

# Initialization - Com-plete Startup Procedure

This chapter describes the Com-plete initialization procedure under the following headings:

- MVS Procedure
  - MVS Required DD Statements
  - MVS Symbolic Parameters
  - Initialize Com-plete System Intercept (VSE)
  - VSE Job Control
  - VSE Logical Partition Assignments
  - VSE Printer Assignment
  - VSE Initialization Files
  - Com-plete Subsystems
- 

## MVS Procedure

For MVS, initialize or start Com-plete by invoking the procedure COMPLETE. When Com-plete is initially installed, this procedure is added to the installation's system procedure library, SYS1.PROCLIB, or any user-defined procedure library.

The following figure illustrates a typical COMPLETE procedure that might be installed for an MVS installation.

## MVS Job Control

During the installation process, you will either alter the supplied COMJCL procedure to suit your requirements, copy it to an installation procedure library, or use it as the basis for a job to be submitted. The following is the procedure in question. The illustrated procedure serves as the basis for the various descriptions and explanations that follow.

```
//COMPLETE PROC PREFIX='COM',
//      SYSPARM=SYSPARM,
//      OPARM=,
//      REG=6000K
//*
//*      Com-plete SYSTEM STARTUP PROCEDURE FOR OS.
//*
//* BEFORE STARTING THIS YOU MUST CHANGE THE STARTUP PROGRAM NAME TO
//* THAT OF THE TLINXX MODULE YOU COPIED TO THE APF AUTHORISED LIBRARY
//* IN THE INSTALLATION JOB #1.
//*
//* FOR A SYSPLEX INSTALLATION, ADD THE RLS PARAMETER TO COMSYS1-4
//*
//IEFPROC EXEC PGM=TLINOS, <----- SEE NOTE ABOVE
//      PARM='&OPARM',
//      REGION=&REG,TIME=1440,DPRTY=(14,14)
```

```

// *
// *****
// STEPLIB DD DISP=SHR,DSN=AN.APF.AUTHORISED.LIBRARY
// *****
// *
// COMPLIB DD DISP=SHR,DSN=&PREFIX..USER.LOAD
// DD DISP=SHR,DSN=&PREFIX..LOAD
// DD DISP=SHR,DSN=&PREFIX..MAPS
// DD DISP=SHR,DSN=APS.LOAD
// * DD DISP=SHR,DSN=THE.CURRENT.ADABAS.SM.LOAD
// *
// COMSYS1 DD DISP=SHR,DSN=&PREFIX..COMSYS.BASE ,RLS=NRI
// COMSYS3 DD DISP=SHR,DSN=&PREFIX..COMSYS.USERDEF ,RLS=NRI
// COMSYS4 DD DISP=SHR,DSN=&PREFIX..COMSYS.CATALOG ,RLS=NRI
// *
// SYSPARM DD DISP=SHR,DSN=&PREFIX..USER.SOURCE(&SYSPARM)
// SYSMAP DD DISP=SHR,DSN=&PREFIX..MAPS
// COMSPL DD DISP=SHR,DSN=&PREFIX..SPOOL
// COMSD DD DISP=SHR,DSN=&PREFIX..SD
// *CAPTUR1 DD DISP=SHR,DSN=&PREFIX..CAPTUR1
// *CAPTUR2 DD DISP=SHR,DSN=&PREFIX..CAPTUR2
// SYSDUMP DD DISP=OLD,DSN=&PREFIX..SYSDUMP
// SYSPRINT DD SYSOUT=X
// SYSRDR1 DD SYSOUT=(X,INTRDR

```

The COMPLETE procedure can be invoked from the operator's console via an MVS START command:

```
S COMPLETE,...
```

or via an MVS batch job:

```
//COMPLETE JOB .....
//IEFPROC EXEC COMPLETE,...
```

In either situation, an understanding of the DD statement functions and usage of the available startup options is required to implement the features of Com-plete.

The following sections use the above example as a basis for defining the Com-plete initialization procedure for MVS.

## MVS Required DD Statements

The required and optional DD statements in the above procedure are described in more detail below:

### STEPLIB

Required

This identifies the authorized load library on which the Com-plete MVS startup module TLINOS resides. No other Com-plete modules need to be available in this library

This data set is only referenced once during initialization and therefore its placement is not an issue.

### COMPLIB

Required

This identifies the PDS library concatenation that effectively becomes the STEPLIB for the duration of the run. This means that all modules loaded during the execution of Com-plete are loaded from this concatenation.

With applymod 79 or 80 set, this will be a highly accessed and performance-critical data set. Without these applymods, the activity on this data set will depend on the number of MVS loads issued from the installation's applications.

### **COMSYSn**

Required

This identifies the VSAM data sets containing various system information required by Com-plete. They will always be highly accessed data sets and should therefore be placed accordingly.

### **SYSMDUMP**

Optional

If not specified, no support can be provided for any problem, as no diagnostic information is available. It identifies where the MVS control program should write a formatted dump in case of an abend or if Com-plete requests such as dump. You are recommended to specify this instead of the SYSABEND or SYSUDUMP DD statements, as more information is available for diagnosis if a problem should occur. You will also note that normally a SYSMDUMP will not take as long to occur. Estimates for the size of the data set specified by this DD statement must be made according to the IBM documentation.

### **CAPTURn**

Optional

Each CAPTURn DD name eg. CAPTUR1,CAPTUR2 etc.. must point to a suitably created VSAM Capture Data Set when capture is being used in the system. When Capture is being used and these data sets are being accessed, the workload on them depends on the amount of data being captured. The data sets must also be placed in such a way that one does not interfere with another, because when one data set is full, it will be unloaded and reinitialized while another capture data set is in use.

### **COMSPL**

Required

This must be used to point to the VSAM COMSPL data set. Refer to the COMSPL description in the chapter *Com-plete Files and Associated User Files* for more details.

### **COMSD**

Required

This must point to the VSAM SD file dataset defined and initialized by TUSDUTIL for this Com-plete.

### **SYSPRINT**

Optional

At termination, or using the *STATS* operator command, statistics are printed to this DD. If the DD is not specified, it will be allocated using dynamic allocation as a sysout data set.

### **SYSRDRn**

Optional

This DD is only necessary if you wish to use the Remote Job Entry (RJE) facilities of Com-plete. It must be assigned to an JES internal reader as in the above example. Depending on the level of RJE activity, up to 9 statement can be specified: SYSRDR1, SYSRDR2 etc..

### **SYSMAP**

Required for the UMAP utility.

Identifies the MVS load library into which UMAP will store maps. This library must also be specified in COMPLIB, so that maps can be loaded from it.

### **SYSPARM**

Required

Identifies the file or library member in which the desired Com-plete system parameters are to be found.

If Com-plete is to be periodically stopped and started in order to test various startup options (for example, number of threads, size of threads, different TIBTAB, etc.), the symbolic parameter &SYSPARM can be used to identify the member containing the desired options. However, a short-term test of a specific option can be affected by use of the PARM option at startup time.

The various system parameters available are described in the section *Startup Options (Sysparms)*.

## **MVS Symbolic Parameters**

You can modify the COMPLETE procedure, including the format and usage of the symbolic parameters. However, the symbolic parameters indicated below are generally sufficient to meet the needs of most installations:

### **&OPARM**

Specifies a character string which is passed to the Com-plete control program via the PARM sysparm (see the section on the startup options for use of this feature).

### **&SYSPARM**

Specifies the member name in the library identified by the DD name SYSPARM, which contains the control statements specifying the startup parameters (sysparms).

You can create multiple members to allow tailoring of Com-plete initialization to meet the specific conditions defined by the control statements.

### **&PREFIX**

Specifies the default high level index (prefix) under which the Com-plete files are cataloged.

## **Initialize Com-plete System Intercept (VSE)**

In a VSE environment, communication between the Com-plete nucleus and the user program is handled with the Com-plete SVC (SUPERVISOR CALL, normally 200).

The program COMSIP is used to dynamically install the SVC without an IPL. COMSIP requires a prior SET SDL for the SVC, and therefore must run in the BG partition.

You only need one system intercept module COMSIP for all Com-pletes running in your system.

### **Executing the COMSIP Program**

Important:

If you have software from CA and/or Macro 4 installed, this step must run **AFTER** the initialization of the CA system and **BEFORE** the initialization of Macro 4.

Insert the following (or the equivalent) in the ASI BG JCL procedure immediately before the **START** of the **POWER** partition so that the Com-plete SVC will be installed automatically during each IPL.

```
// DLBL      SAGLIB, '.....LIBRARY'
// EXTENT   SYS010, vvvvvv
// ASSGN    SYS010, DISK, VOL=vvvvvv, SHR
// LIBDEF   PHASE, SEARCH=SAGLIB.APSvrS, TEMP
SET  SDL
CRSVATBL, SVA
/*
// UPSI     00000000
// EXEC     COMSIP, SIZE=AUTO
```

See also the section **Startup Procedure in VSE Installation** in the Com-plete Installation documentation.

## **VSE Job Control**

For VSE, Com-plete is initialized by running a job that can be left in the POWER reader using a job disposition of LEAVE (DISP=L). The following figure illustrates a typical Com-plete procedure that might be used at a VSE installation.

```

* $$ JOB JNM=JOB COM51,CLASS=2,DISP=D,LDEST=(,????)
* $$ LST CLASS=A,DISP=H,RBS=1000
// JOB JOB COM51 STARTUP FOR Com-plete V51
// OPTION PARTDUMP,NOSYSDMP
// ASSGN SYS009,SYSLST
* $$ LST DISP=D,CLASS=M,LST=SYS009,DEST=(,????)
// LIBDEF PHASE,SEARCH=(SAGLIB.COMvrs,SAGLIB.APSvrs,SAGLIB.ADA???,
SAGLIB.COMUSER,
IJSYSRS.SYSLIB,PRD1.BASE,PRD2.PROD),TEMP
// LIBDEF *,CATALOG=SAGLIB.COMUSER
/*
/*
// DLBL COMCAT,'????????',,VSAM
/*
/* Com-plete COMSYS CONTAINERS
/*
// DLBL COMSYS1,'COM.COMSYS.BASE',,VSAM,CAT=COMCAT
// DLBL COMSYS3,'COM.COMSYS.USERS',,VSAM,CAT=COMCAT
// DLBL COMSYS4,'COM.COMSYS.CATALOG',,VSAM,CAT=COMCAT
/*
/* Com-plete SD-FILE
/*
// DLBL COMSD,'COMPLETE.VSAM.SDFILE',,VSAM,CAT=COMCAT
/*
/* Com-plete PRINTOUT-SPOOL DATA SET
/*
// DLBL COMSPL,'COM.VSAM.SPOOL',,VSAM,CAT=COMCAT
/*
/* Com-plete VSAM DUMPFIL
/*
// DLBL COMDMP,'COM.VSAM.DUMPFIL',,VSAM,CAT=COMCAT
/*
/* Com-plete CAPTURE FILES
/*
/* DLBL CAPTUR1,'COM.VSAM.CAPTURE1',,VSAM,CAT=COMCAT
/* DLBL CAPTUR2,'COM.VSAM.CAPTURE2',,VSAM,CAT=COMCAT
/*
// UPSI 00000000
// EXEC TLINSP,SIZE=AUTO
*
* Com-plete SYSTEM INITIALIZATION PARAMETERS
*
/&
* $$ EOJ

```

The following sections discuss important considerations to be made when initializing Com-plete for VSE. The initialization parameters are defined in the section *Startup Options (Sysparms)*.

## VSE Logical Partition Assignments

You must ensure that a logical partition assignment(s) (VSE LUB) is made for each disk CUU to be used by the Com-plete utilities USERV, UDD and UDZAP.

## VSE Printer Assignment

The USPOOL utility has the ability to route a printout from the Com-plete online spool to a system printer (controlled by POWER). In order to separate this output from the Com-plete job log, a second printer definition must be made in the Com-plete startup deck. The following figure illustrates sample JCL,

including the required assignments:

```
* $$ JOB JNM=COMPLETE,CLASS=2,DISP=L
* $$ LST CLASS=A,DISP=D
// JOB COMPLETE
// OPTION ...
...
// ASSGN SYS009,FEF
* $$ LST CLASS=A,DISP=D,LST=SYS009
```

Note that SYS009 is used by the Com-plete spooling task. All other parameters are installation-dependent.

## VSE Initialization Files

The VSE initialization files are summarized in the following table.

| File Name | File ID                        | LUB  | Description                          |
|-----------|--------------------------------|------|--------------------------------------|
| CAPTURn   | COMPLETE.CAPTURE<br>(optional) | VSAM | Com-plete CAPTURE                    |
| COMSPL    | COMPLETE.SPOOL                 | VSAM | Com-plete spool file                 |
| COMSD     | COMPLETE.SD                    |      | Com-plete SDLIB file                 |
| COMSYS1   | COMPLETE.BASE                  | VSAM | Com-plete System Data Container      |
| COMSYS3   | COMPLETE.USERS                 | VSAM | Com-plete System Data Container      |
| COMSYS4   | COMPLETE.CATALOG               | VSAM | Com-plete System Data Container      |
| COMDMP    | COMPLETE.DUMP                  | VSAM | Com-plete Dump File in case of abend |

## Com-plete Subsystems

The Com-plete nucleus routines are grouped into functional units, so-called subsystems. A number of basic subsystems is required to operate Com-plete, others are optional (meaning they are only required when certain functions within Com-plete are to be used). The following table lists all currently available subsystems and their attributes:

| Subsystem | Function                                      | Basic/Option | Act |
|-----------|---|--------------|-----|
| ACCESS    | Host and batch communication                  | Option       | Yes |
| CAPTURE   | Data logging facility                         | Option       | Yes |
| COMSEC    | Com-plete Security extension (add-on product) | Option       | No  |
| DEBUG     | Application debugging aid                     | Option       | Yes |
| MSGPO     | Message switching / Printout spooling         | Option       | Yes |
| VSAM      | VSAM file control system                      | Option       | Yes |
| VTAM      | VTAM interface                                | Option       | Yes |

Subsystems with "Yes" in the *Act* column are active by default. Optional subsystems not active by default can be enabled with the sysparm SUBSYS-ACTIVATE=. Subsystems active by default can be disabled with the sysparm SUBSYS-IGNORE=.

Disabling a subsystem affects the operation of Com-plete as follows:

1. Sysparms relevant to the particular subsystem are ignored.
2. Modules for that subsystem are NOT loaded during initialization. Therefore, functions carried out by the subsystem are not available until Com-plete is restarted with this subsystem active.
3. Main storage requirements decrease because the subsystem's modules are not loaded.