MPX-32 ™

Revision 3.5

Software Release Notes

April 1990

324-004400-700

ENCORE

LIMITED RIGHTS LEGEND

Encore Computer Corporation
6901 West Sunrise Boulevard
Fort Lauderdale, Florida 33313

History

The *MPX-32 Revision 3.5 Software Release Notes*, Publication Order Number 324-004400-700 was printed in April 1990.

The *MPX-32 Release 3.4 U03 Software Release Notes*, Publication Order Number 324-004420-600 was printed in January 1989.

The *MPX-32 Release 3.4 U02A Software Release Notes*, Publication Order Number 324-003520-000 was printed in July 1989.

The *MPX-32 Release 3.4 U02 Software Release Notes*, Publication Order Number 324-004400-500 was printed in January 1989.

The *MPX-32 Release 3.4 U01 Software Release Notes*, Publication Order Number 324-004400-400 was printed in August 1988.

The *MPX-32 Release 3.4 Software Release Notes*, Publication Order Number 324-004400-300 was printed in February 1988.

This document contains the following pages:

Title page
Copyright page
iii/iv through vi
1-1 through 1-2
2-1 through 2-4
3-1 through 3-31
4-1 through 4-16
5-1 through 5-2
6-1 through 6-3
7-1 through 7-9

CONTENTS

1 OVERVIEW

This document, the *MPX-32 Revision 3.5 Software Release Notes* (SRN), is provided to customers purchasing or upgrading to Revision 3.5. It describes changes to the Mapped Programming Executive (MPX-32) operating system from Revision 3.4 U03 to Revision 3.5 as well as providing other revision-specific information.

This SRN appears on the Master System Distribution Tape (SDT) in file SRN_MPX32_3.5. This upper/lowercase file can be printed from TSM or examined online.

Read this SRN before attempting to install or use MPX-32 Revision 3.5. In addition to providing an overview of the release, this document contains critical, late-breaking information that is not documented elsewhere.

SUMMARY OF CONTENTS

This document contains the following information:

- overview of this document
- contents of this release
- product information
- differences between Revisions 3.4 U03 and 3.5
- installation instructions
- resolved SPRs
- special considerations for Revision 3.5

DOCUMENTATION CONVENTIONS

NOTE: The documentation conventions concerning type fonts apply to only the hardcopy version of this Software Release Notes document.

These are the documentation conventions for this document:

Messages and Examples

Text shown in `this distinctive font` indicates an actual representation of a system message or an example of actual input and output. For example,

```
VOLUME MOUNT SUCCESSFUL
```

or

```
TSM>!ACTIVATE MYTASK
TSM>
```

Italics

Lowercase italic letters identify a generic element that must be replaced with a value. For example,

VOL=*volname*

means replace *volname* with the desired volume name. For example,

```
VOL=VOL1
```

In this SRN, unlike the rest of the MPX-32 manual set, italic letters also indicate document titles.

2 CONTENTS OF THIS RELEASE

Information and model numbers on the various MPX-32 Revision 3.5 distribution components are given in this chapter. The following are components of this distribution:

BINARY DISTRIBUTION

All customers receive one of the following Master System Distribution Tapes (SDTs). The binary tape you have received matches your hardware environment and media format.

Program Binary, SDT bootloadable format

Model Number	Hardware Environment	Media Format
1401-3303	32/67*xx*, SelCONNECTION	9-track, 1600 BPI
1401-330B	32/67*xx*, SelCONNECTION	9-track, 6250 BPI
1401-4303	32/87*xx*,32/97*xx*, 32/2030	9-track, 1600 BPI
1401-430B	32/87*xx*,32/97*xx*, 32/2030	9-track, 6250 BPI
1401-5303	32/2040	9-track, 1600 BPI
1401-530B	32/2040	9-track, 6250 BPI

Revision Date

April 1990.

SOURCE DISTRIBUTION

Source customers receive one of the following source tapes, as well as the binary described above.

Program Source, Volume Manager format

Model Number	License Type	Media Format
1401-3323	Source Single Facility	9-track, 1600 BPI
1401-332B	Source Single Facility	9-track, 6250 BPI
1401-3393	Source Single System	9-track, 1600 BPI
1401-339B	Source Single System	9-track, 6250 BPI

Revision Date

April 1990.

SOFTWARE UPDATE SERVICE (SUS)

Software Update Service (SUS) is a maintenance and enhancement service for software and related documentation. It is available to licensed users on an annual pre-paid basis.

All customers receiving SUS receive one of the following tapes, selected according to hardware environment and desired media format:

Program Binary, SDT bootloadable format

Model Number	Hardware Environment	Media Format
1401-3353	32/67*xx*, SelCONNECTION	9-track, 1600 BPI
1401-335B	32/67*xx*, SelCONNECTION	9-track, 6250 BPI
1401-4353	32/87*xx*,32/97*xx*, 32/2030	9-track, 1600 BPI
1401-435B	32/87*xx*,32/97*xx*, 32/2030	9-track, 6250 BPI
1401-5353	32/2040	9-track, 1600 BPI
1401-535B	32/2040	9-track, 6250 BPI

In addition, SUS source customers receive one of the following tapes:

Program Source, Volume Manager format

Model Number	License Type	Media Format
1401-3323 or	Source Single Facility	9-track, 1600 BPI
1401-3393	Source Single System	9-track, 1600 BPI
1401-332B or	Source Single Facility or System	9-track, 6250 BPI
1401-339B	Source Single System	9-track, 1600 BPI

DOCUMENTATION DELIVERED WITH THIS PRODUCT

The following documents are delivered with MPX-32 Revision 3.5. Only source customers receive the technical manuals.

Reference Manual Volume I
Publication Order Number 323-001551-600

Reference Manual Volume II
Publication Order Number 323-001552-600

Reference Manual Volume III
Publication Order Number 323-001553-600

Reference Manual Volume IV
Publication Order Number 323-001554-600

Technical Manual Volume I
Publication Order Number 322-001551-500

Technical Manual Volume II
Publication Order Number 322-001552-500

Performance Guide
Publication Order Number 321-006360-000

MPX-32 Revision 3.5 Software Release Notes
Publication Order Number 324-004400-700

Reference Manual Set

Volume I contains an overview of the MPX-32 operating system. It discusses task structure and execution, scheduling rules, resource allocation, I/O data structures, and available system services.

Volume II contains job control language (JCL) and the following utilities: Operator Communications (OPCOM), Volume Manager (VOLMGR), Compress, rapid file allocation, shadow memory, ANSI tape labeling, online help, sort/merge, command line recall and edit, and terminal definition (TERMDEF).

Volume III contains installation and system generation (SYSGEN) information, system administrator services, system utilities, and Volume Formatter.

Volume IV contains information about unsupported software, including MPX-32 Demo.

Each volume of the reference manual set contains a complete set of appendices, a glossary of MPX-32 terminology, and an index covering the four volumes.

Technical Manual Set (Source customers only)

Volume I contains information about system task descriptions, system tables and data structures, system macros, SYSGEN, batch tasks, system trace, system initialization, and the IPU.

Volume II contains detailed descriptions of resident modules and handlers.

Performance Guide

Contains information about performance features and tuning.

3 PRODUCT INFORMATION

PRODUCT DESCRIPTION

MPX-32 is a disk-oriented, multiprogramming operating system designed to run on CONCEPT/32 computers. It supports concurrent execution of multiple tasks in interactive, batch, and real-time environments. MPX-32 provides memory management, terminal support, multiple batch streams, and intertask communication.

MPX-32 Revision 3.5 supports the following CONCEPT/32 computers:

- 32/67*xx*
- 32/87*xx*
- 32/97*xx*
- 32/20*xx*
- SelCONNECTION

MPX-32 Revision 3.5 may be installed and used by first-time MPX-32 customers as well as those upgrading from a previous revision of MPX-32.

MPX-32 Revision 3.5 was based on MPX-32 Revision 3.4 U03 and includes all features of that revision as well as new features specific to Revision 3.5.

RELEASE HISTORY

The following is the release history of this product:

DESCRIPTION	LEVEL	DATE
Enhancement Release	3.5	April 1990

PACKAGE FORMAT

All MPX-32 Revision 3.5 tapes are standard 1/2" 9-track magnetic tapes recorded at either 1600 BPI or 6250 BPI, shipped according to your desired format.

DISTRIBUTION TAPE FORMAT

Contents of the SDT

Following is the truncated VOLMGR output. For information about the exact sizes, refer to the file SDT.CONTENTS.

```
            *** V O L U M E   M A N A G E R  3.5***
REWIND          (TAP)
*****  BOT  *****
SDT MASTER          BOOTSTRAP LOADER  06EC (HEX) BYTES
SDT MASTER         @SYSTEM (SYSTEM) MSTRALL       407 (DEC) BLOCKS
SDT        *** EOF ***
SDT MASTER         @SYSTEM (SYSTEM) MSTREXT       428 (DEC) BLOCKS
SDT        *** EOF ***
SDT MASTER         @SYSTEM (SYSTEM) MSTROUT       429 (DEC) BLOCKS
SDT        *** EOF ***
SDT MASTER         @SYSTEM (SYSTEM) J.VFMT        130 (DEC) BLOCKS
SDT        *** EOF ***
SDT MASTER         @SYSTEM (SYSTEM) J.MOUNT        30 (DEC) BLOCKS
SDT MASTER         @SYSTEM (SYSTEM) J.SWAPR        56 (DEC) BLOCKS
SDT MASTER         @SYSTEM (SYSTEM) VOLMGR        220 (DEC) BLOCKS
SDT        *** EOF ***
SDT        *** EOF ***
SAVE               @RT0             (SYSTEM          )INSTALLSDT
SAVE               @RT0             (SYSTEM          )MPXID3.5
SAVE               @RT0             (SYSTEM          )SDT.CONTENTS
SAVE               @RT0             (SYSTEM          )SRN_MPX32_3.5
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE               @RT0             (SYSTEM          )ADMOUNT
SAVE               @RT0             (SYSTEM          )AMOUNT
SAVE               @RT0             (SYSTEM          )ASTAT
SAVE               @RT0             (SYSTEM          )AVOLM
SAVE               @RT0             (SYSTEM          )COMPRESS
SAVE               @RT0             (SYSTEM          )DEVINITL
SAVE               @RT0             (SYSTEM          )HELP
SAVE               @RT0             (SYSTEM          )HELPT
SAVE               @RT0             (SYSTEM          )J.ACCNT
SAVE               @RT0             (SYSTEM          )J.ATAPE
SAVE               @RT0             (SYSTEM          )J.DSCMP
SAVE               @RT0             (SYSTEM          )J.DTSAVE
SAVE               @RT0             (SYSTEM          )J.FORMF
SAVE               @RT0             (SYSTEM          )J.HLP
SAVE               @RT0             (SYSTEM          )J.INIT
SAVE              @RT0            (SYSTEM          )J.LABEL
SAVE              @RT0            (SYSTEM          )J.MDREST
SAVE              @RT0            (SYSTEM          )J.MDSAVE
SAVE       *** EOF **
SAVE       *** EOF ***
SAVE              @RT0            (SYSTEM          )J.MDTI
SAVE              @RT0            (SYSTEM          )J.MOUNT
SAVE              @RT0            (SYSTEM          )J.PRJCT
SAVE              @RT0            (SYSTEM          )J.SHAD
SAVE              @RT0            (SYSTEM          )J.SHUTD
SAVE              @RT0            (SYSTEM          )J.SOEX
SAVE              @RT0            (SYSTEM          )J.SOUT
SAVE              @RT0            (SYSTEM          )J.SSIN1
SAVE              @RT0            (SYSTEM          )J.SSIN2
SAVE              @RT0            (SYSTEM          )J.SWAPR
SAVE              @RT0            (SYSTEM          )J.TDEFI
SAVE              @RT0            (SYSTEM          )J.TINIT
SAVE              @RT0            (SYSTEM          )J.TSET
SAVE              @RT0            (SYSTEM          )J.TSM
SAVE              @RT0            (SYSTEM          )J.UNLOCK
SAVE              @RT0            (SYSTEM          )J.VFMT
SAVE              @RT0            (SYSTEM          )KEY
SAVE              @RT0            (SYSTEM          )KEYWORD
SAVE              @RT0            (SYSTEM          )LIST
SAVE              @RT0            (SYSTEM          )LOGCNT
SAVE              @RT0            (SYSTEM          )LOGTIME
SAVE              @RT0            (SYSTEM          )M.ERR
SAVE              @RT0            (SYSTEM          )OPCOM
SAVE              @RT0            (SYSTEM          )PASSWORD
SAVE              @RT0            (SYSTEM          )PAUSE
SAVE              @RT0            (SYSTEM          )RESTART
SAVE              @RT0            (SYSTEM          )SYSGEN
SAVE              @RT0            (SYSTEM          )TERMOUT
SAVE              @RT0            (SYSTEM          )VOLMGR
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RT0            (SYSTEM          )OH.32
SAVE              @RT0            (SYSTEM          )OH.32_E
SAVE              @RT0            (SYSTEM          )OH.32_OUT
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RT0            (OBJECT          )OH.ACBA
SAVE              @RT0            (OBJECT          )OH.ADA
SAVE              @RT0            (OBJECT          )OH.ALOC
SAVE              @RT0            (OBJECT          )OH.BKDM
SAVE          @RT0          (OBJECT        )OH.DBUG1
SAVE          @RT0          (OBJECT        )OH.DBUG2
SAVE          @RT0          (OBJECT        )OH.DMPMT
SAVE          @RT0          (OBJECT        )OH.EXEC
SAVE          @RT0          (OBJECT        )OH.EXEC2
SAVE          @RT0          (OBJECT        )OH.EXEC3
SAVE          @RT0          (OBJECT        )OH.EXSUB
SAVE          @RT0          (OBJECT        )OH.FISE
SAVE          @RT0          (OBJECT        )OH.IOCS
SAVE          @RT0          (OBJECT        )OH.MDT
SAVE          @RT0          (OBJECT        )OH.MEMM
SAVE          @RT0          (OBJECT        )OH.MEMM2
SAVE          @RT0          (OBJECT        )OH.MONS
SAVE          @RT0          (OBJECT        )OH.MVMT
SAVE          @RT0          (OBJECT        )OH.PTRAC
SAVE          @RT0          (OBJECT        )OH.REMM
SAVE          @RT0          (OBJECT        )OH.REXS
SAVE          @RT0          (OBJECT        )OH.SINIT
SAVE          @RT0          (OBJECT        )OH.SURE
SAVE          @RT0          (OBJECT        )OH.SWAPR
SAVE          @RT0          (OBJECT        )OH.TAMM
SAVE          @RT0          (OBJECT        )OH.TDEF
SAVE          @RT0          (OBJECT        )OH.TSM
SAVE          @RT0          (OBJECT        )OH.VOMM
SAVE     *** EOF ***
SAVE     *** EOF ***
SAVE          @RT0          (OBJECT_E      )OH.ALOC
SAVE          @RT0          (OBJECT_E      )OH.EXSUB
SAVE          @RT0          (OBJECT_E      )OH.FISE
SAVE          @RT0          (OBJECT_E      )OH.MEMM
SAVE          @RT0          (OBJECT_E      )OH.MONS
SAVE          @RT0          (OBJECT_E      )OH.PTRAC
SAVE          @RT0          (OBJECT_E      )OH.REMM
SAVE          @RT0          (OBJECT_E      )OH.REXS
SAVE          @RT0          (OBJECT_E      )OH.TAMM
SAVE          @RT0          (OBJECT_E      )OH.TSM
SAVE          @RT0          (OBJECT_E      )OH.VOMM
SAVE     *** EOF ***
SAVE     *** EOF ***
SAVE          @RT0          (OBJECT        )OH.CALM
SAVE          @RT0          (OBJECT        )OH.CPU
SAVE          @RT0          (OBJECT        )OH.CPU2
SAVE          @RT0          (OBJECT        )OH.ICP
SAVE          @RT0          (OBJECT        )OH.IP00
SAVE          @RT0          (OBJECT        )OH.IP02
SAVE          @RT0          (OBJECT        )OH.IP03
SAVE          @RT0          (OBJECT        )OH.IP04
SAVE          @RT0          (OBJECT        )OH.IP05
SAVE          @RT0          (OBJECT        )OH.IP06
SAVE          @RT0          (OBJECT        )OH.IP07
SAVE          @RT0          (OBJECT        )OH.IP08
SAVE          @RT0          (OBJECT        )OH.IP09
SAVE          @RT0          (OBJECT        )OH.IP0C
SAVE          @RT0          (OBJECT        )OH.IP0F
SAVE          @RT0          (OBJECT        )OH.IP10
SAVE          @RT0          (OBJECT        )OH.IP13
SAVE          @RT0          (OBJECT        )OH.IPAS
SAVE          @RT0          (OBJECT        )OH.IPCL
SAVE          @RT0          (OBJECT        )OH.IPHT
SAVE          @RT0          (OBJECT        )OH.IPIT
SAVE          @RT0          (OBJECT        )OH.IPU
SAVE          @RT0          (OBJECT        )OH.IPUAS
SAVE          @RT0          (OBJECT        )OH.IPUIT
SAVE          @RT0          (OBJECT        )OH.IPVP
SAVE     *** EOF ***
SAVE     *** EOF ***
SAVE          @RT0          (OBJECT        )OH.ASMP
SAVE          @RT0          (OBJECT        )OH.BSMP
SAVE          @RT0          (OBJECT        )OH.CPMP
SAVE          @RT0          (OBJECT        )OH.CTXIO
SAVE          @RT0          (OBJECT        )OH.DCSCI
SAVE          @RT0          (OBJECT        )OH.DCXIO
SAVE          @RT0          (OBJECT        )OH.DPXIO
SAVE          @RT0          (OBJECT        )OH.F8XIO
SAVE          @RT0          (OBJECT        )OH.GPMCS
SAVE          @RT0          (OBJECT        )OH.HSDG
SAVE          @RT0          (OBJECT        )OH.IBLG
SAVE          @RT0          (OBJECT        )OH.IFXIO
SAVE          @RT0          (OBJECT        )OH.LPXIO
SAVE          @RT0          (OBJECT        )OH.MDXIO
SAVE          @RT0          (OBJECT        )OH.MTSCI
SAVE          @RT0          (OBJECT        )OH.MTXIO
SAVE          @RT0          (OBJECT        )OH.MUX0
SAVE          @RT0          (OBJECT        )OH.NUXIO
SAVE          @RT0          (OBJECT        )OH.SLMP
SAVE          @RT0          (OBJECT        )OH.XIOS
SAVE     *** EOF ***
SAVE     *** EOF ***
SAVE          @RT0          (OBJECT_OUT    )OH.ADA
SAVE          @RT0          (OBJECT_OUT    )OH.ALOC
SAVE          @RT0          (OBJECT_OUT       )OH.BKDM
SAVE          @RT0          (OBJECT_OUT       )OH.CALM
SAVE          @RT0          (OBJECT_OUT       )OH.CTXIO
SAVE          @RT0          (OBJECT_OUT       )OH.DBUG1
SAVE          @RT0          (OBJECT_OUT       )OH.DBUG2
SAVE          @RT0          (OBJECT_OUT       )OH.DCSCI
SAVE          @RT0          (OBJECT_OUT       )OH.DCXIO
SAVE          @RT0          (OBJECT_OUT       )OH.DMPMT
SAVE          @RT0          (OBJECT_OUT       )OH.DPXIO
SAVE          @RT0          (OBJECT_OUT       )OH.EXEC3
SAVE          @RT0          (OBJECT_OUT       )OH.EXSUB
SAVE          @RT0          (OBJECT_OUT       )OH.F8XIO
SAVE          @RT0          (OBJECT_OUT       )OH.FISE
SAVE          @RT0          (OBJECT_OUT       )OH.HSDG
SAVE          @RT0          (OBJECT_OUT       )OH.IBLG
SAVE          @RT0          (OBJECT_OUT       )OH.ICP
SAVE          @RT0          (OBJECT_OUT       )OH.IFXIO
SAVE          @RT0          (OBJECT_OUT       )OH.IOCS
SAVE          @RT0          (OBJECT_OUT       )OH.IP00
SAVE          @RT0          (OBJECT_OUT       )OH.IP02
SAVE          @RT0          (OBJECT_OUT       )OH.IP03
SAVE          @RT0          (OBJECT_OUT       )OH.IP04
SAVE          @RT0          (OBJECT_OUT       )OH.IP05
SAVE          @RT0          (OBJECT_OUT       )OH.IP06
SAVE          @RT0          (OBJECT_OUT       )OH.IP07
SAVE          @RT0          (OBJECT_OUT       )OH.IP08
SAVE          @RT0          (OBJECT_OUT       )OH.IP09
SAVE          @RT0          (OBJECT_OUT       )OH.IP0C
SAVE          @RT0          (OBJECT_OUT       )OH.IP0F
SAVE          @RT0          (OBJECT_OUT       )OH.IP13
SAVE          @RT0          (OBJECT_OUT       )OH.IPAS
SAVE          @RT0          (OBJECT_OUT       )OH.IPCL
SAVE          @RT0          (OBJECT_OUT       )OH.IPHT
SAVE          @RT0          (OBJECT_OUT       )OH.IPIT
SAVE          @RT0          (OBJECT_OUT       )OH.IPPF
SAVE          @RT0          (OBJECT_OUT       )OH.IPVP
SAVE     *** EOF ***
SAVE     *** EOF ***
SAVE          @RT0          (OBJECT_OUT       )OH.LPXIO
SAVE          @RT0          (OBJECT_OUT       )OH.MDT
SAVE          @RT0          (OBJECT_OUT       )OH.MDXIO
SAVE          @RT0          (OBJECT_OUT       )OH.MEMM
SAVE          @RT0          (OBJECT_OUT       )OH.MEMM2
SAVE          @RT0          (OBJECT_OUT       )OH.MONS
SAVE          @RT0          (OBJECT_OUT       )OH.MTSCI
SAVE               @RT0             (OBJECT_OUT      )OH.MTXIO
SAVE               @RT0             (OBJECT_OUT      )OH.MVMT
SAVE               @RT0             (OBJECT_OUT      )OH.NUXIO
SAVE               @RT0             (OBJECT_OUT      )OH.PTRAC
SAVE               @RT0             (OBJECT_OUT      )OH.REMM
SAVE               @RT0             (OBJECT_OUT      )OH.REXS
SAVE               @RT0             (OBJECT_OUT      )OH.SINIT
SAVE               @RT0             (OBJECT_OUT      )OH.SURE
SAVE               @RT0             (OBJECT_OUT      )OH.SWAPR
SAVE               @RT0             (OBJECT_OUT      )OH.TAMM
SAVE               @RT0             (OBJECT_OUT      )OH.TDEF
SAVE               @RT0             (OBJECT_OUT      )OH.TSM
SAVE               @RT0             (OBJECT_OUT      )OH.VOMM
SAVE               @RT0             (OBJECT_OUT      )OH.XIOS
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE               @RT0             (HELP            )DS.HLP
SAVE               @RT0             (HELP            )GMPX.HLP
SAVE               @RT0             (HELP            )PROC.HLP
SAVE               @RT0             (HELP            )SAT.HLP
SAVE               @RT0             (HELP            )SVC.HLP
SAVE               @RT0             (HELP            )TOP.HLP
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE               @RT0             (SORT.MERGE      )OJ.SORT.MERGE
SAVE               @RT0             (SYSTEM          )FSORT2
SAVE               @RT0             (SYSTEM          )JJ.SORT
SAVE               @RT0             (SYSTEM          )SORT.DIR
SAVE               @RT0             (SYSTEM          )SORT.LIB
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE               @RT0             (SYSTEM          )M.EQUATESX32
SAVE               @RT0             (SYSTEM          )M.MACLIB
SAVE               @RT0             (SYSTEM          )M.MPXMAC
SAVE               @RT0             (SYSTEM          )M.OSEQUATESX32
SAVE               @RT0             (SYSTEM          )M.SERVICESX32
SAVE               @RT0             (SYSTEM          )MPX.PRO
SAVE               @RT0             (SYSTEM          )MPX.PRO.NOTDEF
SAVE               @RT0             (SYSTEM          )MPX.PRO.TDEF
SAVE               @RT0             (SYSTEM          )MPXPRE
SAVE               @RT0             (SYSTEM          )MPX_EXT
SAVE               @RT0             (SYSTEM          )MPX_NON
SAVE               @RT0             (SYSTEM          )MPX_OUT
SAVE               @RT0             (SYSTEM          )SHUTDOWN
SAVE               @RT0             (SYSTEM          )TDEFLIST
SAVE              @RTO              (SYSTEM          )TERMDEF
SAVE              @RTO              (SYSTEM          )VOLM
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RTO              (SYSTEM          )JH.32
SAVE              @RTO              (SYSTEM          )JH.32_E
SAVE              @RTO              (SYSTEM          )JH.32_OUT
SAVE              @RTO              (SYSTEM          )MSTRALL
SAVE              @RTO              (SYSTEM          )MSTRALLD
SAVE              @RTO              (SYSTEM          )MSTRALLS
SAVE              @RTO              (SYSTEM          )MSTREXT
SAVE              @RTO              (SYSTEM          )MSTREXTD
SAVE              @RTO              (SYSTEM          )MSTREXTS
SAVE              @RTO              (SYSTEM          )MSTROUT
SAVE              @RTO              (SYSTEM          )MSTROUTD
SAVE              @RTO              (SYSTEM          )MSTROUTS
SAVE              @RTO              (SYSTEM          )SG.32
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RTO              (SYSTEM          )ANALYZE
SAVE              @RTO              (SYSTEM          )ARTS
SAVE              @RTO              (SYSTEM          )AUTOJOB
SAVE              @RTO              (SYSTEM          )CHECKDEF
SAVE              @RTO              (SYSTEM          )CONFIG
SAVE              @RTO              (SYSTEM          )DD
SAVE              @RTO              (SYSTEM          )DDUMP
SAVE              @RTO              (SYSTEM          )DEVI
SAVE              @RTO              (SYSTEM          )DISCERR
SAVE              @RTO              (SYSTEM          )DISPTWO
SAVE              @RTO              (SYSTEM          )DOWNDAT2
SAVE              @RTO              (SYSTEM          )DOWNDATE
SAVE              @RTO              (SYSTEM          )DSPC1
SAVE              @RTO              (SYSTEM          )DSPC2
SAVE              @RTO              (SYSTEM          )FOLAP
SAVE              @RTO              (SYSTEM          )GM.CHHST
SAVE              @RTO              (SYSTEM          )GM.CLS
SAVE              @RTO              (SYSTEM          )GM.FLOW
SAVE              @RTO              (SYSTEM          )GM.SEAR
SAVE              @RTO              (SYSTEM          )GM.TDEMO
SAVE              @RTO              (SYSTEM          )LASER
SAVE              @RTO              (SYSTEM          )LMINFO
SAVE              @RTO              (SYSTEM          )LOGMDT
SAVE              @RTO              (SYSTEM          )LOGOFF
SAVE              @RTO              (SYSTEM          )LOOK
SAVE              @RTO              (SYSTEM          )M.CTLOAD
SAVE              @RT0              (SYSTEM           )NEWCOPY
SAVE              @RT0              (SYSTEM           )POOLSCAN
SAVE              @RT0              (SYSTEM           )PORTPROT
SAVE              @RT0              (SYSTEM           )RNVOL
SAVE              @RT0              (SYSTEM           )RT_DEBUG
SAVE              @RT0              (SYSTEM           )SDUTIL
SAVE              @RT0              (SYSTEM           )SPRINT
SAVE              @RT0              (SYSTEM           )SRCH
SAVE              @RT0              (SYSTEM           )SW.CHART
SAVE              @RT0              (SYSTEM           )SW.MODI
SAVE              @RT0              (SYSTEM           )SW.MON
SAVE              @RT0              (SYSTEM           )SWAPMON
SAVE              @RT0              (SYSTEM           )SYSINFO
SAVE              @RT0              (SYSTEM           )TABS
SAVE              @RT0              (SYSTEM           )TERMLOAD
SAVE              @RT0              (SYSTEM           )TSCAN
SAVE              @RT0              (SYSTEM           )UDTS
SAVE              @RT0              (SYSTEM           )US.DSPC
SAVE              @RT0              (SYSTEM           )US.ERR
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RT0              (SYSTEM           )M.PATCH
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RT0              (SYSTEM           )LOGONFLE
SAVE              @RT0              (SYSTEM           )M.CNTRL
SAVE              @RT0              (SYSTEM           )M.KEY
SAVE              @RT0              (SYSTEM           )MPXDIR
SAVE              @RT0              (SYSTEM           )MPXLIB
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RT0              (BUILD.JCL        )JJ.A.HLP
SAVE              @RT0              (BUILD.JCL        )JJ.A.NON
SAVE              @RT0              (BUILD.JCL        )JJ.A.RS1
SAVE              @RT0              (BUILD.JCL        )JJ.A.RS2
SAVE              @RT0              (BUILD.JCL        )JJ.A.RS3
SAVE              @RT0              (BUILD.JCL        )JJ.A.SGN
SAVE              @RT0              (BUILD.JCL        )JJ.A.SWP
SAVE              @RT0              (BUILD.JCL        )JJ.A.TDI
SAVE              @RT0              (BUILD.JCL        )JJ.A.VFM
SAVE              @RT0              (BUILD.JCL        )JJ.A.VOL
SAVE              @RT0              (BUILD.JCL        )JJ.B.LIB
SAVE              @RT0              (BUILD.JCL        )JJ.B.MAC
SAVE              @RT0              (BUILD.JCL        )JJ.COMPR
SAVE              @RT0              (BUILD.JCL        )JJ.CONCT
SAVE          @RT0           (BUILD.JCL       )JJ.ID
SAVE          @RT0           (BUILD.JCL       )JJ.LO.OS
SAVE          @RT0           (BUILD.JCL       )JJ.LO.VOL
SAVE          @RT0           (BUILD.JCL       )JJ.M.ERR
SAVE          @RT0           (BUILD.JCL       )JJ.MSTR
SAVE          @RT0           (BUILD.JCL       )JJ.XX.ER
SAVE          @RT0           (FUP             )JJ.A.ONE
SAVE          @RT0           (FUP             )JJ.C.VOL
SAVE          @RT0           (SYSTEM          )BATCH.OS
SAVE          @RT0           (SYSTEM          )MSTRSDT
SAVE          @RT0           (SYSTEM          )MSTRSRCE
SAVE      *** EOF ***
SAVE      *** EOF ***
SAVE          @RT0           (SOURCE          )SH.HSDG
SAVE          @RT0           (SOURCE          )SH.IBLG
SAVE          @RT0           (SOURCE          )SJ.ERR
SAVE          @RT0           (SOURCE          )SJ.STBLS
SAVE          @RT0           (SOURCE          )SJ.SWAPR2
SAVE          @RT0           (SOURCE          )SJ.VFDPT
SAVE          @RT0           (SOURCE          )SJ.XX.ER
SAVE      *** EOF ***
SAVE      *** EOF ***
SAVE          @RT0           (SYSTEM          )SM.EQUATESX32
SAVE          @RT0           (SYSTEM          )SM.MPXMC
SAVE          @RT0           (SYSTEM          )SM.OSEQUATESX32
SAVE          @RT0           (SYSTEM          )SM.RTMMC
SAVE          @RT0           (SYSTEM          )SM.SERVICESX32
SAVE      *** EOF ***
SAVE      *** EOF ***
SAVE          @RT0           (FUP             )FUP.DIR
SAVE          @RT0           (FUP             )FUP.LIB
SAVE      *** EOF ***
SAVE      *** EOF ***
SAVE          @RT0           (OBJECT          )OJ.ACCNT
SAVE          @RT0           (OBJECT          )OJ.ADMNT
SAVE          @RT0           (OBJECT          )OJ.AMOUNT
SAVE          @RT0           (OBJECT          )OJ.ASTAT
SAVE          @RT0           (OBJECT          )OJ.ATAPE
SAVE          @RT0           (OBJECT          )OJ.AUTO
SAVE          @RT0           (OBJECT          )OJ.AVOL1
SAVE          @RT0           (OBJECT          )OJ.AVOL2
SAVE          @RT0           (OBJECT          )OJ.COMP1
SAVE          @RT0           (OBJECT          )OJ.CRYPT
SAVE          @RT0           (OBJECT          )OJ.DECMP
SAVE          @RT0           (OBJECT          )OJ.DEVL
SAVE            @RTO         (OBJECT        )OJ.DSCMP
SAVE            @RTO         (OBJECT        )OJ.DTSAVE
SAVE            @RTO         (OBJECT        )OJ.ERR
SAVE            @RTO         (OBJECT        )OJ.FORMF
SAVE            @RTO         (OBJECT        )OJ.FREAD
SAVE            @RTO         (OBJECT        )OJ.HELP
SAVE            @RTO         (OBJECT        )OJ.HELPT
SAVE            @RTO         (OBJECT        )OJ.HLP
SAVE            @RTO         (OBJECT        )OJ.INIT
SAVE            @RTO         (OBJECT        )OJ.KEY
SAVE            @RTO         (OBJECT        )OJ.KEYWD
SAVE            @RTO         (OBJECT        )OJ.LABEL
SAVE            @RTO         (OBJECT        )OJ.LIST
SAVE            @RTO         (OBJECT        )OJ.LOGCNT
SAVE            @RTO         (OBJECT        )OJ.LOGTIME
SAVE            @RTO         (OBJECT        )OJ.MDREST
SAVE            @RTO         (OBJECT        )OJ.MDSAVE
SAVE            @RTO         (OBJECT        )OJ.MDTI
SAVE     *** EOF ***
SAVE     *** EOF ***
SAVE            @RTO         (OBJECT        )OJ.MOUNT
SAVE            @RTO         (OBJECT        )OJ.OPCOM
SAVE            @RTO         (OBJECT        )OJ.PAUSE
SAVE            @RTO         (OBJECT        )OJ.PROJ
SAVE            @RTO         (OBJECT        )OJ.PSWD
SAVE            @RTO         (OBJECT        )OJ.REST
SAVE            @RTO         (OBJECT        )OJ.SHAD
SAVE            @RTO         (OBJECT        )OJ.SHUTD
SAVE            @RTO         (OBJECT        )OJ.SOEX
SAVE            @RTO         (OBJECT        )OJ.SOUT
SAVE            @RTO         (OBJECT        )OJ.SSIN
SAVE            @RTO         (OBJECT        )OJ.SWAPR1
SAVE            @RTO         (OBJECT        )OJ.SWAPR2
SAVE            @RTO         (OBJECT        )OJ.TDEFI
SAVE            @RTO         (OBJECT        )OJ.TERMOUT
SAVE            @RTO         (OBJECT        )OJ.TINIT
SAVE            @RTO         (OBJECT        )OJ.TSET
SAVE            @RTO         (OBJECT        )OJ.TSM
SAVE            @RTO         (OBJECT        )OJ.UNLCK
SAVE            @RTO         (OBJECT        )OJ.VFDPT
SAVE            @RTO         (OBJECT        )OJ.VFMT
SAVE     *** EOF ***
SAVE     *** EOF ***
SAVE            @RTO         (OBJECT        )OJ.FMTIO
SAVE            @RTO         (OBJECT        )OJ.OBUTL
SAVE            @RT0            (OBJECT         )OJ.PSCAN
SAVE            @RT0            (OBJECT         )OJ.SDBUG
SAVE            @RT0            (OBJECT         )OJ.SEXEC
SAVE            @RT0            (OBJECT         )OJ.SGINI
SAVE            @RT0            (OBJECT         )OJ.SPH01
SAVE            @RT0            (OBJECT         )OJ.SPH02
SAVE            @RT0            (OBJECT         )OJ.SPH03
SAVE            @RT0            (OBJECT         )OJ.SPH04
SAVE            @RT0            (OBJECT         )OJ.SSCAN
SAVE            @RT0            (OBJECT         )OJ.STACK
SAVE            @RT0            (OBJECT         )OJ.STBLS
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE            @RT0            (BUILD.JCL      )JJ.F.27
SAVE            @RT0            (BUILD.JCL      )JJ.F.87
SAVE            @RT0            (SYSTEM         )DIR.27F
SAVE            @RT0            (SYSTEM         )DIR.87F
SAVE            @RT0            (SYSTEM         )FLOPSAVE
SAVE            @RT0            (SYSTEM         )FLOPSDT
SAVE            @RT0            (SYSTEM         )S3227
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE            @RT0            (PET            )HELP.PET
SAVE            @RT0            (PET            )OH.PET
SAVE            @RT0            (PET            )PET
SAVE            @RT0            (PET            )PET.DOC
SAVE            @RT0            (PET            )PET.ICRT
SAVE            @RT0            (PET            )PET.ICTS
SAVE            @RT0            (PET            )PETIO
SAVE            @RT0            (PET            )SH.PET
SAVE            @RT0            (REDUCE         )HELP.RED
SAVE            @RT0            (REDUCE         )REDUCE
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE            @RT0            (SOURCE         )US.ANA
SAVE            @RT0            (SOURCE         )US.ARTS
SAVE            @RT0            (SOURCE         )US.AUTOJ
SAVE            @RT0            (SOURCE         )US.CHART
SAVE            @RT0            (SOURCE         )US.CHHST
SAVE            @RT0            (SOURCE         )US.CKDEF
SAVE            @RT0            (SOURCE         )US.CLS
SAVE            @RT0            (SOURCE         )US.CONFG
SAVE            @RT0            (SOURCE         )US.DD
SAVE            @RT0            (SOURCE         )US.DDAT2
SAVE            @RT0            (SOURCE         )US.DDATE
SAVE               @RTO          (SOURCE        )US.DDUMP
SAVE               @RTO          (SOURCE        )US.DEVI
SAVE               @RTO          (SOURCE        )US.DISCE
SAVE               @RTO          (SOURCE        )US.DISPT
SAVE               @RTO          (SOURCE        )US.DSPC1
SAVE               @RTO          (SOURCE        )US.DSPC2
SAVE               @RTO          (SOURCE        )US.FLCOM
SAVE               @RTO          (SOURCE        )US.FLCOP
SAVE               @RTO          (SOURCE        )US.FLOW
SAVE               @RTO          (SOURCE        )US.FOLAP
SAVE               @RTO          (SOURCE        )US.LASER
SAVE               @RTO          (SOURCE        )US.LMINFO
SAVE               @RTO          (SOURCE        )US.LOGMDT
SAVE               @RTO          (SOURCE        )US.LOGOFF
SAVE               @RTO          (SOURCE        )US.LOOK
SAVE               @RTO          (SOURCE        )US.MODI
SAVE               @RTO          (SOURCE        )US.POOL
SAVE               @RTO          (SOURCE        )US.PROT
SAVE               @RTO          (SOURCE        )US.RNVOL
SAVE               @RTO          (SOURCE        )US.RTDBG
SAVE               @RTO          (SOURCE        )US.SDUTL
SAVE               @RTO          (SOURCE        )US.SEAR
SAVE               @RTO          (SOURCE        )US.SINFO
SAVE               @RTO          (SOURCE        )US.SPRINT
SAVE               @RTO          (SOURCE        )US.SRCH
SAVE               @RTO          (SOURCE        )US.SWAPMON
SAVE               @RTO          (SOURCE        )US.SWMON
SAVE               @RTO          (SOURCE        )US.TABS
SAVE               @RTO          (SOURCE        )US.TDEMO
SAVE               @RTO          (SOURCE        )US.TLOAD
SAVE               @RTO          (SOURCE        )US.TSCAN
SAVE               @RTO          (SOURCE        )US.UDTS
SAVE               @RTO          (SOURCE        )US.US.ER
SAVE         *** EOF ***
SAVE         *** EOF ***
SAVE               @RTO          (MIPS.SRC      )MIPIDOC
SAVE         *** EOF ***
SAVE         *** EOF ***
SAVE               @RTO          (MIPS.SRC      )MIPMAC
SAVE               @RTO          (MIPS.SRC      )MIPMACX32
SAVE               @RTO          (MIPS.SRC      )MIPREP
SAVE               @RTO          (MIPS.SRC      )MIPSMONX32.M
SAVE               @RTO          (MIPS.SRC      )MIPSYS
SAVE               @RTO          (MIPS.SRC      )MIPS_DIR
SAVE               @RTO          (MIPS.SRC      )MIPS_LIB
SAVE      *** EOF ***
SAVE      *** EOF ***
SAVE             @RT0              (MIPS.SRC         )BIGMAC
SAVE             @RT0              (MIPS.SRC         )BIO.MIP.C
SAVE             @RT0              (MIPS.SRC         )BUILD
SAVE             @RT0              (MIPS.SRC         )CC
SAVE             @RT0              (MIPS.SRC         )COMPIT
SAVE             @RT0              (MIPS.SRC         )DEFINES.MIP
SAVE             @RT0              (MIPS.SRC         )DIO.MIP.C
SAVE             @RT0              (MIPS.SRC         )ERRORFILE
SAVE             @RT0              (MIPS.SRC         )EXMPLE
SAVE             @RT0              (MIPS.SRC         )EXMPLE.A
SAVE             @RT0              (MIPS.SRC         )EXMPLE.O
SAVE             @RT0              (MIPS.SRC         )LOGLIB
SAVE             @RT0              (MIPS.SRC         )LOOPMAC
SAVE             @RT0              (MIPS.SRC         )MIPMAC.A
SAVE             @RT0              (MIPS.SRC         )MIPMACX32.A
SAVE             @RT0              (MIPS.SRC         )MIPREP.C
SAVE             @RT0              (MIPS.SRC         )MIPREP.O
SAVE             @RT0              (MIPS.SRC         )MIPREPJCL1
SAVE             @RT0              (MIPS.SRC         )MIPREPJCL2
SAVE             @RT0              (MIPS.SRC         )MIPSJCL
SAVE             @RT0              (MIPS.SRC         )MIPSMON.A
SAVE             @RT0              (MIPS.SRC         )MIPSMONX32.A
SAVE             @RT0              (MIPS.SRC         )MIPX32JCL
SAVE             @RT0              (MIPS.SRC         )MREP_DIR
SAVE             @RT0              (MIPS.SRC         )MREP_LIB
SAVE             @RT0              (MIPS.SRC         )OPEN.MIP.C
SAVE             @RT0              (MIPS.SRC         )README
SAVE      *** EOF ***
SAVE      *** EOF ***
SAVE             @RT0              (CSWI             )BLD.CSWI
SAVE             @RT0              (CSWI             )CLEAN_UP
SAVE             @RT0              (CSWI             )COPY_FILES
SAVE             @RT0              (CSWI             )CSWI.A
SAVE             @RT0              (CSWI             )CSWI.B
SAVE             @RT0              (CSWI             )CSWI.I
SAVE             @RT0              (CSWI             )J.CSWI.A
SAVE             @RT0              (CSWI             )J.CSWI.B
SAVE             @RT0              (CSWI             )J.CSWI.I
SAVE             @RT0              (CSWI             )J.STPK
SAVE             @RT0              (CSWI             )JT.SURE
SAVE             @RT0              (CSWI             )SG.II
SAVE      *** EOF ***
SAVE      *** EOF ***
SAVE              @RT0              (INTR              )BLD.INTR
SAVE              @RT0              (INTR              )CAT_RESULTS
SAVE              @RT0              (INTR              )CLEAN_UP
SAVE              @RT0              (INTR              )COPY_FILES
SAVE              @RT0              (INTR              )INTRPTB
SAVE              @RT0              (INTR              )INTRPTC
SAVE              @RT0              (INTR              )INTRPTLB
SAVE              @RT0              (INTR              )INTRPTLC
SAVE              @RT0              (INTR              )J.INTRPTB
SAVE              @RT0              (INTR              )J.INTRPTC
SAVE              @RT0              (INTR              )J.INTRPTLB
SAVE              @RT0              (INTR              )J.INTRPTLC
SAVE              @RT0              (INTR              )J.LOOPRT
SAVE              @RT0              (INTR              )J.STPK
SAVE              @RT0              (INTR              )JT.INTR
SAVE              @RT0              (INTR              )LOOPRT
SAVE              @RT0              (INTR              )MOVE_RESULTS
SAVE              @RT0              (INTR              )PRINT_RESULTS
SAVE              @RT0              (INTR              )SG.II
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RT0              (DEMO              )DEMODIR
SAVE              @RT0              (DEMO              )DEMOFILE
SAVE              @RT0              (DEMO              )DEMOLIB
SAVE              @RT0              (DEMO              )GO
SAVE              @RT0              (DEMO              )MPXDBAT
SAVE              @RT0              (DEMO              )MPXDTSM
SAVE              @RT0              (DEMO              )TESTSCR3
SAVE              @RT0              (DEMO              )TESTSCR4
SAVE              @RT0              (DEMO              )TESTTEST
SAVE              @RT0              (DEMO              )TESTX
SAVE              @RT0              (DEMO              )UJ.MPXDE
SAVE              @RT0              (DEMO              )UJ.MPXDL
SAVE              @RT0              (DEMO              )UJ.MPXT
SAVE              @RT0              (DEMO              )US.ACODE
SAVE              @RT0              (DEMO              )US.BKRND
SAVE              @RT0              (DEMO              )US.CPUTM
SAVE              @RT0              (DEMO              )US.DCODE
SAVE              @RT0              (DEMO              )US.EVNTR
SAVE              @RT0              (DEMO              )US.EVNTW
SAVE              @RT0              (DEMO              )US.FICMP
SAVE              @RT0              (DEMO              )US.GBL80
SAVE              @RT0              (DEMO              )US.GFLD
SAVE              @RT0              (DEMO              )US.GMSG
SAVE              @RT0              (DEMO              )US.IBNNR
SAVE              @RT0          (DEMO           )US.INIT
SAVE              @RT0          (DEMO           )US.IOMUL
SAVE              @RT0          (DEMO           )US.IOTS2
SAVE              @RT0          (DEMO           )US.IOTST
SAVE              @RT0          (DEMO           )US.ITERM
SAVE              @RT0          (DEMO           )US.MIXER
SAVE              @RT0          (DEMO           )US.MPXD
SAVE              @RT0          (DEMO           )US.MSG
SAVE              @RT0          (DEMO           )US.PRS
SAVE              @RT0          (DEMO           )US.PRSIE
SAVE              @RT0          (DEMO           )US.PSSCN
SAVE              @RT0          (DEMO           )US.RDDIR
SAVE              @RT0          (DEMO           )US.RESMD
SAVE              @RT0          (DEMO           )US.RESRC
SAVE              @RT0          (DEMO           )US.SCHED
SAVE              @RT0          (DEMO           )US.SMSG
SAVE              @RT0          (DEMO           )US.SMSG2
SAVE              @RT0          (DEMO           )US.STASK
SAVE              @RT0          (DEMO           )US.STREC
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RT0          (DEMO           )V.DEMOFL
SAVE              @RT0          (DEMO           )V.PHASE1
SAVE              @RT0          (DEMO           )V.PHASE2
SAVE              @RT0          (DEMO           )V.PHASE3
SAVE              @RT0          (DEMO           )V.PHASE4
SAVE              @RT0          (DEMO           )V.READY1
SAVE              @RT0          (DEMO           )V.READY2
SAVE              @RT0          (DEMO           )VMGRDEMO
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RT0          (SYSTEM         )BACKROUN
SAVE              @RT0          (SYSTEM         )IBNNR
SAVE              @RT0          (SYSTEM         )IOMUL
SAVE              @RT0          (SYSTEM         )IOTST
SAVE              @RT0          (SYSTEM         )ITERM
SAVE              @RT0          (SYSTEM         )MIXER
SAVE              @RT0          (SYSTEM         )MPXDBAT
SAVE              @RT0          (SYSTEM         )MPXDEMO
SAVE              @RT0          (SYSTEM         )MPXDTSM
SAVE       *** EOF ***
SAVE       *** EOF ***
REWIND          (TAP)
*****   BOT   *****
```

Contents of the Source Tape

For information about the contents of the files on the source tape, refer to the *MPX-32 Reference Manual Volume III*.

Note that this data is the truncated output of a VOLMGR run. For the exact file sizes, refer to the file SRCE.CONTENTS.

```
                  *** V O L U M E   M A N A G E R  3.5 ***
REWIND          (TAP)
*****  BOT  *****
SAVE             @RTO              (SYSTEM          )SRCE.CONTENTS
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE             @RTO              (SOURCE          )SH.ACBA
SAVE             @RTO              (SOURCE          )SH.ADA
SAVE             @RTO              (SOURCE          )SH.ALOC
SAVE             @RTO              (SOURCE          )SH.BKDM
SAVE             @RTO              (SOURCE          )SH.DBUG1
SAVE             @RTO              (SOURCE          )SH.DBUG2
SAVE             @RTO              (SOURCE          )SH.DMPMT
SAVE             @RTO              (SOURCE          )SH.EXEC
SAVE             @RTO              (SOURCE          )SH.EXEC2
SAVE             @RTO              (SOURCE          )SH.EXEC3
SAVE             @RTO              (SOURCE          )SH.EXSUB
SAVE             @RTO              (SOURCE          )SH.FISE
SAVE             @RTO              (SOURCE          )SH.IOCS
SAVE             @RTO              (SOURCE          )SH.MDT
SAVE             @RTO              (SOURCE          )SH.MEMM
SAVE             @RTO              (SOURCE          )SH.MEMM2
SAVE             @RTO              (SOURCE          )SH.MONS
SAVE             @RTO              (SOURCE          )SH.MVMT
SAVE             @RTO              (SOURCE          )SH.PTRAC
SAVE             @RTO              (SOURCE          )SH.REMM
SAVE             @RTO              (SOURCE          )SH.REXS
SAVE             @RTO              (SOURCE          )SH.SINIT
SAVE             @RTO              (SOURCE          )SH.SURE
SAVE             @RTO              (SOURCE          )SH.SWAPR
SAVE             @RTO              (SOURCE          )SH.TAMM
SAVE             @RTO              (SOURCE          )SH.TDEF
SAVE             @RTO              (SOURCE          )SH.TSM
SAVE             @RTO              (SOURCE          )SH.VOMM
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE             @RTO              (SOURCE          )SH.CALM
SAVE             @RTO              (SOURCE          )SH.CPU
SAVE              @RT0              (SOURCE          )SH.CPU2
SAVE              @RT0              (SOURCE          )SH.ICP
SAVE              @RT0              (SOURCE          )SH.IP00
SAVE              @RT0              (SOURCE          )SH.IP02
SAVE              @RT0              (SOURCE          )SH.IP03
SAVE              @RT0              (SOURCE          )SH.IP04
SAVE              @RT0              (SOURCE          )SH.IP05
SAVE              @RT0              (SOURCE          )SH.IP06
SAVE              @RT0              (SOURCE          )SH.IP07
SAVE              @RT0              (SOURCE          )SH.IP08
SAVE              @RT0              (SOURCE          )SH.IP09
SAVE              @RT0              (SOURCE          )SH.IP0C
SAVE              @RT0              (SOURCE          )SH.IP0F
SAVE              @RT0              (SOURCE          )SH.IP10
SAVE              @RT0              (SOURCE          )SH.IP13
SAVE              @RT0              (SOURCE          )SH.IPAS
SAVE              @RT0              (SOURCE          )SH.IPCL
SAVE              @RT0              (SOURCE          )SH.IPHT
SAVE              @RT0              (SOURCE          )SH.IPIT
SAVE              @RT0              (SOURCE          )SH.IPPF
SAVE              @RT0              (SOURCE          )SH.IPU
SAVE              @RT0              (SOURCE          )SH.IPUAS
SAVE              @RT0              (SOURCE          )SH.IPUIT
SAVE              @RT0              (SOURCE          )SH.IPVP
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE              @RT0              (SOURCE          )SH.ASMP
SAVE              @RT0              (SOURCE          )SH.BSMP
SAVE              @RT0              (SOURCE          )SH.CPMP
SAVE              @RT0              (SOURCE          )SH.CTXIO
SAVE              @RT0              (SOURCE          )SH.DCSCI
SAVE              @RT0              (SOURCE          )SH.DCXIO
SAVE              @RT0              (SOURCE          )SH.DPXIO
SAVE              @RT0              (SOURCE          )SH.F8XIO
SAVE              @RT0              (SOURCE          )SH.GPMCS
SAVE              @RT0              (SOURCE          )SH.HSDG
SAVE              @RT0              (SOURCE          )SH.IBLG
SAVE              @RT0              (SOURCE          )SH.IFXIO
SAVE              @RT0              (SOURCE          )SH.LPXIO
SAVE              @RT0              (SOURCE          )SH.MDXIO
SAVE              @RT0              (SOURCE          )SH.MTSCI
SAVE              @RT0              (SOURCE          )SH.MTXIO
SAVE              @RT0              (SOURCE          )SH.MUX0
SAVE              @RT0              (SOURCE          )SH.NUXIO
SAVE              @RT0              (SOURCE          )SH.SLMP
SAVE             @RT0             (SOURCE           )SH.XIOS
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE             @RT0             (SOURCE           )SJ.ACCNT
SAVE             @RT0             (SOURCE           )SJ.ADMNT
SAVE             @RT0             (SOURCE           )SJ.AMOUNT
SAVE             @RT0             (SOURCE           )SJ.ASTAT
SAVE             @RT0             (SOURCE           )SJ.ATAPE
SAVE             @RT0             (SOURCE           )SJ.AUTO
SAVE             @RT0             (SOURCE           )SJ.AVOL1
SAVE             @RT0             (SOURCE           )SJ.AVOL2
SAVE             @RT0             (SOURCE           )SJ.COMP1
SAVE             @RT0             (SOURCE           )SJ.CRYPT
SAVE             @RT0             (SOURCE           )SJ.DECMP
SAVE             @RT0             (SOURCE           )SJ.DEVL
SAVE             @RT0             (SOURCE           )SJ.DSCMP
SAVE             @RT0             (SOURCE           )SJ.DTSAV
SAVE             @RT0             (SOURCE           )SJ.ERR
SAVE             @RT0             (SOURCE           )SJ.FORMF
SAVE             @RT0             (SOURCE           )SJ.FREAD
SAVE             @RT0             (SOURCE           )SJ.HELP
SAVE             @RT0             (SOURCE           )SJ.HELPT
SAVE             @RT0             (SOURCE           )SJ.HLP
SAVE             @RT0             (SOURCE           )SJ.INIT
SAVE             @RT0             (SOURCE           )SJ.KEY
SAVE             @RT0             (SOURCE           )SJ.KEYWD
SAVE             @RT0             (SOURCE           )SJ.LABEL
SAVE             @RT0             (SOURCE           )SJ.LIST
SAVE             @RT0             (SOURCE           )SJ.LOGCNT
SAVE             @RT0             (SOURCE           )SJ.LOGTIME
SAVE             @RT0             (SOURCE           )SJ.MDREST
SAVE             @RT0             (SOURCE           )SJ.MDSAVE
SAVE             @RT0             (SOURCE           )SJ.MDTI
SAVE             @RT0             (SOURCE           )SJ.MOUNT
SAVE             @RT0             (SOURCE           )SJ.OPCOM
SAVE             @RT0             (SOURCE           )SJ.PAUSE
SAVE             @RT0             (SOURCE           )SJ.PROJ
SAVE             @RT0             (SOURCE           )SJ.PSWD
SAVE             @RT0             (SOURCE           )SJ.REST
SAVE             @RT0             (SOURCE           )SJ.TERMOUT
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE             @RT0             (SOURCE           )SJ.FMTIO
SAVE             @RT0             (SOURCE           )SJ.OBUTL
SAVE             @RT0             (SOURCE           )SJ.PSCAN
SAVE               @RTO               (SOURCE          )SJ.SDBUG
SAVE               @RTO               (SOURCE          )SJ.SEXEC
SAVE               @RTO               (SOURCE          )SJ.SGINI
SAVE               @RTO               (SOURCE          )SJ.SHAD
SAVE               @RTO               (SOURCE          )SJ.SHUTD
SAVE               @RTO               (SOURCE          )SJ.SOEX
SAVE               @RTO               (SOURCE          )SJ.SOUT
SAVE               @RTO               (SOURCE          )SJ.SPH01
SAVE               @RTO               (SOURCE          )SJ.SPH02
SAVE               @RTO               (SOURCE          )SJ.SPH03
SAVE               @RTO               (SOURCE          )SJ.SPH04
SAVE               @RTO               (SOURCE          )SJ.SSCAN
SAVE               @RTO               (SOURCE          )SJ.SSIN
SAVE               @RTO               (SOURCE          )SJ.STACK
SAVE               @RTO               (SOURCE          )SJ.STBLS
SAVE               @RTO               (SOURCE          )SJ.SWAPR1
SAVE               @RTO               (SOURCE          )SJ.SWAPR2
SAVE               @RTO               (SOURCE          )SJ.TDEFI
SAVE               @RTO               (SOURCE          )SJ.TINIT
SAVE               @RTO               (SOURCE          )SJ.TSET
SAVE               @RTO               (SOURCE          )SJ.TSM
SAVE               @RTO               (SOURCE          )SJ.UNLCK
SAVE               @RTO               (SOURCE          )SJ.VFDPT
SAVE               @RTO               (SOURCE          )SJ.VFMT
SAVE               @RTO               (SOURCE          )SJ.VPRE
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE               @RTO               (SORT.MERGE      )SOR.MAC
SAVE               @RTO               (SORT.MERGE      )SOR.SRC
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE               @RTO               (SOURCE          )SJ.XX.ER
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE               @RTO               (FUP             )FUPABT
SAVE               @RTO               (FUP             )FUPAUD
SAVE               @RTO               (FUP             )FUPBOO
SAVE               @RTO               (FUP             )FUPCMD
SAVE               @RTO               (FUP             )FUPCPY
SAVE               @RTO               (FUP             )FUPCRE
SAVE               @RTO               (FUP             )FUPCVT
SAVE               @RTO               (FUP             )FUPDAT
SAVE               @RTO               (FUP             )FUPDEB
SAVE               @RTO               (FUP             )FUPDEL
SAVE               @RTO               (FUP             )FUPERR
SAVE         @RTO         (FUP         )FUPFOR
SAVE         @RTO         (FUP         )FUPGBL
SAVE         @RTO         (FUP         )FUPHLP
SAVE         @RTO         (FUP         )FUPIO2
SAVE         @RTO         (FUP         )FUPLOG
SAVE         @RTO         (FUP         )FUPMAN
SAVE         @RTO         (FUP         )FUPMAT
SAVE         @RTO         (FUP         )FUPOPT
SAVE         @RTO         (FUP         )FUPPNB
SAVE         @RTO         (FUP         )FUPPRO
SAVE         @RTO         (FUP         )FUPRCB
SAVE         @RTO         (FUP         )FUPREN
SAVE         @RTO         (FUP         )FUPRES
SAVE         @RTO         (FUP         )FUPRSD
SAVE         @RTO         (FUP         )FUPSAV
SAVE         @RTO         (FUP         )FUPSDT
SAVE         @RTO         (FUP         )FUPSET
SAVE         @RTO         (FUP         )FUPSOR
SAVE         @RTO         (FUP         )FUPSRI
SAVE         @RTO         (FUP         )FUPSRT
SAVE         @RTO         (FUP         )FUPSYS
SAVE         @RTO         (FUP         )FUPTAP
SAVE         @RTO         (FUP         )FUPTIM
SAVE         @RTO         (FUP         )FUPTPA
SAVE         @RTO         (FUP         )FUPTRN
SAVE         @RTO         (FUP         )FUPUTL
SAVE         @RTO         (FUP         )FUPXTN
SAVE     *** EOF ***
SAVE     *** EOF ***
SAVE         @RTO         (FUP         )C_BLD
SAVE         @RTO         (FUP         )C_BRANCH
SAVE         @RTO         (FUP         )C_BSTO
SAVE         @RTO         (FUP         )C_CLSQ
SAVE         @RTO         (FUP         )C_CMPREC
SAVE         @RTO         (FUP         )C_CVNB
SAVE         @RTO         (FUP         )C_CVPB
SAVE         @RTO         (FUP         )C_DEXT
SAVE         @RTO         (FUP         )C_DLNKLR
SAVE         @RTO         (FUP         )C_EXCHNG
SAVE         @RTO         (FUP         )C_GETSI
SAVE         @RTO         (FUP         )C_GOI
SAVE         @RTO         (FUP         )C_INSFR
SAVE         @RTO         (FUP         )C_INSLR
SAVE         @RTO         (FUP         )C_IOUT
SAVE         @RTO         (FUP         )C_LNGX
SAVE            @RT0          (FUP           )C_LOAD
SAVE            @RT0          (FUP           )C_MOVREC
SAVE            @RT0          (FUP           )C_MRGINT
SAVE            @RT0          (FUP           )C_MVA
SAVE            @RT0          (FUP           )C_RDSQM
SAVE            @RT0          (FUP           )C_RLSE
SAVE            @RT0          (FUP           )C_SCSU
SAVE            @RT0          (FUP           )C_SORT
SAVE            @RT0          (FUP           )C_SOX
SAVE            @RT0          (FUP           )C_SREF
SAVE            @RT0          (FUP           )C_SRTX
SAVE            @RT0          (FUP           )C_WFIO
SAVE            @RT0          (FUP           )C_WRSQ
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE            @RT0          (FUP           )CR_BDIVW
SAVE            @RT0          (FUP           )CR_GOSG
SAVE            @RT0          (FUP           )CR_LBVAL
SAVE            @RT0          (FUP           )CR_LDP8
SAVE            @RT0          (FUP           )CR_SBVAL
SAVE            @RT0          (FUP           )CR_SCALE
SAVE            @RT0          (FUP           )CR_SIGN
SAVE            @RT0          (FUP           )FR_ABORT
SAVE            @RT0          (FUP           )F_GETWFS
SAVE            @RT0          (FUP           )F_INIT
SAVE            @RT0          (FUP           )F_IOER
SAVE            @RT0          (FUP           )F_LASPAS
SAVE            @RT0          (FUP           )F_MERGER
SAVE            @RT0          (FUP           )F_OPSQ
SAVE            @RT0          (FUP           )F_RETN
SAVE            @RT0          (FUP           )F_SRT
SAVE            @RT0          (FUP           )F_SRTINT
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE            @RT0          (PET           )ADDVAL.PET.C
SAVE            @RT0          (PET           )ALLOC.PET.C
SAVE            @RT0          (PET           )ASMACT.PET.C
SAVE            @RT0          (PET           )ASSEM.PET.A
SAVE            @RT0          (PET           )ASSEM_HANDLER
SAVE            @RT0          (PET           )ASSMIT
SAVE            @RT0          (PET           )BIGASSM
SAVE            @RT0          (PET           )BIGMAC
SAVE            @RT0          (PET           )BRK.PET.C
SAVE            @RT0          (PET           )BUILD
SAVE            @RT0          (PET           )BUILD_MISC
SAVE            @RTO            (PET            )BUILD_PET
SAVE            @RTO            (PET            )CC
SAVE            @RTO            (PET            )CI.DEFS
SAVE            @RTO            (PET            )CK.CHAR.C
SAVE            @RTO            (PET            )CK.KEY.C
SAVE            @RTO            (PET            )CK.NUM.C
SAVE            @RTO            (PET            )CK.STR.C
SAVE            @RTO            (PET            )CK.TRAN.C
SAVE            @RTO            (PET            )CLEANUP.PET.C
SAVE            @RTO            (PET            )CLOSE.PET.C
SAVE            @RTO            (PET            )CPARSE.C
SAVE            @RTO            (PET            )CSPCODE.PET.A
SAVE            @RTO            (PET            )CSWI.PET.C
SAVE            @RTO            (PET            )CSWT.PET.C
SAVE            @RTO            (PET            )DEFINE.C
SAVE            @RTO            (PET            )DEFINES.PET
SAVE            @RTO            (PET            )DEPENDS.PET
SAVE            @RTO            (PET            )DISPATCH.PET.C
SAVE            @RTO            (PET            )DO.ACT.C
SAVE            @RTO            (PET            )ENDACT.PET.A
SAVE            @RTO            (PET            )ERROR.PET.C
SAVE            @RTO            (PET            )GET.ADD.PET.C
SAVE            @RTO            (PET            )GET.KEY.C
SAVE            @RTO            (PET            )GET.TRAN.C
SAVE            @RTO            (PET            )GETABS.PET.C
SAVE            @RTO            (PET            )GETDATA.PET.C
SAVE            @RTO            (PET            )GLOBALS.PET
SAVE            @RTO            (PET            )HELP.PET
SAVE            @RTO            (PET            )HELP.PET.C
SAVE        *** EOF ***
SAVE        *** EOF ***
SAVE            @RTO            (PET            )ICLT.PET.C
SAVE            @RTO            (PET            )ICLTLOOP.PET.A
SAVE            @RTO            (PET            )IIT.PET.C
SAVE            @RTO            (PET            )INIT.PET.C
SAVE            @RTO            (PET            )INSPTCH.PET.C
SAVE            @RTO            (PET            )IODEFS.PET
SAVE            @RTO            (PET            )IOST.PET.C
SAVE            @RTO            (PET            )IRPCODE.PET.A
SAVE            @RTO            (PET            )LOOPASSM
SAVE            @RTO            (PET            )LOOPMAC
SAVE            @RTO            (PET            )LW
SAVE            @RTO            (PET            )MPXINTSVC.PET.A
SAVE            @RTO            (PET            )NXT.STATE.C
SAVE            @RTO            (PET            )NXT.TRAN.C
SAVE              @RTO             (PET             )OPEN.PET.C
SAVE              @RTO             (PET             )PARSE.PET.C
SAVE              @RTO             (PET             )PART.PET.C
SAVE              @RTO             (PET             )PATCH.PET.C
SAVE              @RTO             (PET             )PCODE.PET.A
SAVE              @RTO             (PET             )PET.C
SAVE              @RTO             (PET             )PET.DOC
SAVE              @RTO             (PET             )PET.ICLT.A
SAVE              @RTO             (PET             )PETCC
SAVE              @RTO             (PET             )PETIO.A
SAVE              @RTO             (PET             )PETSDOC
SAVE              @RTO             (PET             )PET_DIR
SAVE              @RTO             (PET             )PET_LIB
SAVE              @RTO             (PET             )PUTINFO.PET.C
SAVE              @RTO             (PET             )PUTLINE.PET.C
SAVE              @RTO             (PET             )PUTSTD.PET.C
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RTO             (PET             )RCVR.PET.C
SAVE              @RTO             (PET             )READ.PET.C
SAVE              @RTO             (PET             )REDIRECT.PET.C
SAVE              @RTO             (PET             )RTCS.PET.C
SAVE              @RTO             (PET             )RUN.PET.C
SAVE              @RTO             (PET             )SETSTR.PET.C
SAVE              @RTO             (PET             )SETVAL.PET.C
SAVE              @RTO             (PET             )SETVERB.PET.C
SAVE              @RTO             (PET             )SH.PET
SAVE              @RTO             (PET             )SHOW.PET.C
SAVE              @RTO             (PET             )SLEEP.PET.C
SAVE              @RTO             (PET             )SOURCE.PET.C
SAVE              @RTO             (PET             )START.PET.C
SAVE              @RTO             (PET             )STOP.PET.C
SAVE              @RTO             (PET             )STRCPY2.PET.C
SAVE              @RTO             (PET             )TABLE.PET.A
SAVE              @RTO             (PET             )UNPATCH.PET.C
SAVE              @RTO             (PET             )USER.PET.C
SAVE              @RTO             (PET             )UTREAD.PET.A
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE              @RTO             (REDUCE          )ASSMIT
SAVE              @RTO             (REDUCE          )BIGASSM
SAVE              @RTO             (REDUCE          )BIGMAC
SAVE              @RTO             (REDUCE          )BIO.RED.C
SAVE              @RTO             (REDUCE          )BRK.CHK.RED.C
SAVE              @RTO             (REDUCE          )BUILD
SAVE            @RTO            (REDUCE         )BUILD_REDUCE
SAVE            @RTO            (REDUCE         )CC
SAVE            @RTO            (REDUCE         )CHKVERB.RED.C
SAVE            @RTO            (REDUCE         )CI.DEFS
SAVE            @RTO            (REDUCE         )CK.CHAR.C
SAVE            @RTO            (REDUCE         )CK.KEY.C
SAVE            @RTO            (REDUCE         )CK.NUM.C
SAVE            @RTO            (REDUCE         )CK.STR.C
SAVE            @RTO            (REDUCE         )CK.TRAN.C
SAVE            @RTO            (REDUCE         )CLEANUP.RED.C
SAVE            @RTO            (REDUCE         )CNVRT.RED.C
SAVE            @RTO            (REDUCE         )CPARSE.C
SAVE            @RTO            (REDUCE         )DEFINE.C
SAVE            @RTO            (REDUCE         )DEFINES.PET
SAVE            @RTO            (REDUCE         )DEFINES.RED
SAVE            @RTO            (REDUCE         )DIO.RED.C
SAVE            @RTO            (REDUCE         )DISPATCH.RED.C
SAVE            @RTO            (REDUCE         )DO.ACT.C
SAVE       *** EOF ***
SAVE       *** EOF ***
SAVE            @RTO            (REDUCE         )ENDACT.RED.A
SAVE            @RTO            (REDUCE         )ERROR.RED.C
SAVE            @RTO            (REDUCE         )FORMAT.RED.C
SAVE            @RTO            (REDUCE         )GET.KEY.C
SAVE            @RTO            (REDUCE         )GET.TRAN.C
SAVE            @RTO            (REDUCE         )GETCSWI.RED.C
SAVE            @RTO            (REDUCE         )GETIIT.RED.C
SAVE            @RTO            (REDUCE         )GETINFO.RED.C
SAVE            @RTO            (REDUCE         )GETIOST.RED.C
SAVE            @RTO            (REDUCE         )GETSUMS.RED.C
SAVE            @RTO            (REDUCE         )HELP.RED
SAVE            @RTO            (REDUCE         )HELP.RED.C
SAVE            @RTO            (REDUCE         )INIT.RED.C
SAVE            @RTO            (REDUCE         )LOOPASSM
SAVE            @RTO            (REDUCE         )LOOPMAC
SAVE            @RTO            (REDUCE         )LW
SAVE            @RTO            (REDUCE         )NXT.STATE.C
SAVE            @RTO            (REDUCE         )NXT.TRAN.C
SAVE            @RTO            (REDUCE         )OPEN.RED.C
SAVE            @RTO            (REDUCE         )PARSE.RED.C
SAVE            @RTO            (REDUCE         )PUTLINE.RED.C
SAVE            @RTO            (REDUCE         )RCVR.RED.C
SAVE            @RTO            (REDUCE         )REDCC
SAVE            @RTO            (REDUCE         )REDSDOC
SAVE            @RTO            (REDUCE         )REDUCE.C
SAVE                @RT0                 (REDUCE          )RED_DIR
SAVE                @RT0                 (REDUCE          )RED_LIB
SAVE                @RT0                 (REDUCE          )SETSTR.RED.C
SAVE                @RT0                 (REDUCE          )SETVAL.RED.C
SAVE                @RT0                 (REDUCE          )SETVERB.RED.C
SAVE                @RT0                 (REDUCE          )SHOW.RED.C
SAVE                @RT0                 (REDUCE          )SOURCE.RED.C
SAVE                @RT0                 (REDUCE          )SQRT.RED.A
SAVE                @RT0                 (REDUCE          )START.RED.C
SAVE                @RT0                 (REDUCE          )STRCPY2.RED.C
SAVE                @RT0                 (REDUCE          )TABLE.RED.A
SAVE                @RT0                 (REDUCE          )USER.RED.C
SAVE         *** EOF ***
SAVE         *** EOF ***
REWIND            (TAP)
*****   BOT   *****
```

INSTALLATION ENVIRONMENT

The following revision information reflects the revision levels in the system used in the development and testing of MPX-32 Revision 3.5. This does not mean that earlier or later revision levels will not work.

For the boards listed below with no firmware number, the printed circuit assembly number includes the firmware revision level.

Model	Description	Printed Circuit Assembly	Firmware
2345	RTOM	160-103109-001K	
2345	RTOM IT	160-103109-001M	
3034-4	4 MB ISM	160-103723-001D	531-322950-001
73-4030	RMS node	160-103769-001E	531-322148-001
73-4030	RMS node	160-103770-002D	
73-4030	Write Sense Cont	160-103769-001E	531-103148-001
73-4030	Read Sense Cont	160-103770-001D	
7303-8	Dual-ported IMM - 8 MB	160-103351-001F	531-322140-001
8001	IOP DI	160-103509-001C	
8001	IOP	160-103519-001K	531-322740-009
8002	MFP	160-103873-001M	531-103195-011
8002	MFP	160-103874-001E	
8002	MFP	160-103874-0015	531-103197-011
8030	Line printer/floppy disk	160-103448-001A	531-322687-004
8030	Line printer/floppy disk	160-103448-001H	531-322687-006
8031	L. P. F. D. C.	160-103612-001A	531-322775-002
8031	L. P. F. D. C.	160-103612-001B	531-322775-002
8050	High speed tape processor	160-103149-001J	531-322606-002
8050	High speed tape processor	160-103369-001E	
8051	Buffered tape processor	160-103640-001C	531-322795-002
8051	Buffered tape processor	160-103640-001E	531-322795-004
8055	Disk processor II	160-103282-001	
8055	Disk processor II (RPU)	161-103351-001F	531-322626-007
8055	Disk processor II (PROM)	161-103365-001A	531-322627-004
8060	Universal disk processor	160-103282-001E	
8060	Universal disk processor	160-103282-001K	
8060	Universal disk processor	160-103623-001F	531-322645-005
8060	Universal disk processor	160-103623-001J	531-322645-006
8064	HSDP	160-103835-003B	
8064	HSDP	160-103835-003F	

(Continued on next page.)

Model	Description	Printed Circuit Assembly	Firmware
8064	HSDP	160-103858-001	531-103190-001
8064	HSDP	160-103858-001F	531-103190-004
8159	Cache disk accelerator	160-103623-001H	531-322890-004
8191	Disk Proc 8DR Opt.	160-103291-001C	
8512-2	Eight-line asynch	160-103768-001D	531-322624-003
8512-2	Eight-line asynch	160-103768-001E	531-322624-004
8512-2	Eight-line asynch	160-103768-001F	531-322624-005
8515	Ethernet	160-103675-001W	531-322947-005
8516	Ethernet	160-103675-001V	531-322947-005
8516	Ethernet	160-103726-001A	
9130	HSD interface	160-103364-001F	531-322577-009

CMOS Revision Levels:

Description	Printed Circuit Assembly	Firmware
Single slot CMOS CPU	160-103803-001F	

CONCEPT 32/67 revision levels (ACS Ph.I 820K; ACS Ph.II 944G)

Description	Printed Circuit Assembly	Firmware
MS board	160-103654-002C	531-322940-002
CS board	160-103583-001M	531-322801-001
IE board	160-103656-001E	531-322920-001
IOX Universal	160-103063-007	
MS2 SPI	160-103609-001	

CONCEPT 32/87 revision levels: (FSS Rev 2.4M)

CPU boards

Description	Printed Circuit Assembly	Firmware
M (microstore)	160-103563-001	
M (microstore)	160-103563-002	
D board	160-103488-001B	
S board	160-103385-001E	
F board	160-103510-001B	
G board	160-103697-001A	
H board	160-103522-001C	

(Continued on next page.)

Description	Printed Circuit Assembly	Firmware
O board	160-103611-001D	
X board	160-103521-001A	531-322730-001
B (buffer)	160-103564-001C	
B (buffer)	160-103564-002C	
C (cache control)	160-103472-001A	
T (translation)	160-103533-001G	
K (clock)	160-103616-001E	

IPU boards

Description	Printed Circuit Assembly	Firmware
M (microstore)	160-103563-001	
M (microstore)	160-103563-002	
D board	160-103488-001B	
E board	160-103696-001	
S board	160-103385-001E	
F board	160-103510-001B	
G board	160-103695-001A	
P board	160-103479-001	
Q board	160-103480-001A	
O board	160-103611-001D	
I board	160-103610-001B	521-322850-002
A board	160-103586-001G	531-322885-001
L board	160-103593-001E	531-322875-001
X (instrumentation)	160-103521-001A	531-322730-001
B (buffer)	160-103564-001C	
B (buffer)	160-103564-002D	
C (cache control)	160-103588-001C	
T (translation)	160-103533-001G	
K (clock)	160-103616-001E	
H board	160-322522-001D	531-322565-003

CONCEPT 32/97 "9705" Revision Levels: (FSS Rev 2.3G)

Description	Printed Circuit Assembly	Firmware
M (microstore)	160-103563-001	
M (microstore)	160-103563-002	
D board	160-103488-001B	
E board	160-103696-001	

(Continued on next page.)

Description	Printed Circuit Assembly	Firmware
S board	160-103385-001E	
F board	160-103510-001B	
G board	160-103695-001A	
P board	160-103479-001A	531-322735-001
Q board	160-103480-001A	
O board	160-103611-001D	
I board	160-103610-001B	531-322850-002
A board	160-103586-001E	531-322885-001
L (look aside buffer)	160-103593-001L	531-322875-001
X (instrumentation)	160-103521-001A	531-322730-001
B (buffer)	160-103564-001C	
B (buffer)	160-103564-002C	
C (cache control)	160-103588-001C	
T (translation)	160-103533-001L	
K (clock)	160-103616-001E	
H board	160-103522-001D	531-322565-003

CONCEPT 32/97 "9780"

Description	Printed Circuit Assembly	Firmware
M (microstore)	160-103563-001	
M (microstore)	160-103563-002	
D board	160-103488-001B	
E board	160-103696-001	
S board	160-103385-001E	
F board	160-103510-001B	
G board	160-103695-001A	
P board	160-103479-001A	531-322735-001
Q board	160-103480-001A	
O board	160-103611-001D	
I board	160-103610-001B	531-322850-002
A board	160-103586-001D	531-322885-001
L (look aside buffer)	160-103593-001J	531-322875-001
X (instrumentation)	160-103521-001A	531-322730-001
B (buffer)	160-103564-001C	
B (buffer)	160-103564-002D	
C (cache control)	160-103588-001C	
T (translation)	160-103533-001G	
K IV (clock)	160-103616-001D	

PROGRAM ID AND DATE/TIME INFORMATION

Information about version numbers, passed as program identification information, and date/time of compile/assembly is useful in identifying tasks and resident modules. This information is optionally available in object files, subroutine libraries, and load modules. To display this information, use SEARCHER, a task in the MPX-32 Toolkit.

On the 3.5 SDT, the file MPXID3.5 contains the output of a series of SEARCHER runs. This file is an SLOF file from the JJ.ID job stream. Its identification information can be used to verify that the proper versions of the software are on the system.

4 DIFFERENCES BETWEEN REVISION 3.4 U03 AND REVISION 3.5

All changes described in this chapter apply to MPX-32 Revision 3.5 and later revisions, unless otherwise noted.

ENHANCEMENTS

CONCEPT 32/2000 Only Enhancements

Mapped Out Option

The mapped out option is a new execution mode for MPX-32 Revision 3.5 which decouples the resident operating system from the address space of executing tasks. In this mode, resident MPX-32 itself runs unmapped with all references to task address space being resolved using new 'through context' instructions. These new instructions cause a "one-shot" access to the current map register contents during execution of the instruction. Effectively, this removes the requirement that MPX-32 run mapped into the address space of the task. For nonbase mode tasks, this freed address space is returned to the user for additional code and/or data space. Base mode tasks continue to run with MPX-32 mapped into their address space.

The creation of a mapped out MPX-32 image is specified at SYSGEN time via the MACHINE directive or by an explicit assign of OBJ to a compressed mapped out object file. This feature is available on a task by task basis and allows the task to execute with MPX-32 removed from the task's logical address space. Executing a task in the mapped out mode results in more direct (execution) address space for code and directly addressable data with no performance degradation.

Since the user may need to modify source in order to take advantage of this feature, the Catalog ENVIRONMENT directive is the central point of control for initiating the mapped out capability. It has the NOMAPOUT and MAPOUT keywords added to explicitly define a task's mapped out state as well as the SYSMAP keyword that defers the request until runtime. The user can specify the mapped out or mapped in options using:

- CATALOG ENVIRONMENT directive
- TSM MAPOUT or NOMAPOUT directives
- M.PTSK SVC call
- SYSGEN MAPOUT or NOMAPOUT directives

Refer to chapter 7 of this Software Release Notes, *MPX-32 Reference Manual Volume I* Chapter 3, and *MPX-32 Reference Manual Volume III* Chapter 7 for more information about mapped out.

Demand Page Processing

Demand page provides better physical memory management for a multitask environment and allows for execution of tasks which are larger than available physical memory. When demand page processing is in effect for a task, it is loaded on demand into memory in map block increments. As additional logical address space is referenced in a task, the map block (page) needed to satisfy the address is brought into memory (paged in) and added to the task's working set of map blocks. The working set consists of the physical memory mapped into the logical space of the task.

As pages are no longer referenced by the task within a specified amount of time, they are considered aged and are removed from the working set. Modified aged pages are linked to the task's page-out queue or shared memory page-out queue for writing to the swap volume. After they are paged out to the swap file, the physical memory can then be added to the free list (freed) for reuse. Unmodified aged pages are simply freed. Freed pages which still contain valid information from the last allocation can be retrieved. Paged out and freed pages are not part of the task's working set.

When a new memory address not currently in the working set is referenced, the page satisfying that address is first sought from the free list or page-out queue, avoiding I/O from the load module or swap file.

Demand page is the default processing mode on a mapped out system image. It is not available on mapped in images. To be eligible for demand page processing, tasks must be either absolute or running with MPX-32 mapped out of their address space with the TSA moved. If a task is eligible, further control is provided by the SYSGEN DEMAND directive and the Catalog DEMAND/NODEMAND directives.

Refer to *Reference Manual Volume I*, Chapter 3, and *Reference Manual Volume III*, Chapter 7, for more information about demand page.

Enhancements for all CONCEPT/32 Systems

Online Help

Online Help now provides you with the ability to print one or more topics. When in the online help facility, (P)rint is one of the choices. Selecting this choice displays a print menu which allows you to specify your choice of topics to print to SLO or write to a permanent file.

When writing your own help file information, you can select particular occurrences of keyword text, eliminating the highlighting of more than one instance of a keyword on a screen.

For more information about Online Help, refer to *MPX-32 Reference Manual Volume II*, Chapter 9.

Last Access Information

Information about access to task level resources is provided in MPX-32 Revision 3.5. During the close operation of a task level disk resource, the resource descriptor (RD) is updated with the current date, time, and owner name of the last user to access the resource. VOLMGR saves are not recorded. The VOLMGR DELETE, LOG, RESTORE, SAVE, and SET directives can be used to view this information with the BRIEF=N option and conditionally control output with the ACCESSED= parameter.

This information update can be inhibited by specifying the NOLACC SYSGEN directive.

Refer to MPX-32 Reference Manual Volume II, Chapter 3 for more information.

CPU Only Scheduler (H.EXEC3)

The CPU Only Scheduler (H.EXEC3) reduces scheduling overhead with the removal of IPU scheduling and control logic. H.EXEC3 is automatically selected when the IPU directive is omitted at SYSGEN time. H.EXSUB, containing the common subroutines, is also selected automatically.

Refer to *MPX-32 Technical Manual Volume II* for information on the Executive modules.

Extended H.EXEC (H.EXSUB)

The Executive module, H.EXEC, is split into two parts allowing for a module containing subroutines which can reside in extended memory. This applies to Executive modules H.EXEC, H.EXEC2, and H.EXEC3. The module is divided as follows:

- H.EXEC contains the entry points which are called by M.CALL and additional code that can be executed at interrupt level. This module resides in non-extended MPX.

- H.EXSUB contains subroutines that are not executed at interrupt level and can reside in extended MPX-32.

For more information, refer to *MPX-32 Technical Manual Volume II.*

SYSGEN

- **New SYSGEN Directives**

  The following directives have been added to SYSGEN.

  *** (Comment)**

  The asterisk is supplied as a comment designator. In SYSGEN directive files, a line with an asterisk in column one is considered a comment and ignored during processing of the directive file.

  SWAPDEV

  This directive allows the user to specify the default swap device.

  OWNERNAME

  This directive allows optional echoing of the user's owner name to the terminal as the user logs on.

  LOGON

  This directive allows users to log on to MPX-32 concurrently using the same owner name. SYSTEM can be limited to one logon.

  BATCHMSG

  This directive controls message display to the system console and/or terminals.

  NOLACC

  This directive disables the new MPX-32 functionality for tracking last date, time, and owner name of access to task-level resources.

  NOSYSVOL

  This directive inhibits mounting and use of a system volume for RMSS systems.

  TSMEXIT and NOTSMEXIT

  The NOTSMEXIT directive specifies that TSM remains active when not in use. The TSMEXIT directive specifies that TSM exits from the system when inactive.

MAPOUT and NOMAPOUT

The MAPOUT directive designates mapped out as the default execution mode. The NOMAPOUT directive designates mapped in as the default execution mode. These directives apply only to tasks cataloged with the ENVIRONMENT SYSMAP keyword executing on a mapped out system image.

NODEMAND

This directive inhibits demand page processing on a mapped out image that by default supports demand page.

DEMAND

This directive specifies by priority level which tasks are demand page processed on a mapped out image that supports demand page.

AGE, BEGPGOUT and ENDPGOUT

These directives are provided to tune demand page processing environment. AGE specifies the amount of virtual time to pass before an unreferenced page is considered aged. BEGPGOUT and ENDPGOUT specify the minimum and maximum percentages, respectively, of total memory desired for free pages.

- SYSGEN documentation has been improved to clarify several directives. The sample SYSGEN file (SG.32) now includes the directives basic to any system and optional directive entries commented out. Copy this file to use as the basis for your SYSGEN directives file and simply remove the asterisk in column one to include the commented optional directives that pertain to your installation. In addition, SYSGEN has been included in Online Help under System Administrative Tools.

For more information on SYSGEN, refer to the *MPX-32 Reference Manual Volume III*, Chapter 7.

Volume Formatter

The following changes have been made to the Volume Formatter (J.VFMT):

- The DESTROY parameter is now included in the FORMAT directive.
- The default value of the CONFIRM parameter is now Y (yes) for all directives.
- The BOOTFILE parameter allows the user to choose the standard new bootstrap program, the standard old bootstrap program, or a custom bootstrap program when using the FORMAT, REPLACE, and NEWBOOT directives.
- The SSUB parameter allows the user to perform sector substitution on HSDP disks when using the INITIALIZE directive.
- The ISIZE parameter default value has been changed from 400 blocks to 600 blocks.
- For MFP SCSI disks, the appropriate disk bootstrap program is automatically selected based on the revision level of the MFP.

For more information on these enhancements, refer to the *MPX-32 Reference Manual Volume III*, Chapter 13.

Extended TSA (Nonbase Tasks Only)

The extended TSA feature is optionally available to any nonbase task. This feature allows the user to move the task's TSA into the task's extended address space. Positioning the TSA in the task's extended address space results in more executable address space for code and directly addressable data.

The EXTDMPX directive has the optional keywords NOTSA (do not move the TSA) and TSA (move the TSA) added to direct MPX-32 in positioning the task's TSA. Since the user may need to modify source in order to take advantage of this feature, the Catalog EXTDMPX directive is the central point of control for initiating the move TSA capability. It has the SYSTSA keyword added that enables the TSM, M.PTSK, and SYSGEN specifications for moving the TSA. The user can position the TSA by using one of the following:

- CATALOG EXTDMPX directive
- TSM EXTDMPX directive
- M.PTSK SVC call
- SYSGEN EXTDMPX directive

See chapter 7 of this Software Release Notes and *MPX-32 Reference Manual Volume I*, Chapter 3, for more information on the Extended TSA Feature.

TSM

Several enhancements have been added to TSM.

- The ENABLE and DISABLE directives allow the System Administrator to enable or disable one or all TSM devices for logon.
- The RECALL directive is available to recall selected lines of the Command Line Recall and Edit recall buffer. A line from the recalled buffer can be edited and reprocessed.
- To re-establish your logon directory, the CHANGE DIRECTORY directive can be specified without a directory specification. To re-establish your logon project, the CHANGE PROJECT directive can be specified without a project specification.
- The M.TSMPC system service allows access to tab settings, project name, volume name, and directory name information.

Refer to *MPX-32 Reference Manual Volume II* for the new TSM directives. Refer to *MPX-32 Reference Manual Volume I* for the M.TSMPC and M_TSMPC system services.

System Debugger

Several new System Debugger directives allow debugging in physical and virtual (logical) addressing modes. The BA directive can be used to define or redefine the physical or virtual attributes of a base. The RB directive resets all bases to their initial default values. The functionality of base N has been enhanced for Extended TSA and Mapped Out execution. Base U has been defined. The PV and VP directives convert physical addresses to virtual and virtual addresses to physical, respectively. The PS directive now displays the program status doubleword (PSD).

For more information, refer to *MPX-32 Reference Manual Volume III*, Chapter 8.

J.INIT Patch Directives

The keyword MAPOUT has been added to the conditional patch directives /T and /F to selectively modify mapped out images as well as extended and nonextended images within a single patch file.

Refer to *MPX-32 Reference Manual Volume III*, Chapter 9 for details.

Performance Improvements for System Services

The M.CALL nonbase system service has been updated. An SVC type 0 call is compatible with previous revisions of MPX-32 and maintains the blocking state at the time of the call. SVC type 6 calls provide unblocked interrupt processing. The M.OPEN service has been optimized to make an SVC type 3 call only if a scheduling event occurs while M.SHUT is in effect.

For more information, refer to *MPX-32 Technical Manual Volume I*, Chapter 1.

Include a Partition in the Task's Address Space

A static or dynamic partition can now be included at a logical address specified during task execution. This is handled by the M.INCLUDE service when a caller notification packet (CNP) is supplied with bit 4 of the option field set.

For more information, refer to chapter 7 of this Software Release Notes and to the M.INCLUDE service in Chapter 6 (nonbase) or the M_INCLUDE service in Chapter 7 (base) of *MPX-32 Reference Manual Volume I.*

Installing the Master SDT

Three new image files are supplied on the Master SDT tape:

- MSTRALL is the nonextended image for all CONCEPT/32 systems.
- MSTREXT is the extended image for all CONCEPT/32 systems.
- MSTROUT is the mapped out image for CONCEPT 32/2000 systems only.

Associated symbol table and directive files are also supplied.

When booting the system from the Master SDT, the user may now choose to select an extended image or non-extended image. The MSTRALL image is the default image selected for CONCEPT 32/67, CONCEPT 32/87, CONCEPT 32/97 and SelCONNECTION systems. The MSTROUT image is the default image selected for CONCEPT 32/2000 systems, and cannot be optionally selected.

The HSDP disk handler is now included on the master image and in the installation prompt requesting the disk controller type being used. If an improper disk type is entered at this prompt, J.VFMT will detect the error during execution and abort.

In addition, a macro (INSTALLSDT) has been added to automatically restore Master SDT images to disk.

Refer to *MPX-32 Reference Manual Volume III*, Chapter 2, for more information.

Automatic Console Configuration

MPX-32 now has the ability to dynamically configure a windowing console for a CONCEPT 32/2000 system. Refer to the MACHINE directive in *MPX-32 Reference Manual Volume III*, Chapter 7, for more information on this enhancement.

RMSS Support

MPX-32 Revision 3.5 supports the Reflective Memory System Software (RMSS) Revision 3.0. Support for remote disk resource access allows a task on any node of a multiple node reflective memory configuration to access disk resources located on a host node of that configuration and is transparent to the task.

MPX-32 also allows systems to be configured with no system volume for use with RMSS Revision 3.0 by specifying the SYSGEN NOSYSVOL directive.

J.TSM can be directed to exit when inactive by specifying the SYSGEN TSMEXIT directive.

Refer to the *RMSS Revision 3.0 User's Guide* and *Software Release Notes* for more information. The SYSGEN directives are documented in *MPX-32 Reference Manual Volume III*, Chapter 7.

Byte Granularity Disk Read Support

The byte granularity enhancement allows the user to read from unblocked disk files specifying a byte offset zero relative to the start of the file. The read starts at the specified location and continues for the specified transfer count unless truncation occurs.

Refer to the FCB structure in Appendix L of the *MPX-32 Reference Manual*, for more details.

Cyclic HSD I/O Support

This enhancement decreases the overhead of cyclic I/O from the Execute Channel Program (EXCPM) to the High Speed Data (HSD) interface. Cyclic I/O indicates that the I/O command list (IOCL) is issued repetitively with the same buffers and transfer counts to the HSD via the EXCPM entry point. As part of the cyclic I/O overhead reduction, this enhancement also includes permanent I/O queue support.

Refer to the FCB structure for the HSD in *MPX-32 Reference Manual*, Appendix L, and the H.HSDG section of *MPX-32 Technical Manual Volume II* for more information on this enhancement.

MIDL/MEML/Memory

Memory allocation and deallocation is implemented using a linked list to describe the free queue for each type of memory. The table that contains the pointer information is the MPTL (memory allocation pointer list). Each entry is one word corresponding to a map block and contains forward/backward pointers when linked to the free page queues. The MPTL size is one map block when physical memory is 16MB and it is placed in a location which is unmapped from MPX-32. Therefore, it is not included in the logical address space of any task.

A new memory type (D) is added to MPX-32 to support DRAM memory on the CONCEPT 32/2000 processor. SRAM memory on this processor is described by software as E, H, or S.

To support 256MB of physical memory on the CONCEPT 32/2000, the MIDL entries are increased from halfword to fullword entries. The new one word format is valid on CONCEPT 32/2000 processors only. All other CONCEPT processors still use halfword MIDLs. The MATA and MEML bit interpretations changed for all CONCEPT processors.

Refer to *MPX-32 Technical Manual Volume I*, Chapter 2, for more information.

Communications Region

This enhancement reserves 16 words in the Communications Region of MPX-32 for third party vendors. The Communications Region variables are expanded by 128 words positioned just before C.TABLES. C.TABLES is equated to X'D00' and contains the address of the tables. The next 32 words (X'D04' to X'D80') are the fixed user area. The next 80 words are reserved for MPX-32 and the last 16 words are reserved for third party vendors.

Refer to the next section and *MPX-32 Technical Manual Volume I*, Chapter 2, for more information.

CHANGES TO THE SYSTEM STRUCTURE

The operating system's communications region is expanded in MPX-32 Revision 3.5. This change only has an impact on those users who currently specify (using the CDOTS directive in SYSGEN) a user C. area greater than 32 words. These users must re-assemble using the new C.USERVA which contains the address of the new variable size C. area. The existing C. area remains in place but is fixed at 32 words.

New C.s are added to contain the size of each system table. SYSGEN stores these values using the equate from the macro library. This negates future user impact when a system table is expanded. The dispatch queue entry table (DQE) and shared memory table (SMT) are increased and their sizes are in C.DQESIZ and C.SMTSIZ respectively.

The macro library is restructured to eliminate the possibility of a variable having different values in more than one macro. The variables within the tables are in table structure order and equates instead of offsets are used wherever possible. No macros have been removed or eliminated. Some EQUs in the macro library are changed to DEQUs to support the use of these macros in auto-sectioned code.

T.MIDL1 and T.MIDL2 are no longer at X'728'. New locations are assigned to support fullword MIDLs. This change only affects code which accesses these variables.

Existing macro equates which are changed:

```
CC.SVC8  EQU    C.CDOTND+1W   CUSTOM ADDR - SVC '8' TABLES
T.REGS   EQU    X'00900'      PUSHDOWN STACK  (362W)
T.MEML1  EQU    X'00724'      RESERVED FOR MEML STORAGE DW
T.MIDL1  EQU    X'008C8'      RESERVED FOR MIDL STORAGE DW
T.MIDL2  EQU    X'008CA'      RESERVED FOR MIDL STORAGE HW
T.MIDL   EQU    X'02000'      MAP IMAGE DESCR LIST
```

The new macro equates which are in the library are:

Communications Region

```
C.BRFRM  EQU    X'20'         BASE REGISTER FRAME SIZE
C.FRMSIZ EQU    X'80'         STACK FRAME SIZE IS 32W
C.MAXFRM EQU    X'14'         20 FRAMES IN STACK NOW
C.SWPRD  EQU    X'00880'      FOR FUTURE USE BY SWAPPER
C.SWPDEV EQU    X'00884'      CHANNEL/SUBADDR OF SWAP DEV
C.IREGS  EQU    X'00888'      STACK IN TSA OF TASK IN IPU
C.ITSAD  EQU    X'0088C'      TSA OF TASK IN IPU
C.LOSEND EQU    X'00890'      LOGICAL END O.S. PER CONTEXT
C.POSEND EQU    X'00894'      PHYSICAL END O.S.
C.DPGPRI EQU    X'00CD5'      1 BYTE - DP BASE PRIORITY
C.NFRAME EQU    X'00CD6'      1 BYTE - # FRAMES IN STACK
C.MRQLEN EQU    X'00CD7'      FIXED MRQ LENGTH
C.TABLES EQU    X'00D00'      HOLDS ADDR OF MPX-32 TBLS
C.USER   EQU    X'00D04'      FIXED SIZE USER AREA
C.USERND EQU    X'00D80'      END 32W FIXED USER AREA
C.USERVA EQU    X'00D84'      ADDR OF VARIABLE USER AREA
C.TPVA   EQU    X'00D88'      ADDR OF 3RD PARTY VEND. AREA
C.EXEND  EQU    X'00D8C'      1HW - END EXECUTABLE MEMORY
C.FRAME  EQU    X'00D8E'      1HW - STACK FRAME SIZE
C.SCOFDQ EQU    X'00D90'      SCRATCHPAD OFFSET FOR DQE
C.CTSAD  EQU    X'00D94'      TSA ADDR OF CPU TASK
C.AGE    EQU    X'00D98'      VIRTUAL TIME TILL PAGE AGED
C.EFRPG  EQU    X'00D9C'      FREE PAGE HEAD CELL E MEMORY
C.HFRPG  EQU    X'00DA0'      FREE PAGE HEAD CELL H MEMORY
C.SFRPG  EQU    X'00DA4'      FREE PAGE HEAD CELL S MEMORY
C.DFRPG  EQU    X'00DA8'      FREE PAGE HEAD CELL D MEMORY
C.MPFRPG EQU    X'00DAC'      FREE PAGE HC MULTI PROC MEM
C.MPTLA  EQU    X'00DB0'      MEM PTR LIST TABLE ADDRESS
C.CREGS  EQU    X'00DB4'      STACK ADDR FOR CURR CPU TASK
C.MPMAC  EQU    X'00DC6'      1HW TOTAL MULTI MEM AVAIL
C.BEGPGO EQU    X'00DC8'      # MAPS TO BEGIN PAGEOUT
C.ENDPGO EQU    X'00DCA'      # MAPS TO STOP PAGEOUT
C.DMCC   EQU    X'00DCC'      TOTAL CONFIGURED DRAM 1H
```

```
C.DMAC   EQU    X'00DCE'     TOTAL AVAILABLE DRAM 1H
C.BIT1   EQU    X'00DD0'     NEW DW OF BITS
C.COMM   EQU    X'00DE0'     1DW RESERVED FOR COMM-32 PRODUCTS
C.CDOTND EQU    X'00DFC'     END OF CDOTS FOR 3.5
```

New bits in C.BIT are:

```
C.NOSVOL EQU    X'0003C'     NO SYSTEM VOLUME
C.MAPOUT EQU    X'0003D'     SET FOR MAPPED OUT IMAGE
C.NOECHO EQU    X'0003E'     NO ECHO ON OWNERNAME
C.TSA    EQU    X'0003F'     MOVE TSA ON ALL TASKS
```

Bits in new C.BIT1 are:

```
C.TERM   EQU    X'00000'     INHIBIT BATCH MSG TO ALL TY
C.CONS   EQU    X'00001'     INHIBIT BATCH MSG TO CONSOLE
C.NOSYS  EQU    X'00002'     RESTRICT SYSTEM TO 1 TERMINAL
C.TSKOUT EQU    X'00003'     SYSTEM DEFAULT - TSKS RUN MAPOUT
C.NOLACC EQU    X'00005'     NO LAST ACCESS UPDATE
C.DPGSYS EQU    X'00006'     IMAGE SUPPORTS DEMAND PAGE
C.TSMXIT EQU    X'00007'     IF SET TSM EXITS WHEN DONE
C.RACTSM EQU    X'00008'     IF SET TSM IS REACTIVATED
C.ACTSM  EQU    X'00009'     IF SET ACTIVATE TSM
C.PGOPRG EQU    X'0000A'     IF SET PAGE OUT IS IN PROGRESS
C.HIPRI  EQU    X'0000B'     SCHEDULING EVENT HAS OCCURRED
```

File Assignment Table (FAT)

```
DFT.RMIX EQU    X'00030'     REMOTE INDEX DUAL USE FOR WORD 12
```

Bit in DFT.FLGS:

```
DFT.REMR EQU    7            RMSS SUPPORT
```

Dispatch Queue Entry (DQE)

```
DQE.TSAP EQU    X'000A4'     PHYSICAL ADDRESS TSA
DQE.PGOL EQU    X'000AC'     PAGE OUT LIST HEAD CELL
```

Bits in DQE.TSKF:

```
DQE.MAPO EQU    X'0003'      TASK RUNS WITH MPX MAPPED OUT
DQE.NPGO EQU    X'0006'      IF SET, INHIBITS PAGEOUT
```

New wait function codes in DQE.GQFN are:

```
QWTSM    EQU    X'00013'     QUE WAITING FOR TSM
```

File Control Block (FCB)

```
FCB.SKCT EQU    10W             SKIP CNT ON READ W/BYTE GRAN
FCB.OSB  EQU    14              BUFFER IN OS (RESERVED FOR MPX-32)
```

Map Image Descriptor (MIDL)

```
MIDL.SEL EQU    5               32/2000 IF SET...MEMORY IS DRAM
```

Memory Management equates

```
MM.OFFS  EQU    16 OR 0, DEPENDING ON THE FLAG C.FWMID
MM.SHFT  EQU    1 OR 2, DEPENDING ON THE FLAG C.FWMID
MM.SIZE  EQU    MM.SHFT*2
```

Memory or Run Request Queue (MRRQ)

```
MQ.EAPSD EQU    X'00020'      END ACTION PSD W1,W2
MQ.PPTR  EQU    X'00028'      PARAMETER AREA POINTER
MQ.HSIZE EQU    12W           CURRENT LENGTH OF MPX MRRQ
MQ.RTNST EQU    MQ.HSIZE      APPEND TO END OF MRRQ HDR
MQ.NIDST EQU    MQ.RTNST+1W   REMOTE TASK AND NODE ID
RMQ.SIZE EQU    MQ.HSIZE+2W   SIZE OF RMR1 - REMOTE MRRQ
```

Resource Requirement Summary (RRS) types 1 and 6

```
RR.RTA   EQU    X'00011'      REMOTE RESOURCE IS TASK
                              (RESERVED FOR RMSS USE ONLY)
```

Preamble equates

```
PR.FLAG4 EQU    X'000B6'      NEW FLAG FIELD - 1H
PR.CATD  EQU    X'000C4'      SECTOR PTR. CATL DIRECTIVES
PR.CVER  EQU    X'000C8'      VERSION
PR.LMUI  EQU    X'000CC'      8W FOR CATALOG
PR.ALMID EQU    X'000EC'      RESERVED FOR CATALOG
PR.PGO2  EQU    X'00104'      PROGRAM OPTION WORD TWO
PR.AGE   EQU    X'00108'      VIRTUAL TIME TO AGE
PR.PRTGC EQU    X'0010C'      # PROTECTION GRANULES IN CSECT
PR.PRTGD EQU    X'0010E'      # PROTECTION GRANULES IN DSECT
PR.PRTGG EQU    X'00110'      # PROTECTION GRANULES IN GLOBAL
PR.PRTGE EQU    X'00112'      RESERVED FOR EXTENDED GRANULES
```

New bits in PR.FLAG3 are:

```
PR3.ETSA EQU    X'00006'      MOVE TSA TO EXTENDED ENABLED
PR3.RTSA EQU    X'00007'      MOVE TSA REQUESTED
```

New bits in PR.FLAG4:

```
PR4.EMAP EQU    X'00000'      SET, ENABLE MAPOUT
PR4.RMAP EQU    X'00001'      SET, MAPOUT OPTION REQUESTED
PR4.DPG  EQU    X'00002'      SET, DEMAND PAGE THIS TASK
PR4.NDPG EQU    X'00003'      SET, DON'T DEMAND PAGE TASK
```

Shared Memory Table (SMT)

New bit in SMT.FLAG:

```
SMT.LIN  EQU    25            LOGICAL INCLUDE ON THIS SMT
```

Task Service Area (TSA)

```
T.DFCB   EQU    X'00568'      16W FOR DEMAND PAGE FCB
T.TSAOR  EQU    X'00880'      TSA ORIGIN
T.TSASZ  EQU    X'00882'      TSA SIZE IN MAPS
T.ATBIAS EQU    X'00884'      ACTIVATION BIAS
T.LASTP  EQU    X'008C4'      ADDRESS OF LAST STACK PUSH
T.TSAEND EQU    X'008D0'      LOGICAL END TSA
T.ADRTSA EQU    X'008D8'      TSA ADDR FROM EXTDMPX
T.ATADDR EQU    X'008DC'      ACTIVATION TIME BIAS ADDRESS
```

New bits in T.BIT5 are:

```
T.MAPOUT EQU    X'0000A'      TASK RUNS WITH MPX MAPOUT
T.MVTSA  EQU    X'0000B'      MOVE TSA REQUESTED.
T.RTRNOS EQU    X'0000C'      RETURN TO AID OR OS
```

Resource Descriptor (RD)

```
RD.RDOWN EQU    X'000D0'      LAST ACCESS OWNER NAME
```

Unit Definition Table (UDT)

New bits in UDT.BIT3:

```
UDT.HSDP EQU    X'00002'      SET IF HSDP DEVICE
```

New macros added to support fullword MIDL conversions are:

```
M.MIDS   DEFM
M.LDMID  DEFM   P1,P2,P3
M.STMID  DEFM   P1,P2,P3
M.FWHWX  DEFM   RN
```

New macros added to support Mapped Out MPX-32 are:

```
M.PSDGEN DEFM   PC,TYPE,B0,LHW,RHW
M.LF     DEFM   ADDR,XREG
M.STF    DEFM   ADDR,XREG
M.LFBR   DEFM   ADDR,XREG
M.STFBR  DEFM   ADDR,XREG
```

New macros added to support AID debugger are:

```
M.OSREAD DEFM   ARG1,ARG2,ARG3  READ OS LOCATION
M.OSWRIT DEFM   ARG1,ARG2,ARG3  WRITE TO OS LOCATION
M.RTRNOS DEFM                   RETURN TO OS FROM DEBUGGER
```

New macros added to support moving of TSA are:

```
M.GTSAD  DEFM                   GET TSA ADDRESS VIA SVC
M.TSAD   DEFM   R               MACRO TO GET TSA ADDRESS
```

New SYSGEN equates are:

```
G_FWMID  EQU    1               SET = FULLWORD MIDS IN USE
G_MAPFL  EQU    0               SET = MAPOUT/NOMAPOUT SPEC'D
```

5 INSTALLATION INSTRUCTIONS

MPX-32 Revision 3.5 is a Master System Distribution Tape (SDT) and can be installed using the usual SDT installation procedures. Refer to the *MPX-32 Reference Manual Volume III* for these procedures. In addition, see the "IPL Process" section in chapter 7 of this SRN before installing the SDT.

For installation on a CONCEPT 32/2000 system, follow the instructions listed below.

Due to incomplete system integration and testing of MPX-32 Revision 3.5 executing in Mapped Out mode and Demand Page on CONCEPT 32/2000 computers, Encore recommends that users do not install or run production workloads utilizing either the Mapped Out or the Demand Page features at this time. A forthcoming revision, 3.5U01, is scheduled to provide fully integrated and tested support of these and other CONCEPT 32/2000 hardware enhancements. Since the default master image on the CONCEPT 32/2000 is the mapped out image (MSTROUT), we recommend that the following installation procedure be followed when installing or upgrading to MPX-32 Revision 3.5 on CONCEPT 32/2000 systems.

Refer to *CONCEPT 32/2000 Operations* for details on initializing the Amiga PC windowing console and activating windows.

Select MSTRALL as the Master SDT image when booting a CONCEPT 32/2000 computer, as follows:

> Enter panel mode by activating the Panel Window on the windowing console.
>
> Halt the system, if not already halted, by selecting the `halt` command from the Panel Window menu.
>
> Reset the system by selecting the `reset` command from the Panel Window menu.
>
> Clear memory by selecting the `clear mem` command from the Panel Window menu.
>
> Override the default system image by changing the contents of GPR0 to `gpr0:   00000001.`
>
> Load the system by selecting the `ipl` command from the Panel Window menu.

Enter the tape unit's address in the sub-window.

At the debugger prompt (>>), enter TE.

```
>>TE

BOOT FROM A SCSI TAPE? (REPLY Y OR N):
```

Enter Y if booting from a SCSI tape drive.

```
MPX-32 MASTER SDT FOR CONCEPT 32/2000 COMPUTERS
```

To complete the installation of the Master SDT, follow the instructions in *MPX-32 Reference Manual Volume III*, Chapter 2, from the bottom of page 2-13 through page 2-20.

6 RESOLVED SPRs

This section lists the Software Problem Reports (SPRs) filed against MPX-32 that were resolved during the development of revision 3.5. For more information on a particular SPR, consult the SPR database.

Customers who subscribe to the Software Update Service (SUS) can check on the status of any SPR by logging on to a system at Encore Computer Corporation in Fort Lauderdale. This procedure is discussed in the section, Online Viewing of SPRs.

To learn how to report new problems with MPX-32 Revision 3.5 software and documents, refer to the section, MPX-32 Problem Reporting.

LISTING OF RESOLVED SPRs

86000347	87001119	88001014	88002327
86000591	87001120	88001031	89000000
86000645	87001149	88001034	89000006
86000673	87001162	88001123	89000040
86000731	87001199	88001239	89000089
86000808	87001266	88001287	89000121
86000845	87001315	88001329	89000130
86000869	87001343	88001570	89000152
86001173	87001542	88001594	89000155
86001280	87001618	88001598	89000159
86001293	87001623	88001608	89000221
86001367	87001668	88001625	89000226
86001471	87001675	88001626	89000309
87000027	87001684	88001672	89000313
87000177	87001690	88001673	89000335
87000219	87001709	88001697	89000358
87000391	87001809	88001745	89000408
87000392	87001892	88001854	89000482
87000413	87002013	88001856	89000520
87000414	87002135	88001865	89000547
87000416	88000025	88001908	89000549
87000417	88000159	88001912	89000590
87000418	88000309	88001937	89000601
87000422	88000530	88001990	89000609
87000424	88000537	88002025	89000626
87000530	88000539	88002039	89000635
87000531	88000547	88002040	89000650
87000533	88000649	88002041	89000659

87000587	88000669	88002117	89000689
87000647	88000751	88002142	89000712
87000658	88000763	88002153	89000726
87000689	88000827	88002171	89000727
87000778	88000882	88002187	89000761
87000799	88000890	88002249	89000890
87000889	88000985	88002264	89000906
87001070	88001002	88002265	89000910
			90000130

ONLINE VIEWING OF SPRs

Encore Computer Corporation Customer Services maintains a database for SPRs that is accessible to SUS customers. If you are an SUS customer, use a modem to log on to a UTX/32 system at Encore Computer Corporation in Fort Lauderdale. Dial (305) 797-5837 and log in as `cstspr`. Be sure the modem is set to:

- 1200 baud
- full duplex
- 8-bit characters
- one stop bit
- no parity

Call 1-800-TECHAID to report any difficulties with the online database.

PROBLEM REPORTING

There are two ways to report problems with MPX-32 Revision 3.5 software and documents:

- call a toll-free telephone number
- fill out an SPR form

The following sections give more information on the use of these methods.

By Telephone

To report a software or documentation problem by telephone, within the U. S. A., call a customer service representative at 1-800-TECHAID. You are asked to provide the following information:

- the call number (if this is a follow-up call)
- your name
- company name
- contract number
- system number
- telephone number
- machine type
- operating system and revision number
- product and revision number
- problem description

With SPR Forms

A software or documentation problem can be reported by filling out an SPR form and sending the form with the appropriate supplemental information to the address indicated on the form. Forms can be obtained from your local Encore Computer Corporation office.

7 SPECIAL CONSIDERATIONS FOR REVISION 3.5

IPL PROCESS

MPX-32 Revision 3.5 contains many dependencies between resident and non-resident modules. Therefore, you will need to perform an IPL from tape in order to upgrade to MPX-32 Revision 3.5. Use the REPLACE or FORMAT directive at the Volume Formatter prompt to ensure that the master image is installed on the system volume as the default image and the new 3.5 disk bootstrap has been updated on your system. After installing from the SDT tape, build a new default image. If any problems are encountered, an IPL from disk will return the system to the master image.

New master images are larger than the previous ISIZE default in J.VFMT. If upgrading to MPX-32 Revision 3.5, space allocated for the system image on the system volume must be 440 blocks or more, or the volume must be reformatted.

EXTENDED TSA (NONBASE TASKS ONLY)

General Information

The Extended TSA feature is optionally available to any nonbase task executing in the MPX-32 Revision 3.5 environment. This feature allows the user to move the task's TSA into the task's extended address space. Positioning the TSA in the task's extended address space results in more direct (execution) address space for code and directly addressable data.

The TSA varies in size from task to task and is greatly increased by the number and type of I/O resources required by the task. Moving the TSA will increase the user task's direct address space by a minimum of 2 map blocks for tasks with minimal I/O requirements to many more map blocks for tasks with large I/O requirements. Moving the task's TSA will cause no performance degradation.

Tasks Cataloged without the TSA or SYSTSA keywords on the Catalog EXTDMPX directive will have their TSA and extended MPX-32 positioned in the compatible (NOTSA) pre-MPX-32 Revision 3.5 location. When a task is Cataloged using the TSA or NOTSA option on the Catalog EXTDMPX directive any TSM, M.PTSK, or SYSGEN request is ignored. When the task is Cataloged with the SYSTSA option on the Catalog EXTDMPX directive the position of the TSA and extended MPX-32 (if present) will be determined by using the TSM, M.PTSK, or SYSGEN EXTDMPX request in respective order.

The position of the task's TSA is determined during task activation and is fixed for the duration of the task. Each task may choose the position of its TSA.

When the task is executing on an extended MPX-32 image, the extended section of MPX-32 is positioned logically above and contiguous with the TSA. The TSA is positioned at the logical address specified by the EXTDMPX directive keyword; MINADDR, MAXADDR, or map block number. On non-extended MPX-32 images only the TSA option is applicable.

New Communication Region Equates

The Extended TSA feature requires new Communication Region equates be defined and added into MPX-32.

The communication region equates C.TSAD and C.REGS have been maintained for tasks executing in the compatible mode although their use is discouraged. They must not be used by any resident module or non-resident task when the TSA is moved. Any other method for calculating the TSA base address should be carefully inspected for correctness if a moved TSA will be referenced.

C.TSAD was previously used as the index to the start of the TSA for a task running in either the CPU or in the IPU. C.TSAD was the end of MPX-32 plus one byte (excluding extended MPX-32). Presently, C.TSAD is valid only when the task is running with MPX-32 mapped into its logical address space and the task's TSA has not been moved.

C.REGS was previously used to get to the start of the current CPU task's TSA or to the beginning of that task's pushdown/popup stack. T.REGS is the offset from the start of the TSA to the stack. In MPX-32 Revision 3.5, the stack has been moved from the start of the TSA (T.REGS=0) to X'900' into the TSA (T.REGS=X'900'). C.REGS can not be used as a direct index to the stack and code using T.REGS must be reassembled using the new macro library equate value for T.REGS.

New communication variables C.CTSAD, C.CREGS, C.ITSAD, and C.IREGS have been defined and are maintained by their respective CPU (C.C*xxxx*) or IPU (C.I*xxxx*) schedulers.

C.CTSAD and C.ITSAD contain the logical address for the start of the TSA of the task executing in the CPU or IPU, respectively. C.CREGS and C.IREGS contain the logical address for the start of the TSA pushdown/popup stack of the task executing in the CPU or IPU, respectively. These variables may be used by code that is executing in the appropriate processor although their use is discouraged.

A more generic method for determining the TSA address is by using the new M.TSAD macro or the M.GTSAD/M_GTSAD system service call. Resident code sequences that may be executed by either processor must use the M.TSAD macro to obtain the TSA address for the task that is current in that processor. Non-resident tasks that may execute in either processor must use the M.GTSAD (nonbase) or M_GTSAD (base) SVC to obtain their TSA address.

How to Relocate the TSA

User modules that reside within MPX-32 and do not reference data within the TSA of non-resident tasks do not require code changes.

User modules that reside within MPX-32 and reference data within the TSA of non-resident tasks may require code changes:

- When the non-resident task's TSA is repositioned at other than MINADDR, extended addressing must be set to reference data within that task's TSA.
- Resident modules using C.TSAD to obtain the logical start of the non-resident task's TSA must now use the M.TSAD macro to obtain the task's TSA address when it is positioned at other than MINADDR.
- Any I/O done into or out of the TSA must use a 16-word FCB since an 8-word FCB's TCW cannot address the TSA in extended memory.

For tasks not affected by the size of their TSA, MINADDR with the NOTSA keyword is the compatible mode of operation.

Tasks that are affected by the size of their TSA and reference data structures within their TSA may require some code changes before repositioning their TSA.

- When the TSA is positioned at other than MINADDR, the task must set extended addressing to reference data within its TSA.
- Tasks using C.TSAD or C.REGS to obtain the logical start of their TSA must now use the M.GTSAD system service call to obtain their TSA address when it is positioned at other than MINADDR.

RELOCATION OF TASK'S PUSHDOWN/POPUP STACK AND MIDLS WITHIN THE TASK

The task's MIDLs have been moved from X'900' to X'2000' and the task's pushdown/popup stack has been moved from X'0' to X'900' displacement in the TSA. The minimum size of the TSA is now 2 map blocks; the number of pushdown/popup frames in the stack has been increased to 20; and new C.s and T.s are established. Therefore, code sequences that reference the TSA stack and use the new C.s and T.s are not dependent on a particular revision of MPX-32.

Since these are not optional features, any module or task that references T.MIDL, the MIDLs, the MEMLs, T.REGS, or the pushdown/popup stack may be affected:

- If a code sequence uses T.MIDL or T.REGS to calculate the start of the MIDLs or stack, that code will need to be re-assembled using the new values for T.MIDL or T.REGS. Modification to use T.MIDLA or the new M.TSAD macro or M.GTSAD SVC call should also be considered.
- Modifying code sequences to use the new C.FRAME value will assure that code sequences assuming a stack frame size of 32 words will not require modification if the frame size should increase in later revisions of MPX-32.

The MIDLs and MEMLs have been expanded to full words on images built specifically for CONCEPT 32/2000 processors. This makes use of the CONCEPT 32/2000's greater available address space. Resident modules that reference the MIDLs or MEMLs can make use of new macros M.LDMID and M.STMID; and new equates MM.SIZE, MM.OFFS, and MM.SHIFT for manipulating these structures. Non-resident tasks that reference the MIDLs or MEMLs can use an alternate execution path when C.MAPOUT is set to process the MIDLs or MEMLs as full words.

MAPPED OUT OPTION (CONCEPT 32/2000 ONLY)

MPX-32 Revision 3.5 supports a new, optional execution mode termed "Mapped Out." This execution mode is available only on CONCEPT 32/2000 CPUs. In this mode, resident MPX-32 supports task-level code for nonbase tasks starting at logical address zero (e.g. the OS is not included in the address space of the task). Mechanisms employed to accomplish this include new CPU instructions, new Macro Assembler directives, and revisions to MPX-32 code. Detailed information on the mechanisms required will be contained in the documentation accompanying the MPX-32 Revision 3.5U01 software.

The CONCEPT 32/2000 CPU has an enhanced instruction set that includes new 'through context' instructions. Each 'through context' instruction causes the CPU to do a 'one shot' address translation using its map registers regardless of the mapped or unmapped state at the time. The 'through context' instructions enable MPX-32 to reference data within the task's logical address space while MPX-32 is executing in the unmapped mode. The 'standard' instructions and their corresponding enhanced 'through context' instructions are:

Standard Instructions	Enhanced Instructions
LB	LCB
LH	LCH
LW	LCW
LD	LCD
STB	STCB
STH	STCH
STW	STCW
STD	STCD
LEAR	LCRA

The Macro Assembler Revision 3.2 has been enhanced to recognize new directives REIS (Reset Enhanced Instruction Set) and SEIS (Set Enhanced Instruction Set). The REIS directive instructs the assembler to assemble subsequent 'through context' instructions using their corresponding standard instruction opcode and disregard the enhanced instruction set (i.e., in the REIS mode the assembler will assemble a LCD as a LD, etc.). The SEIS directive instructs the assembler to assemble subsequent 'through context' instructions with their enhanced instruction opcode (i.e., in the SEIS mode the assembler will assemble a LCD with the LCD opcode, etc.). The MPX_NON and MPX_EXT PRE files set the REIS mode and define an assembly flag named C.MPXOUT to be false. A new MPX_OUT PRE file sets the SEIS mode and defines C.MPXOUT to be true. This allows one copy of an MPX-32 source file to be assembled in the REIS or SEIS mode based on the MPX-32 PRE file used. It further allows conditional assembly to be done within each source file based upon the C.MPXOUT flag setting.

Load modules built with a previous version of the Utilities Release 3.2 Cataloger run in the compatible mode with MPX-32 mapped into their address space. Load modules built without the NOMAPOUT, MAPOUT, or SYSMAP keywords on the Catalog ENVIRONMENT directive will run in the compatible mode. When a task is Cataloged using the MAPOUT or NOMAPOUT option on the Catalog ENVIRONMENT directive any TSM, M.PTSK, or SYSGEN request is ignored. When the task is Cataloged with the SYSMAP option on the Catalog ENVIRONMENT directive the mapped out or mapped in option may be determined by using the TSM, M.PTSK, or SYSGEN MAPOUT or NOMAPOUT request in respective order. Attempts to execute a mapped out task on other than a mapped out MPX-32 system image are ignored and the task executes mapped in.

The mapped state of MPX-32 with respect to the task is determined during task loading time and is fixed for the duration of the task.

New Communication Region Equates

The Mapped Out feature requires new Communication Region equates be defined and added into MPX-32.

C.TSAD was previously used for address compares of data and program counters by MPX-32 to determine if they were within MPX-32. C.TSAD is the logical end of MPX-32 plus one byte (excluding extended MPX-32) or the logical start of the task's address space. Presently, C.TSAD is valid only when the task is running with MPX-32 mapped into its logical address space (the compatible mode). Two new communication variables have been defined to address this problem. C.POSEND is established by SYSGEN and is the physical end of MPX-32 plus one byte (excluding extended MPX-32). C.POSEND can be used to verify data and program counter value for physical addresses. C.LOSEND is established by the task schedulers upon each context switch and represents the logical end of MPX-32 plus one byte (excluding extended MPX-32) for the current task. C.LOSEND can be used to verify data and program counter values for logical addresses. On mapped in images C.LOSEND will be equivalent to C.POSEND at all times. On a mapped out image C.LOSEND will be equal to C.POSEND when the current task is running mapped in and C.LOSEND will be 0 when the current task is running mapped out.

Conversion to Run in the Mapped Out Mode

A complete list of the guidelines required to convert resident code to execute in the Mapped Out environment is beyond the scope of this SRN section. However, some considerations include:

1. Any memory references to task address space must use the new 'through context' instructions available with CONCEPT 32/2000 CPUs.

2. Resident code itself executes in the unmapped mode (e.g. PSD bit 32 reset).

3. The data move subroutines S.EXEC54 and S.IOCS19 are redefined to S.EXC54A and S.EXC54B as well as S.IOS19A, S.IOS19B, S.IOS19C and S.IOS19D. These routines distinguish among source address space location (task or OS) and target address space location (task or OS). Resident code that previously called either data move subroutine must call the proper routine based on source and target address space locations.

For tasks not affected by the size of MPX-32, NOMAPOUT is the compatible mode of operation.

Tasks that are affected by the size of MPX-32 and do not reference data within the resident operating system may run MAPOUT with no coding changes.

Tasks that are affected by the size of MPX-32 and do reference data within the resident operating system will require coding changes. Two methods are available to choose from: the M.OSREAD/M.OSWRIT system service calls or inclusion of the predefined .MPXTBLS memory partition.

The following considerations apply when executing application-level code in Mapped Out mode:

1. User code cannot reference any data or code in the resident operating system without invoking new system services provided for this purpose.

2. Task-level code cannot place any optionally specified routines at logical address zero (e.g., an I/O error processing routine cannot start at logical zero). By default, a zero value indicates that an optional routine has not been supplied by the user.

3. The location of any given data structure must be consistently in the task address space (logical) or in the OS address space (physical). Before MPX-32 Revision 3.5, nonbase tasks always had the OS, including memory pool, in their logical address space. Thus, intertask message data could be retrieved directly from memory pool without performing a Get Message Data service request. In MPX-32 Revision 3.5, tasks that do not include resident MPX-32 in their logical address space (e.g. run Mapped Out) can no longer directly access data in memory pool.

INTERFACE CHANGES

Changes to S.EXEC68

The subroutine S.EXEC68 has been changed in conjunction with the MRRQ data structure changes. The change in this routine is the basis for the change in no-wait end-action processing.

The subroutine S.EXEC68 in MPX-32 Revision 3.4 U03 was:

```
S.EXEC68, CONSTRUCT AND VECTOR TO USER END ACTION PSD

        ENTER:
            R1=PSB OR FCB ADDRESS
            R6=USER END ACTION ROUTINE ADDRESS
            BL      S.EXEC68
        NO RETURN
```

The new interface for S.EXEC68 in MPX-32 Revision 3.5 is:

```
S.EXEC68, VECTOR TO END ACTION PSD

     ENTER:
          R1=PSB OR FCB ADDRESS
          R6=USER END ACTION ROUTINE ADDRESS  PSD1
          R7=USER END ACTION ROUTINE ADDRESS  PDS2
          BL       S.EXEC68
     NO RETURN
          R2=PSB ADDR
          R3=C.CURR
```

The subroutine S.EXEC68 no longer constructs the target PSD. Instead, the PSD is passed from the IOQ or MRRQ entry. Any user code referencing this subroutine must supply the end-action PSD in R6 and R7 or unpredictable results will occur.

Other Changes

The common subroutines of the routines H.EXEC and H.EXEC2 have been extracted and put into a new file, H.EXSUB. Therefore, downdates of MPX-32 Revision 3.4 U03 and MPX-32 Revision 3.5 will be impractical and patches may have to be re-worked since H.EXSUB can run in extended MPX-32. All user handlers running at interrupt level using any of the subroutines in H.EXSUB in extended mode must be converted. Otherwise, H.EXSUB should not be included in the extended modules.

USING THE 'INCLUDE AT' FEATURE

The M.INCLUDE/M_INCLUDE service is enhanced to allow inclusion of a static or dynamic partition at a specified logical address within the task. When a CNP is supplied to the include service with bit 4 of the option field set, then the service includes the partition at the logical address supplied in CNP word 4. The SMT start page, SMT.PAGE, does not change. It is the logical page start specified at creation. Any inclusion without option field bit 4 set is included at this address. A bit, SMT.LIN, in SMT.FLAG is set when a task includes using the new feature.

Tasks running with MPX-32 mapped out of their address space which desire to access MPX-32 tables and structures use this feature to include those tables. On all images (extended, non-extended or mapped out), there is an entry in the Shared Memory Table for a static partition called .MPXTBLS. This SMT entry is automatically created by SYSGEN and the number of entries specified in the SHARE directive is increased by one. The maps describing .MPXTBLS are at the address contained in C.MIDL; no memory pool is needed. The number of pages in this partition is enough to encompass all addresses up to the first MPX-32 module (usually H.IP00) and rounded to a map block.