



IMS Common Services Layer - An Introduction

By Danny Barr, Senior Product Support Representative

Table of Contents

Common Services Layer overview	3
Automatic RECON Loss Notification (ARLN)	3
Single Point of Control (SPOC)	3
SPOC sample	4
A Cookbook for CSL Setup	6
Step 1: Select names to be used	6
Step 2: Create OM, RM, and SCI initialization members in IMS PROCLIB ..	6
Step 3: Create OM, RM, and SCI members in operating system PROCLIB ..	7
Step 4: Identify the CSL and IMSPLEX to IMS system	7
Summary	8

Common Services Layer overview

No recent IMS enhancement has more potential to improve internal processing than does the Common Services Layer (CSL). The CSL provides many benefits. This document describes the basics of CSL usage and how to get started with the CSL.

The CSL was introduced in IMS Version 8 and has been steadily enhanced in subsequent releases. Many of us have overlooked CSL even though it provides IMS command and DBRC enhancements that make the IMS systems programming and IMS DBA functions easier. If you look at the enhancements to the CSL since IMS Version 8, you will realize that the CSL is here to stay, because it is becoming as necessary for safe, reliable IMS processing as DBRC itself.

The initial release of the CSL provided the following commands:

- > Automatic RECON Loss Notification (ARLN)
- > Single Point of Control (SPOC)
- > IMS TYPE 2 QUERY

The IMS Version 8 CSL requires three address spaces, which are used to form an IMSPLEX:

- > Resource Manager (RM)
- > Operations Manager (OM)
- > Structured Call Interface (SCI) address spaces

An IMSPLEX is a set of IMS systems, similar to an IMS group, that share DBRC and/or databases and can have a common command interface (SPOC). The IMSPLEX is defined using the keywords in the CSL. A cookbook for developing the CSL definitions is attached. (Without reservation, I encourage you to create an IMS system that contains a Common Services Layer so that you can see what CSL can do for your environment.)

An IMS or IMS related program identifies itself as a member of an IMSPLEX with the IMSPLEX parameter on the EXEC statement or through the presence of the exit module DSPSCIX0. Once membership is established with a connection to the CSL, the program, control region, etc. have access to the CSL services. One of these services is ARLN.

Automatic RECON Loss Notification (ARLN)

When ARLN is active, all members of the IMSPLEX accessing a set of RECONS are registered with the CSL. When ARLN is active, only members of the IMSPLEX registered in the RECON can access that set of RECON data sets (assuming that all DBRC access is through an IMS batch or online region control program or an IMS utility that passes control of allocation and deal location of RECON data sets to DBRC). An outsider will not be able to access a set of RECONS for a given IMSPLEX

unless it is identified via module DSPSCIX0 or PARM='IMSPLEX=imsplexname' on the EXEC statement. When something happens to a RECON data set, for example a RECON is discarded, the CSL notifies each registered member program that the RECON has been lost and, if the program observes IMSPLEX rules, the program will immediately deallocate the lost RECON. This function allows RECON reorganization and reinstatement of a DISCARDED RECON data set while IMS subsystems are actively reading and updating DBRC.

Single Point of Control (SPOC)

Another feature of the CSL is the SPOC. SPOC as delivered with IMS can be invoked with the TSO CLIST command "execute '...SDFSEEXEC (DFSAPPL)' 'HLQ (...)'". With this CLIST, IMS commands can be issued to one or more IMS systems in an IMSPLEX through ISPF rather than through an IMS defined terminal such as an IMS Master Terminal or outstand reply.

In IMS Version 8, when CSL is active both ARLN and SPOC are available. If ARLN is not desired, IMS Version 9 or higher is required. The Resource Manager address space communicates with DBRC to provide ARLN. In IMS Version 9, a CSL SPOC can be used without activating ARLN. With a CSL active, all Type 1 and the new Type 2 commands can be issued by an assembler program, a high-level language program a procedural language program like REXX, under TSO or ISPF as an MVS job or task. An example of a batch REXX is provided on the following page.

This example allows an IMS command to be issued in a batch job. When commands are issued using the CSL, the results of the command are returned - making the CSL an attractive Automated Operator Interface (AOI). Within the CSL, the Operations Manager interfaces with the SPOC. The output of the Operations Manager is XML. The example REXX was created using the IBM IMS Version 8 example supplied in the IMS Version 8 Common Services Guide and Reference. If you are adventurous enough to try a CSL or a CSL is available at your location, try running the REXX EXEC as a TSO JOB STEP.

SPOC sample

```
** SAMPLE JCL TO EXEC THE REXX EXEC REXXCMD

/** JCL TO EXECUTE THE REXX EXEC REXXCMD
//REXXCMD EXEC PGM=IKJEFT01,DYNAMNBR=45,COND=EVEN
//STEPLIB DD DISP=SHR,DSN=HLQ.SDFSRESL
/** CLIST REXXCMD IS A MEMBER CLIST IN LIBRARY HLQ.CLIST
//SYSPROC DD DISP=SHR,DSN=HLQ.CLIST
//SYSTSPRT DD SYSOUT=*
/** THREE SETS OF PARAMETERS SEPARATED BY SLASHES
/** FORMAT:
/** implexname/IMSID OR blank for all imsid(s) in the IMSPLEX/the IMS command
//SYSTSIN DD *
  %REXXCMD DMB91//QUERY IMSPLEX SHOW(ALL)
  %REXXCMD DMB91//DIS DB ALL.
/**
//
```

```
** SAMPLE REXX SPOC REXX EXEC CREATED USING THE IMS V8 EXAMPLE
```

```
      MEMBER= REXXCMD
/* REXX */
/* REXXcmd */
  trace off /*trace all or trace off*/
  parse upper arg theInput
  say theInput
  parse var theInput theIMSpIex '/' theIMSids '/' theIMScmd
  say theIMSpIex;
  say "-----";
  say theIMSids;
  say "-----";
  say theIMScmd
  say "-----";
  Address TSO 'CSLULXSB' /* set up the REXX environment */
  if rc > 0 then,
    do
      Say "+++++ TSO Environment WAS NOT established +++++"
      Say "+++++          TERMINATING          +++++"
      exit
    end
  Address IMSSPOC /* set the default host command to IMSSPOC */
                /* Address TSO for TSO commands          */
                /* Address MVS for MVS commands          */
                /* and                                     */
                /* Address IDCAMS for IDCAMS commands    */
                /*                                         */
  if rc > 0 then,
    do
      Say "+++++ Attemp to set TSO default +++++"
      Say "+++++ Host to IMSSPOC failed +++++"
      Say "+++++ rc='imsrc ' reason='imsreason' +++++"
      exit /* Exit if the return code is not zero. */
    end
  /*
  /*
  "IMS" theIMSpIex
                /* Issue the IMS Subcommand to establish */
                /* the implex name.                       */
                /* the IMS command must be issued before */
                /* an other command can be entered.       */
                /*                                         */
  if rc > 0 then,
    do
      Say "+++++ Failure to Join the IMSPLEX = "theIMSpIex
      say '+++++ rc='imsrc 'reason='imsreason
      exit /* Exit if the return code is not zero. */
    end
  /*
  /*
  "route" theIMSids
                /*
                /* example: "route IMSA IMSB IMSC"
                /* The command processors are the specific */
                /* systems that will process the commands. */
                /* If you do not specify a command
                /* processor the previous routing value
                /* is removed and commands will be routed */
                /*
```

```

                /* to all members of the implex (this is */
                /* the default). */
                /* */
                /* */
if rc > 0 then,
do
    say 'rc='imsrc 'reason='imsreason
    exit /* Exit if the return code is not zero. */
end
                /* */
                /* */
cartid = "myrex"
"cart" cartid ;
                /* Use the Command and Respon Token */
                /* (cart) IMS Subcommand to set the name */
                /* of the command and response token. */
                /* This 16-character text string token is */
                /* used to retrieve the command response. */
                /* You must issue the cart subcommand */
                /* before you can issue any IMS COMMANDS. */
                /* */
                /* */
if rc > 0 then,
do
    say 'rc='imsrc 'reason='imsreason
    exit /* Exit if the return code is not zero. */
end
                /* */
                /* */
"WAIT 1:00"
                /* Use the WAIT IMS Subcommand to provide */
                /* a timeout value to OM. The time value */
                /* must be in the form MMM:SS or ssss. */
                /* The maximum value you can specify is */
                /* 9999:59. The Wait subcommand is */
                /* optional. */
                /* */
                /* */
if rc > 0 then,
do
    say 'rc='imsrc 'reason='imsreason
    exit /* Exit if the return code is not zero. */
end
                /* */
                /* */
theIMScmd
say "theIMScmd was issued"
if rc > 0 then say theIMScmd 'rc='imsrc 'reason='imsreason
do
    results = cslulgts('TEMP.', cartid,"1:30")
    /* say 'results='results 'imsrc='imsrc ' reason='imsreason */
    if temp.0 /= '' then
do
    say 'temp.'0'=('temp.0')'
do idx = 1 to temp.0
    say temp.idx
end
end
end
"END"
exit
display_results: procedure expose temp. idx rsp_flag idx2 verbose
if pos(temp.idx,'<msg>') > 0 then,
do
    msg=strip(temp.idx,1,'<msg>')
    say msg
    temp.idx=strip(msg,t,'</msg>')
    say temp.idx
end
return;

```

A Cookbook for CSL Setup

This section provides step-by-step instructions for setting up the CSL.

Step 1: Select names to be used

Make the following decisions and choices:

- > Identify the IMSIDs of one or more IMS systems to include in the IMSPLEX. For example, if two production IMS systems will be included in the IMSPLEX, use IMSIDs IMSA and IMSB.
- > Choose a name (1-5 characters) for the IMSPLEX name to use for this IMSPLEX. For example, PLEX1 can be the IMSPLEX name of the IMSPLEX that includes production IMS systems IMSA and IMSB.
- > From the IMSPLEX name, derive a 3-character name to be used for the CSL initialization suffix. For example, from the IMSPLEX name PLEX1, PX1 can be the CSL initialization suffix. The IMSPLEX name and the CSL initialization suffix are used in the following procedures. The CSL initialization suffix is also used in the IMS PROCLIB DFSPBims member for each participating IMS. (The change to the DFSPBims member is described later in this section.)

Step 2: Create OM, RM, and SCI initialization members in IMS PROCLIB

To create the OM, RM, and SCI initialization members in the IMS PROCLIB, perform the following steps:

1. Copy members CSLOI000, CSLRI000, and CSLSI000 of the SDFSSMPL library to your IMS system PROCLIB.
2. Copy member BPECONFIG of the SDFSSLIB library to member BPECONFIG in your IMS system PROCLIB. (No changes need to be made to this member.)
3. Rename members CSLOI000, CSLRI000, and CSLSI000 in your IMS system PROCLIB by replacing the 000 suffix with your chosen CSL initialization suffix.

For example, PX1 is the CSL initialization suffix for the IMSPLEX named PLEX1. Member CSLOI000 is renamed to CSLOIPX1. Member CSLRI000 is renamed to CSLRIPX1. Member CSLSI000 is renamed to CSLSIPX1.

4. Change the following parameters in the renamed CSLOIxxx member:
 - OMNAME=xxxxx
 - IMSPLEX(NAME=xxxxx)
 - CMDTEXTDSN=your.ims.dsn

For example, using the previous choices and definitions, the following parameters are set in the CSLOIPX1 member. For simplicity, the IMSPLEX name is used as the value of the OMNAME keyword.

- OMNAME=PLEX1
- IMSPLEX(NAME=PLEX1)
- CMDTEXTDSN=your.ims.SDFSDATA

In the DFSPBims member of the IMS PROCLIB, the *ims* suffix is defined by the RGSUF parameter in the operating system PROCLIB that contains the startup JCL for the IMS system. The RGSUF parameter is shown in the following example:

```
//yourims PROC
RGN=128M,RGSUF=IMS,PARM1='IMSID=IMSA,APPLI
D1=IMSA',
PARM2='IRLM=Y,IRLMNM=ARLM'
```

Later, the literal suffix OM is appended to the specified OMNAME value to become the OM member name in the operating system PROCLIB. The name of the OM member in the operating system PROCLIB will be PLEX1OM.

Refer to the appropriate IMS documentation or the IMS system installer for the name of the CMDTEXTDSN library. The lowest level name may be SDFSDATA.

5. Change the following parameters in the renamed CSLRIxxx member:

- IMSPLEX(NAME=xxxxx)
- RMNAME=xxxxx

For example, using the previous choices and definitions, the following parameters are set in the CSLRIPX1 member. For simplicity, the IMSPLEX name is used as the value of the RMNAME keyword.

- IMSPLEX(NAME=PLEX1)
- RMNAME=PLEX1

Later, the literal suffix RM is appended to the specified RMNAME value to become the RM member name in the operating system PROCLIB. The name of the RM member in the operating system PROCLIB will be PLEX1RM.

6. Change the following parameters in the renamed CSLSIxxx member:

- SCINAME=xxxxx
- IMSPLEX(NAME=xxxxx)

For example, using the previous choices and definitions, the following parameters are set in the CSLSIPX1 member. For simplicity, the IMSPLEX name is used as the value of the SCINAME keyword.

- SCINAME=PLEX1
- IMSPLEX(NAME=PLEX1)

Later, the literal suffix *SCI* is appended to the specified SCINAME value to become the SCI member name in the operating system PROCLIB. The name of the SCI member in the operating system PROCLIB will be PLEX1SCI.

Step 3: Create OM, RM, and SCI members in operating system PROCLIB

To create the OM, RM, and SCI members in the operating system PROCLIB, perform the following steps:

1. Copy members CSLOM, CSLRM, and CSLSCI of the SDFSSMPL library to the operating system PROCLIB library.
2. Rename the members as follows:

- Rename member CSLOM to the OM member name, which consists of the value of the OMNAME keyword (as defined in the renamed CSLOlxxx member of the IMS PROCLIB) plus the literal suffix OM.

In the example, the OMNAME keyword value is PLEX1 in member CSLOIPX1 of the IMS PROCLIB, so the renamed CSLOM member is PLEX1OM.

- Rename member CSLRM to the RM member name, which consists of the value of the RMNAME keyword (as defined in the renamed CSLRlxxx member of the IMS PROCLIB) plus the literal suffix RM.

In the example, the RMNAME keyword value is PLEX1 in member CSLRIPX1 of the IMS PROCLIB, so the renamed CSLRM member is PLEX1RM.

- Rename member CSLSCI to the SCI member name, which consists of the value of the SCINAME keyword (as defined in the renamed CSLSlxxx member of the IMS PROCLIB) plus the literal suffix SCI.

In the example, the SCINAME keyword value is PLEX1 in member CSLSIPX1 of the IMS PROCLIB, so the renamed CSLSCI member is PLEX1SCI.

3. In the renamed operating system PROCLIB members, make the following changes as appropriate:

- In each of the renamed operating system PROCLIB members, specify the IMS RESLIB data set name (one to 44 characters) in the following statement:

```
RESLIB='your.ims.reslib'
```

- In each of the renamed operating system PROCLIB members, specify the IMS system PROCLIB to be used with the following statement change:

```
//PROCLIB DD DSN='your.ims.proclib',DISP=SHR
```

- In the OM member, change the 000 value of the OMINIT=000 keyword to the CSL initialization suffix.

For example, in PLEX1OM, change OMINIT=000 to OMINIT=PX1.

- In the RM member, change the 000 value of the RMINIT=000 keyword to the CSL initialization suffix.

For example, in PLEX1RM, change RMINIT=000 to RMINIT=PX1.

- In the SCI member, change the 000 value of the SCIIINIT=000 keyword to the CSL initialization suffix.

For example, in PLEX1SCI, change SCIIINIT=000 to SCIIINIT=PX1.

Step 4: Identify the CSL and IMSPLEX to IMS system

1. To identify the CSL to the IMS systems that are included in the IMSPLEX, add the CSLG keyword to the DFSPBims member (of the IMS system PROCLIB) that corresponds with each IMS system in the IMSPLEX. Code the following keyword and value:

- CSLG=xxx

The keyword value is the CSL initialization suffix that you used to create the CSLOlxxx, CSLRlxxx, and CSLSlxxx members of the IMS PROCLIB.

For example, for the CSL that is defined by IMS PROCLIB members CSLOIPX1 CSLRIPX1, and CSLSIPX1, code CSLG=PX1 in the DFSPBims for production IMSIDs IMSA and IMSB, which are members of IMSPLEX PLEX1.

- To identify the IMSPLEX to the IMS systems that are to be included in the IMSPLEX, add the IMSPLEX keyword to the DFSCGims member (of the IMS system PROCLIB) that corresponds with each IMS system in the IMSPLEX. Code the following keyword and value:

```
IMSPLEX=imsplexname
```

- In IMS version 9.1 and later environments, the Resource Manager address space is not required for support of commands through the IMS CSL. To support a CSL that includes only SCI and OM address spaces, code the RMENV keyword in the DFSCGims member (of the IMS system PROCLIB). The value should be RMENV=N.

Summary

The CSL simplifies systems management tasks. It provides a single system image and single point of control for shared resources across all IMS systems. The CSL enables you to issue Type 1 and Type 2 commands to multiple IMS systems at once.

The IMS Console, which is included in all BMC IMS products, has an interface to the CSL that simplifies IMS management even more than the CSL itself.



ACTIVATE BUSINESS WITH THE POWER OF IT.™

About BMC Software

BMC Software delivers the solutions IT needs to increase business value through better management of technology and IT processes. Our industry-leading Business Service Management solutions help you reduce cost, lower the risk of business disruption, and benefit from an IT infrastructure built to support business growth and flexibility. Only BMC provides best-practice IT processes, automated technology management, and award-winning BMC Atrium technologies that offer a shared view into how IT services support business priorities. Known for enterprise solutions that span mainframe, distributed systems, and end-user devices, BMC also delivers solutions that address the unique challenges of the mid-sized business. Founded in 1980, BMC has offices worldwide and fiscal 2008 revenues of \$1.73 billion. Activate your business with the power of IT. www.bmc.com.

© Copyright 2009 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

IBM and IMS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

January 30, 2009



98479