Runtime Component System Configuration and Administration Guide

Version 1.1

March 31, 2003

Restricted Rights Legend

BMC Software, Inc.
2101 CityWest Blvd.
Houston TX 77042-2827
USA

Contacting BMC Software

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

United States and Canada

Address BMC Software, Inc.
2101 CityWest Blvd.
Houston TX 77042-2827

Telephone 713 918 8800 or
800 841 2031

Fax 713 918 8000

Outside United States and Canada

Telephone (01) 713 918 8800

Fax (01) 713 918 8000

Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see "Before Contacting BMC Software."

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at **http://www.bmc.com/support.html**. From this Web site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software 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 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at **www.bmc.com/support.html**.

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem 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, such as `file system full`
  - — messages from related software

Contents

Figures

Tables

About This Book

This book contains

- an overview of the SMP/E-based installation process for the Runtime Component System (RTCS) and associated RTCS-based products that use the Desktop Installation Assistant
- information about configuring an MVS image to start the RTCS subsystem and execute RTCS-based products
- detailed information about administering RTCS

This book is written for OS/390 and z/OS technical professionals and managers who are involved in the installation, deployment, and management of RTCS-based products, as well as the configuration and management of OS/390 and z/OS systems to support those products.

To use this book, you should be familiar with the following items:

- configuration of your MVS images, DASD volumes, ICF catalog structure, data set naming conventions, and TCP/IP communication environment
- how to install an OS/390 or z/OS systems programming product by using SMP/E
- MVS job control language (JCL), MVS operator commands, and the installed external security manager (ESM)
- your client workstation operating system

  For example, you should know how to start programs and perform common actions in a Microsoft Windows environment (such as choosing menu items and resizing windows).

How This Book Is Organized

This book is organized as follows.

Chapter/Appendix	Description
Chapter 1, “RTCS Infrastructure Overview”	provides conceptual information about the RTCS subsystem
Chapter 2, “Using RTCS Product Libraries”	introduces you to the target libraries that are created or updated by SMP/E during the installation of RTCS and RTCS-based products
Chapter 3, “Configuring a Target MVS Image for RTCS Execution”	explains how to configure an MVS image to be able to start the RTCS subsystem, initialize the System Registry, and then execute RTCS-based products
Chapter 4, “Installing and Configuring RTCS-based Products”	contains installation and configuration information for RTCS-based products
Chapter 5, “RTCS Administration”	contains information about how to use the administrative functions that are available in RTCS
Chapter 6, “Registry Import Utility”	provides information about how to use the RTCS Registry Import Utility
Appendix A, “BMC Software Product Authorization”	contains BMC Software password, licensing, and product authorization information
Appendix B, “Sample JCL and MVS Image Configuration Members”	provides reference information that is related to PROCs and PARMLIB additions

Related Documentation

BMC Software products are supported by several types of documentation:

- online and printed books
- online Help
- release notes and other notices

In addition to this book and the Help, you can find useful information in the publications listed in the following table. As “Online and Printed Books” on page xv explains, these publications are available online and on the product CD.

Document	Description
Desktop Installation Assistant Getting Started	guide that provides information about using the Desktop Installation Assistant and MVS system requirements
Runtime Component System Message Reference	online reference that contains information about messages that are generated by RTCS

Online and Printed Books

The books that accompany BMC Software products are available in online and printed formats. Online books are formatted as Portable Document Format (PDF) files. Some online books are also formatted as HTML files.

To Access Online Books

To view any online book that BMC Software offers, visit the Customer Support page of the BMC Software Web site at **http://www.bmc.com/support_home**. You can also access PDF books from the CD that accompanies your product.

Use the free Acrobat Reader from Adobe Systems to view, print, or copy PDF files. In some cases, installing the Acrobat Reader and downloading the online books is an optional part of the product installation process. For information about downloading the free reader from the Web, go to the Adobe Systems site at **http://www.adobe.com**.

To Request Additional Printed Books

BMC Software provides some printed books with your product order. To request additional books, go to **http://www.bmc.com/support_home**.

Online Help

The RTCS-based products and the Desktop Installation Assistant include Help. Refer to your product-specific documentation for the ways to access the product's Help facilities.

Release Notes and Other Notices

Printed release notes accompany each BMC Software product. Release notes provide current information such as

- updates to the installation and MVS image configuration instructions
- last-minute product information

In addition, BMC Software sometimes provides updated product information between releases (in the form of a flash or a technical bulletin, for example). The latest versions of the release notes and other notices are available on the Web at **http://www.bmc.com/support_home**.

Conventions

This section provides examples of the conventions that are used in this book.

General Conventions

This book uses the following general conventions:

Item	Example
information that you are instructed to type	Type **SEARCH DB** in the designated field. Type **search db** in the designated field. (workstation)
specific (standard) keyboard key names	Press **Enter**.
GUI elements and menu sequences	Choose **File => Open**.
directories, file names, Web addresses, e-mail addresses	The BMC Software home page is at **www.bmc.com**.

code examples, syntax statements, system messages, screen text	`//SYSPRINT DD SYSOUT=*`
	
	`The table` *`tableName`* `is not available.`

Item	Example
emphasized words, new terms	The instructions that you give to the software are called *commands*.
variables	In this message, the variable *fileName* represents the file that caused the error.

This book uses the following types of special text:

Note: Notes contain important information that you should consider.

Warning! Warnings alert you to situations that could cause problems, such as loss of data, if you do not follow instructions carefully.

Tip: Tips contain information that might improve product performance or that might make procedures easier to follow.

Syntax Statements

Syntax statements appear in Courier. The following example shows a sample syntax statement:

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

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

Item	Example
Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words.	*alias* *databaseDirectory*
Brackets indicate a group of options. You can choose at least one of the items in the group, but none of them is required. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option. Unix options are indicated with a hyphen.	[*tableName, columnName, field*]
Braces enclose a list of required items. You must enter at least one of the items. Do not type the braces when you enter the item.	{*DBDName* | *tableName*}

Item	Example
A vertical bar means that you can choose only one of the listed items. In the example, you would choose either *commit* or *cancel*.	`{commit | cancel}`
An ellipsis indicates that you can repeat the previous item or items as many times as necessary.	*`columnName`* `. . .`

Chapter 1 RTCS Infrastructure Overview

This chapter provides conceptual information about the RTCS Subsystem, the RTCS kernel, the RTCS Generalized Server, and their support for RTCS-based products.

This chapter contains the following sections:

Introduction

The Runtime Component System (RTCS) is an infrastructure that uses the latest OS/390 and z/OS system facilities to take advantage of modern enterprise servers. RTCS enables product developers to use the power of parallel sysplex, TCP/IP, UNIX System Services, cross-memory services, data spaces, and the recovery/termination manager (RTM), as well as to properly support most features of the external security managers (ESMs) used with MVS.

RTCS forms the basis for a family of user-friendly, easy-to-install products that are designed to simplify product installation and configuration.

The RTCS infrastructure is established in a target MVS image by a dedicated MVS subsystem control address space that remains active for the life of the IPL. Once initialized, the RTCS Subsystem control address space does not execute any code other than what is necessary to respond to operator commands. All RTCS administrative functions and all RTCS-based products execute in either a dedicated product-specific server address space or in an RTCS Generalized Server address space that is started by RTCS for the exclusive use by one or more RTCS-based products.

Installation and Configuration Overview

Before you can execute an RTCS-based product, you must install RTCS, properly configure each target MVS image on which you intend to use the product, initialize the RTCS Subsystem, and initialize the RTCS System Registry for each target MVS image.

The following steps provide an overview of the tasks that you need to complete to be able to execute RTCS and RTCS-based products. These steps are explained in more detail throughout this guide.

1. Install RTCS. RTCS-based products may also be (and usually are) installed at the same time. It is usually necessary to install RTCS and all RTCS-based products only once. The SMP/E target libraries can be made available to each MVS image using shared DASD volumes, or a copy of each necessary target library can be made for use in production on a specific target MVS image or a group of MVS images.

Note: For detailed information about installation system requirements and using the Desktop Installation Assistant to install RTCS and RTCS-based products, see the *Desktop Installation Assistant Getting Started*.

2. Configure the MVS image to be able to start the RTCS Subsystem. This consists of updating the Program Properties Table (PPT) and the APF library list. This needs to be done once for each target MVS image.

3. Allocate a VSAM linear data set (LDS) for the RTCS System Registry for each target MVS image on which RTCS is to be started.

4. Customize the RTCS Initiator started task (STC) procedure (PROC) JCL DD statements to specify the DSNAMEs that are to be used for the following data sets: the RTCS Subsystem Program Library, the RTCS Product Program Library, the RTCS Hypertext Document Library, the RTCS System Registry VSAM LDS, and the Product Authorization Table Library.

5. Issue the necessary ESM administrative commands to define started task user IDs for the RTCS Initiator, Subsystem, and Generalized Server address spaces, and to grant access to the necessary data sets and other resources.

6. Start the RTCS Initiator, which will start the RTCS Subsystem control address space for the first time, which initializes in an unconfigured mode (because of the empty, newly-allocated System Registry VSAM LDS).

7. After ensuring that the RTCS Subsystem control address space started successfully, initialize the RTCS System Registry for that target MVS image by executing the RTCS Registry Import Utility (RIU).

After you have completed the preceding steps, the RTCS Subsystem is active on that MVS image and no further customization is required. Ensure that RTCS starts as soon as possible on each MVS image during the IPL process. RTCS must successfully initialize before RTCS-based products can be started or executed. One way to accomplish this, for example, is to update the system commands member in SYS1.PARMLIB or the target MVS image's logical PARMLIB data set.

Once RTCS is active on an MVS image, the installation and configuration of RTCS-based products consists of the following steps:

1. Install the RTCS-based product using SMP/E into the same SMP/E environment (distribution and target zones).

Note: For detailed information about installation system requirements and using the Desktop Installation Assistant to install RTCS and RTCS-based products, see the *Desktop Installation Assistant Getting Started*.

2. If necessary, update any copies of the SMP/E target libraries that are used in production on each of the affected MVS images.

3. Update the RTCS System Registry for the RTCS-based product on each MVS image on which the product is to be executed.

RTCS has been designed so that RTCS-based products can be simple to install and easy to configure. Once RTCS is running on an MVS image, most RTCS-based products can be installed and configured by populating the target libraries (which in most cases already exist) and then either running the RIU to update the System Registry or by entering an RTCS command that directs RTCS to discover and automatically configure any newly-installed products.

It is technically possible to install RTCS and several RTCS-based products, as well as configure a target MVS image to be able to run them, in less than 30 minutes, without having to IPL the OS/390 or z/OS target system.

RTCS Kernel and RTCS Subsystem Address Space

Services provided by RTCS are established and maintained on each target MVS image by a dedicated MVS subsystem control address space which runs under the MSTR subsystem and remains active for the life of the IPL.

The following components comprise the RTCS infrastructure:

- RTCS kernel
- RTCS Subsystem address space
- RTCS System Registry
- RTCS Generalized Server address spaces

RTCS Kernel

RTCS runs as an MVS subsystem, which is required by all RTCS-based and RTCS-dependent products. During initialization, the RTCS kernel is loaded into above-the-line common storage by the RTCS Subsystem. All of the code that is directly used by RTCS-based products is physically located in the kernel; there are no references to code in the RTCS address space and none are allowed. Once the RTCS Subsystem has been successfully initialized and the RTCS kernel has been loaded, RTCS services are available to applications and products that have been written to use them.

Only one instance of RTCS exists on each MVS image. After the RTCS kernel is initialized, the RTCS Subsystem control address space remains in the system as the owner of certain MVS resources, but it does not normally execute any code. RTCS-based products interact only with the RTCS kernel, not with the RTCS Subsystem address space itself. Certain MVS operator commands, such as MODIFY, which are directed specifically to the RTCS Subsystem address space, cause some code to be executed in the RTCS Subsystem control address space. Operator commands are normally required only when you want to start an RTCS Generalized Server (product) address space.

Only one instance of RTCS can execute on each MVS image. Once the RTCS kernel is initialized, the RTCS Subsystem control address space remains in the system as the owner of certain MVS resources, but it does not normally execute instructions. RTCS-based products interact only with the kernel and not directly with the RTCS Subsystem address space. Only operator commands such as MODIFY, directed specifically to RTCS, cause any instructions to be executed in the RTCS Subsystem control address space. Operator commands are normally only required when you want to start RTCS Generalized Server (product) address spaces.

RTCS Subsystem Address Space

The RTCS Subsystem address space cannot be shut down. The main responsibilities of the RTCS Subsystem address space are to initialize the kernel and provide the point of ownership for certain MVS resources. MVS exits and cross-memory resources that are established by the RTCS Subsystem are used by functions and code in other permanent MVS system address spaces, which also never terminate. Accordingly, these RTCS functions, MVS resources, and the RTCS kernel code must remain available for the life of the IPL.

Because the RTCS Subsystem address space cannot be stopped, starting it under the primary JES (or any JES) as a started task is not allowed. Instead, it *must* be started under the master subsystem by means of SUB=MSTR. If RTCS is not started this way, you would never be able to perform a normal shutdown of the primary JES.

Because the RTCS Subsystem address space cannot be shut down after it is started, only a limited number of data sets are allocated to it. The RTCS System RegistryVSAM LDS is the only data set that normally remains allocated. The RTCS Subsystem Program Library is dynamically unallocated after the RTCS kernel is initialized.

The RTCS Subsystem Program Library is a separate, dedicated SMP/E target library not shared by any other product. It is dynamically allocated by the RTCS Subsystem address space. The RTCS Subsystem Program Library is required to be a partitioned data set extended (PDSE) program library. This data set should be cataloged in the target MVS image's master catalog; however, this is just a suggestion and is not enforced.

RTCS System Registry

The System Registry is a repository for configuration parameters, product definitions, user profiles, and other typical PARMLIB-type data. The RTCS Initiator validates the specified VSAM cluster and passes its DSNAME to the RTCS Subsystem address space, which later dynamically allocates it during RTCS kernel initialization.

The RTCS System Registry can be seen as read/write auxiliary storage for the RTCS Subsystem and all RTCS-based products. The RTCS System Registry is accessed as if it were in virtual storage, but it is actually stored on DASD in a VSAM LDS. The RTCS System Registry is managed and accessed much like an MVS page data set, except that it is organized in a tree-like structure, similar to the Microsoft Windows registry. This structure speeds up the process of locating data in the registry; complex, self-defining structures can be stored as easily as a byte or string of characters.

The RTCS System Registry is backed by a VSAM LDS cluster, which is defined by the REGISTRY DD statement in the RTCS Initiator started task PROC JCL.

When you define the VSAM LDS cluster to be used for the RTCS System Registry, choose its DSNAME so that it will be cataloged in an image-specific ICF catalog (such as the target MVS image's master catalog or any catalog that contains data sets that are referenced by critical system address spaces during MVS IPL). The catalog in which the RTCS System Registry VSAM LDS is defined is permanently allocated for the life of the IPL. Because it is tightly integrated with the MVS image, the System Registry VSAM LDS should be named so that it will be cataloged in the MVS image's master catalog or a catalog that is not used by any other MVS image. As a last resort, you can use an ICF catalog that is already permanently allocated on a target MVS image (for example, because of use by other critical or permanently started tasks or system address spaces).

The preceding DSNAME and catalog allocation restrictions and suggestions also apply to the RTCS Subsystem and Product Program Libraries, as well as the RTCS Hypertext Document Library. If you want to share RTCS product data sets among multiple target MVS images, consider the DASD volume, catalog availability, and recovery issues in your configuration before choosing the DSNAMEs for the RTCS Subsystem Program Library, the RTCS Product Program Library, and the RTCS Hypertext Document Library.

RTCS products execute in address spaces that deallocate all data sets when they are shut down. Therefore, at many installations, it may only be important to isolate the System Registry VSAM LDS on an MVS image-specific DASD volume.

RTCS Generalized Server Address Space

Code for RTCS-based products does not execute in the RTCS Subsystem address space. Instead, RTCS-based products execute either in their own address space or are started by RTCS in an RTCS Generalized Server address space.

Each RTCS Generalized Server address space can run only products that execute in one particular storage protection key. When a Generalized Server address space is started, it is able to run products that execute in key 0, key 8, or any other valid protection key—but only in one key for any one Generalized Server address space. If products that execute in different protection keys need to be running at the same time, multiple Generalized Server address spaces are started.

A new RTCS Generalized Server address space is started by RTCS using an internally-issued MVS START command when you request that an RTCS-based product be started either by issuing an operator command or by using the online administrative interface *and* either of the following conditions applies:

- The product is defined to require its own dedicated MVS address space. In this case, a new address space dedicated for the use of this product is started.

- The product is defined to execute in a storage key for which there is no eligible Generalized Server address space already started. In this case, a new one is started that is eligible to run any product in the protection key needed for this product.

Products that can execute in multiple address spaces have as many Generalized Server address spaces started for them as the MVS Workload Manager (WLM) determines are necessary. Such address spaces are dedicated to executing such products so that they can be shut down when MVS WLM determines that fewer address spaces are needed.

A product that can execute in any eligible Generalized Server address space is initialized in one that has already been started, if one already exists that is allowed to run products that execute in the required protection key, there is sufficient virtual storage and other resources, and the MVS WLM determines that another address space is not necessary.

Chapter 2 Using RTCS Product Libraries

This chapter describes the product libraries used by the RTCS product family.

This chapter contains the following sections:

Introduction

The RTCS product family uses a limited number of product libraries. The contents of each of these libraries are established by an SMP/E-based installation process. This process constructs target libraries that are then copied for production use on a target MVS image.

Three data sets are used in production. They are the RTCS Hypertext Document Library and the two program library data sets: the RTCS Subsystem Program Library and the RTCS Product Program Library.

All three of these data sets must be allocated as a PDSE. The RTCS Subsystem Program Library contains program objects that are used only by the RTCS Subsystem address space. The RTCS Product Program Library contains program objects that are used by RTCS components and utilities and RTCS-based products, including Generalized Server address spaces. The RTCS Hypertext Document Library contains HTML files that are used by RTCS components and RTCS-based products in Generalized Server address spaces.

The remaining two non-production data sets are the RTCS XML Document Library and the RTCS Sample JCL and Parameter Library. They are normally allocated as PDSEs by the Desktop Installation Assistant, but it is not a requirement. These data sets are used during the MVS image configuration process and are never used by RTCS or RTCS-based products during normal execution.

The distributed target libraries are read-only data sets and should not be modified or customized. The program libraries and document libraries are read-only. Particularly, the sample JCL and Registry Parameter Library should not be changed. As the target library, any maintenance applied using SMP/E would overwrite any changes made by the installation.

Distribution libraries do not need to be customized. Installation and image-specific data needed by RTCS and RTCS-based products is retained in the RTCS System Registry.

Because all data sets are dynamically allocated by RTCS products, there is no need for product-specific started task procedures (PROCs), JCL, or any other customizations. No CLISTs or EXECs need to be installed. TSO commands and programs that are executed in batch are installed dynamically into the system so that JCL does not need to be customized.

RTCS Subsystem Program Library (TOSZRTCS)

This data set, which must be a PDSE, can be used *as is* in the RTCS Initiator PROC, where it must be defined in the STEPLIB DD statement. This library contains only RTCS Subsystem PM3 (DFSMS/MVS Release 4 compatible) program objects. This data set is used only during RTCS Subsystem initialization by the RTCS Initiator address space, and then the data set is dynamically allocated to the RTCS Subsystem address space. After the RTCS kernel has been initialized, this library is dynamically unallocated, and it is not referenced again unless the RTCS kernel needs to be refreshed.

You can use the actual SMP/E TOSZRTCS target data set that was constructed during the installation process. However, BMC Software recommends that you make a copy of this data set for use in production on a target MVS image. This copy should not be an SMS-managed data set. For example, it might be allocated on one of the target MVS image's SYSRES volumes.

RTCS Product Program Library (TOSZLINK)

This data set, which must be a PDSE, can be used *as is* in the RTCS Initiator PROC, where it must be defined in the TASKLIB DD statement. This library contains RTCS component and RTCS-based product PM3 (DFSMS/MVS release 4 compatible) program objects and contains no RTCS Subsystem code. This library is referenced and validated by the RTCS Initiator, which records its DSNAME. However, the library is not used during RTCS Subsystem initialization. This data set is dynamically allocated by the RTCS Generalized Server program (OSZEXEC[*n*]) when an RTCS component or RTCS-based product program is executed. Thus, this data set will be allocated to any address space that is running an RTCS-based product or RTCS component program or utility.

You may use the actual SMP/E TOSZLINK target data set that was constructed during the installation process. However, BMC Software recommends that you make a copy of this data set for use in production on a target MVS image. This copy should not be an SMS-managed data set. For example, it might be allocated on one of the target MVS image's SYSRES volumes.

RTCS Hypertext Document Library (TOSZHTML)

The members of this data set are HTML documents or binary, byte-stream files. Maintenance will be much more convenient if this data set is allocated as a PDSE. This data set can be used *as is* in the RTCS Initiator PROC, where it must be defined in the HTMLDOC DD statement. It will be dynamically allocated by RTCS-based product server address spaces.

You may use the actual SMP/E TOSZHTML target data set that was constructed during the installation process. However, BMC Software recommends that you make a copy of this data set for use in production on a target MVS image. This copy should not be an SMS-managed data set. For example, it might be allocated on one of the target MVS image's SYSRES volumes.

RTCS XML Document Library (TOSZRXML)

This data set is used by the RTCS RIU. This data set can be an ordinary PDS, but it is allocated by the Desktop Installation Assistant as a PDSE for consistency. This data set is never used by the RTCS Subsystem control address space, nor by a Generalized Server address space. The data set contains XML documents that are used in the initial setup of the RTCS System Registry and for product definition after installation of an RTCS-based product. The DSNAME of this data set needs to be specified in a SET statement in the SYSIN input stream each time the RTCS RIU is executed. For more information, see Chapter 6, "Registry Import Utility."

TOSZCNTL Sample JCL and Parameter Library

This data set can be an ordinary PDS, but it is allocated by the Desktop Installation Assistant as a PDSE for consistency. This data set contains several samples that you can use to configure the three RTCS started task PROCs. The data set also contains sample JCL to execute the RTCS RIU and sample parameter input members for RTCS and RTCS-based product registry definitions.

The following members are of particular interest after RTCS is installed, and when a target MVS image is being configured to run RTCS:

- OSZ$INIT
- OSZ$RTCS
- OSZ$EXEC
- OSZ$REGI
- OSZSCHED

The following sections provide instructions on how to use these members.

OSZ$INIT

OSZ$INIT is sample RTCS Initiator started task procedure (PROC) JCL. This member should be copied into SYS1.PROCLIB (or another master subsystem started task procedure library specified in the MVS image's Master JCL); it should be named OSZINIT. The PROC must be located in SYS1.PROCLIB (or equivalent), because OSZINIT must be started under the MVS Master Subsystem (MSTR). For example:

```
S OSZINIT,SUB=MSTR,SSID=RTCS,RTCS=OSZRTCS,EXEC=OSZEXEC
```

The STEPLIB, TASKLIB, REGISTRY, HTMLDOC, and BMCPSWD DD statements must be customized to specify the DSNAMEs of the RTCS Subsystem Program Library, the RTCS Product Program Library, the RTCS System Registry VSAM LDS, and the Product Authorization Table Library.

OSZ$RTCS

OSZ$RTCS is sample RTCS Subsystem started task procedure (PROC) JCL. This member should be copied into SYS1.PROCLIB (or another Master Subsystem started task procedure library specified in the MVS images master JCL); it should be named OSZRTCS; the actual name that is used is specified by the RTCS parameter when the RTCS Initiator is started. The PROC must be located in SYS1.PROCLIB (or equivalent) because OSZRTCS will be started by the RTCS Initiator (internally) under the MVS master (MSTR) subsystem; for example:

```
S OSZRTCS.RTCS,SUB=MSTR
```

Because no parameters, data sets, or DD statements exist in this PROC, customization is unnecessary; simply copy the PROC into the proper data set.

OSZ$EXEC

OSZ$EXEC is sample RTCS Generalized Server address space started task procedure (PROC) JCL. This member should be copied into both of the following locations: SYS1.PROCLIB (or another Master Subsystem started task procedure library that is specified in the MVS image's master JCL) and any JES started task procedure library. It should be named OSZEXEC, but this is not required; the actual name used is specified by the EXEC parameter that is specified when the RTCS Initiator is started.

Because no parameters, data sets, or DD statements exist in this PROC, customization is unnecessary; simply copy the PROC into the proper data set.

OSZ$REGI

OSZ$REGI is sample JCL used to execute the RTCS RIU. Copy this member to an installation-specific data set. In addition to the JOB statement, the JCL has DD statements that need to be customized to specify the DSNAMEs of the SMP/E TOSZRXML target XML document library as well as any customized XML document library.

OSZSCHED

OSZSCHED is a sample set of SYS1.PARMLIB PPT additions for RTCS. This member can be copied *as is* into the appropriate MVS target image-specific PARMLIB data set, either as a new member named SCHED*xx*, or merged into an existing PPT member.

Chapter 3 Configuring a Target MVS Image for RTCS Execution

This chapter explains how to configure a target MVS image in order to be able to start the RTCS Initiator, initialize the RTCS Subsystem, and execute RTCS-based or RTCS-dependent products that require RTCS services.

This chapter contains the following sections:

Introduction

This chapter describes the steps necessary to configure a target MVS image to bring up the RTCS Subsystem and configure RTCS-based products to execute on that newly configured MVS image.

Before you continue, ensure that the RTCS-related SMP/E target libraries exist and are populated with elements for RTCS itself, along with the selected RTCS-based products. For more information on these target libraries, see Chapter 2, “Using RTCS Product Libraries.”

Once RTCS and any RTCS-based products have been installed by SMP/E (or an equivalent process), a small number of configuration steps must be completed for each target MVS image before you can successfully start and initialize RTCS on that image. Basically, the MVS image configuration steps involve important additions that must be made to PARMLIB (for example, SYS1.PARMLIB or an MVS logical parameter library facility data set); the allocation of two PDSE program libraries, a PDSE library used for HTML documents, and a VSAM LDS used to back the RTCS System Registry; copying the started task PROCs to their appropriate procedure libraries (PROCLIBs) and properly customizing them; and APF-authorizing the program libraries. If BMC Software licensed products are to be used with RTCS, then a Product Authorization Table Library must be allocated and initialized or an existing library designated for use by RTCS.

In order for the RTCS Subsystem control address space to be able to establish RTCS in the MVS image as a formal MVS subsystem, the RTCS Initiator and the RTCS Subsystem started task programs must be able to receive control in the proper state and key from the MVS initiator. Started task PROCs to start these address spaces must be placed in the proper STC procedure libraries (PROCLIBs). In addition, required ESM attributes must be established for the RTCS Subsystem and Generalized Server address spaces. The required product data sets must be accessible from the target MVS image (or else copies made). A VSAM LDS for exclusive use by the RTCS System Registry on each MVS image must be allocated. After you have started RTCS for the first time on an MVS image, you must run the RTCS RIU to properly initialize the RTCS System Registry and customize it for that specific MVS image.

After completing the installation of RTCS and any RTCS-based products that were selected, the target data sets shown in Table 3-1 are available for use in configuring MVS to be able to start RTCS and run RTCS-based products:

Table 3-1 SMP/E Target Data Sets

Data Set	Description
tlibprfx.TOSZRTCS	RTCS Subsystem Program Library
tlibprfx.TOSZLINK	RTCS Product Program Library
tlibprfx.TOSZHTML	RTCS Hypertext Document Library
tlibprfx.TOSZRXML	XML Document Library
tlibprfx.TOSZCNTL	Configuration and sample job stream JCL Library
Note: *tlibprfx* represents the site-selected DSNAME prefix used to define and allocate the SMP/E target libraries.	

Of these five data sets, only the first three are required on an MVS image *in production* to execute RTCS and RTCS-based products.You may use the SMP/E target data sets directly for this purpose, but this is not recommended, especially for the two PDSE program libraries (TOSZRTCS and TOSZLINK). Preferably, you should make “production” copies and then use these copies to run RTCS on a target MVS image. In fact, since these data sets are PDSE libraries, it would be a good idea, and it could be an actual requirement in your environment, to make a dedicated copy of each of these three libraries for each target MVS image on which you want to run RTCS. This enables maintenance to be applied to the actual SMP/E target libraries using an allocation that specifies DISP=OLD, and then later copied to the production copies of these data sets that have been made for use on one or more specific MVS images. Such a procedure is commonly and informally employed for certain types of system software products, and so should be familiar to most system programmers. If you do choose to make “production” copies of the SMP/E target data sets, note that *only* the first three libraries listed in the table above need to have copies made. You will not need to make production copies of the TOSZRXML and TOSZCNTL libraries because they contain samples and are used *only* during RTCS Registry Initialization.

In general, it is undesirable to apply maintenance to a product using SMP/E at the same time that the SMP/E target libraries are being used on an MVS image. Results may be inaccurate. Depending on how and when the product accesses elements, such as programs, in those target libraries, it is possible for incorrect results or even ABENDs to occur when library members are accessed which are out of sync with other members that have already been accessed. This is just one reason it is recommended that the RTCS started task PROCs and the RTCS System Registry *not* be configured to use your actual SMP/E target libraries on any *production* machine. However, it is reasonable to configure a test MVS image or LPAR to use SMP/E-maintained libraries directly, where the consequences of an unexpected failure as a result of applying maintenance to libraries that are in use concurrently are usually deemed acceptable.

The following configuration steps assume that separate *production* copies of the first three data sets in Table 3-1 on page 3-3, are to be made and used on your production MVS images. In addition, notice the recommendations about sharing the three production target library copies.

While it is possible to share the RTCS Subsystem and Product Program Libraries (each of which *must* be a PDSE) and the RTCS Hypertext Document Library (which *should* be a PDSE due to certain characteristics of its contents) across multiple MVS images, you *must adhere* to the IBM MVS requirements that are prerequisite to sharing PDSEs in a sysplex. Even though the PDSEs are only being read and not updated, there are several critical requirements that *must* be met in order to share them across multiple MVS images.

In particular, they can only be shared among members of a sysplex (never outside), the GRSplex must match the sysplex, and there must be no additional members accessing the PDSEs from outside the sysplex or outside the GRSplex (CA-MIM *cannot* be used to mediate shared access, even in read-only mode, to PDSE libraries). The PDSE may appear to be or actually become corrupted, or the catalog or the volume where it resides may become corrupted, if you do not strictly adhere to the IBM MVS requirements, even if the libraries are only being read from multiple MVS images.

Warning! Pay close attention to the PDSE library sharing requirements.

Tip: Because of PDSE sharing requirements, BMC Software recommends that unique *production* copies of the three critical target libraries be made on and used exclusively with each individual target MVS image on which RTCS is to be started.

If you have your systems configured properly for PDSE sharing across all members of the sysplex, then you may share the two RTCS program libraries as well as the RTCS Hypertext Document Library (if it is a PDSE).

Target System Configuration

Configuring a target MVS system to be able to run RTCS consists of

- if necessary, allocating and making production copies of the SMP/E target libraries
- allocating or designating a dedicated VSAM LDS for the RTCS System Registry
- customizing the MVS image in order to be able to start RTCS the very first time on the target system (this consists of updating or creating APF and PPT entries in PARMLIB and started task procedures in PROCLIB)
- defining ESM user IDs and establishing necessary attributes and data set access profiles, permits, or rules
- allocating or designating a Product Authorization Table Library for use by RTCS-based licensed products
- starting RTCS on the target system
- using the RTCS RIU to initialize the RTCS System Registry (in effect, defining the “RTCS” product to RTCS) on the target system
- updating the Product Authorization Table, if necessary, for any licensed products that are to be used

When all of this has been accomplished, RTCS will be executing on the target system. You can then install RTCS-based products and configure them for execution without having to configure the target system again. In other words, only access to the RTCS System Registry is required in order to configure RTCS-based products for a specific target system once the products have been installed into the target libraries using SMP/E.

Tip: Another important step to consider is to make any necessary MVS or automation product customizations necessary to ensure RTCS is started automatically at each IPL.

Target system configuration can be accomplished by using the Desktop Installation Assistant or manually using, for example, ISPF PDF. Manual MVS image configuration is straightforward if you have access to a Desktop Installation Assistant-customized OSZINST installation data set or to an MVS image that has already been configured to run RTCS. You must still have access to the necessary MVS target image data sets and catalogs, but in many cases, especially with cloned sysplex MVS image configurations, the existing configuration changes previously made to other target MVS images are already in effect for or easily available to other MVS images that share the same data sets, such as catalogs, PROCLIBs, PARMLIBs, and so on.

At some installations, it will be simpler to configure additional target MVS images manually instead of by using the Desktop Installation Assistant, because of the small number of configuration changes that actually need to be accomplished for an additional image.

Using the Desktop Installation Assistant

You may use the Desktop Installation Assistant to configure an MVS image to be able to run RTCS and any installed RTCS-based products. For more information on the Desktop Installation Assistant, see the *Desktop Installation Assistant Getting Started*.

After you have started the Desktop Installation Assistant, perform the following steps:

Step 1 Sign on to the target MVS system.

Step 2 Select **MVS Image Configuration.**

Step 3 Select either **JOB Submission Mode** or **JCL Customization Mode**.

Essentially the same panels are displayed and the same information is requested regardless of the mode that you select. See the following sections for more information about these modes.

JOB Submission Mode

In JOB Submission mode, the Desktop Installation Assistant submits the MVS image configuration JOB stream directly to the target MVS system, waits for the JOBs to execute, and then examines them for successful execution. JOB Submission mode is simple, straightforward, and most effective if you have all of the necessary ESM authority to make any required MVS target system library updates, execute necessary ESM administrative commands, and enter any required MVS operator commands. If not, then more than one person must be involved to submit the required JOBs, such as a system programmer or security administrator; in this case, the JCL Customization mode may be a better choice.

JCL Customization Mode

In JCL Customization mode, the Desktop Installation Assistant only uploads the target MVS image configuration JOB stream into the designated installation data set (*prefix*.OSZINST). JCL Customization mode should be used if the configuration JOB stream needs to be customized after the Desktop Installation Assistant has uploaded it, or if the JOBs need to be submitted by different individuals or by different user IDs (for example, with different ESM authority).

The JOB stream that is customized by the Desktop Installation Assistant in JCL Customization mode is the same as the one that is used in JOB Submission mode, except for some minor differences. JOB number 70 provides a directory of all uploaded image configuration JOBs (and may indicate the presence of JOBs which are not actually used for a specific MVS target system). Submit all of the uploaded configuration JOBs from the Desktop Installation Assistant installation data set, starting with JOB number 70, one at a time, after checking each JOB for the correct data (such as DSNAMEs), and ensure that each JOB completes successfully before proceeding with the next JOB. There is no need to submit JOBs 89, 91, or 95; they are sample JOBs or just documentation to read; however, it is not a problem if you do submit these JOBs.

Manual Configuration of an MVS Image to Run RTCS

Complete the following steps to manually configure a target MVS image to run RTCS:

Step 1 Create an MVS image-specific RTCS Subsystem Program Library.

The TOSZRTCS target library, which is typically an SMS-managed PDSE, can be used *as is* in the RTCS Initiator started task PROC; however, this is *not* recommended for actual production configurations. No actual SMP/E target data set should be used in production on a target MVS image, since doing so may limit your ability to apply maintenance to the library. If this is what you want to do, proceed to the next step, and simply use *this* data set (the SMP/E TOSZRTCS target library) in the STEPLIB DD statement of the RTCS Initiator PROC, which is usually named OSZINIT (refer to Step 12).

Instead of specifying the SMP/E TOSZRTCS target library directly in the RTCS Initiator PROC, OSZINIT, the recommended procedure is to allocate a new PDSE library specifically for the purpose of running RTCS in production on a specific target MVS image, and then to copy the actual SMP/E TOSZRTCS target library into this new “production” copy. The new, production RTCS Subsystem Program Library production copy *must*, like the original, be a PDSE library. It *should* be allocated as a *non*-SMS-managed PDSE (that is, no STORCLAS specified), but this is not required. However, the data set *must* be cataloged, because it will be dynamically allocated by the RTCS Allocator (OSZMOSYS) to the RTCS Subsystem address space by DSNAME only. It is preferable that this data set be cataloged in the master catalog for the MVS image on which it is to be used, but it may be cataloged in an ICF catalog if you understand the consequences of such an action with respect to catalog and catalog volume availability during the IPL process, which is when RTCS is normally brought up.

It is possible to share this library among multiple MVS images, regardless of whether it is SMS-managed or non-SMS-managed. BMC Software recommends that the RTCS Subsystem Program Library *not* be shared, but instead have one production copy allocated for exclusive use on each target MVS image and cataloged in that target MVS image's master catalog. If you want to share this data set among production MVS images, you *must* ensure that your configuration adheres to all documented IBM MVS requirements for sharing PDSEs.

To allocate and copy the target SMP/E RTCS Subsystem Program Library into a new, non-SMS-managed PDSE, you can use the following sample JCL:

Figure 3-1 Sample JCL to Create RTCS Subsystem Program LIbrary

```
//OSZINJ71 JOB (acct),'CREATE OSZRTCS  PDSE',
//             MSGLEVEL=(1,1),CLASS=A,TIME=1
//*
//*  acct      JOB statement accounting information
//*  xxxxxx    VOL SER on which to allocate .OSZRTCS PDSE
//*  tlibprfx  DSN prefix of RTCS SMP/E target libraries
//*
//ALLOC   EXEC PGM=IEFBR14
//*
//*  Allocate a new non-SMS-managed cataloged PDSE for
//*  the RTCS Subsystem Library. This data set may be
//*  SMS-managed, but the JCL will have to be changed
//*  to allocate an SMS-managed data set. There is no
//*  particular need for the library to be SMS-managed.
//*  This data set CAN be shared by more than one system
//*  if MVS sysplex PDSE sharing requirements are met.
//*
//OSZRTCS   DD DSN=SYS1.RTCS.OSZRTCS,
//             DISP=(NEW,CATLG),
//             UNIT=3390,VOL=SER=xxxxxx,
//             DSNTYPE=LIBRARY,SPACE=(CYL,(10,5,01)),
//             DSORG=PO,RECFM=U,LRECL=0,BLKSIZE=23476
//*
//COPY    EXEC PGM=IEBCOPY,TIME=1,REGION=4M
//*
//* Copy SMP/E Target RTCS Subsystem LIBRARY
//* to the newly-allocated "production" PDSE.
//*
//SYSPRINT  DD SYSOUT=*
//TOSZRTCS  DD DISP=SHR,DSN=tlibprfx.TOSZRTCS
//OSZRTCS   DD DISP=SHR,DSN=*.ALLOC.OSZRTCS
//SYSIN     DD *
         COPY  INDD=((TOSZRTCS,R)),OUTDD=OSZRTCS
//*
```

Note: One of the most convenient places to allocate a non-SMS-managed PDSE library would be on one of the SYSRES volumes for an MVS image or on the volume that contains the master catalog if you are not sharing the master catalog. If you are sharing the master catalog, then it is very likely that you already have an MVS configuration that is eligible for sharing PDSE data sets across the sysplex.

Step 2 Create an MVS image-specific RTCS Product Program Library.

The TOSZLINK target library, which is typically an SMS-managed PDSE, can be used *as is* in the RTCS Initiator started task PROC; however, this is *not* recommended for actual production configurations. No actual SMP/E target data set should be used in production on a target MVS image, since doing so may limit your ability to apply maintenance to the library. If this is what you want to do, proceed to the next step, and simply use *this* data set (the SMP/E TOSZLINK target library) in the TASKLIB DD statement of the RTCS Initiator PROC, which is usually named OSZINIT (refer to Step 12).

Instead of specifying the SMP/E TOSZLINK target library directly in the RTCS Initiator PROC, OSZINIT, the recommended procedure is to allocate a new PDSE library specifically for the purpose of running RTCS products in production on a specific target MVS image, and then to copy the actual SMP/E TOSZLINK target library into this new “production” copy. The new, production RTCS Product Program Library production copy *must*, like the original, be a PDSE library. It *should* be allocated as a *non*-SMS-managed PDSE (that is, no STORCLAS specified), but this is not required. However, the data set *must* be cataloged, because it will be dynamically allocated by the RTCS Allocator (OSZMOSYS) to the RTCS Subsystem address space by DSNAME only. It is preferable that this data set be cataloged in the master catalog for the MVS image on which it is to be used, but it may be cataloged in an ICF catalog if you understand the consequences of such an action with respect to catalog and catalog volume availability during the IPL process, which is when RTCS is normally brought up.

It is possible to share this library among multiple MVS images, regardless of whether it is SMS-managed or non-SMS-managed. BMC Software recommends that the RTCS Product Program Library *not* be shared, but instead have one production copy allocated for exclusive use on each target MVS image and cataloged in that target MVS image’s master catalog. If you want to share this data set among production MVS images, you *must* ensure that your configuration adheres to all documented IBM MVS requirements for sharing PDSEs.

The DSNAME of this library must be specified to the RTCS RIU (refer to Step 17) with a SET statement in its SYSIN input stream. For example:

```
SET ISA.POSZLINK = SYS1.RTCS.OSZLINK
```

To allocate and copy the target SMP/E RTCS Product Program Library into a new, non-SMS-managed PDSE, you can use the following sample JCL:

```
//OSZINJ73 JOB (acct),'CREATE OSZLINK  PDSE',
//             MSGLEVEL=(1,1),CLASS=A,TIME=1
//*
//*  acct      JOB statement accounting information
//*  zzzzzz    VOL SER on which to allocate .OSZLINK PDSE
//*  tlibprfx  DSN prefix of RTCS SMP/E target libraries
//*
//ALLOC   EXEC PGM=IEFBR14
//*
//*  Allocate a new non-SMS-managed cataloged PDSE for
//*  the RTCS Product Library. This data set may be
//*  SMS-managed, but the JCL will have to be changed
//*  to allocate an SMS-managed data set. There is no
//*  particular need for the library to be SMS-managed.
//*  This data set CAN be shared by more than one system
//*  if MVS sysplex PDSE sharing requirements are met.
//*
//OSZLINK   DD DSN=SYS1.RTCS.OSZLINK,
//             DISP=(NEW,CATLG),
//             UNIT=3390,VOL=SER=zzzzzz,
//             DSNTYPE=LIBRARY,SPACE=(CYL,(35,5,01)),
//             DSORG=PO,RECFM=U,LRECL=0,BLKSIZE=23476
//*
//COPY    EXEC PGM=IEBCOPY,TIME=1,REGION=4M
//*
//* Copy SMP/E Target RTCS Product   LIBRARY
//* to the newly-allocated "production" PDSE.
//*
//SYSPRINT  DD SYSOUT=*
//TOSZLINK  DD DISP=SHR,DSN=tlibprfx.TOSZLINK
//OSZLINK   DD DISP=SHR,DSN=*.ALLOC.OSZLINK
//SYSIN     DD *
         COPY  INDD=((TOSZLINK,R)),OUTDD=OSZLINK
//*
```

Step 3 Create an MVS image-specific RTCS Hypertext Document Library.

The TOSZHTML target library, which is typically an SMS-managed PDSE, can be used *as is* in the RTCS Initiator started task PROC; however, this is *not* recommended for actual production configurations. No actual SMP/E target data set should be used in production on a target MVS image, since doing so may limit your ability to apply maintenance to the library. If this is what you want to do, proceed to the next step, and simply specify *this* data set (the SMP/E TOSZHTML target library) to the RTCS RIU (refer to Step 17) with the following SET statement in its SYSIN input stream: `SET ISA.POSZHTML = tlibprfx.TOSZHTML`. Specify this data set (the TOSZHTML target library) in the HTMLDOC DD statement of the RTCS Initiator PROC, which is usually named OSZINIT.

Instead of specifying the SMP/E TOSZHTML target library directly in the RTCS Initiator PROC OSZINIT, the recommended procedure is to allocate a new PDSE library specifically for the purpose of running RTCS products in production on a specific target MVS image, and then to copy the actual SMP/E TOSZHTML target library into this new "production" copy. The new, production RTCS Hypertext Document Library production copy should, like the original, be a PDSE library. It should be allocated as a *non*-SMS-managed PDSE (that is, no STORCLAS specified), but this is not required. However, the data set *must* be cataloged, because it will be dynamically allocated by the RTCS Generalized Server (OSZEXEC[*n*]) to address spaces that require it by DSNAME only. It is preferable that this data set be cataloged in the master catalog for the MVS image on which it is to be used, but it may be cataloged in an ICF catalog if you understand the consequences of such an action with respect to catalog and catalog volume availability during the IPL process, which is when RTCS and some RTCS-based products are normally brought up.

It is possible to share this library among multiple MVS images, regardless of whether it is SMS-managed or *non*-SMS-managed. BMC Software recommends that the RTCS Hypertext Document Library *not* be shared, but instead have one production copy allocated for exclusive use on each target MVS image and cataloged in that target MVS image's master catalog. If you want to share this data set among production MVS images, you *must* ensure that your configuration adheres to all documented IBM MVS requirements for sharing PDSEs.

In addition to specifying the DSNAME of this library in the HTMLDOC DD statement in the RTCS Initiator PROC, which is usually named OSZINIT (refer to Step 12), its DSNAME must be specified to the RTCS RIU (refer to Step 17) with a SET statement in its SYSIN input stream. For example:

```
SET ISA.POSZHTML = SYS1.RTCS.OSZHTML
```

To allocate and copy the target SMP/E RTCS Hypertext Document Library into a new, non-SMS-managed PDSE, you can use the following sample JCL:

```
//OSZINJ75 JOB (acct),'CREATE OSZHTML PDSE',
//             MSGLEVEL=(1,1),CLASS=A,TIME=1
//*
//*  acct      JOB statement accounting information
//*  zzzzzz    VOL SER on which to allocate .OSZHTML PDSE
//*  tlibprfx  DSN prefix of RTCS SMP/E target libraries
//*
//ALLOC   EXEC PGM=IEFBR14
//*
//*  Allocate a new non-SMS-managed cataloged PDSE for
//*  the RTCS HTML Doc Library. This data set may be
//*  SMS-managed, but the JCL will have to be changed
//*  to allocate an SMS-managed data set. There is no
//*  particular need for the library to be SMS-managed.
//*  This data set CAN be shared by more than one system
//*  if MVS sysplex PDSE sharing requirements are met.
//*
//OSZHTML   DD DSN=SYS1.RTCS.OSZHTML,
//             DISP=(NEW,CATLG),
//             UNIT=3390,VOL=SER=zzzzzz,
//             DSNTYPE=LIBRARY,SPACE=(CYL,(25,5,1)),
//             DSORG=PO,RECFM=VB,LRECL=260,BLKSIZE=6504
//*
//COPY    EXEC PGM=IEBCOPY,TIME=1,REGION=4M
//*
//* Copy SMP/E Target RTCS HTML Doc  LIBRARY
//* to the newly-allocated "production" PDSE.
//*
//SYSPRINT  DD SYSOUT=*
//TOSZHTML  DD DISP=SHR,DSN=tlibprfx.TOSZHTML
//OSZHTML   DD DISP=SHR,DSN=*.ALLOC.OSZHTML
//SYSIN     DD *
         COPY  INDD=((TOSZHTML,R)),OUTDD=OSZHTML
//*
```

Step 4 APF-authorize the RTCS Subsystem and Product Program Libraries.

Both the RTCS Subsystem Program Library and the RTCS Product Program Library must be APF-authorized. This can be accomplished on a temporary basis, which will not remain in effect across an IPL of the MVS image, by using the MVS SETPROG operator command.

Use the following command, for example, to add the RTCS Subsystem Program Library to the APF list if the library is, as recommended, *not* SMS-managed:

```
SETPROG APF,ADD,LIBRARY=SYS1.RTCS.OSZRTCS,VOLUME=xxxxxx
```

Use the following command, for example, to add the RTCS Subsystem Program Library to the APF list if the library is SMS-managed:

```
SETPROG APF,ADD,LIBRARY=SYS1.RTCS.OSZRTCS,SMS
```

Use the following command, for example, to add the RTCS Product Program Library to the APF list if the library is, as recommended, not SMS-managed:

```
SETPROG APF,ADD,LIBRARY=SYS1.RTCS.OSZLINK,VOLUME=zzzzzz
```

Use the following command, for example, to add the RTCS Product Program Library to the APF list if the library is SMS-managed:

```
SETPROG APF,ADD,LIBRARY=SYS1.RTCS.OSZLINK,SMS
```

To permanently APF-authorize these two libraries, you must add their DSNAMEs to the appropriate PROG*xx* member of SYS1.PARMLIB (or the appropriate MVS image-specific logical parameter library data set). Then, you must issue the SET PROG=*xx* MVS operator command, which causes MVS to rebuild the list of APF-authorized data sets in storage; or, you may simply IPL the MVS image.

Step 5 Allocate a VSAM LDS cluster for the RTCS System Registry.

A VSAM LDS cluster must be allocated for use by the RTCS Subsystem to back the primary RTCS registry partition using the MVS Data-In-Virtual (DIV) service. The System Registry VSAM LDS *must* be cataloged since it is a VSAM cluster, and because it will be dynamically allocated by the RTCS Allocator (OSZMOSYS) to the RTCS Subsystem address space by DSNAME only.

The VSAM LDS should be cataloged in the master catalog for the target MVS image, although this is not required. Unlike the RTCS Subsystem Program Library, the RTCS System Registry LDS *will* be permanently allocated to the RTCS Subsystem address space for the life of the IPL. Therefore, this VSAM cluster should not be cataloged in an ICF user catalog. But, if you do not want to or it is for some reason inappropriate to catalog this VSAM cluster in the master catalog, then catalog it in an ICF catalog in which other product data sets that are shared sysplex-wide are cataloged (for example, a high-level qualifier of SYS2 instead of SYS1).

Note: Consider allocating the RTCS System Registry VSAM LDS cluster with a name that includes the value of system variable *&SYSNAME*. For example, you could use `DSNAME=SYS1.`*sysname*`.REGISTRY`, where *sysname* is the *&SYSNAME* value for the target MVS image on which you intend to run RTCS. Then you would potentially be able to share and use the same RTCS Initiator started task PROC, which is usually named OSZINIT (refer to Step 12), on all MVS images particularly in a sysplex environment where cloned MVS images are being used with SYSRES volumes and the master catalog being shared.

To allocate a new, non-SMS-managed VSAM LDS to back the RTCS System Registry, you can use this sample JCL:

```
//OSZINJ80 JOB (acct),'ALLOC REGISTRY VLDSC',
//             MSGLEVEL=(1,1),CLASS=A,TIME=1
//*
//*  acct      JOB statement accounting information
//*  yyyyyy    VOL SER on which to allocate .REGISTRY
//*  sysname   System name (&SYSNAME) of this MVS image
//*
//CREATE  EXEC PGM=IEFBR14
//*
//* Allocate a new VSAM Linear Data Set
//* CLUSTER for the RTCS System Registry
//*
//REGISTRY  DD DISP=(NEW,CATLG),
//             VOL=SER=yyyyyy,UNIT=3390,
//             DSN=SYS1.sysname.REGISTRY,
//             RECORG=LS,AMP=AMORG,
//             SPACE=(4096,(24576,0))
//*
```

To calculate the amount of space expressed in terms of number of 4K blocks that you need to allocate for the VSAM LDS, start with 16,384 (which is equivalent to 64 megabytes, requiring 92 cylinders on a 3390), and then add 8 for each unique user ID that you ever expect to be defined to the ESM on that target MVS image that will have access to or use RTCS-based products. For example, for 1024 unique user IDs, the number of blocks should be 16,384+(8*1024) = 24,576. The eight 4K byte blocks are an estimate of the average amount of RTCS System Registry space required to store profile information for each unique user ID encountered.

Step 6 *(optional)* Allocate a new Product Authorization Table Library.

A Product Authorization Table Library may need to be allocated for use by the RTCS Generalized Server to authorize execution of licensed products on the processors in your configuration. If you already have BMC Software licensed products installed, then you might already have such a data set allocated. If so, then it can be used *as is*, and the product license keys (or "passwords") can be added to the product-specific tables in it using an existing facility, such as the TSO ISPF interface or the batch program. But, if you do not have a separate data set dedicated for this purpose, then you will probably need to allocate one for the use of RTCS-based products when executed by means of the RTCS Generalized Server program (OSZEXEC[*n*]). The Product Authorization Table Library *must* be cataloged, since it is dynamically allocated by OSZEXEC[*n*] to RTCS Generalized Server address spaces by DSNAME only.

The Product Authorization Table Library does not need to be cataloged in the master catalog for the target MVS image. However, it will be dynamically allocated by the RTCS Generalized Server in all instances when a licensed product is executed. This means that the data set should be cataloged in a high-availability catalog, such as the master catalog or an ICF catalog in which other product data sets that are shared sysplex-wide are cataloged. This may affect the high-level qualifier of the DSNAME of the data set that you will need to allocate.

The Product Authorization Table Library can be either an ordinary PDS or a PDSE. BMC Software recommends allocating it as a PDSE, but this is not required. If it is allocated as a PDSE, then you cannot place SMS program objects into the same data set (this is an SMS restriction). This library should be used exclusively for maintaining BMC Software licensed product authorization tables.

To allocate a new, non-SMS-managed Product Authorization Table Library as a PDSE, you can use this sample JCL:

```
//OSZINJ84 JOB (acct),'ALLOC REGISTRY VLDSC',
//             MSGLEVEL=(1,1),CLASS=A,TIME=1
//*
//*  acct      JOB statement accounting information
//*  yyyyyy    VOL SER on which to allocate .REGISTRY
//*
//PSWDUV  EXEC PGM=IEFBR14
//*
//* Allocate a New Product Authorization Table Library
//*
//SYSLIB    DD DSN=SYS2.RTCS.BMCPSWD,
//             DISP=(NEW,CATLG,DELETE),
//             SPACE=(CYL,(1,1,1)),DSORG=PO,
//             BLKSIZE=23476,RECFM=U,
//             UNIT=3390,VOL=SER=yyyyyy,
//             DSNTYPE=LIBRARY
//*
```

The product authorization tables (members) contained in this library are typically very small, and it is unlikely that you will need more than a cylinder of space for this data set. In fact, you may not need more than a few tracks for a reasonably large number of RTCS-based products. If you run out of space in this data set while creating or updating tables, you can use IEBCOPY to copy it to another new, larger data set, then delete the old data set and rename the new one to the name of the old one. This is possible because the RTCS Generalized Server does not retain an active allocation for this data set. It is allocated only when the General Server (OSZEXEC[*n*]) is executed and execution of a licensed BMC Software product is requested. It is then immediately dynamically deallocated.

Step 7 Ensure availability of sufficient common area data spaces.

The MAXCAD parameter in member IEASYS*nn* of SYS1.PARMLIB reserves the number of entries available for SCOPE=COMMON data spaces. RTCS uses two common area data spaces. It is unlikely that most installations will need to increase this value, unless many subsystems are using a large number of common area data spaces.

If there is any uncertainty about whether or not there is a sufficient number of available entries, increase the MAXCAD value and IPL the MVS image before attempting to start the RTCS Subsystem. It is possible to make this change in SYS1.PARMLIB and not have to wait for the MVS image to be IPLed; the increased number will be available the next time the system is IPLed. If an IPL is already scheduled or can be expected, increase the number of entries anyway, unless you have previously increased it to ensure that you would have ample entries available for products that may need them.

Step 8 Ensure availability of sufficient system linkage indexes (LXs)

The NSYSLX parameter in member IEASYS*xx* of SYS1.PARMLIB allows you to specify the number of LXs, in addition to those in the MVS system function table, that are to be reserved for system LXs. RTCS uses one system LX. It is unlikely that most installations will need to increase the number of system LXs, unless many subsystems that use a large number of system LXs are being started.

If there is any uncertainty about whether or not an MVS image has enough system LXs available, increase the NSYSLX value and re-IPL the MVS image before attempting to start the RTCS subsystem. It is usually possible to make this change in SYS1.PARMLIB and not have to wait for the MVS image to be IPLed; the increased number will be available the next time that the system is IPLed. If an IPL is already scheduled or can be expected, increase the number of entries anyway, unless you have previously increased it to ensure that you would have plenty of system LXs available for address spaces or products that use them.

Step 9 Add RTCS programs to the MVS PPT.

The RTCS Initiator and RTCS Subsystem, along with the RTCS Generalized Server in most instances, require that they be executed in a started task address space with certain MVS-assigned special attributes. These attributes are assigned by MVS as a consequence of having definitions for the job step program in the PPT. Before RTCS can be started, the MVS PPT must be updated to include these entries. You can do this by updating the SCHED*xx* member that is currently in effect, or by adding a new SCHED*yy* member to SYS1.PARMLIB or a logical parameter library data set and updating the appropriate IEASYS*nn* member to specify, in addition, this new SCHED*yy* member.

The required RTCS PPT additions are provided in member OSZSCHED of the SMP/E TOSZCNTL target library. The contents of member OSZSCHED can be merged into an existing SCHED*xx* member or you can copy OSZSCHED into SYS1.PARMLIB or a logical parameter library data set as a new member named SCHED*yy*. If you update your existing SCHED*xx* member, refresh the PPT by issuing the following MVS operator command:

```
SET  SCH=(xx)
```

If a new SCHED*yy* member is created by copying the sample OSZSCHED member provided into SYS1.PARMLIB, then you can refresh the PPT to have the additional entries in the new member become effective by issuing the following MVS operator command:

```
SET SCH=(xx,yy)
```

The SCH member suffixes specified in a SET command are not additive but replace any existing entries in storage. Hence, any current existing PPT member suffixes must be included in the list in addition to any new ones that you want to add.

Step 10 Select an MVS subsystem ID (SSID) for use by RTCS.

Select an available, four-character MVS SSID for the RTCS Subsystem control address space to use. **RTCS** is the recommended SSID, but any sequence of four characters may be selected. The SSID is used in the following two steps.

Step 11 *(optional)* Update the MVS Subsystem name table for RTCS.

To guarantee the availability of an SSID for exclusive use by the RTCS Subsystem, update the MVS Subsystem Name (SSN) Table. To do this, you must add the selected SSID to the IEFSSN*xx* member in SYS1.PARMLIB, or a logical parameter library data set that is currently in effect for the target MVS image.

For example, to add an entry for **RTCS**, add a statement such as the following to the appropriate IEFSSN*xx* member of SYS1.PARMLIB:

```
SUBSYS      SUBNAME(RTCS)
```

The MVS image can be IPLed to update the subsystem name table.

You can update the subsystem name table without having to IPL the target MVS image by using the following MVS SETSSI operator command:

```
SETSSI  ADD,SUBNAME=RTCS
```

Step 12 Create and customize the RTCS Initiator started task PROC.

A model RTCS Initiator started task PROC [OSZINIT] is exhibited in Appendix B, “Sample JCL and MVS Image Configuration Members.”

The RTCS Subsystem address space is *not* started by the operator, by the installation (as part of the MVS IPL process), or by an automation product. Instead, what needs to be started by one of the above entities in order to bring up RTCS is the RTCS Initiator. Once started, the RTCS Initiator performs certain MVS system checks and validates that the product data sets have been properly allocated. Then, the RTCS Initiator internally starts the RTCS Subsystem address space.

Thus, all started task PROC JCL customizations that are to affect the RTCS Subsystem address space must be made to the PROC used to start the RTCS Initiator, because that is where they are defined. The DSNAMEs allocated by the RTCS Initiator started task PROC are recorded internally by RTCS and subsequently used to dynamically allocate these same data sets in the RTCS Subsystem address space.

A model of an OSZINIT started task PROC is in Appendix B, “Sample JCL and MVS Image Configuration Members.”

12.A Select a name to be used for the RTCS Initiator started task PROC.

This is the name that will be used in an MVS START command that must be used (during IPL, for example) to start the RTCS Initiator address space under the Master (MSTR) Subsystem. The suggested name is **`OSZINIT`**, but this is not required.

12.B Copy member OSZ$INIT from the SMP/E TOSZCNTL target library to SYS1.PROCLIB as the member name you just selected.

It is not required that the PROC be created in SYS1.PROCLIB; instead of SYS1.PROCLIB, any started task procedure library defined in the IEFPDSI DD statement concatenation in the Master Scheduler JCL (member MSTJCL*xx* of SYS1.PARMLIB) may be used. The majority of installations use an unmodified MSTJCL00, and so *must* use SYS1.PROCLIB specifically for those started tasks that are started under the Master Subsystem.

12.C Modify the STEPLIB DD statement in the new PROC to specify the DSNAME of the RTCS Subsystem Program Library (for example, the PDSE created in Step 1).

12.D Modify the TASKLIB DD statement in the new PROC to specify the DSNAME of the RTCS Subsystem Program Library (for example, the PDSE created in Step 2).

12.E Modify the HTMDOC DD statement in the new PROC to specify the DSNAME of the RTCS Hypertext Document Library (for example, the PDSE created in Step 3).

12.F Modify the REGISTRY DD statement in the new PROC to specify the DSNAME of the RTCS System Registry VSAM LDS cluster previously allocated in Step 5 (or the DSNAME of a previously allocated LDS cluster).

12.G Modify the BMCPSWD DD statement in the new PROC to specify the DSNAME of the Product Authorization Table Library previously allocated in Step 6 (or the DSNAME of an existing BMC Software Security Facility Product Authorization Table Library).

12.H Modify the SSID parameter on the PROC statement to specify the SSID that was selected for use by the RTCS Subsystem in Step 10.

12.I Ensure that the RTCS parameter on the PROC statement specifies the PROC name of the RTCS Subsystem Started Task PROC, which will be selected and created in Step 13.

12.J Ensure that the EXEC parameter on the PROC statement specifies the PROC name of the RTCS Generalized Server started task PROC, which will be selected and created in Step 14.

Step 13 Create the RTCS Subsystem started task PROC.

A model RTCS Subsystem started task PROC [OSZRTCS] is exhibited in Appendix B, “Sample JCL and MVS Image Configuration Members.” Note that, other than selecting the actual PROC name to be used, and perhaps some additions to the EXEC statement in order to adhere to local conventions and requirements, such as adding an ACCT parameter, there are *no* changes that should be made to the model PROC. In particular, there should be no DD statements present in the RTCS Subsystem Started Task PROC.

13.A Select a name to be used for the RTCS Subsystem started task PROC.

This is the name that will be used in an MVS START command that will be issued internally by the RTCS Initiator (OSZSIRIS) to start the RTCS Subsystem address space under the Master (MSTR) Subsystem. The suggested name is **OSZRTCS**, but this is not required.

13.B Copy member **OSZ$RTCS** from the SMP/E **TOSZCNTL** target library to **SYS1.PROCLIB** as the member name selected above.

It is not required that the PROC be created in SYS1.PROCLIB; any started task procedure library defined in the IEFPDSI DD statement concatenation in the Master Scheduler JCL (member MSTJCL*xx* of SYS1.PARMLIB) may be used. The majority of installations use an unmodified MSTJCL00, and *must* use SYS1.PROCLIB specifically for those started tasks that are started under the Master Subsystem.

13.C Be sure that you specified the correct name of the RTCS Subsystem started task PROC in the RTCS parameter in the RTCS Initiator PROC that was set up in Step 12.

Step 14 Create the RTCS Generalized Server started task PROCs.

A model RTCS Generalized Server started task PROC [OSZEXEC] is exhibited in Appendix B, “Sample JCL and MVS Image Configuration Members.”

14.A Select a name to be used for the RTCS Generalized Server started task procedures.

The same name must be used for both PROCs, and it must exist by the same name, in up to two separate procedure libraries, potentially. This is the name that will be used in START commands that will be issued internally by RTCS. The suggested name is **OSZEXEC**, but this is not required.

14.B Copy member **OSZ$EXEC** from the SMP/E **TOSZCNTL** target library to **SYS1.PROCLIB** as the member name you selected above.

It is not required that the PROC be created in SYS1.PROCLIB; instead of SYS1.PROCLIB, any started task procedure library defined in the IEFPDSI DD statement concatenation in the Master Scheduler JCL (member MSTJCL*xx* of SYS1.PARMLIB) may be used. The majority of installations use an unmodified MSTJCL00, and so *must* use SYS1.PROCLIB specifically for those started tasks that are started under the Master Subsystem. It is necessary for this PROC to exist in a Master Subsystem PROCLIB, since RTCS Generalized Server address spaces can be started by RTCS with a specification of SUB=MSTR.

14.C Copy member **OSZ$EXEC** from the SMP/E **TOSZCNTL** target library to any valid JES-managed started task PROCLIB as the same member name that was selected in step 14.A.

It is necessary for this PROC to exist in a JES-managed PROCLIB, since RTCS Generalized Server address spaces can be, and usually are, started by RTCS *without* specifying SUB=MSTR. In this case, the RTCS Generalized Server runs under the primary JES, and its PROC is expanded using the usual JES-managed started task PROCLIBs.

14.D Ensure you specified the correct (common) name of the RTCS Generalized Server PROCs in the EXEC parameter in the RTCS Initiator started task PROC set up in Step 12.

Step 15 Establish ESM user IDs for RTCS address spaces.

Appropriately-defined, valid ESM user IDs must be assigned to the three RTCS started task address spaces by MVS and the ESM when they are started. The procedure to accomplish this depends on the specific ESM that is executing on the MVS image. In addition, these user IDs must be given access to the production product libraries that will be accessed by the address spaces running under the assigned user IDs.

This is an area of extreme variability from one installation to another, even those using the same ESM, because of the complex and sometimes subtle interaction between the data set naming conventions in place and existing profiles, permits, or rules. Thus, we can only offer examples for the security administrator or system programmer. The modifications required to the following sample ESM commands and administrative actions may be extensive for your installation.

RACF

```
/* Define data set profiles for the PDSE libraries    */
/* and System Registry.  We define them initially     */
/* with UACC(ALTER) so that the current ID does       */
/* not have to have the SPECIAL attribute to issue    */
/* the PERMIT commands that immediately follow. We    */
/* will alter the UACC in the data set profiles       */
/* after ALTER access by the current user is set.     */

   ADDSD   'SYS1.RTCS.OSZRTCS' -
           GENERIC UACC(ALTER)
   ADDSD   'SYS1.RTCS.OSZLINK' -
           GENERIC UACC(ALTER)
   ADDSD   'SYS1.RTCS.OSZHTML' -
           GENERIC UACC(ALTER)
   ADDSD   'SYS1.RTCS.REGISTRY' -
           GENERIC UACC(ALTER)

/* Ensure that the system programmer executing the    */
/* RTCS configuration JOBs has ALTER access to the    */
/* RTCS Subsystem and Product Libraries, the          */
/* Hypertext Document Library, and the RTCS System    */
/* Registry LDS.  The absence of OWNER(id) does not   */
/* grant such access unless the "*" UserID has the    */
/* SPECIAL attribute, anyway; the point of all of     */
/* this convolution is to obviate the requirement     */
/* that a system programmer running these JOBs have   */
/* the SPECIAL attribute in the first place.          */

   PERMIT  'SYS1.RTCS.OSZRTCS' -
           GENERIC ID(*)    ACCESS(ALTER)
   PERMIT  'SYS1.RTCS.OSZLINK' -
           GENERIC ID(*)    ACCESS(ALTER)
   PERMIT  'SYS1.RTCS.OSZHTML' -
           GENERIC ID(*)    ACCESS(ALTER)
   PERMIT  'SYS1.RTCS.REGISTRY' -
           GENERIC ID(*)    ACCESS(ALTER)

/* Update data set profiles for the PDSE libraries    */
/* and System Registry.  We set the minimum UACC      */
/* needed by a typical, non-strict, customer.         */

   ALTDSD  'SYS1.RTCS.OSZRTCS' -
           GENERIC UACC(NONE)
   ALTDSD  'SYS1.RTCS.OSZLINK' -
           GENERIC UACC(READ)
   ALTDSD  'SYS1.RTCS.OSZHTML' -
           GENERIC UACC(READ)
   ALTDSD  'SYS1.RTCS.REGISTRY' -
           GENERIC UACC(NONE)
```

```
/* Define an STC user ID for use by RTCS Initiator     */
/* and RTCS Subsystem.  Take care to customize         */
/* this properly for your RACF configuration.          */

   ADDUSER OSZRTCS -
           NAME('RTCS Subsystem') -
           NOPASSWORD    -
           DFLTGRP(SYS1) -
           UACC(READ)    -
           LANGUAGE(PRIMARY(ENU)) -
           OMVS( UID(10010) HOME('/home') -
                 PROGRAM('/bin/sh') )

/* Define an STC user ID for use by RTCS Generalized  */
/* Server (products).  Take care to customize this    */
/* properly for your RACF configuration.              */

   ADDUSER OSZEXEC -
           NAME('RTCS General Server') -
           NOPASSWORD    -
           DFLTGRP(SYS1) -
           UACC(READ)    -
           LANGUAGE(PRIMARY(ENU)) -
           OMVS( UID(10010) HOME('/home') -
                 PROGRAM('/bin/sh') )

/* Connect these two new STC user IDs to the STC      */
/* PROCs used by them. We are assuming here that      */
/*    SETROPTS GENERIC(STARTED)                       */
/* is in effect already.                              */

   RDEFINE STARTED OSZINIT.* STDATA( USER(OSZRTCS) )
   RDEFINE STARTED OSZRTCS.* STDATA( USER(OSZRTCS) )
   RDEFINE STARTED OSZEXEC.* STDATA( USER(OSZEXEC) )
   SETROPTS RACLIST(STARTED) REFRESH

/* Allow READ access by the RTCS Initiator to RTCS    */
/* Subsystem Library, the RTCS Product Library,       */
/* and Hypertext Document Library PDSEs.              */

   PERMIT  'SYS1.RTCS.OSZRTCS' -
           GENERIC ID(OSZINIT) ACCESS(READ)
   PERMIT  'SYS1.RTCS.OSZLINK' -
           GENERIC ID(OSZINIT) ACCESS(READ)
   PERMIT  'SYS1.RTCS.OSZHTML' -
           GENERIC ID(OSZINIT) ACCESS(READ)

/* Allow ALTER access by RTCS Initiator to the        */
/* RTCS System Registry.                              */

   PERMIT  'SYS1.RTCS.REGISTRY' -
           GENERIC ID(OSZINIT) ACCESS(ALTER)
```

```
/* Allow READ access by the RTCS Subsystem to RTCS    */
/* Subsystem Library, the RTCS Product Library,       */
/* and Hypertext Document Library PDSEs.              */

   PERMIT  'SYS1.RTCS.OSZRTCS' -
           GENERIC ID(OSZRTCS) ACCESS(READ)
   PERMIT  'SYS1.RTCS.OSZLINK' -
           GENERIC ID(OSZRTCS) ACCESS(READ)
   PERMIT  'SYS1.RTCS.OSZHTML' -
           GENERIC ID(OSZRTCS) ACCESS(READ)

/*                                                    */
/* Allow ALTER access by RTCS Subsystem to the        */
/* RTCS System Registry.                              */

   PERMIT  'SYS1.RTCS.REGISTRY' -
           GENERIC ID(OSZRTCS) ACCESS(ALTER)

/* Allow READ access by RTCS Client Product           */
/* Generalized Server STCs to the RTCS Product        */
/* Library and Hypertext Document Library PDSEs.      */

   PERMIT  'SYS1.RTCS.OSZLINK' -
           GENERIC ID(OSZEXEC) ACCESS(READ)
   PERMIT  'SYS1.RTCS.OSZHTML' -
           GENERIC ID(OSZEXEC) ACCESS(READ)

/* Allow SYSLOG access by RTCS Client Product         */
/* Generalized Server Started Tasks.                  */

   RDEFINE JESSPOOL -
           &RACLNDE.+MASTER+.SYSLOG.*.*.? UACC(READ)
   PERMIT  &RACLNDE.+MASTER+.SYSLOG.*.*.? -
           CLASS(JESSPOOL) ID(OSZEXEC) ACCESS(READ)
```

Checkpoint

At a glance, this is what you have done so far in the configuration process that you conceivably might need to do only once per sysplex, depending upon what level of sharing is taking place, particularly of data sets such as PROCLIBs and the ESM database:

- made production copies of the three run-time RTCS product libraries
- customized the RTCS Initiator started task PROC in SYS1.PROCLIB
- created the RTCS Subsystem started task PROC in SYS1.PROCLIB
- created the RTCS Generalized Server started task PROCs in both a JES-managed started task PROCLIB as well as in SYS1.PROCLIB
- added the RTCS program libraries to the APF list via SETPROG APF
- updated the MVS PPT with SET SCH

- defined the started task ESM user IDs and permitted data set access
- allocated a new Product Authorization Table Library, if necessary

If you are not able to share PDSE libraries among the various target MVS images, and you are not sharing PROCLIBs and ESM databases, then you may need to do all of the above things for each target MVS image.

This is what you have done so far in the MVS image configuration process that you will always need to do once per unique MVS image, regardless of what level of sharing is taking place, even in a sysplex:

- allocated a dedicated, image-specific RTCS Registry VSAM LDS

At this point, everything has been done that is necessary in order to start RTCS on the target MVS image the very first time. There are still things that need to be done after RTCS is started and successfully initialized. For example, the RTCS System Registry VSAM LDS can be initialized only using the RTCS RIU, which requires RTCS to be available in order to execute.

Step 16 Bring up RTCS for the very first time, by issuing the following MVS operator command:

```
S OSZINIT,SUB=MSTR
```

This command brings up RTCS, and starts the RTCS Initiator, which will internally start the RTCS Subsystem.

Ensure that the RTCS Initiator address space started successfully, and that the internally started the RTCS Subsystem address space also started successfully. The RTCS Subsystem *must* be successfully initialized before you can proceed with the next configuration step.

Step 17 Initialize the RTCS System Registry by importing some Registry IMPORT files that are distributed in the SMP/E TOSZRXML target library.

You may use the following sample JCL:

```
//OSZINJ90 JOB (acct),'IMPORT REGISTRY FILE',CLASS=A,
//             MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=1
//*
//*  acct      JOB statement accounting information
//*  tlibprfx  DSN prefix of RTCS SMP/E target libraries
//*
//*******************************************************
//*
//IMPORT  EXEC PGM=OSZEXEC8,TIME=1,REGION=4M,
//             PARM='P=OSZRGIMP,C=BMCPROD'
//SYSPRINT  DD SYSOUT=*
//*
//*  IMPORT XML-FORMAT RTCS REGISTRY IMPORT FILES
//*
//OSZRXML   DD DISP=SHR,DSN=tlibprfx.TOSZRXML
//*
//SYSIN     DD *
SET ISA.SSID     = RTCS
SET ISA.POSZHTML = SYS1.RTCS.OSZHTML
SET ISA.POSZLINK = SYS1.RTCS.OSZLINK
SET ISA.TLA      = OSZ
SET ISA.NAMEKEY  = RTCS
SET ISA.VENDKEY  = BMC
SET ISA.PRODUCT  = "Runtime Component System"
IMPORT OSZRXML(OSZRTYPC)
IMPORT OSZRXML(OSZRSECR)
//*
```

Prior to importing the OSZRTYPC and ORZRSECR Registry IMPORT files (members of the TOSZRXML SMP/E target library), you must set the value of the *ISA.POSZLINK* variable to the DSNAME of the SMP/E target RTCS Product Program Library (or its clone to be used “in production”) in the RTCS RIU SYSIN input stream. This must be done using a SET statement, as illustrated in the preceding sample JCL or in the following example:

```
SET ISA.POSZLINK = SYS1.RTCS.OSZLINK
```

Prior to importing the OSZRTYPC and OSZRSECR Registry IMPORT files (members of the TOSZRXML SMP/E target library), you must set the value of the *ISA.POSZHTML* variable to the DSNAME of the SMP/E target RTCS Hypertext Document Library (or its clone to be used “in production”) in the RTCS Registry Import Utility SYSIN input stream. This must be done using a SET statement, as illustrated in the preceding sample JCL or in the following example:

```
SET ISA.POSZHTML = SYS1.RTCS.OSZHTML
```

The DSNAME of the SMP/E target RTCS XML document library, indicated in the preceding sample JCL, as `tlibprfx.TOSZRXML`, must be specified on an OSZRXML DD statement in the JCL of the step in which you execute the RTCS RIU.

Step 18 Refresh global security parameters.

After the RTCS System Registry has been updated, refresh the Security Manager's global security parameters, which were imported from the XML Registry IMPORT file (member) OSZRSECR, using the following MVS operator command:

```
F RTCS,REFRESH,SECURITY
```

In the above command, *RTCS* is the SSID selected for the RTCS Subsystem.

Step 19 Update MVS automatic commands to automatically start RTCS.

Modify the appropriate COMMND*xx* member in SYS1.PARMLIB to add a START command to start the RTCS Initiator address space automatically. Find the appropriate COMMND*xx* member used by the MVS image and add a line in the appropriate position that specifies:

```
COM='S OSZINIT,SUB=MSTR'
```

Note: The RTCS Subsystem can, and in most instances should, be started before the primary JES.

Chapter 4 Installing and Configuring RTCS-based Products

This chapter explains how to install and configure RTCS-based products after they have been installed into the designated target libraries.

This chapter contains the following sections:

Installing RTCS-Based Products

RTCS-based products must be installed into the same BMC Software SMP/E global, distribution, and target zones in which the RTCS FUNCTION SYSMODs are installed (or are being simultaneously installed). This ensures that all SMP/E-maintained elements for both RTCS and RTCS-based products are applied to the same set of target libraries designated for this purpose.

You may install RTCS-based products at the same time RTCS is installed, or at any time after RTCS has already been installed. The default is for all available RTCS-based products to be installed simultaneously with RTCS. If you select a subset of the RTCS-based products that are available for installation, one or more components distributed as part of a product which you did not select that is used by one of the RTCS-based products which you did select for installation. In this case, all products that share components will automatically be selected for installation, even though you did not explicitly select it. Such components do not require a license, because they are automatically licensed for your use if you have a license to use *any* RTCS-based product.

Note: For detailed information about installation system requirements and using the Desktop Installation Assistant to install RTCS and any RTCS-based products, see the *Desktop Installation Assistant Getting Started*.

Configuring RTCS-Based Products

After a licensed RTCS-based product has been installed, and the designated target libraries have been populated with the product's elements, the product can be configured for execution on any processor for which you have a valid CPU authorization password and CPUID.

When RTCS is installed and a target MVS image is configured to run RTCS, no further changes are needed. You are ready to execute any RTCS-based product after the product has been installed into the target libraries and then copied to your production product libraries (if necessary) and configured.

To execute any RTCS-based product, import its XML Registry IMPORT files into the RTCS System Registry on each MVS image on which you intend to use that product.

If necessary, you must also update the Product Authorization Table Library that is being used for the RTCS-based licensed products to indicate the processors on which you are licensed to execute the products.

Using the Desktop Installation Assistant to Configure an RTCS-based Product

If you use the Desktop Installation Assistant to install an RTCS-based product, you have the option of using it to configure the product that you have just installed. You can also start the Desktop Installation Assistant and use it to configure any previously-installed products that are contained on the accompanying distribution media.

The Desktop Installation Assistant displays one or more panels with configuration options for the product on the target MVS image. The values that you specify for the configuration options will be substituted into the System Registry data structures that are built by the Registry Import Utility for the product. After the necessary XML Registry IMPORT files have been imported using the Registry Import Utility for a specific target MVS image, that product is ready to use on that system. You can start the product by using the Generalized Server, the RTCS START,*product* command, or by using any method described in the product's documentation.

The Desktop Installation Assistant also prompts you to update your Product Authorization Table Library with the CPU authorization password and CPUID for the RTCS-based product that you are configuring.

Manually Configuring an RTCS-based Product

If you do not use the Desktop Installation Assistant to configure your previously-installed RTCS-based products, you can manually create and submit the necessary Registry Import Utility job to do so. Because the RTCS System Registry is not shared among different MVS images, this action must be performed on each target MVS image on which you intend to run the RTCS-based products.

Figure 4-1 shows an example of the basic Registry Import Utility job stream that you need to manually construct in order to configure an RTCS-based product.

Figure 4-1 Registry Import Utility Job Stream Example

```
//OSZINJ90 JOB (acct),'IMPORT REGISTRY FILE',CLASS=A,
//             MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=1
//*
//*  acct      JOB statement accounting information
//*  tlibprfx  DSN prefix of RTCS SMP/E target libraries
//*
//*******************************************************
//*
//IMPORT  EXEC PGM=OSZEXEC8,TIME=1,REGION=4M,
//             PARM='P=OSZRGIMP,C=BMCPROD'
//SYSPRINT  DD SYSOUT=*
//*
//*  IMPORT XML-FORMAT RTCS REGISTRY IMPORT FILES
//*
//OSZRXML   DD DISP=SHR,DSN=tlibprfx.TOSZRXML
//*
//SYSIN     DD *
SET ISA.SSID     = RTCS
SET ISA.POSZHTML = SYS1.RTCS.OSZHTML
SET ISA.POSZLINK = SYS1.RTCS.OSZLINK
SET ISA.TLA      = tla
SET ISA.NAMEKEY  = namekey
SET ISA.VENDKEY  = vendkey
SET ISA.PRODUCT  = "product name"
SET tla.variable = value
IMPORT OSZRXML(tlamembr)
//*
```

Prior to importing any RTCS-based product Registry IMPORT files (members of the TOSZRXML SMP/E target library), you must set the value of the *ISA.POSZLINK* variable to the DSNAME of the SMP/E target RTCS Product Program Library, and the value of the *ISA.POSZHTML* variable to the DSNAME of the SMP/E target RTCS Hypertext Document Library (or its copy) in the RTCS Registry Import Utility SYSIN input stream. This must be done using a SET statement, as illustrated in the sample JCL in Figure 4-1 or in the following example:

```
SET ISA.POSZLINK = SYS1.RTCS.OSZLINK
SET ISA.POSZHTML = SYS1.RTCS.OSZHTML
```

The DSNAME of the SMP/E target RTCS XML Document Library, which is indicated in Figure 4-1 on page 4-4 as tlibprfx.TOSZRXML, must be specified on an OSZRXML DD statement in the JCL for the step executing the RTCS Registry Import Utility.

The names of the XML Registry IMPORT members that need to be imported are listed in the element named *prd*MNFST of the SMP/E TOSZRXML target library.

The values of the other variable elements in the model job stream shown in Figure 4-1 on page 4-4, *ISA.TLA*, *ISA.NAMEKEY*, *ISA.VENDKEY*, and *ISA.PRODUCT*, are listed in the element named *prd*$$RIU of the SMP/E TOSZCNTL target library.

For example, Figure 4-2 on page 4-6 is the actual Registry Import Utility job that would be constructed by the Desktop Installation Assistant for the System Explorer for z/OS product, which has a three-character identifier of *ZSE*. There are three ZSE product-specific Registry IMPORT variables and only one XML Registry IMPORT file (member). The default values are shown for each of these product-specific variables. Each of these variables is defined in the XML Registry IMPORT file (member) with its default value. The Desktop Installation Assistant reads the XML Registry IMPORT file (member) to obtain the data necessary to present the panel that prompts you for this information.

Figure 4-2 Registry Import Utility JOB for *System Explorer for z/OS*

```
//OSZINJ90 JOB (acct),'IMPORT REGISTRY FILE',CLASS=A,
//             MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=1
//*
//*  acct      JOB statement accounting information
//*  tlibprfx  DSN prefix of RTCS SMP/E target libraries
//*
//*******************************************************
//*
//IMPORT  EXEC PGM=OSZEXEC8,TIME=1,REGION=4M,
//             PARM='P=OSZRGIMP,C=BMCPROD'
//SYSPRINT  DD SYSOUT=*
//*
//*  IMPORT XML-FORMAT RTCS REGISTRY IMPORT FILES
//*
//OSZRXML   DD DISP=SHR,DSN=tlibprfx.TOSZRXML
//*
//SYSIN     DD *
SET ISA.SSID     = RTCS
SET ISA.POSZHTML = SYS1.RTCS.OSZHTML
SET ISA.POSZLINK = SYS1.RTCS.OSZLINK
SET ISA.TLA      = ZSE
SET ISA.NAMEKEY  = SYSEXPLR
SET ISA.VENDKEY  = BMC
SET ISA.PRODUCT  = "System Explorer"
SET ZSE.TCP.PORTNUMBER=4080
SET ZSE.TCP.TIMEOUT=300
SET ZSE.ASID=ZSEC
IMPORT OSZRXML(ZSERHTTP)
//*
```

After the Registry Import Utility has been executed on an MVS image to define the data structures for executing that product, you are able to start the product or execute some of its features and functions. Some RTCS-based products have additional, optional configuration features, which are described in the product's documentation.

You might also need to manually create and submit a job to execute the Product Authorization Table Library Update utility (OSZPATLU) to update the specific table for the product. For additional information on this utility, refer to Appendix A, "BMC Software Product Authorization". If the Product Authorization Table has already been updated to indicate the licensed processors for a product (for example, by using another facility or during configuration of the same product on another target MVS image), then it is not necessary to update it again.

Chapter 5 RTCS Administration

This chapter contains information about various administrative functions that a site might need to perform to ensure continued availability and proper functioning of the RTCS Subsystem.

This chapter contains the following sections:

Introduction

Continuous system operation (CSO) refers to a characteristic of software intended to be continuously available without interruption or downtime.

The two most significant issues with respect to the ongoing maintenance and administration of an RTCS Subsystem and associated products on an MVS image are the CSO requirements support and management of the System Registry. This chapter provides information about:

- CSO
- software maintenance
- RTCS System Registry
- automatic RTCS Subsystem startup
- interfacing RTCS with automation products
- configuration changes
- operator commands supported by RTCS.

Continuous System Operation

After the RTCS Subsystem has been started, it never shuts downs for any reason once the RTCS kernel has been loaded and the RTCS Subsystem has been initialized. Any attempt to stop or cancel the RTCS Subsystem address space will be suppressed.

The RTCS Subsystem provides a facility for updating software and configuration information that enables the RTCS kernel and any loaded dynamic link library (DLL) packages to be refreshed dynamically without disruption to ongoing operation. In addition, changes that are made to the RTCS System Registry are immediately effective, in most instances. In those cases where changes to RTCS System Registry configuration parameters are not immediately effective, a facility is provided to cause them to be refreshed when desired.

Even SMP/E-based maintenance of program object elements in the RTCS Subsystem Program Library cannot interfere with ongoing RTCS kernel or subsystem functions because the RTCS Subsystem address space does not retain an allocation for the target or production library after the RTCS kernel is initialized. RTCS dynamically deallocates the data set, permitting it to be updated or even reallocated. When the library is needed again (for example, to refresh the RTCS kernel after application of maintenance), it will be dynamically allocated.

Note: This process applies regardless of whether the SMP/E TOSZRTCS target library is being used directly, or a copy of that library has been made for production use on a particular MVS image.

The RTCS kernel and DLL refresh facility will dynamically allocate the original program libraries allocated by the RTCS Initiator started task PROC (more accurately, libraries with the original DSNAMEs will be dynamically allocated). This means that if these libraries are not the actual SMP/E target libraries, any updated elements (or simply the entire library) must be copied to the original library that is being used in production. Since these libraries are either dynamically deallocated, or can be freed by stopping any product Generalized Server address spaces, updating the production libraries for this purpose is proper, and conventional program library utilities can safely be used to perform this function. In general, because of the way RTCS-based products are designed, it is not necessary to shut down Generalized Server address spaces prior to performing maintenance using SMP/E or copying SMP/E target libraries to MVS image production copies.

Once RTCS has been initialized, the DSNAMEs of the product data sets that are used in production should remain unchanged. Do *not* rename them. Furthermore, in order to refresh or update the elements in the product libraries, you must either be running out of the SMP/E-maintained target libraries directly, or else you must copy the new or updated elements from the SMP/E target libraries to the libraries used in production on each MVS image.

Restrictions

The kernel or DLL elements that you intend to refresh for maintenance or upgrade must have a date and time stamp which is more current than the code which is being refreshed. The RTCS kernel and DLL refresh facility will check to ensure that maintenance has actually been applied and that the affected contents of the library carry a more recent date in order to prevent regression to a back-level version of the code.

RTCS System Registry

The RTCS System Registry is backed by a VSAM linear data set (LDS). All changes that are made to the contents of the System Registry are written to the LDS by the MVS Data in Virtual (DIV) service. Accordingly, each MVS image requires a dedicated VSAM LDS to back the RTCS System Registry for that MVS image.

Since the System Registry is a read/write data set, updated during normal system and product operation, it should be treated as an important system data set. You should consider making a backup of the volume on which the RTCS System Registry resides, but you will normally not be able to make a backup of the VSAM LDS using data set oriented utilities because the RTCS Subsystem keeps it OPEN permanently.

The System Registry contains RTCS and product configuration information, which you can think of as typical PARMLIB-like data. It also serves as the repository for data placed there by RTCS-based products such as end user profile information. In practice, none of this information is critical in the sense that, if lost, it would be difficult to reproduce. RTCS and product configuration information in the System Registry can easily be recreated by executing the Registry Import Utility (RIU) using the same input specifications that were used originally. End user profile information is preferences-oriented, and so nothing of irreplaceable value would be lost.

In most cases, if the DASD volume on which the System Registry VSAM LDS for an MVS image was allocated is lost, it could be simpler and quicker to allocate another VSAM LDS to back the System Registry for that image and reinitialize it using the RTCS RIU. If there were a DASD volume-level backup available, then the VSAM LDS for the System Registry could be restored.

These considerations should affect your choice of the DASD volume on which to allocate the VSAM LDS for the System Registry. If you do not have procedures in place to backup the DASD volume on a regular basis, you should be prepared to reallocate a VSAM LDS and reinitialize it using the RIU using the original input (which you should therefore retain). If you have made any customizations to the RTCS configuration in the System Registry by using the RIU, then any such jobs or RIU input should also be retained for such use. If the VSAM LDS that is used to back the System Registry on an MVS image, or even perhaps its catalog, becomes damaged or is unavailable for allocation for some reason (such as the DASD volume being offline) when the MVS image is IPLed and the RTCS Initiator is started, then RTCS Subsystem initialization will not be able to complete successfully. This means that RTCS-dependent products will not be able to execute. At that point, it will be necessary to correct the problem or restore the VSAM LDS and then start RTCS, or else allocate a new VSAM LDS, start RTCS, and initialize the new System Registry.

Starting RTCS

The RTCS Subsystem must be available in order for RTCS-based and RTCS-dependent products to execute. If the RTCS kernel has not been loaded and initialized, products which use RTCS services will usually terminate abnormally (ABEND); this will typically occur very early during the execution of the product, usually as part of its initialization. Products which use the RTCS Generalized Server (program OSZEXEC[n]) to obtain control in the proper execution key will ABEND immediately, even before any product code is loaded. This action is intended to alert you to the fact that RTCS facilities are not available on the involved MVS image.

Therefore, RTCS should be started as soon as possible during the IPL of an MVS image. Since RTCS executes only under the control of the Master (MSTR) Subsystem, RTCS should be started concurrently with the primary JES, or even prior to it. RTCS does not use any JES services.

RTCS should be started after the external security manager (ESM) has been started and successfully initialized. For RACF, this will always be the case by the time the RTCS Initiator is started, since RACF is initialized before any automatic START commands from PARMLIB are processed during IPL. While it is not a strictly enforced requirement that RTCS be initialized only after ESM initialization, it may be the case that certain products will not function properly unless you have made certain that this is the case. If you have a valid reason to start RTCS prior to ESM initialization, then you should arrange for the ESM or your automation facility to issue the following command *after* the ESM has completed initialization:

```
F RTCS,REFRESH,SECURITY
```

where *RTCS* is the SSID of the RTCS Subsystem.

To start RTCS, use the MVS START command to initiate the RTCS Initiator address space. As a result, the RTCS Initiator internally starts the RTCS Subsystem address space after determining and validating the DSNAMEs of the product data sets to be used by the RTCS Subsystem and the Generalized Server address spaces. The actual START command depends on the member name of the RTCS Initiator Started Task PROC, which is usually located in SYS1.PROCLIB, that was selected during MVS image configuration. The default name used by the Desktop Installation Assistant is OSZINIT.

If OSZINIT was selected, then the MVS START command used to start the RTCS Initiator address space should be:

```
S OSZINIT,SUB=MSTR
```

assuming that the OSZINIT PROC JCL has been properly customized, either by the installation or by the Desktop Installation Assistant, to specify the correct DSNAMEs to be used by RTCS.

The RTCS Initiator will internally start the RTCS Subsystem, which will then issue a series of messages to SYSLOG. Figure 5-1 provides an example of the subset of messages issued:

Figure 5-1 RTCS Subsystem Messages

```
OSZ0050I RTCS ALLOCATOR INITIALIZATION IN PROGRESS
OSZ0001I RTCS SUBSYSTEM INITIALIZATION IN PROGRESS
OSZ0005I RTCS SUBSYSTEM PC SERVICES NOW AVAILABLE
OSZ0029I OBJECT MANAGEMENT SERVICES NOW AVAILABLE
OSZ0079I RTCS SYSTEM REGISTRY IS NOW AVAILABLE
OSZ0022I RTCS MEMORY REGISTRY IS NOW AVAILABLE
OSZ0030I RTCS SUBSYSTEM IS NOW AVAILABLE
OSZ0039I  CONSOLE INTERFACE IS NOW AVAILABLE
```

You can configure an automation package that will automatically start RTCS-dependent products when the following message is issued:

```
OSZ0030I RTCS SUBSYSTEM IS NOW AVAILABLE
```

The following example is an MVS operator command that might automatically be triggered when the OSZ0030I message is issued:

```
F RTCS,START,ZSE
```

This command instructs the RTCS Subsystem to internally start the System Explorer for z/OS (whose TLA is *ZSE*) server address space.

Operator Commands Overview

The RTCS Subsystem address space and Generalized Server address spaces support a limited number of operator commands. Each is entered by using the MVS MODIFY (usually abbreviated as just the letter F) command as follows:

F *cscbName*,*command*,*operand*-1[... ,*operand*-n]

In the preceding example, *cscbName* is the name of the Command Scheduling Control Block (CSCB) that was built by MVS for the address space to which the command is being directed. For the RTCS Subsystem address space, the CSCB name is the RTCS MVS Subsystem ID (SSID). For any Generalized Server address space started by RTCS as a consequence of using the RTCS START,*productcode* command, the CSCB name will be the three-character identifier (for example, *ZSE*). For a batch job, the CSCB name is the job name. For any other started task that was not started by RTCS, the CSCB name is determined by MVS; however, it is usually the name specified after the period following the started task PROC name, if any, otherwise it will be the same as the PROC name.

Figure 5-2 provides a sample subset of the operator messages that are issued when the Generalized Server begins execution. The CSCB name of the Generalized Server address space is *ZSE*.

Figure 5-2 Generalized Server Messages

```
F RTCS,START,ZSE
START  OSZEXEC.ZSE,KEY=4,P=ZSE9INIT,C=BMCPROD
OSZ0031I COMMAND WAS PROCESSED
$HASP100 OSZEXEC  ON STCINRDR
$HASP373 OSZEXEC  STARTED
IEF403I OSZEXEC - STARTED - TIME=08.45.47
OSZ0104I GENERAL SERVER IS STARTING 026
 IN OSZEXEC  (ASID=X'0092',CSCB=ZSE     )
OSZ0039I CONSOLE INTERFACE IS NOW AVAILABLE
```

RTCS Operator Commands

Table 5-1 lists the supported RTCS operator commands.

Table 5-1 RTCS Operator Commands

command,operands	Valid contexts	Explanation
REFRESH,KERNEL	RTCS Subsystem only	RTCS will determine if there is an updated RTCS kernel package in the RTCS Subsystem Program Library (.OSZRTCS) that was specified when the RTCS Initiator address space was started. If so, it will be loaded into extended CSA and will replace the current older RTCS kernel. Use this command to refresh the current kernel, if necessary, after the application of maintenance to the kernel package in the RTCS Subsystem Program Library. You may enter this command at any time; if there is no updated kernel then no action will be performed.
REFRESH,LIBRARY	RTCS Subsystem only	RTCS determines if there are any updated dynamic link library (DLL) packages in the RTCS Subsystem Program Library (.OSZRTCS) that was specified when the RTCS Initiator address space was started. If so, those that have been updated will be loaded into extended CSA, replacing the current DLLs in storage. Use this command to refresh the current DLL routines, if necessary, after application of maintenance to DLL packages in the RTCS Subsystem Program Library. You may enter this command at any time; if there are no updated DLL packages, then no action is performed.
REFRESH,SECURITY	RTCS Subsystem only	The RTCS Security Manager reinitializes every Global Security Parameter from values stored in the RTCS System Registry. This command is normally used only after the RTCS System Registry is initialized for the very first time using the Registry Import Utility (RIU) on a newly-allocated VSAM LDS used to back the System Registry on an MVS image. You can use this command any time if any Global Security Parameters have been changed by the RIU.

Table 5-1 RTCS Operator Commands

command,operands	Valid contexts	Explanation
SECTRACE, {NONE | ON | SIMPLE | EXTENDED | COMPLETE}	RTCS Subsystem only	Indicates the level of diagnostic trace messages which the RTCS Security Manager component is to issue. NONE indicates that no diagnostic trace messages are to be issued. SIMPLE, EXTENDED, and COMPLETE represent three different levels of message trace. ON is equivalent to COMPLETE.
WARNMSG {ON | OFF}	RTCS Subsystem or General Server address spaces.	Indicates whether RTCS functions that are executing in the address space are to issue any warning level messages that are produced. The default status of this for every address space is OFF, but this option can be set to ON for any address space using the RTCS Generalized Server, or for the RTCS Subsystem address space.
START,code[,J=cscb]	RTCS Subsystem only	Directs RTCS to internally start (using the MVS START command) a General Server address space for the RTCS-based product which is indicated by a three-letter identifier (such as ZSE). RTCS will construct the necessary START command and issue it internally. The CSCB name for the General Server will be the same as its identifier, unless you also specify another value using the J= operand.

Security

The RTCS Security Manager component offers an extensive set of functions to RTCS-based and RTCS-dependent products which are accompanied by an extensive customization capability that is made available to the installation.

The RTCS Security Manager uses the RTCS System Registry to store its configuration and any customizations that have been established by the installation. The following configuration records are supported by the RTCS Security Manager and stored in the RTCS System Registry: Global Security Parameters and Global Class Properties.

Global Security Parameters

The RTCS Security Manager provides a set of default global security parameters that are suitable for the majority of installations, ESM products, and MVS configurations. There is generally never any reason to change any of the security parameters, but they have been implemented and are provided specifically for use in those special cases where it is necessary.

RTCS global security parameters (GSP) are customized by creating or updating the CUST GSP data structure in the RTCS System Registry. This updating is done by executing the RTCS RIU to IMPORT the registry IMPORT file (member) which defines the contents of the fields contained in that GSP data structure. A model registry import file is contained in member OSZRSGSP of the TOSZRXML SMP/E target library and a sample job to IMPORT it is contained in member OSZJSGSP of the TOSZCNTL SMP/E target library. The SYSIN input stream of the OSZJSGSP sample job describes each of the individual GSPs, and the values.

After you make any changes to the GSPs, enter the following operator command to make them effective:

```
F RTCS,REFRESH,SECURITY
```

where *RTCS* is the SSID of the RTCS Subsystem.

Global Class Properties

The RTCS Security Manager provides a complete set of default Global Class Properties for all standard MVS SAF resource class names that are used in all currently supported RTCS-dependent and RTCS-based products.

RTCS Global Class Properties (GCP) are customized by creating or updating a CUST GCP data structure for any given resource class name in the RTCS System Registry. This is done by executing the RTCS RIU to IMPORT registry IMPORT file which defines the contents of the fields contained in GCP registry data structures. A model registry IMPORT file is contained in member OSZRSGCP of the TOSZRXML SMP/E target library, and a sample job to import it is contained in member OSZJSGCP of the TOSZCNTL SMP/E target library. The SYSIN input stream of the OSZJSGCP sample job describes each of the individual GCPs that may be specified for a SAF resource class name and the values they may take.

Chapter 6 Registry Import Utility

This chapter provides information about how to use the RTCS Registry Import Utility (RIU).

This chapter contains the following sections:

Introduction

The RTCS RIU is provided to facilitate the use of the RTCS System Registry for product installation and configuration by using an XML document type. The RIU program (OSZRGIMP) processes these XML documents and updates the RTCS System Registry data structures.

Registry import data structures can be parameterized at execution time using the symbol substitution mechanism. An overall set of product configuration data can be described and shipped with the product and then configured for each individual system or product instance. The values of substituted symbols are specified by SET control statements in the SYSIN input stream to the RIU.

The RIU verifies the integrity of the complex data structures against their unique type definition before updating the registry. In addition, substituted data values are validated based on the specific data type required within a data structure. This action protects the registry data from corruption at the syntax level.

Registry Basics

The RTCS System Registry is organized as a hierarchical set of keys and values. A *registry key* is a named container of other keys and values. A *value* is a named data element that has a specific data type.

Each registry key or value is uniquely identified by the set of key names descending from the root key, down to the key or value in question. This set of key names is called a *registry path* and can be written in the same way as a hierarchical file system path. For example, a key named *software* that is immediately under the root key is written "`/software`" and a value named *foo* within that key is written "`/software/foo`".

The type of a registry value can be a simple, predefined data type, such as an integer or string data element, as well as a complex structured data type that contains other complex data types. This factor allows binary data structures to be stored and retrieved directly and facilitates their use programmatically.

All registry keys and values data types can be represented as an XML document.

Figure 6-1 provides an example of the JCL needed to execute the RIU.

Figure 6-1 Registry Import Utility Job

```
//jobname  JOB (acct,room),programmer,NOTIFY=&SYSUID
//IMPORT  EXEC PGM=OSZEXEC8,REGION=4M,
//             PARM='P=OSZRGIMP,C=BMCPROD'
//XMLDOC    DD DSN=USERID.YOUR.PDS,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
SET ISA.SSID     = RTCS
SET ISA.POSZHTML = SYS1.RTCS.OSZHTML
SET ISA.POSZLINK = SYS1.RTCS.OSZLINK
IMPORT XMLDOC(IMPTFILE)
//
```

Figure 6-1 contains the JCL that processes the member IMPTFILE of the data set USERID.YOUR.PDS as a registry import file. The results and messages are sent to the SYSPRINT SYSOUT data set.

The RIU (OSZRGIMP) is invoked using the RTCS Generalized Server (OSZEXEC8). As a result, you do not need to specify the DSNAME for the RTCS Product Program Library; it is dynamically allocated by the RTCS Generalized Server, which is loaded into dynamic LPA as part of RTCS Subsystem initialization.

Required Data Definition Names

The RIU requires a minimal set of data set allocations in order to function properly. This section lists the ddnames of each of these required allocations and explains their purpose.

SYSIN

The RIU executes by interpreting control statements supplied in the data set specified by the SYSIN DD statement. Each record is read one at a time and interpreted as either an individual control statement or a comment.

A comment record begins with an asterisk (*) as the first non-whitespace character. Comments are ignored and can be used for documentation purposes as appropriate for the control statements. For more information on control statement records, see “Control Statements” on page 6-4.

SYSPRINT

The RIU generates standard listing-type output to the data set defined by the SYSPRINT DD statement. This output includes both information and error messages that occur during the processing of the RIU control statements read from SYSIN.

Control Statements

All processing carried out by the RIU is driven by control statement records supplied in the SYSIN DD allocation. A control statement is a single record of input, which begins with a keyword such as SET, IMPORT, or OPTIONS. The keyword must be the first non-whitespace characters on the record.

Each control statement is processed completely before the RIU advances to the next statement. If an error is detected in the syntax of a control statement, all processing terminates before the next record is read. An error message is generated to SYSPRINT that indicates the nature of the problem and provides the SYSIN input record number to help locate the problem.

SET

The SET control statement creates or replaces a name/value mapping used by the RIU to satisfy substitution elements found in registry import files. The SET statement is written as:

```
SET variable = value
```

The value established by this control statement is used for any import files that are subsequently processed.

Substitution variable names can begin with any alphabetic or national symbols on your keyboard.

It is possible to include whitespace (for example, one or more blank characters) in a substitution value, if the value is surrounded by either the single (’) or double (") quotation marks. If you use the single quotation mark, you cannot use a double quotation mark within the substitution value.

IMPORT

The IMPORT control statement directs the RIU to read the registry definitions that are present in the file specified and apply them to the RTCS System Registry.

The following syntax should be used:

```
IMPORT fileName
```

The registry import file can be a sequential data set or a partitioned data set member that is record oriented. The *fileName* can be specified either as a ddname that has been allocated to the job step or as an explicit data set name.

If the ddname form is used, the import *fileName* value can be written directly as the ddname allocated in the job step for this purpose. If the indicated allocation is to a partitioned data set, then the *fileName* would also include the specific member name, written as:

```
IMPORT ddname(member)
```

If the data set name form is used, then the import *fileName* value must indicate that direct specification by being prefixed with the characters *DSN*: and written as:

```
IMPORT DSN:dsname
```

If a partitioned data set is being used, the `IMPORT` statement is written as:

```
IMPORT DSN:dsname(member)
```

OPTIONS

Use the OPTIONS control statement to adjust the behavior of the RIU for all control statements following the specification of that OPTIONS control statement. The following syntax should be used for the OPTIONS control statement:

```
OPTIONS [ [NO]TRACE | [NO]TRACEX | [NO]PRETEND ]
```

The supported options instruct the RIU to generate additional output that describes the actions taken while processing an import file. Each of these options has a corresponding negation beginning with NO, with three additional keywords that can also be specified: NOTRACE, NOTRACEX, and NOPRETEND. The default value of the OPTIONS is off: that is, NOTRACE, NOTRACEX, and NOPRETEND. Multiple options may be specified on a single OPTIONS control statement if each one is separated by a comma.

The TRACE option enables the generation of detailed information regarding the actions actually taken against the RTCS System Registry as a result of processing the import file. This information is useful as a record of what was actually done as a result of the RIU processing. This action includes creating, updating, and verifying keys and values. Each of these messages are generated with informational severity and do not affect any actual RIU processing.

The TRACEX option enables the generation of detailed information regarding the analysis of the XML structure of the import file. This information includes messages regarding the start and end of elements, the attributes processed, and the substitutions done. This information is intended for debugging purposes.

The PRETEND option prevents the RIU from carrying out any updates to the RTCS System Registry, but otherwise completely processes the import file. If the TRACE option is enabled, messages are generated that describe the actions that would have been taken against the RTCS System Registry. The intent of the PRETEND option is to allow a trial-run of one or more import files to determine whether there are any errors before actually committing the changes.

Return Codes and Messages

During the processing of the RIU, several different forms of feedback are provided to indicate the success or failure of each invocation. This section details the return code and general message categories that should be expected.

Return Codes

When the RIU encounters an error in processing that causes it to terminate without continuing, a return code is generated to indicate the nature of the failure. Each return code is detailed in Table 6-1.

Table 6-1 Return Code Table

Return Code	Description
20	Unable to open SYSPRINT. Ensure that the SYSPRINT DD statement is correctly specified and resubmit the JOB.
16	Unable to open SYSIN. Ensure that the SYSIN DD statement is correctly specified and resubmit the JOB.
12	Unable to open a data set associated with an IMPORT control statement. Error messages are generated to SYSPRINT to indicate which IMPORT statement contained the failing data set or data set with member name. If the failing IMPORT statement uses the ddname form of file name specification, check the allocation of that ddname for errors (for example, ensure that the specified member exists).
8	An error occurred during the processing of either an RIU control statement or an XML import file. This return code is accompanied by messages written to SYSPRINT that diagnoses the problem that occurred.

Messages

During the processing of IMPORT control statements and registry import files, the RIU generates a variety of different types of messages. Each message is prefixed with *OSZ*, and ends with one of the following message severity indicators:

- E - Error
- W - Warning
- I - Informational

All RIU messages are in the range OSZ1500*x* - OSZ1599*x*. Each individual message generated by the RIU is documented in the *Runtime Component System Message Reference* that is available on the installation CD.

Using the defaults, the RIU generates a small set of messages for all invocations. This includes:

- output of each control statement as it is being processed
- indication of the start and end of processing of each import file
- completion of the import utility program
- all warning and error messages for situations encountered during processing

With the TRACE option active, the RIU generates additional messages detailing the disposition of processing each control statement, as well as the results of each update to the registry. Each of these additional messages are informational and documents what was done.

With the TRACEX option active, the RIU generates additional messages detailing the progress in parsing the elements and attributes of the XML import file. Each of these additional messages is informational and is intended for the purpose of debugging problems with the operation of the RIU. Any failures in the XML input will be diagnosed with a distinct error message with or without this option.

Appendix A BMC Software Product Authorization

Overview

When Customer Password Response of BMC Software processes a license agreement for a product, it issues CPU authorization passwords. These passwords authorize specific CPUs (also referred to as processors) to run the licensed product. Because BMC Software licenses its products for use on individual CPUs, the passwords are product specific and CPU-specific (one license per product per CPU). To delete or replace an authorized CPU, you must also have a password.

For RTCS-based products, you can use an existing facility that is already installed for existing licensed BMC Software products, or the RTCS Product Authorization Table Library Update utility to apply passwords or to change your licensed CPU configuration.

The types of passwords are as follows:

- Temporary passwords are issued for product trials or in other special circumstances (for example, when a hardware failure prevents you from using the authorized CPU).

- Permanent passwords are issued when you convert to a permanent license, delete or replace a CPU, or modify the properties of a CPU or the product authorization tables.

Note: You do not need to apply passwords or update CPU authorization when you install product maintenance or version upgrades.

Passwords for RTCS-based products can be processed in either of the following ways:

- during the Desktop Installation Assistant MVS Image Configuration procedure

- in a batch interface that uses JCL to execute the Product Authorization Table Library Update utility program, OSZPATLU, provided by RTCS

This appendix describes the process that you use to apply passwords and to reconfigure your CPU, either permanently or temporarily. If you have additional authorization questions or concerns about the Product Authorization utility, contact your BMC Software sales representative.

Product Authorization Tables

When you apply passwords, the BMC Software Product Authorization Table Library Update utility builds or updates product authorization tables in the designated library. The utility uses license keys or passwords to create entries in the a table that define processors authorized to use the specified licensed product.

The types of product authorization tables are as follows:

- The Product Authorization Table Library Update utility builds or updates a permanent product authorization table when you install or apply a permanent password. The permanent table controls which CPUs are licensed to run the product, based on the serial number, the model number, and the submodel number of the CPU.

- The Product Authorization Table Library Update utility builds or updates a temporary product authorization table when you apply a temporary password.

See “Product Authorization Passwords” on page A-4 for more information about permanent and temporary passwords.

Product authorization tables are product-specific members of the Product Authorization Table Library and are identified by the three-character product code, as in the following examples (where the variable *prd* is the three-character product code):

*prd*TBL3P (permanent)

*prd*TBL3T (temporary)

Using a PDSE Product Authorization Table Library

A Product Authorization Table Library cannot contain both program objects and product authorization tables if the data set is allocated as a PDSE (partition data set extended). PDSE cannot contain program objects and data type objects together in the same library. The product authorization table is a data file.

The Desktop Installation Assistant will allocate a PDSE for any new Product Authorization Table Library. If an existing Product Authorization Table Library that is an ordinary PDS is specified, that will work properly also. You may specify to the Desktop Installation Assistant the DSNAME of an existing PDS which does not contain program objects since you cannot mix the two types of members.

The issue can be circumvented by specifying a different library to contain the product authorization tables or by allocating a new library dedicated for this purpose. You should allocate a separate PDSE (or ordinary PDS) for the Product Authorization Table Library and specify this library in the RTCS Initiator started task PROC JCL.

Product Authorization Passwords

Valid passwords can include the following characters:

- the alphanumeric character set, excluding the letters I and O to avoid confusion with the numbers one (1) and zero (0)
- equal sign (=), “at” sign (@), and plus sign (+)

Note: If your keyboard does not have the “at” sign (@), you can use the asterisk (*) in place of @. You can use these two characters (@ and *) interchangeably when typing passwords.

Permanent Passwords

Permanent passwords update a product’s permanent authorization table. Each permanent password has one of the functions described in Table A-1.

Table A-1 Permanent Password Functions

Function	Description
Add	authorizes one new CPU to run the product
Delete	removes one CPU from the table, preventing that CPU from running the product
Replace	replaces one CPU in the table with another CPU, allowing the new CPU to run the product in place of the old CPU

Table A-1 Permanent Password Functions

Function	Description
Modify	modifies one or more properties of one CPU that currently exists in the product authorization table
Reset	modifies the global properties of the product authorization tables

Temporary Passwords

BMC Software issues temporary passwords to customers who are evaluating products on a trial basis or to customers who need to bypass product authorization to run a product temporarily on an unlicensed CPU. Temporary passwords have a specific expiration date, which is part of the password.

How to Apply Passwords

You can apply the new passwords before you completely install the product if you have installed the Product Authorization Table Library Update Utility and have an existing product Authorization Table Library. Also, you can apply the passwords even if the product is not yet running on a specific CPU. For example, if your standard software product installation and configuration procedures require you to install and run the product on a test system or LPAR before migrating it to a production system, you can apply the password for the production system's CPU, even though the product is not yet running there.

RTCS-based products must have their passwords in the library defined by the BMCPSWD DD statement of the RTCS Initiator Started Task PROC JCL.

Passwords are saved in the specified Product Authorization Table Library during the Desktop Installation Assistant's MVS Image Configuration procedure, or by execution of the RTCS Product Authorization Table Library Update Utility (OSZPATLU) using the RTCS Generalized Server (OSZEXEC8) in a batch JOB.

How Products Are Licensed

You must use the Product Authorization Table Library Update utility in the following situations:

- for product trials and permanent licensing
- when upgrading to a new CPU
- when an authorized CPU fails

Note: Although you do not need the Product Authorization Table Library Update utility for product maintenance and version upgrades, you must consider certain issues that are associated with these upgrades. See “Product Maintenance or Version Upgrades” on page A-8 for more information.

Product Trials and Permanent Licensing

During a trial period for a BMC Software product, you can install and use the product on any CPU by using a temporary password that you obtained from your BMC Software sales representative. When you finish the trial and want to obtain a product license, the following rules apply:

- You must purchase a product license for each CPU on which you will run the product.
- BMC Software Customer Password Response issues a permanent password for each combination of CPU and licensed product.
- To enable a product on a CPU, you must add the permanent password that is issued for that CPU. You do not need to reinstall and retest the product.
- You can install multiple passwords in the same Product Authorization Table Library. This capability lets you use the same library to authorize execution of multiple products on multiple CPUs or to install a product at a central site and execute it at remote sites.

CPU Upgrades

When you upgrade to a new CPU, you must obtain a new permanent password for each product that you want to use on that CPU. When you install a new password for a specific licensed product, the old entry in the Product Authorization Table for that product is replaced. The new table entry defines the authorization for the product.

CPU Failures

If a hardware failure or a disaster-recovery situation prevents the use of a licensed CPU, BMC Software can provide a temporary license that lets the product run on a backup CPU for a limited time. Before the temporary license expires, you must acquire a permanent license for the new CPU or you must resume using the original CPU. At the end of the grace period, you can no longer run the affected product on the backup CPU. If the grace period expires, you must obtain a new password to reset the grace period.

Updating Product Authorization Tables

To trigger the grace period, the license validation process must update the Product Authorization Table in the Product Authorization Table Library. If the General Server address space user ID does not have WRITE or UPDATE access to the designated Product Authorization Table Library, problems will occur when you attempt to trigger the grace period. The ESM user ID which was assigned by default to the Generalized Server started task or executing JOB step must have UPDATE or WRITE access.

Before updating the Product Authorization Table Library, the license validation process determines whether the data set is in the system link list (LNKLST). If the data set is in LNKLST, the license validation process does not attempt an update. Normally, for RTCS-based products, the Product Authorization Table Library is not in LNKLST, because placing it there offers no advantage.

Running a Product on an Unlicensed Processor

When you run a product on an unlicensed processor, a 15-calendar-day grace period can be triggered. After this grace period expires, the product will not run or will run with diminished functionality.

Note: The product will continue to function normally when run on a licensed CPU, even if the grace period has been triggered or has expired.

To prevent this situation, you should obtain a RESET password from BMC Software Customer Password Response. If you apply the RESET password before the grace period ends, it updates the product authorization table and makes another 15-calendar-day grace period available.

When the grace period is triggered, the Product Authorization Table Library Update utility and the affected product issue a message that advises you of the expiration date.

Product Maintenance or Version Upgrades

Installing a new maintenance level or upgrading the version or release level of a product has no effect on product authorization. No new passwords are required. You can use the same Product Authorization Table Library that you use in production. If you install products in a test environment before moving them to production, the Product Authorization Table Library must be accessible to the test environment. If you try to run the product on a different CPU, that CPU must also be licensed. You can copy a Product Authorization Table from the production library to a test library allocated for that purpose. To copy the tables for a specific product from the production library to the test library, use the IEBCOPY utility to copy all members with the following names:

*prd*TBL3P

*prd*TBL3T

Do not use the IEBCOPY COPYMOD parameter when copying members from a Product Authorization Table Library.

How to Obtain Passwords

Table A-2 describes the situations in which you need to obtain passwords. For each scenario, the table indicates the type of password that you need (temporary or permanent), what the password does, and how to obtain it.

Table A-2 Password Scenarios

Scenario	PasswordType	Password Function	How to Obtain
You want to begin a free trial period.	temporary	temporarily bypasses authorization checking and allows you to run the product on any CPU for a limited time	BMC Software sales representative
You purchase a license for a new product.	permanent	adds a designated CPU to the list of CPUs that are authorized to run a licensed product	BMC Software sales representative or Customer Password Response (1-800-841-2031)
You stop using an authorized CPU.	permanent	removes a designated CPU from the list of CPUs that are authorized to run a licensed product	BMC Software sales representative or Customer Password Response (1-800-841-2031)

Table A-2 Password Scenarios

Scenario	PasswordType	Password Function	How to Obtain
You upgrade to a new CPU.	permanent	authorizes the transfer of a license from one CPU to another CPU	BMC Software sales representative or Customer Password Response (1-800-841-2031)
You want to run the product on an additional CPU.	permanent	adds a designated CPU to the list of CPUs that are authorized to run a licensed product	BMC Software sales representative or Customer Password Response (1-800-841-2031)
The authorized CPU is not available because of an emergency (such as hardware failure).	temporary	temporarily bypasses authorization checking and allows you to run the product on any CPU for a limited time	BMC Software sales representative, Customer Password Response (1-800-841-2031), or Customer Support (1-800-537-1813)

CPU Information

When you request a permanent product license from BMC Software, you must furnish information about the affected CPUs. For each product that you license, use the worksheet in Table A-3 to record the CPU information and the passwords that you receive from BMC Software. The first line of the table provides a sample entry for a 9X2 with three processors and a CPU ID of 10309-9021-DA.

Note: CPU information is not needed for temporary passwords.

Table A-3 Product Authorization Worksheet

CPU Serial	CPU Type	Version Code	CPU Model	No. of CPUs	Permanent Password
10309	9021	DA	9X2	3	123,456,789,ABC
— — — — —	— — — —	— —	— — —	— —	__ __ __ ,__ __ __, __ __ __ ,__ __ __
— — — — —	— — — —	— —	— — —	— —	__ __ __ ,__ __ __, __ __ __ ,__ __ __
— — — — —	— — — —	— —	— — —	— —	__ __ __ ,__ __ __, __ __ __ ,__ __ __
— — — — —	— — — —	— —	— — —	— —	__ __ __ ,__ __ __, __ __ __ ,__ __ __
— — — — —	— — — —	— —	— — —	— —	__ __ __ ,__ __ __, __ __ __ ,__ __ __

For information about determining your CPU ID, see “Batch Product Authorization” on page A-10 or use the LIST option of Batch Product Authorization.

Batch Product Authorization

This section describes the batch interface that is used to update the Product Authorization Table Library. There is no online Product Authorization Table Library update process provided by RTCS for use by RTCS-based products, but if you have one from another BMC product installed, you may use it.

Using the batch interface, you can perform the following tasks:

- process a password
- obtain current product authorization and processor information

Running Batch Product Authorization

Figure A-1 is sample JCL for a JOB to run the batch Product Authorization Table Library Update utility (OSZPATLU). For descriptions of the information in the JOB, see “Additional Information” on page A-11.

Figure A-1 JCL to Execute Batch Product Authorization Table Library Update Utility

```
//jobname  JOB (acct,room),programmer,NOTIFY=&SYSUID
//*
//PATLU   EXEC PGM=OSZEXEC8,PARM=('P=OSZPATLU,C=BMCPROD',
//             'A=prd')               <== PRODUCT CODE
//SYSPRINT  DD SYSOUT=*
//*
//SYSIN     DD *
****  PROCESS AN ADD PASSWORD AND LIST RESULTS ****
PSWD=AE@,82G,91#,C7$  NEWCPUID=11111-9021
****  PROCESS A DELETE PASSWORD AND LIST RESULTS ****
PSWD=BE@,AD0,32$,7C#  OLDCPUID=31091-9121
**** PROCESS A MODIFY PASSWORD AND LIST RESULTS ****
PSWD=123,456,789,ABC   OLDCPUID=98765-4321
****  PROCESS A REPLACE PASSWORD AND LIST RESULTS ****
PSWD=ARF,56C,##1,C7$   OLDCPUID=31001-3390  NEWCPUID=31091-3381
****  PROCESS A RESET PASSWORD
PSWD=123,456,789,ABC
**** PROCESS A TEMPORARY PASSWORD AND LIST RESULTS ****
PSWD=AE@,B32,#1C,D7#
****  REPORT THE PROCESSOR INFORMATION AND AUTHORIZATION ****
LIST
//
```

Additional Information

The JCL required to execute the batch Product Authorization Table Library Update utility (OSZPATLU) is as follows:

Table A-4 OSZPATLU batch JCL

JCL Statement	Description
JOB	installation-specific
EXEC	executes the RTCS Generalized Server (OSZEXEC8, specifying the RTCS component program to be invoked (OSZPATLU) and passes the product code in the PARM field Replace *prd* with the three-character product code.
SYSPRINT DD	enables the product to issue messages and output from the LIST control statement

Table A-4 OSZPATLU batch JCL

JCL Statement	Description
SYSIN DD	identifies the location of the control statements that define the actions the program is to take For a description of these control statements, see “Control Statements and Keywords” on page A-12.
NOTE: The RTCS version of the batch Product Authorization Table Library Update utility (OSZPATLU) does not require either a STEPLIB DD statement to define the RTCS Product Program Library, or a SYSLIB DD statement to define the Product Authorization Table Library. All of the required data sets are dynamically allocated.	

Control Statements and Keywords

Some tasks require different input parameters depending on the type of password that you are installing. The sample JCL shown in Figure A-1 on page A-11 shows various tasks that you can perform by using the batch version of product authorization. You need to modify the JCL to include only the tasks that you want to perform.

The following syntax rules apply to the control statements:

- Control statements can begin in any column.
- Uppercase letters are required.
- You must insert at least one blank space between individual keywords and data fields. Multiple blank spaces are acceptable.
- To insert comments, type an asterisk (*) in column 1 of each line that contains the comment. Comments following keywords are not allowed.
- You cannot specify the LIST keyword on the same line as PSWD, NEWCPUID, and OLDCPUID.

Table A-5 on page A-13 describes the control statement keywords.

Table A-5 Control Statement Keywords

Keyword	Data	Explanation
PSWD	12-character password formatted as four fields of three characters each, separated by either a comma or a blank (See sample JCL on page A-11.) Twelve contiguous characters are also acceptable.	Valid characters are alphanumeric (excluding letters I and O). Valid special characters are =, +, and @. You can substitute the asterisk (*) for the “at” sign (@) when @ is not available on the keyboard.
NEWCPUID	five-digit serial number, followed by a hyphen and a four-digit model number.	The serial number and model number must be hexadecimal characters separated by a single hyphen.
OLDCPUID	five-digit serial number, followed by a hyphen and a four-digit model number	The serial number and model number must be hexadecimal characters separated by a single hyphen.
LIST	not applicable	Prints a report showing the contents of the product authorization table and information about the processor on which the job ran.

Return Codes

You can receive any of the following return codes when you use the batch Product Authorization Table Library Update utility:

Table A-6 Return Codes

Return Code	Description
0	All requests completed successfully. See the SYSPRINT output for messages about each operation.
4	A LIST was requested, but no tables were in the Product Authorization Table Library.
8	An error prevented completion of all of your requests. See the SYSPRINT output for messages about the error and any completed operations.

Appendix B Sample JCL and MVS Image Configuration Members

This chapter provides reference information regarding the RTCS Started Task PROCs and required MVS PPT table additions. Refer to these examples for assistance in configuring these elements for an MVS image.

This chapter contains the following sections:

RTCS Initiator Started Task PROC

Figure B-1 is an example of the RTCS Initiator PROC.

Figure B-1 RTCS Initiator PROC

```
//OSZINIT PROC SSID=RTCS,RTCS=OSZRTCS,EXEC=OSZEXEC
//OSZINIT EXEC PGM=OSZSIRIS,
//             TIME=1440,
//             REGION=0M,
//             PARM=('SSID=&SSID',
//             'RTCS=&RTCS',
//             'EXEC=&EXEC',)
//STEPLIB   DD DISP=SHR,DSNAME=SYS1.RTCS.TOSZRTCS
//TASKLIB   DD DISP=SHR,DSNAME=SYS1.RTCS.TOSZLINK
//HTMLDOC   DD DISP=SHR,DSNAME=SYS1.RTCS.TOSZHTML
//REGISTRY  DD DISP=SHR,DSNAME=SYS1.&SYSNAME..REGISTRY
//BMCPSWD   DD DISP=SHR,DSNAME=BMC.PROD.PASSWORD
```

This member must be copied into SYS1.PROCLIB, or another PROCLIB that is available for started tasks begun under the Master (MSTR) Subsystem. The STEPLIB, TASKLIB, HTMLDOC, BMCPSWD, and REGISTRY DD statements must be updated to reflect the actual DSNAMEs to be used. These DSNAMEs are established by the Desktop Installation Assistant, if it is used to configure the target MVS image. If you copy this PROC for use on another MVS image, these DD statements may need updating.

The RTCS Initiator address space should be started as early as possible in the MVS IPL process (that is, after the security system has been initialized). For example, the following command could be used:

```
S OSZINIT,SUB=MSTR,RTCS=OSZRTCS,EXEC=OSZEXEC,SSID=RTCS
```

If the PROC statement's default values are correct, use the following command:

```
S OSZINIT,SUB=MSTR
```

The RTCS Initiator address space is not the same address space as the RTCS Subsystem address space. The RTCS Subsystem address space is started by the RTCS Initiator after performing certain validation procedures on the MVS system, the STEPLIB data set, the TASKLIB data set, the HTMLDOC data set, the REGISTRY VSAM LDS used to back the RTCS System Registry, and the BMCPSWD Product Authorization Table Library.

The DSNAMEs of the STEPLIB, TASKLIB, and REGISTRY data sets are determined from this allocation and passed to the RTCS Subsystem address space by the RTCS Initiator internally in MVS ECSA storage. For example, the RTCS Subsystem address space dynamically allocates these data sets and then unallocates the program libraries, thus enabling maintenance to be performed using SMP/E. This is necessary because the RTCS Subsystem address space cannot be terminated.

The Product Authorization Table Library DSNAME and the Hypertext Document Library DSNAME are determined from the BMCPSWD and HTMLDOC allocation here and passed internally to RTCS Generalized Server address spaces in MVS ECSA storage.

The Product Authorization Table Library is a partitioned data set that contains product license table members, which are created and updated by the BMC Software Licensing Facility batch password processing utility. The Product Authorization Table Library must contain at least one table member for each licensed, RTCS-based product. The member names are *prd*TBL3T for a temporary password and *prd*TBL3P for a permanent password (where the variable *prd* is the three-character product code). The RTCS Generalized Server dynamically allocates this data set and performs product license authorization, and then immediately deallocates it.

The RTCS System Registry is an MVS image-specific data set. A separate RTCS Registry VSAM LDS must be allocated for every MVS image on which RTCS is used.

The RTCS System Registry contains configuration information for RTCS and any associated products that are specific to, and tailored expressly for, the MVS image on which the product is being used. The RTCS Subsystem for every MVS image must have its own RTCS Registry data set for the exclusive use of RTCS on that specific MVS image. The RTCS System Registry is read/write and cannot be shared; however, it may be allocated on a shared DASD volume.

The RTCS System Registry is a VSAM LDS and must be allocated at least 64MB of space. RTCS does not attempt to expand the data set to 64MB if less space is allocated; if this occurs, RTCS is not able to initialize. The first time RTCS is started with a newly-allocated, uninitialized VSAM LDS, the data set is initialized so that the RTCS RIU can be executed to initially populate the registry data structures.

The RTCS System Registry VSAM LDS must be cataloged, since it is a VSAM cluster. Because it is an MVS image-specific data set, it is recommended that it be cataloged in the master catalog. Because the VSAM cluster remains allocated for the life of the IPL (since the RTCS address space cannot be shut down), carefully select its DSNAME and catalog.

RTCS Subsystem Started Task PROC

Figure B-2 is an example of the RTCS Subsystem PROC.

Figure B-2 RTCS Subsystem PROC

```
//OSZRTCS PROC SSID=RTCS
//OSZRTCS EXEC PGM=OSZMOSYS,
//             TIME=1440,
//             REGION=0M,
//             PARM=('SSID=&SSID',)
```

Copy this member into SYS1.PROCLIB or another PROCLIB that is available to started tasks that are started under the Master (MSTR) Subsystem.

Specify the member name of this PROC as the value of the RTCS parameter in the OSZINIT started task PROC JCL, or in the START OSZINIT operator command. For example, the following START command can be used:

```
S OSZINIT,SUB=MSTR,RTCS=OSZRTCS,EXEC=OSZEXEC,SSID=RTCS
```

The RTCS Subsystem control address space is started internally by the RTCS Initiator address space, as early as possible in the MVS IPL process. A command such as the following is issued internally:

```
S OSZRTCS.RTCS,SUB=MSTR,SSID=RTCS
```

The SUB=MSTR specification is mandatory because RTCS must start under the Master Subsystem. Because of that, this OSZRTCS started task procedure JCL must be placed in a PROCLIB data set defined in the IEFPDSI concatenation defined in the Master Scheduler JCL: for example, MSTJCL00. This PROCLIB data set is usually SYS1.PROCLIB, but the installation may have some other PROCLIB data set defined in the concatenation (for example, one prior to SYS1.PROCLIB for overrides or one shared by all sysplex images) that may be used.

RTCS Generalized Server Started Task PROC

Figure B-3 is an example of the RTCS Generalized Server PROC.

Figure B-3 RTCS Generalized Server PROC

```
//OSZEXEC PROC KEY=8,P=*,C=*,O=
//OSZEXEC EXEC PGM=OSZEXEC&KEY,
//             TIME=1440,
//             REGION=0M,
//             PARM=('P=&P',
//             'C=&C',
//             '&O',)
```

Copy this member into a JES-managed started task procedure library that is available to started tasks that are started under the primary Job Entry Subsystem. You must also copy this member into SYS1.PROCLIB or another PROCLIB that is available to started tasks that are started under the Master (MSTR) Subsystem.

Specify the member name of this PROC as the value of the EXEC parameter in the OSZINIT started task PROC JCL, or in the START OSZINIT operator command. For example, the following START command can be used:

```
S OSZINIT,SUB=MSTR,RTCS=OSZRTCS,EXEC=OSZEXEC,SSID=RTCS
```

MVS Program Properties Table Additions

The following entries must be added to the MVS program properties table.

Figure B-4 MVS Program Properties Table Entry Additions

```
/** PRODUCT_IDENTIFICATION ****************************************/
/*                                                               */
/* PRODUCT_NAME: RTCS             COMPONENT_ID: KERNEL           */
/* PRODUCT_FMID: ZOSZ111        DIAGNOSIS_PIDS: ZOSZ-111         */
/* CURRENT_RMID: BASE             LAST_UPDATE: 10/22/02          */
/*                                                               */
```

```
/** MEMBER_DESCRIPTION ********************************************/
/*                                                                */
/*      MEMBER: OSZSCHED                                          */
/*    CONTENTS: THIS MEMBER CONTAINS THE PROGRAM PROPERTIES TABLE */
/*              ENTRIES REQUIRED TO PROPERLY INITIATE THE RUNTIME */
/*              COMPONENT SYSTEM INITIATOR, ALLOCATOR, AND KERNEL */
/*              STARTED TASKS, PLUS GENERALIZED SERVER JOB STEPS. */
/*                                                                */
/****************************************************************/
PPT  PGMNAME(OSZSIRIS)            /* RTCS SUBSYSTEM INITIATOR STC    */
     KEY(0)                       /* RTCS SUBSYSTEM INITIATOR KEY 0  */
     PRIV                         /* WLM COMPATIBILITY MODE PGN=000  */
     SYST                         /* NON-TIMED STARTED SYSTEM TASK   */
PPT  PGMNAME(OSZMOSYS)            /* RTCS SUBSYSTEM ALLOCATOR STC    */
     KEY(0)                       /* RTCS SUBSYSTEM ALLOCATOR KEY 0  */
     PRIV                         /* WLM COMPATIBILITY MODE PGN=000  */
     NOCANCEL                     /* KERNEL    CANNOT BE CANCELLED   */
     NOSWAP                       /* NON-SWAPPABLE, CROSS-MEMORY A/S */
     SYST                         /* NON-TIMED STARTED SYSTEM TASK   */
PPT  PGMNAME(OSZIRIAN)            /* RTCS SUBSYSTEM KERNEL    STC    */
     KEY(0)                       /* RTCS SUBSYSTEM KERNEL    KEY 0  */
     PRIV                         /* WLM COMPATIBILITY MODE PGN=000  */
     NOCANCEL                     /* MUST NOT BE CANCELLED           */
     NOSWAP                       /* NON-SWAPPABLE, CROSS-MEMORY A/S */
     SYST                         /* NON-TIMED STARTED SYSTEM TASK   */
PPT  PGMNAME(OSZEXEC0)            /* RTCS SERVER    STARTED TASK - 0 */
     KEY(0)                       /* PRODUCTS WHICH EXECUTE IN KEY 0 */
PPT  PGMNAME(OSZEXEC1)            /* RTCS SERVER    STARTED TASK - 1 */
     KEY(1)                       /* PRODUCTS WHICH EXECUTE IN KEY 1 */
PPT  PGMNAME(OSZEXEC2)            /* RTCS SERVER    STARTED TASK - 2 */
     KEY(2)                       /* PRODUCTS WHICH EXECUTE IN KEY 2 */
PPT  PGMNAME(OSZEXEC3)            /* RTCS SERVER    STARTED TASK - 3 */
     KEY(3)                       /* PRODUCTS WHICH EXECUTE IN KEY 3 */
PPT  PGMNAME(OSZEXEC4)            /* RTCS SERVER    STARTED TASK - 4 */
     KEY(4)                       /* PRODUCTS WHICH EXECUTE IN KEY 4 */
PPT  PGMNAME(OSZEXEC5)            /* RTCS SERVER    STARTED TASK - 5 */
     KEY(5)                       /* PRODUCTS WHICH EXECUTE IN KEY 5 */
PPT  PGMNAME(OSZEXEC6)            /* RTCS SERVER    STARTED TASK - 6 */
     KEY(6)                       /* PRODUCTS WHICH EXECUTE IN KEY 6 */
PPT  PGMNAME(OSZEXEC7)            /* RTCS SERVER    STARTED TASK - 7 */
     KEY(7)                       /* PRODUCTS WHICH EXECUTE IN KEY 7 */
```

Index

A

B

C

D

I

J

L

M

N

O

P

R

S

T

U

V

W

END USER LICENSE AGREEMENT
NOTICE

BY OPENING THE PACKAGE, INSTALLING, PRESSING "AGREE" OR "YES" OR USING THE PRODUCT, THE ENTITY OR INDIVIDUAL ENTERING INTO THIS AGREEMENT AGREES TO BE BOUND BY THE FOLLOWING TERMS. IF YOU DO NOT AGREE WITH ANY OF THESE TERMS, DO NOT INSTALL OR USE THE PRODUCT, PROMPTLY RETURN THE PRODUCT TO BMC OR YOUR BMC RESELLER, AND IF YOU ACQUIRED THE LICENSE WITHIN 30 DAYS OF THE DATE OF YOUR ORDER CONTACT BMC OR YOUR BMC RESELLER FOR A REFUND OF LICENSE FEES PAID. IF YOU REJECT THIS AGREEMENT, YOU WILL NOT ACQUIRE ANY LICENSE TO USE THE PRODUCT.

This Agreement ("**Agreement**") is between the entity or individual entering into this Agreement ("You") and BMC Software Distribution, Inc., a Delaware corporation located at 2101 CityWest Blvd., Houston, Texas, 77042, USA or its affiliated local licensing entity ("BMC"). "You" includes you and your Affiliates. "Affiliate" is defined as an entity which controls, is controlled by or shares common control with a party. THIS AGREEMENT WILL APPLY TO THE PRODUCT, UNLESS (1) YOU AGREED TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT, IN WHICH CASE THAT WEB BASED LICENSE AGREEMENT GOVERNS THE USE OF THE PRODUCT, OR (2) IF YOU DID NOT AGREE TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT AND YOU HAVE A WRITTEN LICENSE AGREEMENT WITH BMC, THEN THAT WRITTEN AGREEMENT GOVERNS THE USE OF THE PRODUCT. THE ELECTRONIC AGREEMENT PROVIDED WITH THE PRODUCT AS PART OF THE INSTALLATION OF THE PRODUCT WILL NOT APPLY. In addition to the restrictions imposed under this Agreement, any other usage restrictions contained in the Product installation instructions or release notes shall apply to Your use of the Product.

PRODUCT AND CAPACITY. "**Software**" means the object code version of the computer programs provided, via delivery or electronic transmission, to You. Software includes computer files, enhancements, maintenance modifications, upgrades, updates, bug fixes, and error corrections.

"**Documentation**" means all written or graphical material provided by BMC in any medium, including any technical specifications, relating to the functionality or operation of the Software.

"**Product**" means the Software and Documentation.

"**License Capacity**" means the licensed capacity for the Software with the pricing and other license defining terms, including capacity restrictions, such as tier limit, total allowed users, gigabyte limit, quantity of Software, and/or other capacity limitations regarding the Software. For licenses based on the power of a computer, You agree to use BMC's current computer classification scheme, which is available at http://www.bmc.com or can be provided to You upon request.

ACCEPTANCE. The Product is deemed accepted by You, on the date that You received the Product from BMC.

LICENSE. Subject to the terms of this Agreement, as well as Your payment of applicable fees, BMC grants You a non-exclusive, non-transferable, perpetual (unless a term license is provided on an order) license for each copy of the Software, up to the License Capacity, to do the following:

(a) install the Software on Your owned or leased hardware located at a facility owned or controlled by You in the country where You acquired the license;

(b) operate the Software solely for processing Your own data in Your business operations; and

(c) make one copy of the Software for backup and archival purposes only (collectively a "**License**").

If the Software is designed by BMC to permit you to modify such Software, then you agree to only use such modifications or new software programs for Your internal purposes or otherwise consistent with the License. BMC grants You a license to use the Documentation solely for Your internal use in Your operations.

LICENSE UPGRADES. You may expand the scope of the License Capacity only pursuant to a separate agreement with BMC for such expanded usage and Your payment of applicable fees. There is no additional warranty period or free support period for license upgrades.

RESTRICTIONS: You agree to **NOT**:
(a) disassemble, reverse engineer, decompile or otherwise attempt to derive any Software from executable code;

(b) distribute or provide the Software to any third party (including without limitation, use in a service bureau, outsourcing environment, or processing the data of third parties, or for rental, lease, or sublicense); or

(c) provide a third party with the results of any functional evaluation or benchmarking or performance tests, without BMC's prior written approval, unless prohibited by local law.

TRIAL LICENSE. If, as part of the ordering process, the Product is provided on a trial basis, then these terms apply: (i) this license consists solely of a non-exclusive, non-transferable evaluation license to operate the Software for the period of time specified from BMC or, if not specified, a 30 day time period ("**Trial Period**") only for evaluating whether You desire to acquire a capacity-based license to the Product for a fee; and (ii) Your use of the Product is on an AS IS basis without any warranty, and **BMC, ITS AFFILIATES AND RESELLERS, AND LICENSORS DISCLAIM ANY AND ALL WARRANTIES (INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT) AND HAVE NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE ("Trial License")**. BMC may terminate for its convenience a Trial License upon notice to You. When the Trial Period ends, Your right to use this Product automatically expires. If You want to continue Your use of the Product beyond the Trial Period, contact BMC to acquire a capacity-based license to the Product for a fee.

TERMINATION. This Agreement shall immediately terminate if You breach any of its terms. Upon termination, for any reason, You must uninstall the Software, and either certify the destruction of the Product or return it to BMC.

OWNERSHIP OF THE PRODUCT. BMC or its Affiliates or licensors retain all right, title and interest to and in the BMC Product and all intellectual property, informational, industrial property and proprietary rights therein. BMC neither grants nor otherwise transfers any rights of ownership in the BMC Product to You. BMC Products are protected by applicable copyright, trade secret, and industrial and intellectual property laws. BMC reserves any rights not expressly granted to You herein.

CONFIDENTIAL AND PROPRIETARY INFORMATION. The BMC Products are and contain valuable confidential information of BMC ("**Confidential Information**"). Confidential Information means non-public technical and non-technical information relating to the BMC Products and Support, including, without limitation, trade secret and proprietary information, and the structure and organization of the Software. You may not disclose the Confidential Information to third parties. You agree to use all reasonable efforts to prevent the unauthorized use, copying, publication or dissemination of the Product.

WARRANTY. Except for a Trial License, BMC warrants that the Software will perform in substantial accordance with the Documentation for a period of one year from the date of the order. This warranty shall not apply to any problems caused by software or hardware not supplied by BMC or to any misuse of the Software.

EXCLUSIVE REMEDY. BMC's entire liability, and Your exclusive remedy, for any defect in the Software during the warranty period or breach of the warranty above shall be limited to the following: BMC shall use reasonable efforts to remedy defects covered by the warranty or replace the defective Software within a reasonable period of time, or if BMC cannot remedy or replace such defective copy of the Software, then BMC shall refund the amount paid by You for the License for that Software. BMC's obligations in this section are conditioned upon Your providing BMC prompt access to the affected Software and full cooperation in resolving the claim.

DISCLAIMER. EXCEPT FOR THE EXPRESS WARRANTIES ABOVE, THE PRODUCT IS PROVIDED "AS IS." BMC, ITS AFFILIATES AND LICENSORS SPECIFICALLY DISCLAIM ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. BMC DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS CAN BE CORRECTED.

DISCLAIMER OF DAMAGES. IN NO EVENT IS BMC, ITS AFFILIATES OR LICENSORS LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RELATING TO OR ARISING OUT OF THIS AGREEMENT, SUPPORT, AND/OR THE PRODUCT (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST COMPUTER USAGE TIME, AND DAMAGE OR LOSS OF USE OF DATA), EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND IRRESPECTIVE OF ANY NEGLIGENCE OF BMC OR WHETHER SUCH DAMAGES RESULT FROM A CLAIM ARISING UNDER TORT OR CONTRACT LAW.

LIMITS ON LIABILITY. BMC'S AGGREGATE LIABILITY FOR DAMAGES IS LIMITED TO THE AMOUNT PAID BY YOU FOR THE LICENSE TO THE PRODUCT.

SUPPORT. If Your order includes support for the Software, then BMC agrees to provide support (24 hours a day/7 days a week) ("**Support**"). You will be automatically re-enrolled in Support on an annual basis unless BMC receives notice of termination from You as provided below. There is a free support period during the one year warranty period.

(a) **Support Terms**. BMC agrees to make commercially reasonable efforts to provide the following Support: (i) For malfunctions of supported versions of the Software, BMC provides bug fixes, patches or workarounds in order to cause that copy of the Software to operate in substantial conformity with its then-current operating specifications; and (ii) BMC provides new releases or versions, so long as such new releases or versions are furnished by BMC to all other enrolled Support customers without additional charge. BMC may refuse to provide Support for any versions or releases of the Software other than the most recent version or release of such Software made available by BMC. Either party may terminate Your enrollment in Support upon providing notice to the other at least 30 days prior to the next applicable Support anniversary date. If You re-enroll in Support, BMC may charge You a reinstatement fee of 1.5 times what You would have paid if You were enrolled in Support during that time period.

(b) **Fees**. The annual fee for Support is 20% of the Software's list price less the applicable discount or a flat capacity based annual fee. BMC may change its prices for the Software and/or Support upon at least 30 days notice prior to Your support anniversary date.

VERIFICATION. If requested by BMC, You agree to deliver to BMC periodic written reports, whether generated manually or electronically, detailing Your use of the Software in accordance with this Agreement, including, without limitation, the License Capacity. BMC may, at its expense, audit Your use of the Software to confirm Your compliance with the Agreement. If an audit reveals that You have underpaid fees, You agree to pay such underpaid fees. If the underpaid fees exceed 5% of the fees paid, then You agree to also pay BMC's reasonable costs of conducting the audit.

EXPORT CONTROLS. You agree not to import, export, re-export, or transfer, directly or indirectly, any part of the Product or any underlying information or technology except in full compliance with all United States, foreign and other applicable laws and regulations.

GOVERNING LAW. This Agreement is governed by the substantive laws in force, without regard to conflict of laws principles: (a) in the State of New York, if you acquired the License in the United States, Puerto Rico, or any country in Central or South America; (b) in the Province of Ontario, if you acquired the License in Canada (subsections (a) and (b) collectively referred to as the "**Americas Region**"); (c) in Singapore, if you acquired the License in Japan, South Korea, Peoples Republic of China, Special Administrative Region of Hong Kong, Republic of China, Philippines, Indonesia, Malaysia, Singapore, India, Australia, New Zealand, or Thailand (collectively, "**Asia Pacific Region**"); or (d) in the Netherlands, if you acquired the License in any other country not described above. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed in its entirety.

ARBITRATION. ANY DISPUTE BETWEEN YOU AND BMC ARISING OUT OF THIS AGREEMENT OR THE BREACH OR ALLEGED BREACH, SHALL BE DETERMINED BY BINDING ARBITRATION CONDUCTED IN ENGLISH. IF THE DISPUTE IS INITIATED IN THE AMERICAS REGION, THE ARBITRATION SHALL BE HELD IN NEW YORK, U.S.A., UNDER THE CURRENT COMMERCIAL OR INTERNATIONAL, AS APPLICABLE, RULES OF THE AMERICAN ARBITRATION ASSOCIATION. IF THE DISPUTE IS INITIATED IN A COUNTRY IN THE ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN SINGAPORE, SINGAPORE UNDER THE CURRENT UNCITRAL ARBITRATION RULES. IF THE DISPUTE IS INITIATED IN A COUNTRY OUTSIDE OF THE AMERICAS REGION OR ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN AMSTERDAM, NETHERLANDS UNDER THE CURRENT UNCITRAL ARBITRATION RULES. THE COSTS OF THE ARBITRATION SHALL BE BORNE EQUALLY PENDING THE ARBITRATOR'S AWARD. THE AWARD RENDERED SHALL BE FINAL AND BINDING UPON THE PARTIES AND SHALL NOT BE SUBJECT TO APPEAL TO ANY COURT, AND MAY BE ENFORCED IN ANY COURT OF COMPETENT JURISDICTION. NOTHING IN THIS AGREEMENT SHALL BE DEEMED AS PREVENTING EITHER PARTY FROM SEEKING INJUNCTIVE RELIEF FROM ANY COURT HAVING JURISDICTION OVER THE PARTIES AND THE SUBJECT MATTER OF THE DISPUTE AS NECESSARY TO PROTECT EITHER PARTY'S CONFIDENTIAL INFORMATION, OWNERSHIP, OR ANY OTHER

PROPRIETARY RIGHTS. ALL ARBITRATION PROCEEDINGS SHALL BE CONDUCTED IN CONFIDENCE, AND THE PARTY PREVAILING IN ARBITRATION SHALL BE ENTITLED TO RECOVER ITS REASONABLE ATTORNEYS' FEES AND NECESSARY COSTS INCURRED RELATED THERETO FROM THE OTHER PARTY.

U.S. GOVERNMENT RESTRICTED RIGHTS. The Software under this Agreement is "commercial computer software" as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulations ("**FAR**") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("**DOD**"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 227.7202 of the DOD FAR Supplement and its successors.

MISCELLANEOUS TERMS. You agree to pay BMC all amounts owed no later than 30 days from the date of the applicable invoice, unless otherwise provided on the order for the License to the Products. You will pay, or reimburse BMC, for taxes of any kind, including sales, use, duty, tariffs, customs, withholding, property, value-added (VAT), and other similar federal, state or local taxes (other than taxes based on BMC's net income) imposed in connection with the Product and/or the Support. This Agreement constitutes the entire agreement between You and BMC and supersedes any prior or contemporaneous negotiations or agreements, whether oral, written or displayed electronically, concerning the Product and related subject matter. No modification or waiver of any provision hereof will be effective unless made in a writing signed by both BMC and You. You may not assign or transfer this Agreement or a License to a third party without BMC's prior written consent. Should any provision of this Agreement be invalid or unenforceable, the remainder of the provisions will remain in effect. The parties have agreed that this Agreement and the documents related thereto be drawn up in the English language. Les parties exigent que la présente convention ainsi que les documents qui s'y rattachent soient rédigés en anglais.

SW EULA Int 030102

The following Copyright notices apply to the DLL (dynamic link library) OSZLIBC:

Portions Copyright (c) 1990 The Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Portions Copyright (c) 1991 by AT&T. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR AT&T MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

Portions (c) Copyright 1986 HEWLETT-PACKARD COMPANY To anyone who acknowledges that this file is provided "AS IS" without any express or implied warranty: permission to use, copy, modify, and distribute this file for any purpose is hereby granted without fee, provided that the above copyright notice and this notice appears in all copies, and that the name of Hewlett-Packard Company not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose.

Portions Copyright (c) 1994, 1997 Cygnus Solutions. All rights reserved. Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed at Cygnus Solutions. Cygnus Solutions may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

All other registered trademarks or trademarks belong to their respective companies.

The following Copyright notice applies to the module OSZHTTPS:

The Apache Software License, Version 1.1

Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ”AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see.

Portions of this software are based upon public domain software originally written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign.

The following Copyright notice applies to the DLL OSZGCC2:

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author’s reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law. If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.) Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the

Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work

based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED-RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b), or DFARS 227.7202, as amended from time to time. Send any contract notices to

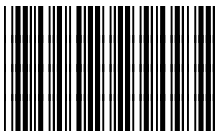
Contractor/Manufacturer:

BMC Software, Inc.

2101 CityWest Blvd.

Houston TX 77042-2827

USA



24189