



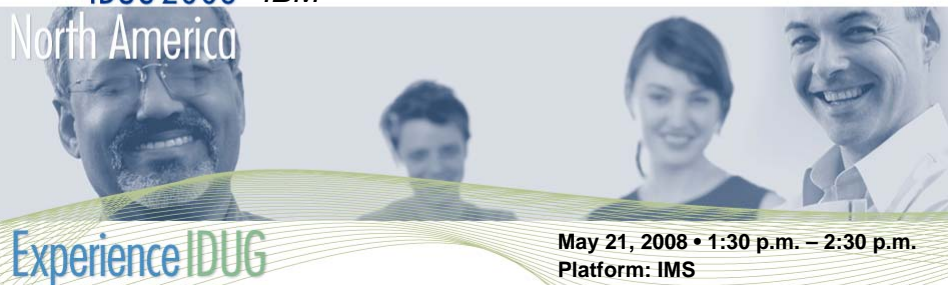
Session: J10

## Preparing for IMS 10: Setting Up the Common Service Layer

IDUG 2008

Diane Goff  
IBM

North America



May 21, 2008 • 1:30 p.m. – 2:30 p.m.  
Platform: IMS

IMS 10, available in October 2007, provides many new functions that require a Common Service Layer (CSL) environment. To get ready for IMS 10, you can begin now to set up and use CSL with your current IMS 9 or IMS 8 systems. This session will first discuss the new functions in IMS 10 that exploit the CSL. Then it will include an overview of CSL and its components, including Operations Manager and Resource Manager. Lastly, this session will go through the steps necessary to set up the CSL environment.

## Objectives

- Learn about the features of IMS 10 that exploit the CSL environment
- Review how CSL can improve systems management for your IMS environments
- Learn about CSL and its major components
- Learn about the two major CSL components, Operations Manager and Resource Manager
- Learn what is necessary to set up the CSL environment

These objectives cover what will be discussed in this session.

## Agenda

- IMS Version 10 New Function using Common Service Layer (CSL)
- Common Service Layer (CSL) Overview
- Overview of Operations Manager (OM)
- Overview of Resource Manager (RM)
- Overview of Structured Call Interface (SCI)
- Setting up the CSL environment
- Starting / Stopping the CSL environment

In preparation for IMS Version 10, I will be discussing the IMS Common Service Layer or CSL. This is the IMS architecture that will be used by many of the major enhancements in IMS Version 10. Since CSL is available in your IMS Version 9 and IMS Version 8 systems today, you can begin your positioning now so that when you migrate to IMS Version 10 you can immediately begin to use many of its new enhancements.

I will first discuss the features of IMS Version 10 that exploit the CSL. Next I will cover an Overview of CSL, discussing its major functions and components. Then I'll spend more time discussing CSL's two major components, Operations Manager (OM) and Resource Manager (RM). Then I'll talk about what is necessary for setting up the CSL environment so that you can begin to use CSL in your IMS environments today.

## IMS Version 10 New Functions using CSL

<ul style="list-style-type: none"> <li>• Dynamic Resource Definition (DRD)</li> <li>• Manage Resources Application</li> <li>• Dynamic Updates of MSC Resources</li> <li>• OM Audit Trail</li> <li>• Batch SPOC Utility</li> </ul> <p>OM (Operations Manager)</p>	<ul style="list-style-type: none"> <li>• Command Enhancements <ul style="list-style-type: none"> <li>• QUERY for Work, Related Resources</li> <li>• QUEUE for message</li> <li>• Resource timestamps</li> <li>• QUERY statistics for MSC Bandwidth</li> <li>• DEDB UPDATE DB</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• ACBLIB Member Online Change</li> <li>• Sysplex Serial Program Management</li> <li>• Global Status</li> </ul> <p>RM (Resource Manager)</p>	<ul style="list-style-type: none"> <li>• Parallel RECON Access (PRA)</li> </ul> <p>SCI (Structured Call Interface)</p>

CSL has three components - Operations Manager (OM), Resource Manager (RM), and Structured Call Interface (SCI) – and this chart lists the major new IMS 10 enhancements that exploit each of these components.

OM has 6 new IMS 10 exploiters:

- DRD for dynamic definitions of programs, transactions, databases, and routing codes
- Manage Resources , a TSO SPOC application to help with DRD
- Dynamic update of MSC resources
- OM Audit Trail, a log stream containing all commands going through OM
- Batch SPOC utility
- Various command enhancements

RM has 3 new IMS 10 exploiters:

- ACBLIB Member Online Change to add/change individual ACBLIB members dynamically
- Sysplex Serial Program Management for enforcing serial programs across an IMSplex with shared queues
- Global Status for maintaining global command status for databases, areas, and transactions in an IMSplex

SCI is used by Parallel RECON Access that improves RECON sharing

Further information on these IMS 10 enhancements is available in the IMS V10 Release Planning Guide at: <http://publib.boulder.ibm.com/epubs/pdf/dfsprgh0.pdf>

## Common Service Layer (CSL) Overview

- An architecture to improve the systems management capabilities for IMS systems
  - Operations management (Operations Manager)
  - Resource management (Resource Manager)
- Provides
  - A single system image (IMSpIex)
  - Ease of use through a single point of control
  - Shared resources across all IMS systems
- Reduces complexity of managing multiple IMS systems

CSL is an architecture focused on improving IMS systems management. It was first available in IMS V8, was enhanced some in IMS V9, and now with IMS V10 there are many new enhancements/exploiters.

The two major focus areas are operations management via the Operations Manager (OM) component and resource management via the Resource Manager (RM) component.

CSL provides a single system image, particularly for an IMSpIex, but also can be used with single IMS systems. It supports IMS TM configurations and DBCTL configurations.

The goal of CSL is to reduce the complexity of managing multiple IMS systems.

## IMSplex Definition

- An IMSplex is a set of IMS address spaces that are working together as a unit and are most likely running in a parallel sysplex (but not required)
- Examples of an IMSplex configuration include:
  - A set of IMS control regions at the V8/V9/V10 level without a CSL that are data sharing or message queue sharing
  - A set of IMS control regions at the V8/V9/V10 level with a CSL that are data sharing and message queue sharing
  - A single IMS control region at the V8/V9/V10 level with a CSL
- Examples of IMSplex components are:
  - IMS subsystems (DB/DC, DBCTL, DCCTL, XRF active, XRF alternate)
  - CQS
  - CSL components (OM, RM, SCI)
  - A batch or DB utility region using DBRC
  - DBRC batch utility

An IMSplex is a set of IMS address spaces that are working together as a unit.

Examples of an IMSplex :

- without a CSL
- with a CSL
- single IMS with CSL

## Common Service Layer (CSL) Components

- Operations Manager (OM)
- Resource Manager (RM)
- Structured Call Interface (SCI)
  
- Based on BPE (Base Primitive Environment)
- Uses CQS (Common Queue Server)
  
- New address spaces
  - OM, RM, SCI, CQS
- New CF structures (optional)
  - Resource, shared queues

There are three components of CSL:

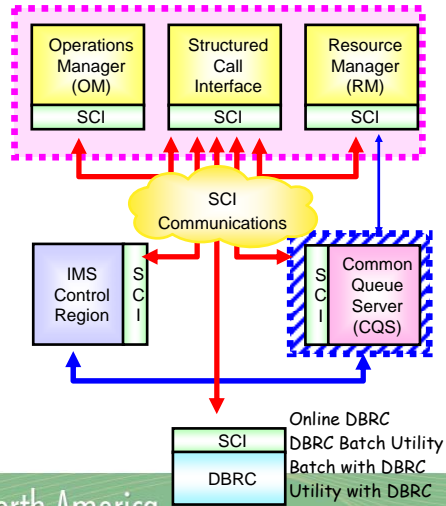
- (1) Operations Manager (OM) for operational ease
- (2) Resource Manager (RM) for sharing resources and coordinating IMSplex-wide processes
- (3) Structured Call Interface (SCI) for coordination and communications within the CSL among its participants

All CSL components use BPE services (available since IMS V6).

CQS can be used for RM functions (optional).

With CSL there are several new address spaces and an optional new coupling facility structure.

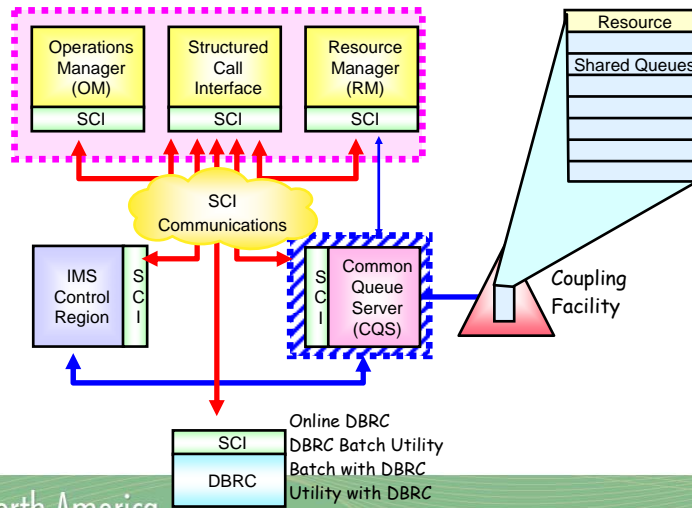
## CSL Architecture (Address Spaces)



This is a diagram of a Common Service Layer environment, showing the new CSL address spaces (OM, SCI, RM, CQS) along with an IMS control region and a DBRC instance as part of single system IMSplex.

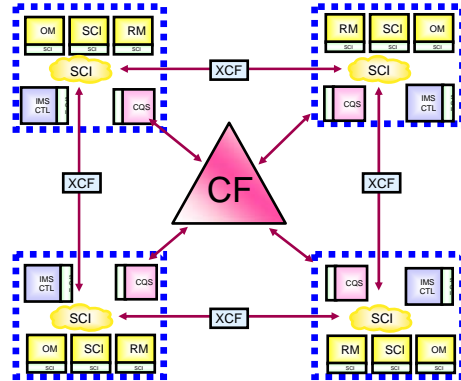


## CSL Architecture (CF Structures)



This CSL diagram shows the addition of the optional coupling facility (CF) structures used by Resource Manager (RM). These optional structures are the resource structure and the shared queues structure.

## IMSplex Configuration



Resource List Structure
LOGR List Structures
Shared Queues List Structures
OSAM Cache Structure
VSAM Cache Structure
Shared VSO Cache Structures
IRLM Lock Structure
VGR List Structure

- In an IMSplex
  - All members share the same CF structures
  - Intra-IMSplex communications is implemented by SCI
    - Uses XCF across z/OS images

For an IMSplex we have multiples of the previous configuration, all communicating through SCI and optionally sharing CF resources.

## Operations Manager (OM) Overview

- Provides 'single point of control' for command entry into an IMSplex
  - Focal point for operations management and automation
- Provides the following services
  - Route commands to IMSplex members registered for the command
  - Consolidate command responses from individual IMSplex members into a single response to present to the command originator
  - Support for new IMSplex commands (type-2 commands) and for existing IMS commands (type-1 commands)
  - An API for IMS commands for automation
  - Command security for authorization using RACF or equivalent plus user exit
  - User exit capability for editing command input and responses
- Configuration
  - One of more OM address spaces required per IMSplex

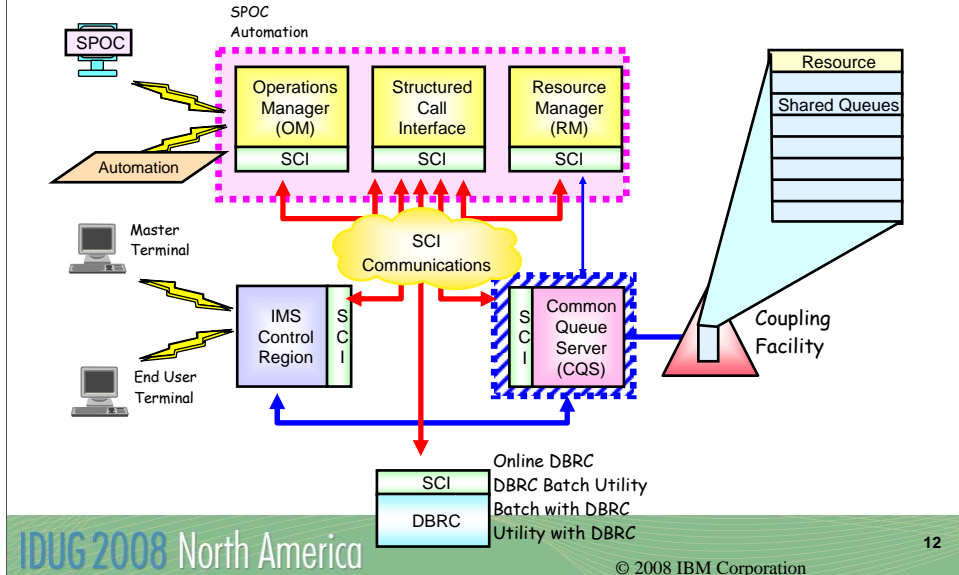
Operations Manager (OM) is the CSL component that focuses on operations management and automation.

It provides the 'single point of control' or SPOC for command entry.

OM provides several services listed above.

One of more OM address spaces are required for an IMSplex (two recommended for backup purposes).

## CSL Architecture – Operations Manager (OM)



The upper left hand corner of this diagram now includes the SPOC and interfaces for automation.

Notice below is the IMS master terminal/end user terminal that provides command services for the IMS control region (older method - type-1 commands only).

## IMS Single Point of Control (SPOC)

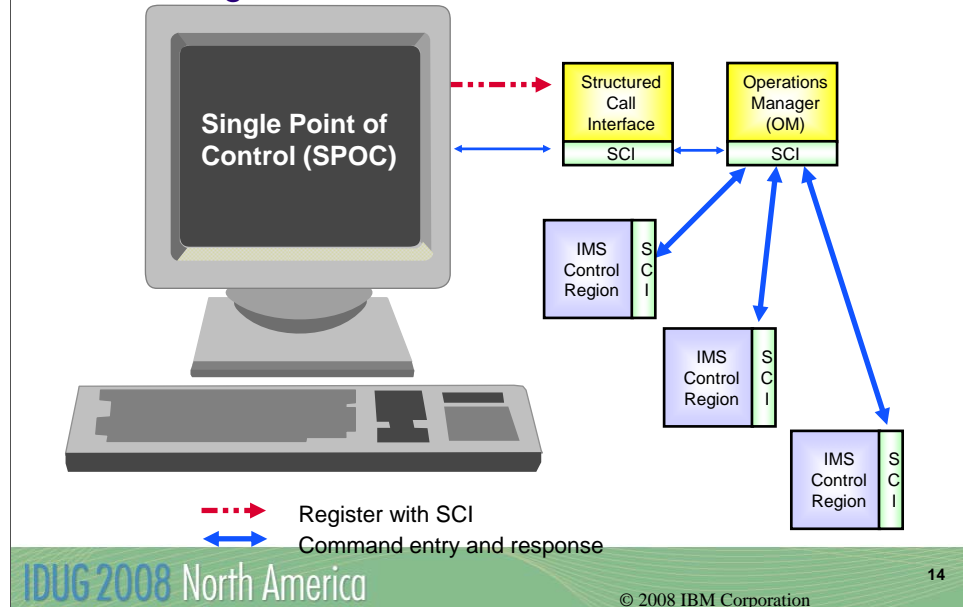
- A SPOC is a program that interfaces between a user and the OM
- From this single point, user can enter commands to any or all IMSs
  - It can run on the mainframe
    - TSO SPOC is provided in IMS
    - REXX SPOC APIs are provided in IMS
  - It can run on the workstation
    - IMS Control Center
- There can be multiple SPOCs in an IMSplex

An IMS SPOC is actually a program that interfaces between a user and OM so that the user can enter commands.

It can run on z/OS – TSO SPOC or REXX SPOC API. These two are provided as part of the IMS product.

It can run on a workstation – the IMS Control Center, which is part of the DB2 UDB Administrative Client and uses IMS Connect for connectivity.

## SPOC Registers with Local SCI



The SPOC uses the CSL component SCI (Structured Call Interface) to

- route commands to IMSplex members from a single console
- consolidate command responses from multiple IMSplex members into a single response

## TSO SPOC (Single Point of Control)

- Runs under z/OS as a TSO/ISPF application
- May or may not be on the same z/OS as OM
  - Must be on a z/OS with an SCI
- Provides a 'green screen' terminal interface from which IMS commands (type-2 or type-1) may be entered by an operator to one or more members of an IMSplex (including DB/DC, DBCTL, DCCTL)
- Formats command responses for display
  - OM response encapsulated in XML
- OM provides security checking
  - TSO userid is used to determine RACF authorization

The TSO SPOC is the typical 'SPOC' used by most customers.

It is provided by IMS and can be started from the IMS Application Menu.

It provides the 'single system image'.

### TSO SPOC

- reformats XML from OM
- type-2 command output displayed in tabular format
- type-1 command output displayed in sequential format

## Type-2 Commands

- INIT (INITiate Process)
- TERM (TERMinate Process)
- UPD (UPDate Resource)
- DEL (DELeTe Resource)
- QRY (QueRY Resource)
- CRE (CREate Resource) (V10)
- QUE (QUEue Message) (V10)

Here is a list of the type-2 commands provided by OM:

- INIT - used for Global Online Change (V8), HALDB Online Reorg (V9). ACBLIB Member Online Change (V10)
- TERM – used with INIT
- UPD - used to update status of resources (V9), for DRD resources (V10)
- DEL - used for LE options (V9), for DRD resources (V10)
- QRY - used for displaying information
- CRE - new command in IMS V10 for DRD
- QUE - new command in IMS V10 to enqueue or dequeue a message for an LTERM or transaction



## SPOC Command Entry Panel

```

File  Display  View  Options  Help  -----
PLX0          IMS Single Point of Control
Command ==> QRY TRAN NAME(A*) SHOW(ALL)
----- Plex .   Route . IMS13   Wait .
Response for:
Override 'Preferences'

F1=Help F3=Exit F4=Showlog F6=Expand F9=Retrieve F12=Cancel
  
```

This is an example of the TSO SPOC command entry panel.

You can set preferences for plexname, routing, and the wait interval.

Commands are submitted on the command line. This example shows issuing a QRY TRAN NAME(A\*) SHOW(ALL) to display information about all transactions whose names start with 'A'.

## Command Response

```

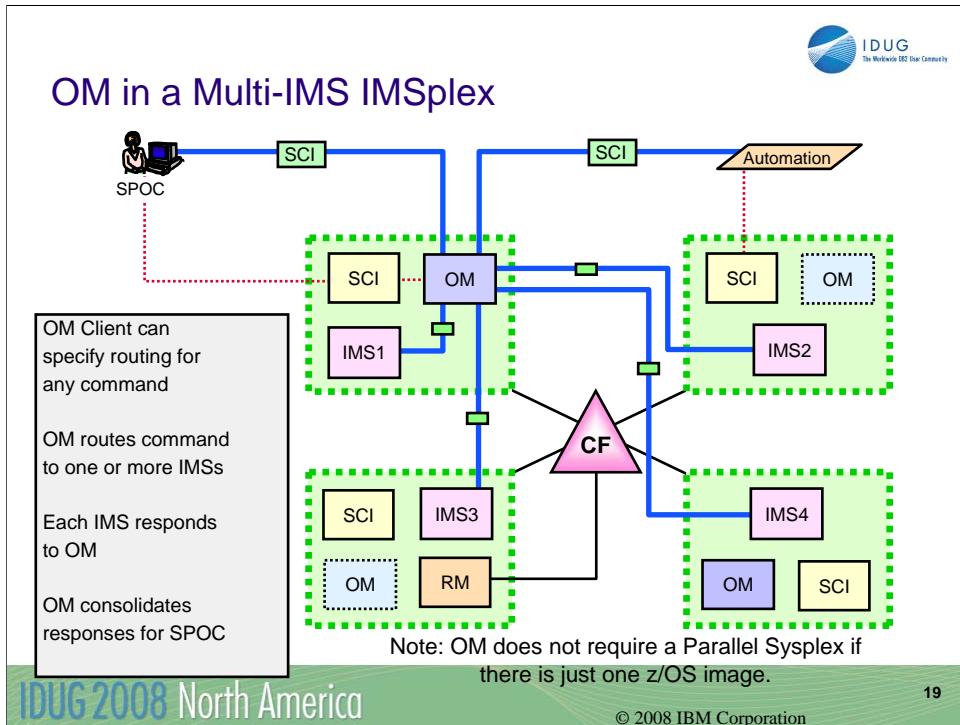
File Display View Options Help -----
                    IMS Single Point of Control
Command ==> _____
----- Plex . _____ Route . IMS13 _____ Wait . _____
Response for: QRY TRAN NAME(A*) SHOW(ALL) More: +>
Trancode MbrName  CC PSBname  LCls  LQCnt  LLCT LPLCT
ADDINV   IMS1     0 INVPSB    4      6     2 65535
ADDINV   IMS3     0 INVPSB    4     12     2 65535
ADDPART  IMS1     0 PARTPSB  23      0 65535 65535
etc.

Display formatted by SPOC from
XML response.

F1=Help F3=Exit F4=Showlog F6=Expand F9=Retrieve F12=Cancel

```

Type-2 command output is displayed in tabular format.

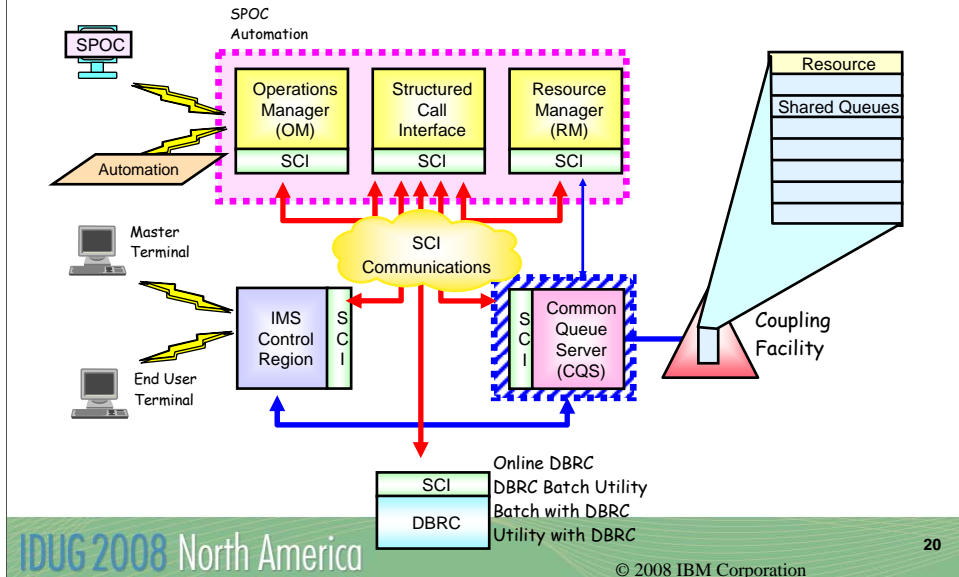


Here is an example of an IMSplex with multiple IMSs in an OM configuration.

Here we have 4 z/OS LPARs. Only 1 OM address space is required (on IMS1) for the IMSplex, though this shows a second OM on IMS4 for backup. IMS2 and IMS3 could also have OM address spaces.

This IMSplex could only have a single IMS, for example, to use OM for DRD for a single system.

## CSL Architecture – Operations Manager (OM)



This diagram summarizes OM as part of the CSL.

## Resource Manager (RM) Overview

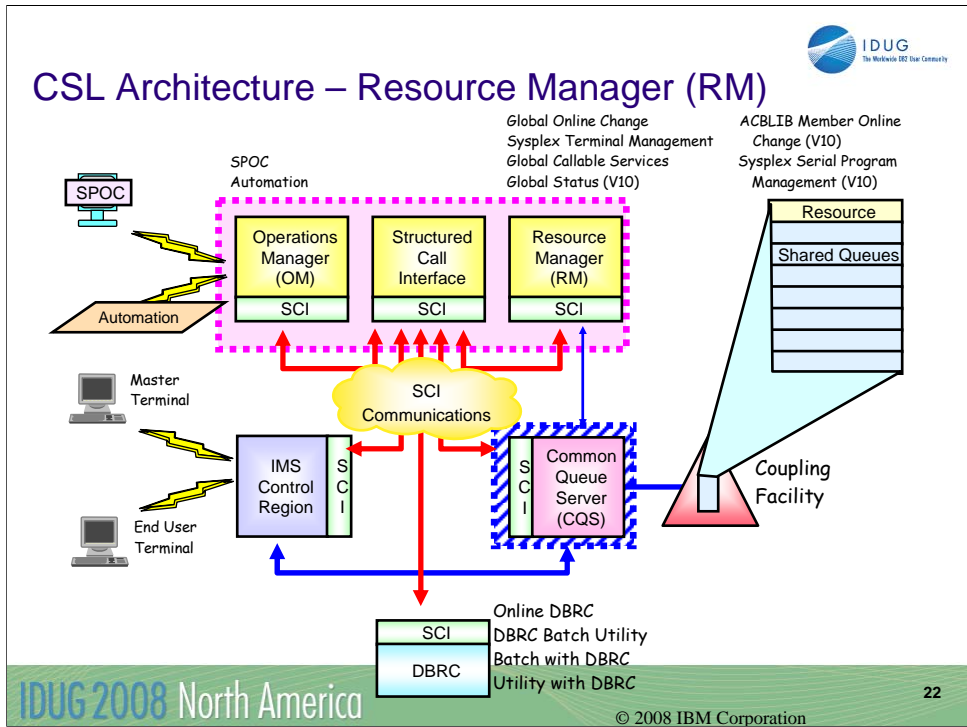
- Provides infrastructure for managing global resources and IMSplex-wide processes
  - IMS is responsible for exploiting RM services
- Provides the following services
  - Maintains global resource information using a resource structure in a Coupling Facility
  - Coordinates IMSplex-wide processes
- Used for the following functions
  - Sysplex Terminal Management (STM) (IMS V8)
  - Global Online Change (GOLC) (IMS V8)
  - Global Callable Services (IMS V8)
  - Global Status (IMS V10)
  - Sysplex Serial Program Management (SSPM) (IMS V10)
  - ACBLIB Member Online Change (IMS V10)

Resource Manager (RM) is the CSL component that focuses on managing global resources in the IMSplex and coordinating IMSplex-wide processes in the IMSplex. IMS is the exploiter of these services.

RM maintains its global resource and process information in the resource structure in the coupling facility.

There are 6 functions in IMS that use these RM facilities:

- STM – manages significant status for VTAM terminals and user resources (IMS V8)
- GOLC – coordinates global online change throughout an IMSplex (IMS V8)
- Global callable services – provides global services for exits (IMS V8)
- Global status – similar to STM but for databases, areas, and transactions (IMS V10)
- SSPM – serial program enforcement across an IMSplex using shared queues (IMS V10)
- ACBLIB MOLC – add/change individual ACBLIB members dynamically (based on GOLC) (IMS V10)

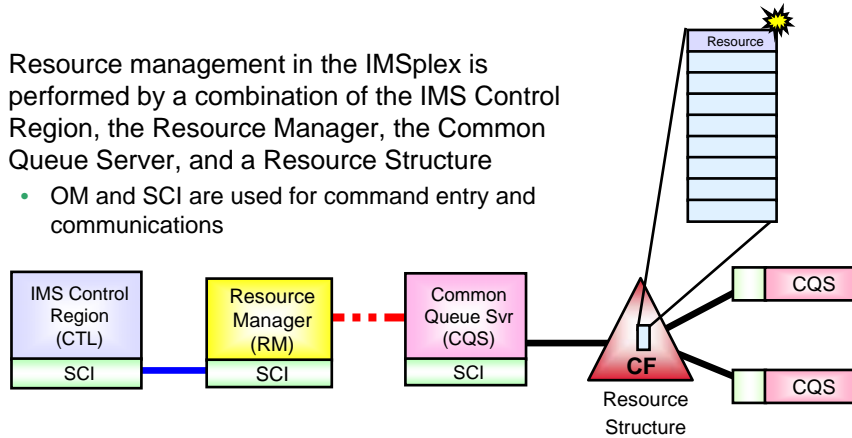


This CSL diagram is updated on the right hand side for Resource Manager.

- New RM address space
- List of functions using RM
- optional Coupling Facility with a Resource Structure
- optional CQS address space if using a Resource Structure
- optional Shared Queues Structure if shared queues is being used

## Resource Manager (RM) Configuration

- Resource management in the IMSplex is performed by a combination of the IMS Control Region, the Resource Manager, the Common Queue Server, and a Resource Structure
  - OM and SCI are used for command entry and communications



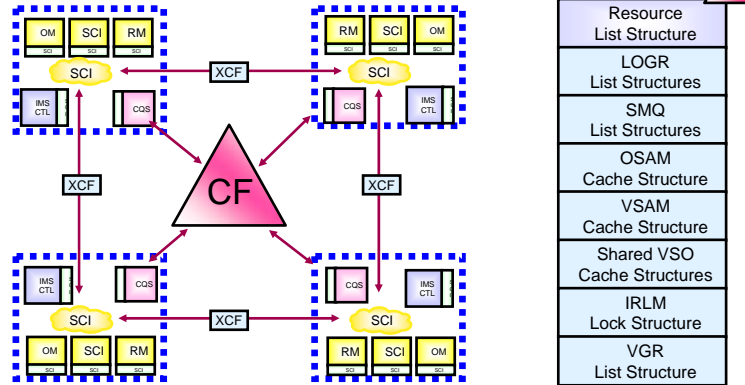
The IMS Control Region is the exploiter of RM services.

CTL interfaces with RM address space for some global processes such as GOLC.

RM address space interfaces with the CQS address space to access the Resource Structure to maintain global resource information.

CQS address space uses shared queues structure if shared queues is implemented.

## IMSpIex RM Configuration



- One or more RM address spaces required per IMSplex in IMS V8
- Zero or more RM address spaces required per IMSplex in IMS V9/V10
  - Need one or more RM address spaces to enable any RM function

Here is a diagram of a configuration using RM in an IMSplex showing the RM address space requirements.

With IMS V8, one or more RM address spaces are required in the IMSplex even if not using any RM functions.

With IMS V9 and IMS V10, zero or more RM address spaces are required in the IMSplex; however to enable any RM function, one or more RM address spaces are required in an IMSplex (second recommended for backup).



## Resource Structure

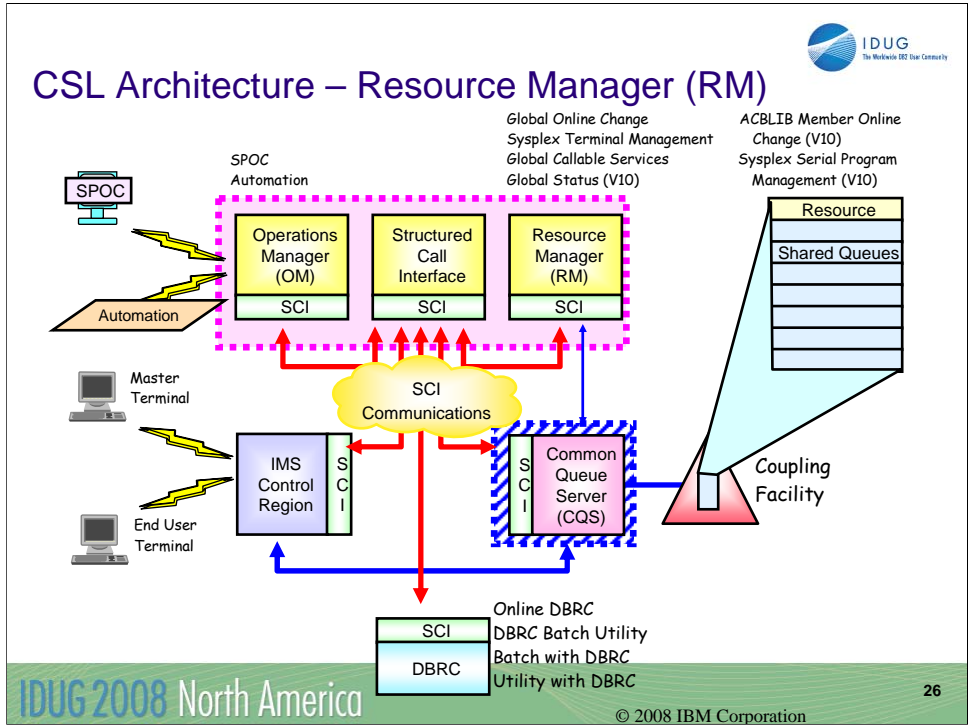
- Resource structure contains global resource information for uniquely named resources
- Resource structure not required for Global Online Change or ACBLIB Member Online Change (V10)
  - i.e., not required for DBCTL
  - Structure will be used if available
- Resource structure required for other IMSplex-wide global processes and access to global resource information
- One resource structure may be defined per IMSplex

Here are the structure requirements for RM.

Remember that the resource structure contains global information for resources and processes and is managed for RM by CQS.

The resource structure is not required for GOLC and MOLC but you do need to have an RM address space. However, the resource structure will be used if it exists.

If you are using any other global information or global processes, then a resource structure is required (STM, Global Status, SSPM).



We have now covered the two major components of CSL, OM and RM, their overall functions and their configuration requirements.

Next we will discuss the other CSL component, Structured Call Interface (SCI).

## Structured Call Interface (SCI) Overview

- Provides communications services among IMSplex members in a single z/OS image and across multiple z/OS images in an IMSplex
- Provides the following services
  - Member registration services (security)
    - OM, RM, CQS, IMS, SPOC, IMS Connect, DBRC
  - Communications services
- Used for the following functions
  - Automatic RECON Loss Notification (ARLN)
  - Parallel RECON Access (PRA) (IMS V10)
- One SCI address space is required on each z/OS image where CSL is active

The major function of SCI is to provide communications among IMSplex members.

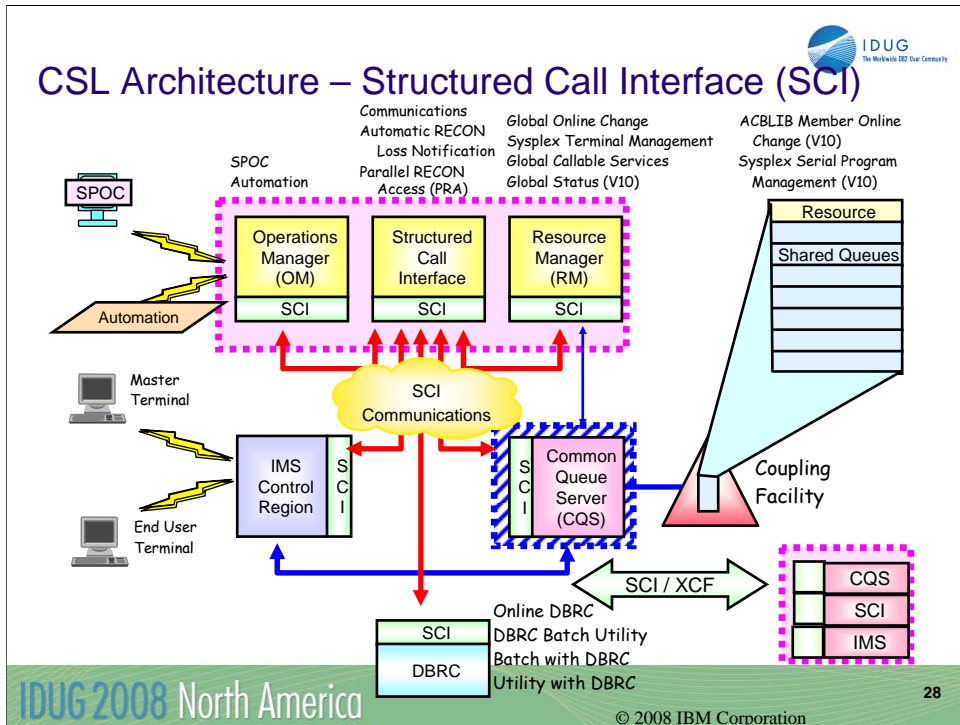
It provides two services:

- member registration services for security
- communications services between members using z/OS XCF

Additional functions that use SCI:

- ARLN – capability that all members of the IMSplex sharing the same RECON are automatically notified of any configuration change (IMS V8)
- PRA - new IMS V10 function for improved RECON sharing

SCI is an important CSL function, therefore we require an SCI address space on each z/OS image where CSL is active.



The CSL architecture diagram is now complete by adding SCI, its address space and its functions.

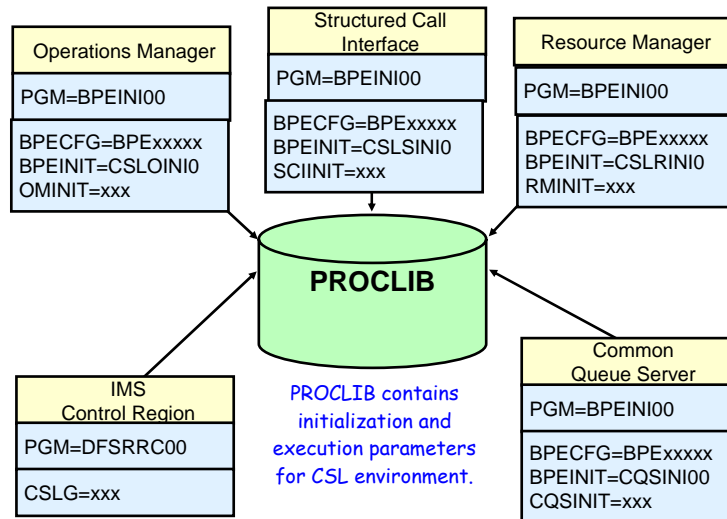
SCI is really the center of a CSL, providing required communications among IMSplex members (IMS control, OM, RM, CQS, DBRC instance) using z/OS XCF services.

## Setting up the CSL Environment

- BPE setup
- SCI setup
- OM setup
- RM setup
- IMS setup
- z/OS / CF setup

All the various CSL components discussed need 'setup'. This setup is not too complex but needs to be done for each component.

## CSL Definition ... High Level View



All CSL components use BPE; therefore, their setup parameters are similar and we will see the commonality as we go through the various component setup details.

There are two types of setup parameters:

- initialization – via an initialization module provided by IMS (via BPEINIT)
- execution – via an initialization PROCLIB member (via xxxINIT)

This presentation covers a high level view of these setup parameters. The details are available in the IMSplex Administration Guide for IMS V10 or the Common Service Layer Guide and Reference for IMS V9 and IMS V8.

## BPE Setup

- Add BPEINI00 to the z/OS Program Properties Table (PPT)
- Two BPE PROCLIB members
  - Contain trace level and user exit information
  - BPE Configuration Parameters (BPECFG=BPExxxxx)
  - BPE User Exit List
    - EXITMBR= in BPECFG=
      - EXITDEF= in EXITMBR
  - Neither are required
    - Can let BPE configuration parameters default
    - Do not need user exits
  - All IMSplex members can share the BPE PROCLIB members
- All CSL address spaces execute BPEINI00

The BPE module, BPEINI00, must be added to the z/OS PPT if it is not already there.

BPE has two PROCLIB members for itself that contain tracing data and user exit data.

- Configuration parameters are specified via BPECFG=BPExxxxx
- The user exit list is provided by the EXITDEF= parameter in the EXITMBR= parameter in BPECFG=

Neither BPE PROCLIB member is required; rather it is recommended to have them default, at least when first setting up CSL.

All IMSplex members can share the BPE PROCLIB members.

## Some CSL component setup rules

- All IMSplex members can share a PROCLIB
- Each CSL component type (OM, RM, SCI) requires its own
  - Initialization module
  - BPEINIT = CSLxINI0 where x is O/R/S
- Each CSL component address space requires its own
  - Initialization PROCLIB member
  - xxxINIT = yyy
    - where xxx is OM/RM/SCI
    - where yyy is a 3 character suffix used for the CSL initialization PROCLIB member CSLxlyyy
      - where x is O/R/S and yyy is a 3 character suffix

Before going into the details of setting up each CSL member, let's first look at some general setup rules.

- All IMSplex members can share a PROCLIB
- Each CSL component type needs its own initialization module, provided by IMS, specified via BPEINIT=
- Each CSL component address space needs its own initialization PROCLIB member
- Parameter formats are common because all CSL components are based on BPE



## SCI setup

- One SCI address space is required on every z/OS image where CSL is active
- SCI started task JCL found in IMS.PROCLIB (CSLSI)
- Each SCI address space
  - Executes BPEINI00
  - SCI initialization module
    - BPEINIT=CSLSINI0
  - SCI initialization PROCLIB member CSLSIxxx
    - SCIINIT=xxx
    - Each instance of SCI has a different initialization PROCLIB member to specify its unique SCI name (and IMSplex name)

CSLSI is the started task JCL.

It executes BPE.

The SCI initialization module is CSLSINI0.

The SCI initialization PROCLIB member is CSLSIxxx.

## SCI setup – Sample JCL for 2 SCI instances

```

//SCI1 PROC      RGN=0,SOUT=A,RESLIB='IMS.SDFSRESL',
//              BPECFG=BPEPLX0,                <<< BPE configuration parms
//              SCIINIT=001,                    <<< default CSLSIxxx member
//              PARM1=                          << PROCLIB member overrides
//SCIPROC EXEC  PGM=BPEINI00,REGION=&RGN,
//              PARM='BPECFG=&BPECFG,
//              BPEINIT=CSLSINI0,              <<< initialize for SCI
//              SCIINIT=&SCIINIT,&PARM1'        <<< SCI PROCLIB member
//STEPLIB      DD      DSN=&RESLIB,DISP=SHR
// ..

//SCI2 PROC      RGN=0,SOUT=A,RESLIB='IMS.SDFSRESL',
//              BPECFG=BPEPLX0,                <<< BPE configuration parms
//              SCIINIT=002,                    <<< default CSLSIxxx member
//              PARM1=                          <<< PROCLIB member overrides
//SCIPROC EXEC  PGM=BPEINI00,REGION=&RGN,
//              PARM='BPECFG=&BPECFG,
//              BPEINIT=CSLSINI0,              <<< initialize for SCI
//              SCIINIT=&SCIINIT,&PARM1'        <<< SCI PROCLIB member
//STEPLIB      DD      DSN=&RESLIB,DISP=SHR
// ..
  
```

This is an example of the JCL needed for setting up two instances of SCI.

Notice both use the same BPE initialization module CSLSINI0.

Each has a different SCIINIT= value because we have 2 instances of SCI with unique NAME parameter settings (next chart).

Parameters in the SCI initialization PROCLIB member can be overridden by the PARM1= parameter on the EXEC statement.

## SCI setup – CSLSI001 example

<code>ARMRST=Y N</code>	<<<	ARM restart enabled?
<code>SCINAME=SCI1</code>	<<<	SCI Name (SCIid = SCI1SC)  Name must be unique within IMSplex; shows up on messages from this component.
<code>IMSPLEX=(NAME=PLX0)</code>	<<<	IMSplex name = CSLPLX0. IMSplex XCF group name. Name must be same for all CSL address spaces, CQS, and IMS.
<code>FORCE=(ALL,[SHUTDOWN])</code>	<<<	Cleanup ECSA [and shutdown]

This is an example of the SCI initialization member for the first SCI instance, CSLSI001.

SCINAME must be unique in the IMSplex. It is 1-6 characters long, and is used to create the SCIID used in SCI messages, suffixed by 'SC'. For this example, the SCIID is 'SCI1SC'.

The NAME= parameter in IMSPLEX= is used to generate the IMSplex XCF group name, prefixed by 'CSL'. For this example, the XCF group name is 'CSLPLX0'.

## OM Setup

- An OM address space is required on at least one z/OS image where CSL is active
  - Recommend a second OM for backup
- OM started task JCL found in IMS.PROCLIB (CSL~~O~~M)
- Each OM address space
  - Executes BPEINI00
  - OM initialization module
    - BPEINIT=CSL~~O~~INI0
  - OM initialization PROCLIB member CSL~~O~~Ixxx
    - ~~O~~MINIT=xxx
    - Each instance of OM has a different initialization PROCLIB member to specify its unique OM name (and IMSplex name)

CSL~~O~~M is the started task JCL.

It executes BPE.

The OM initialization module is CSL~~O~~INI0.

The OM initialization PROCLIB member is CSL~~O~~Ixxx.

You can see the commonality with SCI setup because CSL uses BPE.

## OM setup – Sample JCL for OM instance

```

//OM1 PROC      RGN=0,SOUT=A, RESLIB='IMS.SDFSRESL',
//              BPECFG=BPEPLX0,                <<< BPE configuration parms
//              OMINIT=001,                    <<< default CSLOIxxx member
//              PARM1=                          << PROCLIB member overrides
//OMPROC EXEC   PGM=BPEINI00,REGION=&RGN,
//              PARM='BPECFG=&BPECFG,
//              BPEINIT=CSLOINI0,              <<< initialize for OM
//              OMINIT=&OMINIT,&PARM1'         <<< OM PROCLIB member
//STEPLIB      DD                                DSN=&RESLIB,DISP=SHR
// ..

```

This is an example of the JCL needed for setting up an OM instance.

Notice the use of the BPE initialization module CSLOINI0.

OMINIT= is used to specify the OM initialization PROCLIB member CSLOIxxx.

Parameters in the OM initialization PROCLIB member can be overridden by the PARM1= parameter on the EXEC statement.

## OM setup – CSLOI001 example

```
ARMRST=Y|N          <<< ARM restart enabled?  
OMNAME=OM1         <<< OM Name (OMid = OM1OM)  
IMSPLEX=(NAME=PLX0) <<< IMSplex name = CSLPLX0  
CMDSEC=N|E|R|A     <<< Command Security Level  
                        None, Exit, RACF, All  
CMDLANG=ENU       <<< US English  
CMDTEXTDSN=IMS.TRANABLE <<< Command syntax translation table
```

This is an example of the OM initialization member CSLOI001.

OMNAME must be unique in the IMSplex. It is 1-6 characters long, and is used to create the OMID used in OM messages, suffixed by 'OM'. For this example, the OMID is 'OM1OM'.

The NAME= parameter in IMSPLEX= is used to generate the IMSplex XCF group name, prefixed by 'CSL'. For this example, the XCF group name is 'CSLPLX0'.

## RM Setup

- At least one RM address space is required in an IMSplex to use RM functions (V9/V10)
  - Recommend a second RM for backup
  - If you do not require RM functions, you can configure without an RM by specifying RMENV=N in your DFSCGxxx PROCLIB member
- One RM address space is required in V8 with CSL
- A resource structure is optional in an IMSplex
  - If no resource structure is defined, you can have only one RM address space
  - If a resource structure is defined, you can have one or more RM address spaces
- RM started task JCL found in IMS.PROCLIB (CSLRM)
- Each RM address space
  - Executes BPEINI00
  - RM initialization module
    - BPEINIT=CSLRINI0
  - RM initialization PROCLIB member CSLRxxxx
    - RMINIT=xxx
    - Each instance of RM has a different initialization PROCLIB member to specify its unique RM name (and IMSplex name)

Most customers may just set up OM initially (for DRD), then do RM later for GOLC/MOLC.

There are more rules for the number of RM address spaces than for OM or SCI.

- in IMS V9 and IMS V10, to use RM you need at least one, preferably two RMs to exploit RM functions; use RMENV=N if no RM
- in IMS V8, always need an RM with CSL

Regarding the resource structure, it is optional

- If no resource structure, then you can only have one RM address space
- If resource structure exists, then you can have one or more RM address spaces

CSLRM is the started task JCL.

It executes BPE.

The RM initialization module is CSLRINI0.

The RM initialization PROCLIB member is CSLRxxxx.

## RM setup – Sample JCL for RM instance

```

//RM1 PROC      RGN=0,SOUT=A, RESLIB='IMS.SDFSRESL',
//              BPECFG=BPEPLX0,                <<< BPE configuration parms
//              RMINIT=001,                    <<< default CSLRIxxx member
//              PARM1=                          <<< PROCLIB member overrides
//RMPROC EXEC   PGM=BPEINI00,REGION=&RGN,
//              PARM='BPECFG=&BPECFG,
//              BPEINIT=CSLRINI0,             <<< initialize for RM
//              RMINIT=&RMINIT,&PARM1'         <<< RM PROCLIB member
//STEPLIB      DD      DSN=&RESLIB,DISP=SHR
// ..
  
```

This is an example of the JCL needed for setting up an RM instance.

Notice the use of the BPE initialization module CSLRINI0.

RMINIT= is used to specify the RM initialization PROCLIB member CSLRIxxx.

Parameters in the RM initialization PROCLIB member can be overridden by the PARM1= parameter on the EXEC statement.



## RM setup – CSLRI001 example

```
ARMRST=Y|N          <<< ARM restart enabled?
RMNAME=RM1          <<< RM Name (RMid = RM1RM)
CQSSN=CQS1          <<< CQS NAME
IMSPLEX=(
  NAME=PLX0,         <<< IMSplex Name = CSLPLX0
  RSRCSTRUCTURE(    <<< Resource Structure
  STRNAME=RSRCSTR1) <<< Name
```

This is an example of the RM initialization PROCLIB member CSLRI001.

RMNAME must be unique in the IMSplex. It is 1-6 characters long, and is used to create the RMID used in RM messages, suffixed by 'RM'. For this example, the RMID is 'RM1OM'.

The NAME= parameter in IMSPLEX= is used to generate the IMSplex XCF group name, prefixed by 'CSL'. For this example, the XCF group name is 'CSLPLX0'.

You also need to specify the CQS name and the resource structure name (if used).

## CQS Setup

- A CQS address space is required on every z/OS image where RM exists and a Resource Structure is being used
  - CQS for RM not required if no RM resource structure
- RM uses CQS services to manage the resource structure
- CQS started task JCL found in IMS.PROCLIB (CQS)
- Each CQS address space
  - Executes BPEINI00 (or optionally CQSINIT0)
  - CQS initialization module
    - BPEINIT=CQSINI00
  - CQS initialization PROCLIB member CQSIPxxx
    - CQSINIT=xxx
    - Each instance of CQS has a different initialization PROCLIB member

CQS setup is somewhat similar since it uses BPE also, though it predates CSL.

A CQS address space is required with an RM address space and a resource structure, since CQS is used by RM to access the resource structure.

CQS is the started task JCL.

It executes BPE or optionally CQSINIT0.

The CQS initialization module is CQSINI00.

The CQS initialization PROCLIB member is CQSIPxxx.

## CQS setup – Sample JCL for CQS instance

```

//CQS1 PROC      RGN=0,SOUT=A, RESLIB='IMS.SDFSRESL',
//              BPERES='IMS.SBPERESL',
//              BPECFG=BPEPLX0,                <<< BPE configuration parms
//              CQSINIT=001,                  <<< default CQSIPxxx member
//              PARM1=                        <<< PROCLIB member overrides
//RMPROC EXEC    PGM=BPEINI00,REGION=&RGN,
//              PARM='BPECFG=&BPECFG,
//              BPEINIT=CQSINI00,            <<< initialize for CQS
//              CQSINIT=&CQSINIT,&PARM1'      <<< CQS PROCLIB member
//STEPLIB        DD      DSN=&RESLIB,DISP=SHR
// ..

```

This is an example of the JCL needed for setting up a CQS instance using BPE (not the older PROC).

Notice the use of the BPE initialization module CQSINI00.

CQSINIT= is used to specify the CQS initialization PROCLIB member CQSIPxxx.

Parameters in the RM initialization PROCLIB member can be overridden by the PARM1= parameter on the EXEC statement.

## CQS Setup ...

- CQS has 3 PROCLIB members
  - CQSIPxxx (unique)
    - Identified by CQSINIT parameter on CQS procedure  
**CQSGROUP=SQGP0,**  
**STRDEFG=000,**  
**STRDEFL=001,**  
**IMSPLEX=(NAME=PLX0)**
  - CQSSGxxx (common to all CQSSs)
    - Identifies name of Resource Structure  
**RSRCSTRUCTURE(STRNAME=RSRCSTR0)**
  - CQSSLxxx (unique)
    - Not applicable to resource structure
    - Not required if this CQS not managing shared queue structures

CQS has more setup. It has up to three PROCLIB members.

CQSIPxxx is the CQS initialization PROCLIB member.

CQSSGxxx is used to specify the resource structure for RM.

CQSSLxxx is used if running shared queues.

## IMS Setup

- Changes in control region JCL only if using Global Online Change
  - Remove local //MODSTAT DD statement
  - Add //OLCSTAT DD statement and data set
- One new PROCLIB member
  - DFSCGxxx (or DFSDFxxx in IMS V10)
- One new execution or DFSPBxxx parameter
  - Identifies DFSCGxxx
  - CSLG=xxx
- Two new DFSVSMxx parameters
  - OCMD and CSLT for CSL tracing
- Several new DFSDCxxx parameters
  - SRMDEF and RCVYxxxx for STM

There are a few changes needed for CSL in IMS.

If using GOLC (or MOLC), you need to replace MODSTAT with OLCSTAT.

DFSCGxxx is a new PROCLIB member for specifying CSL parameters. In IMS V10 you can also use DFSDFxxx. Execution or DFSPBxxx parameter CSLG= specifies the suffix for DFSCGxxx.

DFSVSMxx contains new parameters for CSL tracing, OCMD and CSLT.

DFSDCxxx contains new parameters for STM (Sysplex Terminal Management) for status recovery mode.

## z/OS / CF Setup

- CFRM (if using resource structure)
  - Update CFRM Couple Data Set
    - System managed duplexing, system managed rebuild
  - Define the RM resource structure in the CFRM Policy

**STRUCTURE NAME(RSRCSTR0)**  
**INITSIZE(4096)**  
**SIZE(8192)**  
**MINSIZE(2048)**  
**ALLOWAUTOALT(NO|YES)**  
**DUPLEX(DISABLED|ALLOWED|ENABLED)**  
**PREFLIST(CF01,CF02)**

- Activate policy in sysplex

From the z/OS perspective, you need to do some setup if you are using RM with a resource structure.

The CFRM policy needs to be updated to:

- use system managed duplexing or system managed rebuild
- define the resource structure

Ensure you activate the new CFRM policy.

## Starting the CSL Environment

- The sequence in which components are started is important
  - Start SCI first
    - All IMSplex members register with SCI
  - Start (at least one) OM
    - RM and IMS register with OM
    - SPOC registers with OM
  - Start CQS everywhere there is an RM and structure
    - RM registers with CQS
  - Start (at least one) RM
    - Start only one RM if no resource structure
    - IMS registers with RM
  - Start IMS
    - Nobody registers with IMS
  - Start SPOC
    - Anytime after OM started

The CSL setup is now complete. Next we need to activate the various CSL components. This discussion assumes using both OM and RM.

The sequence in which components are started is very important due to interdependencies with registration. If not followed, various 'waiting for xxx' messages will occur.

- 1) SCI
- 2) OM
- 3) CQS
- 4) RM
- 5) IMS
- 6) SPOC

## Starting the CSL Environment ...

- Start SCI first
  - SCI is required by every member of the IMSplex

**S SCI1**  
**CSL0020I SCI READY SCI1SC**
- Start OM second
  - **S OM1**  
**CSL0020I OM READY OM1OM**
- Start CQS if using a resource structure
  - May be started by IMS if using shared queues
  - Must be started by user if only for RM

**S CQS1**  
**CSL0020I CQS READY**  
**CQS1CQS**

  - After Structure is defined and activated
    - CQS connects to structure
  - After SCI is started
    - CQS registers with SCI
  - Before RM is started
    - RM registers with CQS

The next two charts show the recommended sequence for starting the various CSL components, including the z/OS start commands and CSL responses. with the appropriate CSLID, either the OMID, SCIID, RMID, or CQSID.



## Starting the CSL Environment ...

- Start RM after SCI, OM, CQS
  - S RM1**
  - CSL0020I RM READY RM1RM**
- Start IMS after SCI, OM, and RM
  - IMS registers with all 3
  - S IMS1**
- Start SPOC anytime after OM started

The CSL environment is up and running. You can start using the SPOC to familiarize yourself with the new commands/etc. in preparation for IMS Version 10.

## Shutting down the CSL Environment

- Shutdown CSL after shutdown of IMS control region
  - CSL can stay up if restarting IMS control region soon
- Recommend shutting down CSL as a whole unit
  - To shutdown an entire IMSplex
    - Issue the CSL SHUTDOWN CSLPLEX command with the z/OS MODIFY command interface to any SCI in the IMSplex
      - f scijobname,SHUTDOWN CSLPLEX
  - To shutdown CSL on one z/OS image
    - Issue the CSL SHUTDOWN CSLLCL command with the z/OS MODIFY command interface to the SCI on that z/OS image
      - f scijobname, SHUTDOWN CSLLCL
- You can stop individual components in the IMSplex by issuing the z/OS STOP (P) command to the address space you want to stop

CSL environments for test systems are more likely to be shut down than CSL environments for production systems.

First, shut down IMS, then shut down CSL; however, CSL can stay up if IMS will be restarting soon, for example. It depends on the type of outage that is occurring.

It is recommended to shut down CSL as a unit either for the entire IMSplex or for a particular z/OS image using the CSL SHUTDOWN CSLPLEX or CSL SHUTDOWN CSLLCL commands with the z/OS MODIFY command.

If necessary, individual components of the CSL environment can be taken down using the z/OS STOP (P) command.

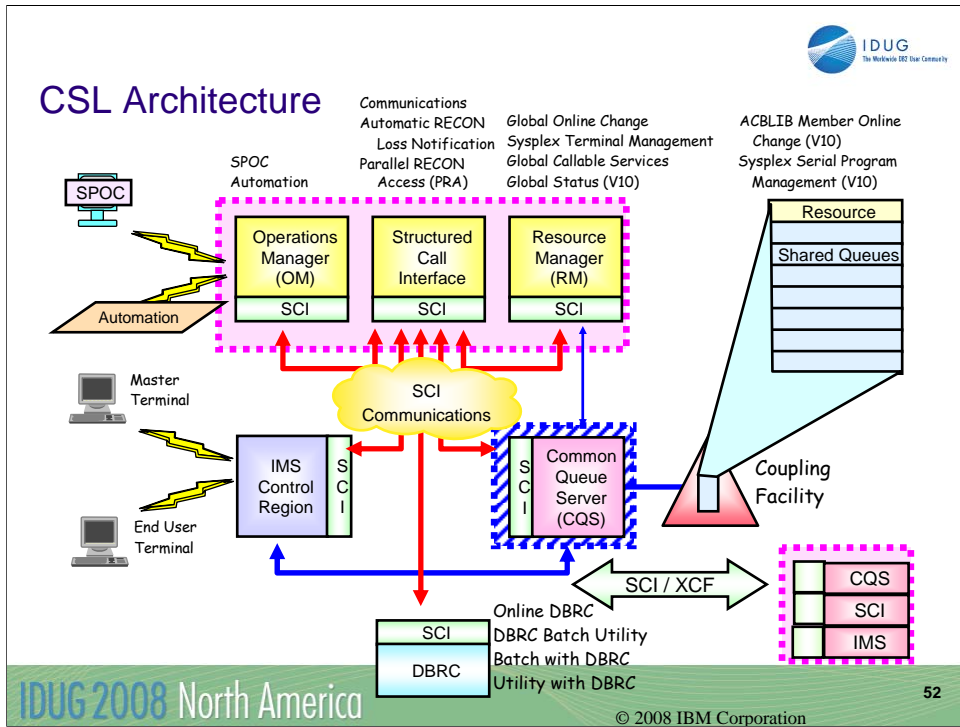
## Common Service (CSL) Summary

- Common Service Layer is part of the evolving IMSplex architecture
  - Required to take advantage of many new IMS V10 functions
- Improvements for Operations Management
- Improvements for Resource Management
  
- Begin implementing CSL now in IMS V9 or IMS V8 so you can be positioned for exploiting new IMS V10 functions

This session has discussed CSL and its components and functions. We've discussed setting up the CSL, and starting and stopping the CSL.

You can begin with IMS V9 or IMS V8 to implement CSL to gain advantages in operations management and resource management, and then be prepared for exploiting new CSL functions in IMS Version 10.

Which components of CSL you will need will depend on the CSL features that you want to use. Perhaps start with OM to be prepared for DRD, then move to RM for ACBLIB Member Online Change.



Completed diagram of the CSL architecture, showing major components, address spaces, and CF structures.

Session J10



Preparing for IMS  
10: Setting Up  
the Common  
Service Layer

Diane Goff

IBM  
dgoff@us.ibm.com