OM88-1020-7

Installation & Quickstart Guide

OMCS/MVS Operations Management and Control System for IBM MVS environments

Version 2 Release 9

January 2008

Manual and Product identifiers.

This manual is identified by a publication number **OM88-1020** combined with a revision level number (sometimes called a dash level) **-7**.

The licenced software product, of which this manual is a part, is identified by a **Product Name** or **Product Acronym** and an **Edition Number**. Edition numbers are made up of a **Version number** and a **Release number** within that version.

The material in this manual is intended to be used with **OMCS/MVS Version 2 Release 9** and later editions of that product. The edition number may appear in licenced material in abbreviated forms like the following: **V2R9**, **V29** and **V2**.

From time to time ASE may issue replacements for, or additions to, the pages in this manual.

Each time the entire manual is replaced by a later version then the revision level (the dash level) part of the publication identifier will be increased by 1.

Related Publications
OM99-0120 - The OMCS/MVS Product Family Concepts and Facilities
OM88-1021 - OMCS/MVS Database Utilities Reference
OM88-1022 - OMCS/MVS General Operations Guide
OM88-1023 - OMCS/ACF Access Control Facility Guide
OM88-1024 - OMCS/AMF Archive Management Facility Guide
OM88-1025 - OMCS/IRM Interactive Report Management Guide.
OM88-1026 - OMCS/JSF Job Scheduling Facility Guide.
OM88-1027 - OMCS/ECL ECL Language Reference.
OM88-1029 - OMCS/MVS Messages and Codes.

Published by
Australian Systems Engineering Pty. Ltd.
20 Andrew Street, Northcote VIC 3070 Australia
telephone (03) 9489 4033
international +61 3 9489 4033
Email: help@ase.com.au

CONTENTS

(page intentionally unused)

INTRODUCTION

Part 1 of this manual provides guidance for the installation of version 2 of the Operations Management and Control System for MVS (OMCS/MVS). **NOTE**: if you are installing a service refresh of OMCS/MVS, you probably will not have to repeat any of the tasks in parts 2 and 3 of this manual. However, you should still briefly review part 2.

Part 2 of this manual provides guidance for those installation tasks which are not part of the SMP/E installation process.

Part 3 of this manual provides a Quickstart Guide to help you get OMCS/V2 up and operational with the least effort. It is not intended to be a long-term substitute for the complete set of manuals identified in the box Related Publications inside the front cover of this manual.

ASE's support for this product began when you first requested that the install material be supplied to you. If at any subsequent time you find that the material in this or any other OMCS manual is confusing, inadequate or incorrect, or if you encounter any problem with the supplied machine-readable material or get unexpected results from the installed product, please call ASE for support.

(page intentionally unused)

INTRODUCTION TO PART 1 - SMP/E INSTALLATION GUIDE

It is beyond the scope of this manual to describe how to use SMP/E. It is assumed the reader is familiar with SMP/E. The following manuals should be consulted for further detail:

SMP/E User's Guide, SA22-7773
SMP/E Commands, SA22-7771

SMP/E Installation consists of the following activities:

- **Preparing install materials.**
  The install materials must be downloaded from the ASE web site and transferred to your z/OS system before installation can proceed.

- **Allocating the OMCS Libraries**
  Eleven OMCS libraries are built by the SMP/E install process.

- **Allocating the SMP/E datasets**
  SMP/E control datasets must be allocated before installation can proceed. Alternatively an existing SMP/E environment may be used.

- **Preparing the SMP/E environment for OMCS installation**
  OMCS installation and maintenance requires some extra SMP/E control dataset entries.

- **SMP/E Installation of OMCS**
  The standard RECEIVE, APPLY and ACCEPT process is used to install OMCS.

- **Maintenance**
  The OMCS product may be accompanied by maintenance. This contains fixes to problems and enhancements developed since the current edition of OMCS was issued. This is installed using SMP/E.

The following table lists the FMIDs which make up the OMCS base product and its optional extensions.

Fmid	Description
ASEB209	OMCS base product
ASEC209	OMCS/IRM - Interactive Report Management
ASED209	OMCS/JSF - Job Scheduling Facility
ASEE209	OMCS/CICS - CICS interface
ASEG209	OMCS/SRF - Statement Reprint Facility

PREPARING INSTALL MATERIALS

The files required to install OMCS and its optional components are available for download at the ASE web site - http://www.ase.com.au/omcs29/omcs29.htm There are two zip files for each component, one containing the SMP/E install files in XMIT format and another containing JCL to extract the install files using TSO RECEIVE command.

Each product zip file has a name of the format ***fmid*.zip** eg. aseb209.zip. The corresponding zipped JCL file has a name of the format **recv*nnnn*.zip** where ***nnnn*** is the last four characters of the fmid eg. recvb209.zip.

Here is a list of the available files. You may only download files for the OMCS components for which you are licensed.

- **aseb209.zip** OMCS V2.9 base product
- **recvb209.zip** JCL to extract the ASEB209 install files using TSO RECEIVE command
- **asec209.zip** OMCS V2.9 Interactive Report Management
- **recvc209.zip** JCL to extract the ASEC209 install files using TSO RECEIVE command
- **ased209.zip** OMCS V2.9 Job Scheduling Facility
- **recvd209.zip** JCL to extract the ASED209 install files using TSO RECEIVE command
- **asee209.zip** OMCS V2.9 CICS report viewing facility
- **recve209.zip** JCL to extract the ASEE209 install files using TSO RECEIVE command
- **aseg209.zip** OMCS V2.9 Statement Reprint Facility
- **recvg200.zip** JCL to extract the ASEG209 install files using TSO RECEIVE command
- **om881020.pdf** This manual in Adobe Acrobat format (PDF)

Unzip the ***fmid*.zip** files on your workstation. You will now have files with names in the format ***fmid*.install.xmit** ie. **aseb209.install.xmit**, **asec209.install.xmit**, etc

Unzip the **recv*nnnn*.zip** files on your workstation. You will now have files with names in the format **recv*nnnn*.jcl** ie. **recvb209.jcl**, **recv209.jcl**, etc

File Transfer

The XMIT files you extracted must be transferred to z/OS datasets. They **MUST NOT** be translated during the file transfer ie. do a "binary" file transfer. Transfer the XMIT files into sequential datasets with the DCB attributes RECFM=FB and LRECL=80. {ASEB209 – approximately 145 tracks, ASEC209 – approximately 145 tracks, ASED209 – approximately 35 tracks, ASEE209 – approximately 5 tracks, ASEG209 approximately 70 tracks}. The remainder of this section assumes your .XMIT datasets will have dataset names of the format ***hlq.fmid*.INSTALL.XMIT** where ***hlq*** is a high level qualifier chosen by you.

The JCL files you extracted must be transferred to z/OS datasets. They **MUST** be translated during the file transfer ie. do a "text" or "ascii" file transfer. Transfer the JCL files into datasets with the DCB attributes RECFM=FB and LRECL=80.

Extract (RECEIVE) SMP/E Files using the supplied JCL

You can use the supplied JCL in the **recv*nnnn*.jcl** files to extract the SMP/E install files. Alternatively you can do it manually which is described in the next section.

The supplied JCL uses the TSO RECEIVE command to extract the SMP/E install files. It assumes that the high level qualifier in the ***hlq.fmid*.INSTALL.XMIT** datasets is your TSO prefix. Here's the JCL from the **recvb209.jcl** file.

```
//RECVB209 JOB 'RECEIVE ASEB209 RELFILES',MSGCLASS=X,REGION=5M
//*--------------------------------------------------------------------
//* This job receives the contents of the distribution XMIT file to
//* create the product relfile datasets for ASEB209.
//*
//* The dataset names in the following commands are unqualified and
//* will assume your TSO userid as the high level qualifier.
//*
//* Note that the blank lines after each RECEIVE command are necessary
//* for this job to function correctly.
//*--------------------------------------------------------------------
//RECEIVE  EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
```

```
  DELETE ASEB209.INSTALL
  RECEIVE INDA(ASEB209.INSTALL.XMIT)

  DELETE ASEB209.SMPMCS
  RECEIVE INDA(ASEB209.INSTALL(SMPMCS))

  DELETE ASEB209.F1
  RECEIVE INDA(ASEB209.INSTALL(F1))

  DELETE ASEB209.F2
  RECEIVE INDA(ASEB209.INSTALL(F2))

  DELETE ASEB209.F3
  RECEIVE INDA(ASEB209.INSTALL(F3))

  DELETE ASEB209.F4
  RECEIVE INDA(ASEB209.INSTALL(F4))

  DELETE ASEB209.F5
  RECEIVE INDA(ASEB209.INSTALL(F5))

  DELETE ASEB209.F6
  RECEIVE INDA(ASEB209.INSTALL(F6))

  DELETE ASEB209.F7
  RECEIVE INDA(ASEB209.INSTALL(F7))

  END
/*
```

Run the job. It will create these datasets for ASEB209 – where ***hlq*** is your TSO prefix.:

- **hlq.ASEB209.SMPMCS**
- **hlq.ASEB209.F1**
- **hlq.ASEB209.F2**
- **hlq.ASEB209.F3**
- **hlq.ASEB209.F4**
- **hlq.ASEB209.F5**
- **hlq.ASEB209.F6**
- **hlq.ASEB209.F7**

If you are also installing other fmids like ASEC209 etc., repeat the whole process with the corresponding **recv*nnn*.jcl** datasets. Note that while ASEB209 has 7 relative files, the other fmids will have a different number of relative files (usually less).

Alternative: Extract (RECEIVE) SMP/E Files manually

Use the TSO RECEIVE command in ISPF option 6 to extract the contents of your ***hlq.fmid.*INSTALL.XMIT** datasets:

```
RECEIVE INDA('hlq.fmid.INSTALL.XMIT')
```

You will see messages like this:

```
INMR901I Dataset ASE.ASEB209.INSTALL from PETER on NODENAME
INMR906A Enter restore parameters or 'DELETE' or 'END' +
```

Press the enter key to accept the defaults. You will then see a series of messages similar to this:

```
                    IEBCOPY MESSAGES AND CONTROL STATEMENTS                    PAGE    1
IEB1135I IEBCOPY  FMID HDZ11F0  SERVICE LEVEL UW81747  DATED 20010911 DFSMS 02.10.00 z/OS    01.02.00
HBB7705  CPU 1247
IEB1035I PETER    ASEPROC  ASEPROC  11:44:07 THU 11 OCT 2007 PARM="
 COPY INDD=((SYS00068,R)),OUTDD=SYS00067
IEB1013I COPYING FROM PDSU  INDD=SYS00068 VOL=OS39M1 DSN=SYS07284.T114404.RA000.PETER.R0100436
IEB1014I          TO PDS  OUTDD=SYS00067 VOL=OS39M1 DSN=PETER.ZSEN150.INSTALL
IEB167I FOLLOWING MEMBER(S) LOADED FROM INPUT DATA SET REFERENCED BY SYS00068
IEB154I F1       HAS BEEN SUCCESSFULLY LOADED
IEB154I F2       HAS BEEN SUCCESSFULLY LOADED
IEB154I F3       HAS BEEN SUCCESSFULLY LOADED
IEB154I F4       HAS BEEN SUCCESSFULLY LOADED
IEB154I F5       HAS BEEN SUCCESSFULLY LOADED
IEB154I F6       HAS BEEN SUCCESSFULLY LOADED
IEB154I F7       HAS BEEN SUCCESSFULLY LOADED
IEB154I SMPMCS   HAS BEEN SUCCESSFULLY LOADED
IEB1098I 8 OF 8 MEMBERS LOADED FROM INPUT DATA SET REFERENCED BY SYS00068
IEB144I THERE ARE 13 UNUSED TRACKS IN OUTPUT DATA SET REFERENCED BY SYS00067
IEB149I THERE ARE 5 UNUSED DIRECTORY BLOCKS IN OUTPUT DIRECTORY
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
INMR001I Restore successful to dataset 'PETER.ASEB209.INSTALL'
***
```

You must now run the TSO RECEIVE command against each of the members in the ***hlq.fmid.*INSTALL** dataset you just created.

```
RECEIVE INDA('hlq.fmid.ASEB209.INSTALL(SMPMCS)')

INMR901I Dataset ASE.ASEB209.SMPMCS from PETER on NODENAME
INMR906A Enter restore parameters or 'DELETE' or 'END' +

INMR001I Restore successful to dataset 'hlq..ASEB209.SMPMCS'
```

```
RECEIVE INDA('hlq.fmid.ASEB209.INSTALL(F1)')

INMR901I Dataset ASE.ASEB209.F1 from PETER on NODENAME
INMR906A Enter restore parameters or 'DELETE' or 'END' +

INMR001I Restore successful to dataset 'hlq..ASEB209.F1'
```

Repeat the RECEIVE command for all of the other members of your ***hlq.***ASEB209.INSTALL dataset ie F1 through F7.

If you are also installing other fmids like ASEC209 etc., repeat the whole process with the corresponding ***hlq.fmid***.INSTALL datasets. Note that while ASEB209 has 7 relative files, the other fmids will have a different number of relative files (usually less).

ALLOCATING THE OMCS LIBRARIES

When a service refresh of OMCS is to be installed, an existing set of OMCS libraries may be reused. If the libraries to be reused are not empty, the disk space required may temporarily be double the space estimates shown below.

The SMP/E install process will build eleven OMCS install libraries - six distribution libraries and five target libraries. The distribution libraries hold the master copy of OMCS maintained by SMP/E. The target libraries make up your running OMCS system.

The OMCS install libraries are listed below along with DASD space estimates. These estimates are large enough for the OMCS base product and the optional extension products OMCS/IRM, OMCS/JSF, OMCS/CICS and OMCS/SRF.

Note that lowest level of the dataset name must be as shown - this is required by SMP/E. However, the higher levels may be whatever the installation requires (eg. SYS3.CPUA.ASELIB or ASE.ASELIB, etc.).

library	RECFM	LRECL	BLKSIZE	number of tracks	number of directory blocks
ASE.AASEECPS	FB	80	27920	268	19
ASE.AASELOAD	U	-	32760	147	48
ASE.AASEMAC	FB	80	27920	6	8
ASE.AASEPANL	FB	80	27920	44	77
ASE.AASESAMP	FB	80	27920	13	15
ASE.AASETLIB	FB	80	27920	1	1
ASE.ASEECPS	FB	80	27920	268	20
ASE.ASELIB	U	-	32760	180	48
ASE.ASEPANL	FB	80	27920	44	77
ASE.ASESAMP	FB	80	27920	13	15
ASE.ASETLIB	FB	80	27920	1	1

Sample JCL for allocating OMCS libraries.

```
 //ALOCDSN  EXEC PGM=IEFBR14
//*
//AASEECPS DD DSN=ASE.AASEECPS,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <-system-determined blksize
//            SPACE=(TRK,(300,30,30)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//AASELOAD DD DSN=ASE.AASELOAD,DISP=(,CATLG),
//            DCB=(RECFM=U,BLKSIZE=32760),
//            SPACE=(TRK,(165,15,60)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//AASEMAC  DD DSN=ASE.AASEMAC,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <- system-determined blksize
//            SPACE=(TRK,(10,5,15),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//AASEPANL DD DSN=ASE.AASEPANL,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <- system-determined blksize
//            SPACE=(TRK,(60,15,100)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
```

```
//AASESAMP DD DSN=ASE.AASESAMP,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <- system-determined blksize
//            SPACE=(TRK,(15,5,20)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//AASETLIB  DD DSN=ASE.AASETLIB,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <- system-determined blksize
//            SPACE=(TRK,(1,1,5)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//ASEECPS  DD DSN=ASE.ASEECPS,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <- system-determined blksize
//            SPACE=(TRK,(300,30,30)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//ASELIB   DD DSN=ASE.ASELIB,DISP=(,CATLG),
//            DCB=(RECFM=U,BLKSIZE=32760),
//            SPACE=(TRK,(230,30,60)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//ASEPANL  DD DSN=ASE.ASEPANL,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <- system-determined blksize
//            SPACE=(TRK,(60,15,100)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//ASESAMP  DD DSN=ASE.ASESAMP,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <- system-determined blksize
//            SPACE=(TRK,(15,5,20)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
//ASETLIB  DD DSN=ASE.ASETLIB,DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0), <- system-determined blksize
//            SPACE=(TRK,(1,1,5)),
//            UNIT=SYSDA,VOL=SER=______ <- insert the desired DASD volume
```

ALLOCATING THE SMP/E DATASETS

This manual will not describe how to allocate SMP/E datasets. The following manuals should be consulted for this information:

SMP/E User's Guide, SA22-7773
SMP/E Commands, SA22-7771

OMCS must **not** be installed in a zone which contains JES2. The customer must decide whether to install OMCS using an existing CSI dataset or create a new CSI dataset. A CSI dataset of approximately three cylinders of 3390 DASD space will be sufficient to install OMCS.

PREPARING THE SMP/E ENVIRONMENT FOR OMCS INSTALLATION

Dynamic Allocation

This step can be skipped when a service refresh is to be installed over an existing OMCS installation.

If you decide to have SMP/E dynamically allocate the OMCS target and distribution libraries, you must define the required DDDEF entries. Here is an example of UCLIN commands to do this in the target zone:

```
SET   BDY(OMCSTGT).         /* set to target zone     */
UCLIN.                      /*                        */
ADD   DDDEF(AASEECPS)       /* DLIB ECP library       */
      DATASET(ASE.AASEECPS) /*                        */
      SHR.                  /*                        */
```

```
 ADD   DDDEF(AASELOAD)       /* DLIB object library    */
       DATASET(ASE.AASELOAD) /*                        */
       SHR.                  /*                        */
 ADD   DDDEF(AASEMAC)        /* DLIB macro library     */
       DATASET(ASE.AASEMAC)  /*                        */
       SHR.                  /*                        */
 ADD   DDDEF(AASEPANL)       /* DLIB panel library     */
       DATASET(ASE.AASEPANL) /*                        */
       SHR.                  /*                        */
 ADD   DDDEF(AASESAMP)       /* DLIB sample library    */
       DATASET(ASE.AASESAMP) /*                        */
       SHR.                  /*                        */
 ADD   DDDEF(AASETLIB)       /* DLIB table library     */
       DATASET(ASE.AASETLIB) /*                        */
       SHR.                  /*                        */
 ADD   DDDEF(ASEECPS)        /* target ECP library     */
       DATASET(ASE.ASEECPS)  /*                        */
       SHR.                  /*                        */
 ADD   DDDEF(ASELIB)         /* target load library    */
       DATASET(ASE.ASELIB)   /*                        */
       SHR.                  /*                        */
 ADD   DDDEF(ASEPANL)        /* target panel library   */
       DATASET(ASE.ASEPANL)  /*                        */
       SHR.                  /*                        */
 ADD   DDDEF(ASESAMP)        /* target sample library  */
       DATASET(ASE.ASESAMP)  /*                        */
       SHR.                  /*                        */
ADD    DDDEF(ASETLIB)        /* target table library   */
       DATASET(ASE.ASETLIB)  /*                        */
       SHR.                  /*                        */
 ENDUCL.                     /*                        */
```

Here is an example of UCLIN commands to define the required DDDEF entries in the distribution zone:

```
SET   BDY(OMCSDLB).          /* set to DLIB zone       */
UCLIN.                       /*                        */
ADD   DDDEF(AASEECPS)        /* DLIB ECP library       */
      DATASET(ASE.AASEECPS) /*                         */
      SHR.                   /*                        */
ADD   DDDEF(AASELOAD)        /* DLIB object library    */
      DATASET(ASE.AASELOAD) /*                         */
      SHR.                   /*                        */
ADD   DDDEF(AASEMAC)         /* DLIB macro library     */
      DATASET(ASE.AASEMAC)  /*                         */
      SHR.                   /*                        */
ADD   DDDEF(AASEPANL)        /* DLIB panel library     */
      DATASET(ASE.AASEPANL) /*                         */
      SHR.                   /*                        */
ADD   DDDEF(AASESAMP)        /* DLIB sample library    */
      DATASET(ASE.AASESAMP) /*                         */
      SHR.                   /*                        */
ADD   DDDEF(AASETLIB)        /* DLIB table library     */
      DATASET(ASE.AASETLIB) /*                         */
      SHR.                   /*                        */
ADD   DDDEF(ASEECPS)         /* target ECP library     */
      DATASET(ASE.ASEECPS)  /*                         */
      SHR.                   /*                        */
ADD   DDDEF(ASELIB)          /* target load library    */
      DATASET(ASE.ASELIB)   /*                         */
      SHR.                   /*                        */
ADD   DDDEF(ASEPANL)         /* target panel library   */
      DATASET(ASE.ASEPANL)  /*                         */
      SHR.                   /*                        */
ADD   DDDEF(ASESAMP)         /* target sample library  */
      DATASET(ASE.ASESAMP)  /*                         */
      SHR.                   /*                        */
ADD   DDDEF(ASETLIB)         /* target table library   */
      DATASET(ASE.ASETLIB)  /*                         */
      SHR.                   /*                        */
ENDUCL.                      /*                        */
```

SMP/E INSTALLATION OF OMCS

OMCS is installed using the "standard" RECEIVE-APPLY-ACCEPT installation method.

RECEIVE

Use the SMP/E ISPF dialog or a batch job to receive the OMCS function(s). Here is an example of a RECEIVE command to do this. The RFPREFIX value is the high level qualifier for the ***hlq.fmid.*F*n*** datasets you created earlier

```
SET BDY(GLOBAL).        /* Set to global zone         */
RECEIVE SELECT(ASEB209, /* OMCS base product          */
               ASEC209, /* OMCS/IRM                   */
               ASED209, /* OMCS/JSF                   */
               ASEE209, /* OMCS/CICS                  */
               ASEG209) /* OMCS/SRF                   */
        RFPREFIX(hlq)   /* hlq for fmid.Fn datasets   */
        SYSMODS.        /*                            */
```

If you are **not** installing OMCS/IRM then **exclude** ASEC209 from the RECEIVE command.
If you are **not** installing OMCS/JSF then **exclude** ASED209 from the RECEIVE command.
If you are **not** installing OMCS/CICS then **exclude** ASEE209 from the RECEIVE command.
If you are **not** installing OMCS/SRF then **exclude** ASEG209 from the RECEIVE command.

When receiving the OMCS functions from DASD datasets the SMPMCS datasets must be allocated as in the following example:

```
//SMPPTFIN DD DISP=SHR,DSN=hlq.ASEB209.SMPMCS
//         DD DISP=SHR,DSN=hlq.ASEC209.SMPMCS
//         DD DISP=SHR,DSN=hlq.ASED209.SMPMCS
//         DD DISP=SHR,DSN=hlq.ASEE209.SMPMCS
//         DD DISP=SHR,DSN=hlq.ASEG209.SMPMCS
```

The RECEIVE process will copy the “relative” files into SMPTLIB datasets. Here are estimates of SMPTLIB space requirements:

smptlib.ASEB209.F1	3 tracks, 1 directory block
smptlib.ASEB209.F2	87 tracks, 7 directory blocks
smptlib.ASEB209.F3	65 tracks, 25 directory blocks
smptlib.ASEB209.F4	7 tracks, 8 directory blocks
smptlib.ASEB209.F5	20 tracks, 40 directory blocks
smptlib.ASEB209.F6	11 tracks, 11 directory blocks
smptlib.ASEB209.F7	2 tracks, 1 directory block
smptlib.ASEC209.F1	2 tracks, 1 directory block
smptlib.ASEC209.F2	110 tracks, 9 directory blocks
smptlib.ASEC209.F3	48 tracks, 14 directory blocks
smptlib.ASEC209.F4	16 tracks, 28 directory blocks
smptlib.ASEC209.F5	5 tracks, 5 directory blocks
smptlib.ASED209.F1	2 tracks, 1 directory block
smptlib.ASED209.F2	26 tracks, 3 directory blocks
smptlib.ASED209.F3	11 tracks, 4 directory blocks
smptlib.ASED209.F4	6 tracks, 7 directory blocks
smptlib.ASEE209.F1	2 tracks, 1 directory block
smptlib.ASEE209.F2	2 tracks, 1 directory block
smptlib.ASEE209.F3	2 tracks, 1 directory block
smptlib.ASEG209.F1	2 tracks, 1 directory block

smptlib.ASEG209.F2	52 tracks, 2 directory blocks
smptlib.ASEG209.F3	27 tracks, 6 directory blocks
smptlib.ASEG209.F4	6 tracks, 4 directory blocks
smptlib.ASEG209.F5	2 tracks, 1 directory block

APPLY

Use the SMP/E ISPF dialog or a batch job to update your target libraries. Here are examples of APPLY CHECK and APPLY commands to do this:

```
SET BDY(OMCSTGT).       /* Set to target zone         */
APPLY SELECT(ASEB209,   /* OMCS base product          */
             ASEC209,   /* OMCS/IRM                   */
             ASED209,   /* OMCS/JSF                   */
             ASEE209,   /* OMCS/CICS                  */
             ASEG209)   /* OMCS/SRF                   */
      REDO              /* replace existing level     */
      CHECK             /* don't update libraries     */
      BYPASS(ID).       /* bypass ID check            */
SET BDY(OMCSTGT).       /* Set to target zone         */
APPLY SELECT(ASEB209,   /* OMCS base product          */
             ASEC209,   /* OMCS/IRM                   */
             ASED209,   /* OMCS/JSF                   */
             ASEE209,   /* OMCS/CICS                  */
             ASEG209)   /* OMCS/SRF                   */
      REDO              /* replace existing level     */
      RETRY(YES).       /* recover from x37 abends    */
```

If you are **not** installing OMCS/IRM then **exclude** ASEC209 from the APPLY command.
If you are **not** installing OMCS/JSF then **exclude** ASED209 from the APPLY command.
If you are **not** installing OMCS/CICS then **exclude** ASEE209 from the APPLY command.
If you are **not** installing OMCS/SRF then **exclude** ASEG209 from the APPLY command.

Your APPLY CHECK and APPLY jobs may finish with RC=4 and give any of the following warning messages:

GIM61903W LMOD lmod WAS NOT DELETED BY SYSMOD sysmod BECAUSE lmod IS NOT IN THE zone ZONE.

GIM32501W SYSMOD sysmod SUPERSEDES SYSMOD sysmod BUT DOES NOT CONTAIN ELEMENT name.

GIM23903W LINK-EDIT PROCESSING FOR SYSMOD sysmod WAS SUCCESSFUL FOR MODULE mod IN LMOD lmod IN THE ASELIB LIBRARY. THE RETURN CODE WAS 04. DATE yy.ddd - TIME hh:mm:ss - SEQUENCE NUMBER nnnnnn.

The above SMP/E message may be associated with one or more of the following link-edit messages:

IEW0461 name

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE; NCAL WAS SPECIFIED, OR THE REFERENCE WAS MARKED FOR RESTRICTED NO-CALL OR NEVERCALL.

IEW2454W 9203 SYMBOL name UNRESOLVED. NO AUTOCALL (NCAL) SPECIFIED.

IEW2651W 511C ESD AMODE 24 CONFLICTS WITH USER_SPECIFIED AMODE 31 FOR ENTRY POINT name.

These messages are normal and can be ignored.

ACCEPT

Use the SMP/E ISPF dialog or a batch job to update your distribution libraries. Here are examples of ACCEPT CHECK and ACCEPT commands to do this:

```
SET BDY(OMCSDLB).       /* Set to DLIB zone           */
ACCEPT SELECT(ASEB209,  /* OMCS base product          */
              ASEC209,  /* OMCS/IRM                   */
              ASED209,  /* OMCS/JSF                   */
              ASEE209,  /* OMCS/CICS                  */
              ASEG209)  /* OMCS/SRF                   */
      REDO              /* replace existing level     */
      CHECK             /* don't update libraries     */
      BYPASS(ID).       /* bypass ID check            */

SET BDY(OMCSDLB).       /* Set to DLIB zone           */
ACCEPT SELECT(ASEB209,  /* OMCS base product          */
              ASEC209,  /* OMCS/IRM                   */
              ASED209,  /* OMCS/JSF                   */
              ASEE209,  /* OMCS/CICS                  */
              ASEG209)  /* OMCS/SRF                   */
      REDO              /* replace existing level     */
      RETRY(YES).       /* recover from X37 abends    */
```

If you are **not** installing OMCS/IRM then **exclude** ASEC209 from the ACCEPT command.
If you are **not** installing OMCS/JSF then **exclude** ASED209 from the ACCEPT command.
If you are **not** installing OMCS/CICS then **exclude** ASEE209 from the ACCEPT command.
If you are **not** installing OMCS/SRF then **exclude** ASEG209 from the ACCEPT command.

Your ACCEPT CHECK and ACCEPT jobs may finish with RC=4 and give any of the the following warning messages:

GIM61903W LMOD lmod WAS NOT DELETED BY SYSMOD sysmod BECAUSE lmod IS NOT IN THE zone ZONE.

GIM32501W SYSMOD sysmod SUPERSEDES SYSMOD sysmod BUT DOES NOT CONTAIN ELEMENT name.

These messages are normal and can be ignored.

POST SMP/E TASKS

The following task is only required if you have installed the OMCS/CICS interface - fmid ASEE209.

Link Edit OMCSCICS

If OMCS/CICS was installed then run a link-edit job like the following example.

FMID ASEE200 must be applied and accepted before the link-edit job is run. An example of this job is supplied in the ASESAMP library as member LINKCICS.

```
//*-------------------------------------------------------------------
//*
//* A SAMPLE JOB FOR RE-LINKING OMCSCICS
//* AFTER INSTALLATION USING SMP/E
//*
//*-------------------------------------------------------------------
//LKED     EXEC PGM=IEWL,
// PARM='LIST,REUS,XREF,NCAL'
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//AASELOAD DD DISP=SHR,DSN=your.AASELOAD     <- your dataset name
//CICSLIB  DD DISP=SHR,DSN=your.CICS.LOADLIB <- your dataset name
//SYSLMOD  DD DISP=SHR,DSN=your.ASELIB       <- your dataset name
//SYSLIN DD *
 INCLUDE CICSLIB(DFHEAI,DFHEAI0)
 INCLUDE AASELOAD(OMCSCICS)
 ORDER DFHEAI,OMCSCICS,DFHEAI0
 ENTRY OMCSCICS
 NAME OMCSCICS(R)
//*
```

After the link-edit is run, SMP/E must be informed of the changes to the OMCSCICS load module. Run an SMP/E JCLIN job using the JCL above as input. This will ensure that program OMCSCICS is link-edited correctly by SMP/E whenever maintenance is applied. Here is a sample SMP/E JCLIN job.

```
//SMPESTEP EXEC PGM=GIMSMP,REGION=5M,DYNAMNBR=120,
//            PARM='PROCESS=WAIT'
//SMPCSI   DD DISP=SHR,DSN=your.CSI
//SMPJCLIN DD DISP=SHR,DSN=your.ASESAMP(LINKCICS)
//SMPCNTL  DD *
  SET BOUNDARY(OMCSTGT).
  JCLIN.
/*
```

MAINTENANCE

There are two types of maintenance you will receive from ASE:

- **PTFs**
  You may receive PTFs with your OMCS product materials and from time to time thereafter. In IBM's terminology this is "preventive service". These PTFs will fix known problems and install enhancements.

- **APAR fixes**
  You will receive APAR fixes from ASE to correct problems you have reported in your OMCS system. In IBM's terminology this is "corrective service". These fixes are to temporarily correct a problem until a PTF can be developed and shipped to customers.

SMP/E Utility and Options entries

APAR fixes which contain source updates for ECL programs require two special utility programs.

- ASMPUPDT which updates ECL source
- ASMPECLC which interfaces to the ECL compiler

These programs must be defined to SMP/E in OPTIONS and UTILITY entries. Here is an example of UCLIN commands to do this:

```
SET   BDY(GLOBAL).          /* set to global zone     */
UCLIN.                      /*                        */
ADD   OPTIONS(ASEOPT)       /* for ASE products       */
      ASM(ASMPECLC)         /* ECL compiler           */
      UPDATE(ASMPUPDT).     /* update utility         */

ADD   UTILITY(ASMPUPDT)     /* update utility         */
      NAME(ASMPUPDT)        /*                        */
      RC(4).                /* RC=4 is OK             */

ADD   UTILITY(ASMPECLC)     /* ECL compiler           */
      NAME(ASMPECLC)        /*                        */
      RC(4).                /* RC=4 is OK             */

ENDUCL.                     /*                        */
```

NOTE: This special **options entry** cannot be used during the initial installation of OMCS. At that stage the programs do not exist and SMP/E will fail during initialisation. The options entry is only required when installing APAR fixes which contain source updates.

PTFs

PTFs may be sent to you by ASE in order to correct known problems and install enhancements. The PTFs should be installed in the same manner as the original product ie. using the RECEIVE-APPLY-ACCEPT process.

Electronic distribution of maintenance

It is now common for ASE to ship both APAR fixes and PTFs directly to customers via email.

Receive PTFs

When a group of PTFs are received they should be assigned to a SOURCEID. This allows the group of PTFs to be easily referenced through this identifier. The identifier should be unique and meaningful eg. PUT9710.

Use the SMP/E ISPF dialog or a batch job to receive the PTFs. Here is an example of a RECEIVE command to do this:

```
SET     BDY(GLOBAL).    /* Set to global zone         */
RECEIVE SYSMODS         /* Receive all SYSMODs        */
        SOURCEID(       /* Assign all PTFs a common   */
         xxxxxxxx).     /*  identifier                */

LIST    SYSMODS         /* List the SYSMOD            */
        MCS             /*  and MCS (cover letter)    */
        SOURCEID(       /*  for only those SYSMODS.   */
         xxxxxxxx).     /*                            */
```

If you are using a batch job to receive the PTFs then allocate the dataset containing the maintenance to DD name SMPPTFIN:

```
//SMPPTFIN DD DSN=SMPMCS,
//            UNIT=TAPE,
//            VOL=SER=______,
//            LABEL=(1,SL),
//            DISP=SHR
```

Apply PTFs

Use the SMP/E ISPF dialog or a batch job to update your target libraries. Here are examples of APPLY CHECK and APPLY commands to do this:

```
SET   BDY(OMCSTGT).     /* Set to target zone         */
APPLY SOURCEID(xxxxxxx) /* Apply this PTF tape        */
      GROUP             /*  and all requisite PTFs    */
      CHECK             /* don't update libraries     */
      BYPASS(ID).       /* bypass ID check            */

SET   BDY(OMCSTGT).     /* Set to target zone         */
APPLY SOURCEID(xxxxxxx) /* Apply this PTF tape        */
      GROUP             /*  and all requisite PTFs    */
      RETRY(YES).       /* recover from X37 abends    */
```

After the APPLY CHECK has run, check the output listing for errors. Consult the SMP/E User's Guide for guidelines on handling any errors that occur. If you cannot solve a problem contact ASE for support.

Your APPLY CHECK and APPLY jobs may finish with RC=4 and give the following warning message:

GIM23903W LINK-EDIT PROCESSING FOR SYSMOD sysmod WAS SUCCESSFUL FOR MODULE mod IN LMOD lmod IN THE ASELIB LIBRARY. THE RETURN CODE WAS 04. DATE yy.ddd - TIME hh:mm:ss - SEQUENCE NUMBER nnnnnn.

The above SMP/E message will be associated with one or more of the following link-edit messages:

IEW0461 name

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE; NCAL WAS SPECIFIED, OR THE REFERENCE WAS MARKED FOR RESTRICTED NO-CALL OR NEVERCALL.

IEW2454W 9203 SYMBOL name UNRESOLVED. NO AUTOCALL (NCAL) SPECIFIED.

These messages are normal and can be ignored.

Accept PTFs

After the PTFs have been applied and before accepting the PTFs you should test your newly updated OMCS system. As a general rule you should not accept PTFs until your updated OMCS system is running successfully in production.

Use the SMP/E ISPF dialog or a batch job to update your distribution libraries. Here are examples of ACCEPT CHECK and ACCEPT commands to do this:

```
SET    BDY(OMCSDLB).    /* Set to DLIB zone           */
ACCEPT SOURCEID(xxxxxx) /* Accept these PTFs          */
       GROUP            /*  include requisite PTFs    */
       CHECK            /* don't update libraries     */
       BYPASS(ID).      /* bypass ID check            */

SET    BDY(OMCSDLB).    /* Set to DLIB zone           */
ACCEPT SOURCEID(xxxxxx) /* Accept these PTFs          */
       GROUP            /*  include requisite PTFs    */
       RETRY(YES).      /* recover from X37 abends    */
```

After the ACCEPT CHECK has run, check the output listing for errors. Consult the SMP/E User's Guide for guidelines on handling any errors that occur. If you cannot solve a problem contact ASE for support.

Your ACCEPT CHECK and ACCEPT jobs may finish with RC=4 and give the following warning messages:

GIM23903W LINK-EDIT PROCESSING FOR SYSMOD sysmod WAS SUCCESSFUL FOR MODULE mod IN LMOD lmod IN THE AASELOAD LIBRARY. THE RETURN CODE WAS 04. DATE yy.ddd - TIME hh:mm:ss - SEQUENCE NUMBER nnnnnn.

GIM24701W SMP/E COULD NOT OBTAIN LINK-EDIT PARAMETERS FOR LOAD MODULE mod IN THE AASELOAD LIBRARY FOR SYSMOD sysmod. DEFAULTS WERE USED.

GIM23903W will be associated with one or more of the following link-edit messages:

IEW0461 name

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE; NCAL WAS SPECIFIED, OR THE REFERENCE WAS MARKED FOR RESTRICTED NO-CALL OR NEVERCALL.

IEW2454W 9203 SYMBOL name UNRESOLVED. NO AUTOCALL (NCAL) SPECIFIED.

These messages are normal and can be ignored.

APAR fixes

APAR fixes are sent to you by ASE in order to correct a problem you have experienced with OMCS. These fixes will be in the form of ECL source updates or module updates in ZAP format. APAR fixes should be installed using a RECEIVE-APPLY process - they are not normally accepted. Consult the SMP/E User's Guide for the pros and cons of accepting corrective service.

APAR fixes are typically distributed via email. They can also be sent via diskette, fax, or as hardcopy by post.

Receive APAR fixes

APAR fixes are usually received individually. Use the SMP/E ISPF dialog or a batch job to receive the APAR fix. Here is an example of a RECEIVE command to do this:

```
SET     BDY(GLOBAL).    /* Set to global zone         */
RECEIVE SELECT(         /* Receive selected APAR fix  */
              xxxxxxxx) /* Specify APAR fix number    */
        SYSMODS.        /*                            */
```

If you are using a batch job to receive the APAR fix then you must include an SMPPTFIN DD statement. Point it to the dataset in which you have placed the APAR fix.

Apply APAR fixes

Use the SMP/E ISPF dialog or a batch job to update your target libraries. Here are examples of APPLY CHECK and APPLY commands to do this:

```
 SET   BDY(OMCSTGT)      /* Set to target zone         */
       OPTIONS(ASEOPT).  /* ASE options entry          */
 APPLY SELECT(           /* Apply selected APAR fix    */
              xxxxxxxx)  /* Specify APAR fix number    */
       CHECK             /* don't update libraries     */
       BYPASS(ID).       /* bypass ID check            */

 SET   BDY(OMCSTGT).     /* Set to target zone         */
       OPTIONS(ASEOPT).  /* ASE options entry          */
 APPLY SELECT(           /* Apply selected APAR fix    */
              xxxxxxxx)  /* Specify APAR fix number    */
       RETRY(YES).       /* recover from X37 abends    */
```

After the APPLY CHECK has run, check the output listing for errors. Consult the SMP/E User's Guide for guidelines on handling any errors that occur. If you cannot solve a problem contact ASE for support.

After the APPLY is complete test your OMCS system before migrating the change to production.

Your APPLY CHECK and APPLY jobs may finish with RC=4 and give the following warning message:

GIM23903W LINK-EDIT PROCESSING FOR SYSMOD sysmod WAS SUCCESSFUL FOR MODULE mod IN LMOD lmod IN THE ASELIB LIBRARY. THE RETURN CODE WAS 04. DATE yy.ddd - TIME hh:mm:ss - SEQUENCE NUMBER nnnnnn.

GIM38201W THERE IS A MODID ERROR FOR ZAP ENTRY mod IN SYSMOD sysmod2.

GIM31904I SYSMOD sysmod2 DOES NOT SPECIFY sysmod1 ON THE PRE OR SUP OPERAND. sysmod1 IS A UMID FOR AN ELEMENT THAT IS BEING INSTALLED.

GIM23903W will be associated with the following link-edit messages:

IEW0461 name

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE; NCAL WAS SPECIFIED, OR THE REFERENCE WAS MARKED FOR RESTRICTED NO-CALL OR NEVERCALL.

IEW2454W 9203 SYMBOL name UNRESOLVED. NO AUTOCALL (NCAL) SPECIFIED.

These messages are normal and can be ignored.

SMP/E will not install multiple superzaps to one module in a single execution of APPLY. If you attempt to do this only one will be applied and the APPLY command will terminate with RC=8 and issue the following message.

GIM41001E ** APPLY PROCESSING FAILED FOR sysmod2 BECAUSE BOTH sysmod2 AND sysmod1 CONTAIN ZAP UPDATES FOR MODULE mod. ONLY ONE OF THESE SYSMODS CAN BE PROCESSED AT A TIME.

GIM22601I APPLY PROCESSING FAILED FOR SYSMOD sysmod2.

PART 2 - THE INSTALLATION GUIDE

The remaining installation tasks consist of the following activities :

- **Installing optional zaps**
  These are optional OMCS modifications that can install using SMP/E usermods.

- **Copying load modules to an authorised linklisted library.**
  The load modules must be placed in an authorised library in the linklist to allow all components to function correctly.

- **Creating the OMCS started task JCL procedure.**
  OMCS uses a started task to perform sysout capture and a variety of other functions. A sample catalogued JCL procedure is supplied in the .ASESAMP library.

- **ISPF Installation.**
  OMCS components that operate in TSO require access to a library of display panels via the ISPPLIB DD name in each TSO address space. You must either modify the allocation of ISPPLIB to include the .ASEPANL library or copy the contents of that library to a library that will be part of that concatenation.

- **Installing the System Command Issuing Module and SVC.**
  Various features of OMCS will only work if the user SVC is installed. The command issuing module and SVC are supplied in source form to allow installation inspection and modification for security purposes and both must be installed into appropriate libraries.

- **Installing Access Control Facility User Exits**
  A supplied sample exit module is to be tailored, assembled and linkedited to allow the access control facility to operate correctly in non-TSO environments.

- **Installing IRM report viewing for CICS**
  OMCS/IRM report viewing via CICS requires a server address space and some CICS resources to be defined.

- **Support**
  Our support for this product began when you first requested that the install material be supplied to you. If at any subsequent time you find that the material in this or any other OMCS manual is confusing, inadequate or incorrect, or if you encounter any problem with the supplied machine-readable material or get unexpected results from the installed product, please contact ASE for support. See the inside cover of this manual for telephone and fax numbers and the email address.

WHAT'S WHAT IN THE ASESAMP LIBRARY

The .ASESAMP library contains miscellaneous files that may be required during the install process. These include sample JCL decks, source for security exits, notes for special situations and so forth. You will be referred to specific members at different points during the install process. This list may not describe every member of the .ASESAMP library.

Member list and brief description

$FUJITSU	Notes on the installation of OMCS in Fujitsu OS4F4 systems
$FUJIZAP	Superzap source for modifications required for Fujitsu OS4F4 systems.
ACFALLOW	Sample ACL for allowing all file accesses
ACFOFF	Sample ACL for turning off OMCS/ACF
ACFSAMPL	Sample ACL illustrating different options
ACF2UIDX	Assembler source for sample security exit for ACF2 users
AMFARJCL	Sample JCL used by OMCS to execute AMF recovery.
AMFRSJCL	Sample JCL used by OMCS to schedule AMF recovery jobs.
CTLDBSR	Sample space reclaim control statements
CTLJBCK	Sample chklib member used by OMCSJBCK.
CTLXWAF	Sample writer control statements
GENACFX	Sample JCL used to assemble and link the ACF security exit module
GENISCM	Sample JCL used to assemble and link the OMCSISCM module
IRMANJCL	JCL model used for IRM report discovery jobs ("A" line command)
IRMBCJCL	JCL model used for IRM bundle closure jobs
IRMBPJCL	JCL model used for IRM bundle print jobs
IRMBRJCL	JCL model used for IRM bundle recovery jobs
IRMGNJCL	JCL model used for miscellaneous IRM batch jobs
IRMRDJCL	JCL model used for IRM report discovery jobs (auto scheduled)
LINKCICS	Sample JCL for link-editing OMCSCICS
LINKIRUV	Sample JCL for link-editing OMCSIRUV (Report Viewing Facility)
LINKTSSJ	Sample JCL for link-editing OMCSTSSJ
OMCSDBSE	Sample ECL source for database selection
OMCSDSSX	Sample space reclaim process manager ECL program
OMCSISCM	Assembler source for the OMCSISCM module
OMCSISFP	Sample SDSF parameters to allow OMCS writers to access the SDSF displays.
OMCSIVP1-4	ECL source for installation verification process
OMCSJBCK	ECL source for step-by-step checking of any job
OMCSJPRX	Sample job processing rule exit.
OMCSRVF	Sample JCL for creating a server started task for CICS
OMCSSTC	A job for the OMCS writer catalogued procedure
OMCSSVC	Assembler source for the OMCS type 3/4 user SVC
PDFORMS	Sample ECL for user printer driver.
PSPBIN	Sample print service program to cause printing to switch between paper bins
PSPDIAG	Sample ECL to see what is passed to the PSP interface during bundle printing.
PSPDUPLX	Sample print service program for duplex printing
PSPFICHE	Sample print service program to fiche the user manifests and reports
PSPPDSM	Sample print service program to put reports into partitioned dataset members
PSPTWOUP	Sample print service program to print reports twoup.
RACFUIDX	Assembler source for sample security exit for RACF users
SRFANJCL	JCL model used for OMCS/SRF analysis jobs
SRFARJCL	JCL model used for OMCS/SRF archive jobs
SRFSRJCL	JCL model used for OMCS/SRF staging pageset build jobs
RUNACF	A job to run the ACF rule compiler in batch
RUNAMAR	A job to run the OMCS archive manager utility OMCSAMAR
RUNAMCO	A job to run the tape consolidation utility OMCSAMCO
RUNAMFC	A job to delete old AMF request records after a specified count of days
RUNAMFS	A job to run to process the AMF request queue.

RUNAMTS	A job to run to prime PSF for AMF request queue processing.
RUNAMUT	A job to run the AMF index cleanup utility OMCSAMUT
RUNASMP	A job to run the ASMPMAIN program
RUNCTLC	A job to run the omcs copy utility OMCSCTLC
RUNDBBU	A job to run the fast backup program OMCSDBBU
RUNDBBUA	A job to dynamically create output tape datasets used by OMCSDBBU
RUNDBEX	A job to run the backup program OMCSDBEX.
RUNDBFM	A job to run the database formatting program OMCSDBFM
RUNDBIM	A job to run the database restore program OMCSDBIM
RUNDBRE	A job to run the database restore program OMCSDBRE
RUNDBSR	A job to run the database space reclaim program OMCSDBSR
RUNDBUT	A job to run the database utility program OMCSDBUT
RUNDBVR	A job to run the database verification and repair program OMCSDBVR
RUNDELNK	A job to delink a load module and create an object deck
RUNECLC	A job to compile an ECL program
RUNECPX	A job to run ECL programs in batch
RUNIRAU	A job to unload/reload accounting information for omcs/irm users and nodes
RUNIRDX	A job to create an irm user/report cross reference report
RUNIVP1-9	Jobs to perform an Installation Verification Process
RUNJDET	A job to run the job detail index maintenance untilty program OMCSJDET.
RUNMFAR	A job to run the microfiche program
RUNRVCU	A job to run the IRM RVF cleanup utility
RUNRVUS	A job to report on report viewing usage statistics
SAMPIRSM	An example of an OMCS/IRM user accounting EXIT (OMCSIRSM)
SDISAMP1	IRM report analysis segment identifier invoked by OMCSXWIR
SDISAMP2	An IRM sysout dataset identification procedure.
STAEXMOD	Sample JCL submitted by OMCSTAEX to modify parms used by ASESTAEX.
ZAPACF	A zap to turn off OMCS/ACF rule checking
ZAPAPPLY	A job to apply zaps with
ZAPOPT	Optional zaps for modifying OMCS
ZAPSMFW	A zap to change the SVC # used by module OMCSSMFW

OPTIONAL ZAPS

Review the members ZAPOPT and ZAPSMFW in the .ASESAMP library. They contain optional modifications to OMCS. With minor changes, the contents of ZAPOPT and ZAPSMFW can be used as the basis for SMP/E usermods.

INSTALLING THE LOAD MODULES

The load library created by the installation process must be **APF authorised.** This library should also be in the linklist.

CREATING THE OMCS STARTED TASK JCL PROCEDURE

Using the supplied sample as a starting point

A sample OMCS writer catalogued JCL procedure is provided as member OMCSSTC in the .ASESAMP install library.

This JCL will be tailored here initially and may be altered from time to time to suit operational requirements

The items to be tailored are :-

```
//IEFPROC  EXEC PGM=OMCSXWMT,PARM='WTR,PSF',REGION=6144K
```

The **WTR** keyword parameter specifies that the sysout capture function of OMCS is required. Once started, the sysout capture function is controlled by a parameter file pointed to be the XWAFIN DD statement (see below).

The **PSF** keyword parameter specifies that the process scheduling function of OMCS is required. If you are not going to use either the Job Scheduling Facility (JSF) or the Interactive Report Management (IRM) extension products then you should omit this keyword.

The **REGION=** JCL parameter specifies a virtual storage limit of 6 megabytes for this started task.

```
//DBASECPX DD DISP=SHR,DSN=....
```

The database to be accessed by the OMCS writer component must have a DD statement present in the JCL. This DD statement must have DD name DBASECPX at this time.

```
//ECPLIB   DD DISP=SHR,DSN=...
```

This DD statement specifies the library or libraries to be searched by the writer whenever an ECL program is required. You must include the .ASEECPS install library in this concatenation so that OMCS components can get at those ECL programs that are required for correct operation.

```
//XWAFIN   DD  DISP=SHR,DSN=...
```

This DD statement points to a file of control statements that will be processed when OMCS is started. A sample is provided in the .ASESAMP library as member CTLXWAF. This file is used to specify things like the sysout class code(s) that OMCS is to process. Refer to the supplied sample for further information.

```
//OMCSSRIN DD  DISP=SHR,DSN=....
```

This DD statement points to a file of control statements that will be read by the Space Reclaim utility program, OMCSDBSR, when it is invoked by the OMCSDSSX ECL program in the OMCS writer environment to deal with a full database situation. This DD statement should be pointed at a member of a library which can be updated later. A sample is provided in the .ASESAMP library as member CTLDBSR. That member contains the following statement:

```
*   #*    WEIGHT(4)
```

This tells space reclaim that it can consider only SYSOUT storage files as candidates for deletion. All other database files will be left strictly alone.

Proper establishment of space reclamation also involves tailoring an ECL program, OMCSDSSX, to submit a batch job.

Please refer to Part 3 of this manual under CONTROLLING AUTOMATIC SPACE RECLAMATION for a description of each of the elements of space reclaim including the tailoring of OMCSDSSX.

Using more than one OMCS writer

When you have a number of OMCS databases you will typically set up one OMCS writer started task for each database. Driving multiple databases from a single writer is supported in version 2. Refer to the member OMCSDBSE in the .ASEECPS install library.

You can simply copy the OMCSV2 JCL and save it as a different member then use ISPF edit to change details such as the name of the database.

ISPF INSTALLATION

For successful operation of OMCS under ISPF you must place the supplied OMCS panels where they can be found and provide for the invocation of OMCS programs as ISPF dialog functions by modifying and extending your existing ISPF panel structure. The following notes assume that the installer has some familiarity with the structure and syntax of ISPF panels.

Invoking the OMCSTSSJ program from an ISPF selection panel

The OMCSTSSJ program provides access to OMCS database contents in TSO.

```
)PROC
&REST=TRUNC(&ZCMD,'.')
&REST=.TRAIL

&ZSEL=TRANS(TRUNC(&ZCMD,'.')
SJ,'PGM(OMCSTSSJ) PARM(&REST @DB=''your.dsname'') NEWAPPL(OMCS) NOCHECK'
```

The OMCSTSSJ program must be told the full dataset name of the database to be accessed via the @DB= keyword in the parameters passed to it. If you have several databases to be accessed then you should set up several options on your selection panel:

```
)PROC
&REST=TRUNC(&ZCMD,'.')
&REST=.TRAIL

&ZSEL=TRANS(TRUNC(&ZCMD,'.')
JCL,'PGM(OMCSTSSJ) PARM(&REST @DB=''OMCS.JCL.DBASE'') NEWAPPL(OMCS) NOCHECK'
STC,'PGM(OMCSTSSJ) PARM(&REST @DB=''OMCS.STC.DBASE'') NEWAPPL(OMCS) NOCHECK'
```

Setting up a single panel for access to a number of OMCS databases.

A sample selection panel is provided in the .PANL distribution library as member OMC@SPAN. This panel can be used as an example showing how to set up a single panel that documents the availability of, and provides access to, a number of OMCS databases using the same techniques shown above. To tie this panel into your existing ISPF selection panel structure you would put an entry like the following into the)PROC section of the appropriate existing panel:

```
)PROC

&ZSEL=TRANS(TRUNC(&ZCMD,'.')

O,'PANEL(OMC@SPAN) NEWAPPL(OMCS)'
```

Remember, if you simply invoke OMCS command processors under ISPF using one of the TSO command interfaces (such as option 6) then the program will work but ISPF display services will not be used and you will not be able to use SPLIT and SWAP.

Placement of OMCS dialog panels

Execution of the OMCS command processor requires that the OMCS dialog panels be available via the DD name ISPPLIB. You must include the .ASEPANL library in the concatenation. Warning: do not compress the panel library that contains OMCS panels.

There are no ISPMLIB, ISPSLIB or ISPCLIB components.

Invoking OMCS command processors correctly under ISPF

OMCS provides several programs that can operate as command processors in TSO. These programs process their own display panels. The resultant screen images can be displayed either with or without the use of ISPF display services. As users we generally prefer to use ISPF display services as it gives us the convenience of the ISPF split screen facility as well as other benefits. OMCS programs will only use ISPF display services if they are invoked according to the rules for ISPF dialog functions. See the examples a little later on.

OMCSTSSJ - the job selection interface program

This program gives users the access they need to the jobs OMCS has captured and filed in its databases. OMCSTSSJ lets you specify what jobs you want to see in a variety of ways and then shows you a scrollable list that you can select from for viewing, purging, hardcopy, etc. OMCSTSSJ also provides access to a number of other OMCS applications such as ACF, JSF, IRM, and to housekeeping facilities like the ECL testbed, database directory, etc.

ECPX - general ECL program executor

The ECPX program can be used to run programs written in ECL, an integral part of the OMCS product. ECPX is used as either a TSO command or as a batch program via JCL. You can write your own ECL programs and execute them using ECPX.

COMMAND ===> ECPX MYECP HENRI BOB CARLOS

will cause ECPX to search the library or libraries allocated to DD name ECPLIB for member MYECP and run it passing the character strings “HENRI”, “BOB” and “CARLOS” to the program.

Please refer to the manual **ECL Language Reference OM88-1027** for information about the ECL language.

OMCSECLC - the ECL program compiler

OMCSECLC is a program which will operate in TSO or batch. It takes an ECL source program and creates a load module from it. Such load modules can then be executed just like any other program and they can operate as TSO command processors. The OMCSECLC program itself was produced in exactly this way.

Running ECL programs in TSO

There are three ways to run ECL programs in TSO:

- You can use the ECPX program (described above) to run an ECP in TSO -or-
- Using a "T" line command against a job in the selection lists that OMCSTSSJ displays -or-
- You can use the OMCSECLC program to create a load module from the ECL source program and then execute that load module.

The following notes apply to the first two of these situations.

All ECL programs are retrieved from the library(ies) allocated to the DD name ECPLIB (or SYSPROC, see below). The normal rules for concatenation of partitioned datasets apply.

ECPLIB may be pre-allocated in the logon proc JCL or it may be allocated by other means after logon.

The ECL interpreter looks for DD name ECPLIB first. If no allocation for ECPLIB is found then it looks for DD name SYSPROC. If neither is allocated then execution will fail with a “member not found” message.

This option can make it convenient to put ECL programs into existing CLIST libraries rather than requiring you to allocate new dedicated ECL libraries.

THE SYSTEM COMMAND ISSUING MODULE AND SVC

Several OMCS components in normal operation may issue a subset of MVS system commands. OMCS components needing this service call a common module, OMCSISCM, to provide it. Module OMCSISCM issues commands to MVS via the MGCR interface and this interface requires that the address space be APF authorised when it is invoked.

OMCSISCM invokes the OMCS svc (a type 3 or 4) to obtain APF authorisation immediately before using the MGCR interface and then again immediately afterward to give up that authorisation. The SVC as supplied by ASE will check that the issuing module has been loaded from an APF authorised library.

Please refer to these members of the install package :-

ASE.ASESAMP(OMCSSVC)

This is the source unit for the svc. ASE will allow limited modification of this source unit for the purposes of maintaining installation security or MVS integrity upon the condition that any modifications are notified to ASE in advance in writing.

ASE.ASESAMP(OMCSISCM)

This is the source unit for the OMCSISCM module.

ASE.ASEMAC($ISCM)

This is the macro containing the OMCSISCM detail.

ASE.ASESAMP(GENISCM)

This is a sample job to be tailored and run to install both the OMCSISCM module and the SVC.

INSTALLING ACCESS CONTROL FACILITY USER EXITS

Do I have to do this now?

Initially you do not need ACF exits to operate OMCS/V2. If you are pressed for time this activity can be performed at the point where internal OMCS security becomes an issue. If this applies in your case then skip this section.

Part 3 of this manual contains a section titled THE OMCS ACCESS CONTROL FACILITY which covers all aspects of the operation of OMCS/ACF. Refer also to the manual ACF Users Guide OM88-1023. This discussion in this section of the install guide is intended to clarify why we need an exit module to establish the real value of UID in non-TSO address spaces.

What is OMCS/ACF and the Access Control Language?

OMCS comes with an integrated security facility (OMCS/ACF). ACF lets you control access to specific database files or groups of files using a set of rules which you create using ISPF EDIT and then compile into each database. Each rule specifies a set of tests and one or more actions to be done if all the tests are satisfied. The complete set of rules is inspected each time a program tries to open or delete any database file. The default action, if all rules are tested without the PREVENT or ALLOW actions being invoked, is PREVENT.

Lets look at an example of a single rule source statement and at the description which follows:

```
IF UID(TASE2) AND DAY(SAT,SUN) AND ACCESS(D) THEN PREVENT
```

IF the user making the current database file request is TASE2 and the day of the week is SATurday or SUNday and the access requested is Delete (ie. something is trying to delete a file) then PREVENT (disallow) the request.

Lets look at another example:

```
IF FILE(*   LOGFILE) AND ACCESS(I) THEN ALLOW AND LOG
```

IF the current database file request is for a file with a filetype of LOGFILE and the access requested is Input (ie. read-only) then ALLOW the request and LOG the request details on the MVS system log.

Each of the things being tested in these rules (**FILE**, **DAY**, **ACCESS** etc) is called a **RESOURCE CHECK ITEM**. OMCS/ACF provides about a dozen such resource check items that you can refer to when you code access rules. You can also define your own resource check items using the exit facilities of OMCS/ACF.

For example, lets assume that you only want users belonging to department MISOPS to have read-only access to SYSLOG stored in an OMCS database. User department is an attribute that is maintained by your system security package, so you need an exit module that can call that package and determine the requesting users department code and pass that information to OMCS/ACF. Assuming you called your exit module MYACFXIT then your ACF rule file would look like the following:

```
EXIT ID(ABC) EP(MYACFXIT) DEFINES(DEPART)  /*define the exit module

IF   FILE(SYSLOG   #*)                     /* if the file contains SYSLOG data
     AND ACCESS(I)                         /* and access is read-only
     AND DEPART(MISOPS)                    /* and department is MISOPS
     THEN ALLOW                            /* then allow the request
```

Or, lets say that you have a database in which the batch work from several TSO users is being captured. Lets assume that, like most TSO users, the jobs that they submit have names that begin with their own logon-ids. If you want every TSO user to have read-only access to the jobs run by other users and complete access (including delete) to jobs whose names begin with their own logon-ids then you would code statements like these:

```
  EXIT ID(ABC) EP(MYACFXIT) DEFINES(UID)

  IF    FILE(*   #*)                          /* if the file contains SYSOUT data
        AND ACCESS(I)                         /* and access is read only
        THEN ALLOW                            /* then allow the request

  IF    FILE(&UID*   #*)                              /* if jobname starts with
LOGONID
        AND ACCESS(D)                         /* and access is DELETE
        THEN ALLOW                            /* then allow the request
```

Finding out what the userid is in non-TSO address spaces

From the preceding paragraphs you can see that many access rules will probably be coded in terms that include the identity of the user making the request. OMCS/ACF defaults UID to the TSO logonid in TSO but it defaults it to eight question marks ???????? in non-TSO address spaces.

We can only overcome this problem by using an exit module which will determine what the true userid is in any address space.

If you have the ACF2 or RACF security packages

Sample exits are supplied in source form as members ACF2UIDX and RACFUIDX in the ASE.ASESAMP library.

Another member, GENACFX, is supplied as an example of how to assemble and link such a module.

The supplied sample exit will require very little modification (it might require none) and it should be assembled and linked at this point so that the OMCS ACF rule checker can have access to a useable definition of the resource name UID in non-TSO address spaces.

A perusal of the sample exit code will show how the security programmer can define further ACF resource names to suit the installations requirements. Please also read the OMCS/ACF Users Guide - OM88-1023

An installation may have a requirement for more than one such exit module. OMCS/ACF allows you to define up to 8 exit modules in a single set of rules.

If you have another security package

In most cases you should be able to use ACF2UIDX as a guide to how to code a suitable exit for your own environment.

Even though the responsibility for such exits can only rest with installation security programmers, you may call ASE support to discuss your specific situation. We will do our best to advise and assist you.

INSTALLING IRM REPORT VIEWING FOR CICS

The OMCS/CICS interface is one of two components involved in supplying Report Viewing through CICS. The first component is a CICS program and transaction. It communicates with the second component, the report viewing application, which resides in a separate address space. It processes requests

(transactions) from many users. This is referred to as a "server" address space as it can service many users simultaneously. These users may be logged on to one or more different CICS regions.

Making OMCSCICS available to your CICS region(s)

The OMCSCICS load module must be in a library allocated to the DFHRPL DD statement in your CICS region(s). Add your ASELIB library to the concatenation. Alternatively, you may copy OMCSCICS into an existing DFHRPL library.

Note that OMCSCICS is the only OMCS load module required by CICS regions.

Defining the OMVT Sub-system

The OMVT subsystem entry must be defined in your system before OMCS/CICS can be used. Add the following line to the current IEFSSNxx member of SYS1.PARMLIB.

```
OMVT,OMCSSSI
```

After the next IPL of your system the OMVT subsystem will be available. Note that the program OMCSSSI must reside in an APF authorised library which is accessible through the MVS link list.

Defining OMCS to Your CICS System(s)

In all CICS systems from which report viewing will be used, the following resources must be defined:

- Program OMCSCICS - this is an assembler program. There are no other special requirements - the CICS defaults may be used.
- Transaction RVF - this must be associated with program OMCSCICS. There are no other special requirements - the CICS defaults may be used.

Creating a Server Started Task

JCL is required for the server address space. This runs as a started task and is where the report viewing application is executed. Here is some sample JCL with descriptions for parameters and DD statements following. This sample is supplied as member OMCSRVF in the ASESAMP library.

```
//OMCSRVF  PROC
//PPCF     EXEC PGM=OMCSPPCF,REGION=6144K,TIME=1440,
// PARM='PGM=OMCSSATR,APPLID=RVF,MAXUSER=100'
//*
//STEPLIB  DD DISP=SHR,DSN=your.ASELIB
//ECPLIB   DD DISP=SHR,DSN=your.ASEECPS
//ISPPLIB  DD DISP=SHR,DSN=your.ASEPANL
//DBASCTL  DD DISP=SHR,DSN=your.dsname  <- audit trail database
//DBASSARV DD DISP=SHR,DSN=your.dsname  <- IRM RVF database
//PPCFDIAG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

Server JCL Parameters

The following parameters are required on the PARM= operand in the server address space JCL.

- **PGM=pgmname**

  This is the name of the program which executes the application. For the report viewing application this is always OMCSSATR.

- **APPLID=xxxx**

  This is the application id (up to four characters). Initially this should be RVF which matches the transaction name defined to CICS. If there is more than one server address space, each one must have a different application id. The CICS transaction name used is usually the same as this application id. For details about connecting to an application see the section entitled "Using the OMCS CICS Transaction".

- **MAXUSER=nnn**

  This is the maximum number of users which may be connected to the application simultaneously.

Server JCL DD Statements

The following DD statements are required in the server address space JCL.

- **//STEPLIB**

  This is the OMCS load library (ASELIB). It must be APF authorised.

- **//ECPLIB**

  This is the OMCS ECL source library (ASEECPS).

- **//ISPPLIB**

  This is the OMCS panel library (ASEPANL).

- **//DBASCTL**

  This is an OMCS database which will contain the report viewing audit trail file - SARV LOGFILE. It may be the same database allocated to DD DBASSARV.

- **//DBASSARV**

  This is the OMCS database which contains the IRM report viewing control files and the reports.

Operator Commands

The server address space is controlled from the MVS console using the following commands.

To start the server:

S OMCSRVF

To stop the server:

P OMCSRVF

or

F OMCSRVF,STOP

If the server fails to stop after a few minutes, you may terminate it with the following command:

F OMCSRVF,CANCEL

Using the OMCS CICS Transaction

Logon to a CICS terminal and sign on. The OMCS transaction may then be started by clearing the screen and entering:

RVF

This starts a session between OMCS/CICS and the report viewing application. The next screen you see will be the primary panel for report viewing.

Running More Than One Server

There may be more than one server address space - each of which may access a different OMCS database. Each server must have a different value of the APPLID parameter. For each application id, define a CICS transaction with the same name. For example, if you set up a new server with an application id of REPT then define another CICS transaction called REPT. The name of the CICS transaction is assumed to be the application id of the server to which connection is required. A user can then connect to the required server by entering a transaction name which is the same as the server application id.

The OMCS CICS transaction will accept a parameter which is interpreted as an application id. For example, this means there are alternative ways to connect to application REPT. Entering

REPT

connects to application REPT while entering

RVF REPT

connects to the same application.

PART 3 - THE QUICKSTART GUIDE

This Quickstart Guide is intended to get you going without you having to spend a lot of time jumping from manual to manual getting basic information. It is not intended to be a long-term substitute for the complete set of manuals identified in the box Related Publications inside the front cover of this manual and in some places you will be referred to some of those manuals for more detailed information.

Quickstart Guide Overview

The Quickstart Guide covers the following:

- Allocating and formatting a database
- The OMCS Access Control Facility
- Starting and stopping the OMCS writer task
- Using the OMCSTSSJ program to access the database
- Controlling automatic space reclamation
- Recovering output from the secondary archive
- Maintaining Job Processing Rules

The Quickstart Guide does not tell you how to use the following OMCS optional extension products. They are fully described in the manuals listed inside the front cover of this manual.

- IRM (interactive Report Management)
- JSF (Job Scheduling Facility)

ALLOCATING AND FORMATTING A DATABASE

You can actually use any block size you like in the range 4096 to 32760, however there are recommended sizes for you to choose from which give you the best utilization of disk space:

Device type	Recommended blocksizes
3380	4276 4820 5492 6356 7476 9076 11476 15476 23476
3390	5064 5726 6518 7548 8906 10796 13682 18452 27998

For a start, use **BLKSIZE=9076** (if its a 3380) or **BLKSIZE=10796** (if its a 3390) and **SPACE=(CYL,100)**

OMCS databases must be formatted before you can use them. Program OMCSDBFM is provided to do this.

A sample job to allocate and format a database is supplied as member RUNDBFM in the ASESAMP library.

The formatting should take about a minute to do 100 cylinders of 3380.

THE OMCS ACCESS CONTROL FACILITY

Why you need to do something about ACF now

OMCS/ACF consists of two parts, a rule compiler and a rule checker. The rule compiler takes rule source language files and creates a ruleset file in the database. The rule checker is an integral part of the database access method. The rule checker insists on finding a ruleset file in the database and checks every request to OPEN or DELETE a file against the ruleset.

When an OMCS database is initially formatted, (see sample job RUNDBFM in the ASE.ASESAMP library) the access method itself creates a rudimentary ruleset file in the database (rulesets are held in a database file called ACCRDATA SYSFILE). This ruleset is equivalent to the Access Control Language (ACL) source statement:

```
IF THEN ACFOFF                    /* turn off OMCS/ACF
```

The ACFOFF action causes the ACF rule checker to turn itself off for the rest of the database session and to log the fact that it is doing so. The two-line message DAIO83I is produced. This default action is intended to remind staff that until someone compiles their own rules the database has no effective security.

As soon as a database has been formatted the responsible person should use the ACF rule compiler to replace this initial ruleset with one which puts into effect the rules that the installation requires. See How to run the ACF rule compiler below.

Disabling OMCS/ACF

If an OMCS database doesn't have an ACCRDATA SYSFILE in it, or if the format of that file is in any way unacceptable, then OMCS cannot be used to access the database except as follows:

A zap is supplied in the ASESAMP library as member ZAPACF. Applying this will turn off the ACF functions in the access method. If you want to restrict this no-ACF situation to just yourself then you should copy the module ASEDAMGR to another authorised library which can be used as a STEPLIB or

ISPLLIB by the relevant TSO user (or batch ACF compiler job), apply the zap to that copy and use the copy until the ruleset has been satisfactorily re-created.

How to run the ACF rule compiler

The rule compiler can be run in TSO or in batch. Batch is attractive when the rule source files are large, when CPU time in TSO is at a premium or when you want something compiled as part of a batch process without having to pay close attention to it.

In TSO: Invoke OMCSTSSJ under ISPF (refer section ISPF INSTALLATION in part 2 if this has not already been done) and invoke the ACF rule compiler like this:

```
OMCS/V2 Job Selection Criteria  ASE.DATABASE 75% -----------------------------
COMMAND ===> acf_                                       ? for field-level help

primary commands ACF,AMF,JSF,IRM,JAL,DBI,DIR,SI,JPR,DDR,X
Enter Job Selection Criteria and press ENTER----------------------------------
jobnames     ,==> TGSG
not before   ===> -2             yyyymmdd, mmdd, dd, tue, 25oct, -10, etc
```

This will bring up a display that lets you compile an ACF rule source file that you created using ISPF EDIT.

```
OMCS/ACF COMPILE DATABASE ACCESS RULES----------------------------------------

 OPTIONS
      R    COMPILE THE RULE SOURCE AND REPLACE DATABASE RULESET
      T    COMPILE THE RULE SOURCE TO CHECK SYNTAX
      E    EDIT THE RULE SOURCE
      M    BROWSE THE MESSAGE FILE AFTER A COMPILE
      D    DECOMPILE AND BROWSE CURRENT RULE SOURCE

 ISPF RULE SOURCE LIBRARY
   PROJECT  ===> ASE
   GROUP    ===> P390
   TYPE     ===> CNTL
   MEMBER   ===>

 OTHER PARTITIONED DATASET
   DSNAME   ===>
```

If your rule source file contains errors then the compiler automatically throws you into a scrollable error list display that should make the problem clear. If this happens then when you have finished browsing (or printing) the error file you should press END to return to the compiler screen.

In batch: A sample job to run the ACF rule compiler is supplied as member RUNACF in the .ASESAMP library. This may be useful when re-organising databases or setting up a new database using batch jobs overnight, etc.

Making up a set of rules just to get going

Assuming you did not get your installer to disable ACF with the supplied zap you can still achieve the same effect with a one-line rule file like this one:

```
     IF THEN ALLOW                      /* let anyone do anything
```

A sample is provided in the installation .ASESAMP library as member ACFALLOW. That member contains the following statements:

```
IF ACCESS(D)              /* IF SOMETHING'S TRYING TO DELETE A FILE
   FILE(/AMF AMFINDEX,    /* AND ITS THE AMF INDEX FILE
        .JOBDETL MASTER,  /* OR THE JOB DETAIL MASTER
        .JOBSEQ MASTER,   /* OR THE JOB SEQUENCE MASTER
        .JOBRULES MASTER) /* OR THE JOBRULES MASTER
   THEN PREVENT AND LOG   /* STOP IT AND TELL

IF THEN ALLOW             /* ALLOW ALL OTHER ACCESSES
```

This allows anyone to do anything except delete some critical OMCS files.

Please refer to the Access Control Facility Users Guide - OM88-1023 for more information.

STARTING AND STOPPING THE OMCS WRITER TASK

Writer parameter file

The operation of the sysout capture function of the OMCS started task is controlled by a parameter file pointed to by the //XWAFIN DD statement in the catalogued JCL procedure. You need to know where that parameter file is and you should review its content before proceeding. Information about the content of this file is contained in member CTLXWAF in the ASE.ASESAMP library.

Using START, STOP and MODIFY commands

(These examples assume that the started task JCL procedure created during the install was called OMCSV2).

Start OMCS like this:

S OMCSV2

OMCS should issue a few messages and then wait for work.

You can now submit work with MSGCLASS set so as to put test jobs onto spool in the appropriate output class.

Issue a stop command like this:

P OMCSV2

When OMCS has finished processing the current job it will stop. Remember that, if the job being processed is very large it might take up to a minute to complete processing before stopping.

You can also stop OMCS by stopping all its sub-functions. These are stopped by using a MODIFY command like this:

F OMCSV2,WTR=STOP

If you believe you must stop OMCS immediately regardless of what it is doing then you can use:

F OMCSV2,CANCEL

Diagnostic aids

The OMCS started task process will write diagnostic information via three DD statements (XWUFDIAG, XWAFDIAG, XWMTDIAG) which have been included as comments in the supplied JCL procedure. If any problem occurs these should be uncommented and checked to ensure that the specified sysout class is satisfactory. Subsequent operation will cause information to be written out which may be of assistance in discussing the problem with ASE.

USING THE OMCSTSSJ PROGRAM TO ACCESS THE DATABASE

Once you have some output in the database you can look at it with the OMCSTSSJ program.

Make sure you can invoke OMCSTSSJ from ISPF

During the installation process the installer should have provided an ISPF option to invoke the OMCSTSSJ program for the database you have just allocated and formatted. If this was not done then it should be done now. Please refer to Part 2 of this manual - **ISPF INSTALLATION**

However, just to get you going you can invoke the OMCSTSSJ program either outside ISPF or inside ISPF under option 6 by typing the following command:

COMMAND ===> OMCSTSSJ @DB=your.database.dsname

(Remember, you wont be able to use ISPF split screen if OMCSTSSJ has been invoked in this way.)

The Job Selection criteria display

You should see a job selection criteria display that looks like this:

```
OMCS/V2 Job Selection Criteria  ASE.DATABASE 75% ------------------------------
COMMAND ===>                                            ? for field-level help

primary commands ACF,AMF,JSF,IRM,JAL,DBI,DIR,SI,JPR,DDR,X
Enter Job Selection Criteria and press ENTER-----------------------------------
jobnames     ,==> TGSG
not before   ===> -9999          yyyymmdd, mmdd, dd, tue, 25oct, -10, etc
not after    ===> -0             yyyymmdd, mmdd, dd, tue, 25oct, -15, etc
symptoms     ,==>
COND codes   ,==> >0                              >4 .. =16,24 .. <12 .. ^' '
assignees    ,==>
programmer   ,==>
lines        ===>             (like >5000 or >5k or <500 etc)
form codes   ,==>
cpu time     ,==>
sort by      ,==> CPU
other ...    ,==>
latest @ top ===>       (Y if selection list is to have latest job at top)
conf delete  ===>       (N if delete confirmation not required)
prt destid   ===>          (system printer destid)
prt class    ===>          (output classcode)
```

This is the alternate primary display for OMCS in TSO. A variety of other OMCS functions can be invoked from here but we will just look at viewing job output for now.

You use this display for job output viewing by entering data in some of the fields and pressing ENTER. In the example shown here, the user was interested in jobs whose names began TGSG, ran during the previous two days and which had at least one COND code which was greater than 0000. The user also requested that the selected entries be sorted on the CPUSEC field before being displayed.

Pressing ENTER produced the display shown below. The header information line has the following fields:

OMCS/V2 JOBS:	fixed constant that always appears
TGSG	the contents of the 'jobnames' criteria field
-2	the contents of the 'not before' criteria field
ASE.DATABASE	the database dataset name
75%	percentage of the database that is available for storing new data
14:13... etc.	current time and date

```
OMCS/V2 JOBS: TGSG -2   ASE.DATABASE 75% 14:13 MON 19941217
COMMAND ===>                                              Scroll ===> CSR
                                                          Line 1     of 1683
S browse Y summar H hardcop P purge D delete U update T test A analyz R restore
lc JOBNAME- JOBID- RUNON DAY STAR STOP ELAPS CPUSEC LINES- MXCC SYMPTOM--- SYST
   TGSGTST0 J06664 DEC15 THU 1743 1743  0:01    0.1  3,042 0004 ABS806     CPUA
   TGSGTST0 J09771 DEC16 FRI 1536 1536  0:01    0.1  3,042 0004 ABS806     CPUA
s  TGSGTSTV J01417 DEC17 SAT 1151 1151  0:01    0.2    206 0004 ABU1006    CPUA
   TGSGTSTV J01419 DEC17 SAT 1151 1152  0:01    0.3    267 0004 ABU1006    CPUA
   TGSGTSTW J01418 DEC17 SAT 1151 1152  0:01    0.3    267 0004 ABU1006    CPUA
y  TGSGTSTV J01421 DEC17 SAT 1152 1152  0:01    0.3    267 0004 ABU1006    CPUA
```

You can use several line commands against this selection list:

S	selects a job for browsing.
Y	displays a dataset summary like the one below.
H	hardcopies the job output using the printer/class on the primary display.
P	deletes just the job and marks the index entry with a vertical bar.
D	deletes both the job and its index entry from the database.
U	displays the job detail data and allows certain fields to be changed.
T	invokes the ECL testbed for this job.
A	invokes the IRM report discovery interface to process this job.
R	recovers the job from the secondary archive and marks the entry with "@".
DI	performs IRM Dataset Identification using the best match rule for the job name.

```
OMCS/V2 DATASET SUMMARY FOR TGSGTSTV #0000067 1151  SAT 17 DEC ----------------
 COMMAND ===>                                             SCROLL ===> CSR
                                                          LINE 1     OF 7
 use S BROWSE H HARDCOPY
 lc PREXNAME PGEXNAME DDNAME--   LINES  PAGES C CPY FORM DESTID-- JES2 DSNAME
             JES2     JESMSGLG      30      1 U   1 STD  LOCAL    .PS000002
             JES2     JESJCL        22      0 U   1 STD  LOCAL    .PS000003
             JES2     JESYSMSG      94      0 U   1 STD  LOCAL    .PS000004
             STEP1    SYSPRINT     108      4 U   1 STD  LOCAL    .PS000101
             STEP1    SYSOUT        11      1 U   3 STD  LOCAL    .PS000102
             STEP1    SYSTERM        2      1 U   1 15X2 LOCAL    .PS000103
                                    16      0     1               .JOBAUDT
```

This dataset summary display (see "Y" line command above) allows two line commands:

S selects a dataset for viewing
H selects a dataset for hardcopy print.

When you have selected a job for viewing you should see a display like the following:

```
OMCS/V2 TGSGTSTV J01417 1151  SAT 17 DEC TGSGTSTV #0000067---------------------
 COMMAND ===>                                                 SCROLL ===> CSR
                                    LINE 1        OF 223      COLS 1:76

 1                       J E S 2  J O B  L O G  --  S Y S T E M  C P U A  --
 0
  11.51.45 JOB01417  IRR010I  USERID TGSG    IS ASSIGNED TO THIS JOB
  11.51.45 JOB01417  ICH70001I TGSG   LAST ACCESS AT 11:45:57 ON SATURDAY, DEC 17
  11.51.45 JOB01417  $HASP373 TGSGTSTV STARTED - INIT    8 - CLASS E - SYS CPUA
  11.51.51 JOB01417  -                                          --TIMINGS (MIN
  11.51.51 JOB01417  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB
  11.51.51 JOB01417  -TGSGTSTV STEP1                04     53    .00    .00
  11.51.52 JOB01417  -TGSGTSTV STEP2                04      6    .00    .00
  11.51.52 JOB01417  IEA995I SYMPTOM DUMP OUTPUT
                      ABEND CODE  USER= 1006 TIME=11.51.52 SEQ=21673 CPU=0000
                      PSW AT TIME OF ERROR  078D1000   00006BC4  ILC 2  INTC
                        ACTIVE LOAD MODULE=OMCSDBUT  ADDRESS=00006928  OFFSET
                        DATA AT PSW  00006BBE - 00181610  0A0D45E0  C3AE4770
                        GPR  0-3  80000000  800003EE  0000D014  00000084
                        GPR  4-7  006ECB00  006E9D70  0000D0B0  FD000000
                        GPR  8-11 006AE1C0  0000CFA8  00000000  00007928
                        GPR 12-15 00006928  0000CFA8  80006BB4  00000000
                      END OF SYMPTOM DUMP
  11.51.52 JOB01417  IEF450I TGSGTSTV STEP3 - ABEND=S000 U1006 REASON=0000000
                             TIME=11.51.52
  11.51.52 JOB01417  -TGSGTSTV STEP3             U1006      0    .00    .00
```

The top line contains summary information from the entry that we selected to get this display;

The third line contains current line number, total lines and the current columns information.

The rest of the screen is a scrollable display of the data.

You can use the following commands in the primary command area on the second line:

N(ext) to skip to the start of the next SYSOUT dataset. You can also specify a skip count, **N 3** will skip three datasets, for example and **N 0** will skip to the start of the current sysout dataset.

P(rev) to skip to the start of the previous SYSOUT dataset, **P 3** will skip three datasets, for example and **P 0** will also skip to the start of the current sysout dataset.

F(ind) to search for text in the file being browsed. The **PREV**, **FIRST** and **LAST** keyword options are supported. **F TOTALS LAST** will start at the bottom of the file and search backward looking for the text string "TOTALS"

PRT to print all or part of the file being viewed. PRT will display a menu. The user completes appropriate fields and presses ENTER to print or END to abandon the print request.

DSI to invoke the DataSet Identification rule editor. A selection list of existing rules will be displayed. You can create new rules with the **ADD** *rulename* primary command and you can update or delete existing rules with **U** and **D** line commands. An **R** line command lets you run the rule processor against the job that you were browsing and then shows you an audit log of the process.

Now, lets go back to the job selection criteria display.

Other fields on the selection criteria display

```
OMCS/V2 Job Selection Criteria  ASE.DATABASE 75% -----------------------------
COMMAND ===>                                            ? for field-level help

primary commands ACF,AMF,JSF,IRM,JAL,DBI,DIR,SI,JPR,DDR,X
Enter Job Selection Criteria and press ENTER-----------------------------------
jobnames     ,==> TGSG
not before   ===> -2             yyyymmdd, mmdd, dd, tue, 25oct, -10, etc
not after    ===>                yyyymmdd, mmdd, dd, tue, 25oct, -10, etc
symptoms     ,==>
COND codes   ,==> >0                              >4 .. =16,24 .. <12 .. ¬' '
assignees    ,==>
programmer   ,==>
lines        ===>             (like >5000 or >5k or <500 etc)
form codes   ,==>
cpu time     ,==>
sort by      ,==> CPU
other ...    ,==>
latest @ top ===>       (Y if selection list is to have latest job at top)
conf delete  ===>       (N if delete confirmation not required)
prt destid   ===>          (system printer destid)
prt class    ===>          (output classcode)
```

The list-support indicator ,==> Whenever a field supports multiple arguments separated by commas we have placed a comma in the start of the field prompt arrow. For example, the jobnames field allows you to specify more than one name, like this:

jobnames ,==> TGSGBKUP,HSKPBKUP

Sort by ,==> As we saw a little earlier, this field can be used to specify one or more field names that the selection list is to be sorted on before it is displayed. You only need to specify as much of the field names as is necessary to allow them to be found in the selection list heading line. Field names should be specified with major name first.

You can also sort the selection list at any time by using the SORT or BY commands in the primary command area on the selection list display.

Other ...,==> allows you to specify all those selection criteria that we couldn't provide specific fields for on the panel due to physical screen size limitations. In this field you can put things like:

other ... ,==> STAR>0800,STAR<1200

This tells OMCS that the selection list is to include only jobs where the STAR (execution start time) time is later than 8am (0800) and is earlier than 12 noon (1200). Here's another:

other... ,==> DAY^(SAT,SUN) (that ^ is supposed to be a **not** sign)

(that says, show me jobs that ran on days other than Saturday or Sunday).

conf delete ===> if set to **N**, lets your delete and purge requests go through without confirmation (though still under the control of ACF). If it's any other value then you will be thrown into a browse display of each job you try to delete and you must type a Y to confirm that the delete is to proceed.

prt destid ===> and **prt class ===>** let you specify the printer destination and classcode to be used for any printing done using **H** (hardcopy) line commands.

Field level help

If you type just a question mark **?** into any field on the criteria display then you will see brief information about the syntax supported for that field in a three line area towards the bottom of the screen.

Primary commands available on the selection criteria display

ACF will display the ACF rule compiler menu

AMF will display the Archive Management Facility menu

DBI will display physical database information

DDR will invoke the Dynamic Database Reconfiguration Facility.

DIR will display a full or partial database directory list.

END will terminate the OMCSTSSJ program

IRM will display the Interactive Report Management menu

JAL will display the Job Audit logfile

JSF will display the Job Scheduling Facility menu

JPR will invoke the Job Processing Rules maintenance function

SI will display MVS system information

Now lets take a closer look at the selection list display

The selection list display

The selection list display is a conventional scrollable list with a heading line, a primary command area, a field heading line and a scrollable data area. Each data line has a line command field at the extreme left side of the screen

The selection list is initially presented to us showing fields like jobname, jobid, run date and start and stop times, etc. Most of the time the information we want will be on this display.

```
OMCS/V2 JOBS: TGSG -2   ASE.DATABASE 75% 14:35 MON 19941217
COMMAND ===>                                                Scroll ===> CSR
                                                            Line 1     of 61
S browse Y summar H hardcop P purge D delete U update T test A analyz R restore
lc JOBNAME- JOBID- RUNON DAY STAR STOP ELAPS CPUSEC LINES- MXCC SYMPTOM--- SYST
   TGSGTST0 J06664 DEC15 THU 1743 1743  0:01    0.1  3,042 0004 ABS806     CPUA
   TGSGTST0 J09771 DEC16 FRI 1536 1536  0:01    0.1  3,042 0004 ABS806     CPUA
s  TGSGTSTV J01417 DEC17 SAT 1151 1151  0:01    0.2    206 0004 ABU1006    CPUA
   TGSGTSTV J01419 DEC17 SAT 1151 1152  0:01    0.3    267 0004 ABU1006    CPUA
   TGSGTSTW J01418 DEC17 SAT 1151 1152  0:01    0.3    267 0004 ABU1006    CPUA
y  TGSGTSTV J01421 DEC17 SAT 1152 1152  0:01    0.3    267 0004 ABU1006    CPUA
```

OMCS keeps other information about jobs. To see that we need to scroll to the right. If we do that once then we see a display like this next one.

```
OMCS/V2 JOBS: TGSG -2   ASE.DATABASE 75% 14:35 MON 19941217                RIGHT
 COMMAND ===>                                               Scroll ===> CSR
                                                            Line 1     of 8
 S browse Y summar H hardcop P purge D delete U update T test A analyz R restore
 lc JOBNAME- JOBID-  ASSIGNED PGMRNAME/DESC------- NOTEPAD----------------------
    TGSGTSTV J00051
    TGSGTSTW J00052
    TGSGTSTW J00054
    TGSGTSTV J00053
    TGSGTSTW J00056
```

And further right shows us more information.

```
OMCS/V2 JOBS: TGSG -2   ASE.DATABASE 75% 14:35 MON 19941217               RIGHT
 COMMAND ===>                                               Scroll ===> CSR
                                                            Line 1     of 8
 S browse Y summar H hardcop P purge D delete U update T test A analyz R restore
 lc JOBNAME- JOBID- ------ ARCDAT ARCDIS SCRDAT PAGES- FORMS--- FCHDAT FCHLOC FC
    TGSGTSTV J00051               950326 941217     17 15X2
    TGSGTSTW J00052               950326 941217     17 15X2
    TGSGTSTW J00054               950326 941217     17 15X2
    TGSGTSTV J00053               950326 941217     17 15X2
    TGSGTSTW J00056               950326 941217     17 15X2
```

And further right again shows us more information, generally less and less interesting!

```
OMCS/V2 JOBS: TGSG -2   ASE.DATABASE 75% 14:35 MON 19941217               RIGHT
 COMMAND ===>                                               Scroll ===> CSR
                                                            Line 1     of 8
 S browse Y summar H hardcop P purge D delete U update T test A analyz R restore
 lc JOBNAME- JOBID- HVOL   NDXDIS CAPTURED--------- RECALLER SDS#---- STLINE FL
    TGSGTSTV J00051        950326 941216-164954-FRI          #0000007
    TGSGTSTW J00052        950326 941216-164957-FRI          #0000007
    TGSGTSTW J00054        950326 941216-165002-FRI          #0000008
    TGSGTSTV J00053        950326 941216-165005-FRI          #0000008
    TGSGTSTW J00056        950326 941216-165112-FRI          #0000009
```

Primary commands available on the job selection list display

BY fieldname [fieldname] sorts the selection list on the fields named.

COLS inserts a columns index line across the top of the scrollable area.,

END returns you to the selection criteria display

F(ind) find commands are supported to allow you to search the list

J [jobnames] [notbefore] [notafter] [symptoms] tells the program that you want to respecify some of the selection arguments and then see the new selection list that results. An example might be where we have made a selection for, say, jobname TGSH (finger slipped on the keyboard) when we meant TGSG. Rather than return to the selection criteria display and change it there we can just type the following command and press ENTER:

J TGSG

You can respecify any of the four criteria described above. When you wish to respecify **symptom**, say, without changing **jobnames**, **notbefore** or **notafter** then you use an asterisk (*****) in place of those criteria you do not want to change. Assuming that we wanted to respecify the symptoms without changing any of the other criteria we would type the following command and press ENTER:

J * * * ABS222,ABS122 (note that blank separates each of the criteria terms)

SORT fieldname [fieldname] sorts the selection list on the fields named.

=... OMCSTSSJ lets you specify any primary command prefixed with a dash **-** or equals **=**. OMCSTSSJ will internally return to its primary display and then inspect the command verb that follows the dash or equals. If the verb is one supported by the OMCSTSSJ primary display then it will be processed. If not then OMCSTSSJ will terminate. Shortly we hope to be able to successfully pass the unprocessed command string back to ISPF so that it will process it.

CONTROLLING AUTOMATIC SPACE RECLAMATION

It is not necessary to immediately do anything about automatic space reclamation. This can be put off until your database fills up for the first time. Depending on the size of your database it could be days or even weeks before it fills up.

When the OMCS writer detects a short-of-free-space condition in the database it will execute the space reclamation process to attempt to relieve the condition. If space reclaim is successful then the writer will continue processing, if not it will stop after writing messages about the situation.

The Automatic Space Reclamation Process

Automatic space reclamation consists of an ECL program, OMCSDSSX, space reclamation utility program, OMCSDBSR and, indirectly, the backup program, OMCSDBEX.

OMCSDSSX - the space reclaim process manager

OMCSDSSX is invoked by the OMCS writer to handle space shortage situations. This program, an example of which is supplied in the ASE.ASESAMP library, is best thought of as a user exit. It requires tailoring at the points indicated in the sample to operate successfully in each installation. Because OMCSDSSX is written in ECL its tailoring should be within the capabilities of operations technical support staff. ASE can be contacted for assistance with this tailoring.

After tailoring, your version of OMCSDSSX must be placed in a library concatenated to ASE.ASEECPS in your writer started task JCL.

The OMCSDBSR program is called by the OMCSDSSX program to work out what can be deleted to meet the free space percentage target that is set by OMCSDSSX.

If OMCSDBSR can meet the target by deleting files that have already been backed up then OMCSDSSX returns to the writer and the writer continues operation.

If the free space target has not been met when OMCSDBSR completes then OMCSDSSX will build and submit a batch job and wait for it to complete. The batch job is built by OMCSDSSX and the structure of that job can be established by the installation. The sample of OMCSDSSX supplied in the .ASESAMP install library builds and submits a job consisting of two steps:

Step1 executes the fast backup program, OMCSDBEX, to do an incremental backup

Step2 executes the OMCSDBSR program to delete or compress files as necessary.

The OMCSDSSX program can tell when the batch job has completed and at that point the sample program returns to the writer to allow it to continue. Should the writer detect a full (or near-full) database condition at a later time then once again OMCSDSSX will be executed and the process described above will be repeated. If the actual free space available when the batch job is submitted is at least 50% of the target that has been set then OMCSDSSX doesn't wait for the batch job to complete before it returns to the writer. This simply reduces the total time that the writer spends waiting for space reclaim to finish.

OMCSDBSR - the space reclaim selective deletion / compression program

The OMCSDBSR program is invoked with a parameter specifying a free space percentage target to be achieved.

OMCSDBSR can be further controlled or influenced in its decisions about what files to delete by control statements which the installation creates either directly as a PDS member or through the tailoring of the OMCSDSSX ECL program. These statements look like:

```
filename filetype WEIGHT(nnn) [CRUNCH]   /* nnn can be 0-32767
```

filename and filetype are explicit or generic arguments that identify files or groups of files in the database. OMCS version 2 stores the output of each job in a database file whose filename is exactly the same as the jobname and whose filetype is #nnnnnnn where nnnnnnn is a decimal number. A typical control file might look like:

```
PINV*  #*    WEIGHT(10)          /* keep these around a bit longer
SYSLOG #*    WEIGHT(5)  CRUNCH   /* compress these after a longer wait
SYSLOG #*    WEIGHT(5)           /* keep the compressed versions longer too
*      #*    WEIGHT(20)          /* all other sysout can go
```

Space reclaim searches this control file from the top for each database file looking for a match or partial match. When one is found, that control statement supplies the WEIGHT number to be used in the following calculation:

scratchindex = ageinhours X sizeinblocks X WEIGHT

example: if a SYSLOG file occupies 100 blocks and was last modified 4 hours ago then its scratchindex would be 100 X 4 X 5 ie. 2000.

Files are sorted into descending order of scratchindex and the file with the largest index is considered first for deletion or compression. Only files that have been backed up since they were last modified are considered deleteable. If space reclaim cannot achieve its free space target percentage by compressing files or deleting files that have already been backed up then it will optionally write out a file of control

statements for the backup program, OMCSDBEX, so that just the files it wants to delete can be backed up so that a subsequent execution of OMCSDBSR can then delete them.

Other control statement keywords such as ADDTOAGE, OLDRTHAN, BIGRTHAN and FORCE give additional control statement flexibility. See Database Utilities Reference - OM88-1021 for a complete description of the capabilities of program OMCSDBSR.

The file compression function of OMCSDBSR

OMCSDBSR can compress files as an alternative to deleting them. This file compression process will reduce a typical JCL listing or SYSLOG file to 1/3rd of its original size and it will achieve even better reductions on conventional computer reports.

The OMCS writer can be asked to compress output at the time it captures it by setting variable JPVASDCM to “C” (compression on) in the appropriate Job Processing Rule . However, you may choose to leave it up to the space reclaim process to make decisions about compressing files and to actually do it. The OMCSDBSR program compresses files in preference to deleting them when the control statement that a particular filename and filetype matches has the keyword CRUNCH coded on it. Of course, once a file has been crunched, a subsequent execution of OMCSDBSR may very well decide to delete the crunched file if it has been backed up. Note, however, that the crunched file will be of the order of 1/3rd the size of the original and this will reduce its scratchindex and hence the likelihood of it being chosen for deletion immediately.

The potential improvement in disk space utilization looks impressive, so why don’t we compress all files immediately? The answer is, of course, that it costs CPU time to compress a file in the first place and it costs CPU time again each time the compressed file is decompressed, for viewing online, for example.

Some output that is captured by OMCS will not be viewed or otherwise referenced at all after it has been stored in the database. Other output, like SYSLOG, may well be viewed scores of times before it leaves the database. The identification of what jobs should be compressed and how long after they reach the database is a tuning exercise that each installation will need to experiment with. Obviously the relative cost and availability of such resources as CPU time, disk space, disk i/o, tape mounts, tape storage, as well as staff time spent waiting for information, could all be taken into account in such an exercise.

Lets consider the case of SYSLOG, for example. If the viewing pattern for this type of output tends to be: 75% of all references occur in the first day, 15% in the second and the rest spread through the next four days then it might be best to compress this file after the first day, that is after 75% of the references have already been made, but with some 80% of the apparent residency requirement still to go. If you are tight on disk space and have plenty of CPU capacity then compression becomes more appealing. If the reverse is true then of course it is less so.

A further complication is the availability of recovery from secondary archive, for example tape. If this is both cheap and rapid, as when using a silo device, then compression may be less attractive. Compression before backup of course will also effectively reduce by 2/3rds the tape data transfer load. It could decrease tape usage by, say, a factor of 2. This would then be an argument for compression earlier rather than later. It could be simply achieved by running the space reclaim utility with an appropriate set of control statements (specifying CRUNCH where appropriate) immediately prior to running the fast backup program. This would apply to jobs submitted by the writers' space reclamation manager (OMCSDSSX) as well as backups scheduled in other ways.

Running backups and other space reclaim jobs

The fast backup and space reclaim programs can be run as simple batch job steps. Sample jobs are provided as members RUNDBEX and RUNDBSR in the .ASESAMP install library. It would be sensible to run a backup overnight so that space reclaim can achieve its targets quickly during the following day by simply deleting or crushing files that had already been backed up.

When the data capture rate exceeds 25% of the database capacity in a single day then you should give thought to scheduling additional backup jobs during the day. This could be done at times of lower user activity, for example at 1200 or at 1750.

Job detail index maintenance

The job detail master file is being continually extended by the OMCS writer adding new entries at the end as it captures jobs. The writer calculates several dates which are stored in the job detail record. One of these is the date at which the index entry itself is to be discarded from the job detail master. A utility program, OMCSJDET, is provided to delete expired index entries. OMCSJDET should be run on a regular basis, probably once per day or once per week, depending on how tight a control is to be maintained over the size of this file. You must set up a job to execute this program according to the JCL sample which follows. The job should be scheduled at a time when database activity is low as it will require exclusive use of the job detail master file for a short period of time.

```
//STEP1    EXEC PGM=OMCSJDET
//SYSPRINT DD   SYSOUT=*
//DBAS     DD   DISP=SHR,DSN=YOUR.DATABASE.NAME
//
```

Why space reclaims' flexibility is important

As you can see there is room for experimentation in the overall logic of space reclaim and, because that logic is controlled by a simple ECL program, OMCSDSSX, you can change the methodology to suit as your needs change, and they will change. For example, you may need a different approach at different times of the day or on different days of the week or in six months time. Each time space reclaim runs you might vary the free space percentage target, vary the WEIGHT or other keyword values or the CRUNCH option for different jobnames or at different times of the month or whatever. This need for installation controlled flexibility is the reason that we have implemented the space reclamation managment logic as an ECL program. The point we are trying to make here is that OMCS version 2 gives you the capability to achieve whatever degree of sophistication you need in database space management.

Dynamic Database Reconfiguration Facility

The Dynamic Database Reconfiguration Facility (DDRF) allows you to ATTACH secondary databases to existing databases. These secondary databases can be defined interactively using the DDR option within OMCSTSSJ or in other environments using the program OMCSDDRB. Not only can these secondary databases be ATTACHed in-flight, as it were, but they can be DETACHed (taken away) again when the need for them has passed. This adds yet another dimension to database space management. It is possible to ATTACH and DETACH secondary databases as part of space reclamation (perhaps during end-of-month processing peaks), and to make even better use of disk space. Secondary databases can be DRAINED or FORCE DRAINed when it is time to DETACH them. This might even occur nightly, after a database backup has been done. Up to 127 secondaries are supported for any primary database.

RECOVERING OUTPUT FROM THE SECONDARY ARCHIVE

OMCS version 2 provides an integrated Archive Management Facility, OMCS/AMF. You can use AMF in two ways:

Recovering job output via the Job selection display

- On the Job Selection list display you can use an **R** line command against an entry to get the output recovered into the database from the secondary archive.

```
OMCS/V2 JOBS: TGSG -2   ASE.DATABASE 75% 14:35 MON 19941217
 COMMAND ===>                                                Scroll ===> CSR
                                                             Line 1     of 61
 S browse Y summar H hardcop P purge D delete U update T test A analyz R restore
 lc JOBNAME- JOBID- RUNON DAY STAR STOP ELAPS CPUSEC LINES- MXCC SYMPTOM--- SYST
    TGSGTST0 J06664 DEC15 THU 1743 1743  0:01    0.1  3,042 0004 ABS806     CPUA
 r  TGSGTST0 J09771 DEC16 FRI 1536 1536  0:01    0.1  3,042 0004 ABS806     CPUA
    TGSGTSTV J01417 DEC17 SAT 1151 1151  0:01    0.2    206 0004 ABU1006    CPUA
    TGSGTSTV J01419 DEC17 SAT 1151 1152  0:01    0.3    267 0004 ABU1006    CPUA
    TGSGTSTW J01418 DEC17 SAT 1151 1152  0:01    0.3    267 0004 ABU1006    CPUA
    TGSGTSTV J01421 DEC17 SAT 1152 1152  0:01    0.3    267 0004 ABU1006    CPUA
```

Recovering files via the AMF menu

- On the Selection Criteria display you can use the **AMF** command to get the AMF menu and from that you can proceed to request the recovery of one or more files.

```
OMCS/V2 Job Selection Criteria  ASE.DATABASE 75% ------------------------------
COMMAND ===> amf_                                      ? for field level help

primary commands ACF,AMF,JSF,IRM,JAL,DBI,DIR,SI,JPR,DDR,X
Enter Job Selection Criteria and press ENTER-----------------------------------
jobnames     ,==> TGSG
not before   ===> -2             yyyymmdd, mmdd, dd, tue, 25oct, -10, etc
```

The AMF primary menu will appear.

```
OMCS/AMF  Archive Management Facility
COMMAND ===>

             1  recover database files by name and time-last-modified
             2  display available backup dataset details
             C  configuration options
             R  reporting options

The following criteria will be used to limit what goes into selection lists:

filename   ===> .JOBDETL  Generic (trailing asterisk) notation
filetype   ===> MASTER    is supported in these two fields.

when-last-modified for database files -or- when-created for archive datasets
not before ===> -999           eg. MAR or 21MAR or 21MAR1995 or TUE:1200 or -3
not after  ===> -0             (refer to help for more info on these fields)


Archive index current detail:
Records: 885     Blocks: 4      Modified: 19941217-121948-MON Created: 19940101

Use the END key to return to the previous display
```

Option 1 lets you request recovery of a file whose filename and filetype you know.

All versions of the file that exist in the archive will be shown to you on a selection list which shows their date and time last modified. You can choose the one you want from that list.

```
OMCS/AMF ---- GLOBAL FILE RECOVERY SELECTION LIST -----------------------------
 COMMAND ===>                                                SCROLL ===> CSR
 SELECT FOR RETRIEVAL WITH "S", UNSELECT WITH "X"            LINE 1     OF 3
    S FILENAME FILETYPE LAST-MODIFIED------ EXPIRY-- ARCHIVED BLOCKS
                        YYYYMMDD-HHMMSS-DAY YYYYMMDD YYYYMMDD
      .JOBDETL MASTER   19941205-131423-WED 19951205 19941205     57
      .JOBDETL MASTER   19941208-152737-SAT 19951208 19941208     17
      .JOBDETL MASTER   19941216-152737-SUN 19951216 19941216     17
```

Option 2 lets you see what backup datasets exist

```
 OMCS/AMF ---- BACKUP DATASET SELECTION LIST ----------------------------------
 COMMAND ===>                                                SCROLL ===> CSR
 SELECT BACKUP DATASET WITH "S"                              LINE 1     OF 19
 lc BACKUP DATASET NAME                          #FILES CREATED---------- EXPIRES-
    TCPOP.IRM.FULLBKUP.G0008V00                      32 19941216-1311-SUN 19951216
    TCPOP.IRM.FULLBKUP.G0009V00                     122 19941217-1632-MON 19951217
```

Selecting an entry on the BACKUP DATASET SELECTION LIST then brings up a scrollable list of the database files that were backed up on that dataset.

```
OMCS/AMF ---- GLOBAL FILE RECOVERY SELECTION LIST -----------------------------
 COMMAND ===>                                                SCROLL ===> CSR
 SELECT FOR RETRIEVAL WITH "S", UNSELECT WITH "X"            LINE 1     OF 34
    S FILENAME FILETYPE LAST-MODIFIED------ EXPIRY-- ARCHIVED BLOCKS
                        YYYYMMDD-HHMMSS-DAY YYYYMMDD YYYYMMDD
 s    BNDLDATA $0000029 19941214-100945-WED 19950113 19950113      1
      BNDLDATA $0000030 19941214-121557-WED 19950113 19950113    368
      BNDLDATA $0000032 19941214-132934-WED 19950113 19950113    143
      BNDLDATA $0000033 19941214-133058-WED 19950113 19950113     18
      BNDLDATA $0000034 19941214-150922-WED 19950113 19950113    168
      BNDLDATA $0000035 19941215-112332-THU 19950114 19950113   1062
      BNDLDATA $0000037 19941215-145123-THU 19950114 19950113      4
      BNDLDATA $0000038 19941216-103009-FRI 19950115 19950113      4
      BNDLDATA $0000039 19941222-064506-THU 19950121 19950113   1198
      BNDLDATA $0000040 19941216-170340-FRI 19950115 19950113     63
      BNDLDATA $0000041 19941222-064521-THU 19950121 19950113    464
      BNDLDATA $0000050 19941220-155845-TUE 19950119 19950113     11
      BNDLDATA $0000051 19941220-163357-TUE 19950119 19950113    106
      BNDLDATA $0000052 19941220-170135-TUE 19950119 19950113     89
      BNDLDATA $0000053 19941220-170213-TUE 19950119 19950113    103
      BNDLDATA $0000054 19941222-160312-THU 19950121 19950113      1
      BNDLDATA $0000055 19941223-120519-FRI 19950122 19950113      5
      BNDLDATA $0000056 19941228-153801-WED 19950127 19950113      5
      BNDLDATA $0000057 19941228-162623-WED 19950127 19950113      5
```

Once you have selected one or more files for recovery from one or more datasets then AMF asks for confirmation before it submits the recovery job.

```
OMCS/V2 ARCHIVE MANAGEMENT - CHOOSE QUEUED OR IMMEDIATE RESTORE SCHEDULING
 COMMAND  ===> _

      Q    Use the queued method of restore scheduling
      I    Use the immediate method of restore scheduling

 FORCE   ===>      ( Y  to force Replacement of an existing file )

 Please ensure the following JOB statement is correct if option I is chosen

 ===> //PETERR JOB RESTORE,MSGCLASS=X,NOTIFY=PETER
 ===> //JOBLIB DD DISP=SHR,DSN=ASE.ASELIB
 ===> //*
 ===> //*


 Say if you want a user notified when the restore completes
 NOTIFY  ===> Y       ( Y OR N )
 LOGONID ===>         ( leave blank for current logonid )

 Type I or Q and press ENTER to schedule the restore -or-
 press END to cancel the restore request
```

JOB PROCESSING RULES

Introducing Job Processing Rules

OMCS typically captures the output of many different jobs, started tasks and TSO user sessions. An installation may want OMCS to process different jobs, or groups of jobs, in different ways. For example, the period of time that output is to be retained will vary between, say, a job that runs many times a day and one that runs only monthly. Job Processing Rules is the mechanism that OMCS provides to allow you to control aspects of how the OMCS writer deals with each job. OMCSTSSJ provides a primary command, JPR, that allows an authorised user to invoke the job processing rules maintenance application.

Each rule is a set of parameter values, or variables, associated with a particular jobname, or generic jobname root (like TPMD*, for example). When the writer first starts to capture a job it creates a set of effective variables for the job by merging values from three sources:

1. a set of defaults hard-coded within the writer program (in CSECT OMCSJPVA);
2. the job rule with a key of asterisk (*) (this is sometimes called the "global" rule);
3. the job rule whose key provides the best-fit match for the jobname.

The effective variable set is built up by using each non-blank value from source (2) or (3) to overlay the corresponding value from a prior source.

For example, the variable JPVAJDRP, which controls how long a job detail entry will be retained by OMCS, would be built up from the three sources as follows. (Assume that the jobname was OMCSBKUP and the best-fit rule was OMCS*).

source	value	merge action	effective value
OMCSJPVA	100	copy value	100
"*" rule	90	copy value	90
"OMCS*" rule	blank	do nothing	90

Hence the OMCS system variable JPVAJDRP would have a value of 90 when OMCS was processing job OMCSBKUP.

The merged variable set that is in effect during the processing of a job is also available to each ECL program that is executed in the writer environment. This means that user-written checking programs can inspect, and be controlled by, data held in Job Processing Rules. (See note on user variables further down).

Each database can have one set of Job Processing Rules. They are stored in a file called .JOBRULES MASTER. When a writer first opens the database it copies the rules to a file called .JOBRULES xxxxxxxx (where xxxxxxxx is the name of the writer). This temporary copy is then used until the writer closes the database. This means that if you make a change to job processing rules you need to stop and restart the writer to make the change effective.

Please refer to the section on Job Processing Rules in the General Operations Guide - OM88-1022, for a detailed description of each job processing variable and instructions for using the maintenance application.

Job processing rules - system variables

The following list of "system" variables gives you an idea of the sort of thing that you can control through job processing rules.

```
variable t default  description
JPVAACCT A BLANK    ACCOUNT STRING FOR MISCELLANEOUS SUBMITTED JOBS
JPVAARGE A BLANK    RUN GLOBAL ECP (Y/BLANK)
JPVAARRP A 100      ARCHIVE RETENTION PERIOD IN DAYS
JPVAARVC A 0        ARCHIVE VERSION COUNT
JPVABCLL A 100      BASIC CHECKING LOG LINE LIMIT
JPVADRQC A BLANK    DUMP RE-Q CLASS CODE OR BLANK
JPVADSDS A BLANK    DELETE SDS FILE AFTER CHECKING (Y/BLANK)
JPVAECFH A BLANK    EC FAILURE HARDCOPY OPTION (A/J/N/BLANK)
JPVAECKS A BLANK    EXECUTION CHECKING SUPPRESSION (Y/BLANK)
JPVAECPL A BLANK    LIST OF ECPS TO BE RUN
JPVAECPN A BLANK    USER ECL PROGRAM INVOKED BY BASIC CHECKING (=LAST)
JPVAECSS A Y        BASIC CHECKING TO CREATE STEP SUMMARY
JPVAHCCL A BLANK    DEFAULT HARDCOPY CLASS CODE (A-Z,0-9 OR BLANK)
JPVAHCEP A BLANK    DEFAULT HARDCOPY ECL PROGRAM NAME
JPVAHCLU A BLANK    DEFAULT HARDCOPY DESTID     (BLANK=>LOCAL)
JPVAHCPG A BLANK    DEFAULT HARDCOPY PROGRAM NAME
JPVAIRM  A BLANK    IRM SUPPORT (N TO SUPPRESS IRM SUPPORT)
JPVAJDRP A 100      JOB DETAIL RECORD RETENTION PERIOD IN DAYS
JPVAJDVC A 100      JOB DETAIL RECORD VERSION CONTROL       (100)
JPVAJSF  A BLANK    JSF SUPPORT (N TO SUPPRESS JSF SUPPORT)
JPVAJRQC A BLANK    JOB  RE-Q CLASS CODE OR BLANK
JPVAMFFD A 0        WAIT-BEFORE-FICHE DELAY IN DAYS
JPVAMFOP A BLANK    MICROFICHE OPTION (A OR BLANK)
JPVAMFSD A 0        SCRATCH-AFTER-FICHE DELAY IN DAYS
JPVAMPIG A BLANK    SYSTEM MSG PREFIXES TO BE IGNORED BY BASIC CHECKING
JPVAMXCC A 0        MAXIMUM COND CODE ALLOWED IN ANY STEP IN THE JOB
JPVANTFU A BLANK    USERIDS TO NOTIFY LIKE "TGSG=E "
JPVAPGSZ A 0        PAGESIZE FOR FORCED PAGINATION (0-32767 OR BLANK)
JPVASDCB A (0000-2300,BLANK) SDS CONSTRUCTION BASE HOUR
JPVASDCM A BLANK    SDS FILE CONSTRUCTION METHOD (C/BLANK)
JPVASDRP A 1        SDS FILE   RETENTION PERIOD IN DAYS
JPVASMR# A 0        SMF RECORD TYPE IF PSEUDO TYPE 6 REQD
JPVAUCLL A 100      USER CHKING LOG LIMIT
```

Job processing rules - user variables

The installation can extend job processing rules by creating their own variables. This provides a simple method of passing parameter or control information to user-written ECL programs that may execute in the writer environment.

(page intentionally unused)

Appendix A - Building an Install Tape

It is possible for SMP/E to receive from tape or DASD. If you were not supplied with OMCS install materials on tape, it is possible to create an OMCS product tape.

You must copy the datasets to a tape or cartridge to create a product tape. The following JCL sample shows how this can be done. Substitute your high level qualifier in the appropriate places. Remove references to fmids which you do not have. If any fmids are removed, you must ensure that the tape file numbers (specified by the L= parameters) are consecutive.

```
//R PROC V=______,           <-substitute the desired tape volume serial
//       U=3480              <-substitute the appropriate device type
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=&ID,DISP=SHR
//OU       DD DSN=&OD,DISP=(OLD,PASS),VOL=(,RETAIN,SER=&V),
//         UNIT=&U,LABEL=(&L,SL)
//         PEND
//S1       EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DISP=SHR,DSN=hlq.ASEB209.SMPMCS
//         DD DISP=SHR,DSN=hlq.ASEC209.SMPMCS
//         DD DISP=SHR,DSN=hlq.ASED209.SMPMCS
//         DD DISP=SHR,DSN=hlq.ASEE209.SMPMCS
//         DD DISP=SHR,DSN=hlq.ASEG209.SMPMCS
//SYSUT2   DD DSN=SMPMCS,DISP=(,KEEP),LABEL=(1,SL),
//            UNIT=3480,                 <-substitute the appropriate device type
//            VOL=(,RETAIN,SER=______)   <-substitute the desired tape volume serial
//ASEBF1   EXEC R,L=2,ID='hlq.ASEB209.F1',OD=ASEB209.F1
//ASEBF2   EXEC R,L=3,ID='hlq.ASEB209.F2',OD=ASEB209.F2
//ASEBF3   EXEC R,L=4,ID='hlq.ASEB209.F3',OD=ASEB209.F3
//ASEBF4   EXEC R,L=5,ID='hlq.ASEB209.F4',OD=ASEB209.F4
//ASEBF5   EXEC R,L=6,ID='hlq.ASEB209.F5',OD=ASEB209.F5
//ASEBF6   EXEC R,L=7,ID='hlq.ASEB209.F6',OD=ASEB209.F6
//ASEBF7   EXEC R,L=8,ID='hlq.ASEB209.F7',OD=ASEB209.F7
//ASECF1   EXEC R,L=9,ID='hlq.ASEC209.F1',OD=ASEC209.F1
//ASECF2   EXEC R,L=10,ID='hlq.ASEC209.F2',OD=ASEC209.F2
//ASECF3   EXEC R,L=11,ID='hlq.ASEC209.F3',OD=ASEC209.F3
//ASECF4   EXEC R,L=12,ID='hlq.ASEC209.F4',OD=ASEC209.F4
//ASECF5   EXEC R,L=13,ID='hlq.ASEC209.F5',OD=ASEC209.F5
//ASEDF1   EXEC R,L=14,ID='hlq.ASED209.F1',OD=ASED209.F1
//ASEDF2   EXEC R,L=15,ID='hlq.ASED209.F2',OD=ASED209.F2
//ASEDF3   EXEC R,L=16,ID='hlq.ASED209.F3',OD=ASED209.F3
//ASEDF4   EXEC R,L=17,ID='hlq.ASED209.F4',OD=ASED209.F4
//ASEEF1   EXEC R,L=18,ID='hlq.ASEE209.F1',OD=ASEE209.F1
//ASEEF2   EXEC R,L=19,ID='hlq.ASEE209.F2',OD=ASEE209.F2
//ASEEF3   EXEC R,L=20,ID='hlq.ASEE209.F3',OD=ASEE209.F3
//ASEGF1   EXEC R,L=21,ID='hlq.ASEG209.F1',OD=ASEG209.F1
//ASEGF2   EXEC R,L=22,ID='hlq.ASEG209.F2',OD=ASEG209.F2
//ASEGF3   EXEC R,L=23,ID='hlq.ASEG209.F3',OD=ASEG209.F3
//ASEGF4   EXEC R,L=24,ID='hlq.ASEG209.F4',OD=ASEG209.F4
```

```
//ASEGF5   EXEC R,L=25,ID='hlq.ASEG209.F5',OD=ASEG209.F5
```

(page intentionally unused)

Publication number: OM88-1020-6

This manual serves as a reference source for data processing staff who use and configure the OMCS/MVS product. Please use this form to comment on the publication in terms of its organisation or content with the understanding that ASE may use or distribute the information that you supply in any way that it believes appropriate without incurring an obligation to you.

If you require further copies of this publication please order them through your ASE representative. If you require assistance in the installation, configuration or use of the OMCS/MVS product please contact ASE for support.

Suggested aspects which might be addressed by your comments include:

Clarity Accuracy Completeness Organisation Examples Legibility

Please give your name and address if you require a reply:

name:

Address:

Occupation:

Comments: