AppleShare IP 6.3
Developer's Kit

AppleShare IP Print Server Security Protocol

Technical Publications

Apple Computer, Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

About This Manual

This manual describes the security protocol that the AppleShare IP Print Server uses to control access to print queues. The protocol is implemented in the LaserWriter 8.6.1 driver, which is compatible with Mac OS 8.5 or later and with the AppleShare IP Print Server 6.1 or later. The protocol uses PostScript document structuring comments to exchange authentication information between a Print Server client and the Print Server.

Conventions Used in This Manual

The Courier font is used to indicate text that you type or see displayed. This manual includes special text elements to highlight important or supplemental information:

Note
Text set off in this manner presents sidelights or interesting points of information. ◆

IMPORTANT
Text set off in this manner—with the word Important—presents important information or instructions. ▲

▲ **WARNING**
Text set off in this manner—with the word Warning—indicates potentially serious problems. ▲

For more information

The following sources provide additional information that may be of interest to Apple Share developers:

- *AppleShare IP Administrator's Manual.* Apple Computer, Inc.
- *Inside Macintosh*. Apple Computer, Inc.

For information on the programming interface for managing users and groups, see the following publication:

- *AppleShare IP 6.3 Developer's Kit: AppleShare Registry Library.* Apple Computer, Inc.

For information on the AppleTalk Filing Protocol (AFP), see the following publications:

- *AppleShare IP 6.3 Developer's Kit: AppleTalk Filing Protocol.* Apple Computer, Inc.
- *AppleShare IP 6.3 Developer's Kit: AppleTalk Filing Protocol Version 2.1 and 2.2.* Apple Computer, Inc.
- *Inside AppleTalk*, Second Edition. Apple Computer, Inc.

For information on controlling an AppleShare file server and handling server events, see the following publication:

- *AppleShare IP 6.3 Developer's Kit: Server Control Calls and Server Event Handling.* Apple Computer, Inc.

For information on user authentication modules (UAMs), see the following publication:

- *AppleShare IP 6.3 Developer's Kit*: *User Authentication Modules.* Apple Computer, Inc.

For information on using an AppleShare IP 6.3 file server and Macintosh File Sharing, see the following manuals:

- *AppleShare Client User's Manual*. Apple Computer, Inc.
- *Macintosh Networking Reference.* Apple Computer, Inc.

AppleShare IP Print Server Security Protocol

The AppleShare IP Print Server uses a security protocol to control access to print queues. The protocol is implemented in the LaserWriter 8.6.1 driver, which can be downloaded from Apple Computer's website and can be installed on computers running Mac OS 8.5 or later. This functionality is built into the LaserWriter driver that comes with Mac OS 8.6 or later.

Note
This document describes the security protocol for version 6.1 or later of the AppleShare IP Print Server. Version 6.0 of the AppleShare IP Print Server also supported a security protocol. That security protocol is no longer supported and is superseded by the protocol described in this document. ◆

The AppleShare IP Print Server Security Protocol supports two security mechanisms:

- *No password*. The no-password security mechanism requires a user name but does not require a password. With this security mechanism, a Print Server queue is configured to monitor the data stream for print jobs that contain an acceptable user name. Alternatively, the LaserWriter 8.6.1 driver can display a dialog box that requires the user to enter a user name but does not require a password. Although the Print Server accepts print jobs for the queue from acceptable users only, it does respond to any queries. This security mechanism is useful for supporting client applications that do not implement the security protocol.
- *Password*. With the password security mechanism, the LaserWriter 8.6.1 driver displays a dialog box that requires the user to enter a user name and password in order to connect to a Print Server queue. Once the network session is authenticated, the Print Server accepts all jobs for the queue that are submitted using the authenticated session. With the exception of the `RBISpoolerID` and `RBIUAMListQuery` queries, the Print Server does not respond

to queries from unauthenticated clients. This security mechanism can only be used by clients that implement the Print Server Security Protocol.

Note
The AppleShare IP Print Server Security Protocol authenticates network sessions; it does not authenticate individual print jobs within a session. In order to implement session authentication, both security mechanisms cause the AppleTalk Printer Access Protocol (PAP), the PAP EOF flag to be ignored. ◆

Document Structuring Comments

To support the Print Server Security Protocol, the AppleShare IP 6.1 or later Print Server responds to the following new document structuring comments:

- `RBISpoolerID`
- `RBIUAMListQuery`
- `RBILogin`
- `RBILoginCont`

RBISpoolerID Comment

The `RBISpoolerID` comment delimits PostScript code that requests information describing the Print Server's product name, version number, and other descriptive text. The Print Server's standard response consists of its product name, version number, and descriptive text.

The code for the `RBISpoolerID` comment is

```
%%?BeginQuery: RBISpoolerID
((NotASpooler) 0.0 ()\n)print
%%?EndQuery: <default>
```

The Print Server's response is of the form

```
"(SpoolerProductNameString) VersionNumber
(OtherDescriptiveString)"<newline>
```

The *SpoolerProductName* and *OtherDescriptiveString* fields are text strings as defined in the Adobe System Incorporated document, *PostScript Language Document Structuring Conventions Specification* version 3.0, available from Adobe's developer support group.

The *VersionNumber* field is an integer or a real number. For example, a version number of 6.1 is valid, but a version number of 6.1.1 is not valid.

The *newline* field terminates the data returned and is defined as a carriage return, a linefeed character, or both.

Table 1-1 lists the responses to the `RBISpoolerID` comment for a variety of devices.

Table 1-1 Responses to the RBISpoolerID comment

Source of Response	Response
A printer	`(NotASpooler) 0.0<newline>`
An AppleShare IP Print Server	`(AppleShare Print Server) 6.1 (6.1.1b3)<newline>`
A NetWare print server	`(Netware Spooler) 5 (5.12Beta1 Developer Series 2<newline>`
A device that does not understand the comment	`(Unknown)<newline>`

RBIUAMListQuery

The `RBIUAMListQuery` comment delimits PostScript code that requests information describing the User Authentication Methods (UAMs) that the Print Server supports.

The code for the `RBIUAMListQuery` comment is

```
%%?BeginQuery: RBIUAMListQuery
(*) == flush
%%?EndQuery: Unknown
```

The response is in the form of a standard list with the listed items in a random order.

Table 1-2 lists the responses to the `RBIUAMListQuery` comment for a variety of devices.

Table 1-2 Responses to the RBIUAMListQuery comment

Source of Response	Response
A printer	`*<`*`newline`*`>`
A spooler that does not recognize the comment	`Unknown<`*`newline`*`>`
An AppleShare IP Print Server that is not configured for the Print Server Security Protocol	*

An AppleShare IP Print Server that is configured for the Print Server Security Protocol. The `NoAuthUAM` entry indicates that a password is not required.	`NoAuthUAM`
	`CleartxtUAM`
	`TwoWayRandnumUAM`
	*

An AppleShare IP Print Server that is configured for the Print Server Security Protocol. The absence of `NoAuthUAM` implies that a password is required.	`CleartxtUAM`
	`TwoWayRandnumUAM`
	*

RBILogin Comment

The `RBILogin` comment delimits PostScript code that starts an authentication process. The standard response is an integer, where 0 indicates that the user was successfully authenticated, any value greater than 0 indicates that the first step in the authentication process was successful, and any negative value indicates failure.

The code for the `RBILogin` comment is

```
%%?BeginQuery: RBILogin UAMName [(username) [(UAMdata)]]
(*) == flush
%%?EndQuery: Unknown
```

Here is an example of the `RBILogin` comment:

```
%%?BeginQuery: RBILogin CleartxtUAM (username) (password)
(*) == flush
%%?EndQuery: Unknown
```

Table 1-3 lists the responses to the `RBILogin` comment for a variety of devices.

Table 1-3 Responses to the RBILogin comment

Source of Response	Response
A printer	`0<newline>`
A spooler that does not recognize the comment	`Unknown<newline>`
A successful authentication with an AppleShare IP Print Server that is configured for the Print Server Security Protocol	`0<newline>`
A successful first step with an AppleShare IP Print Server that is configured for the Print Server Security Protocol	`1 (01234567)<newline>`
A failed authentication with an AppleShare IP Print Server that is configured for the Print Server Security Protocol	`-1 <newline>`

RBILoginCont Comment

The `RBILoginCont` comment delimits PostScript code that continues an authentication process started by the `RBILogin` comment. The standard response is an integer, where 0 indicates that the user was successfully authenticated, any value greater than 0 indicates that this step in the authentication process was successful, and any negative value indicates failure.

The code for the `RBILoginCont` comment is

```
%%?BeginQuery: RBILoginCont (UAMdata) [(More-UAM-data)]
(*) == flush
%%?EndQuery: Unknown
```

The optional *More-UAM-data* field is on the same line as the *UAMdata* field and contains UAM-dependent data that is separated from the *UAMdata* field by a space or a tab.

Here is an example of the `RBILoginCont` comment:

```
%%?BeginQuery: RBILoginCont CleartxtUAM (username) (password)
(*) == flush
%%?EndQuery: Unknown
```

Table 1-4 lists the response to the `RBILoginCont` comment for a variety of devices.

Table 1-4 Responses to the RBILoginCont comment

Source of Response	Response
A printer	`0<newline>`
A spooler that does not recognize the comment	`Unknown<newline>`
A successful authentication and an AppleShare IP Print Server that is configured for the Print Server Security Protocol	`0<newline>`
A successful first step with an AppleShare IP Print Server that is configured for the Print Server Security Protocol.	`1 (01234567)<newline>`

An unsuccessful authentication with an AppleShare IP Print Server that is configured for the Print Server Security Protocol	`-1 <newline>`
	and the following message is generated:
	`%%[Error: SecurityError; SecurityViolation: Unknown user, incorrect password or log on is disabled ]%% <newline>`
	`%%[Flushing: rest of job (to end-of-file) will be ignored ]%% <newline>`

Sample Security Exchanges

This section presents the sequence of client and Print server interactions for the following connection types:

- Explicit connection to a queue with no password required (page 13)
- Implicit Connection to a Queue with No Password Required (page 13)
- Connection to a Queue with Password Required (page 14)

Explicit connection to a queue with no password required

Here is the sequence of exchanges between the client and the Print Server for an explicit connection to a queue when no password is required.

Client sends:	%!PS-Adobe-3.0 Query %%?BeginQuery: RBIUAMListQuery (*) = flush %%?EndQuery: * %%EOF
Server responds:	NoAuthUAM CleartxtUAM TwoWayRandNumUAM *
Client sends:	%!PS-Adobe-3.0 Query %%?BeginQuery: RBILogin NoAuthUAM (*username*) (0) = flush %?EndQuery: Unknown %%EOF
Server responds:	0

Implicit Connection to a Queue with No Password Required

Here is the sequence of exchanges between the client and the Print Server for an implicit connection to a queue when no password is required. This example uses a LaserWriter 8 driver that doesn't support the security protocol.

Client sends:	%!PS-Adobe-3.0 Query %%?BeginQuery: ADOIsBinaryOK? ... %%?EndQuery: Unknown ...
Server responds:	True ...
Client sends:	%!PS-Adobe-3.0 Query %%Title: (Untitled) %%Creator: (QuarkXPress\252: LaserWriter 8 8.4.2) %%CreationDate: (12:02 PM Tuesday July 29, 1998) %%For: (*username*) %%Pages: 14 %%EndComments ... %%EOF

In this example, the server obtains the user name from the “%%For:” comment. If this comment were missing, the Print Server would generate a PostScript error when it receives an end-of-job signal (that is, the PAP EOF flag). To prevent the needless spooling of large documents, the server also generates a PostScript error if the session isn’t authenticated by the time the server sees the “%%EndComments” comment.

Connection to a Queue with Password Required

Here is the sequence of exchanges between a client and the Print Server for an connection to a queue with a password required.

Client sends:	%!PS-Adobe-3.0 Query %%?BeginQuery: RBIUAMListQuery (*) = flush %%?EndQuery: * %%EOF
Server responds:	CleartxtUAM TwoWayRandNumUAM *
Client sends:	%!PS-Adobe-3.0 Query %%?BeginQuery: RBILogin TwoWayRandNumUAM (*username*) (0) = flush %%?EndQuery = Unknown %%EOF

Server responds: (*UAMdata1* is an 8-byte random number challenge.)	`1 <UAMdata1>`

Client sends: (*UAMdata2* is *UAMdata1* encoded with the user's password. *UAMdata3* is an 8-byte random number challenge.)	`%!PS-Adobe-3.0 Query`
	`%%?BeginQuery: RBILoginCont <UAMdata2> <UAMdata3>`
	`(0) = flush`
	`%%?EndQuery: Unknown`
	`%%EOF`

Server responds: (The server uses the user's password to decode *UAMdata2* and compares it with *UAMdata1*. If they match, the user is authenticated. The server creates *UAMdata4* by encoding *UAMdata3* with the user's password and sends *UAMdata4* to the client. At its option, the client can use the user's password to decode *UAMdata4* and compare it with *UAMdata3*. If they match, the client is certain that it is communicating with the intended server and not an entity that is masquerading as the server.)	`0 <UAMdata4>`

If authentication fails, the Print Server generates the following error and closes the connection:

```
%%[ Error: SecurityError; SecurityViolation: Unknown user, incorrect
password or log on is disabled ]%% <newline>

%%[Flushing: rest of job (to end-of-file) will be ignored ]%% <newline>
```