BMC Subsystem Optimizer for zEnterprise Administration Guide

Supporting
Version 3.0 of BMC Subsystem Optimizer for DB2
Version 3.0 of BMC Subsystem Optimizer for IMS
Version 3.0 of BMC Subsystem Optimizer for MQ

December 2016
Contacting BMC Software

Several methods are available for contacting BMC Software.

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>1 713 918 8800</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2103 CITYWEST BLVD</td>
<td>or</td>
<td>or</td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td>1 800 841 2031</td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>+01 713 918 8800</td>
<td>+01 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 2014-2016 BMC Software, Inc.

BMC, BMC Software, the BMC logo, the BMC Software logo, and other BMC marks are the exclusive properties of BMC Software, Inc. and are registered or may be registered in the U.S. and in other countries. BladeLogic and other BladeLogic marks are the exclusive properties of BladeLogic, Inc. and are registered or may be registered in the U.S. and in other countries. All other trademarks or registered trademarks are the property of their respective owners.

Explorer is a trademark or registered trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

CICS, CICSPlex, DB2, IBM, IBM z, IMS, MQSeries, MVS, RACF, WebSphere, z/OS, z10, zEnterprise, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2103 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

Support website
You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

■ Read overviews about support services and programs that BMC offers
■ Find the most current information about BMC products
■ Search a database for problems similar to yours and possible solutions
■ Order or download product documentation
■ Download products and maintenance
■ Report a problem or ask a question
■ Subscribe to receive proactive e-mail alerts
■ Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail
In the United States and Canada, if you need technical support and do not have access to the web, call 1 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:yourSupportContractID, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC
Have the following information available so that Customer Support can begin working on your issue immediately:

■ Product information
  — Product name
  — Product version (release number)
  — License number and password (trial or permanent)
■ Operating system and environment information
  — Machine type
  — Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  — System hardware configuration
  — Serial numbers
  — Related software (database, application, and communication) including type, version, and service pack or maintenance level
■ Sequence of events leading to the problem
■ Commands and options that you used
■ Messages received (and the time and date that you received them)
  — Product error messages
  — Messages from the operating system
  — Messages from related software
License key and password information

If you have questions about your license key or password, contact Customer Support through one of the following methods:

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)
- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.
Contents

About this book ................................................................. 9
Related publications .................................................................. 9
Conventions ............................................................................... 10
Syntax statements ........................................................................ 10
Summary of changes ................................................................... 11

Chapter 1  Overview of BMC Subsystem Optimizer for zEnterprise  13
Why use Subzero? ....................................................................... 14
  What Subzero does .................................................................. 14
  Benefits of Subzero over other solutions .................................. 16
  Example deployment scenarios for Subzero ................................ 18
Subzero architecture ..................................................................... 27
  Summary of the redirection workflow ......................................... 28
  Overview of components and technologies ................................ 29
Subzero requirements ..................................................................... 33
  Hardware requirements ............................................................ 33
  Operating system requirements .................................................. 34
  Subsystem requirements ........................................................... 35
  Subzero security requirements ................................................... 38
  MainView requirements ............................................................ 39

Chapter 2  Subzero considerations and environmental tuning  41
Subzero considerations .............................................................. 41
  Performance considerations ...................................................... 41
  Subsystem access considerations .............................................. 42
  System Management Facilities (SMF) considerations .................... 43
  Considerations for operational procedures and policies .............. 44
Tuning your environment for Subzero .......................................... 45

Chapter 3  Setting up Subzero and associated components  47
Installing and starting components ............................................ 47
Defining Subzero redirections .................................................... 48
  Identifying candidates for redirection ......................................... 49
  Using BMC Cost Analyzer for zEnterprise with Subzero .............. 53
Setting Subzero configuration options in LGC ............................ 54
  Overview of Subzero configuration options in the LGC .............. 56
About this book

This book contains detailed information about the associated product or products. This preface explains the special conventions that the book uses, and how to access related publications.

If applicable, the preface also summarizes the major changes included in the latest release of the product.

Related publications

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:


- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:
  — Documentation Center (primary center and secured center)
  — Support Central (at http://www.bmc.com/support/mainframe-demonstrations)
  — BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)


Products with online interfaces also offer online Help via the F1 key or, for graphical user interfaces (GUIs), via a Help button.
Tip

If you prefer hardcopy documentation, you can order it from your BMC sales representative or from Support Central. Also, from Support Central you can subscribe to receive proactive e-mail alerts when BMC issues notices.

Conventions

This document uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.

- Variable text in path names, system messages, or syntax is displayed in italic text:
  
  `testsyst/instance/fileName`

- Menu sequences use a symbol to convey the sequence. For example, `Actions => Create Test` instructs you to choose the `Create Test` command from the `Actions` menu.

Syntax statements

This topic explains conventions for showing syntax statements.

A sample statement follows:

```
COMMAND KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4={YES | NO} fileName...
```

The following table explains conventions for syntax statements and provides examples:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
</table>
| Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words. | `alias`  
`databaseDirectory`  
`serverHostName` |
Summary of changes

This topic summarizes product changes and enhancements by version number and release date.

Version 3.0.00, December 2016

This release adds the following new features:

- You can now configure the IBM MQ product to communicate with the IBM IMS, DB2, and CICS subsystems on remote LPARs. This feature is available to users with a license for the BMC Subsystem Optimizer for MQ product.

- You can now configure Subzero to connect to alternative target LPARs (up to two), if the primary target LPAR is unavailable.

  You configure alternative targets in the DB2 Product Configuration (LGC) component, and enable or disable this feature by using the BRD, ALTS command.

- The number of threads is no longer limited to 8,192. Subzero can now run an unlimited number of threads, which are dynamically defined according to system requirements.

- Subzero now supports an enhanced infrastructure that allows additional monitoring by the MainView products. Specific enhancements are delivered by the individual MainView products.
This release of BMC Subsystem Optimizer for DB2 supports IBM DB2 Version 12.

**Version 2.0.00, September 2015**

This release adds the following new features:

- You can now configure an IBM MQ queue-sharing group (QSG) to cache shared object definitions on an IBM DB2 subsystem that is running on a remote LPAR.

- All STOP commands now support the FORCE option. For example `/dbcssid BRD,STOP,FORCE` forcibly stops all address spaces on the local LPAR.

- The BRD,DISPLAY,MAP command now also displays any redirectional conflicts such as redirection to unavailable targets, or to unsupported DB2 or IMS subsystems.

Following the 2.0.00 release in September 2015, the following significant features were made available via PTFs accompanying small program enhancements (SPE):

- You can run DB2 and IMS batch jobs on remote LPARs. Specifically, for batch jobs that use the call attachment facility (CAF) to execute DB2 SQL statements, you can now redirect those jobs to a remote DB2 subsystem. You can also specify that the DB2 job steps run on the same remote LPAR, which reduces latency for very large data requests.

- Subzero now lets you specify a naming pattern for Subzero subsystem servers based on the job name of the client application that is using the Subzero connection.

**Version 1.0.00, May 2014**

Following the original release in May 2014, the following significant features were made available via PTFs accompanying small program enhancements (SPE):

- Subzero now provides an installation verification program (IVP) that you can use to ensure that the product is installed and set up correctly in your environment.

- Subzero now supports the IBM IMS Open Thread Environment (OTE). IBM issued APARs to enable OTE for IMS Version 12 (APAR PI29194) and 13 (APAR PI29195).

- Enhanced Subzero messages now provide information that you can use in a BRD,KILL command to terminate the task. These messages identify the waiting or suspended thread and its associated Subzero task number.
Overview of BMC Subsystem Optimizer for zEnterprise

BMC Subsystem Optimizer for zEnterprise (Subzero) products run in an IBM z/OS sysplex environment. Subzero redirects subsystem access requests that are issued by applications on one logical partition (LPAR) to subsystems that are running on a different LPAR.

Subzero replaces supported subsystem access routines, such as those that application programs running under the IBM CICS product use to access IBM DB2 or IBM IMS subsystems or communicate IBM MQ (formerly called IBM WebSphere MQ) messages. Subzero also replaces supported subsystem access routines that MQ uses to communicate with DB2 and IMS subsystems. Subzero products include:

- BMC Subsystem Optimizer for DB2
- BMC Subsystem Optimizer for IMS
- BMC Subsystem Optimizer for MQ

Note
In this document, the term Subzero refers to any BMC Subsystem Optimizer product. The term includes a qualifier (such as for DB2) only when the information is specific to that product.

The following topics provide background information that you need for configuring and using the product:

- “Why use Subzero?” on page 14
- “Subzero architecture” on page 27
- “Subzero requirements” on page 33
Why use Subzero?

By removing the constraint that applications and subsystems must run on the same LPAR, Subzero offers new ways to optimize subsystem deployment in your environment.

What Subzero does

With Subzero redirecting subsystem requests, you can separate subsystems from the applications that need access to those subsystems.

For example, Figure 1 on page 15 illustrates a z/OS sysplex environment without Subzero, where applications and their accessed subsystems must run on the same LPAR. Two LPARs are active, and each one has:

- Applications running under CICS and issuing requests for access to DB2 databases
- A DB2 subsystem that accesses databases, fulfills the application requests, and returns results to the applications
In contrast, Subzero can optimize subsystem deployment by redirecting requests from LPAR 1 to LPAR 2 (Figure 2 on page 16):

- On LPAR 1, applications are running under CICS and issuing requests for access to DB2 databases.
- Subzero redirects those requests to a DB2 subsystem that is running on LPAR 2.
The DB2 subsystem on LPAR 2 accesses databases, fulfills the application requests, and returns results to the applications through Subzero.

Figure 2: Example environment with Subzero

Redirection is completely transparent to the applications and subsystems participating in the redirection: no changes are needed in the application programs or subsystems. Within Subzero, your definitions for redirection are easy to change without affecting system or subsystem operations.

Benefits of Subzero over other solutions

By enabling applications and subsystems to run on different LPARs, Subzero lets you deploy software products on LPARs in a way that:

- Optimizes sub-capacity license fees
■ Expands strategic options for remote subsystem processing and system outages

■ Simplifies system administration tasks because you can eliminate low-usage subsystems through consolidation

Subzero offers the following benefits over other possible solutions:

■ No changes to application programs or CICS transaction definitions are required for redirection. The altered processing environment is completely transparent to the application program and CICS.

■ Subzero supports concurrent, remote connectivity to DB2 and IMS subsystems. MQ, CICS, IMS, and DB2 subsystems can run on different LPARs.

■ Subzero allows connections to remote and local subsystems:
  — Local subsystem connectivity is automatic and is accomplished by using native connectivity interfaces. Subzero searches for the target subsystem on the local LPAR before searching remote LPARs.
  — Backout, fallback, and recovery scenarios for redirection do not require application or CICS changes. If redirection fails, you can simply connect to a local subsystem and continue processing.
  — Remote LPARs do not require CICS regions for subsystem connectivity. Subzero provides a true remote connection to the remote subsystem, not a redundant, mirrored environment that has subsystem connectivity.

■ You maintain your redirection configurations through an ISPF interface. You can alter the configuration dynamically while CICS, MQ (formerly called IBM WebSphere MQ), IMS, and DB2 environments are active.

■ Subzero client interface tasks and subsystem server address spaces automatically start and stop as needed to support redirection requirements. No operational changes are required for these elements.

■ You can configure an IMS subsystem to perform SQL transactions on a DB2 subsystem running on a remote LPAR.

■ You can now configure an IBM MQ subsystem to communicate with a DB2, IMS, or CICS subsystem on a remote LPAR.

■ You can now configure a queue-sharing group (QSG) for the IBM MQ product to cache shared object definitions on an IBM DB2 subsystem that is running on a remote LPAR.
You can configure IMS batch message processing (BMP) regions to access DB2 databases remotely.

Example deployment scenarios for Subzero

The following scenarios illustrate how you might deploy Subzero to reduce subsystem proliferation, expand flexibility, and simplify administration tasks in the environment:

- “Separating DB2 subsystems from CICS subsystems” on page 19
- “Consolidating IMS subsystems on a single LPAR” on page 20
- “Reducing an IMS footprint” on page 20
- “Separating high-volume activity from low-volume activity” on page 21
- “Consolidating subsystems in a test environment” on page 22
- “IMS subsystems calling DB2 subsystems on a remote LPAR” on page 24
- “MQ caching shared object definitions on DB2” on page 24
Separating DB2 subsystems from CICS subsystems

Assume that you have two DB2 and two CICS subsystems on one LPAR, and the same setup on a second LPAR. Deploying Subzero lets you separate the DB2 and CICS subsystems by moving the DB2 subsystems to a third LPAR.

Figure 3: Deployment example—DB2 separated from CICS
Consolidating IMS subsystems on a single LPAR

For this scenario, assume that IMS is deployed on three LPARs. Deploying Subzero lets you consolidate IMS on one LPAR.

Figure 4: Deployment example—IMS on a single LPAR

Reducing an IMS footprint

For this scenario, assume that:

- IMS is deployed on three LPARs
- The IMS activity is relatively low in comparison to the DB2 activity on LPARs 1 and 3
- Overall activity is low on LPAR 2
Deploying Subzero lets you reduce the IMS footprint to a single LPAR without changing the high-volume DB2 subsystems.

**Figure 5: Deployment example—Reduced IMS footprint**

---

**Separating high-volume activity from low-volume activity**

Assume that DB2 and IMS are each deployed on three LPARs. Activity demands are mixed, with some high-volume activity and some low volume.

Deploying Subzero lets you improve performance by:

- Handling high-volume DB2 activity on a separate LPAR (LPAR 1 in Figure 6 on page 22)

- Handling high-volume IMS activity on a separate LPAR (LPAR 3)
- Combining low-volume IMS and DB2 activity on a single LPAR (LPAR 2)

**Figure 6: Deployment example—Volume-based separation of activities**

**Consolidating subsystems in a test environment**

Assume that you have an environment for testing applications, where:

- Volume and performance are not important
- The test environment has three LPARs
- Each LPAR deploys CICS, IMS, and DB2
Deploying Subzero lets you use a single LPAR for CICS, one for IMS, and one for DB2.

**Figure 7: Deployment example—Consolidation in a test environment**
**IMS subsystems calling DB2 subsystems on a remote LPAR**

Assume that you have two IMS and two DB2 subsystems on one LPAR, and the same setup on a second LPAR. Deploying Subzero lets you separate the IMS and DB2 subsystems by moving the DB2 subsystems to a third LPAR.

**Figure 8: Deployment example—IMS subsystems calling DB2 subsystems on a remote LPAR**

**MQ caching shared object definitions on DB2**

With Subzero, you can now configure a queue-sharing group (QSG) for the IBM MQ product (formerly called IBM WebSphere MQ) to cache shared object definitions on an IBM DB2 subsystem that is running on another LPAR.
Note
Previously the DB2 subsystem had to run on the same LPAR as MQ. As a result, if you were subject to variable workload license charges, you would pay for DB2 at the peak rate for the LPAR.

For example, assume that you have two MQ subsystems that are using separate DB2 subsystems to share object definitions on one LPAR, and the same setup on a second LPAR. Subzero now lets you separate the MQ and DB2 subsystems by moving the DB2 subsystems to a third LPAR.
Figure 9: Deployment example—MQ caching shared object definitions on DB2
Subzero architecture

Subzero uses a set of components and supporting technologies to redirect subsystem requests in the z/OS environment.

Figure 10 on page 27 shows the interaction between components and technologies that support Subzero:

- DB2 Component Services (DBC)
- DB2 Product Configuration (LGC)
- Runtime Component System (RTCS)
- Subzero controlling agent
- Subzero subsystem server

Note
In this figure, the applications running under CICS and accessing a DB2 subsystem are an example of one type of subsystem access that Subzero supports.
The following topics explain how Subzero works:

- “Summary of the redirection workflow” on page 28 summarizes the process for completing a redirection, specifying which component or technology is involved in each step.

- “Overview of components and technologies” on page 29 describes how the components and technologies work.

Summary of the redirection workflow

The following process summarizes how Subzero completes a redirection, specifying which component or technology is involved in each step.

1. A DB2 Component Services (DBC) address space starts on each z/OS logical partition (LPAR) that will participate in Subzero redirection.

2. Each DBC starts a Subzero controlling agent on the same LPAR.

3. One controlling agent (internally designated as the master) requests the current Subzero option set from DB2 Product Configuration (LGC). The option set identifies the participants in Subzero redirection activities as clients and targets. A client is the subsystem access routine and LPAR to which applications are sending requests for subsystem access. A target is the subsystem and LPAR to which Subzero will redirect those requests.

4. LGC obtains the option set from its datastore, which is stored in a Runtime Component System (RTCS) registry. LGC returns the option set to the master controlling agent, which shares the configuration information with the other controlling agents.

5. If the option set identifies that a client is on an LPAR, the controlling agent on that LPAR initializes a task to handle that client.

6. An application program issues the first subsystem access request as usual. (No application program changes are required.)

7. The first request is routed to the Subzero controlling agent on the client LPAR.

8. The controlling agent on the client LPAR receives the first request, and sends it to the controlling agent on the target LPAR.

9. In response to the first request, the target controlling agent initializes a Subzero subsystem server on the target LPAR, and passes the request to that new server.
10 The subsystem server establishes a connection with the target subsystem, which fulfills the request and returns the results to the subsystem server.

To the target subsystem, the subsystem server appears to be an ordinary subsystem access routine. (No target subsystem changes are required.)

11 The subsystem server establishes an XCF connection with the client on the client LPAR, and sends the fulfilled response to the client.

To the client, the subsystem server appears to be an ordinary subsystem access routine. From this point, the client and the subsystem server communicate directly with each other through XCF. DBC and the controlling agents no longer participate.

12 The application program receives the results of the request.

Overview of components and technologies

Subzero uses the following components and technologies:

- “DB2 Component Services (DBC)” on page 29
- “DB2 Product Configuration (LGC) and other configuration components” on page 30
- “Subzero controlling agent” on page 31
- “Subzero subsystem servers” on page 32
- “Recovery and diagnostic facilities” on page 32

DB2 Component Services (DBC)

The DB2 Component Services (DBC) component provides an infrastructure in which Subzero can initialize its own product services and obtain DBC-provided services.

DBC runs as a persistent address space in each LPAR that is participating in Subzero redirection activities. DBC provides the following major services for Subzero:

- Initializes the Subzero controlling agent, based on startup instructions in a script that was customized during installation and stored in the DBC repository
- Supports Subzero console commands that you can use to control Subzero servers manually
- Coordinates activities and communicates with other active DBC address spaces throughout the sysplex
Provides information to Subzero about subsystems that are participating in redirection, such as
— The LPARs on which the subsystems are running
— Libraries required for access
— When the subsystems start and stop

In most production environments, DBC is executed as a z/OS started task. Typically, the operating system starts the task automatically from a JCL procedure in the system procedure library. In a trial or test environment, you can start DBC manually as a batch job.

**Note**

DBC must start before any CICS regions that are participating in Subzero redirection.

DBC is shared with other BMC products in the environment, as applicable.

For detailed information about DBC, see the *BMC Infrastructure Components Administration Guide*.

**DB2 Product Configuration (LGC) and other configuration components**

DB2 Product Configuration (LGC) manages the configuration values that you establish for Subzero.

LGC provides a simple ISPF interface for you to specify configuration values. These values control how Subzero redirects traffic between identified client ("from" or source) redirection participants and target ("to" or destination) participants.

LGC uses a Runtime Component System (RTCS) registry to store and access your specified configuration values; this set of values is called an option set. LGC communicates with other LGC agents in the sysplex environment to manage its datastore.

LGC runs as an agent within DB2 Component Services (DBC). The master Subzero controlling agent communicates with LGC to obtain the latest Subzero configuration values. The controlling agents use these values to initialize Subzero tasks and servers as needed for redirection.

LGC and RTCS are shared with other BMC products in the environment, as applicable.

For detailed information about LGC, see the *BMC Infrastructure Components Administration Guide*.
Subzero controlling agent

The Subzero controlling agent executes on every LPAR that is participating in Subzero redirection. The controlling agent performs the following tasks:

- Communicates with DBC
- Communicates with other Subzero controlling agents
- If internally designated as the master, communicates with LGC to obtain current Subzero configuration values, and shares them with other Subzero controlling agents
- Automatically interfaces with clients on this LPAR as needed, based on the defined configuration values
  
  If configuration values define any potential clients on this LPAR, the controlling agent prepares to redirect subsystem access requests to the appropriate target subsystem on a remote LPAR. The controlling agent serves all clients on this LPAR.
  
  The controlling agent establishes a Subzero cross-memory environment and cross-system coupling facility (XCF) communication between the client address space and the remote subsystem, as follows:
  
  — Creates a reusable linkage index (LX) within the client address space
  
  — Allocates approximately 250 KB in the extended common service area (ECSA) for executable cross-memory routines, and approximately 4 MB of 64-bit common memory objects for XCF message buffers

---

*Note*

After establishing the communications environment, the controlling agent is no longer involved in redirection; therefore, it consumes minimal CPU resources. The controlling agent terminates automatically when it is no longer needed.

- Automatically starts and stops Subzero subsystem servers as needed, when a controlling agent on a client LPAR requests access to a subsystem that is participating in redirection

DBC starts the controlling agent automatically as a separate address space. The controlling agent runs as an external agent under the control of DBC. The default address space job name of a controlling agent is BRDCNTL. All of the Subzero controlling agents that are running on the different LPARs in the sysplex communicate with each other.

Subzero controlling agents and subsystem servers share a common code base, simplifying operations and maintenance.
Subzero subsystem servers

A Subzero subsystem server receives redirected subsystem access requests from a client, communicates with the target subsystem, and returns request results to the client.

When the Subzero controlling agent on a target LPAR receives the first redirected request to a target subsystem, the controlling agent starts a subsystem server automatically. The subsystem server receives the request and connects to the target subsystem. The target subsystem sends the fulfilled request to the subsystem server. The subsystem server sends the fulfilled request directly to the client through XCF. The client sends subsequent access requests directly to the subsystem server.

**Note**

If the target subsystem is unavailable and you have defined alternative targets, Subzero connects to an alternative target subsystem.

The controlling agent terminates a subsystem server automatically when the server is no longer needed (such as when the corresponding subsystem has terminated). To control a subsystem server manually, you can issue console commands.

A subsystem server runs as a separate address space on the target LPAR. DBC manages the address space as an external agent. One subsystem server is required to serve each defined redirection relationship between a client (running on the client LPAR) and a target subsystem (running on the target LPAR).

The default address space job name of the subsystem server is BRDssid, where ssid is the subsystem ID of the target subsystem. A subsystem server address space uses minimal system resources, limited to those necessary to enable access to the remote subsystem. It does not use ECSA or common memory objects and does not create any cross-memory environments.

Recovery and diagnostic facilities

Extensive recovery and diagnostic facilities ensure that all Subzero components handle problems correctly and provide robust information about events and actions.

Subzero uses the following diagnostic and tracing facilities. For more information, see “Troubleshooting Subzero problems” on page 127.
### Facility | Description
--- | ---
Processing log and diagnostic tracing | Subzero servers write messages to the processing log in response to:
- Major events in the environment (such as established and lost connections and error conditions)
- Internal and external commands
You can enable diagnostic tracing to obtain more detailed information. The BRDPRINT data set (in the DBC address space) contains the processing log. The BRDTRACE and BRDSVLOG data sets (in the Subzero address space) contain diagnostic trace data.

GTF process tracing | You can enable Subzero components to generate generalized trace facility (GTF) tracing records. These records show the process flow through the various client and subsystem server routines.
**Note:** GTF tracing increases overhead; enable it only at the request of BMC Customer Support.

Abend recovery | The following abend recovery services produce an SVC dump and terminate cleanly with an error message or a condition code that is passed to another routine:
- The associated recovery routine (ARR) provides recovery services for cross-memory routines.
- The functional recovery routine (FRR) provides recovery services for SRB routines.
- The Extended Specify Task Abnormal Exit (ESTAE) service provides recovery services for client and subsystem servers.

---

**Subzero requirements**

The following topics describe required and optional elements for Subzero support in your environment.

**Hardware requirements**

Subzero executes on the IBM zSeries hardware platform and requires an IBM z10 Enterprise class (or later) system.
Subzero depends on coupling facility resources to accomplish product objectives. Your coupling facility hardware might need to be adjusted or upgraded. For more information, see “Tuning your environment for Subzero” on page 45.

**Operating system requirements**

Subzero operates across a single basic or parallel sysplex environment. Subzero requires z/OS Version 1.12 or later, including the following z/OS resources:

- Linkage Index (LX) Reuse Facility
- 64-bit Pause Services
- Long Displacement Facility
- Execute Immediate Facility
- Execute Extensions Facility

For the latest information about the compatibility of Subzero with a specific release of z/OS, use the BMC Solution and Product Availability and Compatibility Utility on the BMC Support Central website (http://www.bmc.com/support).

**Note**

You might need to tune your z/OS environment to accommodate Subzero requirements. For more information, see “Tuning your environment for Subzero” on page 45.

**Address space requirements**

Subzero requires a set of address spaces to perform product functions in the z/OS environment:

<table>
<thead>
<tr>
<th>Address Space</th>
<th>Subzero requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Component Services (DBC)</td>
<td>One DBC address space running on each LPAR that is participating in Subzero redirection activities</td>
</tr>
<tr>
<td></td>
<td>This address space typically runs as a persistent started task.</td>
</tr>
<tr>
<td>Subzero controlling agent</td>
<td>One controlling agent address space running on each LPAR that is participating in Subzero redirection activities</td>
</tr>
<tr>
<td></td>
<td>The DBC address space initializes the controlling agent automatically, based on the configuration information in the Subzero option set.</td>
</tr>
</tbody>
</table>
## Address Space

| Subzero subsystem server | One subsystem server address space running for each target subsystem (such as DB2, MQ, or IMS) address space that is participating in Subzero redirection activities. The subsystem server runs on the LPAR that is running the target subsystem. The DBC address space initializes a subsystem server address space automatically when a controlling agent address space attempts to connect to the target subsystem. |

## XCF requirements

Subzero uses the cross-system coupling facility (XCF) of z/OS to communicate with product and associated components across different logical partitions (LPARs) in a sysplex environment.

The processing group name that uniquely identifies a set of Subzero servers is also used as the name of the XCF group that those servers join. The processing group name is defined in the Subzero option set.

## Subsystem requirements

Subzero supports the following types of subsystem access:

- Application programs that run under CICS and issue calls for access to DB2 databases
- Application programs that run under CICS and issue calls for access to IMS databases
- Application programs that run under CICS and issue calls for access to MQ messages
- IMS subsystems that issue calls to a DB2 subsystem
- Application programs that use the instrumentation facility interface (IFI) to issue commands to DB2 subsystems
- A queue-sharing group (QSG) for IBM MQ that caches and accesses shared object definitions on a DB2 subsystem

An application can use multiple types of subsystem access routines.

The following topics describe requirements for the subsystems that Subzero supports:
CICS requirements

Subzero can work with the IBM CICS product acting as a transaction manager for application programs that are accessing supported types of subsystems (such as DB2 or IMS).

Subzero supports CICS Version 4.1 and later. For the latest information about the compatibility of Subzero with a specific release of CICS, use the BMC Solution and Product Availability and Compatibility Utility on the BMC Support Central website (http://www.bmc.com/support).

Subzero requires the following modifications to the CICS environment:

- You must add the Subzero product library to the CICS startup JCL.
- You must add the Subzero BRDPLT00 program to the CICS program list table (PLT).

\[\text{Note}\]

DBC and the Subzero controlling agent must be executing on the LPAR before a CICS region that is participating in Subzero redirection starts on that LPAR.

For more information, see “Configuring CICS for Subzero” on page 60.

Subzero allows CICS regions to connect to local or remote subsystems:

- When connecting to local subsystems, Subzero turns control over to the native subsystem interface.
- When connecting to remote subsystems, Subzero replaces the native subsystem interface with the Subzero interface. In this case, Subzero does not use or require any native subsystem interface routines.
- To switch between remote and local connectivity, simply disconnect from the current subsystem and reconnect by using a different subsystem name.
Note
Before the reconnection, you might need to change the Subzero redirection option to specify a different target subsystem.

DB2 requirements

To use Subzero as an interface between a queue-sharing group (QSG) for IBM MQ and IBM DB2, between DB2 and CICS, or between IBM IMS and DB2, you must have a license for the BMC Subsystem Optimizer for DB2 product.

Subzero supports DB2 Version 10.1 and later. For the latest information about the compatibility of Subzero with a specific release of DB2, use the BMC Solution and Product Availability and Compatibility Utility on the BMC Support Central website (http://www.bmc.com/support).

Subzero supports the following types of application program calls to DB2 databases:

- Structured Query Language (SQL)
  Subzero supports all SQL calls that are normally available within the application execution environment (such as CICS).

- Instrumentation Facility Interface (IFI)
  Subzero supports most IFI calls—including COMMAND, WRITE, and READS (synchronous read)—that are normally available within the application execution environment (such as CICS).
  Subzero does not support READA (asynchronous read) because this call requires the application to issue a z/OS WAIT request. CICS application programs must avoid WAIT requests.

Note
If you are using the DB2 group attach facility to define the connection between CICS and DB2, you can specify the defined DB2 group ID in a Subzero option set. The group ID is valid for the original (source) subsystem, the target (remote) subsystem, or both.

IMS requirements

To use Subzero as an interface between IBM CICS and IBM IMS, you must have a license for the BMC Subsystem Optimizer for IMS product. To use Subzero as an interface between IBM IMS and IBM DB2, you must have a license for the BMC Subsystem Optimizer for DB2 product.

Subzero supports IMS Version 12.1 and later. For the latest information about the compatibility of Subzero with a specific release of IMS, use the BMC Solution and

Subzero supports all Data Language I (DL/I) application program calls to IMS databases. Subzero supports all DL/I calls that are normally available within the application execution environment (such as CICS).

When a client (such as a CICS region) makes a connection to an IMS system, a Subzero subsystem server address space starts on the remote LPAR to work with that connection. This Subzero address space requires that the DFSPZPxx module, which contains startup parameters for the IMS Database Resource Adapter (DRA), can be loaded into memory. The system attempts to load the required DFSPZPxx member from the following sources:

1. IMS RESLIB specification within the DFSPZPxx member itself
2. STEPLIB or JOBLIB concatenation in the Subzero startup procedure (BRDPROC)
3. System linklist or link pack area (LPA)

You can ensure that DFSPZPxx can be loaded by modifying the STEPLIB in BRDPROC to include the library that contains DFSPZPxx. If the client contains an IMS RESLIB specification that provides only the DFSPZPxx ddname (the IMS RESLIB data set name is null), specify that same ddname in BRDPROC.

MQ requirements

To use Subzero as an interface between IBM MQ and IBM DB2, IBM IMS, or IBM CICS, you must have a license for the BMC Subsystem Optimizer for MQ product.

Subzero supports MQ Versions 7.00, 7.10, 8.00, and 9.00.

For the latest information about the compatibility of Subzero with a specific release of MQ, use the BMC Solution and Product Availability and Compatibility Utility on the BMC Support Central website (http://www.bmc.com/support).

Subzero security requirements

Subzero works with existing APF and SAF facilities for security and access control, as explained in the following topics:

- “APF authorization requirements” on page 39
- “SAF requirements” on page 39
Once the DBC and its agents are running in your environment, you must perform the following tasks:

- Grant the DBC authority to start and stop the DBC address spaces.
- Define the address spaces as started tasks.

For more information, see *BMC Infrastructure Components Administration Guide*.

**APF authorization requirements**

Subzero and its associated address spaces must start in an APF-authorized state.

All libraries in the STEPLIB concatenation of the following components must be APF authorized:

- DB2 Component Services (DBC) address space
- Subzero procedure (BRDPROC)

Other components, including Subzero client and subsystem servers, inherit DBC’s authorization to perform their authorized functions.

**SAF requirements**

You can secure Subzero and associated components by using your standard external security manager, such as the IBM RACF product, through the System Authorization Facility (SAF).

For more information, see “Securing Subzero components with SAF” on page 73.

**MainView requirements**

You can use BMC MainView products to obtain information about Subzero activities in your environment. Subzero works with or without MainView.

Subzero and its redirection activities are visible in views that the following BMC MainView products produce:

- MainView *for CICS*
- MainView *for DB2*
- MainView for DBCTL
- MainView *for IMS*Online
Certain configuration steps are required for these MainView products to work with Subzero. For more information about these steps and about viewing Subzero information in MainView views, see “Monitoring Subzero components and activities with MainView” on page 139. For general information about using MainView products, see the MainView user documentation.

**Note**

The MainView for z/OS product displays information about XCF members that Subzero uses. You can follow hyperlinks to this information from views in other MainView products that display information about Subzero components. No configuration steps are needed to use Subzero with MainView for z/OS.

Subzero and MainView products might require maintenance (PTFs) to work correctly with each other. For information about required maintenance, see the BMC Subsystem Optimizer for zEnterprise release notes.
Subzero considerations and environmental tuning

The following topics highlight important information about Subzero interactions with other elements in your environment, and explain how to tune your environment to better support Subzero redirection:

- “Subzero considerations” on page 41
- “Tuning your environment for Subzero” on page 45

Subzero considerations

Before implementing Subzero, you should be aware of special considerations that might apply in the following areas:

- “Performance considerations” on page 41
- “Subsystem access considerations” on page 42
- “System Management Facilities (SMF) considerations” on page 43
- “Considerations for operational procedures and policies” on page 44

Performance considerations

Subzero might extend the overall response time for subsystem access requests; transferring a subsystem access request through XCF to remote LPARs naturally takes longer than accessing subsystems with local, cross-memory techniques.

Therefore, you should consider the potential performance effects of redirection in the following areas:
- Locking
  Database locks might be held longer while waiting for a single transaction to complete all of its database interaction and commit any updates.

- Transaction queuing
  Transaction residency might be higher due to the elongated subsystem access, and transaction queues can become deeper. More client address spaces might be required to provide better transaction balancing and throughput.

- Performance monitoring
  Subzero is transparent to the client (such as applications that run under CICS) and the target (remote) subsystem. In a monitoring product, the client and target subsystems reflect connectivity to each other. However, for hyperlinks to work between the client and subsystem monitors, the monitors must recognize that the hyperlink target is running on a remote LPAR. Attempting to connect the monitor to a Subzero server causes errors: the control block structures that the monitor analyzes do not recognize the Subzero internal control blocks.
  MainView products provide maintenance to recognize and support a Subzero environment.

  Tuning efforts can mitigate performance issues. For more information, see “Tuning your environment for Subzero” on page 45.

### Subsystem access considerations

Subzero supports standard techniques for application programs to interact with supported subsystems. However, applications that use nonstandard or undocumented techniques for performing routine functions within the subsystem might need modifications to work correctly with Subzero.

- Subzero supports only publicly documented access to subsystems. No cross-memory connection exists between the client address space (such as CICS) and the subsystem.
  Any attempt to access control blocks in either environment from the opposing environment is likely to fail.

- Subzero does not attempt to emulate environments or simulate functions other than those required to process standard subsystem initialize, read, update, and terminate functions.
  Subzero does not ensure the integrity of control block pointers, except for those required for standard, documented subsystem access.
System Management Facilities (SMF) considerations

If you use SMF data to account for resource consumption, you might need to combine SMF data for accurate information when Subzero redirection occurs.

When a client address space (such as CICS) requests a connection to a target subsystem, Subzero starts a subsystem server address space on the target LPAR. This subsystem server is dedicated to processing requests from that client; a one-to-one relationship exists between the client address space and the subsystem server that Subzero starts on behalf of that client.

While initializing the server, Subzero propagates all Workload Manager (WLM) job accounting, performance monitoring, delay monitoring, and classification data from its associated client address space.

SMF data that z/OS generates for a Subzero subsystem server mirrors the SMF data of the client address space. For example, SMF 30 job accounting records for the subsystem server contain the same accounting information as the records for the client address space. However, the performance and resource data are specific to the subsystem server.

SMF data in an environment with CICS and DB2

When CICS monitoring is enabled, CICS generates SMF 110 records. In a Subzero environment, the SMF 110 subtype 1 record understates CPU utilization for DB2 activity.

In a locally connected CICS and DB2 environment in which Subzero is not active, the DB2 Open Thread Environment (OTE) charges its time directly back to the CICS task control block (TCB). (OTE is available with CICS Version 2.2 and DB2 Version 6 and later.) As a result, CICS generates SMF 110 subtype 1 records that reflect the total CICS and DB2 CPU time.

In a remotely connected CICS and DB2 environment in which Subzero is active, the CICS TCB is suspended while the remote database access request is processed. Because no direct connection exists to the DB2 thread, CICS generates SMF 110 subtype 1 records that reflect only the CICS CPU time and no DB2 CPU time. However, Subzero passes the CICS Correlator ID to DB2. This action allows DB2 to generate the SMF 101 record with the required data for matching that record with the SMF 110 record that CICS produces.

For accurate accounting and capacity management reporting, you will need to combine the SMF 101 data produced by DB2 with the SMF 110 data produced by CICS.
SMF data in a CICS and IMS environment

In a locally connected CICS and IMS environment in which Subzero is not active, IMS passes statistical information for each PSB scheduled and unscheduled interval. CICS provides this information to various exits and maintains internal statistics for downstream reporting.

In a remotely connected CICS and IMS environment in which Subzero is active, Subzero passes this statistical information from the remote IMS system to its client CICS region; this approach allows CICS and all of its exits and downstream reporting to function normally.

If the appropriate PTFs are applied, MainView for IMS generates the FA and F9 IMS log records with the proper IMS accounting and performance data.

Considerations for operational procedures and policies

Using Subzero might result in changes to existing operational procedures and policies.

When you implement Subzero in a production environment, you might want to:

■ Consolidate and eliminate instances of subsystems
■ Create or delete LPARS
■ Quickly redirect subsystem access requests to different subsystems to respond to outages or remote-processing requirements

These activities might require changes to the following elements in your environment:

■ Operational procedures (for example, starting DBC components and Subzero before starting CICS regions that require redirection services)
■ Automated processes
■ Disaster recovery policies
■ Change management processes and policies
■ Documentation
Tuning your environment for Subzero

Tuning is not important in a test environment unless you are evaluating performance or notice a performance problem. However, you should evaluate and tune your environment as needed before implementing Subzero in production.

You can contact BMC Customer Support to discuss any concerns and how to address them.

Preparing your z/OS environment for Subzero

Subzero redirection activities can increase the load on a z/OS environment. You might need to adjust the environment to ensure that adequate resources are available. Shifting subsystem workloads from one LPAR to another might also shift operating system resources that process client application programs.

Assume that you shift the millions of service units (MSUs) required to process CICS transactions on LPAR 1. As a result, the MSUs required to process the associated subsystem access requests might shift to LPAR 2.

Subzero transports subsystem access requests and their related responses across LPARs. Consequently, the transported data is duplicated across the system, and resides in the memory of both the client and target LPARs. To eliminate stress on memory resources, the Subzero controlling agent acquires data storage for its largest blocks above the bar, in 64-bit common memory objects. Although above-the-bar storage is theoretically inexhaustible, you must assign sufficient system paging packs to manage this storage.

To prepare a z/OS system for Subzero

1. Evaluate the number of volumes that are assigned to your paging subsystem, and increase the number if needed.

2. Evaluate your CPU assignment percentages, and adjust as needed for possible shifts in resource usage.

Tuning XCF components for Subzero

Subzero redirections use cross-system coupling facility (XCF) resources. Setup XCF using standard procedures and parameters to ensure the functionality of Subzero. Contact BMC Customer Support for advice.
Setting up Subzero and associated components

Setting up Subzero for use in your environment is a straightforward process of installing and starting product components, specifying redirection options, and making a few changes as needed in the environment.

When you complete the following tasks, Subzero will be ready to redirect subsystem access requests:

- “Installing and starting components” on page 47
- “Defining Subzero redirections” on page 48
- “Configuring CICS for Subzero” on page 60
- “Securing Subzero components with SAF” on page 73
- “Setting up IMS or DB2 batch job redirection” on page 75
- “Configuring MQ for redirection” on page 66

Installing and starting components

You use the standard BMC Installation System to install and start Subzero components. The Installation System is an ISPF application that performs the following tasks:

- Obtains the compressed base installation files from BMC, decompresses them, and installs them as the Subzero product load library on your z/OS system

**Note**

BMC neither supports nor recommends installing the installation of the Subzero product load library in partitioned data set extended (PDSE) data sets.
■ Collects information about your environment and the products that you select
   Using this information, the Installation System creates installation jobs that use IBM SMP/E facilities and conventions to establish FMIDs on your system.

■ Collects information for setting up product components
   Using this information, the Installation System creates jobs that define, start, and initialize required components.

■ Creates JCL to execute the Product Authorization utility, which installs CPU authorization passwords, also known as product license keys.

For complete information about performing these tasks, see the following documents:

■ Installation System Reference Manual
■ Installation System Quick Start
■ Installation System release notes
■ BMC Subsystem Optimizer for zEnterprise release notes
■ BMC Infrastructure Components Administration Guide

Once the DBC and its agents are running in your environment, you must perform the following tasks:

■ Grant the DBC authority to start and stop the DBC address spaces.
■ Define the address spaces as started tasks.

For information about setting up DBC and its agents to work with Subzero, view the Quick Course "Subsystem Optimizer for zEnterprise - Setting up procs for execution."

You can now set up Subzero for redirection.

Defining Subzero redirections

To define how Subzero performs redirections, you identify the candidates to participate as clients and targets. You then specify information about those candidates in a Subzero option set.

For more information, view the Quick Course "Subsystem Optimizer for zEnterprise - Redirecting CICS to Remote Subsystems."
The following topics explain how to define Subzero redirections:

- “Identifying candidates for redirection” on page 49
- “Setting Subzero configuration options in LGC” on page 54
- “Overview of Subzero configuration options in the LGC” on page 56

Identifying candidates for redirection

Before you can configure Subzero, you must decide which clients and targets should participate in redirection activities. Clients are the source (original, local, or "from") applications, subsystems, and LPARs; targets are the remote (destination, or "to") subsystems and LPARs.

Although this topic offers a procedure to help you identify candidates for redirection, you can take any approach that works for you. Some general guidelines follow:

- Your approach is likely to vary depending on whether you are configuring a test environment or a production environment:
  
  — In an early test environment, your candidate choices might focus on simplicity, convenience, and proof that Subzero works with representative elements in your production environment.
  
  — In a later test environment, your choices might focus on "stress" (throughput and capacity) testing, performance, and operational issues.

  Note

  You can also deploy Subzero in a test environment not to test Subzero itself, but to reduce the test environment's administrative complexity and costs. In this case, you should choose candidates as you would for a production environment.

  — In a production environment, your choices might focus on balancing cost savings and operational flexibility against performance needs.

- You can choose specific LPARs by using complete system IDs, or specify the LPARs generically by using the asterisk (*) wildcard character. While identifying candidates, consider developing a consistent approach for specific and generic specifications.

- Multiple redirection definitions can specify the same subsystem ID for the client or target subsystem, some with wildcard characters and others with specific
subsystem IDs. In this case, Subzero uses the definition that has the most specific subsystem ID.

You can use the `/dbc BRD,DISPLAY,MAP` console command to see the results of generic specifications.

For example: the option set contains the following definitions:

<table>
<thead>
<tr>
<th>IMS Subsystems To Redirect</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ IMS1 on LPAR *_____ will redirect to IMID on LPAR *_____</td>
</tr>
<tr>
<td>+ IMS1 on LPAR SY*_____ will redirect to IMIB on LPAR *_____</td>
</tr>
<tr>
<td>+ IMS2 on LPAR SYS*_____ will redirect to IMS on LPAR *_____</td>
</tr>
<tr>
<td>+ IMS3 on LPAR SYSA_____ will redirect to IMSA on LPAR *_____</td>
</tr>
<tr>
<td>+ IMS4 on LPAR SYSB_____ will redirect to IMIB on LPAR *_____</td>
</tr>
</tbody>
</table>

Redirection occurs as follows:

— IMS1 on LPAR BOB redirects to IMID
— IMS1 on LPAR SYT1 redirects to IMIB.
— IMS2 on LPAR SYSC redirects to IMIS.
— IMS3 on LPAR SYSA redirects to IMSA.
— IMS4 on LPAR SYSB redirects to IMIB.

To identify candidates for redirection

1. Start by making a list of potential candidates to participate in redirection:

<table>
<thead>
<tr>
<th>What to identify</th>
<th>Guidelines</th>
<th>Your potential candidates</th>
</tr>
</thead>
</table>
| At least two LPARs for redirection activities | ■ These LPARs must operate in the same sysplex environment.  
■ For early testing, you can choose any LPAR that meets the minimum system requirements (“Operating system requirements” on page 34). For later testing and production, ensure that the LPARs are configured to handle Subzero workloads as described in “Tuning your environment for Subzero” on page 45. | |
<table>
<thead>
<tr>
<th>What to identify</th>
<th>Guidelines</th>
<th>Your potential candidates</th>
</tr>
</thead>
</table>
| At least one representative subsystem (such as a DB2 subsystem or IMS control region) | - For early testing, you can choose small or simplified representative subsystems.  
- To ensure complete coverage in later testing, you might choose larger and more complex representatives.  
- For production, you might include all subsystems in the list of potential candidates, and then rule out those that should not participate. | |
| At least one representative set of application programs that access the listed subsystem or subsystems | - For early testing, you can choose small-volume or simplified applications.  
- To ensure complete coverage in later testing, you might choose larger-volume and more complex applications.  
- For production, you might include all applications in the list of potential candidates, and then rule out those that should not participate. | |

2 From your list of potential candidates, choose the subsystems and LPARs that you want to use as clients and targets:
### Original subsystem

Choose the subsystem ID (such as DB2A or IMS1) to which your client applications send subsystem access requests:

- For simplicity in early testing, choose the subsystem ID to which your test applications are already sending subsystem access requests.

- For later testing, you can choose the existing subsystem ID to which your applications are already sending subsystem access requests, or you can choose a new subsystem ID.

This subsystem is not required to run on the same LPAR as the applications; it does not even need to be a real subsystem. For example, you could define an arbitrary standard ID that indicates Subzero will redirect the requests, and your application programs could send requests to that ID.

If your specified ID matches a subsystem that is running on the same LPAR as the client applications, Subzero will pass control to the native subsystem access routines; that is, no redirection will occur; in effect, you can specify redirection options to be used only when the original subsystem is not available on the client LPAR.

You can specify the same subsystem ID for the original subsystem and the target subsystem.

### Client LPAR

Choose the system ID of the LPAR that is running the applications that send requests.

### Target subsystem

Choose the subsystem ID of the "real" (destination) subsystem—the one that will receive and process redirected access requests.

Through Subzero, the target subsystem receives redirected requests from the client applications, processes the requests, and returns results.

Optionally, you can select alternative target subsystems that Subzero will use if the primary target is unavailable.

### Target LPAR

Choose the system ID of the LPAR that is running the destination subsystem.

The target LPAR cannot be the same as the client LPAR.

You can specify the same subsystem ID for the original subsystem and the target subsystem.
Using BMC Cost Analyzer *for zEnterprise* with Subzero

The BMC Cost Analyzer *for zEnterprise* product can help you identify candidates for Subzero redirection, model redirection changes, and predict the potential cost savings from redirection.

Cost Analyzer provides tools for analyzing, optimizing, and predicting IBM product license costs that are based on IBM Monthly License Charges (MLCs) and IBM Sub-Capacity Variable Workload License Charges (VWLC). With Cost Analyzer, you can:

- Use the Reporting tool to identify the best candidates for redirection
- Use the Planning tool to determine the effect of redirection on MLC software usage and costs

For more information, see the *BMC Cost Analyzer for zEnterprise User Guide*.

Examples of using the Cost Analyzer Reporting tool with Subzero

The following examples illustrate ways to use the Cost Analyzer Reporting tool with Subzero:

---

**Example**

1. In the Cost Analyzer Monthly Summary report, identify your peak 4-hour rolling average (4HRA).

2. Within the peak 4HRA interval, identify IBM MLC products that are using a proportionally small number of millions of service units (MSUs). These products might be good candidates for redirection to another LPAR.
Example

1. Using the Cost Analyzer Reporting tool, start from the Summary page and click a 4HRA MSU Utilization chart for the DB2 or IMS subsystem. Cost Analyzer displays an LPAR view that shows all LPARs on which that subsystem is active.

2. Using the Aggregated Workload view with the Subsystem Address Space workload type, look for LPARs that have a combination of IBM CICS with IMS or DB2.
   When you use the Subsystem Address Space workload type, Cost Analyzer aggregates all of the subsystem address spaces by type (CICS, IMS, DB2, and so on).

Examples of using the Cost Analyzer Planning tool with Subzero

After identifying redirection candidates with the Cost Analyzer Reporting Tool, use the Cost Analyzer Planning tool to model the effects of redirection on source and target LPARs.

Example

1. Create a plan with the Subsystem Address Space workload type.

2. Within the plan, move IMS or DB2 workloads from one LPAR to another, specifying the option to delete the IMS or DB2 workload from the source LPAR and add it to the target LPAR.

You can increase the accuracy of the model as follows:

- Increase the moved workloads slightly to account for Subzero overhead.

- When CICS and DB2 are on the same LPAR, DB2 CPU usage is charged to the CICS address space. When you move the DB2 portion of the workload to a different LPAR, DB2 CPU usage is charged to the Subzero address space. (This usage is not Subzero overhead.) To account for this effect, decrease the CICS workload and increase the DB2 workload by a corresponding amount.

Setting Subzero configuration options in LGC

Use the following procedure to specify how Subzero should redirect subsystem access requests from a client to a target.
Additional guidelines are as follows:

- LGC creates XML-based option sets to contain your Subzero configuration values. For a description of each option that you can set, see “Overview of Subzero configuration options in the LGC” on page 56.

- You can change Subzero configuration values while Subzero is active. The new values do not become effective until you reload the configuration in Step 7 on page 56.

- The DB2 Component Services (DBC) must be running on the LPAR that you designate as a new client or target LPAR. If the IBM CICS product started before DBC on a new client LPAR, you must restart CICS.

For more information about setting values in LGC, see the BMC Infrastructure Components Administration Guide.

**To set Subzero configuration values**

1. Execute the LGCISPF CLIST (located in the product SAMP data set).
   
   The default location for LGCISPF is in userLibHLQ.UXXSAMP or HLQ.BMCSAMP.

2. From the DB2 Product Configuration Main Menu, select Manage Product Options.

3. From the Product Option Sets panel, place your cursor on the plus sign (+) to the left of BMC Subsystem Optimizer for zEnterprise, and take one of the following actions:
   - To create a new Subzero option set for this product, enter I, and then enter redirection values.
   - To work with an existing Subzero option set, press Enter, put your cursor on the plus sign (+) to the left of that option set name, and take one of the following actions:
     - Enter E to edit the values in the option set.
     - Enter C to create a new option set that contains values copied from this option set.

   **Tip**
   Entering B allows you to browse (view) the values in the option set.

4. On the option set panel, select an option group, and take one of the following actions:
Enter I next to an option name to create an option set, and enter valid values in the editable fields.

Enter C next to an option name to copy that option set, and type over the displayed values to change them. (This action is not valid for General Options.)

5 When ready to save your changes, press F3 (the END command).

6 If you created a new option in Step 3 on page 55, specify the option’s name and description in the Create New Option Set window, and press Enter.

For the name, use the DBC group name that is associated with this configuration.

7 Activate the new or changed option set by entering the following command on the system console:

```
/dbc BRD,CONFIG,RELOAD
```

dbc is the subsystem ID of the DBC address space.

8 If you configured a new client LPAR and CICS started before the DBC on that LPAR, restart CICS.

Overview of Subzero configuration options in the LGC

This topic describes the Subzero configuration options that tell Subzero how to redirect subsystem access requests.

---

*Note*

In clientLPAR and targetLPAR fields, you can specify an asterisk (*) wildcard character in the value to make it generic:

- Specify * to match any system ID.

- Specify a character string followed by * to match any system ID that begins with that string. For example, specify SYS* to match SYSA, SYSB, SYS1, and so on.

---
<table>
<thead>
<tr>
<th>Section</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Options</td>
<td>XCFGROUP</td>
<td>This option enables you to specify the name of the XCF group that Subzero will use to process redirected requests. Contact your IBM z/OS system administrator to assign a group name for Subzero or help you choose one. No other XCF software running in the sysplex can use this name. <strong>Note:</strong> No z/OS system changes are required for Subzero to use this name.</td>
</tr>
</tbody>
</table>
|                  | (Advanced Option) Name subsystem servers based upon source JOBNAME | *(optional)* You can name the target subsystem servers based on the source job name. Enter a source job name, and then enter a pattern using standard symbols and the following BMC proprietary symbols:  
  - &SRC_JOBNAME  
    The job name of the application that is requesting a connection  
  - &ATT_NAME  
    The attach facility name. This name depends on the application, but for IBM CICS, use the CICS APPLID.  
  - &ATT_TYPE  
    The attach facility type (either IMS or DB2)  
  - &ATT_SSID  
    The SSID/GROUP name or IMSID for the connection |
### Defining Subzero redirections

<table>
<thead>
<tr>
<th>Section</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| DB2 SSIDs/Groups to Redirect | originalSSID on clientLPAR will redirect to targetSSID on targetLPAR | This option defines how Subzero should redirect requests between specified DB2 subsystems. You can insert or copy as many options of this type as needed. This option contains the following fields:  
  - At originalSSID, specify the subsystem ID of the original DB2 subsystem to which applications send subsystem access requests. You can specify an individual DB2 subsystem ID or a DB2 group ID.  
  - At clientLPAR, specify the system ID of the LPAR that is running the application programs.  
  - At targetSSID, specify the subsystem ID of the "real" (destination) DB2 subsystem that will receive and process redirected subsystem access requests. You can specify an individual DB2 subsystem ID or a DB2 group ID.  
  - At targetLPAR, specify the system ID of the LPAR that is running the real DB2 subsystem.  
  Tip: You can expand this option to display a Notes field, which you can use to enter comments or other reminders. The value is optional and has no default. |
| Redirect to targetSSID on targetLPAR | (optional) This option defines the alternative DB2 target subsystems to use if the primary target subsystem is unavailable. You can specify up to two alternative targets.  
1. Enter values for alternative targetSSID and targetLPAR.  
2. In the Automatically enable alternatives? field, enter Y.  
   You can use the BRD,ALTS,ENABLE or BRD,ALTS,DISABLE command to override this value.  
   For more information, see “Overview of Subzero console commands” on page 95. |
<table>
<thead>
<tr>
<th>Section</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| IMS Subsystems to Redirect  | originalSSID on clientLPAR will redirect to targetSSID on targetLPAR | This option defines how Subzero should redirect requests between specified IBM IMS control regions. You can insert or copy as many options of this type as needed. This option contains the following fields:  
  - At originalSSID, specify the subsystem ID of the original IMS control region to which applications send subsystem access requests.  
  - At clientLPAR, specify the system ID of the LPAR that is running the application programs.  
  - At targetSSID, specify the subsystem ID of the "real" (destination) IMS control region that will receive and process redirected subsystem access requests.  
  - At targetLPAR, specify the system ID of the LPAR that is running the real IMS subsystem.  
  Tip: You can expand this option to display a Notes field, which you can use to enter comments or other reminders. The value is optional and has no default. |
|                             | redirect to targetSSID on targetLPAR       | *(optional)* This option defines the alternative IMS target subsystems if the primary target subsystem is unavailable. You can specify up to two alternative targets.  
  1. Enter values for alternative targetSSID and targetLPAR.  
  2. In the Automatically enable alternatives? field, enter Y.  
  You can use the BRD, ALTS,ENABLE or BRD,ALTS,DISABLE commands to override this value.  
  For more information, see “Overview of Subzero console commands” on page 95. |
## Configuring CICS for Subzero

Use the following procedure to configure each CICS region that will participate in Subzero redirection activities.

<table>
<thead>
<tr>
<th>Section</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| MQSeries Subsystems To Redirect | **originalMQID** on **clientLPAR** will redirect to **targetMQID** on **targetLPAR** | This option defines how Subzero should redirect requests between specified MQ subsystems. You can insert or copy as many options of this type as needed. This option contains the following fields:  
  - At **originalMQID**, specify the subsystem ID of the original MQ subsystem to which applications send subsystem access requests.  
  - At **clientLPAR**, specify the system ID of the LPAR that is running the application programs.  
  - At **targetMQID**, specify the subsystem ID of the "real" (destination) MQ subsystem that will receive and process redirected subsystem access requests.  
  - At **targetLPAR**, specify the system ID of the LPAR that is running the real MQ subsystem.  
  
  **Tip:** You can expand this option to display a **Notes** field, which you can use to enter comments or other reminders. The value is optional and has no default.  

|                                  | **redirect to **targetMQID** on **targetLPAR** | *(optional)* This option defines the alternative MQ target subsystems if the primary target subsystem is unavailable. You can specify up to two alternative targets.  
  1. Enter values for alternative *targetMQID* and *targetLPAR*.  
  2. In the **Automatically enable alternatives?** field, enter **Y**.  
     You can use the BRD,ALTS,ENABLE or BRD,ALTS,DISABLE commands to override this value.  
     For more information, see “Overview of Subzero console commands” on page 95.  

---

**Configuring CICS for Subzero**

Use the following procedure to configure each CICS region that will participate in Subzero redirection activities.
For more information, view the Quick Course "Subsystem Optimizer for zEnterprise - Customizing CICS to include Subzero."

To configure CICS for Subzero

1. Modify the CICS startup JCL by adding the Subzero load library to the following concatenations:
   - STEPLIB
   - DFHRPL

   BMC recommends that you add the Subzero library as the last data set in the concatenation.

2. Create or modify the CICS program list table (PLT) to include the Subzero PLT program (BRDPLT00):

   **Note**
   The PLT specifies which programs are executed in the post-initialization phase of CICS startup.

   - If creating a new PLT, create a member in your CICS table library. The member name should adhere to the naming conventions in your environment. Enter the following statements in the member. `xx` is the two-character suffix of the PLT.
     ```
     DFHPLT TYPE=INITIAL,SUFFIX=xx
     DFHPLT TYPE=ENTRY,PROGRAM=BRDPLT00
     DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
     DFHPLT TYPE=FINAL
     END
     ```

   - If modifying a PLT, modify the PLT source and include an entry for program name BRDPLT00. Insert the entry before the `PROGRAM=DFHDELIM` statement:
     ```
     ...
     DFHPLT TYPE=ENTRY,PROGRAM=BRDPLT00
     DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
     ...
     ```

3. Assemble and link the new or updated PLT.

4. Update the PLTPI parameter in the CICS system initialization table (SIT) to identify this updated PLT to CICS:
   ```
   PLTPI=xx
   ```

   `xx` is the two-character suffix of the PLT.

5. Ensure that the DBC address space is running, and that the Subzero controlling agent is active.
DBC and Subzero must start before any CICS regions that are participating in Subzero redirection. Otherwise, applications will use the native IMS or DB2 interfaces instead of redirections.

6 Restart CICS.

CICS is now using the specified PLT, and is ready for Subzero participation.

## Configuring IMS to communicate with DB2

Use the following procedures to enable IMS to communicate with remote DB2 subsystems via Subzero.

There are two implementation procedures that can be used to enable IMS connections to remote DB2 subsystems.

- Static IMS connectivity to DB2 via Subzero, see “Static IMS connectivity to DB2 via Subzero” on page 63.
- Dynamic IMS connectivity to DB2 via Subzero, see “Dynamic connectivity to DB2 via Subzero” on page 64.

You can use a combination of implementation procedures. You can force some DB2 connection definitions to use Subzero, and allow the others to be dynamically altered for remote or local connectivity at connection time.

Both of these implementation procedures require that you have:

- Configured and started the Subzero environment on the local and remote LPARs
- Configured remote connectivity using the DB2 Product Configuration (LGC) component

To also use Subzero as an interface between IBM IMS and IBM DB2, you must have a license for the BMC Subsystem Optimizer for DB2 product.

For more information, view the Quick Course "Subsystem Optimizer for zEnterprise - Redirecting IMS/TM to remote DB2 subsystems."
Static IMS connectivity to DB2 via Subzero

This implementation procedure forces IMS to connect to specific DB2 subsystems via Subzero. You may want to use this implementation method in an environment used to validate Subzero functionality. The method is minimally invasive, but offers the least flexibility.

**Note**

This implementation procedure does not allow IMS to connect to a local DB2 subsystem using native DB2 interfaces. Even when the DB2 subsystem is on the same LPAR, it will connect via Subzero. Consequently, your IMS, DB2 and Subzero Subsystem Server address spaces are all on the same LPAR.

This implementation procedure does not support DB2 Group Attach Names. Your dependent region SSM member parameters must specify DB2 subsystem names; not DB2 Group Attach Names.

To create a static connection to DB2 via Subzero

1. Change the IMS Configuration.

   a. Edit the DB2 external Subsystem member in your IMS PROCLIB.

      This is a member name of `iiiiissss` where `iiii` is the IMS ID and `ssss` is the SSM suffix specified in your IMS startup. For example, IMSA with an SSM=DB2A would require a member of IMSADB2A in your IMS PROCLIB.

      Change the SSM member parameters from `ESMT=DSNMIN10` to `ESMT=BRDMIN10`.

      **Example**

      

      ```
      SST=DB2,SSN=T2HZ,LIT=SYS1,ESMT=DSNMIN10,REO=R,CRC=%
      ``

      is changed to:

      ```
      SST=DB2,SSN=T2HZ,LIT=SYS1,ESMT=BRDMIN10,REO=R,CRC=%
      ``

      **Note**

      SSN can be DB2 source name in LGC options.

     b. Add the Subzero product library to the DFSESL DD statement in your IMS Control Region JCL.

   c. Add the Subzero product library to the DFSESL DD statement in your IMS Dependent Region JCL.
Note
You do not need to remove any DB2 load libraries however, they are not needed for remote DB2 connectivity and can be removed unless there are application program requirements for them to be present (for example DSNTIAR to format messages).

2 Implement the IMS configuration changes.

You do not need to stop and restart IMS in order to implement your changes. You can simply stop and restart the subsystem definitions.

a Issue a /STOP SUBSYS ALL command to IMS. This will stop the current connections to DB2.

b Issue a /START SUBSYS ALL command to IMS. This will restart the current connection to DB2 using the new SSM parameters.

3 Look for BRDM001 and/or BRD0150I messages indicating that your DB2 Subsystem connections are started and being redirected by Subzero:

BMCBRD0150I Redirecting T2HZ to DB2 DEHZ on system SYSM
BRDM001I IMS/TM T13P CONNECTED TO SUBSYSTEM T2HZ

4 Restart your IMS dependent regions.

Dynamic connectivity to DB2 via Subzero

This implementation procedure offers the most flexibility. IMS and DB2 subsystems can be moved to different LPARs. Subzero makes the remote connections whenever necessary, and allows the local connections to occur whenever possible.

This implementation procedure requires the implementation of an IMS usermod, and subsequent stop and restart of IMS. It offers the following additional features:

- You do not need to modify any IMS SSM member parameters. IMS SSM parameters are modified in real-time based upon Subzero definitions and DB2 subsystem availability.

- IMS dependent regions can connect to DB2 Group Attach Names as well as specific DB2 Subsystems.

For example, the IMS control region running on SYSA can connect to DB2B and DB2C running on SYSB and SYSC respectively. Assuming DB2B and DB2C are part of DB2 Group DB2G, the IMS dependent regions can specify an SSM member with a SSN=DB2G in order to establish a remote connection to an available DB2 subsystem on SYSB or SYSC.
IMS can connect to local DB2 subsystems using native interfaces and remote DB2 subsystems using Subzero.

For example, IMSA on SYSA may have a SSN=DB2B definition.

If DB2B is running on SYSB, Subzero will make the remote connection without the use of any native DB2 interfaces.

If DB2B is running on SYSA, Subzero will simply pass the connection to the native DB2 interface routines without any further execution of Subzero code.

The IMS usermod front-ends the IBM SSM member edit routine (DFSESI00) and performs a real-time scan of the Sysplex for DB2 subsystems. When running inside the IMS control region, the Subzero front-end looks for the individual DB2 subsystems on the local LPAR. If found, the SSM member parameter is not modified for that DB2 connection. If the DB2 subsystem is found on a remote LPAR and there is a Subzero definition for it, the Subzero front-end routine will modify the in-core SSM member parameter data to enable the Subzero initialization table (BRDMIN10) to make the remote connection to DB2.

When running inside the IMS dependent regions, the Subzero SSM member edit routine scans the remote connection table to find any remote DB2 subsystems. If one is found, Subzero will again, modify the in-core SSM member parameter data to enable the Subzero initialization table. If the IMS dependent region SSM member data specifies a DB2 Group Attach Name, Subzero will look for a remote DB2 subsystem within that group. If found, it will alter the in-core SSM member parameter data to connect to the specific DB2 Subsystem. Like IMS, Subzero will find the first available DB2 subsystem within the DB2 group and make the remote connection.

To create a dynamic connection to DB2 via Subzero

1. Apply the IMS usermod:
   a. Edit the usermod - BRDSMP01. It can be found in SAMPLIB.
      
      Modify the following elements as instructed within the JCL:
      
      - Subzero high-level qualifier
      - IMS Distribution load library
      - IMS RESLIB
      - IMS SMP/E CSI
      - IMS SMP/E Target Zone
      - Modify the +++VER to specify only the entry for the correct IMS version

   b. Run the job. By default, this does an APPLY CHECK of the usermod. If the APPLY CHECK is successful, you can remove the CHECK parameter and rerun the job.
Note
Do not accept this usermod. Reject this usermod before applying IMS maintenance and then re-apply the usermod afterwards.

2 Change your IMS startup JCL:
   a Add the Subzero product library to the DFSESL statement in your IMS Control Region JCL.
   b Add the Subzero product library to the DFSESL DD statement in your IMS Dependent Region JCL.

Note
You do not need to remove any DB2 load libraries however, they are not needed for remote DB2 connectivity and can be removed unless there are application program requirements for them to be present (e.g. DSNTIAR to format messages).

3 Stop and re-start IMS.

4 Look for BRD0022I and BRD0023I messages indicating that remote subsystem connectivity is available. For example:

   BMCBRD00023I BMC Subsystem Optimizer for zEnterprise IMS initialized by subsystem AXL2

   BMCBRD00022I BMC Subsystem Optimizer for zEnterprise is active for remote DB2 connections

5 Look for BRDM001 and/or BRD0150I messages indicating that your DB2 Subsystem connections are started and being redirected by Subzero. For example:

   BMCBRD0150I Redirecting T2HZ to DB2 DEHZ on system SYSM

   BRDM001I IMS/TM T13P CONNECTED TO SUBSYSTEM T2HZ

Configuring MQ for redirection

To use IBM MQ requests without Subzero, you must run the MQ subsystem on the same LPAR as the IMS, DB2, or CICS subsystem. Subzero enables you to run the MQ subsystem on a remote LPAR and reduce variable workload license charges.

Note
You must have a license for BMC Subsystem Optimizer for MQ.
**Example of MQ redirection using Subzero**

For this scenario, assume that you have a DB2 and an MQ subsystem on LPAR A, and an IMS and MQ subsystem on LPAR B. Deploying Subzero lets you separate the DB2 and IMS subsystems from the MQ subsystem by moving the MQ subsystems to a third LPAR.

**Setting up MQ to communicate with CICS**

Use the following procedure to enable IBM MQ to communicate with CICS via Subzero.

Subzero can redirect the following Message Queue Interface (MQI) calls that CICS supports:

- CALL MQOPEN – Open object
- CALL MQCLOSE – Close object
- CALL MQINQ – Inquire object attributes
- CALL MQPUT – Put message
- CALL MQPUT1 – Put one message
- CALL MQGET – Get message
Before you begin

You must have a license for BMC Subsystem Optimizer for MQ.

To set up MQ to communicate with CICS

1 Configure each CICS region that will participate in Subzero redirection activities.
   See “Configuring CICS for Subzero” on page 60.

2 Set up the Subzero MQ redirection options in the LGC.
   See “Setting Subzero configuration options in LGC” on page 54.

Setting up MQ to communicate with IMS

Use the following procedure to enable IBM IMS to communicate with MQ via Subzero.

Subzero can redirect the following Message Queue Interface (MQI) calls that IMS supports:

- CALL MQBACK – Back out changes
- CALL MQCLOSE – Close object
- CALL MQCMIT – Commit changes
- CALL MQCONN – Connect queue manager
- CALL MQCONNX – Connect queue manager extended
- CALL MQDISC – Disconnect queue manager
- CALL MQGET – Get message
- CALL MQINQ – Inquire object attributes
- CALL MQOPEN – Open object
- CALL MQPUT – Put message
- CALL MQPUT1 – Put one message
CALL MQSET – Set object attributes
CALL MQSTAT – Retrieve Status Information
CALL MQSUB – Register Subscription
CALL MQSUBRQ – Subscription Request

**Before you begin**

You must have a license for BMC Subsystem Optimizer for MQ.

This implementation procedure includes an optional step, which requires the implementation of an IMS usermod, and subsequent stop and restart of IMS. Using the usermod offers the following additional features:

- You do not need to modify any IMS SSM member parameters. IMS SSM parameters are modified in real-time based upon Subzero definitions and MQ subsystem availability.
- IMS can connect to local MQ subsystems using native interfaces and remote MQ subsystems using Subzero.

If you do not want to apply the usermod, skip **Step 1**, and in IMS PROCLIB(IMSIDMQID) DD, change the name of the member CSQQESMT to BRDQESMT.

**IMSID** is the IMS subsystem ID, and **MQID** is the MQ subsystem ID.

For example change:

Q7K1,MQM1,CSQQESMT,,R,(...) to Q7K1,MQM1,BRDQESMT,,R,(...)

**To set up MQ to communicate with IMS**

1. **Apply the IMS usermod:**
   a. Edit the usermod - BRDSMP01. It can be found in SAMPLIB.

   Modify the following elements as instructed within the JCL:

   - Subzero high-level qualifier
   - IMS Distribution load library
   - IMS RESLIB
   - IMS SMP/E CSI
   - IMS SMP/E Target Zone
   - Modify the ++VER to specify only the entry for the correct IMS version
b Run the job. By default, this does an APPLY CHECK of the usermod. If the APPLY CHECK is successful, you can remove the CHECK parameter and rerun the job.

**Note**
Do *not* accept this usermod. Reject this usermod before applying IMS maintenance and then re-apply the usermod afterwards.

2 Change your IMS startup JCL:

a Add the Subzero product library to the DFSESL DD statement in your IMS control region JCL.

b Add the Subzero product library to the DFSESL DD statement in your IMS dependent region JCL.

**Note**
You can remove any MQ load libraries, as they are not required for remote connectivity, unless the application program uses them. For example, an application program might require MQ Data Conversion services to perform ASCII to EBCDIC conversions.

3 Implement the IMS configuration changes:

**Note**
You *only* need to stop and restart IMS in order to implement your changes if you applied the usermod. If you did not apply the usermod, you can simply stop and restart the subsystem definitions.

a Issue a `/STOP SUBSYS ALL` command to IMS to stop the current connections to MQ.

b Issue a `/START SUBSYS ALL` command to IMS to restart the current connection to MQ using the new SSM parameters.

You should see the following messages:

```
BMCBRDM0026I MQID connection is being managed by Subzero (DBCID)
CSQQ0001 IMS/TM IMSID connected to queue manager MQID
```

4 Restart your IMS dependent regions.

5 Set up the Subzero MQ redirection options in the LGC.

See “Setting Subzero configuration options in LGC” on page 54.
Setting up MQ to communicate with DB2

Use the following procedure to enable IBM DB2 to communicate with MQ via Subzero.

Subzero can redirect the following DB2 SQL functions:

- MQREAD
- MQREADALL
- MQREADALLCLOB
- MQREADCLOB
- MQRECEIVE
- MQRECEIVEALL
- MQRECEIVEALLCLOB
- MQRECEIVECLOB
- MQSEND

Before you begin

You must have a license for the BMC Subsystem Optimizer for MQ product.

To set up MQ to communicate with DB2

1. From the BRDLOAD library, copy the following files to the DB2 DSNEXIT library:
   - BRDBCON
   - BRDBDSC
   - BRDBSRV
   - BRDLOCSS

2. Rename the following files:
   - BRDBCON to CSQBCON
   - BRDBDSC to CSQBDSC

3. Set up the Subzero MQ redirection options in the LGC.
   
   See “Setting Subzero configuration options in LGC” on page 54.
Configuring an MQ QSG to communicate with DB2

Use the following procedure to configure an IBM MQ queue-sharing group (QSG) to cache shared-object definitions on an IBM DB2 subsystem that is running on a remote LPAR.

Before you begin

To use Subzero as an interface between IBM MQ and IBM DB2, you must have a license for the BMC Subsystem Optimizer for MQ product.

Ensure that the DB2 Product Configuration (LGC) component is running.

Enabling MQ to communicate with DB2

1. Make the following changes to the Subzero load library:
   a. Copy module DSNARRS from the DB2 DSNLOAD library to module DSNARA00 in your Subzero load library.
      
      **Example**
      
      `BRD.V20.BMCLINK(DSNARRS)`
      
   b. Copy module BRDARRS in your Subzero load library as DSNARRS.

      **Note**
      
      If you subsequently apply DB2 maintenance to DSNARRS or apply Subzero maintenance to BRDARRS, you must copy and rename the modules again.

2. Verify that the following modules in the Subzero load library are listed in the MQ master address space (noting that other modules might also be present):
   - DSNARA00—Contents identical to DSNARRS in the DB2 DSNLOAD library
   - BRDLOCSS—Subzero module
   - BRDCCINT—Subzero module
   - DSNARRS—Contents identical to the BRDARRS Subzero module

3. In the MQ master address space, edit the JCL to insert the Subzero load library before the DB2 load library in the STEBLIB configuration.

   You can insert these after the MQ load libraries, and before any DSNLOADs.

   **Note**
   
   All data sets in the STEPLIB DD must be APF authorized.

4. Edit the Subzero redirection configuration in the LGC:
a From the DB2 Product Configuration Main Menu, select **Manage Product Options**.

b From the Product Option Sets panel, place your cursor on the plus sign (+) to the left of **BMC Subsystem Optimizer for zEnterprise** and press **Enter**.

c Select the option set that you want to edit.

d Insert a new entry by typing `I` next to **DB2 SSID/GROUPs to Redirect** and pressing **Enter**.

e Change the new entry as needed.

For example DKJ2 on LPAR * will redirect to DKJ1 on LPAR *.

f Press **F3** to save your entries.

5 Enter the following command to reload the map:

```
/dbcssid BRD,CONFIG,RELOAD
```

6 Start Subzero:

a Start Subzero on the MQ LPAR by entering `/ssid BRD,START`.

Replace `ssid` with the DBC subsystem ID.

b Start Subzero on the DB2 LPAR by entering `/ssid BRD,START`.

7 Start MQ.

8 Verify that the following messages are displayed, confirming that the process has been successful:

- **BMCMRD0026I**
- **BMCMRD0026I**
- **BMCBRD0150I**

## Securing Subzero components with SAF

Use the following procedure to secure Subzero components through your standard external security manager, (such as the IBM RACF product) and the System Authorization Facility (SAF).
Note
For information about securing DBC resources, see the *BMC Infrastructure Components Administration Guide*.

This procedure gives other BMC products access to the STATAPI resource; STATAPI allows those products to work with Subzero through the Subzero application programming interface (BRDAPI). For example, the MainView products for CICS, DB2, DBCTL, and IMS Online can work with Subzero through BRDAPI.

Before you begin
You might need to consult with your environment’s security administrator to prepare and execute the definition.

To define the STATAPI resource to RACF

1 Specify JCL to define the STATAPI resource to your SAF package, and permit access to the resource.

The resource name is *HLQ.lparName.dbcGroup.BRD.STATAPI.PF*. When a product function attempts to access the defined resource, DBC supplies the high-level qualifier that is assigned to DBC at startup, the system ID of the current LPAR, and the DBC group name. MainView provides the BRD and STATAPI portions of the resource name, and DBC appends the PF portion of the name (to indicate a product function).

Tip
When you define a resource for the BRDAPI, you can specify the wildcard * in values if you want the definition to apply to multiple systems.

Example
The following JCL shows how you might define the STATAPI resource to RACF:

```
//RACFBAT JOB ...
//TSO EXEC PGM=IKJEFT01,REGION=4M,DYNAMNBR=30
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *
RDEFINE FACILITY *.*.BRD.STATAPI.PF
PERMIT *.*.BRD.STATAPI.PF CLASS(FACILITY) +  
   ID(MAINVIEW) ACCESS(UPDATE)
SETROPTS CLASSACT(FACILITY)
SETROPTS RACLIST(FACILITY) REFRESH
RLIST FACILITY *.*.BRD.STATAPI.PF AUTHUSER
```

2 Execute the job, and view the output to determine that the required access is permitted.
Setting up IMS or DB2 batch job redirection

To set up batch job redirection, you need to install and prepare APPLICATION RESTART CONTROL and MainView Batch Optimizer, and to configure the DB2 Product Configuration (LGC) common component.

Note
Subzero does not support the following implementations for IMS batch job redirection:

- Remote access to the I/O or to the alternate I/O program communication blocks (PCB)
  Subzero cannot redirect the application program to access the IMS I/O PCB or Alternate I/O PCB on a remote IMS system. Also, Subzero cannot redirect access to the IMS message queues (no transaction-oriented BMP support) or to the IMS command interface.

- Remote access to generalized sequential access method (GSAM) databases
  AR/CTL for IMS can provide support for Application Sequential Access Method (ASAM) database access, which you can implement to replace GSAM database access. For more information, contact BMC Customer Support.

- Scheduling a PSB that contains a PCB with a processing option (PROCOPT) of L (LOAD) or LS (LOAD SEQUENTIAL)
  The PSB is the third parameter of the BMP/DLI/DBB execution parameter.

Complete the tasks in the following table in the order shown.

Table 1: Summary of setup tasks

<table>
<thead>
<tr>
<th>In this product or component</th>
<th>Complete these tasks</th>
<th>Using these instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MainView Batch Optimizer</td>
<td>Create a new job policy in the control data set, add the DIRECT action to the policy's global options, and activate the policy.</td>
<td>“Adding the DIRECT command to a job policy definition” on page 79 “Activating a job policy in Job Optimizer” on page 80</td>
</tr>
<tr>
<td></td>
<td>Add the DIRECT JTL statement for each job that you want to execute on a remote subsystem.</td>
<td>“Adding the DIRECT JTL statement” on page 81</td>
</tr>
<tr>
<td></td>
<td>Edit the MainView Batch Optimizer Subsystem commands member to activate the Subzero for DB2 interface at startup.</td>
<td>“Activating the Subzero interface” on page 81</td>
</tr>
</tbody>
</table>
Batch job deployment architecture and workflow

You can redirect batch job SQL requests to a remote DB2 subsystem, redirect IMS DL/I calls issued by batch jobs to a remote IMS subsystem, or perform the job steps on the remote LPAR where the subsystem resides.

The following example describes the process for DB2 batch job requests and requires a license for Subzero for DB2. The process for IMS batch job requests is similar, but requires a license for Subzero for IMS.

1. You use the APPLICATION RESTART CONTROL for DB2 (AR/CTL) product to define that you want to redirect the SQL requests to a DB2 subsystem on a remote LPAR or alternatively perform the job on the remote LPAR in order to reduce latency.

2. AR/CTL calls Subzero.

3. According to the action that you selected in AR/CTL, Subzero takes one of the following actions:
   - Redirects the SQL requests to the remote DB2 subsystem
   - Calls MainView Batch Optimizer–Advanced, which performs the job steps consecutively on the remote LPAR
This deployment uses capabilities that APPLICATION RESTART CONTROL for DB2 and MainView Batch Optimizer provide. For more information, see “Setting up IMS or DB2 batch job redirection” on page 75.

Figure 11: Deployment example for batch job support—redirection of SQL requests
Including the DFSPZP module for IMS Batch Support

Subzero IMS Batch support requires a Database Resource Adapter (DRA) Startup Parameter Table, this module must have a member name of DFSPZPSZ.

1. Modify the following DFSPZPSZ sample as appropriate for your environment.

```plaintext
DFSPRP DSECT=NO,
FUNCLV=3,           Function level 3
DDNAME=BRDRESL,     DDNAME FOR DRA RESLIB
DSNAME=BMCBRD.IMS.SDFSRESL,
DBCTLID=BRD,        DBCTL IDENTIFIER
USERID=BRDUSR,      USER IDENTIFIER
```
MINTHRD=1,          MINIMUM NUMBER OF THREADS
MAXTHRD=1,          MAXIMUM NUMBER OF THREADS
TIME=2,            IDENTIFY TIMER VALUE DEFAULT
FPBUF=0,            NUMBER OF FP BUFFERS PER THREAD
FPBOF=0,            NUMBER OF FP OVERFLOW BUFFERS
SOD=A,             SNAP DATASET OUTPUT CLASS
AGN=,              AGN, no longer supported
TIMEOUT=60,         DRA TERM TIMEOUT VALUE
IDRETRY=5,         Identify Retry count
PCBLOC=31,          Local PCB storage option
CNBA=0,            Total FP NBA buffers this Conn
OPENTHRD=DISABLE,      OTT option
GENSNAP=NO,         Generate SNAP dumps YES or NO
IMSPLEX=,          IMS PLEX ODBA to use for ODBM
ODBMNAME=,         ODBM Name this ODBA to use

2 Ensure that DFSPZPSZ is APF authorized.

3 Link DFSPZPSZ to IMS RESLIB.

Setting up LGC

This topic explains how to set options in the DB2 Product Configuration (LGC) component to support the Subzero batch job feature.

In the LGC, you must configure the redirection information for the client and target subsystems. For more information, see “Setting Subzero configuration options in LGC” on page 54 or “Overview of Subzero configuration options in the LGC” on page 56.

Setting up MainView Batch Optimizer

This topic explains how to set up MainView Batch Optimizer to support the Subzero batch job features.

Adding the DIRECT command to a job policy definition

Use the following procedure to add the DIRECT command to the policy definition for each job that you want to redirect in the control data set.

Note

The default job policy member is JOBPOL00.

To add the DIRECT command to a job policy definition

1 Access the MainView Batch Optimizer user interface.
To do so, you can use the REXX EXEC member BSSBISP in data set BMC.BSS.INSTALL.

2 At the product logo panel, press Enter.

3 On the Objects List panel, type E (Edit) in the action entry field to the left of the job policy that you want to edit, and press Enter.

The Job Optimization Policy panel displays the selected policy's information and selection criteria.

4 In the Selection Criterion list, type E (Edit) next to <New>, and press Enter.

5 In the Job Policy Definition panel, enter DIRECT in the Action field.

Figure 13: Example Job Policy Definition panel

6 Press F3 and select option 1 to save your changes and exit.

For more information, see the topic about accessing the user interface in the MainView Batch Optimizer Job Optimizer Reference Manual.

Activating a job policy in Job Optimizer

The job policy becomes active when you start the MainView Batch Optimizer Subsystem on the images in the BatchPlex.

Your site's initial job policy was created during the MainView Batch Optimizer customization process. The name of the job policy is defined within the BatchPlex definition. For a new job policy, you must activate the policy on each IBM MVS image in the BatchPlex.
If you created a new policy, or edited an existing policy to include the new DIRECT keyword, you cannot activate the policy on a previous maintenance level of MainView Batch Optimizer.

To activate a job policy

1 Enter the following command:

```
mbos BSL POLICY ACTIVATE policyName
```

*mbos* is the subsystem ID of the MainView Batch Optimizer Subsystem that is active on the MVS image.

*policyName* is the name of an existing job policy definition in the control data set. Omitting *policyName* reactivates the currently active job policy.

Adding the DIRECT JTL statement

Use the following procedure to add the DIRECT job transformation language (JTL) statement.

1 Add the following JTL statement immediately after a job in the MainView Batch Optimizer JCL:

```
//**BSLCNTL DIRECT
```

For more information about JTL statements, see the *MainView Batch Optimizer Job Optimizer Reference Manual*.

Activating the Subzero interface

Use the following procedure to activate the Subzero interface in MainView Batch Optimizer each time that MainView Batch Optimizer starts.

To activate the Subzero interface at startup

1 In the Subsystem commands member BCSCMDxx for MainView Batch Optimizer, add the following command:

```
mbos BSL SUBZERO ENABLE [IGNORE | WARN | FAIL]
```

Optionally, you can include the IGNORE, WARN, or FAIL keyword if you want to specify how MainView Batch Optimizer should react if a step fails to split:
Setting up AR/CTL

This topic explains how to set up AR/CTL for IMS to support the Subzero batch job feature.

Editing the AR/CTL BCSS started task

Use the following procedure to edit the AR/CTL BMC Consolidated Subsystem (BCSS) started task (for more information about the BCSS, see the APPLICATION RESTART CONTROL Administrator Guide).

1. Use the $Q03APFJ job to add the AR/CTL version 4.1.00 product load library and Subzero version 2.0.00 product load library to an APF-authorized load library.

   This job is created as part of the Application Enhancement Series Install System (AESIS) application. AESIS creates customized batch jobs that complete the installation and customization process of the AR/CTL products. For more information, see the configuration chapter of the APPLICATION RESTART CONTROL Customization Guide.

2. Add the APF-authorized load library, containing the AR/CTL and Subzero load libraries, to the BCSS STEPLIB DD.

3. Verify that the required passwords are available in the password data set:

<table>
<thead>
<tr>
<th>To use this product</th>
<th>In this environment</th>
<th>You need licenses for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subzero for DB2</td>
<td>An IMS or IMS replacement environment</td>
<td>AR/CTL for IMS Subzero for DB2</td>
</tr>
<tr>
<td></td>
<td>A non IMS environment</td>
<td>AR/CTL for DB2 Subzero for DB2</td>
</tr>
</tbody>
</table>
To use this product | In this environment | You need licenses for
--- | --- | ---
Subzero for IMS | An IMS environment | AR/CTL for IMS, Subzero for IMS

- Use of Subzero for DB2 in an IMS or IMS replacement environment requires AR/CTL for IMS, AR/CTL for DB2, and Subzero for DB2 licenses.

- Use of Subzero for DB2 in a non IMS environment requires AR/CTL for DB2 and Subzero for DB2 licenses.

- Use of Subzero for IMS in an IMS environment requires AR/CTL for IMS and Subzero for IMS licenses.

4. Issue the following command (where bcssid is the subsystem identifier) to reinitialize the BCSS:

```
bcssid REINIT AES
```

5. Check the system log to ensure that the REINIT command completed successfully.

The BCSS responds with messages indicating that it has initialized the AR/CTL components, and status messages.

**Creating a program registration record**

To set up AR/CTL to participate in application program execution without JCL changes, you can create one or more program registration records in the REGISET.

**To create a program registration record**

*Note*

For more information about this task, see the [*APPLICATION RESTART CONTROL Administrator Guide*](#).

1. Access the Application Enhancement Series primary menu, and select option 1 (AES records).

2. On the Application Enhancement Series Records panel, select option 3 (Non IMS Program manual registration) or option 7 (IMS Program registration).

3. On the Limit List of Records panel, type or verify asterisks in all fields, and press Enter.

4. On the List Records panel, enter the **ADD** command.
5 On the Add Record panel, type the qualifiers to use in the record key, type the requested information (if applicable), and press **Enter** to validate the information.

---

**Note**

To allow a job step that abends on one IMS system to be restartable on a different IMS system, set the IMSID to wildcard characters in the program registration record. Although the IMSID is not in the key of the restart control record, the IMSID is in the key of the program registration record that identifies the application program for AR/CTL participation.

---

6 Enter the **END** command to save the record and exit from the panel.

### Creating a Subzero program registration record

Use the following procedure to create a Subzero IMS or DB2 program registration record.

1 Access the Application Enhancement Series primary menu, and select option **1** (AES records).

2 On the Application Enhancement Series Records panel, select one of the following options:
   - Option 12 (Subzero IMS program registration)
   - Option 11 (Subzero DB2 program registration)

3 On the Limit List of Records panel, type or verify asterisks in all fields, and press **Enter**.

4 On the List Records panel, enter the **ADD** command.

5 On the Add Record panel, type the qualifiers to use in the record key, type the requested information (if applicable), and press **Enter** to validate the information.

6 Enter the **END** command to save the record and exit from the panel.

### Creating an environment registration record

To provide access to AR/CTL execution modules and services, you must create one or more environment registration records in the REGISET.

Different records are used for non IMS environments and IMS environments:
A non IMS environment is any environment where IMS is not active, or where application programs that execute in the environment do not use IMS type calls and structures.

An IMS environment is any environment where IMS is active, or where IMS is not active but IMS compatible application programs use IMS type calls and structures.

To create an environment registration record

*Note*
For more information about this task, see the **APPLICATION RESTART CONTROL Administrator Guide**.

1. Access the Application Enhancement Series primary menu, and select option 1 (AES records).
2. On the Application Enhancement Series Records panel, select option 2 (Non IMS Environment registration) or option 6 (IMS Environment registration).
3. On the Limit List of Records panel, type or verify asterisks in all fields, and press Enter.
4. On the List Records panel, enter the ADD command.
5. On the Add Record panel, type the qualifiers to use in the record key, type the requested information (if applicable), and press Enter to validate the information.
6. Enter the END command to save the record and exit from the panel.

Deactivating Subzero redirection in an AR/CTL step

Use the following procedure if you want to stop batch job redirection from AR/CTL.

1. Enter the relevant DD statement:
   - To stop DB2 batch job redirection:
     ```
     //AESSZ$2N DD DUMMY
     ```
   - To stop IMS batch job redirection:
     ```
     //AESSZ$IN DD DUMMY
     ```
Subzero installation verification program for CICS

Subzero provides an installation verification program (IVP) that you can use to ensure that the product is installed and set up correctly in your environment. The IVP (a transaction named BRD1) runs on the IBM CICS product to update your IBM DB2 database, IBM IMS database, or both.

With the IVP, you can verify that a CICS transaction can be run against a local database, or against a remote database (via Subzero) without changing the program, the CICS transaction definition, or the DB2 bind process. For example, you can run the IVP with your current CICS and database configuration, and then run it again after enabling Subzero and placing the database on a different LPAR.

Supported databases

The Subzero installation verification program (IVP) works with the following IBM databases:

- The DB2 database is part of the DB2 IVP that IBM supplies in job DSNTEJ1 in the DSNSAMP library. The Subzero IVP updates the dsn8vvr.DEPT and dsn8vvr.EMP tables.

  Note
  The dsn8vvr qualifier depends on your release of DB2. You provide this qualifier in the job that binds the IVP.

- The IMS database is named DI21PART and is part of the IMS IVP that IBM supplies. Subzero updates the PARTROOT segment in the DFHSAM04 program specification block (PSB).

Sample library members

The following members of the sample library contain programs and JCL for compiling, defining, and running the installation verification program (IVP):

<table>
<thead>
<tr>
<th>Member</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRDIV$D</td>
<td>Documentation about how to use the IVP</td>
</tr>
<tr>
<td>BRDIVAL</td>
<td>JCL to pre-process, assemble, and link the IVP</td>
</tr>
<tr>
<td>BRDIVBNN</td>
<td>JCL to run the DB2 BIND command</td>
</tr>
</tbody>
</table>
As indicated in the JCL comments, you must edit the JCL according to your environment, as follows:

- JOB statement
- Subzero high-level qualifiers (HLQs) and data set names
- (If applicable) DB2 version number, HLQ, and subsystem ID (SSID)
- (If applicable) IMS version number, HLQ, and whether to access SDFSRESL or RESLIB
- (If applicable) CICS version number, HLQ, and console name

Using the IVP

Use the following procedure to prepare the Subzero IVP for use, and to execute the IVP either as a batch program or through a CICS console.

**Before you begin**

Complete all procedures to install and set up Subzero in your environment.

**To prepare and run the IVP**

1. Edit and run the JCL in member BRDIVAL to assemble and link the IVP.

2. If applicable, on every DB2 system in which the Subzero IVP will run, edit and run the JCL in member BRDIVBN to bind the IVP.

3. Edit and run the JCL in member BRDIVCD to define the IVP transaction (BRD1) to CICS.

4. Run the IVP transaction.

You can run the IVP transaction from a CICS terminal or in batch by editing and running the JCL in member BRDIVEX.
You can run BRDIVPEX in either of the following ways:

- Use the REXX EXEC script in member BRDIVCMD to take advantage of the TSO CONSOLE command. This technique is valid only if the CICS subsystem has a defined console. (The IBM CICS IVP defines a console.) The REXX script can issue any operator command and display the output.

- Issue operator commands through JCL. In this case, the command output is written to the console and the CICS JES message log.

Select the method that works best for you, and delete or comment out the unneeded step from the BRDIVEX job.

To control the actions that the IVP is taking, provide input to the BRD1 transaction.

For more information, see “IVP transaction input” on page 88.

The IVP issues messages to explain the actions it is taking. All IVP messages start with the string **BRD**.

**IVP transaction input**

While the Subzero IVP transaction (BRD1) is running, you can use the following syntax to issue input to BRD1:

```
BRD1 action [psb] [count] [key]
```

Observe the following guidelines:

- The *action* option is required if you specify any of the other options.

- Include at least one space between options.

- To omit an option that is between other specified options, substitute a comma (,) or period (.) and retain the required space. The comma or period is a placeholder, not a separator. (The comma or period is not required for options omitted from the end of the statement.)
**Example**

To add one record to the DB2 database with a key of 12345:

```
BRD1 DB2 , 1 12345
```

To add the default number of records to the IMS database through the default PSB:

```
BRD1 IMS
```

To add the default number of records with a key of 12345 to the IMS and DB2 databases:

```
BRD1 BOTH , , 12345
```

The variables represent the following values:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>action</strong></td>
<td>The action that the IVP should take. Replace <em>action</em> with one of the following values:</td>
</tr>
<tr>
<td></td>
<td>— <strong>DB2</strong> (or <strong>D</strong>) updates the DB2 database.</td>
</tr>
<tr>
<td></td>
<td>— <strong>IMS</strong> (or <strong>I</strong>) updates the IMS database.</td>
</tr>
<tr>
<td></td>
<td>— <strong>BOTH</strong> (or <strong>B</strong>) updates the DB2 and IMS databases.</td>
</tr>
<tr>
<td></td>
<td>— <strong>DL</strong> lists all rows that the IVP added to the DB2 EMP table.</td>
</tr>
<tr>
<td></td>
<td>— <strong>DD</strong> deletes all rows that the IVP added to the DB2 EMP table.</td>
</tr>
<tr>
<td></td>
<td>— <strong>IL</strong> lists all records that the IVP added for the IMS PARTROOT segment.</td>
</tr>
<tr>
<td></td>
<td>— <strong>DL</strong> lists all rows that the IVP added to the DB2 EMP table.</td>
</tr>
<tr>
<td></td>
<td>— <strong>ID</strong> deletes all records that the IVP added to the IMS PARTROOT segment.</td>
</tr>
</tbody>
</table>

| **psb** | The name of the IMS program specification block (PSB) that the IVP should use. You can omit this option if you are using the default value DFHSAM04, or if the IVP is not updating the IMS database. |

| **count** | The number of records or rows that the IVP should add to the database. If you omit this option, the IVP uses the default value (3). |

| **key** | The unique key (1 through 8 numeric digits) for the record or row that the IVP is adding to the database. **Note:** If the IVP is adding more than one record or row, a duplication error occurs. |

- Entering `BRD1` with no options displays an abbreviated listing of input options; entering `BRD1 ?` displays syntax Help.
Subzero installation verification program for IMS

Subzero provides an Installation Verification Program (IVP) that you can use to ensure the product is installed and set up correctly in your IBM IMS Transaction Manager (TM) Resource Adapter (IMS TM) environment. The IVP transaction, named BRD4, runs as an online IMS transaction that accesses and updates your IBM DB2 database.

With the IVP, you can verify that the transaction can access both a locally connected and remotely connected (via Subzero) DB2 database without changing the program or the IMS transaction definitions. For example, you can run the IMS IVP transaction while locally connected to a DB2 subsystem and then again after enabling Subzero to connect to a remote DB2 subsystem.

Supported databases

The Subzero installation verification program (IVP) works with the DB2 IVP that IBM supplies in job DSNTEJ1 in the DSNSAMP library. The Subzero IVP updates the dsn8vvr.DEPT and dsn8vvr.EMP tables.

Note
The dsn8vvr qualifier depends on your release of DB2. You provide this qualifier in the job that binds the IVP.

Sample library members

The following members of the sample library contain programs and JCL for assembling and defining the installation verification program.

<table>
<thead>
<tr>
<th>Member</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRDIV$D</td>
<td>Documentation on how to use the IVP</td>
</tr>
<tr>
<td>BRDIVAL4</td>
<td>JCL to pre-process, assemble and link the IVP</td>
</tr>
<tr>
<td>BRDIVBN4</td>
<td>JCL to run the DB2 BIND command</td>
</tr>
<tr>
<td>BRDIVP4G</td>
<td>Sample IMS Gen card for the BRD4 transaction</td>
</tr>
<tr>
<td>BRDIVP4P</td>
<td>Sample IMS Program Specification Block (PSB) for the BRD4 Transaction</td>
</tr>
<tr>
<td>BRDIVP4</td>
<td>Assembler program that issues the DB2 database SQL requests</td>
</tr>
</tbody>
</table>

As indicated in the sample library members, you will need to modify the JCL to accommodate your environmental requirements:
Prepare your IMS system for the IVP

1. Add the IVP transaction and application to your IMS system. Using the sample provided in BRDIVP4G, you will need to modify your IMS system definition parameters.

2. Perform an IMS MODBLKS gen.


4. Perform an ACB gen.

5. Copy both MODBLKS and ACB libraries to the inactive libraries defined to your IMS system.

6. Perform an IMS Online Change for both MODBLKS and ACBLIB.

Prepare the IVP

1. Edit and run the JCL in BRDIVAL4. This will assemble and link the IVP program.

2. On every DB2 system that the Subzero IVP will run, edit and run the JCL in member BRDIVBN4.

Executing the IMS IVP transaction

Perform the following procedure to execute the IMS IVP transaction.

Note
You can define the transaction and application to your IMS system using dynamic tools such as BMC Delta for IMS

1. Log onto your IMS system.

2. Execute the IVP transaction (BRD4).

   `BRD4 action,,count`

   where:
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Values</th>
</tr>
</thead>
</table>
| action    | ■ DB2—inserts BMC created records into the DB2 table  
|           | ■ DL—List BMC created records in the DB2 table  
|           | ■ DD—Delete BMC created records from the DB2 table |
| count     | Number of rows to insert into the table |

3  Issue the following transactions from your IMS terminal:

a  Enter **BRD4 DB2,,5**.

This transaction inserts 5 rows into the DB2 table and randomly deletes one row.

**Example**

```
BRD4: Starting.  Compiled on 08/05/15 07.23
BRD4: Input: DB2,,5,,,,
BRD4: DB2 - Start DEHZ
BRD4: DB2 - SSID:DEHZ  Userid:MVSAXL1  Schema:MVSAXL1
BRD4: DB2 - Selected DEPT row for: A00 000010 A00 SPIFFY COMPUTER SERVICE
BRD4: DB2 - DEPT table insert okay, BMC Boulett Moores Cloer
BRD4: DB2 - EMP table insert okay, 506138 Alexander P Rodriguez
BRD4: DB2 - EMP table insert okay, 562983 Bea D Martinez
BRD4: DB2 - EMP table insert okay, 691674 Donna R Gilbert
BRD4: DB2 - EMP table insert okay, 298049 Ben V Taylor
BRD4: DB2 - EMP table insert okay, 166902 Johnie F Quidditch
BRD4: DB2 - 5 rows inserted.
BRD4: DB2 - updated 0 rows in EMP table for Lastname: Lorre
BRD4: DB2 - Selected row for delete: 166902 Johnie F Quidditch
BRD4: DB2 - EMP row deleted okay, 166902 Johnie F Quidditch
BRD4: DB2 - Finished processing
 BRD4: Ended with  
RC=0
```

b  Enter **BRD4 DL**.

This transaction lists the remaining DB2 table entries.
Enter **BRD4 DD**.

This transaction deletes the DB2 table entries.
Controlling Subzero

You can control Subzero, DB2 Component Services (DBC), and associated components with console commands. In addition, you can modify the values in JCL procedures and XML scripts to change certain characteristics of the components.

**Note**

Most command examples in the following topics include the slash that is required when you enter a command in the System Display and Search Facility (SDSF). If you are using another method to enter commands, omit the slash.

Controlling Subzero with console commands

Subzero provides a set of console commands for manually controlling product components and obtaining information.

For example, you can perform any of the following tasks:

- “Starting, restarting, and refreshing Subzero address spaces and tasks” on page 99
- “Terminating Subzero address spaces and tasks” on page 100
- “Displaying statistics about Subzero components” on page 104
- “Displaying information about Subzero options” on page 105
- “Reloading Subzero configuration options and license keys” on page 106
- “Clearing the BRDPRINT data set” on page 107

Overview of Subzero console commands

You can use Subzero console commands for various purposes, including controlling address spaces and getting information. You can enter Subzero console commands...
in a system console, in the System Display and Search Facility (SDSF), or in another facility that accepts console commands.

Observe these guidelines when entering the console commands:

- Enter console commands in SDSF in the following format:
  
  `/dbc BRD,command,commandParameters`

  — `dbc` is the DBC subsystem ID or DBC group name that is managing Subzero.

  — The BRD parameter tells DBC to direct the command to the Subzero controlling agent.

  — `command` is the literal command name.

  — `commandParameters` specifies the parameters for the command (if any).

- Enter a comma but no spaces between BRD, the command name, and the command parameters.

- You can route a command to multiple LPARs by using standard system commands.

- Some commands accept a product instance identifier (PIID) or hexadecimal address space identifier (ASIDX) to identify an address space. You can obtain the PIID and ASIDX from message BMCDBC0131I in the DBCPRINT data set:

  
  ```
  BMCDBC0131I DPR process='productCode.FMID.piid(procName)' PROCID(procID) ASID=decimalAddressSpaceID(hexadecimalAddressSpaceID) STKN='stoken' is status
  ```

  Table 2 on page 97 describes the available Subzero console commands.

**Tip**

To write a list of the latest commands to the BRDPRINT data set, enter the `/dbcssid BRD,HELP` command.

**WARNING**

The FORCE parameter tells Subzero to detach its tasks and terminate without prejudice. If a task is a thread connected to the subsystem (for example IMS or DB2), detach the task could cause the subsystem to abend.

BMC recommends using the FORCE parameter only if all previous attempts to stop the address space have failed. Terminating a process that is still in use can have unpredictable results.
Table 2: Subzero console commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| BRD,ALTS,ENABLE, `subsystemType`, FROM_ID=`subsystemID`, FROM_LPAR=`lpar` | Redirects to alternative targets if the primary task is unavailable. The variables are as follows:  
  - The `subsystemType` is the type of subsystem (MQ, DB2, or IMS)  
  - `subsystemID` specifies the MQ, DB2, or IMS subsystem. Wild cards are supported.  
  - *(optional)* `lpar` specifies the LPAR where the subsystem is installed (wildcards are supported). If you do not use this parameter, the value equates to `*`, which tells Subzero to use alternative targets on matching subsystems on all LPARS. |
<p>| BRD,ALTS,D DISABLE, <code>subsystemType</code>, FROM_ID=<code>subsystemID</code>, FROM_LPAR=<code>lpar</code> | Does not redirect to alternative targets if the primary task is unavailable. |
| BRD,CONFIG,RELOAD | Reread the Subzero option set (containing configuration information) from the repository, and make the new configuration active. |
| BRD,DISPLAY,MAP | Display a map showing how Subzero is configured to redirect subsystem access requests from client to target subsystems, including details of alternative targets. The map also shows any redirection conflicts. |
| BRD,DISPLAY,MAP,FUTURE | Display the configuration that would result if the BRD,CONFIG,RELOAD command were executed now. The configuration from the updated Subzero option set is read and compared with the configuration that is currently in effect. This information includes changes to the status of enabling or disabling alternative targets. |
| BRD,DISPLAY,ASIDX=<code>serverASIDX</code>, STORAGE | Display storage statistics for the Subzero address space that has the specified hexadecimal address space ID (ASIDX) on this LPAR. |
| BRD,DISPLAY,CLIENT, STORAGE | Display storage statistics about the Subzero task that handles clients on this LPAR. |
| BRD,DISPLAY,DB2ID=<code>subsystemID</code>, STORAGE | Display storage statistics for Subzero address spaces that are associated with the specified DB2 subsystem ID or group ID on this LPAR. |
| BRD,DISPLAY,IMSID=<code>IMSID</code>, STORAGE | Display storage statistics for Subzero address spaces that are associated with the specified IMS region on this LPAR. |</p>
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRD,DISP,PIID=serverPIID,STORAGE</td>
<td>Display storage statistics for the Subzero address space that has the specified product instance ID (PIID) on this LPAR.</td>
</tr>
<tr>
<td>BRD,HELP</td>
<td>Display available operator command syntax for Subzero.</td>
</tr>
<tr>
<td>BRD,KILL,ASIDX=serverASIDX,TASK=SubzeroTaskNumber</td>
<td>Terminate the specified task, which is executing in the Subzero server address space that has the specified ASIDX.</td>
</tr>
<tr>
<td>BRD,KILL,CLIENT,TASK=SubzeroTaskNumber</td>
<td>Terminate the specified task that is executing in the Subzero controlling agent.</td>
</tr>
<tr>
<td>BRD,KILL,PIID=serverPIID,TASK=SubzeroTaskNumber</td>
<td>Terminate the specified task, which is executing in the Subzero server address space that has the specified PIID.</td>
</tr>
<tr>
<td>BRD/LICENSE,RELOAD</td>
<td>Reload all available Subzero product license keys (CPU authorization passwords), and validate them.</td>
</tr>
<tr>
<td>BRD,PRTFLUSH</td>
<td>Close the BRDPRINT and BRDERROR data sets that Subzero uses within the DBC on this LPAR, and create new data sets.</td>
</tr>
<tr>
<td>BRD,PRTFLUSH,ALL</td>
<td>Close the BRDPRINT and BRDERROR data sets that Subzero uses within the DBC on all LPARs, and create new data sets.</td>
</tr>
<tr>
<td>BRD,REFRESH</td>
<td>Stop Subzero on this LPAR, and restart it from the DBC repository definitions.</td>
</tr>
<tr>
<td>BRD,RESTART</td>
<td>Stop Subzero on this LPAR, and restart it based on current in-memory DBC definitions.</td>
</tr>
<tr>
<td>BRD,START</td>
<td>Start Subzero on this LPAR if not already running.</td>
</tr>
<tr>
<td>BRD,STOP</td>
<td>Stop all Subzero address spaces on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,ASIDX=serverASIDX</td>
<td>Stop the Subzero server address space that has the specified ASIDX on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,CLIENT</td>
<td>Stop the Subzero controlling agent task that handles clients on this LPAR. <strong>Note:</strong> Use this command only at the direction of BMC Customer Support.</td>
</tr>
<tr>
<td>BRD,STOP,DB2ID=subsystemID</td>
<td>Stop all Subzero address spaces that are associated with the specified DB2 subsystem ID or group ID on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,IMSID=IMSID</td>
<td>Stop all Subzero address spaces that are associated with the specified IMS subsystem on this LPAR.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>BRD,STOP,MQID=\textit{mqID}</td>
<td>Stop all Subzero subsystem server address spaces that are performing work on the specified MQ subsystem on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,PIID=\textit{serverPIID}</td>
<td>Stop the Subzero server address space that has the specified PIID on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,FORCE</td>
<td>Forcefully stop all Subzero address spaces on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,FORCE,ASIDX=\textit{serverASIDX}</td>
<td>Forcefully stop the Subzero subsystem server address space on this LPAR that matches the specified ASIDX.</td>
</tr>
<tr>
<td>BRD,STOP,FORCE,CLIENT</td>
<td>Forcefully stop the Subzero client intercept task on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,FORCE,DB2ID=\textit{subsystemID}</td>
<td>Forcefully stop all Subzero subsystem server address spaces that are performing work on the specified DB2 subsystem on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,FORCE,IMSID=\textit{subsystemID}</td>
<td>Forcefully stop all Subzero subsystem server address spaces that are performing work on the specified IMS subsystem on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,FORCE,MQID=\textit{mqID}</td>
<td>Forcefully stop all Subzero subsystem server address spaces that are performing work on the specified MQ subsystem on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,FORCE,PIID=\textit{serverPIID}</td>
<td>Forcefully stop the Subzero subsystem server address space on this LPAR that has the specified PIID on this LPAR.</td>
</tr>
</tbody>
</table>

**Starting, restarting, and refreshing Subzero address spaces and tasks**

Use the following procedures to manually start Subzero address spaces and tasks on the LPAR in which you enter the command.

**To start a Subzero controlling agent**

1. Enter the following command to tell DBC to initialize the Subzero controlling agent on this LPAR:

   \texttt{/dbcssid BRD,START}

   This command is useful if the controlling agent has stopped (because of a command or failure), and you want it to start running again without restarting DBC. If the controlling agent is already running on this LPAR, DBC ignores the command.
To start a Subzero controlling agent task that handles clients

1 Enter the following command to request that the Subzero controlling agent initialize the task that handles clients on this LPAR:

```
/dbcssid BRD,START,CLIENT
```

This command is useful if the task that handles clients has stopped (because of a command or failure), and you want it to start running again without restarting the controlling agent. The task will initialize only if the current configuration indicates that this LPAR is a potential client LPAR for redirection. If the task is already running on this LPAR, the controlling agent ignores the command.

To restart a Subzero controlling agent

1 Enter the following command to tell DBC to stop the Subzero controlling agent on this LPAR, and restart it by using the definitions that are currently in DBC's memory:

```
/dbcssid BRD,RESTART
```

This command causes Subzero to use the latest load modules from the load library. This command is useful when you need to "bounce" (stop and restart) the controlling agent (such as after applying maintenance), but you have not changed the Subzero startup procedure. If the controlling agent is not already running on this LPAR, DBC starts it.

To refresh a Subzero controlling agent

1 Enter the following command to tell DBC to stop the Subzero controlling agent on this LPAR, and restart it by using the definitions that are in the DBC repository:

```
/dbcssid BRD,REFRESH
```

This command is useful when you have changed the BRD startup procedure, and you want the changes to take effect immediately. This command also causes Subzero to use the latest load modules from the load library. If the controlling agent is not already running on this LPAR, DBC starts it.

Terminating Subzero address spaces and tasks

Use the following procedures to manually terminate Subzero address spaces and tasks on the LPAR in which you enter the command.
Tip

The information that precedes the BMCBRD message number in the BMCPRINT listing displays the:

- ID of the LPAR on which you should enter the BRD,STOP or BRD,KILL command
- Address space ID that is running the address space or task to be terminated

For example, the following message shows that the LPAR ID is SYSM, and the address space ID is 0129:

13:36:40 SYSM ASID(0129): BMCBRD0350W IMS Thread waiting for DBCTL response ...

For waiting or suspended Subzero tasks, Subzero messages provide information that you can use in a BRD,KILL command to terminate the task. These messages identify the waiting or suspended thread and its associated Subzero task number.

- In the Subzero subsystem server, the server monitors thread connections between itself and the supported IBM IMS or DB2 subsystem. Subzero issues message BMCBRD0350W or BMCBRD0351W (for IMS or DB2, respectively) when both of the following conditions exist:
  - A thread has issued a subsystem request (for example, an IMS database request).
  - The thread has waited for a response for longer than 10 seconds.

- In the Subzero controlling agent, the agent monitors client thread connections between the client address space (IBM CICS) and its remote Subzero subsystem server. Subzero issues message BMCBRD0352W if a client thread has waited for more than 60 seconds for a response from the remote server.

WARNING

Before you terminate a Subzero address space, execute the processes and procedures that properly terminate a connection between a client and its connected subsystem. If a server shuts down while active connections exist between the client and its remote subsystem, abends can occur in the Subzero, client, and subsystem address spaces.

BMC recommends using the STOP,FORCE commands only if all previous attempts to stop the address space have failed. Terminating a process that is still in use can have unpredictable results.
To stop all Subzero address spaces

1 Enter the following command to tell DBC to stop executing all Subzero controlling agents and subsystem servers on this LPAR:

```
/dbcssid BRD,STOP
```

The forcible STOP command format is:

```
/dbcssid BRD,STOP,FORCE
```

This command is useful for terminating all Subzero activity on an LPAR. If no Subzero address spaces are running on this LPAR, DBC ignores the command.

To stop the Subzero controlling agent task that handles clients

1 Enter the following command to tell the Subzero controlling agent to stop execution of the task that handles clients on this LPAR:

```
/dbcssid BRD,STOP,CLIENT
```

The forcible STOP command format is:

```
/dbcssid BRD,STOP,FORCE,CLIENT
```

This command is useful if you need to stop the task without stopping the controlling agent. This command terminates redirection for all clients. If the task is not running on this LPAR, the controlling agent ignores the command.

To stop a specified Subzero subsystem server

1 Enter the following command to tell DBC to stop executing a specified subsystem server on this LPAR:

```
/dbcssid BRD,STOP,subsytemType=subsytemID
```

The forcible STOP command format is:

```
/dbcssid BRD,STOP,FORCE,subsytemType=subsytemID
```

In this command, `subsytemType` indicates the type of the target subsystem (DB2ID, or IMSID) that is associated with this subsystem server. `subsytemID` specifies the IMS or DB2 subsystem ID (or DB2 group ID) of the target subsystem.

This command is useful if you need to stop the subsystem server without stopping the controlling agent or the associated target subsystem; for example, you might use this command because a subsystem server problem occurred. This command stops redirection between the client application and the target.
subsystem. If the application is still sending requests, it receives a failure. If the subsystem server is not running on this LPAR, the controlling agent ignores the command.

**To stop a specified Subzero address space**

1. Enter the following command to tell DBC to stop executing a specified Subzero address space on this LPAR:

   `/dbcssid BRD,STOP,IDType=ID`

   The forcible STOP command format is:

   `/dbcssid BRD,STOP,FORCE,IDType=ID`

   The variables are as follows:

   - **IDType** indicates the type of identifier you are entering (PIID or ASIDX).
   - **ID** specifies the product instance ID or hexadecimal address space ID of the address space that you want to stop.

   This command is useful if you need to stop the specified address space. If that address space is not running on this LPAR, DBC ignores the command.

**To terminate (kill) a waiting task in a Subzero subsystem server**

1. Enter the following command to terminate execution of a waiting thread (task) in a Subzero subsystem server on this LPAR:

   `/dbcssid BRD,KILL,IDType= subsystemID ,TASK=SubzeroTaskNumber`

   The variables are as follows:

   - **IDType** indicates the type of identifier you are entering product instance ID (PIID) or hexadecimal address space ID (ASIDX).
   - **subsystemID** specifies the PIID or ASIDX of the task to be terminated.
   - **SubzeroTaskNumber** specifies the internal task identifier of the Subzero task in the subsystem server.

   You can obtain the PIID and ASIDX and Subzero task number from message BMCBRD0350W (for an IMS thread) or BMCBRD0351W (for a DB2 thread).

   This command is valid for terminating the specified task if that task is in a wait state, as indicated by message BMCBRD0350W or BMCBRD0351W. If that task is
not in a wait state or is not running on this LPAR, the address space ignores the command.

**To terminate (kill) a waiting task in a Subzero controlling agent**

1. Enter the following command to terminate execution of a waiting thread (task) in the Subzero controlling agent on this LPAR:

   ```
   /dbcssid BRD,KILL,CLIENT,TASK=SubzeroTaskNumber
   ```

   *SubzeroTaskNumber* specifies the internal task number of the Subzero task in the controlling agent. You can find the task number in the BMCBRD0352W message.

   This command is valid for terminating the specified task if that task is in a wait state, as indicated by message BMCBRD0352W. If that task is not in a wait state or is not running on this LPAR, the controlling agent ignores the command.

**Displaying statistics about Subzero components**

Use the following procedures to obtain statistical information about Subzero address spaces and tasks.

In response, Subzero will write relevant messages to the BRDPRINT data set in the DBC address space.

**To display statistics for a specified address space**

1. Enter the following command to obtain information about internal memory for a specified address space on this LPAR:

   ```
   /dbcssid BRD,DISPLAY,IDType=ID,STORAGE
   ```

   The displayed information follows message BMCBRD6184I in the BRDPRINT data set.

   The variables are as follows:

   - *IDType* indicates the type of identifier you are entering:
     - **DB2ID** for a DB2 subsystem ID or group ID that is associated with a subsystem server
     - **IMSID** for an IMS control region ID that is associated with a subsystem server
     - **PIID** for the product instance ID of the address space
— **ASIDX** for the hexadecimal address space ID of the address space

- **ID** specifies the identifier of the address space.

**To display statistics for a specified controlling agent task that handles clients**

1. Enter the following command to obtain information about internal memory for the controlling agent task that handles clients on this LPAR:

   `/dbcssid BRD,DISPLAY,STORAGE`

   The displayed information follows message BMCBRD6181I in the BRDPRINT data set.

**Displaying information about Subzero options**

Use the following procedures to display the Subzero options that define the current state of redirection activities, and the future state of redirection activities if the options are reloaded.

**To display current options**

1. Enter the following command on any LPAR that belongs to the DBC group:

   `/dbcssid BRD,DISPLAY,MAP`

   In response, Subzero writes information about options that are currently in effect. This information follows message BMCBRD6170I in the BRDPRINT data set in the DBC address space.

   The information also specifies, in BMC6292W message rows, any redirectional conflicts such as redirection to unavailable targets, or to unsupported DB2 or IMS subsystems. This information appears below the relevant controlling agent/client status row.

**To display future options**

1. Enter the following command on any LPAR that belongs to the DBC group:

   `/dbcssid BRD,DISPLAY,MAP,FUTURE`

   In response, Subzero writes information about how the redirection activities, which are currently in effect will change when the option set is reloaded from the repository in response to the BRD,CONFIG,RELOAD command. The information follows message BMCBRD6120I in the BRDPRINT data set in the DBC address space.
Reloading Subzero configuration options and license keys

Use the following procedures to reload option sets or product license keys (CPU authorization passwords) manually. You can issue these commands at almost any time during Subzero execution.

The option set contains configuration options that specify how to redirect subsystem access requests between client and target participants. Reloading the option set causes any changes to take effect immediately. For participants that have not changed, loading a new option set has no effect.

**Example**

The following scenario describes the effect of making a typical change to an option set and using the reload command to make the change effective:

1. The option set that is currently in effect contains this definition: **DB2A on SYSA will redirect to DB2B on SYSB.**

2. An application is currently running on SYSA and sending requests to DB2A, which Subzero is redirecting to DB2B on SYSB.

3. You change the option set so that it contains this definition: **DB2A on SYSA will redirect to DB2C on SYSC.**

4. You enter the `/dbcssid BRD,CONFIG,RELOAD` command.

5. The Subzero controlling agent recognizes that DB2B is no longer defined as a target subsystem. In response, it requests a shutdown of the existing subsystem server for DB2B on SYSB.

6. The next subsystem access request that the application sends to DB2A fails because the subsystem server for redirection to DB2B has stopped.

7. The application sends a request to reconnect to DB2A.

8. The controlling agent on SYSC receives the reconnection request because the new target is DB2C on SYSC, and starts a subsystem server for DB2C.

9. The subsystem server for DB2C on SYSC establishes an XCF connection with the application, and now receives subsequent subsystem access requests from that application.

**To reload an option set**

1. Make the required changes to the Subzero option set, and save your changes.
For more information, see “Defining Subzero redirections” on page 48.

2 Enter the following command to tell the Subzero controlling agent to obtain the latest version of the option set from the LGC repository, and use that set instead of the current set:

```
/dbcssid BRD,CONFIG,RELOAD
```

To reload license information

1 Make the required changes to install, change, or delete passwords in your password data set, and save your changes.

For more information, see the Installation System Reference Manual.

2 Enter the following command to revalidate passwords within the product:

```
/dbcssid BRD,LICENSE,RELOAD
```

Clearing the BRDPRINT data set

Use the following procedure to close the Subzero BRDPRINT and BRDERROR data sets and create new data sets.

For example, you might want to clear the data sets if they have become too big to use efficiently.

To clear the BRDPRINT data set on this LPAR

1 Enter the following command in the LPAR for which you want to clear the Subzero print data sets:

```
/dbcssid BRD,PRTFLUSH
```

To clear the BRDPRINT data set for all LPARs

1 Enter the following command in any LPAR in the DBC group:

```
/dbcssid BRD,PRTFLUSH,ALL
```
Controlling Subzero with BRDPROC and BRDINIT

Subzero provides a JCL procedure (BRDPROC) and initialization script (BRDINIT) that DBC uses to automate startup of Subzero tasks. They are included in the BRDSAMP library. Optionally, you can edit BRDPROC and selected statements in BRDINIT to accommodate your site’s preferences.

The following topics provide more information:

■ “Optionally editing the Subzero startup procedure (BRDPROC)” on page 108
■ “Optionally editing the Subzero initialization script (BRDINIT)” on page 109

Optionally editing the Subzero startup procedure (BRDPROC)

During customization, the Installation System creates a JCL procedure named BRDPROC, which initializes Subzero agents on an LPAR. You can change BRDPROC as needed.

The Installation System creates a copy of BRDPROC in the JCL data set and customizes the procedure with your installed libraries:

```
//BRDPROC PROC ACC=,
//             TIM=1440
//EAGENT   EXEC PGM=BRDJANUS,ACCT=&ACC,TIME=&TIM,MEMLIMIT=NOLIMIT
//STEPLIB  DD DISP=SHR,DSN=@hlq.BMCLINK
//         DD DISP=SHR,DSN=SYS1.CSSLIB
//DBCPRINT DD SYSOUT=*;
//BMCPSWD  DD DISP=SHR,DSN=@hlq.BMCPSWD
```

Table 3: Description of statements

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC</td>
<td>Specifies BRDPROC as the procedure name and includes the ACC parameter in case the EAGENT step requires account information</td>
</tr>
<tr>
<td>BRDJANUS</td>
<td>Checks elements in the environment and starts the DBC agent stub; the agent stub ultimately starts the Subzero controlling agent code as a task. Subzero agents start with a JOBNAME parameter to distinguish tasks that are based on the Subzero XML definition to DBC.</td>
</tr>
</tbody>
</table>
### Statement Description

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td><em>(required)</em> Includes the BMC data sets that contain BRD, DBC, LGC, SCC, USC, and Dignus modules. The library names depend on options specified during installation. If one or more IMS subsystems will participate in Subzero redirection, the STEPLIB statement might need to include the library that contains the DFSPZPxx module; this module contains startup parameters for the IMS database resource adapter (DRA). For more information, see “IMS requirements” on page 37. All libraries in the STEPLIB concatenation must be APF authorized.</td>
</tr>
<tr>
<td>DBCPRINT</td>
<td>Identifies the data set to which the DBC external agent stub writes a small amount of information</td>
</tr>
<tr>
<td>BMCPSWD</td>
<td>Identifies the data set that contains the BMC license keys (CPU authorization passwords) that enable Subzero use in your environment</td>
</tr>
<tr>
<td>BRDPRINT and BRDERROR (not shown)</td>
<td>Identify the data sets that will contain message and error information generated during execution. The product dynamically allocates these data sets if they are not defined in the procedure.</td>
</tr>
</tbody>
</table>

**Note**

Do not attempt to start Subzero manually. DBC must always start Subzero, either automatically during DBC startup or in response to a BRD,START console command.

### To edit the startup procedure

1. Access BRDPROC from the JCL library.
2. Edit BRDPROC as required.

### Optionally editing the Subzero initialization script (BRDINIT)

During customization, the Installation System creates an initialization script named BRDINIT, which contains XML statements that define Subzero parameters to DBC. You can change certain definitions in the script as needed.

**Note**

Do not edit statements other than those described in this topic.

The Installation System stores the script as member BRDINIT in the runtime XML library (BMCXML or xxXML) that is defined to DBC:

```xml
<DBCUTIL>
  <OPTIONS>
    <MAXRESPONSES>1</MAXRESPONSES>
  </OPTIONS>
  <COMMANDS>
</DBCUTIL>
```
To edit the initialization script

1. Access BRDINIT in the runtime XML library.

2. Edit any of the following statements (but do not edit anything else in the script):

   - `<PROCESS>` specifies the procedure that DBC uses to start Subzero agents. For `procName`, you can specify any valid procedure name instead of the default BRDPROC. Ensure that you also use this name as the name of the corresponding member in a data set in your SYSPROC concatenation.

   - JOBNAME=jobName specifies the job name of the Subzero controlling agent. For `jobName`, you can specify any valid job name (up to eight characters) instead of the default BRDCNTL. By default, the job names of Subzero subsystem servers are based on the JOBNAME value unless you define a naming pattern for the subsystem servers:

     —DBC retains up to the first four characters of the controlling agent's JOBNAME.

     —Subzero appends the four character subsystem ID of the associated DB2 or IMS subsystem.
—The name is padded with blanks to ensure an eight character JOBNAME.

In the job name, you can specify any symbolic value that DBC supports.

---
**Example**

JOBNAME=SSID.CNTL

---
During execution, the system constructs the name by substituting current values for the symbolic values. In this example, the system substitutes the DBC subsystem ID.

---
**Note**

Ensure that the constructed length of the job name does not exceed 8 characters.

---
For information about supported symbolic values, see the *BMC Infrastructure Components Administration Guide*.

---
**Note**

For more information about defining a naming pattern for the subsystem servers, see “Overview of Subzero configuration options in the LGC” on page 56.

---

### Managing DBC operations

DBC is the supervising address space that manages Subzero address spaces. This topic explains how to start the DBC address space manually (if needed) or stop the DBC subsystem, and briefly describes DBC commands that you can use.

For complete information, see the *BMC Infrastructure Components Administration Guide*.

---

### Overview of the DBC started task procedure

DBC typically runs as a persistent started task in each IBM z/OS system that participates in Subzero redirections.

For a permanent implementation, DBC should run as a started task. For a trial implementation, you can run DBC as a batch job, but the JES initiator will remain busy as long as DBC continues to run.
**Example**

This example shows JCL for a typical started task:

```plaintext
//DBC PROC SSID=,GRP=,REPMODE='  
//DBC EXEC PGM=DBCMAIN,REGION=OM,ACCT=&ACC,TIME=&TIM,  
//                  PARM='SSID=&SSID,GROUP=&G,TRACE=&T'  
//STEPLIB DD DSN=BMCBRD.TEST.BMCLINK,DISP=SHR  
//DBCPRINT DD SYSOUT=*,RECFM=VA  
//BMCPSWD DD DSN=BMCBRD.TEST.BMCLINK,DISP=SHR  
//DBCPARMS DD DSN=BMCBRD.TEST.BMCSAMP($DBC&SSID),DISP=SHR  
//DBCSECUR DD DISP=DSN=BMCBRD.TEST.BMCSAMP($SEC&SSID),DISP=SHR
```

Sample JCL for executing DBC as a started task is located in member DBC$STC of the SAMP library.

The JCL includes these statements:

- **PROC** specifies parameters for the procedure, if needed.
- **EXEC** specifies the program (DBCMAIN) to execute, accounting information, and parameters.
- **STEPLIB** specifies the library containing the DBC executable load modules. All libraries in STEPLIB must be APF authorized.
- **DBCPRINT** identifies the data set to contain DBC messages.
- *(optional)* **BMCPSWD** identifies the data set that contains your product authorization passwords (license keys).
- **DBCPARMS** identifies the data set member that contains DBC initialization parameters. During customization, the Installation System populates this data set with parameters. You can change these parameters as needed.
- **DBCSECUR** identifies the data set member that contains DBC security parameters. During customization, the Installation System populates this data set with parameters. You can change these parameters as needed.

## Starting the DBC address space manually

You must start DBC in each IBM z/OS system that participates in Subzero redirections. You can start DBC manually if it is not automatically started.

### To start the DBC address space manually

1. Display the JCL procedure (PROC) to execute the DBC address space, and edit it if needed.
2 Enter the following command to execute the PROC:

```
/S procName
```

where `procName` is the name of the PROC.

---

**Note**

DBC must start before any CICS subsystem that will participate in Subzero redirection.

---

### Stopping the DBC subsystem

Use the following procedure to stop the DBC subsystem through an IBM z/OS console command. You identify the DBC subsystem to stop through the subsystem ID (`ssid`) or XCF group value (`group`).

---

**WARNING**

Shutting down the DBC subsystem also shuts down all associated Subzero servers used by BMC Subsystem Optimizer for zEnterprise. If a server shuts down while active connections exist between the client and its remote subsystem, abends can occur in the Subzero, client, and subsystem address spaces.

Before you terminate the DBC subsystem or any Subzero address space, execute the processes and procedures that properly terminate a connection between a client and its connected subsystem.

---

### To stop the DBC subsystem

1 Enter one of the following z/OS commands:

- `ssid STOP`
- `group STOP`
- `MODIFY ssid,STOP`

---

**Example**

Assume that a DBC subsystem has an SSID of DBC1. You can stop this subsystem by issuing the STOP command as a MODIFY command:

```
F DBC1,STOP
```
WARNING

Some product processes might require an extended amount of time to terminate successfully. If DBC shutdown is delayed, review DBCPRINT for the following message to determine the cause of the delay:

BMCDBC0232I … DBC is waiting for product \textit{productName} to shutdown.

If the DBC does not shut down after a reasonable amount of time, you can use the following command to terminate the delayed process:

\texttt{ssid STOP,FORCE}

Use caution when doing so; terminating a process that is still in use can have unpredictable results. BMC recommends that you use the \texttt{STOP,FORCE} command only if preceding attempts to stop the DBC have failed.

## DBC commands

You can interact with the DBC subsystem by using any of the following commands.

The \textit{ssid} variable refers to the subsystem ID of the DBC subsystem.

### Table 4: DBC commands

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ssid DBC,DIAG</td>
<td>Sends diagnostic information to the DBCPRINT and DBCTRACE data sets. BMC Support might direct you to execute this command to assist in diagnosing problems.</td>
</tr>
<tr>
<td>/ssid DBC,DB2C,ssid.command</td>
<td>Executes a DB2 command in the DB2 subsystem identified by &quot;ssid&quot; and returns the result to DBCPRINT. This command can be used to execute any native DB2 command that the user is authorized for. <strong>Note</strong>: This command cannot be used for DSN subcommands.</td>
</tr>
<tr>
<td>/ssid DBC,RESETPRINT</td>
<td>Closes the DBCPRINT data set and allocates a new one in order to clear data from DBCPRINT.</td>
</tr>
<tr>
<td>/ssid DBC,RESETTRACE</td>
<td>Clears the DBCTRACE data set and allocates a new one.</td>
</tr>
<tr>
<td>/ssid STOP</td>
<td>Stops the DBC subsystem.</td>
</tr>
<tr>
<td>/ssid DBC,TRACEOFF</td>
<td>Disables DBC tracing.</td>
</tr>
<tr>
<td>/ssid DBC,TRACEON</td>
<td>Enables DBC tracing. DBC writes trace data to the DBCTRACE data set. The DBCTRACE data set is dynamically allocated. <strong>WARNING</strong>: If you leave tracing enabled, be aware that the DBCTRACE data set can grow to a significant size. BMC recommends using caution when using the \texttt{TRACEON} command.</td>
</tr>
</tbody>
</table>

114  
BMC Subsystem Optimizer for zEnterprise Administration Guide
**Table 5: DISPLAY Command Options**

<table>
<thead>
<tr>
<th>DISPLAY object (alternative syntax)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCTS (PRODUCTS, PRD, PDXD) [,ppp]</td>
<td>Displays the Product Definition XML Document (PDXD) from the DBC repository. If no parameters are specified, information is displayed for all products. If the three-character product code is specified, DBC displays information for only that product. If multiple PIIDS exist for that product code, they are all displayed.</td>
</tr>
<tr>
<td>CMDS [,ppp]</td>
<td>Displays all operator command definitions. If no parameters are specified, commands for all products are displayed. If the three-character product code is specified, DBC displays only commands for that product.</td>
</tr>
<tr>
<td>EVENT (EV),eventName [subid]</td>
<td>Displays all subscription information the named event. If the subid is specified, only information for the named subscription is displayed.</td>
</tr>
<tr>
<td>CMD,cmdtext</td>
<td>Displays the ACTION XML for the named command.</td>
</tr>
<tr>
<td>AGENTS (AG) [,ppp]</td>
<td>Displays all agents that are currently executing. This command generates message BMCDBC0126I, which indicates the TCB address of the agent and whether it is a job step TCB. If no parameters are specified, this command displays information for all products. If the three-character product code is specified, displays currently execution agents for only the product specified.</td>
</tr>
<tr>
<td>DISPLAY object (alternative syntax)</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>PROCESSES</td>
<td>Displays all active PROCESSES. If no other parameters are specified, PROCESSES for all products are displayed. If the three-character product code is specified, displays currently active PROCESSES for only the product specified.</td>
</tr>
<tr>
<td>DB2 [,ssid][,LOCAL</td>
<td>ALL ]</td>
</tr>
<tr>
<td>IMS [,imsid][,LOCAL</td>
<td>ALL ]</td>
</tr>
<tr>
<td><strong>DISPLAY object (alternative syntax)</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>MQ [,ssid][,LOCAL</td>
<td>ALL ]</td>
</tr>
<tr>
<td>SRBS [,ppp ]</td>
<td>Displays currently active SRBs. If no parameters are specified, SRBs for all products are displayed. If the three-character product code is specified, displays currently active SRBs only for the product specified</td>
</tr>
<tr>
<td>STATS [,TCBS,name] [,MEM][,QUEUES]</td>
<td>If no parameters are specified, this command displays all available statistics. name displays statistics for only the named task. MEM displays memory usage statistics. QUEUES displays queue usage statistics.</td>
</tr>
<tr>
<td>DSNS</td>
<td>Displays all data sets currently allocated and specifies if the data set allocation results from STEPLIB, product, or agent definition</td>
</tr>
<tr>
<td>SOCKETS</td>
<td>Displays the state of all sockets currently in use.</td>
</tr>
</tbody>
</table>
Managing DBC operations
Maintaining Subzero

This chapter provides recommendations for maintaining the Subzero product.

Recommended responses to Subzero component failures

This topic explains what happens when the address space for a required Subzero component terminates, either normally or due to failure. The topic also recommends how to respond to the terminations.

Failure types

Normal termination occurs when you issue an appropriate shutdown command to the address space:

<table>
<thead>
<tr>
<th>The address space for . . .</th>
<th>Terminates normally when you issue . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBC</td>
<td>*/P dbcid</td>
</tr>
<tr>
<td>Subzero controlling agent</td>
<td>dbcid BRD,STOP</td>
</tr>
<tr>
<td>Subzero subsystem server</td>
<td>dbcid BRD,STOP,ASIDX=nnnnn</td>
</tr>
</tbody>
</table>

In contrast, the following types of address-space terminations indicate component failures:

Cancel

You issue a CANCEL command for the address space.

Force

You issue a z/OS FORCE command following an unsuccessful CANCEL command.
**Note**

Using the FORCE command to terminate a Subzero address space can cause unpredictable results.

---

**Abend**

A software failure causes the address space to terminate abnormally.

**LPAR termination**

An LPAR with an active Subzero environment terminates.

## Results and recovery after a component terminates or fails

For the Subzero components, Table 1 on page 57 summarizes the events that occur when a component terminates normally or due to failure. Table 7 on page 123 recommends steps to take in response to the terminations or failures.

The following guidelines apply:

- In these tables, the failure descriptions reflect worst-case scenarios. Your experience might be more moderate based on environmental variations at your site. However, BMC recommends developing recovery/restart strategies that could accommodate the worst-case scenarios. Use Table 7 on page 123 as a guide.

- In some cases, a particular termination will trigger another termination. For example, terminating the DBC address space also terminates the address space for the Subzero controlling agent. In that case, you should review the termination results for the DBC and for the controlling agent.

- Following a termination, the address space takes independent actions to attempt recovery and cleanup. The tables in this topic include some redundancy, reflecting the redundancy in the failure-recovery process.

### Table 6: Termination results for Subzero components

<table>
<thead>
<tr>
<th>Component</th>
<th>Normal termination</th>
<th>Cancel/Force/Abend failure</th>
<th>LPAR failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTCS</td>
<td>Not applicable</td>
<td>The DBC address space</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>terminates.</td>
<td></td>
</tr>
<tr>
<td>DBC</td>
<td>The address space</td>
<td>The address space</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>for the Subzero</td>
<td>for the Subzero</td>
<td></td>
</tr>
<tr>
<td></td>
<td>controlling agent</td>
<td>controlling agent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>terminates.</td>
<td>terminates.</td>
<td></td>
</tr>
<tr>
<td>LGC</td>
<td>The address space</td>
<td>The address space</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>for the Subzero</td>
<td>for the Subzero</td>
<td></td>
</tr>
<tr>
<td></td>
<td>controlling agent</td>
<td>controlling agent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>terminates.</td>
<td>terminates.</td>
<td></td>
</tr>
</tbody>
</table>
### Recommended responses to Subzero component failures

<table>
<thead>
<tr>
<th>Component</th>
<th>Normal termination</th>
<th>Cancel/Force/Abend failure</th>
<th>LPAR failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subzero controlling agent</td>
<td>1 All address spaces for Subzero subsystem servers terminate.</td>
<td>1 All address spaces for Subzero subsystem servers terminate.</td>
<td>1 XCF notifies other members in the Subzero XCF group that a member has terminated.</td>
</tr>
<tr>
<td></td>
<td>2 Active client threads abend with a U3549-AAC error. The active connection to the target subsystem terminates.</td>
<td>2 Active client threads abend with a U3549-AAC error. The active connection to the target subsystem terminates.</td>
<td>2 Subzero address spaces on non-failing LPARs are responsible for terminating any active threads or connections that are facilitated by them and are associated with the terminating Subzero or client address space.</td>
</tr>
<tr>
<td></td>
<td>3 XCF notifies other members in the Subzero XCF group that a member has terminated.</td>
<td>3 XCF notifies other members in the Subzero XCF group that a member has terminated.</td>
<td></td>
</tr>
</tbody>
</table>

### Multiple LPAR components

<table>
<thead>
<tr>
<th>Subsystem server or servers</th>
<th>(DB2 only) If a client is configured to reconnect, the subsystem server automatically restarts.</th>
<th>(DB2 only) If a client is configured to reconnect, the subsystem server automatically restarts.</th>
<th>(DB2 only) If a client is configured to reconnect, the subsystem server automatically restarts.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Active client threads abend with a U3549-AAC error. The active connection to the target subsystem terminates.</td>
<td>2 Active client threads abend with a U3549-AAC error. The active connection to the target subsystem terminates.</td>
<td>2 Subzero address spaces on non-failing LPARs are responsible for terminating any active threads or connections that are facilitated by them and are associated with the terminating Subzero or client address space.</td>
</tr>
<tr>
<td></td>
<td>3 XCF notifies other members in the Subzero XCF group that a member has terminated.</td>
<td>3 XCF notifies other members in the Subzero XCF group that a member has terminated.</td>
<td>2 Subzero address spaces on non-failing LPARs are responsible for terminating any active threads or connections that are facilitated by them and are associated with the terminating Subzero or client address space.</td>
</tr>
</tbody>
</table>
### Component | Normal termination | Cancel/Force/Abend failure | LPAR failure
---|---|---|---
Client subsystem or subsystems | 1. The Subzero subsystem server terminates the active threads and the connection to the target subsystem. 2. The Subzero subsystem server enters a wait state, anticipating an automatic reconnection from the client LPAR. If the reconnection does not occur within 5 minutes, the subsystem server terminates. **Note:** Closing a client connection has the same effect. | 1. The client address space might attempt recovery to terminate active threads and connections. 2. The Subzero subsystem server enters a wait state, anticipating an automatic reconnection from the client LPAR. If the reconnection does not occur within 5 minutes, the subsystem server terminates. **Note:** Closing a client connection has the same effect. | Not applicable
Target subsystem or subsystems | The associated Subzero subsystem server address space terminates. | If the target subsystem has relevant recovery routines, those routines communicate their failure back to their client. | Not applicable

- The client (for example, IBM CICS) and the remote target (for example, IBM DB2) maintain unit-of-recovery (UOR) data. In the event of a failure, these systems resolve any in-doubt UORs at reconnection time. Reconnection can occur with or
without using Subzero; using Subzero does not change any database or transaction recovery processes.

- The goal of all Subzero recovery routines (ESTAEs, ARRs, FRRs, and resource managers) is to meet the following requirements:
  
  — Ensure that the client connection and threads terminate properly.
  
  — Ensure that the connection and threads are available for automatic or manual reconnection without needing to stop and restart any additional component address spaces.

Table 7: Recommended recovery steps following component terminations

<table>
<thead>
<tr>
<th>Component</th>
<th>Recommended steps after a normal, Cancel, Force, or Abend termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTCS</td>
<td>1 Restart RTCS by using the console command <code>S RTCSProcName</code>.</td>
</tr>
<tr>
<td>DBC</td>
<td>1 Restart DBC by using the console command <code>S DBCProcName</code>. Note: DBC should automatically start the LGC agent and the controlling agent. If either agent fails to start, start it manually.</td>
</tr>
<tr>
<td>LGC</td>
<td>1 Restart LGC by using the console command <code>dbcid LGC,START</code>.</td>
</tr>
</tbody>
</table>
| Subzero controlling agent        | 1 Restart the controlling agent by using the console command `dbcid BRD,START`.  
  2 Restart the client connection (for example, restart a CICS client with CICS DSNC START or, for a batch, by resubmitting the batch job). |
| Subsystem server or servers      | The subsystem server will reconnect automatically, if still active, or will restart automatically when the client connection restarts (for example, with CICS DSNC START or, for a batch, when the batch job is resubmitted). |
| Client subsystem or subsystems   | 1 Restart the client connection (for example, with CICS DSNC START or, for a batch, by resubmitting the batch job). The subsystem server will reconnect automatically, if still active, or restart automatically when the client connection restarts (for example, with CICS DSNC START or, for a batch, when the batch job is resubmitted). |
| Target subsystem or subsystems   | 1 Restart the target subsystem.  
  2 Restart the client connection (for example, with CICS DSNC START or, for a batch, by resubmitting the batch job). |
### Sample scenarios

This topic provides sample scenarios to help you resolve Subzero failures.

#### Subsystem failure

This topic describes a subsystem failure involving a DB2 sharing group.

For more information, see Knowledge Article 000115342 on BMC Support Central.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td>Assume that you have an IBM DB2 sharing group on LPARs A and B and a CICSPlex on LPARs C and D, using group attach.</td>
</tr>
<tr>
<td>Component failure</td>
<td>The DB2 subsystem fails on LPAR A.</td>
</tr>
<tr>
<td>User response</td>
<td>Restart DB2 on LPAR A or LPAR B in light mode.</td>
</tr>
</tbody>
</table>
| System response| 1  CICS reconnects through Subzero to the restarted DB2 group member to resolve any broken threads.  

**Note:** Reconnection occurs whether you restart DB2 on LPAR A or LPAR B.  

2  The DB2 group member shuts down in light mode.  

3  Depending on the cause of the failure, the DB2 group member either restarts normally or does not restart:  

   - If it restarts normally, CICS connects to it immediately.  

   - If it does not restart, CICS connects to the next DB2 group member, or connects as specified by the Subzero connection configuration. |
## DBC failure

This topic describes a DBC subsystem failure.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Environment</strong></td>
<td>Assume that a MQ shared queue on LPAR A is using Subzero to communicate with DB2 on LPAR B. Also IBM IMS on LPAR A is using Subzero to communicate with DB2 on LPAR B.</td>
</tr>
<tr>
<td><strong>Component failure</strong></td>
<td>The DBC subsystem fails on LPAR A.</td>
</tr>
</tbody>
</table>
| **User response** | 1. Restart DB2 on LPAR A or LPAR B in light mode.  
2. Issue `/STA SUBSYS xxx` to reconnect IMS to DB2.  
   Replace `xxx` with the name of an individual subsystem or with `ALL` to apply to all subsystems. |
| **System response** | 1. Restart any transactions that stopped due to DB2 unavailability.  
2. The MQ shared queue reconnects with DB2 on LPAR B.  
3. MQ message flow on LPAR A resumes. |
Subzero and associated components automatically recover from most failures. However, you might encounter a situation that requires troubleshooting.

The following topics can help you with troubleshooting efforts:

- “Resources for troubleshooting” on page 127
- “Potential problems” on page 130
- “Frequently asked questions about Subzero” on page 131

Resources for troubleshooting

If a Subzero problem occurs, you can use a variety of resources (in addition to BMC Customer Support) for obtaining information and diagnosing the problem.

Subzero messages

Subzero and associated components produce error, warning, and informational messages.

- The beginning of the message number tells you which component issued the message:
  - BMCBRD for Subzero messages
  - BMCDBC for DBC messages
  - BMCLGC (or BMC) for LGC messages

- Message locations vary according to which component issued the message:
Subzero issues messages to the BRDPRINT and BRDERROR data sets. During early initialization, Subzero uses the BRDPRINT data sets in controlling agent address spaces. After connecting to DBC, Subzero broadcasts messages to the BRDPRINT data set in the DBC address spaces on all LPARs in the processing group. Subzero allocates BRDPRINT dynamically.

DBC issues messages to the DBCPRINT data set in the DBC address space. The DBC external agent stub in Subzero address spaces (controlling agent and subsystem server) also issues messages to DBCPRINT in those address spaces.

For message explanations and recommended user responses, see the Messages Library in the BMC Documentation Center (https://webapps.bmc.com/infocenter/index.jsp).

**Note**

If an error occurs that Subzero could not resolve programmatically, Subzero can issue user abend code U3549.

---

**Online Help**

When working with a Subzero option set in the LGC ISPF interface, you can access online Help as follows:

- To display information about a field, place your cursor in the field and enter the HELP command (F1).

- To view product-specific information about the panel, use the Help pull-down (option 2) or enter the PRODHELP command on the command line.

From the system console of an LPAR that is participating in Subzero redirection, entering the following command lists all available Subzero commands:

```
/dbc BRD,HELP
```

**Documentation needed when reporting a problem**

If a problem occurs, provide the following resources to BMC Customer Support BMC for use in diagnosing the problem.

- Issue the Subzero DIAG command to print the internal trace information: `/dbcid BRD,DIAG`.  

128  

*BMC Subsystem Optimizer for zEnterprise Administration Guide*
- Provide the JES JOBLOG for the affected address spaces:
  — Client (for problems affecting CICS, IMS/TM, MQ, DB2, etc)
  — Subzero (for problems affecting Controlling Agent, Subsystem Server)
- SVC dump data set

  **Note**
  If multiple SVC dumps occur, the first dump from each address space is usually sufficient.

- Console dump

  If an address space appears to be not responding (hung):
  1. Take a console dump of the address space and include the Subzero Controlling Agent address space. For example:
     Enter `/DUMP COMM='SUBZERO HUNG CICS REGION'`.
  2. At the reply:
     Enter `/R mnnn,JOBNAME=(CICSRGN, SUBZRGN, DBCRGN), END`.

### Diagnostic tracing (DEBUG command)

To collect information about a recurring problem, BMC Customer Support might ask you to enable diagnostic tracing via DEBUG commands and send the trace output to BMC.

The trace output is located in the BRDTRACE and BRDSVLOG data sets in the address space in which you enter the DEBUG command.

Tracing is a resource-intensive activity. Use the following diagnostic tracing commands only at the direction of BMC Customer Support:

<table>
<thead>
<tr>
<th>This command</th>
<th>Enables or disables debug tracing for</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRD,DEBUG,{ON</td>
<td>OFF}, CLIENT</td>
</tr>
<tr>
<td>BRD,DEBUG,{ON</td>
<td>OFF}, DB2ID=ssid</td>
</tr>
<tr>
<td>BRD,DEBUG,{ON</td>
<td>OFF}, IMSID=imsid</td>
</tr>
<tr>
<td>BRD,DEBUG,{ON</td>
<td>OFF}, PIID=server_piid</td>
</tr>
</tbody>
</table>
This command: Enables or disables debug tracing for:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRD,DEBUG,{ON</td>
<td>OFF},ASIDX=server_asidx</td>
</tr>
<tr>
<td>BRD,DEBUG,{ON</td>
<td>OFF},AGENT</td>
</tr>
</tbody>
</table>

Potential problems

The following topics explain how to address various problems and situations that you might encounter when using Subzero.

Inability to connect to a remote subsystem

If your DBC address spaces and client address spaces are started but Subzero cannot connect to a remote subsystem, one of the following problems might be the cause:

- The DBC address space might not have started before the CICS region.
  
  In this case, "bounce" (stop and restart) the CICS region.

- The Subzero redirection options might be configured incorrectly.

  The client address spaces on this LPAR must be configured to redirect requests to remote LPARs. Review the redirection options to ensure that a client (source) LPAR matches a target (destination) LPAR. After correcting the options, enter the `/dbcssid BRD,RELOAD` command to activate the corrected options.

Inadvertent connection to a local subsystem instead of a remote one

If Subzero is not redirecting the requests for a client to the remote target subsystem that is specified in the redirection options, the client might be connecting to a local subsystem.

This problem can occur if the "from" (local, or client) subsystem ID matches the subsystem ID of a subsystem that is active on the client (local) LPAR. Subzero always attempts to connect to a subsystem that is active on the local LPAR before attempting to connect to a target subsystem on a remote LPAR.
Example

You have DB2A on SYSA, and you want to redirect the CICS system running on SYSA to connect to DB2B on SYSB. You must take one of the following actions:

- Stop DB2A on SYSA. You can specify DB2A within your redirection options, and continue to connect the CICS system to DB2A. Without a local DB2A subsystem, Subzero will redirect requests to the remote subsystem that is specified in your redirection criteria.

- As the client subsystem, specify a "fictitious" subsystem ID (one that is not in use at your site—such as ADB2 in this example).

For your redirection criteria, use "ADB2 on LPAR * will redirect to DB2B on LPAR SYSB." Then, connect your CICS region to ADB2 instead of DB2A. Assuming that no ADB2 DB2 subsystem ID is active on the local LPAR, Subzero will redirect requests from the CICS region to DB2B on SYSB. After modifying your redirection options, you must enter the `/dbcssid BRD,RELOAD command to activate the changed options.

Frequently asked questions about Subzero

The following topics answer common questions that you might have when using Subzero:

- “General questions” on page 131
- “Operational questions” on page 133
- “Performance questions” on page 136
- “Infrastructure questions” on page 137

General questions

What is the BRDSNAP data set in my CICS region?

Enabling the Subzero interface enables a Subzero ESTAE routine for each remotely connected thread. If an abend occurs, this ESTAE routine will take a SNAP dump of the task and stores the dump in BRDSNAP so that required diagnostic data is available for BMC Customer Support.
Is it easier to move the CICS region or the DB2 or IMS subsystem to a different LPAR?

When implementing Subzero, many organizations find that it is easier to move a CICS region to different LPAR, and leave a DB2 or IMS subsystem on its original LPAR. Site processes can affect the decision about which address spaces to move.

Does Subzero support redirection to a DB2 data-sharing group?

Yes. If you are using the DB2 group attach facility to define the connection between CICS and DB2, you must specify the DB2 group ID that is defined to CICS as the original (source) subsystem in a Subzero option set.

Note
Subzero does not support DB2 data-sharing subgroups.

Does Subzero affect the data in software monitors for IMS, DB2, and CICS?

Yes. Because Subzero could connect to remote subsystems, some data might be missing or contain different names.

BMC provides an API to externalize Subzero information for software monitors. You can ask your vendor to work with BMC to enable their monitor to use the Subzero API.

BMC MainView products already use the Subzero API.

Can I configure Subzero to redirect data requests from CICS/IMS TM to a DB2 or IMS instance running on an LPAR that is in the same sysplex, but on a different CPC (or CEC) than where CICS is running?

Yes, but be aware that this may increase CPU consumption. For best performance, run Subzero environments (for example, CICS to DB2) on LPARs that are on the same CPC.

Does Subzero support threadsafe transactions?

Subzero supports IBM CICS transactions that have been defined as threadsafe in both key 8 and key 9.
How do you handle DDF (direct to DB2) when moving a DB2 subsystem to a new LPAR

Subzero does not redirect access to DB2 through the distributed data facility (DDF). If you are using DDF, Subzero will redirect only the data requests from CICS. If you move CICS and leave DB2 on its current LPAR, no changes are needed for your DDF environment (because the DB2 subsystem has not moved). In contrast, if you move the DB2 subsystem, you must make changes so that the remote DDF requests can connect to DB2 on its new LPAR.

Operational questions

How does Subzero affect policies for my Workload Manager (WLM) services?

When a client address space makes a connection request to a remote subsystem, the Subzero controlling agent starts a Subzero subsystem server on the remote system. This server handles the thread activity from a single client address space (for example, a CICS region).

Subzero passes the classification data for the client address space to the remote subsystem server, which uses the data to reclassify the address space. Classification data for the client thread passes between the client and the Subzero subsystem server. This data serves as the classification data for WLM performance and delay monitoring.

Does Subzero require a sysplex environment?

Yes, Subzero works within a single basic or parallel sysplex environment. Subzero does not run across different sysplex environments.

How can I collect diagnostic information from IBM CICS or IMS TM, and from DB2 or IMS for troubleshooting?

Potentially, you will need information from two LPARs: the local LPAR where the client is running, and the remote LPAR where the DB2 or IMS subsystem is running.

All of the information that you normally would gather still exists; however, it is now in two systems instead of one. This situation is similar to diagnosing a problem in a data-sharing environment, in which you need to gather information from the individual LPARs of your data-sharing group.
The situation is also similar to diagnosing a problem if you have implemented IBM MQ by using Open Transaction Manager Access (OTMA) to split your IMS environment away from the MQ LPARs.

**How do my charge-back systems reconcile the breakout of CICS and remote subsystem activity?**

When a client address space makes a connection request to a remote subsystem, the Subzero controlling agent starts a Subzero subsystem server on the remote system. This server handles the thread activity from a single client address space (for example, a CICS region).

Subzero passes the job accounting data from the client address space to the remote subsystem server, which uses the data to reclassify the address space. Thread statistics that the remote subsystem generates are returned to the client address space in the expected format. The client address space can generate any statistics or accounting data, as it normally would, from the data that the remote subsystem supplies.

Subzero propagates the SMF accounting data from the client address space to the Subzero subsystem server address space that is dedicated to servicing the thread activity. However, the SMF data will be spread across multiple LPARs.

The Subzero controlling agent address space is a multiple-user address space. Multiple client address spaces use its XCF and cross-memory communication services. All Subzero users on the LPAR should share the system resource charges that are associated with this address space.

For complete information, see “System Management Facilities (SMF) considerations” on page 43.

**How do I know that Subzero has been enabled within a CICS region?**

You should see the BMCBRD0022I message for IMS, DB2, or both within the JESLOG for the CICS region.

**How do I know that a CICS region is connected to a remote subsystem?**

Subzero issues these messages in the JESLOG for the CICS region:

- BMCBRD0026I indicates that Subzero is managing the subsystem connection.
BMCBRD0150I indicates that the subsystem has been redirected to another subsystem on another LPAR.

**Why did my CICS region connect to a local subsystem instead of a remote subsystem?**

Remote subsystem redirection occurs only when the specified subsystem is not active on the local LPAR. When a CICS region attempts to connect to a subsystem, Subzero checks for the specified subsystem on the local LPAR first:

- If that subsystem is found on the local LPAR, Subzero passes control to the native interface routines and no longer participates in the job step. At this point, CICS connects to the local subsystem through the native interface.

- If the specified subsystem is *not* found on the local LPAR, Subzero checks for a configuration option that redirects the connection to a remote subsystem; if such an option is found, Subzero performs the redirection.

**Example**

CICS and DB2A are active on SYSA. DB2B is active on SYSB. Your Subzero configuration options specify that Subzero should redirect DB2A to DB2B. Subzero responds as follows:

- If DB2A is running on SYSA when CICS on SYSA attempts to connect to DB2A, Subzero passes control to the native interfaces, which establish a local DB2A connection.

- If DB2A is not running on SYSA when CICS on SYSA attempts to connect to DB2A, Subzero establishes the remote connection to DB2B, as specified in your configuration options.

**How many CICS regions can connect to a remote IMS or DB2 subsystem?**

Subzero can handle an unlimited number of threads for remote connections. Subzero dynamically increases and decreases the number of threads according to application requirements.

**If Subzero stops functioning, can connections to IMS and DB2 be reestablished without recycling CICS?**

Yes. You can reconnect to the remote subsystem or a local subsystem without stopping and restarting CICS.
What happens if a DB2 or IMS subsystem starts on the local LPAR after Subzero connects to that same subsystem on a remote LPAR?

Subzero continues to maintain the connection to the DB2 or IMS subsystem on the remote LPAR.

To stop using the remote connection and start using the local connection, you must tell CICS to stop and start its database connection. At that point, Subzero connects to the local version of the subsystem.

What happens if my application program experiences an IMS or DB2 abend?

Thread abends on the remote system are propagated back to the local thread. For example, if your IMS PSB Schedule request failed due to a U0456 abend (PSB Stopped), that abend is propagated back to the client thread.

What happens if the Subzero subsystem server fails?

If the Subzero subsystem server abends, the controlling agent communicates the failure to the registered client address space (CICS). It also notifies IMS thread manager task (BRDITHDM) and/or posts the DB2 connection termination ECB to allow normal thread and connection cleanup within the client address space. When the Subzero subsystem server abends and the controlling agent is still running, an attempt is made automatically to restart the subsystem server. After multiple restart attempts, the subsystem server will not automatically restart. In this case, you must manually restart it.

Performance questions

What transaction throughput can Subzero handle?

Subzero scales with your z/OS environment. Too many elements factor into the transaction rate to provide an absolute number, but BMC testing with more than 1000 transactions per second with 100 LUs indicates no significant decline in performance.

If your own testing indicates that throughput is an issue, you might address the issue by adding CICS instances. This approach reduces your CICS MLC savings
because the additional CICS instances adds to the peak consumption on that LPAR, but the improved throughput ensures that you can meet SLAs.

**What effect does Subzero have on a high-availability environment?**

In addition to helping to reduce MLC costs, Subzero can help you maintain a high-availability environment.

You can define Subzero routing to provide alternative connections between CICS and a DB2 or IMS database. Subzero can then route data requests to a running DB2 or IMS instance if the DB2 or IMS instance that is defined for the initial Subzero connection becomes unavailable.

**Infrastructure questions**

**Does DBC have to be started before CICS?**

Yes, DBC must be running before you start CICS. Without DBC, Subzero does not receive control to redirect requests for applications that connect to subsystems through CICS. If CICS is started and DBC is not running, you must restart CICS if you want to use Subzero.

**If DBC is recycled, does CICS have to be recycled, too?**

No. If DBC stops and restarts on the client (local) LPAR or the target (remote) LPAR, Subzero disconnects CICS from the target subsystems. When DBC restarts, Subzero automatically reestablishes the connections in CICS to the target subsystems without a CICS recycle.

**Does Subzero work with my existing DBC address spaces?**

Yes, if you are already using DBC (and other infrastructure components) with other BMC products, you should use the same components for Subzero. You do not need to use a separate set of components.

**What is the purpose of the controlling agent?**

The controlling agent consolidates and coordinates Subzero across the sysplex environment. The controlling agent starts when DBC starts and runs on all LPARs using Subzero. The controlling agent is responsible for keeping the maps up to date,
starting the Subzero common code, and defining/starting the Subzero subsystem servers.

The controlling agent initializes storage areas and cross-memory environments only on LPARs that can be used as a source for a data request. You define the source LPARs when you configure redirection options.

**What happens if the controlling agent fails?**

If the controlling agent abends and ESTAE routines cannot recover from the failure, each registered client is notified of the failure. For DB2, the DB2 connection termination ECD will be posted. For IMS, the IMS thread manager task (BRDITHDM) will be notified.

To restart the controlling agent, issue the DBC command dbcid BRD,START. There is no automatic restart.

After restarting the controlling agent, CICS must reconnect to IMS (CDBC transaction) or DB2 (DSNC transaction) to restart the Subzero subsystem server. When CICS attempts to reconnect to the remote IMS or DB2 subsystem, DBC will automatically restart the Subzero subsystem server. WTOs are produced, so you can automate the restart.

**Is the DB2 Component Services (DBC) component involved in every communication between the controlling agent and the subsystem server?**

No. Once the remote Subzero subsystem server is active and connected, data requests are passed directly from the client address space (CICS) to the remote subsystem server. The response is sent back to the client address space through XCF.

**Why is the Runtime Component System (RTCS) required?**

Subzero itself does not use any RTCS services. However, Subzero uses the DB2 Product Configuration (LGC) infrastructure, which requires RTCS.
Monitoring Subzero components and activities with MainView

BMC MainView products can provide information about Subzero address spaces, cross-system coupling facility (XCF) groups that Subzero address spaces are using, and the relationships between clients and targets for redirection.

The following MainView products can provide Subzero information:

- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online

**Note**
The MainView for z/OS product displays information about XCF members that Subzero uses. You can follow hyperlinks to this information from views in other MainView products that display information about Subzero components. No configuration steps are needed to use Subzero with MainView for z/OS.

For more information, view the following Quick Courses:

- "Subsystem Optimizer for zEnterprise - MainView Support for Subzero"
- "Subsystem Optimizer for zEnterprise - Impact on MainView products"

The following topics explain how to view Subzero information in these products. These topics assume that you are already familiar with how to use your MainView product.

- “Enabling MainView to work with Subzero” on page 140
- “Summary of Subzero information in MainView views and containers” on page 141
- “Subzero environment views” on page 142
Enabling MainView to work with Subzero

For MainView to work with Subzero, you must install required MainView maintenance, provide System Authorization Facility (SAF) access to a Subzero application programming interface (API), and enable collection of Subzero data.

To enable MainView to work with Subzero

1. Obtain and install the required PTFs for your MainView products.

   A list of required PTFs is located in the BMC Subsystem Optimizer for zEnterprise release notes.

2. Provide access to the STATAPI resource in your system authorization facility (such as RACF).

   MainView products require this resource in order to use the Subzero application programming interface (BRDAPI). For instructions, see “Securing Subzero components with SAF” on page 73.

3. Modify the BBIISPxx member of the BBIPARM library to specify the following Subzero parameters:

   - Specify BRDDCOLL=YES to collect Subzero component data.

     - **Note**
       To disable collection of Subzero data, specify BRDDCOLL=NO or omit this parameter.

     - (optional) To limit Subzero data collection to a single XCF group, specify BRDGROUP=xcfGroupName. xcfGroupName is the name of the Subzero XCF group. To collect data for all Subzero XCF groups, omit the BRDGROUP parameter.

   You must specify these parameters for at least one BBI-SS product address space (PAS) on each LPAR in which Subzero is running. Only one PAS collects data. However, if this PAS terminates and the parameters are specified for another PAS that is running on this LPAR, the second PAS will take over data collection.

4. Add the Subzero runtime library to the STEPLIB concatenation in the PAS JCL.
Verify that the following message is displayed, indicating that Subzero data collection is activated:

```
BBQBE104I Starting Subzero data collection
```

## Summary of Subzero information in MainView views and containers

MainView views and containers display Subzero information as applicable. The following table summarizes which MainView views and containers display Subzero information.

<table>
<thead>
<tr>
<th>Primary information</th>
<th>MainView view or container name</th>
<th>MainView product</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCF groups defined for use by Subzero components</td>
<td>Subzero Groups (RDGROUP)</td>
<td>All</td>
</tr>
<tr>
<td>Subzero address spaces currently executing in the environment</td>
<td>Subzero Servers (RDSRVR)</td>
<td>All</td>
</tr>
<tr>
<td>Relationships between Subzero clients and targets for redirection</td>
<td>Subzero Topology (RDTOPOL)</td>
<td>All</td>
</tr>
<tr>
<td>Subzero flow</td>
<td>EZExplorer\Configurations \Subzero Flow</td>
<td>All</td>
</tr>
<tr>
<td>Subzero topology</td>
<td>EZExplorer\Configurations \Subzero Topology</td>
<td>All</td>
</tr>
<tr>
<td>CICS regions (within the current context) accessing IMS through a DBCTL attachment</td>
<td>CREGDBC</td>
<td>MainView for CICS</td>
</tr>
<tr>
<td>Details about a specified CICS region accessing IMS through a DBCTL attachment</td>
<td>CREGDBCD</td>
<td>MainView for CICS</td>
</tr>
<tr>
<td>CICS regions (within the current context) accessing DB2 through connections between CICS and DB2</td>
<td>CDB2CON</td>
<td>MainView for CICS</td>
</tr>
<tr>
<td>Details about a specified CICS region accessing DB2 through a connection between CICS and DB2</td>
<td>CDB2COND</td>
<td>MainView for CICS</td>
</tr>
<tr>
<td>IMS subsystem that has CICS connections</td>
<td>IMSCONR</td>
<td>MainView for IMS Online and MainView for DBCTL</td>
</tr>
</tbody>
</table>
Subzero environment views

In all MainView products that support Subzero, you can access views that primarily provide information about the Subzero environment. The following views have the Subzero environment as their primary focus:

- “Subzero Groups (RDGROUP) view” on page 142
- “Subzero Servers (RDSRVR) view” on page 143
- “Subzero Redirect Targets (RDTOPOL) view” on page 144

Subzero Groups (RDGROUP) view

The Subzero Groups view, which you can access directly by using the RDGROUPE command, displays the XCF groups that Subzero components are using in the sysplex environment.

You can use this view to answer questions such as:

- Which Subzero groups (environments) are deployed in the sysplex?
- Which LPARs are involved in Subzero activities in the sysplex?
- Which DBC address spaces are managing Subzero address spaces?
- Which BBI-SS PASs are collecting data for the Subzero groups?
The following example of the Subzero Groups view shows a relatively large number of Subzero groups.

**Figure 14: Subzero Groups (RDGROUP) view**

---

**Subzero Servers (RDSRVR) view**

The Subzero Servers view, which you can access directly by using the RDSRVR command, lists the Subzero address spaces that are deployed in your environment.

You can use this view to answer questions such as:

- Which Subzero servers are deployed in the sysplex, and what is their status?
- Which LPAR is running a particular Subzero server?
- How many Subzero address spaces are running on a particular LPAR?
- Which target subsystem is associated with a particular Subzero address space?
The following example of the Subzero Servers view shows a relatively small number of Subzero address spaces.

**Figure 15: Subzero Servers (RDSRVR) view**

![Subzero Servers (RDSRVR) view](image)

**Subzero Redirect Targets (RDTOPOL) view**

The Subzero Redirect Targets view, which you can access directly by using the RDTOPOL command, lists information about the redirection participants that are active in the Subzero environment.

You can use this view to answer questions such as:

- Which Subzero subsystem server is working with a particular target subsystem?
- Where is Subzero redirecting subsystem access requests for a particular client (such as a CICS region)?
- Which target system is servicing the redirected subsystem access requests?
The following example of the Subzero Redirect Targets view shows a relatively small number of participants.

Figure 16: Subzero Redirect Targets (RDTOPOL) view

Subzero containers in MainView Explorer

MainView Explorer provides out-of-the-box solutions that illustrate the flow of work and relationships between Subzero components.
You can navigate to or invoke these Subzero containers from the EZExplorer/Configurations folder in the navigation tree of any MainView product that supports Subzero (Figure 17 on page 146).

**Figure 17: Subzero containers in MainView Explorer**

The following Subzero containers are available:

- “Subzero Flow container” on page 146
- “Subzero Topology container” on page 149

**Subzero Flow container**

The Subzero Flow container illustrates how a Subzero group is redirecting requests for subsystem access from clients (such as CICS regions) on local LPARs to targets (such as DB2 subsystems and IMS control regions) on different LPARs.
Figure 18 on page 147 shows the flow for a single Subzero group.

**Figure 18: Example Subzero Flow container**

The group is identified by its XCF group name, JANQDBC3. In this group, Subzero is redirecting requests:

- From a client CICS region named CICSJAN3 that is running on LPAR IMSA (using the Subzero controlling agent, J3SACTLA)
- To a target DB2 subsystem named DEHZ that is running on LPAR SYSM (through the Subzero subsystem server named J3SMDEHZ)
- To a target IMS control region named J121 that is running on LPAR SYSM (through the Subzero subsystem server named J3SMJ121)

**Note**

In a flow container, you can hover your cursor over the end nodes to display additional attributes of the target.
Figure 19 on page 148 shows three Subzero groups that are redirecting subsystem access requests from multiple CICS regions to various DB2 and IMS subsystems.

**Figure 19: Example Subzero Flow container for multiple Subzero groups**

The BRDMPL Subzero group is redirecting requests:

- From a CICS system named CICSJAN1 that is running on LPAR IMSA (through the controlling agent address space named RDAMLPCA)
- To a DB2 system named DEHZ that is running on LPAR SYSM (through a subsystem server named RDAMDEHZ)

The JANQDBC2 Subzero group is redirecting requests:

- From two CICS systems, named CICSJAN2 and CICSJAN4, that are running on LPAR SYSM (through the controlling agent address space named BRD2CNTL)
- To an IMS system named J122 that is running on LPAR IMSA (through a subsystem server named BRD2J122)

The JANQDBC3 Subzero group is redirecting requests:

- From a CICS system named CICSJAN3 that is running on LPAR IMSA (through the controlling agent address space named J3SACTLA)
- To an IMS system named J121 that is running on LPAR SYSM (through a subsystem server named J3SMJ121)
To a DB2 system named DEHZ that is running on LPAR SYSM (through a subsystem server named J3SMDEHZ)

**Subzero Topology container**

The Subzero Topology container illustrates the relationships between the participants in Subzero redirection.

**Figure 20 on page 150** shows the topology for a single Subzero group. The group is identified by its XCF group name, JANQDBC3. In this group, Subzero is redirecting requests:

- From a client CICS region named CICSJAN3 that is running on LPAR IMSA (using the Subzero controlling agent, J3SACTLA)
- To a target DB2 subsystem named DEHZ that is running on LPAR SYSM (through the Subzero subsystem server named J3SMDEHZ)
- To a target IMS control region named J121 that is running on LPAR SYSM (through the Subzero subsystem server named J3SMJ121)
**Note**

In a topology container, you can hover your cursor over the end nodes to display additional attributes of the target.

Figure 20: Subzero Topology container

---

**Subzero information in other MainView views**

The following table describes each field that displays Subzero information in the indicated MainView views if Subzero support is enabled in MainView:
<table>
<thead>
<tr>
<th>Fields for Subzero information</th>
<th>Description</th>
<th>Product Views</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subzero Server</td>
<td>Identifies the subsystem ID of the Subzero address space that is associated with this entity. You can hyperlink from this field to display the RDSRVR view that displays information about the participants in Subzero redirection.</td>
<td>CREGDBC CREGDBCD CDB2CON CDB2COND</td>
<td>MainView for CICS</td>
</tr>
<tr>
<td>Subzero Server System or Subzero System</td>
<td>Identifies the system ID of the LPAR where the Subzero address space that is associated with this entity is executing. You can hyperlink from this field to display the RDSRVR view that displays information about the participants in Subzero redirection.</td>
<td>CREGDBC CREGDBCD CDB2CON CDB2COND</td>
<td>MainView for CICS</td>
</tr>
<tr>
<td>ID</td>
<td>Identifies the subsystem ID of the Subzero subsystem server that is redirecting requests from a CICS region to an IMS subsystem.</td>
<td>IMSCONR</td>
<td>MainView for DBCTL MainView for IMS Online</td>
</tr>
<tr>
<td>Subsys Type</td>
<td>Identifies a Subzero subsystem server as type Subzero.</td>
<td>IMSCONR</td>
<td>MainView for DBCTL MainView for IMS Online</td>
</tr>
<tr>
<td>CICS SYS Name (Redirected by BMC)</td>
<td>Identifies the LPAR from which Subzero redirected the selected transaction instance.</td>
<td>ITASTRAC</td>
<td>MainView for DBCTL MainView for IMS Online</td>
</tr>
</tbody>
</table>
Glossary

4HRA

Four–hour rolling average, also known as rolling four-hour average, R4HA, and R4. An average of the CPU consumption of a given logical partition (LPAR). The 4HRA is measured in Millions of Service Units (MSUs) used over a given four-hour period.

A

ASIDX

Hexadecimal address space identifier.

B

BMC Cost Analyzer for zEnterprise

A BMC product that provides tools for analyzing, optimizing, and predicting software license cost for IBM MLC products.

BMC Intelligent Capping for zEnterprise (iCap)

A BMC product that dynamically optimizes defined capacity settings for LPARs defined to a central processor complex (CPC), also known as a central electronic complex (CEC).

BMC Subsystem Optimizer for zEnterprise

A family of BMC products that redirect subsystem access requests that applications issue on one LPAR to subsystems that are running on a different LPAR.

See also “Subzero” on page 159.
BRD

The three-character product code that BMC uses to identify Subzero components, such as load modules, programs, and address spaces.

BRDAPI

The Subzero application programming interface (API). Programs and products can use BRDAPI to access the subsystem connectivity and performance statistics that Subzero maintains.

BRDCNTL

The default name of the Subzero controlling agent.

BRDINIT

The name of the Subzero initialization script.

BRDPROC

The default name of the Subzero startup procedure.

C

CDS

Couple data set. A z/OS data set containing information about a sysplex, its systems, cross-system coupling facility (XCF) groups, and their members.

client

The original (local, source, or "from") subsystem access routine and LPAR to which applications are sending requests for subsystem access, and from which Subzero is redirecting requests to the target.

controlling agent

A Subzero address space that runs on each LPAR that is participating in Subzero redirection. Controlling agents communicate with DBC, LGC, and controlling agents on other LPARS. As needed, controlling agents interface with clients on this LPAR and start and stop Subzero subsystem servers.

cross-system coupling facility
DB2 Component Services

See “DBC” on page 155.

DB2 Product Configuration

See “LGC” on page 156.

DBC

DB2 Component Services. This BMC address space provides an infrastructure in which BMC products can initialize their own product services and obtain DBC-provided services. DBC usually runs as a persistent started task. For more information, see the BMC Infrastructure Components Administration Guide.

E

ECSA

Extended common service area, also known as extended common storage area. This area is a major element of IBM z/OS virtual storage above the 16-MB line.

F

four-hour rolling average

See “4HRA” on page 153.

G

GTF

Generalized trace facility. This IBM z/OS service program records significant system events such as I/O interrupts, SVC interrupts, program interrupts, and external interrupts.
LGC

DB2 Product Configuration. A DBC agent that manages configuration options for various BMC products. LGC provides an ISPF interface for specifying options, and stores and retrieves sets of options through a Runtime Component System (RTCS) registry. For more information, see the *BMC Infrastructure Components Administration Guide*.

local subsystem or local LPAR

*See “client” on page 154.*

LPAR

Logical partition. A virtualized image of a hardware computing system. The LPAR can include shared and dedicated resources that are assigned from a pool of resources.

MainView

A family of BMC products for monitoring, analyzing, managing, and optimizing mainframe environments.

master controlling agent

The Subzero component that obtains the current Subzero option set from LGC and shares configuration information with other controlling agents. DBC automatically designates and manages the master controlling agent, which is transparent to all administrators and users.

MLCs

Monthly License Charges. A method that IBM uses to determine charges for software products, based on aggregate usage of the operating system and other products, such as IBM z/OS, CICS, DB2, IMS, and WebSphere MQ.

MSUs
Millions of Service Units. A measure of CPU time consumption, calculated as number of CPU seconds used per hour, multiplied by the service units per seconds (SU/sec) coefficient.

The SU/sec coefficient depends on the CPC type and model and normally is the same for all LPARs on a CPC.

O

option set

A collection of configuration values that define how Subzero should redirect requests from clients to targets. LGC manages option sets for Subzero. You use the LGC ISPF interface to enter and change your values.

P

PIID

Product instance identifier. When creating a subsystem server, Subzero defines the PIID as a key that uniquely identifies the subsystem server to DBC.

PLT

Program list table. A CICS structure that specifies which programs to execute in the post-initialization phase of CICS startup.

R

R4

See “4HRA” on page 153.

redirection

The Subzero process that enables an application running on one LPAR to send requests to a subsystem (such as DB2 or IMS) running on a different LPAR. You specify options that identify the original (source) subsystem and LPAR and the target (remote) subsystem and LPAR to participate in redirection. When an application attempts to connect to the original subsystem (and that subsystem is not running on the same LPAR as the application), Subzero replaces that connection with a connection to the target subsystem. Redirection is transparent to the application and target subsystem.
remote subsystem or remote LPAR

See “target” on page 159.

rolling four-hour average

See “4HRA” on page 153.

RTCS

Runtime Component System. A BMC infrastructure component that LGC uses to manage option sets for Subzero.

Runtime Component System

See “RTCS” on page 158.

S

SAF

System Authorization Facility. A z/OS interface that programs can use to communicate with an external security manager, such as IBM RACF.

SMF

System Management Facilities. A component of the IBM z/OS system that collects and records system and job-related information.

STATAPI

The SAF resource that controls access to the Subzero application programming interface.

subsystem server

A Subzero component that receives redirected subsystem access requests from a client, communicates with the target subsystem, and returns request results to the client. The controlling agent starts and stops subsystem servers automatically as needed. One subsystem server is required to serve each defined redirection relationship between a client and a target.
Subzero

A generic term for all products in the BMC Subsystem Optimizer for zEnterprise product family.

System Management Facilities

See “SMF” on page 158.

T

target

The destination, remote, or "to" subsystem and LPAR to which Subzero redirects application requests. For example, the target might be an IBM DB2 subsystem, or an IBM IMS region.

W

WLM

Workload Manager. A component that controls access to system resources based on performance goals set by the system administrator.

X

XCF

Cross-system coupling facility. An IBM z/OS component that enables authorized applications on one LPAR to communicate with applications on the same LPAR or a different one.
Index

4HRA
  definition 153
64-bit Pause Services 34

A
abend code U3549 127
abend propagation 136
accounting records. See SMF
  address spaces
    requirements 34
    stopping 100
agents. See controlling agents or LGC
APF requirements 39
API for Subzero 73
applications
  no changes needed for Subzero 17
  supported types 35
architecture of Subzero 27
ASIDX
  definition 153
authorization passwords. See license keys
automated startup 108

B
backout scenarios 17
BBIISPxx member 140
BBQBE104I message 140
benefits of Subzero 17
BMC Cost Analyzer for zEnterprise product 53
BMC Intelligent Capping for zEnterprise
  definition 153
BMCBRD0022I message 134
BMCBRD0026I message 134
BMCBRD0150I message 134
BMCPSWD data set 108, 111
BRD
  definition 154
BRD1 transaction 86
BRDAP1 application programming interface 73, 140
BRDCNTL address space 31
BRDDCOLL parameter 140
BRDERRO data set 108, 127
BRDGROUP parameter 140
BRDINIT initialization script 109
BRDPLT00 PLT program 60
BRDPRINT data set 107, 108, 127
BRDPROC procedure
  APF requirements 39
  named in initialization script 109
  setup 108
  source of DFSPZPxx module 37
BRDSNAP data set 131
BRDssid address space 32
BSL POLICY ACTIVATE command 80

C
CDB2CON view 150
changes to the product 11
CICS
  configuring for Subzero 60
  connection to local and remote subsystems 36
  DBC must start first 30
  IMS connections 37
  no transaction definition changes needed for
    Subzero 17
  PLT changes 60
  remote connection 134
  requirements 36
  SMF records 43
  startup JCL 60
  Subzero enabled 134
  support 36
clearing the BRDPRINT data set 107
client interface tasks 31
clientLPAR field 56
clients
  definition 154
  identifying candidates 49
CLIST for LGC 54
commands
  BSL POLICY ACTIVATE 80
  CONFIG,RELOAD 106
  console 95
  DBC 114
  DEBUG 129
  DIAG 114
  DISPLAY 104
  DISPLAY,MAP 105
  HELP 96, 128
  KILL 100
  LICENSE,RELOAD 106
  overview 96
  PRTFLUSH 107
  REFRESH 99
  RELOAD 106
  RESETPRINT 114
  RESETTRACE 114
  RESTART 99
  START 99
  STOP 100, 114
  Subzero 96
  TRACEFLUSH 114
  TRACEOFF 114
  TRACEON 114
components
  controlling agents 31
  DBC 29
  diagnostic facilities 32
  LGC 30
  overview 27, 29
  recovery facilities 32
  RTCS 30
  subsystem servers 32
CONFIG,RELOAD command 106
configuration options 56
considerations
  operational procedures and policies 44
  overview 41
  performance 41
  SMF 43
  subsystem access 42
console commands 95
containers
  Subzero Flow 146
  Subzero Topology 149
controlling agents
  definition 154
  displaying statistics 104
  job name 109
  overview 31
  refreshing with new startup values 99
  restarting 99
  starting 99
  stopping 100
conventions, documentation 10
Cost Analyzer product 53
CPU authorization passwords. See license keys
Create New Option Set pop-up panel 54
CREGDBC view 150
CREGDBCD view 150
cross-system coupling facility. See XCF

D

Database Resource Adapter 37
DB2
  access considerations 42
  data sharing group 132
  example of separation from CICS 19
  group IDs 37, 56
  requirements 37
  SMF records 43
  support 37
DB2 Component Services. See DBC
DB2 Product Configuration. See LGC
DB2 Product Configuration Main Menu 54
DB2 SSIDs/Groups to Redirect section 56
DBC
  APF requirements 39
  definition 155
  overview 29
  securing 73
  started task 111
  starting during installation 47
  startup procedure 112
  stopping 113
DBC commands
DIAG 114
DISPAGENTS 114
DISPDNS 114
RESETPRINT 114
RESETTRACE 114
STOP 114
TRACEFLUSH 114
TRACEOFF 114
TRACEON 114
DBC$STC sample 111
DBCPARMS data set 111
DBCPRT PRINT data set 108, 111
DBCSECUR data set 111
DEBUG command 129
deployment examples 18
DFSPZPxx module 37
DIAG command 114
diagnostic data sets 128
DISPLAY command 104
DISPLAY,MAP command 105
DL/I calls, supported 37
documentation information 9
dump data set 128

E

EC SA
definition 155
environment registration 84
error messages 127
examples
consolidating subsystems in a test environment 22
DB2 separated from CICS 19
DB2 separated from IMS 24
deployment 18
IMS consolidated on a single LPAR 20
reduced IMS footprint 20
volume-based separation of activities 21
Execute Extensions Facility 34
Execute Immediate Facility 34

F

fallback scenarios 17
FAQs 131
flushing the BRDPRINT data set 107
FMIDs 47
frequently asked questions 131

G

General Options section 56
generic LPAR specifications 49
GTF
definition 155

H

hardware requirements 33
HELP command 96
Help, online 128

I

IFI calls, supported 37
IMS
access considerations 42
CICS connections 37
Database Resource Adapter 37
database adapter 37
example of consolidation on a single LPAR 20
development example 20
example of reducing the footprint 20
requirements 37
restart on different IMS system 83
SMF records 43
support 37
IMS environment registration record 84
IMS program registration record 83
IMS Subsystems to Redirect section 56
IMSCONR view 150
Index Term 78, 79, 81, 82, 84, 85, 124, 131, 133, 136, 137
initialization script 109
installation
IVP 86
installing Subzero components 47
ISPF interface 54
ITASTRAC view 150

J

job policies
activating 80
**K**

KILL command 100

**L**

LGC
- definition 155, 156
- Help in ISPF interface 128
- ISPF interface 54
- overview 30
LGCISPF CLIST 54
license keys
  - in DBC started task procedure 111
  - installing 47
  - reloading 106
LICENSE,RELOAD command 106
Linkage Index Reuse Facility 34
local subsystems
  - connections 36, 135
  - definition 156
  - inadvertent connection 130
  - precedence over remote subsystems 17
locks, database 41
Long Displacement Facility 34
LPAR
  - definition 156
LX Reuse Facility 34

**M**

MainView
  - accounting records 43
  - definition 156
  - enabling for Subzero 140
  - requirements 39
Subzero information in views and containers 141
  - support 39
  - using with Subzero 139
master controlling agent
  - definition 156
messages, overview of Subzero 127
Millions of Service Units. See MSUs
MLCs
  - definition 156
monitoring software 132
MQ redirection using Subzero example 66
MSUs (Millions of Service Units)
  - definition 156

**N**

non IMS environment registration record 84
non IMS program registration record 83
Notes field 56

**O**

online Help 128
operating Subzero 95
operating system requirements 34
option sets
  - creating 54
  - definition 157
  - displaying future options in effect 105
  - displaying options in effect 105
  - modifying 54
originalSSID field 56

**P**

panels
  - Create New Option Set 54
  - DB2 Product Configuration Main Menu 54
  - option set 54
  - Product Option Sets 54
passwords. See license keys
performance
  - considerations 41
  - monitoring products 41
PIID
  - definition 157
PLT
  - definition 157
PLT changes for Subzero 60
PLTPI parameter 60
problems with Subzero 127
  - local connections 130
  - remote connections 130
processing group 35
Product Authorization utility 47
product changes 11
product license keys. See license keys
Product Option Sets panel 54
program list table. See PLT
program registration 83
PRTFLUSH command 107
publications, related 9

R
R4HA. See 4HRA
RDGROUP view 142
RDSRVR view 143
RDTOPOP view 144
recovery scenarios 17
Redirect Targets view 144
redirection
   defining 48
   definition 157
   guidelines for candidates 49
   ISPF interface for defining 54
   options 56
   workflow 28
redirection conflicts 105
REFRESH command 99
registration
   environment 84
   program 83
related publications 9
RELOAD command 106
remote subsystems
   connection problems 130
   connections 36
   definition 158
   inadvertent connection to local 135
   Subzero messages if connected 134
reporting a problem 128
requirements
   address space 34
   APF 39
   CICS 36
   DB2 37
   hardware 33
   IMS 37
   MainView 39
   operating system 34
   security 38
   subsystems 35
   XCF 35
RESETPRINT command 114
RESETTRACE command 114
restart
   on different IMS system 83
RESTART command 99
RTCS 30
   definition 158
Runtime Component System. See RTCS

S
SAF
   definition 158
   securing Subzero components 73
scenarios
   deployment examples 18
   See also examples
SDSF 95
security
   DBCSECUR data set 111
   STATAPI resource 73
security requirements 38
SMF
   CICS 43
   considerations 43
   DB2 43
   definition 158, 159
   IMS 43
   reconciling redirected requests 134
SMP/E 47
SQL calls, supported 37
START command 99
starting DBC during installation 47
startup automation 108
startup procedure
   DBC 111, 112
   Subzero 108
STATAPI resource 73, 140
   definition 158
statistics, displaying 104
STOP command 100, 114
stopping DBC 113
subsystem access considerations 42
subsystem servers
   definition 158
   displaying statistics 104
   overview 32
   stopping 100
Subzero
architecture 27
automating startup 108
benefits 17
client interface tasks 31
components 27
considerations 41
console commands 96
controlling 95
controlling agents 31
definition 159
deployment examples 18
enabling for MainView 140
FAQs 131
installation 47
operating 95
overview 13
performance 41
products 13
recovery and diagnostic facilities 32
redirection definitions 48
redirection example 14
redirection options 56
redirection workflow 28
requirements 33
securing components 73
setting up 47
subsystem servers 32
troubleshooting 127
tuning 41
using with MainView 139
Subzero Flow container 146
Subzero Groups view 142
Subzero Servers view 143
Subzero Topology container 149
summary of changes 11
SVC dump data set 128
syntax statement conventions 10
system authorization facility. See SAF
System Display and Search Facility. See SDSF
System Management Facilities. See SMF

T
targetLPAR field 56
targets
definition 159
identifying candidates 49
targetSSID field 56
terminating address spaces 100
testing
consolidating subsystems 22
guidelines 49
topology view 149
trace facility 129
TRACEFLUSH command 115
TRACEOFF command 114
TRACEON command 114
transaction queues 41
transaction throughput 136
troubleshooting Subzero problems 127
tuning 41
overview 45

U
U3549 abend code 127

V
views
CDB2COND 150
CREGDBC 150
CREGDBCD 150
IMSCONR 150
ITASTRAC 150
RDGROUP 142
RDSRVR 143
RDTOPOL 144
Redirect Targets 144
Subzero Groups 142
Subzero Servers 142

W
waiting tasks, terminating 100
wildcard character 49
WLM 133
WLM (Workload Manager)
definition 159
workflow, redirection 28
Workload Manager. See WLM

X
XCF
definition 159
  group name 35
  performance 41
  requirements 35
XCFGROUP option 56

Z
  z/OS
    requirements 34