

Oracle® Solaris Cluster Data Service for MaxDB Guide

Copyright © 2000, 2010, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related software documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	5
1 Installing and Configuring HA for MaxDB	11
HA for MaxDB Overview	11
Overview of the Installation and Configuration Process for HA for MaxDB	12
Planning the HA for MaxDB Installation and Configuration	13
Configuration Requirements	14
Supported Configurations of This Data Service	14
Configuration Considerations	16
Configuration Planning Questions	17
Installing and Configuring MaxDB	18
▼ How to Install and Configure MaxDB	18
▼ How to Enable MaxDB to Run in a Cluster	20
Verifying the MaxDB Installation and Configuration	22
▼ How to Verify MaxDB Installation and Configuration on Each Node	22
Installing the HA for MaxDB Packages	23
▼ How to Install the HA for MaxDB Packages	24
Configuring the HASToragePlus Resource Type to Work With HA for MaxDB	25
▼ How to Register and Configure an HASToragePlus Resource	26
Registering and Configuring HA for MaxDB	27
Setting HA for MaxDB Extension Properties	28
Administering SAP xserver as a User Other Than Root	28
▼ How to Register and Configure an SAP xserver Resource	28
▼ How to Register and Configure a MaxDB Resource	31
Tuning the HA for MaxDB Fault Monitors	34
Factors That Affect the Interval Between Fault Monitor Probes	35
Operations by the HA for MaxDB Fault Monitors During a Probe	35
Faults Detected by the HA for MaxDB Fault Monitors	35

Forcing the MaxDB Database Instance to Be Restarted if the Parent Kernel Process Is Terminated	36
Verifying the HA for MaxDB Installation and Configuration	37
▼ How to Verify the Operation of the MaxDB Fault Monitor	37
▼ How to Verify the Operation of the SAP xserver Fault Monitor	39
Upgrading the SUNW.sap_xserver Resource Type	40
Information for Registering the New Resource Type Version	40
Information for Migrating Existing Instances of the Resource Type	41
A HA for MaxDB Extension Properties	43
SUNW.sapdb Extension Properties	43
SUNW.sap_xserver Extension Properties	46
Index	49

Preface

Oracle Solaris Cluster Data Service for MaxDB Guide explains how to install and configure HA for MaxDB.

Note – This Oracle Solaris Cluster release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC, SPARC64, AMD64, and Intel 64. In this document, x86 refers to the larger family of 64-bit x86 compatible products. Information in this document pertains to all platforms unless otherwise specified.

This document is intended for system administrators with extensive knowledge of Oracle software and hardware. Do not use this document as a planning or presales guide. Before reading this document, you should have already determined your system requirements and purchased the appropriate equipment and software.

The instructions in this book assume knowledge of the Oracle Solaris Operating System and expertise with the volume-manager software that is used with Oracle Solaris Cluster software.

Using UNIX Commands

This document contains information about commands that are specific to installing and configuring Oracle Solaris Cluster data services. The document does *not* contain comprehensive information about basic UNIX commands and procedures, such as shutting down the system, booting the system, and configuring devices. Information about basic UNIX commands and procedures is available from the following sources:

- Online documentation for the Oracle Solaris Operating System
- Oracle Solaris Operating System man pages
- Other software documentation that you received with your system

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P-2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#

Related Documentation

Information about related Oracle Solaris Cluster topics is available in the documentation that is listed in the following table. All Oracle Solaris Cluster documentation is available at <http://docs.sun.com>.

Topic	Documentation
Data service administration	<i>Oracle Solaris Cluster Data Services Planning and Administration Guide</i> Individual data service guides
Concepts	<i>Oracle Solaris Cluster Concepts Guide</i>
Overview	<i>Oracle Solaris Cluster Overview</i>
Software installation	<i>Oracle Solaris Cluster Software Installation Guide</i>
System administration	<i>Oracle Solaris Cluster System Administration Guide</i>
Hardware administration	<i>Oracle Solaris Cluster 3.3 Hardware Administration Manual</i> Individual hardware administration guides
Data service development	<i>Oracle Solaris Cluster Data Services Developer's Guide</i>
Error messages	<i>Oracle Solaris Cluster Error Messages Guide</i>
Command and function reference	<i>Oracle Solaris Cluster Reference Manual</i>

For a complete list of Oracle Solaris Cluster documentation, see the release notes for your release of Oracle Solaris Cluster at <http://docs.sun.com>.

Related Third-Party Web Site References

Third-party URLs that are referenced in this document provide additional related information.

Note – Oracle is not responsible for the availability of third-party web sites mentioned in this document. Oracle does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Oracle will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

See the following web sites for additional resources:

- Documentation (<http://docs.sun.com>)
- Support (<http://www.oracle.com/us/support/systems/index.html>)
- Training (<http://education.oracle.com>) – Click the Sun link in the left navigation bar.

Oracle Welcomes Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of its documentation. If you find any errors or have any other suggestions for improvement, go to <http://docs.sun.com> and click Feedback. Indicate the title and part number of the documentation along with the chapter, section, and page number, if available. Please let us know if you want a reply.

Oracle Technology Network (<http://www.oracle.com/technetwork/index.html>) offers a range of resources related to Oracle software:

- Discuss technical problems and solutions on the [Discussion Forums](http://forums.oracle.com) (<http://forums.oracle.com>).
- Get hands-on step-by-step tutorials with [Oracle By Example](http://www.oracle.com/technology/obe/start/index.html) (<http://www.oracle.com/technology/obe/start/index.html>).
- Download [Sample Code](http://www.oracle.com/technology/sample_code/index.html) (http://www.oracle.com/technology/sample_code/index.html).

Getting Help

If you have problems installing or using Oracle Solaris Cluster, contact your service provider and provide the following information:

- Your name and email address (if available)
- Your company name, address, and phone number
- The model number and serial number of your systems
- The release number of the Oracle Solaris Operating System (for example, Oracle Solaris 10)
- The release number of Oracle Solaris Cluster (for example, Oracle Solaris Cluster 3.3)

Use the following commands to gather information about each node on your system for your service provider.

Command	Function
<code>prtconf -v</code>	Displays the size of the system memory and reports information about peripheral devices
<code>psrinfo -v</code>	Displays information about processors
<code>showrev -p</code>	Reports which patches are installed
<code>prtdiag -v</code>	Displays system diagnostic information
<code>/usr/cluster/bin/clnode show-rev</code>	Displays Oracle Solaris Cluster release and package version information

Also have available the contents of the `/var/adm/messages` file.

Installing and Configuring HA for MaxDB

This chapter explains how to install and configure HA for MaxDB. From version 7.5, SAP DB is distributed under the name MaxDB for MySQL (MaxDB). In this book, all versions of this database are referred to as MaxDB.

Note – If you are using the Solaris 10 OS, you can install and configure this data service to run in the non-global zone. HA for MaxDB is supported in non-global zones.

This chapter contains the following sections.

- “HA for MaxDB Overview” on page 11
- “Overview of the Installation and Configuration Process for HA for MaxDB” on page 12
- “Planning the HA for MaxDB Installation and Configuration” on page 13
- “Installing and Configuring MaxDB” on page 18
- “Verifying the MaxDB Installation and Configuration” on page 22
- “Installing the HA for MaxDB Packages” on page 23
- “Configuring the HAStoragePlus Resource Type to Work With HA for MaxDB” on page 25
- “Registering and Configuring HA for MaxDB” on page 27
- “Tuning the HA for MaxDB Fault Monitors” on page 34
- “Verifying the HA for MaxDB Installation and Configuration” on page 37
- “Upgrading the SUNW.sap_xserver Resource Type” on page 40

HA for MaxDB Overview

To eliminate single points of failure in an MaxDB system, HA for MaxDB provides the following features:

- Fault monitoring and automatic failover for the MaxDB application. You must configure HA for MaxDB as a failover data service.

- Fault monitoring and automatic restart for SAP xserver. You must configure SAP xserver as a scalable data service.

For conceptual information about failover data services and scalable data services, see the [Oracle Solaris Cluster Concepts Guide](#).

Each component of MaxDB has data service that protects the component when the component is configured in Oracle Solaris Cluster. See the following table.

Note – The files that are associated with the `SUNW.sap_xserver` resource type are supplied with the Oracle Solaris Cluster HA for SAP liveCache data service. The Oracle Solaris Cluster HA for SAP liveCache data service is installed when you install HA for MaxDB data service.

TABLE 1-1 Protection of MaxDB Components by Oracle Solaris Cluster Data Services

MaxDB Component	Data Service
MaxDB	HA for MaxDB The resource type is <code>SUNW.sapdb</code> .
SAP xserver	Oracle Solaris Cluster HA for SAP liveCache The resource type is <code>SUNW.sap_xserver</code> .
NFS file system	Oracle Solaris Cluster HA for NFS The resource type is <code>SUNW.nfs</code> . For more information about this data service, see Oracle Solaris Cluster Data Service for Network File System (NFS) Guide .

Overview of the Installation and Configuration Process for HA for MaxDB

The following table summarizes the tasks for installing and configuring HA for MaxDB and provides cross-references to detailed instructions for performing these tasks. Perform the tasks in the order that they are listed in the table.

TABLE 1-2 Tasks for Installing and Configuring HA for MaxDB

Task	Cross-Reference
Plan the HA for MaxDB installation and configuration	Your SAP documentation. <i>Oracle Solaris Cluster Data Services Planning and Administration Guide</i> “Planning the HA for MaxDB Installation and Configuration” on page 13
Install and configure MaxDB	“Installing and Configuring MaxDB” on page 18
Verify the MaxDB installation and configuration	“Verifying the MaxDB Installation and Configuration” on page 22
Install the HA for MaxDB packages	“Installing the HA for MaxDB Packages” on page 23
Configure the HAS _{torageP} lus resource to work with HA for MaxDB	“Relationship Between Resource Groups and Device Groups” in <i>Oracle Solaris Cluster Data Services Planning and Administration Guide</i> “Synchronizing the Startups Between Resource Groups and Device Groups” in <i>Oracle Solaris Cluster Data Services Planning and Administration Guide</i> “Configuring the HAS _{torageP} lus Resource Type to Work With HA for MaxDB” on page 25
Register and configure the HA for MaxDB data service	“Registering and Configuring HA for MaxDB” on page 27
(Optional) Tune the HA for MaxDB fault monitors	“Tuning the HA for MaxDB Fault Monitors” on page 34
Verify the HA for MaxDB installation and configuration	“Verifying the HA for MaxDB Installation and Configuration” on page 37
(Optional) Upgrade the SUNW.sap_xserver resource type	“Upgrading the SUNW.sap_xserver Resource Type” on page 40

Planning the HA for MaxDB Installation and Configuration

This section contains the information that you need to plan your HA for MaxDB installation and configuration.

Note – HA for MaxDB can be configured to run in a whole root or a sparse root non-global zone, if required.

Note – Before you begin, consult your MaxDB documentation for configuration restrictions and requirements that are not imposed by Oracle Solaris Cluster software. For information about restrictions that the Oracle Solaris Cluster software imposes, see the Oracle Solaris Cluster documentation.

Configuration Requirements

The configuration requirements in this section apply only to HA for MaxDB.



Caution – If your data service configuration does not conform to these requirements, the data service configuration might not be supported.

For requirements that apply to all data services, see “[Configuration Guidelines for Oracle Solaris Cluster Data Services](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

MaxDB Software Version Requirements

Use MaxDB version 7.4 or compatible versions.

HA for MaxDB Configuration Requirements

Configure HA for MaxDB as a failover data service. You cannot configure HA for MaxDB as a scalable data service. For more information, see the following sections:

- “[How to Enable MaxDB to Run in a Cluster](#)” on page 20
- “[How to Register and Configure a MaxDB Resource](#)” on page 31

SAP xserver Configuration Requirements

To enable client applications to access HA for MaxDB, you must use SAP xserver. Configure SAP xserver as a scalable data service. Do *not* configure SAP xserver as a failover data service.

Configure SAP xserver so that SAP xserver starts on all nodes to which the MaxDB resource can fail over. To implement this configuration, ensure that the node list of the SAP xserver resource group contains all nodes that are in the node list of the MaxDB resource group. For more information, see “[How to Register and Configure an SAP xserver Resource](#)” on page 28.

Supported Configurations of This Data Service

The HA for MaxDB data service supports configurations that conform to the requirements in “[Configuration Requirements](#)” on page 14.

If you plan to use MaxDB with other highly available SAP applications, you must also configure the Oracle Solaris Cluster data services for those applications. For more information, see the following table.

SAP Application	Oracle Solaris Cluster Data Service	Associated Document
SAP R/3	Oracle Solaris Cluster HA for SAP	Oracle Solaris Cluster Data Service for SAP Guide
SAP liveCache	Oracle Solaris Cluster HA for SAP liveCache	Oracle Solaris Cluster Data Service for SAP liveCache Guide
SAP Web Application Server	Oracle Solaris Cluster HA for SAP Web Application Server	Oracle Solaris Cluster Data Service for SAP Web Application Server Guide

The examples that follow show these supported configurations of HA for MaxDB:

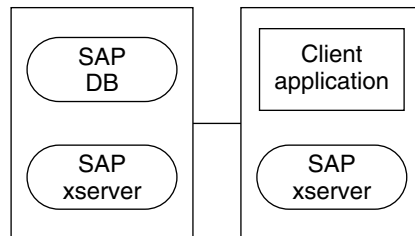
- Two-node cluster configuration
- Four-node cluster configuration with SAP R/3
- Four-node cluster configuration with SAP R/3 and SAP liveCache

Note – HA for MaxDB might support additional configurations. However, you must contact your Oracle service provider for information about additional configurations.

EXAMPLE 1-1 Two-Node Configuration

This example shows a two-node configuration in which a client application accesses the MaxDB resource through the SAP xserver resource. The characteristics of this configuration are as follows:

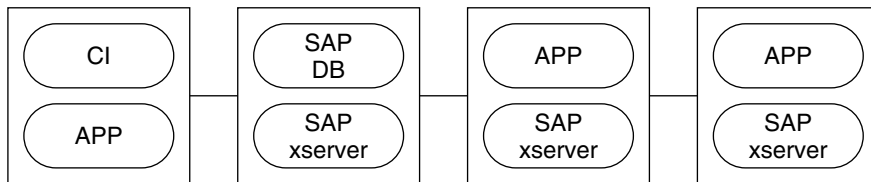
- The MaxDB resource is configured as a failover data service.
- The SAP xserver resource is configured as a scalable data service.



EXAMPLE 1-2 Four-Node Configuration With SAP R/3

This example shows a four-node configuration in which MaxDB is used with SAP R/3. This configuration uses multiple Advanced Planner & Optimizer (APO) application servers. The characteristics of this configuration are as follows:

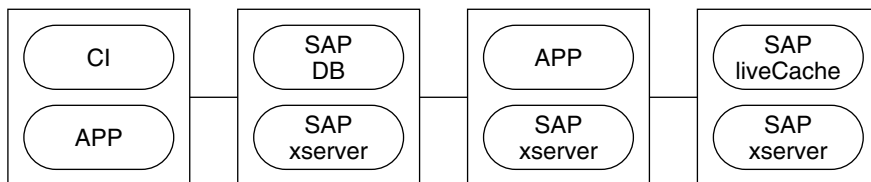
- The SAP APO Central Instance (CI) resource is configured as a failover data service.
- The MaxDB resource is configured as a failover data service.
- The SAP xserver resource is configured as a scalable data service.
- APO application server (APP) resources are configured as scalable data services.



EXAMPLE 1-3 Four-Node Configuration With SAP R/3 and SAP liveCache

This example shows a four-node configuration in which MaxDB is used with SAP R/3 and SAP liveCache. This configuration uses multiple APO application servers. The characteristics of this configuration are as follows:

- The SAP APO CI resource is configured as a failover data service.
- The MaxDB resource is configured as a failover data service.
- The SAP xserver resource is configured as a scalable data service.
- APP resources are configured as scalable data services.
- The SAP liveCache resource is configured as a failover data service.



Configuration Considerations

The configuration considerations in the subsections that follow affect the installation and configuration of HA for MaxDB.

Device Group for the MaxDB Application

Ensure that you create a device group for the MaxDB application as follows:

- Install MaxDB on its own global device group. For more information, see [“Installing and Configuring MaxDB” on page 18](#). This separate global device group for MaxDB ensures that the MaxDB resource can depend on the HASToragePlus resource only for MaxDB.
- Create an HASToragePlus resource for the global device group on which MaxDB is installed. For more information, see [“Configuring the HASToragePlus Resource Type to Work With HA for MaxDB” on page 25](#).
- Ensure that the resource for MaxDB depends on the HASToragePlus resource for the global device group on which MaxDB is installed. For more information, see [“Registering and Configuring HA for MaxDB” on page 27](#).

Dependencies of the MaxDB Application on SAP xserver

Configure MaxDB so that MaxDB starts only on a node where SAP xserver is running. To implement this configuration, configure resources and resource groups as follows:

- Ensure that the resource for MaxDB depends on the resource for SAP xserver.
- Create on the MaxDB resource group a strong positive affinity for the SAP xserver resource group.

For more information, see [“Registering and Configuring HA for MaxDB” on page 27](#).

Administration of SAP xserver by a User Other Than Root

You might be required to administer SAP xserver as a user other than root. In this situation, you must create and define that user as follows:

- You must create that user on all cluster nodes or zones that master SAP xserver.
- You must define that user when you register and configure HA for MaxDB. For more information, see [“Registering and Configuring HA for MaxDB” on page 27](#).

Configuration Planning Questions

Answer the questions in this section to plan the installation and configuration of HA for MaxDB. Write the answers to these questions in the space that is provided on the data service worksheets in [Appendix C, “Data Service Configuration Worksheets and Examples,” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*](#).

- Which resource group will you use for the MaxDB application resource and for the logical host name for the MaxDB resource?

Use the answer to this question when you perform the procedure [“How to Enable MaxDB to Run in a Cluster” on page 20](#).

- What is the logical host name for the MaxDB resource? Clients access the data service through this logical host name.

Use the answer to this question when you perform the following procedures:

- “How to Install and Configure MaxDB” on page 18
- “How to Enable MaxDB to Run in a Cluster” on page 20
- Where will the system configuration files reside?
See *Oracle Solaris Cluster Data Services Planning and Administration Guide* for the advantages and disadvantages of using the local file system instead of the cluster file system.

Installing and Configuring MaxDB

To enable HA for MaxDB to make MaxDB highly available, additional installation and configuration operations are required. These operations supplement the standard installation and standard configuration of MaxDB.

During a standard installation, MaxDB is installed with a physical host name. To enable MaxDB to run in a cluster, you must modify MaxDB to use a logical host name.

For information about the standard installation and standard configuration of MaxDB, see the following documentation:

- If you are using MaxDB with SAP R/3, see the SAP R/3 documentation for information about how to install and configure SAP R/3 with MaxDB.
- If you are using MaxDB independently of SAP R/3, see the MaxDB documentation.

▼ How to Install and Configure MaxDB

- 1 On one node of the cluster, install the MaxDB software.**
Ensure that you install MaxDB on its own global device group.
- 2 Perform a standard configuration of MaxDB.**
- 3 Create the .XUSER.62 file in the home directory of the operating system (OS) user who administers the MaxDB instance.**

Note – If you are using MaxDB 7.5 or MaxDB 7.6, the UNIX user identity of the OS user who administers the MaxDB database is `sdb`. You must specify this user in `DB_User` extension property when you create the MaxDB resource in a later step. Otherwise, the MaxDB fault monitor cannot probe the MaxDB database.

- a. **Create a plain text file that contains information about the database user who administers the MaxDB database instance.**

For information about the content of this file, see the MaxDB documentation. For the name of the server on which the database is running, specify the logical host name for the MaxDB resource that you specified in “Configuration Planning Questions” on page 17.

For an example of the content of this file, see [Example 1–4](#).

- b. **As MaxDB admin user, generate the `.XUSER.62` file from the plain text file that you created in [Step a](#).**

Use the MaxDB command `xuser` for this purpose.

```
# xuser -b user-info-file
```

`-b user-info-file` Specifies the plain text file from which you are generating the `.XUSER.62` file

- 4 **Copy the `/usr/spool/sql` directory and its contents from the node on which you installed MaxDB to all nodes where resources for MaxDB and SAP xserver will run.**

To ensure that the same owner owns the directory and its contents on all nodes, use the [tar\(1\)](#) command and the [rsh\(1\)](#)

```
# tar cfB - /usr/spool/sql | rsh destination tar xfB -
```

`destination` Specifies the node to which you are copying the `/usr/spool/sql` directory and its contents

The following example shows a plain text file that contains information about a database user who administers an MaxDB instance.

- 5 **If you are installing MaxDB 7.5 or MaxDB 7.6, perform this step. If you are installing a previous version, proceed to the next step.**

Copy the `/etc/opt/sdb` directory and its contents from the node on which you installed MaxDB to all nodes where resources for MaxDB and SAP xserver will run.

To ensure that the same owner owns the directory and its contents on all nodes, use the [tar\(1\)](#) command and the [rsh\(1\)](#)

```
# tar cfB - /etc/opt/sdb | rsh destination tar xfB -
```

`destination` Specifies the node to which you are copying the `/etc/opt/sdb` directory and its contents

- 6 **If you are using MaxDB version earlier than 7.6.03.09, comment out `/net` in `/etc/auto_master` file and remove `nis` from the automount entry in `/etc/nsswitch.conf` file.**

Example 1–4 Information About a Database User Who Administers a MaxDB Instance

```

DEFAULT
dbm
dbm
TST
svr-1
  blank line
  blank line
-1
-1
  blank line

```

This example specifies the following information about a database user who administers an MaxDB instance:

- The user key that is used to address this combination of XUSER data is named DEFAULT.
- The user name and password of the database user are dbm.
- The name of the MaxDB instance is TST.
- The logical host name for the MaxDB resource is svr-1.
- No structured query language (SQL) mode is specified.
- The default time-out value of the MaxDB instance is used.
- The default isolation level of the MaxDB instance is used.

For more information, see the MaxDB documentation.

▼ How to Enable MaxDB to Run in a Cluster

- 1 **Create a failover resource group to contain the MaxDB application resources and the logical host name for the MaxDB resource.**

Use the resource group that you identified when you answered the questions in “[Configuration Planning Questions](#)” on page 17.

```
# clresourcegroup create -n node-zone-list sapdb-rg
```

sapdb-rg Specifies that the resource group that you are creating is named *sapdb-rg*.

-n node-zone-list Specifies a comma-separated, ordered list of zones that can master this resource group. The format of each entry in the list is *node*. In this format, *node* specifies the node name and *zone* specifies the name of a non-global Solaris zone. To specify the global zone, or to specify a node without non-global zones, specify only *node*.

This list is optional. If you omit this list, the global zone of each cluster node can master the resource group.

- 2 Ensure that all network resources that you intend to you use are added to your name service database.

- 3 Add a logical host name resource to the failover resource group that you created in [Step 1](#).

```
# clreslogicalhostname create -g sapdb-rg \  
-h sapdb-logical-hostname sapdb-logical-hostname  
  
-g sapdb-rg
```

Specifies that the logical host name's database resource is to be added to the failover resource group that you created in [Step 1](#).

```
-h
```

Specifies the host name list. You must use this option either when more than one logical host needs to be associated with the new MaxDB resource or when the logical host does not have the same name as the MaxDB resource itself. MaxDB is the resource for the *sapdb-logical-hostname* that you specified in “[Configuration Planning Questions](#)” on [page 17](#).

```
sapdb-logical-hostname
```

Specifies the logical host name of the server on which the database is running. This host name must be the logical host name for the MaxDB resource that you specified in “[Configuration Planning Questions](#)” on [page 17](#).

Note – If you require a fully qualified hostname, you must specify the fully qualified name with the `-h` option and you cannot use the fully qualified form in the resource name.

- 4 Enable the resource group that you created in [Step 1](#).

```
# clresourcegroup online -emM sapdb-rg
```

```
-emM    Enables the resource group sapdb-rg created in Step 1 and moves it to the MANAGED  
state.
```

Example 1–5 Enabling MaxDB to Run in a Cluster

This example shows the sequence of commands that are required to enable MaxDB to run in a cluster. The commands are run on only one cluster node.

1. The following command creates a failover resource group to contain the MaxDB application resources and the logical host name for the MaxDB resource. The resource group is named *sapdbrg*. The *sapdbrg* resource group can be brought online on all cluster nodes or zones.

```
# clresourcegroup create sapdbrg
```

2. The following command adds a logical host name resource to the *sapdbrg* resource group. The logical host name of the server on which the database is running is *srvr-1*. When the *sapdbrg* resource group is brought online, an attempt is made for each node to discover a network interface on the subnet that the host name list identifies.

```
# clreslogicalhostname create -g sapdbrg -h sdrvr-1 -d sapdbl
```

Note – If you require a fully qualified hostname, you must specify the fully qualified name with the `-h` option and you cannot use the fully qualified form in the resource name.

3. The following command moves the `sapdbrg` resource group to the `MANAGED` state, brings the resource group online, and monitors the resources in the resource group.

```
# clresourcegroup online -emM sapdbrg
```

Verifying the MaxDB Installation and Configuration

Before you install the HA for MaxDB packages, verify that the MaxDB software is correctly installed and configured to run in a cluster. This verification does *not* verify that the MaxDB application is highly available because the HA for MaxDB data service is not yet installed.

▼ How to Verify MaxDB Installation and Configuration on Each Node

Perform this procedure on each node or zone that can master the MaxDB resource group.

- 1 Log in as superuser to a node or zone that can master the MaxDB resource group.
- 2 Switch the MaxDB resource group to the node that you logged in to in [Step 1](#).

```
# clresourcegroup switch -n node sapdb-rg
```

`-n node` Specifies the node to which the MaxDB resource group is to be switched

`sapdb-rg` Specifies that the MaxDB resource group `sapdb-rg` is to be switched to another node

- 3 Confirm that the MaxDB database can be started and be stopped.

- a. Become the OS user who administers the MaxDB database.

```
# su - os-sapdb-adm-user
```

`os-sapdb-adm-user` Specifies the UNIX user identity of the OS user who administers the MaxDB database. This user's home directory contains the `.XUSER.62` file that was created during the installation and configuration of MaxDB. For more information, see [“Installing and Configuring MaxDB” on page 18](#). You specify this user when you set

the `DB_User` extension property as explained in [“How to Register and Configure a MaxDB Resource”](#) on page 31.

If you are using MaxDB 7.5 or MaxDB 7.6, the UNIX user identity of the OS user who administers the MaxDB database is `sdb`.

b. Start the SAP xserver.

```
$ x_server start
```

c. Manually start the MaxDB database instance on the node that you logged in to in [Step 1](#).

```
$ dbmcli -U sapdb-adm-key db_online
```

`-U sapdb-adm-key` Specifies that the `dbmcli` command is run with the user key of the database user who administers the MaxDB instance. This user key is created during the installation and configuration of MaxDB. For more information, see [“Installing and Configuring MaxDB”](#) on page 18. You specify this user key when you set the `User_Key` extension property as explained in [“How to Register and Configure a MaxDB Resource”](#) on page 31.

d. Confirm that the MaxDB database instance is started.

e. Manually stop the MaxDB database instance.

```
$ dbmcli -U sapdb-adm-key db_offline
```

`-U sapdb-adm-key` Specifies that the `dbmcli` command is run with the user key that you used for starting the database in [Step c](#)

f. Confirm that the MaxDB database instance is stopped.

Installing the HA for MaxDB Packages

If you did not install the HA for MaxDB packages during your initial Oracle Solaris Cluster installation, perform this procedure to install the packages. To install the packages, use the `installer` program.

Note – You need to install the HA for MaxDB packages in the global cluster and not in the zone cluster.

▼ How to Install the HA for MaxDB Packages

Perform this procedure on each cluster node where you are installing the HA for MaxDB packages.

You can run the `installer` program with a command-line interface (CLI) or with a graphical user interface (GUI). The content and sequence of instructions in the CLI and the GUI are similar.

Note – Even if you plan to configure this data service to run in non-global zones, install the packages for this data service in the global zone. The packages are propagated to any existing non-global zones and to any non-global zones that are created after you install the packages.

Before You Begin Ensure that you have the Oracle Solaris Cluster installation media.

If you intend to run the `installer` program with a GUI, ensure that your `DISPLAY` environment variable is set.

1 On the cluster node where you are installing the data service packages, become superuser.

2 Load the Oracle Solaris Cluster installation media into the DVD-ROM drive.

If the Volume Management daemon `vold(1M)` is running and configured to manage DVD-ROM devices, the daemon automatically mounts the DVD-ROM on the `/cdrom` directory.

3 Change to the installation wizard directory of the DVD-ROM.

- **If you are installing the data service packages on the SPARC platform, type the following command:**

```
# cd /cdrom/cdrom0/Solaris_sparc
```

- **If you are installing the data service packages on the x86 platform, type the following command:**

```
# cd /cdrom/cdrom0/Solaris_x86
```

4 Start the installation wizard.

```
# ./installer
```

5 When you are prompted, accept the license agreement.

6 From the list of Oracle Solaris Cluster agents under Availability Services, select the data service for MaxDB.

- 7 If you require support for languages other than English, select the option to install multilingual packages.**
English language support is always installed.
- 8 When prompted whether to configure the data service now or later, choose Configure Later.**
Choose Configure Later to perform the configuration after the installation.
- 9 Follow the instructions on the screen to install the data service packages on the node.**
The installation wizard displays the status of the installation. When the installation is complete, the wizard displays an installation summary and the installation logs.
- 10 (GUI only) If you do not want to register the product and receive product updates, deselect the Product Registration option.**
The Product Registration option is not available with the CLI. If you are running the installation wizard with the CLI, omit this step.
- 11 Exit the installation wizard.**
- 12 Unload the installation media from the DVD-ROM drive.**
 - a. To ensure that the DVD-ROM is not being used, change to a directory that does *not* reside on the DVD-ROM.**
 - b. Eject the DVD-ROM.**

```
# eject cdrom
```

Next Steps Go to [“Configuring the HASStoragePlus Resource Type to Work With HA for MaxDB”](#) on page 25.

Configuring the HASStoragePlus Resource Type to Work With HA for MaxDB

For maximum availability of the MaxDB database, resources that HA for MaxDB requires must be available before the MaxDB database instance is started. An example of such a resource is the file system where programs and libraries for the MaxDB runtime environment reside. To ensure that these resources are available, configure the HASStoragePlus resource type to work with HA for MaxDB.

For information about the relationship between resource groups and disk device groups, see “Relationship Between Resource Groups and Device Groups” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

Configuring the HASStoragePlus resource type to work with HA for MaxDB involves the following operation:

- Registering and configuring an HASStoragePlus resource

▼ How to Register and Configure an HASStoragePlus Resource

Perform this procedure on any one cluster node.

1 Register the SUNW.HASStoragePlus resource type.

```
# clresourcetype register SUNW.HASStoragePlus
```

2 Create an HASStoragePlus resource for the global device group on which MaxDB is installed.

Create this resource in the MaxDB resource group. This resource must perform an affinity switchover for all global devices that are defined for this resource.

```
# clresource create -d -g sapdb-rg \  
-t SUNW.HASStoragePlus -p filesystemmountpoints=mountpoint-list \  
-p globaldevicepaths=sapdb-device-group \  
-p affinityon=TRUE hsp-resource
```

-d

Specifies that a new resource is to be created in a disabled state.

-g *sapdb-rg*

Specifies that the resource is to be added to the MaxDB resource group.

-t SUNW.HASStoragePlus

Specifies that the resource is an instance of the SUNW.HASStoragePlus resource type.

-p *filesystemmountpoints=mountpoint-list*

Specifies a list of valid mount points for the file system.

-p *globaldevicepaths=sapdb-device-group*

Specifies the name of the global device group on which the MaxDB software is installed.

-p affinityon=TRUE

Specifies that this resource performs an affinity switchover for all global devices that are defined for this resource.

hsp-resource

Specifies that the resource that you are creating is named *hsp-resource*.

The resource is created in the enabled state.

Example 1-6 Creating an HAStoragePlus Resource

```
# clresource create -g sapdbrg \  
-t SUNW.HAStoragePlus -p filesystemmountpoints=/global/sapdbdata \  
-p globaldevicepaths=sapdbdg -p affinityon=TRUE hsprs
```

This example shows the creation of a SUNW.HAStoragePlus resource that has the following characteristics:

- The resource is named hsprs.
- The resource is a member of a resource group that is named sapdbrg. The creation of this resource group is shown in [Example 1-5](#).
- The resource is an instance of the SUNW.HAStoragePlus resource type. The registration of this resource type is not shown in this example.
- The mount point for the file system is /global/sapdbdata.
- The MaxDB software is installed on a global device group that is named sapdbdg.
- The hsprs resource performs an affinity switchover for all global devices that are defined for this resource.

Next Steps Go to [“Registering and Configuring HA for MaxDB” on page 27](#).

Registering and Configuring HA for MaxDB

To enable HA for MaxDB to make MaxDB highly available, configure Oracle Solaris Cluster data services as follows:

- Configure SAP xserver as a scalable data service.
- Configure HA for MaxDB as a failover data service.

Before you perform this procedure, ensure that the HA for MaxDB data service packages are installed.



Caution – One SAP xserver serves multiple MaxDB instances and, if SAP liveCache is used, multiple SAP liveCache instances in the cluster. Therefore, do *not* configure more than one SAP xserver resource on the same cluster. If more than one SAP xserver resource runs on the same cluster, conflicts between the SAP xserver resources occur. These conflicts cause all SAP xserver resources to become unavailable. If you attempt to start the SAP xserver a second time, the attempt fails. The error message `Address already in use` is also displayed.

Setting HA for MaxDB Extension Properties

The sections that follow contain instructions for registering and configuring resources. These instructions explain how to set *only* extension properties that HA for MaxDB requires you to set. For information about all HA for MaxDB extension properties, see [Appendix A, “HA for MaxDB Extension Properties.”](#) You can update some extension properties dynamically. You can update other properties, however, only when you create or disable a resource. The Tunable entry indicates when you can update a property.

To set an extension property of a resource, include the following option to modify the resource:

```
-p property=value
-p property      Identifies the extension property that you are setting
value           Specifies the value to which you are setting the extension property
```

You can also use the procedures in [Chapter 2, “Administering Data Service Resources,”](#) in *Oracle Solaris Cluster Data Services Planning and Administration Guide* to configure resources after the resources are created.

Administering SAP xserver as a User Other Than Root

You might be required to administer SAP xserver as a user other than root. In this situation, you must create and define that user as follows:

- You must create that user on all cluster nodes or zones that master SAP xserver.
- You must define that user when you register and configure an SAP xserver resource. To define the user who administers SAP xserver, set the Xserver_User extension property when you create an SAP xserver resource. For more information about the Xserver_User extension property, see [“SUNW.sap_xserver Extension Properties”](#) on page 46.

▼ How to Register and Configure an SAP xserver Resource

- 1 Become superuser on a cluster node.
- 2 Register the SUNW.sap_xserver resource type.


```
# clresourcetype register SUNW.sap_xserver
```

3 Create a scalable resource group for the SAP xserver resource.

Configure SAP xserver so that SAP xserver starts on all nodes to which the MaxDB resource can fail over. To implement this configuration, ensure that the node list of the SAP xserver resource group contains all nodes that are in the node list of the MaxDB resource group. This resource group is created when the procedure “[How to Enable MaxDB to Run in a Cluster](#)” on page 20 is performed.

```
# clresourcegroup create -n node-zone-list \
-p Maximum primaries=nodes-in-sapdb-rg \
-p Desired primaries=nodes-in-sapdb-rg xserver-rg
```

-n *node-zone-list*

Specifies a comma-separated, ordered list of zones that can master this resource group. The format of each entry in the list is *node*. In this format, *node* specifies the node name and *zone* specifies the name of a non-global Solaris zone. To specify the global zone, or to specify a node without non-global zones, specify only *node*.

This list is optional. If you omit this list, the global zone of each cluster node can master the resource group.

-p *Maximum primaries=nodes-in-sapdb-rg*

Specifies the maximum number of nodes on which the SAP xserver resource can start. This number is equal to the number of nodes that are in the node list of the MaxDB resource group. You must specify the same number as the value of the *Desired primaries* property.

-p *Desired primaries=nodes-in-sapdb-rg*

Specifies the desired number of nodes on which the SAP xserver resource can start. This number is equal to the number of nodes that are in the node list of the MaxDB resource group. You must specify the same number as the value of the *Maximum primaries* property.

xserver-rg

Specifies that the resource group that you are creating is named *xserver-rg*.

4 Add the HASStoragePlus resource to the SAP xserver resource group.

```
# clresource create -g xserver-rg \
-t SUNW.HASStoragePlus \
-p filesystemmountpoints=mountpoints \
-p affinityon=false xserver-storage-resource
```

-g *xserver-rg*

Specifies that the resource that you are creating is added to the resource group *xserver-rg*.

-t *SUNW.HASStoragePlus*

Specifies that the resource type of the resource you are creating is *SUNW.HASStoragePlus*.

-p *filesystemmountpoints=mountpoint, ...*

Specifies the mount points for the resource group you are creating.

-p *affinityon=false*

Indicates that the SAP xserver resource does not have to be colocated with the device group.

xserver-storage-resource

Specifies that the resource that you are creating is named *xserver-storage-resource*.

The resource is created in the enabled state.

For more details on how to set up an HASStoragePlus resource, see “Enabling Highly Available Local File Systems” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

5 Create an SAP xserver resource in the resource group that you created in Step 3.

```
# clresource create -d -g xserver-rg \  
-t SUNW.sap_xserver \  
-p resource_dependencies=xserver-storage-resource xserver-resource
```

-g *xserver-rg*

Specifies that the resource is to be added to the resource group that you created in [Step 3](#)

-t *SUNW.sap_xserver*

Specifies that the resource is an instance of the *SUNW.sap_xserver* resource type

-p *resource_dependencies=xserver-storage-resource*

Sets a resource dependency between *xserver-storage-resource* and the HASStoragePlus resource we created in [Step 4](#).

xserver-resource

Specifies that the resource that you are creating is named *xserver-resource*.

The resource is created in the enabled state.

6 Enable the resource group that you created in Step 3.

```
# clresourcegroup online -emM xserver-rg
```

-emM Enables the resource group and moves it to the MANAGED state.

xserver-rg Specifies the name of the resource group.

Example 1-7 Configuring a SUNW.sap_xserver Resource

This example shows the sequence of commands that are required to configure an SAP xserver resource. The commands are run on only one cluster node.

1. The following command creates a scalable resource group to contain an SAP xserver resource for a four-node cluster. The resource group is named *xsvrrg*. The *xsvrrg* resource group can be brought online on all cluster nodes.

```
# clresourcegroup create \  
-p Maximum primaries=4 \  
-p Desired primaries=4 xsvrrg
```

2. The following command adds the HASStoragePlus resource to the SAP xserver resource group. The *xserver-storage-resource* is called *xsstorrs* and the mount point, *mtp1*.

```
# clresource create -d -g xsrvrrg \
-t SUNW.HASStoragePlus \
-p filesystemmountpoints=mp1 \
-p affinityon=false xsstorrs
```

The resource is created in the enabled state.

3. The following command creates an SAP xserver resource that is named `xsvrrs` in the `xsrvrrg` resource group. The SAP xserver resource is an instance of the `SUNW.sap_xserver` resource type. The `xserver-storage-resource` is `xsstorrs`. The registration of this resource type is not shown in this example.

```
# clresource create -d -g xsrvrrg -t SUNW.sap_xserver \
-p resource_dependencies=xsstorrs xsvrrs
```

The resource is created in the enabled state.

4. The following command moves the `xsrvrrg` resource group to the `MANAGED` state and brings the resource group online.

```
# clresourcegroup online -emM xsrvrrg
```

▼ How to Register and Configure a MaxDB Resource

- 1 Register the `SUNW.sapdb` resource type.

```
# clresourcetype register SUNW.sapdb
```

- 2 Create a MaxDB resource in the MaxDB resource group.

Ensure that the MaxDB resource depends on the following resources:

- The `HASStoragePlus` resource for the global device group on which MaxDB is installed.
- The SAP xserver resource.

When you create this resource, specify the following information about the MaxDB database instance. This information is created when MaxDB is installed and configured as explained in [“Installing and Configuring MaxDB” on page 18](#).

- The name of the MaxDB database.
- The UNIX user identity of the OS user who administers the MaxDB database instance. If you are using MaxDB 7.5 or MaxDB 7.6, the UNIX user identity of the OS user who administers the MaxDB database is `sdb`.
- The user key of the database user who administers the MaxDB database instance.

```
# clresource create -d -g sapdb-rg \
-t SUNW.sapdb \
-p DB_Name=db-name \
-p DB_User=os-sapdb-adm-user \
-p User_Key=sapdb-adm-key \
```

- p **resource_dependencies**=*hsp-resource , xserver-resource sapdb-rs*
 - d
Specifies that the resource that you are creating is not immediately enabled.
 - g *sapdb-rg*
Specifies that the resource is to be added to the MaxDB resource group.
 - t *SUNW . sapdb*
Specifies that the resource is an instance of the *SUNW . sapdb* resource type.
 - p **DB_Name**=*db-name*
Specifies the name of the MaxDB database instance in uppercase.
 - p **DB_User**=*os-sapdb-adm-user*
Specifies the UNIX user identity of the OS user who administers the MaxDB database. This user's home directory contains the *.XUSER . 62* file that was created during the installation and configuration of MaxDB. For more information, see [“Installing and Configuring MaxDB” on page 18](#).
 - p **User_Key**=*sapdb-adm-key*
Specifies the user key of the database user who administers the MaxDB database instance. This user key is created during the installation and configuration of MaxDB. For more information, see [“Installing and Configuring MaxDB” on page 18](#).
 - p **resource_dependencies**=*hsp-resource , xserver-resource*
Specifies that the MaxDB resource depends on the following resources.
 - The *HAStoragePlus* resource for the global device group on which MaxDB is installed
 - The *SAP xserver* resource
 - sapdb-rs*
Specifies that the resource that you are creating is named *sapdb-rs*.
- The resource is created in the enabled state.

3 Ensure that the MaxDB resource group is brought online only on a node or zone where the SAP xserver resource group is online.

To meet this requirement, create on the MaxDB resource group a strong positive affinity for the SAP xserver resource group.

- ```
clresourcegroup set -p rg_affinities=++xserver-rg sapdb-rg
```
- p **rg\_affinities**=++*xserver-rg*  
Specifies that the MaxDB resource group declares a strong positive affinity for the SAP xserver resource group.
  - sapdb-rg*  
Specifies that the MaxDB resource group is to be modified.



**4 Enable the MaxDB resource group.**

```
#clresourcegroup onLine -emM sapdb-rg
```

-emM Enables the MaxDB resource group and moves it to the MANAGED state.

*sapdb-rg* Specifies the name of the MaxDB resource group to be brought online.

**5 (Optional) Consider configuring your cluster to prevent noncritical resource groups from being brought online on the same node or zone as the MaxDB resource group.**

You might plan to run lower-priority services on a node to which the MaxDB resource can fail over. In this situation, consider using resource group affinities to shut down the noncritical services when the MaxDB resource fails over to the node.

To specify this behavior, declare on the resource group for each noncritical service a strong negative affinity for the MaxDB resource group.

```
clresourcegroup set -p rg_affinities=--sapdb-rg noncritical-rg
```

-p

Sets the resource group property to the specified value.

*noncritical-rg*

Specifies the name of the noncritical resource group.

**Example 1–8 Creating a SUNW.sapdb Resource**

```
clresource create -d -g sapdbrg -t SUNW.sapdb \
-p DB_Name=TST -p DB_User=dbadmin \
-p User_Key=DEFAULT -p resource_dependencies=hsprs,xsrvrrs sapdbrs
```

The resource is created in the enabled state.

This example shows the creation of a SUNW.sapdb resource that has the following characteristics:

- The resource is named sapdbrs.
- The resource is a member of a resource group that is named sapdbrg. The creation of this resource group is shown in [Example 1–5](#).
- The resource is an instance of the SUNW.sapdb resource type. The registration of this resource type is not shown in this example.
- The MaxDB database instance that is associated with this resource is named TST.
- The UNIX user identity of the OS user who administers the MaxDB database is dbadmin.
- The user key of the database user who administers the MaxDB database is DEFAULT.
- The MaxDB resource depends on the following resources:
  - An HASStoragePlus resource that is named hsprs. The creation of the hsprs resource is shown in [Example 1–6](#).

- A `SUNW.sap_xserver` resource that is named `xsvrrrs`. The creation of the `xsvrrrs` resource is shown in [Example 1-7](#).

## Tuning the HA for MaxDB Fault Monitors

Fault monitoring for the HA for MaxDB data service is provided by the following fault monitors:

- The MaxDB fault monitor
- The SAP xserver fault monitor

Each fault monitor is contained in a resource whose resource type is shown in the following table.

TABLE 1-3 Resource Types for HA for MaxDB Fault Monitors

| Fault Monitor | Resource Type                 |
|---------------|-------------------------------|
| MaxDB         | <code>SUNW.sapdb</code>       |
| SAP xserver   | <code>SUNW.sap_xserver</code> |

System properties and extension properties of these resources control the behavior of the fault monitors. The default values of these properties determine the preset behavior of the fault monitors. The preset behavior should be suitable for most Oracle Solaris Cluster installations. Therefore, you should tune the HA for MaxDB fault monitors *only* if you need to modify this preset behavior.

Tuning the HA for MaxDB fault monitors involves the following tasks:

- Setting the interval between fault monitor probes
- Setting the timeout for fault monitor probes
- Defining the criteria for persistent faults
- Specifying the failover behavior of a resource

For more information, see “[Tuning Fault Monitors for Oracle Solaris Cluster Data Services](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*. Information about the HA for MaxDB fault monitors that you need to perform these tasks is provided in the subsections that follow.

HA for MaxDB also enables you to control how the fault monitor responds if the MaxDB parent kernel process is not running. For more information, see “[Forcing the MaxDB Database Instance to Be Restarted if the Parent Kernel Process Is Terminated](#)” on page 36.

Tune the HA for MaxDB fault monitors when you register and configure HA for MaxDB. For more information, see “[Registering and Configuring HA for MaxDB](#)” on page 27.

## Factors That Affect the Interval Between Fault Monitor Probes

To determine whether SAP xserver and the MaxDB database instance are operating correctly, the HA for MaxDB fault monitors probe these resources periodically. The optimum interval between fault monitor probes depends on the time that is required to respond to a fault in a resource. This time depends on how the complexity of the resource affects the time that is required for operations such as restarting the resource.

For example, SAP xserver is a much simpler resource and can be restarted much quicker than MaxDB. Therefore, the optimum interval between fault monitor probes of SAP xserver is shorter than the optimum interval between probes of MaxDB.

## Operations by the HA for MaxDB Fault Monitors During a Probe

The optimum timeout for fault monitor probes depends on the operations that a fault monitor performs to probe the resource.

### Operations by the MaxDB Fault Monitor During a Probe

During a probe, the MaxDB fault monitor performs the following operations:

1. The MaxDB fault monitor determines whether the MaxDB database instance is online.
2. If the MaxDB database instance is online, the MaxDB fault monitor determines whether the parent kernel process of the MaxDB database instance is running. You can control how the fault monitor responds if the parent kernel process is not running. For more information, see [“Forcing the MaxDB Database Instance to Be Restarted if the Parent Kernel Process Is Terminated”](#) on page 36.
3. The MaxDB fault monitor determines whether SAP xserver is available. This fault monitoring supplements the fault monitoring that the SAP xserver fault monitor provides.

### Operations by the SAP xserver Fault Monitor During a Probe

During a probe, the SAP xserver fault monitor determines whether SAP xserver is available.

## Faults Detected by the HA for MaxDB Fault Monitors

Faults that each HA for MaxDB fault monitor detects are described in the subsections that follow.

## Faults Detected by the MaxDB Fault Monitor

The MaxDB fault monitor detects the following faults in MaxDB:

- A status of the MaxDB database instance that is not **ONLINE**, for example, **OFFLINE** or **ADMIN**
- Unexpected termination of the parent kernel process of the MaxDB database instance

The MaxDB fault monitor also detects the unavailability of SAP xserver. This fault monitoring supplements the fault monitoring that the SAP xserver fault monitor provides.

---

**Note** – If the MaxDB fault monitor detects that SAP xserver is unavailable twice within the retry interval, the MaxDB fault monitor restarts MaxDB. By restarting MaxDB, the fault monitor ensures that the MaxDB database fails over to another node when SAP xserver is persistently unavailable.

---

## Faults Detected by the SAP xserver Fault Monitor

The SAP xserver fault monitor detects following faults:

- **Unavailability of SAP xserver.** Unavailability of SAP xserver is also detected by the MaxDB fault monitor.
- **Persistent system errors.** A persistent system error is a system error that occurs four times within the retry interval. If a persistent system error occurs, the fault monitor restarts SAP xserver.

## Recovery Actions in Response to Detected Faults

To minimize the disruption that transient faults in a resource cause, a fault monitor restarts the resource in response to such faults. For persistent faults, more disruptive action than restarting the resource is required:

- For the MaxDB resource, the fault monitor fails over the resource to another node. The MaxDB resource is a failover resource.
- For the SAP xserver resource, the fault monitor takes the resource offline. The SAP xserver is a scalable resource.

## Forcing the MaxDB Database Instance to Be Restarted if the Parent Kernel Process Is Terminated

By default, unexpected termination of the parent kernel process does *not* cause the MaxDB fault monitor to restart the MaxDB database instance. The MaxDB database instance can continue to function without the parent kernel process. Restarting the MaxDB database instance in this situation might cause unnecessary unavailability of the MaxDB database instance. Therefore,

you should force the MaxDB database instance to be restarted *only* if you require a feature that the parent kernel process provides. An example of such a feature is maintaining the integrity of the log history.

To force the MaxDB database instance to be restarted if the parent kernel process is terminated, set the `Restart_if_Parent_Terminated` extension property of the `SUNW.sapdb` resource to `True`.

## Verifying the HA for MaxDB Installation and Configuration

After you install, register, and configure HA for MaxDB, verify the HA for MaxDB installation and configuration. Verifying the HA for MaxDB installation and configuration determines if the HA for MaxDB data service makes the MaxDB application highly available.

Verifying the HA for MaxDB installation involves verifying the operation of the following fault monitors:

- The MaxDB fault monitor
- The SAP xserver fault monitor

### ▼ How to Verify the Operation of the MaxDB Fault Monitor

Perform this procedure on each node where MaxDB can run.

- 1 Log in as superuser to a node or zone that can master the MaxDB resource group.
- 2 Switch the MaxDB resource group to the node or zone that you logged in to in [Step 1](#).

```
clresourcegroup switch -n node sapdb-rg
```

*node* Specifies the node to which the MaxDB resource group is to be switched

*sapdb-rg* Specifies the name of the MaxDB resource group is to be switched to another node

- 3 Abnormally terminate MaxDB.
  - a. Determine the process identities of all kernel processes for the MaxDB database instance that you are running.

```
ps -ef | grep kernel | grep db-name
```

*db-name* Specifies the name of the MaxDB database instance in uppercase

- b. Kill all kernel processes for the MaxDB database instance that you are running.

```
kill -9 sapdb-kernel-pid
```

*sapdb-kernel-pid* Specifies the process identities of the MaxDB kernel processes that you determined in [Step a](#)

**4 Confirm that the HA for MaxDB fault monitor performs the appropriate operation from the following list:**

- Restarting the MaxDB resource
- Failing over the MaxDB resource to another node

The expected behavior of the fault monitor depends on the failure history of the resource and the value of the `FailOver_enabled` extension property. For more information, see the following sections:

- [“Tuning the HA for MaxDB Fault Monitors” on page 34](#)
- [“SUNW.sapdb Extension Properties” on page 43](#)

**5 Terminate MaxDB normally.**

**a. Become the OS user who administers the MaxDB database.**

```
su - os-sapdb-adm-user
```

*os-sapdb-adm-user* Specifies the UNIX user identity of the OS user who administers the MaxDB database. This user's home directory contains the `.XUSER.62` file that was created during the installation and configuration of MaxDB. For more information, see [“Installing and Configuring MaxDB” on page 18](#). You specify this user when you set the `DB_User` extension property as explained in [“How to Register and Configure a MaxDB Resource” on page 31](#). If you are using MaxDB 7.5 or MaxDB 7.6, the UNIX user identity of the OS user who administers the MaxDB database is `sdb`.

**b. Manually stop the MaxDB database instance.**

```
dbmcli -U sapdb-adm-key db_offline
```

*-U sapdb-adm-key* Specifies that the `dbmcli` command is run with the user key of the database user who administers the MaxDB database. This user key is created during the installation and configuration of MaxDB. For more information, see [“Installing and Configuring MaxDB” on page 18](#). You specify this user key when you set the `User_Key` extension property as explained in [“How to Register and Configure a MaxDB Resource” on page 31](#).

**6 Confirm that the HA for MaxDB fault monitor performs the appropriate operation from the following list:**

- Restarting the MaxDB resource

- Failing over the MaxDB resource to another node

The expected behavior of the fault monitor depends on the failure history of the resource and the value of the `Fai Lover_enabled` extension property. For more information, see the following sections:

- [“Tuning the HA for MaxDB Fault Monitors” on page 34](#)
- [“SUNW.sapdb Extension Properties” on page 43](#)

## ▼ How to Verify the Operation of the SAP xserver Fault Monitor

Perform this procedure on each node that can master SAP xserver.

- 1 Log in to a node or zone that can master SAP xserver.

- 2 Abnormally terminate SAP xserver.
  - a. Determine the process identities of all SAP xserver processes.
 

```
ps -ef | grep vserver
```
  - b. Kill all SAP xserver processes.
 

```
kill -9 xserver-pid
```

*xserver-pid* Specifies the process identities of the SAP xserver processes that you determined in [Step a](#)

- 3 Confirm that the SAP xserver fault monitor restarts the SAP xserver resource.

- 4 Terminate SAP xserver normally.
  - a. Become the OS user who administers SAP xserver.
 

```
su - os-sapxsrvr-adm-user
```

*os-sapxsrvr-adm-user* Specifies the UNIX user identity of the OS user who administers SAP xserver. By default, this user is root. You can specify this user by setting the `Xserver_User` extension property. For more information, see [“SUNW.sap\\_xserver Extension Properties” on page 46](#).
  - b. Manually stop the SAP xserver.
 

```
x_server stop
```

- 5 Confirm that the SAP xserver fault monitor restarts the SAP xserver resource.

## Upgrading the SUNW.sap\_xserver Resource Type

The SUNW.sap\_xserver resource type is supplied with the Oracle Solaris Cluster HA for SAP liveCache data service. The Oracle Solaris Cluster HA for SAP liveCache data service is installed when you install HA for MaxDB data service. Upgrade the SUNW.sap\_xserver resource type if all conditions in the following list apply:

- You are using version 3.0 5/02 asynchronous release, version 3.1 5/03, version 3.1 10/03, or 3.1 4/04 of the Oracle Solaris Cluster HA for SAP liveCache data service.
- You plan to use the HA for MaxDB data service with your existing version of the Oracle Solaris Cluster HA for SAP liveCache data service.
- You need to use the new features of the SUNW.sap\_xserver resource type.

For general instructions that explain how to upgrade a resource type, see “[Upgrading a Resource Type](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*. The information that you need to complete the upgrade of the SUNW.sap\_xserver resource type is provided in the subsections that follow.

## Information for Registering the New Resource Type Version

The relationship between a resource type version and the release of Oracle Solaris Cluster data services is shown in the following table. The release of Oracle Solaris Cluster data services indicates the release in which the version of the resource type was introduced.

| Resource Type Version | Oracle Solaris Cluster Data Services Release |
|-----------------------|----------------------------------------------|
| 1.0                   | 3.0 5/02 asynchronous release                |
| 2                     | 3.1 4/04                                     |
| 3.1                   | 3.2                                          |

To determine the version of the resource type that is registered, use the `clresource list` command.

The resource type registration (RTR) file for this resource type is `/opt/SUNWscld/xserver/etc/SUNW.sap_xserver`.



## Information for Migrating Existing Instances of the Resource Type

The information that you need to migrate instances of the SUNW.sap\_xserver resource type is as follows:

- You can perform the migration only when the resource is unmonitored.
- If you need to use the new features of the SUNW.sap\_xserver resource type, the required value of the `Type_version` property is 2.
- If you need to specify the directory that contains programs and libraries for the SAP xserver runtime environment, set the `Independent_Program_Path` extension property. For more information, see “[SUNW.sap\\_xserver Extension Properties](#)” on page 46.

The following example shows a command for editing an instance of the SUNW.sap\_xserver resource type.

**EXAMPLE 1-9** Editing an Instance of the SUNW.sap\_xserver Resource Type During Upgrade

```
clresource set -p Independent_Program_Path=/sapdb/indep_prog \
-p Type_version=2 sapxserver-rs
```

This command edits a SUNW.sap\_xserver resource as follows:

- The SUNW.sap\_xserver resource is named `sapxserver-rs`.
- The `Type_version` property of this resource is set to 2.
- The independent program path is `/sapdb/indep_prog`.



## HA for MaxDB Extension Properties

---

Extension properties for HA for MaxDB resource types are described in the following sections.

- “[SUNW.sapdb Extension Properties](#)” on page 43
- “[SUNW.sap\\_xserver Extension Properties](#)” on page 46

For details about system-defined properties, see the [r\\_properties\(5\)](#) man page and the [rg\\_properties\(5\)](#) man page.

### SUNW.sapdb Extension Properties

The SUNW.sapdb resource type represents the MaxDB application in a Oracle Solaris Cluster configuration. The extension properties of this resource type are as follows:

#### dbmcli\_Start\_Option

The option that is passed to the `dbmcli` command to start the MaxDB database instance.

---

**Note** – For MaxDB version 7.4.3, set this property to `db_online`.

---

|                  |                        |
|------------------|------------------------|
| <b>Data type</b> | String                 |
| <b>Default</b>   | <code>db_online</code> |
| <b>Range</b>     | Not applicable         |
| <b>Tunable</b>   | When disabled          |

#### DB\_Name

The name of the MaxDB database instance in uppercase. This name is created when MaxDB is installed and configured as explained in “[Installing and Configuring MaxDB](#)” on page 18.

|                  |                    |
|------------------|--------------------|
| <b>Data type</b> | String             |
| <b>Default</b>   | No default defined |

**Range** Not applicable

**Tunable** When disabled

#### DB\_User

The UNIX user identity of the operating system (OS) user who administers the MaxDB database instance. This user's home directory contains the .XUSER.62 file that was created during the installation and configuration of MaxDB. For more information, see [“Installing and Configuring MaxDB” on page 18](#).

**Data type** String

**Default** No default defined

**Range** Not applicable

**Tunable** When disabled

#### Failover\_enabled

Specifies whether the fault monitor fails over the MaxDB resource if the number of attempts to restart exceeds `Retry_count` within the time that `Retry_interval` specifies. The possible values of this extension property are as follows:

- `True` – Specifies that the fault monitor fails over the MaxDB resource
- `False` – Specifies that the fault monitor does *not* fail over the MaxDB resource

**Data type** Boolean

**Default** `True`

**Range** Not applicable

**Tunable** Any time

#### Independent\_Program\_Path

The full path to the directory that contains the following programs and libraries for the MaxDB application:

- Programs that are independent of the database software version
- Libraries for the client runtime environment

HA for MaxDB determines the path to the `dbmccli` command from the value of this property. The `dbmccli` command resides in the `bin` subdirectory of the directory that this property specifies.

**Data type** String

**Default** `/sapdb/programs`

**Range** Not applicable

**Tunable** When disabled

**Monitor\_retry\_count**

The maximum number of restarts by the process monitor facility (PMF) that are allowed for the fault monitor.

|                  |                  |
|------------------|------------------|
| <b>Data type</b> | Integer          |
| <b>Default</b>   | 4                |
| <b>Range</b>     | No range defined |
| <b>Tunable</b>   | Any time         |

**Monitor\_retry\_interval**

The period of time in minutes during which the PMF counts restarts of the fault monitor.

|                  |                  |
|------------------|------------------|
| <b>Data type</b> | Integer          |
| <b>Default</b>   | 2                |
| <b>Range</b>     | No range defined |
| <b>Tunable</b>   | Any time         |

**Pid\_dir\_Path**

The full path to the directory under which files that store the process identities of MaxDB kernel processes are created. The process identities of MaxDB kernel processes are stored in the following files:

- *pid-dir/ppid/db-name*
- *pid-dir/pid/db-name*

The replaceable items in these file paths are as follows:

- *pid-dir* is the directory that the `Pid_Dir_Path` extension property specifies
- *db-name* is the name of the MaxDB database instance that the `DB_Name` extension property specifies

|                  |                             |
|------------------|-----------------------------|
| <b>Data type</b> | String                      |
| <b>Default</b>   | <code>/var/spool/sql</code> |
| <b>Range</b>     | Not applicable              |
| <b>Tunable</b>   | When disabled               |

**Probe\_timeout**

The timeout value in seconds that the fault monitor uses to probe an MaxDB database instance.

|                  |           |
|------------------|-----------|
| <b>Data type</b> | Integer   |
| <b>Default</b>   | 90        |
| <b>Range</b>     | 30–99,999 |

**Tunable** Any time

#### Restart\_if\_Parent\_Terminated

Determines whether the fault monitor restarts the MaxDB database instance if the parent kernel process is terminated. The possible values of this extension property are as follows:

- True – Specifies that the fault monitor restarts the MaxDB database instance if the parent kernel process is terminated
- False – Specifies that the fault monitor does *not* restart the MaxDB database instance if the parent kernel process is terminated

**Data type** Boolean

**Default** False

**Range** Not applicable

**Tunable** Any time

#### User\_Key

The user key of the database user who administers the MaxDB database instance. This user key is created when MaxDB is installed and configured as explained in [“Installing and Configuring MaxDB” on page 18](#).

**Data type** String

**Default** No default defined

**Range** Not applicable

**Tunable** When disabled

## SUNW.sap\_xserver Extension Properties

The SUNW.sap\_xserver resource type represents SAP xserver in a Oracle Solaris Cluster configuration. The extension properties of this resource type are as follows:

#### Confdir\_List

The full path to the directory that contains the MaxDB software and MaxDB database instance.

**Data type** String

**Default** /sapdb

**Range** Not applicable

**Tunable** At creation

**Independent\_Program\_Path**

The full path to the directory that contains the following programs and libraries for SAP xserver:

- Programs that are independent of the database software version
- Libraries for the client runtime environment

HA for MaxDB determines the path to the `x_server` command from the value of this property. The `x_server` command resides in the `bin` subdirectory of the directory that this property specifies.

|                              |                    |
|------------------------------|--------------------|
| <b>Data type</b>             | String             |
| <b>Default</b>               | No default defined |
| <b>Range</b>                 | Not applicable     |
| <b>Tunable</b>               | When disabled      |
| <b>Introduced in release</b> | 3.1 4/04           |

**Monitor\_retry\_count**

The maximum number of restarts by the PMF that are allowed for the fault monitor.

|                  |                  |
|------------------|------------------|
| <b>Data type</b> | Integer          |
| <b>Default</b>   | 4                |
| <b>Range</b>     | No range defined |
| <b>Tunable</b>   | Any time         |

**Monitor\_retry\_interval**

The period of time in minutes during which the PMF counts restarts of the fault monitor.

|                  |                  |
|------------------|------------------|
| <b>Data type</b> | Integer          |
| <b>Default</b>   | 2                |
| <b>Range</b>     | No range defined |
| <b>Tunable</b>   | Any time         |

**Probe\_timeout**

The timeout value in seconds for fault monitor probes.

|                  |                  |
|------------------|------------------|
| <b>Data type</b> | Integer          |
| <b>Default</b>   | 120              |
| <b>Range</b>     | No range defined |
| <b>Tunable</b>   | Any time         |

**Soft\_Stop\_Pct**

The percentage of the Stop method timeout that is used to stop SAP xserver by using the SAP utility `x_server stop`. If this timeout is exceeded, the SIGKILL signal is used to stop all SAP xserver processes.

|                  |               |
|------------------|---------------|
| <b>Data type</b> | Integer       |
| <b>Default</b>   | 50            |
| <b>Range</b>     | 1–100         |
| <b>Tunable</b>   | When disabled |

**Xserver\_User**

The UNIX user identity of the OS user who administers SAP xserver.

|                  |                |
|------------------|----------------|
| <b>Data type</b> | String         |
| <b>Default</b>   | root           |
| <b>Range</b>     | Not applicable |
| <b>Tunable</b>   | At creation    |



# Index

---

## A

- Address already in use error message, 27
- affinities
  - resource groups
    - creating, 32
    - planning, 17
- affinity switchover, 26

## C

- caution notice, multiple SAP xserver resources, 27
- `clnode` command, 9
- clusters, running MaxDB in, 20–22
- commands, node information, 8
- `Confdir_List` extension property, `SUNW.sap_xserver` resource type, 46
- configuring
  - device groups
    - performing, 25–27
    - planning, 17
  - file systems
    - performing, 25–27
    - planning, 17
  - HA for MaxDB
    - overview, 12–13
    - performing, 27–34
    - planning, 13–18
  - HASStoragePlus resource
    - performing, 25–27
    - planning, 17
  - MaxDB application, 18–22

## creating

- HASStoragePlus resource, 26
- MaxDB resource, 31
- MaxDB resource group, 20
- SAP xserver resource, 30
- SAP xserver resource group, 29
- `.XUSER.62` file, 18

## D

- database instance, *See* MaxDB database instance
- database users
  - identity of OS user, 44
  - sample definition, 20
  - user key for, 46
- `DB_Name` extension property, 43
- `DB_User` extension property, 44
- `dbmcli` command
  - path to, 44
  - start option, 43
- `dbmcli_Start_Option` extension property, 43
- defining
  - MaxDB instance user, 18
  - SAP xserver user
    - planning for nondefault, 17
    - specifying nondefault, 28
- dependencies
  - device groups
    - creating, 31
    - planning, 17

dependencies (*Continued*)

- MaxDB resource
  - creating, 31
  - planning, 17

## device groups

- configuring
  - performing, 25–27
  - planning, 17
- installation of MaxDB, 18
- MaxDB application, 17

## directories

- See also* paths
- /usr/spool/sql, 19

**E**

editing, resource type instances, 41

## enabling

- MaxDB resource group, 33
- SAP xserver resource group, 30

## extension properties

- See also* properties
- See also* system properties
- SUNW.sap\_xserver resource type, 46–48
- SUNW.sapdb resource type, 43–46

**F**

Failover\_enabled extension property, SUNW.sapdb resource type, 44

## fault monitors

- faults detected by, 35–36
- MaxDB
  - faults detected, 36
  - resource type for, 34
  - verifying operation of, 37–39
- probe interval, 35
- probe timeout, 35
- response to faults, 35–36
- SAP xserver
  - faults detected, 36
  - resource type for, 34
  - verifying operation of, 39

fault monitors (*Continued*)

- tuning, 34–37

## faults

- recovery actions, 36
- responses to, 35–36

## file systems

- configuration
  - performing, 25–27
  - planning, 17
- protection by data services, 12

## files

- process identity, 45
- RTR, 40
- system configuration, 18

**G**

global zone, 11, 24

**H**

## HA for MaxDB

- overview, 11–12
- configuration
  - performing, 27–34
  - planning, 13–18
  - with HAStoragePlus resource, 25–27
- fault monitors, 34–37
- installing, 23–25
  - verifying installation, 37–39
- SAP liveCache application, use with, 14–16
- SAP R/3 application, use with, 14–16
- service configuration, 14
- software packages, installing, 23–25

## HAStoragePlus resource

- configuration
  - performing, 25–27
  - planning, 17
- dependency of MaxDB resource on
  - creating, 31
  - planning, 17
- MaxDB device group, 17
- resource group for, 26

help, 8–9

## I

Independent\_Program\_Path extension property  
     SUNW.sap\_xserver resource type, 46  
     SUNW.sapdb resource type, 44  
 installing  
     HA for MaxDB, 23–25  
         overview, 12–13  
         verifying installation, 37–39  
     MaxDB application, 18–22  
 intervals, fault monitor probes, 35

## K

kernel processes  
     identities, 45  
     termination of parent  
         extension property, 46  
         restarting MaxDB after, 36–37

## L

libraries  
     MaxDB  
         SUNW.sap\_xserver resource type, 46  
         SUNW.sapdb resource type, 44  
     SAP xserver, 47  
 local zones, *See* non-global zones  
 locations, *See* paths  
 log history, preserving, 36–37  
 logical host names  
     resource group for  
         creating, 20  
         enabling, 22  
         planning, 17  
     resources  
         creating, 21  
         planning, 18

## M

MaxDB application  
     configuring, 18–22  
     device group for, 17  
     fault monitor, 34  
     installing, 18–22  
     paths to programs and libraries  
         SUNW.sap\_xserver resource type, 46  
         SUNW.sapdb resource type, 44  
     processes  
         paths to process identity files, 45  
         termination of parent, 36–37, 46  
     protection by data services, 12  
     resource group for  
         creating, 20  
         enabling, 22  
         planning, 17  
     running in cluster, 20–22  
     software versions, 14  
     termination of parent kernel process, 36–37  
     verifying installation and configuration, 22–23  
 MaxDB database instance  
     database user, 46  
     name, 43  
     operating system user, 44  
     starting  
         dbmcli command options, 43  
         MaxDB verification, 23  
     stopping, 23  
 MaxDB resource  
     creating, 31  
     dependency on other resources  
         creating, 31  
         planning, 17  
 MaxDB resource group  
     affinity for SAP xserver resource group  
         creating, 32  
         planning, 17  
 maximum values  
     nodes for SAP xserver, 29  
 restarts  
     SUNW.sap\_xserver resource type, 47  
     SUNW.sapdb resource type, 45  
 messages file, 9

migrating, resource type instances, 41

Monitor\_retry\_count extension property  
SUNW.sap\_xserver resource type, 47  
SUNW.sapdb resource type, 44

Monitor\_retry\_interval extension property  
SUNW.sap\_xserver resource type, 47  
SUNW.sapdb resource type, 45

**N**

names, MaxDB database instance, 43

network addresses, *See* logical host names

NFS file system, protection by data services, 12

nodes, for SAP xserver and MaxDB, 14

non-global zones, 11, 24

**O**

operating system users  
of MaxDB database instance  
defining, 18  
of MaxDB database instance  
extension property for, 44  
of MaxDB database instance  
sample definition, 20  
of SAP xserver  
defining, 17  
extension property for, 48

Oracle Solaris Cluster HA for SAP liveCache, relation to  
SUNW.sap\_xserver resource type, 12

overriding, default SAP xserver users, 28

**P**

packages, 23–25

parent kernel processes  
extension property, 46  
restarting MaxDB after termination of, 36–37

paths  
dbmcli command, 44  
MaxDB programs and libraries  
SUNW.sap\_xserver resource type, 46

paths (*Continued*)  
MaxDB programs and libraries  
SUNW.sapdb resource type, 44  
process identity files, 45  
SAP xserver programs and libraries, 47  
x\_server command, 47

persistent faults, recovery actions, 36

Pid\_Dir\_Path extension property, 45

Probe\_timeout extension property  
SUNW.sap\_xserver resource type, 47  
SUNW.sapdb resource type, 45

processes  
identities, 45  
termination of parent  
extension property, 46  
restarting MaxDB after, 36–37

programs  
MaxDB  
SUNW.sap\_xserver resource type, 46  
SUNW.sapdb resource type, 44  
SAP xserver, 47

properties  
*See also* extension properties  
*See also* system properties  
Type\_version, 41

protection, MaxDB application, 12

prtconf -v command, 9

prtdiag -v command, 9

psrinfo -v command, 9

**R**

recovery actions, after faults, 36

registering  
SUNW.HASStoragePlus resource type, 26–27  
SUNW.sap\_xserver resource type  
during initial set up, 28–31  
during upgrade, 40  
SUNW.sapdb resource type, 31–34

resource groups  
HASStoragePlus resource, 26  
logical host name  
creating, 20  
enabling, 22

- resource groups (*Continued*)
    - logical host names
      - planning, 17
    - MaxDB
      - creating, 20
      - enabling, 22
      - planning, 17
    - SAP xserver
      - creating, 29
      - enabling, 30
      - planning, 14
  - resource type registration (RTR) file, 40
  - resource types
    - fault monitors, 34
    - migrating instances of, 41
    - SUNW.HASStoragePlus
      - See also* HASStoragePlus resource
      - registering, 26–27
    - SUNW.nfs, 12
    - SUNW.sap\_xserver
      - extension properties, 46–48
      - initial registration, 28–31
      - instantiating, 30
      - protection by, 12
      - registration during upgrade, 40
      - relation to Oracle Solaris Cluster HA for SAP
        - liveCache, 12
      - upgrading, 40–41
    - SUNW.sapdb
      - extension properties, 43–46
      - initial registration, 31–34
      - instantiating, 31
      - protection by, 12
  - resources
    - logical host name
      - creating, 21
      - planning, 18
    - MaxDB
      - creating, 31
    - SAP xserver
      - creating, 30
  - responses, to faults, 35–36
  - Restart\_if\_Parent\_Terminated extension property
    - SUNW.sapdb resource type, 46
  - Restart\_if\_Parent\_Terminated extension property (*Continued*)
    - tuning, 37
  - restarts
    - interval between
      - SUNW.sap\_xserver resource type, 47
      - SUNW.sapdb resource type, 45
    - maximum allowed
      - SUNW.sap\_xserver resource type, 47
      - SUNW.sapdb resource type, 45
  - restrictions
    - zones, 11, 24
  - RTR (resource type registration) file, 40
- S**
- SAP liveCache application, use with HA for
    - MaxDB, 14–16
  - SAP R/3 application, use with HA for MaxDB, 14–16
  - SAP xserver
    - administration by nondefault users
      - planning, 17
      - specifying, 28
    - caution notice, 27
    - fault monitor, 34
    - multiple instances of, 27
    - nodes for, 14
    - operating system user, 48
    - paths to programs and libraries, 47
    - protection by data services, 12
    - resource group for
      - creating, 29
      - enabling, 30
      - planning, 14
    - service configuration, 14
    - starting, 23
    - stopping
      - command for, 39
      - time allowed for, 48
    - upgrading, 40–41
  - SAP xserver resource, 30
    - dependency of MaxDB resource on
      - creating, 31

SAP xserver resource (*Continued*)

- dependency of MaxDB resource on
  - planning, 17
- SAP xserver resource group
  - affinity of MaxDB resource group for
    - creating, 32
    - planning, 17
- show-rev subcommand, 9
- showrev -p command, 9
- SIGKILL signal, 48
- Soft\_Stop\_Pct extension property, 47
- software packages, 23–25
- starting
  - MaxDB database instance
    - dbmcli command options, 43
    - MaxDB verification, 23
  - SAP xserver, 23
- stopping
  - MaxDB database instance, 23
  - SAP xserver
    - command for, 39
    - time allowed for, 48
- SUNW.HASStoragePlus resource type
  - See also* HASStoragePlus resource
  - registering, 26–27
- SUNW.nfs resource type, 12
- SUNW.sap\_xserver resource type
  - extension properties, 46–48
  - instantiating, 30
  - protection by, 12
  - registering
    - during initial setup, 28–31
    - during upgrade, 40
  - relation to Oracle Solaris Cluster HA for SAP
    - liveCache, 12
  - resource type versions, 40
  - upgrading, 40–41
- SUNW.sapdb resource type
  - extension properties, 43–46
  - instantiating, 31
  - protection by, 12
  - registering, 31–34
- system configuration files, location, 18
- system properties, effect on fault monitors, 34

**T**

- technical support, 8–9
- timeouts
  - fault monitor
    - guidelines for setting, 35
    - SUNW.sap\_xserver resource type, 47
    - SUNW.sapdb resource type, 45
  - stop method, 48
- transient faults, recovery actions, 36
- tuning, fault monitors, 34–37
- Type\_version property, 41

**U**

- upgrading, SAP xserver, 40–41
- User\_Key extension property, 46
- users
  - of MaxDB database instance
    - database, 46
    - operating system, 44
  - of MaxDB database instance
    - sample definition, 20
  - of SAP xserver, 48
    - planning for nondefault, 17
    - specifying nondefault, 28
- /usr/spool/sql directory, 19

**V**

- /var/adm/messages file, 9
- verifying, MaxDB application, 22–23
- version requirements, MaxDB application, 14
- versions, resource types, 40

**X**

- x\_server command, path to, 47
- Xserver\_User extension property
  - specifying, 28
  - SUNW.sap\_xserver resource type, 48
- .XUSER.62 file
  - creating, 18

.XUSER.62 file (*Continued*)

and DB\_User extension property, 44  
example, 20

## **Z**

zones, 11, 24

