BMC Subsystem Optimizer for 

zEnterprise Administration Guide

Supporting

Version 1.0 of BMC Subsystem Optimizer for DB2
Version 1.0 of BMC Subsystem Optimizer for IMS

December 2014
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>2101 CITYWEST BLVD</td>
<td>or</td>
<td>1 800 841 2031</td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td></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- BMC Software, Inc.

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

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

CICS, DB2, IBM, IMS, 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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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, 2101 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**
- Product documentation ................................................................. 9
- Conventions .................................................................................. 9
- Syntax statements ........................................................................ 10

### Chapter 1 Overview of BMC Subsystem Optimizer for zEnterprise 13
- Why use Subzero? ......................................................................... 13
  - What Subzero does .................................................................... 14
  - Benefits of Subzero over other solutions .................................. 16
  - Example deployment scenarios for Subzero .................... 17
- Subzero architecture ..................................................................... 22
  - Summary of the redirection workflow ........................................ 24
  - Overview of components and technologies ............................... 25
- Subzero requirements ................................................................... 29
  - Hardware requirements ............................................................ 29
  - Operating system requirements ............................................... 30
  - Subsystem requirements .......................................................... 31
  - Subzero security requirements ................................................. 34
  - MainView requirements ............................................................ 35

### Chapter 2 Subzero considerations and environmental tuning 37
- Subzero considerations ................................................................ 37
  - Performance considerations ..................................................... 37
  - Subsystem access considerations .............................................. 38
  - System Management Facilities (SMF) considerations ............... 39
  - Considerations for operational procedures and policies ............ 40
- Tuning your environment for Subzero .......................................... 41
  - Preparing your z/OS environment for Subzero ....................... 41
  - Tuning XCF components for Subzero ...................................... 42

### Chapter 3 Setting up Subzero and associated components 45
- Installing and starting components .............................................. 45
- Defining Subzero redirections ....................................................... 46
  - Identifying candidates for redirection ....................................... 47
- Using BMC Cost Analyzer for zEnterprise with Subzero ............ 51
- Collecting information about redirection candidates .................. 53
How do my charge-back systems reconcile the breakout of CICS and remote subsystem activity? .................................................................88
How does Subzero affect policies for my Workload Manager (WLM) services? .............................................................................................89
Why did my CICS region connect to a local subsystem instead of a remote subsystem? ..................................................................................89
Does DBC have to be started before CICS? ............................................90
If DBC is recycled, does CICS have to be recycled, too? ........................90
How many CICS regions can connect to a remote IMS or DB2 subsystem? .................................................................................................90
If Subzero stops functioning, can connections to IMS and DB2 be reestablished without recycling CICS? ........................................................91
Does Subzero affect the data in software monitors for IMS, DB2, and CICS? .................................................................................................91
What transaction throughput can Subzero handle? ................................91
What effect does Subzero have on a high-availability environment? ......91
Does Subzero work with my existing DBC address spaces? ................92
Does Subzero require assignment of an XCF transport class? ..........92
Does Subzero require a sysplex environment? ....................................92
What should I do to gather diagnostic information from CICS and DB2 or IMS for troubleshooting? .............................................................92
Is it easier to move the CICS region or the DB2 or IMS subsystem to a different LPAR? .................................................................92
What happens if a DB2 or IMS subsystem starts on the local LPAR after Subzero connects to that same subsystem on a remote LPAR? ......93
When does the Subzero controlling agent start, and where does it run? ... 93

Appendix A        Monitoring Subzero components and activities with MainView 95
Enabling MainView to work with Subzero ..............................................96
Summary of Subzero information in MainView views and containers ......97
Subzero environment views .................................................................98
    Subzero Groups (RDGROUP) view ....................................................98
    Subzero Servers (RDSRVR) view ......................................................99
    Subzero Redirect Targets (RDTOPOL) view ......................................100
Subzero containers in MainView Explorer ............................................101
    Subzero Flow container .................................................................102
    Subzero Topology container ..........................................................105
Subzero information in other MainView views ....................................106

Glossary 109

Contents 7
About this book

This book contains detailed information about using BMC Subsystem Optimizer for zEnterprise (Subzero) products. This book is intended for system administrators, database administrators, and others who are responsible for implementing and using the products in IBM z/OS, DB2, IMS, and CICS systems.

Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the support website at http://www.bmc.com/support.

**Note**

Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at http://www.adobe.com.

The software also offers online Help. To access Help, press F1 within any product or click the **Help** button in graphical user interfaces (GUIs).

Product documentation

From the Support Central website (http://www.bmc.com/support), you can:

- Link to the BMC Documentation Center (https://webapps.bmc.com/infocenter/index.jsp) to browse documentation sets

- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are included in the Documentation Center

- View individual product documents (books and notices) within the “A – Z Supported Product List”

You can order hardcopy documentation from your BMC sales representative or from the support site. You can also subscribe to proactive alerts to receive e-mail messages when notices are issued.
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: `testsys/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>
<tbody>
<tr>
<td>Items in italic type represent variables that you must replace with a name</td>
<td><code>alias</code></td>
</tr>
<tr>
<td>or value. If a variable is represented by two or more words, initial</td>
<td><code>databaseDirectory</code></td>
</tr>
<tr>
<td>capitals distinguish the second and subsequent words.</td>
<td><code>serverHostName</code></td>
</tr>
<tr>
<td>Brackets indicate optional items. Do not type the brackets when you enter</td>
<td><code>[tableName, columnName, field]</code></td>
</tr>
<tr>
<td>the option. A comma means that you can choose one or more of the listed</td>
<td><code>[-full, -incremental, -level]</code></td>
</tr>
<tr>
<td>options. You must use a comma to separate the options if you choose more</td>
<td></td>
</tr>
<tr>
<td>than one option.</td>
<td></td>
</tr>
<tr>
<td>Braces indicate that at least one of the enclosed items is required. Do</td>
<td>`{DBDName</td>
</tr>
<tr>
<td>not type the braces when you enter the item.</td>
<td>`UNLOAD device=(disk</td>
</tr>
<tr>
<td></td>
<td>`{-a</td>
</tr>
<tr>
<td>Convention</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>A vertical bar means that you can choose only one</td>
<td>{commit</td>
</tr>
<tr>
<td>of the listed items. In the example, you would</td>
<td></td>
</tr>
<tr>
<td>choose either commit or cancel.</td>
<td></td>
</tr>
<tr>
<td>An ellipsis indicates that you can repeat the</td>
<td>columnName…</td>
</tr>
<tr>
<td>previous item or items as many times as necessary.</td>
<td></td>
</tr>
</tbody>
</table>
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 applications issue 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. Subzero products include:

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

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 13
- “Subzero architecture” on page 22
- “Subzero requirements” on page 29

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 14 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

Figure 1: Example environment without Subzero
In contrast, Subzero can optimize subsystem deployment by redirecting requests from LPAR 1 to LPAR 2 (Figure 2 on page 15):

- 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. CICS, IMS, and DB2 subsystems can all 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, 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.
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 18
- “Consolidating IMS subsystems on a single LPAR” on page 19
- “Reducing an IMS footprint” on page 19
- “Separating high-volume activity from low-volume activity” on page 20
- “Consolidating subsystems in a test environment” on page 21
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 21)

- 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**

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

**Figure 8 on page 23** 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.

**Figure 8: Subzero components and technologies**

The following topics explain how Subzero works:

- “Summary of the redirection workflow” on page 24 summarizes the process for completing a redirection, specifying which component or technology is involved in each step.

- “Overview of components and technologies” on page 25 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...
Overview of components and technologies

Subzero uses the following components and technologies:

- “DB2 Component Services (DBC)” on page 25
- “DB2 Product Configuration (LGC) and other configuration components” on page 26
- “Subzero controlling agent” on page 27
- “Subzero subsystem servers” on page 28
- “Recovery and diagnostic facilities” on page 28

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.

**Note**
DBC originally worked only with BMC products for the IBM DB2 environment but now works with other environments, too.

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.

**Note**
LGC originally worked only with BMC products for the IBM DB2 environment but now works with other environments, too.
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 ID 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.

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 ID 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 83.
### 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 (or later) Business Class 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 XCF components for Subzero” on page 42.

### 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 “Preparing your z/OS environment for Subzero” on page 41.

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

*Note*
You might need to tune your XCF components to accommodate Subzero requirements. For more information, see “Tuning XCF components for Subzero” on page 42.

**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

An application can use multiple types of subsystem access routines.

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

- “CICS requirements” on page 32
- “DB2 requirements” on page 32
- “IMS requirements” on page 33
**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).

*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 58.

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**

Subzero can work with the IBM DB2 product acting as a database manager. To use Subzero with 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

Subzero can work with the IBM IMS product acting as a database manager. To use Subzero with IMS, you must have a license for the BMC Subsystem Optimizer for IMS 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 Product Availability and Compatibility Utility on the BMC Support Central website (http://www.bmc.com/support).

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.

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 34
- “SAF requirements” on page 34

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 60.
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 95. 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 37
- “Tuning your environment for Subzero” on page 41

Subzero considerations

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

- “Performance considerations” on page 37
- “Subsystem access considerations” on page 38
- “System Management Facilities (SMF) considerations” on page 39
- “Considerations for operational procedures and policies” on page 40

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.

- **Cross-system coupling facility (XCF) communication**
  The increased volume of XCF traffic might require changes to your XCF environment. For more information, see “Tuning XCF components for Subzero” on page 42.

- **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 41.

### 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.

---

**BMC Subsystem Optimizer for zEnterprise Administration Guide**
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

If you are concerned about environmental or resource issues affecting your use of Subzero, you can tune your cross-system coupling facility (XCF) and other elements in the environment.

Note
Typically, 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.

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.

Shifting subsystem workloads from one LPAR to another might also shift operating system resources that process client application programs. For example, 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.

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

Because Subzero redirections use cross-system coupling facility (XCF) resources, XCF components directly influence Subzero performance. You might need to adjust your XCF configuration to ensure that adequate XCF resources are available.

For more information, view the Quick Course "Subsystem Optimizer for zEnterprise - XCF configuration considerations."

The following XCF components can affect the overall performance of Subzero:

<table>
<thead>
<tr>
<th>XCF component</th>
<th>Effect on Subzero performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sysplex Failure Management (SFM) MEMSTALLTIME parameter</td>
<td>SFM can recognize and address XCF communication failures, based on user-specified conditions. The MEMSTALLTIME parameter limits how long XCF will wait before issuing message IXC431I, at which point SFM takes action to resolve the problem. (For example, SFM might cancel the member that is using the most XCF resources.) The default setting for MEMSTALLTIME is NO, directing SFM to take no action for stalled members. <em>This value might be inappropriate for Subzero.</em></td>
</tr>
<tr>
<td>Couple data set (CDS)</td>
<td>Subzero does not implement any services that require access to the Couple Data Set.</td>
</tr>
<tr>
<td>Coupling facility communication paths and XCF message buffers (as defined with the CLASSDEF statement)</td>
<td>Subzero maintains threads to its remote servers. An insufficient number of communication paths can cause message delays and stall coupling facility processing. An insufficient number of XCF message buffers, or improperly sized buffers, can cause an excessive amount of message blocking and deblocking for Subzero. Result can include higher overhead and longer delays.</td>
</tr>
<tr>
<td>Integrated Coupling Facility (ICF) processors</td>
<td>If ICF processors are running close to capacity, the additional volume of XCF messaging from Subzero could overload them. However, because organizations rarely use the full capacity of their existing ICF processors, Subzero is unlikely to cause a problem.</td>
</tr>
</tbody>
</table>

To tune XCF components for Subzero

1. Evaluate the MEMSTALLTIME parameter, and adjust the value as needed to ensure that XCF handles stalled members appropriately.

   **Tip**
   For more information, see the IBM documentation for setting up a z/OS sysplex.

2. Ensure that Subzero is using the appropriate communication path and XCF buffer size:
a Identify the XCF communication path (CLASSDEF) that Subzero will use.

Subzero uses an assigned CLASSDEF (if available) or dynamically chooses the undesignated CLASSDEF that has the maximum throughput capacity. Subzero finds the CLASSDEF with the largest number of outbound paths, and multiplies the number of paths by the buffer size that is associated with that CLASSDEF.

**Example**

Assume that two CLASSDEFS have the following configuration:

<table>
<thead>
<tr>
<th>CLASSDEF name</th>
<th>Buffer size</th>
<th>PATHIN</th>
<th>PATHOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALLBUF</td>
<td>2048</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>LARGEBUF</td>
<td>12288</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

If using the SMALLBUF CLASSDEF, Subzero can send 819,200 (2048 multiplied by 400) bytes asynchronously. If using the LARGEBUF CLASSDEF, Subzero can send 614,400 (12288 multiplied by 50) bytes asynchronously. Subzero chooses the SMALLBUF CLASSDEF buffer size for its XCF messaging.

b Evaluate the assigned or dynamically chosen CLASSDEF, and make adjustments if needed.

- The CLASSDEF must be able to accommodate each potential thread that Subzero might use to connect to a target subsystem. You can assign Subzero to a different CLASSDEF, or increase the capacity of the assigned or chosen CLASSDEF.

- Subzero automatically resizes its buffers to match the buffer size of the assigned or dynamically chosen CLASSDEF. You can override this dynamic buffer-size algorithm if needed by specifying the Subzero SET MSGBUFFERSIZE command.

3 Evaluate ICF processor and coupling facility link usage.

Ensure that the available capacity is adequate for the increased workload of every subsystem request from all client address spaces.
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 45
- “Defining Subzero redirections” on page 46
- “Configuring CICS for Subzero” on page 58
- “Securing Subzero components with SAF” on page 60

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, and decompresses them on your z/OS system
- 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 release notes
- BMC Subsystem Optimizer for zEnterprise release notes

At the end of the process, DBC and its agents should be running in your environment. You can now set up Subzero for redirection.

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."

### 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 47
- “Using BMC Cost Analyzer for zEnterprise with Subzero” on page 51
- “Collecting information about redirection candidates” on page 53
- “Setting Subzero configuration options in LGC” on page 54
- “Overview of Subzero configuration options” 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.
Example

The option set contains the following definitions:

```
- IMS Subsystems To Redirect
  + IMS1    on LPAR  *_______  will redirect to  IMID  on LPAR
  + IMS1    on LPAR  SY*_____  will redirect to  IMIB  on LPAR
  + IMS1    on LPAR  SYS*____  will redirect to  IMIS  on LPAR
  + IMS1    on LPAR  SYSA____  will redirect to  IMSA  on LPAR
  + IMS1    on LPAR  SYSB____  will redirect to  IMIB  on LPAR
```

Redirection occurs as follows:

- IMS1 on LPAR SYSA redirects to IMSA.
- IMS1 on LPAR SYSC redirects to IMIS.
- IMS1 on LPAR SYT1 redirects to IMIB.
- IMS1 on LPAR SYSB redirects to IMIB.
- IMS1 on LPAR BOB redirects to IMID

You can record information about your redirection candidates in the table in “Collecting information about redirection candidates” on page 53. This information will be required when you configure Subzero.

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 30). 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 41. |                           |
### What to identify

<table>
<thead>
<tr>
<th>At least one representative subsystem (such as a DB2 subsystem or IMS control region)</th>
<th>Guidelines</th>
<th>Your potential candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ For early testing, you can choose small or simplified representative subsystems.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ To ensure complete coverage in later testing, you might choose larger and more complex representatives.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ For production, you might include all subsystems in the list of potential candidates, and then rule out those that should not participate.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>At least one representative set of application programs that access the listed subsystem or subsystems</th>
<th>Guidelines</th>
<th>Your potential candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ For early testing, you can choose small-volume or simplified applications.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ To ensure complete coverage in later testing, you might choose larger-volume and more complex applications.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ For production, you might include all applications in the list of potential candidates, and then rule out those that should not participate.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. From your list of potential candidates, choose the subsystems and LPARs that you want to use as clients and targets:
What you need to choose | Guidelines | Your choice
---|---|---
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. |  |
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**

In the Cost Analyzer Monthly Summary report, find all of the LPARs that are running IBM DB2 or IMS subsystems. Based on the MSU consumption of those subsystems, determine whether consolidating some of the subsystems through redirection could reduce overall consumption.
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.

Note
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.
Collecting information about redirection candidates

To enter Subzero configuration values efficiently, you can collect information about the candidates that you have identified to participate in redirection.

You must define a processing group for a set of candidates. You can define multiple processing groups (containing different candidates) for different purposes, such as test and production, geographic regions, and type of subsystem. Each processing group is known by its assigned XCF group name.

Table 1 on page 53 lists the configuration values that you will need to specify for each candidate. You can use this table to collect information for defining the processing group. For more information about the values, see “Setting Subzero configuration options in LGC” on page 54.

Table 1: Configuration for processing group ____________________

<table>
<thead>
<tr>
<th>Original subsystem</th>
<th>Client LPAR</th>
<th>Target subsystem</th>
<th>Target LPAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: DB2A</td>
<td>Example: SYSA</td>
<td>Example: DB2B</td>
<td>Example: SYSB</td>
</tr>
</tbody>
</table>

The values have the following meanings:

Original subsystem

The subsystem ID (such as DB2A or IMS1) to which applications send subsystem access requests

Client LPAR

The system ID of the LPAR that is running the application programs that are issuing subsystem access requests
Target subsystem

The subsystem ID of the "real" subsystem that will receive and process redirected subsystem access requests

Target LPAR

The system ID of the LPAR that is running the real subsystem

Setting Subzero configuration options in LGC

Use the following procedure to set Subzero configuration options in the DB2 Product Configuration (LGC) ISPF interface. These options specify how Subzero should redirect subsystem access requests from a client to a target.

LGC creates XML-based option sets to contain your Subzero configuration values. You can create multiple option sets and activate a particular set by issuing a command.

For a description of each option that you can set, see “Overview of Subzero configuration options” on page 56.

You can change Subzero configuration values at almost any time. DBC must be running on the LPAR that you designate as a new client or target LPAR. If CICS started before DBC started 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 Installation System copied the customized LGCISPF CLIST to a data set as follows:

- If you chose the installation option to concatenate user libraries ahead of runtime libraries, LGCISPF is in userLibHLQ.UXXSAMP. (The userLibHLQ variable is specified during installation.)

- If you chose the installation option to consolidate user libraries with runtime libraries, LGCISPF is in HLQ.BMCSAMP. (The HLQ variable is specified during installation.)
As an alternative to using LGCISPF, you can access the configuration panels from within the Installation System.

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 option set for this product, enter **I**.

- 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** would let you simply browse (view) the values in the option set.

4 On the option set panel, you can work with options as follows:

- In the option fields, type the values that you want to use. For information about Subzero options, see “Overview of Subzero configuration options” on page 56.

- To insert a new option that contains default values, enter **I** next to an option name. (This action is not valid for **General Options**.)

- To create a new option that contains values copied from this option, enter **R** next to an option name. (This action is not valid for **General Options**.) Type over the displayed values to change them.

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

6 If this option set is new, specify its name and description on the Create New Option Set pop-up, 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`

In this command, `dbc` is the subsystem ID of the DBC address space.

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

## Overview of Subzero configuration options

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>Specify the name of the XCF group that Subzero will use to process redirected requests. Contact your 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>
### DB2 SSIDs/Groups to Redirect

<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. |

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.
## 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.

   Note
   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.
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:

```plaintext
//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 RACLST(FACILITY) REFRESH  
RLIST FACILITY *.*.BRD.STATAPI.PF AUTHUSER
```
2 Execute the job, and view the output to determine that the required access is permitted.
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 66
- “Terminating Subzero address spaces and tasks” on page 67
- “Displaying statistics about Subzero components” on page 69
- “Displaying information about Subzero options” on page 70
- “Reloading Subzero configuration options and license keys” on page 71
- “Clearing the BRDPRINT data set” on page 72

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 64 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.

---

### Table 2: Subzero console commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRD,CONFIG,RELOAD</td>
<td>Reread the Subzero option set (containing configuration information) from the repository, and make the new configuration active.</td>
</tr>
<tr>
<td>BRD,DEBUG ...</td>
<td>Enable or disable debug tracing for Subzero. Use this command only at the direction of BMC Customer Support. For more information, see “Diagnostic tracing (DEBUG command)” on page 85.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>BRD,DISPLAY,MAP</td>
<td>Display a map showing how Subzero is configured to redirect subsystem access requests from client to target subsystems.</td>
</tr>
<tr>
<td>BRD,DISPLAY,MAP,FUTURE</td>
<td>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.</td>
</tr>
<tr>
<td>BRD,DISPLAY,CLIENT,STORAGE</td>
<td>Display storage statistics about the Subzero task that handles clients on this LPAR.</td>
</tr>
<tr>
<td>BRD,DISPLAY,DB2ID= subsystemID,STORAGE</td>
<td>Display storage statistics for Subzero address spaces that are associated with the specified DB2 subsystem ID or group ID on this LPAR.</td>
</tr>
<tr>
<td>BRD,DISPLAY,IMSID=IMSID,STORAGE</td>
<td>Display storage statistics for Subzero address spaces that are associated with the specified IMS region on this LPAR.</td>
</tr>
<tr>
<td>BRD,DISPLAY,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,DISPLAY,ASIDX= serverASIDX,STORAGE</td>
<td>Display storage statistics for the Subzero address space that has the specified hexadecimal address space ID (ASIDX) on this LPAR.</td>
</tr>
<tr>
<td>BRD,HELP</td>
<td>Display available operator command syntax for Subzero.</td>
</tr>
<tr>
<td>BRD,KILL,PIID= serverPIID,TASK=taskID</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,KILL,ASIDX= serverASIDX,TASK=taskID</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,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,START,CLIENT</td>
<td>Request initialization of the Subzero controlling agent task that handles clients on this LPAR. The task will initialize only if the current configuration indicates that this LPAR is a potential client LPAR for redirection.</td>
</tr>
<tr>
<td>BRD,STOP</td>
<td>Stop all Subzero address spaces on this LPAR.</td>
</tr>
<tr>
<td>BRD,STOP,CLIENT</td>
<td>Stop the Subzero controlling agent task that handles clients on this LPAR.</td>
</tr>
</tbody>
</table>
Command | Description
---|---
BRD,STOP,DB2ID=subsystemID | Stop all Subzero address spaces that are associated with the specified DB2 subsystem ID or group ID on this LPAR.
BRD,STOP,IMSID=IMSID | Stop all Subzero address spaces that are associated with the specified IMS subsystem on this LPAR.
BRD,STOP,PIID=serverPIID | Stop the Subzero server address space that has the specified PIID on this LPAR.
BRD,STOP,ASIDX=serverASIDX | Stop the Subzero server address space that has the specified ASIDX on this LPAR.

### 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:

   `/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.

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

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

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

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 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=subsytemID,TASK=taskID
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 is running the task to be terminated.
- **taskID** specifies the identifier of the task.

You can obtain the PIID and ASIDX and the task ID 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.

### 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
     - **PIIID** 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.

**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 that 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 46.

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. 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 73
- “Optionally editing the Subzero initialization script (BRDINIT)” on page 74

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
```

**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 any of the following statements:

   - PROC specifies BRDPROC as the procedure name and includes the ACC parameter in case the EAGENT step requires account information.
EXEC identifies the program to execute (BRDJANUS) and the execution parameters to use. BRDJANUS 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.

STEPLIB must include 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, STEPLIB 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 33.

All libraries in the STEPLIB concatenation must be APF authorized.

DBCPRINT identifies the data set to which the DBC external agent stub writes a small amount of information.

BMCPSWD identifies the data set that contains the BMC license keys (CPU authorization passwords) that enable Subzero for use in your environment.

BRDPRINT and BRDERRO (not shown) 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.

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:

```
<!-- BRD VERSION 1.00 DBC INIT -->
<DBCUTIL>
  <OPTIONS>
    <MAXRESPONSES>1</MAXRESPONSES>
  </OPTIONS>
  <COMMANDS>
    <COMMAND>
      <REPOSITORY>
        <UPDATE>
          <TYPE>INITPROD</TYPE>
        </UPDATE>
      </REPOSITORY>
    </COMMAND>
  </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>`<procName> 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 instead of the default BRDCNTRL. The job names of Subzero subsystem servers are based on this job name; DBC replaces the last four characters with the subsystem ID of the associated DB2 or IMS subsystem.

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.
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:

```
//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`

   `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`
   - `F group,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 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:

```
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 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 ssid variable refers to the subsystem ID of the DBC subsystem.

Table 3: 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, DBC2, 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, TRACEFLUSH</td>
<td>Forces unwritten trace information to the DBCTRACE data set. This command is useful if trace was disabled. DBC writes any trace data that exists in an internal buffer that is used to maintain trace information.</td>
</tr>
<tr>
<td>( ssid ) DBC, PRODCMDS, ppp</td>
<td>Displays all commands for product ppp. <strong>(Only displays commands that conform to the ppp,command format)</strong>.</td>
</tr>
<tr>
<td>( ssid ) DBC, DISPLAY, object</td>
<td>The details of the DBC, DISPLAY command for each object type are documented in the following table.</td>
</tr>
</tbody>
</table>

Table 4: DISPLAY Command Options

<table>
<thead>
<tr>
<th>DISPLAY object (alternative syntax)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCTS (PRODUCTS, PRD, PDXD)</td>
<td>Displays the Product Definition XML Document (PDXD) from the DBC repository. If no parameters are specified, information is displayed for all products</td>
</tr>
<tr>
<td>( ), ppp</td>
<td>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</td>
<td>Displays all operator command definitions. If no parameters are specified, commands for all products are displayed.</td>
</tr>
<tr>
<td>( ), ppp</td>
<td>If the three-character product code is specified, DBC displays only commands for that product.</td>
</tr>
<tr>
<td>EVENT (EV), eventName</td>
<td>Displays all subscription information the named event</td>
</tr>
<tr>
<td>subid</td>
<td>If the subid is specified, only information for the named subscription is displayed.</td>
</tr>
<tr>
<td>DISPLAY object (alternative syntax)</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>CMD,cmdtext</strong></td>
<td>Displays the ACTION XML for the named command.</td>
</tr>
<tr>
<td><strong>AGENTSS (AE)</strong></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.</td>
</tr>
<tr>
<td>,ppp</td>
<td>Displays currently execution agents for only the product specified</td>
</tr>
<tr>
<td><strong>PROCESSES</strong></td>
<td>Displays all active PROCESSES. If no other parameters are specified, PROCESSES for all products are displayed.</td>
</tr>
<tr>
<td>,ppp</td>
<td>Displays currently active PROCESSES for only the product specified</td>
</tr>
<tr>
<td><strong>DB2</strong></td>
<td>Displays the DB2INFO block for DB2 subsystems. If no parameters are specified, this command displays DB2INFO blocks for all subsystems known to the DBC group.</td>
</tr>
<tr>
<td>,ssid</td>
<td>If the ssid is specified, DB2INFO blocks for only matching DB2 subsystems are displayed. You can partially use wildcards for this value by specifying an asterisk at the end (e.g. DE* selects all DB2 subsystems with ssid beginning &quot;DE&quot;).</td>
</tr>
<tr>
<td>,LOCAL</td>
<td>ALL</td>
</tr>
<tr>
<td><strong>IMS</strong></td>
<td>Displays the IMSINFO block for IMS subsystems. If no parameters are specified, this command displays IMSINFO blocks for all subsystems known to the DBC group.</td>
</tr>
<tr>
<td>,imsid</td>
<td>If the imsid is specified, IMSINFO blocks for only matching IMS subsystems are displayed. You can partially use wildcards for this value by specifying an asterisk at the end (e.g. MX* selects all IMS subsystems with IMSID beginning &quot;MX&quot;).</td>
</tr>
<tr>
<td>,LOCAL</td>
<td>ALL</td>
</tr>
<tr>
<td><strong>MQ</strong></td>
<td>Displays the MQINFO block for WebSphere MQ subsystems. If no parameters are specified, this command displays MQ blocks for all subsystems known to the DBC group.</td>
</tr>
<tr>
<td>,ssid</td>
<td>If the ssid is specified, MQINFO blocks for only matching MQ subsystems are displayed. You can partially use wildcards for this value by specifying an asterisk at the end (e.g. MX* selects all MQ subsystems with SSID beginning &quot;MX&quot;).</td>
</tr>
<tr>
<td>,LOCAL</td>
<td>ALL</td>
</tr>
<tr>
<td><strong>SRBS</strong></td>
<td>Displays currently active SRBs. If no parameters are specified, SRBs for all products are displayed.</td>
</tr>
<tr>
<td>DISPLAY object (alternative syntax)</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>,ppp</td>
<td>Displays currently active SRBs only for the product specified</td>
</tr>
<tr>
<td>STATS</td>
<td>If no parameters are specified, this command displays all available statistics.</td>
</tr>
<tr>
<td>,TCBS, {	extit{name}}</td>
<td>Displays statistics for only the named task</td>
</tr>
<tr>
<td>,MEM</td>
<td>Displays memory usage statistics</td>
</tr>
<tr>
<td>,QUEUES</td>
<td>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>
Troubleshooting Subzero problems

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 83
- “Potential problems” on page 86
- “Frequently asked questions about Subzero” on page 87

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 PRODHHELP 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
```

**Diagnostic data sets**

If a problem occurs, BMC Customer Support might ask you to send data sets to BMC for use in diagnosing the problem.

For example, if an abnormal termination occurs, BMC Customer Support might request the following resources:
- SVC dump data set

   **Note**
   A complete system dump is required. An abbreviated dump, such as might be useful in applications programming environments, is inadequate.

- JES JOBLOG data set

- A listing of the Subzero configuration values (option set), which is located in the DBCPRINT data set

In addition, output data sets from the following areas might also be helpful when diagnosing problems:

<table>
<thead>
<tr>
<th>Component or participant</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBC address spaces</td>
<td>Client and target LPARs</td>
</tr>
<tr>
<td>Subzero controlling agents</td>
<td>Client and target LPARs</td>
</tr>
<tr>
<td>CICS regions</td>
<td>Client LPAR</td>
</tr>
<tr>
<td>Subzero subsystem servers</td>
<td>Target LPAR</td>
</tr>
<tr>
<td>DB2 master address spaces</td>
<td>Target LPAR</td>
</tr>
<tr>
<td>IMS control regions</td>
<td>Target LPAR</td>
</tr>
</tbody>
</table>

**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>This command</td>
<td>Enables or disables debug tracing for</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BRD,DEBUG,{ON</td>
<td>OFF},PIID=server_piid</td>
</tr>
<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.

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.
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.

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.

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.

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 39.

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.

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.

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.

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

Subzero can handle a maximum of 8,192 individual threads for remote connections. This amount represents the accumulated number of threads from all client address spaces to a remote subsystem on any LPAR within the sysplex.
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.

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.

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.
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.

Does Subzero require assignment of an XCF transport class?

No, you are not required to assign an XCF transport class to the XCF group that Subzero is using. Subzero automatically selects the largest "pipe" it can find. However, you can assign a transport class if you want to do so.

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.

What should I do to gather diagnostic information from CICS and 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 WebSphere by using Open Transaction Manager Access (OTMA) to split your IMS environment away from the WebSphere LPARs.

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.
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.

When does the Subzero controlling agent start, and where does it run?

The Subzero controlling agent starts when DBC starts and runs on each LPAR that DBC runs on.

The controlling agent is responsible for setting up the connection between a client on the local LPAR and a subsystem on a remote LPAR, and for defining and starting Subzero subsystem servers.
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 96
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 60.
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.

5 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</td>
<td>CDB2CON</td>
<td>MainView for CICS</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 98
- “Subzero Servers (RDSRVR) view” on page 99
- “Subzero Redirect Targets (RDTOPOL) view” on page 100

### Subzero Groups (RDGROUP) view

The Subzero Groups view, which you can access directly by using the RDGROUP 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 9: 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 10: Subzero Servers (RDSRVR) view

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 11: 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 12 on page 102).

Figure 12: Subzero containers in MainView Explorer

The following Subzero containers are available:

- “Subzero Flow container” on page 102
- “Subzero Topology container” on page 105

**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 13 on page 103 shows the flow for a single Subzero group.

**Figure 13: 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 14 on page 104 shows three Subzero groups that are redirecting subsystem access requests from multiple CICS regions to various DB2 and IMS subsystems.

**Figure 14: 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 15 on page 106 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 15: 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 115.
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
See “XCF” on page 115.

D

DB2 Component Services

See “DBC” on page 111.

DB2 Product Configuration

See “DB2 Product Configuration” on page 111.

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 109.

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 110.*

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.

M

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 109.

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 115.

rolling four-hour average

See “4HRA” on page 109.

RTCS

Runtime Component System. A BMC infrastructure component that LGC uses to manage option sets for Subzero.

Runtime Component System

See “RTCS” on page 114.

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 114.

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 109
64-bit Pause Services 30

A
abend code U3549 83
abend propagation 88
accounting records. See SMF
address spaces
   requirements 30
   stopping 67
agents. See controlling agents or LGC
APF requirements 34
API for Subzero 60
applications
   no changes needed for Subzero 16
   supported types 31
architecture of Subzero 22
ASIDX
   definition 109
authorization passwords. See license keys
automated startup 73

B
backout scenarios 16
BBIISPxx member 96
BBQBE104I message 96
benefits of Subzero 16
BMC Cost Analyzer for zEnterprise product 51
BMC Intelligent Capping for zEnterprise
   definition 109
BMCBRD0022I message 87
BMCBRD0026I message 87
BMCBRD0150I message 87
BMCPSWD data set 73, 76
BRD
   definition 110
BRDAPPI application programming interface 60, 96
BRDCNTL address space 27
BRDDCOLL parameter 96
BRDERROR data set 73, 83
BRDGROUP parameter 96
BRDINIT initialization script 74
BRDPRT00 PLT program 58
BRDPRINT data set 72, 73, 83
BRDPROC procedure
   APF requirements 34
   named in initialization script 74
   setup 73
   source of DFSPZPxx module 33
BRDSNAP data set 88
BRDssid address space 28

C
candidates for redirection 53
CDB2CON view 106
CDS (couple data set) 42
CICS
   configuring for Subzero 58
   connection to local and remote subsystems 32
   DBC must start first 26
   IMS connections 33
   no transaction definition changes needed for
      Subzero 16
   PLT changes 58
   remote connection 87
   requirements 87
   SMF records 39
   startup JCL 58
   Subzero enabled 87
   support 32
CLASSDEF statement 42
clearing the BRDPRINT data set 72
client interface tasks 27
clientLPAR field 56
clients
  definition 110
  identifying candidates 47
CLIST for LGC 54
commands
  CONFIG,RELOAD 71
  console 63
  DBC 78
  DEBUG 85
  DIAG 78
  DISPLAY 69
  DISPLAY,MAP 70
  HELP 64, 84
  KILL 67
  LICENSE,RELOAD 71
  overview 64
  PRTFLUSH 72
  REFRESH 66
  RELOAD 71
  RESETPRINT 78
  RESETTRACE 78
  RESTART 66
  START 66
  STOP 67, 78
  Subzero 64
  TRACEFLUSH 78
  TRACEOFF 78
  TRACEON 78
components
  controlling agents 27
  DBC 25
  diagnostic facilities 28
  LGC 26
  overview 22, 25
  recovery facilities 28
  RTCS 26
  subsystem servers 28
CONFIG,RELOAD command 71
configuration options 56
considerations
  operational procedures and policies 40
  overview 37
  performance 37
  SMF 39
  subsystem access 38
console commands 63
containers
  Subzero Flow 102
  Subzero Topology 105
controlling agents
  definition 110
  displaying statistics 69
  job name 74
  overview 27
  refreshing with new startup values 66
  restarting 66
  starting 66
  stopping 67
conventions, documentation 10
Cost Analyzer product 51
couple data set 42
CPU authorization passwords. See license keys
Create New Option Set pop-up panel 54
CREGDBC view 106
CREGDBCD view 106
cross-system coupling facility. See XCF

D

Database Resource Adapter 33
DB2
  group IDs 32, 56
  access considerations 38
  data sharing group 88
  example of separation from CICS 18
  group IDs 32, 56
  requirements 32
  SMF records 39
  support 32
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 34
  definition 111
  overview 25
  securing 60
  started task 76
  starting during installation 45
  startup procedure 77
  stopping 77
DBC commands
E

ECSA
definition 111

error messages 83
tables
examples
consolidating subsystems in a test environment 21
DB2 separated from CICS 18
deployment 17
IMS consolidated on a single LPAR 19
reduced IMS footprint 19
volume-based separation of activities 20

Execute Extensions Facility 30
Execute Immediate Facility 30

F

fallback scenarios 16
FAQs 87
flushing the BRDPRINT data set 72
FMIDs 45
frequently asked questions 87

G

General Options section 56
generic LPAR specifications 47
GTF
definition 111

H

hardware requirements 29
HELP command 64
Help, online 84

I

ICF processors 42
IFI calls, supported 32
IMS
access considerations 38
CICS connections 33
Database Resource Adapter 33
example of consolidation on a single LPAR 19
example of reducing the footprint 19
requirements 33
SMF records 39
support 33
IMS Subsystems to Redirect section 56
IMSCONR view 106
initialization script 74
installing Subzero components 45
Integrated Coupling Facility (ICF) processors 42
ISPF interface 54
ITASTRAC view 106
IXC4311 message 42

K

KILL command 67

L

LGC
definition 111, 112
Help in ISPF interface 84
ISPF interface 54
overview 26
LGCISPF CLIST 54
license keys
   in DBC started task procedure 76
   installing 45
   reloading 71
LICENSE,RELOAD command 71
Linkage Index Reuse Facility 30
local subsystems
   connections 32, 89
   definition 112
   inadvertent connection 86
   precedence over remote subsystems 16
locks, database 37
Long Displacement Facility 30
LPAR
   definition 112
LX Reuse Facility 30

M

MainView
   accounting records 39
   definition 112
   enabling for Subzero 96
   requirements 35
   Subzero information in views and containers 97
   support 35
   using with Subzero 95
master controlling agent
   definition 112
MEMSTALLTIME parameter 42
messages, overview of Subzero 83
Millions of Service Units. See MSUs
MLCs
   definition 112
monitoring software 91
MSGBUFFERSIZE command 42
MSU allocations 41
MSUs (Millions of Service Units)
   definition 112

N

Notes field 56

O

online Help 84

operating Subzero 63
operating system requirements 30
option sets
   creating 54
   definition 113
   displaying future options in effect 70
   displaying options in effect 70
   modifying 54
originalSSID field 56

P

paging subsystem 41
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 37
   monitoring products 37
PIID
   definition 113
PLT
   definition 113
PLT changes for Subzero 58
PLTPI parameter 58
problems with Subzero 83
local connections 86
remote connections 86
processing group 31
Product Authorization utility 45
product license keys. See license keys
Product Option Sets panel 54
program list table. See PLT
PRTFLUSH command 72

R

R4HA. See 4HRA
RDGROUP view 98
RDSRVR view 99
RDTOPOL view 100
recovery scenarios 16
Redirect Targets view 100
redirection
collecting candidate information 53
defining 46
definition 113
guidelines for candidates 47
ISPF interface for defining 54
options 56
workflow 24
REFRESH command 66
RELOAD command 71
remote subsystems
connection problems 86
connections 32
definition 114
inadvertent connection to local 89
Subzero messages if connected 87
requirements
address space 30
APF 34
CICS 32
DB2 32
hardware 29
IMS 33
MainView 35
operating system 30
security 34
subsystems 31
XCF 31
RESETPRINT command 79
RESETTRACE command 79
RESTART command 66
RTCS 26
definition 114
Runtime Component System. See RTCS

S

SAF
definition 114
securing Subzero components 60
scenarios
deployment examples 17
See also examples
SDSF 63
security
DBCSECUR data set 76
STATAPR resource 60
security requirements 34
SET MSGBUFFERSIZE command 42
SMF

CICS 39
considerations 39
DB2 39
definition 114, 115
IMS 39
reconciling redirected requests 88
SMP/E 45
SQL calls, supported 32
START command 66
starting DBC during installation 45
startup automation 73
startup procedure
DBC 76, 77
Subzero 73
STATAPR resource 60, 96
definition 114
statistics, displaying 69
STOP command 67, 79
stopping DBC 77
subsystem access considerations 38
subsystem servers
definition 114
displaying statistics 69
overview 28
stopping 67
Subzero
architecture 22
automating startup 73
benefits 16
client interface tasks 27
components 22
considerations 37
console commands 64
t tunneling 63
controlling agents 27
definition 115
deployment examples 17
enabling for MainView 96
FAQs 87
installation 45
operating 63
overview 13
performance 37
products 13
recovery and diagnostic facilities 28
redirection definitions 46
redirection example 14
redirection options 56
redirection workflow 24
requirements 29
securing components 60
setting up 45
subsystem servers 28
troubleshooting 83
tuning 37
using with MainView 95
Subzero Flow container 102
Subzero Groups view 98
Subzero Servers view 99
Subzero Topology container 105
SVC dump data set 84
syntax statement conventions 10
Sysplex Failure Management (SFM) 42
system authorization facility. See SAF
System Display and Search Facility. See SDSF
System Management Facilities. See SMF

t

targetLPAR field 56

targets
  definition 115
  identifying candidates 47
targetSSID field 56
terminating address spaces 67
testing
  consolidating subsystems 21
guidelines 47
topology view 105
trace facility 85
TRACEFLUSH command 79
TRACEOFF command 79
TRACEON command
transition queues 37
transaction throughput 91
troubleshooting Subzero problems 83
tuning 37
  overview 41
  z/OS 41

U

U3549 abend code 83

V

views
  CDB2COND 106
  CREGDBC 106
  CREGDBCD 106
  IMSCONNR 106
  ITASTRAC 106
  RDGROUP 98
  RDSRVR 99
  RDTOPOL 100
Redirect Targets 100
Subzero Groups 98
Subzero Servers 99

W

waiting tasks, terminating 67
wildcard character 47
WLM 89
WLM (Workload Manager)
definition 115
workflow, redirection 24
Workload Manager. See WLM

X

XCF
definition 115
group name 31
performance 37
requirements 31
tuning for Subzero 42
XCFGROUP option 56

Z

z/OS
requirements 30
tuning for Subzero 41