

# **NCS ARCHITECTED INTERFACES**

## **EXTERNAL SPECIFICATIONS**



June 6, 1989

\* HP Confidential \*

Copyright © 1989 HEWLETT-PACKARD COMPANY



# PRODUCT IDENTIFICATION

SECTION

1

## 1.1 PRODUCT NAME

New Commercial Spooler Architected Interfaces

## 1.2 PRODUCT MNEMONIC



NCS AIF

## 1.3 PRODUCT ABSTRACT

The New Commercial Spooler (NCS) is the new native mode spooler for the MPE XL operating system. It is scheduled for release concurrent with MPE XL Release 2.1, which is currently planned for second quarter of 1990. NCS will be part of FOS and will replace the existing spooling system, which currently consists of Module 79, Module 80, and the SPOOK utility. In addition to its external feature set described in the New Commercial Spooler External Specifications, special interfaces will be provided to allow access to MPE XL internal spooler information and functionality via Architected Interfaces.

Architected Interfaces (AIFs) are a series of services which provide competitive interfaces for use by software suppliers and internal or external solutions creators. They allow easy and high performance access, manipulation, or interception of HP proprietary operating system and subsystem information and/or processes. Through these supported and documented interfaces, supportability and reliability of MPE XL and MPE XL software products will be increased.

The NCS AIFs are currently scheduled to be available concurrent with release 2.1 and release of the New Commercial Spooler. This document describes the NCS subset of the Architected Interfaces project.

**HP Computer Museum**  
**[www.hpmuseum.net](http://www.hpmuseum.net)**

**For research and education purposes only.**



## 2.1 OVERVIEW

NCS AIFs provide a software interface between non-OS software and the New Commercial Spooler functionality of MPE XL. It provides controlled access to NCS functionality and NCS internal data structures, i.e. the Spooler Process Information Table and the Spoolfile Directory. This software interface, executing at User Privileged Mode, provides a window into NCS internal operations. User Privileged Mode, which is also called HPPA Hardware Level 2, maps directly to MPE V Privileged Mode.

The AIFs abstract internal data structures, allowing developers to access data items by name or number. This abstraction frees the developer from the pain of rewriting an application whenever the operating system internals change. This should equate into higher reliability and supportability for software products utilizing these interfaces.

## 2.2 DEFINITION

The first release of AIFs contains a set of interfaces to the MPE XL operating system providing two types of access: information access and functionality access. The information access interfaces for NCS AIFs provide read/write access to the Spoolfile Directory (SPFDIR) and the Spooler Process Information Table (SPIT). The functionality access interfaces provide users with the capability to control spooling processes and to manage spoolfiles. These two types of interfaces are described in more detail in the following paragraphs.

### 2.2.1 Information Access

The information access interfaces provide access to system tables. A list of example information access interface groupings are provided below.

- System Configuration Information
- System Wide Information
- Device Information
- Directory Information
- Global File Information
- Job/Session Information
- Local File Information
- Ports Information
- Process Information
- Spooler Information

## 2.2.2 Functionality Access

The functionality access interfaces provide access to different operating system functionalities. A list of example functionality interfaces is provided below.

- Lock Management
- Device Management
- High Level IO Management
- Ports Management
- Spooler Management

The Spooler Management functionality interfaces will be composed of the following NCS AIFs:

- aif\_\_spp\_\_openq
- aif\_\_spp\_\_shutq
- aif\_\_spp\_\_start
- aif\_\_spp\_\_stop
- aif\_\_spp\_\_suspend
- aif\_\_spp\_\_resume
- aif\_\_spp\_\_release
- aif\_\_spf\_\_link
- aif\_\_spf\_\_list

## 2.3 PRODUCT ENVIRONMENT

### 2.3.1 Hardware Requirements and Restrictions

The NCS AIFs will run on HP Precision Architecture Series 900 HP3000 systems. They will not run on the Classic HP3000 family systems. The peripherals supported will be all spoolable peripherals that are supported on the HPPA 3000 systems.

### 2.3.2 Software Requirements and Restrictions

The NCS AIFs are scheduled to be generally available concurrent with MPE Release 2.1, or in second quarter of Fiscal Year 1990.

Supported languages for NCS AIFs are Pascal and C.

This section describes the two types of interfaces provided with NCS AIFs, the NCS information access AIFs and the NCS functionality access AIFs.

Also described in this section is a system wide information access interface which provides a global view of the current spoolfiles on the system.

In the last part of the section, we describe the data types used for NCS AIF, system logging, and error checking conventions. A glossary of terms is also included for reference.

## 3.1 INFORMATION ACCESS AIFS

The NCS AIFs for information access provide direct access to the NCS internal tables. The design of the AIFs is to abstract the internal tables and provide uniform interfaces. The class of NCS objects about which information can be accessed is comprised of the Spooler Process Information Table (SPIT) and the Spoolfile Directory (SPFDIR). For each of the tables, there are two AIF procedures: a *Get* and a *Put*. The *Get* interface will return information identified by the input keys for a specific spooling process or spoolfile. The *Put* interface will accept input from the user and update relevant tables to obtain a consistent, updated state of the system. The *Put* interface will allow the user to specify the same keys as the corresponding *Get* interface, however it will allow access to only a subset of the items provided by the *Get* interface. NCS AIF is assigned a range of 1000 numbers, from 8000 to 8999, to identify its information access items and errors.

Each information access procedure will attempt to lock all the tables associated with that object. Thus, for spoolfiles, the GUFID and the SPFDIR will be locked and for spooling processes, the SPIT will be locked. Locking may also be done explicitly by the user via general Architected Interfaces.

The information access procedures will probe any address before they write to it. For system tables, they will ensure that the values being placed into the tables fall into the correct range. In addition to routine checks, these procedure will try to enforce some amount of consistency before updating any system tables.

## 3.2 FUNCTIONALITY ACCESS AIFS

This class of NCS AIFs provides functionality for managing spoolfiles as well as spooling processes. It provides the means of starting, stopping, resuming, suspending and releasing devices. It also allows users to link files into the spooling system.

## 3.3 SYSTEM WIDE INFORMATION ACCESS

This is a special kind of interface. It provides global information on the current state of the system. The data returned by this interface are actually keys that can be passed to other information access interfaces



so that more detailed information can be acquired. It will probably be the first procedure called in most system management applications.

On calling the procedure, the user can specify an object class and a list of criterion to be applied to the objects in the class. The procedure will apply the selection criterion on each object in the specified class and return only names of qualifying objects. A list of fast access keys can be optionally returned. These keys are known to other AIFs and can be passed for fast access.

An additional feature is that a context-key is returned if there are more qualifying keys than provided for in the user defined array. A subsequent call to the system wide interface could use this key as the starting point of a search.

A user can call this routine, for example, to ask for all spoolfiles on the system targeted for LDEV 6 and have priority higher than 8. In this case, the procedure will return a list of spoolfile numbers and the UFID of the spoolfiles to be printed on LDEV 6 and has priority higher than 8.

### 3.4 DATA TYPES

Most of the information exchange across the NCS AIFs is accomplished through the use of scalar types, which do not require any special handling. The scalar types include integers, long integers, short integers, character arrays and booleans.

For record types, the documentation will provide the Pascal record declaration as well as the packing of the fields as implemented on HP Pascal compiler. This information should suffice to make the call usable from Pascal and C. The packing information will be kept up to date through timely documentation updates.

Listed below are the generic data types and their mnemonics that are used to declare the data types for the parameters and the values returned.

Mnemonic	Generic Data Type
I32	32-bit signed integer.
I64	64-bit signed integer.
BIT32	32-bit signed integer, or data that takes 32 bits of storage.
BIT64	64-bit signed integer, or data that takes 64 bits of storage.
B	Boolean.
C	Character.
@32	32-bit address.
@64	64-bit address.
A	Array. Used in combination with other types. For example, CA represents an array of elements, each element containing an ASCII character value; BA represents an array of elements, each element containing a boolean value.
Rec	Record.

### 3.5 SYSTEM LOGGING

System Logging, which is a MPE logging facility for recording subsystem errors and resource usage, will be used to produce an audit trail of all NCS AIFs information *put* and NCS AIFs functionality accesses. This will increase AIFs supportability by providing trace information for use in debugging.

### 3.6 ERROR MANAGEMENT

The NCS AIFs have two parameters for the returning of status information. One of them is an integer called *overall\_status*, which indicates the status of the call, on the whole. The other parameter is an array of integers called *item\_status\_array*. This array returns status information on each individual item to be transferred. It is returned by all the information access procedures.

The *overall\_status* is an integer which obeys most of the conventions of *hpe\_status*. A zero return indicates normal execution. A positive number indicates the index of the last item which caused an error. More than one error can be returned in the *item\_status\_array*, but it is the user's responsibility to walk the whole status array to look for them. A negative number indicates an error condition for the overall

call. The rightmost halfword contains the AIF subsystem number. The leftmost halfword encodes the actual error number.

The *item\_\_status\_\_array* is returned by the information access AIFs. In case of error, an element of this array will return an error which follows all the conventions of *hpe\_\_status*. A status of zero indicates normal execution, a negative number indicates an error and a positive indicates a warning. Typical errors returned in this array would be: invalid item number, protection violation and value for *put* not in its legal range.

## 3.7 GLOSSARY OF TERMS

The following are terms and names of the data types used in this ES.

### Terms

**BANNERS** The generic term for either the header or trailer of a printout. It contains the identification information for the listing.

**DATA FILE** A spoolfile that has been input from a device or STREAMed using the :DATA command. The data in it will be used later by an interactive session or a batch job.

**DEVICE\_NAME\_TYPE** This is the data type used to interface with the spooler and spoolfile management routines. It is declared as *device\_\_name\_\_type* = packed array [1..18] of char in Pascal.

**FILENAME\_TYPE** Most of the AIF interfaces accept and return file names in a single standard format. This is a fully qualified file name that is returned. Its format is as follows :

```
filename_type = record
    filename : packed array[1..16] of char;
    group    : packed array[1..16] of char;
    account  : packed array[1..16] of char;
end;
```

All the three arrays are always returned padded with blanks on the right. Most interfaces accept them, when a file name is input, to be flushed with blanks on the right.

**ITEM\_ARRAY\_TYPE** This will be a user defined data type. The interfaces expect this type to be of the form : *Array [1..x] of globalanyptr*. Here *x* could be any number.

**ITEMNUM\_ARRAY\_TYPE** This will be a user defined type. The interfaces accept this type to be of the form : *Array [1..x] of integer*. Here *x* could be any number.

**ITEMSTATUS\_ARRAY\_TYPE** This will be a user defined type. The interfaces accept this type to be of the form : *Array [1..x] of integer*. Here *x* could be any number.

**LINKED SPOOLFILE** A spoolfile that has an entry in the SPFDIR and resides in the HPSPOOL

account. Input spoolfiles are in @.IN.HPSPOOL. Output spoolfiles are in @.OUT.HPSPOOL. If a user copies a spoolfile from OUT.HPSPOOL to his local group and account, that copy has no entry in the SPFDIR and is therefore not a linked spoolfile. See Spooler management routine AIFSpoolfLink for more details.

## NCS

The New Commercial Spooler. This is the new MPE XL native mode spooler, which will replace the existing CM SPOOLER and SPOOK. It is scheduled for release on MPE XL 2.1.

## PRIVATE SPOOLFILE

A spoolfile which is created with the PRIVATE option specified. The PRIVATE option is intended for applications which produce sensitive output. Private spoolfiles have more stringent access and attribute restrictions than non-private spoolfiles.

For more detailed description of private spoolfiles, see section 3 and section 5 of the NCS ES.

## RETURN\_ARRAY\_TYPE

The system wide interface returns values in arrays of this type. The actual structure of the array may vary depending upon the kind of keys passed, but the general form is : *Array [1..x] of appropriate type*. Here *x* is any integer and *appropriate type* is specified in AIFSysWideGet.

## SEARCH\_KEY\_TYPE

The system wide interface returns arrays of keys. The number of keys returned in a call depends upon the space allocated by the user. If more keys can be returned then this is indicated in status. Also, a special key is returned which can be used in a call to AIFSysWideGet to start the scan from that search key and not repeat the keys returned before. The type of the search key depends upon the other parameters input to the interface.

## SELECTION EQUATION

A method of selecting, or filtering, one or more spoolfiles from a larger group of spoolfiles according to user-specified criteria. For example, it is possible to select all spoolfiles created by a specific user or which are in a particular state (or both). The selection is not an independent command but is a feature of the NCS commands :LISTSPF and .SPOOLF. Further details may be found in the NCS ES.

## SPFDIR

Spoolfile Directory. This is the internal table used by the New Commercial Spooler to keep information about spoolfiles.

## SPF\_ID\_TYPE

Most NCS AIF routines accept and return the spoolfile ID in variables of this data type. It takes 32 bits of storage. The first 31 bits specify the spoolfile ID number in the form of a 31-bit positive integer and the last bit indicates whether it is an input or an output spoolfile, 0 for input and 1 for output. Its format is as follows:

```
SPF_ID_TYPE = packed record
  case boolean of
    true: ( id_number : bit31;
           i_or_o_flag : bit1 );
    false:( all       : integer);
  end;
```

## SPIT

Spooling Process Information Table. This is the internal table used by the New Commercial Spooler to keep information about spooling processes.

## Product Organization

### *SPOOLER*

A program that manages input from or output to non-sharable devices so that they appear to be shared among several users. The input spooler collects data from an input device (usually a tape drive) and places it in a disc file for later use by a CI or use process. The output spooler builds an output spoolfile on disc, and oversees the orderly selection and printing of spoolfiles.

### *SPOOLFILE*

The spoolfiles generated by the file system for the NCS are ordinary disc files. This prevents input and output spoolfiles from being lost during system boots as they are today. A new file type identifies the files as spoolfiles and allows them to be managed as such. Two new file codes have also been assigned, 1515 for input spoolfiles and 1516 for output spoolfiles. Input spoolfiles are created in the IN group of the reserved account HPSPOOL and output spoolfiles in the OUT group of the HPSPOOL account.

### *SYSTEM LOGGING*

A facility to record the occurrence of specific events and system resource usage into the system log files on a job/session basis. System logging types may be enabled or disabled by the System Manager.

### *UFID*

It stands for Unique File Identifier. It is a unique handle to a single file throughout the life of a system. It is unique even across system boots. It has the following record declaration.

```
ufid_type = record
    ufid : packed array[1..20] of char;
end
```

# DETAILED SPECIFICATIONS

SECTION

4

This section provides the detailed specifications for the NCS AIF Product.

## 4.1 INFORMATION ACCESS

The detailed specifications of the information access procedures is organized as follows. A subsection is devoted to each information access procedure. For each procedure, we first describe the calling sequence and the data type of the parameter accepted by the procedure. Then, a list of all the items returned or accepted by the procedure is documented. For each item, the corresponding item number and the type of the data returned and expected are described in detail. For *Put* interfaces, a range check will be conducted on each item.

## 4.1.1 System Wide Information

The System Wide AIF is used to obtain keys (UFIDs, PIDs, ...) that are needed for accessing other AIFs. For this reason it differs in style and usage from information access and functionality access AIFs.

### 4.1.1.1 CALLING SEQUENCE.

#### Procedure AIFSysWideGet(

```

var      aif__area           :integer;
anyvar   return__array1     :return__array__type;
anyvar   return__array2     :return__array__type;
anyvar   num__array__entry  :integer;
var      overall__status    :integer;
anyvar   crit__item__array  :itemnum__array__type;
anyvar   crit__item__v__array :item__array__type;
anyvar   crit__item__st__array :item__status__array__type;
anyvar   search__key        :search__key__type)

```

```

option uncheckable__anyvar
extensible__gateway 5
default__parms( return1__array=nil, return__array2=nil,
                num__array__entrys=0, search__key=nil);

```

*aif\_\_area* (I32) This key specifies the AIF area for which system wide information is required.

*return\_\_array1* (@64A) This array of user definable size contains pointers to keys which can be used to access other AIFs. If a nil address (the default value) is passed in this parameter, no keys will be returned.

*return\_\_array2* (@64A) This array of user definable size contains pointers to keys which can be used to access other AIFs. If a nil address (the default value) is passed in this parameter, no keys will be returned.

*num\_\_array\_\_entrys* (I32) The number of entrys returned in *return\_\_array1* or *return\_\_array2*. If *return\_\_array1* and *return\_\_array2* were nil pointers the returned value would be the number of instances of the required *aif\_\_area* and *crit\_\_item\_\_array* (optional) parameters.

*overall\_\_status* (I32) This will hold the overall status of the call. A negative value indicates an error in the overall call, not specific to any item. A 0 indicates no error. A positive number indicates an index into the *crit\_\_item\_\_st\_\_array*, indicating the first item signaling an error condition.

*crit\_\_item\_\_array* (I32A) This array will denote the items on which selection criteria must be applied. The valid items will depend upon the *aif\_\_area* passed. For example, if information is desired about processes, then the criteria may be process-state, capabilities, etc.

*crit\_item\_v\_array (@32A)* This array contains pointers to the values of the criteria items selected in *crit\_item\_array*.

*crit\_item\_st\_array (I32A)* This array will return status about each of the selection criteria. It will indicate error conditions when the criteria values are incorrect, if the criteria is no longer supported, etc.

*search\_key (@32)* In the event the *return\_array1* and *return\_array2* are not large enough to contain all the returned values of the specified criteria, a *search\_key* will be returned. On a subsequent *AIFSysWideGet* call the *search\_key* would eliminate collating values which have already been returned. The key will be of the type contained in *return\_array1* (or *return\_array2* if the first is nil).

**AIF\_\_AREA Keys and Returned Array Information**

aif_area num	aif_area name	return_array1	return_array2
1000	Job/Session	PID	Job/Session #
2000	Process	PID	
5000	File	UFID	File name
6000	Directory Objects		Directory Name
8000	Spoolfiles	UFID	Spoolfile num
11000	Ports	Port ID/Message ID	Port name

**Examples :**

To obtain the number of output spoolfiles with priority > 7:

```

Procedure AIFSysWideGet(
    aif_area,                = [ 8000 {SPOOLFILES} ]
    nil,
    nil,
    num_array_entries,
    overall_status,
    crit_item_array,        = [ 8502 {Output Priority}, 8502 ]
    crit_item_v_array,      = [ 8,14 ]
    crit_item_st_array);
    
```



## 4.1.1.2 CRITERIA DESCRIPTIONS .

## System Wide Spoolfile Criteria Items

Num	Name (Type) & Description																																																												
8501	<p><b>File State(I32)</b></p> <p>The state of the spoolfile. The valid states are:</p> <table border="1"> <thead> <tr> <th data-bbox="320 667 392 695">FLAG</th> <th data-bbox="448 667 535 695">STATE</th> <th data-bbox="560 667 617 695">I/O</th> <th data-bbox="642 667 776 695">SPOOLFILE</th> <th data-bbox="816 667 934 695">MEANING</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>= Open</td> <td></td> <td>I</td> <td>Job or data file being accessed</td> </tr> <tr> <td>1</td> <td>= Active</td> <td></td> <td>I</td> <td>Job/data file inputting from inspool dev</td> </tr> <tr> <td>2</td> <td>= Create</td> <td></td> <td>O</td> <td>Device opened and being written to</td> </tr> <tr> <td>3</td> <td>= Defer</td> <td></td> <td>O</td> <td>Defer option specified for file</td> </tr> <tr> <td>4</td> <td>= Ready</td> <td>I/O</td> <td></td> <td>File ready to be accessed or printed</td> </tr> <tr> <td>5</td> <td>= Transfer</td> <td></td> <td>O</td> <td>File being transferred</td> </tr> <tr> <td>6</td> <td>= Print</td> <td></td> <td>O</td> <td>Outspoolfile being printed</td> </tr> <tr> <td>7</td> <td>= Problem</td> <td></td> <td>O</td> <td>Abnormality preventing file from printing</td> </tr> <tr> <td>8</td> <td>= Del_Pending</td> <td></td> <td>O</td> <td>File to be deleted after closing</td> </tr> <tr> <td>9</td> <td>= Retain</td> <td></td> <td>O</td> <td>Copies printed, SPSAVE option specified</td> </tr> <tr> <td>10</td> <td>= (Reserved)</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FLAG	STATE	I/O	SPOOLFILE	MEANING	0	= Open		I	Job or data file being accessed	1	= Active		I	Job/data file inputting from inspool dev	2	= Create		O	Device opened and being written to	3	= Defer		O	Defer option specified for file	4	= Ready	I/O		File ready to be accessed or printed	5	= Transfer		O	File being transferred	6	= Print		O	Outspoolfile being printed	7	= Problem		O	Abnormality preventing file from printing	8	= Del_Pending		O	File to be deleted after closing	9	= Retain		O	Copies printed, SPSAVE option specified	10	= (Reserved)			
FLAG	STATE	I/O	SPOOLFILE	MEANING																																																									
0	= Open		I	Job or data file being accessed																																																									
1	= Active		I	Job/data file inputting from inspool dev																																																									
2	= Create		O	Device opened and being written to																																																									
3	= Defer		O	Defer option specified for file																																																									
4	= Ready	I/O		File ready to be accessed or printed																																																									
5	= Transfer		O	File being transferred																																																									
6	= Print		O	Outspoolfile being printed																																																									
7	= Problem		O	Abnormality preventing file from printing																																																									
8	= Del_Pending		O	File to be deleted after closing																																																									
9	= Retain		O	Copies printed, SPSAVE option specified																																																									
10	= (Reserved)																																																												
8502	<p><b>Priority (I32)</b></p> <p>The output priority of the spoolfile which is a value from 0 to 14.</p> <p>This criteria will return the spoolfile number or UFID for all spoolfiles with priority = to the specified number. To request a range use consecutive criteria. The first priority will be the lower &gt;= limit and the second priority the upper &lt;= limit.</p>																																																												
8504	<p><b>Disposition (I32)</b></p> <p>Whether the output spoolfile is to be saved or purged after it has been printed. The values are:</p> <ul style="list-style-type: none"> <li>1 = Save</li> <li>2 = Purge (default)</li> </ul>																																																												

## System Wide Spoolfile Criteria Items (cont.)

Num	Name (Type) & Description
8509	<p><b>Stdlist of Aborted Job (I32)</b></p> <p>Whether the spoolfile is the stdlist of an aborted job. 1 for yes and 0 for not stdlist or not stdlist of aborted job.</p>
8511	<p><b>Copies (I32)</b></p> <p>The total number of copies to be printed for the spoolfile.</p> <p>This criteria will return the spoolfile number or UFID of all spoolfiles with copies = the number specified. To specify a range use consecutive criteria. The first number of copies will be the lower &gt;= limit and the second number of copies will be the upper &lt;= limit.</p>
8512	<p><b>Ready Date (I32)</b></p> <p>The calendar date when the spoolfile was created. The format returned in the 32 bit integer is:</p> <p style="padding-left: 40px;">Bits(0:16) - Unused Bits(16:7) - The year of the century Bits(23:9) - The day of the year</p> <p>This criteria will return the spoolfile number or UFID of all spoolfiles with the specified calendar date. To request a range of dates use consecutive Ready Date criteria. The first date will be the lower &gt;= limit and the second date the upper &lt;= limit.</p>
8514	<p><b>Number of Pages(I32)</b></p> <p>The number of pages in the spoolfile. A positive number indicates the actual number of pages in the spoolfile. A negative number indicates that the spoolfile has never been printed before and the number is only an estimation.</p>
8516	<p><b>Owner User/Account (CA32)</b></p> <p>The user and account name of the creator of the spoolfile. The first 16 bytes is the user name and the second 16 bytes is the account name. The names should be left justified and blank padded. Currently, only the first 8 bytes of each field is used.</p>

## System Wide Spoolfile Criteria Items (cont.)

Num	Name (Type) & Description
8517	<p><b>Job/Session #(Bit32)</b></p> <p>The job/session number under which the spoolfile was created. A job/session number with a "" indicates that the job or session is not current to the system, but rather that of a spoolfile restored onto the current system. The format returned in the 32-bit integer is:</p> <p style="text-align: center;">Bits(0:2) - S'/S/J/J'</p> <p style="text-align: center;">Bits(2:30) - The job/session number</p> <p>The value of the first two bits indicates:</p> <p style="text-align: center;">0 = S'</p> <p style="text-align: center;">1 = S</p> <p style="text-align: center;">2 = J</p> <p style="text-align: center;">3 = J'</p>
8518	<p><b>Job Name (CA16)</b></p> <p>The job name under which the spoolfile was created.</p>
8519	<p><b>File Designator (CA16)</b></p> <p>The file designator of the spoolfile, i.e., the file name of the spoolfile.</p>
8520	<p><b>Target Device (CA18)</b></p> <p>An identifier (of <i>device__name__type</i>) specifying the destination logical device name/number/class of the spoolfile.</p>
8525	<p><b>Forms ID (CA16)</b></p> <p>The forms ID of the spoolfile.</p>
8528	<p><b>Number of Records (Bit32)</b></p> <p>This criteria will return the spoolfile number or UFID of all spoolfiles with number of records = the specified number. To request a range use consecutive number of records criteria. The first number will be the lower &gt;= limit and the second number the upper &lt;= limit.</p>
8600	<p><b>Input/Output (B)</b></p> <p>Whether the spoolfile is an input or an output file.</p>

## 4.1.1.3 ERROR MESSAGES .

## AIFSysWideGet Error Messages

AIFERR	1) Message 2) Cause 3) Action
-1	1) Caller not in privileged mode. 2) The process calling this interface was at HW ring level 3. 3) Call GETPRIVMODE to promote process to HW ring level 2.
-3	1) Unable to access item <i>n</i> 's status in <i>item__st__array</i> . 2) The <i>item__st__array</i> had fewer items than <i>itemnum__array</i> . 3) Make <i>item__st__array</i> have as many items as <i>itemnum__array</i>
-4	1) AIFs not enabled. 2) The MPE XL AIF Product is not enabled. 3) Use AIFAccessOn with appropriate VendorId.
-18	1) Invalid <i>aif__area</i> key . 2) The <i>aif__area</i> indicated does not exist. 3) See AIFSysWideGet parameter definition of <i>aif__area</i> .

## 4.1.2 AIF\_\_SPP\_\_GET

This procedure allows access to information about spooler processes. It accepts a device name or device number which identifies the spooler process that the information is to be returned on.

### 4.1.2.1 CALLING SEQUENCE .

Procedure aif\_\_spp\_\_get(

<i>anyvar</i>	<i>itemnum__array</i>	: <i>itemnum__array__type</i> ;
<i>anyvar</i>	<i>item__array</i>	: <i>item__array__type</i> ;
<i>anyvar</i>	<i>item__status__array</i>	: <i>item__status__array__type</i> ;
<i>var</i>	<i>overall__status</i>	: <i>integer</i> ;
<i>readonly</i>	<i>spooler__device</i>	: <i>device__name__type</i> )

option extensible 5  
uncheckable\_\_anyvar;

*itemnum\_\_array* (I32A) An array of integers where each element is the item number of an item to be returned. If there are *n* item numbers being requested, element *n+1* must be a zero to indicate the end of the element list.

*item\_\_array* (@64A) An array of *n* long pointers where each element is a long pointer that points to an area of the type appropriate for the corresponding item number in the *itemnum\_\_array*. The information to be retrieved is placed in these areas.

*item\_\_status\_\_array* (I32A) An array of *n* integers where each element will hold on output, the status for the corresponding item. A zero indicates no error, a negative value indicates an error condition and a positive value indicates a warning.

*overall\_\_status* (I32) This will hold the overall status of the call. A negative value indicates an error in the overall call, not specific to any particular item. A zero indicates no error. A positive number indicates an index into the *item\_\_status\_\_array*, indicating the first element signaling an error condition.

*spooler\_\_device* (*device\_\_name\_\_type*) The device name or logical device (ldev) number owned by the spooler process for which information is desired. A logical device number must be converted into ASCII character string before being passed to this routine. The device name should be left justified and blank padded.

## 4.1.2.2 ITEM DESCRIPTIONS .

## Spooler Process Get Items

Num	Name(Type) & Description
8001	<b>Ldev Number(I32)</b> Returns the LDEV number of the device.
8002	<b>Process Pin(I32)</b> Returns the PIN of the Spooler process for the given device.
8003	<b>Current Spoolfile ID(Bit32)</b> Returns the integer Spoolfile ID of the Spoolfile that is currently being printed on the device.
8004	<b>Process Kind(I32)</b> Returns the process kind of the Spooler. The values are:  1 = Inspool 2 = Outspool
8005	<b>Process State(I32)</b> Returns the state of the spooler process. The states are:  0 = Initialization 1 = Release 2 = Start 3 = Stop 4 = Stop_Pending 5 = Suspend 6 = Suspend_Pending 7 = Resume 8 = Active 9 = Shut_Pending 10 = Idle

## Spooler Process Get Items (contd.)

Num	Name(Type) & Description
8006	<p><b>Finishing Strategy(I32)</b></p> <p>Returns the finishing strategy of the device. This is one of the options that can be specified for the SPOOLER xx;STOP/SUSPEND command. It is only valid when the spooler process is being suspended or stopped. The possible values are:</p> <p style="padding-left: 40px;">0 = None (device not being suspended/stopped)  1 = Now (default)  2 = End of Copy</p>
8007	<p><b>Banner Strategy(I32)</b></p> <p>Returns whether banner is to be printed in between copies of the same document. AROUND means print header before the first copy and trailer after the last copy. BETWEEN means print header and trailer for every copy.</p> <p style="padding-left: 40px;">0 = Between (default)  1 = Around</p>
8008	(reserved)
8009	<p><b>Device Outfence(I32)</b></p> <p>Returns the current outfence of the device. The range is 0 through 14.</p>
8010	<p><b>Suspend keep flag(I32)</b></p> <p>This is one of the options that can be specified for the SPOOLER xx; SUSPEND command. The field is valid only when the spooler process is being suspended. The return is whether the spooler is to retain ownership of the spoolfile or to close the spoolfile and return it to the READY state.</p> <p style="padding-left: 40px;">0 = None (suspend not currently in effect)  1 = Keep (default)  2 = NoKeep</p>



## 4.1.2.3 ERROR MESSAGES .

## Error Messages for AIF\_SPP\_GET

AIFERR	1) Message 2) Cause 3) Action
-1	1) Caller not in privileged mode. 2) The process calling this interface was at HW ring level 3. 3) Call GETPRIVMODE to promote process to HW ring level 2.
-2	1) Could not lock the required resources. 2) The required resources are locked up by the system. 3) Call AIFLockGet to explicitly lock, or try again.
-8001	1) Invalid item number passed in <i>itemnum_array</i> . 2) A non-zero, invalid item number was passed in <i>itemnum_array</i> . 3) Pass an appropriate value and end with a zero.
-8002	1) Invalid data address passed for item <i>n</i> in <i>item_array</i> . 2) An address not accessible to the caller was in <i>item_array</i> . 3) Pass only addresses in accessible spaces in <i>item_array</i> .
-8003	1) Unable to access item <i>n</i> 's pointer in <i>item_array</i> . 2) The <i>item_array</i> had fewer items than <i>itemnum_array</i> . 3) Make <i>item_array</i> have as many items as <i>itemnum_array</i> .
-8004	1) Invalid <i>spooler_device</i> . 2) The device indicated by <i>spooler_device</i> is not spooled. The device indicated by <i>spooler_device</i> is not spoolable. The device indicated by <i>spooler_device</i> is not in the system configuration. The <i>spooler_device</i> is not a valid device name. 3) StartSpool on the device if the device is spoolable and not spooled.



### 4.1.3 AIF\_\_SPP\_\_PUT

This procedure allows write access to spooler process information. It accepts a device name or device number which identifies the spooler process for which information in the Spooler Process Information Table is to be updated.

#### 4.1.3.1 CALLING SEQUENCE .

```

Procedure aif__spp__put(
    anyvar      itemnum__array      : itemnum__array__type;
    anyvar      item__array          : item__array__type;
    anyvar      item__status__array  : item__status__array__type;
    var         overall__status      : integer;
    readonly    spooler__device      : device__name__type)

```

```

option extensible 5
    uncheckable__anyvar;

```

*itemnum\_\_array* (I32A) An array of integers where each element is the item number of an item to be modified. If there are  $n$  items being updated, element  $n+1$  must be a zero to indicate the end of the element list.

*item\_\_array* (@64A) An array of  $n$  long pointers where each element is a long pointer that points to an area of the type appropriate for the corresponding item number in the *itemnum\_\_array*. The items will be updated with information contained in these areas.

*item\_\_status\_\_array* (I32A) An array of  $n$  integers where each element will hold on output, the status for the corresponding item. A zero indicates no error, a negative value indicates an error condition and a positive value indicates a warning.

*overall\_\_status* (I32) This will hold the overall status of the call. A negative value indicates an error in the overall call, not specific to any particular item. A zero indicates no error. A positive number indicates an index into the *item\_\_status\_\_array*, indicating the first element signaling an error condition.

*spooler\_\_device* (*device\_\_name\_\_type*) The device name or logical device (ldev) number owned by the spooler process for which information is to be modified. An ldev number must be converted into ASCII character string before being passed to this routine. The device name should be left justified and blank padded.

## 4.1.3.2 ITEM DESCRIPTIONS .

## Spooler Process Put Items

Num	Name(Type) & Description
8007	<p><b>Banner Strategy(I32)</b></p> <p>This is the banner printing strategy of the device. The valid values are <b>AROUND</b> and <b>BETWEEN</b>. <b>AROUND</b> means print header before the first copy and trailer after the last copy. <b>BETWEEN</b> means print header and trailer for every copy.</p> <p style="padding-left: 40px;">0 = Between (default) 1 = Around</p>
8009	<p><b>Device Outfence(I32)</b></p> <p>The outfence of the device. The range of accepted values are 0 through 14.</p>

## 4.1.3.3 ERROR MESSAGES .

## Error Messages for AIF\_\_SPP\_\_PUT

AIFERR	1) Message 2) Cause 3) Action
-1	1) Caller not in privileged mode. 2) The process calling this interface was at HW ring level 3. 3) Call GETPRIVMODE to promote process to HW ring level 2.
-2	1) Could not lock the required resources. 2) The required resources are locked up by the system. 3) Call AIFLockGet to explicitly lock, or try again.
-8001	1) Invalid item number passed in <i>itemnum__array</i> . 2) A non-zero, invalid item number was passed in <i>itemnum__array</i> . 3) Pass an appropriate value and end with a zero.
-8002	1) Invalid data address passed for item <i>n</i> in <i>item__array</i> . 2) An address not accessible to the caller was in <i>item__array</i> . 3) Pass only addresses in accessible spaces in <i>item__array</i> .
-8003	1) Unable to access item <i>n</i> 's pointer in <i>item__array</i> . 2) The <i>item__array</i> had fewer items than <i>itemnum__array</i> . 3) Make <i>item__array</i> have as many items as <i>itemnum__array</i> .
-8004	1) Invalid <i>spooler__device</i> . 2) The device indicated by <i>spooler__device</i> is not spooled. The device indicated by <i>spooler__device</i> is not spoolable. The device indicated by <i>spooler__device</i> is not in the system configuration. The <i>spooler__device</i> is not a valid device name. 3) StartSpool on the device if the device is spoolable and not spooled.

## 4.1.4 AIF\_\_SPF\_\_GET

This procedure allows access to information about spoolfiles. It accepts a spoolfile ID or spoolfile UFID which identifies the spoolfile that the information is to be returned on. UFID is the faster mechanism of the two. If both are provided, then the UFID and the spoolfile ID will be checked against each other to make sure that they match the same file.

### 4.1.4.1 CALLING SEQUENCE .

```
Procedure aif__spf__get(
    anyvar      itemnum__array      : itemnum__array__type;
    anyvar      item__array         : item__array__type;
    anyvar      item__status__array : item__status__array__type;
    var         overall__status     : integer;
    var         spf__ufid           : ufid__type;
    var         spf__id             : spf__id__type)
```

```
option extensible 6
uncheckable__anyvar
default__parms (
    spf__ufid := nil
    spf__id   := nil);
```

- itemnum\_\_array* (I32A) An array of integers where each element is the item number of an item to be returned. If there are  $n$  item numbers being requested, element  $n+1$  must be a zero to indicate the end of the element list.
- item\_\_array* (@64A) An array of  $n$  long pointers where each element is a long pointer that points to an area of the type appropriate for the corresponding item number in the *itemnum\_\_array*. The information to be retrieved is placed in these areas.
- item\_\_status\_\_array* (I32A) An array of  $n$  integers where each element will hold on output, the status for the corresponding item. A zero indicates no error, a negative value indicates an error condition and a positive value indicates a warning.
- overall\_\_status* (I32) This will hold the overall status of the call. A negative value indicates an error in the overall call, not specific to any particular item. A zero indicates no error. A positive number indicates an index into the *item\_\_status\_\_array*, indicating the last element signaling an error condition.
- spf\_\_ufid* (ufid\_\_type) The UFID of the spoolfile for which information is desired.
- spf\_\_id* (spf\_\_id\_\_type) The spoolfile ID of the spoolfile for which information is desired. Since the UFID allows faster access to the spoolfile information, don't pass the *spf\_\_id* if the UFID is available.

## 4.1.4.2 ITEM DESCRIPTIONS .

## Spoolfile Get Items

Num	Name(Type) & Description																																																												
8501	<p><b>File State(I32)</b></p> <p>The state of the spoolfile. The valid states are:</p> <table border="0"> <thead> <tr> <th>FLAG</th> <th>STATE</th> <th>I/O</th> <th>SPOOLFILE</th> <th>MEANING</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>=</td> <td></td> <td></td> <td>Open</td> </tr> <tr> <td>1</td> <td>=</td> <td></td> <td></td> <td>Active</td> </tr> <tr> <td>2</td> <td>=</td> <td></td> <td></td> <td>Create</td> </tr> <tr> <td>3</td> <td>=</td> <td></td> <td></td> <td>Defer</td> </tr> <tr> <td>4</td> <td>=</td> <td></td> <td></td> <td>Ready</td> </tr> <tr> <td>5</td> <td>=</td> <td></td> <td></td> <td>Transfer</td> </tr> <tr> <td>6</td> <td>=</td> <td></td> <td></td> <td>Print</td> </tr> <tr> <td>7</td> <td>=</td> <td></td> <td></td> <td>Problem</td> </tr> <tr> <td>8</td> <td>=</td> <td></td> <td></td> <td>Del_Pending</td> </tr> <tr> <td>9</td> <td>=</td> <td></td> <td></td> <td>Spsave</td> </tr> <tr> <td>10</td> <td>=</td> <td></td> <td></td> <td>(Reserved)</td> </tr> </tbody> </table>	FLAG	STATE	I/O	SPOOLFILE	MEANING	0	=			Open	1	=			Active	2	=			Create	3	=			Defer	4	=			Ready	5	=			Transfer	6	=			Print	7	=			Problem	8	=			Del_Pending	9	=			Spsave	10	=			(Reserved)
FLAG	STATE	I/O	SPOOLFILE	MEANING																																																									
0	=			Open																																																									
1	=			Active																																																									
2	=			Create																																																									
3	=			Defer																																																									
4	=			Ready																																																									
5	=			Transfer																																																									
6	=			Print																																																									
7	=			Problem																																																									
8	=			Del_Pending																																																									
9	=			Spsave																																																									
10	=			(Reserved)																																																									
8502	<p><b>Priority(I32)</b></p> <p>Returns the output priority of the spoolfile. The range is 0 through 14.</p>																																																												
8503	<p><b>Restartable(I32)</b></p> <p>Returns whether the job is restartable. RESTART is a parameter of the :JOB command. This item applies only to \$STDIN of a job.</p> <p>0 = Not Restartable 1 = Restartable</p>																																																												
8504	<p><b>Disposition(I32)</b></p> <p>Returns whether the output spoolfile is to be saved or purged after it has been printed. The values are:</p> <p>1 = Save 2 = Purge (default)</p>																																																												

## Spoolfile Get Items (contd.)

Num	Name(Type) & Description
8505	<p><b>Private(I32)</b></p> <p>Returns whether the spoolfile is a private spoolfile. 0 for not private and 1 for private spoolfile. All input spoolfiles are created with the private option.</p>
8506	<p><b>Forms Message(I32)</b></p> <p>Returns whether the spoolfile has a forms message associated with it. 0 for no message and 1 for yes.</p>
8507	<p><b>Incomplete(I32)</b></p> <p>Returns whether the spoolfile is incomplete for some reason. 0 for complete and 1 for incomplete.</p>
8508	<p><b>Job/Data File(I32)</b></p> <p>Returns whether the spoolfile is a job or data file. The values are:</p> <p style="padding-left: 40px;">0 = Neither (for output spoolfile)  1 = Job File  2 = Data File</p>
8509	<p><b>Stdlist of Aborted Job(I32)</b></p> <p>Returns whether the spoolfile is the \$stdlist of an aborted job. 1 for yes and 0 for not \$stdlist or not \$stdlist of aborted job.</p>
8510	<p><b>Spoolfile ID(I32)</b></p> <p>Returns the spoolfile ID in the following format:</p> <p style="padding-left: 40px;">Bits(0:31) - The spoolfile ID number  in the form of 31-bit positive integer  Bits(31:1) - 0 for input and 1 for output spoolfile</p>
8511	<p><b>Copies(I32)</b></p> <p>Returns the total number of copies to be printed for the spoolfile.</p>

## Spoolfile Get Items (contd.)

Num	Name(Type) & Description
8512	<p><b>Ready Date(I32)</b></p> <p>Returns the calendar date when the spoolfile was created. The format returned in the 32-bit integer is the same as that returned by the CALENDAR intrinsic:</p> <p style="padding-left: 40px;">Bits(0:16) - Unused  Bits(16:7) - The year of the century  Bits(23:9) - The day of the year</p>
8513	<p><b>Ready Time(I32)</b></p> <p>Returns the time (hours/minutes/seconds/tenths of seconds) when the spoolfile was created. The format returned in the 32-bit integer is the same as that returned by the CLOCK intrinsic:</p> <p style="padding-left: 40px;">Bits(0:8) - The hour of the day  Bits(8:8) - The minute of the hour  Bits(16:8) - The seconds  Bits(24:8) - The tenths of seconds</p>
8514	<p><b>Number of Pages(I32)</b></p> <p>Returns the number of pages in the spoolfile. A positive number indicates the actual number of pages in the spoolfile. A negative number indicates that the spoolfile has never been printed before and the number is only an estimation.</p>
8515	<p><b>Page To Start At(I32)</b></p> <p>Returns the page to restart at if the printing of the spoolfile has been suspended.</p>
8516	<p><b>Owner User/Account(CA32)</b></p> <p>Returns the user and account name of the creator of the spoolfile. The first 16 bytes is the user name and the second 16 bytes is the account name. The names should be left justified and blank padded. Currently, only the first 8 bytes of each field is used.</p>

## Spoolfile Get Items (contd.)

Num	Name(Type) & Description
8517	<p><b>Job/Session #(Bit32)</b></p> <p>Returns the job/session number under which the spoolfile was created. A job/session number with a "" indicates that the job or session is not current to the system, but rather that of a spoolfile restored onto the current system. The format returned in the 32-bit integer is:</p> <p style="padding-left: 40px;">Bits(0:2) - S'/S/J/J'</p> <p style="padding-left: 40px;">Bits(2:30) - The job/session number</p> <p>The value of the first two bits indicates:</p> <p style="padding-left: 40px;">0 = S'</p> <p style="padding-left: 40px;">1 = S</p> <p style="padding-left: 40px;">2 = J</p> <p style="padding-left: 40px;">3 = J'</p>
8518	<p><b>Job Name(CA16)</b></p> <p>Returns the job name under which the spoolfile was created.</p>
8519	<p><b>File Designator(CA16)</b></p> <p>Returns the file designator of the spoolfile, i.e., the file name of the spoolfile.</p>
8520	<p><b>Target Device(device __name __type)</b></p> <p>Returns the destination logical device name or device class for the spoolfile.</p>
8521	<p><b>Device Record Size(Bit32)</b></p> <p>Returns the record size of the target device of the spoolfile.</p>
8522	<p><b>Device Type(Bit16)</b></p> <p>Returns the device type of the target device of the spoolfile.</p>
8523	<p><b>Device Subtype(Bit16)</b></p> <p>Returns the device subtype of the target device of the spoolfile.</p>
8524	<p><b>Completed Copy Count (I32)</b></p> <p>Returns the number of copy that has already been printed.</p>



## Spoolfile Get Items (contd.)

Num	Name(Type) & Description
8525	<b>Forms ID(CA16)</b> Returns the forms ID of the spoolfile.
8526	<b>Spoolfile UFID(ufid__type)</b> Returns the UFID of the spoolfile.
8527	<b>Active Device(device__name__type)</b> Returns the name of the last device that has picked up a copy of the spoolfile and is currently processing the spoolfile.
8528	<b>Number of Records(Bit32)</b> Returns the number of records in the spoolfile. However, the number is not known until the file leaves the CREATE state.
8529	<b>Number of Sectors(I32)</b> Returns the number of sectors in the spoolfile.
8530	<b>Environment File Name(CA36)</b> Returns the environment file name of the spoolfile. This is only valid if the system has not been rebooted.
8531	<b>Foptions(Bit16)</b> Returns the Foptions of the spoolfile. This is only valid if the system has not been rebooted.
8532	<b>Aoptions(Bit16)</b> Returns the Aoptions of the spoolfile. This is only valid if the system has not been rebooted.
8533	<b>File Open Flag(I32)</b> Returns whether the spoolfile is currently opened by HPFOPEN or FOPEN and whether the file is opened or exclusively opened. The values are:  0 = The file is not opened 1 = The file is opened exclusively 2 = The file is opened

## 4.1.4.3 ERROR MESSAGES .

## Error Messages for AIF\_SPF\_GET

AIFERR	1) Message 2) Cause 3) Action
-1	1) Caller not in privileged mode. 2) The process calling this interface was at HW ring level 3. 3) Call GETPRIVMODE to promote process to HW ring level 2.
-2	1) Could not lock the required resources. 2) The required resources are locked up by the system. 3) Call AIFLockGet to explicitly lock, or try again.
-4	1) AIFs not enabled. 2) The MPE XL AIF Product is not enabled. 3) Use AIFTurnOn with appropriate VendorId.
-8501	1) Invalid item number passed in <i>itemnum__array</i> . 2) A non-zero, invalid item number was passed in <i>itemnum__array</i> . 3) Pass an appropriate value and end with a zero.
-8502	1) Invalid data address passed for item <i>n</i> in <i>item__array</i> . 2) An address not accessible to the caller was in <i>item__array</i> . 3) Pass only addresses in accessible spaces in <i>item__array</i> .
-8503	1) Unable to access item <i>n</i> 's pointer in <i>item__array</i> . 2) The <i>item__array</i> had fewer items than <i>itemnum__array</i> . 3) Make <i>item__array</i> have as many items as <i>itemnum__array</i> .
-8505	1) Invalid <i>spf__ufid</i> . 2) The <i>spf__ufid</i> passed in does not exist on the system. 3) Call Get__SystemWide__Info for a list of valid UFID.
-8506	1) Invalid <i>spf__id</i> . 2) The <i>spf__id</i> passed in does not exist on the system. 3) Call Get__SystemWide__Info for a list of valid spoolfile IDs.
-8507	1) <i>spf__ufid</i> and <i>spf__id</i> do not match the same file. 2) The file indicated by <i>spf__ufid</i> is not the same as <i>spf__id</i> . 3) Call Get__SystemWide__Info for a list of valid UFIDs and spoolfile IDs.

## 4.1.5 AIF\_\_SPF\_\_PUT

This procedure allows write access to information about spoolfiles. It accepts a spoolfile ID or spoolfile UFID which identifies the spoolfile and updates the information in the spoolfile directory and file label extension. UFID is the faster mechanism of the two. If both the UFID and the spoolfile ID are provided, they will be checked against each other to make sure that they match the same file.

### 4.1.5.1 CALLING SEQUENCE .

Procedure `aif__spf__put`(

```

    anyvar    itemnum__array      : itemnum__array__type;
    anyvar    item__array         : item__array__type;
    anyvar    item__status__array : item__status__array__type;
    var       overall__status     : integer;
    var       spf__ufid           : ufid__type;
    var       spf__id            : spf__id__type)

```

option extensible 6

```

    uncheckable__anyvar
    default__parms (
        spf__ufid := nil,
        spf__id  := nil);

```

- itemnum\_\_array* (I32A) An array of integers where each element is the item number of an item to be modified. If there are  $n$  items being updated, element  $n+1$  must be a zero to indicate the end of the element list.
- item\_\_array* (@64A) An array of  $n$  long pointers where each element is a long pointer that points to an area of the type appropriate for the corresponding item number in the *itemnum\_\_array*. The items will be updated with information contained in these areas.
- item\_\_status\_\_array* (I32A) An array of  $n$  integers where each element will hold on output, the status for the corresponding item. A zero indicates no error, a negative value indicates an error condition and a positive value indicates a warning.
- overall\_\_status* (I32) This will hold the overall status of the call. A negative value indicates an error in the overall call, not specific to any particular item. A zero indicates no error. A positive number indicates an index into the *item\_\_status\_\_array*, indicating the last element signaling an error condition.
- spf\_\_ufid* (*ufid\_\_type*) The UFID of the spoolfile for which information is to be modified.
- spf\_\_id* (*spf\_\_id\_\_type*) The ID of the spoolfile for which information is to be modified. Since the UFID allows faster access to the spoolfile information, don't pass the *spf\_\_id* if the UFID is available.

## 4.1.5.2 ITEM DESCRIPTIONS .

## Spoolfile Put Items

Num	Name(Type) & Description																																																				
8501	<p><b>File State(I32)</b></p> <p>The state of the spoolfile. The valid states are:</p> <table border="0"> <thead> <tr> <th data-bbox="263 611 329 638">FLAG</th> <th data-bbox="391 611 474 638">STATE</th> <th data-bbox="505 611 714 638">I/O SPOOLFILE</th> <th data-bbox="763 611 879 638">MEANING</th> </tr> </thead> <tbody> <tr> <td colspan="4" data-bbox="263 646 1321 659">-----</td> </tr> <tr> <td data-bbox="294 674 314 701">0</td> <td data-bbox="329 674 437 701">= Open</td> <td data-bbox="582 674 602 701">I</td> <td data-bbox="652 674 1151 701">Job or data file being accessed</td> </tr> <tr> <td data-bbox="294 709 314 737">1</td> <td data-bbox="329 709 474 737">= Active</td> <td data-bbox="582 709 602 737">I</td> <td data-bbox="652 709 1295 737">Job/data file inputting from inspool dev</td> </tr> <tr> <td data-bbox="294 745 314 772">2</td> <td data-bbox="329 745 474 772">= Create</td> <td data-bbox="582 745 602 772">O</td> <td data-bbox="652 745 1264 772">spoolfile created and being written to</td> </tr> <tr> <td data-bbox="294 781 314 808">3</td> <td data-bbox="329 781 458 808">= Defer</td> <td data-bbox="582 781 602 808">O</td> <td data-bbox="652 781 1228 808">Defer option specified for spoolfile</td> </tr> <tr> <td data-bbox="294 816 314 844">4</td> <td data-bbox="329 816 458 844">= Ready</td> <td data-bbox="571 816 628 844">I/O</td> <td data-bbox="652 816 1228 844">File ready to be accessed or printed</td> </tr> <tr> <td data-bbox="294 852 314 879">5</td> <td data-bbox="329 852 509 879">= Transfer</td> <td data-bbox="582 852 602 879">O</td> <td data-bbox="652 852 1007 879">File being transferred</td> </tr> <tr> <td data-bbox="294 888 314 915">6</td> <td data-bbox="329 888 458 915">= Print</td> <td data-bbox="582 888 602 915">O</td> <td data-bbox="652 888 1264 915">Outspoolfile being printed on a device</td> </tr> <tr> <td data-bbox="294 924 314 951">7</td> <td data-bbox="329 924 489 951">= Problem</td> <td data-bbox="582 924 602 951">O</td> <td data-bbox="652 924 1310 951">Abnormality preventing file from printing</td> </tr> <tr> <td data-bbox="294 959 314 987">8</td> <td data-bbox="329 959 555 987">= Del_Pending</td> <td data-bbox="582 959 602 987">O</td> <td data-bbox="652 959 1167 987">File to be deleted after closing</td> </tr> <tr> <td data-bbox="294 995 314 1022">9</td> <td data-bbox="329 995 474 1022">= Spsave</td> <td data-bbox="582 995 602 1022">O</td> <td data-bbox="652 995 1279 1022">Copies printed, SPSAVE option specified</td> </tr> <tr> <td data-bbox="294 1031 329 1058">10</td> <td data-bbox="329 1031 540 1058">= (Reserved)</td> <td></td> <td></td> </tr> </tbody> </table>	FLAG	STATE	I/O SPOOLFILE	MEANING	-----				0	= Open	I	Job or data file being accessed	1	= Active	I	Job/data file inputting from inspool dev	2	= Create	O	spoolfile created and being written to	3	= Defer	O	Defer option specified for spoolfile	4	= Ready	I/O	File ready to be accessed or printed	5	= Transfer	O	File being transferred	6	= Print	O	Outspoolfile being printed on a device	7	= Problem	O	Abnormality preventing file from printing	8	= Del_Pending	O	File to be deleted after closing	9	= Spsave	O	Copies printed, SPSAVE option specified	10	= (Reserved)		
FLAG	STATE	I/O SPOOLFILE	MEANING																																																		
-----																																																					
0	= Open	I	Job or data file being accessed																																																		
1	= Active	I	Job/data file inputting from inspool dev																																																		
2	= Create	O	spoolfile created and being written to																																																		
3	= Defer	O	Defer option specified for spoolfile																																																		
4	= Ready	I/O	File ready to be accessed or printed																																																		
5	= Transfer	O	File being transferred																																																		
6	= Print	O	Outspoolfile being printed on a device																																																		
7	= Problem	O	Abnormality preventing file from printing																																																		
8	= Del_Pending	O	File to be deleted after closing																																																		
9	= Spsave	O	Copies printed, SPSAVE option specified																																																		
10	= (Reserved)																																																				
8502	<p><b>Priority(I32)</b></p> <p>Must be a value from 0 to 14. A value higher than the OUTFENCE of the device causes the spoolfile to be printed.</p>																																																				
8504	<p><b>Disposition(I32)</b></p> <p>Indicates whether the output spoolfile is to be saved or purged after it has been printed. The valid values are:</p> <table border="0"> <tbody> <tr> <td data-bbox="417 1402 437 1430">1</td> <td data-bbox="458 1402 555 1430">= Save</td> </tr> <tr> <td data-bbox="417 1438 437 1465">2</td> <td data-bbox="458 1438 740 1465">= Purge (default)</td> </tr> </tbody> </table>	1	= Save	2	= Purge (default)																																																
1	= Save																																																				
2	= Purge (default)																																																				
8505	<p><b>Private(I32)</b></p> <p>Must be of the value of either 0 or 1. 0 for not private and 1 for private spoolfile. All input spoolfiles are created with the private option.</p>																																																				
8507	<p><b>Incomplete(I32)</b></p> <p>Must be of the value of either 0 or 1. 1 means that the spoolfile is incomplete for some reason, 0 means that the spoolfile is complete.</p>																																																				

## Spoolfile Put Items (contd.)

Num	Name(Type) & Description
8511	<p><b>Copies(I32)</b></p> <p>Changes the total number of copies to be printed for the spoolfile. The file will be deleted if this count is lower than the number of copies already completed and the file is not marked as SPSAVE.</p>
8512	<p><b>Ready Date(I32)</b></p> <p>This is the calendar date that the spoolfile was created. This field cannot be modified if the file is in the CREATE state. The format of the integer returned is the same as that returned by the CALENDAR intrinsic:</p> <p style="padding-left: 40px;">Bits(0:16) - Unused Bits(16:7) - The year of the century Bits(23:9) - The day of the year</p>
8513	<p><b>Ready Time(I32)</b></p> <p>This is the time (hours/minutes/seconds/tenths of seconds) when the spoolfile was created. This field cannot be modified if the file is in the CREATE state. The format of the integer returned is the the same as that returned by the CLOCK intrinsic:</p> <p style="padding-left: 40px;">Bits(0:8) - The hour of the day Bits(8:8) - The minute of the hour Bits(16:8) - The seconds Bits(24:8) - The tenths of seconds</p>
8514	<p><b>Number of Pages(I32)</b></p> <p>Modifies the number of pages in the spoolfile. A positive number indicates the actual number of pages in the spoolfile. A negative number indicates that the spoolfile has never been printed before and the number is only an estimation.</p>
8515	<p><b>Page To Start At(I32)</b></p> <p>Changes the page to restart at if the printing of the spoolfile has been suspended.</p>
8516	<p><b>Owner User/Account(CA32)</b></p> <p>Modifies the user and account name of the creator of the spoolfile. The first 16 bytes is the user name and the second 16 bytes is the account name. The names should be left justified and blank padded. Currently, only the first 8 bytes of each field is used.</p>

## Spoolfile Put Items (contd.)

Num	Name(Type) & Description
8517	<p><b>Job/Session #(Bit32)</b></p> <p>Changes the job/session number under which the spoolfile was created. A job/session number with a "" indicates that the job or session is not current to the system, but rather that of a spoolfile restored onto the current system. The 32-bit input should be in the following format</p> <p style="padding-left: 40px;">Bits(0:2) - S'/S/J/J'</p> <p style="padding-left: 40px;">Bits(2:30) - The job/session number</p> <p>The value of the first two bits indicates:</p> <p style="padding-left: 40px;">0 = S'</p> <p style="padding-left: 40px;">1 = S</p> <p style="padding-left: 40px;">2 = J</p> <p style="padding-left: 40px;">3 = J'</p>
8518	<p><b>Job Name(CA16)</b></p> <p>Changes the job name under which the spoolfile was created.</p>
8519	<p><b>File Designator(CA16)</b></p> <p>Changes the file designator of the spoolfile, i.e., the file name of the spoolfile.</p>
8520	<p><b>Target Device(device__name__type)</b></p> <p>Changes the destination logical device for the spoolfile. It can be a device name, a device number or a device class. A device number must be in the format of ASCII character string. A 'put' operation on the <b>target device</b> would actually move the spoolfile entry from the old target device queue to the newly targeted device queue.</p>
8524	<p><b>Completed Copy Count(I32)</b></p> <p>Modifies the number of copy that has been printed already.</p>
8525	<p><b>Forms ID(CA16)</b></p> <p>Changes the forms ID of the spoolfile.</p>

4.1.5.3 ERROR MESSAGES .

Error Messages for AIF\_SPF\_PUT

AIFERR	1) Message 2) Cause 3) Action
-1	1) Caller not in privileged mode. 2) The process calling this interface was at HW ring level 3. 3) Call GETPRIVMODE to promote process to HW ring level 2.
-2	1) Could not lock the required resources. 2) The required resources are locked up by the system. 3) Call AIFLockGet to explicitly lock, or try again.
-4	1) AIFs not enabled. 2) The MPE XL AIF Product is not enabled. 3) Use AIFTurnOn with appropriate VendorId.
-8501	1) Invalid item number passed in <i>itemnum_array</i> . 2) A non-zero, invalid item number was passed in <i>itemnum_array</i> . 3) Pass an appropriate value and end with a zero.
-8502	1) Invalid data address passed for item <i>n</i> in <i>item_array</i> . 2) An address not accessible to the caller was in <i>item_array</i> . 3) Pass only addresses in accessible spaces in <i>item_array</i> .
-8503	1) Unable to access item <i>n</i> 's pointer in <i>item_array</i> . 2) The <i>item_array</i> had fewer items than <i>itemnum_array</i> . 3) Make <i>item_array</i> have as many items as <i>itemnum_array</i> .
-8505	1) Invalid <i>spf_ufid</i> . 2) The <i>spf_ufid</i> passed in does not exist on the system. 3) Call Get_SystemWide_Info for a list of valid UFID.
-8506	1) Invalid <i>spf_id</i> . 2) The <i>spf_id</i> passed in does not exist on the system. 3) Call Get_SystemWide_Info for a list of valid spoolfile IDs.
-8507	1) <i>spf_ufid</i> and <i>spf_id</i> do not match the same file. 2) The file indicated by <i>spf_ufid</i> is not the same as <i>spf_id</i> . 3) Call Get_SystemWide_Info for a list of valid UFIDs and spoolfile IDs.

## 4.2 FUNCTIONALITY ACCESS

There are two groups of NCS functionality access routines, one for managing spoolfiles and the other for managing spooling processes. For each routine, the calling sequence and data type of the parameter are described in detail.



## 4.2.1 AIF \_\_SPP\_\_OPENQ

This procedure opens a spool queue for the specified logical device, device name, or device class. It is the programmatic interface for executing the MPE command OPENQ.

### 4.2.1.1 CALLING SEQUENCE .

Procedure aif\_\_spp\_\_openq(  
    *readonly*      *device*                                   : *device\_\_name\_\_type*;  
    *var*            *status*                                 : *integer*)

option extensible 2;

*device*  
(*device\_\_name\_\_type*)      The device name, number or device class for which a spool queue is to be opened. A device number must be converted into ASCII character string before being passed to this routine. The device name should be left justified and blank padded.

*status (I32)*                This parameter, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.

## 4.2.2 AIF \_\_SPP \_\_SHUTQ

This procedure closes the spool queue for the specified logical device, device name, or device class. It is the programmatic interface for executing the MPE command SHUTQ.

### 4.2.2.1 CALLING SEQUENCE .

```
Procedure aif__spp__shutq(
    readonly    device          : device__name__type;
    var         status         : integer)
```

option extensible 2;

*device*  
(*device\_\_name\_\_type*)      The device name, number or device class for which a spool queue is to be closed. A device number must be converted into ASCII character string before being passed to this routine. The device name should be left justified and blank padded.

*status (I32)*              This parameter, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.

### 4.2.3 AIF \_\_SPP\_\_ START

This procedure creates and activates a new spooler process to handle spoolfiles destined for the specified logical device, device name, or device class. An OPENQ is done by default when this procedure is invoked, unless the SHUTQ parameter is specified. This is the programmatic interface for executing the MPE commands STARTSPOOL and SPOOLER dev;START.

#### 4.2.3.1 CALLING SEQUENCE .

```
Procedure aif__spp__start(
    readonly    device           : device__name__type;
    var         status          : integer;
               q__state        : integer)
```

```
option extensible 3
    default__parms (q__state := 1);
```

*device*  
(*device\_\_name\_\_type*)      The device name, number or device class for which the spooling process is to be initiated. A device number must be converted into ASCII character string before being passed to this routine.

*status (I32)*              This parameter, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.

*q\_\_state (I32)*            The *q\_\_state* parameter is used to indicate whether the spooling queue is to be enabled or disabled when the spooler process is initiated. The default is OPENQ for starting a spooler process. If SHUTQ is specified, it prevents users from generating spoolfiles on that device. It does not prevent the user from printing previously generated spoolfiles. The valid inputs are:

```
0 = no change to the current q__state of the spooling process
1 = openq (default)
2 = shutq
```

## 4.2.4 AIF\_\_SPP\_\_STOP

This procedure terminates spooling to the specified logical device, device name, or device class. The spooling processes associated with the devices are also terminated. A SHUTQ is done by default when this procedure is invoked, unless the OPENQ parameter is specified. This is the programmatic interface for executing the MPE commands STOPSPPOOL and SPOOLER dev;STOP.

### 4.2.4.1 CALLING SEQUENCE .

```
Procedure aif__spp__stop(
    readonly    device          : device__name__type;
    var         status          : integer;
              finish          : integer;
              q__state        : integer)
```

```
option extensible 4
    default__parms (
        finish := 1,
        q__state := 0 );
```

*device*  
(*device\_\_name\_\_type*)      The device name, number or device class for which the spooling process is to be terminated. A device number must be converted into ASCII character string before being passed to this routine.

*status* (I32)              This parameter, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.

*finish* (I32)             This is the finishing strategy for stopping the spooling process. The valid inputs are:

```
1 = now          (default)
2 = end of copy
```

*q\_\_state* (I32)            The *q\_\_state* parameter indicates whether the spooling queue is to remain open or disabled when the spooling process terminates. The default is SHUTQ for terminating a spooler process. If OPENQ is specified, it allows users to generate spoolfiles on that device even when the spooling process has been terminated. The valid inputs are:

```
0 = no change to the current q__state of the spooling process
1 = openq
2 = shutq (default)
```

## 4.2.5 AIF \_\_SPP\_\_SUSPEND

This procedure suspends the spooling processes for the specified logical device, device name or device class. The associated spooler processes remain alive, but inactive. It is valid to call this procedure only when the spooler is in the ACTIVE or IDLE state, or to accelerate a previous SUSPEND;FINISH to a SUSPEND;NOW. The default options are KEEP, FINISH NOW and RELATIVE OFFSET 0. This is the programmatic interface for executing the MPE commands SUSPENDSPOOL and SPOOLER dev;SUSPEND.

### 4.2.5.1 CALLING SEQUENCE .

#### Procedure aif\_\_spp\_\_suspend(

```

    readonly    device           : device__name__type;
    var         status           : integer;
              finish           : integer;
              keep             : integer;
              direction        : integer;
              offset          : integer;
              q__state        : integer)

```

option extensible 7

```

    default__parms (
        finish := 1,
        keep   := 1,
        direction := 0,
        offset := 0,
        q__state := 0);

```

*device*  
(*device\_\_name\_\_type*)      The device name, number or device class for which the spooling process is to be suspended. A device number must be converted into ASCII character string before being passed to this routine.

*status (I32)*              This parameter, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.

*finish (I32)*              This is the finishing strategy for suspending the spooling process. The valid inputs are:

```

    1 = now          (default)
    2 = end of copy

```

*keep (I32)*                This parameter tells the spooler whether to retain ownership of the currently printing spoolfile or to close the file and return it to the READY state. The valid inputs are:

```

    1 = keep        (default)
    2 = nokeep

```

*direction* (I32)

This parameter tells the spooler how to apply the *offset* parameter.

- 0 = relative offset specified in the *offset* parameter (default)
- 1 = absolute offset specified in the *offset* parameter

*offset* (I32)

*Offset* is an integer representing a page offset, either absolute or relative, within the spoolfile. Together with the *direction* parameter, it tells the spooler where to resume when the file is picked up again for printing. If 'absolute' is specified in *direction*, printing resumes at that page, absolute from the beginning of the file. If 'relative' is specified in *direction*, then depending on whether *offset* is positive or negative, the offset is adjusted either forward or backward relative to the current location, by the number of pages specified. No matter which combination of offsets is specified, the final location is limited by the bounds of the file. The default for offset is 0. If the printing of a spoolfile is to be resumed from the beginning of the file, pass absolute for *direction* and 0 for *offset*.

*q\_\_state* (I32)

The *q\_\_state* parameter indicates whether the spooling queue is to be opened or disabled when the spooling process is suspended. The default is not to change the current *q\_\_state* of the spooler process. The valid inputs are:

- 0 = no change to the current *q\_\_state* of the spooling process (default)
- 1 = openq
- 2 = shutq

## 4.2.6 AIF \_\_SPP\_\_RESUME

This procedure resumes the suspended spooling process for the specified logical device, device name or device class. The spooler must be in the SUSPEND state. If the spooler retained a spoolfile when it was SUSPENDED and the spoolfile was not subsequently RELEASED, then the *offset* parameter can be specified. If it is not specified, then output will resume where it was left off. It is the programmatic interface for executing the MPE commands RESUMESPOOL and SPOOLER dev;RESUME.

### 4.2.6.1 CALLING SEQUENCE .

```
Procedure aif__spp__resume(
    readonly    device          : device__name__type;
    var         status          : integer;
               direction       : integer;
               offset          : integer;
               q__state        : integer)
```

```
option extensible 5
default__parms (
    direction := 0,
    offset   := 0,
    q__state := 0 );
```

*device*  
(*device\_\_name\_\_type*)      The device name, number or device class for which the spooling process is to be resumed. A device number must be converted into ASCII character string before being passed to this routine.

*status* (I32)              This parameter, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.

*direction* (I32)          This parameter tells the spooler how to apply the *offset* parameter.

0 = relative offset specified in the *offset* parameter (default)  
1 = absolute offset specified in the *offset* parameter

*offset* (I32)              *Offset* is an integer representing a page offset, either absolute or relative, within the spoolfile. Together with the *direction* parameter, it tells the spooler where to resume when the file is picked up again for printing. If 'absolute' is specified in *direction*, printing resumes at that page, absolute from the beginning of the file. If 'relative' is specified in *direction*, then depending on whether *offset* is positive or negative, the offset is adjusted either forward or backward relative to the current location, by the number of pages specified. No matter which combination of offsets is specified, the final location is limited by the bounds of the file. The default for offset is 0. If the printing of a spoolfile is to be resumed from the beginning of the file, pass absolute for *direction* and 0 for *offset*.

*q\_\_state* (I32)            The *q\_\_state* parameter indicates whether the spooling queue is to be opened

or disabled when the spooling process is resumed. The default is not to change the current q\_\_state of the spooler process. The valid inputs are:

- 0 = no change to the current q\_\_state of the spooling process (default)
- 1 = openq
- 2 = shutq



## 4.2.7 AIF\_\_SPP\_\_RELEASE

This procedure releases the spoolfile that is currently 'kept' by the specified suspended spooler. It closes the spoolfile and return it to the READY state. An *offset* may be specified to change the resumption point of the spoolfile the next time it is selected for printing. It is the programmatic interface for executing the MPE command SPOOLER dev;RELEASE.

### 4.2.7.1 CALLING SEQUENCE .

```
Procedure aif__spp__release(
    readonly    device           : device__name__type;
    var         status          : integer;
              direction        : integer;
              offset           : integer;
              q__state         : integer)
```

```
option extensible 5
    default__parms (
        direction := 0,
        offset    := 0
        q__state  := 0 );
```

*device*  
(*device\_\_name\_\_type*)      The device which is currently 'keeping' a spoolfile due to an earlier SUSPEND with the KEEP option. A device number must be converted into ASCII character string before being passed to this routine.

*status* (I32)              This parameter, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.

*direction* (I32)          This parameter tells the spooler how to apply the *offset* parameter.

0 = relative offset specified in the *offset* parameter (default)  
1 = absolute offset specified in the *offset* parameter

*offset* (I32)              *Offset* is an integer representing a page offset, either absolute or relative, within the spoolfile. Together with the *direction* parameter, it tells the spooler where to resume when the file is picked up again for printing. If 'absolute' is specified in *direction*, printing resumes at that page, absolute from the beginning of the file. If 'relative' is specified in *direction*, then depending on whether *offset* is positive or negative, the offset is adjusted either forward or backward relative to the current location, by the number of pages specified. No matter which combination of offsets is specified, the final location is limited by the bounds of the file. The default for offset is 0. If the printing of a spoolfile is to be resumed from the beginning of the file, pass absolute for *direction* and 0 for *offset*.

*q\_\_state* (I32)            The *q\_\_state* parameter indicates whether the spooling queue is to be opened or disabled when the spooling process is resumed. The default is not to

change the current q\_\_state of the spooler process. The valid inputs are:

0 = no change to the current q\_\_state of the spooling process (default)

1 = openq

2 = shutq



## 4.2.8 AIF\_\_SPF\_\_LINK

Spoolfiles with NCS are permanent disk files. They are created implicitly by sending data to a spooled output device. Explicitly, spoolfiles can be created by the HPFOPEN intrinsic or by the MPE BUILD command. A spoolfile created by HPFOPEN with the 'link option' and the 'linked device option' resides in the HPSPOOL account. It has an entry in the separately maintained spoolfile directory and is linked into the spoolfile queue. This is called a 'linked' spoolfile and it is known to the spooling processes. A spoolfile created by the MPE BUILD command with the ;SPOOL option or by the HPFOPEN intrinsic without the link option can reside in any user directory. It does not have an entry in the spoolfile directory and is not linked into the spoolfile queue. A spoolfile created in such a way is described as 'unlinked'. To clarify this further, "A linked spoolfile must reside in the HPSPOOL account, but a spoolfile that resides in the HPSPOOL account is not necessarily linked".

To print an unlinked spoolfile, the spoolfile must first be copied to the HPSPOOL account and linked into the spoolfile directory. This procedure provides both the copying and the linking as described.

One other application of this procedure is that a user may save a copy of a spoolfile from the HPSPOOL account in his own directory. This can be achieved by the MPE command COPY. The user copy of the spoolfile is not linked. Later on, when this spoolfile is to be printed, the user can call AIF\_\_SPF\_\_LINK to copy and link the spoolfile to the HPSPOOL account. However, the spoolfile queue must be open for the device before a copy of the spoolfile can be created in the HPSPOOL account.

Certain attributes of the spoolfile can be altered while linking a spoolfile by calling this routine. If the target device information exists in the file label extension, then that device will be used as the default. The *device* parameter may be specified to override the existing target device. If there is no target device in the file label extension, the *device* must be specified when calling AIF\_\_SPF\_\_LINK or an error results. Other attributes that can be changed for the spoolfiles are *priority*, *copies*, *spsave* and *defer*.

Either a UFID or a spoolfile name can be specified for calling this procedure. However, the UFID is the faster mechanism of the two. If both are provided, the UFID and the spoolfile name will be checked against each other to make sure that they match the same file. This is the programmatic interface for executing the MPE command SPOOLF spoolid;PRINT.

### 4.2.8.1 CALLING SEQUENCE .

```

Procedure aif__spf__link(
    readonly    source__spf           : filename__type;
    var         target__spf__id       : spf__id__type;
    var         target__spf__ufid     : ufid__type;
    var         status                : integer;
    target__device : device__name__type;
    priority    : integer;
    copies      : integer;
    spsave      : integer;
    defer       : integer)

option extensible 9
    default__parms (
        device := "",
        priority := 8,

```

```
copies := 1,
spsave := 0,
defer := 0);
```

<i>source_spf</i> ( <i>filename__type</i> )	The name of the spoolfile to be copied and linked.
<i>target_spf_id</i> ( <i>spf_id__type</i> )	The spoolfile id of the spoolfile created and linked to the HPSPPOOL account.
<i>target_spf_ufiled</i> ( <i>ufid__type</i> )	The UFID of the spoolfile created and linked to the HPSPPOOL account.
<i>status (I32)</i>	This parameter, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.
<i>target_device</i> ( <i>device__name__type</i> )	This parameter, if specified, will be used as the target device for printing the spoolfile. Whether this parameter is specified or not, the spoolfile queue for the device must be open or an error results. The device name should be left justified and blank padded.
<i>priority (I32)</i>	This parameter is used for setting the output priority of the newly created spoolfile in the HPSPPOOL account. The default priority is 8. The valid range is 0 to 13.
<i>copies (I32)</i>	This parameter is used to set the number of copies to be printed for the newly created spoolfile in the HPSPPOOL account. The default is 1. The valid range is 1 to 65535.
<i>spsave (I32)</i>	This parameter is used to set the SPSAVE flag for the newly created spoolfile in the HPSPPOOL account. SPSAVE flag directs the spooler to save the spoolfile in the HPSPPOOL account after it has been printed. The default is not to save the spoolfile. The valid inputs are:  0 = no spsave 1 = spsave
<i>defer (I32)</i>	This parameter is used to set the file state of newly created spoolfile. If defer is specified, the spoolfile will not be printed. The default is not to defer the printing of the spoolfile. The valid inputs are:  0 = no defer 1 = defer

## 4.2.9 AIF\_\_SPF\_\_LIST

This procedure returns the UFID and the ID of the spoolfiles that satisfy the selection criteria specified in the *sel\_eq* parameter.

The spoolfile queues are organized by device class and device name. Within each queue, the spoolfile is sorted by priority and ready date/time. This procedure scans the spoolfile queue and finds all spoolfiles satisfying the specified selection criteria. The spoolfiles returned in the arrays will be in the same sequence as listed by the LISTSPF command, i.e., sorted by device, priority and ready date/time.

Spoolfile UFIDs are returned in the *spf\_ufid\_array* and spoolfile IDs are returned in the *spf\_id\_array*. Either or both arrays can be defaulted to *nil*, in this case, no spoolfile ID or UFID will be returned. The number of spoolfiles qualifying the selection criteria is returned in the *spf\_count* parameter.

The list of spoolfile attributes that can be selected on are:

- device name
- file designator
- spoolfile ID
- number of pages in the spoolfile
- form id
- spoolfile state
- job name
- disposition (SPSAVE or PURGE)
- number of copies
- priority
- job number
- number of records in the spoolfile
- owner of the spoolfile
- stdlist of aborted job
  - spoolfile ready date

### 4.2.9.1 CALLING SEQUENCE .

```
Procedure aif__spf__list(
    var          sel_eq          : sel_eq__type;
    anyvar       spf_ufid_array  : ufid_array__type;
    anyvar       spf_id_array    : spf_id_array__type;
    var          spf_count       : integer;
    var          status          : integer)

option extensible 6
    uncheckable__anyvar
    default__parms (
        sel_eq      := nil,
        spf_ufid_array := nil,
        spf_id_array := nil);
```

*sel\_eq (sel\_eq\_type)* This parameter holds the selection criteria to be used as a filter for selecting the spoolfile to be returned by this routine.

It takes the form of Pascal string. The maximum length is 277 characters, not including the 'SELEQ' keyword. The actual parameter should be delimited by the beginning and ending square brackets, i.e. [ and ]. Blanks count as characters. For example, to list all spoolfiles owned by the user JON.DOE and are in the READY state, pass [(OWNER = JON.DOE) AND (STATE = READY)] in the *sel\_eq* parameter. The format of the selection equation is described in appendix B of this document or in the NCS ES under the LISTSPF and the SPOOLF commands.

If defaulted to *nil*, all spoolfiles on the system are returned.

*spf\_ufid\_array (ufid\_array\_type)* This variable, on procedure return, will hold the UFID of the spoolfile qualifying the selection criteria. *UFID\_array\_type* is an array whose element is of type UFID and each UFID takes 20 bytes of storage.

*spoolfile\_id\_array (spf\_id\_array\_type)* This variable, on procedure return, will hold the file ID of the spoolfile selected. *Spf\_id\_array\_type* is an array whose element is of type *Bit32*. Bit 0 through 30 is a 31-bit positive integer holding the spoolfile ID. Bit 31 is a flag, 0 indicates an input spoolfile and 1 indicates an output spoolfile.

*spf\_count (I32)* This variable, on input, holds the number of elements in the UFID and spoolfile ID array. On output, holds the number of the spoolfiles returned in the arrays. If the arrays are too small to hold all the qualifying spoolfiles, *spf\_count* will return the total number of qualifying spoolfiles instead of the number of entries returned.

*status (I32)* This variable, on procedure return, will hold the status of the call. A negative value indicates an error. A zero indicates no error. A positive number indicates a warning.



# LIST OF AIF ERRORS

## A.1 ERROR MESSAGES

### Error Messages for NCS AIFs

AIFERR	1) Message    2) Cause    3) Action
-1	1) Caller not in privileged mode. 2) The process calling this interface was at HW ring level 3. 3) Call GETPRIVMODE to promote process to HW ring level 2.
-3	1) Unable to access item <i>n</i> 's status in <i>item__st__array</i> . 2) The <i>item__st__array</i> had fewer items than <i>itemnum__array</i> . 3) Make <i>item__st__array</i> have as many items as <i>itemnum__array</i>
-4	1) AIFs not enabled. 2) The MPE XL AIF Product is not enabled. 3) Use AIFACCESSOn with appropriate VendorId.
-18	1) Invalid <i>aif__area key</i> . 2) The <i>aif__area</i> indicated does not exist. 3) See AIFSysWideGet parameter definition of <i>aif__area</i> .
-8001	1) Invalid item number passed in <i>itemnum__array</i> . 2) A non-zero, invalid item number was passed in <i>itemnum__array</i> . 3) Pass an appropriate value and end with a zero.
-8002	1) Invalid data address passed for item <i>n</i> in <i>item__array</i> . 2) An address not accessible to the caller was in <i>item__array</i> . 3) Pass only addresses in accessible spaces in <i>item__array</i> .
-8003	1) Unable to access item <i>n</i> 's pointer in <i>item__array</i> . 2) The <i>item__array</i> had fewer items then <i>itemnum__array</i> . 3) Make <i>item__array</i> have as many items as <i>itemnum__array</i> .
-8004	1) Invalid <i>spooler__device</i> . 2) The device indicated by <i>spooler__device</i> is not spooled. The device indicated by <i>spooler__device</i> is not spoolable. The device indicated by <i>spooler__device</i> is not in the system configuration. The <i>spooler__device</i> is not a valid device name. 3) StartSpool on the device if the device is spoolable and not spooled.



## Error Messages for NCS AIFs

AIFERR	1) Message    2) Cause    3) Action
-8501	1) Invalid item number passed in <i>itemnum__array</i> . 2) A non-zero, invalid item number was passed in <i>itemnum__array</i> . 3) Pass an appropriate value and end with a zero.
-8502	1) Invalid data address passed for item <i>n</i> in <i>item__array</i> . 2) An address not accessible to the caller was in <i>item__array</i> . 3) Pass only addresses in accessible spaces in <i>item__array</i> .
-8503	1) Unable to access item <i>n</i> 's pointer in <i>item__array</i> . 2) The <i>item__array</i> had fewer items than <i>itemnum__array</i> . 3) Make <i>item__array</i> have as many items as <i>itemnum__array</i> .
-8505	1) Invalid <i>Spoolfile__UFID</i> . 2) The <i>Spoolfile__UFID</i> passed in does not exist on the system. 3) Call <i>Get__SystemWide__Info</i> for a list of valid UFID.
-8506	1) Invalid <i>Spoolfile__name</i> . 2) The <i>Spoolfile__name</i> passed in does not exist on the system. 3) Call <i>Get__SystemWide__Info</i> for a list of valid filenames.
-8507	1) <i>UFID</i> and <i>Filename</i> do not match the same file. 2) The file indicated by <i>UFID</i> is not the same as <i>Filename</i> . 3) Call <i>Get__SystemWide__Info</i> for a list of valid UFIDs and filenames.

## Table of Contents

<b>Section 1</b>	
<b>PRODUCT IDENTIFICATION</b>	
1.1 Product Name . . . . .	1-1
1.2 Product Mnemonic . . . . .	1-1
1.3 Product Abstract . . . . .	1-1
<b>Section 2</b>	
<b>PRODUCT SPECIFICATIONS</b>	
2.1 Overview . . . . .	2-1
2.2 Definition . . . . .	2-1
2.2.1 Information Access . . . . .	2-1
2.2.2 Functionality Access . . . . .	2-2
2.3 Product Environment . . . . .	2-2
2.3.1 Hardware Requirements and Restrictions . . . . .	2-2
2.3.2 Software Requirements and Restrictions . . . . .	2-2
<b>Section 3</b>	
<b>PRODUCT ORGANIZATION</b>	
3.1 Information Access AIFs . . . . .	3-1
3.2 Functionality Access AIFs . . . . .	3-1
3.3 System Wide Information Access . . . . .	3-1
3.4 Data Types . . . . .	3-2
3.5 System Logging . . . . .	3-3
3.6 Error Management . . . . .	3-3
3.7 Glossary of Terms . . . . .	3-4
<b>Section 4</b>	
<b>DETAILED SPECIFICATIONS</b>	
4.1 Information Access . . . . .	4-1
4.1.1 System Wide Information . . . . .	4-2
4.1.1.1 Calling Sequence . . . . .	4-2
4.1.1.1 Criteria Descriptions . . . . .	4-4
4.1.1.2 Error Messages . . . . .	4-7
4.1.2 AIF_SPP_GET . . . . .	4-8
4.1.2.1 Calling Sequence . . . . .	4-8
4.1.2.1 Item Descriptions . . . . .	4-9
4.1.2.2 Error Messages . . . . .	4-11
4.1.3 AIF_SPP_PUT . . . . .	4-12
4.1.3.1 Calling Sequence . . . . .	4-12
4.1.3.1 Item Descriptions . . . . .	4-13
4.1.3.2 Error Messages . . . . .	4-14
4.1.4 AIF_SPF_GET . . . . .	4-15
4.1.4.1 Calling Sequence . . . . .	4-15
4.1.4.1 Item Descriptions . . . . .	4-16
4.1.4.2 Error Messages . . . . .	4-21

## Table of Contents

4.1.5 AIF__SPF__PUT . . . . .	4-22
4.1.5.1 Calling Sequence . . . . .	4-22
4.1.5.1 Item Descriptions . . . . .	4-23
4.1.5.2 Error Messages . . . . .	4-26
4.2 Functionality Access . . . . .	4-27
4.2.1 AIF__SPP__OPENQ . . . . .	4-28
4.2.1.1 Calling Sequence . . . . .	4-28
4.2.2 AIF__SPP__SHUTQ . . . . .	4-29
4.2.2.1 Calling Sequence . . . . .	4-29
4.2.3 AIF__SPP__START . . . . .	4-30
4.2.3.1 Calling Sequence . . . . .	4-30
4.2.4 AIF__SPP__STOP . . . . .	4-31
4.2.4.1 Calling Sequence . . . . .	4-31
4.2.5 AIF__SPP__SUSPEND . . . . .	4-32
4.2.5.1 Calling Sequence . . . . .	4-32
4.2.6 AIF__SPP__RESUME . . . . .	4-34
4.2.6.1 Calling Sequence . . . . .	4-34
4.2.7 AIF__SPP__RELEASE . . . . .	4-36
4.2.7.1 Calling Sequence . . . . .	4-36
4.2.8 AIF__SPF__LINK . . . . .	4-38
4.2.8.1 Calling Sequence . . . . .	4-38
4.2.9 AIF__SPF__LIST . . . . .	4-40
4.2.9.1 Calling Sequence . . . . .	4-40

## Appendix A

### LIST OF AIF ERRORS

A.1 Error Messages . . . . .	A-1
------------------------------	-----