z/OS Runtime Component

Release 8.1.3

November 2013

IKAN Solutions N.V.
Kardinaal Mercierplein 2
B-2800 Mechelen
BELGIUM

Table of Contents

CHAPTER 1
Introduction

1.1. Related Products

- MetaStore Manager (and the corresponding batch component MSBSTORE)
- MetaMap Manager (and the corresponding batch component MSBMAP)
- Generator Manager (and the corresponding batch component MSBGEN)

1.2. Terminology

MDL	MetaSuite Definition Language
MXL	MetaSuite Export Language
MGL	MetaSuite Generated Language
MRL	MetaSuite Run Language

CBL	COBOL source
COB	COBOL source
SQB	SQL COBOL source
ECO	Embedded SQL COBOL source
<Ins>	MetaSuite installation folder on the client side

1.3. System Requirements

CPU	Pentium Processor or higher
System RAM	Minimum of 96 MB
Hard disk space	Minimum 300 MB of free disk space for storage of MetaSuite software and .NET Framework.

Operating System	• Windows Vista • Windows Seven • Windows XP with Service Pack 3
Software	• Microsoft .NET Framework 2.0 (already included in Windows Vista) • File transfer program (e.g. PC Support) • MT9750 terminal emulator

1.4. Pre-installation Requirements

Before installing the runtime component, you must:

1. Install MetaSuite and select *IBM COBOL on z/OS* as Generator environment.

   For more detailed information, refer to the *Installation and Setup Guide*.

2. Create the MetaSuite Generator Dictionary for z/OS.

   For more detailed information, refer to the *Generator Manager User Guide*.

CHAPTER 2
About This Manual

This guide describes how to install the MetaSuite OS/390 and Z/OS runtime components. More specifically, it describes the installation of the following MetaSuite components:

- Base product
- MetaSuite Database Interfaces

The instructions for these components refer to additional information found in separate documents. Be sure to have those documents available during the installation.

2.1. Prerequisites

Product installers are expected to be familiar with their host operating systems and software installation processes.

2.2. Related Publications

The MetaSuite User and Reference Guides describe the different MetaSuite components and provide examples for using MetaSuite. Those guides should be available for reference during the installation and test procedures described here.

The following table gives an overview of the complete MetaSuite documentation set.

Release Information	Release Notes 8.1.3
Installation Guides	• BS2000/OSD Runtime Component • DOS/VSE Runtime Component • Fujitsu Windows Runtime Component • MicroFocus Windows Runtime Component • MicroFocus UNIX Runtime Component • OS/390 and Z/OS Runtime Component • OS/400 Runtime Component • VisualAge Windows Runtime Component • VisualAge UNIX Runtime Component • VMS Runtime Component
User Guides	• INI Manager User Guide • Installation and Setup Guide • Introduction Guide • MetaStore Manager User Guide • MetaMap Manager User Guide • Generator Manager User Guide

Technical Guides	• ADABAS File Access Guide • IDMS File Access Guide • IMS DLI File Access Guide • RDBMS File Access Guide • XML File Access Guide • Runtime Modules • User-defined Functions User Guide

In addition, you may want to refer to the related IBM manuals for your operating system, which explain how to code JCL and how to interpret IBM system messages and codes.

If you are unfamiliar with MetaSuite, the following technical description provides you with a brief overview.

The MetaSuite System	MetaSuite is designed for data retrieval, extraction, conversion and reporting. It includes a workstation-based graphical user interface and a mainframe runtime component.
MetaSuite Database Interfaces	MetaSuite can access data from a number of database management systems, using the same commands, program structure and retrieval techniques used for non-database files. Each database interface is available as an optional enhancement to the base product.
MetaMap Manager	MetaMap Manager is the MetaSuite tool used to define models. Such models are intuitively built by describing overall program specifications, input file definitions (data and process) and target file definitions (data and process).
MetaStore Manager	MetaStore Manager is a tool that provides metadata maintenance and documentation services.
Generator Manager	The Generator Manager is the system administration tool.All kinds of basic functionalities and customization possibilities are supported by this tool.

CHAPTER 3
Overview

Product installers are expected to be familiar with their host operating systems and with the software installation process on z/OS.

3.1. The Stages of Installation

Stage	Description
Stage 1 - Completing the pre-installation questionnaire	This questionnaire collects all the information needed during the installation.
Stage 2 - Building the runtime installation	Customize your installation based on the questionnaire. Allocate all the necessary libraries. Copy members of the MetaSuite install directory into the appropriate libraries.
Stage 3 - Compiling and linking the runtime system	Compile and link the MetaSuite runtime modules.

Note: MetaSuite Runtime operates under the restrictions imposed by data security systems such as RACF, Top Secret, or ACF2, and by database management systems to be accessed. When you install MetaSuite, be sure that the intended users have access to the appropriate MetaSuite data sets and libraries.

CHAPTER 4

Stage 1 - Completing the Pre-installation Questionnaire

The information in this questionnaire is required for the MetaSuite installation jobs, the MetaSuite procedures and the MetaSuite template code for JCL (MRL).

In most cases, the symbolic markers that appear in the left column and are bracketed in asterisks (for example, **SYSLIB**), are used in these installation text files. You may use any editor to substitute the symbolic markers with your values.

The following parameters should be specified:

4.1. JCL Substitution Parameters

Installation Parameter	Your Value	Description/Choices
ACC Job Accounting Parameters		Job accounting information for installation batch jobs. This can be a single accounting value or a list of values such as: (93,PRGMMNG,9301,87-0000)

Installation Parameter	Your Value	Description/Choices
JCLS Job Class		1-character JOB CLASS used as the default job class for all installation and user batch jobs. Specify a class that allows programs as large as 4096K to execute.
MCLS Message Class		1-character MESSAGE CLASS. This must be a legal print output class. If MetaSuite users will want to view JCL messages for the jobs that they submit, set this to the print output class used by the z/OS Spooling System.
MLVL Message Level		MESSAGE LEVEL specification; the default is (1,1). This value is the default message level for installation and user batch jobs.
PERMDA Permanent Disk Unit		DISK UNIT specification (the JCL UNIT= parameter) for all permanent data sets, catalogued or not. The default is SYSDA.
TEMPDA Temporary Disk Unit		DISK UNIT specification (the JCL UNIT= parameter) used for all temporary data sets. The default is SYSDA.
PCLS SYSOUT Class for Listings		1-character SYSOUT print class. This must be a legal print output class. If MetaSuite users will want to view output created by the jobs that they submit, set this to the print output class used by the z/OS Spooling System.
QUAL Data set Qualifier		MetaSuite High Level Data set Qualifier. This qualifier will be used to build default data set names for this installation.For example, `METASUIT.070101` **Note:** DO NOT terminate this qualifier with a period.

4.2. COBOL Compiler Defaults

Installation Parameter	Your Value	Description/Choices
COBOL Compiler Program	IGYCRCTL	NAME of the COBOL compiler program used to compile your MetaSuite application programs.
CBLLOAD COBOL Compiler Load Library		DATASET NAME of the Load library that contains the compiler resident and transient service routines.

Installation Parameter	Your Value	Description/Choices
CBLLIB COBOL Subroutine Library		DATASET NAME of the library that contains the COBOL resident and transient service routines. This library will be used as SYSLIB in the link step for application programs, and used in the STEPLIB for all MetaSuite job-steps.
CBLPGM		Name of the MetaSuite application program to compile.

4.3. Linkage Editor Defaults

Installation Parameter	Your Value	Description/Choices
Link Program	IEWL	NAME of the Linkage Editor program in use in your computer system.
Link-Editor Parameters	LET CALL LIST XREF	PARAMETER LIST to be used by the Linkage Editor when linking your MetaSuite application programs.

4.4. Sort Program Defaults

Installation Parameter	Your Value	Description/Choices
SRTLIB Sort Runtime Library		DATASET NAME of the library containing the SORT service routines used for sort processing by MetaSuite applications (sort runtime library). This library will be included in the STEPLIB in the run-step of MetaSuite application jobs.
SRTTRK Sort Work Space		NUMBER OF TRACKS to be allocated for each of the sort work files used by generated programs. The MetaSuite catalogued procedures set up SORTWK01, SORTWK02, SORTWK03, and SORTWK04 data sets for sort work space, and allocate this number of tracks to each of them. This should be as large as is practical to prevent users from running out of sort work space when running generated programs. A good starting point might be 10 tracks on a 3380.

4.5. Database Interface Values

ADABAS

Installation Parameter	Your Value	Description/Choices
ADALIB Subroutine Load Library		NAME of the ADABAS load module library that contains the database management service routines used to access the database.

DATACOM/DB

Installation Parameter	Your Value	Description/Choices
DCOMLIB Subroutine Load Library		NAME of the DATACOM load module library that contains the database management service routines used to access the database.

DBS/DB

Installation Parameter	Your Value	Description/Choices
DBSLIB Subroutine Load Library		NAME of the DBS load module library that contains the database management service routines used to access the database.

DB2 for z/OS

Installation Parameter	Your Value	Description/Choices
DB2SSID DB2 Subsystem Identifier (SSID)		DB2 SUBSYSTEM IDENTIFIER (SSID) in use at your site (typically DSN).
DB2REGN DB2 Region Size		TSO TERMINAL MONITOR PROGRAM (TMP) region size. Enter the number including 'K', i.e., 4096K. The TSO TMP is used to invoke DB2.
DB2LIB DB2 Load Library		DATASET NAME of the DB2 system load library, containing DB2 service routines and program modules.
DB2QUAL		DB2 table qualifier to be used as AUTHID for each referenced table.

Installation Parameter	Your Value	Description/Choices
DB2PREF DB2 User-id		TSO TERMINAL MONITOR PROGRAM (TMP) user-id.
DB2RMLIB DB2 DBRM Library		DATASET NAME of the library to store the DBRM created by user programs. If one does not exist, create it prior to using the MetaSuite DB2 Interface.
DB2CLIB* DB2 CLIST Library		DATASET NAME of the DB2 system CLIST library, containing the TSO CLISTs that run the DB2 programs.
Pre-compiler Parameters	HOST(COBOL) APOSTSQL STDSQL(NO) TIME(ISO) QUOTE	PARAMETER LIST to be used by the DB2 pre-compiler when processing generated MetaSuite COBOL programs. **Remark:** Programs generated with V6.03 or lower still need DATE(ISO) as well

IDMS

Installation Parameter	Your Value	Description/Choices
IDMSLIB Subroutine Load Library		NAME of the IDMS load library containing the IDMS database management routines (including IDMS, IDMSCANC and the "IDMSOPTI" module).
SYSCTL IDMS SYSCTL Name		MetaSuite programs can run in either local or central mode. If you want your programs to run in central mode but do not specify the name of an IDMSOPTI module, use this parameter to specify the dataset name of the SYSCTL file that provides central mode access.

Installation Parameter	Your Value	Description/Choices
IDMSOPTI Module Name		MODULE NAME. MetaSuite applications which access the IDMS database can either run in "local mode" (which means that the JCL for the database files must be provided in the MetaSuite job stream) or in "central mode" (which means that no database JCL is needed with the exception of SYSCTL, below). The way that your programs run may be controlled by the JCL or by a module called the IDMSOPTI module. This module is constructed by your IDMS system administrator, who determines its name. If you want to use the IDMSOPTI module to control the technique for accessing the database, then supply its name for this parameter.

IMS

Installation Parameter	Your Value	Description/Choices
RESLIB		DATASET NAME of the IMS resident module library containing the DLITCBL module.
DBDLIB		DATASET NAME of the library containing the IMS DBD that defines the databases that you will be accessing.
PSBLIB		DATASET NAME of the library containing the IMS PSB that you will be using in accessing the database.
DBIMSR Region Type		REGION TYPE for running under IMS. Valid values are DLI and BMP.
DBIMSP Additional Parameters		ADDITIONAL PARAMETERS needed in order to invoke program DFSRRC00 for running under IMS. You do not have to enter the first three parameters (region, type, MetaSuite program name and PSB name) as these are always present by default. When entering a value for this field, please always begin with a comma.

TOTAL

Installation Parameter	Your Value	Description/Choices
TOTLIB Subroutine Load Library		NAME of the TOTAL load module library **TOTLIB** that contains the database management service routines.

CHAPTER 5

Stage 2 - Build the Runtime Installation Libraries

This stage indicates how to continue after you have completed the questionnaire.

5.1. Installation Jobs

The following jobs require parameter substitution:

```
MSALLOC.JCL
mkrts813.JCL
MKCOB.JCL
MKCDB2.JCL
MKCIDMS.JCL
MKCDB2ID.JCL
MKCADA.JCL
```

Please refer to Appendix B - MetaSuite Library Contents (page 19) for a description of the function of the jobs.

5.2. Installation Procedures

The following procedures require parameter substitution:

```
MKCOB.PRC
MKCADA.PRC
MKCDB2.PRC
MKCIDMS.PRC
```

Please refer to Appendix B - MetaSuite Library Contents (page 19) for a description of the function of the procedures.

5.3. Generated MRL

When a MetaSuite Generator program is run, there is a template run script generated as well, which is stored in `x:\installdirectory\GENOS390\MRL\MXL-name.mrl`. This script needs parameter substitution.

Please refer to the *Generator Manuager User Guide* for more information on how the code tables for the template run scripts are built and how they can be customized.

5.4. Copy MetaSuite Procedures

Use a File Transfer Program (FTP ASCII CRLF Format) to copy the tailored MetaSuite procedures from the MetaSuite installation directory into the procedure library:

```
MKCOB.PRC
MKCDB2.PRC
MKCIDMS.PRC MKCADA.PRC
```

5.5. Copy MetaSuite Jobs

Use a File Transfer Program (FTP ASCII CRLF Format) to copy the tailored MetaSuite jobs from the MetaSuite installation directory into the MetaSuite job library:

```
MSALLOC.JCL
mkrts813.JCL
MKCOB.JCL
MKCADA.JCL
MKCDB2.JCL
MKCIDMS.JCL
MKCDB2ID.JCL
```

5.6. Copy MetaSuite Runtime Sources

Use a File Transfer Program (FTP ASCII CRLF Format) to copy the MetaSuite runtime source code from the MetaSuite installation directory into the MetaSuite source library:

```
MSRTS813.PDS
```

This is a flat file, no real PDS file, but it can be converted to a PDS file on mainframe.

5.7. Allocate MetaSuite Libraries

Submit MSALLOC.JCL to allocate all necessary MetaSuite libraries.

CHAPTER 6

Stage 3 - Compile and Link the Runtime System

This stage finalizes the installation:

This job compiles the MetaSuite runtime routines, places the resulting object modules into the MetaSuite Object Library and will link the Runtime modules into the MetaSuite Load Library.

6.1. Compiler Parameters

Before compiling the runtime modules, determine what compiler parameters to use for your site. You need to consult with your Systems Programmer to determine what these are for your data center. IKAN Solutions strongly recommends that you include the following compiler parameters:

Compiler Parameters	Description
DYNAM	Dynamic loading.
RES	Resident.
NOADV	We add a carriage control character.
OBJ	Generates object code.
FLAG(E)	List error messages only.
LIB	Processes COPY, BASIS, or REPLACE statements.
QUOTE	Literal delimiter
NOCMPR2	No compatibility with COBOL II Version 2einforced
NOFLAGMIG	No incompatibility flags with COBOL II Version 2 enforced

These compiler directives are already set in `mkrts813.JCL`

6.2. mkrts813 Job Steps

The runtime modules are provided in source format. See Appendix B - MetaSuite Library Contents (page 19) for an explanation of the functions of each of these modules.

Step	Programs Produced
MSCOB	In-stream procedure to compile the COBOL runtime modules using the IGYCRCTL compiler program. Review this step with your System Programmer. Some sites use the COB2UC procedure instead of the IGYCRCTL program.
UPDTE	Expand Source members in source library
XXX	The next steps are the compiles of the runtime modules. The step name corresponds to the last 6 characters in the module name.
MSLNK	Link runtime modules

Compiler Messages

COBOL compiler messages begin with IGY. Note that some runtime module compiles get a return code of 04, which is only a warning message and can be safely ignored.

Return Codes

The compile and bind steps should produce a return code of "0000" or "0004".

Step MSLNK should produce a return code value of "0004" and the link output listing should contain many "UNRESOLVED EXTERNAL REFERENCE" error messages.

The external references will be for MSxxx813 modules and IGZ modules. A return code of "0008" or higher for this step indicates a problem.

Region Size

The mkrts813 program specifies a region size of 2048K for selected steps. Ensure that the job card does not override it with a smaller region size.

Incompatibility

The compiler parameters for your MetaSuite-generated programs and the MetaSuite runtime modules must be compatible. A symptom of incompatibility can be a return code of 8 on your link, or a SOC1 abend in the run step.

Note: This is unlikely to happen, since the JCL for mkrts813 is tailored with the same COBOL compiler parameters that are used in the MetaSuite generated program. This may be a concern if you have tailored the in-line procedure supplied in mkrts813 or if you tailor the compiler directives in the template code for the generation of MetaSuite programs.

Appendix A - Estimating Track Sizes

You will need to estimate sizes for Target files, if the disposition is NEW (including work files). The sizes you assign to these files are important because if the allocated space is insufficient, the job will fail in the run step with "insufficient space" errors.

This appendix describes how to estimate track sizes.

A.1. Number of Tracks for Files

The number of tracks depends on:

- Number of records you expect to output
- Number of records per block

  Divide the block-size by the maximum record-size specified in the file definition.

A.2. Number of Blocks per Track

Consult with your systems staff to determine the number of blocks of this size that can fit on 1 track. Use the following formula:

```
#blocks = (#records / #records per block)
**TRK** = (#blocks / #blocks per track)
```

where **TRK** is the estimated number of tracks.

A.3. Tracks User for Workfiles

A work-file is required for every file that is read with a Sort procedure that includes the SORT or EXTRACT commands. The MetaSuite program generator calculates record and block sizes for the work-file during program generation.

Therefore, the estimation formula relies on the following information:

- Number of records

  Estimate the number of records to be written to this file (i.e., not excluded in the Sort procedure).

- Number of fields per record

  Count the number of fields referenced from the input file (the formula will account for the size of these fields).
- Track capacity of the disk device

  Refer to the following table:

Device	2314 or 2319	3330	3340	3350	3375	3380	3390
Tracks	2000	6000	4000	8000	14000	20000	20000

Use the following formula:

```
**TRK** = (#records * #fields) / track-capacity
```

where **TRK** is the estimated number of tracks.

A.4. Number of Tracks for Target Files

Use the following formula:

```
**TRK** = (#records * #fields) / track-capacity
```

where **TRK** is the estimated number of tracks.

Device	2314 or 2319	3330	3340	3350	3375	3380	3390
Tracks	1000	3000	2000	7000	7000	10000	12500

Other information needed for the estimate is:

- Expected number of records

  Estimate the total number of DETAIL and TOTAL records that will be output for the Target file.
- Number of fields in the longest record

  Count the number of fields on the longest non-accumulate DETAIL or TOTAL command for the Target file. Use the following formula:

  TRK = (#records * #fields)/track-capacity

  where **TRK** is the number of estimated tracks needed.

Appendix B - MetaSuite Library Contents

The following members are stored in the MetaSuite system libraries. The members are grouped in this Appendix according to the general function they serve. Each member name is followed by a brief description of its specific function.

Note: The scripts used in the following sections are for documentary purposes only.

B.1. Job Library

This library consists of installation and compilation jobs.
Each job must be tailored by the system installer during the installation of MetaSuite.

Member Name	Description
MSALLOC	Allocate libraries
mkrts813	Compile, link runtime modules for COBOL 370 or COBOL 390
MKCOB	Compile, link generated program
MKCADA	Compile, link generated Adabas program
MKCDB2	Compile, link generated DB2 program
MKCIDMS	Compile, link generated IDMS program
MKCDB2ID	Compile, link generated DB2/IDMS program

Note: The database JCL will be available only if the respective database interfaces have been purchased and installed.

MSALLOC Job

```
//MSALLOC  JOB **ACC**,
//         MSGCLASS=**MCLS**,
//         MSGLEVEL=**MLVL**,
//         REGION=4096K,
//         CLASS=**JCLS**
//* ALLOCATE A PROCEDURE LIBRARY FOR METASUITE PROCEDURES
//ALLOC0   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//PROCLIB  DD  DSN=**QUAL**.PROCLIB,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(TRK,(5,2,1)),
//             DCB=(RECFM=FB,DSORG=PO,LRECL=80,BLKSIZE=2960)
//* ALLOCATE A SOURCE LIBRARY FOR METASUITE RUN-TIME
//ALLOC1   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//SRCLIB   DD  DSN=**QUAL**.SRCLIB,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(TRK,(5,2,1)),
//             DCB=(RECFM=FB,DSORG=PO,LRECL=80,BLKSIZE=2960)
//* ALLOCATE AN OBJECT LIBRARY FOR METASUIT RUN-TIME
//ALLOC2   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//OBJLIB   DD  DSN=**QUAL**.OBJLIB,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(TRK,(25,5,5)),
//             DCB=(RECFM=FB,DSORG=PO,LRECL=80,BLKSIZE=2960)
//* ALLOCATE A LOAD LIBRARY FOR METASUITE RUN-TIME
//ALLOC3   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//LOADLIB  DD  DSN=**QUAL**.LOADLIB,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(TRK,(400,,20)),
//             DCB=(RECFM=U,DSORG=PO)
//* ALLOCATE A USER SOURCE LIBRARY FOR METASUITE GENERATED PROGRAMS
//ALLOC4   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//SRCLIB   DD  DSN=**QUAL**.MGLLIB,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(CYL,(10,10,5)),
//             DCB=(RECFM=FB,DSORG=PO,LRECL=80,BLKSIZE=2960)
//* ALLOCATE A USER LOAD LIBRARY FOR METASUITE GENERATED PROGRAMS
//ALLOC5   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//LOADLIB  DD  DSN=**QUAL**.USER.LOADLIB,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(CYL,(50,,60)),
//             DCB=(RECFM=U,DSORG=PO)
//* ALLOCATE A PARM LIBRARY FOR RUN-TIME VARIABLES
//ALLOC6   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//PARMLIB  DD  DSN=**QUAL**.PARMLIB,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(CYL,(10,10,5)),
//             DCB=(RECFM=FB,DSORG=PO,LRECL=255,BLKSIZE=2550)
//* ALLOCATE EMPLOYEE TEST DATA FILE
//ALLOC7   EXEC PGM=IEFBR14,COND=EVEN
```

```
//SYSPRINT DD  SYSOUT=**PCLS**
//EMPL     DD  DSN=**QUAL**.EMPLOYEE,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(TRK,(5,2,1)),
//             DCB=(RECFM=FB,DSORG=PS,LRECL=204,BLKSIZE=2040)
//* ALLOCATE PAYROLL TEST DATA FILE
//ALLOC8   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//PAYROLL  DD  DSN=**QUAL**.PAYROLL,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(TRK,(5,2,1)),
//             DCB=(RECFM=FB,DSORG=PS,LRECL=157,BLKSIZE=1570)
//* ALLOCATE A JOB LIBRARY FOR METASUITE
//ALLOC9   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//JCLLIB   DD  DSN=**QUAL**.MRLLIB,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(CYL,(10,10,5)),
//             DCB=(RECFM=FB,DSORG=PO,LRECL=80,BLKSIZE=2960)
//ALLOC10   EXEC PGM=IEFBR14,COND=EVEN
//SYSPRINT DD  SYSOUT=**PCLS**
//DEPARTM  DD  DSN=**QUAL**.DEPARTM,
//             DISP=(NEW,CATLG),UNIT=**PERMDA**,
//             SPACE=(TRK,(5,2,1)),
//             DCB=(RECFM=FB,DSORG=PS,LRECL=31,BLKSIZE=310)
```

mkrts813 Job

```
//MSRTS    JOB **ACC**,
//         MSGCLASS=**MCLS**,
//         MSGLEVEL=**MLVL**,
//         REGION=4096K,
//         CLASS=**JCLS**
//*
//MSCOB PROC MEMBER=
//COB      EXEC PGM=IGYCRCTL,REGION=2048K,COND=(5,LT),
// PARM=('OBJ,FLAG(E),QUOTE,LIB,NOADV,DYNAM,RES,NOCMPR2,NOFLAGMIG')
//STEPLIB   DD DSN=**CBLLOAD**,DISP=SHR
//SYSLIN    DD DSN=**QUAL**.OBJLIB(&MEMBER),DISP=SHR
//SYSIN     DD DSN=**QUAL**.SRCLIB(&MEMBER),DISP=SHR
//SYSLIB    DD DSN=**QUAL**.SRCLIB,DISP=SHR
//SYSTERM   DD SYSOUT=**PCLS**
//SYSPRINT  DD SYSOUT=**PCLS**
//SYSUT1    DD UNIT=**TEMPDA**,SPACE=(CYL,(1,1)),DISP=(NEW,PASS)
//SYSUT2    DD UNIT=**TEMPDA**,SPACE=(CYL,(1,1)),DISP=(NEW,PASS)
//SYSUT3    DD UNIT=**TEMPDA**,SPACE=(CYL,(1,1)),DISP=(NEW,PASS)
//SYSUT4    DD UNIT=**TEMPDA**,SPACE=(CYL,(1,1)),DISP=(NEW,PASS)
//SYSUT5    DD UNIT=**TEMPDA**,SPACE=(CYL,(1,1)),DISP=(NEW,PASS)
//SYSUT6    DD UNIT=**TEMPDA**,SPACE=(CYL,(1,1)),DISP=(NEW,PASS)
//SYSUT7    DD UNIT=**TEMPDA**,SPACE=(CYL,(1,1)),DISP=(NEW,PASS)
// PEND
//*
//UPDTE   EXEC PGM=IEBUPDTE,REGION=512K,PARM=NEW
//SYSPRINT DD SYSOUT=**PCLS**
//SYSIN    DD DSN=**QUAL**.SRCLIB(MSRTC390),DISP=SHR
//SYSUT1   DD DUMMY
```

```
//SYSUT2   DD DSN=**QUAL**.SRCLIB,DISP=SHR
//* COMPILE RUNTIME
//ASC      EXEC MSCOB,MEMBER=MSASC813
//BIN      EXEC MSCOB,MEMBER=MSBIN813
//CAL      EXEC MSCOB,MEMBER=MSCAL813
//CCX      EXEC MSCOB,MEMBER=MSCCX813
//CIX      EXEC MSCOB,MEMBER=MSCIX813
//DBX      EXEC MSCOB,MEMBER=MSDBX813
//DCX      EXEC MSCOB,MEMBER=MSDCX813
//DMX      EXEC MSCOB,MEMBER=MSDMX813
//DTX      EXEC MSCOB,MEMBER=MSDTX813
//EBC      EXEC MSCOB,MEMBER=MSEBC813
//ENX      EXEC MSCOB,MEMBER=MSENX813
//EOJ      EXEC MSCOB,MEMBER=MSEOJ813
//HEX      EXEC MSCOB,MEMBER=MSHEX813
//HXT      EXEC MSCOB,MEMBER=MSHXT813
//HXX      EXEC MSCOB,MEMBER=MSHXX813
//IMS      EXEC MSCOB,MEMBER=MSIMS813
//INI      EXEC MSCOB,MEMBER=MSINI813
//LID      EXEC MSCOB,MEMBER=MSLID813
//PFX      EXEC MSCOB,MEMBER=MSPFX813
//RNX      EXEC MSCOB,MEMBER=MSRNX813
//RST      EXEC MSCOB,MEMBER=MSRST813
//SYX      EXEC MSCOB,MEMBER=MSSYX813
//SZX      EXEC MSCOB,MEMBER=MSSZX813
//TDX      EXEC MSCOB,MEMBER=MSTDX813
//TMX      EXEC MSCOB,MEMBER=MSTMX813
//TRM      EXEC MSCOB,MEMBER=MSTRM813
//HFX      EXEC MSCOB,MEMBER=MSHFX813
//LSX      EXEC MSCOB,MEMBER=MSLSX813
//SPX      EXEC MSCOB,MEMBER=MSSPX813
//* LINK RUNTIME
//MSLNK   EXEC PGM=IEWL,REGION=2048K
//SYSLIN   DD *
   ENTRY MSASC813
   INCLUDE OBJLIB(MSASC813)
   NAME MSASC813(R)
   ENTRY MSBIN813
   INCLUDE OBJLIB(MSBIN813)
   NAME MSBIN813(R)
   ENTRY MSCAL813
   INCLUDE OBJLIB(MSCAL813)
   NAME MSCAL813(R)
   ENTRY MSCCX813
   INCLUDE OBJLIB(MSCCX813)
   NAME MSCCX813(R)
   ENTRY MSCIX813
   INCLUDE OBJLIB(MSCIX813)
   NAME MSCIX813(R)
   ENTRY MSDBX813
   INCLUDE OBJLIB(MSDBX813)
   NAME MSDBX813(R)
   ENTRY MSDCX813
   INCLUDE OBJLIB(MSDCX813)
   NAME MSDCX813(R)
   ENTRY MSDMX813
   INCLUDE OBJLIB(MSDMX813)
   NAME MSDMX813(R)
   ENTRY MSDTX813
```

```
INCLUDE OBJLIB(MSDTX813)
NAME MSDTX813(R)
ENTRY MSEBC813
INCLUDE OBJLIB(MSEBC813)
NAME MSEBC813(R)
ENTRY MSENX813
INCLUDE OBJLIB(MSENX813)
NAME MSENX813(R)
ENTRY MSEOJ813
INCLUDE OBJLIB(MSEOJ813)
NAME MSEOJ813(R)
ENTRY MSHEX813
INCLUDE OBJLIB(MSHEX813)
NAME MSHEX813(R)
ENTRY MSHXT813
INCLUDE OBJLIB(MSHXT813)
NAME MSHXT813(R)
ENTRY MSHXX813
INCLUDE OBJLIB(MSHXX813)
NAME MSHXX813(R)
ENTRY MSIMS813
INCLUDE OBJLIB(MSIMS813)
NAME MSIMS813(R)
ENTRY MSINI813
INCLUDE OBJLIB(MSINI813)
NAME MSINI813(R)
ENTRY MSLID813
INCLUDE OBJLIB(MSLID813)
NAME MSLID813(R)
ENTRY MSPFX813
INCLUDE OBJLIB(MSPFX813)
NAME MSPFX813(R)
ENTRY MSRNX813
INCLUDE OBJLIB(MSRNX813)
NAME MSRNX813(R)
ENTRY MSRST813
INCLUDE OBJLIB(MSRST813)
NAME MSRST813(R)
ENTRY MSSYX813
INCLUDE OBJLIB(MSSYX813)
NAME MSSYX813(R)
ENTRY MSSZX813
INCLUDE OBJLIB(MSSZX813)
NAME MSSZX813(R)
ENTRY MSTDX813
INCLUDE OBJLIB(MSTDX813)
NAME MSTDX813(R)
ENTRY MSTMX813
INCLUDE OBJLIB(MSTMX813)
NAME MSTMX813(R)
ENTRY MSTRM813
INCLUDE OBJLIB(MSTRM813)
NAME MSTRM813(R)
ENTRY MSHFX813
INCLUDE OBJLIB(MSHFX813)
NAME MSHFX813(R)
ENTRY MSLSX813
INCLUDE OBJLIB(MSLSX813)
NAME MSLSX813(R)
```

```
   ENTRY MSSPX813
   INCLUDE OBJLIB(MSSPX813)
   NAME MSSPX813(R)
//SYSLMOD   DD DSN=**QUAL**.LOADLIB,DISP=SHR
//OBJLIB    DD DSN=**QUAL**.OBJLIB,DISP=SHR
//SYSLIB    DD DSN=**CBLLIB**,DISP=SHR
//SYSUT1    DD UNIT=**TEMPDA**,SPACE=(CYL,(1,1)),
//             DISP=(NEW,PASS)
//SYSPRINT  DD SYSOUT=**PCLS**
```

MKCOB Job

```
//**USERID**C JOB **ACC**,
//         MSGCLASS=**MCLS**,
//         MSGLEVEL=**MLVL**,
//         REGION=4096K,
//         CLASS=**JCLS**,
//         NOTIFY=**USERID**
// JCLLIB  ORDER=(**QUAL**.PROCLIB)
//MKCOB    EXEC MKCOB,
//             SRCLIB='**QUAL**.MGLLIB',
//             MEMBER='**CBLPGM**',
//             PRT='**PCLS**',
//             CPRT='SYSOUT=**PCLS**',
//             ULIB='**QUAL**.USER.LOADLIB'
```

MKCADA Job

```
//MKCADA   JOB **ACC**,
//         MSGCLASS=**MCLS**,
//         MSGLEVEL=**MLVL**,
//         REGION=4096K,
//         CLASS=**JCLS**
// JCLLIB  ORDER=(**QUAL**.PROCLIB)
//MKCADA   EXEC MKCADA,
//             SRCLIB='**QUAL**.USER.SRCLIB',
//             MEMBER='**CBLPGM**',
//             PRT='**PCLS**',
//             CPRT='SYSOUT=**PCLS**',
//             ULIB='**QUAL**.USER.LOADLIB'
```

MKCDB2 Job

```
//MKCDB2  JOB **ACC**,
//         MSGCLASS=**MCLS**,
//         MSGLEVEL=**MLVL**,
//         REGION=4096K,
//         CLASS=**JCLS**
// JCLLIB  ORDER=(**QUAL**.PROCLIB)
//CDB2     EXEC MKCDB2,
//             RLIB='**QUAL**.USER.LOADLIB',
```

```
//             SRCLIB='**QUAL**.USER.SRCLIB',
//             MEMBER='**CBLPGM**',
//             PRT='**PCLS**',
//             CPRT='SYSOUT=**PCLS**',
//             ULIB='**QUAL**.USER.LOADLIB'
//LKED.SYSIN  DD *
 INCLUDE DB2LIB(DSNELI)
 NAME  **CBLPGM**(R)
//*
```

MKCIDMS Job

```
//MKCIDMS  JOB **ACC**,
//         MSGCLASS=**MCLS**,
//         MSGLEVEL=**MLVL**,
//         REGION=4096K,
//         CLASS=**JCLS**
// JCLLIB  ORDER=(**QUAL**.PROCLIB)
//CIDMS    EXEC MKCIDMS,
//             RLIB='**QUAL**.USER.LOADLIB',
//             SRCLIB='**QUAL**.USER.SRCLIB',
//             MEMBER='**CBLPGM**',
//             PRT='**PCLS**',
//             CPRT='SYSOUT=**PCLS**',
//             ULIB='**QUAL**.USER.LOADLIB'
//LKED.SYSIN  DD *
  INCLUDE IDMSLIB(IDMS,IDMSCANC)
  NAME **CBLPGM**(R)
//*
```

MKCDB2ID Job

```
//MKCDB2ID JOB **ACC**,
//         MSGCLASS=**MCLS**,
//         MSGLEVEL=**MLVL**,
//         REGION=4096K,
//         CLASS=**JCLS**
// JCLLIB  ORDER=(**QUAL**.PROCLIB)
//CDB2     EXEC MKCDB2,
//             RLIB='**QUAL**.USER.LOADLIB',
//             SRCLIB='**QUAL**.USER.SRCLIB',
//             MEMBER='**CBLPGM**',
//             PRT='**PCLS**',
//             CPRT='SYSOUT=**PCLS**',
//             ULIB='**QUAL**.USER.LOADLIB'
//LKED.SYSIN  DD *
 INCLUDE DB2LIB(DSNELI)
 INCLUDE IDMSLIB(IDMS,IDMSCANC)
 NAME  **CBLPGM**(R)
//LKED.IDMSLIB  DD DSN='**IDMSLIB**',DISP=SHR
//*
```

B.2. Procedures Library

This library consists of prototype JCL procedures.

Member Name	Description
MKCOB.PRC	Make a COBOL executable
MKCADA.PRC	Make a COBOL executable with ADABAS/C
MKCDB2.PRC	Make a COBOL executable with DB2 for z/OS
MKCIDMS.PRC	Make a COBOL executable with IDMS

Note: The database procedures will be available only if the respective database interfaces have been purchased and installed.

MKCOB Procedure

```
//MKCOB PROC SRCLIB='**QUAL**.USER.SRCLIB',
//      ULIB='**QUAL**.USER.LOADLIB',
//      CLIB='**CBLLOAD**',
//      LLIB='**CBLLIB**',
//      MEMBER=MEMBER,
//      TEMPDA=**TEMPDA**,
//       PRT='**PCLS**',
//      COMPGM=IGYCRCTL,
//      CPARM='OBJ,FLAG(E),QUOTE,LIB,NOADV,DYNAM,RES',
//        CPRT='SYSOUT=**PCLS**',
//      LNKPGM=IEWL,
//      LPARM='LIST,XREF,LET,CALL',
//      RLIB='**QUAL**.LOADLIB'
//*
//*  PARAMETER   USAGE
//*  ---------   ------------------------------------------
//*  ULIB         USER LOAD LIBRARY FOR GENERATED PROGRAMS
//*  CLIB        COBOL COMPILER LIBRARY
//*  LLIB        COBOL SERVICE ROUTINE LIBRARY
//*  MEMBER      METASUITE GENERATED PROGRAM
//*  TEMPDA       DISK TYPE FOR GENERATED PROGRAM AND WORK FILES
//*  PRT         SYSOUT PRINT CLASS
//*  COMPGM      COBOL COMPILER PROGRAM NAME
//*  CPARM       COBOL COMPILER PARAMETERS
//*  CPRT         COBOL LISTING - CODE CPRT='SYSOUT=A' TO SEE IT
//*  LNKGPM      LINKAGE EDITOR PROGRAM NAME
//*  LPARM       LINKAGE EDITOR PARAMETERS
//*  RLIB          RUNTIME LIBRARY FOR THE METASUITE RUNTIME MODULES
//*
//*********************************************************
//COB    EXEC PGM=&COMPGM,
//            PARM='RES,NOSEQ,&CPARM'
//STEPLIB  DD DSN=&CLIB,DISP=SHR
```

```
//SYSTERM  DD SYSOUT=&PRT
//SYSPRINT DD &CPRT
//SYSUT1   DD DSN=&&SYSUT1,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT2   DD DSN=&&SYSUT2,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT3   DD DSN=&&SYSUT3,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT4   DD DSN=&&SYSUT4,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT5   DD DSN=&&SYSUT5,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT6   DD DSN=&&SYSUT6,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT7   DD DSN=&&SYSUT7,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSLIN   DD DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=&TEMPDA,
//            SPACE=(80,(500,100))
//SYSIN    DD DSN=&SRCLIB(&MEMBER),DISP=(SHR)
//LKED   EXEC PGM=&LNKPGM,COND=(5,LT,COB),REGION=2048K,
//            PARM='&LPARM'
//SYSLIN  DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//SYSLMOD  DD DSN=&ULIB(&MEMBER),DISP=SHR
//SYSLIB   DD DSN=&LLIB,DISP=SHR
//         DD DSN=&RLIB,DISP=SHR
//         DD DSN=&ULIB,DISP=SHR
//SYSUT1   DD UNIT=&TEMPDA,SPACE=(1024,(50,20))
//SYSPRINT DD SYSOUT=&PRT
```

MKCADA Procedure

```
//MKCADA  PROC SRCLIB='**QUAL**.USER.SRCLIB',
//            ULIB='**QUAL**.USER.LOADLIB',
//          CLIB='**CBLLOAD**',
//          LLIB='**CBLLIB**',
//           MEMBER='CBLPGM',
//            TEMPDA=**TEMPDA**,
//            PRT='**PCLS**',
//          COMPGM=IGYCRCTL,
//             CPARM='OBJ,FLAG(E),QUOTE,LIB,NOADV,XREF,DYNAM,RES',
//            CPRT='SYSOUT=**PCLS**',
//          LNKPGM=IEWL,
//          LPARM='LIST,XREF,LET,CALL',
//          RLIB='**QUAL**.LOADLIB'
//*
//*  PARAMETER   USAGE
//*  ---------   ------------------------------------------
//*  ULIB         USER LOAD LIBRARY FOR GENERATED PROGRAMS
//*  CLIB        COBOL COMPILER LIBRARY
//*  LLIB        COBOL SERVICE ROUTINE LIBRARY
//*  MEMBER      METASUITE GENERATED PROGRAM
//*  TEMPDA       DISK TYPE FOR GENERATED PROGRAM AND WORK FILES
//*  PRT         SYSOUT PRINT CLASS
//*  COMPGM       COBOL COMPILER PROGRAM NAME
//*  CPARM        COBOL COMPILER PARAMETERS
//*  CPRT          COBOL LISTING - CODE CPRT='SYSOUT=A' TO SEE IT
//*  LNKGPM       LINKAGE EDITOR PROGRAM NAME
//*  LPARM        LINKAGE EDITOR PARAMETERS
//*  RLIB           RUNTIME LIBRARY FOR THE METASUITE RUNTIME MODULES
//*
//*************************************************************
//COB    EXEC PGM=&COMPGM,
//            PARM='RES,NOSEQ,&CPARM'
```

```
//STEPLIB  DD DSN=&CLIB,DISP=SHR
//SYSTERM  DD SYSOUT=&PRT
//SYSPRINT DD &CPRT
//SYSUT1   DD DSN=&&SYSUT1,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT2   DD DSN=&&SYSUT2,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT3   DD DSN=&&SYSUT3,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT4   DD DSN=&&SYSUT4,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT5   DD DSN=&&SYSUT5,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT6   DD DSN=&&SYSUT6,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT7   DD DSN=&&SYSUT7,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSLIN   DD DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=&TEMPDA,
//            SPACE=(80,(500,100))
//SYSIN    DD DSN=&SRCLIB(&MEMBER),DISP=(SHR)
//LKED   EXEC PGM=&LNKPGM,COND=(5,LT,COB),REGION=2048K,
//            PARM='&LPARM'
//SYSLIN  DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//SYSLMOD  DD DSN=&ULIB(&MEMBER),DISP=SHR
//SYSLIB   DD DSN=&LLIB,DISP=SHR
//         DD DSN=&RLIB,DISP=SHR
//         DD DSN=&ULIB,DISP=SHR
//ADALIB   DD DSN=&ADALIB,DISP=SHR
//SYSUT1   DD UNIT=&TEMPDA,SPACE=(1024,(50,20))
//SYSPRINT DD SYSOUT=&PRT
```

MKCDB2 Procedure

```
//MKCDB2   PROC RLIB='**QUAL**.LOADLIB',
//       SRCLIB='**QUAL**.SRCLIB',
//       ULIB='**QUAL*.USER.LOADLIB',
//       CLIB='**CBLLOAD**',
//       LLIB='**CBLLIB**',
//       DB2LIB='**DB2LIB**',
//       DB2RMLIB='**DB2RMLIB**',
//       MEMBER=MEMBER,
//       TEMPDA=**TEMPDA**,
//        PRT='**PCLS**',
// PCPARM='HOST(COBOL),QUOTE,APOSTSQL,ST
SQL(NO),DATE(ISO),TIME(ISO)',
//       PCBLKS=500,
//       COMPGM=IGYCRCTL,
//        CPARM='OBJ,FLAG(E),QUOTE,LIB,NOADV,DYNAM,RES',
//          CPRT='SYSOUT=**PCLS**',
//        LNKPGM=IEWL,
//        LPARM='LIST,XREF,LET,CALL'
//*
//* DB2 PRECOMPILE, COMPILE, AND LINK
//*
//* PARAMETER   USAGE
//* ---------   ------------------------------------------
//* RLIB        METASUITE SYSTEM LOAD LIBRARY
//* ULIB       USER LOAD LIBRARY FOR GENERATED PROGRAMS
//*             CODE ULIB='&&TEMPLIB' FOR AD-HOC PROGRAM REQUESTS
//* CLIB       COBOL COMPILER LIBRARY
//* LLIB       COBOL LINK-EDIT SYSLIB
//* DB2LIB     DB2 SYSTEM LOAD LIBRARY
//* DB2LIB2    OPTIONAL SECOND DB2 SYSTEM LOAD LIBRARY
```

```
//* DBRMLIB    DB2 SYSTEM DBRM LIBRARY
//* MEMBER     GENERATED PROGRAM
//* TEMPDA      DISK TYPE FOR GENERATED PROGRAM AND WORK FILES
//* PRT        SYSOUT PRINT CLASS
//* PCPARM     DB2 PRECOMPILER PARAMETERS
//* PCBLKS     DB2 PRECOMPILER WORK FILE BLOCKS
//* COMPGM     COBOL COMPILER PROGRAM NAME
//* CPARM      COBOL COMPILER PARAMETERS
//* CPRT        COBOL LISTING - CODE CPRT='SYSOUT=A' TO SEE IT
//* LNKGPM     LINKAGE EDITOR PROGRAM NAME
//* LPARM      LINKAGE EDITOR PARAMETERS
//*
//*  ADDITIONAL  REQUIRED DATASETS
//*
//*  DATASET     USAGE
//*  ---------   -----------------------------------------
//*
//*  LKED.SYSIN  LINKAGE EDITOR CONTROL COMMAND INPUT FILE.  THIS
//*              MUST CONTAIN THE FOLLOWING STATEMENTS:
//*
//*               INCLUDE DB2LIB(DSNELI)
//*                NAME **CBLPGM**(R)
//*
//PRECMP EXEC PGM=DSNHPC,
//            PARM='&PCPARM'
//DBRMLIB  DD DSN=&DB2RMLIB(&MEMBER),DISP=SHR
//STEPLIB  DD DSN=&DB2LIB,DISP=SHR
//SYSCIN   DD DSN=&&DSNHOUT,DISP=(MOD,PASS),UNIT=&TEMPDA,
//            SPACE=(800,(500,500))
//SYSPRINT DD SYSOUT=&PRT
//SYSLIB   DD DUMMY
//SYSUT1   DD DSN=&&SYSUT1,UNIT=&TEMPDA,SPACE=(800,(&PCBLKS,&PCBLKS))
//SYSUT2   DD DSN=&&SYSUT2,UNIT=&TEMPDA,SPACE=(800,(&PCBLKS,&PCBLKS))
//SYSTERM  DD SYSOUT=&PRT
//SYSIN    DD DSN=&SRCLIB(&MEMBER),DISP=(SHR)
//COB      EXEC PGM=&COMPGM,COND=(5,LT,PRECMP),
//            PARM='RES,NOSEQ,&CPARM'
//STEPLIB  DD DSN=&CLIB,DISP=SHR
//SYSTERM  DD SYSOUT=&PRT
//SYSPRINT DD &CPRT
//SYSUT1   DD DSN=&&SYSUT1,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT2   DD DSN=&&SYSUT2,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT3   DD DSN=&&SYSUT3,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT4   DD DSN=&&SYSUT4,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT5   DD DSN=&&SYSUT5,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT6   DD DSN=&&SYSUT6,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT7   DD DSN=&&SYSUT7,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSLIN   DD DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=&TEMPDA,
//            SPACE=(80,(500,100))
//SYSIN    DD DSN=&&DSNHOUT,UNIT=&TEMPDA,DISP=(OLD,DELETE)
//LKED   EXEC PGM=&LNKPGM,COND=(5,LT,COB),REGION=2048K,
//            PARM='&LPARM'
//SYSLIN   DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD DDNAME=SYSIN
//SYSLMOD  DD DSN=&ULIB(&MEMBER),DISP=SHR
//SYSLIB   DD DSN=&LLIB,DISP=SHR
//         DD DSN=&DB2LIB,DISP=SHR
//         DD DSN=&RLIB,DISP=SHR
//DB2LIB   DD DSN=&DB2LIB,DISP=SHR
```

```
//SYSUT1   DD UNIT=&TEMPDA,SPACE=(1024,(50,20))
//SYSPRINT DD SYSOUT=&PRT
```

MKCIDMS Procedure

```
//MKCIDMS PROC SRCLIB='**QUAL**.SOURCE',
//             ULIB='**QUAL**.USER.LOAD',
//             CLIB='**CBLLOAD**',
//             LLIB='**CBLLIB**',
//             IDMSLIB='**IDMSLIB**',
//             MEMBER=MEMBER,
//             TEMPDA=**TEMPDA**,
//             PRT='**PCLS**',
//              COMPGM=IGYCRCTL,
//              CPARM='OBJ,FLAG(E),QUOTE,LIB,NOADV,DYNAM,RES',
//             CPRT='SYSOUT=**PCLS**',
//              LNKPGM=IEWL,
//              LPARM='LIST,XREF,LET,CALL',
//             RLIB='**QUAL**.LOADLIB'
//*
//*  PARAMETER   USAGE
//*  ---------   ----------------------------------------
//*  ULIB         USER LOAD LIBRARY FOR GENERATED PROGRAMS
//*  CLIB        COBOL COMPILER LIBRARY
//*  LLIB        COBOL LINK-EDIT SYSLIB
//*  IDMSLIB     IDMS SYSTEM LOAD LIBRARY
//*  MEMBER      GENERATED PROGRAM
//*  TEMPDA       DISK TYPE FOR GENERATED PROGRAM AND WORK FILES
//*  SPBLKS       NUMBER OF BLOCKS FOR GENERATOR SPOOL FILES
//*  PRT          SYSOUT PRINT CLASS
//*  COMPGM       COBOL COMPILER PROGRAM NAME
//*  CPARM        COBOL COMPILER PARAMETERS
//*  CPRT          COBOL LISTING - CODE CPRT='SYSOUT=A' TO SEE IT
//*  LNKGPM       LINKAGE EDITOR PROGRAM NAME
//*  LPARM        LINKAGE EDITOR PARAMETERS
//*  RLIB          RUNTIME LIBRARY FOR METASUITE RUNTIME MODULES
//*
//*  ADDITIONAL  REQUIRED DATASETS
//*
//*  DATASET     USAGE
//*  ---------   ----------------------------------------
//*  LKED.SYSIN  LINKAGE EDITOR CONTROL COMMAND INPUT FILE.  THIS
//*               MUST CONTAIN THE FOLLOWING STATEMENT:
//*
//*             INCLUDE IDMSLIB(IDMS,IDMSCANC)
//*
//*              IF DESIRED, THE NAME OF THE IDMSOPTI MODULE MAY
//*             BE ADDED TO THE LIST OF NAMES.
//*
//COB    EXEC PGM=&COMPGM,
//            PARM='RES,NOSEQ,&CPARM'
//STEPLIB  DD DSN=&CLIB,DISP=SHR
//SYSTERM  DD SYSOUT=&PRT
//SYSPRINT DD &CPRT
//SYSUT1   DD DSN=&&SYSUT1,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT2   DD DSN=&&SYSUT2,UNIT=&TEMPDA,SPACE=(460,(700,100))
```

```
//SYSUT3   DD DSN=&&SYSUT3,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT4   DD DSN=&&SYSUT4,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT5   DD DSN=&&SYSUT5,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT6   DD DSN=&&SYSUT6,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSUT7   DD DSN=&&SYSUT7,UNIT=&TEMPDA,SPACE=(460,(700,100))
//SYSLIN   DD DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=&TEMPDA,
//            SPACE=(80,(500,100))
//SYSIN    DD DSN=&SRCLIB(&MEMBER),DISP=(SHR)
//LKED   EXEC PGM=&LNKPGM,COND=(5,LT),REGION=2048K,
//            PARM='&LPARM'
//SYSLIN   DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD DDNAME=SYSIN
//SYSLMOD  DD DSN=&ULIB(&MEMBER),DISP=SHR
//SYSLIB   DD DSN=&LLIB,DISP=SHR
//         DD DSN=&RLIB,DISP=SHR
//         DD DSN=&ULIB,DISP=SHR
//IDMSLIB  DD DSN=&IDMSLIB,DISP=SHR
//SYSUT1   DD UNIT=&TEMPDA,SPACE=(1024,(50,20))
//SYSPRINT DD SYSOUT=&PRT
```

B.3. MetaSuite Source Library

This library includes the source code for runtime support modules of the MetaSuite system. These are supplied in the event that it becomes necessary to recompile the support modules due to changes in your environment (a new release of the COBOL compiler might require the recompilation).

MetaSuite Runtime Modules

Use a File Transfer Program (FTP) to copy the MetaSuite runtime source code (all the .cbl files) from the `MetaSuite\GENSz_zOS\SYSTEM` folder into the MetaSuite source library.

Note: If your COBOL compiler does not support Unicode (i.e., the Unicode functions "display-of" and "national-of" with the option "second argument" to provide a code page (for example 1208 to do the conversion to or from UTF-8)), some runtime programs can not be compiled.

Appendix C- Troubleshooting

This chapter describes some of the problems that may be encountered both when installing and when using MetaSuite.

C.1. Installation

The most common cause of installation problems is that the runtime modules have been compiled with compiler options that are incompatible with the user's generated program.

C.2. Compiler Options

The compiler options used to compile your user programs must exactly match the compiler options used for the MetaSuite runtime parameters. This problem can manifest itself in a number of different abends, the two most common being a S0C1 or a U519.

This is a common occurrence when your CPARM override is out of sync with the runtime module compiler parameters, as well as when you are trying to link to a user-written utility module which wasn't taken into consideration during the MetaSuite installation process. Programs must be recompiled and relinked so that all have compatible compiler options.

C.3. Linker Options

Depending upon the default parameters for the editor at your site, you may need to make two modifications to the Link Edit parameters.

1. You may need to increase your link edit region size. Symptoms can be a variety of different error messages from the Linkage Editor and/or messages from the run step about the condition of the COBOL code. Messages in the run step will generally give the impression that something is wrong with the COBOL code. You can correct this problem by increasing the REGION size, as follows:

   ```
   //GENEXEC EXEC  MSproc,REGION.LKED=xxxxK,
   ```

   and adding the following override statement:

   ```
   //    LPARM='SIZE=(y,z),LIST,XREF,LET,CALL'
   ```

   where MSproc is the procedure name, xxxx is the new region size, and y,z are the size parameters for the load module.

   Note: Consult with your Systems Programmer to determine the effectiveness of the above JCL Statements

for your program in your environment.

2. Your may need to increase the size of SYSUT1. A symptom of this problem might be that your jobs are abending with a B37 on the SYSUT1 data set. You can correct this problem by including the following override of SYSUT1 after the LKED.SYSLIB statements:

```
//LKED.SYSUT1   DD   UNIT=DISK,SPACE=(3000,(100,50))
```

C.4. COBOL Compiler

Depending upon the default parameters for the size of the load set file, you may abend in the compile step with a B37 on the &&LOADSET file. You can correct this problem by overriding the COBOL step SYSLIN statement as follows:

```
//COB.SYSLIN   DD   DSN=&&LOADSET,DISP=(,PASS),
//          UNIT=SYSDA,SPACE=(160,(10000,3000))
```

The above statement should be placed just prior to the Link Edit SYSLIB DD statements.

C.5. JCL Region Size

The current version of MetaSuite requires 2048K to run. If you find that is insufficient, change the region size on your default job card. You can also specify the REGION parameter in your job card information if you are building your own JCL.

C.6. Installation Abends and Errors

During installation testing, it is not uncommon to have installation mistakes show up as errors or abends. The following is a list of the known types of abends encountered during installation.

SCO1

If the abend occurs in the RUN step of the first MetaSuite job that you are trying to execute, this usually means that the **CPARM** for the procedures and the runtime modules are out of sync. It can also mean that the **CPARM** parameters are in conflict with the link edit parameters (NORES compiler option and NCAL link edit option for the runtime modules).

Appendix D - System Messages and Return Codes

There are two kinds of ABENDs that you may encounter; system abends and user abends. The system abends that we have listed here are a brief attempt to cover the most commonly encountered abends. More general information can be found on all such IBM messages in their System Messages and Codes manual.

If the list of probable causes does not help you resolve your problem, take the steps listed below to generate more diagnostic information.

If you are still unable to resolve your problem, call IKAN Solutions Technical Support.

Please have the following information available for the technician:

- JCL used to run the program
- MDL, or Dictionary DDL (whichever is appropriate)
- MXL, which is another name for the MetaSuite's 4GL language
- Use the proc override CPRT='SYSOUT=*' in order to force a COBOL listing
- COBOL listing with the MAP,LIST (COBOL II) compiler options in effect (add these to the CPARM proc override, remembering to repeat all the default options as well)
- Complete Job output, from the JES log right down to the actual dump.

Keep in mind that you should review all of your SYSOUT before calling for technical assistance. For example, an apparent problem in your RUN step might actually be only a symptom of a problem in the LKED (link edit) step. Many times the solution is pointed out for you in the JES log.

D.1. System Abends

Starred (**) information indicates the most usual cause for a given abend.

001	** Possible I/O error. Check the block size of the file. R2=DCB of bad file. ** With IEC161I 052-084 message. Pointing to a file means that the file is in use. Record length or block size is not correct as specified in the JCL. True I/O error on tape or disk. Attempt to read past EOF. Recording mode V(ariable) assumed or specified and file is F(ixed) or vice versa. Device Malfunction

002	** Possible I/O error. Check DCB information for file. R2-> DCB of bad file Wrong record length (Indexed Sequential) Too many tracks specified for cylinder. This can occur in the RUN step and is accompanied by the message IEC036I. The abend and the message number is embedded in a lot of system information. Look at the lines around the message for a MetaSuite (PPTxxxx) entity name. In the RUN step the message will point to PPTFnn. The usual reason is that the file is a multi-record file which has been defined to MetaSuite without record keys. Sometimes the reason is the record length is incorrect.
013	** Record length or block size is incorrect R2 +X'52' indicates LRECL Trying to add a member of a PDS to a sequential data set Looking for a PDS member in a sequential data set
031	No data on input file Error occurred while processing using QISAM (could be out of sequence key if loading an ISAM data set). Insufficient disk space
0C1	**New Installation: Runtime modules have been compiled with different compiler parameters than the generated program. (See Appendix C- Troubleshooting on page 32.) **If R15 = 0048 or 0050 or 0052 or any combination, cause is either a missing DD card, a misspelled DDname, or moving data to/from an unopened file. **If R15 = 0, you are probably missing a subroutine; check the link edit map. **If R15 = anything else, probably destroyed core memory by moving data with a bad subscript, or by moving to a field which does not exist because of a closed file, or because EOF was reached. If interrupt address is within the program or close to it, the probable cause is subscript or index problem. Review the program logic with particular attention to any routines that fill tables. Program compiled RES is trying to call a NORES subroutine.
0C2	Privileged operation exception Incorrect or missing DD statement Run-away subscript
0C3	Execute exception; faulty program logic
0C4	RUN Step Causes: **Assignment was made to a target file field in an INITIAL procedure. **Moving data to/from a record which does not exist because the file is closed or EOF has been reached. **Missing or misspelled DD statement. **Block size and record size specified as equal in a variable length file. Invalid subscript or index; (i.e. exceeding a table via a subscript) Inclusion of a STOP RUN statement in the input or output **Accompanied with message number IEA705I with RC 878-10. The message text indicates that the error occurred during a GETMAIN. You need to increase your REGION allocation on the JOB card.
0C5	Bad subscript: an address has been calculated outside the bounds of available storage, or you have exceeded the table limit. Moving data to/from a record that does not exist because file is closed or EOF is reached. (I.e. Attempt to reference an I/O area before a read or open was issued.) A (COBOL sort verb) sort has failed but you did not check the sort-return special register. Improper exit from a performed routine. A subroutine's argument which was expected to be a binary fullword (BINARY PIC S9(5) or integer) is a halfword (BINARY PIC S9(4)).

0C6	Bad subscript: an address has been calculated outside the bounds of available storage, or you have exceeded the table limit. A (COBOL sort verb) sort has failed but you did not check the sort-return special register. A subroutine's argument which was expected to be a binary fullword (BINARY PIC S9(5) or integer) is a halfword (BINARY PIC S9(4)).
0C7	**Input record numeric field contains blanks, decimal points, or commas in numeric field. (You many want to use SYS-NUMERIC-CHECK feature. Another option is to define all input fields as MIXED; check these MIXED FIELDS for non-numeric data; and then optionally move numeric or "altered" values to a numeric work field. **The figurative constants ZERO or LOW-VALUES were moved to a group level field which had subordinate packed types fields. OR Data was moved from a display field into a COMP or COMP-3 field at a group level, so no conversion was done. **USAGE clause was omitted or incorrect for input record numeric field **Data anomalies in the input data **Subscript or index value exceeded maximum and invalid data was accessed. **Move from ZONED to MIXED field does not move as numeric data. Subsequent referencing of that MIXED field as numeric data can cause this problem. **Data field is not initialized or is initialized improperly. This can be especially troublesome when a MIXED field redefines a numeric field, but the MIXED field has the INITIAL clause defined to it. Linkage section is incorrect: arguments are in wrong order, too many/few arguments, wrong usage clause, or wrong length parameters. (COBOL sort verb) sort has failed.
0C8	Fixed point overflow (integer too large)
0C9	Attempt to divide by zero Fixed-point divide error (integer too large)
0CA	Decimal overflow (decimal number too large)
0CB	Attempt to divide by zero Decimal divide error (decimal quotient too large)
0CC	Exponent overflow
0CD	Exponent underflow
0CF	Attempt to divide floating point number by zero
0F1	Supervisor abend: can be caused by failure to allocate space for an index when creating an ISAM file
0F2	Invalid arguments passed to a supervisor call. Assembler only.
0F3	A machine error occurred and the job was aborted
106	**If R15 = 0C, Return Code = 14, then region is too small. Sometimes accompanied by a message saying 'FETCH FAILED FOR MODULE module-name'. Increase the REGION allocation on your JOB card. If R15 = 0C, there was an invalid record type in the load module If R15 = 0E, there was an invalid address in the load module If R15 = 0F, OS cannot diagnose the error. (It could be there is no directory for library)
122	Operator cancelled job with a dump (see code 222)
12D	During execution of an overlay program, the segment descriptor table was destroyed. Probably occurred because of a bad subscript, or data was moved to a record that does not exist because the file is closed or EOF has been reached.
137	Error at the end of volume of magnetic tape

13C	**While running a DB2 job, a system file is left open. The results may be ok, but the return code is not. Result of having the wrong runtime modules at runtime or the wrong COBOL II libraries at runtime
213	**Disk data set of JOBLIB or STEPLIB is not found. Check spelling of data set name **DISP parameter specified as OLD or SHR for an output data set. DISP=MOD was specified and the volume serial number was specified, but no space had previously been allocated. Wrong volume specified
214	Error during the close of magnetic tape file
222	Operator cancelled job. Could be because program was in a loop (check CPU time and number of lines printed), or asked for a non-existent tape or disk (check messages in HASP system log for messages from the operator indicating the problem area) or tried to access protected data set (check system log).
237	A verification error occurred in label processing at end of volume. Tape label block count did not agree with DCB block count, probably because a block of data was missed or skipped due to a hardware error. Wrong tape serial or dsname.
322	**Job cancelled because a specified step or task time limit was reached. **Specified print lines exceeded or specified punch limit exceeded. **Check for a loop before you raise the limits and resubmit. A typical looping problem occurs when a DO WHILE loop is coded with no changes in the value of the controlling variable.
337	End of file occurred but no end of file routine address was specified.
413	Tape or disk open error File opened, but no volume serial number specified for file. (Tape) I/O error when reading volume label. Volume sequence number greater than number of volumes allocated to data set. A non-1600 BPI tape was allocated to a 1600-BPI only drive See LBLPRINT and DUMPER
422	Insufficient job queue space. Your job has too much JCL. Split it into two or more jobs.
513	Attempt to open more than one file on tape simultaneously
522	The job step exceeded the maximum allowable wait time. Should never happen.
606	Not enough main storage to load module. Check your region specifications.
613	Error occurred when reading or writing a tape label. Either there is no header label on the input tape or there are problems in positioning the tape.
706	Program is not executable. This is usually a link edit problem
713	The expiration date has not yet been reached, and the operator cancelled the job step. Message IEC107D should be in the log at the front of your listing, identifying the file.
737	Concatenating disk data set not found.
804	**Region is too small to allow program to be loaded or to allocate necessary buffers or I/O routines. Increase region size on the job card, and always allow at least an extra 8K in case you need to get a dump.
806	**If R15 = 04, the program was not found. The link edit step may have failed, and the program does not yet exist in the load library. If R15 = 08, there was an I/O error when trying to load the program. (It could be there is no directory for the library.)

80A	**Not enough main storage. Increase the region size on the JOB or EXEC card. Usually accompanied by error message IEA705I 'Error during GETMAIN ...'. If this is a sort with a large block size on sort-in/sort-out, make the REGION at least 14K more than CORE= keyword in the PARM field.
813	**Tape data set name does not match the DSName on the DD card. Check the spelling of the DSName and the volume serial. Run LBLPRINT to print the contents of the tape labels.
878	**This usually indicates a region size problem. Increase the region specified for the job and run it again.
A0A	**Program did not have enough main storage work space for your source program. Increase the size in the parm field, and increase the region allocation on the JOB card.
B0A	**Region is too small. Increase the region on the JOB card If this is a link edit, the keyword SIZE=(value1,value2) in parm has too small a value1. Increase value1 considerably and make the REGION 10K larger than value1.
B37	**Space allocation problem. Check the space utilization for the data sets. Condense the offending PDS, and rerun the job. (There should be an accompanying message in the JES log with the name of the file.)
C03	**TSO end of job, but a file was left open. This error is usually associated with a DB2 batch job. **If this problem occurs at the end of the generate step, the COBOL code can be captured to a permanent data set and the job can be restarted at the Precompile or Compile step.
D37	**All space allocated by the primary extent has been used. No secondary extent was specified. Allocate a new, larger data set and copy the old one to the new one.
E37	**While writing to a PDS, no more space was available on the disk, or all space allocated to the PDS has been used. Check the space utilization for the data set.
F13	Conflicting or missing data set parameter, or a PDS member not found.
F37	Concatenated disk data set not found.

D.2. System Abends

Starred (**) information indicates the most usual cause for a given abend.

001	** Possible I/O error. Check the block size of the file. R2=DCB of bad file. ** With IEC161I 052-084 message. Pointing to a file means that the file is in use. Record length or block size is not correct as specified in the JCL. True I/O error on tape or disk. Attempt to read past EOF. Recording mode V(ariable) assumed or specified and file is F(ixed) or vice versa. Device Malfunction

002	** Possible I/O error. Check DCB information for file. R2-> DCB of bad file Wrong record length (Indexed Sequential) Too many tracks specified for cylinder. This can occur in the RUN step and is accompanied by the message IEC036I. The abend and the message number is embedded in a lot of system information. Look at the lines around the message for a MetaSuite (PPTxxxx) entity name. In the RUN step the message will point to PPTFnn. The usual reason is that the file is a multi-record file which has been defined to MetaSuite without record keys. Sometimes the reason is the record length is incorrect.
013	** Record length or block size is incorrect R2 +X'52' indicates LRECL Trying to add a member of a PDS to a sequential data set Looking for a PDS member in a sequential data set
031	No data on input file Error occurred while processing using QISAM (could be out of sequence key if loading an ISAM data set). Insufficient disk space
0C1	**New Installation: Runtime modules have been compiled with different compiler parms than the generated program. (See Appendix C- Troubleshooting on page 32.) **If R15 = 0048 or 0050 or 0052 or any combination, cause is either a missing DD card, a misspelled DDname, or moving data to/from an unopened file. **If R15 = 0, you are probably missing a subroutine; check the link edit map. **If R15 = anything else, probably destroyed core memory by moving data with a bad subscript, or by moving to a field which does not exist because of a closed file, or because EOF was reached. If interrupt address is within the program or close to it, the probable cause is subscript or index problem. Review the program logic with particular attention to any routines that fill tables.Program compiled RES is trying to call a NORES subroutine.
0C2	Privileged operation exception Incorrect or missing DD statement Run-away subscript
0C3	Execute exception; faulty program logic
0C4	RUN Step Causes: **Assignment was made to a target file field in an INITIAL procedure. **Moving data to/from a record which does not exist because the file is closed or EOF has been reached. **Missing or misspelled DD statement. **Block size and record size specified as equal in a variable length file. Invalid subscript or index; (i.e. exceeding a table via a subscript) Inclusion of a STOP RUN statement in the input or output **Accompanied with message number IEA705I with RC 878-10. The message text indicates that the error occurred during a GETMAIN. You need to increase your REGION allocation on the JOB card.
0C5	Bad subscript: an address has been calculated outside the bounds of available storage, or you have exceeded the table limit. Moving data to/from a record that does not exist because file is closed or EOF is reached. (I.e. Attempt to reference an I/O area before a read or open was issued.) A (COBOL sort verb) sort has failed but you did not check the sort-return special register. Improper exit from a performed routine. A subroutine's argument which was expected to be a binary fullword (BINARY PIC S9(5) or integer) is a halfword (BINARY PIC S9(4)).

0C6	Bad subscript: an address has been calculated outside the bounds of available storage, or you have exceeded the table limit. A (COBOL sort verb) sort has failed but you did not check the sort-return special register. A subroutine's argument which was expected to be a binary fullword (BINARY PIC S9(5) or integer) is a halfword (BINARY PIC S9(4)).
0C7	**Input record numeric field contains blanks, decimal points, or commas in numeric field. (You many want to use SYS-NUMERIC-CHECK feature. Another option is to define all input fields as MIXED; check these MIXED FIELDS for non-numeric data; and then optionally move numeric or "altered" values to a numeric work field. **The figurative constants ZERO or LOW-VALUES were moved to a group level field which had subordinate packed types fields. OR Data was moved from a display field into a COMP or COMP-3 field at a group level, so no conversion was done. **USAGE clause was omitted or incorrect for input record numeric field **Data anomalies in the input data **Subscript or index value exceeded maximum and invalid data was accessed. **Move from ZONED to MIXED field does not move as numeric data. Subsequent referencing of that MIXED field as numeric data can cause this problem. **Data field is not initialized or is initialized improperly. This can be especially troublesome when a MIXED field redefines a numeric field, but the MIXED field has the INITIAL clause defined to it. Linkage section is incorrect: arguments are in wrong order, too many/few arguments, wrong usage clause, or wrong length parameters. (COBOL sort verb) sort has failed.
0C8	Fixed point overflow (integer too large)
0C9	Attempt to divide by zero Fixed-point divide error (integer too large)
0CA	Decimal overflow (decimal number too large)
0CB	Attempt to divide by zero Decimal divide error (decimal quotient too large)
0CC	Exponent overflow
0CD	Exponent underflow
0CF	Attempt to divide floating point number by zero
0F1	Supervisor abend: can be caused by failure to allocate space for an index when creating an ISAM file
0F2	Invalid arguments passed to a supervisor call. Assembler only.
0F3	A machine error occurred and the job was aborted
106	**If R15 = 0C, Return Code = 14, then region is too small. Sometimes accompanied by a message saying 'FETCH FAILED FOR MODULE module-name'. Increase the REGION allocation on your JOB card. If R15 = 0C, there was an invalid record type in the load module If R15 = 0E, there was an invalid address in the load module If R15 = 0F, OS cannot diagnose the error. (It could be there is no directory for library)
122	Operator cancelled job with a dump (see code 222)
12D	During execution of an overlay program, the segment descriptor table was destroyed. Probably occurred because of a bad subscript, or data was moved to a record that does not exist because the file is closed or EOF has been reached.
137	Error at the end of volume of magnetic tape

13C	**While running a DB2 job, a system file is left open. The results may be ok, but the return code is not. Result of having the wrong runtime modules at runtime or the wrong COBOL II libraries at runtime
213	**Disk data set of JOBLIB or STEPLIB is not found. Check spelling of data set name **DISP parameter specified as OLD or SHR for an output data set. DISP=MOD was specified and the volume serial number was specified, but no space had previously been allocated. Wrong volume specified
214	Error during the close of magnetic tape file
222	Operator cancelled job. Could be because program was in a loop (check CPU time and number of lines printed), or asked for a non-existent tape or disk (check messages in HASP system log for messages from the operator indicating the problem area) or tried to access protected data set (check system log).
237	A verification error occurred in label processing at end of volume. Tape label block count did not agree with DCB block count, probably because a block of data was missed or skipped due to a hardware error. Wrong tape serial or dsname.
322	**Job cancelled because a specified step or task time limit was reached. **Specified print lines exceeded or specified punch limit exceeded. **Check for a loop before you raise the limits and resubmit. A typical looping problem occurs when a DO WHILE loop is coded with no changes in the value of the controlling variable.
337	End of file occurred but no end of file routine address was specified.
413	Tape or disk open error File opened, but no volume serial number specified for file. (Tape) I/O error when reading volume label. Volume sequence number greater than number of volumes allocated to data set. A non-1600 BPI tape was allocated to a 1600-BPI only drive See LBLPRINT and DUMPER
422	Insufficient job queue space. Your job has too much JCL. Split it into two or more jobs.
513	Attempt to open more than one file on tape simultaneously
522	The job step exceeded the maximum allowable wait time. Should never happen.
606	Not enough main storage to load module. Check your region specifications.
613	Error occurred when reading or writing a tape label. Either there is no header label on the input tape or there are problems in positioning the tape.
706	Program is not executable. This is usually a link edit problem
713	The expiration date has not yet been reached, and the operator cancelled the job step. Message IEC107D should be in the log at the front of your listing, identifying the file.
737	Concatenating disk data set not found.
804	**Region is too small to allow program to be loaded or to allocate necessary buffers or I/O routines. Increase region size on the job card, and always allow at least an extra 8K in case you need to get a dump.
806	**If R15 = 04, the program was not found. The link edit step may have failed, and the program does not yet exist in the load library. If R15 = 08, there was an I/O error when trying to load the program. (It could be there is no directory for the library.)

80A	**Not enough main storage. Increase the region size on the JOB or EXEC card. Usually accompanied by error message IEA705I 'Error during GETMAIN ...'. If this is a sort with a large block size on sort-in/sort-out, make the REGION at least 14K more than CORE= keyword in the PARM field.
813	**Tape data set name does not match the DSName on the DD card. Check the spelling of the DSName and the volume serial. Run LBLPRINT to print the contents of the tape labels.
878	**This usually indicates a region size problem. Increase the region specified for the job and run it again.
A0A	**Program did not have enough main storage work space for your source program. Increase the size in the parm field, and increase the region allocation on the JOB card.
B0A	**Region is too small. Increase the region on the JOB card If this is a link edit, the keyword SIZE=(value1,value2) in parm has too small a value1. Increase value1 considerably and make the REGION 10K larger than value1.
B37	**Space allocation problem. Check the space utilization for the data sets. Condense the offending PDS, and rerun the job. (There should be an accompanying message in the JES log with the name of the file.)
C03	**TSO end of job, but a file was left open. This error is usually associated with a DB2 batch job. **If this problem occurs at the end of the generate step, the COBOL code can be captured to a permanent data set and the job can be restarted at the Precompile or Compile step.
D37	**All space allocated by the primary extent has been used. No secondary extent was specified. Allocate a new, larger data set and copy the old one to the new one.
E37	**While writing to a PDS, no more space was available on the disk, or all space allocated to the PDS has been used. Check the space utilization for the data set.
F13	Conflicting or missing data set parameter, or a PDS member not found.
F37	Concatenated disk data set not found.

D.3. User Abends

U0002	If the job was running under COBOL II, this problem could be associated with an IBM sort bug. There are no consistent messages, but there is usually an indication that the abend occurred in the sort. If this occurs, you will need to speak with your IBM rep. The APAR number is PL69542, and the PTF number is UL81337.
U0003	This is a catch-all abend. If this occurs in the compile step, it usually means that the generated program is so large that is exceeds the COBOL compiler region and buffer size. These are SIZ and BUF options on the compiler parameter list (CPARM) of the EXEC statement. To discover what the default levels were, look for the COBOL compiler options in effect for the program which abended. You will find it in the listing from that job. If they were not generated for your job, rerun the job after adding CPRT='SYSOUT=*' to the EXEC statement. Add the SIZ and BUF options to your CPARM list and increase the size of these options. CAUTION! Whenever changing or adding a CPARM, you must copy the entire list of CPARMs from the procedure to the EXEC statement and then add or change a parameter.
U0046	This abend occurs when certain work files run out of space during the RUN step. Messages will point to SORTWnn. You will need to increase the SSPACE allocation on the EXEC statement.

U0519	This problem is usually encountered during the installation process at a COBOL II site. It indicates a problem in the incompatibility between the runtime modules and the user program. Please bring this problem to the attention of the person who installed MetaSuite. Also refer to the Appendix C- Troubleshooting (page 32).
U1006	When accompanied by IGZ006I 'THE REFERENCE TO TABLE 'table-name' BY VERB NUMBER 'nn' ON LINE 'nnnnnn' ADDRESSED AN AREA OUTSIDE THE REGION OF THE TABLE', this abend indicates that the link edit region size needs to be increased. Add REGION.LKED=400K to the EXEC card as well as the following override: PARM.LKED=('SIZE=(90000,60000),LIST,LET,XREF,CALL')
U1035	Accompanied by message IGZ035I 'THERE WAS AN UNSUCCESSFUL OPEN OR CLOSE OF FILE PPTFnn' IN PROGRAM 'program-name'. This indicates that there is an incorrect or missing DD card in the JCL. The file name is named PPTFnn, where nn is the number that relates to the order of the files specified in the MXL.