PRINTRONIX AUTO ID

PXML Application Developer Reference Manual

Thermal Series Printers

Trademark Acknowledgements

Printronix and PSA are registered trademarks of Printronix, Inc.

PXML, T8000, T6000, SL4M, T4M, SL5000r, and T5000r are trademarks of Printronix, Inc.

Table of Contents

1 *Overview*

Introduction

This manual describes a communication model for remotely managing Printronix Auto ID printers. It is for application developers to implement custom printer management schemes. **The current version of PXML supported is 2.1.**

Two facets to the management model include:

- Management messages and the operations and events that they describe.
- Transport mechanism that moves the messages back and forth between the printer and the client software.

The main component of the model is the Printronix XML Device Management Language, which describes the management and response messages. PXML is an XML based language that allows a client application to issue commands to a printer and receive responses from the printer. Responses can be either a response to a command (solicited responses), or responses generated by events that occur during printer operation (unsolicited responses).

Please visit the Printronix website (www.PrintronixAutoID.com) to view the printer models and firmware versions available with PXML.

PXML provides the following features:

- printer power-up modes – reset and download.
- different types of printer status reporting.
- job status reporting, including RFID tag and online data validation information.
- configuration change.
- menu changes.

Constraints

- UCP (for GPIO) and PXML cannot be used at the same time in the system. Only one can be active at any time.
- Only one client may connect to the management port at a time.
- The client is responsible for maintaining the TCP management connection to the printer; reconnect as necessary. The management connection may be lost at any time due to printer power-down or network failure. Responses will be lost when the management connection is down since there is no caching or retransmission mechanism.
- It is possible for a client to connect to the management port in the middle of a response message, which at the same time is being sent to a previous client. This is possible if the connection to the previous client is dropped before the message is complete. It is the responsibility of the new client to read and discard the remainder of the message.
- All responses are not synchronized, so a message immediately returned after a command may not be the one that applies to the command. The messages are in XML format and are guaranteed to be complete messages, except when the management connection is lost and reestablished while a message was being sent from the printer.
- XML messages must start with the <?xml> header so that invalid messages can be ignored by the printer.

- The TCP transport mechanism does not provide a separate framing protocol, so the input stream must be processed by an XML parser to determine the end of a message. Alternatively, message framing can be determined by searching for the <?xml> header, then capturing all data until the </pxml> trailer is found.
- One response may span more than one message to relieve the printer from having to buffer output messages. Messages that form a single response are guaranteed to be sent in the correct sequence, and a flag is provided to denote the final message of the sequence.
- For firmware download, the client is responsible for waiting for the printer to enter download mode before sending data. The client is responsible for waiting until the printer fully boots with the new firmware before sending configuration data.
- Notification of start and end of print job must be initiated by some other means, typically be placing PTX_SETUP commands in the print data stream.

2 *PXML*

PXML Selection

PrintNet Enterprise (PNE) is the Printronix Auto ID proprietary management protocol used by the Printronix PrintNet Enterprise software. PXML is the XML based management protocol that serves as a direct interface between third party software. Since version 2.0, both PNE and PXML are active at the same time. However, PXML and GPIO (UCP) share the same channel and only one can be active at any time.

The factory default is port 3001 for PNE protocol and 3007 for PXML protocol.

PTX_SETUP Commands to Activate UCP

This sets up UCP using the Ethernet port on port number 3007:

```
!PTX_SETUP
CONFIG-MPI_SELECT;UCP
CONFIG-PXML_PORT_NUM;3007
PTX_END
```

PTX_SETUP Commands to Activate PXML

This section defines PTX_SETUP commands that are used to activate PXML. These commands are:

`CONFIG-MPI_SELECT;`*`value`*
where *value* is one of the following:
UCP
PXML

`CONFIG-PXML_PORT;value`
where *value* is one of the following:
DISABLE
ETHERNET (if Ethernet is not available, try Adapter if it is enabled)

`CONFIG-PXML_PORT_NUM;value`
where *value* selects the active PXML Port Number.

NOTE: The printer automatically reboots after this command.

This following sets up PXML using the Ethernet port on port number 3007:

```
!PTX_SETUP
CONFIG-MPI_SELECT;PXML
CONFIG-PXML_PORT;ETHERNET
CONFIG-PXML_PORT_NUM;3007
PTX_END
```

Transport

The printer provides a TCP server socket that can be opened by a client as a telnet session. Only one socket is allowed to be open at a time.

The client is responsible for opening and maintaining the session. The connection could be broken at any time, either by a network error or by the printer being powered down. The client must detect that the connection is broken and reconnect to the port. Most TCP stacks do not immediately report that the connection is down, so the client must ‘ping’ the printer on a regular basis.

No separate framing protocol is provided, so the client must continuously parse the incoming data stream for XML messages. The client may send XML messages at any time.

Messaging

All messages are in XML format and are described by a formal XML schema. Command messages are messages that are sent by the client to perform some action or request information.

Response messages are messages that are sent by the printer, either as a response to a command message or as an unsolicited message that is triggered by a change of printer state or an event that occurs during a print job.

All responses are not synchronized. When a command asks for a response, it adds a request to an event queue. A reply to a command may be interspersed with unsolicited messages, although all messages are guaranteed to be well formed XML.

To protect the printer against unauthorized use, commands that affect printer operation require a password to be embedded in the message. This password is the same as the printer's 'telnet' root password.

Message Root

All PXML messages are wrapped with a common root element, named *pxml*.

The standard XML header `<?xml version="1.0"?>` must be sent at the beginning of each message so that the XML parser can discard faulty messages.

Password Security

For inbound messages (messages to the printer), an optional password can be specified. For those commands that require a password, the string is checked against the printer's telnet password; the message is rejected if the password does not match. For commands that do not require a password, the string is ignored. Only commands that change printer setup requires a password. For information inquiry commands a password is not required. By default, the NIC password is zero, or an empty string. To change the NIC password, please refer to the Printronix Auto ID *Network Interface Card User's Manual*.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml password="mypassword">
   .
   .
   .
   </pxml>
```

Example of setting a password through telnet:

```
>set user passwd root mypassword
```

Message Unique ID

For inbound messages (messages to the printer), an optional *requestID* number can be specified. The generated response message will have the same *requestID* attached to it.

None of the unsolicited messages will have any *requestID* attribute. If the request does not include any ID, the response will have the *requestID* of 0. Label, job, and RFID messages are unsolicited and do not include requested ID. The label and job responses use the job ID, not the *requestID*.

The *requestID* value in any command sent to the printer needs to be between 1 and 4294967294. An example of a request sent to the printer with a *requestID* include:

```
<?xml version="1.0"?>
   <pxml requestID="203">
      <info>
         <get type="printer"/>
      </info>
   </pxml>
```

Response coming from the printer:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml requestID="203">
      <info>
         <server pxmlVersion="2.1" />
      </info>
   </pxml>.
```

Status

When certain events occur, the printer can send status messages to the client as responses to the client's request or as unsolicited notification. For example, if a fault occurs, messages are sent to inform the client about the fault condition. By default, the printer does not send unsolicited messages.

The management client can however, select the unsolicited messages to be sent by the printer. This helps reduce the amount of traffic on the management interface.

Job Markers

Printing can cause automatic replies due to the data that is being printed, such as RFID tag data. Markers can be placed in the print stream to indicate the start and end of a job. Print data can be queued so that the markers are processed simultaneously with the print data. When a marker is processed, it sends an automatic reply. This guarantees that a management client listening to the management data stream reads the replies in the order they were processed on the print data stream.

The maximum job number is 65535 (Decimal), for printers released before Oct 2006, and 4294967295 for printers released later.

The host can print the following command to indicate the start of a print job. Any number can be used as the job number to identify the job. This example uses 1234.

```
!PTX_SETUP
PRINTJOB-START;1234
PTX_END
```

When the command is eventually processed, a Job Status Message is sent with a *jobStart* type attribute.

The end of a job can also be indicated with a Job End marker:

```
!PTX_SETUP
PRINTJOB-END;1234
PTX_END
```

When this command is eventually processed, a Job Status Message is sent with a *jobEnd* type attribute.

Get Status Request

The client sends the following command to request the printer to send status.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <get type="fault"/>
      </status>
   </pxml>
```

The type attribute is used to specify the type of status to send, which can include the any of the following:

Table 1. Printer Attribute and Status – Get Status Request

Attribute	Status
engine	Send current engine status.

fault	Send current fault status.
display	Automatic panel display status.

Automatic Status Select

Changes in the printer state and some print operations can cause unsolicited status messages to be sent to the management client. It is possible for the management client to select the unsolicited messages to be sent. This can reduce the amount of traffic on the management interface. The printer will send an ACK message to confirm the status select command.

The printer default for all unsolicited status messages is disabled. To receive any type of status messages, the select message is necessary to enable that type of unsolicited status messages.

To enable or disable unsolicited messages, the management client sends the following command:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <select type="job" enable="true" version="2"/>
      </status >
   </pxml>
```

The type attribute is used to specify the type of status reports that are enabled or disabled. Possible options for type include:

Table 2. Printer Attribute and Status – Automatic Status Select

Attribute	Status
engine	Automatic engine status.
fault	Automatic fault status.
display	Automatic panel display status.
job	Automatic job status. **NOTE:** Only this option can have version 1 or 2 attributes. Version=1 generates version 1 of *rfidTagDetail* element. Version=2 generates version 2 of *rfidTagDetail* element. Refer to “RFIDTagResponseMessage” on page 12.

The enable attribute turns on or off automatic status for the specific type. The valid parameters are Boolean type: false, 0, true, and 1.

NOTE: Job status messages are provided in an unsolicited fashion and cannot be requested individually. The engine and fault status messages can be received by request or as changes happen in the system, if the select command is enabled.

Version attribute is only acceptable for the type=*job*.

Job Status Reply Message

Job Start and Job End markers in the print data stream generate the Job Status Reply message. To receive these automatic start and end of the job, the select command should be sent once to enable the automatic messages as described in “Automatic Status Select” on page 10.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <job type="jobStart">
            <jobDetail id="1234"/>
         </job>
```

```
      </status>
   </pxml>
```

type = The type of the job information, specified by the marker. Possible values include: *jobStart*, *jobEnd*, *rfid*, *label*, *errorLabel*, *partialLabel*, and *errorReport*.

NOTE: The types *errorLabel*, *partialLabel* and *errorReport* are generated for PGL jobs only.

id = The identifier specified by the marker. The *id* attribute only appears for types *jobStart* and *jobEnd*.

The end of the job that is not completed successfully is announced by sending the following message:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <job type="jobEnd">
            <jobDetail id="1234" failure="1"/>
         </job>
      </status>
   </pxml>
```

Label Response Message

The Label Response message is sent out at the end of each page or label. To receive the label status message the select job message should be sent prior to the start of the job or at system start up.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <job type="label">
            <labelDetail failure="0"/>
         </job>
      </status>
   </pxml>
```

The attributes for the *labelDetail* include: *failure* = This is the result of printing a label.

Possible Boolean values are *true*, *false*, 0 (pass) or 1 (fail).

If the emulation encounters errors, error messages will be sent out as they are recognized from the emulation. The error report option for PGL on the front panel should be set to “on” or “host” to receive this message.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <job type="errorReport">
            <jobDetail id="1234" error="135"/>
         </job>
      </status>
   </pxml>
```

If the emulation generates an error page, an error label message is reported by sending the following:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <job type="errorLabel">
            <labelDetail failure="0"/>
         </job>
      </status>
   </pxml>
```

If part of the label does not print, a partial page message is reported by sending the following:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<pxml>
   <status>
      <job type="partialLabel">
         <labelDetail failure="0"/>
      </job>
   </status>
</pxml>
```

If the error is caused by RFID reader error such as RFID read, write error, or ODV verification error, the message will read as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <job type="label">
            <labelDetail failure="1"/>
         </job>
      </status>
   </pxml>
```

RFID Tag Response Message

Whenever an RFID Tag is encoded, a message generates. Depending on the version of the job selected for automatic status select, one of the following messages is returned.

If job version is set to one, *rfidTagDetail* version=1 is sent by the printer. This message is only sent for writing to EPC field.

If job version is set to two, *rfidTagDetail* version=2 is sent by the printer. This message is sent for reading or writing to EPC, TID, and USR fields.

The attributes for the *rfidTagDetail* include:

version = Version of the detail information

failure = A Boolean that indicates if the tag failed.

Version 1 Example:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <job type="rfid">
            <rfidTagDetail version="1" failure="false">
               <property name="type" value="Alien Squiggle 64"/>
               <property name="length" value="64"/>
               <property name="sequence" value="0000"/>
               <property name="epc" value="0123456789ABCDEF"/>
               <property name="prechecked" value="false"/>
               <property name="retries" value="00"/>
               <property name="errorCount" value="00"/>
               <property name="totalTagCount" value="00"/>
               <property name="totalTagFailures" value="00"/>
            </rfidTagDetail>
         </job>
      </status>
   </pxml>
```

The properties for *rfidTagDetail* version 1 include those shown in Table 3.

Table 3. Printer Attribute and Properties - RFID Tag Response Message, Version 1

Attribute	Properties
type	The tag type description.
length	The number bits in the raw tag data.
sequence	The number of RFID operations (read and write) performed including this operation since power up.
epc	The raw tag data in xsd:hexBinary format. (The number of digits depends on the number of tag data bits.)
prechecked	A Boolean that indicates that a tag has been pre-checked.
retries	The number of bad tags that have already been voided while attempting to write this tag data.
errorCount	The number of errors seen when reading or writing this tag.
totalTagCount	Same as the statistics for the menu “Tag Write Cnt” in Hex.
totalTagFailures	Same as the statistics for the menu “Failed Tag Count” in Hex.

Version 2 Example:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <job type="rfid">
            <rfidTagDetail version="2"  failure="false">
               <property name="chain"  value="single"/>
               <property name="length" value="64"/>
               <property name="operation" value="write"/>
               <property name="fieldType" value="EPC"/>
               <property name="totalDatalength"  value="64"/>
               <property name="type"  value="Alien Squiggle 64"/>
               <property name="sequence"  value="0000"/>
               <property name="prechecked"value="false"/>
               <property name="retries"   value="00"/>
               <property name="errorCount"   value="00"/>
               <property name="totalTagCount"    value="00"/>
               <property name="totalTagFailures" value="00"/>
               <property name="data"value="0123456789ABCDEF"/>
            </rfidTagDetail>
         </job>
      </status>
   </pxml>
```

The properties for *rfidTagDetail* include those shown in Table 4.

Table 4. Printer Attribute and Properties - RFID Tag Response Message, Version 2

Attribute	Properties
length	The number of bytes in the data property.
operation	Read or write. This will not be present in the middle or last chain.
fieldType	USR, TID, EPC, and ACS. This will not be present in the middle or last chain.
type	The tag type description. This will not be present in the middle or last chain.
totalDatalength	The total number of bits sent in all of the chained messages related to this operation. This will not be present in the middle or last chain.
sequence	The number of RFID operations (read and write) performed including this operation since power up. This will not be present in the middle or last chain.
prechecked	A Boolean indicating that a tag has been prechecked. This will not be present in the middle or last chain.
retries	The number of bad tags that have already been voided while attempting to write this tag data. This will not be present in the middle or last chain.
errorCount	The number of errors seen when reading or writing this tag. This will not be present in the middle or last chain.
totalTagCount	Same as the statistics for the menu “Tag Write Cnt” in Hex. This will not be present in the middle or last chain.
totalTagFailures	Same as the statistics for the menu “Failed Tag Count” in Hex. This will not be present in the middle or last chain.
data	The raw tag data in xsd:hexBinary format. The number of digits depends on the number of tag data bytes.
chain	First, last, middle, and single. If data is sent in a sequence of chained messages, there is more than one message to contain all data. Single means all data is sent in a single message.

NOTE: With the *chain* attribute, the Host application needs to collect all data in the first, middle, and last message to construct the data that was written to or read from the tag.

Display Response Message

Messages are generated for each line of the display when the message on the operator panel changes, or when requested by a Get Status command.

```
<?xml version="1.0"?>
   <pxml>
      <status>
         <display row="1" text="MENU MODE"/>
      </status>
   </pxml>
```

There are two attributes in the message as shown in Table 5.

Table 5. Printer Attribute and Properties - Display Response Message

Attribute	Properties
row	The row number, for multiple row displays. The first row is ‘1’.
text	The text on the row, as seen by the operator.
Printers released after June 2008 support UTF-8 encoding which allows the control panel to display messages in all supported languages.	

Fault Response Message

When a fault occurs or when requested, the following message generates.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <fault alert="2001" group="0002"/>
      </status>
   </pxml>
```

There are two attributes in the message as shown in Table 6. For more information on alerts, see the full definition in “Alerts” on page 35.

Table 6. Printer Attribute and Properties - Fault Response Message

Attribute	Properties
alert	The alert number.
group	The group number. This can be used to prioritize the fault.

Engine Status Response Message

When engine status changes or when requested, the following message generates.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <status>
         <engine state="idle/>
      </status>
   </pxml>
```

There is one attribute in the message: *state* = The engine state. Possible values include those shown in Table 7.

Table 7. Printer Values and Properties - Engine Status Response Message

Values	Properties
fault	The printer is in a fault state.
idle	Online but not printing.
offline	Offline
pause	Processing print data, but not printing.
printing	Processing and printing print data.
present	Label present.

Statistics

The printer provides print statistic information on request. The statistic information is grouped according to the type of information.

RFID Statistic Get Request

```
<?xml version="1.0"?>
   <pxml>
      <statistics>
         <get type="rfid" />
      </statistics>
   </pxml>
```

There is one attribute in the message: *type* = The type will specify what kind of statistics is requested. Currently the only supported type is *rfid*.

RFID Statistic Clear Request

This request clears the RFID statistics counters in the printer and requires a password. As a result, the RFID read, write, failed, and voided counters start at zero again.

```
<?xml version="1.0"?>
   <pxml>
      <statistics>
         <clear type="rfid" />
      </statistics>
   </pxml>
```

There is one attribute in the message: *type* = The type specifies the kind of statistics requested. Currently the only supported type is *rfid*.

RFID Statistic Response

For RFID get statistics, the response is returned in the following format:

```
<?xml version="1.0"?>
   <pxml>
      <statistics>
         <RFID>
            <property name="TagWriteCount" value="1000"/>
            <property  name="TagFailedCount" value="5"/>
            <property name="TagVoidedCount" value="0"/>
            <property name="TagReadCount" value="30"/>
         </RFID>
      </statistics>
   </pxml>
```

The properties are in decimal format. Descriptions of the fields are provided in Table 8.

Table 8. RFID Statistic Response

Values	Properties
TagWriteCount	Total number of tag writes since the last clear command.
TagFailedCount	Total number of tag writes fail since the last clear command.
TagVoidedCount	Total number of tags voided by the validator.
TagReadCount	Total tag read commands.

Reboot Request

This command reboots the printer. There is no response to this command, and the management connection will be broken when the printer reboots.

The download port is 3010. Any incoming data is in flash file format. These files have a special header that is recognized by Printronix Auto ID printers. This data is loaded into the flash file system of the printer before the printer reboots. For example, the client sends the following command:

```
<?xml version="1.0"?>
   <pxml password="abc123">
      <reboot type="download" timeout="2"/>
   </pxml>
```

There are two attributes in the message:

1. *type* = The type of reboot. Possible options include:
    a. *reset* for normal reboot.
    b. *download* for rebooting in download mode.
    c. *factory* for rebooting with Factory Config as Power-up Config. This changes the power-up Config to be the Factory Config. It also optimizes FLASH. Nothing is deleted from the FLASH.
    d. *optimize*, *reboot and optimize FLASH*, and *optional EMC*. Optimize means the part of unavailable FLASH, that was marked for deleted files, will be available for reuse.
2. *timeout* = An optional timeout value (defaults to 1 minute) defines how long the printer will wait for flash data before it aborts the download, discards any partial data, and reboots.

Information Request

Use this info command to obtain general static information from the printer.

Information Get Request

The get info command requests a particular kind of information based on type.

```
<?xml version="1.0"?>
   <pxml>
      <info>
         <get type="printer"/>
      </info>
   </pxml>
```

There is one attribute in the message: *type* = The type of information. Possible options are *printer*, *server*, and *rfidTagOption*.

Printer Information Response

This printer information message is sent to a host in response to obtain information for *type=printer*.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <info>
         <printer>
            <property name="model"  value="T8304"/>
            <property name="manufacturer" value="Printronix"/>
            <property name="version"   value="V1.09A"/>
            <property name="partNumber"   value="   "/>
            <property name="serialNumber" value="23445634"/>
            <property name="hres"   value="0300"/>
            <property name="vres"   value="0300"/>
```

```
            <property name="ramSizeMB" value="0032"/>
            <property name="flashSizeMB"   value="0008"/>
            <property name="flashAvailable" value="02613959"/>
            <property name="flashAvailableB" value="02613959"/>
            <property name="flashUnAvailableB" value="3959"/>
            <property name="emcSizeMB" value="0016"/>
            <property name="emcAvailableB" value="02613959"/>
            <property name="emcUnavailableB" value="3959"/>
            <property name="displayLines" value="2"/>
            <option name="RFID" state="absent"/>
            <option name="ODV"  state="absent"/>
            <option name="ENET" state="present"/>
            <option name="XNET" state="absent"/>
            <option name="WENET"state="absent"/>
            <option name="WXNET"state="absent"/>
            <option name="NET"  state="present"/>
            <option name="CTHI" state="absent"/>
            <option name="GPIO" state="absent"/>
            <option name="CLOCK"state="absent"/>
        </printer>
    </info>
</pxml>
```

The various property names are described in Table 9. Options names are shown in Table 10.

Table 9. Property Descriptions

Property Name	Unit	Description
model		Printer model name.
manufacturer		Printronix
version		Printer firmware version
partNumber		Printer firmware part number.
serialNumber		Printer unique serial number.
hres	DPI	Horizontal resolution in dots per inch.
vres	DPI	Vertical resolution in dots per inch.
ramSizeMB	Megabyte	Total RAM installed.
FlashSizeMB	Megabyte	Total Flash installed.
flashAvailableB	Bytes	FLASH available.
flashUnavailableB	Bytes	Size of FLASH that is not available, until the printer performs an optimize reboot operation.
emcSizeMB	Megabyte	Total EMC installed (0=not available)
emcAvailableB	Bytes	EMC available
emcUnavailableB	Bytes	Size of EMC that is not available, until the printer performs an optimize reboot operation.
displayLines	Decimal	Maximum number of front panel display lines.

Table 10. Option Descriptions

Property Name	Description
RFID	RFID hardware.
ODV	Barcode verifier hardware.
NET	Network connection available.
ENET	Embedded Network Card.
XNET	External Network Card.
WENET	Wireless Embedded Network Card.
WXNET	Wireless External Network Card.
CTHI	CTHI card.
GPIO	General Purpose IO card.
CLOCK	Real Time Clock.

Server Information Response

This server information message is sent to a host in response to obtain information for *type=server*.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <info>
         <server pxmlVersion="2.1" />
      </info>
   </pxml>
```

rfidTagOption Information Response

This RFID tag ID and names information message is sent to a host in response to get information for *type=rfidTagOption*.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <info>
         <rfidTagOption>
            <option name="Alien Squiggle 64" class="Gen2"/>
            ...
         </rfidTagOption>
      </info>
   </pxml>
```

Valid class names include: Class 0, Class 0+, Class 1, Class Zuma, Class 1.19, and Gen 2. Examples of valid tag type descriptions include those shown in Table 11.

The number of options available for RFID tags depends on the software version and the RFID reader installed inside the printer. The tag names are unique for all printers.

Table 11. Valid Tag Descriptions

Alien Squiggle 64	RAFUCode 477 96	TI Dallas G2
Alien Squiggle 96	Impinj Zuma 64	Avery AD220 G2
Alien M-TAG 64	ImpZ Prop 96	Imp Banjo G2
Alien M-TAG 96	EPC Gen2 96	Imp Prop G2
RAF Omni 313 64	Omron Wave	Alien Squig G2
RAF Omni 432 96	Rafsec 478	RAF Square G2
Matrics1020 64	X-Ident PH58 96	RAF Short G2
Matrics1020 96	Avery AD410 IN	X-Ident PH60 96
Matrics2020 64	Avery BL	ImpZ Triflex 96
Matrics2020 96	Alien Itag 96	Flex Wing
RAFUCode 450 96	Alien SupS 96	

Printer ACK or Error Response

Some commands do not return an immediate response; in this case the printer will send an ACK message indicating that it received the request. For example, the status select requests will enable or disable a particular kind of unsolicited message from the printer. After the printer receives this kind of message sends an ACK message to the host, indicating that the message was received.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <ack result="success" />
   </pxml>
```

If there is a syntax error in the XML request message, the following error message is sent:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <ack result="fail">
         <details row="1" column="0" message="Invalid Element"/>
      </ack>
   </pxml>
```

Some of the error messages include: Invalid Element, Invalid Attribute, Invalid Data, Invalid Password, and Command (not available for this printer model).

PXML Inactive Error Response

This message indicates that PXML is inactive and the PXML host message cannot be processed. To activate PXML refer to “PXML Selection” on page 7.

This message is only received for PXML version 1.0. For version 2, PNE and PXML can coexist in the printer and UCP does not return any error messages, if it gets PXML messages on its port.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <ack result="fail">
         <details message="PXML port error"/>
      </ack>
   </pxml>
```

Printer Setting

The printer setting is different from info because it is not static information. The setting information can be requested from the printer. Based on the type attribute value, a different kind of response is sent to the host. The user can also set some of the settings. Each set and get type is described in the following sections.

Printer Setting Get Request

The following request message generates the property response that has information related to the type requested in this message.

```
<?xml version="1.0"?>
   <pxml>
      <setting>
         <get type="network"\>
      </setting>
   </pxml>
```

Valid options for type include *network*, *realTime*, *rfidSetting*, and *pglErrorReport*.

Printer Setting Select Request for rfidSetting

The following select request message for *rfidSetting* generates unsolicited *rfidSetting* response messages anytime there is a change in RFID setting within the printer.

```
   <pxml>
      <setting>
         <select type="rfidSetting" enable="true"\>
      </setting>
   </pxml>
```

The only valid option for type is *rfidSetting*.

Time Setting Response

This printer setting message is sent to a host in response to obtain settings for *type=time*. An error message is sent back if the time setting is not available in the printer. The value is in XML time format.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <setting>
         <property name="realTime" value="2005-08-10T08:20:10"/>
      </setting>
   </pxml>
```

Time Set Request

An error message is sent back if the time setting is not available in the printer. This request message sets the new time for the printer and requires a password.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml password="abc123">
      <setting>
         <set name="realTime" value="2005-08-10T08:20:10"/>
      </setting>
   </pxml>
```

PGL Error Handling Response

This printer setting message is sent to the host in response to obtain the setting request for *type=pglErrorReport*. The valid parameters are *on*, *off*, *debugMode*, *fault*, and *host*.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <setting>
         <property name="pglErrorReport" value="off"/>
```

```
      </setting>
   </pxml>
```

PGL Error Handling Set Request

To disable printing of the PGL error page, enter the following PXML command to change the front panel setting for PGL error handling.

NOTE: Set this for every job or save the configuration so the setting does not change.

```
<?xml version="1.0"?>
   <pxml>
      <setting>
         <set name="pglErrorReport" value="host"\>
      </setting>
   </pxml>
```

Possible choices for *value* = (options) *off*, *on*, *debugMode*, *fault*, or *host* as described in Table 12.

Table 12. Choices for PGL Error Handling Setting

Value options	Description
off	No error message prints or is sent to the host.
on	An error message prints when an error occurs.
debug Mode	The test file prints as text with the error message under the command line which has the error.
fault	The error message prints, “IGP/PGL Error” displays on the front panel, and the printer stops printing. Clear the error by pressing the ONLINE key before resuming printer operation.
host	Sends the error message to the host.

Network Setting Response

More than one network connection may be present, such as embedded (ENET), wireless embedded (WENET), external (XNET) and wireless external (WXNET) options. This printer setting message is sent to a host to obtain setting requests for *type=network*. Only the network connections (ENET, WENET, XNET, WXNET) that are present in the system have descriptions included in this message.

Printer name, location, and description are per stack. ENET and WENET share the same stack and they share the same name, location and description, called EMBEDDED properties. Also, XNET and WXNET share the same stack, and are EXTERNAL properties. If the printer has both the external and embedded NIC, then the user should try to set the same properties for both EMBEDDED and EXTERNAL, since they share the same physical printer.

An ACK message is sent back to indicate whether the request is successful or unsuccessful.

```
<?xml version="1.0"?>
   <pxml>
      <setting>
         <network type="WENET">
            <property name="macAddress" value="2345679"/>
            <property name="DHCPenable" value="1"/>
            <property name="ipAdrress" value="255.255.255.255"/>
            <property name="subnetMask" value="255.255.255.255"/>
            <property name="gatewayAddress" value="255.255.255.255"/>
         </network>
         <network type="ENET">
            <property name="macAddress" value="2345679"/>
            <property name="DHCPenable" value="1"/>
```

```
            <property name="ipAdrress" value="255.255.255.255"/>
            <property name="subnetMask" value="255.255.255.255"/>
            <property name="gatewayAddress" value="255.255.255.255"/>
         </network>
         <network type="EMBEDDED">
            <property name="dnsAddress" value="255.255.255.255"/>
            <property name="printerName"value="name"/>
            <property name="description"value="description string"/>
            <property name="location"  value="manufacturing"/>
         </network>
         <network type="EXTERNAL">
            <property name="dnsAddress" value="255.255.255.255"/>
            <property name="printerName"value="name"/>
            <property name="description"value="description string"/>
            <property name="location"  value="manufacturing"/>
         </network>
         <network type="pxml">
            <property name="port"   value="3007"/>
         </network>
      </setting>
   </pxml>
```

Network Set Request

The network connection needs to set one connection at a time. This request requires a password.

IMPORTANT **The application needs to send a reboot request to activate the network setup changes. An ACK message is sent back as a response. Longer values for name, location, and description are truncated. The addresses are all valid IP address and they will not be verified. It is recommended to check for network setting and verification of the desired setup before rebooting.**

The maximum number of characters for name, location, and description is 29 one byte characters. Only ASCII characters are recommended for these fields. Do not use characters two or more bytes long.

The value of *DHCPenable* is a Boolean type.

```
<?xml version="1.0"?>
<pxml password="abc123">
   <setting>
      <network type="WENET">
         <set name="DHCPenable" value="0"/>
         <set name="ipAddress"   value="255.255.255.255"/>
         <set name="subnetMask"  value="255.255.255.255"/>
         <set name="gatewayAddress"value="255.255.255.255"/>
      </network>
   </setting>
</pxml>
```

Example: Set The PXML port

```
<?xml version="1.0"?>
<pxml password="abc123">
   <setting>
      <network type="pxml">
         <set name="port" value="3007"/>
      </network>
   </setting>
</pxml>
```

RFID Setting Response

This printer setting message is sent to a host in a response to obtain setting request for type=rfidSetting.

```
<?xml version="1.0"?>
   <pxml>
      <setting>
         <rfidSetting>
            <property name="readerEnable" value="1"/>
            <property name="tagName"   value=""/>
            <property name="EPCdataSize"   value=""/>
            <property name="USRdataSize"   value=""/>
            <property name="TIDdataSize"   value=""/>
            <property name="RSVdataSize"   value=""/>
            <property name="error Handling"   value=""/>
            <property name="printerRetry" value=""/>
            <property name="readerRetry"   value=""/>
            <property name="maxRetryError"value=""/>
         </rfidSetting>
      </setting>
   </pxml>
```

All numbers are in decimal value.

Valid values for *errorHandling* include *none*, *stop*, and *overstrike*.

RFID Set Request

This is a host request to change the RFID setup in the printer; it requires a password. An ACK message is sent back to show whether the request was successful or unsuccessful.

The user can ask for *rfidSetting* to see if the new requested values are in effect. All the numbers are in decimal values.

NOTE: The *TagName* value must match one of the RFID options from the printer. This setting is case sensitive. Some examples of valid tagName values are included in Table 11.

```
<?xml version="1.0"?>
<pxml  password="abc123">
   <setting>
      <rfidSetting>
         <set name="readerEnable"   value="1"/>
         <set name="tagName"  value="Alien Squiggle 96"/>
         <set name="errorHandling"  value="overstrike"/>
         <set name="printerRetry"   value="10"/>
         <set name="readerRetry" value="4"/>
         <set name="maxRetryError" value="1"/>
      </rfidSetting>
   </setting>
</pxml>
```

The range of the valid numbers for *printerRetry* and *readerRetry* are the same as the front panel setting. *maxRetryError* has a Boolean value, matching the front panel setting.

Printer Configuration

Load Configuration Request

The following request message generates the property response that contains information related to the type requested in this message.

```
<?xml version="1.0"?>
   <pxml>
      <configSetting>
         <load name="1"/>
      </configSetting>
   </pxml>
```

Valid options for *name* are 1-8, factory, and *powerUp*. The printer sends an ACK message to confirm. The "Loading Configuration Failed" error message is sent if the requested configuration is not present in the printer.

Save Configuration Request

This requests that the current configuration be saved to the given configuration and as the power up configuration if the power up flag is set. If the given configuration is *auto* then the current configuration is saved to the last loaded configuration unless that configuration is the factory configuration. If the last loaded configuration is the factory configuration, then the configuration is saved to the first available unused configuration. If no configurations are available, then an error response is sent.

```
<?xml version="1.0"?>
   <pxml>
      <configSetting>
         <save name="1" powerUp="1"/>
      </configSetting>
   </pxml>
```

Valid options for name are 1-8 or *auto*. Field *powerUp* has a Boolean value.

Value 1 or *true* means make the saved configuration as the power up configuration. Value 0 or *false* means save the configuration without changing the power up.

Save Configuration Response

This message is a response to saving configuration. It returns the saved configuration number or an error.

```
<?xml version="1.0"?>
   <pxml>
      <configSetting>
         <property name="savedConfig" value="1"/>
      </configSetting>
   </pxml>
```

Valid options for value is 1-8 or *failed*.

Printer Storage

IMPORTANT: EMC (Extended Memory Cartridge) applies to SD storage as well.

Flash or EMC Storage Request

This printer setting message is sent to a host in response to obtain setting request for *type=flash* or *emc*.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
```

```
      <storage type="flash">
         <get type="unprotected"/>
      </storage>
   </pxml>
```

Flash or EMC File System Directory Response

This printer setting message is sent to a host in response to obtain information setting for *type=unprotected*. The file size is in hex values. This will be the list of unprotected files only.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <storage>
         <flash type="unprotected">
            <file name="form1.pgl"  size="000034"/>
            <file name="form2.pgl"  size="000124"/>
         </flash>
      </storage>
   </pxml>
```

For EMC the response will be:

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <storage>
         <emc type="unprotected">
            <file name="form1.pgl"  size="000034"/>
            <file name="form2.pgl"  size="000124"/>
         </emc>
      </storage>
   </pxml>
```

If no EMC is available in the printer, the following will be returned. Since EMC is optional storage, no error message generates.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml>
      <storage>
      </storage>
   </pxml>
```

Flash or EMC Delete File Request

The request is sent to delete a single unprotected flash file. This request does require a password. This command also accepts the “*.ext”, to delete all files with a given extension.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml password="abc123" >
      <storage type="flash">
         <delete name="form1.pgl"/>
      </storage>
   </pxml>
```

To delete a file from EMC set type=EMC.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml password="abc123" >
      <storage type="emc">
         <delete name="form1.pgl"/>
      </storage>
```

```
   </pxml>
```

If the file is not present in the file system, no error message generates. The file is deleted.

Flash or EMC Read File Request

The request is sent to read or upload a single unprotected file from the given storage type. This request does not require a password.

```
<?xml version="1.0" encoding="UTF-8"?>
   <pxml password="password">
      <storage type="flash">
         <read name="roll.frm"/>
      </storage>
   </pxml>
```

The response would be:

```
<?xml version="1.0" encoding="UTF-8"?>
   <storage type="flash">
          <file name="filename.frm" size="68">
             <base64Data>data</base64Data>
          </file>
   </storage>
</pxml>
```

FLASH or EMC Read File Response

The response to reading file request will return the file content in base64Binary format. The file size given is hex.

```
<?xml version="1.0" encoding="UTF-8"?>
   <storage type="flash">
      <file name="filename.frm" size="68">
         <base64Data>data</base64Data>
      </file>
   </storage>
</pxml>
```

Alerts

General Alerts

Table 13. General Alerts

Printer State	Alert	Group
No Fault	0000	NoFault (0)
Paper Out	2001	mediaInput (2)
Paper Jam	2002	mediaPath (4)
Host System Requesting Operator's Attention	2006	intervention (9)
Serial Interface Buffer Overrun	2020	intervention (9)
Serial Line Parity Error	2023	intervention (9)
Serial Interface Framing Error	2025	intervention (9)
Paper Out Error Has Timed Out	2031	mediaInput (2)
Paper Jam Error Has Timed Out	2032	mediaPath (4)
Ribbon Stall Error Has Timed Out	2034	marker (5)
Buffer Overflow	2041	intervention (9)
Printer Is Hot	2060	intervention (9)
Flash File System Is Full	2219	warning (0)
Flash File System Needs More DRAM	2220	intervention (9)
Flash File Overwrite Error	2221	Intervention (9)
Flash File System Is Invalid	2222	intervention (9)
Flash File System Write Error	2223	intervention (9)
Twinax Graphic Check Error	2224	intervention (9)
Bad VFU Channel	2300	intervention (9)
Barcode Fails Specification	2301	barcode (7)
PPM Generated Fault	2302	intervention (9)

Thermal Alerts

Table 14. Thermal Alerts

Printer State	Alert	Group
Ribbon Fault	2400	marker (5)
Print Head Is Hot	2401	intervention (9)
EC Software Fail	2402	intervention (9)

Printer State	Alert	Group
Gap Is Not Detected	2404	label (11)
Ribbon Installed in Direct Mode	2405	marker (5)
Cutter Has Fault	2406	cutter (6)
Barcode Fails Specification	2407	barcode (7)
Missing Barcode	2408	scanner (8)
Decodability Fault	2410	scanner (8)
Defects Failure	2411	scanner (8)
Percent Decode Fault	2412	scanner (8)
Symbol Contrast Fault	2413	scanner (8)
Quiet Zones Failure	2414	barcode (7)
Encodation Fault	2415	barcode (7)
Calibration Warning	2416	intervention (9)
Signal Clipping	2417	intervention (9)
Print Head Is Open	2418	marker (5)
Head Power Fail	2420	intervention (9)
Power Supply 24V Fail	2421	intervention (9)
Power Supply 40V Fail	2422	intervention (9)
Ribbon Is Broken	2423	marker (5)
Ribbon Load Bad	2424	marker (5)
Barcode Checksum Failure	2430	barcode (7)
Verifier Data Invalid	2431	scanner (8)
Verifier Motor Speed Failure	2432	intervention (9)
Verifier EC Fault	2433	intervention (9)
Verifier Not Installed	2434	scanner (8)
Verifier Not Enabled	2435	scanner (8)
Battery Voltage Too Low	2442	powerCart (12)

RFID Alerts

Table 15. RFID Alerts

Printer State	Alert	Group
RFID Maximum Tag Retry	2439	rfid (13)

Printer State	Alert	Group
RFID Communication Error	2441	rfid (13)
RFID Tag Failed	2443	rfid (13)
RFID Max. Tag Retry Timeout	2444	rfid (13)
RFID Tag Failed Timeout	2445	rfid (13)
RFID Data Error	2446	rfid (13)
RFID Read Only Tag	2447	rfid (13)
RFID Lock not supported	2448	rfid (13)
RFID MAX RETRY Dumping Form	2845	rfid (13)

Extended Memory (EMC) Alerts

Table 16. Extended Memory (EMC) Alerts

Printer State	Alert	Group
Invalid EMC Installed	2833	intervention (9)
EMC Not Found	2838	intervention (9)
EMC Write Err	2839	intervention (9)
EMC Removed	2840	intervention (9)
EMC Invalid Type	2841	intervention (9)

Line Matrix Alerts

Table 17. Line Matrix Alerts

Printer State	Alert	Group
Paper Out	2001	mediaInput (2)
Paper Jam	2002	mediaPath (4)
Ribbon Ink Out	2005	consumables (10)
Power Stacker Jammed	2017	mediaOutput (3)
Power Stacker Fault	2019	mediaOutput (3)
Ribbon Ink Out Timeout	2035	consumables (10)
Firmware Error	2045	intervention (9)
Hammer Coil Open	2056	intervention (9)
Shuttle Jam Error	2058	intervention (9)
Exhaust Fan Fault	2062	intervention (9)

Printer State	Alert	Group
Hammer Bank Fan Fault	2065	intervention (9)
Power Supply Voltage Failure	2081	intervention (9)
Power Supply 8.5V Fail	2082	intervention (9)
Intake Fan Fault	2083	intervention (9)
Power Supply 48V Fail	2084	intervention (9)
Controller Voltage Failure	2085	intervention (9)
Controller 15V	2086	intervention (9)
Controller 23.5V	2088	intervention (9)
Ribbon Stall	2089	marker (5)
Cover/Door Open	2090	intervention (9)
Ribbon Driver Circuit	2092	marker (5)
Upper Driver Short	2101	intervention (9)
Lower Driver Short	2102	intervention (9)
Coil Hot	2107	intervention (9)
Stack Overflow	2110	mediaOutput (3)
Stack Underflow	2111	mediaOutput (3)
Undefined OpCode	2112	intervention (9)
Protected Instruction	2113	intervention (9)
Illegal Operand Access	2114	intervention (9)
Illegal Instruction Access	2115	intervention (9)
Illegal External Bus Access	2116	intervention (9)
A to D Overrun	2117	intervention (9)
Undefined Interrupt	2118	intervention (9)
TCB Corrupted	2119	intervention (9)
Access Null Pointer	2120	intervention (9)
Paper Not at Speed (Impact)	2121	intervention (9)
Paper Not Scheduled	2122	intervention (9)
Paper Busy Too Long	2123	intervention (9)
Paper FIFO Overflow	2124	intervention (9)
Paper FIFO Underflow	2125	intervention (9)
Paper Feed Bad Table	2126	intervention (9)

Printer State	Alert	Group
Paper Feed Illegal State	2127	intervention (9)
Paper Feed Invalid Command	2128	intervention (9)
Paper Feed Invalid Parameter	2129	intervention (9)
Paper Feed Incompletely Energized	2130	intervention (9)
Paper Feed Unexpected Interrupt	2131	intervention (9)
Ribbon Invalid Command	2132	intervention (9)
Ribbon Invalid State	2133	intervention (9)
Platen Invalid Command	2134	intervention (9)
Platen Invalid State	2135	intervention (9)
Platen Invalid Parameter	2136	intervention (9)
Shuttle Invalid Command	2137	intervention (9)
Shuttle Invalid Parameter	2138	intervention (9)
Shuttle Over Speed	2139	marker (5)
EC Stopped	2200	intervention (9)
Driver Circuit Bad	2201	intervention (9)
Hammer Bank Not Installed	2202	intervention (9)
Hammer Coil Bad	2203	intervention (9)
Ribbon Ink Out	2226	consumables (10)

Validate Requests and Responses using XML Schema

Use the following declaration to validate a PXML command instance against the PXML schema.

Below is an example of a normal command that a client sends to the printer:

```
<?xml version="1.0"?>
   <pxml>
      <info>
         <get type="printer"/>
      </info>
   </pxml>
```

Since the printer does not perform validation using a schema file, there is no need to add a schema location definition for normal operation. If there is a question as to whether an XML command matches the schema, use a validating parser to test the command. The command must be modified to add a schema declaration as shown:

```
<?xml version="1.0"?>
   <pxml  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="http://www.printronix.com/pxml"
          xsi:schemaLocation="http://www.printronix.com/pxml  pxml.xsd">
      <info>
```

```
      <get type="printer"/>
   </info>
</pxml>
```

3 Formal PXML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema
   elementFormDefault="qualified"
   targetNamespace="http://www.printronix.com/pxml"
   xmlns="http://www.printronix.com/pxml"
   xmlns:doc="http://www.printronix.com/pxml/doc" version="1.0">

   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            XML Schema for Printronix Device Management Language (PXML)
         </doc:purpose>
         <doc:copyright>
            Copyright (C) 2005 Printronix, Inc. All Rights Reserved
         </doc:copyright>
         <doc:version>"1.2 03"</doc:version>
      </xsd:documentation>
   </xsd:annotation>

   <xsd:element name="pxml">
      <xsd:annotation>
         <xsd:documentation>
            <doc:purpose>
               Root element for all PSXML messages.
               This element wraps all messages to and from the printer.
            </doc:purpose>
            <doc:type toplevel="true" node="container"/>
            <doc:parameter name="password">
               Password supplied on messages sent to the printer. This must
               match the printer's telnet root password. Required for
               messages that could affect printer operation.
            </doc:parameter>
            <doc:parameter name="requestID">
               This is a number that will be sent back with the corresponding
               response.
            </doc:parameter>
         </xsd:documentation>
      </xsd:annotation>
   <xsd:complexType>
      <xsd:choice minOccurs="1" maxOccurs="1">
         <xsd:element name="ack" type="ackType"/>
         <xsd:element name="info" type="infoType"/>
         <xsd:element name="status" type="statusType"/>
         <xsd:element name="statistics" type="statisticsType"/>
         <xsd:element name="reboot" type="rebootType"/>
         <xsd:element name="setting" type="settingType"/>
         <xsd:element name="configSetting" type="configSettingType"/>
         <xsd:element name="storage" type="storageType"/>
      </xsd:choice>
```

```
      <xsd:attribute name="password" type="xsd:string" use="optional"/>
      <xsd:attribute name="requestID" type="xsd:integer" use="optional"/>
   </xsd:complexType>
</xsd:element>

<xsd:complexType name="ackType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports acknowledgement of a command.
         </doc:purpose>
         <doc:parameter name="details"> error details.
         </doc:parameter>
         <doc:type node="reply"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="0" maxOccurs="1">
      <xsd:element name="details" type="errorDetailsType"/>
   </xsd:choice>
   <xsd:attribute name="result" use="required">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="success"/>
            <xsd:enumeration value="fail"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>
   <xsd:complexType name="errorDetailsType">
      <xsd:annotation>
         <xsd:documentation>
            <doc:purpose>
               Object that contails the error details.
            </doc:purpose>
            <doc:type node="response" autonomous="true"/>
         </xsd:documentation>
      </xsd:annotation>
      <xsd:attribute name="row" type="xsd:integer" use="optional"/>
      <xsd:attribute name="column" type="xsd:integer" use="optional"/>
      <xsd:attribute name="message" type="xsd:string" use="required"/>
   </xsd:complexType>

   <xsd:complexType name="rebootType">
      <xsd:annotation>
         <xsd:documentation>
            <doc:purpose>
               Reboot the printer with the option of downloading flash file.
            </doc:purpose>
         <doc:type node="request"/>
         <doc:restriction> Password required.
         </doc:restriction>
         <doc:parameter name="type">
            Specifies the reboot type. This can be either a normal reset, or
            can reboot into download mode.
         </doc:parameter>
         <doc:parameter name="timeout">
```

```
            Specifies the maximum amount of time, in minutes, that the
            printer will wait between flash data characters before aborting
            download and rebooting. This only applies when 'type' is
            'download'.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type" use="optional" default="reset">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="reset"/>
            <xsd:enumeration value="download"/>
            <xsd:enumeration value="factory"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="timeout" type="xsd:integer" use="optional"
      default="1"/>
</xsd:complexType>

<xsd:complexType name="infoType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Commands and responses for getting printer information.
         </doc:purpose>
         <doc:type node="container"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="1">
      <xsd:element name="get" type="getInfoType"/>
      <xsd:element name="printer" type="printerInfoType"/>
      <xsd:element name="server" type="serverInfoType"/>
      <xsd:element name="rfidTagOption" type="rfidTagOptionInfoType"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="getInfoType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests device info.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:response type="printerInfoType"/>
         <doc:response type="serverInfoType"/>
         <doc:response type="rfidTagOptionInfoType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="printer"/>
            <xsd:enumeration value="server"/>
            <xsd:enumeration value="rfidTagOption"/>
         </xsd:restriction>
```

```
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="printerInfoType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports management service info.
         </doc:purpose>
         <doc:type node="reply"/>
         <doc:parameter name="option"> Describes a printer option.
         </doc:parameter>
         <doc:parameter name="property"> Describes a printer property.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="option" type="printerInfoOptionType"/>
      <xsd:element name="property" type="propertyType"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="printerInfoOptionType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports printer options.
         </doc:purpose>
         <doc:type node="reply"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="name" type="xsd:string" use="required"/>
   <xsd:attribute name="state" use="required">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="absent"/>
            <xsd:enumeration value="present"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="rfidTagOptionInfoType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports supported tag options.
         </doc:purpose>
         <doc:type node="reply"/>
         <doc:parameter name="option"> Describes a printer option.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="option" type="optionType"/>
```

```
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="optionType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object holds option (name/value pairs).
         </doc:purpose>
         <doc:type node="reply"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="name" type="xsd:string" use="required"/>
   <xsd:attribute name="class" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="serverInfoType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports management service info.
         </doc:purpose>
         <doc:type node="reply"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="pxmlVersion" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="propertyType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object holds properties (name/value pairs).
         </doc:purpose>
         <doc:type node="reply"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="name" type="xsd:string" use="required"/>
   <xsd:attribute name="value" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="statusType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Printer status commands and responses.
         </doc:purpose>
         <doc:type node="container"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="1">
      <xsd:element name="get" type="getStatusType"/>
      <xsd:element name="select" type="selectStatusType"/>
      <xsd:element name="display" type="displayStatusType"/>
      <xsd:element name="fault" type="faultStatusType"/>
      <xsd:element name="engine" type="engineStatusType"/>
```

```
      <xsd:element name="job" type="jobStatusType"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="getStatusType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Request the printer to send current status.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:parameter name="type">
            Indicates the type of status to be returned.
         </doc:parameter>
         <doc:parameter name="version">
             Indicates the version of response for job status.
         </doc:parameter>
         <doc:response type="displayStatusType"/>
         <doc:response type="faultStatusType"/>
         <doc:response type="engineStatusType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type" use="required">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="display"/>
            <xsd:enumeration value="fault"/>
            <xsd:enumeration value="engine"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="version" type="xsd:integer" use="optional"/>
</xsd:complexType>

<xsd:complexType name="selectStatusType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Request the printer select autonomous operation.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:parameter name="type">
             Indicates the type of autonomous status.
         </doc:parameter>
         <doc:parameter name="enable">
             Indicates if autonomous responses should be sent.
         </doc:parameter>
         <doc:response type="ackType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="display"/>
            <xsd:enumeration value="fault"/>
            <xsd:enumeration value="engine"/>
```

```
            <xsd:enumeration value="job"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="enable" type="xsd:boolean"/>
</xsd:complexType>

<xsd:complexType name="displayStatusType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object that contains operator panel display text.
         </doc:purpose>
         <doc:type node="response" autonomous="true"/>
         <doc:parameter name="text"> The displayed text.
         </doc:parameter>
         <doc:parameter name="row">
             Which text row, for multi-line display.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="text">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:minLength value="1"/>
            <xsd:maxLength value="16"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="row" type="xsd:integer" use="optional"/>
</xsd:complexType>

<xsd:complexType name="faultStatusType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object that contains fault information.
         </doc:purpose>
         <doc:type node="response" autonomous="true"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="alert" type="xsd:integer" use="required"/>
   <xsd:attribute name="group" type="xsd:integer" use="required"/>
</xsd:complexType>

<xsd:complexType name="engineStatusType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object that contains engine status.
         </doc:purpose>
         <doc:type node="response" autonomous="true"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="state" use="required">
      <xsd:simpleType>
```

```
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="fault"/>
            <xsd:enumeration value="idle"/>
            <xsd:enumeration value="offline"/>
            <xsd:enumeration value="pause"/>
            <xsd:enumeration value="printing"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="jobStatusType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object that contains job status.
         </doc:purpose>
         <doc:parameter name="type"> The type of job status.
         </doc:parameter>
         <doc:parameter name="jobDetail"> The job details
         </doc:parameter>
         <doc:parameter name="labelDetail"> The label details
         </doc:parameter>
         <doc:parameter name="rfidTagDetail"> The rfid details
         </doc:parameter>
         <doc:type node="response" autonomous="true"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="1">
      <xsd:element name="jobDetail" type="jobDetailType"/>
      <xsd:element name="labelDetail" type="labelDetailType"/>
      <xsd:element name="rfidTagDetail" type="rfidTagDetailType"/>
   </xsd:choice>
   <xsd:attribute name="type">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="jobStart"/>
            <xsd:enumeration value="jobEnd"/>
            <xsd:enumeration value="label"/>
            <xsd:enumeration value="rfid"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="jobDetailType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object that contains details of the job.
         </doc:purpose>
         <doc:type node="response" autonomous="true"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="id" type="xsd:integer" use="required"/>
</xsd:complexType>
```

```
<xsd:complexType name="rfidTagDetailType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object that contains RFID tag details.
         </doc:purpose>
         <doc:type node="response" autonomous="true"/>
         <doc:parameter name="property"> Describes an RFID Tag property.
         </doc:parameter>
         <doc:parameter name="version">
            Describes RFID Tag telemetry version.
         </doc:parameter>
         <doc:parameter name="failure">
            Indicates if RFID Tag passed or failed.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="property" type="propertyType"/>
   </xsd:choice>
   <xsd:attribute name="version" type="xsd:integer" use="required"/>
   <xsd:attribute name="failure" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="labelDetailType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object that contains label or page detail information.
         </doc:purpose>
         <doc:type node="response" autonomous="true"/>
         <doc:parameter name="property"> Describes a label property.
         </doc:parameter>
         <doc:parameter name="failure"> Indicates if label passed or failed.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="property" type="propertyType"/>
   </xsd:choice>
   <xsd:attribute name="failure" type="xsd:boolean" use="required"/>
</xsd:complexType>

<xsd:complexType name="statisticsType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Commands and responses for getting and clearing statistics.
         </doc:purpose>
         <doc:type node="container"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="1">
      <xsd:element name="get" type="getStatisticsType"/>
      <xsd:element name="clear" type="clearStatisticsType"/>
```

```
      <xsd:element name="RFID" type="RFIDstatisticsType"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="getStatisticsType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests statistics for a given type.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:response type="RFIDstatisticsType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="RFID"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="clearStatisticsType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests to clear statistics for a given type.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:response type="ackType">
            Returns an 'ack' element specifying the success of the set
            operation.
         </doc:response>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="RFID"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="RFIDstatisticsType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports management service RFID statistics. This will
            match front panel menu setting for RFID statistics.
         </doc:purpose>
         <doc:type node="reply"/>
         <doc:parameter name="property"> Describes a RFID statistic.
         </doc:parameter>
      </xsd:documentation>
```

```
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="property" type="propertyType"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="settingType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Commands and responses for getting a group of printer setting.
         </doc:purpose>
         <doc:type node="container"/>
         </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="get" type="getSettingType"/>
      <xsd:element name="set" type="setSettingType"/>
      <xsd:element name="select" type="selectSettingType"/>
      <xsd:element name="realTime" type="propertyType"/>
      <xsd:element name="network" type="networkSettingType"/>
      <xsd:element name="rfidSetting" type="rfidSettingType"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="getSettingType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests a group of device setting.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:response type="propertyType"/>
         <doc:response type="networkSettingType"/>
         <doc:response type="rfidSettingType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="realTime"/>
            <xsd:enumeration value="network"/>
            <xsd:enumeration value="rfidSetting"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="networkSettingType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports network setting.
         </doc:purpose>
         <doc:type node="container"/>
         <doc:parameter name="type">
```

```
            Describes a network connection type.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="set" type="setSettingType"/>
      <xsd:element name="property" type="propertyType"/>
   </xsd:choice>
   <xsd:attribute name="type">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="ENET"/>
            <xsd:enumeration value="WENET"/>
            <xsd:enumeration value="XNET"/>
            <xsd:enumeration value="WXNET"/>
            <xsd:enumeration value="EMBEDDED"/>
            <xsd:enumeration value="EXTERNAL"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="rfidSettingType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports rfid setting.
         </doc:purpose>
         <doc:type node="container"/>
         <doc:parameter name="property"> Describes a printer property.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="set" type="setSettingType"/>
      <xsd:element name="property" type="propertyType"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="selectSettingType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Request the printer select autonomous operation.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:parameter name="type">
            Indicates the type of autonomous setting.
         </doc:parameter>
         <doc:parameter name="enable">
            Indicates if autonomous responses should be sent.
         </doc:parameter>
         <doc:response type="ackType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type">
```

```
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="rfidSetting"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="enable" type="xsd:boolean"/>
</xsd:complexType>

<xsd:complexType name="setSettingType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports printer options.
         </doc:purpose>
         <doc:type node="request"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="name" type="xsd:string" use="required"/>
   <xsd:attribute name="value" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="configSettingType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Commands and responses for saving and loading configuration.
         </doc:purpose>
         <doc:type node="container"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="1">
      <xsd:element name="load" type="configSettingLoadType"/>
      <xsd:element name="save" type="configSettingSaveType"/>
      <xsd:element name="property" type="propertyType"/>
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="configSettingLoadType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests loading of a saved configuration.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:response type="ackType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="name" use=”required”>
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
            <xsd:enumeration value="3"/>
            <xsd:enumeration value="4"/>
            <xsd:enumeration value="5"/>
```

```
            <xsd:enumeration value="6"/>
            <xsd:enumeration value="7"/>
            <xsd:enumeration value="8"/>
            <xsd:enumeration value="factory"/>
            <xsd:enumeration value="powerUp"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="configSettingSaveType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests a group of device setting.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:response type="configSettingType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="name" use=”required”>
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
            <xsd:enumeration value="3"/>
            <xsd:enumeration value="4"/>
            <xsd:enumeration value="5"/>
            <xsd:enumeration value="6"/>
            <xsd:enumeration value="7"/>
            <xsd:enumeration value="8"/>
            <xsd:enumeration value="auto"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:attributename="powerUp"type="xsd:boolean"use="required"/>
</xsd:complexType>

<xsd:complexType name="storageType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Commands and responses for getting and deleting flash files.
         </doc:purpose>
         <doc:type node="container"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="1">
      <xsd:element name="get" type="getStorageType"/>
      <xsd:element name="delete" type="deleteStorageType"/>
      <xsd:element name="flash" type="flashStorageType"/>
      <xsd:element name="emc" type="emcStorageType"/>
      <xsd:element name="file" type="fileStorageType"/>
   </xsd:choice>
   <xsd:attribute name="type">
      <xsd:simpleType>
```

```
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="flash"/>
            <xsd:enumeration value="emc"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="getStorageType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests directory of a particular type.
         </doc:purpose>
         <doc:type node="request"/>
         <doc:parameter name="type">
            Indicates the type of storage information to be returned.
         </doc:parameter>
         <doc:response type="flashStorageType"/>
         <doc:response type="emcStorageType"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="type">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="unprotected"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="deleteStorageType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests to delete an unprotected file.
         </doc:purpose>
         <doc:type node="request"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="readStorageType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object requests to upload an unprotected file from printer.
         </doc:purpose>
         <doc:type node="request"/>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="flashStorageType">
   <xsd:annotation>
```

```
      <xsd:documentation>
         <doc:purpose>
            Object reports the flash directory of some type.
         </doc:purpose>
         <doc:type node="reply"/>
         <doc:parameter name="type">
            Indicates the type of storage information to be returned.
         </doc:parameter>
         <doc:parameter name="file"> Indicates the file information.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="file" type="fileStorageType"/>
   </xsd:choice>
   <xsd:attribute name="type" use="optional">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="unprotected"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="emcStorageType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object reports the emc directory of some type.
         </doc:purpose>
         <doc:type node="reply"/>
         <doc:parameter name="type">
            Indicates the type of storage information to be returned.
         </doc:parameter>
         <doc:parameter name="file"> Indicates the file information.
         </doc:parameter>
      </xsd:documentation>
   </xsd:annotation>
   </xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="file" type="fileStorageType"/>
   </xsd:choice>
   <xsd:attribute name="type" use="optional">
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="unprotected"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="fileStorageType">
   <xsd:annotation>
      <xsd:documentation>
         <doc:purpose>
            Object holds file properties.
         </doc:purpose>
         <doc:type node="reply"/>
```

```
      </xsd:documentation>
   </xsd:annotation>
   <xsd:choice minOccurs="0" maxOccurs="1"
      <xsd:element name="base64Data" type="xsd:base64Binary"/>
   </xsd:choice>
   <xsd:attribute name="name" type="xsd:string" use="required"/>
   <xsd:attribute name="size" type="xsd:string" use="required"/>
</xsd:complexType>

</xsd:schema>
```

Contact Information

Printronix Auto ID Customer Support

The Printer Place 708-597-4222
www.theprinterplace.com