

RAPID

Version 3 Release 4.3

Installation and Administration



Rapid is one of the portfolio of products marketed and supported by TBS Software Inc. This portfolio includes products to facilitate enterprise-wide messaging and office automation across heterogeneous platforms, and products which complement IBM's OfficeVision/MVS, enriching its functionality and usability, and simplifying the development of mail-enabled applications.

[SMTP-Send for z/OS and OS/390](#)

SMTP-Send for z/OS and OS/390 lets you easily integrate e-mail distribution capabilities into your MVS Batch, TSO and CICS applications.

[OfficePath/SMTP-Send for z/OS and OS/390](#)

OfficePath/SMTP-Send for z/OS and OS/390 replaces your SNADS e-mail network with industry-standard SMTP. Because it is designed to work with OfficePath or RAPID, no changes are required to your existing mail-enabled line-of-business applications.

OfficePath/Batch-TSO Provides a bi-directional gateway to OV/MVS, or PS/CICS, to simplify the development of mail-enabled Batch applications; it includes TSO dialogs to mail data sets from TSO.

OfficePath/CICS Provides a bi-directional gateway to OV/MVS, or PS/CICS, to simplify the development of mail-enabled CICS applications.

OfficePath/SNADS Provides a SNADS gateway to OV/MVS or PS/CICS. This enables your mail-enabled CICS or Batch application, or any OV/MVS or PS/CICS user, to mail to/from any user on the SNADS network. Can be used to mail between multiple OV/MVS and/or PS/CICS systems, IBM's Mail Exchange, or to/from LAN-based mail systems.

RAPID/Batch-TSO Provides a bi-directional gateway to DISOSS to simplify the development of mail-enabled Batch applications. It includes TSO dialogs to mail datasets from TSO.

RAPID/CICS Provides a bi-directional gateway to DISOSS to simplify the development of mail-enabled CICS applications.

PIXI Allows users to move almost any documents between IBM MVS Office systems and personal workstations (PC, PS/2 etc.) in a simple, cost-effective manner.

CLASY Provides automatic backup and archiving of mail in Personal Services or PS/CICS. Makes more effective use of on-line storage by moving non-essential documents off-line.

DSR Monitors the storage of documents in Personal Services or DISOSS and the Application Services Document Cabinet.

Hupdate/Rupdate	Provides on-line administration of the DISOSS HUP data set and/or Routing data set.
OV/Assist	Substantially simplifies the administration of IBM's OfficeVision/MVS.
OV/Context	Allows a user to search for documents in their File Cabinet which contain a particular text-string.
OV/Re-Router	Provides automatic re-routing of a Personal Services user's mail to any other user on the e-mail network.
OV/Reminder	Automatically distributes a Note or document to other e-mail users or -Basket on a specified day.
OV/Fax	Integrates FAX facilities with the Office System enabling any connected user, or mail-enabled application, to send or receive Faxes. Application Fax and AutoFax features provide added integration with business applications.
OV/Bulletin	Integrates bulletin-board facilities with the Office system. Enables any connected user, or mail-enabled application, to post a bulletin. CICS users can view the bulletins, and request a copy.

For further information on any of these products please contact the address on the following page.

RAPID and *OfficePath* are registered trademarks of **Keane Ltd.**

IBM, *AS/400*, *PS/2* and *OS/2* are registered trademarks and *OfficeVision* is a trademark of **International Business Machines Corporation.**

CLASY, *SMTP-Send* and *OfficePath/SMTP-Send* are trademarks of **TBS Software Inc.**

The names and other products and companies mentioned may be registered trademarks or trademarks.

Preface

RAPID Version 3 Release 4.3

This manual was last revised on 21 February 2008.

For **sales information** please contact:

TBS Software Inc. (Head Office)
8300 Woodbine Ave
Suite 101
Markham, Ontario
Canada
L3R 9Y7

Tel: 905-940-9373 Fax: 905-940-9376

If you wish to communicate electronically for sales information, email to: sales@tbssoft.com

Product Support is available through electronic mail at support@tbssoft.com or via telephone or fax at:

Tel: 905-940-9373 Ext 777 Fax: 905-940-9376

Table of Contents

- [Preface](#)
 - [Figures](#)
 - [Introduction](#)
 - [Purpose of this book](#)
 - [Audience for this book](#)
 - [Pre-requisites](#)
 - [About This Release](#)
 - [Change history](#)
 - [Migration from earlier RAPID versions](#)
 - [Other books you may need](#)
 - [Product Overview](#)
-

PART ONE - Installing the RAPID system

- [Introduction](#)
- [Unloading the RAPID system](#)
 - [ZPCONTL - The Control File](#)
 - [ZPLOOKUP - The Keyword file](#)
 - [ZPMAILBX - The Mailbox File](#)
 - [ZPMAILSK - The Mailsack File](#)
 - [ZPMESSG - The Message File](#)
 - [ZPRDRF - The Rationalise Distribution file](#)
 - [CICS Table Updates](#)
 - [DCT](#)
 - [FCT](#)
 - [PCT](#)
 - [PPT](#)
 - [CICS Job Control Changes](#)
- [Updating DISOSS for RAPID](#)
 - [CICS Program Names](#)
 - [CICS Transaction Names](#)
 - [CICS DD Names](#)
 - [GCID support](#)
 - [Language Support](#)
 - [DISOSS parameters](#)
 - [Internal Reader Length](#)
 - [Error reporting on master console](#)
 - [Automatic message file delete](#)
 - [Batch interface](#)

- [RFT conversion](#)
 - [ZAPTABA Table](#)
 - **[Password Validation Exit](#)**
 - **[Permitted Document Size Exit](#)**
 - **[Installation Checklist](#)**
 - [Testing the Installation](#)
-

PART TWO - Administering the RAPID system

- [Starting RAPID](#)
 - [Closing RAPID](#)
 - [Autostart/close](#)
 - [RAPID timers](#)
 - [General Information](#)
 - [ZPINTRDR - Internal reader](#)
 - [ZPMAILBX - The Mailbox File](#)
 - [ZPMAILSK - The Mailsack File](#)
 - [ZPMESSG - The Message File](#)
 - [ZPLOOKUP - The Keyword File](#)
 - [ZPCONTL - The Control File](#)
 - [ZPRDRF - The Rationalise Distribution file](#)
 - [DSVAPIQ - The DISOSS APIQ](#)
 - [RAPID message file browse](#)
 - [The Online Browse facility](#)
 - [The Batch print/cleanup utility](#)
 - [Control File Maintenance](#)
 - [Main Menu screen](#)
 - [User Screen](#)
 - [JCL Screen](#)
 - [Location Screen](#)
 - [Copylist Screen](#)
 - [System Screen](#)
 - [Interface Screen](#)
 - [Message Screen](#)
 - [RDRF Screen](#)
-

APPENDIX

- **[Appendix A. Customisation Record](#)**
 - **[Appendix B. Program Temporary Fixes \(PTFs\)](#)**
-

Index

- **Appendix. Index**
-

List of Illustrations

1. [The RAPID system](#)
2. [Contents of the Installation Tape](#)
3. [Example JCL to Unload the RAPID.INSTALL Library](#)
4. [Example JCL to Unload the RAPID Libraries](#)
5. [Example JCL to Initialise the Control File](#)
6. [Example JCL to Initialise the Keyword File](#)
7. [Example JCL to Initialise the Mailbox File](#)
8. [Example JCL to Initialise the Mailsack File](#)
9. [Example JCL to Initialise the Message dataset](#)
10. [Example JCL to Initialise the RDRF file](#)
11. [DCT Table Updates](#)
12. [FCT Table Updates](#)
13. [PCT Table Updates - 1 of](#)
14. [PCT Table Updates - 2 of](#)
15. [PPT Table Updates - 1 of](#)
16. [PPT Table Updates - 2 of](#)
17. [CICS DD Cards](#)
18. [Control Statements for the HUP Utility](#)
19. [The ZAPTABA Code \(part 1 of 2\)](#)
20. [The ZAPTABA Code \(part 2 of 2\)](#)
21. [Example JCL to Re-link APIRAPID](#)
22. [PLT sample entry](#)

Introduction

Purpose of this book

This manual describes the installation and maintenance procedures for the Realtime Application Program Interface to DISOSS (RAPID), which provides a generalised interface between DISOSS and other applications.

RAPID is the generic term for RAPID/Base (which is the actual interface software), RAPID/TSO, RAPID/Batch and RAPID/CICS (which are sub-systems to interface to different environments).

Audience for this book

This manual is intended for the System Programmer responsible for the installation and maintenance of CICS based Office Systems software.

Other books you may need

There are 3 volumes that accompany the RAPID system software:

- *RAPID Installation and Administration* manual, which explains how to install, customize and administer the RAPID system.
- *RAPID Problem Determination* manual, which describes the detailed working of RAPID, and lists messages and codes.
- *RAPID User Guide*, which provides information on how to develop and install an interface between RAPID and other systems.

Pre-requisites

The successful implementation of RAPID assumes the prior installation of the following IBM software products:

- CICS/ESA Version 3.2.1 or later
- DISOSS Version 3, 5665-290, Release 2.0, 3.0, or
- Either
 COBOL OS/VS Library, 5740-LM1, Release 2.3

or

 COBOL OS/VS Compiler and Library, 5740-CB1, Release 2.3

In addition to the above, the successful implementation of the RAPID/TSO interface assumes the prior installation of TSO/ISPF.

If your installation does not support the above products, or is using different versions, please contact your local support group; the addresses and telephone numbers are given on the front page of this manual.

NOTE: If your installation uses the IBM Program Offering EDT then you will need to customise the RAPID program names, as described later in this manual, because the APISGNON and APIACTIV program ids exist on the EDT library.

About This Release

Users installing RAPID for the first time can skip this section and move onto Part One of this manual.

Change history

From a System Programmer's perspective, this release of RAPID is similar to previous RAPID releases, the main changes being

Version 3.4.3

- Various amendments have been made to enable RAPID to function in the year 2000 and beyond. Dates are held internally as 7 digits, including a century indicator (previously always zero) as the first digit. Where dates are entered with two digit years (e.g. in the

API), the year is assumed to be in the range 1984-2083. In order for RAPID to be able to determine the century, CICS/ESA V3.2.1 or later must be installed.

- Modifications to improve error handling in RAPIDGT3, previously available as FIX342A have been included.

Version 3.4.2

- New installation verification transactions ZAPF and ZAPT.
- RAPUSRC1 user exit now uses a subject field length of 60 characters.
- The 'Change Password' clist now supplies the 'reply' details.
- The 'Change Password' command now works with user exit RAPUSRCK.
- ZAPM now issues a warning message when a large number of COD requests are outstanding.
- ZAP6 now refreshes the screen when ENTER is pressed with no function details entered.
- RAPIDGS2 now uses the latest version of the MVS enqueue program.
- RAPIDGS3 does not attempt to open the CARDFILE DD when the parm passed is 'P'.
- RAPIDGS3 does not leave an MVS enqueue if the SYSOUT DD is missing.
- RAPIDGS3 does not include lines which begin with the x'03' control character.
- Message RAPID116 no longer contains the RAPID version number.
- ZAPQ and ZAPK transactions do not 'deadlock' each other.
- RAPRPORT now correctly spells 'successfully'.
- APIOBTN now reports the correct user id when logging the 'user not signed-on' message.
- APIMAIL/APIOUT enhanced so that the ZPMAILBX file can be allocated using an FCT entry.
- RAPIDGS3 enhanced to reject documents that exceed a defined size limit.
- APIGS3/APIGT3 enhanced to make error codes returned more detailed.
- APIMAIL enhanced to allow bad requests to be removed from the mailbox and allow processing to continue.

Version 3.4.1

- Bug causing version 3.4.0 to not display the correct version number fixed.
- Bug causing external references to not be delivered fixed.
- Bug causing profile fields to be incorrectly displayed when input from a variable length file fixed.

Version 3.4

- RAPID now has a password validation user exit, RAPUSRCK. This may be used to code installation specific validation or modification of userid, password, subject and document name in RAPID/Batch or RAPID/TSO.
- All known bugs have been fixed.

Version 3.3.5

- RAPID now supports the 'NOTE' document type. This must only be used with applications which are coded to support notes.
- RAPID Internal userid (default APIMASTR) can now be modified using the on-line panels.
- 'RAPID171W APIQ format not recognised' after PTF UN90019 has been corrected. RAPID now uses the API interface module for all processing and so is independent of any subsequent changes to the APIQ layout.
- Occasional deadlock between ZAPJ and ZAPN corrected. The problem only occurs if there is an outstanding OBTAIN command when RAPID is restarted.
- Informational messages (eg RAPID301I DIU field not recognised) can now be suppressed by deleting the /MSG record from the control file.
- List intervals for users can now be controlled using the value on the user record.

Version 3.3.4

- Support for more than 50 recipients. Removing the restriction on COPYLIST and RDRF processing.

Note: Only 50 recipients may be specified on a single input request.

- 'EXEC CICS SUSPEND' command in APIREPLY/APIOUT to allow other tasks access to CPU.
- RAPID/TSO CLISTS re-written to improve usability.
- Dataset open/close using 'EXEC CICS' commands.
- '31st December' bug fixed.

Version 3.3.3

- New program added to control file maintenance to maintain COPYLIST records.
- MVS ENQ changed to a conditional ENQ in the on-line transactions.
- RFT conversion altered to optionally include RPE character at end of each page.
- 8100I conversion changed to increase right margin value.
- RAPID changed to recognise 'C357000C' status from DISOSS. This is returned when a document is cancelled by a remote node.

Version 3.3.2

- a new message file enquiry system, controlled by transactions ZAPY and ZAPV.
- a new Control File Maintenance system, controlled by transactions ZAPX, ZAP0, ZAP1, ZAP3, ZAP5, ZAP6, ZAP7, ZAP8 and ZAPH.
- two changed transactions; ZAPL and ZAPO which invoke new programs APIPOBTN and APIPSIGN respectively.
- change to the layout and processing of the message file.
- removal of the list of signed-on users from the 'D'-record.
- mailbox integrity now controlled by MVS ENQ in CICS 1.6 and 1.7.
- optional routing of RAPID error messages to operator console.

- change the program flow to remove 'single-threading' restriction.

The RAPID User Guide contains a list of functional changes.

Migration from earlier RAPID versions

This list is for quick reference only and you should read the rest of this installation section before attempting the upgrade.

You can make use of the re-naming functions (see) if you wish - however **users of existing RAPID sub-systems (for example TELEX, MESSAGING) should not change the CICS DD names without checking the interface documentation.**

Note: If you have used the RAPID/TSO Interface you should be aware that the RAPSSEND clist has been replaced by two other CLISTS RAPSDOC and RAPSMSG which can be used to 'Send a document' or 'Send a message' respectively. RAPSSEND is now an 'Edit macro' which can be used to send the file which you are currently editing. Refer to the user Guide for more information.

Migration from RAPID version 3.4.2

1. Replace your current RAPID.LOAD library with the new one.

Migration from RAPID version 3.4.1

1. Add PPT entries for APIVER, APITEST, APIUTEST (all COBOL), APITAA, APIUAA (both maps) and PCT entries for ZAPF (program APIVER), ZAPT (program APITEST) and ZAPU (program APIUTEST)
2. Replace your current RAPID.LOAD library with the new one.
3. Replace the RAPID.PANELS library in your TSO procedure.
4. Replace the RAPID.CLIST library in your TSO procedure.
5. Run the MIGRATE job on the RAPID.JCL library.

Migration from RAPID version 3.3.5

1. Add PPT entries for APIVER, APITEST, APIUTEST (all COBOL), APITAA, APIUAA (both maps) and PCT entries for ZAPF (program APIVER), ZAPT (program APITEST) and ZAPU (program APIUTEST)
2. Replace your current RAPID.LOAD library with the new one.
3. Recustomise the parameter table (ZAPTABA) and assemble it.
4. Replace the RAPID.PANELS library in your TSO procedure.
5. Replace the RAPID.CLIST library in your TSO procedure.
6. Run the MIGRATE job on the RAPID.JCL library.

Migration from RAPID version 3.3.4

1. Add PPT entries for APIPURGE, APIVER, APITEST, APIUTEST (all COBOL), APITAA, APIUAA (both maps) and PCT entries for ZAPF (program APIVER), ZAPT (program APITEST) and ZAPU (program APIUTEST)
2. Replace your current RAPID.LOAD library with the new one.
3. Recustomise the parameter table (ZAPTABA) and assemble it.
4. Replace the RAPID.PANELS library in your TSO procedure.
5. Replace the RAPID.CLIST library in your TSO procedure.
6. Run the MIGRATE job on the RAPID.JCL library.

Migration from RAPID version 3.3.x

1. Add PPT entries for APIPURGE, APICF410, APICF411, APIVER, APITEST, APIUTEST (all COBOL), APICF4A, APITAA, APIUAA (all maps) and PCT entries for ZAP4 (program APICF411), ZAPF (program APIVER), ZAPT (program APITEST) and ZAPU (program APIUTEST)
2. Replace your current RAPID.LOAD library with the new one.
3. Recustomise the parameter table (ZAPTABA) and assemble it.
4. Replace the RAPID.PANELS library in your TSO procedure.
5. Replace the RAPID.CLIST library in your TSO procedure.
6. Run the MIGRATE job on the RAPID.JCL library.

Migration from RAPID version 3.x

1. Make sure all outstanding requests have been processed, then close the RAPID/CICS system.
2. Replace your current RAPID.LOAD library with the new one.
3. Recustomise the parameter table (ZAPTABA) and assemble it.
4. Replace the CICS PCT and PPT entries with the new entries from the install library
5. Replace your current RAPID.LOAD library with the new one.
6. Delete and redefine the ZPMESSG file (using the job from the new JCL library).
7. All usersets which have RAPID users as the obtain-id must be changed so that each userset has a unique obtain-id. The new obtain-id's should be added to the DISOSS HUP and the RAPID control file.
8. Replace the RAPID.PANELS library in your TSO procedure.
9. Replace the RAPID.CLIST library in your TSO procedure.
10. Run job MIGRATE from the RAPID.JCL library to update your RAPID control file with the new records required. Follow the instructions provided in the member.

Migration from RAPID version 2.x

1. Make sure all outstanding requests have been processed, then close the RAPID/CICS system.
2. Replace your current RAPID.LOAD library with the new one.
3. Recustomise the parameter table (ZAPTABA) and assemble it.
4. Replace the CICS PCT and PPT entries with the new entries from the install library
5. Add a HUP entry for RAPMASTR as a RAPID/API user.

6. Run job MIGRATE from the RAPID.JCL library to update your RAPID control file with the new records required. Follow the instructions provided in the member.
7. Delete and redefine the ZPMESSG file (using the job from the new JCL library).
8. Check that any RAPID/TSO procedures in your PROCLIB execute program RAPRPORT, not APIRPORT.
9. Replace the RAPID.PANELS library in your TSO procedure.
10. Replace the RAPID.CLIST library in your TSO procedure.
11. Customise the JCL records for BATCHUSR, BATCHDSN and INTRDR on the RAPID control file. Only the JOB card should be altered.
12. Note that the ZDISPRT 'serious error messages' file has now been removed and all messages are directed to CSMT.
13. All usersets which have RAPID users as the obtain-id must be changed so that each userset has a unique obtain-id. The new obtain-id's should be added to the DISOSS HUP and the RAPID control file.

The RAPID.CLIST members will need re-customising to reflect the dataset names in use at your installation.

Product Overview

RAPID provides a single generalised bi-directional link between DISOSS and other applications.

RAPID shields the application programmer from the intricacies of DIA/DCA, IBM's Office Systems Architectures, and from the asynchronous nature of DISOSS processing. It also allows the application programmer to interface to DISOSS from environments other than CICS.

RAPID consists of a series of COBOL (CICS and BATCH) and ASSEMBLER programs together with a number of maintenance and diagnostic routines.

[Figure 1](#) gives a diagrammatic representation of the RAPID system.

Input to RAPID is via two (identical) VSAM files, the MAILBOX and the MAILSACK. The Mailbox file is for interfaces that do not reside in the RAPID/DISOSS CICS region. The Mailsack is for CICS interfaces that are running in the same CICS region as RAPID.

On the outward-bound side RAPID accepts documents from DISOSS and actions them in one of two basic ways:

1. Writes them to an internal reader together with customised JCL
2. Writes them to a designated VSAM file

Details of how the documents are to be processed are held on the RAPID control file by means of a 'interface' record. Subsequent processing can then be actioned as required; RAPID, therefore, requires one, or more, interface programs for the end-user.

For RAPID, the "RAPID User Guide" describes the interfaces that are supplied as part of the RAPID package, and also explains how to create your own.

For RAPID/Base, additional documentation will have been supplied for the relevant sub-system.

Figure 1. The RAPID system



PART ONE - Installing the RAPID system

Introduction

This section is designed for use by the Systems Programmer responsible for the installation of CICS-based application software and assumes the prior knowledge of CICS table administration procedures and VSAM file creation.

Users who are migrating either from a previous RAPID release should read "[About This Release](#)" before attempting the move.

The RAPID system is supplied on an IBM standard label tape that contains the files shown in [Figure 2](#). The tape serial number is PDNnnn where nnn is a three digit number, obtained from the tape. The following chapters detail the steps required to unload the tape and prepare the CICS/DISOSS region to accept RAPID.

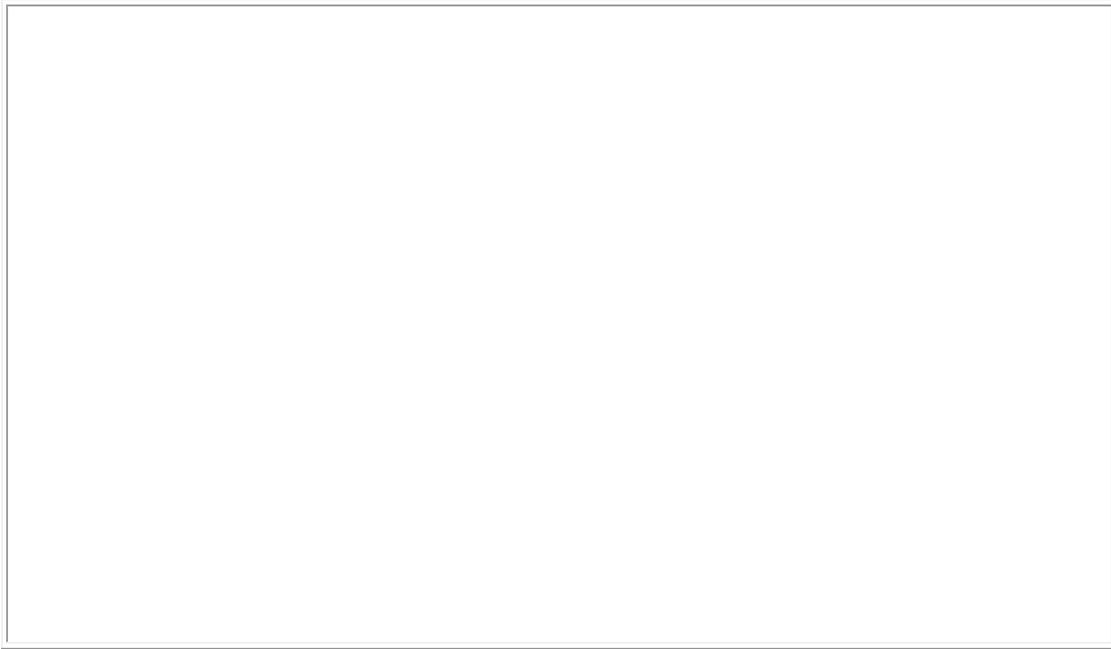
The JCL library holds set-up data for the VSAM files used by RAPID. The data contains hexadecimal characters so care should be taken to ensure that any necessary editor options are invoked that allow the modification of information without the loss of special characters. All JCL provided is example JCL only and will need tailoring to your own installation standards. All disc space parameters are based on IBM 3380 (single density) devices and may need to be converted for your own equipment.

Dataset names will need to be tailored to your installation standards as well as all points of reference to these names (such as JCL or TSO dialogues). CICS transactions, programs and filenames can be customised if required - read before returning to this section, so that you are clear about what is involved.

A 'quick reference' checklist of the installation actions is provided in "[Installation Checklist](#)".

All transaction, program, dataset and DD names referred to in this manual are the supplied default names. [Appendix A, "Customisation Record"](#) contains a form for you to record any customisation made.

Figure 2. Contents of the Installation Tape

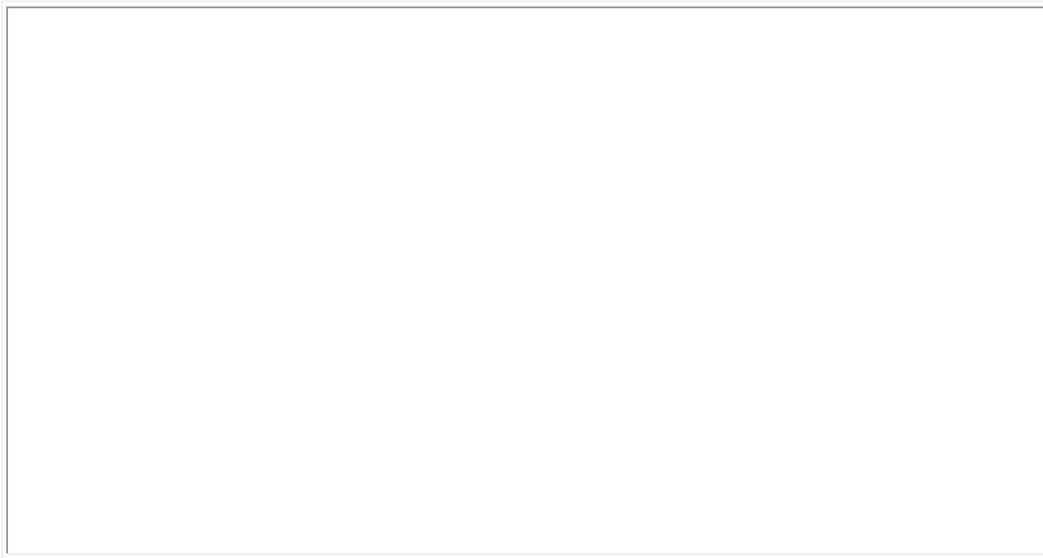


Unloading the RAPID system

The first file on the tape (dataset RAPID.INSTALL) holds the members to start the installation process.

An example of the JCL needed to unload the installation library is shown in [Figure 3](#). This will need tailoring to your installation standards.

Figure 3. Example JCL to Unload the RAPID.INSTALL Library



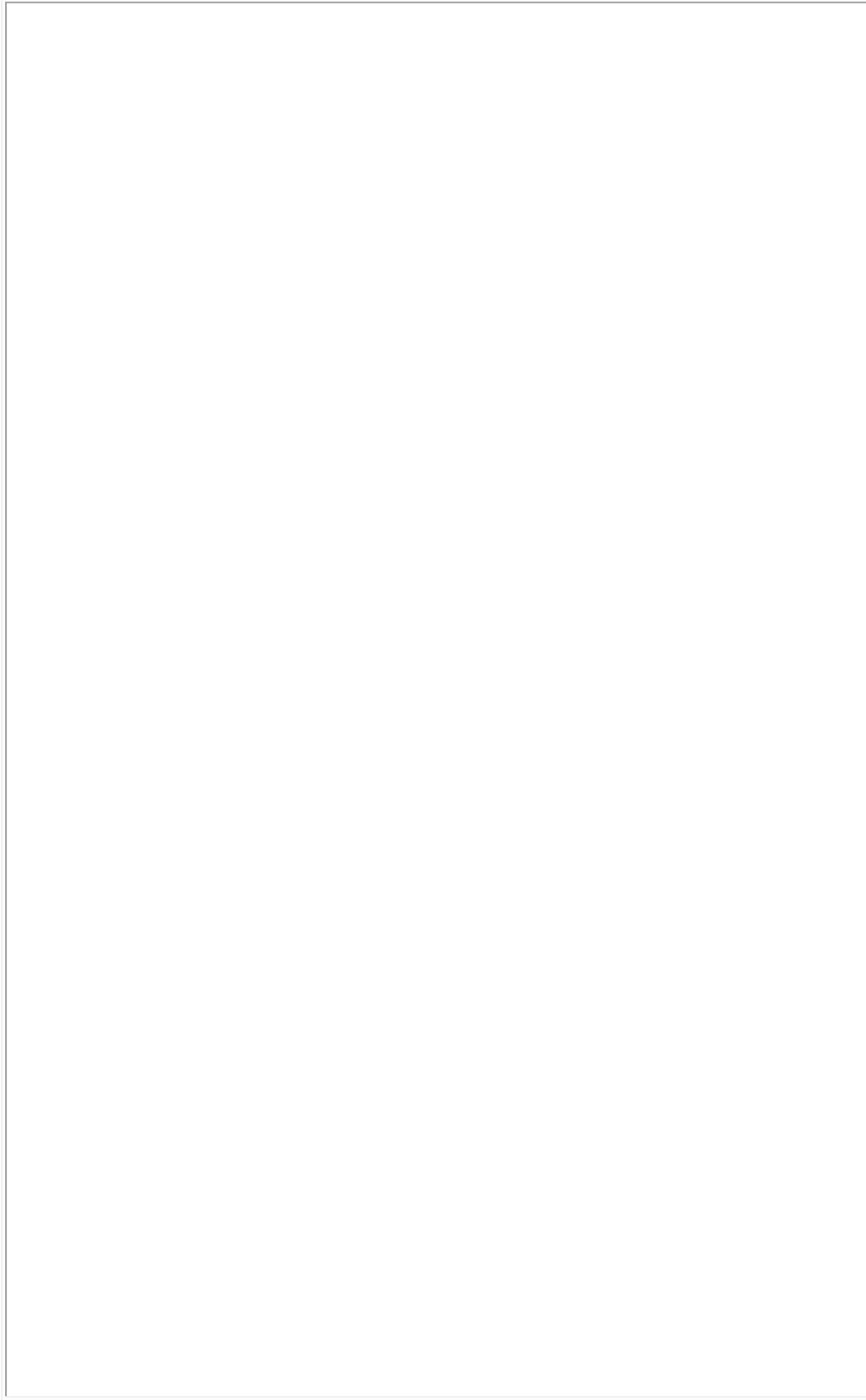
The RAPID.INSTALL library not only contains JCL to unload the core of the RAPID system, but also holds members that detail the required CICS system table updates.

The last member of this library (member **LOAD**) is the JCL needed to create and load the remaining libraries in the RAPID system. This member is set up to unload the full RAPID system. RAPID/Base users should tailor the JCL to remove references to the CLIST and PANELS libraries. After tailoring this JCL to your installation standards and successfully executing the job, the next stage in the installation process can then take place.

The JCL contained in the **LOAD** member is shown in [Figure 4](#).

Figure 4. Example JCL to Unload the RAPID Libraries





This section describes the steps involved in the creation and initialisation of the RAPID files. A description of the use of each file and any maintenance considerations are contained in .

The RAPID.JCL library unloaded from the second file of the installation tape contains a variety of members, some of which are used to create the files in the RAPID system. For each file to be created, there is a member on the JCL library whose name matches the relevant dataset's DD name.

The control file, **ZPCONTL**, will require further, installation-dependent, customisation. This is dealt with in a later section.

All the records written onto the VSAM files during the creation process are produced through program RAPSETUP except for the ZPCONTL file which uses the batch maintenance program RAPCTL. Both the RAPSETUP and the RAPCTL program are on the RAPID.LOAD library.

RAPSETUP takes in card-image data (created by a utility such as the TSO editor), through DD statement INCARDFL. The statements contain data that describes the format (record length, fields to fill in, etc) that the output record is to take. The record is then written onto a dataset referenced by DD statement OUTPUTFL and copied onto the required VSAM file using the IDCAMS REPRO facility.

RAPCTL takes in card-images, through DD statement CARDFILE, containing parameters which are formatted into records and written to the control file.

Each of the jobs that create the VSAM files have DELETE statements before the DEFINE statements on the first IDCAMS step. This allows the JCL to be used to re-create the file, for example during file maintenance. Note that this will cause a non-zero return code if the file does not already exist.

If the following jobs are being used after RAPID has been installed, the RAPID system must be shut down and the files closed in CICS before running.

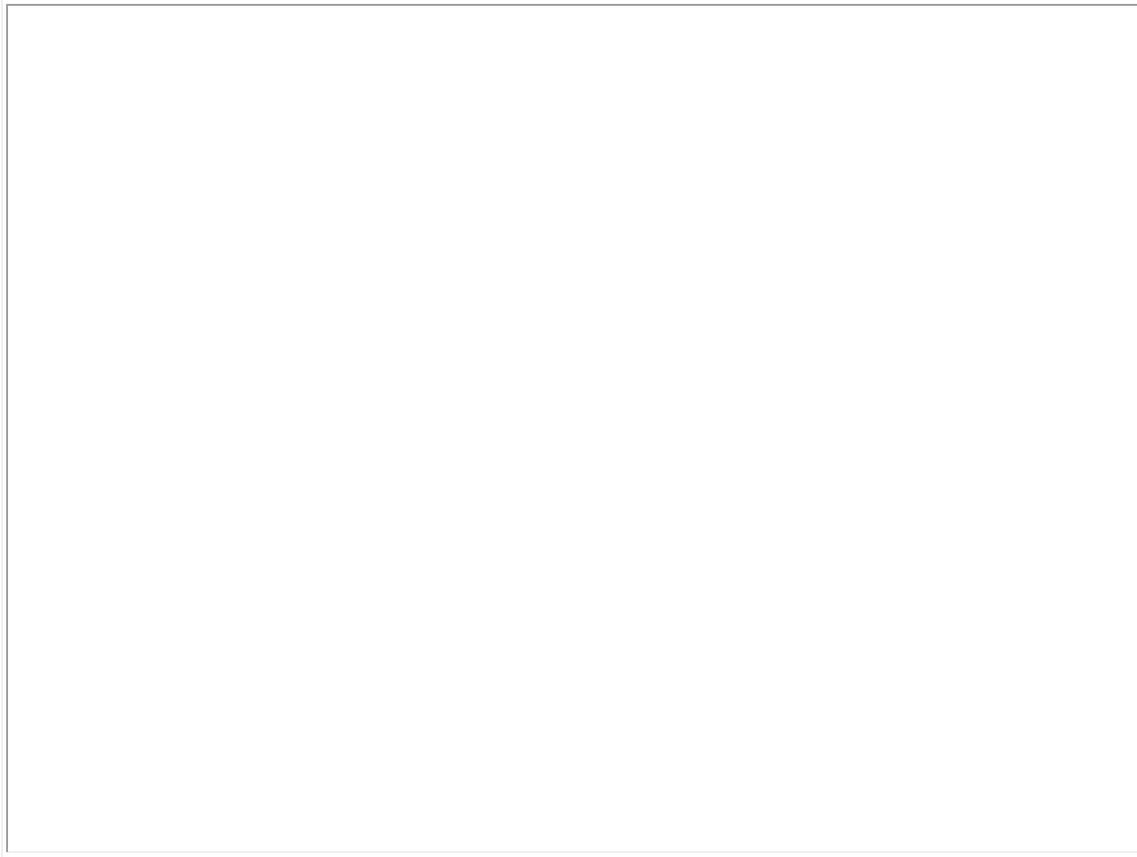
ZPCONTL - The Control File

The Control File is another VSAM KSDS. The JCL shown in [Figure 5](#) can be used to create it.

The correct input DD statement for CARDFILE should be selected before running this job.

Figure 5. Example JCL to Initialise the Control File



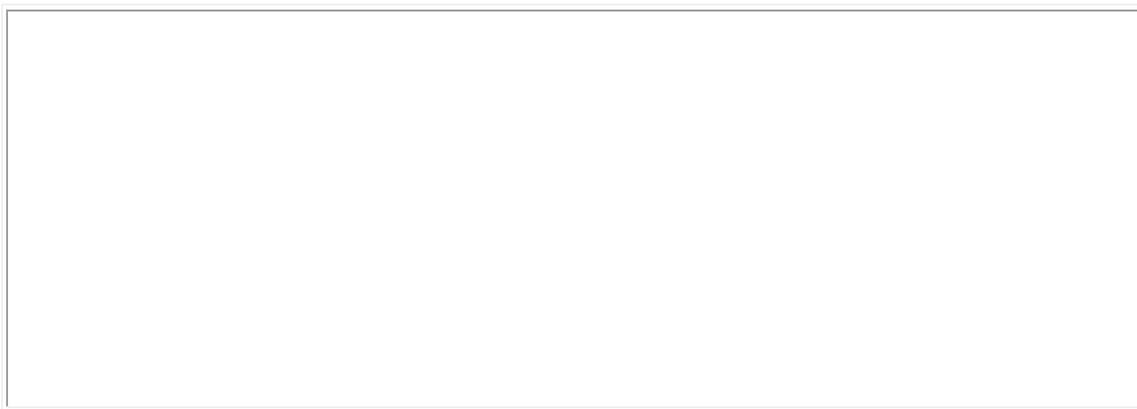


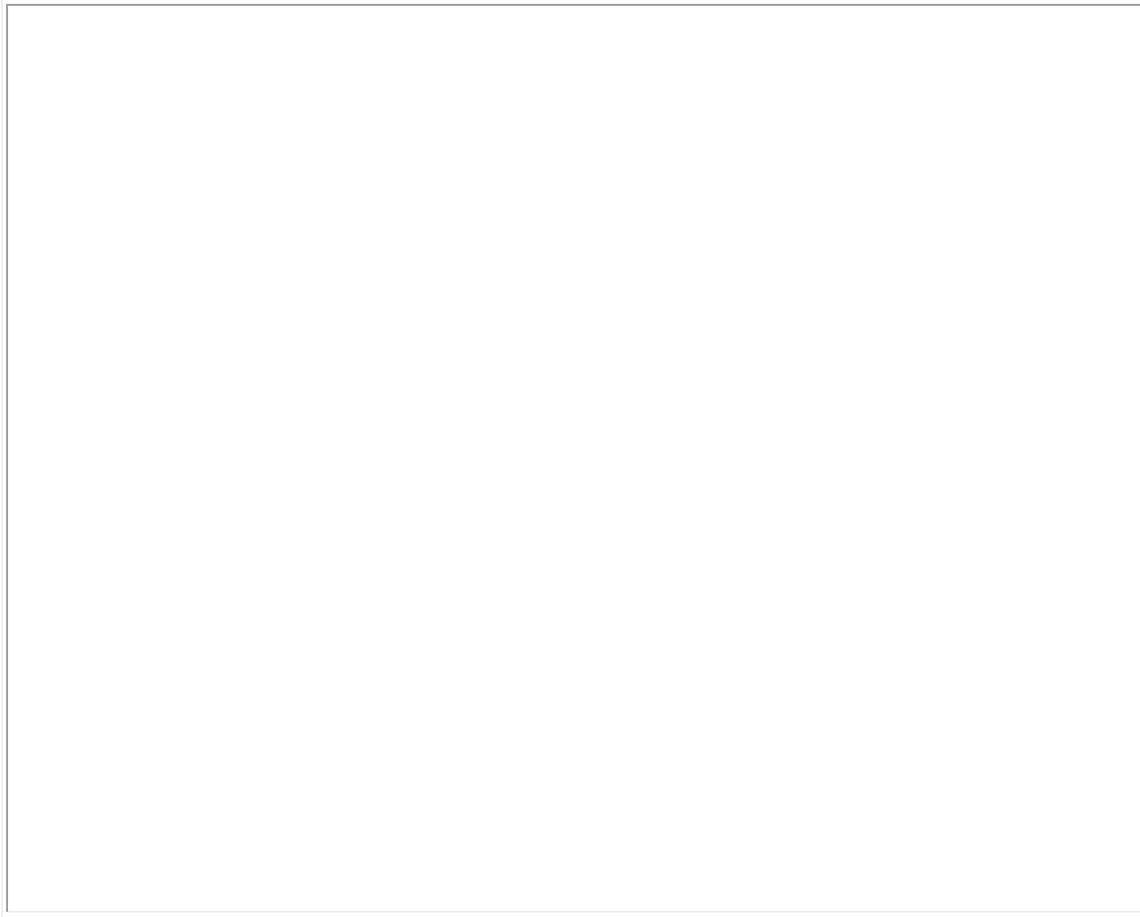
The ITT input data should NOT be used for your installation.

ZPLOOKUP - The Keyword file

The Keyword File is a VSAM KSDS. The JCL shown in [Figure 6](#) can be used to create it.

Figure 6. Example JCL to Initialise the Keyword File

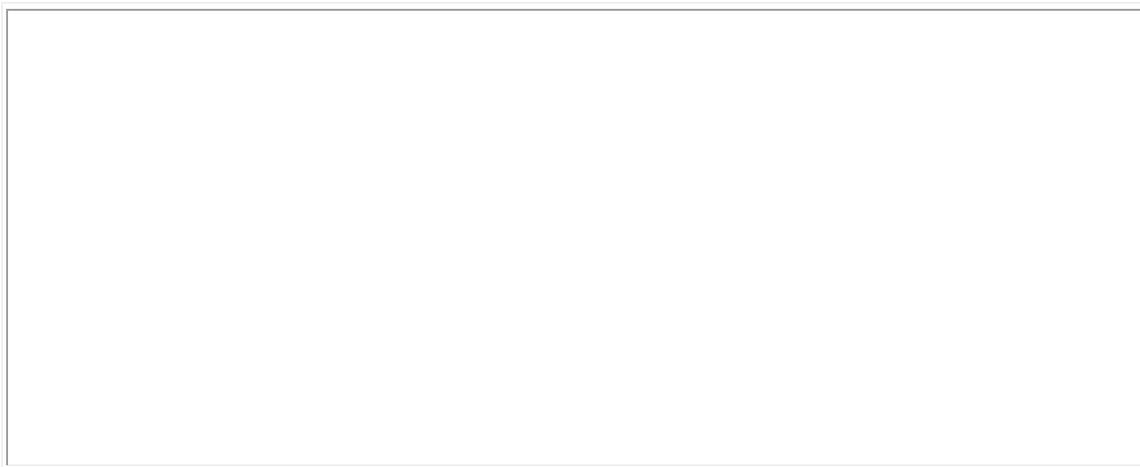


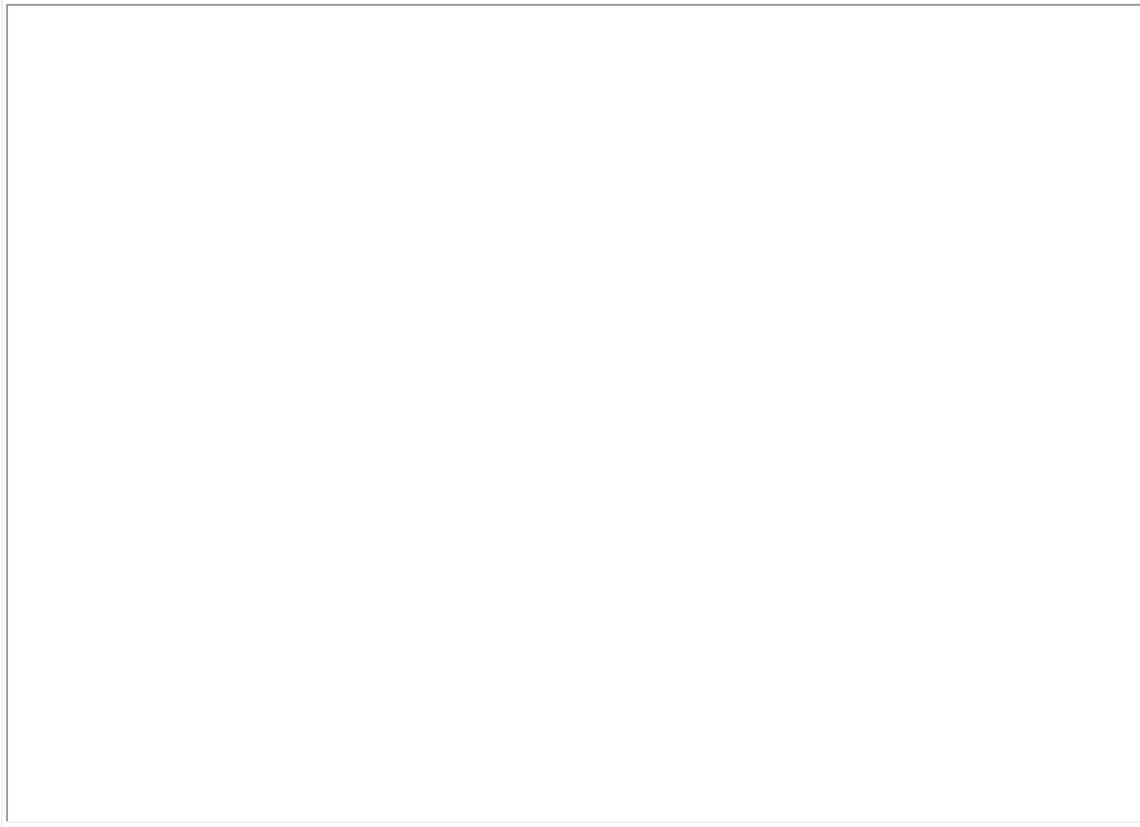


ZPMAILBX - The Mailbox File

The Mailbox File is a VSAM KSDS and the JCL shown in [Figure 7](#) can be used to create it.

Figure 7. Example JCL to Initialise the Mailbox File

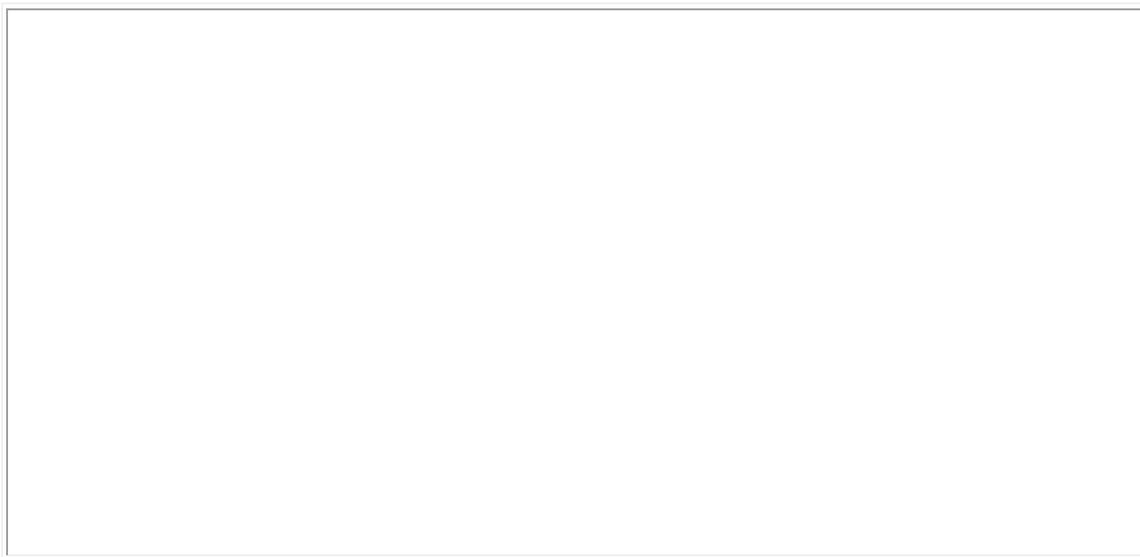


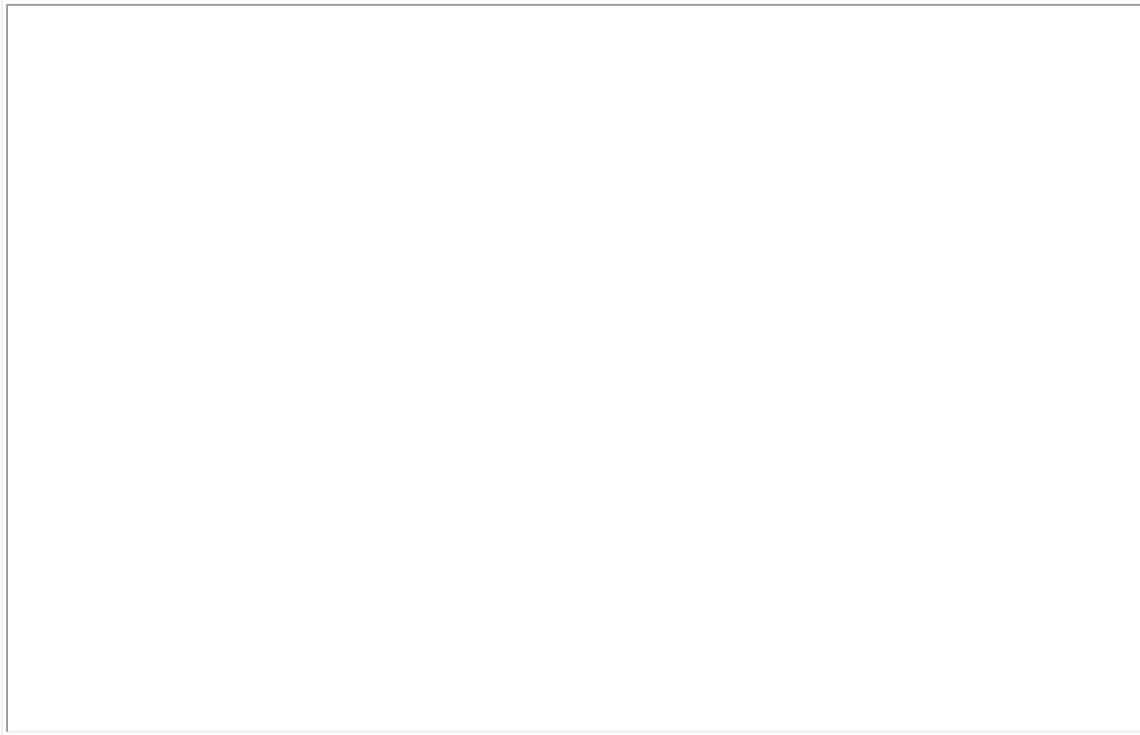


ZPMAILSK - The Mailsack File

The Mailsack File is also a VSAM KSDS. The JCL shown in [Figure 8](#) can be used to create it.

Figure 8. Example JCL to Initialise the Mailsack File

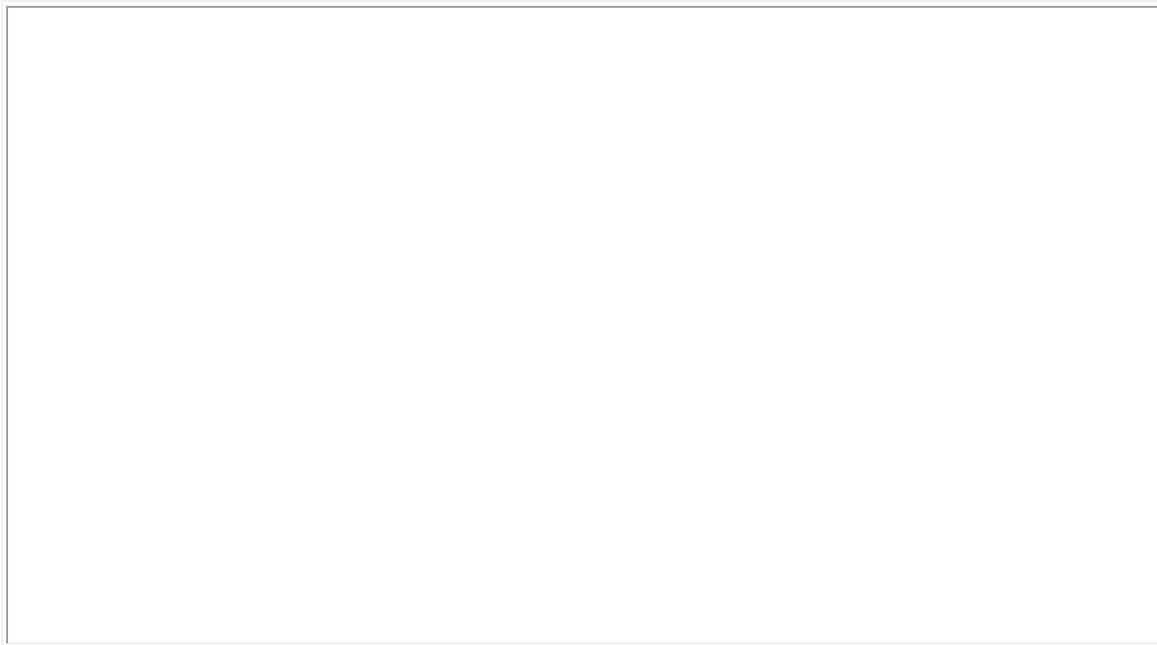


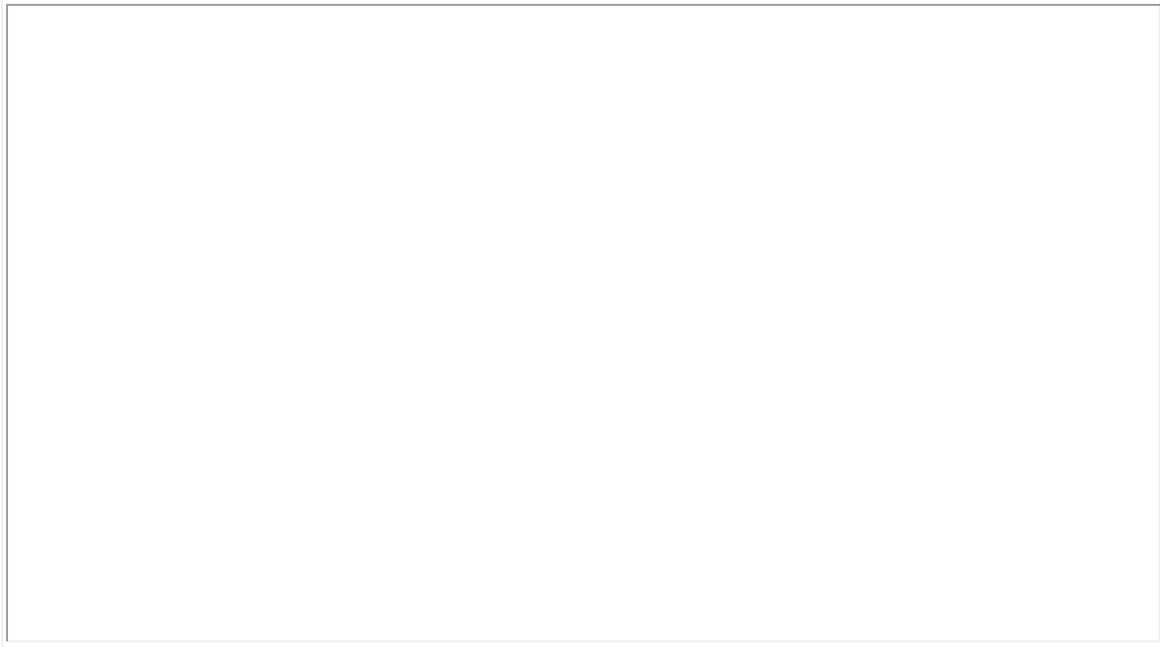


ZPMESSG - The Message File

The Message File is another VSAM KSDS. The JCL shown in [Figure 9](#) can be used to create it.

Figure 9. Example JCL to Initialise the Message dataset

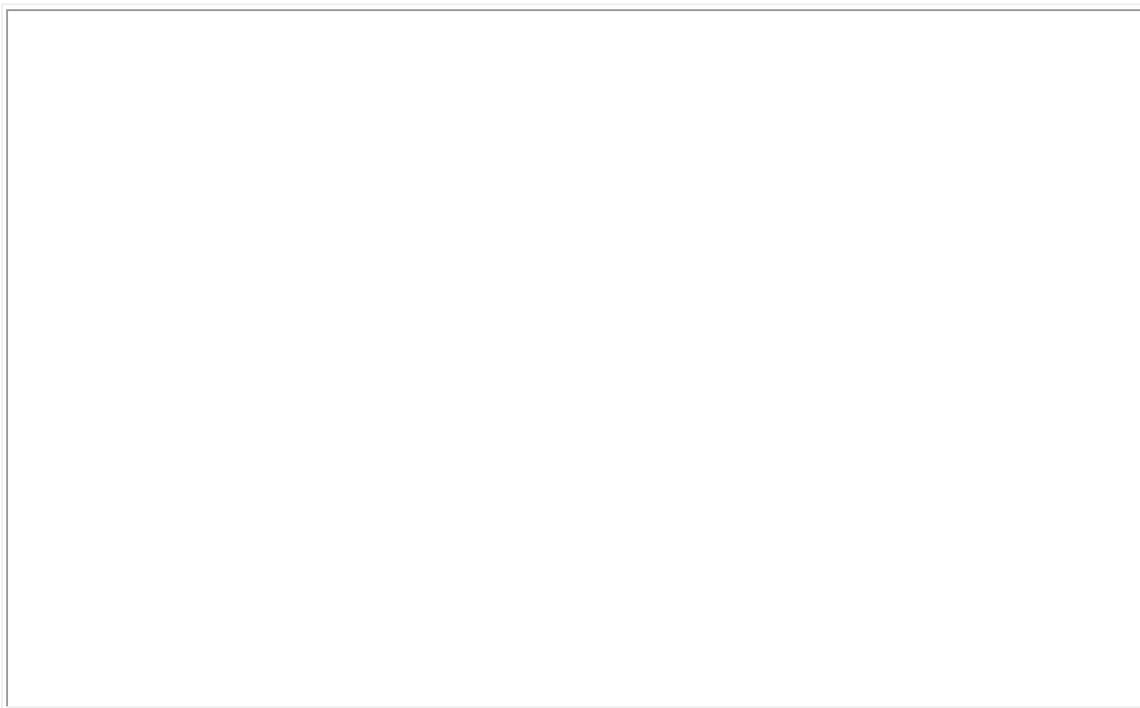


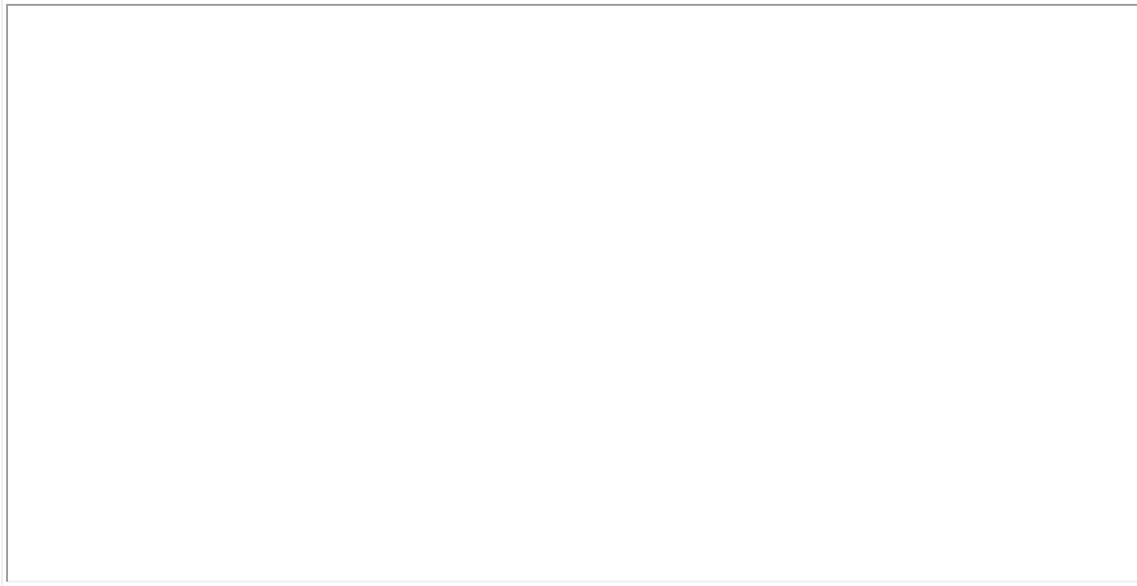


ZPRDRF - The Rationalise Distribution file

The Rationalise Distribution file is VSAM KSDS. The JCL shown in [Figure 10](#) can be used to create it.

Figure 10. Example JCL to Initialise the RDRF file





Dataset names should be tailored to your installation standards. CICS transactions, programs, and CICS file DD names can be customised if required - see . This manual refers to the supplied default names at all times.

Most of the programs used in RAPID are written in COBOL. Consequently, the DISOSS/CICS system that RAPID will run in should have language support for COBOL included in the PCP sysgen and the COBOL subroutine library available in the DFHRPL concatenation, or be an authorised library, depending on CICS system.

The CICS journal 1 (DFHJ01A) must have a minimum block size of 12K.

NOTE: The recovery mechanisms used by RAPID and DISOSS require the Dynamic Logging facilities offered by CICS. In order for the system to operate correctly the following criteria must be met:

- All transactions defined with DTB=YES
- All files defined with LOG=YES
- CICS system must be AUTO started, not COLD started.

If these conditions are not met then the system will fail unpredictably.

CICS Table Updates

The **INSTALL** library contains members that can be used to update CICS tables. A description of each member follows.

DCT

The DCT entry is for an internal reader for submitting jobs to the JES queues. This can be modified to point to your standard internal reader if preferred.

The record length of the internal reader is set at the standard 80 characters. However you can define a wider record if you wish, for example, to pass print-image documents out from DISOSS. Change the DCT entry to the new length (maximum 255) and then change the entry in the ZAPTABA program - see .

Member **APIDCT** contains the CICS DCT table updates and is shown in [Figure 11](#). Remember that TYPE=EXTRA entries must go after all the other DCT entries.

[Figure 11. DCT Table Updates](#)

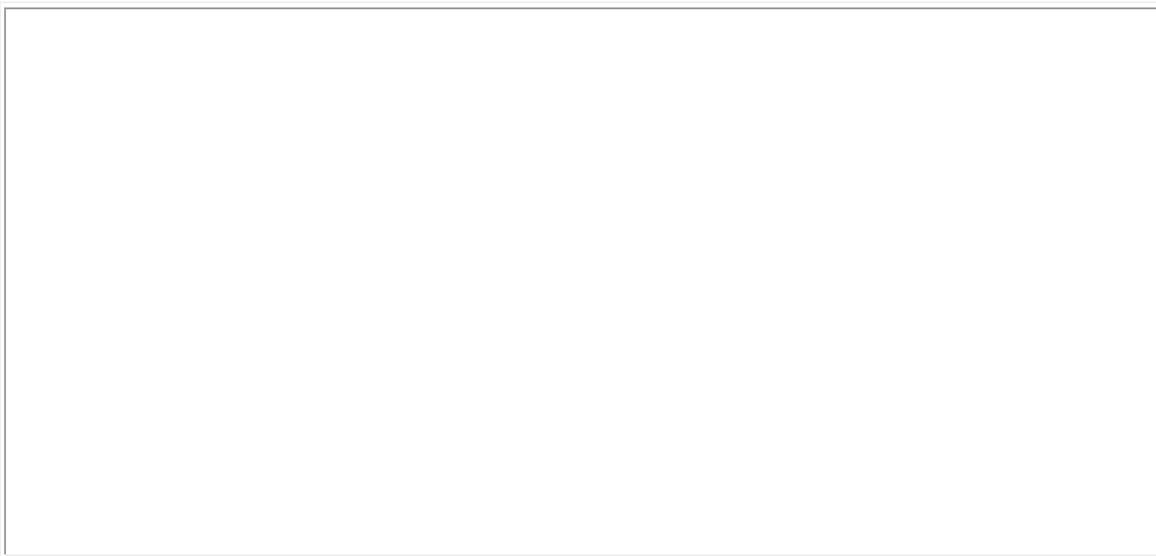


The JCL for the two files referenced here is described in "[CICS Job Control Changes](#)".

[FCT](#)

Member **APIFCT** contains the CICS FCT table updates in macro format and is shown in [Figure 12](#). The tables are also available in RDO format in member **APIFCTRD**.

[Figure 12. FCT Table Updates](#)





The JCL for the six files referenced here is described in "[CICS Job Control Changes](#)".

PCT

Member **APIPCT** contains the CICS PCT table updates and is shown in [Figure 13](#) and [Figure 14](#).

Member **APIPTRD** contains the CICS PCT RDO updates.

The entries *ZAPA* and *ZAPZ* are the manual start-up and close-down transactions. Most of the remaining transactions are invoked automatically by RAPID as and when required. Transactions can be made SPURGE if required.

Apart from *ZAPA* and *ZAPZ* all the RAPID background transactions **MUST** be in the same TCLASS (set to 1 as a default). The setting of the MAXTASK value for this class will clearly have implications on system performance. A high value will give high RAPID through-put, but will require greater CICS resource. A low value will give lower through-put, but will place less demands on CICS. These factors will need to be balanced for your own requirements.

Figure 13. PCT Table Updates - 1 of

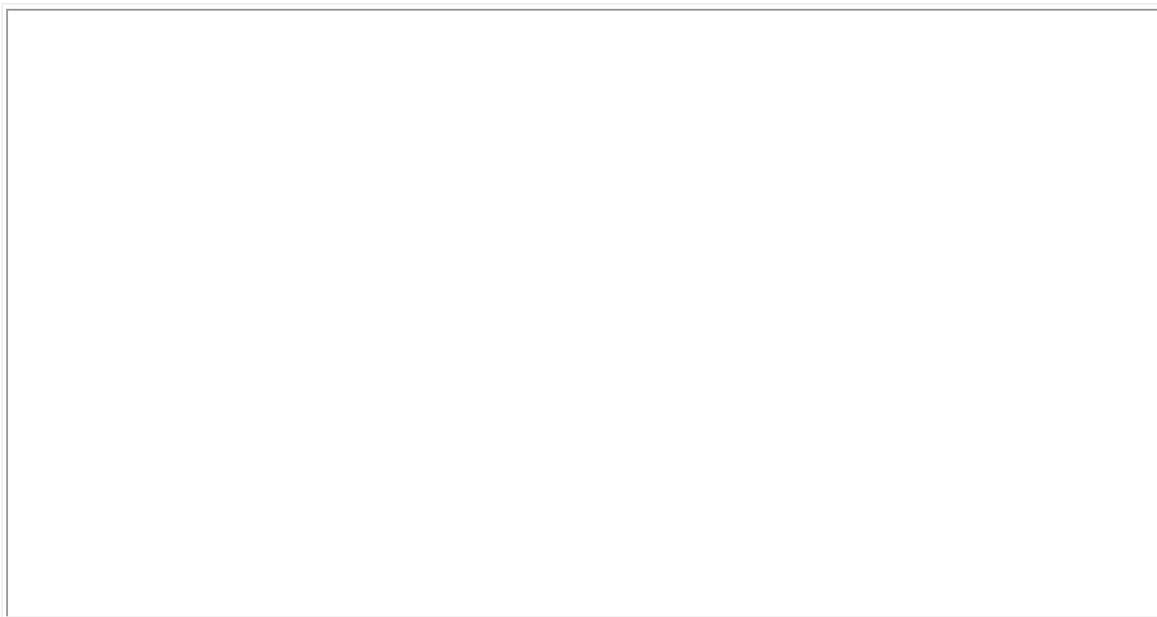
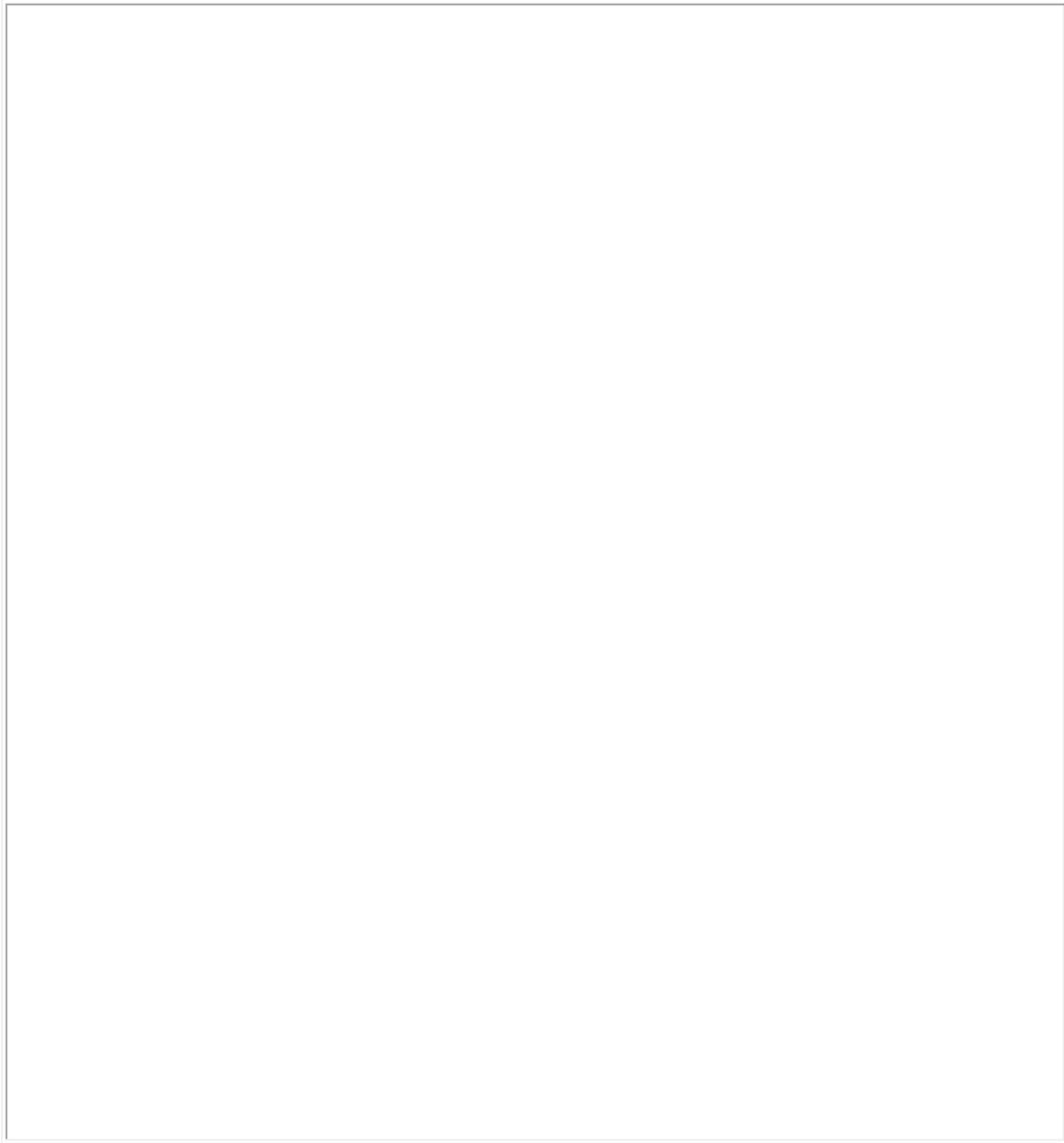




Figure 14. PCT Table Updates - 2 of



PPT

Member **APIPPT** contains the CICS PPT table updates and is shown in [Figure 15](#) and [Figure 16](#).

Member **APIPTRD** contains the CICS PPT RDO updates.

Most of the programs have been written in COBOL except for the screen maps (APICFMA, APICFVA and APICFYA) and programs ZAPTABA and APIRAPID which are in Assembler. There are no special requirements such as program residency.

[Figure 15. PPT Table Updates - 1 of 2](#)

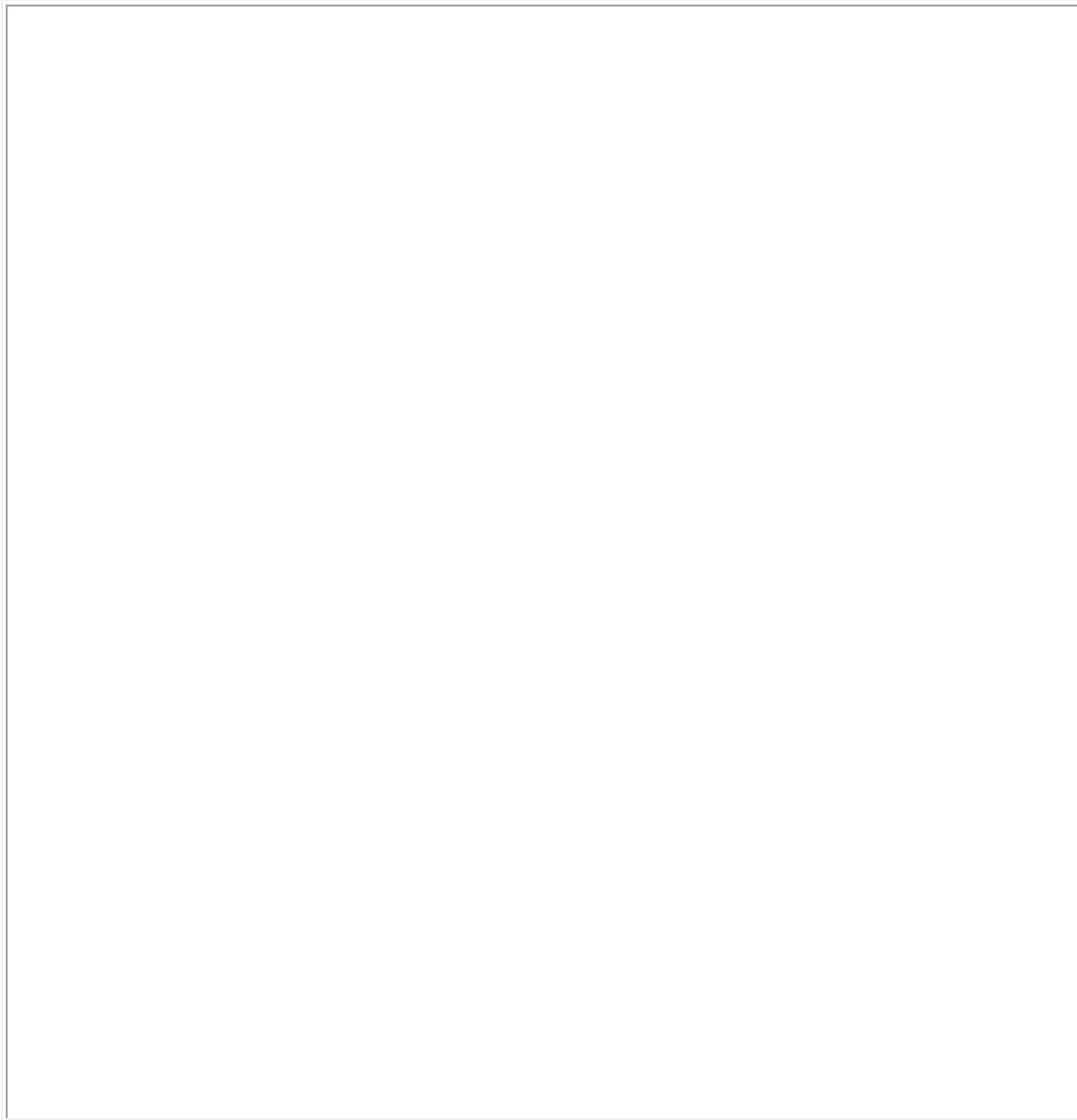
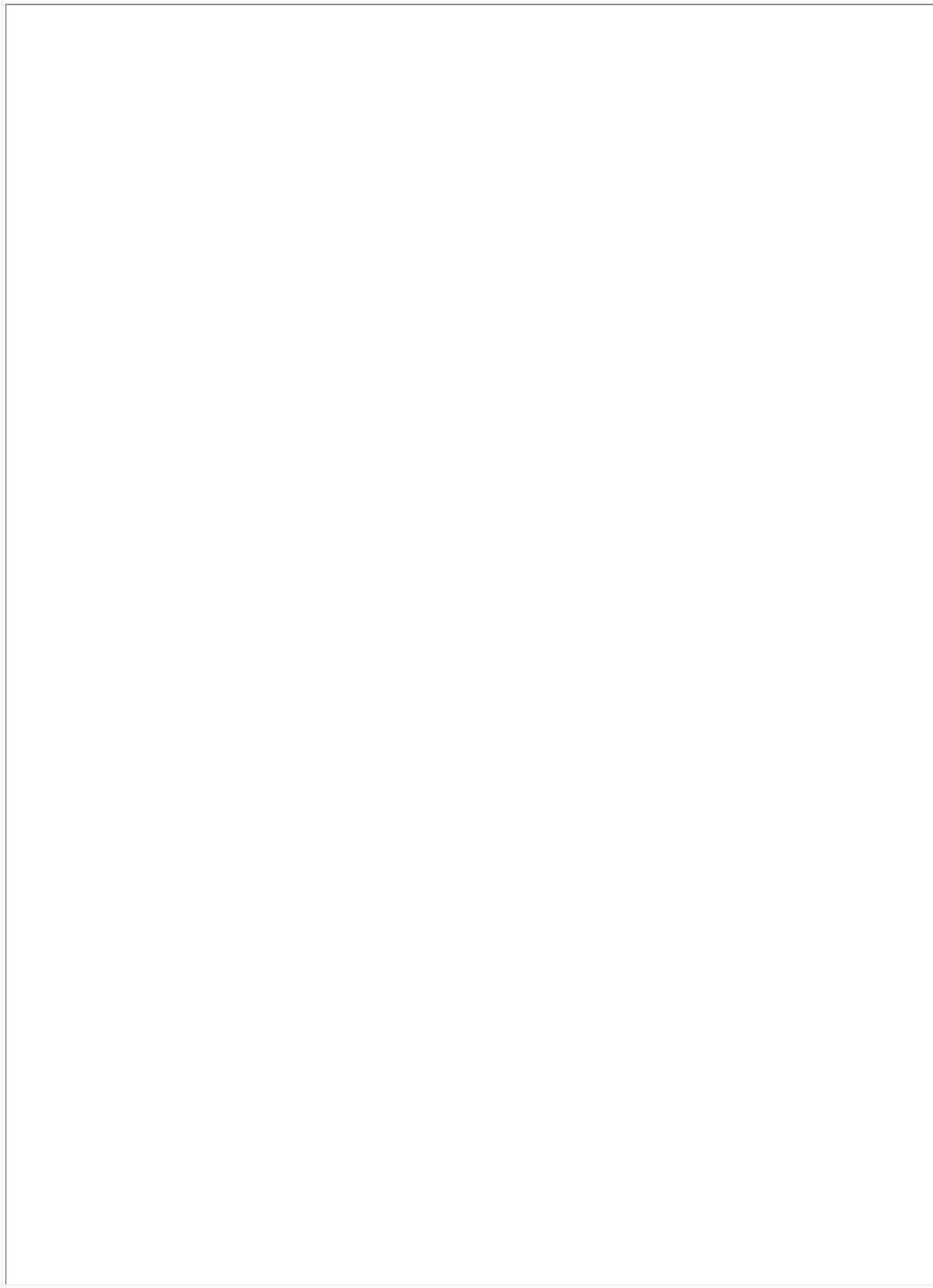


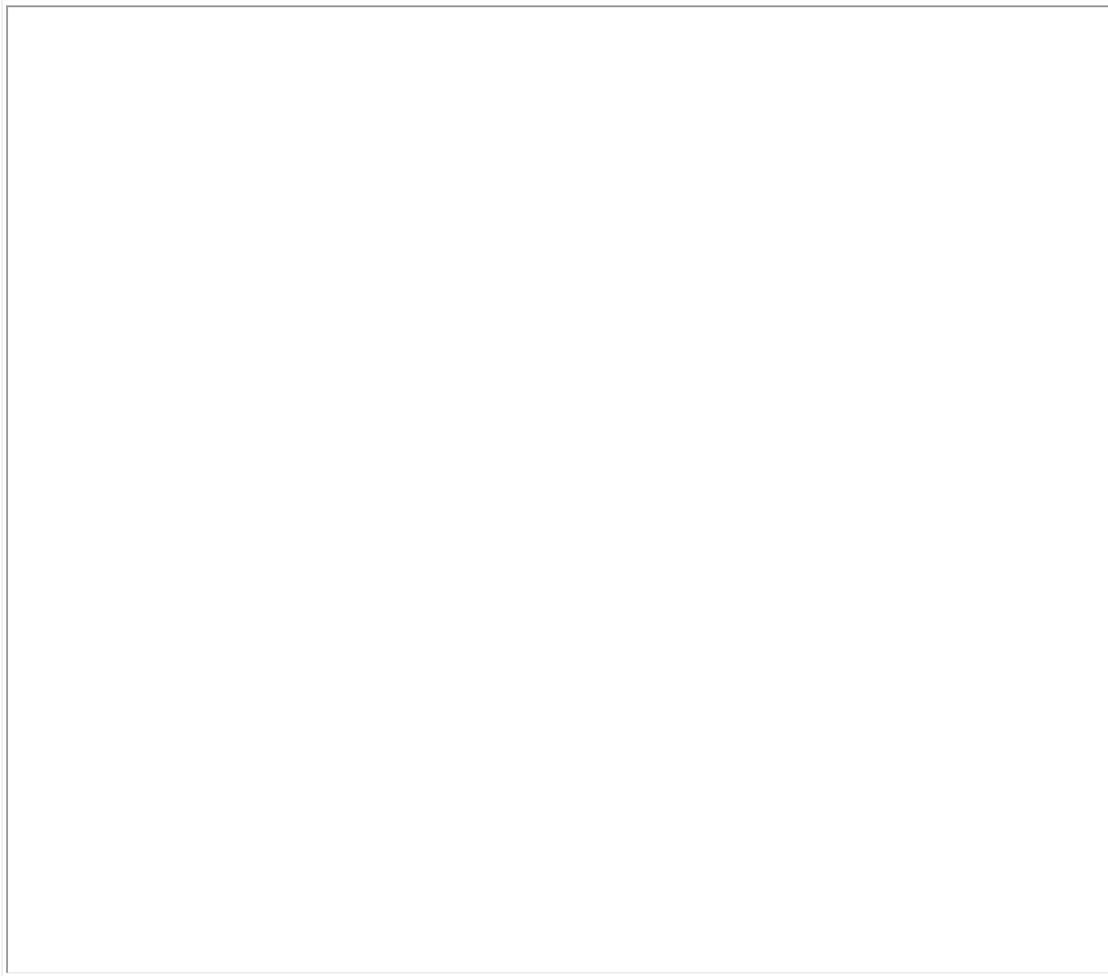
Figure 16. PPT Table Updates - 2 of 2



CICS Job Control Changes

The DD cards required in CICS are described in [Figure 17](#). The RAPID.LOAD library will have been unloaded from the installation tape.

[Figure 17. CICS DD Cards](#). Please ensure any dataset name changes are reflected here.



Updating DISOSS for RAPID

The RAPID system uses the API interface to carry out its task of document transfer in and out of DISOSS. To do this, a number of special DISOSS users must be set up.

Member HUP on the JCL library details the entries for the DISOSS users required on the HUP. The default location DSVHOST is used, so please ensure that any installation dependent changes are made before the HUP is updated; the content of the HUP member is shown in .

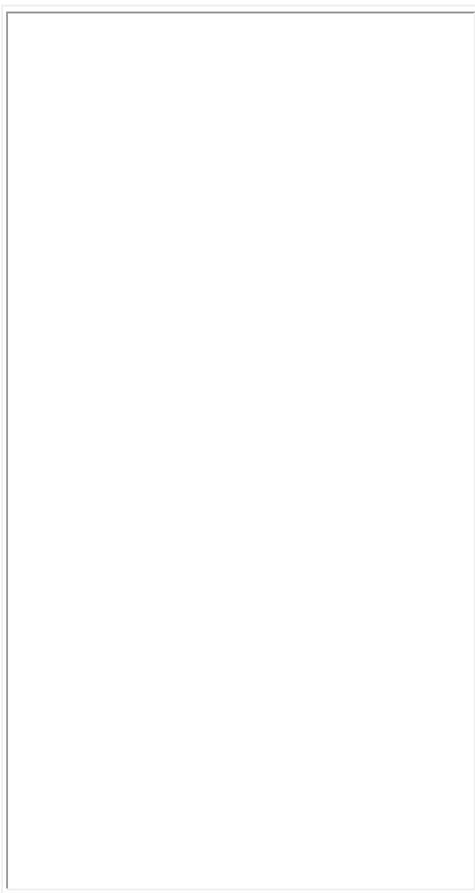
If you have customised the APIQ (or have not defined one) then you will need to make the relevant changes to the HOSTDEF for the APIQ, and add the DISOSS API modules in the PPT. Refer to your DISOSS manual for instructions on how to do this. If you require DIU tracing, the blocksize of DISOSS journal 13 should be greater than the record size of the APIQ.

The DD name of the APIQ and the DISOSS release level should be identified to RAPID in the customisation table - see .

The APIQ should also be defined with a minimum of 1 megabyte.

NOTE: The FILEOPT=ALL parameter in the DISOSS HOSTDEF is incompatible with the use of usersets. If you require usersets for a RAPID interface then either change FILEOPT to 8100 or contact your local support centre.

Figure 18. Control Statements for the HUP Utility



RAPID contains a customisation table to allow for variations from the standard system. For example, if the CICS based program names, transaction names, or DD names supplied with RAPID are unsuitable for any reason they can be customised to your own specifications.

The table also contains GCID support tables, DISOSS parameters, and other local variations. [Appendix A, "Customisation Record"](#) contains a check list for you to record any changes made.

Note: It is strongly recommended that you reassemble the parameter table (ZAPTABA on the RAPID.JCL library). This will ensure that the values which are used are suitable for your organisation.

Users of some RAPID interfaces cannot rename the CICS DD names - please check the interface documentation.

CICS Program Names

The 3-character prefix of the CICS programs, supplied with a default of 'API', can be customised to any suitable program name prefix, for example, choosing IMI will make program APIMAIL become IMIMAIL.

Decide on a 3-character name that is suitable as the program prefix to replace API then re-name all the RAPID load library members currently prefixed API (that is, all the CICS programs) with the new prefix, for example APIINIT to IMIINIT. Be careful only to alter the 3 prefix characters. You will also need to change the CICS PPT and PCT entries for these programs to reflect the new names.

The new prefix will then need to be identified to RAPID with the ZAPTABA table - see ["ZAPTABA Table"](#).

Program ZAPTABA must be prefixed with the TRANSACTION name prefix - see the next section.

CICS Transaction Names

The 3-character prefix of the CICS transaction names, supplied with the default of 'ZAP', can be customised to any suitable transaction name prefix - for example, selecting IMI will make transaction ZAPA become IMIA. Note that the customisation program ZAPTABA must also have the TRANSACTION prefix, so check that the transaction prefix you use is also suitable as a program name prefix.

The CICS PCT will need to be altered to reflect the new transaction names and the new prefix will then need to be identified to RAPID with the ZAPTABA table - see ["ZAPTABA Table"](#).

The HUP entries for the RAPID users will also need updating to reflect this transaction name.

CICS DD Names

These can be renamed as required. You cannot alter the batch program DD names. You will need to alter both the CICS FCT and JCL entries to reflect the new DD names. They will then need to be identified to RAPID with the ZAPTABA table - see "[ZAPTABA Table](#)".

GCID support

RAPID assumes UK English as the local GCID. Other users will need to set up the correct local GCID using the ZAPTABA table - see "[ZAPTABA Table](#)".

Language Support

RAPID generates confirmation messages which may be customised to the local language. The fields in the ZAPTABA table are DELMSG (delivered message), CANMSG (cancelled message), INVMSG (invalid recipient message) and ERRMSG (general error message). These messages may contain substitution fields:

- '@R' The node which detected the status
- '@N' The NAK code returned by the node
- '@D' The date the error was detected
- '@T' The time the error was detected

The DATEFORM parameter determines whether the date is substituted as DD/MM/YY (if 'UK') or MM/DD/YY (if 'US') in the error message. These values are contained in the ZAPTABA table - see "[ZAPTABA Table](#)".

DISOSS parameters

The DD name of the DISOSS APIQ (default DSVAPIQ) and the release of DISOSS in use (default 3.3) need to be identified to RAPID. Both these values are contained in the ZAPTABA table - see "[ZAPTABA Table](#)".

Internal Reader Length

The Internal Reader used by RAPID/Batch is set at 80 characters. This may cause wide documents to be truncated when passed to a batch environment. To change this, the CICS DCT entry for ZPINTRDR should be changed to the desired record length (maximum 254), and the entry in the ZAPTABA table altered accordingly - see "[ZAPTABA Table](#)".

Note: When DISOSS translates documents to print-image it will truncate the document at 132 characters (a print line) so in practice the maximum value is 133 (132 characters + print control).

Error reporting on master console

RAPID may be customised to report any errors on the master console. RAPID error messages are divided into 4 types (Information, Warning, Error and Severe error). Each type of message may be optionally logged on the master console. Each type consists of two fields a description field and a routing code. The description field identifies the type of message, and is contained on the ZAPTABA table - see "[ZAPTABA Table](#)".

X'2000' informational message (scrolling)

x'4000' error message (non-scrolling)

The routing code field identifies the destination of the message.

X'4000' master console information

x'8000' master console action

If both the description code and the routing code are set to x'0000' messages will not be written to the operator console.

Further information on WTO may be found in the IBM manual

OS SUPERVISOR SERVICES AND MACRO INSTRUCTIONS GUIDE

Automatic message file delete

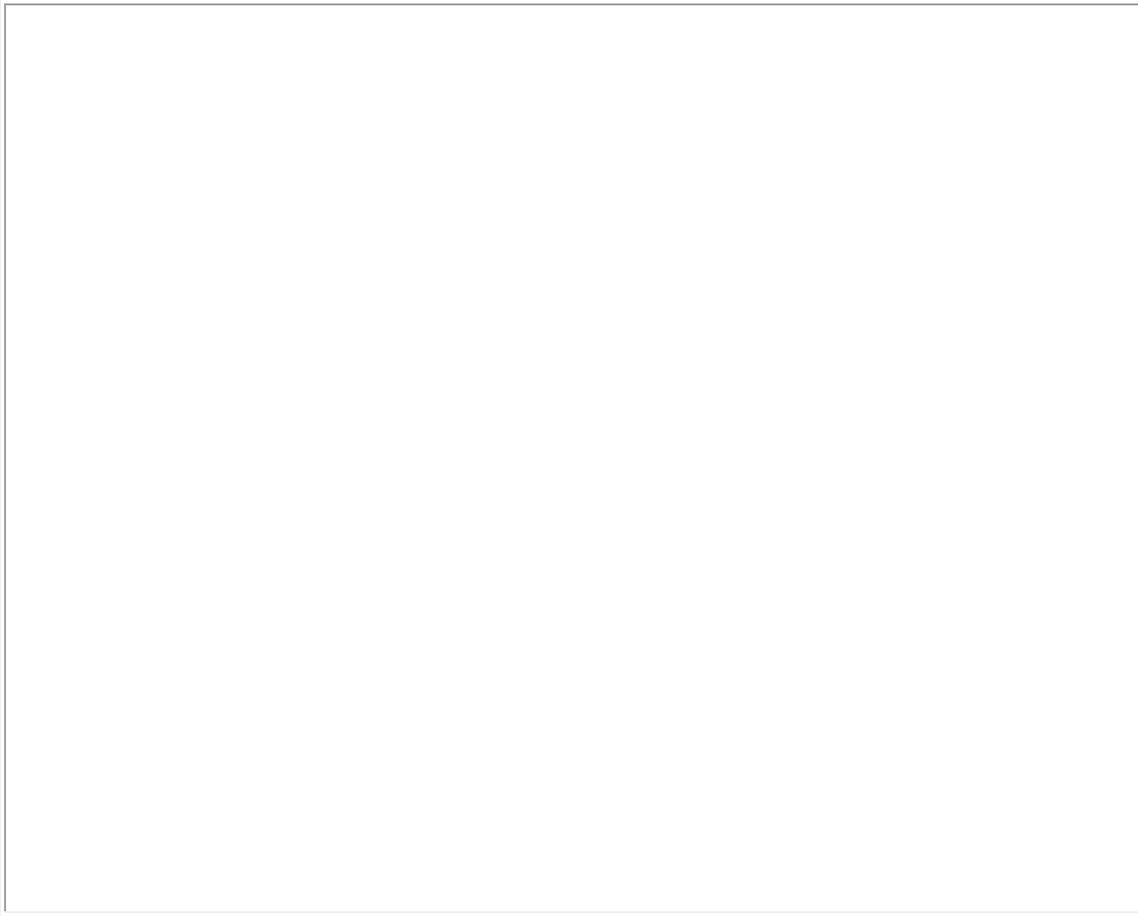
RAPID will delete completed messages from the message file after a given number of days. The RETAINDY parameter specifies the number of days to keep completed actions on the message file. The deletion is based on the last-updated date for the message.

If a value of 0 is specified then all messages, except those created today, are deleted.

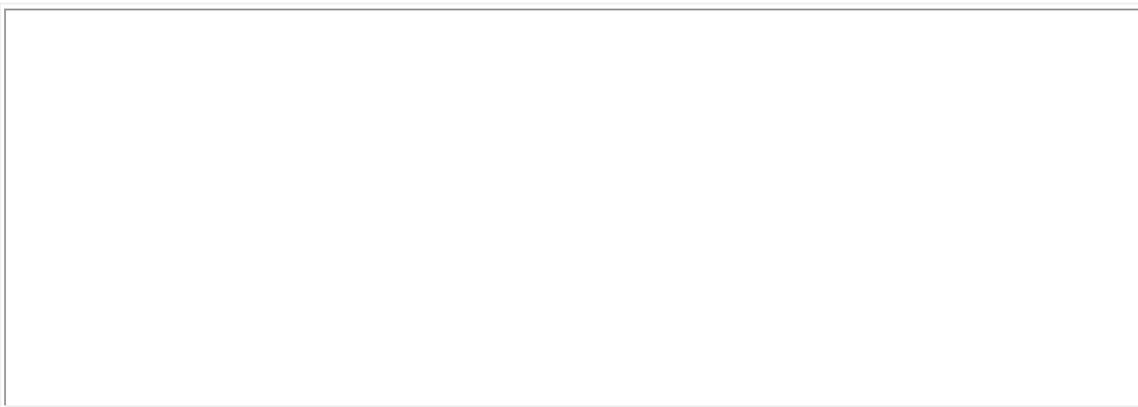
-

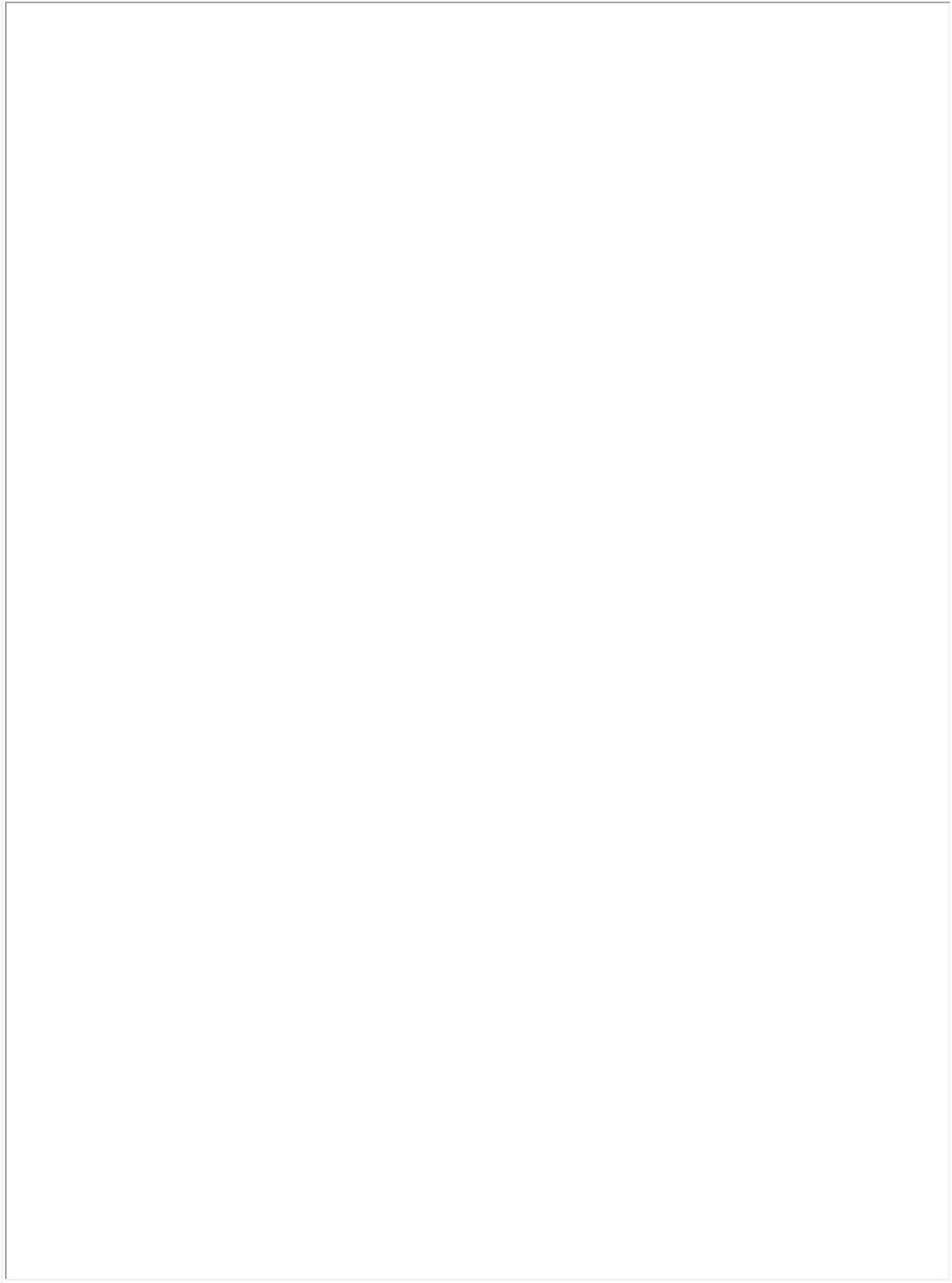
The code will need altering to reflect any customisation and is exemplified in [Figure 19](#) and [Figure 20](#).

[Figure 19. The ZAPTABA Code \(part 1 of 2\)](#)



[Figure 20. The ZAPTABA Code \(part 2 of 2\)](#)



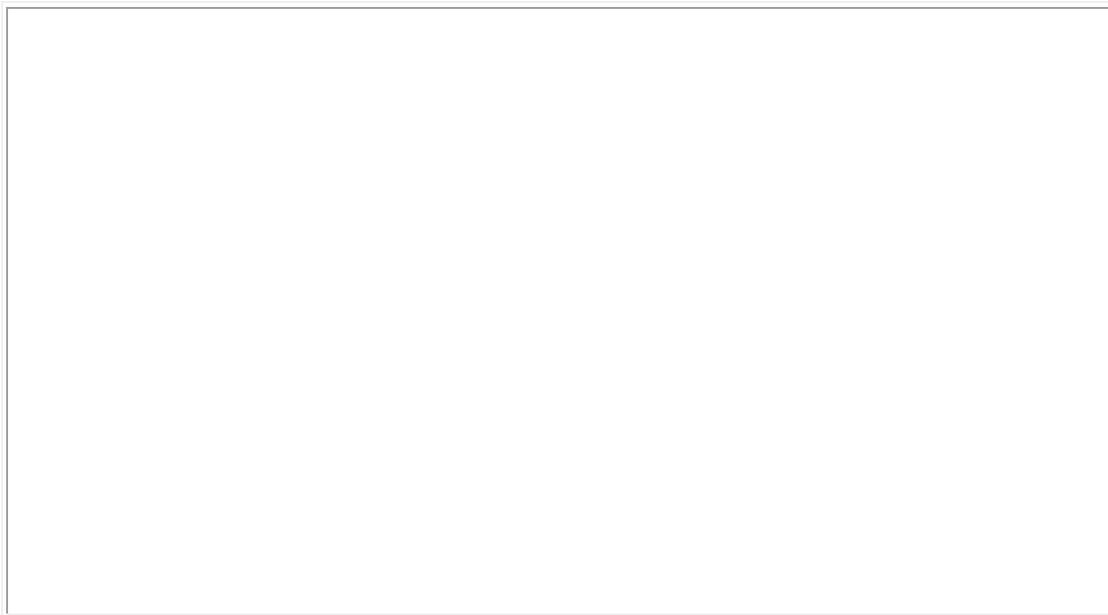


When complete the table will need to be assembled. Member ASSEMBLE on the RAPID.JCL library contains example JCL for this job, or you may wish to use your own installation standard batch assembly method.

If, as part of your customisation, you decide to rename the ZAPTABA program itself it must have the TRANSACTION prefix for your RAPID system NOT the program name prefix.

If you have changed the transaction prefix, program APIRAPID will need to be re link-edited with an INCLUDE for the new ZAPTABA after you have executed the assembly successfully. Member LINK on the RAPID.JCL library gives an example job to achieve this. The JCL is also shown in .

Figure 21. Example JCL to Re-link APIRAPID



Global Resource Serialisation (GRS) is a pre-requisite for RAPID to run in a Channel-to-Channel (CTC) connected processor environment.

RAPID controls access to the mailbox file (and also any batch outtray files) using an MVS ENQ command which the second processor will not be aware unless GRS is being used. You may need to add an entry to GRS to allow the RAPID ENQ to be transmitted to the second processor.

The ENQ issued by RAPID is in the form QNAME=RAPID RNAME=dsname of mailbox (or outtray). QNAME is 8 characters and RNAME is 44 characters, both fields are space filled.

Password Validation Exit

There is a password validation program, RAPUSRCK, available on the **ASSM** library. This is only relevant for the RAPID/Batch and RAPID/TSO subsystems, and you can ignore this section if you do not have either of them.

RAPUSRCK is invoked by the batch interface program whenever a batch request is made. The program is passed the userid, password, subject and document name which it is to validate. The program can be customised to either:

- set a non-zero return code, in which case the request is rejected by the batch interface program with return code '8xx' (where xx is the return code issued by RAPUSRCK)
- modify any of the fields passed, in which case the request continues using the returned data.

The default program supplied does not reject any requests or update any fields.

Member RAPUSRC1 contains the code required to

- Change the user-id to RPDMAIL
- Remove the password
- Add the user-id of the TSO user submitting the job to the beginning of the subject, moving the subject entered to the right, and truncating as necessary.

If RAPUSRCK is altered, assemble it using a standard batch assembly and relink to the batch interface program using member LINKEXIT on the **JCL** library.

Permitted Document Size Exit

There is a permitted document size program, RAPDOCSZ, available on the **ASSM** library. This is only relevant for the RAPID/Batch and RAPID/TSO subsystems, and you can ignore this section if you do not have either of them.

RAPDOCSZ is invoked by the batch interface program whenever a batch request is made. The program is called and returns the maximum permitted document size, in K, that the batch program will transfer and an indicator as to whether the control file can be read to override this document size value (the value on the control file will only be used if it is smaller than the default value). The program can be customised to change the returned value and indicator.

The default program supplied returns a value of 1000K (1 meg) and a control file read indicator value of 'N' (No).

If RAPDOCSZ is altered, assemble it using a standard batch assembly and relink to the batch interface program using member LINKEXIT on the **JCL** library.

Note, this program supplies a default value to the batch interface program. If, when the batch interface program is run, the RAPID control file is allocated then the value used is the one off this control file (the value is maintained on-line using the Control File Maintenance function).

Installation Checklist

You have now completed the initial installation process. As a result of carrying out the preceding steps, you should now have actioned each of the following:

- Three libraries unloaded from tape:
 1. RAPID.INSTALL.
 2. RAPID.JCL.
 3. RAPID.LOAD.
- For RAPID/TSO only, two additional libraries:
 1. RAPID.CLIST.
 2. RAPID.PANELS.
- Six VSAM files created and initialised:
 1. ZPCONTL.
 2. ZPLOOKUP.
 3. ZPMAILBX.
 4. ZPMAILSK.
 5. ZPMESSG.
 6. ZPRDRF.
- DD statements added to the CICS JCL for the above files, plus:
 1. ZPINTRDR.
 2. The RAPID.LOAD library in the DFHRPL list.

Note: The ZPMAILBX file must be defined in the CICS JCL and **not** using the DSNAMES option in the FCT.

- The following DISOSS-CICS tables updated with the data from the members on the RAPID.INSTALL library indicated:
 1. DCT with member APIDCT.
 2. FCT with member APIFCT.
 3. PCT with member APIPCT.
 4. PPT with member APIPPT.
 5. PLT updated (if required).
- The DISOSS Host User Profile (HUP) updated with the entries in member HUP on the RAPID.JCL library.

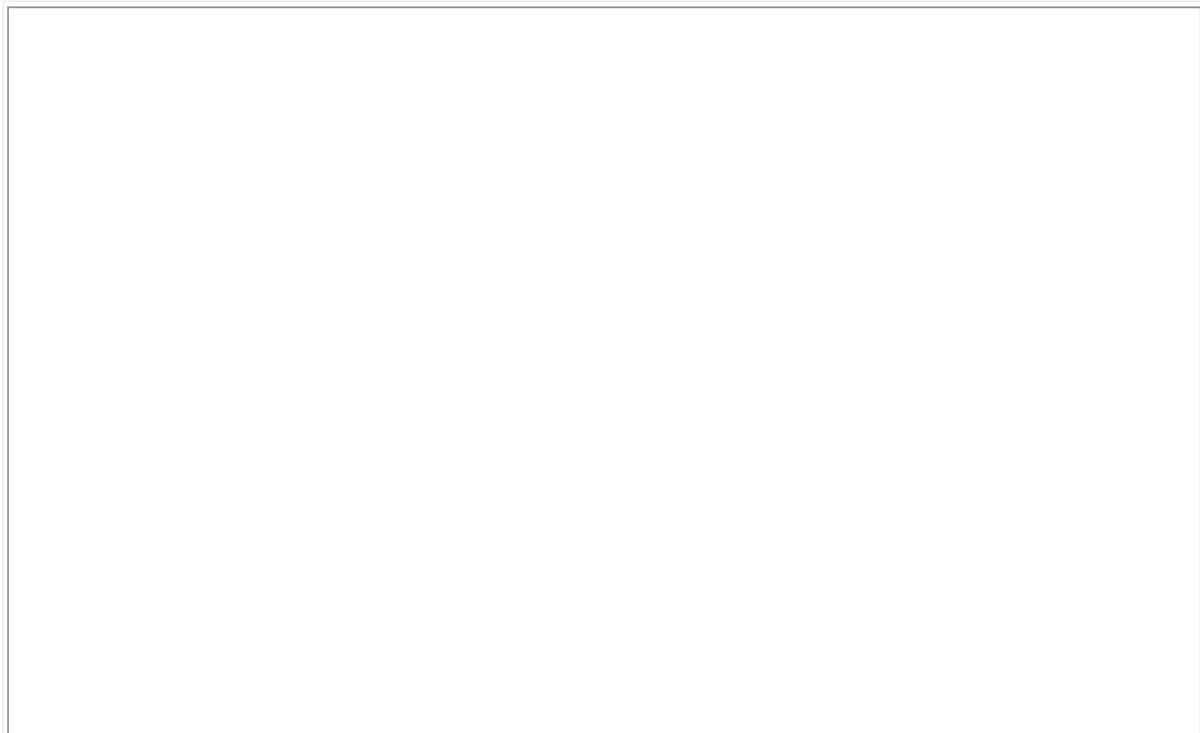
- Support for COBOL generated into CICS and COBOL subroutine libraries available to CICS.
 - Any customisation changes actioned as detailed in .
 - Ensure DISOSS Journal 13 is larger than the APIQ record size.
 - Ensure CICS Journal 1 has a minimum blocksize of 12k.
 - Ensure the DISOSS APIQ is a minimum of 1 megabyte (larger if traffic is likely to be heavy).
 - If CICS 2.1.2 is being used, ensure PTF UN56003 is applied.
-

Testing the Installation

The first step in testing the installation is to verify that all the programs, transactions and files required for RAPID are defined in CICS.

To verify the definitions:

1. Execute transaction ZAPF. This should display an 'Installation Check Log' screen which shows a list of RAPID programs, transactions and files.
2. Press ENTER to scroll forward through the list. If a required entry is not found it will be prefixed with '>>>>' and followed by an error message.
3. At the end of the list, page 06, summary totals for transactions, T=, programs, P= and files, F=, are shown. The summary total is in the form 'found total/required total'. For programs the total should be 78, transactions 29 and files



Any errors should be corrected before proceeding further.

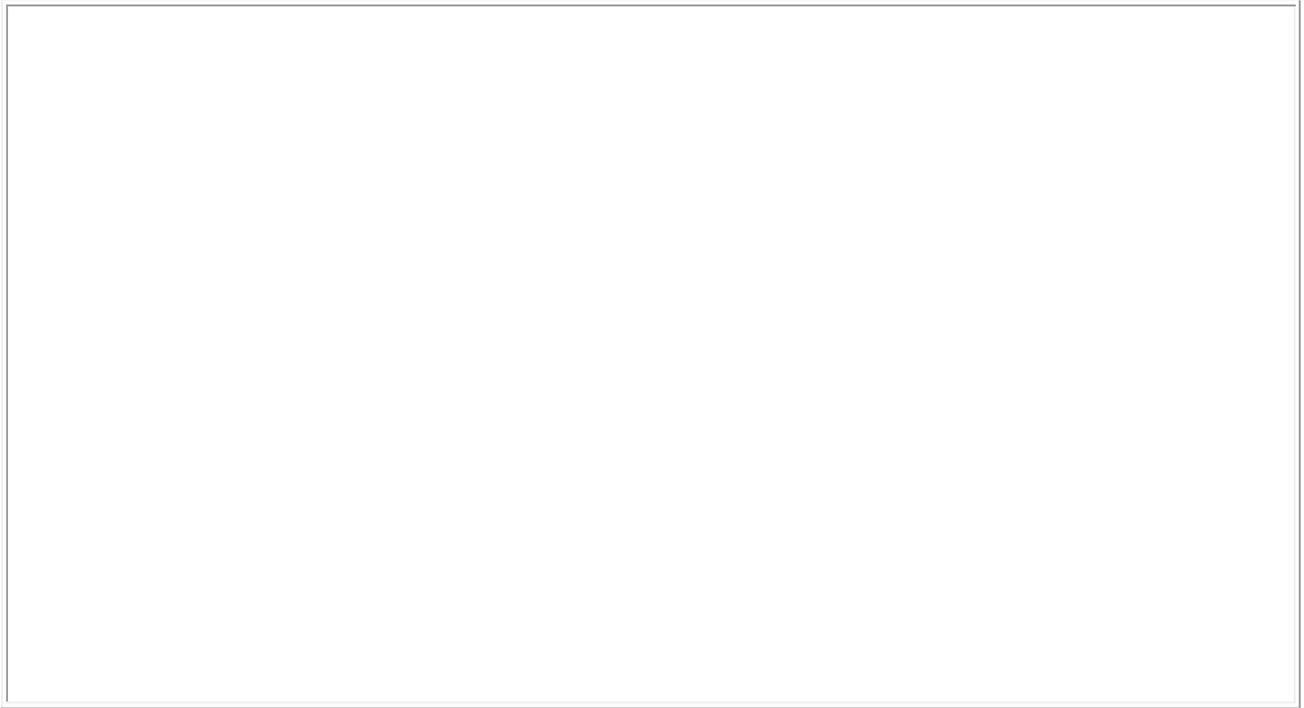
When the definition verification has been completed RAPID can be started manually by entering transaction ZAPA. The terminal will give a confirmation message that a request to start RAPID has been issued; at this stage RAPID should be active. The standard CICS active task display transaction

will show a number of transactions all prefixed with 'ZAP' executing and completing. This is RAPID operating.

The second step in testing the installation is to verify that RAPID is operating correctly.

To verify correct operation:

1. Execute transaction ZAPT. This should display a blank 'Test Send Facility' screen.
2. Press PF key 6. This should display default document text.
3. Move the cursor to the 'Send To' field and enter the required userid.
4. Move the cursor to the 'Address' field and enter the required address.
5. Move the cursor to the 'Password' field and enter the required password for 'APIMASTR'.
6. Press PF key 2 to send the document. The message 'Document sent via RAPID' should appear.
7. You can now either:
 - Press PF4 to exit the transaction
 - Alter the details and press PF2 to send the item to another userid.
 - Press PF7, PF8, or, PF9 to send different types of request.



If the recipient has not received the mail item after 10 minutes then check the RAPID message file (see "[RAPID message file browse](#)") and refer to the Problem Determination Guide.

PART TWO - Administering the RAPID system

Starting RAPID

RAPID can be started manually by entering transaction ZAPA. The terminal will give a confirmation message that a request to start RAPID has been issued; at this stage RAPID should be active. The standard CICS active task display transaction

will show a number of transactions all prefixed with 'ZAP' executing and completing. This is RAPID in action.

Closing RAPID

To shut RAPID enter transaction ZAPZ. This will disable all RAPID transactions except ZAPA (start-up transaction).

Autostart/close

RAPID may be started automatically when CICS is loaded by including the program APIRAPID in the PLTPI list. The RAPID system may also be stopped during the CICS system quiesce by including APIRAPID in the PLTSD list. A sample PLT entry is shown in [Figure 22](#).

Figure 22. PLT sample entry



RAPID timers

RAPID processes documents on the Mailsack and Mailbox on a time interval. The length of this interval is controlled by the SYSTEM record on the RAPID control file - see "[Control File Maintenance](#)".

General Information

This section provides general information about the maintenance and purpose of the various RAPID system files. Each file will be described separately.

All the RAPID files are VSAM KSDS, so they can be monitored and maintained using the standard utility IDCAMS. However, because of the way CICS and VSAM work, any action other than listing the composition and content of the file will require them to be closed in CICS (after having shut down RAPID).

The RAPID system will normally perform its own housekeeping functions on all files.

However, in common with all VSAM KSDS files, it is possible that the VSAM KSDS indices will eventually become congested. There is, unfortunately, no way to monitor the condition of the indices automatically. This means that at some point the RAPID system may fail to transfer documents.

To avoid this, it is recommended that the IDCAMS LISTCAT utility be run on a regular basis against all the files (weekly or monthly dependant upon activity).

An example of the command format is:

If the output shows a large number of CI and CA splits, or a significant reduction in the number of freespace bytes, a re-organisation of the files, using IDCAMS REPRO, is recommended.

For backup and recovery purposes, all the VSAM files can be copied and restored using the standard IDCAMS REPRO or similar utility.

ALWAYS SHUT RAPID DOWN BEFORE CLOSING ANY OF THE SYSTEM FILES.

NOTE: Because of the interconnection of the RAPID system files, if the Control File (ZPCONTL) is deleted and re-defined, all the other files should be cleaned down and re-initialised at the same time.

The JCL members, detailed in Part One , can be used for the clean-down.

ZPINTRDR - Internal reader

There is no physical file for this DDcard. It should point to a standard system internal reader.

It is used by RAPID to load various jobs onto the JES system input queue when the DISOSS EXEC route is not appropriate.

The types of job likely to be submitted are:

1. TSO message sent by DISOSS users to TSO users.
 2. Confirmation of delivery sent by RAPID back to the TSO user.
 3. Jobs sending documents out of DISOSS. Eg DATASET or DCFOUT.
-

ZPMAILBX - The Mailbox File

This file is used for passing documents to RAPID from systems that are external to the RAPID/CICS region, for example, RAPID/BATCH.

Information on this file is transferred at regular intervals into DISOSS and then deleted from the ZPMAILBX file.

Data integrity is ensured by the RAPID/BATCH and RAPID/CICS programs issuing ENQ/DEQ macros before accessing the file.

Because of the nature of this file, it can be regarded as highly volatile, so regular monitoring of its VSAM status (such as CI/CA splits) should be made.

ZPMAILSK - The Mailsack File

This file is used to hold all the requests sent to RAPID by systems running in the same CICS-DISSOSS region. The file is also used as a scratch-pad work-area by RAPID.

The file is read-write to CICS. After a record has been processed it will be deleted from the file. As a consequence, unless traffic is heavy and RAPID cannot keep pace, then this file should not fill up.

Because of the nature of this file, it can be regarded as highly volatile, so regular monitoring of its VSAM status (such as CI/CA splits) should be made.

**NOTE: This file is used as a work-area by the RAPID programs.
If the file becomes full, it should be re-organised.**

ZPMESSG - The Message File

This file is used to hold messages about the activities carried out by RAPID. All work performed by RAPID will cause a record, keyed by date and time, to be written onto this file. The file can be accessed through an online browse system - details of how to use and access the data on this file are contained in ["RAPID message file browse"](#).

Records on this file are deleted by RAPID after a period controlled by an entry on the ZAPTABA table (see) or by the batch message file program (see ["RAPID message file browse"](#)).

This file can also be regarded as highly volatile so regular monitoring of its VSAM status should be made. The file can be re-created as shown in .

The file should contain both a high and low values record.

ZPLOOKUP - The Keyword File

This file is used by the RAPID-TSO sub-system to provide a means to validate keywords to be stored with documents. DISOSS, at present, does not allow for the validation of parameters associated with a document, so the enforcement of 'In-House' standards can only be achieved by user awareness.

The online maintenance system (see ["Control File Maintenance"](#)) allows the administrator to add Keywords to this file which will then be used by the TSO sub-system to validate any Keywords entered by a user. This file can be used by any system which will create documents to be sent to DISOSS via RAPID.

If Keyword validation is not required, simply do not create any entries on the file. The file itself can be deleted if desired (beware if the CICS JCL has been updated with a DD card for this file).

The file itself is a VSAM KSDS and is keyed on the whole record (29 bytes).

ZPCONTL - The Control File

The Control File is singularly the most important file in the RAPID system. It contains all the information RAPID requires in its daily execution.

The file itself contains a number of different record types that drive the successful operation of the RAPID system.

As with most of the RAPID files the Control File is volatile and should be monitored regularly.

The contents of the file (eg. timer values, keywords etc) can be maintained online using the online maintenance system detailed in "[Control File Maintenance](#)".

ZPRDRF - The Rationalise Distribution file

This file is used to record the processing of outgoing distributions to RAPID "out-trays" designated for RDRF processing.

When a distribution for one of these "out-trays" is obtained, the RDRF file is used to check whether this document has already been distributed to another RAPID user on that system. In this way a single document, with a recipient list, can be delivered to the interface system.

The file is read-write to CICS, and RAPID will keep track of records processed on the file. After a record has been processed, it will be deleted from the file. As a consequence, unless traffic is heavy and RAPID cannot keep pace, this file should not fill up.

Because of the nature of this file, it can be regarded as highly volatile, so regular monitoring of its VSAM status (such as CI/CA splits) should be made.

DSVAPIQ - The DISOSS APIQ

This file is actually a DISOSS file, but is included here because it is the communication channel between RAPID (or PS/CICS) and DISOSS. All documents send into/out of DISOSS pass through this file before/after being placed on the DELQ. As requests are read from the file they are deleted, so the file is self-maintaining although it will need re-organisation from time to time.

If necessary the file can be re-organised, using IDCAMS REPRO, to increase the free space; ensure that the RAPID system is shut-down before closing this file.

Note: This file should not normally be re-created as documents may be lost. If, however, the APIQ is re-created then the RAPID Mailsack (ZPMAILSK) and Message (ZPMESSG) files should also be re-created at the same time.

RAPID message file browse

Rapid contains a log of all its actions in the MESSAGES file (ZPMESSG). The contents of this file are extremely useful in diagnosing problems as it holds information such as:

1. RAPID return codes
2. Next transaction to be invoked
3. Document being processed
4. Function being performed
5. DISOSS user involved
6. API return code (NAK CODE)

A full list of codes, together with further information on the major ones, is contained in the RAPID Reference and Problem Determination Guide.

The Online Browse facility

The online browse is invoked by transaction ZAPY or by entering type MESSAGE from control file maintenance. The screen will display the messages in a compressed form. By typing either I(input),A(all) or O(output) in the FUNCTION field at the top of the screen, the screen display can be restricted to show only records that are input to or output from DISOSS. The default option of ALL is used if this field is left blank.

The fields displayed on the compressed screen are|

MESG-NUM	The number associated with the message. This item consists of one character followed by a number. A character of X signifies that the request is complete, and M signifies that the request is in progress.
CREATED	The date and time that the document was created. The format is dd/mm/yy hh mm ss.
LAST UPDATE	The date and time that the document was last updated. The format is dd/mm/yy hh mm ss.
COMMAND	The command that RAPID has issued to DISOSS for a user.
REQUESTOR	The user the command has been issued for.
RETCD	The return code. This is the reason why this message was generated.
REPCD	The reply code. This indicates how far RAPID has got in the processing of the request.

Any messages which resulted in non-zero NAK codes from DISOSS will be highlighted.

By typing 'S' in the 'CMD' field, the 'FURTHER DETAILS' enquiry screen is selected. This screen gives more information on the selected request.

The additional fields displayed on the further details screen are|

DOCUMENT NAME	The name of the document or the first 44 characters of the message.
1ST RECIPIENT	The first DISOSS user specified as a recipient of the document.
CORRELATION	The user supplied correlation data in HEX format.
NEXT TRANSID	The transaction identifier of the next cics transaction which will be executed.
MAILSACK KEY	The key of the record on the mailsack in HEX format.
ISSUE PROG	The program that wrote the message to the message file.
DETECT PROG	The last program that updated the message file.
SIGNED ON USER	The DISOSS user who issued the request on behalf of the requestor. NOTE: This is not necessarily the same as the REQUESTOR.
DESTINATION	The two character destination code indicating where the request was sent.
REPLY DESTN	The destination to which the reply to the command will be sent.
CONFIRMATION	The option that specifies whether a reply was requested or not.
DISOSS NAK	The response code received from DISOSS. The code should be interpreted using the DISOSS diagnosis guide.
RECIPIENTS	The number of DISOSS users the document was sent to.
CONFIRMED	The number of recipients for whom confirmation of delivery has been received by RAPID.
OPTIONS	This field displays any options specified by the requestor when the command was issued.
REPLY DATA	This field is mainly used when there is a non zero NAK code from DISOSS. It displays the responses from DISOSS, and so it is particularly useful if an error has occurred during processing.

An in-progress record can be deleted from the further details screen by typing D in the FUNCTION field. You will be asked for confirmation before the record is deleted. To cancel a delete request change the function field back to E or press PF12. Once deleted the REPCD is set to 999 and the request cannot be recovered.

The Batch print/cleandown utility

A hardcopy of the message file contents may be produced by running the batch program RAPMGCLP. Sample JCL to do this is provided given in MESSAGES on the RAPID.JCL library. This program will produce a print-out giving information similar to the message file enquiry further details screen for each record on the message file. The program is executed with a PARM statement. Specifying a parm value of 999 will cause the program to delete all X (completed) records after they have been printed. A parm value of 000 will print the records without deleting them.

Control File Maintenance

This section describes the online facility to alter the RAPID control file. Control file maintenance program is invoked using transaction ZAPX.

Note: Not all the record types listed below will be relevant for your own installation - refer to your interface documentation for details. References to the RAPID User Guide only apply to the full RAPID system; RAPID/Base users should ignore references to records that do not apply to them.

Main Menu screen

This screen will be displayed either after initial entry, when an invalid option is selected or when RETURN is pressed.

To select an option enter the function and type in the fields at the top of the screen. For certain types of record the key of the record to be processed may also be entered. Valid types are given in the following list:

1. User
2. JCL
3. Location
4. Copylist
5. System
6. Interface
7. Message
8. RDRF

Valid functions are:

1. Enquire
2. Insert
3. Delete
4. Amend

The key field is ignored for the SYSTEM and MESSAGE screens. For the other screens the records may be browsed through using the PREV REC and NEXT REC keys.

If all the details for a record cannot be displayed on one screen then the BACKWARD and FORWARD keys may be used to view the record.

User Screen

This screen allows maintenance of the RAPID users. The RAPID User Guide gives details of how to define RAPID users via this screen.

JCL Screen

This screen allows maintenance of the JCL required for RAPID certain RAPID interfaces. The RAPID User Guide describes the JCL that is required by RAPID and gives details of how to use this screen.

Location Screen

This screen allows the location-to-dca records to be set up. The records identify the DCA levels that each different location can receive. The records are used to convert print image documents into a form the location can receive.

Note: These records are only used if a DCA level is not explicitly specified in the request.

The DCA level field may be specified as any value, however RAPID can only perform translations to DCA levels:

- 2 Final-form text
- 3 5520I text
- 7 8100I text
- 11 Revisable-form text

Copylist Screen

This screen allows you to create and amend distribution lists which may be used by RAPID when sending documents into DISOSS. The list contains a number of qualified user names (ie userid and address). When a record is created/amended the userid and address must both be entered. A user may be removed from the list by amending the record and blanking out the userid and address fields.

A distribution list is identified in a send request (using RAPIDGS3/APIGS3) by specifying a recipient of **Copylist name** with address **COPYLIST**

Note: Copylist's may be used to allow more than 50 recipients in a single send request.

System Screen

This screen allows various control fields used by the RAPID system be modified. This record may not be inserted or deleted.

The fields which may be amended are:

DCA LEVEL	The default location-to-dca translation value. See location screen for a description of the use.
PROCESSING INT	The time interval between scans of the input files (ZPMAILSK and ZPMAILBX). This interval will determine how often RAPID scans for new requests on the Mailsack and Mailbox files.
OUTTRAY INT	The interval which RAPID will wait before submitting a job to process a batch outtray. When a request is written to a batch outtray RAPID will submit a batch job to process it.
MAILBOX REQ	If this is set to N then RAPID when RAPID is restarted it will not scan the Mailbox file for requests. This may be used if the batch interface is not required by your installation.
MAXIMUM DOC SIZE	The maximum permitted size, in K, of a document transferred by the batch interface program.

The following fields are displayed on the screen but may not be amended:

MESSAGE NUM	The next message number which will be used by RAPID.
SIGNED-ON USERS	A list of the users which RAPID signed-on when it started up. This list is created dynamically from the interface records each time RAPID starts up.

Note: The first user in the **SIGNED-ON USERS** list is the RAPID internal userid. This field may be modified if required, however it should only be altered to a user who is defined on the control file. After the userid has been altered, the RAPID system **MUST** be restarted using the ZAPA transaction.

Interface Screen

This screen allows an interface records to be amended. The RAPID User Guide describes how to set-up a new interface.

Message Screen

This screen is the online message file browse facility which is described in more detail in ["RAPID message file browse"](#).

RDRF Screen

This screen allows the user to browse the RDRF file for the interface specified in the key field. BACKWARD and FORWARD may be used to page through the entries. The fields which are displayed on the screen are:

CODE	This identifies the type of entry. The code may be: 0 RDRF processing in progress for this request 5 RDRF processing completed for this request
DDN	The DISOSS assigned DDN for the document which is being processed. This field gives the name of the sender.
TIMESTAMP	The date and time the last action was taken on the document.
MESSAGE	The last message on the message file relating to the document.
NO OF RECIP	The number of recipients after RDRF processing.
DOC NAME	The document name or 1st 44 characters of message.

Most of the RAPID files are very volatile with records constantly being added and deleted. As a consequence it is possible that the files may become disorganised. This disorganisation may cause a performance degradation of RAPID or at worst it may cause RAPID to terminate.

It is, therefore, imperative that the RAPID files be monitored on a regular basis. The regularity will depend on the level of activity and this can only be judged by monitoring RAPID in its production environment.

This monitoring should be sensitive to the number of CI/CA splits, the availability of freespace and extents. Ideally, there should be no CI/CA splits and there should be no extents (i.e. the file should be within its primary space allocation).

Should the number of splits become excessive, the freespace reduce significantly or the extents become excessive, a reorganisation of the file is recommended using the standard VSAM utilities. The reorganisation itself may not, for example, remove all the extents. In these cases the files should be backed up, the cluster definition changed, the cluster deleted and redefined and the file restored from the backup.

For further information on the RAPID system files, their contents and volatility, see the RAPID Installation and Administration Guide chapter titled The RAPID System Files.

Appendix B. Program Temporary Fixes (PTFs)

There are no PTFs available with this release of RAPID.

Index

Appendix. Index

Special Characters

[A](#) [B](#) [C](#) [D](#) [I](#) [K](#) [L](#) [M](#) [O](#) [R](#) [U](#) [Z](#)

Special Characters

- [\(1\)](#)
- creation JCL [\(64\)](#)

A

- APICF410 [\(26\)](#)
- APICF411 [\(27\)](#)
- APICFMA [\(79\)](#)
- APICFVA [\(80\)](#)
- APICFYA [\(81\)](#)
- APIDCT [\(73\)](#)
- APIFCT [\(74\)](#)
- APIGS3 [\(132\)](#)
- APIPCT [\(75\)](#)
- APIPOBTN [\(15\)](#)
- APIPPT [\(78\)](#)
- APIPSIGN [\(16\)](#)
- APIPURGE [\(25\)](#)
- APIRAPID [\(83\)](#), [\(92\)](#), [\(101\)](#)
- APITEST [\(19\)](#)
- APIUTEST [\(20\)](#)
- APIVER [\(18\)](#)
- ASSEMBLE [\(90\)](#)

B

- BATCHDSN [\(34\)](#)
- BATCHUSR [\(33\)](#)

C

- Control File [\(116\)](#)
 - creation JCL. [\(50\)](#)
- COPYLIST .

D

- Disoss APIQ [\(121\)](#)
- DSVAPIQ [\(120\)](#)

I

- INSTALL ([41](#)), ([71](#))
- Internal Reader ([40](#)), ([87](#))
- Internal reader ([105](#))
- INTRDR ([35](#))

K

- Keyword File ([114](#))
 - creation JCL. ([54](#))

L

- Language Support ([86](#))

M

- Mailbox ([17](#)), ([103](#)), ([107](#))
 - description of ([37](#))
- Mailbox File
 - creation JCL ([58](#))
- Mailsack ([102](#)), ([109](#))
 - description of ([39](#))
- Mailsack File
 - creation JCL ([62](#))
- Message File ([111](#))
 - creation JCL ([66](#))

O

- out-trays ([119](#))

R

- RAPCTL ([46](#))
- RAPIDGS3 ([131](#))
- RAPMASTR ([30](#))
- RAPMGCLP ([128](#))
- RAPREPORT ([32](#))
- RAPSETUP
 - description of ([45](#))
- Rationalise Distribution file ([118](#))
 - creation JCL ([70](#))

U

- User Exits
 - Password Validation ([94](#))
 - Permitted Document Size ([95](#))

Z

- ZAP1 ([6](#))
- ZAP3 ([7](#))
- ZAP4 ([28](#))
- ZAP5 ([8](#))
- ZAP6 ([9](#))
- ZAP7 ([10](#))

- ZAP8 ([11](#))
- ZAPA ([76](#)), ([97](#)), ([99](#))
- ZAPF ([21](#)), ([96](#))
- ZAPH ([12](#))
- ZAPL ([13](#))
- ZAPO ([5](#)), ([14](#))
- ZAPT ([22](#)), ([98](#))
- ZAPTABA ([24](#)), ([72](#)), ([82](#)), ([84](#)), ([85](#)), ([89](#)), ([91](#)), ([112](#))
- ZAPU ([23](#))
- ZAPV ([3](#))
- ZAPX ([4](#))
 - Message file enquiry ([130](#))
- ZAPY ([2](#))
 - online browse facility ([127](#))
- ZAPZ ([77](#)), ([100](#))
- ZPCONTL ([115](#))
 - creation JCL. ([48](#))
 - the control file ([43](#))
- ZPINTRDR ([88](#)), ([104](#))
- ZPLOOKUP ([113](#))
 - creation JCL. ([52](#))
- ZPMAILBX ([106](#))
 - creation JCL ([56](#))
- ZPMAILSK ([108](#)), ([122](#))
 - creation JCL ([60](#))
- ZPMESSG ([29](#)), ([31](#)), ([110](#)), ([123](#))
 - Message file enquiry ([125](#))
- ZPRDRF ([117](#))
 - creation JCL ([68](#))