

InfoPrint ProcessDirector

Developing Web Services

G550-1152-02

InfoPrint Solutions Company

InfoPrint ProcessDirector

Developing Web Services

G550-1152-02

InfoPrint Solutions Company

Note:

Before using this information and the product it supports, read the information in “Notices” on page 57.

Third edition (November 2009)

This edition applies to InfoPrint ProcessDirector for Linux, Version 1 Release 3 Modification 1 (Program Number 5765-G79) with PTF U820528, InfoPrint ProcessDirector for AIX, Version 1 Release 3 Modification 1 (Program Number 5765-G74) with PTF US00105, and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces G550-1152-01.

Internet

Visit our home page: <http://www.infoprint.com>

You can send comments by e-mail to printpub@infoprint.com or by mail to:

InfoPrint Solutions Company
6300 Diagonal Hwy 002J
Boulder, CO 80301-9270
U.S.A.

This product is or contains commercial computer software and commercial computer software documentation developed exclusively at private expense. As specified in Federal Acquisition Regulation 12.212 in the case of civilian agencies and Defense Federal Acquisition Regulation Supplement 227.7202 in the case of military agencies, use, duplication and disclosure by agencies of the U.S. Government shall solely be in accordance with the accompanying International Program License Agreement in case of software products and in accordance with the licensing terms specified in the product’s documentation in the case of hardware products.

© Copyright InfoPrint Solutions Company 2008, 2009.

Contents

About this publication	v	Chapter 14. Get Tagged Logical Elements within a job	29
Who should read this publication	v	Chapter 15. Get Tagged Logical Element values within a job.	31
Related information	v	Chapter 16. Login	33
Chapter 1. Web services in InfoPrint ProcessDirector	1	Chapter 17. Logout	35
Usage notes	1	Chapter 18. Print again.	37
Notation and conventions for Web service descriptions.	2	Chapter 19. Process again	39
Chapter 2. Connect and disconnect input devices	3	Chapter 20. Set object properties	41
Chapter 3. Continue jobs	5	Chapter 21. Start printers.	45
Chapter 4. Count candidate printer jobs	7	Chapter 22. Stop jobs	47
Chapter 5. Count object log entries	9	Chapter 23. Stop printers.	49
Chapter 6. Count objects	11	Chapter 24. Submit a job	51
Chapter 7. Delete jobs	13	Chapter 25. Submit a job using a job ticket	53
Chapter 8. Enable and disable objects	15	Appendix. HTTP return codes	55
Chapter 9. Get allowed actions	17	Notices	57
Chapter 10. Get candidate printer jobs	19	Trademarks	59
Chapter 11. Get object log	21		
Chapter 12. Get object properties	23		
Chapter 13. Get objects	25		

About this publication

This publication provides technical information about using Web services that use Representational State Transfer (REST) software architecture principles to interact with InfoPrint® ProcessDirector for Linux®, Version 1 Release 3.1 (Program Number 5765-G79) with PTF U820528 or InfoPrint ProcessDirector for AIX® Version 1 Release 3.1 (Program Number 5765-G74) with PTF US00105.

Who should read this publication

This publication is for system programmers who need to configure Web services to access functions of InfoPrint ProcessDirector from other applications.

This publication does not provide general education about Web services; it provides the information that a programmer who is already familiar with Web services needs to work with the Web services that accompany InfoPrint ProcessDirector.

Related information

For information about InfoPrint products, see these Web pages:

- InfoPrint Solutions Company Web site (<http://www.infoprint.com>)
- InfoPrint Information Center (<http://www.infoprint.com/infocenter>)

For information about using InfoPrint ProcessDirector and Advanced Function Presentation™ (AFP™) printing, see the information center and the online help, which you can access from the Web-based user interface.

Chapter 1. Web services in InfoPrint ProcessDirector

InfoPrint ProcessDirector provides Web services that use Representational State Transfer (REST) software architecture principles so you can access its functions from other applications.

RESTful Web services send requests and receive results using the Hypertext Transfer Protocol (HTTP). Each Web service has a Universal Resource Indicator (URI) associated with it. When you create requests, you modify the URI to include specific information about the object that you are interested in or the action that you want to do. These modifications can be simple, such as inserting the name of a printer object into the URI, or highly complex, such as using Extensible Markup Language (XML) to describe the criteria to use when searching for objects and how they should be sorted when they are returned. If you code XML elements to include in your requests, they must be Universal Resource Locator (URL)-encoded before they are added to the URI.

With InfoPrint ProcessDirector, you can use Web services to request information about print jobs and other objects in the system. You can also use them to set properties and do actions such as submitting jobs, enabling and disabling objects, and starting and stopping printers.

Usage notes

The Web services provided with InfoPrint ProcessDirector follow specific conventions. When you integrate Web services with your application, refer to these usage notes to make sure that your implementation functions properly.

- You must use the **Login** Web service to establish a connection and receive a credential token from the InfoPrint ProcessDirector server before you can use any other Web services. All of the other Web services require that you include a credential token in the request header of the URI. If you do not, you receive an immediate 401 Unauthorized HTTP error.
- You can specify one or more languages for a Web service as part of the request header, **Accept-Language**. If you do not specify a language or if the server cannot find a match for the language specified, the system uses US English.
- You cannot translate actions and URIs into other languages because they are fixed-character strings. For example, "**action=**" remains the same for all languages.
- DELETE, GET, and HEAD requests by convention do not contain entity-bodies. As a result, you must place the XML content for **query** (criteria), **sort**, and **propertyset** elements in the URI query string. You must URL-encode the XML that you place in those variables before it is added to the URI. You can use the Java™ URLEncoder class or a similar utility in another language.
- HEAD requests do not return a response body. All **count** requests submitted as HEAD services return the count under the **result_count** header of the response.
- Record sets retrieved might change between invocations of any service, especially when you use sorting or filtering functions; successive identical requests might not yield the same records.
- You must wrap any XML that you provide as input in the entity-body in a root node, such as:

```
<input>
  <message ... />
  <message ... />
</input>
```

- If you activate Secure Sockets Layer (SSL) support, InfoPrint ProcessDirector uses the SSL protocol for all HTTP communications, including Web services. If SSL support is active, you must use a Web services client that supports SSL to invoke Web services.

Notation and conventions for Web service descriptions

The descriptions of InfoPrint ProcessDirector Web services follow this general format.

Each description begins with the purpose of the Web service and continues with the technical details. Technical details include:

HTTP method

All Web services use DELETE, GET, HEAD, POST, or PUT.

URI

The URI that you modify to invoke a Web service. Text in italics must be changed; text not in italics must stay the same.

Request headers

Sends the credential token that you receive when you log in to the server and, as an option, sets the language that you want to use.

Query variables

Describes the options and information that a request can include. Each Web service uses different allowed variables. Some variables are required, some are optional.

Return codes

Lists the HTTP return codes that are valid for the Web service.

Response body on success

Describes what the response from the InfoPrint ProcessDirector system looks like when a request is processed successfully.

Most of the descriptions include examples of requests; many also contain sample responses.

Chapter 2. Connect and disconnect input devices

This Web service sends a request to connect or disconnect one or more input devices.

HTTP method

POST

URI

/aiwservices/v1/inputdevice/*ID1;ID2;ID3;...*

Request headers

ippdcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Indicates whether the action is to connect or disconnect the input device.

- action=connect
- action=disconnect

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Examples

Sample request

Connect more than one input device.

```
http://hostname:15080/aiwservices/v1/inputdevice/  
HotFolderAFP;HotFolderFileName?action=connect
```

Chapter 3. Continue jobs

This Web service submits a request to resume printing a job that is currently stopped at a manual checkpoint.

HTTP method

POST

URI

/aiwservices/v1/job/*ID1;ID2;ID3;...*

Request headers**ippdcredential**

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

action=continue

Return codes

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 500 Internal Server Error
- 501 Not Implemented

Examples**Sample request**

Continue more than one job.

```
http://hostname:15080/aiwservices/v1/job/  
10000000;10000001;10000002?action=continue
```

Chapter 4. Count candidate printer jobs

This Web service counts the jobs that can be printed on a specific printer.

HTTP method

HEAD

URI

/aiwservices/v1/printer/*ID*/job

Request headers

ippdcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Note: GET, HEAD, and DELETE requests by convention do not contain entity-bodies. The XML content for **query**, **sort**, and **propertyset** elements must be part of the URI query string. XML placed in these variables must be URL-encoded using Java URLEncoder class or a similar utility in another language before they are added to the URI.

query = <query />

Limits the jobs that are counted based on the value for a given property. For example, you can request the number of candidate jobs that have more than 10,000 pages for a certain printer. If this variable is not specified, all candidate jobs are counted.

Attributes:

property

The database property name for the property that you want to use as the selection criteria.

operator

The comparison operator. Valid operators are:

EQ Equal

GE Greater than or equal to

GT Greater than
IS_NULL Value is empty
LE Less than or equal to
LIKE Values that follow the **LIKE** operator are case-sensitive. A wildcard character (%) can be used in the value.
LT Less than
NE Not equal
NOT_LIKE Values that follow the **NOT_LIKE** operator are case-sensitive. A wildcard character (%) can be used in the value.

value

The value of the selected property that you want to base your selection criteria on.

Return codes

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

HEAD requests do not return a response body; all **count** HEAD services return the count under the **result_count** header of the response.

Examples

Sample request

Count the candidate jobs for the printer **IP4000** that have more than 10,000 pages.

`http://hostname:15080/aiwservices/v1/printer/IP4000/job`

Query variables:

`query = <query property="Job.TotalPages" operator="GT" value="10000"/>`

Chapter 5. Count object log entries

This Web service requests the number of messages that currently exist in the message log for a specific object. The request can include a timeframe.

HTTP method

HEAD

URI

/aiwservices/v1/objectType/ID/log

Valid values for *objectType* are:

- inputdevice
- inputfile
- instance
- job
- jobtype
- mediasize
- mediatype
- printer
- steptemplate
- user
- workflowsystem

Request headers

ippcredential

`<credential token="value" />`

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Indicates the start and end times for the log entries that are counted. Both values are in the format: **yyyymmdd**.

rangeStart=yyyymmdd

rangeEnd=yyyymmdd

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

HEAD requests do not return a response body. All **count** HEAD services return the count under the **result_count** header of the response.

Examples

Sample request

Request the number messages that were written to the log for job **10000000** from July 7, 2008 through July 10, 2008.

`http://hostname:15080/aiwservices/v1/job/10000000/log`

Query variables:

`rangeStart=20080707`
`rangeEnd=20080710`

Chapter 6. Count objects

This Web Service requests the number of objects that belong to a certain object type. The request can include filtering options to refine the count further.

HTTP method

HEAD

URI

/aiwservices/v1/objectType

Valid values for *objectType* are:

- inputdevice
- inputfile
- instance
- job
- jobtype
- mediasize
- mediatype
- printer
- steptemplate
- user
- workflowsystem

Request headers

ippcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Note: GET, HEAD, and DELETE requests by convention do not contain entity-bodies. The XML content for **query**, **sort**, and **propertyset** elements must be part of the URI query string. XML placed in these variables must be URL-encoded using Java URLEncoder class or a similar utility in another language before they are added to the URI.

query = <query />

Filters the returned list of objects. For example, a query can request the names of all printers that are in the **Intervention required** state. If this variable is not specified, all objects of the requested type are returned.

Attributes:

property

The database property name for the property that you want to use as the selection criteria.

operator

The comparison operator. Valid operators are:

EQ Equal

GE Greater than or equal to

GT Greater than

IS_NULL

Value is empty

LE Less than or equal to

LIKE Values that follow the **LIKE** operator are case-sensitive. A wildcard character (%) can be used in the value.

LT Less than

NE Not equal

NOT_LIKE

Values that follow the **NOT_LIKE** operator are case-sensitive. A wildcard character (%) can be used in the value.

value

The value of the selected property that you want to base your selection criteria on.

Return codes

- 200 OK
- 401 Unauthorized
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

HEAD requests do not return a response body. All **count** HEAD services return the count under the **result_count** header of the response.

Examples

Sample request

Get the number of jobs that are in **Complete** state.

`http://hostname:15080/aiwservices/v1/job`

Query variables:

`query = <query property="Job.State" operator="EQ" value="Complete"/>`

Chapter 7. Delete jobs

This Web service submits a request to delete one or more jobs from the system.

HTTP method

DELETE

URI

/aiwservices/v1/job/*ID1;ID2;ID3;...*

Request headers

ippdcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Return codes

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 500 Internal Server Error
- 501 Not Implemented

Examples

Sample request

Delete more than one job.

`http://hostname:15080/aiwservices/v1/job/10000000;10000001;10000002`

Chapter 8. Enable and disable objects

This Web service sends a request to enable or disable an InfoPrint ProcessDirector object.

HTTP method

POST

URI

/aiwservices/v1/objectType/ID1;ID2;ID3;...

Supported *objectTypes* are:

- inputdevice
- instance
- jobtype
- printer
- steptemplate

Request headers

ippdcredential

<credential token="value" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Indicates whether the action is to enable or disable the object.

- action=enable
- action=disable

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Note: If any of the objects does not exist, a 404 Not Found is returned.
However, all of the objects that do exist are enabled or disabled.

Examples

Sample request

Disable more than one printer.

`http://hostname:15080/aiwservices/v1/printer/printer1;printer2?action=disable`

Chapter 9. Get allowed actions

This Web service requests a list of actions that the current user ID can do for a specific object type.

HTTP method

GET

URI

/aiwservices/v1/objectType/actions

Valid values for *objectType* are:

- inputdevice
- inputfile
- instance
- job
- jobtype
- mediasize
- mediatype
- printer
- steptemplate
- user
- workflowsystem

Request headers

ippcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Return codes

- 200 OK
- 401 Unauthorized
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

InfoPrint ProcessDirector returns a <resultset> The <resultset> contains <result> elements that contain the allowed actions as strings.

Notes:

1. Null values are represented as self-closing <property> elements with no value attribute.
2. Empty values are represented as <property> elements with no value attribute and no embedded <value> tags. They are not self-closing.

Examples

Sample request

Obtain actions that the current user can do on printers.

`http://hostname:15080/aiwservices/v1/printer/actions`

Sample response

```
<resultset>
  <result value="action_Printer_EditPrinterAttributes"/>
  <result value="action_Printer_StartPrinter"/>
  <result value="action_Printer_DisablePrinter"/>
  <result value="action_Printer_EnablePrinter"/>
  <result value="action_Printer_ShowPrinterJobs"/>
  <result value="action_Printer_ViewPrinterAttributes"/>
  <result value="action_Printer_StopPrinter"/>
  <result value="action_Printer_ViewPrinterLog"/>
</resultset>
```

Chapter 10. Get candidate printer jobs

This Web service requests the jobs that are candidates to be printed on a specific printer. The service can also request the values of the properties of those jobs.

HTTP method

GET

URI

/aiwservices/v1/printer/*ID*/job

Request headers

ippdcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Note: GET, HEAD, and DELETE requests by convention do not contain entity-bodies. The XML content for **query**, **sort**, and **propertyset** elements must be part of the URI query string. XML placed in these variables must be URL-encoded using Java URLEncoder class or a similar utility in another language before they are added to the URI.

property = <property />

Indicates which property value to retrieve for the jobs that are returned. If this variable is not included, the default set of variables for jobs is returned. To retrieve more than one property, enclose the <property> elements in a <propertyset>.

Attributes:

name

The database property name of the property to retrieve for the specified object.

propertyset = <propertyset />

Encloses more than one <property> element.

Return codes

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

InfoPrint ProcessDirector returns a <resultset> that contains <object> elements. The <object> elements list the candidate jobs and the requested properties. Only jobs that match the scheduling properties for the printer and that are in the **PrintJobs** step and **Unassigned** state are returned.

Notes:

1. Null values are represented as self-closing <property> elements with no value attribute.
2. Empty values are represented as <property> elements with no value attribute and no embedded <value> tags. They are not self-closing.

Examples

Sample request

Obtain candidate jobs for the printer **IP4000**.

`http://hostname:15080/aiwservices/v1/printer/IP4000/job`

Chapter 11. Get object log

This Web service requests all the messages that were written to the error and message logs for a specific object. The request can include a timeframe to limit the number of messages that are returned.

HTTP method

GET

URI

/aiwservices/v1/objectType/ID/log

Valid values for *objectType* are:

- inputdevice
- inputfile
- instance
- job
- jobtype
- mediasize
- mediatype
- printer
- steptemplate
- user
- workflowsystem

Request headers

ippcredential

`<credential token="value" />`

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Indicates the start and end times for the log entries that are returned. Both values are in the format: **yyyymmdd**.

rangeStart=yyyymmdd

rangeEnd=yyyymmdd

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

InfoPrint ProcessDirector returns a <resultset> that contains <message> elements.

Notes:

1. Null values are represented as self-closing <property> elements with no value attribute.
2. Empty values are represented as <property> elements with no value attribute and no embedded <value> tags. They are not self-closing.

Examples

Sample request

Request the log messages for job 10000000.

`http://hostname:15080/aiwservices/v1/job/10000000/log`

Query variables:

`rangeStart=20080707`
`rangeEnd=20080710`

Sample response

```
<resultset>
<result>
  <message action="This is an information message that requires no response."
  code="AIWP0049I" message="InfoPrint ProcessDirector stopped the job 10000000."
  name="10000000" severity="I" time="2008-07-09 20:19:45.126"/>
</result>
<result>
  <message action="This is an information message that requires no response."
  code="AIWP003I" message="Job 10000000: moved from
  Prepare.AFP.UseInlineFormDefinition.Stopped to
  Prepare.AFP.UseInlineFormDefinition.Stopped" name="10000000" severity="S"
  time="2008-07-09 20:19:45.127"/>
</result>
</resultset>
```

Chapter 12. Get object properties

This Web service requests the property values for an object based on a list of attributes.

HTTP method

GET

URI

/aiwservices/v1/objectType/ID

Valid values for *objectType* are:

- inputdevice
- inputfile
- instance
- job
- jobtype
- mediasize
- mediatype
- printer
- steptemplate
- user
- workflowsystem

Request headers

ippcredential

`<credential token="value" />`

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Requests a set of properties for an object. If this variable is not specified, all properties of the object are returned. In this context, you can only specify the property name; you cannot specify a property value.

Contains one or more `<property>` elements.

property

Attribute:

name

The database property name of the property that you want to query for the specified object.

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

Returns a single <object> that contains nested <property> elements.

Notes:

1. Null values are represented as self-closing <property> elements with no value attribute.
2. Empty values are represented as <property> elements with no value attribute and no embedded <value> tags. They are not self-closing.

Examples**Sample request**

Obtain **JobType.description** and **JobType.Steps** properties for the **AFP** job type.

`http://hostname:15080/aiwservices/v1/jobtype/AFP`

Query variables:

```
propertyset = <propertyset>
  <property name="JobType.Description" />
  <property name="JobType.Steps" />
</propertyset>
```

Chapter 13. Get objects

This Web service requests a list of objects of a certain type. The request can include filtering and sorting options for the list.

HTTP method

GET

URI

/aiwservices/v1/objectType

Valid values for *objectType* are:

- inputdevice
- inputfile
- instance
- job
- jobtype
- mediasize
- mediatype
- printer
- steptemplate
- user
- workflowsystem

Request headers

ippcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

Note: GET, HEAD, and DELETE requests by convention do not contain entity-bodies. The XML content for **query**, **sort**, and **propertyset** elements must be part of the URI query string. XML placed in these variables must be URL-encoded using Java URLEncoder class or a similar utility in another language before they are added to the URI.

query = <query />

Filters the returned list of objects. For example, a query can request the names of all printers that are in the **Intervention required** state. If this variable is not specified, all objects of the requested type are returned.

Attributes:

property

The database property name for the property that you want to use as the selection criteria.

operator

The comparison operator. Valid operators are:

EQ Equal

GE Greater than or equal to

GT Greater than

IS_NULL

Value is empty

LE Less than or equal to

LIKE Values that follow the **LIKE** operator are case-sensitive. A wildcard character (%) can be used in the value.

LT Less than

NE Not equal

NOT_LIKE

Values that follow the **NOT_LIKE** operator are case-sensitive. A wildcard character (%) can be used in the value.

value

The value of the selected property that you want to base your selection criteria on.

sort = <sort />

Sorts the list of objects that is returned. If this variable is not specified, the data is returned in no particular sorted order. Only one sort property can be specified. InfoPrint ProcessDirector can sort on most properties, but not all of them.

Attributes:

name

The database property name of the property that you want to use for sorting.

direction

The order that you want the results to appear in. Valid values are:

asc Sort the returned objects in ascending order.

desc Sort the returned objects in descending order.

propertyset = <propertyset>

Requests a certain set of properties for the returned list of objects. If this variable is not specified, all properties of the object are returned. In this context, you can only specify the property name; you cannot specify a property value.

Contains one or more <property> elements.

property

Attribute:

name

The database property name of the property that you want to return for the objects in the returned list.

Return codes

- 200 OK
- 401 Unauthorized
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

InfoPrint ProcessDirector returns a <resultset>, which contains <result> elements in this format:

```
<resultset>
  <result>
    <object name="value1"/>
    <object name="value2"/>
    ...
    <object name="valuen"/>
  </result>
</resultset>
```

Notes:

1. Null values are represented as self-closing <property> elements with no value attribute.
2. Empty values are represented as <property> elements with no value attribute and no embedded <value> tags. They are not self-closing.

Examples**Sample request**

Get a list of jobs that are in **Complete** state, sorted on their submit times. Retrieve only the **Job.Class** and **Job.Forms** property values.

`http://hostname:15080/aiwservices/v1/job`

Query variables:

```
query = <query property="Job.State" operator="EQ" value="Complete"/>
sort = <sort name="Job.SubmitTime" direction="asc"/>
propertyset = <propertyset>
  <property name="Job.Class" />
  <property name="job.Forms" />
</propertyset>
```

Sample request

Get a list of all the jobs that meet one of these sets of criteria:

- In class **A**, with the number of copies set to **1** or more
- In class **A**, or currently assigned to one of these printers: **PacificColor**, **PacificCutsheet**, **PacificBlackWhite**
- In class **A**, with a **Job number** less than **16739578** or a description that contains the word **hold**

Query variables:

```
<query property="Job.Class" operator="EQ" value="A">
  <AND>
    <query property="Job.Copies" operator="GE" value="1" />
  </AND>
  <OR>
    <query property="Job.CurrentPrinter" operator="LIKE" value="Pacific%" />
  </OR>
```

```
<AND>  
  <query property="Job.ID" operator="LT" value="16739578">  
    <OR>  
      <query property="Job.Description" operator="LIKE" value="%hold%">  
    </OR>  
  </query>  
</AND>  
</query>
```

Chapter 14. Get Tagged Logical Elements within a job

This Web service requests a list of all the Tagged Logical Element (TLE) names that are used in a job.

HTTP method

GET

URI

/aiwservices/v1/job/*ID*/TLE

Request headers

ippdcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

A <resultset> containing one <result> for each TLE in the job. The **value** attribute of each <result> is a string that contains the name of the TLE.

Notes:

1. Null values are represented as self-closing <property> elements with no value attribute.
2. Empty values are represented as <property> elements with no value attribute and no embedded <value> tags. They are not self-closing.

Examples

Sample request

Get all TLEs used in job 10000000.

`http://hostname:15080/aiwservices/v1/job/10000000/TLE`

Sample response

```
<resultset>  
  <result value="Customer name"/>  
  <result value="Address"/>  
  <result value="Account number"/>  
  <result value="Sheet"/>  
</resultset>
```

Chapter 15. Get Tagged Logical Element values within a job

This Web service submits a request for all the values that are used for a specific Tagged Logical Element (TLE) in a job. Both Page Level and Group Level TLE values are returned.

HTTP method

GET

URI

/aiwservices/v1/job/ID/TLE/tleName

Request headers

ippdcredential

<credential token="value" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

A <resultset> containing one <result> for each TLE value in the job. Each <result> contains one <tle> element with these attributes:

name

The job ID specified in the URI

tleName

The TLE specified in the URI

tleValue

One of the values used in the job for the specified TLE

page

The page of the job where the TLE is used

Notes:

1. Null values are represented as self-closing <property> elements with no value attribute.
2. Empty values are represented as <property> elements with no value attribute and no embedded <value> tags. They are not self-closing.

Examples**Sample request**

Get all TLE values of the TLE **Customer name** for job **10000000**.

`http://hostname:15080/aiwservices/v1/job/10000000/TLE/Customer+name`

Sample response

```
<resultset>
  <result>
    <tle name="10000000" tleName="Customer name" tleValue="Pamela Clark-Anders" page="15"/>
  </result>
  <result>
    <tle name="10000000" tleName="Customer name" tleValue="Carlos Ferreira" page="18"/>
  </result>
</resultset>
```

Chapter 16. Login

This Web service lets you log in to InfoPrint ProcessDirector Web services so you can submit additional requests.

If the login is successful, the login service returns a credential token. The primary server manages the validity of the token. The token must be included in all Web services requests except this one.

HTTP method

GET

URI

/aiwservices/v1/login

Request headers

ippdlogin

```
<login name="username" pwd="password" />
```

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Return codes

- 200 OK
- 401 Unauthorized
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

When the login action succeeds, InfoPrint ProcessDirector returns:

```
<credential token="value"/>
```

The token is valid until a **logout** request is submitted.

Sample request

```
http://hostname:15080/aiwservices/v1/login
```

Sample response

```
<credential token="6533823204038603430"/>
```

Chapter 17. Logout

This Web service sends a request to InfoPrint ProcessDirector to log out the user or application that is using Web services.

The logout service invalidates a given credential token. Further requests using that credential are refused. The user or application must use the **Login** service to get a new credential.

HTTP method

DELETE

URI

/aiwservices/v1/login

Request headers

ippdcredential

<credential token="value" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Return codes

- 200 OK
- 500 Internal Server Error
- 501 Not Implemented

Chapter 18. Print again

This Web service submits a request to reprint all or part of a job.

HTTP method

POST

URI

/aiwservices/v1/job/*ID*

Request headers

ippdcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

- action=printAgain
- Job.PageRange=*x* or Job.PageRange=*y-z*
- Job.Copies=*n*

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Examples

Sample request

Print a job again, reprinting all pages three times.

```
http://hostname:15080/aiwservices/v1/job/1000000?action=PrintAgain&Job.PageRange=&Job.Copies=3
```

Chapter 19. Process again

This Web service submits a request to move a job to a previous step in its job type for reprocessing.

HTTP method

POST

URI

/aiwservices/v1/job/*ID1*

Request headers**ippdcredential**

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

action=processAgain

Request body

<input> element with a <checkpoint> node. The <checkpoint> XML must be retrieved from the **Job.RestartSteps** property and must be specified or an error occurs. The <checkpoint> does not need to be URL-encoded because it is part of the body of the request.

Return codes

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Examples

Sample request

Run **Process Again** on job **10000000** specifying the phase **Receive** and step **SetJobPropsFromSteps**.

`http://hostname:15080/aiwservices/v1/job/10000000action=processAgain`

with request body:

```
<checkpoint children="0" name="1000000" phase="Receive" process="AFP"
progress="Staging" restart="Receive" step="SetJobPropsFromSteps"
timestamp="2008-07-11 17:28:07.25"/>
```

Chapter 20. Set object properties

This Web service sets selected properties for a list of objects.

HTTP method

PUT

URI

/aiwservices/v1/objectType/ID1[;ID2;ID3;...]

Valid values for *objectType* are:

- inputdevice
- inputfile
- instance
- job
- jobtype
- mediasize
- mediatype
- printer
- steptemplate
- user
- workflowsystem

Request headers

ippcredential

<credential token="value" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Request body

The properties to set on the specified objects, enclosed in an <object> element. The <object> is a construct of the specific object type (such as **printer** or **input device**), database property names, and the new values for the properties. The <object> should not have an ID set because the ID is set automatically from the URI given (ID1; ... IDn).

Query variables

object = <object />

Specifies the object type as requested in the URI. <object> variables enclose <property> variables.

Attributes:

type The object type.

property = <property />

Indicates the properties to update and their new values.

Attributes:

name The database property name of the property.

value The value to set for the property.

Return codes

- 200 OK
- 206 Partial Content
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

Either a return code 200 or 206. A 206 Partial Content return code means that the set succeeded on some objects, but not all. An example of a 206 Partial Content return code is when three IDs were specified but only two could be updated by the request.

Examples**Sample requests****Setting a property with one value**

Set the description on several jobs. In this case, one of the jobs does not exist.

`http://hostname:15080/aiwservices/v1/job/10000000;10000001;10000002`

Request body:

```
<object type="job">
  <property name="Job.Description" value="Set BY Web Services"/>
</object>
```

Setting a property with multiple values

Set the output bins on a printer.

`http://hostname:15080/aiwservices/v1/printer/Kezar`

Request body:

```
<object type="Printer">
  <property name="Printer.OutputBin">
    <value>Finisher Upper Tray</value>
    <value>Finisher Shift Tray</value>
    <value>Finisher Proof Tray</value>
  </property>
</object>
```

Setting custom property values**Resetting a property with a single value**

`http://hostname:15080/aiwservices/v1/printer/Kezar`

Request body:

```
<object type="Printer">
  <property name="Printer.StapleCapable" value=""/>
</object>
```

Resetting a property with multiple values

Request body:

```
<object type="Printer">
  <property name="Printer.OutputBin">
    <value>$</value>
  </property>
</object>
```

Note:

\$ indicates default sizes, default types, or default bins.

Setting a property with a single value to Any [object].

For example, setting **Requested printer** to **Any printer**.

`http://hostname:15080/aiwservices/v1/job/10000000`

Request Body:

```
<object type="Job">
  <property name="Job.RequestedPrinter" value="*"/>
</object>
```

Setting a property with multiple values to All [objects].

For example, setting **Output bins available** to **All bins**.

`http://hostname:15080/aiwservices/v1/printer/Kezar`

Request Body:

```
<object type="Printer">
  <property name="Printer.OutputBin"/>
  <value>*</value>
</property>
</object>
```

Note: * indicates any printer, all sizes, all types, or all bins.

Sample response

```
206
<resultset>
  <result>
    <message action="Job 10000001 was not found." code="AIWP6022E"
      message="Job 10000001 was not found." name="10000001" severity="E"
      time="2008-07-10 17:03:50.968" type="Job"/>
  </result>
</resultset>
```

Chapter 21. Start printers

This Web service starts an InfoPrint ProcessDirector printer object so it can send jobs to the printer for printing.

HTTP method

POST

URI

/aiwservices/v1/printer/*ID1;ID2;ID3;...*

Request headers

ippdcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

action=start

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Note: If one of the printers does not exist, a 404 Not Found error is returned and none of the printers are started. A 200 OK error indicates that the request has been accepted, code-executed, and no errors were encountered. It does not indicate the action is complete, but that the request is complete and successful, and no further status is forthcoming.

Examples

Sample request

Start more than one printer.

<http://hostname:15080/aiwservices/v1/printer/printer1;printer2?action=start>

Chapter 22. Stop jobs

This Web service submits a request to stop one or more jobs when processing completes for the current step. If the job is in the **printing** state, InfoPrint ProcessDirector stops it as soon as possible and the rest of the job is discarded.

HTTP method

POST

URI

/aiwservices/v1/job/ID1;ID2;ID3;...

Request headers

ippdcredential

<credential token="value" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

action=stop

Return codes

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 500 Internal Server Error
- 501 Not Implemented

Examples

Sample request

Stop more than one job.

```
http://hostname:15080/aiwservices/v1/job/  
10000000;10000001;10000002?action=stop
```

Chapter 23. Stop printers

This Web service requests that InfoPrint ProcessDirector stop the printer object. The service must indicate when to stop the printer—before the current job is complete or after.

HTTP method

POST

URI

/aiwservices/v1/printer/*ID1;ID2;ID3;...*

Request headers

ippdcredential

<credential token="*value*" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Query variables

The **action** variable must be included. The **when** and **NPRO** variables are not required. If they are not included, the system uses the default option.

action=stop

when

Indicates when the printer should stop relative to the job that is currently printing.

now (default)

Stop the printer as soon as possible, without completing the current print job.

afterCopy

For a multi-copy print job, stop the printer after the current copy of the print job is finished printing.

afterJob

Stop the printer after the current job is finished printing.

NPRO

Indicates whether the printer should do a non-process run out (NPRO) when it stops printing.

- **yes** or **true**
- **no** or **false** (default)

Return codes

- 200 OK
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

If one of the printers does not exist, a 404 Not Found error is returned and none of the printers are stopped. A 200 OK message indicates that the request has been accepted, code-executed, and no errors were encountered. It does not indicate the action is complete, but that the request is complete and successful, and no further status is forthcoming.

Examples

Sample request

Stop more than one printer.

`http://hostname:15080/aiwservices/v1/printer/printer1;printer2?action=stop`

Chapter 24. Submit a job

This Web service submits one or more files on the local file system to a hot folder input device to be processed as a job.

HTTP method

POST

URI

/aiwservices/v1/inputdevice/InputDeviceID/job

Request headers

ippdcredential

<credential token="value" />

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Request body

One or more files on the local file system, including print, overrides, and list files, if required. The **Content-Type** of the request should be **multipart/form-data**. You can use a client API like **commons-httpclient** or a command line tool like **cURL** to POST the file.

Return codes

- 202 Accepted
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented

Response body on success

InfoPrint ProcessDirector sends a return code 202 to indicate whether the command was accepted for processing. No other information is returned.

Examples

Sample request

Submit an AFP file called **test.afp** to the hot folder input device named **HotFolderAFP**.

`http://hostname:15080/aiwservices/v1/inputdevice/HotFolderAFP/job`

Using Commons HttpClient

```
....
// Set up the Post method with the URI to invoke
PostMethod method = new PostMethod(
    "http://hostname:15080/aiwservices/v1/inputdevice/HotFolderAFP/job");

// add credential header
File f = new File("c:/test.afp");
Part[] parts = { new FilePart(f.getName(), f) };
method.setRequestEntity(new MultipartRequestEntity(parts,
    method.getParams()));
HttpClient client = new HttpClient();
int status = client.executeMethod(filePost);
....
```

Using Commons HttpClient over SSL

```
....
// Set up the Post method with the URI to invoke
Protocol easyhttps = new Protocol("https", new EasySSLProtocolSocketFactory(), 15443);
String uriString = http://hostname:15080/aiwservices/v1/inputdevice/HotFolderAFP/job;
URI uri = new URI(uriString, true);
PostMethod method = new PostMethod(uri.getPathQuery());
HostConfiguration hc = new HostConfiguration();
hc.setHost(uri.getHost(), uri.getPort(), easyhttps);

// add credential header
File f = new File("c:/test.afp");
Part[] parts = { new FilePart(f.getName(), f) };
method.setRequestEntity(new MultipartRequestEntity(parts, method.getParams()));
HttpClient client = new HttpClient();
int statusCode = client.executeMethod(hc, method);
....
```

Using cURL

```
curl -F "upload=@c:\test.afp" -H "ippdcredential:<credential
token='-2921201405796490177'/>"
http://hostname:15080/aiwservices/v1/inputdevice/HotFolderAFP/job
```

Note: To include more than one file:

- In Commons HttpClient, insert additional file definitions, such as:
 - `File a = new File(file2.afp);`
 - `File b = new File(myfile.pdf);`
 - `File l = new File(mylist.lst);`
 - `File o = new File(overrides.oth);`
- In cURL, insert additional **-F** flags:
 - `-F "upload=@c:\file2.afp"`
 - `-F "upload=@c:\myfile.pdf"`
 - `-F "upload=@c:\mylist.lst"`
 - `-F "upload=@c:\overrides.oth"`

Chapter 25. Submit a job using a job ticket

This Web service submits a file on the local file system to a hot folder input device using a Job Definition Format (JDF) job ticket. You can use the job ticket to override the values of certain job attributes.

HTTP method

POST

URI

/aiwservices/v1/inputdevice/InputDeviceID/job

InputDeviceID is the name of the target input device in the InfoPrint ProcessDirector system.

Request headers

ippdcredential

`<credential token="value" />`

Accept-Language (optional)

lang

Supported language values:

de	German
en	US English (default)
es	Spanish
fr	French
it	Italian
ja	Japanese
pt_BR	Brazilian Portuguese
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Request body

The job ticket file on the local file system and any other required files. The job ticket file might contain either the absolute path to the data files or just the data file names. If an absolute path is specified, you do not have to include the data files. If the job ticket file contains just the file name, the data files must be sent first, followed by the job ticket file.

The **Content-Type** of the request should be **multipart/form-data**. You can use a client API like **commons-httpclient** or a command line tool like **cURL** to POST the file.

Return codes

- 202 Accepted
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found
- 500 Internal Server Error

- 501 Not Implemented

Response body on success

InfoPrint ProcessDirector sends a return code 202 to indicate whether the command was accepted for processing. No other information is returned.

Examples

Sample request

Submit a JDF file and a data file to an input device named **HotFolderAFPJDF**.

`http://hostname:15080/aiwservices/v1/inputdevice/HotFolderAFPJDF/job`

Using Commons HttpClient:

```
....
// Set up the Post method with the URI to invoke
PostMethod method = new PostMethod(
"http://hostname:15080/aiwservices/v1/InputDeviceJDFHF/HotFolderJDF/job");
// add credential header
File dataFile = new File("c:/test.pdf");
File jdfFile = new File("c:/test.jdf");
Part[] parts = {
    new FilePart(dataFile.getName(), dataFile),
    new FilePart(jdfFile.getName(), jdfFile)
};
method.setRequestEntity(new MultipartRequestEntity(parts, method.getParams());

HttpClient client = new HttpClient();
int status = client.executeMethod(filePost);
....
```

Using Commons HttpClient over SSL

```
....
// Set up the Post method with the URI to invoke
Protocol easyhttps = new Protocol("https", new EasySSLProtocolSocketFactory(), 15443);
String uriString = http://hostname:15080/aiwservices/v1/inputdevice/HotFolderAFPJDF/job;
URI uri = new URI(uriString, true);
PostMethod method = new PostMethod(uri.getPathQuery());
HostConfiguration hc = new HostConfiguration();
hc.setHost(uri.getHost(), uri.getPort(), easyhttps);

// add credential header
File dataFile = new File("c:/test.pdf");
File jdfFile = new File("c:/test.jdf");
Part[] parts = {
    new FilePart(dataFile.getName(), dataFile),
    new FilePart(jdfFile.getName(), jdfFile)
};
method.setRequestEntity(new MultipartRequestEntity(parts, method.getParams());
HttpClient client = new HttpClient();
int statusCode = client.executeMethod(hc, method);
....
```

Using cURL

```
curl -F "upload=@c:/test.pdf" -F "upload=@c:/test.jdf" -H "ippcredential:
<credential token='-2921201405796490177'/">
http://hostname:15080/aiwservices/v1/inputdevice/HotFolderJDF/job
```

Note: To include more than one file:

- In Commons HttpClient, insert additional file definitions, such as:
 - `File b = new File(myfile.pdf);`
 - `File l = new File(mylist.lst);`
- In cURL, insert additional **-F** flags:
 - `-F "upload=@c:\myfile.pdf"`
 - `-F "upload=@c:\mylist.lst"`

Appendix. HTTP return codes

When you submit a request using Web services, HTTP provides these messages that you can use to help determine if your request was processed correctly.

InfoPrint ProcessDirector Web services support these HTTP return codes:

200 OK

The request has been processed with no errors.

202 Accepted

The request has been accepted for processing, but the processing has not been completed. There is no facility for resending a status code from an asynchronous operation such as this.

206 Partial Content

Some elements of the request were processed successfully. Messages in the response body indicate the portions that were not successful. If all elements of a multi-ID request succeed, a return code 200 results; if all fail, a return code 500 should result.

400 Bad Request

The request contains incorrect data or invalid syntax and cannot be processed. Resubmitting the request does not solve this problem.

401 Unauthorized

No credentials were supplied, or the supplied credential does not permit access to the requested service.

404 Not Found

The request included an object that did not exist at the time of the request.

409 Conflict

The request data given conflicts with itself or with the current state of the resource.

500 Internal Server Error

The server encountered an unexpected condition that prevented it from filling the request. This return code is expected to be the most common (or general) failure code. Details should be contained in a message element or elements in the entity-body.

501 Not Implemented

The resource does not support the request. Resubmitting the request does not solve this problem.

503 Service Unavailable

The Web service cannot complete the request because of a communication failure. Resubmitting the request might solve the problem.

Notices

This information was developed for products and services offered in the U.S.A.

InfoPrint Solutions Company may not offer the products, services, or features discussed in this document in other countries. Consult your local InfoPrint Solutions Company representative for information on the products and services currently available in your area. Any reference to an InfoPrint Solutions Company product, program, or service is not intended to state or imply that *only* that InfoPrint Solutions Company product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any InfoPrint Solutions Company intellectual property rights may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-InfoPrint Solutions Company product, program, or service.

References in this document to InfoPrint Solutions Company products, product features, programs or services do not imply that InfoPrint Solutions Company intends to make such products, product features, programs or services available in all countries in which InfoPrint Solutions Company operates or does business.

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

InfoPrint Solutions Company, LLC
6300 Diagonal Hwy 002J
Boulder, CO 80301-9270
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the InfoPrint Solutions Company Intellectual Property Department in your country or send inquiries, in writing, to:

InfoPrint Solutions Company, LLC
6300 Diagonal Hwy 002J
Boulder, CO 80301-9270
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INFOPRINT SOLUTIONS COMPANY PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. InfoPrint Solutions Company may make improvements and/or changes in the product(s) described in this publication at any time without notice.

Any references in this information to non-InfoPrint Solutions Company Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this InfoPrint Solutions Company product and use of those Web sites is at your own risk.

InfoPrint Solutions Company may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

InfoPrint Solutions Company, LLC
6300 Diagonal Hwy 002J
Boulder, CO 80301-9270
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by InfoPrint Solutions Company under terms of the InfoPrint Solutions Company Customer Agreement, InfoPrint Solutions Company International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-InfoPrint Solutions Company products was obtained from the suppliers of those products, their published announcements or other publicly available sources. InfoPrint Solutions Company has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-InfoPrint Solutions Company products. Questions on the capabilities of non-InfoPrint Solutions Company products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to InfoPrint Solutions Company, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming

interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. InfoPrint Solutions Company, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

These terms are trademarks or registered trademarks of Ricoh Co., Ltd., in the United States, other countries, or both:

- Advanced Function Presentation
- AFP
- InfoPrint
- Infoprint
- Ricoh

These terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

- AIX
- IBM

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

InfoPrint Solutions Company

Program Number: 5765-G79
5765-G74

Printed in USA

G550-1152-02

