TBS Software Inc.

OfficePath

Version 1 Release 6.1

Reference and Diagnosis Guide

OfficePath Reference and Diagnosis

Copyright

Trademarks

Edition Information

September 2021

This edition applies to Version 1 Release 6.1 of OfficePath. It also applies to subsequent releases until otherwise indicated in new editions.
Inquiries may be directed to:

TBS Software Inc.
2201-278 Bloor St E.
Toronto, ON
Canada
M4W 3M4

Phone: 905 940-9373
Fax: 905 940-9376
Web: www.tbssoft.com
eMail: support@tbssoft.com

Page Break

Table of Contents

List of Illustrations

Chapter 1. Introduction

Purpose of this book

This book contains information about the transactions used by OfficePath and the error messages issued by the main transactions. The book is intended to give the application/system programmer enough information to determine the reason for an OfficePath problem and the action necessary to recover from it.

Audience for this book

This book is designed for use by the systems programmer responsible for the installation of CICS-based application software and/or the application programmer writing an interface to OfficePath. It assumes a prior knowledge of CICS application programming, and DISOSS DIA/DCA Architecture and Protocol.

How this book is organized

This book is divided into three parts:

Diagnosing problems	This part provides general information on the resolution of problems.
Internal Logic	This part describes the main background transactions and files used by OfficePath.
Error Messages	This part lists the error messages produced by the OfficePath programs.

Other books you may need

There are four volumes that accompany the OfficePath system software:

- *OfficePath Installation and Administration* manual, which explains how to install, customize and administer the OfficePath system.
- *OfficePath Reference and Diagnosis* manual, which describes the detailed working of OfficePath, and also gives a list of the most common problems encountered.

- *OfficePath Programming Interfaces* manual, which provides information on how to develop and install an interface between an OfficePath system and an application based in CICS, MVS/BATCH or TSO.
  This manual is only supplied if the CICS or Batch features have been purchased.
- *OfficePath Library Retrieval Feature Installation and User Guide*, which provides information on how to install and use the Library Retrieval feature, providing on-line search and retrieval of documents from an existing DISOSS library.
  This manual is only supplied if the Library Retrieval feature has been purchased.

In addition, the following IBM manuals may be required:

- *DisplayWrite/370 Printing Guide* (SH12-5659), which describes how to configure DisplayWrite/370 in order to print documents.
- *DisplayWrite/370 Version 2 Programming Interfaces for customers* (SH12-5660), which describes the commarea used to interface to DisplayWrite/370.
- *DISOSS Interchange Architecture Reference Guide* (SH30-3276), which describes the data formats used for by the SNADS interface and also for communicating with Personal Services.
- *Installing OV/MVS Release 2.0* (SH21-0515), which describes how to customize Personal Services.

Chapter 2. Product Overview

What is OfficePath?

OfficePath is a CICS based node in the Office Systems Network. It provides "store and forward" Document Distribution Service facilities for routing from a **source** to one or more **destinations**.

The **source** can be one of the following:

Personal Services This can be either PS/CICS 1.3 or the Personal Services component of OfficeVision/MVS.

An API application An API application which uses the same subset of DISOSS commands as Personal Services[1].

A SNADS node Any other SNADS node. This may be another OfficePath system.

A CICS Application The Mailsack VSAM KSDS is used as an in-tray between the CICS application and OfficePath.

A Batch Application The Mailbox VSAM KSDS is used as an in-tray between the batch application and OfficePath.

TSO A TSO User can use the supplied CLISTs to send TSO datasets and/or messages.

The **destination** can be one of the following types:

Personal Services This can be either PS/CICS 1.3 or the Personal Services component of OfficeVision/MVS.

An API application An API application which uses the same subset of DISOSS commands as Personal Services[1].

A SNADS node Any other SNADS node. This may be another OfficePath system.

A CICS Application A VSAM KSDS, called an out-tray, is used as the interface between the CICS application and OfficePath.

An Internal Reader The document text is included, as in-stream data, in a JCL deck submitted to an internal reader.

[1] The OfficePath programs intercept DISOSS API requests by replacing the API module DSVAW100 (or DMDAW130 module for Personal Services). This module is invoked via an EXEC CICS LINK command which contains a list of storage addresses. This means that OfficePath must be in the same region as the API programs.

A Batch Application The document text is written to a VSAM KSDS, called an out-tray, and a batch job is submitted to process it.

TSO The document is sent to the TSO user using an internal reader. The user can view the document using the supplied CLISTs.

Services not provided by OfficePath

The following facilities are not provided by OfficePath:

1. Document Library Services

   The FILE, SEARCH, RETRIEVE and DELETE commands, which may be issued from within a DISOSS API application (for example, Personal Services), are not supported. These requests will be rejected, with an error status message.

2. Password Validation

   OfficePath does not support password validation. It is the responsibility of CICS applications to carry out Password validation. OfficePath includes a sample user exit, which is invoked by the batch programs, and may be used to develop a password validation mechanism.

3. RGN (Routing Group Name)

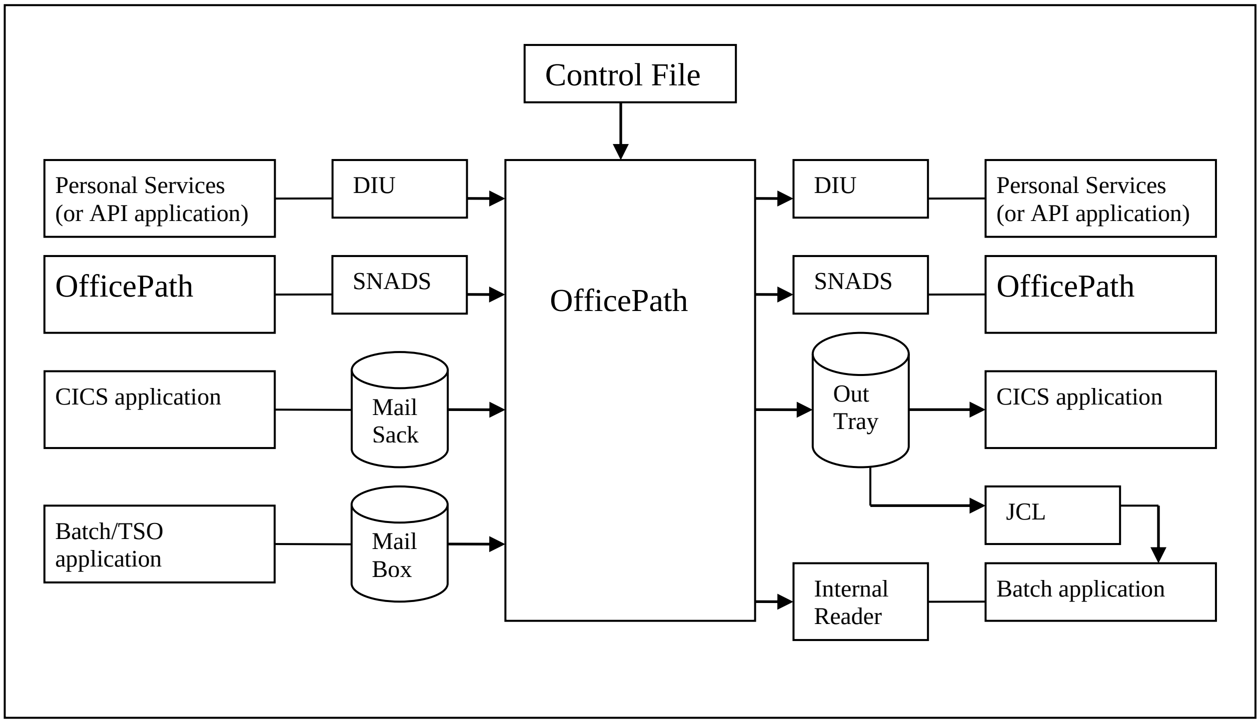
   OfficePath does not support the RGN: it is left blank for requests sent to another node and is ignored for requests from other nodes.

Note: The OfficePath Library Retrieval Feature provides on-line search and retrieval of documents from an existing DISOSS library. Full details are in the *OfficePath Library Retrieval Feature Installation and User Guide*.

System Overview

OfficePath is represented schematically as follows:

Figure 1. OfficePath schematic arrangement



In summary, OfficePath provides the following:

- Connectivity to Personal Services or API application
  OfficePath communicates with PS in the same way as DISOSS, using IBM's Document Interchange Architecture (DIA). However, OfficePath only supports a limited subset of DISOSS functions, specifically:

  SIGNON All DCA levels and GCIDs are ignored. The values from the OfficePath control file are used.

  SIGNOFF No options can be specified.

  REQDIST All options not involving library documents.

  LIST Unformatted LIST command only.

  OBTAIN Obtain all documents only.

- The following features are **not** supported:

USERSETS OfficePath does not support USERSETS. However, OfficePath provides an equivalent facility by allowing users to be registered generically (for example all users at the same address can be registered with a single registration record).

FORMAT41 Format 41 addressing in either originator or recipient fields is not supported.

- Connectivity to other OfficePath systems.

  OfficePath communicates with other OfficePath systems using IBM's SNA Distribution Services architecture (SNADS).

- Connectivity to CICS, batch and TSO applications.

  OfficePath communicates with your applications using proprietary mechanisms which are described in the *OfficePath Programming Interfaces* manual.

- Conversion between the following document types defined by IBM's Document Content Architecture:
  - Revisable Form Text (RFT)
  - Final Form Text (FFT)
  - EDITABLE Notes and Documents

  Document conversion is performed by "formatter" programs. For RFT documents, **DisplayWrite/370** can be used to perform conversions; alternatively OfficePath provides sample programs to perform the conversions.

PART ONE - DIAGNOSING PROBLEMS

Chapter 3. Recommended approach to diagnosis

This chapter describes the recommended approach when attempting to identify/diagnose a problem. Generally the realization that a problem has occurred comes some time after the situation that caused the error. In these cases, the best approach may be to try the request again; this will re-generate any error messages, making them easier to locate. If possible the "SEND" should be to one recipient only, as this is easier to trace through the log.

If possible, perform the diagnosis in the following sequence:

1. Re-send the request (using one recipient).

   If the request initiated in Personal Services the recipient causing the error can usually be identified by using the Personal Services "Work with Status information" option. The recipient causing the error will be the one with a status of "ACCEPTED".

2. Use the **PIPY** transaction to locate the request.

   The PIPY transaction is described in the *OfficePath Installation and Administration* manual. There should be an "M" record for the request on the log (this will be a "SEND" command from the originator's userid). Check the status on the "M" record against the appropriate table in "Sample transaction flows". This will show the extent to which OfficePath has processed the request.

3. Check the CICS log (CSMT) for error messages.

   These will be in the form PIPnxxx, where "PIPn" is the transaction issuing the message and xxx is the message number. The description of these error messages can be found in "Messages written to CSMT".

4. Correct/delete the invalid request.

   Attempt to correct the error, using the error message description and the transaction description from "Interaction between modules", attempt to correct the error. Some of the common errors which occur are described in "Common problems".

   If the request is in error, then it can be deleted as described in "Purging invalid or corrupt requests from PIPLOG".

5. Restart OfficePath using the **PIPS**

   This will clean-up the files and restart any "in-flight" requests. The **PIPS** transaction can be invoked either from "blank" CICS screen or by pressing PF2 from within the "Log File Browse" (PIPY). If the **PIPS** transaction fails then you should crosscheck the files as described in "Crosschecking OfficePath files". If any other transactions fail then you may need to purge requests as described in "Purging invalid or corrupt requests from PIPLOG".

6. Request additional support.

   Contact the TBS Software Ltd Product Support team as described in "Requesting Additional Support"

OfficePath contains sophisticated recovery techniques which it uses during its restart processing. These are used to ensure the integrity of the files and to restart any in-flight requests. Whenever an error has been corrected, the OfficePath system should be restarted to ensure that data on all its files is consistent.

Chapter 4. Common problems

This chapter describes the recommended approach to resolve some of the common problems which can occur.

Personal Services not receiving mail or status

Mail has been sent to a Personal Services user who has not received it (the "M" record is still on the PIPLOG file), or the Personal Services user has sent mail and its status has not been updated. This may occur, for example, in the following cases:

- If the recipient's application has not generated a status

  If the "Returns status information" flag is set on the Control File record for the recipient destination, then OfficePath will be waiting for the Application to return a status. Check the request on the log-file to see how far the processing has got. If the Application is unable to generate a status, then this flag should be set to "N".

- If a new Personal Services user is defined and mail is then sent to him **before** he has signed-on for the first time.
- If a "LIST" or "OBTAIN" request has been deleted from PIPLOG.
- If a "LIST" or "OBTAIN" request has been rejected by Personal Services.
- If PTF UP80565 (or replacement) has not been applied to PS 1.3.

Note: This can also occur if the PIPLOG dataset has been deleted and re-defined.

Solution: Personal Services maintains internal flags which are used to determine the current LIST/OBTAIN status. If a LIST/OBTAIN request is in progress, Personal Services will not issue another LIST/OBTAIN request, but instead sets a flag indicating that it needs to issue another request when the current one has completed. To **force** Personal Services to carry out a LIST/OBTAIN:

1. Sign-on the user

   Personal Services resets the flags at sign-on time. It may issue a LIST/OBTAIN command if the flags were set.

2. Send another mail item to the User

   Note: This must be from another node/application or internal distribution (PS user to PS user) will be used.

3. Send an item from the user to an invalid userid

   This is will cause Personal Services to receive a status message.

4. Sign-off, and then sign-on, the User

   This should reset the flags and cause a new LIST to be created as no response has occurred from the previous requests.

   This last step should only be attempted if all previous steps have failed.

5. Delete the user's MCD record.

   The record can be found on the DMDMC30 dataset. The key of the record is the userid preceded by an "M". Once the record has been deleted, sign-on as the user and send a mail item to an invalid userid.

Requests sent by an Application are not confirmed

A document is sent from an Application (CICS or Batch) and the originator's status for the request is not updated to complete, or the COD document is not returned, even though the recipient has received it. The possible causes are:

1. The reply destination and/or user specified by an **application** are invalid.

   The reply destination must be defined on the Control File and must be of type **OUTTRAY** or **INTRDR**. The Username must point to a destination of type **OUTTRAY** or **INTRDR**.

   Note: An error message should have been written to CSMT indicating that COD was turned off for this request.

2. A recipient **application** has not generated a status.

   If the "Returns status information" flag is set on the Control File record for the recipient destination, then OfficePath will expect the application to return a status. Check the request on the log-file to see how far it has been processed. If the application is unable to generate a status, then this flag should be set to "N".

3. An error has occurred while returning the COD document.

   If the request record on PIPLOG is still an "M" record (with status **COD NFY**), then the problem is in OfficePath. Follow the instructions for diagnosing OfficePath problems. If the request is an "X" record then the problem is in the destination system receiving the COD document. Follow that system's instructions for diagnosing problems.

Certain characters are not displayed correctly

A document has been sent point-to-point through OfficePath. When the recipient views the document, the text is not displayed correctly (for example, the pound sterling character is displayed as dollar). Possible reasons are:

1. Requests sent from an Application are not specifying the correct GCID.

   The Application can specify a GCID or allow it to default.

2. The "LOCAL GCID" field on the CUSTOM record on the Control File is incorrect.
3. The GCID specified on the recipient destination record on the Control File is incorrect.
4. There is no GCID translation table.

   Check CSMT for any error messages.

5. Personal Services Terminal GCID table is incorrect.

   The DMDTMTGT table identifies the GCID which Personal Services uses when sending data to the terminal. The default value supplied is US.

Document cannot be viewed or contains garbage

A document arrives in Personal Services, but one of the following occurs when the document is viewed:

- "DDD7114 Document has incorrect format"
- "DDD7002 The output disk cannot be accessed"
- The document text contains garbage.

Possible causes are:

1. The sender specified the wrong document type.

   Check that the document can be viewed by the sender and, for an Application, that the document type specified on the send is correct.

2. The document has no document name.

Try viewing the document after filing it in the File Cabinet (specifying a name). Correct the Application to ensure that a name is always supplied.

3. The formatter program is incorrect.

   If the request has been formatted (that is converted to a different document type) then check the formatter program.

No requests going to/coming from an Application

A document has been sent to an Application, but has not arrived, or sent from an Application and has not arrived. Possible causes are:

1. No INPUT record for Application In-tray

   If the In-tray to which the application writes is not defined to OfficePath then no request on it will be processed. Define the in-tray to OfficePath as described in the *OfficePath Programming Interfaces* manual.

2. In-tray processor has stopped

   If the INPUT request for the In-tray (on PIPLOG) has a last-updated time which is earlier than the current time, then the In-tray processor has stopped. Follow the OfficePath Diagnosis procedure as described in "Recommended approach to diagnosis".

3. Request was invalid.

   If the request is logged as an "X" record, then the request was invalid. Check the source system which issued the request.

   If the request is logged as an "M" record coming from an Application, then check that there are no problems with the recipient.

   If the request is logged as an "M" record going to an Application, then check CSMT for OfficePath errors.

4. Application has not processed the request

   The fault is in the Application. Check for errors in the Application as described in the Application documentation.

No requests going to/coming from a remote node

If you suspect that requests to, or from, a remote SNADS node are not being processed, then check the following:

1. Link not acquired

   Use the PIPZ transaction to check that the link to the remote system is acquired.

2. No sessions available

   Use the "OPE" command to ensure all sessions are bound.

3. Check CSMT for errors

   Look for any PIPSxxx error messages on terminals associated with the link to the remote node.

4. Close/re-Open link

   Shut-down the link using the "CLO" and "REL" commands. If necessary, use "PUR" command to end any outstanding sessions. Re-Open the link.

Note: If OfficePath is failing to communicate with a remote node then any requests coming from that node will not have been logged on the log-file of this (local) OfficePath. Requests which are waiting to be sent to that remote node will have been logged on this local log-file.

NOTES arrive as EDITABLE documents

When a NOTE is sent to a Personal Services user, with Receipt Acknowledgement, it arrives as an EDITABLE document rather than as a NOTE. This problem only occurs for NOTES which have been distributed using OfficePath, not NOTES which used Personal Services internal distribution. The problem is due to a bug in the Personal Services DIU parser which does not process the attribute byte correctly. The attribute byte contains the NOTE and the RA indicators. If the RA indicator is ON, then the NOTE indicator is ignored and the document treated as EDITABLE.

This is a Personal Services bug which has been addressed by OV/MVS PTF UY56372.

AFCL abend in OfficePath Transactions

OfficePath transactions abend with abend code AFCL while executing a CICS WRITE command. This occurs if the length of the record which is being written is larger than the CICS DFHJ01A/B block size. The CICS Journal 1 **must** have a block-size greater than 8K for OfficePath to run correctly.

ATSP abend in OfficePath transactions

OfficePath transactions abend with abend code ATSP when attempting to write to temporary storage. This will occur if Temporary Storage Table contains a generic entry which includes the "PIP" prefix. [2]

The OfficePath transactions use AUXILIARY Temporary Storage Queues as an extended Working Storage area for the program. The transaction ensures that the queue does not exist by issuing a DELETEQ TS command before writing to the TSQ. All processing which requires data from the TSQ is done within the transaction which creates it. As such, the TSQ does not need to be recoverable.

The solution is to modify the Temporary Storage Table (TST) so that the PIP TSQs are not recoverable.

Deadlock between transactions DMDC and DMDD or DMDF.

This occurs under OfficeVision/MVS only and is due to an inconsistent approach to updating the DMDMC30 dataset. These transactions are never invoked directly by a user, only in response to a request issued by a user.

If this problem occurs then you **must** ensure that transactions DMDC, DMDD and DMDF are single threaded [3] to prevent a deadlock. This should not impact on-line performance since these transactions are only invoked for background processing.

[2] The TSQ names used all have the same prefix as the OfficePath transactions (the supplied value is "PIP").

[3] This can be done by putting the transactions in the same transaction class (TCLASS or TRANCLASS) and setting the maximum number of tasks for that class to 1. Refer to CICS documentation for more information on transaction classes.

PIPGS3B abends or produces unexpected results

PIPGS3B uses the password validation exit (PIPUSRCK) to validate the user specified. The supplied version of this exit changes the user to the RACF userid submitting the job and this userid is extracted from system control blocks. The locations of these control blocks have changed in different releases of the MVS operating system and so unexpected results may occur of the program is run on a different release of MVS from that used when the program was originally assembled.

If this happens then PIPUSRCK should be reassembled and linked, as described in the OfficePath Installation and Administration manual.

Chapter 5. Purging invalid or corrupt requests from PIPLOG

This chapter describes how the OfficePath log file browse transaction can be used to delete invalid requests. This method should only be used in the last resort if the request is causing system problems and it is not possible to complete the request by changing the status of the recipients. The recipient status can be updated using the **PIPY** transaction as described in "Recipient Details Panel" in "Chapter 40. Message File Browse" of the *OfficePath Installation and Administration* guide.

If a request is purged, then:

- Any recipients who have not yet received the document, will not receive it!
- No status information will be returned to the sender.
- The Auditing exit will not be invoked.
- The log file status will be updated to **CANCEL**.

A request can be purged from the message file browse panel by selecting the item. This will display the "Message Details" panel for the item. Change the function to "PURGE". At this point you will be prompted to press *enter* again in order to confirm the request. On pressing *enter*, the request is deleted.

Multiple requests may be purged simultaneously by entering 'P' against the requests on the message file browse panel, and pressing *enter*. The panel is then redisplayed with the selected requests highlighted, and *enter* must be pressed again to confirm the purge.

After deleting requests the OfficePath system should be re-started immediately.

Note: Failure to re-start OfficePath after purging a request may result in unexpected ABENDs because the OfficePath files will not be synchronized.

Chapter 6. Holding Mail and Status Requests

As part of problem determination you may require to prevent new mail and status requests from being sent. This can be achieved by using the on-line Control File Maintenance transaction - PIPX:

1. ENQUIRE on the DESTINATION you wish to "hold"
2. Change the function depending on what to want to hold:

HOLDALL	to prevent all MAIL AND STATUS requests from being sent
HOLDM	to prevent all MAIL requests from being sent
HOLDS	to prevent all STATUS requests from being sent

3. Press <ENTER> to action the request: the panel will be updated with informational text.

To "release" the mail and status do the following:

1. To "release" the mail and status change the function to AMEND and press <ENTER>.
2. Restart OfficePath.

Chapter 7. Crosschecking OfficePath files

If the OfficePath files are corrupted then they may become out of synch leading to abends reporting the NOTFND condition or messages reporting missing header or trailer records.

If this occurs then OfficePath may be restarted using the FILES option. This option crosschecks the files, purging any corrupt requests and inserting missing header or trailer records before restarting OfficePath

To use this option type PIPS FILES on a blank CICS screen. Any errors found will be written to the CICS log.

Chapter 8. Requesting Additional Support

If, after reference to this manual, you cannot resolve the problem, you will need to contact Technical Support - the following information should be obtained before doing so:

1. Details of any CICS abends, including times, transactions involved and abend codes.
2. Details of OfficePath, CICS and Personal Services levels. If the problem concerns a SNADS connection, details of the receiving SNADS node type, release level etc, will also be required.
3. Access to the following transactions may be required in order to diagnose the problem.

PIPX	On-line maintenance transaction
PIPY	On-line log file enquiry
PIPZ	On-line link monitor
CECI	CICS command interpreter
CEMT	CICS master transaction
CEDF	CICS debugging tool
TMON	or another CICS monitor

4. **Note:** The CICS internal transactions usually require a special CICS sign-on. Please make sure that you have access to these transactions **before** you call the Support Team.
5. Access to the CICS log/operator console display

   Any OfficePath error messages will be written to the CICS log. If you cannot access the log then Technical Support cannot determine the problem.

   Note: Please make sure that you have the correct log. If you issue a "find" command for "PIPS" (while viewing the log), you should find a PIPS116 message.

Contact Technical Support as described on the front of this manual (see **Error! Reference source not found.**)**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**

When through to the Technical Support:

1. Technical Support will first determine whether you have a maintenance agreement or if the product is still under warranty; if neither of these apply all support activities are chargeable on a time and materials basis.
2. Technical Support will then attempt to resolve the problem over the telephone.
3. If it is not possible to resolve the problem on the telephone, Technical Support may need to do further investigative work on the problem.

4. Technical Support may request OfficePath trace information. This trace information is obtained by first setting the appropriate trace flag on the SYSTEM record maintained using the PIPX transaction, and then reproducing the problem. The trace flag should then be reset and the requested trace information printed and sent to Technical Support.
5. If it is still not possible to resolve the problem additional information, such as CICS dumps may need to be sent. These dumps may be sent on 3480 (NOT 3490) cartridge or electronically if the dump is not too large).

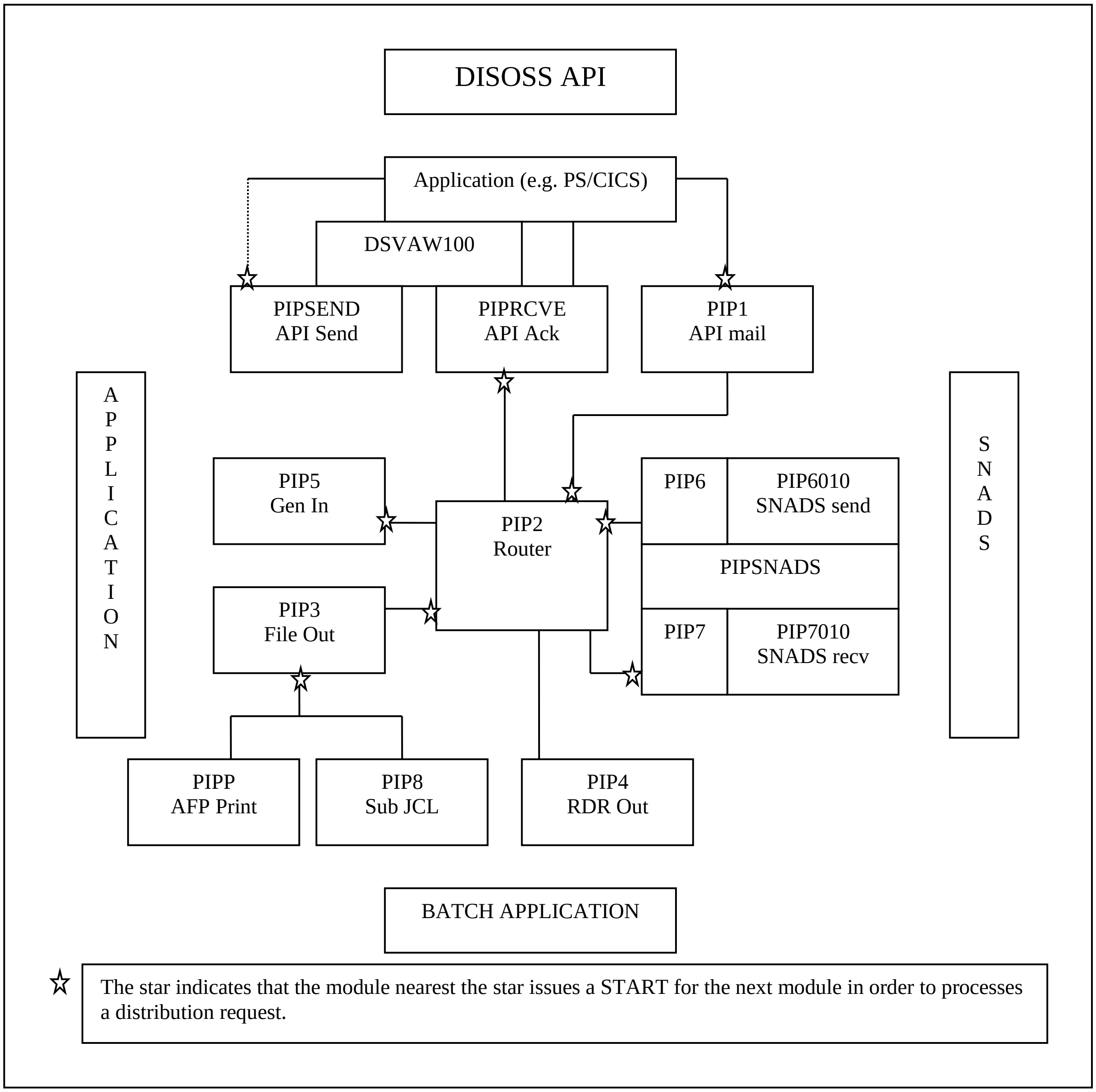
PART TWO - INTERNAL LOGIC

Chapter 9. Interaction between modules

Overview of transaction interaction

This chapter shows how the OfficePath modules interact with each other to process distribution requests. This is intended to be used in combination with an error message to determine why and, if necessary, where an error occurred. A short description of the function of each of the modules is given. This may be useful in determining the reason for the error.

Figure 2. Transaction flow within OfficePath



☆ The star indicates that the module nearest the star issues a START for the next module in order to processes a distribution request.

Personal Services does not use the DSVAW100 module, which is normally used by API applications. It has its own version of this program which is called DMDAW130. Functionally the two modules are identical.

DSVAW100 - DISOSS API module

DISOSS API programs invoke the DSVAW100 module [4] whenever they wish to communicate with DISOSS. This module normally reads/writes data from the DSVAPIQ dataset. The "data" is a DIU which is being sent to DISOSS or received from DISOSS.

When OfficePath is installed, the DSVAW100 (and DMDAW130) module is replaced by a module on the OfficePath **LOAD** library. The OfficePath DSVAW100 module intercepts the requests, before they are written to/read from the DSVAPIQ, and passes the DIU to the PIPSEND module (for a send request) or gets a DIU from the PIPRCVE module (for a receive request).

Note: The "DISOSS API" module uses the term **SEND** to indicate issuing a request to DISOSS (for example SIGNON and LIST) and the term **RECEIVE** to indicate receiving a response (for example ACK and DELIVER) from DISOSS.

PIPSEND - API Request processor

PIPSEND parses DIUs sent from a DISOSS API application to determine the request type (for example Request Distribution, List, Obtain, Sign-On, File, ...). The processing will depend on the type of request which has been issued:

REQUEST-DIST The document text is written to the PIPTEXT file, a PIPSTAT record is created for each recipient (with processed flag set to N), and a PIPLOG record is created with the status set to **PENDING**. The appropriate API reply transaction [5] is started.

PRINT REQUEST The document text is written to the PIPTEXT file, a PIPSTAT record to **ACCEPTED**) is created for the printer name (using the host print location from the SYSTEM record), and, a PIPLOG record is created with the status set to ACCEPTED. The PIP2 transaction is started to process the request.

[4] Personal Services actually invokes a program called DMDAW130 but this module is functionally equivalent to the DISOSS module.

[5] For Personal Services the transactions are:

DMDC	Response of all API requests expect Obtain
DMDF	Response to Obtain command
DMDD	New Mail notification transaction

SIGN-OFF This request is parsed but no action is performed. Personal Services does not expect a response so no reply transaction is started. The request is not recorded on PIPLOG.

Other A PIPLOG record is created [6] and the reply transaction is started. For a LIST command the reply transaction is started with an interval (20 sec) for efficiency (this allows potentially more than one status to be reported).

Note: The SIGN-ON command is recognized and acknowledged, but any document types specified in it are ignored. The "acceptable" document types are identified by the appropriate destination record on the PIPCNTL file.

PIPRCVE - API Request Acknowledge

PIPRCVE, which is invoked by the API response transaction, builds a reply DIU and returns it to the calling application. It reads the PIPLOG file to determine the type of request which was issued and, generates either an ACKNOWLEDGE or a DELIVER DIU depending on the request issued.

SIGN-ON A SIGNON REPLY is generated indicating that the SIGN-ON was successful. The log record is deleted if "Log all requests" is set to "N".

LIST A DELIVER DIU containing an unformatted LIST DOCUMENT is generated by reading the PIPMAIL file (containing a list of mail sent to the user) and the PIPSTAT file with processed flag set to 'P' (containing a list of status information, by recipient, for each document sent by the user). The PIPSTAT records are updated and the processed flag set to 'Y'. The log record is deleted if "Log all requests" is set to "N".

OBTAIN A DELIVER DIU is created for each entry on the PIPMAIL file. The PIPSTAT file, for the sender, is updated to indicate that the document has been **DELIVERED** (with processed flag set to 'P') and PIP2 is STARTed to return the updated status to the sender.

The PIPMAIL entry is deleted after it has been processed.

REQUEST-DIST An ACKNOWLEDGE DIU is created indicating that the request has been accepted by OfficePath. The PIP2 transaction is STARTed to process the recipients.

Other An ACKNOWLEDGE DIU is created which contains a NAK code indicating whether or not the request was processed. For all unsupported requests (for example FILE or SEARCH) a NAK code of C30307 (Unsupported Function) is returned. For a print request which specified an invalid printer a NAK code of

[6] The log record is created even if logging is turned off. The record is then deleted when the reply has been returned.

C40209 (Invalid Printer) is returned. The log record is deleted if "Log all requests" is set to "N".

PIP1 - API Mail/Status processor

This transaction is used to notify the API application when either new mail or status information arrives for a DISOSS API user. If the DEST record indicates that the user is a Personal Services user (the HUD name value on the DEST panel is non-blank) then the PIP1 transaction attempts to validate the user by reading the Personal Services HUD file. Otherwise, the user validation program PIPAPIV is invoked to verify the userid.

If the userid is valid, then the program checks the PIPSTAT file for a 'list' record which indicates if a LIST or OBTAIN command is currently being processed. If a request is currently active then the transaction ends, otherwise the mail/status transaction from the DEST record is STARTed. The DISOSS API module will then issue LIST and/or OBTAIN commands to get the status information and/or mail.

If the userid is not valid, then the PIPSTAT entry for the recipient will be updated to **INVALID** (and processed flag set to 'P') and PIP2 is STARTed to return the updated status to the sender.

Note: If a **PSFAMILY** record is present for the recipient's address then PIP1 will check the recipient against each HUD dataset in turn. In this case the values from the DEST record are ignored. However, you must enter a transaction and HUD dataset name on the DEST panel so that OfficePath knows that the recipient is Personal Services. The default values are DMDHU30 and DMDD respectively.

PIP2 - Document/Status routing and document conversion

PIP2 will be invoked either to process distribution requests or to process status information (including being started by the start-up transaction in order to purge an active - "M" - request).

Distribution Requests

PIP2 examines all the recipients of a distribution request by reading through the PIPSTAT records (for the request) and processed each one as follows:

1. Check the PIPCNTL file for a matching user record. The search order is:
    a. Address-Userid
    b. Address-*
    c. *-*

2. The user record contains a destination-id. PIP2 will then read that destination record to determine the type of mail which can be received and the mail processor (API, OUTTRAY, INTRDR, or SNADS) which will be invoked to deliver the mail.
3. The validation exit - PIPEXIT1 - is invoked so that additional validation (including alteration) of each recipient can be performed: for full details of the exit see the *OfficePath Installation and Administration* If the user (or node) is changed then the user and destination records will be re-read.
4. If a document conversion is required, then the document text will be translated (using a formatter program) and the translated version stored on the PIPTEXT file.

   The document translation is determined by:

   OUTPUT type: the OUTPUT document type requested by the Application (CICS or Batch)

   Current type whether the document can be received without conversion: this is determined by the originating application - if this is an API application it cannot be EDITABLE (it will be converted to FF TEXT before it reaches OfficePath)

   Destination type the first document type, from the destination record, to which the current document can be converted (that is for which there is a formatter program)

   PIPEXIT1 The exit can override the document type to force a particular conversion.

   Note: Document conversion to a particular document type is only performed once for each request. Thus if two recipients are both to receive an RFT document as EDITABLE, the document conversion from RFT to EDITABLE is performed once and the converted copy sent to both recipients.

   The interface to the formatter program is that used by DisplayWrite/370 and is documented in the DisplayWrite/370 V2 Program Interfaces for Customers manual.

5. If the recipient is valid then
   a. An entry will be created on the PIPMAIL file for the recipient.

      If the destination specifies "Accumulate Recipients", then all recipients on that destination will be written to one PIPMAIL record and the mail processor will be STARTed once only to process all the recipients. For example, SNADS type destinations are implicitly defined with "Accumulate Recipients" to ensure that the document text is sent across the LU6.2 link only once, with a list of recipients, rather than once per recipient. "Accumulate Recipients" may only used with application programs which have been coded to respond to multiple recipients.

   b. The appropriate mail processor (PIP1, PIP3, PIP4, or PIP6) will then be STARTed to distribute the mail.

otherwise (user record is set to **INVALID**, **CANCEL**, or **ERROR**)

c. the PIPSTAT record will be updated with the status (and processed flag set to 'P')
d. the appropriate processor STARTed to return the updated status to the sender.

The PIPEXIT1 program will be invoked to indicate the end of the recipients at which time the exit program can request that additional recipient are added or that the document text be passed to the program. Any additional recipients will undergo the same processing as the original recipients with the exception that the exit program will not be invoked to validate them.

Status Processing

PIP2 will be invoked each time it is required that status information is to be returned to the sender, and each time a request is complete or has been cancelled by the start-up transaction PIPS.

1. When status information is to be returned the appropriate processor (PIP1, PIP3, PIP4 or PIP6) is invoked depending on the destination type of the sender.
2. If the mail distribution request is complete, or has been cancelled by the start-up transaction, the processing is as follows:
   a. The active log record is deleted (changed from an "M" to an "X").
   b. The Audit Exit program - PIPEXIT2 - is invoked passing the request profile: for full details of the exit see the *OfficePath Installation and Administration*
   c. For each recipient the exit program is invoked passing the details from the PIPSTAT record. The PIPSTAT records are then deleted if the keep status records flag is set to 'N'.
   d. The exit program is invoked to indicate the end of the recipients at which time the exit program can request that the document text be passed to it.
   e. The corresponding PIPTEXT records are deleted.

PIP3 - Out-tray Mail/Status processor

This transaction is used to process mail or status information which is sent to an out-tray for processing by an application. The application can be either a CICS or BATCH application.

If mail is sent to an out-tray recipient, then the PIP3 transaction will read the PIPMAIL file for the request. It will then copy the document from the PIPTEXT file to the out-tray. The PIPSTAT record for the recipient is updated to **DELIVERED**. The PIPMAIL entry will be deleted after it has been processed.

If status information is returned to an out-tray, then the PIP3 transaction will read the PIPSTAT records to get the updated status. Any PIPSTAT records with processed flag set to 'P' are used to create a COD document. The PIPSTAT processed flag is then updated to 'Y'.

Note: If the destination is a batch out-tray, then PIP3 will serialize access to it using an MVS ENQ. The PIP3 transaction will also submit JCL, if present on PIPCNTL, to process the file. If a JCL record for the destination exists on PIPCNTL, then the PIP3 transaction will check whether any JCL is waiting to be submitted. If there is no JCL waiting then the PIP3 transaction will START the PIP8 transaction, on a time interval, to submit the JCL. The time interval is controlled by a field on the control file SYSTEM record.

If the destination is a CICS out-tray, you may specify the name of a transaction to be STARTed whenever data has been written to the out-tray. The transaction can be STARTed on a terminal, if required.

The recipient destination record can specify that the application will generate a status ACK (on an in-tray file) which will be used by the PIP5 transaction to update the PIPSTAT file. In this case the PIP3 transaction will not update the PIPSTAT file when the document is written to the out-tray. This feature may only be used for applications which have been coded to generate a status ACK.

PIP4 - Internal Reader Mail/Status processor

This transaction is used to deliver mail or status information to a batch application (by sending the document/status as in-stream data).

If mail is sent to an internal reader recipient, then PIP4 will create a batch job containing the document text as in-stream data. The document text is taken from the PIPTEXT file. The PIPSTAT records for the recipient will be updated to **DELIVERED**. The PIPMAIL entry will be deleted to indicate it has been processed.

The JCL for the batch job will be taken from the PIPCNTL file, using the search order:

1. Address.User
2. Address.*
3. Destination

Note: The JCL, which is submitted, can be modified with substitutions from the profile of the document which has been sent. For example, the name of the document substituted as a dataset name in the JCL.

The recipient destination record can specify that the application will generate a status ACK (on an in-tray file) which will be used by the PIP5 transaction to update the PIPSTAT file. In this case the PIP4 transaction will not update the PIPSTAT file when the document is written to the internal reader. This feature may only be used for applications which have been coded to generate a status ACK.

If status information is returned to an internal reader, then the PIP4 transaction will read the PIPSTAT records to get the updated status. Any PIPSTAT records with processed flag 'P' are used to create a COD document. The PIPSTAT flag is then updated to 'Y'. The COD document is included as in-stream data in a batch job. The JCL for the batch job is taken from the PIPCNTL file, using the reply destination.

For a status request the JCL for the batch job is taken from the PIPCNTL file using the reply destination.

PIP5 - In-tray processor

This transaction processes requests which are written to in-trays by applications. Multiple PIP5 transactions are started by OfficePath start-up processing - one to process each in-tray. Each PIP5 transaction has a special PIPLOG log record which it updates every time it reads the in-tray. This is used as an indication that PIP5 is processing the in-tray and a mechanism to ensure against multiple PIP5 transactions processing a single in-tray.

When a distribution request is found on the in-tray, PIP5 creates a PIPLOG record for the request, writes the document text to the PIPTEXT file and the recipients to the PIPSTAT file. If the application uses copylists, then these are expanded and the list of recipients added to the PIPSTAT file. The PIP2 transaction is then STARTed to process the request.

When a status ACK request is found on the in-tray, PIP5 updates the PIPSTAT record for the recipient (setting processed flag to 'P') and STARTs PIP2 to return the status to the sender. The status ACK request is then deleted from the in-tray.

If no requests are found on the in-tray the transaction re-STARTs itself using the interval on the control file SYSTEM record.

Note: PIP5 ensures that the sender's userid and the reply destination refer to **OUTTRAY** or **INTRDR** type destinations. If they do not, then COD is set to NO (no confirmation is generated). This is to ensure that COD information, requested by an application, is directed to an application and not, for example, to Personal Services or a SNADS node.

PIPSNADS - SNADS protocol module

The PIPSNADS module handles the LU6.2 communication for SNADS. It is invoked by the PIP6 and PIP7 transaction codes. The link protocol is handled using LU6.2 unmapped (EXEC CICS GDS) commands at SYNCLEVEL

The module is invoked as PIP6 when mail/status is sent to a SNADS node. In this case it LINKs to the PIP6010 program to build a SNADS DISTRIBUTE DIU which is then sent.

The module is invoked as PIP7 when mail/status is received from a SNADS node. In this case a LINK is made to the PIP7010 program in order to parse the received DIU.

Note: The PIP7 transaction has a transaction alias of X'21F0F0F2'. This is the transaction name which is reserved for SNADS processors. Thus, the PIP7 transaction will be STARTed whenever CICS receives a SNADS distribution from another node (for example an OfficePath system in another CICS system).

PIP6010 - Create SNADS DIU module

This module is used to create a SNADS DISTRIBUTE DIU which is sent to another node. The PIPSNADS module, invoked as transaction PIP6, LINKs to this module (PIP6010), which builds the DIU, and then the PIPSNADS module sends the DIU which is returned.

If mail is sent to a SNADS recipient, PIP6 will read the PIPMAIL record for the request. It will create a type=DATA DIU containing the document text, from PIPTEXT, and the list of recipients, from PIPMAIL. The PIPMAIL entry is deleted after the request has been processed.

If status information is returned to a SNADS sender, PIP6 will read the PIPSTAT file and create a type=STATUS DIU. All PIPSTAT entries with processed flag set to 'P' will be sent in the DIU. The PIPSTAT processed flag is then updated to 'Y'.

PIP7010 - Parse SNADS DIU module

This module is used to parse a SNADS DISTRIBUTE DIU which has been sent from another node. The PIPSNADS module, invoked as transaction PIP7, receives a DIU, LINKs to this module (PIP7010), which processes the request contained in the DIU, and then PIPSNADS completes the LU6.2 conversation. The received DIU may be either a type=DATA (a request distribution) or type=STATUS (an acknowledgment).

If the request is a request distribution, then a PIPLOG record is created. The document text is written to the PIPTEXT file and the recipients are written to the PIPSTAT file with processed flag set to 'N'. The PIP2 transaction is then STARTed to process the request.

If the request is an acknowledgment, then the PIPSTAT records for the recipients being acknowledged are updated (with processed flag set to 'P') and PIP2 is STARTed to return the updated status to the sender.

Note: For a request distribution, the sender is validated on the PIPCNTL file. If the sender is not a **SNADS** type destination, then COD is set to NO (no confirmation is generated). This is to prevent the COD being sent to an Application, or to Personal Services.

PIP8 - Batch Out-tray JCL submit

This transaction is invoked to submit application JCL to process an out-tray defined as BATCH. The transaction is STARTed by the PIP3 transaction after it has written a request to the out-tray. The PIP8 transaction reads the JCL from the PIPCNTL file and submits it, using the internal reader name from the control file SYSTEM record. The PIPLOG record is deleted to indicate that the JCL has been submitted.

Note: No JCL parameter substitution is performed by OfficePath.

PIPS - Start-up Transaction

The PIPS transaction is used to start OfficePath. This transaction may be entered from a terminal, invoked directly from PIPY, or invoked automatically by using PIPSPLT as a PLTPI entry. This transaction tidies-up all the OfficePath files, excluding in-trays and out-trays, by ensuring that:

- The special "INPUT" records on PIPLOG are deleted.
- Any PIPLOG "X" records older than the value on the control file SYSTEM record are physically deleted (removed from the file).
- Any PIPLOG "M" records older than the value on the control file SYSTEM record are set to cancelled and PIP2 is started to complete the request.
- Any records on any of the OfficePath internal files which do not have a corresponding entry on the PIPLOG file are physically deleted (removed from file).
- The PIP5 transaction is STARTed for each of the INPUT files.
- Any "in-flight" requests are re-STARTed.

PIPP - Advanced Function Print Feature

Requests sent to the AFP feature of OfficePath are written to a file and the PIPP transaction is STARTed. This transaction then reads the request from the file and invokes DisplayWrite/370 to perform the print.

The file may contain host-print or mail-room print requests. The program distinguishes between mail-room print requests and host-print requests by the recipient address. If the recipient address matches the host-print location (from the SYSTEM record) then it is assumed to be a host-print otherwise it is assumed to be a mail-room print.

Host-print requests use the recipient address as the LDE name when invoking DW/370.

Mail-room print requests use the name of the DEST record to which the recipient's USER record points as the LDE name.

Chapter 10. Files used by OfficePath

This chapter briefly describes the contents of each file used by OfficePath. It also includes information about the usage of records on the files, which may be used to assist in sizing the files.

Note: Because of the interconnection between the files, one file should never be deleted/redefined, since other files may point to records on it. The files should only be re-organized, or deleted, as a group.

PIPCNTL - Control File

This file contains control records used by the OfficePath modules. The following information is stored on the control file:

User These records identify all valid users, either explicitly or generically. User records point to destination records.

Destination These records identify the types of document users can receive and also the type of processor to be used when sending mail.

JCL These records contain JCL which is submitted by OfficePath. The JCL is used to invoke batch applications.

COPYLIST These records contain lists of recipients which are used as a shared distribution list facility. These can only be accessed by Application interfaces.

INTRAY These records identify the input files which OfficePath is to process.

PS Family These records are used to allow OfficePath to support multiple Personal Services systems (which do not share the HUD and DPD/DRD datasets) without requiring all Personal Services users to be explicitly defined to OfficePath.

SYSTEM This record contains various values used to control OfficePath operation.

All these records may be maintained using the on-line maintenance transaction PIPX.

In addition the control file contains:

- GCID tables
- Formatter tables
- Error messages
- Help information

These records may only be updated by the batch PIPCTL program.

This file should be fairly static in size. The OfficePath transactions only read/update records. Records are added/deleted using the on-line maintenance transactions.

PIPLOG - Log file

This file is keyed on the record type ("M" or "X") and the message number (4 bytes binary).

The file consists of two types of record, "M" and "X". It is used as an activity monitor and as an audit trail. Each request which is sent through OfficePath generates an entry on the PIPLOG file. This entry is updated by the OfficePath transactions as processing is completed and is thus an indication of how far a request has progressed.

When a request is made to OfficePath, a PIPLOG "M" entry is created. This record is then updated by the OfficePath transactions. The "M" record contains information which is used by the OfficePath transaction to re-START the request following a system or transaction failure. The record remains as an "M" record until all the recipients have been processed and the status has been returned to the sender. At this point the "M" record is deleted and rewritten as an "X" record.

The "X" records can be used as an audit trail to trace through all requests which have been processed by OfficePath. "X" records are retained on the log file for a number of days, specified on the control file SYSTEM record.

LIST and SIGNON requests are written to this file as "M" records however they are deleted once they have been processed unless "Log all requests" is set to "Y".

The size of the PIPLOG file will depend on the number of requests processed by OfficePath (one record per request) each day, on whether OfficePath is configured to log all requests, and on the number of days for which the log records are retained (from SYSTEM record). The file is updated frequently by OfficePath, which inserts and deletes records. As a consequence of this, the file will require re-organization from time to time.

PIPMAIL - Mail Waiting for recipient

This file is keyed on the recipient (2 x 8 bytes), the message number (4 bytes binary), and a sequence number (4 bytes binary).

The file consists of a list of mail waiting to be processed (by a mail/status processor) for a recipient. The file consists of one record per item of mail per recipient. The PIPMAIL entry points to a document on the PIPTEXT file. PIPMAIL entries are created by the PIP2 transaction and are deleted by the mail/status processor when the mail is delivered. The PIPMAIL file is read during startup processing so that all outstanding mail can be re-STARTed.

This file has may have several records created on it as the result of one distribution request. The records are deleted after they have been processed, therefore the maximum size of this file will depend on the maximum number of outstanding mail items. This file will require re-organization from time to time.

Note: If the destination specifies "accumulate recipients", then multiple recipients (up to 50) may be written to one PIPMAIL record.

PIPSTAT - Recipient Status Information

This file is keyed on the originator (2 x 8 bytes), a one byte status flag, the message number (4 bytes binary), and the recipient (2 x 8 bytes).

The file consists of a list of recipients, and their status, for documents which have been distributed. It is used (by the status processors) to generate the COD information for the sender. The file contains one record per recipient per distribution. The records are created by the relevant input processor (PIPSEND, PIP5 or PIP7) and initially contain a status of **ACCEPTED** and the processed flag is set to 'N'. As distribution requests are processed by the transactions, the recipient status information is updated to **DELIVERED**, **INVALID**, **CANCELLED** or **ERROR,** as appropriate and the processed flag is set to 'P'. When the status has been reported to the originator then the processed flag is set to 'Y'.

Records are deleted either when the request is complete or when requests are removed from PIPLOG, depending on the setting on the SYSTEM record. The size of this file thus depends on either the number of incomplete requests or the total number of requests on the log. This file will require re-organization from time to time.

PIPTEXT - Document text file

This file is keyed on the message number (4 bytes binary), the DCA level (2 bytes binary), the GCID (4 binary), and a sequence number (2 bytes binary).

The file contains the DIA Profile and document content of distribution requests. Profile and document records are created by the input processors (PIPSEND, PIP5 or PIP7). If document conversion is performed, the converted document is also stored here. The document text is deleted by the PIP2 transaction once the request is complete.

This file will contain at least one copy of the document in an incomplete distribution request. The records will be deleted when the request is complete (to all recipients). This file will require re-organization from time to time. Note that multiple copies of a document are necessary if the recipients require the document in different formats (RFT, Editable, ...).

In-Trays - Input files

One or more in-trays (ZPMAILSK and/or ZPMAILBX) are defined on the control file. Application programs place all requests onto an in-tray. When requests are processed by the PIP5 transaction they are "moved" from the in-tray to PIPTEXT and PIPSTAT files.

The in-tray will be emptied every time the PIP5 transaction runs. This file will require re-organization from time to time.

Out-Trays - Output files

Each out-tray destination has the name of an out-tray. The same out-tray may be "shared" by several destinations. Records are written to the out-tray by the PIP3 transaction.

It is the responsibility of the Application which processes the out-tray to remove records as it processes them. This file will require re-organization from time to time.

Chapter 11. Sample transaction flows

This chapter gives some examples of the transaction flow which occurs when a document is distributed through OfficePath. Several examples of different types of distribution are given. These can be used as a template to determine where an error has occurred when a document does not arrive, or a confirmation is not returned.

The examples given are:

1. Document sent from Personal Services to another SNADS node.

   Mail sent from OV/MVS to IBM Mail Exchange would go though this processing.

2. Document received from SNADS for a Personal Services user.

   Mail received from OV/400 (or other SNADS node) would go though this processing.

3. Document sent from an Application to a Personal Services user.

   Documents sent by customer written applications would go through this processing.

4. Document sent from Personal Services to a VSAM file.

   Documents sent to CICS Applications would go through this processing. In this example the application processing the out-tray will have been coded to "return Status information".

5. Document sent from Personal Services to a Batch Internal Reader.

   Documents sent to BATCH Applications would go through this processing.

6. Documents sent from a Batch Application to a CICS application.

   Documents sent via the AUTOFAX feature of Application/FAX would go through this processing. In this example the application processing the out-tray will **not** have been coded to "return Status information".

Notes:

- All examples assume distribution to one recipient only.
- Most Personal Services to Personal Services distributions should use the internal distribution facility (in PS/CICS 3.1 and OV/MVS). OfficePath will only be used if the requestor and the recipient -
  1. Have different user defaults set up for the type of document they can receive.
  2. Are in different Personal Services transaction families.

In either case, the document will be translated in accordance with the OfficePath recipient destination record NOT the Personal Services user defaults.

- The processing is similar for documents sent to BATCH VSAM or to CICS Applications. The differences are:
    1. The Batch in-tray is opened/closed and an MVS ENQ is issued.
    2. Batch JCL is submitted instead of a transaction being STARTed.

Send from Personal Services to another SNADS node

Trans	Processing	Key	CMD	Sender	Recipient
DMDB	**User selects the send option in PS**: PIPSEND is invoked from PS with a ***send*** The request is validated and a log entry created. The DMDC transaction is then STARTed to receive the ACK	**Ma**	**SEND**	**PENDING**	
DMDC	**Receive the ACK**: PIPRCVE is invoked from PS. It builds an ACK DIU, which is returned to PS, and updates the log record. PIP2 is STARTed to process the ***send***	Ma	SEND	**ACCEPTED**	**DCA CONV**
PIP2	**Document conversion/routing**: PIP2 validates the recipients using the USER records from the PIPCNTL file. The document is converted, if necessary, using the information from the DESTINATION records. The PIP6 transaction is STARTed on the LU 6.2 link since the recipient is SNADS.	Ma	SEND	ACCEPTED	**IN PROG**
PIP6	**SNADS Mail/Status send processor**: A SNADS distribution DIU is built and the document is sent to the remote node. The log file is updated to indicate that the mail has been forwarded to the remote node, but the recipient status is not changed at this time. The remote	Ma	SEND	ACCEPTED	**COMPLETE**

	node will return a STATUS DIU: this DIU will indicate whether or not the recipient was valid.				
PIP7	**SNADS Mail/Status receive processor**: The SNADS STATUS DIU is received from the remote node. This is used to update the recipient status for the request. PIP2 is then STARTed to return the status	Ma	SEND	ACCEPTED	COMPLETE
PIP2	**Status routing**: PIP2 STARTs transaction PIP1 to process the status	Ma	SEND	**COD NFY**	COMPLETE
PIP1	**PS Mail/Status processor**: START the DMDD transaction	Ma	SEND	COD NFY	COMPLETE
DMDD	**Personal Services**: LINKs to the PIPSEND module with a list command. PIPSEND creates a log record for the list and STARTs DMDC to process the response. ***See note 1***.	Ma **Mb**	SEND **LIST**	COD NFY **PENDING**	COMPLETE
DMDC	**Personal Services**: LINKs to the PIPRCVE module to get the list response. PIPRCVE builds a DELIVER DIU containing the status information in a list document. Both the list and the send are now complete as all information has been returned. PIP2 is STARTed to delete the status records as the request is now complete.	Ma **X**b	SEND LIST	**COMPLETE** **COMPLETE**	COMPLETE
PIP2	**Status**: Deletes the status records and turns the log "M" record into an "X" record.	**X** a	SEND	COMPLETE	COMPLETE

Notes:

1. The LIST request will always be logged, regardless of the parameter on the SYSTEM record on the control file. The system record setting only causes the record to be deleted rather than written as an 'X' record.

Request from SNADS to a Personal Services user

Trans	Processing	Key	CMD	Sender	Recipient
PIP7	**SNADS Mail/Status receive processor:** The mail is received from the remote node. A log record is created and PIP2 is STARTed to process the request	**Ma**	**SEND**	**ACCEPTED**	**DCA CONV**
PIP2	**Document conversion/routing**: PIP2 validates the recipients using the USER records on the PIPCNTL file. The document is convert, if necessary, using the information from the DESTINATION records. The PIP1 transaction is STARTed since the recipient is a PS user.	Ma	SEND	ACCEPTED	**IN PROG**
PIP1	**PS Mail/Status processor**: PIP1 validates the recipient by reading the HUD file and the mail transaction is STARTed. If there is a **PSFAMILY** record for the recipient's address then OfficePath will read each of the HUD dataset, on that record, in turn. Otherwise the HUD from the DEST record is used.	Ma	SEND	ACCEPTED	IN PROG
DMDD	**Personal Services**: PIPSEND is invoked from PS with a ***list*** The request is validated and a log record is created. PIPSEND then STARTs DMDC. See note	Ma **Mb**	SEND **LIST**	ACCEPTED **PENDING**	IN PROG
DMDC	**Personal Services**: PIPRCVE is invoked from PS. It builds a ***list document*** is returned to PS. Since the ***list document*** indicates that new mail has arrived, PS LINKs to PIPSEND with an OBTAIN DIU command to get the mail. DMDF is STARTed.	Ma **Mc** Xb	SEND **OBTAIN** LIST	ACCEPTED **PENDING** **COMPLETE**	IN PROG
DMDF	**Personal Services**: PIPRCVE is invoked from PS. PIPRCVE builds	Ma Xc	SEND **OBTAIN**	**COD NFY** **COMPLETE**	COMPLETE

	a DELIVER DIU which contains the mail item. The mail is now ***delivered*** and PIP2 is STARTed to return the status.				
PIP2	**Status routing**: PIP2 STARTs the PIP6 transaction to process the status.	Ma	SEND	**COD NFY**	COMPLETE
PIP6	**SNADS Mail/Status send processor**: A SNADS ACK DIU is built to return the status to the SNADS node. The log record is updated to complete. PIP2 is STARTed to delete the status records as the request is now complete.	Ma	SEND	**COMPLETE**	**COMPLETE**
PIP2	**Status**: Deletes the status records and turns the log "M" record into an "X" record.	**X**a	SEND	COMPLETE	

Notes:

1. The LIST request will always be logged, regardless of the parameter on the SYSTEM record on the control file. The system record setting only causes the record to be deleted rather than written as an 'X' record.

Send from Personal Services user to a VSAM file

Trans	Processing	Key	CMD	sender	recipient
DMDB	**User selects the send option in PS**: PIPSEND is invoked from PS with a ***send*** The request is validated and a log record is created. DMDC is STARTed.	**Ma**	**SEND**	**PENDING**	
DMDC	**Personal Services**: PIPRCVE is invoked from PS. PIPRCVE builds an ACK DIU, updates the log record and STARTs the PIP2 transaction.	Ma	SEND	**ACCEPTED**	**DCA CONV**
PIP2	**Document conversion/routing**: PIP2 validates the recipients using	Ma	SEND	**ACCEPTED**	**IN PROG**

	the USER records from the PIPCNTL file. The document is converted, if necessary, using the information from the DESTINATION records. The PIP3 transaction is STARTed since the recipient is a VSAM outtray.				
PIP3	**VSAM Out-tray Mail/Status processor**: The document is copied onto the out-tray and application mail transaction, from the DESTINATION record, is STARTed. ***See notes 1 and 2.***	Ma	SEND	ACCEPTED	**COMPLETE**
XXXX	**Application**: Read/Delete the request using the CICS (or Batch) interface program. The application validates the recipient/request and generates an ACK response. This is written to the intray and, optionally, PIP5 is STARTed. ***See notes 1 and 3.***	Ma	SEND	ACCEPTED	COMPLETE
PIP5	**Application Intray processor**: Read the ACK request from the intray. Update the log record and START PIP2 to return status. ***See note 1.***	Ma	SEND	**COD NFY**	COMPLETE
PIP2	**Status routing**: PIP2 STARTs the PIP1 transaction to process the status.	Ma	SEND	**COD NFY**	COMPLETE
PIP1	**PS Mail/Status processor**: The DMDD transaction is STARTed to notify PS of the updated status.	Ma	SEND	**COD NFY**	COMPLETE
DMDD	**Personal Services**: PIPSEND in invoked by PS with a ***list*** The request is validated and a log record is created. The DMDC transaction is STARTed. See note	Ma **Mb**	SEND **LIST**	COD NFY **PENDING**	COMPLETE
DMDC	**Personal Services**: PIPRCVE is invoked by PS. PIPRCVE builds a ***list document*** which contains the updated status information. PIP2 is STARTed to delete the status records	Ma **X**b	SEND **LIST**	**COMPLETE** **COMPLETE**	COMPLETE

	as the request is now complete.				
PIP2	**Status**: Deletes the status records and turns the log "M" record into an "X" record.	**X** a	SEND	COMPLETE	

Notes:

1. If the application cannot return the status then OfficePath can be instructed to update the status when the request is written to the out-tray. In this case the PIP3 transaction will set the status to COD NFY (instead of ACCEPTED) and the PIP2 transaction will then be STARTed. The XXXX and PIP5 steps will only occur if the destination record specifies that the application will return a status.
2. If the application runs in batch, then the PIP8 transaction is STARTed. This transaction submits JCL which can be used to invoke the application.
3. The application response to a distribution (that is ACK) can be postponed until the mail has reached its final destination. For example, if the application is a messaging-system the application could postpone the ACK until the distribution has reached the messaging-system user's in-basket.
4. The LIST request will always be logged, regardless of the parameter on the SYSTEM record on the control file. The system record setting only causes the record to be deleted rather than written as an 'X' record.

Request from an Application to Personal Services

Trans	Processing	Key	CMD	sender	recipient
XXXX	**Application**: Write a ***send*** request to the intray. Optionally, START PIP5.				
PIP5	**Application In-tray processor**: The document is moved from the intray onto the PIPTEXT file. A log file record is created and each of the recipients is written to the PIPSTAT file. The PIP2 transaction is STARTed to process the ***send See note 1.***	**Ma**	**SEND**	**ACCEPTED**	**DCA CONV**
PIP2	**Document conversion/routing**: PIP2 validates the recipients using the USER records on the	Ma	SEND	ACCEPTED	**IN PROG**

	PIPCNTL file. The document is converted, if necessary, using the information from the DESTINATION records. The PIP1 transaction is STARTed since the recipient is a PS user.				
PIP1	**PS Mail/Status processor** PIP1 validates the recipient by reading the HUD file. If the recipient is valid then the mail transaction is started otherwise a status message is returned. ***See note 2.***	Ma	SEND	ACCEPTED	IN PROG
DMDD	**Personal Services**: PIPSEND in invoked from PS with a ***list*** The request is validated and a log record is created. PIPSEND then STARTs DMDC. ***See note 4.***	Ma **Mb**	SEND **LIST**	ACCEPTED **PENDING**	IN PROG
DMDC	**Personal Services**: PIPRCVE is invoked from PS. It builds a ***list document*** is returned to PS. Since the ***list document*** indicates that new mail has arrived, PS LINKs to PIPSEND with an OBTAIN DIU command to get the mail. DMDF is STARTed.	Ma **Mc** **Xb**	SEND **OBTAIN** LIST	ACCEPTED **PENDING** **COMPLETE**	IN PROG
DMDF	**Personal Services**: PIPRCVE is invoked from PS. PIPRCVE builds a DELIVER DIU which contains the mail item. The mail is now ***delivered*** and PIP2 is STARTed to return the status.	Ma **Xc**	SEND **OBTAIN**	**COD NFY** **COMPLETE**	**COMPLETE**
PIP2	**Status routing**: PIP2 STARTs transaction PIP1 to process the status	Ma	SEND	**COD NFY**	COMPLETE
PIP3	**VSAM Out-tray Mail/Status processor** An ACK document is written to the outtray. PIP2 is STARTed to delete the status records as the request is now complete. The application transaction is STARTed. ***See note 4.***	Ma	SEND	**COMPLETE**	
PIP2	**Status**: Deletes the status records and turns the log "M" record into	**X** a	SEND	COMPLETE	

	an "X" record.				
XXXX	**Application**: Read/Delete the outtray using the CICS/Batch interface program.				

Notes:

1. CICS applications can START the PIP5 transaction after writing a request to the in-tray (this is documented in the *OfficePath Programming Interfaces* manual) or wait until the INPUT timer expires.
2. If there is a **PSFAMILY** record for the recipient's address then OfficePath will read each of the HUD dataset, on that record, in turn. Otherwise the HUD from the DEST record is used.
3. The LIST request will always be logged, regardless of the parameter on the SYSTEM record on the control file. The system record setting only causes the record to be deleted rather than written as an 'X' record.
4. If the application runs in batch, then the PIP8 transaction is STARTed by PIP3. The PIP8 transaction submits JCL which can be used to invoke the application which processes the COD.

Send from Personal Services to an Internal Reader

Trans	Processing	Key	CMD	sender	recipient
DMDB	**User selects the send option in PS**: PIPSEND is invoked from PS with a ***send*** The request is validated and a log entry is created. The DMDC transaction is then STARTed.	**Ma**	**SEND**	**PENDING**	
DMDC	**Personal services**: PIPRCVE is invoked from PS. PIPRCVE builds an ACK DIU, updates the log record and STARTs the PIP2 transaction.	Ma	SEND	**ACCEPTED**	**DCA CONV**
PIP2	**Document conversion/routing**: PIP2 validates the recipients using the USER records from the PIPCNTL file. The document is converted, if necessary, using the information from the DESTINATION records. The PIP4 transaction is STARTed since	Ma	SEND	ACCEPTED	**IN PROG**

	the recipient is an Internal reader.				
PIP4	**Internal RDR Mail/Status processor**: The document is merged with JCL from the PIPCNTL file. It is then submitted to the INTRDR. ***See note 1.***	Ma	SEND	ACCEPTED	**COMPLETE**
	BATCH APPLICATION: Process the document included as instream data. The application generates an ACK response which is written to the intray. ***See notes 1 and 2.***	Ma	SEND	ACCEPTED	COMPLETE
PIP5	**Application In-tray processor**: Read the ACK request from the intray. Update the log record and START PIP2 to return status.	Ma	SEND	**COD NFY**	COMPLETE
PIP2	**Status Routing** PIP2 STARTs the PIP1 transaction to process the status.	Ma	SEND	**COD NFY**	COMPLETE
PIP1	**PS Mail/Status processor** The DMDD transaction is STARTed to notify PS of the updated status.	Ma	SEND	**COD NFY**	COMPLETE
DMDD	**Personal Services**: PIPSEND is invoked by PS with a ***list*** The request is validated and a log record is created. The DMDC transaction is STARTed. See note	Ma **Mb**	SEND **LIST**	COD NFY **PENDING**	COMPLETE
DMDC	**Personal Services**: PIPRCVE is invoked from PS. PIPRCVE builds a ***list document*** which contains the updated status information. The send request is now complete. PIP2 is STARTed to delete the status records as the request is now complete.	Ma **X**b	SEND LIST	**COMPLETE** **COMPLETE**	COMPLETE
PIP2	**Status**: Deletes the status records and turns the log "M" record into an "X" record.	**X**a	SEND	COMPLETE	

Notes:

1. If the application cannot return the status then OfficePath can be instructed to update the status when the request is written to the internal reader. In this case the PIP4 transaction will set the status to COD NFY (instead of ACCEPTED) and the PIP2 transaction will

then be STARTed. The batch application and the PIP5 steps will only occur if the destination record specifies that the application will return a status.

2. The application response to a distribution (that is ACK) can be postponed until the mail has reached it's final destination.
3. The LIST request will always be logged, regardless of the parameter on the SYSTEM record on the control file. The system record setting only causes the record to be deleted rather than written as an 'X' record.

Request from an Batch Application to a CICS Application

Trans	Processing	Key	CMD	sender	recipient
	Application: Write a ***send*** request to the intray.				
PIP5	**Application In-tray processor**: The document is moved from the intray onto the PIPTEXT file. A log file record is created and each of the recipients is written to the PIPSTAT file. The PIP2 transaction is STARTed to process the ***send***	**Ma**	**SEND**	**ACCEPTED**	**DCA CONV**
PIP2	**Document conversion/routing**: PIP2 validates the recipients using the USER records on the PIPCNTL file. The document is converted, if necessary, using the information from the DESTINATION records. The PIP3 transaction is STARTed since the recipient is a VSAM out-tray.	Ma	SEND	ACCEPTED	**IN PROG**
PIP3	**VSAM Out-tray Mail/Status processor**: The document is copied onto the out-tray and application mail transaction, from the DESTINATION record, is STARTed. PIP2 will be started to process the status. ***See note 1***	Ma	SEND	**COD NFY**	**COMPLETE**
XXXX	**Application**: Read/Delete the request using the CICS interface program.				

PIP2	**Status routing**: PIP2 STARTs transaction PIP4 to process the status	Ma	SEND	**COD NFY**	COMPLETE
PIP4	**Internal RDR Mail/Status processor** An ACK document is merged with JCL from the PIPCNTL file. It is then submitted to the internal reader. PIP2 is STARTed to delete the status records as the request is now complete. ***See note 2***	Ma	SEND	**COMPLETE**	
PIP2	**Status**: Deletes the status records and turns the log "M" record into an "X" record.	**X**a	SEND	COMPLETE	
	Application: The JCL executes to process the ACK.				

Notes:

1. More advanced applications can be coded to "return status information" in this case the sender status will remain as "ACCEPTED" until the status is returned by the application.
2. The application response to a distribution (that is ACK) can be postponed until the mail has reached its final destination.

PART THREE - ERROR MESSAGES

Chapter 12. CICS Background transactions

The background OfficePath transactions write any error messages to CSMT. In addition, if the error is serious, the transaction will abend with ABEND code Annn (where nnn is the message number).

In the event of an error in the PIPSEND or PIPRCVE modules no abend is produced. The program takes a transaction dump and then returns to the calling module with an error code.

API return codes

PIPSEND/PIPRCVE are modules invoked by DISOSS API applications (for example OV/MVS, CA-eMAIL). If an error occurs in these modules no abend is produced but the program takes a transaction dump (code Dnnn) and then returns an error code to the calling application. The following error codes are used:

0 No error

3 End of data

15 I/O Error - The API application should invoke a ROLLBACK

24 SYSID Error - OfficePath is not available (for example a Remote System is down or the file PIPCNTL cannot be found)

Messages written to CSMT

Structure of the Messages

Note: All messages have an identification containing the transaction issuing the message, a four digit number and a severity indicator. For reference purposes all messages are shown as if issued by transaction "PIPS".

OfficePath writes the error messages to the CICS log (transient data queue CSMT) with a length of 80 bytes. Some messages may be longer than this and the character "\" is used to split one line into two. Various "placemarkers" are used to place variable data into the message text: these are detailed below.

@C The CICS response (such as NOTFND or NOTOPEN)

@D Message dependent data (variable length)

@X Message dependent data (variable length) displayed as hexadecimal characters

@F The EIBFN (for example READ or START) corresponding to the CICS command in error.

@R the EIBRSRCE (usually a file name) corresponding to the CICS command in error.

@T The transaction where the error was detected.

@M The OfficePath message number, in HEX, of the request being processed.

@Z A return code issued by an invoked program.

@S The current date and time.

The "\"character is be used to split a line when writing to the CICS log.

Usually there should be no requirement to change the text of any of the error messages. Placemarkers should not be changed, inserted, or removed. However, you may choose to insert the @S placemarker for some of the messages.

A number of the messages are displayed for information only; under some circumstances a large number of these may be generated. If you do not want these messages they can be deleted. Such Messages are:

PIPS047I Mail sent to printer @D CANCELLED - Request @M.

PIPS048I Cancelled user @D, change causes duplicate. Request @M.

PIPS049I User @U has a node of @D in the HUD. Request @M.

PIPS050W No GCID table found. No translation performed.

PIPS051W No COPYLIST record found. No expansion performed.

PIPS052W No DEST record for @D: Confirmation ignored. Request @M.

PIPS053W No USER record for @D: Receipt Ack ignored. Request @M.

PIPS054W Duplicate user @D ignored. Request @M.

PIPS056I @S Document size exceeded.\Request @M originating from @U.

PIPS057I @S HOP count exceeded.\Request @M originating from @U.

PIPS058W Unexpected status received from recipient @D.\Request @M.

PIPS059I No conversion for DCA Level @Z for recipient @D.\Request @M.

PIPS113W No STATUS records for request @M, request will be cancelled

PIPS114W No TEXT records for request @M, request will be cancelled

PIPS115W @S Another copy of the startup program is already running

PIPS116I @S Startup request for OfficePath. Version @D

PIPS117I @S Startup successful. OfficePath expires in @D days.

PIPS120I Vital @D record added to @R.

PIPS202W Restart error: @C during EXEC CICS @F @R @X\Request @M.

PIPS251I SNADS send error detected. Data @X \ Request @M requeued.

PIPS301I DIU field @X not recognized. Field ignored.

PIPS302I PIPAPIV module not found. User @D assumed valid.

Description of the Messages

The meaning of each message is as follows:

PIPS001I Control file not available.:

Explanation: The control file is closed or the remote system is not available.

System action: Return code 24 is passed back to the API application.

User response: Open the control file or ensure remote system is available.

PIPS003E Unexpected end-of-data processing request @M.:

Explanation: The DIU parser reached the end of the DIU before the DIU Suffix was found. The DIU that was sent may be corrupt or incomplete.

System action: The request is terminated with a dump.

User response:

- Check the message on the PIPLOG file to determine which request failed.
- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS012E Unrecognised command @D issued. Request @M.:

Explanation: The Personal Services transaction has issued an API command which is invalid. This may indicate an error in the Personal Services transaction or a storage violation.

System action: The request is terminated with a dump.

User response:

- Check the CICS log for any storage violation messages.
- Check the message on the PIPLOG file to determine the request which failed.
- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS047I Mail sent to printer @D CANCELLED - Request @M.:

Explanation: A recipient in a distribution had the same location at the deafult print location, as specified on the SYSTEM record. It is not permitted to **send**mail to printers.

System action: The status for the "changed" recipient is set to CANCEL.

User response: None.

PIPS048I Cancelled user @D, change causes duplicate. Request @M.:

Explanation: A recipient has been changed, either via the user validation exit program (PIPEXIT1) or via the redirector feature. The new user matches one of the other recipients in the request.

System action: The status for the "changed" recipient is set to CANCEL.

User response: None.

PIPS049W User @U has a node of @D in the HUD. Request @M.:

Explanation: Personal Services user ids are validated by reading the Personal Services HUD data set. OfficePath found a record for the user id, however the node (location) did not match. This may be because the user has changed his location and not signed onto OfficePath.

System action: Processing continues, the request will not arrive until the user signs on again.

User response: Get the user to sign on again.

PIPS050W No GCID table found. No translation performed.:

Explanation: A request required a GCID which could not be found on the control file. The request could be an attempt to write to an out-tray, or to read from an in-tray. This means that the characters in the document profile/text may not be interpreted correctly.

System action: Processing continues, the document is assumed to be in multi-lingual GCID and no translation will be performed.

User response:

- If the problem was on an in-tray, identify the application which sent the request. If the GCID supplied was incorrect, correct it.
- If the problem was on an out-tray, identify the GCID specified for the out-tray. If the GCID supplied was incorrect, correct it.

- If the GCID supplied was correct then you will need to add a new GCID translation table. Refer to the *OfficePath Installation and Administration* manual for details on how to do this.

PIPS051W No COPYLIST record found. No expansion performed.:

Explanation: An application specified a COPYLIST which was not found on the control file.

System action: Processing continues. The COPYLIST is treated as a single recipient.

User response:

- Identify the application which created the request.
- Correct the application or add the COPYLIST.

PIPS052W No DEST record for @D: Confirmation ignored. Request @M.:

Explanation: OfficePath either cannot find a DEST record @D or the DEST record is not suitable for this request.

If the request has come from another SNADS node then OfficePath will look for a reply DEST record using the REN of the originating system. In this case the DEST record must be of type SNADS.

If the request has come from the CICS or Batch application interface then the DEST record name is taken from the "TO SYS" option of the REPLY parameter. In this case the DEST record must be of type OUTTRAY or INTRDR.

This message will also be produced by the installation verification transaction PIPT. In this case the DEST will be PIPTEST.

System action: Processing of the request continues, but no confirmation will be generated.

User response:

- Identify the application which produced the request.
- Correct the application or add the destination to the control file.

If the message is produced by the PIPT transaction, then you should ignore the message. **Do not create a DEST record for PIPTEST unless explicitly requested to do so by TBS Software Ltd support personnel.**

PIPS053W No USER record for @D: Receipt Ack ignored. Request @M.:

Explanation: A request, originating from either the Batch or CICS interface program, which specified receipt acknowledgment was processed by OfficePath. The userid @D specified as the sender either was not defined on the control file or the DEST record is neither an OUTTRAY or INTRDR type.

System action: Processing of the request continues, but receipt acknowledgment is suppressed.

User response:

- Identify the application which created the request.
- Correct the application or correct/insert USER and DEST records.

PIPS054W Duplicate user @D ignored. Request @M.:

Explanation: A request, originating from SNADS, has the same recipient specified twice. Most Office systems eliminate duplicate recipients before the request is sent, but in this case the duplicate has not been removed.

System action: The duplicate recipient is ignored. The document will not be delivered twice nor will two acknowledgments for that recipient be generated.

User response:

- Identify the application which created the request.

PIPS056I @S Document size exceeded.\Request @M originating from @U.:

Explanation: The size of a document is greater than the maximum permitted by this OfficePath system.

System action: The request is rejected with an exception code of C3111B (document size exceeded).

User response:

- Either increase the maximum document size or tell the originator to stop sending such large documents.

PIPS057I @S HOP count exceeded.\Request @M originating from @U.:

Explanation: The HOP count has been exceeded: the request has gone through too many SNADS nodes.

System action: The request is reject with a status of ERROR.

User response:

- Investigate this as it might be caused by a requested LOOPING between a number of nodes.

PIPS058W Unexpected status received from recipient @D.\Request @M.:

Explanation: An acknowledgement, originating from SNADS, has status information for a recipient who was not part of the original distribution, or for whom status information has already been received.

System action: The information is ignored.

User response:

- This may indicate that an application or node is confirming recipients multiple times. If the problem occurs repeatedly then check the recipients in an attempt to locate the source problem.

PIPS059I No conversion for DCA Level @Z for recipient @D.\Request @M:

Explanation: DCA level conversion is necessary if the recipient user cannot receive mail with the DCA level of the original document. The conversion parameters are held on the /FMT records on the OfficePath control file. This message indicates that no suitable conversion could take place.

System action: The recipient status is set to CANCELLED.

User response:

- If you are expecting the conversion to take place check that the /FMT records have been correctly created.

PIPS060E No TSQ records found. Request @M.:

Explanation: An internal logic error has occurred. The OfficePath modules communicate using TS Queues. A module could not find the TSQ.

System action: Processing of the request is terminated with a dump.

User response:

- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS061E DIU Prefix missing from command. Request @M.:

Explanation: The DIU parser did not find the DIU Prefix. The DIU that was sent may be corrupt or in-complete.

System action: The request is terminated with a dump.

User response:

- Check the message on the PIPLOG file to determine the request which failed.
- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS062E DIU command unit missing/invalid. Request @M.:

Explanation: The DIU parser did not recognize the command unit in a DIU. The DIU that was sent may be corrupt or incomplete.

System action: The request is terminated with a dump.

User response:

- Check the message on the PIPLOG file to determine the request which failed.
- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS063E DIU document unit missing. Request @M.:

Explanation: The DIU parser did not find the DIU document unit. The DIU that was sent may be corrupt or incomplete.

System action: The request is terminated with a dump.

User response:

- Check the message on the PIPLOG file to determine the request which failed.
- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS064E DIU suffix missing (or not at end). Request @M.:

Explanation: The DIU parser did not find the DIU suffix, or the suffix was not last in the DIU. The DIU that was sent may be corrupt or incomplete.

System action: The request is terminated with a dump.

User response:

- Check the message on the PIPLOG file to determine the request which failed.
- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS087E Invalid START of OfficePath\START data - @X.:

Explanation: The OfficePath transaction was started with data which did not match the expected START data.

System action: Program terminates with an ABEND.

User response:

- Determine how the transaction was STARTed.
- If the transaction was STARTed by an application program, correct the program to pass the correct START data.

Note: Application transactions should only START PIP5. No other OfficePath transactions should be STARTed. Details of the START data which must be used can be found in the *OfficePath Programming Interfaces* manual.

PIPS100E No text records found for request @M.:

Explanation: The PIP2 transaction has detected a NOTFND condition when trying to read records on the PIPTEXT file. The records should be present.

System action: Processing of the request is terminated with a dump.

User response:

- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS101E No status information to process\ Request @M.:

Explanation: One of the mail/status processors, or the PIP2 transaction, was STARTed to process status information. When the transaction executed no status information was found for the request. This indicates that an internal logic error has occurred.

System action: Processing of the request is terminated with a dump.

User response:

- Check the request on the PIPLOG file.
- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS113W No STATUS records for request @M, request will be cancelled:

Explanation: During the start-up processing a request was found on the log file (PIPLOG) without the required corresponding status records (PIPSTAT). This message can only occur if OfficePath is started with the " FILES" option, see "Crosschecking OfficePath files".

System action: The "M" record is converted to an "X" record and processing continues.

User response: Check for damaged files: perhaps a VSAM re-organization has failed.

PIPS114W No TEXT records for request @M, request will be cancelled:

Explanation: During the start-up processing a request was found on the log file (PIPLOG) without the required corresponding text records (PIPTEXT). This message can only occur if OfficePath is started with the " FILES" option, see "Crosschecking OfficePath files".

System action: The "M" record is converted to an "X" record and processing continues.

User response: Check for damaged files: perhaps a VSAM re-organization has failed.

PIPS115W @S Another copy of the startup program is already running:

Explanation: There are two copies of the start-up program (PIPS010) active; the res-count is greater that one.

System action: Only one copy of the program will execute at any one time, the other(s) will terminate.

User response: Check that a copy of the start-up program has not become "stuck". Make sure that the PIPS010 program is **NOT** defined as RESIDENT in the PPT.

PIPS116I Startup request for OfficePath. Version @D:

Explanation: You have typed in the PIPS transaction and OfficePath has started (or re-started), or the PIPSPLT program has started OfficePath when the CICS system was loaded. This message appears on the CICS system log and on the terminal when transaction PIPS is invoked from a terminal.

System action: Informational message. OfficePath initialization has completed and PIP5 transaction(s) have been STARTed to process the input files.

User response:

- Relax, all is well.

PIPS117W Startup successful. OfficePath expires in @D days.:

Explanation: You have OfficePath on a trial period. This trial will end in @D days because the software will stop working.

System action: Warning message. System functions will continue as normal (for @D days, anyway!).

User response:

- Inform your management that they need to decide whether they are going to keep this product or not.
- Carry on with your normal activities.

PIPS118E OfficePath license has expired.:

Explanation: You have ignored the PIP117W message and as a result your software has stopped working. No distribution requests will be processed. However, users will be able to sign-on to Personal Services as normal.

System action: Error message. All distribution functions will cease working. Sign-on will be unaffected.

User response:

- Either, contact TBS Software Ltd for a new license number and run the PIPCTL program using the new number.
- Or, remove the OfficePath software from your machine.

PIPS119E Vital @D record missing from @R.:

Explanation: OfficePath requires that the files PIPTEXT, PIPMAIL, PIPSTAT, and PIPLOG contain a header and a trailer record. This message is produced during the start-up processing for OfficePath.

System action: The start-up transaction, PIPS, will abend.

User response:

- Find out why the header/trailer record is missing and correct the problem, then re-run the PIPS transaction.
- If you are happy that no data has been lost, crosscheck the files as detailed in "Crosschecking OfficePath files".

PIPS120I Vital @D record added to @R.:

Explanation: The PIPS transaction started with the **FILES** option and detected that a header/trailer record was missing from one of the files.

System action: The missing record has been inserted.

User response: None: you have selected the **FILES** option to repair the file structure.

PIPS121E Invalid key length of @D for file @R.:

Explanation: The PIPS transaction validates certain characteristics of the files: this message indicates that the key length is incorrect.

System action: The start-up transaction, PIPS, will abend.

User response: Check and correct the definition of the file.

PIPS122E Invalid record length of @D for file @R.:

Explanation: The PIPS transaction validates certain characteristics of the files: this message indicates that the record length is incorrect.

System action: The start-up transaction, PIPS, will abend.

User response: Check and correct the definition of the file.

PIPS126S I/O Error on file @R. VSAM return codes @X.:

Explanation: An error was returned by CICS when OfficePath was attempting to OPEN or CLOSE a batch in-tray or out-tray. The file was @R and the VSAM return code @X contains the following fields (displayed in hex):

REG15	CL2	Value from register
ERROR	CL2	Field from VSAM ACB

Common return codes are:

000800A8 Open failed. Dataset not available for type of processing requested.

This return code means that another job is accessing the dataset.

System action: Processing of the request is terminated with a dump. If the request is for an INPUT file, no further processing of that file will occur.

User response: Access to the Batch in-trays and out-trays is serialized by the MVS ENQ command. This return code indicates that a job has run which is not using the ENQ or the ENQ is not being transmitted from another machine.

- Check on the system log to find out if any batch jobs were running at the time the error occurred. Check that these jobs are using the correct version of the OfficePath batch program.
- Check that the GRS Resource Name List (or equivalent) has the correct entries for OfficePath. This is described in the *OfficePath Installation and Administration* manual.
- Check that batch jobs running on other machines are using the correct version of the OfficePath batch program.

Note: Running the IDCAMS utility program, or a user written program, against the in-tray/out-tray will produce this message since they will open the dataset without issuing an ENQ. Access to the in-tray/out-tray must only be from the Batch interface modules described in the *OfficePath Programming Interfaces* manual.

PIPS127S Error @D returned by PIPSOUT. Contact TBS Software Ltd.:

Explanation: The PIPSOUT module, which is used internally by OfficePath, returned an error code. This indicates an internal logic error.

System action: Processing of the request is terminated with a dump.

User response:

- Print the Dump. Contact TBS Software Ltd.

PIPS128S DYNALLOC error @X on file @R. Request @M.:

Explanation: Whenever OfficePath attempts to access file @R which is defined on its control file as a batch file (the "Outtray processed in Batch" flag is set), it issues an MVS ENQ to serialize access to the file. OfficePath uses the DYNALLOC routine to get the DSNAME of the file, which is used in the ENQ. The DYNALLOC routine returned an error code @X (in hex). This error may occur when OfficePath is accessing an INPUT file or an OUTTRAY file.

Common return codes are:

0438 DD Name not found.

DYNALLOC could not find a file allocated to the DD Name @R.

System action: Processing of the request is terminated with a dump. If the request is for an INPUT file, no further processing of that file will occur.

User response:

- Check that the DSNAME parameter is specified in the FCT definition for this file or that the file is allocated in the CICS JCL. The DSNAME option causes the file to be dynamically allocated/de-allocated when it is opened/closed.
- If the request number is not zero then the problem has occurred for an OUTTTRAY. Check the request on the log file and determine which recipient caused the error. Correct the error and restart OfficePath.
- If the request number is zero then the problem has occurred on an input file. Check the status of the INPUT records on the log file. The INPUT record whose LAST UPDATED field has expired is the file which had the error.
- Correct the error and restart OfficePath.

PIPS129S Error returned from CEMT: @D.:

Explanation: The OfficePath Remote Link Monitor, part of Control file maintenance, uses CEMT to acquire/release remote links and to put them in/out of service. Error @D has occurred using CEMT.

System action: The remote link is not actioned.

User response:

- Check the remote link exists.
- Perform the action using CEMT directly.

PIPS200S @C during EXEC CICS @F @R @X\ Request @M.:

Explanation: CICS returned @C in response to the EXEC CICS @F command on resource @R (If the resource is a file then @X, is the key used).

System action: Processing of the request is terminated with a dump.

User response:

- Correct the error and restart OfficePath.

PIPS201E Request @M purged by Personal Services.:

Explanation: Personal Services rejected a DIU sent to it by OfficePath.

System action: The request is deleted from the log file. Personal Services will produce a dump and ignore the DIU.

User response:

- Check the request on the log-file.
- Print the Dump. Contact TBS Software Ltd.

PIPS202W Restart error: @C during EXEC CICS @F @R @X\Request @M.:

Explanation: The OfficePath startup transaction was attempting to continue the processing of an outstanding request. A CICS error was issued during the processing.

System action: The startup transaction continues processing the next request. The request which caused the problem will not be restarted.

User response: Investigate why the request could not be processed. If the request is invalid (for example, a SNADS connection has been removed) then it can be deleted from the PIPLOG file. "Purging invalid or corrupt requests from PIPLOG" describes how to do this.

PIPS203E Formatting error; Retcode @Z - @D\ Request @M.:

Explanation: One of the formatter programs, which are used for document conversion/printing, issued a return code @Z. The text (@D) may contain more information regarding the error.

System action: The request is not processed.

User response: The most common formatting error message is DKL7503 which is returned by DisplayWrite/370 when it cannot find an LDE entry. DisplayWrite/370 requires LDE entries in order to convert or print documents. Check that you have defined the necessary LDE entries in the PIPCLIST clist.

PIPS204E Return code @Z from PIPGT3 program\ Request @M.:

Explanation: This message is used by the OfficePath DisplayWrite/370 print feature.

System action: The request is not processed. Any other OfficePath DisplayWrite/370 print requests will be queued.

User response: This message indicates an internal logic error:

- Check the request on the log-file.
- Print the Dump.
- Contact TBS Software Ltd.

PIPS205E Control @R record @D missing for request @M.:

Explanation: A required record from the control - PIPCNTL - file is missing; @R is the record type (SYSTEM, DEST, USER, JCL,...) and @D is the name of the missing record.

System action: Processing of the request is terminated with a dump.

User response:

- Correct the error and restart OfficePath.

PIPS250E Error during SNADS processing\Data @X.:

Explanation: An error was returned by one of the EXEC CICS GDS commands used during SNADS distributions. The @X field contains the following information (displayed in hex):

Field	**Picture**	**Description**
RETCODE	CL6	Equivalent to CICS EIBRCODE field
FLAGS1	CL7	Flags identifying conversation state
CDBERRCD	CL4	LU 6.2 Error code
FLAGS2	CL1	More flags
RESERVED	CL12	Not used

The possible values for the RETCODE and CDBERRCD fields are documented in the *CICS Intercommunications Facilities Guide*. Common values for CDBERRCD are:

A000 ***Remote transaction prematurely terminated***: the remote SNADS transaction has ABENDed.

084C ***Specified PROCESS is not available***: the remote SNADS transaction is not defined or is disabled.

0864 ***An ISSUE ABEND has been received***: the remote SNADS transaction instructed OfficePath to ABEND.

0889 ***An ISSUE ERROR has been received***: the remote SNADS transaction rejected the request which was sent.

System action: Processing of the request is terminated with a dump.

User response: If the remote transaction has abended then the DIU which is being sent may be invalid. Otherwise the error code indicates that the link is not defined correctly in one or both systems.

- Check the log on the remote node for any error messages.
- Check the definitions for the link.
- Correct the error and restart OfficePath.

PIPS251I SNADS send error detected. Data @X \ Request @M requeued.:

Explanation: The OfficePath SNADS processor was abended due to a read time-out while trying to send a request to another SNADS device. This situation can occasionally occur it both SNADS devices attempt to send mail at exactly the same time.

System action: The request is requeued. The SNADS transaction will attempt to process it again after the interval specified on the SYSTEM record has expired.

User response: No action is usually required. If this message repeats many times for a particular request, it may indicate that the receiving system is unable to process the request and so the request must be cancelled.

PIPS300E Length error in DIU field @X.:

Explanation: The OfficePath DIU parser found a DIU field which had an invalid length. The DIU field which was in error is @X.

System action: Processing of the request is terminated with a dump.

User response:

- Print the dump and contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

PIPS301I DIU field @X not recognised. Field ignored.:

Explanation: The OfficePath DIU parser found a DIU field which it does not recognize. IBM may have added a new DIU field to the architecture. This should not cause a problem since the field length can be determined and so the parser can ignore the field.

System action: Processing continues with the next field in the DIU.

User response:

- Contact, and warn, TBS Software Ltd.

PIPS302W PIPAPIV module not found. User @D assumed valid.:

Explanation: Mail (or status) information has been sent to a DISOSS API type application. The DEST record indicates that the user exit program should be used to validate the recipient @D. OfficePath could not LINK to the exit program and so has assumed that the userid is valid.

System action: The mail/status transaction is notified that mail has arrived. It is the API's responsibility to continue processing the request if the userid is valid.

User response: For test/evaluation purposes the PIPAPIV program does not have to be defined. The only restriction is that you will never have a recipient status of INVALID returned (even if the recipient is not defined to the API application). Any mail sent to an "invalid" recipient will simply accumulate on the OfficePath files since the application will never "obtain" the mail. OfficePath will not delete the request because it is not delivered. For a production environment you should ensure that you have created a PIPAPIV module. Instructions for doing this are in "Chapter 34." of the *OfficePath Installation and Administration* guide.

Transaction ABEND/DUMP codes

When one of the OfficePath transactions encounters a serious error it will produce a transaction dump. With the exception of the PIPSEND and PIPRCVE modules the transaction dump is produced by the transaction issuing an EXEC CICS ABEND. The PIPSEND and PIPRCVE transactions will produce a dump using EXEC CICS DUMP and return an error code to Personal Services.

The dump code will be Axxx for a transaction ABEND or Dxxx for a DUMP.

The following codes may be produced:

UMSG Unable to LINK to OfficePath Error message module.

Check PPT entry for PIPMSG module.

UMCA Message module commarea incorrect.

Internal logic error in program. Contact TBS Software Ltd.

UTCA Trace module commarea incorrect.

Internal logic error in program. Contact TBS Software Ltd.

Annn nnn corresponds to an error message number.

The error message text can be found on CSMT. The meaning of the error messages is described in "Messages written to CSMT"

Note: In this case, the abending program will be PIPMSG (the program that writes the message text to CSMT), you should take particular note of the abending transaction when diagnosing the problem.

Appendix

Appendix A. Batch Application Interface Return Codes

This section lists the returns codes which can be issued by the OfficePath Batch application interface feature. Use of these programs is described in the *OfficePath Programming Interfaces*

Return codes from input processor

The input module can be invoked from batch JCL or from a TSO CLIST. If it is invoked from batch JCL then a return code (and when appropriate, an error message) are given, otherwise only the return code is given.

Use of this module is described fully in the *OfficePath Programming Interfaces* manual.

The possible return codes are:

0000 Processing completed successfully

0099 Invalid PARM value

The PARM= statement does not contain one of the valid values for this program.

0100 Unable to access LISTDSN

The LISTDSN file could not be opened, check

1. The PARM= statement is valid for this type of request
2. The file is a sequential file or member of a PDS
3. System log for any error messages associated with this file

0101 No records on LISTDSN

The LISTDSN file is empty.

0102 Logic error in LISTDSN

One of the requests on the LISTDSN file is not in the correct format.

0103 Error reading LISTDSN

An error occurred while reading the file, check the system messages for more information.

0104 Error processing document text.

Either document text has been found when processing a message or document text was not found when it was expected. Check that the PARM= value matches the type of request being made.

0105 Internal logic error

The function passed to a sub-module was invalid. Contact Icom Solutions.

0106 Parameter sequence error

When a request is made, the COMMAND parameter must be first. The other parameters may appear in any order

Note: Sub-parameters must appear immediately after the relevant parameter.

0107 Parameter blank or invalid

One of the supplied parameters is either invalid, or the parameter value is blank.

Note: If you want a parameter to be blank, it should not be included in the request. Only the MESSAGE parameter may contain blank lines.

0110 Internal logic error

An error occurred while sending DCA LEVEL sub-parameters. Check that the PARM= value matches the type of request being made.

0150 Maximum document size exceeded

The maximum document size specified on the SYSTEM record in the OfficePath control file or in the document size validation exit (PIPDOCSZ) has been exceeded.

02xx Parameter data invalid

Parameter number xx contains invalid data.

03xx Parameter invalid for request

Parameter number xx is invalid for this type of request.

04xx Sub-parameter invalid for request

Parameter number xx has a sub-parameter which is invalid for this type of request.

05xx Mandatory parameter missing

Parameter number xx is required for this type of request.

06xx Mandatory sub-parameter missing

Parameter number xx requires a sub-parameter to be supplied for this type of request.

07xx Error accessing ZPMAILBX, the intray dataset.

The COBOL Status code returned was xx.

08xx User ID rejected by user exit

The user-validation exit rejected the supplied userid with a return code xx. The return code xx should be checked with your systems programmer.

1030 Error accessing CARDFILE

An error occurred while accessing the CARDFILE dataset, check

1. The PARM= statement is valid for this type of request
2. The file is a sequential file or member of a PDS
3. System log for any error messages associated with this file

Where xx is used in the above examples, it refers to the number of the affected parameter as given in the following list:

00	COMMAND
01	DATE-TIME
02	DEST-CODE
03	DCA LEVEL
04	REPLY
05	USER NAME
06	PASSWORD
07	QUALIFIER^(reserved for RAPID compatibility)
08	FROM DATE^(reserved for RAPID compatibility)
09	TO DATE^(reserved for RAPID compatibility)
10	PRIORITY
11	PERSONAL
12	PROF GCID
13	MESSAGE
14	RECIPIENT
15	PROCESS
16	CREATED
17	DOC DATE
18	EXPIRES

19	CLASS
20	DOC NAME
21	FCAB REF
22	SUBJECT
23	AUTHOR
24	KEYWORD
25	ACCESS
26	CORRELATE
27	UPDATED
28	DIST CORR

See "COBOL file status codes" for the meanings of the COBOL return codes.

Return codes from output processor

The output module can be invoked via batch JCL or using a TSO CLIST. If the program is invoked from batch then it will display a message and issue a return code. Otherwise it will only issue a return code.

Use of this module is described fully in the *OfficePath Programming Interfaces* manual.

The possible return codes are:

0099 Invalid PARM value

The PARM= statement does not contain one of the valid values for this program.

07xx Error accessing OUTTRAY

The COBOL status code returned was xx.

0900 Error accessing LISTDSN

An error occurred while accessing the LISTDSN dataset. Check that:

1. The PARM= statement matches the files which are supplied
2. The file is a sequential file or member of a PDS

3. System log for any error messages associated with this file

1030 Error accessing CARDFILE

An error occurred while accessing the CARDFILE dataset. Check that:

1. The PARM= statement matches the files which are supplied
2. The file is a sequential file or member of a PDS
3. System log for any error messages associated with this file

See "COBOL file status codes" for the meanings of the COBOL return codes.

COBOL file status codes

Error codes which are returned from file accesses by the Batch Interface programs are COBOL status codes, not VSAM. Descriptions of these codes can be found in the *IBM VS COBOL for OS/VS* manual.

The following status codes may be returned:

10 End of file

22 Duplicate record

23 Record not found

90 Other error. No further information: could be file empty on open for input, or not empty on open for output

91 File is password protected

92 Logic error in program: for example record on file longer than program is expecting or I/O before successful OPEN.

93 Resource not available: data set in use (See PIPS126S message).

96 No DD card for file

Appendix B. CICS Application Interface Return Codes

This section lists the return codes which can be issued by the OfficePath CICS application interface feature. Use of these programs is described in the *OfficePath Programming Interfaces* manual.

Return codes from input processor

The input module is invoked via a CICS LINK command from an application program. The program will return a non-zero return code in the commarea if an error occurs in the processing of the request.

Use of this module is described fully in the *OfficePath Programming Interfaces* manual. The return codes and parameter numbers are also described in that manual and are shown here for completeness only.

The return codes which may be given are:

0000 Command successfully processed

0104 Error processing document text

Either document text has been found when processing a message, or document text was not found when it was expected. Check that the format of the request on the TSQ is correct.

0105 Internal logic error

The function passed to a sub-module was invalid. Contact Icom Solutions.

0106 Parameter sequence error

When a request is made, the COMMAND parameter must be first. The other parameters may appear in any order

Note: Sub-parameters must appear immediately after the relevant parameter.

0107 Parameter blank or invalid

One of the supplied parameters is invalid, or the parameter value is blank.

Note: If you want a parameter to be blank, it should not be included in the request. Only the MESSAGE parameter may contain blank lines.

02xx Invalid parameter data

Parameter number xx contains invalid data.

03xx Parameter invalid for request

Parameter number xx is invalid for this type of request.

04xx Sub-parameter invalid for request

Parameter number xx has a sub-parameter which is invalid for this type of request.

05xx Mandatory parameter missing

Parameter number xx is required for this type of request.

06xx Mandatory subparameter missing

Parameter number xx requires a sub-parameter to be supplied for this type of request.

1rrr Error reading from TSQ, CICS EIBRESP is rrr.

A CICS error occurred while reading an item from the TSQ.

2rrr Error writing to in-tray, CICS EIBRESP is rrr.

A CICS error occurred while writing an item to the intray.

Some of the EIBRESP values you may receive are:

012	FILENOTFOUND
015	DUPREC
018	NOSPACE
019	NOTOPEN
044	QIDERR

xx is the number of the parameter in question, as detailed in the following list:

00	COMMAND
01	DATE-TIME
02	DEST-CODE
03	DCA LEVEL
04	REPLY
05	USER NAME

06	PASSWORD
07	QUALIFIER^(reserved for RAPID compatibility)
08	FROM DATE^(reserved for RAPID compatibility)
09	TO DATE^(reserved for RAPID compatibility)
10	PRIORITY
11	PERSONAL
12	PROF GCID
13	MESSAGE
14	RECIPIENT
15	PROCESS
16	CREATED
17	DOC DATE
18	EXPIRES
19	CLASS
20	DOC NAME
21	FCAB REF
22	SUBJECT
23	AUTHOR
24	KEYWORD
25	ACCESS
26	CORRELATE
27	UPDATED
28	DIST CORR

Return codes from output processor

The output module is invoked via a CICS LINK command from an application program. The program will return a non-zero return code in the commarea if an error occurs in the processing of the request.

Use of this module is described fully in the *OfficePath Programming Interfaces* The return codes and parameter numbers are also described in that manual and are shown here for completeness only.

The return codes which may be given are:

0000 Command successfully processed

0001 Application out-tray is empty

No requests were found on the specified Application Out-tray.

0100 Internal logic error

An error was returned by a sub-module. Check that the Destination record for the Out-tray on the Control File specifies 'Uses Interface program'. If not, then alter the Destination record and delete all requests from the Out-tray. If the Destination record is correct then contact the TBS Software Ltd Support Team as described in "Requesting Additional Support"

1rrr Error writing to TSQ

A CICS error occurred while writing the request to the TSQ, CICS EIBRESP is rrr.

2rrr Error reading from out-tray, CICS EIBRESP is rrr.

A CICS error occurred while reading a request from the Out-tray.

3rrr Error deleting from out-tray. CICS EIBRESP is rrr.

A CICS error occurred while deleting the request from the Out-tray.

Some of the EIBRESP values you may receive are:

012	FILENOTFOUND
018	NOSPACE
019	NOTOPEN
044	QIDERR

Appendix C. NAK Codes

OfficePath can only issue four NAK (Negative Acknowledgement) codes:

C30307 Unauthorized access - the request is for a command not supported by OfficePath (for example FILE, SEARCH, RETRIEVE, and DELETE).

C3111B The request has been rejected because the document is too large for OfficePath to process.

C40209 An invalid printer name has been specified for a print request.

C40B17 Error detected - the recipient's status is set to ERROR.

Appendix D. OfficePath Dataset Layouts

Note: These layouts are provided for diagnosis purposes only. They may be changed in future releases of OfficePath.

The following characters are used to specify the field types:

G	a group level with sub-fields
C	Character format
D	packed decimal format
F	Fullword, binary format
H	Halfword, binary format

PIPSTAT

RECORD LENGTH = 200

KEY LENGTH = 37

Offset	**Field Name**	**T**	**Length**	**Field Description**
The key of the file				
0	FS-STAT-KEY	G	37	The key of the file
0	FS-SENDER-LOCN	C	8	The sender's node id
8	FS-SENDER-USER	C	8	The sender's user id
16	FS-PROCESSED-FLAG	C	1	The state of the status reporting: **N** the status has not been reported to OfficePath **P** the status has been reported to OfficePath, but not to the originator **Y** the status has been reported to OfficePath and to the originator
17	FS-ORIG-MSGNO	F	4	The message number of the owning request
21	FS-RECIP-LOCN	C	8	The recipient's node id
29	FS-RECIP-USER	C	8	The recipient's user id
The remainder of the file				

37	FS-STAT-CODE	G	4	The recipient status
37	FS-STAT-TYPE	C	2	The recipient status type
39	FS-STAT-CC	C	2	The recipient status contents
41	FS-STAT-DATE	D	4	The date associated with the return of the status
45	FS-STAT-TIME	D	4	The time associated with the return of the status
49	FS-STAT-NODE	D	8	The node which returned the status
57	FS-STAT-NAK	C	3	The NAK code (optional)
60	FS-STAT-NAK-DATA	C	80	The NAK data (optional)
140	FS-RCVD-DCA	H	2	not used
142	FS-RCVD-GCID	B	4	not used
146	FS-REAL-RECIP	G	16	The original recipient specified (if the recipient was changed either through PIPEXIT1 or through the redirection feature). It is set to high values if the recipient was added to the request.
146	FS-REAL-RECIP-LOCN	C	8	The original node id
154	FS-REAL-RECIP-USER	C	8	The original user id
162	FS-RDRF-FLAG	C	1	used with FS-DEST to determine the associated mail records when recipients are accumulated
163	FS-DEST	C	8	the name of the destination used
171	FILLER	C	29	low values

PIPMAIL

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FM-MAIL-KEY	G	22	The key of the file
0	FM-RECIP-LOCN	C	8	The recipient's node id
8	FM-RECIP-USER	C	8	The recipient's user id

16	FM-ORIG-MSGNO	F	4	The message number of the owning request
20	FM-SEQUENCE-NO	H	2	a sequence number if there are more than fifty recipients
The remainder of the file				
22	FM-DCA	H	2	The DCA level
24	FM-GCID	B	4	The document GCID
28	FM-COUNT	H	2	The number of recipients
30	FM-RECIPIENT (x50)	G	16	The date associated with the return of the status
	FM-RLOCN	C	8	The recipient's node id
	FM-RUSER	C	8	The recipient's user id

PIPLOG

RECORD LENGTH = 566

KEY LENGTH = 5

The are three record types on the PIPLOG file.

Message Records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FL-LOG-KEY	G	5	The key of the file
0	FL-TYPE	C	1	"M" or "X"
0	FL-MSGNO	F	4	the message number
The input details				
5	FL-REQUESTOR	G	16	The requestor (sender)
5	FL-REQ-LOCN	C	8	the requestor's node id
13	FL-REQ-USER	C	8	the requestor's user id
21	FL-FRST-DATE-TIME	G	8	The date and time that the request was created
21	FL-FRST-DATE	D	4	The date that the request was created
25	FL-FRST-TIME	D	4	The time that the request was created

29	FL-SEND-STATUS	H	2	The status with respect to the sender
31	FL-COMMAND	C	1	The command being executed: **A** - ACK **B** - SIGNOFF **C** - CANCEL **D** - DELETE **E** - EXECUTE **F** - FILE **G** - RETRIEVE **I** - INPUT **J** - JCL **L** - LIST **M** - MODIDY **N** - SIGNON **O** - OBTAIN **R** - SEND **S** - SEARCH **T** - FORMAT **V** - CHANGE
32	FL-CORRELATION	C	16	The correlation data
48	FL-INPUT-FILE	C	8	The name of the input file
56	FL-INPUT-DCA	H	2	The input DCA level
58	FL-INPUT-GCID	B	4	The input GCID
62	FL-PROFILE-TYPE	C	1	The type of the profile: **D** - document **M** - message **P** - profile **N** - note
63	FL-DOC-NAME	C	44	The document name
107	FL-HOP-COUNT	H	2	The current HOP count
109	FL-DOC-SIZE	F	4	The size the document text

113	FILLER	C	14	Low values
The COD details				
127	FL-COD	C	1	Whether COD has been requested (Y on N)
128	FL-REPLY-DEST	C	8	The name of the reply destination
136	FL-REPLY-DETAILS	C	44	the reply data set name
180	FL-REPLY-FILE	C	8	The where the reply will be written to
188	FL-NAK	C	3	The NAK code
191	FL-REP-DATA	G	80	The reply data
191	FL-REP-DDN	C	20	The distribution document name
211	FL-REP-DTIME	C	8	The reply date and time
219	FILLER	C	52	
271	FL-INPUT-TYPE	C	1	set the "3'" from requests made via an intray
272	FL-REPLY-DIUS-SEPARATE	C	1	Whether the reply DIUs should be sent separated for API applications
273	FILLER	C	18	low values
The recipient details				
291	FL-RECIP1	C	16	the node and user id of the first recipient
307	FL-LAST-DATE-TIME	G	8	The date and time the request was last updated
307	FL-LAST-DATE	D	4	The date the request was last updated
311	FL-LAST-TIME	D	4	The time the request was last updated
315	FL-RECIP-STATUS	H	2	The status with respect to the recipients
317	FL-NO-RECIP	H	2	The number of recipients
319	FL-NO-PROCESSED	H	2	The number of recipients where the mail has been successfully sent
321	FL-NO-CONFIRMED	H	2	The number of recipients for whom mail has been confirmed
323	FL-NO-REJECTED	H	2	The number of recipients for whom mail has been rejected
325	FL-NEW-RECIP1	C	16	The node and user id of the first recipient if it has been changed
341	FILLER	C	2	low values
The next transaction of be invoked				
343	FL-TRANSID	C	4	the transaction id
347	FL-START-	G	36	the start data for the transaction

	DATA			
other details				
383	FL-TERMID	C	4	the termid for the transaction to be started on
387	FL-NEW-ORIG	C	16	indicating a change of originator

Header record

Offset	Field Name	T	Length	Field Description
0	FL-LOG-KEY	C	5	low values
5	FL-LAST-MESSAGE	F	4	The last message number used

Trailer record

Offset	Field Name	T	Length	Field Description
0	FL-LOG-KEY	C	5	high values
5	FL-LAST-RESTART-ALL	G	36	The text displayed at the foot of the log file browse panel
5	FL-LAST-RESTART-TEXT	C	16	"LAST RESTARTED: "
21	FL-LAST-RESTART-DATE	C	8	The date that OfficePath was last restarted
29	FL-LAST-RESTART-FILLER	C	4	" AT "
33	FL-LAST-RESTART-TIME	C	8	The time that OfficePath was last restarted

PIPCNTL

There are a number of different records on the control file.

The Customize record

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-CUST-KEY	G	17	The key of the record
0	FC-CUST-TYPE-C	C	1	"C"
1	FILLER	C	16	spaces
The remainder of the record				
17	FC-CUST-LOCAL-GCID	B	4	The GCID specified as "local"
21	FC-CUST-RETAIN-DAYS	C	2	the number of days that "X" records are kept
23	FC-CUST-ERR-LOG	C	4	The name of the CICS TDQ to which error messages are written
27	FC-CUST-DEFLOCN	C	8	The local node id
35	FC-CUST-TIMER	D	4	The frequency at which the application input files are scanned for requests
39	FC-CUST-ENQ-INTERVAL	D	4	The interval OfficePath waits after an unsuccessful MVS ENQ
43	FC-CUST-SUB-INTERVAL	D	4	the time for which OfficePath will wait, after writing a request to a batch outtray, before it submits the JCL to process the outtray.
47	FC-CUST-SUB-INTRDR	C	4	the name of the internal reader which will be used to submit the JCL to process the batch file
41	FC-CUST-HOP	H	2	the number of nodes which a distribution may pass through before it is cancelled.
43	FC-CUST-LNUMBER	C	8	not used
51	FC-CUST-PRTLOCN	C	8	used for print requests from API application where a "DISOSS" print is specified
59	FC-CUST-LOGFLAGS.	C	3	Three 1-byte flags indicating whether LIST, SIGN-ON and SIGN-OFF requests should be logged.
62	FC-CUST-DEFDOCN	C	44	The default document name to be used the incoming request does not specify one
106	FC-CUST-PURGE	C	2	the number of days before "M" records are deleted
108	FC-CUST-OPTTIME	D	4	The time at which the PIPS transaction will be automatically re-run to "restart" OfficePath.
112	FC-CUST-OUTPUT-INT	D	4	the amount of time OfficePath will wait before retrying a failing send request

116	FC-CUST-ERROR-MAX	H	2	not used
118	FC-CUST-LAST-USER	C	8	who last updated this record
126	FC-CUST-LAST-TERM	C	8	the terminal used for the update
134	FC-CUST-LAST-ABSTIME	C	8	when it was last updated
142	FC-CUST-COMMENTS	C	44	comments
186	FC-CUST-MAX-DOC-SIZE	F	4	the maximum document size which can be processed by the system
190	FC-CUST-STATUS	C	1	whether the status records should be kept for completed requests
191	FC-CUST-TEMP-STORAGE	C	1	whether main or auxiliary temporary storage should be used
192	FC-CUST-TRACE	C	3	three one-byte flag used to determine the level of tracing to take place

Destination records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-DEST-KEY	G	17	The key of the records
0	FC-DEST-TYPE-D	C	1	"D"
1	FILLER	C	8	spaces
9	FC-DEST	C	8	the name of the destination
The remainder of the records				
17	FC-DEST-TYPE	C	1	"1" - API, "3" - OUTTRAY, "4" - INTRDR, "6" - SNADS
18	FC-DEST-DCA (x11)	H	2	the supported DCA levels
40	FC-DEST-GCID (x10)	B	4	the supported GCIDs
80	FC-DEST-RDRF-F	C	1	whether the destination should accumulate recipients
81	FC-DEST-GT3-F	C	1	whether the outtray destination used the PIPGT3 interface program to receive the mail items
82	FC-DEST-CONF-F	C	1	whether the destination will return a

				confirmation of delivery
83	FC-DEST-ASA-F	C	1	whether the destination supports ASA control characters
84	FC-DEST-BATCH-F	C	1	whether the outtray is to be processed in batch
85	FC-DEST-NOTE-F	C	1	whether the destination support documents of type NOTE
86	FILLER	C	4	low values
90	FC-DEST-TRAN	C	4	the name of the transaction to be invoked to process the outtray
94	FC-DEST-FACILITY	C	4	either the TCT for SNADS or the TDQ for INTRDR
98	FC-DEST-FILE	C	8	the name of the outray
106	FC-DEST-DCB	C	2	the output record format
108	FC-DEST-RECLEN	H	2	the line length to be used
110	FC-DEST-LAST-MSGNO	F	4	?
114	FC-DEST-LAST-USER	C	8	the user who last amended the record
122	FC-DEST-LAST-TERM	C	8	the terminal at which the record was amended
130	FC-DEST-LAST-ABSTIME	C	8	when the record was last amended
130	FC-DEST-COMMENTS	C	64	comments
194	FC-DEST-HOLD-FLAG	C	1	the mail and/or status request are "on hold"

Formatter records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-FMT-KEY	G	17	The key of the record
0	FC-FMT-TYPE-F	C	1	"F"
1	FILLER	C	8	spaces
9	FC-FMT-DCA-IN	H	2	the input (source) DCA level
11	FILLER	C	6	spaces
The remainder of the record				
17	FC-FMT-DETAIL	G	10	

	(x10)			
	FC-FMT-DCA	H	2	the target DCA level
	FC-FMT-PROG	C	8	the formatter program

GCID records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-GCID-KEY	G	17	The key of the record
0	FC-GCID-TYPE-G	C	1	"G"
1	FC-GCID-FROM-GPID	H	2	the source GPID
3	FILLER	C	6	spaces
9	FC-GCID-TO-GCID	B	4	the target GCID
13	FILLER	C	4	spaces
The remainder of the record				
17	FC-GCID-TABLE	C	192	the characters x"40" to x"FF"

Help records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-HELP-KEY	G	17	The key of the record
0	FC-HELP-TYPE-H	C	1	"H"
1	FC-HELP-TYPE	C	9	corresponding to the option
1	FC-HELP-SCREEN-NO	C	3	the page number
The remainder of the record				
17	FC-HELP-LINE (x16)	C	78	a line of text

Input records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-INP-KEY	G	17	The key of the record
0	FC-INP-TYPE-I	C	1	"I"
1	FILLER	C	8	spaces
9	FC-INP-FILE	C	8	the name of the mailsack
The remainder of the record				
17	FC-INP-BATCH-F	C	1	whether the mailsack is processed in batch
18	FC-INP-LAST-USER	C	8	the last user to modify the record
26	FC-INP-LAST-TERM	C	8	the terminal used to modify the record
34	FC-INP-LAST-ABSTIME	C	8	when the record was last amended
42	FC-INP-COMMENTS	C	64	comments

JCL records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-JCL-KEY	G	17	The key of the file
0	FC-JCL-TYPE-J	C	1	"J"
1	FC-JCL-LOCN	C	8	the location associated with the JCL
9	FC-JCL-USER	C	8	the user id associated with the JCL
The remainder of the file				
17	FC-JCL-LINE (x56)	C	72	the lines of the JCL
4049	FC-JCL-NUM-LINES	H	2	the number of lines of JCL
4051	FC-JCL-LAST-USER	C	8	the user who last amended the record
4059	FC-JCL-LAST-TERM	C	8	the terminal at which the record was amended
4065	FC-JCL-LAST-ABSTIME	C	8	when the record was last amended

Copylist records

Offset	Field Name	T	Length	Field Description

The key of the file				
0	FC-LIST-KEY	G	17	The key of the file
0	FC-LIST-TYPE-L	C	1	"L"
1	FILLER	C	8	spaces
9	FC-LIST	C	8	the name of the copylist
The remainder of the file				
17	FC-LIST-NO-RECIP	H	2	the number of recipient in the copylist
19	FC-LIST-RECIP (x254)	G	16	the location and user id of each recipient in the list

Message records

Offset	**Field Name**	**T**	**Length**	**Field Description**
The key of the file				
0	FC-MSG-KEY	G	17	The key of the record
0	FC-MSG-TYPE-M	C	1	"M"
1	FILLER	C	8	spaces
9	FC-MSG-CODE	C	4	the message number
13	FILLER	C	4	spaces
The remainder of the record				
17	FC-MSG-TEXT	C	78	the text of the message

Option records

Offset	**Field Name**	**T**	**Length**	**Field Description**
The key of the file				
0	FC-OPT-KEY	G	17	The key of the record
0	FC-OPT-TYPE-O	C	1	"O"
1	FC-OPT	C	16	the name of the option
The remainder of the record				
17	FC-OPT-DESCRIPTION	C	40	a text description

57	FC-OPT-PROGRAM	C	8	the program to be invoked

PS Family records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-PSFAM-KEY	G	17	The key of the record
0	FC-PSFAM-TYPE-P	C	1	"P"
1	FC-PSFAM-NODE	C	8	the node id
9	FILLER	C	8	spaces
The remainder of the record				
17	FC-PSFAM-ENTRY x12	G	12	
18	FC-PSFAM-HUD	C	8	the name of the PS Host User Dataset
26	FC-PSFAM-TRAN	C	4	the name of the mail/status transaction
161	FC-PSFAM-LAST-USER	C	8	the user who last amended the record
169	FC-PSFAM-LAST-TERM	C	8	the terminal at which the record was amended
177	FC-PSFAM-LAST-ABSTIME	C	8	when the record was last amended
185	FC-PSFAM-COMMENTS	C	64	comments

Transaction destination records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-TRAN-KEY	G	17	The key of the record
0	FC-TRAN-TYPE-T	C	1	"T"
1	FILLER	C	8	spaces
9	FC-TRAN-PREFIX	C	3	the first three characters of the mail/status transaction
12	FILLER	C	5	spaces

The remainder of the record				
17	FC-TRAN-LOCN	C	1	the associated node id

User records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FC-USER-KEY	G	17	The key of the file
0	FC-USER-TYPE-U	C	1	"U"
1	FC-USER-LOCN	C	8	the user's node
9	FC-USER	C	8	the user's id
The remainder of the file				
17	FC-USER-DEST	C	8	the destination to be used
25	FC-USER-DTERM	C	4	optional terminal to be used for debugging
29	FILLER	C	4	spaces
33	FC-USER-LAST-USER	C	8	the user who last amended the record
41	FC-USER-LAST-TERM	C	8	the terminal at which the record was amended
49	FC-USER-LAST-ABSTIME	C	8	when it was amended
47	FC-USER-COMMENTS	C	44	comments
33	FC-USER-REDIRECT	C	16	the node and user of the "new" user
49	FC-USER-REDIR-TYPE	C	1	the type of redirection
50	FC-USER-REDIR-DATE	C	6	the starting date for redirection

PIPTEXT

There are different record types on the text file.

Profile records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FT-TEXT-KEY	G	12	The key of the file
0	FT-MSGNO	F	4	The message number associated with the request
4	FT-DCA-LEVEL	H	2	The dca level of the text on the file
6	FT-GCID	B	4	The GCID of the text on the file
10	FT-KEY-PAGE	H	2	The page number
Profile				
12	FILLER	C	6	
18	FT-DOC-TYPE	H	2	The input DCA level
20	FT-DOC-TYPE-ON-EXIT	H	2	The requested output DCA level
22	FT-CHARS-PER-LINE	H	2	The number of characters on a line
24	FT-PROFILE-TYPE	C	1	'D' - Document, 'M' - Message, 'P' - Profile, 'N' - Note
25	FT-COMMAND	C	1	The command to be executed (see FL-COMMAND on PIPLOG)
26	FT-ACTION-IND	C	1	'2' - DELIVERED, '3' - CANCELLED, '5' - INVALID
27	FT-COD	C	1	'Y'
28	FT-COD-SYSTEM	C	8	Where the replies (CODs) will go
36	FT-COD-DATASET	C	44	Either name name of the dataset to which CODs will be written or the location and user of the recipient
80	FT-SENDER	C	16	The node and user id of the sender
96	FT-PASSWORD	C	8	For compatibility with RAPID
104	FL-CORRELATION	C	16	Correlation data
120	FT-REPLY-DETAILS	G	103	
120	FT-NAK-CODE	C	3	The negative acknowledgement code

123	FT-DIST-DOC-NAME	C	20	The distribution document name (DDN): sender location, sender id and sequence number.
143	FT-DIST-DATE-TIME	C	6	The date and time of the distribution
149	FT-DIST-DINA	C	44	Distribution name
193	FT-SYSTEM-CODE	C	13	The system code
206	FT-HEADER-TPN	C	4	Usually X'20F0F0F0'
210	FT-STATUS-TPN	C	4	Usually X'20F0F0F0'
214	FT-SERVER-TPN	C	4	Usually X'20F0F0F1'
218	FT-DIST-DINA-LL	H	2	The length of the FT-DIST-DINA field
223	FT-RAPID-RESVD	C	100	reserved
323	FT-PAGE0-GCID	C	4	The GCID for the profile - usually multi-lingual
327	FT-QUALIFIER-F	C	1	D - DDN, L - LADN, S - SRL
328	FT-QUALIFIER	C	20	The data corresponding to the previous flag
348	FT-PRIORITY-F	C	1	"Y" or "N"
349	FT-PERSONAL-F	C	1	"Y" or "N"
350	FT-APPEND-F	C	1	"Y" or "N"
351	FT-RA-F	C	1	"Y" or "N"
360	FT-MESSAGE	C	256	The message
616	FT-RECIPIENT-AREA	C	800	50 recipients
1416	FT-DOC-PROFILE	G		
1416	FT-CREATE-DATE	C	6	The document creation date
1422	FT-DOC-DATE	C	4	The document date
1426	FT-EXPIRE-DATE	C	4	The expiry date
1430	FT-LAST-UPDATED	C	6	The last updated date
1436	FT-DOC-GCID	C	4	The document GCID - usually multi-lingual
1440	FT-DOC-CLASS	C	16	The document class
1456	FT-DOC-NAME	C	44	The document name
1500	FT-SUBJECT	C	60	The subject
1560	FT-ACCESS-CODES (x8)	C	5	Up to eight access codes
1600	FT-KEYWORD (x20)	C	60	Up to eight keywords
2800	FT-AUTHOR (x20)	C	20	Up to twenty authors
3200	FT-DOC-CONTENTS	G		
3200	FT-FCAB-REF	C	60	The external reference
3200	FT-LINES	H	2	The number of lines per page

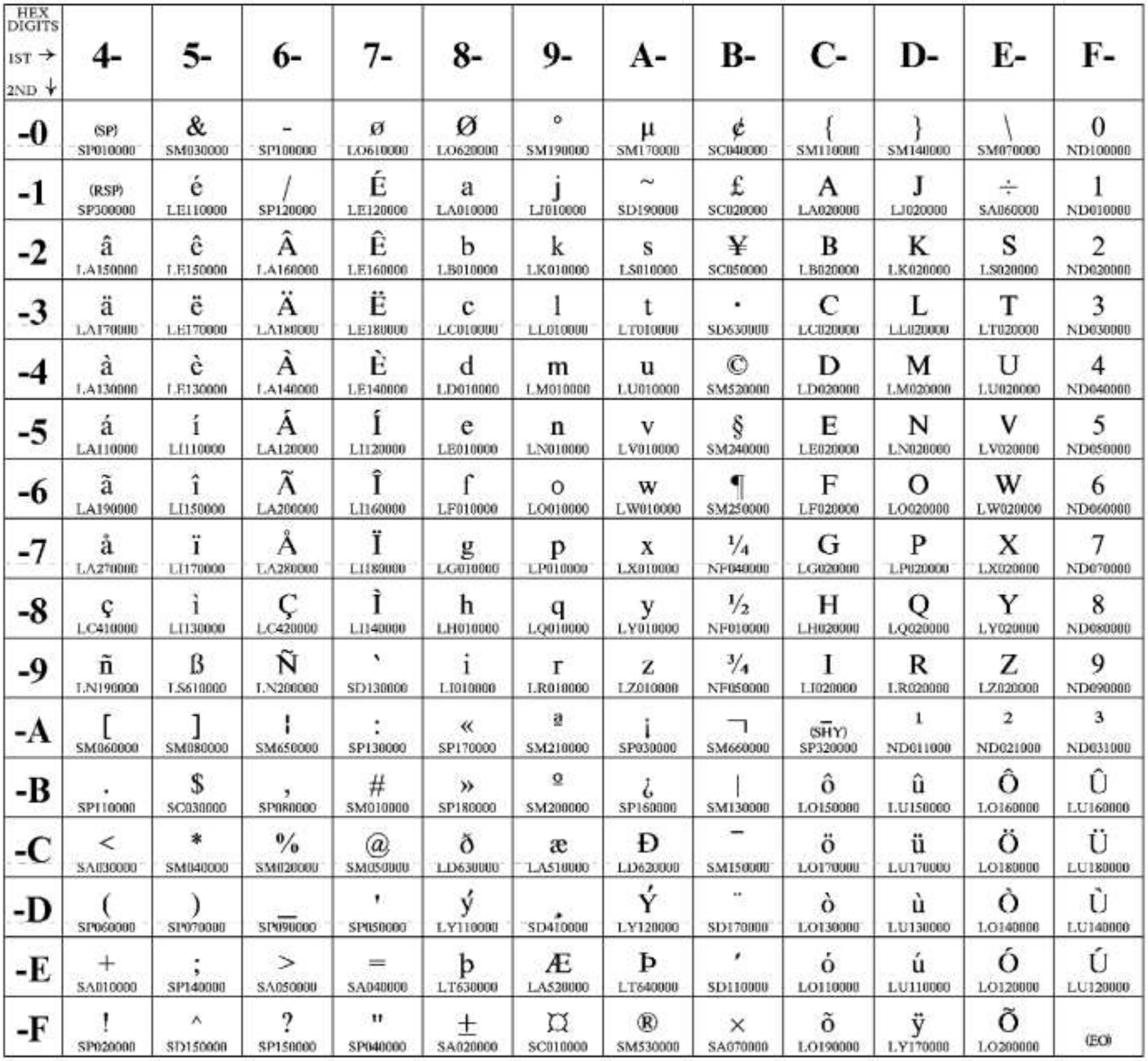
3202	FT-GLOBAL-FONT	H	2	The font used
3204	FT-FONT-WIDTH	H	2	The font width
3206	FT-LINE-DEPTH	H	2	The font height
3208	FT-TOP-MARGIN	H	2	The top margin
3210	FT-BOT-MARGIN	H	2	The bottom
3212	FT-LEFT-MARGIN	H	2	The left-hand margin
3214	FT-RIGHT-MARGIN	H	2	The right-hand margin
3216	FT-DOC-PAGES	H	2	The number of pages in the document

Text records

Offset	Field Name	T	Length	Field Description
The key of the file				
0	FT-TEXT-KEY	G	12	The key of the file
Document text				
24	FT-DATA-AREA	C	8168	The document text

Appendix E. Multi-lingual Code Page Characters

The table below shows the characters in the multi-lingual code page.



HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	µ SM170000	¢ SC040000	{ SM110000	} SM140000	\ SM070000	0 ND100000
-1	(RSP) SP300000	é LE110000	/ SP120000	É LE120000	a LA010000	j LJ010000	~ SD190000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	· SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	§ SM240000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ì LI130000	Ç LC420000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	` SD130000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	[SM060000	] SM080000	¦ SM650000	: SP130000	« SP170000	ª SM210000	¡ SP030000	¬ SM660000	(SHY) SP320000	¹ ND011000	² ND021000	³ ND031000
-B	. SP110000	$ SC030000	, SP080000	# SM010000	» SP180000	º SM200000	¿ SP160000	| SM130000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	@ SM050000	ð LD630000	æ LA510000	Ð LD620000	¯ SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	! SP020000	^ SD150000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00500

Index