、通知の配信方式を指定します。

number-of-documents

ジョブ内のドキュメントの数を示します。

object-class

クラスをジョブとして指定します。

output-bin

プリント・ジョブが出力されるプリンタ排出トレイを示します。

有効な値は、top (上段排出トレイ)、bottom (下段排出トレイ)、middle (中段排出トレイ)、side (サイド排出トレイ)、face-up (おもて面排出トレイ)、left (左側排出トレイ)、right (右側排出トレイ)、および large (大容量排出トレイ) です。

physical-printers-requested

ジョブの印刷先となる、1 つまたは複数の物理プリンタを示します。

previous-job-state

最後に状態が変更される前の、ジョブの状態を示します。

printer-name-requested

プリント要求に使用される論理プリンタを示します。

printers-assigned

ジョブが割り当てられる物理プリンタを示します。この属性は、ジョブがスーパーバイザに送られると、スプーラによって設定されます。

started-printing-time

ジョブが印刷を開始した時刻を示します。

submission-time

ジョブに対して最後のプリント要求が実行された時刻を示します。この属性は、`pdresubmit` コマンドで変更されることがあります。

total-job-octets

ジョブのサイズをバイト数で示します。

user-name

プリント・サービスへのアクセスを要求しているユーザの名前を示します。

A.5 ドキュメント属性

binding-edge

綴じるシートの端を指定します。

有効な値は、`bottom-edge` (下端)、`right-edge` (右端)、`top-edge` (上端)、および `left-edge` (左端) です。

bottom-margin

指定した印刷方向で見たときの、論理ページの下端とテキスト領域の下端との間隔を、文字数で指定します。

content-orientation

最もよく使用するドキュメントの方向を指定します。

有効な値は、`portrait` (たて)、`landscape` (よこ)、`reverse-portrait` (たて (逆方向))、および `reverse-landscape` (よこ (逆方向)) です。

copy-count

ドキュメントの印刷部数を指定します。

`copy-count` の値が `maximum-copies-supported` の値を超える場合には、ジョブは拒否されます。

default-character-set

コード文字セットの指定を必要とするドキュメントのページの省略時の値として、サーバが使用するコード文字セットを示します。

default-font

他の方法ではフォントを指定しないドキュメント・ページで使用するフォントを示します。

default-input-tray

ドキュメントの印刷が開始される前に、スーパーバイザが省略時の値として設定する給紙トレイを指定します。

この属性が指定されていない場合には、スーパーバイザはプリンタの省略時の値を使用します。

有効な値は、top (上段給紙トレイ)、middle (中段給紙トレイ)、bottom (下段給紙トレイ)、envelope (封筒専用トレイ)、manual (手差しトレイ)、large-capacity (大容量給紙トレイ)、main (標準給紙トレイ)、および side (サイド給紙トレイ) です。

default-medium

ドキュメントで使用する用紙を指定します。

スプーラはこの属性を使用してスケジューリングを実行し、ASCII から PostScript への翻訳プログラムは、この属性を使用して印刷する用紙のサイズを決定します。

document-file-name

ファイルへの完全パスを含めて、ファイルの名前を指定します。

document-format

ドキュメントのプリント・フォーマットを指定します。

document-name

ドキュメントに意味のある名前を指定します。この属性は、クライアントによってファイル名として設定されるか、または、プリント・ジョブが実行されたときに設定することができます。

document-sequence-number

ジョブ内でのドキュメントのシーケンス番号を指定します。

document-state

ドキュメントの状態を示します。

有効な値は、transfer-pending (転送待ち)、pending (処理待ち)、processing (処理中)、completed (完了)、および printing (印刷中) です。

finishing

ドキュメントに適用されるフィニッシング・オブジェクト，またはフィニッシング・プロセスの流れを指定します。

有効な値は，staple (ホチキスどめ)，staple-top-left (ホチキスどめ (左上隅))，staple-bottom-left (ホチキスどめ (左下隅))，staple-top-right (ホチキスどめ (右上隅))，staple-bottom-right (ホチキスどめ (右下隅))，saddle-stitch (中綴じ)，edge-stitch (端綴じ)，punch (パンチ)，および cover-bind (製本) です。

footer-text

各ページのフッタのテキストを指定します。

header-text

各ページのヘッダのテキストを指定します。

highlight-colour

強調印刷色を指定します。

有効な値は，red (赤)，blue (青)，cyan (シアン)，magenta (マゼンタ)，yellow (黄色)，cardinal (深紅)，royalblue (ロイヤルブルー)，ruby (ルビー)，violet (バイオレット)，black (黒)，および name (指定色) です。

highlight-colour-mismatch-action

強調色のミスマッチ・オプションを指定します。

有効な値は，abort，ignore，および operator です。

highlight-colour-rendering-algorithm

強調色のレンダリング・アルゴリズムを指定します。

有効な値は，automatic，colourToHighlight，colourTables，presentation，および pictorial です。

highlight-mapping-colour

参照色となる色を指定します。

有効な値は，red (赤)，blue (青)，cyan (シアン)，magenta (マゼンタ)，yellow (黄色)，cardinal (深紅)，royalblue (ロイヤルブルー)，ruby (ルビー)，violet (バイオレット)，black (黒)，および name (指定色) です。

initial-value-document

属性の省略値の設定に使用する初期値ドキュメントを指定します。ドキュメントの initial-value-document は、プリンタの initial-value-document よりも優先されます。

left-margin

指定した印刷方向で見たときの、論理ページの左端とテキスト領域の左端との間隔を、文字数で指定します。

length

テキスト領域の長さを文字数で指定します。

modification-filter

変換フィルタリングの前に、ドキュメント・データに適用される変更フィルタの名前を指定します。

no-filtering

翻訳フィルタリングまたは変更フィルタリングを使用不能にします。

number-pages

ページ番号付けをするかどうかを示します。

number-up

片面に印刷するページ・イメージの数を指定します。

有効な値は、none または 0、simple-1-up または 1、simple-2-up または 2、simple-4-up または 4 です。

object-class

オブジェクトのクラスを示します。

有効な値は、job です。

octet-count

ドキュメントのサイズをオクテット数で指定します。

output

ドキュメントが印刷される用紙の出力処理を示します。

有効な値は、no-page-collate です。

page-media-select

指定されたページを、指定された用紙に印刷する必要があることを示します。この属性で指定されていないページは、省略時の用紙に印刷されます。

この属性は、特定のプリント・ハードウェアでのみサポートされていることに注意してください。

page-order-received

ドキュメントのページがフォーマットされるページ順序を指定します。

有効な値は、unknown、first-to-last、last-to-first です。

page-select

印刷する 1 ページ、または一続きのページを指定します。

plex

ページ・イメージを片面印刷、または両面印刷に対応するように設定するかどうかを指定します。

有効な値は、simplex (片面)、duplex (両面 (長辺とじ))、および tumble (両面 (短辺とじ)) です。

repeated-tab-stops

タブ・ストップ間の文字数を指定します。

reset-printer

ドキュメントを印刷した後、プリンタをリセットするかどうかを指定します。

有効な値は、true/yes、および false/no です。

right-margin

指定した印刷方向で見たときの、論理ページの右端とテキスト領域の右端との間隔を、文字数で指定します。

sides

ジョブを用紙の片面に印刷するか、または両面に印刷するかを指定します。

thickening-specification

ドキュメント全体またはドキュメント内のビットマップ・イメージの陰影パラメータを設定します。

有効な値は、entire-document (ドキュメント全体)、および bitmap-images (ビットマップ・イメージ) です。

top-margin

論理ページの上端とテキスト領域との間隔を、文字数で指定します。

transfer-method

ドキュメントをサーバに転送する方法を示します。

有効な値は、with-request、socket、および file-reference です。

translation-filter

翻訳フィルタの自動呼び出しを指定変更します。

width

テキスト領域の幅を文字数で指定します。

x-image-shift

ページ・イメージがレンダリングされる用紙に対して、ページ・イメージの位置をシフトするように設定します。シフトする方向は、イメージの X 軸方向です。

y-image-shift

ページ・イメージがレンダリングされる用紙に対して、ページ・イメージの位置をシフトするように設定します。シフトする方向は、イメージの Y 軸方向です。

プリンタ・データ・シート

この付録には、プリント・システムのスーパーバイザによって使用可能にできる、プリンタ機種ごとのデータ・シートが含まれています。日本語環境では、次のプリンタについて、データ・シートを利用することができます。なお、英語環境で使用可能なプリンタについては、英語版 Advanced Printing Software の『System Administration and Operation Guide』を参照してください。

Digital PrintServer 17
Digital PrintServer 20
Digital PrintServer 32

表 B-1: プリンタ・データ・シート: Digital PrintServer 20 Level 2 PS

プリンタ属性ファイル名:

Digital_PrintServer20_Level2PS.paf

プリンタ・モデル:

Digital PrintServer 20

maximum-printer-speed	connection-method	connection-level
13 ppm	ip-socket	2
printer-tcpip-port-number	parallel	2
	serial	4
10001		

xxx-supported-attributes

document-formats-supported = PostScript simple-text
input-trays-supported = top bottom [large capacity] [envelope]
output-bins-supported = top
sides-supported = 1, 2

xxx-ready-attributes

document-formats-ready = PostScript simple-text
native-document-formats-ready = PostScript
input-trays-ready = top bottom
output-bins-ready = top
sides-ready = 1, 2

インプリメンテーションに関する注意事項

汎用給紙トレイが上段トレイです。
コントロール・パネルを使用してネットワーク・カードを構成してください。
角かっこで囲まれた属性値がある場合、これらはプリンタ・オプションであり、プリンタ属性ファイルには含まれません。オプションがインストールされている場合には、pdset コマンドまたは pdcreate コマンドを使用して、値を属性に追加してください。
例: pdset -x "input-trays-supported+=large-capacity" ppname

表 B-2: プリンタ・データ・シート: Digital PrintServer 32 Level 2 PS

プリンタ属性ファイル名:

Digital_PrintServer32_Level2PS.paf

プリンタ・モデル:

Digital Printserver 32

maximum-printer-speed

32 ppm

xxx-supported-attributes

document-formats-supported = PostScript simple-text

input-trays-supported = top middle large-capacity

output-bins-supported = top side

sides-supported = 1, 2

xxx-ready-attributes

document-formats-ready = PostScript simple-text

native-document-formats-ready = PostScript

input-trays-ready = top middle large-capacity

output-bins-ready= top side

sides-ready= 1, 2

インプリメンテーションに関する注意事項

角かっこで囲まれた属性値がある場合、これらはプリンタ・オプションであり、プリンタ属性ファイルには含まれません。オプションがインストールされている場合には、pdset コマンドまたは pdcreate コマンドを使用して、値を属性に追加してください。

例: pdset -x "input-trays-supported+=large-capacity" ppname

表 B-3: プリンタ・データ・シート: DIGITAL PrintServer LPS17 Level 2 PS

プリンタ属性ファイル名:

Digital_PrintServer17_Level2PS.paf

プリンタ・モデル:

Digital PrintServer 17 Level2 PS

maximum-printer-speed	connection-method	connection-level
17 ppm	digital-printserver	5
printer-tcpip-port-number		
n/a		

xxx-supported-attributes

document-formats-supported = PostScript simple-text

input-trays-supported = top bottom

output-bins-supported = top

side

sides-supported = 1 [2]

xxx-ready-attributes

document-formats-ready = PostScript simple-text

native-document-formats-ready = PostScript

input-trays-ready = top bottom

output-bins-ready= top side

sides-ready= 1 [2]

インプリメンテーションに関する注意事項

共通プリンタ・アクセス・プロトコル (Common Printer Access Protocol) です。角かっこで囲まれた属性値がある場合、これらはプリンタ・オプションであり、プリンタ属性ファイルには含まれません。オプションがインストールされている場合には、pdset コマンドまたは pdcreate コマンドを使用して、値を属性に追加してください。

例: pdset -x "input-trays-supported+=large-capacity" ppname

ONC バインディング・エントリ

C.1 ONC バインディング・エントリ

ローカル・ファイルおよび NIS ネーミング・サービスでは、ONC バインディング・エントリ形式のエントリを使用して、プリント・オブジェクトを識別します。プリント・システムは、このようなエントリを `/etc/printers.conf` ファイルに格納したり、このファイルから検索したりします。

ONC バインディング・エントリ形式の構文は、`/etc/printcap` や `/etc/remote` ファイルで使用される構文と似ていますが、ONC バインディング・エントリ形式では異なったキー名を使用する点が異なります。

`printer-configuration` の構文は、次のとおりです。

```
ONCEntry ::= NameList { : KeyValuePair } +
NameList ::= PrimaryName { ' | ' Alias } +
KeyValuePair ::= Name = { ValueDP | ... }
ValueDP ::= Hostname, ProgramNumber, ProgramVersion,
Authentication, SubType, ServerName [ , Protoserver ], spooling-type
```

この構文の記号の意味は、次のとおりです。

```
{ ... } + = 1 つ以上
|      = または
|      = シンボル |
[ ... ] = オプション
```

また、パラメータの意味は、次の表のとおりです。

パラメータ	説明
ONCEntry	1 つの PrimaryName、オプションの Aliases、および 1 つまたは複数の KeyValuePair からなる、プリンタ・テーブル・エントリ。
PrimaryName of Object	オブジェクトを識別するためのプライマリ名。
Aliases	オブジェクトの別名で、ゼロでも複数あってもよい(オプション)。

KeyValuePair	属性名およびその値からなる属性。ここでは、タイプ ValueDP の値だけが表示される。
ValueDP	プリンタ・アドレス・バインディング (paddr), サーバ・アドレス・バインディング (saddr), キュー・アドレス・バインディング (qaddr) に対応するサーバ・バインディング。
Hostname	オブジェクトおよびそれが関連付けられているサーバを持つホスト・コンピュータの名前。
ProgramNumber	サーバまたは ProtoServer の RPC プログラム番号。サーバ・プログラム番号は、サーバが動的に作成する場合も、スプーラに対して特別に作成される場合もある。
ProgramVersion	Protoserver の RPC バージョン番号。
Authentication	使用される RPC 認証方式。プリント・システムでは、sys だけがサポートされている。
SubType	オブジェクトについての説明。lp は論理プリンタ・オブジェクト, pp は物理プリンタ・オブジェクト, qu はキュー・オブジェクト, sl はスプーラ・サーバ・オブジェクト, sv はスーパーバイザ・サーバ・オブジェクト。
ServerName	オブジェクトに関連付けられているサーバの名前。
ProtoServer	オプション。パラメータの値が 1 の場合は、ProtoServer が使用される。パラメータの値が 0 の場合は、ProtoServer は使用されない。ローカル・ファイルおよび NIS ネーミング・サービスの場合、このフィールドの値は 1 でなければならない。
Spooling-type	インバウンド・ゲートウェイ翻訳が行われるプリント・パラダイムの名前。プリント・システムでは、dpa だけがサポートされている。

プリンタの構成例

```
elmtree:paddr=papers,105004,1,sys,lp,wooden,1,dpa
```

この例のそれぞれのフィールドの意味は、次のとおりです。

- elmtree はオブジェクトの名前で、プリンタ・アドレス・バインディング情報が含まれています。
- プリンタ・アドレス・バインディング情報には、RPC バインディング情報の他に、papers の RPC ホスト名、プログラム番号 105004 (Protoserver RPC 番号)、およびプログラムのバージョン番号 1 が含まれます。
- elmtree オブジェクトは、sys の認証方式を持ちます。

- elmtree は、wooden という名前の関連付けられたサーバ・オブジェクトを持つ、論理プリンタ・オブジェクトです。
- Protoserver デーモン (UNIX ホスト・コンピュータ 'papers' 上にある) により、スーパー・サーバ (wooden) に接続するために使用される適切な RPC 動的プログラム番号 (Dynamic Program Number) が決定されます。
- dpa は、LPD インバウンド・ゲートウェイ・ジョブのプリント・パラダイムです。

2 つの論理プリンタおよび 1 つのキューを hostC 上に持つスーパーと、2 つの物理プリンタを hostD 上に持つスーパーバイザからなる構成の場合、完全な構成ファイルは、次のようになります。

```
spooler1:saddr=hostC,105004,1,sys,s1,spooler1,1,dpa
superv1:saddr=hostD,105004,1,sys,sv,superv1,1,dpa
lp1:paddr=hostC,105004,1,sys,lp,spooler1,1,dpa
lp2:paddr=hostC,105004,1,sys,lp,spooler1,1,dpa
ql:qaddr=hostC,105004,1,sys,qu,spooler1,1,dpa
pp1:paddr=hostD,105004,1,sys,pp,superv1,1,dpa
pp2:paddr=hostD,105004,1,sys,pp,superv1,1,dpa
```

この構成ファイルは、ローカル・ファイル・ネーミング環境にあるすべてのホストに存在し、NIS 環境の NIS データベースを更新するために使用する必要があります。ネーミング・サービスの構成については、第 3 章を参照してください。

ただし、ローカル・ファイル環境では、プリント・システムは ONC のネーミング・コンポーネントを使用せず、ONC RPC に依存していることに注意してください。

日本語出力のための設定

この付録では、日本語ファイルを出力するための方法と、サポートされる日本語プリンタで指定できるオプションについて説明します。

D.1 日本語ファイル出力のための設定

Advanced Printing Software で、日本語ファイルを出力する方法には、次の 2 つの方法があります。

D.1.1 スーパーバイザ (pdspvr) を使用する方法

プリンタがパラレル・ポートやシリアル・ポートに接続されている場合、次の手順で設定を行いません。

1. スーパーバイザおよびスプーラを作成します。

`/usr/pd/scripts/` ディレクトリにある `cresupspl_sup.sh` シェル・スクリプトを使用してスーパーバイザとスプーラを作成します。

2. 物理プリンタ、論理プリンタ、およびキューを作成します。

`pdprintadmin` (GUI) あるいは `pdcreate` (CLI) を使用して、物理プリンタ、論理プリンタ、およびキューを作成します。`pdprintadmin` (GUI) を利用すると、以下のように、物理プリンタ、論理プリンタ、キューを簡単に作成することができます。

- a. コマンド行より `pdprintadmin` を実行します。
 - b. GUI が表示されたら、物理デバイス・タブをクリックします。
 - c. 物理デバイス・タブが一番手前側に表示されている状態で、画面上部のプルダウン/メニューでデバイスの追加をクリックします。
 - d. ダイアログのメッセージに従って必要な情報を入力します。
3. 日本語トランスレーション・フィルタを設定します。

次のシェル・スクリプトを使用して日本語トランスレーション・フィルタを設定します。

- `/usr/pd/scripts/setfilter_escp_euc.sh -- EUC ESC/P 用
フィルタ`
- `/usr/pd/scripts/setfilter_escp_sjis.sh -- SJIS ESC/P 用
フィルタ`
- `/usr/pd/scripts/setfilter_ps_euc.sh -- EUC PostScript 用
フィルタ`
- `/usr/pd/scripts/setfilter_ps_euc.sh -- SJIS PostScript 用
フィルタ`

次のシェル・スクリプトを利用することにより、設定されているフィルタを調べることができます。

```
/usr/pd/scripts/check_filter.sh
```

D.1.2 Lpd Outbound Gateway スーパーバイザ (pdspvlpr) を使用する 場合 (lpd 印刷)

プリンタが TCP/IP ネットワーク経由で接続されていて lpd ポートをサポートしている場合、次の手順で設定を行います。

ネットワーク・インタフェースを備えているほとんどの日本語プリンタは lpd ポートをサポートしていますが、日本語プリンタで Raw TCP ポートをサポートしているものはあまりありません。プリンタがサポートしているポートの詳細については、それぞれのプリンタのマニュアルを参照してください。

1. Lpd Outbound Gateway スーパーバイザ、およびスプーラを作成します。
シェルスクリプト `/usr/pd/scripts/cresupspl_obg.sh` を使用して Lpd Outbound Gateway スーパーバイザ、およびスプーラを作成します。
2. 物理プリンタ、論理プリンタ、およびキューを作成します。
`pdprintadmin` (GUI) あるいは `pdcreate` (CLI) を使用して、物理プリンタ、論理プリンタ、およびキューを作成します。物理プリンタの作成時には、プリンタ内部の lpd プリンタ名が必要となります。

例) **lp (OKI MICROLINE 903PS3 の場合)**

lpd プリンタ名についてはそれぞれプリンタのマニュアルを参照してください。

pdprintadmin (GUI) を利用すると、以下のように、物理プリンタ、論理プリンタ、およびキューを簡単に作成することができます。

- a. コマンド行より pdprintadmin を実行します。
- b. GUI が表示されたら、物理デバイス・タブをクリックします。
- c. 物理デバイス・タブが一番手前側に表示されている状態で、画面上部のプルダウン・メニューでデバイスの追加をクリックします。
- d. ダイアログのメッセージに従って必要な情報を入力します。

Lpd Outbound Gateway スーパーバイザ (pdspvlpr) を使用して印刷を行うためには、pdpspr というフィルタ・コマンドを pdpr と組み合わせて使用します。

```
% pdpspr kanji.txt | pdpr
```

pdpspr, pdpr コマンドの詳しい使用方法については、それぞれのコマンドのリファレンス・ページを参照してください。

pdprint (GUI) では、Lpd Outbound Gateway スーパーバイザ (pdspvlpr) を利用した日本語プリンタへの印刷ができない場合があります。その場合は pdpspr と pdpr を使用してください。

pdpspr フィルタとは別に、日本語ファイルを印刷するための機能として jpdpr というコマンドも用意されています。jpdpr コマンドでは、実際には pdpspr フィルタと pdpr コマンドを呼び出して、印刷を実現する方法をとっています。

jpdpr コマンドの詳しい使用方法については、リファレンス・ページを参照してください。

D.2 プリンタの制御

日本語 Advanced Printing Software では、出力部数、印刷方向などの印刷オプションを指定することができます。どのようなオプションを指定できるかについては、プリンタの仕様やプリンタの接続方法によって異なります。

パラレル・ポート/シリアル・ポート/rawTCP 接続でスーパーバイザを使用する場合

プリンタがパラレル・ポートやシリアル・ポートに接続されていて、スーパーバイザ (pdspvr) を使用している場合の、各サポート・プリンタにおける CLI および GUI での指定可能オプションを表 D-1 および表 D-2 に示します。

表 D-1: プリンタの制御 (CLI 使用時)

プリンタ機種名	出力部数	用紙サイズ	印刷方向	片面両面
日本語 DIGITAL PrintServer17				
EPSON LP-9200 PS3				
EPSON LP-8400 PS3				
OKI MICROLINE 701N3				NA ^c
OKI MICROLINE 703N3				NA ^c
OKI MICROLINE 900PS3LT				NA ^c
OKI MICROLINE 903PS3				NA ^c
RICOH IPSiO NX710				NA ^c
RICOH IPSiO NX800 ^a				
RICOH IPSiO NX1100 ^a				
FUJI XEROX DocuPrint201PS ^b	×			NA ^c
EPSON LP-9200 SX	NA ^c		×	×
OKI MICROLINE 8370 SE	NA ^c		NA ^c	NA ^c
OKI MICROLINE 8580 SE	NA ^c		NA ^c	NA ^c

^aRICOH IPSiO NX800, RICOH IPSiO NX1100 においては出力トレイ制御も可能です。

^brawTCP 接続機

^cハードウェアが未対応のものです。

表 D-2: プリンタの制御 (GUI 使用時)

プリンタ機種名	出力部数	印刷方向	片面両面
日本語 DIGITAL PrintServer17			
EPSON LP-9200 PS3			×
EPSON LP-8400 PS3			×

D-4 日本語出力のための設定

表 D-2: プリンタの制御 (GUI 使用時) (続き)

プリンタ機種名	出力部数	印刷方向	片面両面
OKI MICROLINE 701N3			NA ^a
OKI MICROLINE 703N3			NA ^a
OKI MICROLINE 900PS3LT			NA ^a
OKI MICROLINE 903PS3			NA ^a
RICOH IPSiO NX710			NA ^a
RICOH IPSiO NX800			×
RICOH IPSiO NX1100			
FUJI XEROX DocuPrint201PS ^b			NA ^a
EPSON LP-9200 SX		×	×
OKI MICROLINE 8370 SE		NA ^a	NA ^a
OKI MICROLINE 8580 SE		NA ^a	NA ^a

^aハードウェアが未対応のものです。

^brawTCP 接続機

TCP/IP ネットワーク接続で Lpd Outbound Gateway スーパバイザを使用する場合

プリンタが TCP/IP ネットワーク経由で接続されていて Lpd Outbound Gateway スーパバイザ (pdspvlpr) を使用している場合、次の印刷オプションを指定することができます。Lpd Outbound Gateway を利用して印刷する場合、Advanced Printing Software の構造上、日本語 GUI での印刷はできません。

表 D-3: プリンタの制御 (CLI 使用時)

プリンタ機種名	出力部数	用紙サイズ	印刷方向	片面両面
日本語 DIGITAL PrintServer17				
EPSON LP-9200 PS3				
EPSON LP-8400 PS3				
OKI MICROLINE 701N3				NA ^a
OKI MICROLINE 703N3				NA ^a

表 D-3: プリンタの制御 (CLI 使用時) (続き)

プリンタ機種名	出力部数	用紙サイズ	印刷方向	片面両面
OKI MICROLINE 900PS3LT				NA ^a
OKI MICROLINE 903PS3				NA ^a
RICOH IPSiO NX710				NA ^a
RICOH IPSiO NX800				
RICOH IPSiO NX1100				
EPSON LP-9200 SX	×	×	×	×
OKI MICROLINE 8370 SE	×	NA ^a	NA ^a	NA ^a
OKI MICROLINE 8580 SE	×	NA ^a	NA ^a	NA ^a

^aハードウェアが未対応のものです。

この用語集には、マニュアルで使用する用語、頭字語、および略語の定義を示します。

定義

DPA プロトコル (DPA protocol)

クライアント / サーバ間の情報交換で使用される、ISO DPA 標準の通信方式です。

ISO DPA

ISO (国際標準化機構) の Document Printing Application Standard 10175 で定義されているドキュメント印刷標準です。

LPD クライアント (LPD Client)

LPD プロトコルによって通信を行うクライアントです。

LPD プロトコル (LPD protocol)

パークレー・バージョンの UNIX オペレーティング・システムが使用する、プリント・システム用のライン・プリンタ・デーモン・プロトコルです。RFC 1179 に記述されています。

POSIX

移植性の高いオペレーティング・システム・インタフェースの標準化を目的とした IEEE 標準群です。本書では、POSIX は印刷についてのオペレーティング・システム・インタフェース標準を指します。

アクセス制御リスト (ACL: Access Control List)

一定のエンティティ (ユーザ, オペレータ, 管理者およびサーバなど) が、操作や機能を実行することを許可されているかどうかを表すエンティティのリストです。

オブジェクト (Object)

名前付きの物理エンティティまたは論理エンティティについての、情報提供と記述のために使用される属性の集合です。プリント・システム分散システムでは、オブジェクトは ISO DPA 標準に準拠します。

オペレータ (Operator)

POSIX および X/OPEN で定義されるユーザのクラスの1つで、セキュリティによって制御されます。オペレータは、ジョブおよびプリンタの一時停止や再開、およびプリンタの使用可能/使用不能設定の切り換えのような日常的な操作を行うことができます。

管理者 (Administrator)

セキュリティによって制御されるユーザのクラスの1つです。管理者はポリシーを設定し、プリンタやサーバのような永続的なプリント・システム・オブジェクトの作成、変更、および削除を行うことができます。オペレータおよびエンド・ユーザも参照してください。

キュー (Queue)

論理プリンタと物理プリンタを区分化するスプーラ・オブジェクトです。通常、1つのプリンタは単一のキューにのみ存在するので、キューは複数のプリンタを別個のセットに区分化するものと言えます。省略時のプリント・システム・スプーラ構成は、単一キュー構成です。1つのキューの中では、1つの論理プリンタを、そのキューの1つ以上の物理プリンタに関連付けることができます。省略時には、すべての物理プリンタに関連付けられます。

加えて、キューはそのキューの論理プリンタに対して実行されたジョブに順序を付けます。このように付けられた順序を選択順序といいます。この順序付けのためのアルゴリズムは構成可能です (FIFO, 最短ジョブを優先)。

許可 (Authorization)

(認証された、または認証されていない) エンティティが特定の操作を行えるかどうかを、アクセス・ポリシーから判定するプロセスです。

サーバ (Server)

同一のコンピュータ上で、またはネットワーク経由で、1つ以上のクライアントに代わって、印刷機能を管理するソフトウェア・コンポーネントです。プリント・システムにおいて、この用語は、通常、スプーラまたはスーパーバイザのいずれか一方のプリント・サーバを指し、両方を指すことはありません。正確には、スプーラは、それに直接アクセスする多様な API, CLI, および GUI クライアントに対するプリント・サーバです。ただし、スーパーバイザがスプーラに対するプリント・サーバである場合は、スプーラがスーパーバイザのクライアントにもなります。

出力デバイス (Output device)

プリンタです。イメージやドキュメントをレンダリングする機能を持つ物理デバイスまたはハードウェアであり、マーキング・エンジンです。同義語に、「物理デバイス」があります。

スプーラ (Spooler)

クラス・サーバの ISO DPA および POSIX オブジェクト。サーバ・オブジェクトを特殊化したものです。プリント・クライアントからプリント・システム操作を受け取って、適切なプリンタ・スーパーバイザでプリント・ジョブをスケジューリングする名前付きサーバです。一般的なスプーラはマルチスレッドで、複数の印刷操作を同時に処理する機能を持ちます。

スーパーバイザ (Supervisor)

クラス・サーバの ISO DPA および POSIX オブジェクト。サーバ・オブジェクトを特殊化したものです。プリント・ジョブの印刷用に特定のプリンタを制御する静的プロセス、またはソフトウェアです。スーパーバイザは、ジョブ要求の解釈について全面的な責任を負います。これには、ファイル内容の解釈、およびプリント要求パラメータによって提供される指示とファイルの内容との調整が含まれます。スーパーバイザは、フォント、フォーム、その他の電子的リソースのプリンタへのロード、およびファイルのプリンタへの転送をはじめとして、ジョブの印刷を完全に制御します。

セキュリティ・サービス (Security service)

サーバによってネットワーク環境全体に提供されるサービスで、他のネットワーク・サービスに対するクライアント・アクセスの制御に使用します。アクセスは認証と許可の2段階のプロセスによって行われます。認証とはクライアントの識別情報を保証することであり、許可とはクライアントがサービスにアクセスするための許可を持っていることを確認することです。

属性 (Attribute)

オブジェクトのプロパティまたは特性です。特定のオブジェクトに関連する情報項目で、タイプ指示子と1つ以上の値からなります。

通知 (Notification)

サーバにサービスを要求したクライアントに対し、サーバから返される結果およびエラー・レポートです。一般に、通知を実現するために必要なメカニズムは、システム・インフラストラクチャによって提供されます。ほとんどのシステム環境では、通知のための簡便な手段として電子メールを使用することができます。

認証 (Authentication)

要求を行ったユーザを、ユーザのセキュリティ証明 (credential) のチェックと認定により識別して、確認するプロセスです。

ネーム・サービス (Name service)

サーバによって、1つのネットワーク環境中に提供される機能のセットです。これにより、クライアント・アプリケーションはユーザ定義またはシステム定義のオブジェクトにアクセスできます。

物理デバイス (Physical device)

出力デバイスと同義です。

物理プリンタ (Physical printer)

クラス・プリンタの ISO DPA および POSIX ソフトウェア・オブジェクト。レーザー・プリンタ・ハードウェアやレンダリング・システムのような、出力デバイスを表す名前付きプリンタ・オブジェクトです。物理プリンタと出力デバイスの両方を同時に意味することもあります。

ユーザ (User)

POSIX および X/OPEN で定義されるユーザのクラスの1つで、セキュリティによって制御されます。プリント・ジョブを実行し、ステータスを取得する (特権のない) ユーザです。

ローカライズ (Localization)

ソフトウェアを特定の言語または地域に適合させるプロセスです。ローカライズでは、国際化される製品を現地の必要条件に合わせるために、必要な翻訳を行い、フォーマットを定義し、テーブルやその他のデータを提供します。

ロケール (Locale)

ネイティブ言語環境の明示的なモデルと定義です。プログラムのロケールは、そのコードセット、日付/時刻フォーマット、通貨フォーマット、小数点フォーマット、および照合順序を定義します。

論理プリンタ (Logical printer)

1つ以上の物理プリンタ・オブジェクトをグループ化するために使用する、クラス・プリンタの名前付きオブジェクトです。論理プリンタは、負荷分散のために、または物理プリンタ・オブジェクトによって指定される属性とは別の省略時の属性を指定するために使用します。

A

ACL

作成, 4-5

説明, 1-8

apx_caa_setup, 5-11

B

BSD アウトバウンド・ゲートウェイ・

スーパーバイザ

説明, 1-10

E

/etc/printcap, 10-2

L

LDAP

構成, 3-6

**Lightweight directory access
protocol**

(LDAP を参照)

LPD インバウンド・ゲートウェイ

/etc/printcap ファイル, 10-2

構成, 10-1

LPD インバウンド・ゲートウェイ・

マッピング

DEC プロトコル拡張, 10-7t

Sun プロトコル拡張, 10-6

Xerox DocuPrint 拡張, 10-14t

Xerox DocuSP プロトコル拡張,
10-11t

Xerox プロトコル拡張, 10-8t

ジョブとファイルのメッセージ,
10-3

プリント・メッセージ, 10-5

N

NIS

設定, 3-3

P

PAF

説明, 1-12

pdclean

キューをクリーン処理する, 6-2

サーバをクリーン処理する, 5-20

pdconntf

説明, 1-12

pdcreate

キューの作成, 6-1

物理プリンタの作成, 6-8

論理プリンタの作成, 6-14

pddelete

サーバからジョブを消去する, 5-20

- サーバの削除, 5-24
 - 物理プリンタ, 6-12
 - 論理プリンタの削除, 6-18
 - pddisable**
 - サーバを使用不能にする, 5-19
 - 物理プリンタを使用不能にする, 6-11
 - 論理プリンタ, 6-15
 - pdenable**
 - キューを使用可能にする, 6-2
 - サーバを使用可能にする, 5-19
 - 物理プリンタ, 6-10
 - 論理プリンタ, 6-15
 - pdl**
 - 論理プリンタ属性のリスト, 6-16
 - pdmakedb**, 5-1
 - 例, 5-3
 - pdmolddb**, 5-4
 - pdntfs**, 9-1
 - 説明, 1-12
 - pdpause**
 - キューの一時停止, 6-2
 - サーバの一時停止, 5-21
 - ジョブの一時停止, 7-5
 - 物理プリンタの一時停止, 6-11
 - pdpr**
 - ジョブの印刷, 7-2
 - pdpromote**, 7-5
 - pdq**
 - 論理プリンタ上のジョブの表示, 6-16
 - pdresubmit**, 7-3
 - pdresume**
 - キューの再開, 6-2
 - サーバの再開, 5-22
 - ジョブの再開, 7-6
 - 物理プリンタの再開, 6-12
 - pdset**
 - ACL の作成, 4-5
 - サーバ属性の設定, 5-7
 - データ・フィルタの定義, 8-3
 - pdshowdb**, 5-3
 - pdshutdown**, 5-23
 - pdsplr**, 5-16
 - pdspvr**, 5-17
 - printers.conf**
 - NIS ネーム・サービス, 3-3
 - ローカル・ファイル・ネーム・サービス, 3-1
-
- ## R
-
- rcp.pts**
 - 説明, 1-12
 - ready** 属性
 - 説明, 1-6
-
- ## S
-
- supported** 属性
 - 説明, 1-6
-
- ## T
-
- TruCluster**
 - における構成, 3-8, 5-8
-
- ## X
-
- xxx-ready** 属性, 1-6
 - xxx-supported** 属性, 1-6

あ

アウトバウンド・ゲートウェイ
説明, 11-3
アウトバウンド・ゲートウェイ・マッ
ピング
DIGITAL プロトコル拡張, 11-5t
Sun プロトコル拡張, 11-4t
Xerox DocuSP プロトコル拡張,
11-8t
Xerox プロトコル拡張, 11-5t
ジョブとファイル・メッセージ,
11-4t
アクセス制御
説明, 1-8
アクセス制御リスト
説明, 1-8

い

イベント通知
イベント・クラス, 9-2
キュー・イベント, 9-3
サーバ・イベント, 9-2
ジョブ・イベント, 9-4
説明, 1-9
通知サーバ, 9-1
通知方法, 9-2
プリンタ・イベント, 9-4

き

キュー
キュー属性のリスト, 6-3

作成, 6-1
使用可能にする, 6-2
状態, 6-2
ジョブの削除, 6-2
説明, 1-5

こ

高可用性アプリケーション
構成, 3-8, 5-8
リソース・プロファイル, 3-9, 5-8
コマンド行クライアント・プログラム
説明, 1-10
コンソール通知デーモン
説明, 1-12

さ

サーバ
一時停止, 5-21
起動, 5-16
削除, 5-24
作成, 5-1
サーバ属性の設定, 5-7
サーバの再開, 5-22
サーバの状態, 5-15
シャットダウン, 5-23
使用可能にする, 5-19
使用不能にする, 5-19
サーバ・データベース
作成, 5-1
フィールド値の表示, 5-3
フィールド値の変更, 5-4
サーバ・デーモン

説明, 1-12
サーバ・プログラム
説明, 1-10

し

初期値ジョブ
説明, 1-6
初期値ドキュメント
説明, 1-6
ジョブ
一時停止, 7-5
印刷, 7-2
印刷時刻の指定, 7-2
昇格, 7-5
ジョブの再開, 7-6
ジョブの再実行, 7-3
ジョブの破棄, 7-8
ジョブの保持, 7-7
ジョブの保留, 7-6
説明, 1-5
ジョブ属性
job-discard-time, 7-8
job-hold, 7-6
job-retention-period, 7-7

す

スクリプト
apx_caa_setup, 5-11
スプーラ
起動, 5-16
作成, 5-1
使用可能にする, 5-19
使用不能にする, 5-19
説明, 1-10

スーパーバイザ
起動, 5-17
作成, 5-1
使用可能にする, 5-19
使用不能にする, 5-19
説明, 1-10
フィルタの呼び出し, 8-7
スーパーバイザ属性
filter-definition, 8-2, 8-4

つ

通知サーバ
説明, 1-12

と

ドキュメント
説明, 1-5
ドキュメント属性
no-filtering, 8-4
translation-filter, 8-4
ドキュメント・データ・フィルタ
excluded—filters 属性, 8-4
filter-definition 属性, 8-2
modification-filter 属性, 8-4
no-filtering 属性, 8-4
translation-filter 属性, 8-4
trn_textps, 8-9
trn_textps_j, 8-9
エラー処理, 8-8
作成, 8-8
説明, 8-1
フィルタの呼び出し, 8-7
特権
説明, 1-8

ね

Name services

- LDAP, 3-6
- 構成, 3-1

ふ

- 物理プリンタ, 6-8
 - 一時停止, 6-11
 - オプション属性の設定, 6-9
 - 再開, 6-12
 - 削除, 6-12
 - 作成, 6-4
 - 作成例, 6-8
 - 使用可能, 6-10
 - 使用不能, 6-11
 - シリアル・ポート属性, 6-7
 - 説明, 1-5
 - プリンタ・アドレスの設定, 6-4
 - プリンタ接続方法, 6-5
 - プリンタ接続レベル, 6-6
- プリンタ属性
 - modification-filter, 8-4
- プリンタ属性ファイル
 - 説明, 1-12
- プロトサーバ
 - 説明, 1-12

へ

- 変更フィルタ, 8-2
 - 作成, 8-8

- 説明, 1-13
- 定義, 8-3

ほ

- 翻訳フィルタ, 8-1
 - trn_asciips, 1-13
 - trn_textps, 8-9
 - trn_textps_j, 8-9
 - 作成, 8-8
 - 説明, 1-13
 - 定義, 8-3

り

- リソース・プロファイル, 3-9, 5-8

ろ

- 論理プリンタ
 - 削除, 6-18
 - 作成, 6-13
 - 実行されたジョブのリスト, 6-16
 - 使用可能にする, 6-15
 - 使用不能にする, 6-15
 - 説明, 1-5
 - 属性のリスト, 6-16
- ローカル・ファイル・ネーム・サービス
 - printers.conf, 3-1
 - 構成, 3-1
 - 説明, 3-1