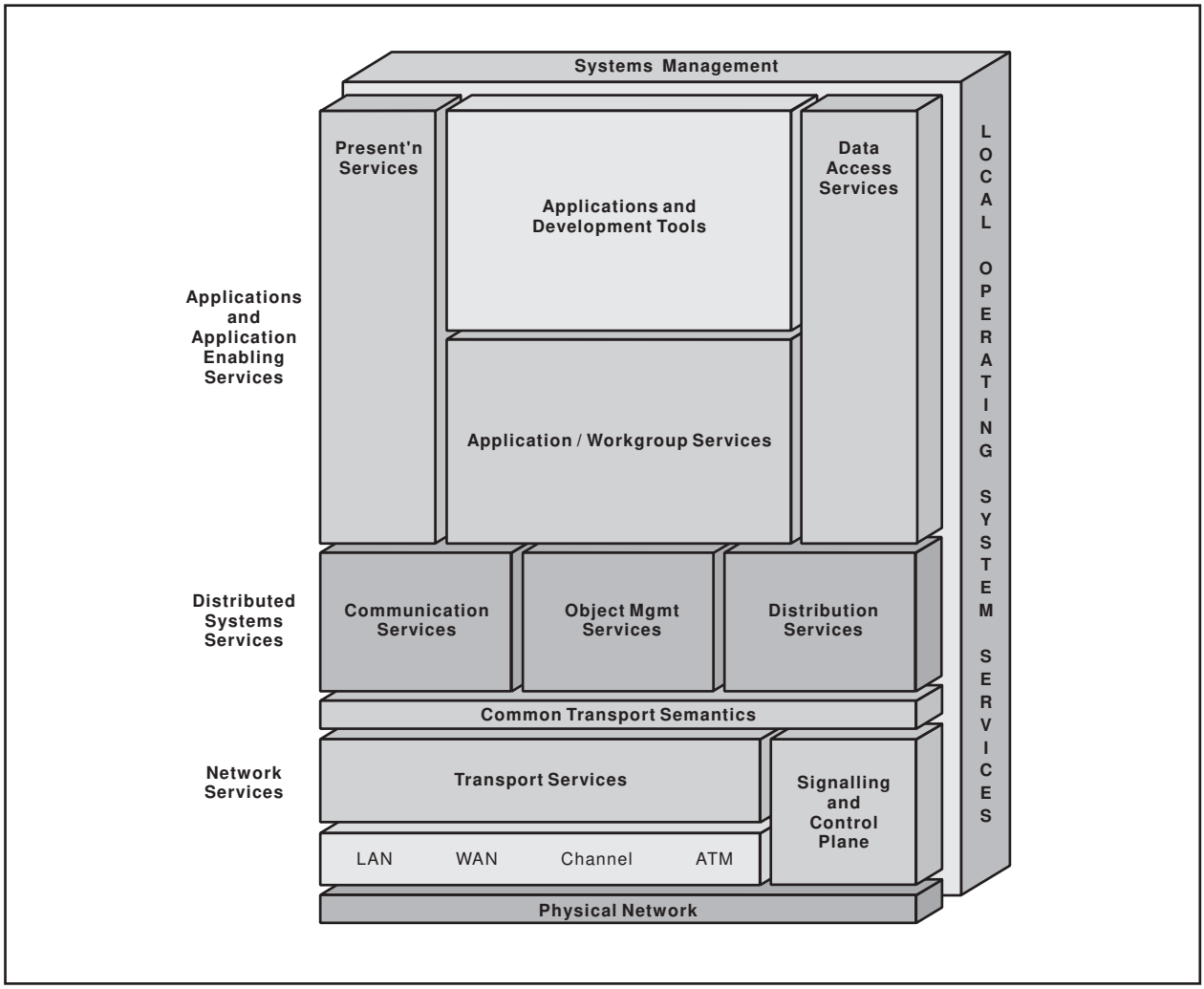
Open Blueprint

Print Resource Manager



GC23-3924-01

Open Blueprint

Print Resource Manager

GC23-3924-01

About This Paper

Open, distributed computing of all forms, including client/server and network computing, is the model that is driving the rapid evolution of information technology today. The Open Blueprint structure is IBM's industry-leading architectural framework for distributed computing in a multivendor, heterogeneous environment. This paper describes the Print resource manager component of the Open Blueprint and its relationships with other Open Blueprint components.

The Open Blueprint structure continues to accommodate advances in technology and incorporate emerging standards and protocols as information technology needs and capabilities evolve. For example, the structure now incorporates digital library, object-oriented and mobile technologies, and support for internet-enabled applications. Thus, this document is a snapshot at a particular point in time. The Open Blueprint structure will continue to evolve as new technologies emerge.

This paper is one in a series of papers available in the *Open Blueprint Technical Reference Library* collection, SBOF-8702 (hardcopy) or SK2T-2478 (CD-ROM). The intent of this technical library is to provide detailed information about each Open Blueprint component. The authors of these papers are the developers and designers directly responsible for the components, so you might observe differences in style, scope, and format between this paper and others.

Readers who are less familiar with a particular component can refer to the referenced materials to gain basic background knowledge not included in the papers. For a general technical overview of the Open Blueprint, see the *Open Blueprint Technical Overview*, GC23-3808.

Who Should Read This Paper

This paper is intended for audiences requiring technical detail about the Print Resource Manager in the Open Blueprint. These include:

- Customers who are planning technology or architecture investments
- Software vendors who are developing products to interoperate with other products that support the Open Blueprint
- Consultants and service providers who offer integration services to customers

© **Copyright International Business Machines Corporation 1995, 1996. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Print Resource Manager 1
Introduction 1
Background 1

Print Resource Manager Components 5
Print Resource Manager Structure 7
Security 10
Distributed Time Services 11
Scalability 11
Auxiliary Support 11
References 15

Appendix A. Notices 17
Trademarks 17

Appendix B. Communicating Your Comments to IBM 19

Figures

1. The Print Environment of the 1990's 2
2. Print Resource Manager Client, Servers and Objects 5
3. Print Resource Manager Structure 7
4. Attribute Defaulting 8
5. Authorizing Access to Printer Functions 11

© Copyright IBM Corp. 1995, 1996

Print Resource Manager

This description of the Print resource manager represents a strategic architecture for the provision of distributed print/print management. It provides common APIs and cross platform consistent services. Full implementation of the architecture is a long-term objective.

Introduction

Printing systems show great variations across operating platforms as well as between hardware devices. This requires applications to support a variety of programming interfaces and functional variations, and hardware vendors to implement device drivers which are sensitive to a specific operating environment.

The Print resource manager removes these requirements by providing a common set of application programming interfaces which support print job submission and management, device management and output, and consistent print system behavior.

Background

In the early 1980s, most printing was limited to an isolated facility, under the centralized control of experienced operators. Applications were created with an understanding of the printer device their output would be sent to, and usually produced relatively simple line data (for example, 1403).

In the mid-1980s, several trends began to shape printing in the enterprise, creating the heterogeneous print environment portrayed in Figure 1 on page 2. These trends included:

- The growth in personal workstations brought with it a corresponding explosion in the quantity and variety of desktop printers. The effect was to dramatically increase the number of print system interfaces and printer protocols (such as, HPPCL, IPDS, and PPDS).
- At the same time, the use of intelligent workstations fostered an increase in the diversity of applications generating printable output along with the complexity of the document formats themselves (color PostScript, compound AFPDS, etc.). In many cases, it became difficult to locate and select a printer device that was compatible with the document's format.
- Through the implementation of LAN server-attached network-attached printers, much of the printing of host-generated data was moved from a few, central locations closer to the end-user work area. Further, the same printers were often used to support local, LAN-based printing. Printers could no longer be centrally monitored for interventions or supplies outages.
- The role of the end-user and operator/administrator, so clearly defined under the centralized, production printing paradigm, began to blur with distributed printing. Day-to-day operational control such as adding paper, removing output, or fixing simple paper jams became the responsibility of a department member.

In response to these changes, a number of standards and initiatives have emerged. These include:

- The definition of the Printer MIB in RFC 1759 from the Internet Engineering Task Force
- The Desktop Management Interface Specification and the Common printer MIB/MIF specification from the Desktop Management Task Force
- The Network Printer Alliance Protocol (NPAP) specification
- The International Organization for Standards (ISO) Document Printing Application (DPA) ISO-10175-1

© Copyright IBM Corp. 1995, 1996

- The IEEE POSIX 1387.4 standard
- The X/Open Printing System Interoperability Specification
- The Palladium Print System.

The Open Blueprint Print resource manager has been influenced by all of the above; the following sections focus on the latter four, which form the basis for print interoperability in an open, distributed system environment.

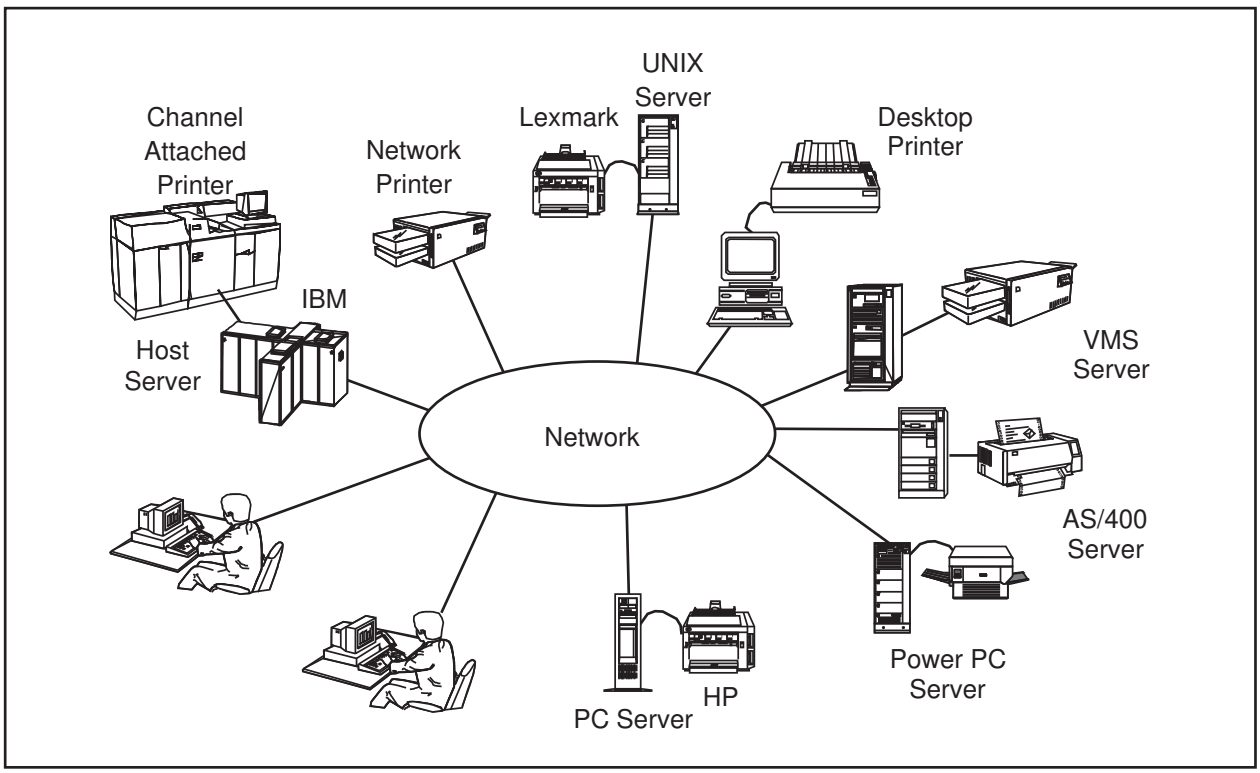


Figure 1. The Print Environment of the 1990's

ISO Distributed Print Application 10175-1

The *International Organization for Standards (ISO) Document Printing Application (DPA) 10175-1 standard*, defines the objects, attributes and operations for the distributed print environment. It assumes that a document is ready for delivery to a hardware device such as a printer, plotter, microfiche or fax machine. What remains is the queuing, preparation, rendering and finishing of the fully composed document at a device.

The ISO DPA describes the following:

- Classes of print-related objects. Each object is precisely defined using attributes, and provided with a consistent definition of how the object relates to another object (behavior). Examples of print objects include job, document, printer, server, and font.
- Attributes which are defined in terms of the objects they may describe, and the values (attribute values) they may contain. As an example, a “document” object may be partially defined by the attribute “data stream,” which in turn may be defined by the value “PostScript.”

- Operations which act on objects and attributes to provide access to and control of the distributed print system. For example, ISO DPA defines operations for submitting, interrupting, pausing or cancelling a print job and operations which permit inquiry about the status, capabilities and characteristics of a print server.

IEEE POSIX 1387.4 Standard

The IEEE POSIX 1387 family of standards defines a set of interfaces to support system administration, with emphasis on the distributed environment. The *POSIX System Administration - Part 4: Printing Interfaces 1387.4* standard was created to overcome some of the limitations in the ISO DPA 10175 standard. It supports and extends the ISO DPA 10175 definition of managed objects and operations, but also defines a consistent set of application and command line interfaces for distributed printing. The IEEE PSOIX 1387.4 describes the following:

- The ISO DPA object set is extended by defining new objects with their own set of attributes and values, as well as by adding new attributes and values to existing ISO DPA objects. For example, the object printer is redefined into physical printers and logical printers to provide a separation between the real, physical devices, and an applications “view” of the resource.
- The ISO set of operations is extended to include object creation, set (for setting attribute values) and delete.
- An Application Programming Interface (API) is added to support the calls to compose, submit, and capture operational requests/responses between clients and servers.
- Command Line Interfaces (CLI) are added to provide a consistent, human-centric interface which is functionally equivalent to the POSIX API set. The CLI consists of 17 commands and several hundred attributes which support each of the ISO and POSIX operations.

X/Open Printing System Interoperability Specification

Taken together, ISO DPA 10175 and POSIX 1387.4 fall short of completely defining the communications between distributed print systems clients and servers. An emerging standard called the *X/Open Submission, A Printing System Interoperability Specification (PSIS) Extensions to the ISO DPA 10175 and POSIX 1387.4*, is intended to overcome a number of the limitations. PSIS addresses:

- Implementation-specific techniques for using distributed services such as the OSF Distributed Computing Environment (DCE) or Sun's Open Network Computing (ONC).
- Client-to-server and server-to-server communications.
- Portability for interfaces to modules that may be supplied by third parties (for example, job scheduling algorithms, notification services or printer control).

Palladium Print System

Palladium is an architecture for a client/server-based distributed printing system that today supports both the ISO DPA 10175 and POSIX 1387.4 standards. It was developed to address the deficiencies with print systems in the distributed environment.

Work on Palladium began in 1988 as part of the Athena Project at the Massachusetts Institute of Technology. IBM, Digital Equipment Corporation, and Hewlett-Packard Corporation all participated in its initial design and development. In 1991, a Palladium prototype was submitted to the Open Software Foundation in response to the printing services portion of the Distributed Management Environment (DME) Request for Technology.

Several vendors, including IBM, DAZEL Corporation, Sun Microsystems, and Hewlett-Packard Corporation, have announced intentions to deliver products based on Palladium technology.

Print Resource Manager Components

The Print resource manager extends the Palladium technology, and implements the ISO DPA 10175, POSIX 1387.4 and PSIS standards, to provide the interfaces and services to access and control print drivers and print subsystems in the distributed network.

A client, spooler, supervisor and 11 managed objects provide the basic building blocks of the system (see Figure 2). *Clients* accept application requests, and in turn issue requests to both spoolers and supervisors. *Spoolers* are servers that accept requests from many clients, validate job and document attributes with those of the intended print destination, provide queuing and scheduling, and then route jobs/documents to the appropriate printer. *Supervisors* are servers that provide a bridge to the printer device support (print device driver or print subsystem).

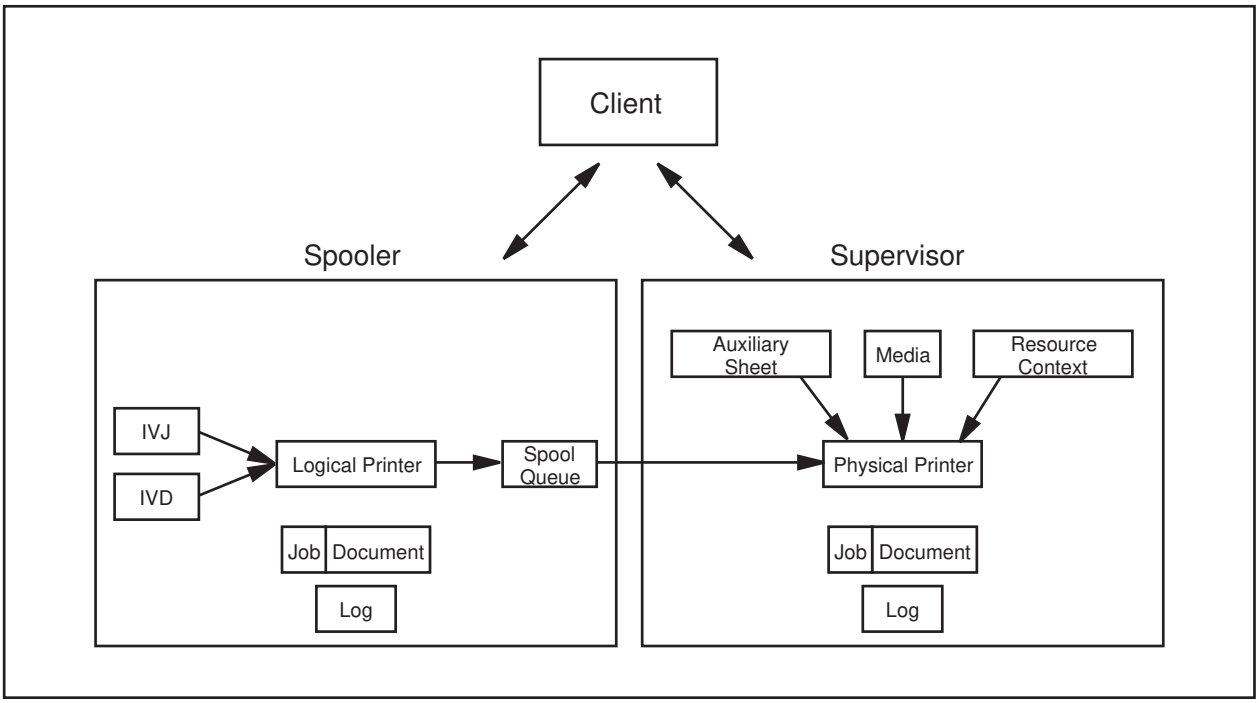


Figure 2. Print Resource Manager Client, Servers and Objects

Objects

Spoolers and supervisors manage the following objects:

Documents — Created in response to a print request. Documents convey the formatted data and summary information about the data's characteristics and processing (print media, fonts, page size, orientation, page order, etc.). A document is always associated with a single job. Documents may reside in spoolers or supervisors.

Jobs — Consist of one or more documents. Jobs contain common job information (name, originator, accounting, etc.), and convey information about error and event notification, scheduling, printer selection, security and processing status. A job may be associated with one or more documents. Jobs may reside in spoolers or supervisors.

© Copyright IBM Corp. 1995, 1996

Logical Printers

Print jobs are submitted to logical printers, a collection of characteristics (duplex, paper type, etc.), which are referenced as a single entity. Logical printers provide several important functions:

1. Validation of documents and jobs to ensure their attribute values are compatible with those of both the logical printer itself and physical printer(s) down-line, effectively restricting access to a printer based on attribute values.
2. Attribute defaulting which is the association of specific job and document attribute values with a logical printer.
3. The separation of the application-defined print destination (logical printer) from the physical printer. A logical printer can feed one or more physical devices, and conversely, one physical printer can be fed by multiple logical printers.

Logical printers are contained in spoolers.

Queues

Receives jobs from one or more logical printers, and holds them for one or more physical printers. A scheduler is associated with each queue to place the jobs in the order in which they will be printed. Scheduling may be based on a variety of algorithms including (but not limited to) FIFO, job size and job id. Queues are contained in spoolers.

Logs

Supports the system by tracing jobs and holding error messages. Logs are contained in spoolers and supervisors.

Initial Value Jobs (IVJ)

Provide sets of job default properties for a logical printer. IVJs are provided with a print request or are associated with a logical printer. Because of IVJs, applications do not need to define all the attribute values for each job they submit, only those of particular importance, such as number of copies or plex. IVJs are contained in spoolers.

Initial Value Documents (IVD)

Like Initial Value Jobs, IVDs provide sets of document default properties for logical printers. One or more IVDs may be provided with a print request. An IVD may also be associated with a logical printer. Because of IVDs, applications do not need to define all the attribute values for each document they submit. IVDs are contained in spoolers.

Physical Printers

Representations of actual printers and their features. Physical printers are contained in supervisors.

Auxiliary Sheets

Represent start, separator, and end sheets to be printed along with the job output. Separator sheets are printed between documents in a job, and often list the job owner, name and print date. Auxiliary sheets are contained in supervisors.

Resource Contexts

Represent the locations of sets of printing resources, such as files needed for form definitions or finishing equipment (staplers, binders, etc.). Resource contexts are contained in supervisors.

Media

Represent physical material on which output can be rendered. For example, paper or microfiche. Media are contained in supervisors.

Print Resource Manager Structure

The client, spooler and supervisor interact with each other to process printing and administrative requests. Figure 3 depicts the structures which support this processing. The following discussion is based on a printer scenario, but could be applied to any number of physical devices (plotter, microfiche, etc.).

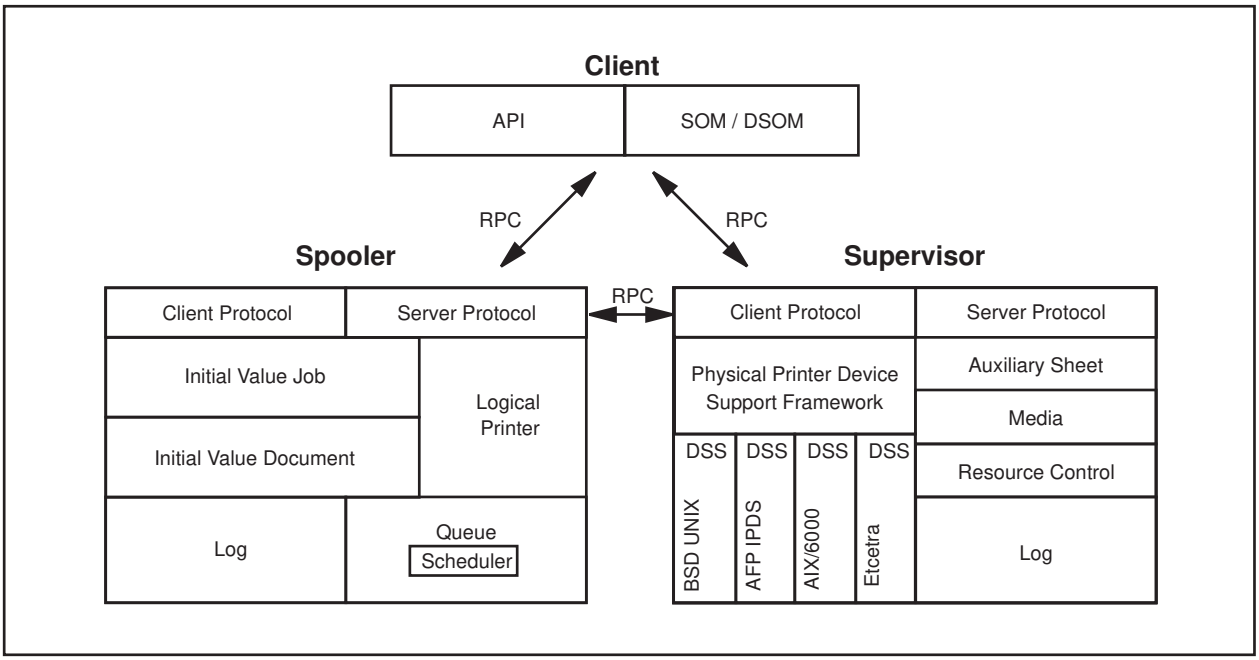


Figure 3. Print Resource Manager Structure

Client

The client can simultaneously process a number of requests from one or more applications. Requestor access to the client is through POSIX 1387.4-compliant API's, or by an object-oriented application using the Open Blueprint Object Request Broker.

The client relies on Open Blueprint security services to provide single login support to the Print resource manager controlled system, and to restrict, or enable, access to any object or operation.

Open Blueprint Directory Services are used to determine the location of logical printers, queues, spoolers, physical printers, and supervisors. The client communicates with spoolers and supervisors on behalf of applications to create, query, modify and delete print objects. The request identifies the services to be performed against one or more objects. The request itself is passed to either the spooler or supervisor using Open Blueprint Remote Procedure Call (RPC). Print data is transferred using Open Blueprint RPC or passed as an external reference.

Spooler

The spooler is primarily responsible for ensuring that jobs and documents are selected for processing on the appropriate physical printer. The spooler supports notification, and provides the services for creating, deleting and managing logical printers, queues, logs, IVJs, IVDs, jobs and documents.

The spooler receives the client request to create a document or job in association with a logical printer. In concert with the print request, the application specifies attributes which further define how the job is to be printed (simplex, duplex, fan fold, number of copies, type of paper, etc.). When an attribute has not been

set, defaulting, based on the Initial Value Job and Initial Value Document Objects, is used to assign values. This provides a method to separate the specification of one or more attribute values from the requesting application. At the same time, defaulting can be used to ensure that repetitively printed matter maintains a consistent look and feel. For example, (see Figure 4) two logical printers, feeding the same set of physical printers, can be defined so that one implies duplex printing on letter size paper and the second implies simplex printing on legal size paper.

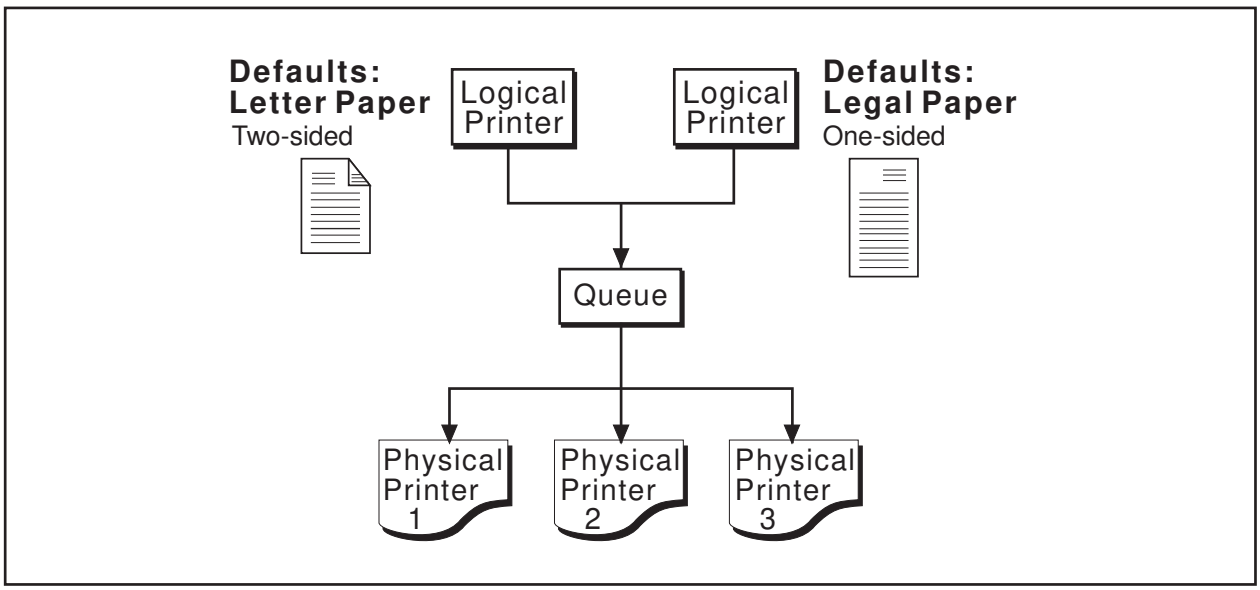


Figure 4. Attribute Defaulting

The IVD and IVJ are associated with specific logical printers, specified by the requesting application, or can be omitted entirely.

Both logical and physical printer validations steps are performed during the process of creating a job. The logical printer job validation compares requested job attributes to the logical printer for the administrative purpose of:

- Restricting physical printer function.
- Controlling physical printer selection.
- Restricting physical printer access.

Further discussion is covered in “Security” on page 10.

The physical printer job validation compares requested job attributes to those of the available physical printers (physical printers which can be accessed by the queue that the logical printer feeds). Attributes which are not supported by the physical printer(s) cause a job or document to be rejected. This has the effect of removing or filtering jobs or documents which are incompatible with the physical printers serviced by the logical printer.

Logical printers are associated with a single queue, which in turn, bounds which physical printer(s) a job may be assigned to. This association can be changed, but only to another queue in the same spooler. Queues can receive jobs from any number of logical printers within the same spooler, and may send jobs to any number of physical printers residing in any number of supervisors. This structure provides the flexibility to add, modify or remove physical printers without the knowledge of the requesting application.

Once on the queue, job scheduling is based on predetermine algorithms, and the availability of a physical device which is capable of printing the job. The Print resource manager defaults to managing queues using priority FIFO processing. Another algorithm may be substituted using services which support the

X/Open PSIS standard. Independent of the algorithm, other conditions can affect queue processing. For example, it is possible for the job to be behind five previously submitted jobs, but print ahead of them. This may occur if a job does not require a print function the other jobs do, allowing it to be scheduled to an idle printer that could not print the waiting jobs. Or, using the *job promotion* services in the Print resource manager, an application can request that a job be placed at the top of a queue. Application access to this function is through the client API.

Throughout this process, an application can monitor the job using notification or query services. Notification services provide the application with asynchronous feedback on selected events during job processing (job aborted, job cancelled, job completed, etc.). Spoolers, supervisors, printers, queues and jobs can all generate notifications. The notification may be delivered to an application using a variety of messaging services including the Open Blueprint Mail resource manager.

Query services are used to request information on the state of a job while resident on a queue or in the printer. An application may also affect the progress of a job by modifying attributes, changing priorities, moving a job to a new queue, or cancelling the job altogether.

The spooler functions support a large number of potential application request scenarios through the ability of a client to create, query, modify and delete the seven spooler objects (logical printers, queues, logs, IVJs, IVDs, jobs and documents). The following are a few examples of these functions:

- In selecting a logical printer, an application may **query** several spoolers to locate a PostScript compatible printer or list every enabled printer on a specific floor of a building.
- A system management application can receive Print resource manager notifications about print-related events (for example, a queue exceeds a threshold value) and then **modify** the attributes of a logical printer to reduce the workload.
- Levels of service can be preset and then maintained by regulating the print capacity which supports a logical printer. For example, a physical printer object can be **created** and added to the Print resource manager without changing either the logical printer or the application which references the logical printer.
- A job or document can be stored in the spooler, printed many times, and then **deleted** when it is no longer required.

Supervisor

Supervisors communicate with many clients and servers using Open Blueprint RPC. The supervisor provides the interface to the physical printer device driver or printer subsystem. The supervisor is primarily responsible for managing jobs and documents once they have been assigned to a physical printer.

A single supervisor can support multiple physical printers. The printers may be located anywhere in the network and attach to the supervisor through network or local attachments (for example, a TCP/IP network or parallel port).

The nucleus of the supervisor is the object-oriented device support framework (DSF). DSF is a device and platform-independent set of object classes which provide the interface between the Print resource manager and specific hardware represented by a device support system (DSS). It provides a minimum set of default behavior (expected actions) for the supervisor and hardware. Examples of the default behavior include listing job attributes, enabling, disabling or deleting printers, and cancelling, pausing or resuming jobs. There is no visibility of the DSF at the application level.

On the other hand, an application may be aware of a DSS because of the need to understand printer attributes and capabilities. The DSS provides three primary functions:

- The device-specific support required by a printer or family of printers.
- Extensions to attributes and values which are unique to a specific printer or printer family.
- Customization of the physical printer or printer subsystem.

A number of tasks are directly dependent on the DSS implementation, including the creation of a physical printer object, and setting printer attributes and values. Using a DSS, a hardware vendor can write device support which is less dependent on a specific operating system. The Print resource manager supports many DSSs, for example the AIX/6000 native print support, simple BSD UNIX print, and the Advanced Function Printing (AFP) Intelligent Printer Data Stream (IPDS) protocol. Additional DSSs may be added to increase device support, or to support platform-specific capabilities.

The supervisor controls one or more physical printer objects. Each physical printer object contains an attribute which identifies the name of the queue it is associated with. During physical printer initialization, the Open Blueprint Directory is searched for the queue name, to determine, in turn, the spooler the queue resides in. This establishes the binding between the physical printer, and the spooler/queue.

The physical printer object initiates a print operation by signaling the spooler that it is ready for a print job. The spooler queue then submits a print job to the physical printer object. This process causes the job to be bound to a specific physical device, and results in an association with resource context, media and auxiliary sheet objects. The job or document is passed through the correct DSS, and then to the print device driver or subsystem.

At print job completion, notification is passed from the supervisor to the spooler and back to the appropriate application. The definition of print job completion is dependent on the specific print driver or subsystem. In some cases, notification can only be sent when the job is physically passed to the print driver. In more sophisticated systems such as with AFP IPDS printers, notification is sent when the last page has actually printed. In either case, the job or document “lives” (persists) until it is no longer required. This may be until the supervisor has notified the spooler that the job successfully printed, or until the application requests that the job be purged.

The spooler may communicate with the supervisor during print processing. The communication provides a vehicle to cancel or pause a printing job, or list a job's status. Conversely, the supervisor can inform the spooler and/or client that it has completed pausing or deleting a job, or that an error condition exists (for example, a paper jam).

The supervisor functions support a number of potential scenarios through the ability of a client to create, query, modify and delete the seven supervisor objects (physical printers, logs, media, jobs, documents, resource context and auxiliary sheet). The most significant of these is the ability to add, change or eliminate physical printers independent of any change to an application.

Security

The Open Blueprint Access Control resource manager is used to manage access to printers, spoolers, supervisors and print objects, and print system services. This authorization may be combined to effectively allow the Print resource manager to be used to establish a hierarchy which can fence application access to and control over the print system.

For example, in Figure 5 on page 11, a scenario is portrayed in which security services are used to implement controlled access over the production of multiple copies and the use of legal-size paper. A pool of three physical printers is fed by two logical printers. Logical printer 1 supports printing multiple copies of a single job, and printing on either letter or legal-size paper. The Open Blueprint Access Control resource manager is used to selectively authorize application access to logical printer 1. Logical printer 2

has unrestricted application access, but is restricted to printing jobs which require a single copy on letter-sized paper.

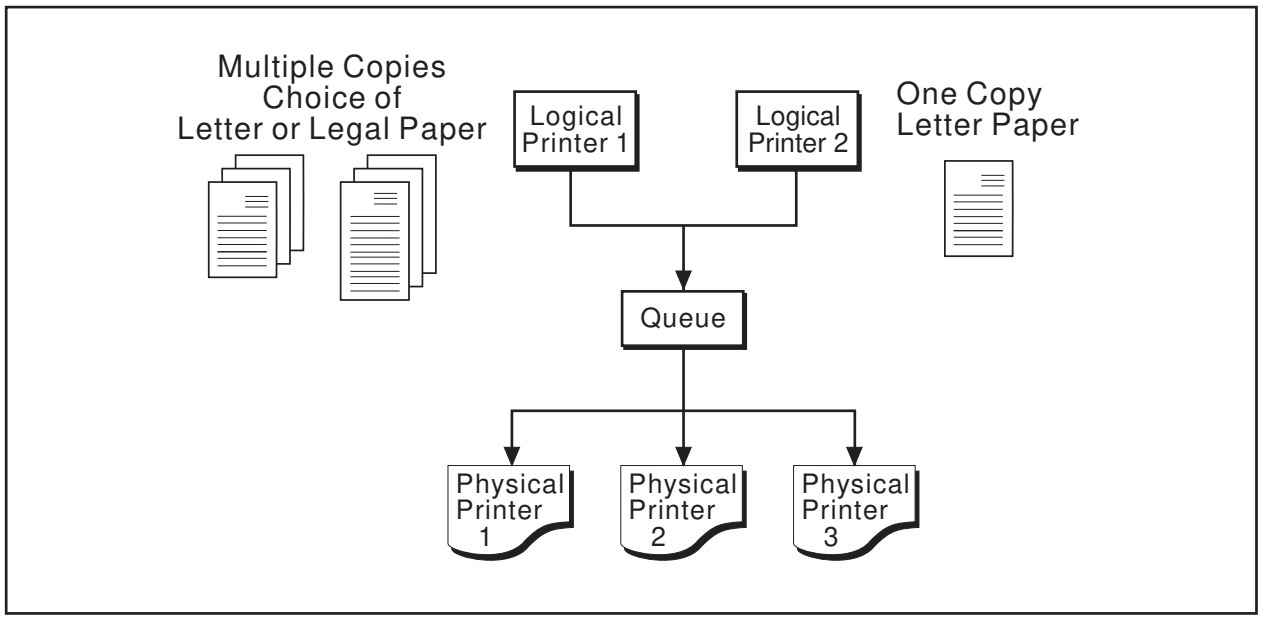


Figure 5. Authorizing Access to Printer Functions

Authorization may be implemented on one or more of the following:

- **Printers** - Authorization to submit print jobs is determined at the logical printer. The logical printer may control access to a single physical printer or multiple printers, all of which are associated with the same queue. Unauthorized jobs or documents are rejected before they are placed on the queue.
- **Spoolers, supervisors and print objects** - In general, application access is checked at the server level first followed by the object level. Authorization on jobs is requested at the owner level first (job owner), then the queue the job resides in, followed by the server if the job is not in a queue.
- **Print System Services** - Access control can be placed on any Print resource manager service (such as creating an object, promoting jobs in a queue, etc.).

Distributed Time Services

Open Blueprint Time Services are used to provide the clock synchronization that enable the Print resource manager to determine event sequences, duration and scheduling.

Scalability

The Open Blueprint resource manager scales to the size of the network through the ability to define any number of spoolers and supervisors.

Auxiliary Support

The Print resource manager provides three applications which immediately exploit the client interface. These applications are Gateway services, a Command Line Interface, and a System Administrator Graphical User Interface.

Gateway Services

The Print resource manager Gateway services provide limited access to end-user services by mapping existing print methods to Print resource manager APIs. These applications provide controlled accessibility for print requestors that are not logged onto the Open Blueprint security services. The net effect is to allow for the gradual migration of end-user applications to the Print resource manager interfaces long after print drivers, subsystems and devices have been placed under Print resource manager Control. The degree of support of Print resource manager Services varies by application, though it is roughly equivalent to basic UNIX and PC network printing. The following are the four gateway services:

1. Local system access using system-specific print interfaces such as the AIX lp, lpr, qprt, and enque commands. This requires an instance of the Print resource manager Client to reside on the system that the local system is logged into. The gateway supports print submission, listing printer status and the jobs in a queue, and job removal.
2. LPR print protocol access from any system which uses the X/Open 1179 TCP/IP Protocol. LPR commands are passed over TCP/IP Sockets to gateway services which map the command and their parameters to Print resource manager APIs. The gateway supports print submission, list (printer up/down, jobs in a queue) and job removal. This same technique can be used to give LAN Server user access to Print resource manager controlled printers.
3. OS/2 access through extensions to the OS/2 network printer support. The extensions enable Print resource manager logical printers to be identified to the OS/2 system. An end-user or application can access the services using standard OS/2 interfaces such as drag-and-drop or application printing. Requests are passed from the OS/2 system over TCP/IP Sockets to gateway services. The gateway supports print submission and list (printer up/down, jobs in a queue) functions.
4. Windows access using an application which defines Print resource manager logical printers to the Windows system. Application access to the printers is through standard Windows print interfaces. Requests are passed from the Windows system over TCP/IP Sockets to gateway services. The gateway supports print submission.

Command Line Interface

The Print resource manager command line interface (CLI) is an implementation of 16 of the POSIX 1387.4 commands. The commands and their definitions are as follows:

pdls Retrieves selected attributes of the specified object instance from servers.

pdmod Modifies or sets the attributes of one or more previously submitted print jobs or documents within a job.

pdpr Creates and submits a single print job, consisting of one or more documents, to a spooler for printing on a printer.

pdq Requests the status of one or more print jobs.

pdresubmit Resubmits a previously submitted job to a specified logical printer on the same or different spooler.

pdrm Cancels (removes) one or more previously submitted print jobs or documents within a job.

pdclean Removes all jobs on the specified server or queue.

pdcreate Creates print objects.

pddelete Deletes print objects.

pddisable Disables the specified server, printer, or queue from accepting print requests.

pdenable Allows the specified server, printer, or queue to accept print requests.

pdpause Pauses an idle or currently-printing job, physical printer, server, or queue. Requests are still accepted.

pdpromote Advances the specified job to the beginning of a queue which services the next appropriate printer available.

pdresume Resumes a currently paused job, physical printer, server, or queue.

pdset Sets the attributes of the specified object.

pdshutdown Shuts down the specified server after all jobs have printed, all current jobs have printed, or immediately.

System Administrator Graphical User Interface

The system administrator graphical user-interface (GUI) application is a Motif-based interface which supports the administration and operation of the distributed print network. The GUI supports all command line functions except print submission.

The GUI is object-oriented at the user-interface level. Print resource manager objects and actions are represented by icons which populate windows according to their position in the print system hierarchy. The graphical representation of the hierarchy allows the user to view the entire configuration or portions of the configuration as needed. The icons incorporate visual cues that allow the states of the objects and significant events to be represented graphically. For example, the icon for a paused job is different from the icon for a job that is progressing through the queue normally.

Any collection of objects represented in a window or portion of a window can be filtered according to attribute values. For example, a window showing printers can be filtered to show only those printers that support PostScript and duplex. In addition, a Find capability allows the user to search through a collection of objects for a specific object by specifying the name of the object or other distinguishing attributes.

Object creation is accomplished through drag-and-drop of template objects. The GUI application provides a base set of templates for common printer types. The templates can be replicated and customized.

Objects and actions interact primarily through drag-and-drop mechanisms. For example, an Enable action can be dragged to a printer to make the printer able to accept print jobs. A printer can be detached from a queue and attached to another queue by dragging the printer from one queue to another. Print jobs can be rerouted by dragging them from a queue to a logical printer.

Notifications are represented at the interface as icon changes, pop-up messages, or entries in an event log. Notification can be customized so that the user can choose whether or not to log certain events and whether or not to surface events for a specific object. For example, the user can turn off event notifications for a printer that will be disabled for an extended period of time.

The attributes for each object class are presented in a notebook style dialog. The notebooks organize the attributes logically so they can be referenced easily and in an appropriate context. A single notebook for each object allows the user to query or modify the attributes of the object.

The GUI application can be run stand-alone or may be integrated with selected Open Blueprint Systems Management Applications (NetView/6000) to provide a single monitoring point for all components of the network.

References

1. *IEEE, POSIX 1387.4, System Administration - Part 4: Printing Interfaces*.
2. *ISO 10175-1, Information Technology - Document Printing Application*
3. *X/Open Submission, A Printing System Interoperability Specification (PSIS) Extensions to the ISO DPA 10175 and POSIX 1387.4*.

Appendix A. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Function Printing
AFP
AIX
IBM
IBMLink
Intelligent Printer Data Stream
IPDS
Netview
Open Blueprint
OS/2

The following terms are trademarks of other companies:

BSD UNIX	AT&T Bell Laboratories Inc.
DCE	Open Software Foundation, Inc.
Digital	Digital Equipment Corporation
Hewlett-Packard	Hewlett-Packard Company
ONC	Sun Microsystems, Inc.
Motif	Open Software Foundation, Inc.
Open Software Foundation	Open Software Foundation, Inc.
OSF	Open Software Foundation, Inc.
Palladium	Massachusetts Institute of Technology
PostScript	Adobe Systems Incorporated
POSIX	Institute of Electrical and Electronic Engineers
Sun	Sun Microsystems, Inc.
Sun Microsystems	Sun Microsystems, Inc.
UNIX	UNIX Systems Laboratories, Inc.
Windows	Microsoft Corporation
X/Open	X/Open Company Limited

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

© Copyright IBM Corp. 1995, 1996

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Appendix B. Communicating Your Comments to IBM

If you especially like or dislike anything about this paper, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply. Feel free to comment on specific error or omissions, accuracy, organization, subject matter, or completeness of this paper.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

- If you prefer to send comments by FAX, use this number:

  United States and Canada: 1-800-227-5088.

- If you prefer to send comments electronically, use one of these ID's:
  - Internet: **USIB2HPD@VNET.IBM.COM**
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**

Make sure to include the following in your note:

- Title of this paper
- Page number or topic to which your comment applies

© Copyright IBM Corp. 1995, 1996

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC23-3924-01