

Setting up to use MVS Download

[Back to Administrator procedures](#)

Important information for PSF for OS/2 users

If you have decided to use the MVS Download function of Infoprint Manager to replace your PSF for OS/2 system, you should be aware of these differences between the two products:

- PSF for OS/2 uses SNA to send data from the host to the server; MVS Download uses TCP/IP.
- To use PSF for OS/2, you had to install and configure IBM Communications Manager for OS/2 or IBM Communications Server for OS/2; you do not need to install the equivalent product on your Windows system to use MVS Download.
- The Distributed Print Function (DPF) feature of PSF for OS/2 receives IPDS data from the host system. MVS Download receives AFPDS and line data, as well as PCL or PostScript and passes it to Infoprint Manager to be transformed before it is printed.
- In PSF for OS/2, AFP resources are sent from the host system to the print server along with the job. In MVS Download, AFP resources are not sent with the job; they must either be sent inline in the data stream or they must be resident on your Infoprint Manager server before any print jobs that use them are sent.
- If you use the Distributed Print Function (DPF) feature of PSF for OS/2, your resources are managed automatically, so any changes you made to your resource files were automatically incorporated into your print jobs. MVS Download does not have an automatic resource management feature. If you make changes to your resource files, those changes will not be automatically incorporated into print jobs you submit through MVS Download. You will need to copy the changed resource files to the computer that your Infoprint Manager server runs on.

Configuring MVS Download

The MVS Download feature of Infoprint Manager uses a *Destination Control File (DCF)* to map Job Control Language (JCL) parameters to Infoprint Manager submission parameters. A default DCF called `mvsdmap.txt` is provided with Infoprint Manager; it is installed in the `<install path>\var\pd\mvsd` directory, where `<install path>` is the directory that you installed Infoprint Manager in. MVS Download receives JCL parameters from the host system, applies the DCF to it, and generates a corresponding list of Infoprint parameters. The list of Infoprint parameters is then passed to the MVS Download Exit Program (or *Exit*) so Infoprint Manager can process the job.

The MVS Download feature of Infoprint Manager also invokes an Exit for every file received. This Exit is responsible for performing some action on the received file. Two sample Exits are provided with Infoprint Manager: one written in Microsoft Visual C++ Version 6.0 and one written in Microsoft Visual Basic Version 6.0. Both samples perform the same action: submit the file to Infoprint Manager using `pdpr` and log any errors. The exit also stop the MVS Download receiver if the `pdpr` fails.

There are essentially five tasks that you must perform to be able to print using the MVS Download feature of Infoprint Manager:

1. Create the Infoprint destination or destinations that you want to submit your print jobs to.
2. Make the AFP resources that you need to use for your print jobs available to Infoprint Manager.
3. **[Optional]**Modify the DCF.
4. **[Optional]**Modify the Exit Program.
5. Create an MVS Download receiver that will accept print jobs from the host and tie all of the other elements together.

All printing configurations must complete steps one, two, and five. Refer to the *IBM Infoprint Manager for Windows NT and Windows 2000: Getting Started* for information on creating destinations. Refer to “Making your AFP resources available to Infoprint Manager” on page 16 for information about completing step 2. If you can use the sample DCF and Exit that are provided with Infoprint Manager (the default configuration), you do not have to perform steps three and four. Refer to “Setting up a default MVS Download configuration” on page 3 for instructions on completing step five for the default configuration. You can use the default configuration if all of the following conditions apply:

1. On jobs that are sent to this MVS Download receiver, the DEST listed on the job when it is received from the host (converted to lower case) identifies the Infoprint actual or logical destination that the job should be submitted to

OR

Jobs with no DEST specified will be submitted to the Infoprint actual or logical destination that you set as the default for this receiver.

2. You do not use Infoprint job scheduling functions based on media type.
3. You will not submit AFPDS jobs with inline resources and request multiple copies.
4. When submission of a print job fails (for example, because the server is stopped or because the DEST requested does not exist), you will accept the default action that the receiver automatically stops.
5. You will not use MVS Exit 15.

If you do not meet all of the criteria, you will need to modify the DCF, the Exit, or both. Some of the reasons that you would need to modify the DCF are:

1. You want to use alternate attributes or no attributes for some mvs_keywords.
2. You do not want the DCF to change the DEST value it receives to lower case.
3. You want to use Infoprint job scheduling based on media type.
4. You will submit AFPDS jobs with inline resources and request multiple copies.
5. You see that some of the MVS to Infoprint mappings in the sample DCF are not what you want.
6. You want to identify MVS job defaults that are not communicated from MVS (for example PAGEDEF).
7. You want to identify Infoprint submission parameters to use on all jobs (for example, the font resource directory on your Windows system).
8. You use Host MVS Exit 15.

Refer to the following sections for instructions on modifying your DCF and Exit, and creating an MVS Download receiver:

- “Understanding and using the MVS Download destination control file” on page 4
- “Understanding the MVS Download Exit Program” on page 14
- “Creating an MVS Download Receiver” on page 15

Setting up a default MVS Download configuration

Once you have made your resources available, the only tasks you need to perform on your Windows system to set up a Default configuration are:

- Create an Infoprint destination.
Refer to the *IBM Infoprint Manager for Windows NT and Windows 2000: Getting Started* for information on creating destinations.
- Create and configure an MVS Download receiver that uses the default DCF and Exit to send print jobs to that destination.
Refer to “Creating an MVS Download Receiver for a default configuration” for instructions.

Creating an MVS Download Receiver for a default configuration

Before you can begin sending jobs from the MVS JES spool to Infoprint Manager through the MVS Download feature, follow this procedure to set up an MVS Download Receiver:

1. Open the Infoprint Manager Management Console.
2. Left click the **MVS Download Receiver** item in the left pane to select it.
3. Right click the **MVS Download Receiver** item in the left pane to open a pop-up menu.
4. In the menu that pops up, select **New**.
The **Add MVS Receiver** dialog opens.
5. Fill in the fields according to the following specifications. Refer to the Management Console online help for more information about the dialog and the fields it contains.
 - **Port Number:** Enter the port number that you want this receiver to use when it communicates with the host system. This number must match the port number that is specified in the Routing Control File used by the MVS host Download FSA for this computer’s IP address.
 - **Target Destination:** Select the default Infoprint destination that you want this receiver to submit jobs to from the drop-down list. Infoprint Manager will only submit a job to this destination if the jobs does not have a DEST specified when it comes from the host system.
 - **Destination Control File:** Do not change this path.
 - **Exit Program Name:** Do not change this path.
 - In the **Processing Options** group box, both options are selected by default and can be left that way.

Note: When the **Retain command files** check box is selected, Infoprint Manager saves the control files for any MVS Download print jobs that failed in print submission. Therefore, when you want to resubmit those jobs, you do not have to download them from the host system again; the files are still on your Infoprint server. You can deselect this box, but any jobs that fail in processing will need to be resubmitted from the host to be printed.

Operators should evaluate any processing failures, correct the problems that caused the failures, and resubmit the jobs either from the host system (if this box is deselected) or from the receiver (if this box is selected).

For additional information, refer to the Operator Procedure Deleting Orphan Files and Resubmitting Jobs to MVS Download.

6. Click **OK**.

One or more pop-up informational messages may appear; read them and click **OK** to dismiss them.

7. Select the **MVS Download Receivers** item in the left pane of the Management Console.

You should see the receiver you just created in the right pane. The receiver status may be **STOPPED**, but it should start momentarily. Click the **Refresh View** button on the toolbar until the status changes to **RUNNING**.

Once the receiver is running, it is ready to receive print jobs from the host system.

Understanding and using the MVS Download destination control file

The MVS Download Destination Control File (DCF) is a flat text file that consists of Control Statements, blank lines, and comments. Each line has a maximum length of 1023 bytes. Tabs are treated as blank spaces.

Note: If you change the DCF, the changes will not become active until you stop and restart the receivers that use that DCF.

The DCF maps the MVS keywords and parameters it receives from the host to the **pdpr** `-x` and `-p` attributes and parameters. The supplied MVS Download Exit uses **pdpr** to submit jobs to the target destination. To modify any of the other **pdpr** options, you must change the Exit Routine.

The sample DCF that comes with Infoprint Manager is set to perform the mappings listed in Table 1.

Note: This table only lists the Mapping statements in the sample DCF. It does not reflect the Default statements and Global statements that are set.

Table 1. Mappings in the Sample DCF

mvs_definition	Infoprint Attribute and Value
ACCOUNT	account-text
ADDRESS1	address1-text
ADDRESS2	address2-text
ADDRESS3	address3-text
ADDRESS4	address4-text
BUILDING	building-text
CC=NO	carriage-control-type=none
CC=YES	ignored
CCTYPE=ANSI	carriage-control-type=ansi-ebcdic
CCTYPE=MACHINE	carriage-control-type=machine

Table 1. Mappings in the Sample DCF (continued)

CCTYPE=ASCII	carriage-control-type=ansi-ascii
CHARS	chars
CLASS	printer-pass-through=-opa=class
COPIES	copy-count
DATAACK=UNBLOCK	data-fidelity-problem-reported=all
DATAACK=BLOCK	data-fidelity-problem-reported=none
DATAACK=BLKPOS	data-fidelity-problem-reported=character
DATAACK=BLKCHAR	data-fidelity-problem-reported=position
DATATYPE=LINE	document-format=line-data
DEPT	department-text
DEST	target-destination-name and printer-pass-through=-opa=destination
DUPLEX=NO	sides=1
DUPLEX=NO	plex=simplex
DUPLEX=NORMAL	sides=2
DUPLEX=NORMAL	plex=simplex
DUPLEX=TUMBLE	sides=2
DUPLEX=TUMBLE	plex=tumble
FCB	page-definition
FILEFORMAT=RECORD	new-line-option=counted-4-octet-aligned
FILEFORMAT=STREAM	new-line-option=lf
FORMDEF	form-definition
FORMS	printer-pass-through=-opa=forms
INTRAY	printer-pass-through=-obin
IPADDR	printer-pass-through=-opa=ipaddr
JOBID	printer-pass-through=-opa=jobid
JOBNAME	job-name and job-owner
NAME	name-text
NODEID	node-id-text
OFFSETXB	x-image-shift-back
OFFSETXF	x-image-shift
OFFSETYB	y-image-shift-back
OFFSETYF	y-image-shift
OUTBIN	output-bin
OVERLAYB	overlay-back
OVERLAYF	overlay-front
PAGEDEF	page-definition
PRMODE=SOSI1	shift-out-shift-in=one
PRMODE=SOSI2	shift-out-shift-in=two
PRMODE=SOSI3	shift-out-shift-in=three

Table 1. Mappings in the Sample DCF (continued)

PRMODE=[<i>anything else</i>]	ignored
PROGRAMMER	programmer-text
PRTQUEUE	printer-pss-through=-oprtqueue
RESFMT=P240	font-resolution=P240
RESFMT=P300	font-resolution=P300
ROOM	room-text
SEGMENTID	printer-pass-through=-opa=segmentid
SYSOUT	printer-pass-through=-opa=class
TITLE	title-text
TRC	table-characters-reference
UCS	chars
USERID	user-id-text

Types of control statements

There are three kinds of Control Statements in a DCF:

1. **Default statements** identify a default parameter for an `mvs_keyword` when one is not received from MVS. They consist of only an `mvs_definition`.

Note: If you include multiple Default statements that use the same `mvs_keyword`, Infoprint only uses the last one.

2. **Global statements** identify an Infoprint parameter that should be used on all data set submissions. This statement uses GLOBAL as the `mvs_definition` value.

Note: If you include multiple Global statements that use the same `infoprint_attribute`, Infoprint only uses the last one.

3. **Mapping statements** match `mvs_keyword` with their Infoprint equivalents. These statements must contain both an `mvs_keyword` and a control. If there is no Mapping statement listed in the DCF for one or more of the `mvs_keywords` and parameters that are received, Infoprint adds those `mvs_keywords` and parameters as unchanged printer-pass-through attributes.

Notes:

- a. If you include multiple Mapping statements that use the same `mvs_definition` but map to different `infoprint_attributes`, all of them will be honored.
- b. If you include multiple Mapping statements that use the same `mvs_keyword` and map to the same `infoprint_attribute`, the last one mapped will be used.
- c. If you define a Mapping statement that maps to an `infoprint_attribute` that is also used in a Global statement, and the Mapping statement is used due to the MVS Keywords in a received file, the Mapping statement will override the Global statement.
- d. If no matching Mapping statement is found for a received `mvs_keyword` and parameter, it will automatically be added to the printer-pass-through parameter. A null Mapping statement can be used to throw the `mvs_keyword` away entirely. A null Mapping statement contains an `mvs_definition` and a control, but no `infoprint_definition`.

Syntax of control statements

Each Control Statement consists of three parts as described below. The entire statement must reside on a single line (it can't contain a carriage return until it is complete) and it must follow this syntax:

```
mvs_definition [control [infoprint_definition]]
```

mvs_definition: The *mvs_definition* identifies a keyword that is generated by the host and, for Default and some Mapping statements, its associated parameter. It follows this format:

```
mvs_keyword[=mvs_parameter]
```

Note: In Control Statements, *mvs_definitions*, *controls*, and *infoprint_definitions* themselves cannot contain any white space (spaces or tabs). The first occurrence of white space separates the *mvs_definition* from the control and the second separates the control from the *infoprint_definition*. As a result, if the elements contain white spaces, the DCF will not complete the mapping correctly.

These values are described below.

mvs_keyword: The *mvs_keyword* represents the string that Infoprint Manager receives from an MVS system. It is what the MVS system uses to refer to information about a print job (for example, whether or not it should be duplexed or what overlay it uses, if any). The keywords identified in Table 1 on page 4 are the standard set of *mvs_keywords* that are available for you to use. The keywords that MVS uses to refer to these job characteristics are not the same as the Infoprint attribute names that do the same things.

mvs_parameter: The optional *mvs_parameter* is a case-sensitive free-format field. The purpose of an *mvs_parameter* depends upon what type of Control Statement it is used with.

- **Default Statement**

The *mvs_parameter* value is required. This value identifies the default value of the *mvs_keyword* when the *mvs_keyword* has not been sent by the host system with a job. For example, in the Defaults section of the *mvsdmap.txt* file, *CHARS=GT13* is identified. Therefore, *GT13* is the *mvs_parameter* default font value, and will be used whenever Infoprint Manager does not receive a font value from the host system.

- **Global Statement**

The *mvs_parameter* value is ignored.

- **Mapping Statement**

The *mvs_parameter* value is optional. If an *mvs_parameter* value is specified for an *mvs_keyword* on a Mapping statement, the *mvs_keyword* is not always mapped to the *infoprint_definition* specified. The *mvs_keyword* is only mapped to the *infoprint_definition* when the *mvs_keyword* and the *mvs_parameter* combination that comes from the host system matches the *mvs_keyword* and *mvs_parameter* combination specified on the Mapping statement.

For example, these four lines appear in the Mapping section of the *mvsdmap.txt* file:

```
CCTYPE :: carriage-control-type=none
CCTYPE=ANSI :: carriage-control-type=ansi-ebcdic
CCTYPE=MACHINE :: carriage-control-type=machine
```

CCTYPE=ASCII :: carriage-control-type=ansi-ascii

If Infoprint Manager receives a job from the host system with the `mvs_keyword` CCTYPE=ANSI specified, Infoprint Manager will map that keyword to the Infoprint attribute `carriage-control-type` and set its parameter to `ansi-ebcdic`. If the next job comes with the `mvs_keyword` CCTYPE=MACHINE specified, Infoprint Manager will map that keyword to the Infoprint attribute `carriage-control-type` and set its parameter to `machine` instead. But, if Infoprint Manager receives a job with the `mvs_keyword` CCTYPE set to anything besides ANSI, MACHINE, or ASCII, it will map that keyword to the Infoprint attribute `carriage-control-type` and set its parameter to `none`.

For certain `mvs_keywords`, special `mvs_parameters` are recognized. These keywords are listed in Table 2.

Table 2. Special `mvs_parameters`

<code>mvs_keyword</code>	<code>mvs_parameters</code>
CC	YES NO
CCTYPE	ANSI MACHINE ASCII
DATAACK	BLOCK UNBLOCK BLKPOS BLKCHAR
DATATYPE	LINE
DUPLEX	NO NORMAL TUMBLE
FILEFORMAT	RECORD STREAM
RESFMT	P240 P300
TRC	YES NO

control: A *control* is a symbol that is used for delineation and which can also tell the MVS Download receiver what it has to do to the `mvs_keyword` if it converts the `mvs_keyword` to an Infoprint keyword. Every Mapping statement contains a control. Some of the controls listed below are used in the `mvsdmap.txt` file.

- :: Don't do any special processing. (Standard delineation)
- :+ Change all of the characters in the `mvs_parameter` into upper case when converting it to an `infoprint_parameter`.
- :- Change all of the characters in the `mvs_parameter` into lower case when converting it to an `infoprint_parameter`.
- !: Do not convert this `mvs_parameter` into an `infoprint_parameter`. This control is only necessary if no `infoprint_parameter` is specified on this statement.

Note: In Control Statements, `mvs_definitions`, `controls`, and `infoprint_definitions` themselves cannot contain any white space (spaces or tabs). The first occurrence of white space separates the `mvs_definition` from the control and the second separates the control from the `infoprint_definition`. As a result, if the elements contain white spaces, the DCF will not complete the mapping correctly.

Controls behave differently in the different types of Control statements.

- In **Default** statements, you cannot specify a control.
- In **Global** statements, you must specify a control, but the operator is only recognized as a delineator; no matter what operator you use, it acts like the `::` control.
- In **Mapping** statements, you must specify a control.

infoprint_definition: The *infoprint_definition* identifies the attribute and parameter that will be specified on the **pdpr** command when Infoprint Manager sends the job to the printer. The *infoprint_definition* identifies an Infoprint attribute and (optionally) the value that it should be set to. *infoprint_definitions* follow this syntax:

```
infoprint_attribute[=infoprint_parameter]
```

Note: In Control Statements, *mvs_definitions*, controls, and *infoprint_definitions* themselves cannot contain any white space (spaces or tabs). The first occurrence of white space separates the *mvs_definition* from the control and the second separates the control from the *infoprint_definition*. As a result, if the elements contain white spaces, the DCF will not complete the mapping correctly.

The *infoprint_attribute* is the Infoprint attribute name as it is listed in the *Infoprint Manager: Reference* under *-x* options, plus the specially recognized *target-destination-name*, which is used to identify the *-p* option of the **pdpr** command. The *infoprint_parameter* is the value that is assigned to that attribute.

Infoprint_definitions work differently in each kind of control statement.

- In **Default statements**, you cannot specify an *infoprint_definition*.
- In **Global statements**, you must specify an *infoprint_definition*.
- In **Mapping statements**, you should specify an *infoprint_definition* to map the *mvs_definition* to. However, if you specify an *infoprint_attribute*, you do not have to specify an *infoprint_parameter* with it. If you only list an *infoprint_attribute*, the *mvs_parameter* value from the *mvs_keyword* becomes the *infoprint_attribute* value.

Notes:

1. If you do not provide an *infoprint_definition* in a Mapping statement, the information contained in that **mvs_keyword** will be discarded.
 2. If you provide multiple Mapping statements that lead to the same attribute only one will be used, but it is impossible to predict which one that will be.
- To identify an *infoprint_attribute* that you want to use on the *-x* option of the **pdpr** command, omit the *-x*, for example: *media-id*.
 - To identify an *infoprint_attribute* that you want to use on the *printer-pass-through* attribute of the **pdpr** command, prefix the *infoprint_attribute* with an *-o* flag for example, *-ojobid*. All generated-*o* attributes will be collected and added to a single *printer-pass-through* attribute.
 - To identify an *infoprint_attribute* that you want to use on the *-opa* option of the *printer-pass-through* attribute of the **pdpr** command, prefix the *infoprint_attribute* with an *-opa*, for example, *-opa:class*. All generated-*opa* attributes will be collected and added to a single *-opa* keyword on the *printer-pass-through* attribute.
 - To identify that the *mvs_definition* is to map to the name of the target destination for submission, specify *target-destination-name*, which is a specially-recognized *infoprint_attribute* that overrides the third parameter that is passed to the sample MVS Download Exit.
 - To handle complex, multi-valued *infoprint_attributes*, begin the *infoprint_parameter* with as many colons (:) as necessary to move the parameter into the correct position. For example, the *results-profile* attribute follows this syntax:

```
-x "results-profile=name@node:method:number:'message text':bin"
```

In use, the attribute would look like this:

```
-x "results-profile=nr645@bld25:pickup:2:'Please read this'"
```

So, if you wanted to map the COPIES value to the 'number' parameter of results-profile, you would enter two colons (::) in the mapping statement. The two colons indicate that the program should skip over the first two parameters in the results-profile (name@node and method), and insert the value in the third position (number). The mapping statement would look like this:

```
COPIES :: results-profile=::
```

You can include a hard-coded infoprint_parameter in the statement, but you do not have to. If you do not include one, the normal rules for determining the resulting infoprint_parameter apply.

Changing the sample DCF

Copying the sample DCF: Follow these steps to create a copy of the sample DCF provided with Infoprint Manager.

1. Open the file <install path>\var\pd\mvsdmap.txt, (where <install path> is the directory where Infoprint Manager is installed) in a text editor such as Notepad.
2. Select **File** ->**Save as**, and save the file into a directory outside of the <install path> directory. Make note of the directory and file name for reference later.

Notes:

- a. You can change the file name if you like, but do not need to.
- b. If you do not save the file to another directory outside the <install path> directory, it could be overwritten or deleted if you ever need to reinstall, apply service, or upgrade Infoprint Manager.
- c. This step creates a copy of the sample/default file and leaves the original, unedited version in the directory listed above so you can use it to create another customized DCF later if you need to.

Editing the DCF: Refer to this section to edit either the sample DCF provided with Infoprint Manager or an existing DCF.

1. Open the file you want to edit in a text editor.
Notice that most of the lines begin with a *. Every line that begins with a * is a line that is *commented out*, so Infoprint Manager ignores it during processing. When you need to comment out a line, insert a * at the beginning of the line; to uncomment a line, delete the * at the beginning of the line.
2. Read the notes and instructions in the file before you make any changes to it.
3. Change the Control Statements as you need to.
 - In the **Defaults** section, you can set the defaults as you need them to be.
 - In the **Globals** section, the only value that is set is carriage-control-type attribute value.

The statements that are commented out are examples of how to specify alternate resource search orders.

- The first example specifies a location that Infoprint Manager will search for all fonts (resource-context-font=C:\fonts\reslib).
- The second example specifies a location and search order for page definitions (resource-context-pagedef=C:\joeski\pagedefs;c:\bills\pdefs).

You should put any other **pdpr** parameters to be specified on the **-x** option on all jobs here.

- In the **Mapping** section, there are two sections: *Commonly changed mappings* and *Less commonly changed mappings*. Follow the instructions provided in the text file to make your changes, keeping these additional considerations in mind:
 - If you plan to send print jobs to different logical destinations based upon the media type of the job, you should uncomment the `FORMS :: default-medium` mapping.
 - The default mapping uses the MVS JCL keyword `DEST` to specify the target destination (printer) that the downloaded jobs are sent to. If you comment out this mapping or remove the `target-destination-name` from the right side, all jobs are submitted to the target destination that you specify in the **Target destination** field when you set up your MVS Download Receiver.
 - If your host MVS system uses the newer `PRTQUEUE` JCL keyword to identify the target-destination, comment out the `DEST :: target-destination-name` mapping and uncomment the lines:

```
PRTQUEUE :: target-destination-name
DEST      ::
```

Debugging your changed DCF: The sample DCF contains a `DEBUG` statement that is commented out. Follow these steps to use this statement to check your DCF.

1. Uncomment the `DEBUG` statement in the DCF that you changed.
2. Save the file.
3. Stop the MVS Download receiver that uses this DCF.
4. Restart the MVS Download receiver that uses this DCF.
5. Submit a job from MVS that runs through this MVS Download receiver. Information will be written into the server log. You can view the server log by clicking the **Server Log** item in the left pane of the Management Console.
6. Once you are satisfied with the results, open the DCF and comment the `DEBUG` statement out again.
7. Save the file.
8. Stop the MVS Download receiver that uses this DCF.
9. Restart the MVS Download receiver that uses this DCF.

Changing the sample DCF- advanced information

If you use MVS Exit 15, understand that the predefined *mvs_keywords* are internally converted using Table 3 on page 12. If you need to add lines to the DCF you will need to be aware of these internal mappings and what additional information your lines need to contain.

For example, the sample DCF contains the line:

```
DATACK=BLKPOS :: data-fidelity-problem-reported=character
```

But, because in Table 3 on page 12 `DATACK` maps to `-odatac`, the following line would produce the same results:

```
-odatac=blkpos :: data-fidelity-problem-reported=character
```

When the MVS host actually sends `-odatac=blkpos`, this mapping statement is used. For information on the actual keywords that the MVS host sends, refer to the

publication Print Services Facility for OS/390: Download for OS/390, (S544-5624), which you can download from the IBM Printing Systems website at <http://www.ibm.com/printers>.

Note: The `mvs_definition` (all of the information to the left of the control) is case-sensitive and must exactly match the information that comes from the host system for the mapping to be made.

Some parameter information is received from the MVS host as `-o` keywords and other information is received as sub-keywords of the `-opa` keyword. For example, CLASS information is received as `-opa=class=xxx`.

When the DCF is processed by an MVS Download receiver, the receiver looks at each `mvs_keyword` to determine if it will be received from MVS as a keyword or as a sub-keyword of the `-opa` keyword. The receiver tries to convert the specified `mvs_keyword` according to Table 3. If the receiver doesn't find a mapping conversion for the specified `mvs_keyword`, the `mvs_keyword` is left unchanged. If the result of the conversion begins with `-o`, the receiver assumes that the `mvs_keyword` will be received as a keyword. If the result of the conversion does not begin with `-o`, the receiver assumes that the `mvs_keyword` will be received as a sub-keyword of the `-opa` keyword.

For example, the `mvs_keyword` SYSOUT is internally converted (according to Table 3) to `class`. Since `class` does not begin with `-o`, the receiver assumes that `class` is a sub-keyword of the `-opa` keyword.

Table 3. MVS keyword internal conversions

mvs_keyword	MVS Values Passed to Infoprint
ACCOUNT	-AC
ADDRESS1	-oaddress1
ADDRESS2	-oaddress2
ADDRESS3	-oaddress3
ADDRESS4	-oaddress4
BUILDING	-obu
CC	-occ
CCTYPE	-occtype
CHARS	-ochars
CLASS	class
COPIES	-ocop
DATAACK	-odatac
DATATYPE	-odata
DEPT	-ode
DEST	destination
DUPLEX	-odu
FCB	-opagedef
FILEFORMAT	-ofileformat
FORMDEF	-of
FORMS	forms
INTRAY	-oin

Table 3. MVS keyword internal conversions (continued)

IPADDR	-oipdest
JOBID	jobid
JOBNAME	-ojobn
NAME	-ona
NODEID	-ono
OFFSETXB	-ooffxb
OFFSETXF	-ooffxf
OFFSETYB	-ooffyb
OFFSETYF	-ooffyf
OUTBIN	-outbin
OVERLAYB	-oovlyb
OVERLAYF	-oovlyf
PAGEDEF	-opagedef
PRMODE	-oprmode
PROGRAMMER	-opr
PRTQUEUE	-oprqueue
RESFMT	-ore
ROOM	-oro
SEGMENTID	segmentid
SYSOUT	class
TITLE	-oti
TRC	-otrc
UCS	-ochars
USERID	-ous

MVS Exit 15 allows users to add sub-keywords to the `-opa` keyword only. The `mvs_keyword` that would be used in the DCF is the (case-sensitive) sub-keyword only. For example, if an MVS Exit 15 added `OUTGRP=n`, where *n* was the number of the dataset in the job, and you wanted to map this value to the `printer-pass-through=-opa=segmentid` parameter (instead of the supplied mapping), you would replace the existing DCF mapping statement of:

```
SEGMENTID      :: -opa:segmentid
```

with

```
OUTGRP         :: -opa:segmentid
```

If you wish to add multiple sub-keywords to the `-opa` keyword using MVS Exit 15, each keyword/parameter pair must be delineated with a comma. It is highly recommended that you not use any blank spaces or tabs around the comma or around the "=" sign that you place between your generated sub-keyword and the parameter.

For more information, refer to the publication *Print Services Facility for OS/390: Download for OS/390, (S544-5624)*, which you can download from the IBM Printing Systems website at <http://www.ibm.com/printers>.

Understanding the MVS Download Exit Program

The MVS Download Exit Program (or *Exit*) takes the Infoprint Manager submission parameters generated by the DCF and uses them to process the print job appropriately. The Exit is invoked once per file received from MVS.

Two sample exit routines are provided with Infoprint Manager: one written in Microsoft Visual C++ Version 6.0 and one written in Microsoft Visual Basic Version 6.0. The Visual C++ file set includes: `mvdsdsubm.dsp` and `mvdsdsubm.opt` (project files), `mvdsdsubm.dsw` (a workspace file), and `mvdsdsubm.c` (a source file). The Visual Basic file set includes `mvdsdsubm.vbp` (a project file), `mvdsdsubm.vbw` (a workspace file), and `mvdsdsubm.frm` (a source file). They are installed in the `<install path>\exits\mvdsd` directory, where `<install path>` is the directory that you installed Infoprint Manager in.

Using the sample MVS Download exit program

In the sample `mvdsdsubm` files, the Exit performs the following operations using the five parameters that are passed to it. Both of the sample Exits perform the same functions.

1. Issues a **pdpr** print command to submit the file (parameter #1) to the specified target destination (parameter #3), with the indicated Infoprint print parameters (parameter #5).
2. If the job prints successfully, the Exit will erase the print job file (parameter #1).
3. If the job does not print, the Exit will log an error in the MVS Download error file and terminate the receiver.

Parameters	Definition
#1	The name of the file that contains data received from the MVS JES spool
#2	MVS JES destination control parameters
#3	Infoprint target destination (printer) where the job is submitted
#4	MVS Download Receiver name
#5	Infoprint print parameters

Customizing the sample exit program

You can customize either the `mvdsdsubm.c` (written in Visual C++) or `mvdsdsub.frm` (written in Visual Basic) to allow the Exit to perform differently. For example, you might want to modify the Exit so that it ignores the Infoprint Manager print parameters (parameter #5) and just processes the MVS JES destination control parameters themselves (parameter #2). Or, you could remove the **pdpr** call and change it to any applicable program or OS command call. For instance, you could have the Exit copy or archive the data file (parameter #1).

Note: If you specify a DCF that contains no statements, Infoprint will pass parameter #5 as a null string.

Copying the sample exit: If you want to customize the sample Exit, you should not change the sample itself, because if you ever need to reinstall Infoprint Manager, all of your changes will be lost. Instead, copy the files that you want to use (either the Visual C++ version or the Visual Basic version) into a directory outside of the directory the you have Infoprint Manager installed in and make your changes to that version.

Note: You must have access to either Microsoft Visual C++ Version 6.0 or Microsoft Visual Basic Version 6.0 to customize the Exit.

You can follow these steps to complete the process:

1. Log on to the system that your Infoprint Manager server is installed on.
2. Decide which directory you would like to store your new exit in or create a new one for it.
3. Navigate to the <install path>\exits\mvds directory, where <install path> is the directory that you installed Infoprint Manager in, using Windows Explorer or the **My Computer** icon.

You will see seven files. Four of them are the Visual C++ fileset and the other three are the Visual Basic fileset.

4. Copy the files for the version of the Exit you want to modify into the directory you selected or created in step two.
 - For the Visual C++ version, copy: mvdsdsubm.dsp, mvdsdsubm.opt, mvdsdsubm.dsw, and mvdsdsubm.c
 - For the Visual Basic version, copy: mvdsdsubm.vbp, mvdsdsubm.vbw, and mvdsdsubm.frm
5. Open the copy of the project and make your changes.
6. Save and compile the Exit.

Note: By default, the sample Visual C++ Exit builds a **Debug** compile version of the Exit. The pre-installed executable version of the sample (mvdsdsubm.exe) is a **Release** version. You can change the Active Configuration of the **mvdsdsubm** project to build the Release version of the Exit.

Creating an MVS Download Receiver

Before you can begin sending jobs from the MVS JES spool to Infoprint Manager through MVS Download, follow this procedure to set up an MVS Download Receiver.

1. Open the Infoprint Manager Management Console.
2. On the menu bar, select **Edit** —> **New** —> **MVS Download Receiver**.
The **Add MVS Receiver** dialog opens.
3. Fill in the fields according to the following specifications. Refer to the Management Console online help for more information about the dialog and the fields it contains.
 - **Port Number:** Enter the port number that you want this receiver to use when it communicates with the host system. This number must match the port number that is specified in the Routing Control File used by the MVS host Download FSA for this computer's IP address.
 - **Target Destination:** Select the Infoprint destination that you want this receiver to submit jobs to from the drop-down list. Infoprint Manager will only submit a job to this destination when the job does not have a target-destination-name value assigned to it after its mvs_keywords have been processed against the DCF.
 - **Destination Control File:** Click **Browse** and locate the destination control file that you want to use with this receiver. Select the file and click **Save**.

- **Exit Program Name:** If you have not modified the exit program, do not change this path. If you have modified the exit program, click **Browse** and locate the exit program file that you modified and compiled. Select the file and click **Save**.
- In the **Processing Options** group box, both of the options are selected by default.
 - When the **Retain command files** check box is selected, Infoprint Manager saves all of the files that pertain to MVS Download jobs that fail in processing. When MVS Download print jobs are processed successfully, these files are cleaned up regardless of what you select here. If you need to resubmit failed jobs, you do not have to download them from the host system again; the files are still on your Infoprint server. If you deselect this box, you will have to download any jobs that fail in processing from the host again to print them.

Note: You can diagnose errors by clicking the **Server Log** item in the left pane of the Management Console.

Operators should evaluate any processing failures, correct the problems that caused the failures, and resubmit the jobs.

Refer to the Operator Procedure Deleting Orphan Files and Resubmitting Jobs to MVS Download.

- When the **Enable parallel processing** check box is selected, this MVS receiver will process multiple jobs simultaneously. When it is cleared, the receiver will wait until the Exit for one print job completes before it starts to process the next job.

This option is normally left selected unless a custom Exit has been written that requires only one instance of the exit to run at a time and in the order the files are received.

4. Click **OK**.

One or more pop-up informational messages may appear; read them and click **OK** to dismiss them.

5. Select the **MVS Download Receivers** item in the left pane of the Management Console.

You should see the receiver you just created in the right pane. The receiver status may be **STOPPED** but it should start momentarily. Click the **Refresh View** button on the toolbar until the status changes to **RUNNING**.

If the receiver does not start within a few minutes, select **Edit** → **MVS Download Manager** and manually start the receiver by selecting it and clicking **Start**.

Note: Receivers that will not start more than likely have destination control file errors. You can diagnose errors by clicking the **Server Log** item in the left pane of the Management Console.

Making your AFP resources available to Infoprint Manager

The AFP resources you use for your printing jobs must be available to Infoprint Manager so the jobs process correctly. This section assumes that these resources currently reside on your host system in AFP format, and that you (or someone) can locate them.

Note: If you currently use the Distributed Print Function (DPF) of PSF for OS/2, the resource files you use are in Intelligent Printer Data Stream (IPDS) format. However, the Advanced Function Presentation (AFP) versions of those files exist on your host system as well. Make sure that you use the AFP versions with MVS Download.

There are several ways that you can make your resources available to Infoprint Manager; three of them are described below. Read through the options and decide which is best for your configuration.

Manually transfer the resources from the host to your Windows system

You can move your resources directly from your host system using any method you like, though the most common is ftp. If you decide to ftp your files, make sure that you transfer the files as *binary* data, not as *text*. In general, the steps you should follow are:

1. Create a directory on your Windows system that will receive your resources.
2. ftp your resources to that directory as binary data.
3. Identify that directory to Infoprint Manager.

You can either identify the directory using a resource context object or by changing the default resource search path in the Management Console. For instructions on creating a resource context object, refer to the Administration Procedure Creating a New Resource-Context Object. For information on changing the default resource search path, consult the online help for the Management Console under **Service Configuration**.

This method is effective and relatively easy to implement. However, every time you change your AFP resources, you will have to ftp them to the correct directory on your Windows system. If you do not change your resources often, this method may work well for you. If you change your AFP resources often, you may want to consider a different option.

Store your resources in a central location

Another option is to store your AFP resources on any machine in your network (including the host system that they currently reside on), and NFS mount a directory on the Windows system that your Infoprint Manager server is installed on to that machine. Then, identify that directory to Infoprint Manager just as in the option above (either by using a resource context object or by changing the default resource search path). To implement this option, you will need to install NFS server and client software, which is available from a variety of vendors.

The main drawback to this solution is that it generates a significant amount of network traffic as the system retrieves the resources. The increase in traffic and time needed to retrieve the resources when needed could cause printer clutching.

Move the resources using MVS Download

A third option is to move your AFP resources using the MVS Download feature of Infoprint Manager itself. You can achieve this by setting up an MVS Download receiver that receives jobs and copies them into a particular directory without printing them. You would only use this receiver to move your resources. Setting up this configuration is different depending on your environment. Each time you modify a resource, you submit a job that sends the modified resource to each system that you are running an Infoprint Manager server on. To accomplish this task, you may need to customize the JES inish deck, the Host MVS Download routing control data set, the Destination Control File (DCF), and the Exit Program. The modifications needed will depend on your existing MVS Download

configuration used for sending jobs, although the process of getting the AFP resources to the Infoprint Manager computer is the same in all cases:

1. Place the resource on the spool for MVS Download to process with identification that it is an AFP resource, the resource name, and the desired target path. (Target path is optional)
2. Set up the routing control data set to send the AFP resources to unique MVS Download receivers on the Infoprint Manager system. (Optional)
3. Set up the routing control data set to not prefix each record with a record length field for AFP resources.
4. Set up the DCF to map the desired resource path/name to the target-destination-name. (Optional)
5. Modify the supplied Exit to copy the received AFP resource into the proper resource directory with the provided resource name. The full functionality of this Exit would depend on the options chosen above. For example:
 - a. If the target path was not supplied, the Exit would either need to have it hardcoded or determine it using other means.
 - b. If both AFP resources and print jobs are being sent to the same receiver, the exit would need to distinguish between the two by some mechanism and take the proper action for each.
 - c. If the DCF does not map the resource path/name to the target-destination-name, it will either need to parse it from the other parameters or determine it using other means.

This option is more difficult to set up than the other two, but is less time consuming once you have it implemented, because every time you change your AFP resources all you have to do is resubmit a job for transmission to your Windows systems. If you have multiple Windows systems to update or can further automate this process, this method may work well for you. One possible configuration example follows.

Example: Assume that JES is configured to schedule all jobs with DEST=NTPRT1 to an MVS Download FSA that downloads them to an Infoprint Manager system that then uses the default supplied DCF and Exit to submit the jobs to an IPDS printer, **ntprt1**. The MVS Download FSA uses a routing control data set that contains the following statement:

```
DEST=NTPRT1
IPADDR=dept01.myststem
PORTNUM=5002
RETRYNUM=3
RETRYINTV=60
```

We have decided the following about the configuration:

- Leave the existing MVS Download receiver (on our Windows system) alone.
- Create a new receiver to processes the AFP resources, but continue using the existing MVS Download FSA.
- Pass only the AFP resource name, not the directory path for it to be stored in.
- Use the FORMS keyword to identify the job as an AFP resource since the Routing Control Statements can use FORMS to direct files to different receivers.
- Use the NAME JCL keyword to identify the AFP resource name because it sounds right.

In the JCL job for putting modified AFP resources on the spool, we have an OUTPUT statement with the following:

```
OUTPUT      DEST=NTPRT1,FORMS=AFPRES,NAME='COMYFNT.300',....
```

No changes to the JES inish deck are required since the MVS Download FSA already receives jobs with DEST=NTPRT1.

We change the routing control file as follows to send this job to a separate receiver and to prefix these resources with no length field. Jobs that do not have FORMS=AFPRES would continue to be sent to the existing receiver.

```
DEST=NTPRT1
FORMS=AFPRES
IPADDR=dept01.mysystem
PORTNUM=5003
SEND_REC_LENGTH=NO
RETRYNUM=3
RETRYINTV=60
DEST=NTPRT1
IPADDR=dept01.myststem
PORTNUM=5002
RETRYNUM=3
RETRYINTV=60
```

Now we need to define a new receiver on the dept01.mysystem machine. The new receiver will use port number 5003 (as identified in the routing control statement), a modified DCF, and a user-written Exit.

We make a copy of the supplied DCF and modify it to change the mapping of the NAME keyword from MVS. The existing statement:

```
NAME :: name-text
```

would be changed to:

```
NAME :: target-destination-name
```

Also, the existing statement that maps DEST to target-destination-name would be commented out as follows:

```
* DEST :: target-destination-name
```

Due to this change in the DCF, the Exit will now receive the AFP resource name as the third passed parameter.

Next, we make a copy of the supplied Exit (either the Visual Basic or Visual C++ version). We modify the Exit so it copies the received file instead of submitting it to Infoprint Manager using **pdpr**. The Exit must use the first passed parameter as the name of the resource file and the third passed parameter to derive the target resource file name, because the DCF mapped it that way. Since we did not pass the desired resource directory name in the JCL, the Exit needs to determine the drive and directory and pre-pend it to the target resource file name. Since this Exit will only be used on a receiver that receives AFP resources, it doesn't need to identify the file as an AFP resource and have a paths for both copying resources and submitting non-resources to Infoprint Manager to be printed. The Exit can also be coded to detect and react to error situations as appropriate.

Lastly, we use the Management Console to create the receiver that will use the DCF and Exit that we created. In this example, the **Target Destination** that we specify when we create the receiver is irrelevant and will only be passed to the Exit if no NAME is specified in the JCL that submitted the resource.

Back to Administrator procedures