

Oracle® Solaris Cluster Data Services Reference Manual

Copyright © 2000, 2011, 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 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 that 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.

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. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

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.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

Contents

Preface	5
OSC40DS 4	9
custom_action_file(4)	10
OSC40DS 5	17
ORCL.obiee_cluster_controller(5)	18
ORCL.obiee_presentation_service(5)	20
ORCL.obiee_scheduler(5)	22
ORCL.obiee_server(5)	24
SUNW.apache(5)	26
SUNW.asm_diskgroup(5)	31
SUNW.dns(5)	39
SUNW.iws(5)	43
SUNW.jsas(5)	48
SUNW.jsas-na(5)	52
SUNW.krb5(5)	56
SUNW.ldom(5)	59
SUNW.nfs(5)	63
SUNW.oracle_listener(5)	67
SUNW.oracle_server(5)	69
SUNW.slas(5)	74
SUNW.sap_as(5)	78
SUNW.sap_ci(5)	81
SUNW.sapdb(5)	84
SUNW.sapeng(5)	89
SUNW.sap_livecache(5)	95

SUNW.saprepl(5)	98
SUNW.sapscs(5)	104
SUNW.sapwebas(5)	110
SUNW.sap_xserver(5)	116
SUNW.sblgtwy(5)	119
SUNW.sblsrvr(5)	121
SUNW.scalable_asm_diskgroup_proxy(5)	123
SUNW.scalable_asm_instance(5)	127
SUNW.scalable_asm_instance_proxy(5)	134
SUNW.scalable_rac_listener(5)	139
SUNW.scalable_rac_server(5)	144
SUNW.scalable_rac_server_proxy(5)	152
SUNW.sybase(5)	161
SUNW.wls(5)	165
Index	171

Preface

The *Oracle Solaris Cluster Data Services Reference Manual* provides reference information about resource types for data services that are supplied with Oracle Solaris Cluster software. This book is intended for experienced system administrators with extensive knowledge of Oracle software and hardware. This book is not to be used as a planning or presales guide. The information in this book assumes knowledge of the Solaris Operating System and expertise with the volume manager software that is used with Oracle Solaris Cluster software.

Both novice users and those familiar with the Solaris Operating System can use online man pages to obtain information about their SPARC based system or x86 based system and its features.

A man page is intended to answer concisely the question “What does this command do?” The man pages in general comprise a reference manual. They are not intended to be a tutorial.

Note – Oracle Solaris Cluster software runs on two platforms, SPARC and x86. The information in this book pertains to both platforms unless otherwise specified in a special chapter, section, note, bulleted item, figure, table, or example.

Overview

The following contains a brief description of each man page section and the information it references:

- Section 4 outlines the formats of various files. The C structure declarations for the file formats are given where applicable.
- Section 5 contains miscellaneous Oracle Solaris Cluster documentation such as descriptions of resource types.

The following is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. For example, if no bugs can be reported, no BUGS section is included. See the `int ro` pages for more information and detail about each section, and `man(1)` for general information about man pages.

NAME	This section gives the names of the commands or functions that are documented, followed by a brief description of what they do.
SYNOPSIS	<p>This section shows the syntax of commands or functions. If a command or file does not exist in the standard path, its full path name is shown. Options and arguments are alphabetized, with single-letter arguments first, and options with arguments next, unless a different argument order is required.</p> <p>The following special characters are used in this section:</p> <ul style="list-style-type: none">[] Brackets. The option or argument that is enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.. . . Ellipses. Several values can be provided for the previous argument, or the previous argument can be specified multiple times, for example, “filename . . .”. Separator. Only one of the arguments separated by this character can be specified at a time.{ } Braces. The options and/or arguments enclosed within braces are interdependent. All characters within braces must be treated as a unit.
PROTOCOL	This section occurs only in subsection 3R and indicates the protocol description file.
DESCRIPTION	This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. DESCRIPTION does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, and functions are described under USAGE.
IOCTL	This section appears on pages in Section 7 only. Only the device class that supplies appropriate parameters to the <code>ioctl(2)</code> system call is called <code>ioctl</code> and generates its own heading. <code>ioctl</code> calls for a specific device are listed alphabetically (on the man page for that specific device).

	<p><code>ioctl</code> calls are used for a particular class of devices. All these calls have an <code>io</code> ending, such as <code>mtio(7I)</code>.</p>
OPTIONS	<p>This section lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.</p>
OPERANDS	<p>This section lists the command operands and describes how they affect the actions of the command.</p>
OUTPUT	<p>This section describes the output – standard output, standard error, or output files – generated by the command.</p>
RETURN VALUES	<p>If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or -1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions that are declared void do not return values, so they are not discussed in RETURN VALUES.</p>
ERRORS	<p>On failure, most functions place an error code in the global variable <code>errno</code> that indicates why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.</p>
USAGE	<p>This section lists special rules, features, and commands that require in-depth explanations. The subsections that are listed here are used to explain built-in functionality:</p> <ul style="list-style-type: none">CommandsModifiersVariablesExpressionsInput Grammar
EXAMPLES	<p>This section provides examples of usage or of how to use a command or function. Wherever possible, a complete</p>

	<p>example, which includes command-line entry and machine response, is shown. Whenever an example is given, the prompt is shown as <code>example%</code>, or if the user must be superuser, <code>example#</code>. Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS, and USAGE sections.</p>
ENVIRONMENT VARIABLES	<p>This section lists any environment variables that the command or function affects, followed by a brief description of the effect.</p>
EXIT STATUS	<p>This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion, and values other than zero are returned for various error conditions.</p>
FILES	<p>This section lists all file names that are referred to by the man page, files of interest, and files created or required by commands. Each file name is followed by a descriptive summary or explanation.</p>
ATTRIBUTES	<p>This section lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. See attributes(5) for more information.</p>
SEE ALSO	<p>This section lists references to other man pages, in-house documentation, and outside publications.</p>
DIAGNOSTICS	<p>This section lists diagnostic messages with a brief explanation of the condition that caused the error.</p>
WARNINGS	<p>This section lists warnings about special conditions that could seriously affect your working conditions. WARNINGS is not a list of diagnostics.</p>
NOTES	<p>This section lists additional information that does not belong anywhere else on the page. NOTES covers points of special interest to the user. Critical information is never covered here.</p>
BUGS	<p>This section describes known bugs and, wherever possible, suggests workarounds.</p>

R E F E R E N C E

OSC40DS 4

Name custom_action_file – file that defines custom behavior of fault monitors for HA Oracle server resources and Oracle 9i RAC server resources

Description A custom action file is a plain text file. The file contains one or more entries that define the custom behavior of fault monitors for the following resources:

- **HA Oracle server resources.** These resources are instances of the [SUNW.oracle_server\(5\)](#) resource type.
- **Oracle 9i Real Application Clusters (RAC) server resources.** These resources are instances of the [SUNW.scalable_rac_server\(5\)](#) resource type.

Each entry defines the custom behavior for a single database management system (DBMS) error, a single timeout error, or several logged alerts. A maximum of 1024 entries is allowed in a custom action file.

Note – Each entry in a custom action file overrides the preset action for an error, or specifies an action for an error for which no action is preset. Create entries in a custom action file *only* for the preset actions that you are overriding or for errors for which no action is preset. Do *not* create entries for actions that you are not changing.

An entry in a custom action file consists of a sequence of keyword-value pairs that are separated by semicolons. Each entry is enclosed in braces.

The format of an entry in a custom action file is as follows:

```
{
[ERROR_TYPE=DBMS_ERROR|SCAN_LOG|TIMEOUT_ERROR;]
ERROR=error-spec;
[ACTION=SWITCH|RESTART|STOP|NONE;]
[CONNECTION_STATE=co|di|on|*;]
[NEW_STATE=co|di|on|*;]
[MESSAGE="message-string"]
}
```

White space may be used between separated keyword-value pairs and between entries to format the file.

The meaning and permitted values of the keywords in a custom action file are as follows:

ERROR_TYPE

Indicates the type of the error that the server fault monitor has detected. The following values are permitted for this keyword:

DBMS_ERROR	Specifies that the error is a DBMS error.
SCAN_LOG	Specifies that the error is an alert that is logged in the alert log file.
TIMEOUT_ERROR	Specifies that the error is a timeout.

The `ERROR_TYPE` keyword is optional. If you omit this keyword, the error is assumed to be a DBMS error.

ERROR

Identifies the error. The data type and the meaning of *error-spec* are determined by the value of the `ERROR_TYPE` keyword as shown in the following table.

<code>ERROR_TYPE</code>	Data Type	Meaning
<code>DBMS_ERROR</code>	Integer	The error number of a DBMS error that is generated by Oracle
<code>SCAN_LOG</code>	Quoted regular expression	A string in an error message that Oracle has logged to the Oracle alert log file
<code>TIMEOUT_ERROR</code>	Integer	The number of consecutive timed-out probes since the server fault monitor was last started or restarted

You must specify the `ERROR` keyword. If you omit this keyword, the entry in the custom action file is ignored.

ACTION

Specifies the action that the server fault monitor is to perform in response to the error. The following values are permitted for this keyword:

<code>NONE</code>	Specifies that the server fault monitor ignores the error.
<code>STOP</code>	Specifies that the server fault monitor is stopped.
<code>RESTART</code>	Specifies an action that depends on the type of resource for which the fault monitor that is being customized: <ul style="list-style-type: none"> ▪ HA Oracle server resource. Specifies that the server fault monitor stops and restarts the entity that is specified by the value of the <code>Restart_type</code> extension property of the <code>SUNW.oracle_server</code> resource. ▪ Oracle 9i RAC server resource. Specifies that the server fault monitor stops and restarts the Oracle 9i RAC server resource.
<code>SWITCH</code>	Specifies that the server fault monitor switches over the database server resource group to another node.

Note – Do *not* specify the `SWITCH` keyword in the custom action file for an Oracle 9i RAC server fault monitor. For the Oracle 9i RAC server fault monitor, the `SWITCH` keyword performs no action.

The **ACTION** keyword is optional. If you omit this keyword, the server fault monitor ignores the error.

CONNECTION_STATE

Specifies the required state of the connection between the database and the server fault monitor when the error is detected. The entry applies only if the connection is in the required state when the error is detected. The following values are permitted for this keyword:

- * Specifies that the entry always applies, regardless of the state of the connection.
- co Specifies that the entry applies only if the server fault monitor is attempting to connect to the database.
- on Specifies that the entry applies only if the server fault monitor is online. The server fault monitor is online if it is connected to the database.
- di Specifies that the entry applies only if the server fault monitor is disconnecting from the database.

The **CONNECTION_STATE** keyword is optional. If you omit this keyword, the entry always applies, regardless of the state of the connection.

NEW_STATE

Specifies the state of the connection between the database and the server fault monitor that the server fault monitor must attain after the error is detected. The following values are permitted for this keyword:

- * Specifies that the state of the connection must remain unchanged.
- co Specifies that the server fault monitor must disconnect from the database and reconnect immediately to the database.
- di Specifies that the server fault monitor must disconnect from the database. The server fault monitor reconnects when it next probes the database.

The **NEW_STATE** keyword is optional. If you omit this keyword, the state of the database connection remains unchanged after the error is detected.

MESSAGE

Specifies an additional message that is printed to the resource's log file when this error is detected. The message must be enclosed in double quotes. This message is additional to the standard message that is defined for the error.

The **MESSAGE** keyword is optional. If you omit this keyword, no additional message is printed to the resource's log file when this error is detected.

Examples **EXAMPLE 1** Changing the Response to a DBMS Error to Restart

```
{  
  ERROR_TYPE=DBMS_ERROR;  
}
```

EXAMPLE 1 Changing the Response to a DBMS Error to Restart *(Continued)*

```

ERROR=4031;
ACTION=restart;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Insufficient memory in shared pool.";
}

```

This example shows an entry in a custom action file that overrides the preset action for DBMS error 4031. This entry specifies the following behavior:

- In response to DBMS error 4031, the action that the server fault monitor performs is restart.
- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.
- The following message is printed to the resource's log file when this error is detected:
 Insufficient memory in shared pool.

EXAMPLE 2 Ignoring a DBMS Error

```

{
ERROR_TYPE=DBMS_ERROR;
ERROR=4030;
ACTION=none;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="";
}

```

This example shows an entry in a custom action file that overrides the preset action for DBMS error 4030. This entry specifies the following behavior:

- The server fault monitor ignores DBMS error 4030.
- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.
- No additional message is printed to the resource's log file when this error is detected.

EXAMPLE 3 Changing the Response to a Logged Alert

```

{
ERROR_TYPE=SCAN_LOG;
ERROR="ORA-00600: internal error";
}

```

EXAMPLE 3 Changing the Response to a Logged Alert *(Continued)*

```
ACTION=RESTART;  
}
```

This example shows an entry in a custom action file that overrides the preset action for logged alerts about internal errors. This entry specifies the following behavior:

- In response to logged alerts that contain the text `ORA-00600: internal error`, the action that the server fault monitor performs is restart.
- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.
- No additional message is printed to the resource's log file when this error is detected.

EXAMPLE 4 Changing the Maximum Number of Consecutive Timed-Out Probes

```
{  
  ERROR_TYPE=TIMEOUT;  
  ERROR=2;  
  ACTION=NONE;  
  CONNECTION_STATE=*;  
  NEW_STATE=*;  
  MESSAGE="Timeout #2 has occurred."  
}
```

```
{  
  ERROR_TYPE=TIMEOUT;  
  ERROR=3;  
  ACTION=NONE;  
  CONNECTION_STATE=*;  
  NEW_STATE=*;  
  MESSAGE="Timeout #3 has occurred."  
}
```

```
{  
  ERROR_TYPE=TIMEOUT;  
  ERROR=4;  
  ACTION=NONE;  
  CONNECTION_STATE=*;  
  NEW_STATE=*;  
  MESSAGE="Timeout #4 has occurred."  
}
```

```
{  
  ERROR_TYPE=TIMEOUT;  
  ERROR=5;
```

EXAMPLE 4 Changing the Maximum Number of Consecutive Timed-Out Probes *(Continued)*

```
ACTION=RESTART;  
CONNECTION_STATE=*;  
NEW_STATE=*;  
MESSAGE="Timeout #5 has occurred. Restarting."  
}
```

This example shows the entries in a custom action file for increasing the maximum number of consecutive timed-out probes to five. These entries specify the following behavior:

- The server fault monitor ignores the second consecutive timed-out probe through the fourth consecutive timed-out probe.
- In response to the fifth consecutive timed-out probe, the action that the server fault monitor performs is restart.
- The entries apply regardless of the state of the connection between the database and the server fault monitor when the timeout occurs.
- The state of the connection between the database and the server fault monitor must remain unchanged after the timeout occurs.
- When the second consecutive timed-out probe through the fourth consecutive timed-out probe occurs, a message of the following form is printed to the resource's log file:

Timeout #*number* has occurred.
- When the fifth consecutive timed-out probe occurs, the following message is printed to the resource's log file:

Timeout #5 has occurred. Restarting.

See Also [SUNW.oracle_server\(5\)](#), [SUNW.scalable_rac_server\(5\)](#)

R E F E R E N C E

OSC40DS 5

Name ORCL.obiee_cluster_controller, obiee_cluster_controller – resource type implementation for HA-Oracle Business Intelligence Enterprise Edition (Oracle BI EE) Cluster Controller

Description The ORCL.obiee_cluster_controller resource type represents the HA Oracle BI EE Cluster Controller in an Oracle Solaris Cluster configuration.

The HA-Oracle BI EE Cluster Controller resource is configured in a failover resource group when creating an Oracle BI EE installation that consists of more than one BI EE Server. The node lists for the resource groups that contain the primary and secondary cluster controllers must not contain common nodes.

You must set the following properties for an Oracle BI EE Cluster Controller resource by using the `clresource(1CL)` command.

Standard Properties The standard resource property `Failover` is set for all failover resource types.

See the `r_properties(5)` man page for a complete description of the following resource properties.

`Failover_mode`

Default SOFT

Tunable Any time

`Retry_count`

Default 2

Tunable Any time

`Retry_interval`

Default 1330

Tunable Any time

`Thorough_probe_interval`

Default 30

Tunable Any time

Extension Properties `BI_Install_Directory`

Type string (required). This property is set to the absolute path of the Oracle BI software installation directory. You can modify this parameter only when it is disabled.

`BI_Data_Directory`

Type string (required). This property is set to the absolute path of the Oracle BI software data directory. You can modify this parameter only when it is disabled.

Run_64_bit

Type Boolean (required). Default is TRUE. If this property is set to TRUE, then the 64-bit version of the binary is run. If it is set to FALSE, then the 32-bit version of the binary is run. You can modify this parameter only when it is disabled.

Cluster_Controller_Role

Type enumerated (required). Default is PRIMARY. This property determines whether the resource represents a primary or secondary cluster controller. Valid values are PRIMARY or SECONDARY. You can modify this parameter only when it is disabled.

Debug_Level

Type enumerated (optional). Default is 0. Valid values are 0, 1, and 2. This property specifies the debug level. You can modify this parameter at any time.

See Also [pmfadm\(1M\)](#) [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name ORCL.obiee_presentation_service, obiee_presentation_service – resource type implementation for HA-Oracle Business Intelligence Enterprise Edition (Oracle BI EE) Presentation Service

Description The ORCL.obiee_presentation_service resource type represents the HA Oracle BI EE Presentation Service in an Oracle Solaris Cluster configuration. The HA-Oracle BI EE Presentation Service resource can be configured in one of two ways:

- In a failover resource group when the BI EE Cluster Controller is not used.
- In a multi-master resource group when the BI EE Cluster Controller is used.

You must set the following properties for an Oracle BI EE Presentation Service resource by using the `clresource(1CL)` command.

Standard Properties The standard resource property `Failover` is set for all failover resource types.

See the [r_properties\(5\)](#) man page for a complete description of the following resource properties.

`Failover_mode`

Default SOFT

Tunable Any time

`Retry_count`

Default 2

Tunable Any time

`Retry_interval`

Default 1330

Tunable Any time

`Thorough_probe_interval`

Default 30

Tunable Any time

Extension Properties `BI_Install_Directory`

Type string (required). This property is set to the absolute path of the Oracle BI software installation directory. You can modify this parameter only when it is disabled.

`BI_Data_Directory`

Type string (required). This property is set to the absolute path of the Oracle BI software data directory. You can modify this parameter only when it is disabled.

Run_64_bit

Type Boolean (required). Default is TRUE. If this property is set to TRUE, then the 64-bit version of the binary is run. If it is set to FALSE, then the 32-bit version of the binary is run. You can modify this parameter only when it is disabled.

Debug_Level

Type enumerated (optional). Default is 0. Valid values are 0, 1, and 2. This property specifies the debug level. You can modify this parameter at any time.

See Also [pmfadm\(1M\)](#) [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name ORCL.obiee_scheduler, obiee_scheduler – resource type implementation for HA Oracle Business Intelligence Enterprise Edition (Oracle BI EE) Scheduler

Description The ORCL.obiee_scheduler resource type represents the HA Oracle BI EE Scheduler in an Oracle Solaris Cluster configuration. The HA-Oracle BI EE Scheduler resource can be configured in one of two ways:

- In a failover resource group when the BI EE Cluster Controller is not used.
- In a multi-master resource group when the BI EE Cluster Controller is used.

You must set the following properties for an Oracle BI EE Scheduler resource by using the `clresource(1CL)` command.

Standard Properties The standard resource property `FailOver` is set for all failover resource types.

See the `r_properties(5)` man page for a complete description of the following resource properties.

`Failover_mode`

Default SOFT

Tunable Any time

`Retry_count`

Default 2

Tunable Any time

`Retry_interval`

Default 1330

Tunable Any time

`Thorough_probe_interval`

Default 30

Tunable Any time

Extension Properties `BI_Install_Directory`

Type string (required). This property is set to the absolute path of the Oracle BI software installation directory. You can modify this parameter only when it is disabled.

`BI_Data_Directory`

Type string (required). This property is set to the absolute path of the Oracle BI software data directory. You can modify this parameter only when it is disabled.

`Run_64_bit`

Type Boolean (required). Default is TRUE. If this property is set to TRUE, then the 64-bit version of the binary is run. If it is set to FALSE, then the 32-bit version of the binary is run. You can modify this parameter only when it is disabled.

Debug_Level

Type enumerated (optional). Default is 0. Valid values are 0, 1, and 2. This property specifies the debug level. You can modify this parameter at any time.

See Also [pmfadm\(1M\)](#) [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name ORCL.obiee_server, obiee_server – resource type implementation for HA Oracle Business Intelligence Enterprise Edition (Oracle BI EE) Server

Description The ORCL.obiee_server resource type represents the HA Oracle BI EE Server in an Oracle Solaris Cluster configuration. The HA-Oracle BI EE Server resource can be configured in one of two ways:

- In a failover resource group when the BI EE Cluster Controller is not used.
- In a multi-master resource group when the BI EE Cluster Controller is used.

You must set the following properties for an Oracle BI EE Server resource by using the `clresource(1CL)` command.

Standard Properties The standard resource property `FailOver` is set for all failover resource types.

See the `r_properties(5)` man page for a complete description of the following resource properties.

`Failover_mode`

Default SOFT

Tunable Any time

`Retry_count`

Default 2

Tunable Any time

`Retry_interval`

Default 1330

Tunable Any time

`Thorough_probe_interval`

Default 30

Tunable Any time

Extension Properties `BI_Install_Directory`

Type string (required). This property is set to the absolute path of the Oracle BI software installation directory. You can modify this parameter only when it is disabled.

`BI_Data_Directory`

Type string (required). This property is set to the absolute path of the Oracle BI software data directory. You can modify this parameter only when it is disabled.

`Run_64_bit`

Type Boolean (required). Default is TRUE. If this property is set to TRUE, then the 64-bit version of the binary is run. If it is set to FALSE, then the 32-bit version of the binary is run. You can modify this parameter only when it is disabled.

Debug_Level

Type enumerated (optional). Default is 0. Valid values are 0, 1, and 2. This property specifies the debug level. You can modify this parameter at any time.

See Also [pmfadm\(1M\)](#) [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.apache, apache – resource type implementation for failover and scalable Apache Web Server

Description The Apache Web Server data service for Oracle Solaris Cluster is configured as a resource managed by the Oracle Solaris Cluster Resource Group Manager (RGM).

You must set the following properties on an Apache resource by using `clresource(1CL)`.

Standard Properties The standard resource properties `Scalable`, `Network_resources_used`, `Resource_dependencies`, `Port_list`, `Load_balancing_policy`, and `Load_balancing_weights` are common to all scalable resource types.

The `SUNW.apache` resource type supports two modes. The first mode is a scalable mode that exploits the cluster networking facility to permit the Apache resource to run on multiple nodes simultaneously. The second mode is a failover mode, in which the Apache resource runs on only one node at a time. The `Scalable` property is set at resource creation time to indicate the mode in which the service operates. The default is `FALSE` (failover mode).

See [Seer_properties\(5\)](#) for a complete description of the following resource properties.

`Load_balancing_policy`

Default LB_WEIGHTED

Tunable At creation

`Load_balancing_weights`

Default NULL

Tunable Any time

`Network_resources_used` (string array)

A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties `Resource_dependencies`, `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart`.

This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the `Resource_dependencies` property.

Category Conditional/Optional

Default The empty list

Tunable At creation

`Resource_dependencies` (string array)

Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the [clreslogicalhostname\(1CL\)](#) and [clressharedaddress\(1CL\)](#) man pages for more information.

You can specify an alternate kind of dependency by using the `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart` property instead of the `Resource_dependencies` property. For more information, see the `r_properties(5)` man page.

Category Optional

Default The empty list

Tunable Any time

`Port_list`

Default 80/tcp

Tunable At creation

`Retry_count`

Minimum 0

Maximum 10

Default 2

Tunable Any time

`Retry_interval`

Minimum 0

Maximum 3600

Default 620

Tunable Any time

`Thorough_probe_interval`

Minimum 0

Maximum 3600

Default 60

Tunable Any time

Extension Properties `Bin_dir`

Type string. Indicates the location of Apache Web server binaries. You must specify this property at resource creation time.

Monitor_retry_count

Type integer. Default is 4. Minimum is -1. Controls the restarts of the fault monitor. This property indicates the number of times the fault monitor is restarted by the process monitor facility and corresponds to the `-n` option passed to the `pmfadm(1M)` command. The number of restarts is counted in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not the web server. The restarts of the web server are controlled by the system-defined properties `Thorough_Probe_Interval`, `Retry_Interval`, and `Retry_Count`, as specified in their descriptions. See `clresource(1CL)`. You can modify the value for this property at any time.

Monitor_retry_interval

Type integer. Default is 2. Minimum is -1. Indicates the time in minutes, over which the failures of the fault monitor are counted, and corresponds to the `-t` option passed to the `pmfadm(1M)` command. If the number of times the fault monitor fails exceeds the value of `Monitor_retry_count`, the fault monitor is not restarted by the process monitor facility. You can modify the value for this property at any time.

Monitor Uri_List

Type string array. Default is "". Introduced in release 3.1 10/03. This property enables you to ensure that application components are responding by querying the configured URIs. The `Monitor Uri_List` property is used for detailed fault monitoring of Oracle Solaris Cluster HA for Apache Web Server. The fault monitor periodically runs the HTTP GET command for the URIs. The monitor takes action if the HTTP request returns with response code 500 "Internal Server Error" or if the application server does not respond. An example URI setting is `http://logical-hostname/App/tester`. If the configured URIs are implemented by using a servlet in the web server, detailed monitoring of the web server Java Virtual Machine (JVM) is possible.

Probe_timeout

Type integer. Defaults to 90. Minimum is 15. This property is the time-out value (in seconds) used by the fault monitor to probe an Apache instance. You can modify the value for this property at any time.

Examples **EXAMPLE 1** Creating a Failover Apache Resource

For this example to work, the data service must first be installed. This example creates a failover Apache resource named `apache-failover` in an existing resource group named `web-rg`. `web-rg` is assumed to contain a `LogicalHostname` resource, which identifies the logical hostname associated with the resource group. Another assumption is that the `Port_list` property defaults to `80/tcp`, that is, the Apache instance is listening on port 80.

```
example# clresourcetype register SUNW.apache
example# clresource create -g web-rg -t SUNW.apache \
-p Bin_dir=/global/apache/https-web/bin apache-failover
```

In this example, the Apache resource created is named `apache-failover`, which listens on port 80, with a corresponding Apache instance in the directory `/global/apache/https-web`.

EXAMPLE 2 Creating a Scalable Apache Resource

For this example to work, the data service must first be installed. This example creates a scalable Apache resource named `apache-scalable` in a resource group named `web-rg`, which is configured to run simultaneously on all four nodes of a four-node cluster. The `apache-scalable` resource is configured to listen on port 8080 and uses the IP addresses as configured in a `SharedAddress` resource named `www_foo_com`, which is contained in the resource group `foo_com_RG`.

```
example# clresourcegroup create -p Maximum primaries=4 \
-p Desired primaries=4 -p RG_dependencies=foo_com_RG web-rg
example# clresource type register SUNW.apache
example# clresource create -g web-rg -t SUNW.apache \
-p Bin_dir=/global/apache/https-web/bin \
-p Port_list=8080/tcp -y Scalable=TRUE \
-p Network_resources_used=www_foo_com apache-scalable
```

EXAMPLE 3 Setting `Monitor_uri_list` for Scalable Oracle Solaris Cluster HA for Apache Instance

The following example shows how to set the `Monitor_uri_list` extension property when a scalable Oracle Solaris Cluster HA for Apache instance is added to a configuration. The `Monitor_uri_list` extension property is not supported with a secure Oracle Solaris Cluster HA for Apache instance.

(Add an insecure Apache instance with default load balancing.)

```
example# clresource create -g resource-group-1 \
-t SUNW.apache -p Bin_dir=/opt/apache/bin \
-p Monitor_Uri_list=http://schost-1:8000/servlet/monitor \
-p Network_resources_used=schost-1,... \
-p Scalable=True -p Port_list=8000/tcp apache-insecure-1
```

EXAMPLE 4 Setting `Monitor_uri_list` for Failover Oracle Solaris Cluster HA for Apache Instance

The following example shows how to set the `Monitor_uri_list` extension property when a failover Oracle Solaris Cluster HA for Apache instance is added to a configuration. The `Monitor_uri_list` extension property is not supported with a secure Oracle Solaris Cluster HA for Apache instance.

(Add an insecure Apache application resource instance.)

```
# clresource create -g resource-group-1 \
-t SUNW.apache -p Bin_dir=/opt/apache/bin \
-p Monitor_Uri_list=http://schost-1:80/servlet/monitor \
-p Network_resources_used=schost-1 \
-p Scalable=False -p Port_list=80/tcp apache-insecure-1
```

Files Bin_dir/apachectl

The `apachectl start` command is used by HA-Apache to start a non-SSL Apache web server.

The `apachectl startssl` command is used by HA-Apache to start an Apache web server that uses `mod_ssl`.

Bin_dir/httpsdctl

The `httpsdctl start` command is used by HA-Apache to start an Apache-ssl web server.

Bin_dir/keypass

This file must be created for an Apache web server that uses `mod_ssl` for handling `https` requests. Only the owner should have read, write, or execute permissions to this file. All other users must not have permissions to this file.

If the web server does not use encrypted private keys, the contents of this file are irrelevant. For a web server that uses encrypted private keys, this file is called during resource startup with `host:port` and `algorithm` as its two arguments. The web server expects the pass phrase for the key corresponding to that host and port combination to be printed to `stdout`.

For example, for a secure web server listening on ports 8080 and 8888 that use RSA encrypted private keys for both ports, the `keypass` file could look like the following:

```
#!/bin/ksh
host='echo $1 | cut -d: -f1'
port='echo $1 | cut -d: -f2'
algorithm=$2

if [ "$host" = "button-1.eng.sun.com" \
  -a "$algorithm" = "RSA" ]; then
case "$port" in
  8080) echo passphrase-for-8080;;
  8888) echo passphrase-for-8888;;
esac
fi
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/apache

See Also [pmfadm\(1M\)](#), [scha_resource_get\(3HA\)](#), [clresourcetype\(1CL\)](#), [clresourcegroup\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [scalable_service\(5\)](#)

, *Oracle Solaris Cluster Data Services Planning and Administration Guide*

Name SUNW.asm_diskgroup, asm_diskgroup – resource type implementation for the Oracle Automated Storage Management (ASM) disk group managed by Oracle Solaris Cluster

Description The SUNW.asm_diskgroup resource type represents the ASM disk group in an Oracle Solaris Cluster configuration.

Note – The SUNW.asm_diskgroup resource type can be used when using Oracle Cluster Ready Services (CRS) or not.

The SUNW.asm_diskgroup resource type is a failover or multiple-master resource type. A single resource of this type can run on multiple nodes concurrently, but does not use network load balancing.

Each SUNW.asm_diskgroup resource represents a ASM disk group. Each single instance ASM disk group is uniquely identified by the value of the `asm_diskgroups` extension property on the node where the instance is running.

The ASM disk group resource should only be mounted if the ASM instance is available on the same cluster node. Additionally, ASM disk groups should be mounted if any required storage resources are enabled on the cluster node. To ensure that these requirements are met, configure the ASM disk group resource as follows:

- Create a strong positive affinity between the ASM disk group resource group and the following resource groups:
 - The single instance or clustered ASM resource group
 - Any resource group that contains storage resources for Oracle files
- Create an offline-restart dependency between the ASM disk group resource and the following resources:
 - The single instance or clustered ASM instance resource
 - Any storage resources for Oracle files that you are using

Create these dependencies and affinities when you configure clustered ASM instance resources for the Oracle Solaris Cluster Support for Oracle data service or Oracle Solaris Cluster Support for Oracle RAC data service.

To register this resource type and create instances of this resource type, use one of the following:

- The `clsetup(1CL)` utility, specifying the option for configuring Oracle Solaris Cluster Support for Oracle Real Application Clusters
- The following sequence of Oracle Solaris Cluster maintenance commands:
 - To register this resource type, use the `clresourcetype(1CL)` command.
 - To create instances of this resource type, use the `clresource(1CL)` command.

Standard Properties For a description of all standard resource properties, see the [r_properties\(5\)](#) man page.

Standard resource properties are overridden for this resource type as follows:

Failover_mode

Default SOFT

Tunable Any time

Load_balancing_policy

Default LB_WEIGHTED

Tunable At creation

Load_balancing_weights

Default ""

Tunable Any time

Monitor_check_timeout

Minimum 60

Default 300

Monitor_start_timeout

Minimum 60

Default 300

Monitor_stop_timeout

Minimum 60

Default 300

Network_resources_used

Default ""

Default Any time

Port_list

Default ""

Tunable Anytime

Retry_count

Maximum 10

Default 2

Tunable	Anytime
Retry_Interval	
Maximum	3600
Default	370
Tunable	Anytime
Scalable	
Default	FALSE
Tunable	At creation
Start_timeout	
Minimum	60
Default	300
Stop_timeout	
Minimum	60
Default	300
Through_probe_interval	
Minimum	3600
Default	60
Tunable	ANYTIME
Update_timeout	
Minimum	60
Default	300
Validate_timeout	
Minimum	60
Default	300

Extension Properties The extension properties of the SUNW.asm_diskgroups resource type are as follows.

asm_diskgroups

This property specifies the Oracle ASM disk group. If required, more than one ASM disk group can be specified as a comma separated list.

Data Type	String array
Range	Not applicable

Default	“”
Tunable	When disabled

Child_mon_level(integer)

Provides control over the processes that are monitored through the Process Monitor Facility (PMF). This property denotes the level to which the forked child processes are monitored. Omitting this property or setting this property to the default value is the same as omitting the -Cooption for `pmfadm(1M)`. All child processes and their descendants are monitored.

Category	Optional
Default	-1
Tunable	When disabled

debug_level

Note – All SQL*Plus and srmgr messages that the ASM disk group resource issues are written to the log file `/var/opt/SUNWscor/oracle_asm/message_log. ${RESOURCE}`.

This property indicates the level to which debug messages for the Oracle ASM disk group resources are logged. When the debug level is increased, more debug messages are written to the system log `/var/adm/messages` as follows:

0	No debug messages
1	Function Begin and End messages
2	All debug messages and function Begin/End messages

You can specify a different value of the `debug_level` extension property for each node that can master the resource.

Data Type	Integer
Range	0–2
Default	0
Tunable	Any time

Failover_Enabled(boolean)

Allows the resources to fail over. If this property is set to `False`, failover of the resource is disabled. You can use this property to prevent the application resource from initiating a failover of the resource group.

Note – Use the `Failover_mode` property instead of the `Failover_enabled` extension property because `Failover_mode` better controls failover behavior. For more information, see the descriptions of the `LOG_ONLY` and `RESTART_ONLY` values for `Failover_mode` in [r_properties\(5\)](#).

Category	Optional
Default	True
Tunable	When disabled

Log_level

Specifies the level, or type, of diagnostic messages that are logged by GDS. You can specify `None`, `Info`, or `Err` for this property. When you specify `None`, diagnostic messages are not logged by GDS. When you specify `Info`, both information and error messages are logged. When you specify `Err`, only error messages are logged.

Category	Optional
Default	Info
Tunable	Any time

Network_aware(boolean)

This property specifies whether an application uses the network.

Category	Optional
Default	False
Tunable	At creation

Monitor_retry_count

This property specifies the number of PMF restart allowed for the fault monitor.

Default	4
Tunable	Any time

Monitor_retry_interval

This property specifies the number of PMF restart allowed for the fault monitor.

Default	2
Tunable	Any time

probe_command(string)

Specifies the command that periodically checks the health of the single instance ASM.

Category	Required
Default	<code>"/opt/SUNWscor/oracle_asm/bin/asm_control probe -R %RS_NAME -G %RG_NAME -T %RT_NAME"</code>

Tunable	None
Probe_timeout(integer)	This property specifies the timeout value, in seconds, for the probe command.
Category	Optional
Default	30 seconds
Tunable	Any time
Start_command(string)	Specifies the command that mounts the ASM diskgroup.
Category	Required
Default	<code>"/opt/SUNWscor/oracle_asm/bin/asm_control start -R %RS_NAME -G %RG_NAME -T %RT_NAME"</code>
Tunable	None
Stop_command(string)	Specifies the command that dismounts the ASM diskgroup.
Category	Required
Default	<code>"/opt/SUNWscor/oracle_asm/bin/asm_control stop -R %RS_NAME -G %RG_NAME -T %RT_NAME"</code>
Tunable	None
Stop_signal(integer)	Specifies the command that send stop signal to the ASM diskgroup.
Category	Optional
Default	15
Tunable	When disabled
Validate_command(string)	Specifies the absolute path to the command that validates the application, although currently not used.
Category	Optional
Default	NULL
Tunable	When disabled

Examples EXAMPLE 1 Creating a Failover asm_diskgroup Resource

This example shows the commands for performing the following operations to create a failover asm_diskgroup resource on a two-node cluster:

EXAMPLE 1 Creating a Failover asm_diskgroup Resource *(Continued)*

1. Creating the asm-dg-rg resource group
2. Registering the SUNW.asm_diskgroup resource type
3. Setting the resource group affinity
4. Adding the asm-dg-rs resource to the asm-dg-rg resource group
5. Setting the asm_diskgroups extension property

The example makes the following assumptions:

- The bash shell is used.
- A resource group that is named asm-inst-rg exists and contains a resource of type SUNW.scalable_asm_instance that is named asm-inst-rs.

```
phys-schost-1# clresourcetype register SUNW.asm_diskgroup
phys-schost-1# clresourcegroup create asm-dg-rg
phys-schost-1# clresourcegroup set -p rg_affinities=++asm-inst-rg asm-dg-rg
phys-schost-1# clresource create -g asm-dg-rg \
-t SUNW.asm_diskgroup \
-p asm_diskgroups=dg1 \
-p resource_dependencies_offline_restart=asm-instance-rs \
-d asm-dd-rs
phys-schost-1# clresourcegroup online -M asm-dg-rg
phys-schost-1# clresource enable asm-dg-rs
```

EXAMPLE 2 Creating a Scalable asm_diskgroup Resource

This example shows the commands for performing the following operations to create a scalable multi-master asm_diskgroup resource on a two-node cluster:

1. Creating the asm-dg-rg resource group
2. Registering the SUNW.asm_diskgroup resource type
3. Setting the resource group affinity
4. Adding the asm-dg-rs resource to the asm-dg-rg resource group
5. Setting the asm_diskgroups extension property for two ASM disk groups

The example makes the following assumptions:

- The bash shell is used.
- A resource group that is named asm-inst-rg exists and contains a resource of type SUNW.scalable_asm_instance_proxy that is named asm-inst-rs.
- A resource group that is named scal-mp-rg exists and contains a resource of type SUNW.ScalMountPoint that is named qfs-mp-rs.

```
phys-schost-1# clresourcetype register SUNW.asm_diskgroup
phys-schost-1# clresourcegroup create -g asm-dg-rg
phys-schost-1# clresourcegroup set -p rg_affinities=++asm-inst-rg asm-dg-rg
```

EXAMPLE 2 Creating a Scalable asm_diskgroup Resource *(Continued)*

```
phys-schost-1# clresource create -g asm-dg-rg \  
-t SUNW.asm_diskgroup \  
-p asm_diskgroups=dg1,dg2 \  
-p resource_dependencies_offline_restart=asm-instance-rs, qfs-mp-rs \  
-d asm-dg-rs  
phys-schost-1# clresourcegroup online -M asm-dg-rg  
phys-schost-1# clresource enable asm-dg-rs
```

See Also [*Oracle Solaris Cluster Data Services Planning and Administration Guide*](#)

Name SUNW.dns, dns – resource type implementation for failover Domain Name Service (DNS)

Description The DNS data service for Oracle Solaris Cluster is configured as a resource managed by the Oracle Solaris Cluster Resource Group Manager (RGM). You must set the following properties on a DNS resource.

See [r_properties\(5\)](#) for a complete description of the following resource properties.

Standard Properties **Network_resources_used** (string array)

A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties `Resource_dependencies`, `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart`.

This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the `Resource_dependencies` property.

Category Conditional/Optional

Default The empty list

Tunable When disabled

Resource_dependencies (string array)

Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the [clreslogicalhostname\(1CL\)](#) and [clressharedaddress\(1CL\)](#) man pages for more information.

You can specify an alternate kind of dependency by using the `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart` property instead of the `Resource_dependencies` property. For more information, see the [r_properties\(5\)](#) man page.

Category Optional

Default The empty list

Tunable Any time

Port_list

Default 53/udp

Tunable At creation

For DNS resources, the value of 53/udp is the only recommended value.

Retry_count

Minimum 0

Maximum 10

Default 2

Tunable Any time

Retry_interval

Minimum 0

Maximum 3600

Default 750

Tunable Any time

Thorough_probe_interval

Minimum 0

Maximum 3600

Default 60

Tunable Any time

Extension Properties **Confdir_list**

Type string array. This property is the path name to the configuration directory that contains the file named `.conf` of the DNS resource. You must specify only one value for this property at resource creation time.

DNS_mode

Type string array. This property is the configuration file to be used for starting DNS. The default is `conf`, which means that the DNS is started by using the `named.conf` file located in the directory pointed to by the value of the `Confdir_list` property. A value of `boot` means that DNS is started with the `named.boot` file as the configuration file. You can specify the value of this property at resource creation time only.

Monitor_retry_count

Type integer. Default is 4. The Minimum value is -1. This property controls the restarts of the fault monitor. It indicates the number of times the fault monitor is restarted by the process monitor facility and corresponds to the `-n` option passed to the `pmfadm(1M)` command. The number of restarts is counted in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not DNS. The restarts of DNS are controlled by the system-defined

properties `Thorough_Probe_Interval` and `Retry_Interval` and `Retry_Count`, as specified in the description of those system-defined properties. See [clresource\(1CL\)](#). You can modify the value for this property any time.

`Monitor_retry_interval`

Type integer. Default is 2. The Minimum value is -1. Indicates the time (in minutes) over which the failures of the fault monitor are counted and corresponds to the `-t` option passed to the `pmfadm(1M)` command. If the number of times the fault monitor fails exceeds the extension property `Monitor_retry_count`, the fault monitor is not restarted by the Process Monitor Facility. You can modify the value for this property any time.

`Probe_timeout`

Type integer. Default is 120. The Minimum value is 15. Indicates the time-out value (in seconds) used by the fault monitor to probe a DNS instance. You can modify the value for this property any time.

Examples **EXAMPLE 1** Initiating a Failover DNS Resource

For this example to work, you must first install the data service. This example instantiates a failover DNS resource named `dnss` in a resource group named `dns-rg`. `dns-rg` is assumed to contain at least one `LogicalHostname` resource, which identifies the logical hostnames associated with the resource group.

```
example# clresourcetype register SUNW.dns
example# clresource create -g dns-rg -t SUNW.dns \
-p Confdir_list=/global/dns dnss
```

In this example, the DNS resource created is named `dnss`, which listens on port 53, with a corresponding configuration directory path name `/global/dns`. The configuration file that is used for starting the DNS resource is named `.conf`, located under `/global/dns`.

EXAMPLE 2 Instantiating a Failover DNS Resource

For this example to work, the data service must first be installed. This example instantiates a failover DNS resource named `dns` in a resource group named `dns-rg`, which uses the `LogicalHostname` resource `lh-specific`.

```
example# clresourcetype register SUNW.dns
example# clresource create -g dns-rg -t SUNW.dns \
-p Confdir_list=/global/dns \
-p Network_resources_used=lh-specific dns-lh
```

In this example, the `LogicalHostname` resource `lh-specific` must be a resource in the `dns-rg` resource group.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/dns

See Also [named\(1M\)](#), [pmfadm\(1M\)](#), [scha_resource_get\(3HA\)](#), [clresourcetype\(1CL\)](#), [clresourcegroup\(1CL\)](#), [named.conf\(4\)](#), [attributes\(5\)](#), [r_properties\(5\)](#)

Name	SUNW.iws, iws – resource type implementation for failover and scalable Sun Java System Web Server
Description	<p>The SUNW.iws resource type represents the Sun Java System Web Server application in an Oracle Solaris Cluster configuration.</p> <p>You must set the following properties on an SUNW.iws resource by using <code>clresource(1CL)</code>.</p>
Standard Properties	<p>The standard resource properties <code>Scalable</code>, <code>Network_resources_used</code>, <code>Port_list</code>, <code>Load_balancing_policy</code>, and <code>Load_balancing_weights</code> are common to all scalable resource types.</p> <p>The SUNW.iws resource type supports two modes. The first mode is a scalable mode that exploits the cluster networking facility to permit the iws resource to run on multiple nodes simultaneously. The second mode is a failover mode, in which the iws resource runs on only one node at a time. The <code>Scalable</code> property is set at resource creation time to indicate the mode in which the service operates. The default is <code>FALSE</code> (failover mode).</p> <p>See r_properties(5) for a complete description of the following resource properties.</p> <p><code>Load_balancing_policy</code></p> <p style="padding-left: 2em;">Default LB_WEIGHTED</p> <p style="padding-left: 2em;">Tunable At creation</p> <p><code>Load_balancing_weights</code></p> <p style="padding-left: 2em;">Default NULL</p> <p style="padding-left: 2em;">Tunable Any time</p> <p><code>Network_resources_used</code> (string array)</p> <p>A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties <code>Resource_dependencies</code>, <code>Resource_dependencies_weak</code>, <code>Resource_dependencies_restart</code>, or <code>Resource_dependencies_offline_restart</code>.</p> <p>This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the <code>Resource_dependencies</code> property.</p> <p style="padding-left: 2em;">Category Conditional/Optional</p> <p style="padding-left: 2em;">Default The empty list</p> <p style="padding-left: 2em;">Tunable At creation</p> <p><code>Resource_dependencies</code> (string array)</p> <p>Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.</p>

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the [clreslogicalhostname\(1CL\)](#) and [clressharedaddress\(1CL\)](#) man pages for more information.

You can specify an alternate kind of dependency by using the `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart` property instead of the `Resource_dependencies` property. For more information, see the `r_properties(5)` man page.

Category Optional

Default The empty list

Tunable Any time

`Port_list`

Default 80/tcp

Tunable At creation

`Retry_count`

Minimum 0

Maximum 10

Default 2

Tunable Any time

`Retry_interval`

Minimum 0

Maximum 3600

Default 620

Tunable Any time

`Thorough_probe_interval`

Minimum 0

Maximum 3600

Default 60

Tunable Any time

Extension Properties `Confdir_list`

Type string array. This property is a comma-separated list of path names. Each element in the list is the path name of an `iWS` instance directory. If an `iWS` instance is in secure mode,

then the directory must contain a file named `keypass`, which contains the secure key password needed to start this instance. You must specify this property at resource creation time.

Monitor_retry_count

Type integer. Default is 4. Minimum value is -1. Controls the restarts of the fault monitor. This property indicates the number of times the fault monitor is restarted by the process monitor facility and corresponds to the `-n` option passed to the `pmfadm(1M)` command. The number of restarts is counted in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not the web server. The restarts of the web server are controlled by the system-defined properties `Thorough_Probe_Interval`, `Retry_Interval`, and `Retry_Count`, as specified in their descriptions. See `clresource(1CL)`. You can modify the value for this property at any time.

Monitor_retry_interval

Type integer. Default is 2. Minimum value is -1. Indicates the time in minutes, over which the failures of the fault monitor are counted, and corresponds to the `-t` option passed to the `pmfadm(1M)` command. If the number of times the fault monitor fails exceeds the value of `Monitor_retry_count`, the fault monitor is not restarted by the Process Monitor Facility. You can modify the value for this property at any time.

Monitor Uri_List

Type string array. Default is `""`; introduced in release 3.1 10/03. This property allows you to ensure that application components are responding by querying the configured URIs. The `Monitor Uri_List` property is used for detailed fault monitoring of Sun Java System Web Server. The fault monitor periodically runs the HTTP GET command for the URIs. The monitor takes action if the HTTP request returns with response code 500 "Internal Server Error" or if the application server does not respond. An example URI setting is `http://logical-hostname/App/tester`. If the configured URIs are implemented by using a servlet in the web server, detailed monitoring of the web server Java Virtual Machine (JVM) is possible.

Probe_timeout

Type integer. Default is 90. Minimum value is 15. This property is the time out value (in seconds) that is used by the fault monitor to probe an iWS instance. You can modify the value for this property at any time.

Examples **EXAMPLE 1** Creating a Failover iWS Resource in an Existing Group

For this example to work, the data service must first be installed. This example creates a failover iWS resource named `webserver-failover` in an existing resource group named `web-rg`. `web-rg` is assumed to contain a `LogicalHostname` resource, which identifies the logical hostname associated with the resource group. Another assumption is that the `Port_list` property defaults to `80/tcp`; that is, the iWS instance is listening on port 80.

```
example# clresourcetype register SUNW.iws
example# clresource create -g web-rg -t SUNW.iws \
```

EXAMPLE 1 Creating a Failover iWS Resource in an Existing Group (Continued)

```
-p Confdir_list=/global/iws/https-web webserver-failover
```

In this example, the iws resource created is named `webserver-failover`, which listens on port 80, with a corresponding iWS instance in the directory `/global/iws/https-web`.

EXAMPLE 2 Creating a Scalable iWS Resource

For this example to work, the data service must first be installed. This example creates a scalable iWS resource named `webserver-scalable` in a resource group named `web-rg`, which is configured to run simultaneously on all four nodes of a four-node cluster. The `webserver-scalable` resource is configured to listen on port 8080 and uses the IP addresses as configured in a `SharedAddress` resource named `www_foo_com`, which is contained in the resource group `foo_com_RG`.

```
example# clresourcegroup create \  
-p Maximum primaries=4 -p Desired primaries=4 \  
-p RG_dependencies=foo_com_RG web-rg  
example# clresourcetype register SUNW.iws  
example# clresource create -g web-rg -t SUNW.iws \  
-p Confdir_list=/global/iws/https-web \  
-p Port_list=8080/tcp -p Scalable=TRUE  
\-p Network_resources_used=www_foo_com webserver-scalable
```

EXAMPLE 3 Creating a Failover iWS Resource Listening on a Specified Port

For this example to work, the data service must first be installed. This example creates a failover iWS resource named `webserver-secure`, which listens on port 443 in an existing resource group named `web-rg`.

```
example# clresourcetype register SUNW.iws  
example# clresource create -g web-rg -t SUNW.iws \  
-p Confdir_list=/global/iws/https-web \  
-p Port_list=443/tcp webserver-secure
```

In this example, the directory `/global/iws/https-web` must contain a file named `keypass`, which contains the secure key password needed to start the secure web server.

EXAMPLE 4 Creating a Scalable iWS Resource That Contains Two iWS Instances

For this example to work, the data service must first be installed. This example creates a scalable iWS resource named `webserver-paired`, which contains two iWS instances, one secure and the other non secure. The probe timeout is reduced from the default value of 30 seconds to 20 seconds. The resource listens on the IP addresses contained in two `SharedAddress` resources named `www_foo_com` and `www_foobar_com`. The `Load_balancing_policy` is set to be “sticky” so that a given client always goes to the same cluster node irrespective of whether it contacts the secure port or the non secure one.

EXAMPLE 4 Creating a Scalable iWS Resource That Contains Two iWS Instances *(Continued)*

```
example# clresourcetype register SUNW.iws
example# clresource create -g web-rg -t SUNW.iws \
-p Confdir_list=/global/iws/https-web-not-secure,/global/iws/https-web-secure \
-p Port_list=80/tcp,443/tcp -p Probe_timeout=20 -y Scalable=TRUE \
-p Network_resources_used=www_foo_com,www_foobar_com
\ -p Load_balancing_policy=LB_STICKY webserver-paired
```

EXAMPLE 5 Setting Monitor_uri_list for Scalable Sun Java System Web Server Instance

The following example shows how to set the Monitor_uri_list extension property when a scalable Sun Java System Web Server instance is added to a configuration. The Monitor_uri_list extension property is not supported with a secure Sun Java System Web Server instance.

(Add an insecure Sun Java System application resource instance with default load balancing.)

```
example# clresource create -g resource-group-1 -t SUNW.iws \
-p Confdir_List=/opt/SunONE/https-web-not-secure-1 \
-p Monitor_Uri_list=http://schost-1:8000/servlet/monitor \
-p Scalable=True -y Network_resources_used=schost-1 \
-p Port_list=8000/tcp SunONE-insecure-1
```

EXAMPLE 6 Setting Monitor_uri_list for Failover Sun Java System Web Server Instance

The following example shows how to set the Monitor_uri_list extension property when a failover Sun Java System Web Server instance is added to a configuration. The Monitor_uri_list extension property is not supported with a secure Sun Java System Web Server instance.

(Add an insecure Sun Java System application resource instance.)

```
example# clresource create -g resource-group-1 -t SUNW.iws \
-p Confdir_list=/opt/SunONE/conf \
-p Monitor_Uri_list=http://schost-1:80/servlet/monitor \
-p Scalable=False -y Network_resources_used=schost-1 \
-p Port_list=80/tcp web-not-secure-1
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/iplanet-web-server

See Also [clresourcetype\(1CL\)](#), [clresourcegroup\(1CL\)](#), [scha_resource_get\(1HA\)](#), [pmfadm\(1M\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [scalable_service\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.jsas, jsas – resource type implementation for failover and multiple masters Sun Java System Application Server

Description The SUNW.jsas resource type represents the Sun Java System Application Server application in an Oracle Solaris Cluster configuration.

Standard properties and extension properties that are defined for the SUNW.jsas resource type are described in the subsections that follow. To set these properties for an instance of the SUNW.jsas resource type, use the `clresource(1CL)` commands.

Standard Properties See [r_properties\(5\)](#) for a complete description of the following resource properties.

Network_resources_used (string array)

A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties `Resource_dependencies`, `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart`.

This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the `Resource_dependencies` property.

Category Conditional/Optional

Default The empty list

Tunable When disabled

Resource_dependencies (string array)

Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the [clreslogicalhostname\(1CL\)](#) and [clressharedaddress\(1CL\)](#) man pages for more information.

You can specify an alternate kind of dependency by using the `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart` property instead of the `Resource_dependencies` property. For more information, see the `r_properties(5)` man page.

Category Optional

Default The empty list

Tunable Any time

Port_list**Default** Null string**Tunable** Any time**Retry_count****Maximum** 10**Default** 2**Tunable** Any time**Retry_interval****Maximum** 3600**Default** 1220**Tunable** Any time**Thorough_probe_interval****Maximum** 3600**Default** 120**Tunable** Any time

Extension Properties The extension properties of this resource type are as follows:

Adminuser

The DAS administrative user name.

Data type String array**Default** No default defined**Range** Not applicable**Tunable** At creation**Confdir_list**

The complete path to the Sun Java System Application Server installation directory.

Data type String array**Default** /opt/SUNWappserver**Range** Not applicable**Tunable** At creation**Domaindir**

The full path to the domain directory.

Data type String
Default Null
Range Not applicable
Tunable At creation

Domain_name
The domain name.

Data type String
Default No default defined
Range Not applicable
Tunable At creation

Monitor Uri List
List of URIs to be probed. The Application Server agent sends HTTP/1.1 GET requests to each of the listed URIs.

The only response code that results in a failover of the resource is the response code 500 (Internal Server Error).

Data type String array
Default Null
Range Not applicable
Tunable Any time

Passwordfile
The full path to the file that contains the DAS administrative password.

Data type String
Default No default defined
Range Not applicable
Tunable At creation

Probe_timeout
The timeout value (in seconds) for the probe.

Data type Integer
Default 180
Range Minimum = 2
Tunable Any time

Examples EXAMPLE 1 Creating a Resource for SUNW.jsas

For this example to work, you must first install the Oracle Solaris Cluster HA for Sun Java System Application Server data service. This data service includes all the packages to make Sun Java System Application Server highly available.

This example creates a failover Domain Administration Server (DAS) resource. The resource is created in an existing failover resource group. To create this resource the following commands are run:

```
# clresourcetype register SUNW.jsas
# clresource create -g das-rg -t SUNW.jsas \
-p Adminuser=admin \
-p Domain_name=new-domain \
-p Passwordfile=/global/disk1/passwordfile das-rs
```

In this example, the DAS resource group is named `das-rg`, the DAS resource is named `das-rs`, and values are specified for the extension properties that have no defaults.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/glassfish-app-server

See Also [clresourcetype\(1CL\)](#), [clresourcegroup\(1CL\)](#), [scha_resource_get\(3HA\)](#), [pmfadm\(1M\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [scalable_service\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.jsas-na, jsas-na – resource type implementation Sun Java System Application Server

Description The SUNW.jsas-na resource type represents the Node Agent component of the Sun Java System Application Server application in an Oracle Solaris Cluster configuration.

Standard properties and extension properties that are defined for the SUNW.jsas-na resource type are described in the subsections that follow. To set these properties for an instance of the SUNW.jsas-na resource type, use the `clresource(1CL)` command.

Standard Properties See `r_properties(5)` for a complete description of the following resource properties.

Network_resources_used (string array)

A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties `Resource_dependencies`, `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart`.

This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the `Resource_dependencies` property.

Category Conditional/Optional

Default The empty list

Tunable When disabled

Resource_dependencies (string array)

Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the `clreslogicalhostname(1CL)` and `clressharedaddress(1CL)` man pages for more information.

You can specify an alternate kind of dependency by using the `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart` property instead of the `Resource_dependencies` property. For more information, see the `r_properties(5)` man page.

Category Optional

Default The empty list

Tunable Any time

Port_list

Default Null string

Tunable Any time

Retry_count

Maximum 10

Default 2

Tunable Any time

Retry_interval

Maximum 3600

Default 1220

Tunable Any time

Thorough_probe_interval

Maximum 3600

Default 120

Tunable Any time

Extension Properties The extension properties of this resource type are as follows:

Adminhost

The host name of the Domain Administration Server.

Data type String

Default Null string

Range Not applicable

Tunable When disabled

Adminport

The port on which the administration server is listening.

Data type Integer

Default 4849

Range Not applicable

Tunable Any time

Adminuser

The Domain Administration Server (DAS) administrative user name.

Data type String

Default Null string

Range Not applicable

Tunable When disabled

Agentdir

The full path to the Node Agents directory.

Data type String

Default Null string

Range Not applicable

Tunable When disabled

Confdir_list

The full path to the Sun Java System Application Server installation directory.

Data type String array

Default /opt/SUNWappserver

Range Not applicable

Tunable When disabled

Passwordfile

The full path to the file that contains the DAS administrative password and master password.

Data type String

Default Null string

Range Not applicable

Tunable When disabled

Probe_timeout

The timeout value (in seconds) for the probe.

Data type Integer

Default 180

Range Minimum = 2

Tunable Any time

Examples EXAMPLE 1 Creating a Resource for SUNW.jsas-na

For this example to work, you must first install the Oracle Solaris Cluster HA for Sun Java System Application Server data service. This data service includes all the packages to make Sun Java System Application Server highly available.

EXAMPLE 1 Creating a Resource for SUNW.jsas-na (Continued)

This example creates a failover Node Agent resource. The resource is created in an existing failover resource group. To create this resource the following commands are run:

```
# clresourcetype register SUNW.jsas-na
# clresource create -g na-rg -t SUNW.jsas-na \
-p Agentdir=/global/disk1/my-domain \
-p Adminuser=admin \
-p Adminhost=host1 \
-p Passwordfile=/global/disk1/passwordfile na-rs
```

In this example, the Node Agent resource group is named na-rg, the Node Agent resource is named na-rs, and values are specified for the extension properties that have no defaults.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/glassfish-app-server

See Also [clresourcetype\(1CL\)](#), [clresourcegroup\(1CL\)](#), [scha_resource_get\(3HA\)](#), [pmfadm\(1M\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [scalable_service\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.krb5, krb5 – resource type implementation of the Kerberos KDC server

Description SUNW.krb5 is the resource type that uses the SUNW.sckrb5/etc/SUNW.krb5 file to store the resource properties required to drive the high-availability of the Kerberos server.

Standard Properties Standard resource properties are overridden for this resource type as follows:

Cheap_probe_interval

Maximum	3600 seconds
Minimum	10 seconds
Default	30 seconds
Tunable	Any time

Network_resources_used (string array)

A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties Resource_dependencies, Resource_dependencies_weak, Resource_dependencies_restart, or Resource_dependencies_offline_restart.

This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the Resource_dependencies property.

Category Conditional/Optional

Default The empty list

Tunable When disabled

Resource_dependencies (string array)

Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the [clreslogicalhostname\(1CL\)](#) and [clressharedaddress\(1CL\)](#) man pages for more information.

You can specify an alternate kind of dependency by using the Resource_dependencies_weak, Resource_dependencies_restart, or Resource_dependencies_offline_restart property instead of the Resource_dependencies property. For more information, see the [r_properties\(5\)](#) man page.

Category Optional

Default The empty list

Tunable Any time

Port_list

Note – Port 88 is associated with krb5kdc(1M) and port 749 belongs to kadmind(1M).

Maximum	Not applicable
Default	88/tcp, 749/tcp, and 88/udp
Tunable	At creation

Retry_count

Maximum	10
Default	2
Tunable	Any time

Retry_interval

Maximum	3600 seconds
Default	1570 seconds
Tunable	Any time

Thorough_probe_interval

Maximum	3600 seconds
Default	300 seconds
Tunable	Any time

For more information about standard properties, see the [r_properties\(5\)](#) man page.

Extension Properties The extension properties associated with the SUNW.krb5 resource type are as follows:

Monitor_retry_count

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

Data type	Integer
Default	4
Range	No range defined
Tunable	Any time

Monitor_retry_interval

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

Data type	Integer
------------------	---------

Default	2 minutes
Range	No range defined
Tunable	Any time

Probe_timeout

The time-out value in seconds that the fault monitor uses to probe a Kerberos instance.

Data type	Integer
Minimum	2
Default	90 seconds
Range	No range defined
Tunable	Any time

Examples EXAMPLE 1 Instantiating a Failover Kerberos Resource

This example shows how to instantiate a failover Kerberos resource.

Before you work through this example, ensure that the Oracle Solaris Cluster HA for Kerberos is installed.

Instantiate a Kerberos resource named `krb5-rs` in a resource group named `krb5-rg`. The `krb5-rg` resource group contains at least one logical hostname resource which identifies the logical hostnames associated with the resource group.

```
# clresourcetype register SUNW.krb5
# clresource create -g krb5-rg -t SUNW.krb5 krb5-rs
```

The resource `krb5-rs` listens on port 88 for `krb5kdc` and 749 for `kadmind`.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/kerberos

See Also [kinit\(1\)](#), [kadmin\(1M\)](#), [kadmind\(1M\)](#), [krb5kdc\(1M\)](#), [scrgadm\(1M\)](#), [svcadm\(1M\)](#), [r_properties\(5\)](#),

Name SUNW.Idom, Idom – resource type implementation for failover guest domains.

Description The LDoms Guest Domains data service for Oracle Solaris Cluster is configured as a resource managed by the Oracle Solaris Cluster Resource Group Manager (RGM).

You must set the following properties on an LDoms resource by using `clresource(1CL)` man page.

Standard Properties The standard resource property `Failover` is set for all failover resource types.

`Resource_dependencies` (string array)

Specifies a list of resources upon which a resource depends. This list includes any `HASStoragePlus` resources that are used by a resource. The default value for this property is null.

You can specify an alternate kind of dependency by using the `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart` property instead of the `Resource_dependencies` property. For more information, see the `r_properties(5)` man page.

Category Optional

Default The empty list

Tunable Any time

`Failover_mode` (enum)

Default SOFT

Tunable When disabled

`Retry_count` (integer)

Default 2

Tunable When disabled

`Retry_interval` (integer)

Default 370

Tunable When disabled

`Thorough_probe_interval` (integer)

Default 60

Tunable Any time

Extension Properties `Domain_name` (string)

Indicates the name of the LDom's guest domain.

Category Required

Default No default

Tunable When disabled

`Migration_type` (enum)

This property specifies the type of migration when the resource is being switched over or a failover is triggered. Possible values are MIGRATE and NORMAL.

Category Optional

Default MIGRATE

Tunable Any time

`Plugin_probe` (string)

This property specifies the command that checks the status of the logical-guest domain. This should return with an exit status of 0 if the guest domain is running properly. The probe status is used within the HA-xVM probing algorithm to determine whether to restart the guest domain locally or to failover the guest domain to another node.

Category Optional

Default Null

Tunable Any time

`password_file` (string)

This property specifies the complete path to the file containing the target host password required for guest domain migration.

Category Required if the `Migration_type` property is set to MIGRATE.

Default No default

Tunable When disabled

`Log_level` (enum)

Specifies the level, or type of diagnostic messages that are logged by the LDom's data service. You can specify None, Info, or Err for this property. When you specify None, diagnostic messages are not logged by the LDom's data service. When you specify Info, both information and error messages are logged. When you specify Err, only error messages are logged.

Category Optional

Default Info

Tunable Any time

Debug_level (integer)

This property indicates the level to which the debug messages of the Solaris Cluster agent for LDOMs guest domain component are logged.

Category	Required
Default	Info
Tunable	When disabled

Monitor_retry_count (integer)

This property controls the restarts of the fault monitor. It indicates the number of times the fault monitor is restarted by the process monitor facility and corresponds to the `-n` option passed to the `pmfadm(1M)` command. The number of restarts is counted in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not guest domains. The restarts of the guest domain resources are controlled by the system-defined properties `Thorough_Probe_Interval`, `Retry_Interval`, and `Retry_Count`, as specified in their descriptions. See `clresource(1CL)`. You can modify the value for this property at any time.

Default	4
Tunable	Any time

Monitor_retry_interval (integer)

Indicates the time (in minutes) over which the failures of the fault monitor are counted and corresponds to the `-t` option passed to the `pmfadm(1M)` command. If the number of times the fault monitor fails exceeds the value of `Monitor_retry_count`, the fault monitor is not restarted by the process monitor facility. You can modify the value for this property at any time.

Default	2
Tunable	Any time

Probe_timeout (integer)

Indicates the time-out value (in seconds) used by the fault monitor to probe an LDOM instance. You can modify the value for this property at any time.

Default	30
Tunable	Any time

Examples EXAMPLE 1 Creating a Failover LDOMs Guest-Domain Resource

The following example creates a failover LDOMs guest-domain resource named `ldom-rs` in an existing resource group named `ldom-rg`. In this example, `ldg1` is the domain name of the `ldom-rs` resource. The fault monitor of the LDOMs guest-domain resource uses the `plugin_probe` script to check the status of the LDOMs guest domain.

EXAMPLE 1 Creating a Failover LDOMs Guest-Domain Resource *(Continued)*

```
example# clresourcetype register SUNW.ldom
example# clresource create -g ldom-rg -t SUNW.ldom \
-p Domain_name=ldg1 -p Password_file=/global/ldom/pass \
-p Plugin_probe="/opt/SUNWscxvm/bin/ppkssh -P \
fmuser:/export/fmuser/ .ssh/id_dsa:ldg1:multi-user-server:online" \
ldom-rs
```

Note – You should install the LDOMs guest domain data service before performing the steps in the example.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/dns

See Also [pmfadm\(1M\)](#), [scha_resource_get\(3HA\)](#), [clresourcetype\(1CL\)](#), [clresourcegroup\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.nfs, nfs – resource type implementation for Oracle Solaris Cluster HA for NFS

Description The nfs resource type implementation operates on a set of share commands stored in a “per-resource” file. The format of this file is that each line consists of a [share_nfs\(1M\)](#) command. This file’s location is relative to the Pathprefix property of the containing resource group. This file must reside as SUNW.nfs/dfstab.*resource-name* under the Pathprefix directory that contains the resource group.

Standard Properties See [r_properties\(5\)](#) for a complete description of the following resource properties.

Cheap_probe_interval

Minimum 10

Maximum 3600

Default 20

Tunable Any time

Network_resources_used (string array)

A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties Resource_dependencies, Resource_dependencies_weak, Resource_dependencies_restart, or Resource_dependencies_offline_restart.

This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the Resource_dependencies property.

Category Conditional/Optional

Default The empty list

Tunable When disabled

Resource_dependencies (string array)

Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the [clreslogicalhostname\(1CL\)](#) and [clressharedaddress\(1CL\)](#) man pages for more information.

You can specify an alternate kind of dependency by using the Resource_dependencies_weak, Resource_dependencies_restart, or Resource_dependencies_offline_restart property instead of the Resource_dependencies property. For more information, see the [r_properties\(5\)](#) man page.

Category Optional

Default The empty list

Tunable Any time

Thorough_probe_interval

Minimum 60

Maximum 3600

Default 120

Tunable Any time

Extension Properties Lockd_nullrpc_timeout

Type integer. Default is 120. Minimum value is 60. This property indicates the time out value (in seconds) to use when probing lockd.

Monitor_retry_count

Type integer. Default is 4. Minimum value is -1. This property controls the restarts of the fault monitor. It indicates the number of times that the fault monitor is restarted by the Process Monitor Facility (PMF) and corresponds to the -n option passed to the [pmfadm\(1M\)](#) command. The number of restarts is counted in a specified time window (see the property [Monitor_retry_interval](#)). Note that this property refers to the restarts of the fault monitor itself, not the NFS daemons.

Monitor_retry_interval

Type integer. Default is 2. Minimum value is -1. This property indicates that the failures of the fault monitor are counted and corresponds to the -t option passed to the [pmfadm\(1M\)](#) command. If the number of times the fault monitor fails exceeds the extension property [Monitor_retry_count](#), the fault monitor is not restarted by the Process Monitor Facility.

Mountd_nullrpc_restart

Type Boolean. Default is TRUE. Indicates if mountd should be restarted when a null rpc call fails.

Mountd_nullrpc_timeout

Type integer; defaults to 120. Minimum value is 60. This property indicates the time out value (in seconds) to use when probing mountd.

Nfsd_nullrpc_restart

Type Boolean. Default is FALSE. This property indicates if nfsd should be restarted when a null rpc call fails.

Nfsd_nullrpc_timeout

Type integer. Default is 120. Minimum value is 60. This property indicates the time out value (in seconds) to use when probing nfsd.

Rpcbind_nullrpc_reboot

Type Boolean. Default is FALSE. Indicates if the system is to be rebooted when a null rpc call on rpcbind fails.

Rpcbnd_nullrpc_timeout

Type integer. Default is 120. Minimum value is 60. This property indicates the time out value (in seconds) to use when probing rpcbind.

Statd_nullrpc_timeout

Type integer. Defaults to 120. Minimum value is 60. This property indicates the time out value (in seconds) to use when probing statd.

Files *dfstab.resource-name* The file is in *dfstab* format, which contains the list of share commands to be managed by the resource. This file must reside in the *SUNW.nfs* subdirectory under the *Pathprefix* directory of the containing resource group.

*/tmp/.hanfs/** Critical state files used by the implementation.

Examples EXAMPLE 1 Instantiating a Failover NFS Resource

For this example to work, the data service must first be installed. This example instantiates a failover NFS resource named *hanfs-rs* in a resource group named *hanfs-rg*. The *hanfs-rg* resource group is assumed to contain at least one logical hostname resource, which identifies the logical hostnames associated with the resource group.

```
example# clresourcetype register SUNW.nfs
example# clresource create -g hanfs-rg -t SUNW.nfs hanfs-rs
```

The resource group *hanfs-rg* must contain a valid path name as its *Pathprefix* property. A file named *dfstab.hanfs-rs* must reside in the subdirectory *SUNW.nfs* under the *Pathprefix* directory.

EXAMPLE 2 Instantiating a Failover NFS Resource

For this example to work, the data service must first be installed. This example instantiates a failover NFS resource named *sap-nfs* in a resource group named *sap-rg*. The system-defined property *Thorough_probe_interval* is set to 30 for this resource. The *Network_resources_used* property is set to a logical hostname *relo-sap*, which must reside in the same resource group, *sap-rg*.

```
example# clresourcetype register SUNW.nfs
example# clresource create -g sap-rg -t SUNW.nfs \
-p Thorough_probe_interval=30 \
-p Network_resources_used=relo-sap sap-nfs
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/nfs

See Also `lockd(1M)`, `mountd(1M)`, `nfsd(1M)`, `pmfadm(1M)`, `rpcbind(1M)`, `scha_resource_get(3HA)`, `clresourcetype(1CL)`, `clresourcegroup(1CL)`, `share(1M)`, `statd(1M)`, `rpc(3NSL)`, `share_nfs(1M)`, `attributes(5)`, `r_properties(5)`

Oracle Solaris Cluster Data Services Planning and Administration Guide

Notes The path names being shared by means of `dfstab.resource-name` must be unique across all resources, and they cannot be present in the system `dfstab` file on any cluster node.

The implementation supports customization of the `/etc/init.d/nfs.server` script to start the `nfsd` daemon with a customized set of options.

The `SUNW.nfs` subdirectory under the `Pathprefix` directory of the containing resource group is also used by `statd` to save its state.

Name SUNW.oracle_listener, oracle_listener – resource type implementation for the Oracle listener

Description The SUNW.oracle_listener resource type represents the Oracle listener in an Oracle Solaris Cluster configuration. The HA Oracle listener resource is configured with an HA Oracle server resource..

You must set the following properties for an Oracle listener resource by using `clresource(1CL)`.

Standard Properties The standard resource property `FaiLover` is set for all failover resource types.

See [r_properties\(5\)](#) for a complete description of the following resource properties.

`Failover_mode`

Default: NONE

Tunable: Any time

`Retry_count`

Minimum: -1

Default: -1

Tunable: Any time

`Retry_interval`

Minimum: -1

Maximum: 2592000

Default: 600

Tunable: Any time

`Thorough_probe_interval`

Minimum: 1

Maximum: 2592000

Default: 30

Tunable: Any time

Extension Properties `Listener_name`

Type string. Default is LISTENER. Defines the name of the listener to be started. This name must match the corresponding entry in the `listener.ora` configuration file. You can change this property only when the resource is disabled.

`Oracle_Home`

Type string. This property is set to the Oracle parent directory that contains the binaries, logs, and parameter files. You can modify this property only when the resource is disabled.

Probe_timeout

Type integer. Default is 180. Maximum value is 99999. The time-out value in seconds that the fault monitor uses to probe an Oracle listener. You can modify this property at any time.

User_env

Type string. Default is NULL. This property is set to the name of the file that contains the environment variables to be set before listener startup or shutdown. You can modify this property at any time.

Examples EXAMPLE 1 Creating a Failover oracle_listener Resource

For this example to work, you must first install the data service.

The following example creates a failover oracle_listener resource named ora_listener in an existing resource group named oracle-rg. oracle-rg is assumed to contain a LogicalHostname resource, which identifies the logical host name associated with the resource group, and an oracle_server resource, which identifies the Oracle server associated with the resource group. Another assumption is that the configuration file listener.ora was created with the correct port number for the listener to listen at.

```
example# clresourcetype register SUNW.oracle_listener
example# clresource create -g oracle-rg \
-t SUNW.oracle_listener -p ORACLE_HOME=/oracle \
-p LISTENER_NAME=ORALIST ora_listener
```

In this example, the Oracle listener resource created is named ora_listener, which has its ORACLE_HOME under /oracle. The listener name is ORALIST, which matches the corresponding entry in the configuration file listener.ora.

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [r_properties\(5\)](#), [SUNW.oracle_server\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.oracle_server, oracle_server – resource type implementation for HA Oracle server

Description The SUNW.oracle_server resource type represents the HA Oracle server in an Oracle Solaris Cluster configuration. The HA Oracle server resource is configured with an Oracle listener resource..

You must set the following properties for an Oracle server resource by using `clresource(1CL)`.

Standard Properties The standard resource property `Failover` is set for all failover resource types.

See `r_properties(5)` for a complete description of the following resource properties.

`Failover_mode`

Default: SOFT

Tunable: Any time

`Retry_count`

Minimum: -1

Maximum: 9999

Default: 2

Tunable: Any time

`Retry_interval`

Minimum: 0

Maximum: 2592000

Default: 1330

Tunable: Any time

`Thorough_probe_interval`

Minimum: 1

Maximum: 2592000

Default: 30

Tunable: Any time

Extension Properties `Alert_log_file`

Type string. This property is set to the absolute path of the Oracle alert log file. The Oracle software logs alerts in this file. The Oracle server fault monitor scans the alert log file for new alerts at the following times:

- When the server fault monitor is started

- Each time that the server fault monitor queries the health of the server

If an action is defined for a logged alert that the server fault monitor detects, the server fault monitor performs the action in response to the alert.

You can modify this parameter any time.

When the `Db_unique_name` extension property is used, the `Alert_log_file` extension property is ignored and can be omitted from the HA-Oracle configuration.

Auto_end_bkp

Type Boolean. Default is FALSE. This property instructs the `oracle_server` START method to automatically recover the database during startup if the database had previously been interrupted during a hot backup.

If this property is set to TRUE, and the `oracle_server` START method detects the RDBMS error `ORA-01113 file %s needs media recovery` during startup, all files left in hot backup mode (as indicated by the `sys.v$backup` view). These files are automatically taken out of hot backup mode by using the command:

```
alter database datafile 'filename' end backup;
```

The database is then opened for use.

If this property is set to FALSE, the `oracle_server` START method takes no recovery action following an `ORA-01113` error, and the status of the resource is set to `FAULTED`. Manual intervention is required at this stage.

This property can be modified at any time.

Connect_cycle

Type integer. Default is 5. The Maximum value is 99999. The Oracle server fault monitor connects to the database periodically by using the user ID and password specified in `Connect_string`. The monitor disconnects after executing the number of probes specified in this property and then reconnects. You can modify this property at any time.

Connect_string

Type string. This property is set to the user ID and password of the database user in fault-monitor transactions. This property is specified as follows:

```
userid/password
```

As part of the HA Oracle setup, you must define the database user ID and password before enabling the server resource and its fault monitor. To use Solaris authentication, type a slash (/) instead of a user ID and password. This property must be set for standby databases as well. This property is used by the fault monitor after the physical standby database is transitioned to a primary database. You can modify this property at any time.

When the `Db_unique_name` extension property is used, the `Connect_string` extension property is ignored and can be omitted from the HA-Oracle configuration.

Custom_action_file

Type string. Default is NULL. This property specifies the absolute path of the file that defines the custom behavior of the Oracle Solaris Cluster HA for Oracle server fault monitor. The format of this file is defined in the [custom_action_file\(4\)](#) man page. You can modify this property at any time.

Dataguard_role

Type string. Default is NULL. This property specifies the role of the database. The permitted values for this property are as follows:

NONE

Specifies that no standby database instances are configured for the primary database instance.

PRIMARY

Specifies that the database is a primary database instance for which standby database instances are configured.

STANDBY

Specifies that the database role is standby. This value is used by Oracle Solaris Cluster HA for Oracle data service along with the Standby_mode property value to determine the role of the database.

IN_TRANSITION

Specifies that the database is undergoing a role reversal process. This value must be set, when a role reversal process is to be applied to the database. The Dataguard_role and Standby_mode properties must be set after the role reversal process is complete to reflect the correct role of the database.

You can modify this property at any time.

Db_unique_name

Type string. This property is required when a single-instance HA for Oracle data service is deployed using Oracle 11g release 2 Grid Infrastructure for Clusters. The db_unique_name extension property specifies the unique name of the single-instance database that is being deployed.

When the db_unique_name extension property is used, the following extension properties are ignored and can be omitted from the HA-Oracle configuration:

- Alert_log_file
- Connect_string

You can modify this property at any time.

Debug_level

Type integer. Default is 1. Maximum value is 100. This property indicates the level to which debug messages from the fault monitor of the Oracle server component are logged. When the debug level is increased, more debug messages are written to the log files. These

messages are logged to the file `/var/opt/SUNWscor/oracle_server/message_log.rs`, where *rs* is the name of the resource that represents the Oracle server component. You can modify this property at any time.

Oracle_Home

Type string. This property is set to the Oracle parent directory that contains the binaries, logs, and parameter files. You can modify this property only when the resource is disabled.

Oracle_Sid

Type string. This property is set to the name of the Oracle database instance (also called the Oracle system identifier). You can modify this property only when the resource is disabled.

Parameter_file

Type string. This property is set to the Oracle parameter file, which starts the database. If this property is not set, it defaults to `$ORACLE_HOME/dbs/init$ORACLE_SID.ora`. If the default value is not found, Oracle Solaris Cluster HA for Oracle checks for `$ORACLE_HOME/dbs/spfile$ORACLE_SID.ora`. You can modify this property at any time.

Probe_timeout

Type integer. Default is 300 seconds. Maximum value is 99999. This property is the timeout value (in seconds) that is used by the fault monitor to probe an Oracle server instance. You can modify this property at any time.

Restart_type

Type enumeration. Default is `RESOURCE_RESTART`. This property specifies the entity that the server fault monitor restarts when the response to a fault is restart. The permitted values for this property are as follows:

<code>RESOURCE_RESTART</code>	Specifies that only this resource is restarted.
<code>RESOURCE_GROUP_RESTART</code>	Specifies that all resources in the resource group that contains this resource are restarted.

You can modify this property at any time.

Standby_mode

Type string. Default is `LOGICAL`. This property specifies the mode of the standby database. This property is used by Oracle Solaris Cluster HA for Oracle data service only when the `Dataguard_role` property is set to `STANDBY` to determine the type of standby database. The permitted values for this property are as follows:

<code>LOGICAL</code>	Specifies a logical standby database.
<code>PHYSICAL</code>	Specifies a physical standby database.

You can modify this property at any time.

User_env

Type string. Default is NULL. This property is set to the name of the file that contains the environment variables to be set before database startup or shutdown. You can modify this property at any time.

Wait_for_online

Type Boolean. Default is TRUE. This property specifies whether the oracle_server START method waits for the database to be online before exiting. If this property is set to FALSE, oracle_server START executes the commands to start the database but does not wait for it to come online before exiting. You can modify this property at any time.

Examples EXAMPLE 1 Creating a Failover oracle_server Resource

For this example to work, the data service must first be installed.

The following example creates a failover oracle_server resource named ora_server in an existing resource group named oracle-rg. oracle-rg is assumed to contain a LogicalHostname resource, which identifies the logical hostname associated with the resource group, and an oracle_listener resource, which identifies the Oracle listener associated with the resource group.

```
example# clresourcetype register SUNW.oracle_server
example# clresource create -g oracle-rg \
-t SUNW.oracle_server -p CONNECT_STRING=scott/tiger \
-p ORACLE_SID=oraSID -p ORACLE_HOME=/oracle \
-p ALERT_LOG_FILE=/oracle/admin/oraSID/bdump/alert_oraSID.log ora_server
```

In this example, the Oracle server resource created is named ora_server, which has its ORACLE_HOME under /oracle. The SID of the is ora_server is oraSID. Its fault monitor uses the user ID scott and the password tiger to connect to the database. The alert log file scanned by the fault monitor for any errors that have occurred is at /oracle/admin/oraSID/bdump/alert_oraSID.log.

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [custom_action_file\(4\)](#), [SUNW.oracle_listener\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.s1as, s1as – resource type implementation for failover and multiple masters Sun Java System Application Server (s1as)

Description The SUNW.s1as resource type represents the Sun Java System Application Server application in an Oracle Solaris Cluster configuration.

You must set the following properties on an SUNW.s1as resource by using [scrgadm\(1M\)](#).

Standard Properties See [r_properties\(5\)](#) for a complete description of the following resource properties.

`Network_resources_used` (string array)

A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties `Resource_dependencies`, `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart`.

This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the `Resource_dependencies` property.

Category Conditional/Optional

Default The empty list

Tunable When disabled

`Resource_dependencies` (string array)

Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the [clreslogicalhostname\(1CL\)](#) and [clressharedaddress\(1CL\)](#) man pages for more information.

You can specify an alternate kind of dependency by using the `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart`

property instead of the Resource_dependencies property. For more information, see the r_properties(5) man page.

Category Optional

Default The empty list

Tunable Any time

Port_list

Default: No default

Tunable: Any time

Retry_count

Maximum: 10

Default: 2

Tunable: Any time

Retry_interval

Default: 3600

Default: 1220

Tunable: Any time

Thorough_probe_interval

Default: 3600

Default: 120

Tunable: Any time

Extension Properties Confdir_list

Type string array. This property is a path name set to *install-dir/domains/domain/server* which is the path name of an s1as instance directory. You must specify this property at resource creation time.

Monitor Uri_List

Type string array; defaults to "". This property allows you to ensure that application components are responding by querying the configured URIs. The Monitor Uri_List property is used for detailed fault monitoring of Sun Java System Application Server. The fault monitor periodically performs HTTP GET command for the URIs. The monitor takes action if the HTTP request returns with response code 500 "Internal Server Error" or if the application server does not respond. An example URI setting is `http://logical-hostname/App/tester`. If the configured URIs are implemented by using a Servlet in the application server, detailed monitoring of the application server JVM (Java Virtual Machine) is possible. Either the Monitor Uri_List or the Network_Resources_Used and the Port_List properties must be set. If Network_Resources_Used, Port_List, and Monitor Uri_List are all set, the fault

monitor will probe the ports and the URIs provided. Setting `Port_List` to include the IIOP listeners helps to ensure that the application server is listening and responding to IIOP requests.

`Probe_timeout`

Type string; defaults to 120 seconds. Minimum is 2. This property is tunable anytime and sets the timeout value for the probe.

Examples EXAMPLE 1 Creating a Failover s1as Resource in an Existing Group

For this example to work, you must first install the `SUNWscs1as` data service package. This example creates a failover `s1as` resource named `appsrv-rs` in an existing resource group named `appsrv-rg`. The `appsrv-rg` resource group contains a `LogicalHostname` resource, which identifies the logical hostname associated with the resource group.

(Register the `SUNW.s1as` resource type.)

```
# scrgadm -a -t SUNW.s1as
```

(Create a Sun Java System Application Server resource and add it to the resource group.)

```
# scrgadm -a -j appsrv-rs -g appsrv-rg \\  
-t SUNW.s1as \\  
-x Confdir_list=/global/appsrv/domains/scdomain/server1 \\  
-y Network_resources_used=schost-1 \\  
-y Port_list=80/tcp,3700/tcp \\  
-x Monitor Uri_list=http://schost-1:80/servlet/monitor
```

In the preceding example, the `s1as` resource created is named `appserver-rs`, with a corresponding `s1as` instance in the directory `/global/appsrv/domains/scdomain/server1`.

EXAMPLE 2 Creating a Failover s1as Resource Listening on a Specified Port

For this example to work, you must first install the `SUNWscs1as` data service package. This example creates a failover `s1as` resource named `appserver-secure`, which listens on port 443 in an existing resource group named `app-rg`.

```
example# scrgadm -a -t SUNW.s1as  
example# scrgadm -a -j appserver-secure -t SUNW.s1as  
-g app-rg  
-x Confdir_list=/global/s1as/domains/domain1/server2  
-y Network_resources_used=schost-1  
-x Monitor Uri_list=http://schost-1:80/servlet/monitor  
-y Port_list=443/tcp
```

EXAMPLE 3 Configuring s1as in a Scalable Resource Group

In this example a scalable resource group, `s1as-rg`, is created with `Maximum primaries` and `Desired primaries` equal to three. A Sun Java System Application Server resource named `scalable-app-server1` is then created in this resource group with a configuration directory

EXAMPLE 3 Configuring slas in a Scalable Resource Group *(Continued)*

of /global/slas/domains/domain1/server1. The URI `http://localhost:8000/servlets/testervlet` is specified to the extension property `Monitor_uri_list`.

```
example# scrgadm -a -g slas-rg
-y Maximum primaries=3 -y Desired primaries=3
example# scrgadm -a -g slas-rg -j scalable-app-server1 -t SUNW.slas
-x Confdir_list=/global/slas/domains/domain1/server1
-x Monitor_uri_list=http://localhost:8000/servlets/testervlet
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/glassfish-app-server

See Also [scrgadm\(1M\)](#), [scswitch\(1M\)](#), [scha_resource_get\(3HA\)](#), [pmfadm\(1M\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [scalable_service\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.sap_as, sap_as, SUNW.sap_as_v2, sap_as_v2 – resource type implementations for Oracle Solaris Cluster HA for SAP application server

Description The sap_as or SUNW.sap_as resource type represents Oracle Solaris Cluster HA for SAP as a failover data service.

The sap_as_v2 or SUNW.sap_as_v2 resource type represents Oracle Solaris Cluster HA for SAP as a failover data service or a scalable data service.

The Resource Group Manager (RGM) manages the SAP data service for Oracle Solaris Cluster software. If you are setting up the Oracle Solaris Cluster HA for SAP application server as a failover data service, configure it as a logical hostname resource and an SAP application server resource. If you are setting up the Oracle Solaris Cluster HA for SAP application server as a scalable data service, configure it as a scalable SAP application server resource.

Use the `clresource(1CL)` command or a resource configuration GUI to set the following properties on an SAP application server resource.

Standard Properties See [r_properties\(5\)](#) for a complete description of the following resource properties.

Failover_mode	Default: SOFT Tunable: Any time
Retry_count	Default: 3 Tunable for SUNW.sap_as_v2: When disabled Tunable for SUNW.sap_as: Any time
Retry_interval	Default: 3600 Tunable for SUNW.sap_as_v2: When disabled Tunable for SUNW.sap_as: Any time
Thorough_probe_interval	Default: 60 Tunable: Any time

Extension Properties SAPSID

Type string. No default exists for this field. You must provide the value when you create the resource. The value is the SAP system name or *SAPSID*. You can modify this property only when you have disabled the resource.

As_instance_id

Type string. This value is a two-digit SAP system number or instance ID. No default exists for this field. You must provide the value when you create the resource. You can modify this property only when you have disabled the resource.

As_services_string

Type string. The default is D, which is a string of services that the application server provides. You can modify this property only when you have disabled the resource.

Monitor_retry_count

Type integer. The default is 4. This property controls fault-monitor restarts. The property indicates the number of times the process monitor facility (PMF) restarts the fault monitor. The property corresponds to the -n option passed to the `pmfadm(1M)` command. The RGM counts the number of restarts in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not the SAP application server. You can modify the value for this property at any time.

Monitor_retry_interval

Type integer. The default is 2. This property indicates the time window in minutes during which the RGM counts fault-monitor failures. The property corresponds to the -t option passed to the `pmfadm(1M)` command. If the number of times the fault monitor fails exceeds the extension property `Monitor_retry_count`, the PMF does not restart the fault monitor. You can modify the value for this property at any time.

As_db_retry_interval

Type integer. The default is 30. This property indicates the time window in seconds to wait between attempts to connect to the database before starting the SAP application server at startup time, if the database is unavailable. You can modify this property only when you have disabled the resource.

As_startup_script

Type string. The Oracle Solaris Cluster HA for SAP data service uses the startup script name to start the SAP application server. No default exists for this field. You must supply the value when you create the resource. You can modify this property only when you have disabled the resource.

Stop_sap_pct

Type integer. The default is 95. This property indicates the percentage of the `Stop_timeout` value that the Oracle Solaris Cluster HA for SAP data service uses to stop SAP processes with the SAP shutdown script before calling PMF to terminate the SAP processes. You can modify this property only when you have disabled the resource.

As_shutdown_script

Type string. This value is the shutdown script name, which the Oracle Solaris Cluster HA for SAP data service uses to shut down the SAP application server. No default exists for this field. You must supply the value when you create the resource. You can modify this property only when you have disabled the resource.

Probe_timeout

The default is 120. This property indicates the timeout value in seconds for the probe. You can modify the value for this property at any time.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/sap

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [attributes\(5\)](#), [SUNW.sap_ci\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.sap_ci, sap_ci, SUNW.sap_ci_v2, sap_ci_v2 – resource type implementations for Oracle Solaris Cluster HA for SAP central instance.

Description The Resource Group Manager (RGM) manages the SAP data service for Oracle Solaris Cluster software. Configure the Oracle Solaris Cluster HA for SAP central instance as a logical hostname resource and an SAP central instance resource.

Use the `clresource(1CL)` command or a resource configuration GUI to set the following properties on an SAP central-instance resource.

Standard Properties See `r_properties(5)` for a complete description of the following resource properties.

Failover_mode	Default: SOFT
	Tunable: Any time
Retry_count	Default: 3
	Tunable: Any time
Retry_interval	Default: 3600
	Tunable: Any time
Thorough_probe_interval	Default: 60
	Tunable: Any time

Extension Properties SAPSID

Type string. No default exists for this field. You must provide the value when you create the resource. The value is the SAP system name or *SAPSID*. You can modify this property only when you have disabled the resource.

Ci_instance_id

Type string. The default is 00. This value is a two-digit SAP system number or instance ID. This value is a two-digit SAP system number or instance ID. You can modify this property only when you have disabled the resource.

Ci_services_string

Type string. The default is DVEBMGS, which is a string of services that the central instance provides. You can modify this property only when you have disabled the resource.

Message_server_name

Type string. No default value exists. This value is the name of the message server. You can modify this property only when you have disabled the resource.

Monitor_retry_count

Type integer. The default is 4. This property controls fault-monitor restarts. The property indicates the number of times the process monitor facility (PMF) restarts the fault monitor. The property corresponds to the `-n` option passed to the `pmfadm(1M)` command. The RGM

counts the number of restarts in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not the SAP central instance. You can modify the value for this property at any time.

`Monitor_retry_interval`

Type integer. The default is 2. This property indicates the time window in minutes during which the RGM counts fault-monitor failures. The property corresponds to the `-t` option passed to the `pmfadm(1M)` command. If the number of times the fault monitor fails exceeds the extension property `Monitor_retry_count`, the PMF does not restart the fault monitor. You can modify the value for this property at any time.

`Probe_timeout`

For `SUNW.sap_ci`, the default is 120. For `SUNW.sap_ci_v2`, the default value is 60. This property indicates the time-out value in seconds for the probes. You can modify the value for this property at any time.

`Check_ms_retry`

Type integer. The default is 2. This property indicates the maximum number of times the SAP message-server check can fail before the fault monitor reports a total failure. You can modify this property at any time.

`Ci_start_retry_interval`

Type integer. The default is 30. This property indicates the time window in seconds to wait between attempts to connect to the database before starting the SAP central instance at startup time, if the database is unavailable. You can modify this property only when you have disabled the resource.

`Ci_startup_script`

Type string. The Oracle Solaris Cluster HA for SAP data service uses the startup script name to start the SAP central instance. No default value exists. You must supply the value when you create the resource. You can modify this property only when you have disabled the resource.

`Stop_sap_pct`

Type integer. The default is 95. This property indicates the percentage of the `Stop_timeout` value that the Oracle Solaris Cluster HA for SAP data service uses to stop SAP processes with the SAP shutdown script before calling PMF to terminate the SAP processes. You can modify this property only when you have disabled the resource.

`Ci_shutdown_script`

Type string. This value is the shutdown script name, which the Oracle Solaris Cluster HA for SAP data service uses to shut down the SAP central instance. No default value exists. You must supply the value when you create the resource. You can modify this property only when you have disabled the resource.

`Lgtst_ms_with_logicalhostname`

Type boolean. The default is `TRUE`. This property indicates how to check the SAP message server with the `lgtst` utility. The `lgtst` utility requires a hostname (IP address) as the

location for the SAP message server. This hostname can be either an Oracle Solaris Cluster logical hostname or a localhost (loop back) name. If you set this resource property to TRUE, use a logical hostname. Otherwise, use a local hostname. This property indicates whether the fault monitor probes the message server using the utility `lgtst` with the logical hostname. If you change this value to FALSE, the fault monitor uses the loopback address instead of the logical hostname with the utility `lgtst`. You can modify the value for this property at any time.

Shutdown_dev

Type boolean. The default is FALSE. This property indicates whether the RGM shuts down a development system before starting the SAP central instance. If you set this value to TRUE, you must also set the extension properties `Dev_sapsid` and `Dev_shutdown_script`. You can modify this property only when you have disabled the resource.

Dev_sapsid

Type string. This value is the development system name or *SAPSID*. You must set this property if you set the property `Shutdown_dev` to the value TRUE. You can modify this property only when you have disabled the resource.

Dev_shutdown_script

Type string. This value is the name of the shutdown script, which the RGM uses to shut down the SAP development system. No default value exists. You must supply the value when you create the resource if you set the property `Shutdown_dev` to the value TRUE. You can modify this property only when you have disabled the resource.

Dev_stop_pct

Type integer. The default is 20. This property indicates the percentage of the `Start_timeout` value that the Oracle Solaris Cluster HA for SAP data service uses to shut down the development system before starting the SAP central instance. The Oracle Solaris Cluster HA for SAP data service will not use this property if you set the property `Shutdown_dev` to the value FALSE. You can modify this property only when you have disabled the resource.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/sap

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresourcegroup\(1CL\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.sap_as\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.sapdb, sapdb – resource type implementation for Oracle Solaris Cluster HA for MaxDB

Description The SUNW. sapdb resource type represents the MaxDB application in an Oracle Solaris Cluster configuration. The MaxDB application requires the SAP xserver system. Therefore, you must set a dependency between the MaxDB resource group and the SAP xserver resource group. Create this dependency when you register and configure the Oracle Solaris Cluster HA for MaxDB data service.

Standard properties and extension properties that are defined for the SUNW. sapdb resource type are described in the subsections that follow. To set these properties for an instance of the SUNW. sapdb resource type, use the [cl_resourcetype\(1CL\)](#) command.

Standard Properties Standard resource properties are overridden for this resource type as follows:

Retry_Count

Maximum 10

Default 2

Tunable Any time

Retry_Interval

Maximum 3600

Default 850

Tunable Any time

Thorough_Probe_Interval

Maximum 3600

Default 120

Tunable Any time

For a description of these standard resource properties, see [r_properties\(5\)](#).

Extension Properties 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.

Data type String

Default db_online

Range Not applicable

Tunable When disabled

DB_Name

The name of the MaxDB database instance in uppercase. This name is created when MaxDB is installed and configured.

Data type String

Default No default defined

Range Not applicable

Tunable When disabled

DB_User

The UNIX user identity of the operating system (OS) user that 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.

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

Note – The `Failover_mode=RESTART_ONLY` setting matches the behavior of the `Failover_enabled=False` setting. The `Failover_mode=LOG_ONLY` setting goes a step further and prevents resources from restarting. Use the `Failover_mode` property instead of the `Failover_enabled` extension property to better control failover behavior. For more information, see the descriptions of the `LOG_ONLY` and `RESTART_ONLY` values for `Failover_mode` in [r_properties\(5\)](#).

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

Oracle Solaris Cluster HA for MaxDB determines the path to the `dbmcli` command from the value of this property. The `dbmcli` 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.

Data type Integer

Default 4

Range No range defined

Tunable Any time

Monitor_retry_interval

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

Data type Integer

Default 2

Range No range defined

Tunable 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

Data type String

Default /var/spool/sql

Range Not applicable

Tunable When disabled

Probe_timeout

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

Data type Integer

Default 90

Range 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 that administers the MaxDB database instance. This user key is created when MaxDB is installed and configured.

Data type String

Default No default defined

Range Not applicable

Tunable When disabled

Examples EXAMPLE 1 Creating a SUNW.sapdb Resource

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

EXAMPLE 1 Creating a SUNW.sapdb Resource (Continued)

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 not shown in this example.
- 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 that administers the MaxDB database is dbadmin.
- The user key of the database user that administers the MaxDB database is DEFAULT.
- The MaxDB resource depends on an HAStoragePlus resource that is named hsprs. The creation of the hsprs resource is not shown in this example.

This example does not show the creation of the logical host resource that the MaxDB resource uses.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/sapdb

See Also [clresourcegroup\(1CL\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.sap_xserver\(5\)](#)

Name SUNW.sapenq, sapenq – resource type implementation for the SAP enqueue server component of Oracle Solaris Cluster HA for SAP Web Application Server

Description The SUNW.sapenq resource type represents the SAP enqueue server component in an Oracle Solaris Cluster configuration. This resource type is part of a set of resource types for the SAP Web Application Server platform. The other components are the SAP replica server (represented by the SUNW.saprepl resource type), the SAP message server (represented by the SUNW.sapscs resource type), and the SAP web application server component (represented by the SUNW.sapwebas resource type).

The SAP enqueue server resource and the SAP message server resource must be in the same failover group (called the SAP central services resource group), because they fail over together. The SAP replica server resource must be in a different failover resource group from the SAP enqueue server resource, because the SAP replica server resource must not fail over with the SAP enqueue server resource.

The resource group affinities must be set to ensure that the SAP central services resource group fails over to the node where the SAP replica resource group has been running and that the SAP replica resource group fails over to another available node.

The resource dependencies must be set to ensure that the SAP replica server resource depends on the SAP enqueue server resource being online.

With the resource group affinities and resource dependencies set as described above, if the SAP enqueue server experiences any hardware or software failure, the SAP central services resource group will fail over to the node where the SAP replica resource group has been running and the SAP replica resource group will fail over to another available node. If the SAP message server experiences any failure, the SAP message server resource will be restarted locally a configurable number of times before a failover is initiated.

Create all these dependencies when you configure the Oracle Solaris Cluster HA for SAP Web Application Server data service.

Standard properties and extension properties that are defined for the SUNW.sapenq resource type are described in the subsections that follow. To set these properties for an instance of the SUNW.sapenq resource type, use the `clresource(1CL)` command.

Standard Properties Standard resource properties are overridden for this resource type as follows:

Retry_Count

The value of this property must be 0 if the SAP replica server is running. If the standalone SAP enqueue server is running without the SAP replica server, this property can be set to a non-zero value.

Maximum 2

Default 0

Tunable Any time

Retry_Interval**Maximum** 3600**Default** 960**Tunable** Any time**Thorough_Probe_Interval****Maximum** 3600**Default** 120**Tunable** Any time

For a description of these standard resource properties, see [r_properties\(5\)](#).

Extension Properties The extension properties of this resource type are as follows:

Child_mon_level

The child process monitoring level for the process monitor facility (PMF). This property is equivalent to the -C option of pmfadm.

The default value of -1 indicates that child process monitoring will not be performed. Positive values indicate the desired level of child process monitoring.

Data type Integer**Default** -1**Range** No range defined**Tunable** Any time**Enqueue_Instance_Number**

The two-digit instance number for the SAP enqueue server. This is the value of SAPSYSTEM in the SAP profile for the SAP enqueue server.

Data type String**Default** No default defined**Range** Not applicable**Tunable** When disabled**Enqueue_Profile**

The full path to the SAP enqueue server profile.

Data type String**Default** No default defined**Range** Not applicable

Tunable When disabled

Enqueue_Server

The full path to the SAP enqueue server executable.

Data type String

Default No default defined

Range Not applicable

Tunable When disabled

Enqueue_Server_Monitor

The full path to the SAP enqueue server monitor executable.

Data type String

Default *directory*/ensmon, where *directory* is the full path to the directory where the SAP enqueue server executable is stored, as specified by the extension property Enqueue_Server.

Range Not applicable

Tunable When disabled

Log_Directory

The directory for the startup and monitor log files.

Data type String

Default The home directory of the administration user, as specified by the extension property SAP_User.

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 SAP enqueue server fault monitor.

Data type Integer

Default 4

Range No range defined

Tunable Any time

Monitor_retry_interval

The interval in minutes between restarts of the SAP enqueue server fault monitor.

Data type Integer

Default 2
Range No range defined
Tunable Any time

Probe_timeout

The timeout value in seconds that the SAP enqueue server fault monitor uses to probe an SAP enqueue server instance.

Data type Integer
Default 120
Range Minimum = 2; no maximum defined
Tunable Any time

SAP_User

The administration user for the SAP enqueue server.

Data type String, where letters are in lowercase
Default No default defined
Range Not applicable
Tunable When disabled

Stop_signal

The signal that is sent to the application to stop the SAP enqueue server application.

Data type Integer
Default 2 (equivalent to SIGINT)
Range 1–37
Tunable When disabled

Examples EXAMPLE 1 Creating Resources for SUNW.sapenq, SUNW.sapscs, and SUNW.saprepl

For this example to work, you must first install the Oracle Solaris Cluster HA for SAP Web Application Server data service, which includes all the packages to make the SAP Web Application Server components highly available.

The failover SAP central services resource group contains the SAP enqueue server resource, the SAP message server resource, and the logical host resource. The following commands are an example of creating the SAP central services resource group:

```
# clresourcegroup create central-rg
# clreslogicalhostname create -g central-rg -h central-lh \\  
-N sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4 central-lh \\  
central-lh-rs
```

EXAMPLE 1 Creating Resources for SUNW.sapenq, SUNW.sapsacs, and SUNW.saprepl (Continued)

To bring online the SAP central services resource group, the following command is run:

```
# clresourcegroup -emM central-rg
```

The failover SAP replica resource group contains the SAP replica server resource and a logical host resource. The following commands are an example of creating the SAP replica resource group:

```
# clresourcegroup create repl-rg
# clreslogicalhostname create -g repl-rg -h repl-lh \
-N sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4 repl-lh-rs
```

To bring online the SAP replica server resource group, the following command is run:

```
# clresourcegroup -emM repl-rg
```

Setting weak positive resource group affinity between the SAP central services resource group and the SAP replica resource group ensures that, in case of failover, the SAP central services resource group fails over to the node where the SAP replica resource group has been running. The following command is an example of setting this affinity:

```
# clresourcegroup set -p RG_affinities+=repl-rg central-rg
```

The two resource groups must be mastered on different nodes before the strong negative affinity can be set. Therefore, either the SAP central services resource group or the SAP replica resource group must be switched to another node. The following command is an example of switching the SAP central services resource group to another node:

```
# clresourcegroup switch -n node2 central-rg
```

Setting strong negative resource group affinity between the SAP replica resource group and the SAP central services resource group ensures that, in case of failover, after the SAP central services resource group fails over to the node where the SAP replica resource group has been running, the SAP replica resource group will fail over to another available node. The following command is an example of setting this affinity:

```
# clresourcegroup set -p RG_affinities=-central-rg repl-rg
```

To register the resource types, the following commands are run:

```
# clresourcetype register SUNW.sapenq
# clresourcetype register SUNW.sapsacs
# clresourcetype register SUNW.saprepl
```

To create the SAP enqueue server resource in the SAP central services resource group, the following command is run:

EXAMPLE 1 Creating Resources for SUNW.sapenq, SUNW.sapscs, and SUNW.saprepl (Continued)

```
# clresource create -g central-rg -t SUNW.sapenq \\  
-p Enqueue_Profile=/usr/sap/SC3/SYS/profile/SC3_SCS01_central-lh \\  
-p Enqueue_Server=/sapmnt/SC3/exe/enserver \\  
-p SAP_User=sc3adm -p Enqueue_Instance_Number=01 enq-rs
```

To create the SAP message server resource in the SAP central services resource group, the following command is run:

```
# clresource create -g central-rg -t SUNW.sapscs \\  
-p SAP_SID=SC3 -p SAP_Instance_Number=01 \\  
-p SAP_Instance_Name=SCS01 -p Msg_Server_Port=3601 msg-rs
```

To create the SAP replica server resource in the SAP replica resource group, the following command is run:

```
# clresource -g repl-rg -t SUNW.saprepl \\  
-p Replica_Profile=/usr/sap/SC3/SYS/profile/SC3_REP01 \\  
-p Replica_Server=/sapmnt/SC3/exe/enrepserver \\  
-p SAP_User=sc3adm -p Resource_Dependencies=enq-rs repl-rs
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/sapenq

See Also [clresource\(1CL\)](#), [clresourcegroup\(1CL\)](#), [clresourcetype\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.saprepl\(5\)](#), [SUNW.sapscs\(5\)](#), [SUNW.sapwebas\(5\)](#)

Name SUNW.sap_livcache, sap_livcache – resource type implementation for failover SAP liveCache database

Description The SAP liveCache data service is managed by the Oracle Solaris Cluster Resource Group Manager (RGM) and is configured as a LogicalHostname resource, a SAP liveCache database resource.

The SAP liveCache database depends on the SAP xserver which is managed by data service SUNW.sap_xserver. Dependency should be set between the SAP liveCache resource group and the SAP xserver resource group.

You must set the following properties for a SAP liveCache database resource using the `clresource(1CL)` command.

Standard Properties See [r_properties\(5\)](#) for a description of the following resource properties.

Retry_count	Maximum: 10
	Default: 2
	Tunable: Any time
Retry_interval	Maximum: 3600
	Default: 620
	Tunable: Any time
Thorough_probe_interval	Maximum: 3600
	Default: 60
	Tunable: Any time

Extension Properties **Monitor_retry_count** (integer) Default is 4. This property controls the restarts of the fault monitor. It indicates the number of times the fault monitor is restarted by the process monitor facility and corresponds to the `-n` option passed to the `pmfadm(1M)` command. The number of restarts is counted in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not SAP liveCache. SAP liveCache restarts are controlled by the system-defined properties `Thorough_Probe_Interval`, `Retry_Interval`, and `Retry_Count`, as specified in their descriptions. You can modify the value for this property at any time.

Monitor_retry_interval (integer) Default is 2. Indicates the time in minutes over which the failures of the fault monitor are counted and corresponds to

the `-t` option passed to the `pmfadm(1M)` command. If the number of times the fault monitor fails exceeds the value of `Monitor_retry_count` within this period, the fault monitor is not restarted by the process monitor facility. You can modify the value for this property at any time.

- `Probe_timeout` (integer) Default is 90. Minimum value is 30. Indicates the time-out value (in seconds) used by the fault monitor to probe a SAP liveCache database instance. You can modify the value for this property at any time.
- `Failover_enabled` (boolean) Defaults to TRUE. Indicates whether to failover or not when `retry_count` is exceeded during `retry_interval`. You must specify this property at resource creation time.

Note – The `Failover_mode=RESTART_ONLY` setting matches the behavior of the `Failover_enabled=False` setting. The `Failover_mode=LOG_ONLY` setting goes a step further and prevents resources from restarting. Use the `Failover_mode` property instead of the `Failover_enabled` extension property to better control failover behavior. For more information, see the descriptions of the `LOG_ONLY` and `RESTART_ONLY` values for `Failover_mode` in [r_properties\(5\)](#).

- `Livecache_Name` (string array) This property is the name of the liveCache database instance. Note the name is in uppercase (LC-NAME). You must specify this property at resource creation time.
- `Confdir_list` (string array) This property only has one value which is the directory for livecache software and instance directories. You must specify this property at resource creation time.

Default is `/sapdb`.

Examples EXAMPLE 1 Configuration Example

For this example to work, you must first install the data service.

The following example creates a failover SAP liveCache database resource named `lc-rs` in an existing resource group called `lc-rg`. `lc-rg` must contain a `LogicalHostName` resource.

```
# clresourcetype register SUNW.sap_livecache
# clresource create -g lc-rg -t SUNW.sap_livecache \\  
-p LiveCache_Name=LC4 lc-rs
```


EXAMPLE 1 Configuration Example *(Continued)*

In this example, LC4 is the SAP liveCache database instance name. The rest of the extension properties use the default values.

After the SAP liveCache database resource group and the SAP xserver resource group are created, set the dependency between them.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/sap-livcache

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresourcegroup\(1CL\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#)

Name SUNW.saprepl, saprepl – resource type implementation for the SAP replica server component of Oracle Solaris Cluster HA for SAP Web Application Server

Description The SUNW.saprepl resource type represents the SAP replica server component in an Oracle Solaris Cluster configuration. This resource type is part of a set of resource types for the SAP Web Application Server platform. The other components are the SAP enqueue server (represented by the SUNW.sapenq resource type), the SAP message server (represented by the SUNW.sapscs resource type), and the SAP web application server component (represented by the SUNW.sapwebas resource type).

The SAP enqueue server resource and the SAP message server resource must be in the same failover group (called the SAP central services resource group), because they fail over together. The SAP replica server resource must be in a different failover resource group from the SAP enqueue server resource, because the SAP replica server resource must not fail over with the SAP enqueue server resource.

The resource group affinities must be set to ensure that the SAP central services resource group fails over to the node where the SAP replica resource group has been running and that the SAP replica resource group fails over to another available node.

The resource dependencies must be set to ensure that the SAP replica server resource depends on the SAP enqueue server resource being online.

With the resource group affinities and resource dependencies set as described above, if the SAP enqueue server experiences any hardware or software failure, the SAP central services resource group will fail over to the node where the SAP replica resource group has been running and the SAP replica resource group will fail over to another available node. If the SAP message server experiences any failure, the SAP message server resource will be restarted locally a configurable number of times before a failover is initiated.

Create all these dependencies when you configure the Oracle Solaris Cluster HA for SAP Web Application Server data service.

Standard properties and extension properties that are defined for the SUNW.saprepl resource type are described in the subsections that follow. To set these properties for an instance of the SUNW.saprepl resource type, use the `clresource(1CL)` command.

Standard Properties Standard resource properties are overridden for this resource type as follows:

Retry_Count

Maximum 2

Default 2

Tunable Any time

Retry_Interval

Maximum 3600

Default 620
Tunable Any time

Thorough_Probe_Interval

Maximum 3600
Default 120
Tunable Any time

For a description of these standard resource properties, see [r_properties\(5\)](#).

Extension Properties The extension properties of this resource type are as follows:

Child_mon_level

The child process monitoring level for the process monitor facility (PMF). This property is equivalent to the -C option of pmfadm.

The default value of -1 indicates that child process monitoring will not be performed. Positive values indicate the desired level of child process monitoring.

Data type Integer
Default -1
Range No range defined
Tunable Any time

Log_Directory

The directory for the startup and monitor log files that are created by the SAP replica server application.

Data type String
Default The home directory of the administration user, as specified by the extension property SAP_User.
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 SAP replica server fault monitor.

Data type Integer
Default 4
Range No range defined
Tunable Any time

Monitor_retry_interval

The interval in minutes between restarts of the SAP replica server fault monitor.

Data type Integer

Default 2

Range No range defined

Tunable Any time

Probe_timeout

Currently unused. The timeout value in seconds that the SAP replica server fault monitor uses to probe an SAP replica server instance. The replica server is started by PMF and monitored by PMF. No additional probing is currently performed by the fault monitor.

Data type Integer

Default 30

Range Minimum = 2; no maximum defined

Tunable Any time

Replica_Profile

The full path to the SAP replica server profile.

Data type String

Default No default defined

Range Not applicable

Tunable When disabled

Replica_Server

The full path to the SAP replica server executable.

Data type String

Default No default defined

Range Not applicable

Tunable When disabled

SAP_User

The administration user for the SAP replica server.

Data type String, where letters are in lowercase

Default No default defined

Range Not applicable

Tunable When disabled

Stop_signal

The signal that is sent to the application to stop the SAP replica server application.

Data type Integer

Default 2 (equivalent to SIGINT)

Range 1–37

Tunable Any time

Examples EXAMPLE 1 Creating Resources for SUNW.sapenq, SUNW.sapses, and SUNW.saprepl

For this example to work, you must first install the Oracle Solaris Cluster HA for SAP Web Application Server data service, which includes all the packages to make the SAP Web Application Server components highly available.

The failover SAP central services resource group contains the SAP enqueue server resource, the SAP message server resource, and the logical host resource. The following commands are an example of creating the SAP central services resource group:

```
# clresourcegroup create central-rg
# clreslogicalhostname create -g central-rg -l central-lh \
-N sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4 central-lh-rs
```

To bring online the SAP central services resource group, the following command is run:

```
# clresourcegroup -emM central-rg
```

The failover SAP replica resource group contains the SAP replica server resource and a logical host resource. The following commands are an example of creating the SAP replica resource group:

```
# clresourcegroup create repl-rg
# clreslogicalhostname create -g repl-rg -l repl-lh \
-N sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4 repl-lh-rs
```

To bring online the SAP replica server resource group, the following command is run:

```
# clresourcegroup -emM repl-rg
```

Setting weak positive resource group affinity between the SAP central services resource group and the SAP replica resource group ensures that, in case of failover, the SAP central services resource group fails over to the node where the SAP replica resource group has been running. The following command is an example of setting this affinity:

```
# clresourcegroup set -p RG_affinities=+repl-rg central-rg
```

The two resource groups must be mastered on different nodes before the strong negative affinity can be set. Therefore, either the SAP central services resource group or the SAP replica

EXAMPLE 1 Creating Resources for SUNW.sapenq, SUNW.sapses, and SUNW.saprepl (Continued)

resource group must be switched to another node. The following command is an example of switching the SAP central services resource group to another node:

```
# clresourcegroup switch -n Node2 central-rg
```

Setting strong negative resource group affinity between the SAP replica resource group and the SAP central services resource group ensures that, in case of failover, after the SAP central services resource group fails over to the node where the SAP replica resource group has been running, the SAP replica resource group will fail over to another available node. The following command is an example of setting this affinity:

```
# clresourcegroup set -p RG_affinities=--central-rg repl-rg
```

To register the resource types, the following commands are run:

```
# clresourcetype register SUNW.sapenq
# clresourcetype register SUNW.sapses
# clresourcetype register SUNW.saprepl
```

To create the SAP enqueue server resource in the SAP central services resource group, the following command is run:

```
# clresource create -g central-rg -t SUNW.sapenq \\  
-p Enqueue_Profile=/usr/sap/SC3/SYS/profile/SC3_SCS01_central-lh \\  
-p Enqueue_Server=/sapmnt/SC3/exe/enserver \\  
-p SAP_User=sc3adm \\  
-p Enqueue_Instance_Number=01 sapenq-rs
```

To create the SAP message server resource in the SAP central services resource group, the following command is run:

```
# clresource create -g central-rg -t SUNW.sapses \\  
-p SAP_SID=SC3 -p SAP_Instance_Number=01 \\  
-p SAP_Instance_Name=SCS01 \\  
-p Msg_Server_Port=3601 msg-rs
```

To create the SAP replica server resource in the SAP replica resource group, the following command is run:

```
# clresource create -g repl-rg -t SUNW.saprepl \\  
-p Replica_Profile=/usr/sap/SC3/SYS/profile/SC3_REP01 \\  
-p Replica_Server=/sapmnt/SC3/exe/enrepserver \\  
-p SAP_User=sc3adm \\  
-p Resource_Dependencies=enq-rs repl-rs
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	ha-cluster/data-service/saprepl

See Also [clresource\(1CL\)](#), [clresourcegroup\(1CL\)](#), [clresourcetype\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.sapenq\(5\)](#), [SUNW.sapscs\(5\)](#), [SUNW.sapwebas\(5\)](#)

Name SUNW.sapscs, sapscs – resource type implementation for the SAP message server component of Oracle Solaris Cluster HA for SAP Web Application Server

Description The SUNW.sapscs resource type represents the SAP message server component in an Oracle Solaris Cluster configuration. This resource type is part of a set of resource types for the SAP Web Application Server platform. The other components are the SAP enqueue server (represented by the SUNW.sapenq resource type), the SAP replica server (represented by the SUNW.saprepl resource type), and the SAP web application server component (represented by the SUNW.sapwebas resource type).

The SAP enqueue server resource and the SAP message server resource must be in the same failover group (called the SAP central services resource group), because they fail over together. The SAP replica server resource must be in a different failover resource group from the SAP enqueue server resource, because the SAP replica server resource must not fail over with the SAP enqueue server resource.

The resource group affinities must be set to ensure that the SAP central services resource group fails over to the node where the SAP replica resource group has been running and that the SAP replica resource group fails over to another available node.

The resource dependencies must be set to ensure that the SAP replica server resource depends on the SAP enqueue server resource being online.

With the resource group affinities and resource dependencies set as described above, if the SAP enqueue server experiences any hardware or software failure, the SAP central services resource group will fail over to the node where the SAP replica resource group has been running and the SAP replica resource group will fail over to another available node. If the SAP message server experiences any failure, the SAP message server resource will be restarted locally a configurable number of times before a failover is initiated.

Create all these dependencies when you configure the Oracle Solaris Cluster HA for SAP Web Application Server data service.

Standard properties and extension properties that are defined for the SUNW.sapscs resource type are described in the subsections that follow. To set these properties for an instance of the SUNW.sapscs resource type, use the `clresourcetype(1CL)` command.

Standard Properties Standard resource properties are overridden for this resource type as follows:

Retry_Count

Maximum 5

Default 2

Tunable Any time

Retry_Interval

Maximum 3600

Default 970

Tunable Any time

Thorough_Probe_Interval

Maximum 3600

Default 120

Tunable Any time

For a description of these standard resource properties, see [r_properties\(5\)](#).

Extension Properties The extension properties of this resource type are as follows:

Failover_Enabled

Specifies whether to fail over when `Retry_Count` is exceeded during `Retry_Interval`.

Data type Boolean

Default TRUE

Range TRUE or FALSE

Tunable When disabled

Note – The `Failover_mode=RESTART_ONLY` setting matches the behavior of the `Failover_enabled=False` setting. The `Failover_mode=LOG_ONLY` setting goes a step further and prevents resources from restarting. Use the `Failover_mode` property instead of the `Failover_enabled` extension property to better control failover behavior. For more information, see the descriptions of the `LOG_ONLY` and `RESTART_ONLY` values for `Failover_mode` in [r_properties\(5\)](#).

Monitor_Retry_Count

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

Data type Integer

Default 4

Range No range defined

Tunable Any time

Monitor_Retry_Interval

The interval in minutes between restarts of the SAP message server fault monitor.

Data type Integer

Default 2

Range No range defined

Tunable Any time

Msg_Server_Monitor

The SAP message server probe executable.

Data type String

Default /usr/sap/<SAP_SID>/SYS/exe/run/msprot

Range Not applicable

Tunable When disabled

Msg_Server_Port

The listen port of the SAP message server.

If no value is specified for this property, the initial default value is 0. In this case a derived default value is calculated to be $3600 + \text{SAP_Instance_Number}$. If the listen port of the SAP message server to be probed is not equivalent to $3600 + \text{SAP_Instance_Number}$, for example, in the case of two SAP message servers, specify a value for this property.

Data type Integer

Default 0

Range 0 – 65535

Tunable When disabled

Probe_Timeout

The timeout value in seconds that the SAP message server fault monitor uses to probe an SAP message server instance.

Data type Integer

Default 120

Range Minimum = 2; no maximum defined

Tunable Any time

SAP_Instance_Name

The name of the SAP message server instance. This is `INSTANCE_NAME` in the SAP profile.

Data type String

Default None defined

Range Not applicable

Tunable When disabled

SAP_Instance_Number

The two-digit SAP system number for the SAP message server instance. This is `SAPSYSTEM` in the SAP profile.

Data type String
Default None defined
Range Not applicable
Tunable When disabled

SAP_SID

The SAP system ID. This is SAPSYSTEMNAME in the SAP profile.

Data type String
Default None defined
Range Not applicable
Tunable When disabled

SAP_User

The administration user for the SAP message server.

Data type String, where letters are in lowercase
Default <\$SAP_SID>adm
Range Not applicable
Tunable When disabled

Scs_Shutdown_Script

The full path to the shut-down script for the instance.

Data type String
Default /usr/sap/<SAP_SID>/SYS/exe/run/stopsap
Range Not applicable
Tunable When disabled

Scs_Startup_Script

The full path to the startup script for the instance.

Data type String
Default /usr/sap/<SAP_SID>/SYS/exe/run/startsap
Range Not applicable
Tunable When disabled

Examples EXAMPLE 1 Creating Resources for SUNW.sapenq, SUNW.sapscs, and SUNW.saprepl

For this example to work, you must first install the Oracle Solaris Cluster HA for SAP Web Application Server data service, which includes all the packages to make the SAP Web Application Server components highly available.

The failover SAP central services resource group contains the SAP enqueue server resource, the SAP message server resource, and the logical host resource. The following commands are an example of creating the SAP central services resource group:

```
# clresourcegroup create central-rg
# clreslogicalhostname create -g central-rg -l central-lh \
-N sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4 central-lh-rs
```

To bring online the SAP central services resource group, the following command is run:

```
# clresourcegroup -emM central-rg
```

The failover SAP replica resource group contains the SAP replica server resource and a logical host resource. The following commands are an example of creating the SAP replica resource group:

```
# clresourcegroup create -g repl-rg
# clreslogicalhostname create -g repl-rg -l repl-lh \
-N sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4 repl-lh-rs
```

To bring online the SAP replica server resource group, the following command is run:

```
# clresourcegroup -emM repl-rg
```

Setting weak positive resource group affinity between the SAP central services resource group and the SAP replica resource group ensures that, in case of failover, the SAP central services resource group fails over to the node where the SAP replica resource group has been running. The following command is an example of setting this affinity:

```
# clresourcegroup set -p RG_affinities+=repl-rg central-rg
```

The two resource groups must be mastered on different nodes before the strong negative affinity can be set. Therefore, either the SAP central services resource group or the SAP replica resource group must be switched to another node. The following command is an example of switching the SAP central services resource group to another node:

```
# clresourcegroup switch -n Node2 central-rg
```

Setting strong negative resource group affinity between the SAP replica resource group and the SAP central services resource group ensures that, in case of failover, after the SAP central services resource group fails over to the node where the SAP replica resource group has been running, the SAP replica resource group will fail over to another available node. The following command is an example of setting this affinity:

EXAMPLE 1 Creating Resources for SUNW.sapenq, SUNW.sapscs, and SUNW.saprepl (Continued)

```
# clresourcegroup set -p RG_affinities=--central-rg repl-rg
```

To register the resource types, the following commands are run:

```
# clresourcetype register SUNW.sapenq
# clresourcetype register SUNW.sapscs
# clresourcetype register SUNW.saprepl
```

To create the SAP enqueue server resource in the SAP central services resource group, the following command is run:

```
# clresource create -g central-rg -t SUNW.sapenq \\  
-p Enqueue_Profile=/usr/sap/SC3/SYS/profile/SC3_SCS01_central-lh \\  
-p Enqueue_Server=/sapmnt/SC3/exe/enserver \\  
-p SAP_User=sc3adm -p Enqueue_Instance_Number=01 enq-rs
```

To create the SAP message server resource in the SAP central services resource group, the following command is run:

```
# clresource create -g central-rg -t SUNW.sapscs \\  
-p SAP_SID=SC3 -p SAP_Instance_Number=01 \\  
-p SAP_Instance_Name=SCS01 \\  
-p Msg_Server_Port=3601 msg-rs
```

To create the SAP replica server resource in the SAP replica resource group, the following command is run:

```
# clresource create -g repl-rg -t SUNW.saprepl \\  
-p Replica_Profile=/usr/sap/SC3/SYS/profile/SC3_REP01 \\  
-p Replica_Server=/sapmnt/SC3/exe/enrepserver \\  
-p SAP_User=sc3adm \\  
-p Resource_Dependencies=enq-rs repl-rs
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/sapscs

See Also [clresource\(1CL\)](#), [clresourcegroup\(1CL\)](#), [clresourcetype\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.sapenq\(5\)](#), [SUNW.saprepl\(5\)](#), [SUNW.sapwebas\(5\)](#)

Name SUNW.sapwebas, sapwebas – resource type implementation for the SAP web application server component of Oracle Solaris Cluster HA for SAP Web Application Server

Description The SUNW.sapwebas resource type represents the following components in an Oracle Solaris Cluster Configuration: SAP web application server; J2EE Engine; SAP web Application Server with the J2EE Engine. This resource type is part of a set of resource types for the SAP Web Application Server platform. The other components are the SAP enqueue server (represented by the SUNW.sapenq resource type), the SAP replica server (represented by the SUNW.saprepl resource type), and the SAP message server (represented by the SUNW.sapscls resource type).

The components represented by the SUNW.sapwebas resource may be configured as a failover resource or a scalable resource.

The SAP web application server component resource depends on the database resource being online. The J2EE Engine component depends on the database resource and the SAP message server. You create these dependencies when you configure the Oracle Solaris Cluster HA for SAP Web Application Server data service and the J2EE Engine data service.

Standard properties and extension properties that are defined for the SUNW.sapwebas resource type are described in the subsections that follow. To set these properties for an instance of the SUNW.sapwebas resource type, use the `clresource(1CL)` command.

Standard Properties Standard resource properties are overridden for this resource type as follows:

Retry_Count

Maximum 5

Default 2

Tunable Any time

Retry_Interval

Maximum 4600

Default 4320

Tunable Any time

Thorough_Probe_Interval

Maximum 3600

Default 120

Tunable Any time

For a description of these standard resource properties, see [r_properties\(5\)](#).

Extension Properties The extension properties of this resource type are as follows:

Monitor_Retry_Count

The maximum number of restarts by the process monitor facility (PMF) that are allowed for the SAP web application server component fault monitor.

Data type Integer

Default 4

Range No range defined

Tunable Any time

Monitor_Retry_Interval

The interval in minutes between restarts of the SAP web application server component fault monitor.

Data type Integer

Default 2

Range No range defined

Tunable Any time

Probe_Timeout

The timeout value in seconds that the SAP web application server component fault monitor uses to probe an SAP web application server component instance.

Data type Integer

Default 120

Range Minimum = 2; no maximum defined

Tunable Any time

SAP_Instance_Name

The name of the SAP web application server component instance. This is `INSTANCE_NAME` in the SAP profile.

Data type String

Default None defined

Range Not applicable

Tunable When disabled

SAP_Instance_Number

The two-digit SAP system number for the SAP web application server component instance. This is `SAPSYSTEM` in the SAP profile.

Data type String

Default None defined
Range Not applicable
Tunable When disabled

SAP_SID

The SAP system ID. This is SAPSYSTEMNAME in the SAP profile.

Data type String
Default None defined
Range Not applicable
Tunable When disabled

SAP_User

The administration user for the SAP web application server component.

Data type String, where letters are in lowercase
Default <SAP_SID>adm
Range Not applicable
Tunable When disabled

SAP_Instance_Type

The Instance type on the specified Host. The possible values of this extension property are as follows:

- ABAP – Specifies that SAP Web AS ABAP central instance is installed on the host.
- J2EE – Specifies that SAP Web AS Java engine is deployed on the host.
- ABAP_J2EE – Specifies that SAP Web AS ABAP and SAP Web AS Java engine are deployed on the host.

Data type Enum
Default ABAP
Range Not applicable
Tunable When disabled

SAP_Logdir

The directory for the startup and monitor log files.

Data type String
Default The home directory of the administration user, as specified by the extension property SAP_User.
Range Not applicable

Tunable Any time

Webas_Shutdown_Script

The full path to the shut down script for the instance.

Data type String

Default /usr/sap/<SAP_SID>/SYS/exe/run/stopsap

Range Not applicable

Tunable When disabled

Webas_Startup_Script

The full path to the start-up script for the instance.

Data type String

Default /usr/sap/<SAPSID>/SYS/exe/run/startsap

Range Not applicable

Tunable When disabled

Webas_Use_Pmf

Determines if the startup script process tree is run under Process Monitor Facility (PMF). The possible values of this extension property are as follows:

- True - Specifies that the startup script process tree is run under PMF.
- False - Specifies that the startup script process tree is *not* run under PMF.

Data type Boolean

Default TRUE

Range Not applicable

Tunable When disabled

Examples EXAMPLE 1 Creating a Failover Resource for SUNW.sapwebas

For this example to work, you must first install the Oracle Solaris Cluster HA for SAP Web Application Server data service, which includes all the packages to make the SAP Web Application Server components highly available.

The failover resource group for the SAP web application server component contains the SAP web application server component resource and the logical host resource. The following commands are an example of creating the failover resource group for the SAP web application server component:

```
# clresourcegroup create fo-webas-rg
# clreslogicalhostname create -g fo-webas-rg -l webas-lh \
-n sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4
```

EXAMPLE 1 Creating a Failover Resource for SUNW.sapwebas (Continued)

To bring online the failover resource group for the SAP web application server component, the following command is run:

```
# clresourcegroup -emM fo-webas-rg
```

To register the resource type, the following command is run:

```
# clresourcetype register SUNW.sapwebas
```

To create a SAP web application server component resource in the failover resource group, the following command is run:

```
# clresource create -g fo-webas-rg -t SUNW.sapwebas \  
-p SAP_SID=SC3 -p SAP_Instance_Number=08 \  
-p SAP_Instance_Name=D08 \  
-p Resource_Dependencies=db-webas-rs,msg-rs webas-rs
```

EXAMPLE 2 Creating a Scalable Resource for SUNW.sapwebas

For this example to work, you must first install the Oracle Solaris Cluster HA for SAP Web Application Server data service, which includes all the packages to make the SAP Web Application Server components highly available.

The scalable resource group for the SAP web application server component contains the SAP web application server component resource and the logical host resource. The following commands are an example of creating the scalable resource group for the SAP web application server component:

```
# clresourcegroup create \  
-p Maximum primaries=4 \  
-p Desired primaries=4 \  
sc-webas-rg  
# clreslogicalhostname create -g sc-webas-rg -l webas-lh \  
-N sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4
```

To bring online the scalable resource group for the SAP web application server component, the following command is run:

```
# clresourcegroup -emM sc-webas-rg
```

To register the resource type, the following command is run:

```
# clresourcetype register SUNW.sapwebas
```

To create a SAP web application server component resource in the scalable resource group, the following command is run:

EXAMPLE 2 Creating a Scalable Resource for SUNW.sapwebas (Continued)

```
# clresource create -g sc-webas-rg -t SUNW.sapwebas \\  
-p SAP_SID=SC3 -p SAP_Instance_Number=08 \\  
-p SAP_Instance_Name=D08 \\  
-p Resource_Dependencies=db-webas-rs,msg-rs webas-rs
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/sapwebas

See Also [clresource\(1CL\)](#), [clresourcegroup\(1CL\)](#), [clresourcetype\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.sapenq\(5\)](#), [SUNW.sapscs\(5\)](#), [SUNW.saprepl\(5\)](#)

Name SUNW.sap_xserver, sap_xserver – resource type implementation for scalable SAP xserver

Description The SAP xserver data service for Oracle Solaris Cluster is managed by the Oracle Solaris Cluster Resource Group Manager (RGM) and is configured as a scalable SAP xserver resource.

You must set the following properties on an SAP xserver resource using the `clresource(1CL)` command.

Standard Properties See [r_properties\(5\)](#) for a description of the following resource properties.

Retry_count

Maximum: 10

Default: 2

Tunable: Any time

Retry_interval

Maximum: 3600

Default: 620

Tunable: Any time

Thorough_probe_interval

Maximum: 3600

Default: 60

Tunable: Any time

Extension Properties `Confdir_list`

Type string. The full path to the directory that contains the software and database instance of the applications that depend on the xserver. These applications can be any combination of the following applications:

- SAP DB
- SAP liveCache

Default is `/sapdb`. You can modify the value for this property only when you have disabled the resource.

`Independent_Program_Path`

Type string. 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

A `SUNW.sap_xserver` resource 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.

You can modify the value for this property only when the resource is disabled.

Monitor_retry_count

Type integer; default is 4. This property controls the restarts of the fault monitor. It indicates the number of times the fault monitor is restarted by the process monitor facility and corresponds to the `-n` option passed to the `pmfadm(1M)` command. The number of restarts is counted in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not the SAP xserver. The SAP xserver restarts are controlled by the system-defined properties `Thorough_Probe_Interval` and `Retry_Interval`, as specified in their descriptions. See `clresource(1CL)`. You can modify the value for this property at any time.

Monitor_retry_interval

Type integer, default is 2. Indicates period of time in minutes during which the PMF counts restarts of the fault monitor and corresponds to the `-t` option passed to the `pmfadm` command. If the number of times the fault monitor fails exceeds the value of `Monitor_retry_count` within this period, the fault monitor is not restarted by the process monitor facility. You can modify the value for this property at any time.

Probe_timeout

Type integer; default is 90. Indicates the time-out value (in seconds) used by the fault monitor to probe a SAP xserver instance. The minimum value is 30. You can modify the value for this property at any time.

Soft_Stop_Pct

Type integer. This property is 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. You can modify the value for this property at any time.

Default is 50.

Xserver_User

Type string array. This property is the SAP xserver system administrator user name. You can modify the value for this property only when you have disabled the resource.

Default is root.

Examples EXAMPLE 1 Configuration Example

For this example to work, you must first install the data service.

The following example creates a scalable SAP xserver resource named `xsvr-rs` in a resource group called `xsvr-rg`. The `xsvr-rg` resource group does not contain a `SharedAddress` resource.

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

EXAMPLE 1 Configuration Example *(Continued)*

```
# clresourcetype register SUNW.sap_xserver
# clresource create -g xsvr-rg -t SUNW.sap_xserver xsrv-rs
```

In this example, SAP xserver will run on 4 cluster nodes. The SAP xserver extension properties use the default values.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/sap-livecache

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [clresourcegroup\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#)

Name SUNW.sblgtwy, sblgtwy – resource type implementation for failover Siebel gateway

Description The Siebel data service for Oracle Solaris Cluster is managed by the Oracle Solaris Cluster Resource Group Manager (RGM) and is configured as a Siebel gateway resource and one or more Siebel server resources.

You must set the following properties for a Siebel gateway resource using the `clresource(1CL)` command.

Standard Properties See [r_properties\(5\)](#) for a complete description of the following resource properties.

Retry_count

Maximum: 10

Default: 2

Tunable: Any time

Retry_interval

Maximum: 3600

Default: 730

Tunable: Any time

Thorough_probe_interval

Maximum: 3600

Default: 60

Tunable: Any time

Extension Properties `Confdir_list`

Type string array. This property is the path name to the Siebel gateway root directory. You can specify the value at resource creation time only.

`Monitor_retry_count`

Type integer. Default is 4. This property controls the restarts of the fault monitor. It indicates the number of times the fault monitor is restarted by the process monitor facility and corresponds to the `-n` option passed to the `pmfadm(1M)` command. The number of restarts is counted in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault monitor itself, not the Siebel gateway. Siebel gateway restarts are controlled by the system-defined properties `Thorough_Probe_Interval` and `Retry_Interval`, as specified in their descriptions. You can modify the value for this property at any time.

`Monitor_retry_interval`

Type integer. Default is 2. Indicates the time (in minutes) over which the failures of the fault monitor are counted, and corresponds to the `-t` option passed to the `pmfadm` command. If the number of times the fault monitor fails exceeds the value of

Monitor_retry_count within this period, the fault monitor is not restarted by the process monitor facility. You can modify the value for this property at any time.

Probe_timeout

Type integer Default is 120. Indicates the time-out value (in seconds) used by the fault monitor to probe a Siebel gateway instance. You can modify the value for this property at any time.

Examples EXAMPLE 1 Configuration Example

For this example to work, you must first install the data service.

The following example creates a failover Siebel gateway resource named sblgtwy-rs in an existing resource group called siebel-rg. siebel-rg is assumed to contain a LogicalHostName resource.

```
# clresourcetype register SUNW.sblgtwy
# clresource create -g siebel-rg -t SUNW.sblgtwy \
-p Confdir_list=/global/siebel/gtwy sblgtwy-rs
```

In this example, /global/siebel/gtwy is the Siebel gateway root directory.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/siebel

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [clresourcegroup\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.sblsrvr, sblsrvr – resource type implementation for failover Siebel server

Description The Siebel data service for Oracle Solaris Cluster is managed by the Oracle Solaris Cluster Resource Group Manager (RGM) and is configured as a Siebel gateway resource and one or more Siebel server resources.

You must set the following properties on an Siebel server resource using `clresource(1CL)` command.

Standard Properties See [r_properties\(5\)](#) for a description of the following resource properties.

Retry_count

Maximum: 10

Default: 2

Tunable: Any time

Retry_interval

Maximum: 3600

Default: 1700

Tunable: Any time

Thorough_probe_interval

Maximum: 3600

Default: 120

Tunable: Any time

Extension Properties Confdir_list

Type string array. This property is the path name to the Siebel server root directory. You can specify the value at resource creation time only.

Siebel_enterprise

Type string. This property is set to the name of the Siebel enterprise. You can specify the value at resource creation time only.

Siebel_server

Type string. This property is set to the name of the Siebel server. You can specify the value at resource creation time only.

Monitor_retry_count

Type integer; default is 4. This property controls the restarts of the fault monitor. It indicates the number of times the fault monitor is restarted by the process monitor facility and corresponds to the `-n` option passed to the `pmfadm(1M)` command. The number of restarts is counted in a specified time window (see the property `Monitor_retry_interval`). Note that this property refers to the restarts of the fault

monitor itself, not the Siebel server. Siebel server restarts are controlled by the system-defined properties `Thorough_Probe_Interval`, `Retry_Interval`, and `Retry_Count`, as specified in their descriptions. You can modify the value for this property at any time.

`Monitor_retry_interval`

Type integer. Default is 2. Indicates the time in minutes, over which the failures of the fault monitor are counted, and corresponds to the `-t` option passed to the `pmfadm` command. If the number of times the fault monitor fails exceeds the value of `Monitor_retry_count`, the fault monitor is not restarted by the process monitor facility. You can modify the value for this property at any time.

`Probe_timeout`

Type integer. Default is 300. This property is the time-out value (in seconds) used by the fault monitor to probe a Siebel server instance. You can modify the value for this property at any time.

Examples **EXAMPLE 1** Configuration Example

For this example to work, you must first install the data service.

The following example creates a failover Siebel server resource named `sblsrvr-rs` in an existing resource group called `siebel-rg`. `siebel-rg` is assumed to contain a `LogicalHostName` resource.

```
# clresourcetype register SUNW.sblsrvr
# clresource create -g siebel-rg -t SUNW.sblsrvr \
-p Confdir_list=/global/siebel/srvr \
-p siebel_enterprise=sieb_ent \
-p siebel_server=button-1 sblsrvr-rs
```

In this example, `/global/siebel/srvr` is the Siebel server root directory. The Siebel enterprise is `sieb_ent` and Siebel server name is `button-1`.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/siebel

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresourcetype\(1CL\)](#), [clresource\(1CL\)](#), [clresourcegroup\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.scalable_asm_diskgroup_proxy, scalable_asm_diskgroup_proxy – resource type implementation for the clustered Oracle Automated Storage Management (Oracle ASM) disk group managed by Oracle Solaris Cluster

Description The SUNW.scalable_asm_diskgroup_proxy resource type represents the clustered Oracle ASM disk group in an Oracle Solaris Cluster configuration. This resource type is introduced for use starting with Oracle 11g release 2 configurations.

Note – The SUNW.scalable_asm_diskgroup_proxy resource type can only be used when using Oracle Grid Infrastructure for Clusters.

The SUNW.scalable_asm_diskgroup_proxy resource type is a multiple-master resource type. A single resource of this type can run on multiple nodes concurrently, but does not use network load balancing.

Each SUNW.scalable_asm_diskgroup_proxy resource represents a clustered Oracle ASM disk group. Each clustered Oracle ASM disk group is uniquely identified by the value of the `asm_diskgroups` extension property on the node where the instance is running. The Oracle ASM disk group resource should only be mounted if the Oracle ASM instance is available on the same cluster node. Additionally, Oracle ASM disk groups should only be mounted if any required storage resources are enabled on the cluster node. To ensure that these requirements are met, configure the Oracle ASM disk group resource as follows:

- Create a strong positive affinity between the Oracle ASM disk group resource group and the following resource groups:
 - The clustered Oracle ASM resource group
 - Any resource group that contains storage resources for Oracle files
- Create an offline-restart dependency between the clustered Oracle ASM disk group resource and the following resources:
 - The clustered Oracle ASM instance resource
 - Any storage resources for Oracle files that you are using

Create these dependencies and affinities when you configure clustered Oracle ASM disk group resources for the Oracle Solaris Cluster HA for Oracle data service or the Oracle Solaris Cluster Support for Oracle RAC data service.

To register this resource type and create instances of this resource type, use one of the following:

- The `clsetuputility`, specifying the option for configuring Oracle Solaris Cluster Support for Oracle Real Application Clusters
- The following sequence of Oracle Solaris Cluster maintenance commands:
 - To register this resource type, use the `clresourcetypecommand`.
 - To create instances of this resource type, use the `clresourcecommand`.

Standard Properties For a description of all standard resource properties, see the [r_properties\(5\)](#) man page.

Standard resource properties are overridden for this resource type as follows:

Prenet_start_timeout

Minimum	60
Default	300

Prenet_stop_timeout

Minimum	60
Default	300

Extension Properties The extension properties of the SUNW.scalable_asm_diskgroup_proxy resource type are as follows.

asm_diskgroups

This property specifies the single-instance Oracle ASM disk group. If required, more than one single-instance Oracle ASM disk group can be specified as a comma separated list.

Data Type	String array
Range	Not applicable
Default	No default defined
Tunable	When disabled

debug_level

Note – All SQL*Plus and srvmgr messages that the Oracle ASM disk group resource issues are written to the log file `/var/opt/SUNWscor/oracle_asm/message_log. ${RESOURCE}`.

This property indicates the level to which debug messages for the Oracle ASM disk-group resources are logged. When the debug level is increased, more debug messages are written to the system log `/var/adm/messages` as follows:

- 0 No debug messages
- 1 Function Begin and End messages
- 2 All debug messages and function Begin/End messages

You can specify a different value of the `debug_level` extension property for each node that can master the resource.

Data Type	Integer
Range	0–2
Default	0

Tunable	Any time
proxy_probe_timeout	
	This property specifies the timeout value, in seconds, that the proxy monitor uses when checking the status of the clustered Oracle ASM disk group resource for which this resource is acting as a proxy.
Data Type	Integer
Range	5–120
Default	60
Tunable	Any time
proxy_probe_interval	
	This property specifies the interval, in seconds, between probes of the clustered Oracle ASM disk group resource for which this resource is acting as a proxy.
Data Type	Integer
Range	5–120
Default	30
Tunable	Any time

Examples **EXAMPLE 1** Creating a scalable_asm_diskgroup_proxy Resource

This example shows the commands for performing the following operations to create a scalable multi-master `scalable_asm_diskgroup_proxy` resource on a two-node cluster:

1. Creating the `asm-dg-rg` resource group
2. Registering the `SUNW.scalable_asm_idiskgroup_proxy` resource type
3. Setting the resource group affinity
4. Adding the `asm-dg-rs` resource to the `asm-dg-rg` resource group
5. Setting the `asm_diskgroups` extension property for one Oracle ASM disk group

The example makes the following assumptions:

- The bash shell is used.
- A resource group that is named `asm-inst-rg` exists and contains a resource of type `SUNW.scalable_asm_instance_proxy` that is named `asm-inst-rs`.
- A resource group that is named `scal-mp-rg` exists and contains a resource of type `SUNW.ScalMountPoint` that is named `scal-mp-rs`, for Oracle files.

```
phys-schost-1# clresourcetype register SUNW.scalable_asm_diskgroup_proxy
phys-schost-1# clresourcegroup create -S asm-dg-rg
phys-schost-1# clresourcegroup set -p rg_affinities=++asm-inst-rg asm-dg-rg
phys-schost-1# clresource create -g asm-dg-rg \
-t SUNW.scalable_asm_diskgroup_proxy \
```

EXAMPLE 1 Creating a scalable_asm_diskgroup_proxy Resource (Continued)

```
-p asm_diskgroups=data1 \<\  
-p resource_dependencies_offline_restart=casm-inst-rs,qfs-mp-rs \<\  
-d asm-dg-rs
```

```
phys-schost-1# clresourcegroup online -M asm-dg-rg
```

```
phys-schost-1# clresource enable asm-dg-rs
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/oracle-database

See Also [Oracle Solaris Cluster Data Services Planning and Administration Guide](#)

Name SUNW.scalable_asm_instance, scalable_asm_instance – resource type implementation for the Oracle single instance Automated Storage Management (ASM) managed by Oracle Solaris Cluster

Description The SUNW.scalable_asm_instance resource type represents the single instance ASM in an Oracle Solaris Cluster configuration.

Note – Use the SUNW.scalable_asm_instance resource type only if you are not using Oracle Cluster Ready Services (CRS). If you are using Oracle Cluster Ready Services, use the SUNW.scalable_asm_instance_proxy resource type.

The SUNW.scalable_asm_instance resource type is a multiple-master resource type. A single resource of this type can run on multiple nodes concurrently, but does not use network load balancing.

Each SUNW.scalable_asm_instance resource represents all Oracle single instance ASM instances. Each instance of the single instance ASM is uniquely identified by the value of the oracle_sid extension property on the node where the instance is running.

The oracle_sid extension property is a per-node property. A single resource of this type can take a different value of this property for each node.

To register this resource type and create instances of this resource type, use one of the following:

- The `clsetup(1CL)` utility, specifying the option for configuring Oracle Solaris Cluster Support for Oracle Real Application Clusters
- The following sequence of Oracle Solaris Cluster maintenance commands:
 1. To register this resource type, use the `clresourcetype(1CL)` command.
 2. To create instances of this resource type, use the `clresource(1CL)` command.

Standard Properties For a description of all standard resource properties, see the `r_properties(5)` man page.

Standard resource properties are overridden for this resource type as follows:

Failover_mode

Default SOFT

Tunable Any time

Load_balancing_policy

Default LB_WEIGHTED

Tunable At creation

Load_balancing_weights

Default ""

Tunable	Any time
Monitor_check_timeout	
Minimum	60
Default	300
Monitor_start_timeout	
Minimum	60
Default	300
Monitor_stop_timeout	
Minimum	60
Default	300
Network_resources_used	
Default	""
Default	Any time
Port_list	
Default	""
Tunable	Anytime
Retry_count	
Maximum	10
Default	2
Tunable	Anytime
Retry_Interval	
Maximum	3600
Default	370
Tunable	Anytime
Scalable	
Default	FALSE
Tunable	At creation
Start_timeout	
Minimum	60

Default	300
Stop_timeout	
Minimum	60
Default	300
Through_probe_interval	
Minimum	3600
Default	60
Tunable	ANYTIME
Update_timeout	
Minimum	60
Default	300
Validate_timeout	
Minimum	60
Default	300

Extension Properties The extension properties of the SUNW.scalable_asm_instance resource type are as follows.

Child_mon_level(integer)

Provides control over the processes that are monitored through the Process Monitor Facility (PMF). This property denotes the level to which the forked child processes are monitored. Omitting this property or setting this property to the default value is the same as omitting the `-Coption` for `pmfadm(1M)`. All child processes and their descendents are monitored.

Category Optional
Default -1
Tunable When disabled

debug_level

Note – All SQL*Plus and srmgr messages that the single instance ASM resource issues are written to the log file `/var/opt/SUNWscor/oracle_asm/message_log. ${RESOURCE}`.

This property indicates the level to which debug messages for the Oracle single instance ASM resources are logged. When the debug level is increased, more debug messages are written to the system log `/var/adm/messages` as follows:

0	No debug messages
---	-------------------

1	Function Begin and End messages
2	All debug messages and function Begin/End messages

You can specify a different value of the `debug_level` extension property for each node that can master the resource

Data Type	Integer
Range	0–2
Default	0
Tunable	Any time

`Failover_Enabled`(boolean)

Allows the resources to fail over. If this property is set to `False`, failover of the resource is disabled. You can use this property to prevent the application resource from initiating a failover of the resource group.

Note – Use the `Failover_mode` property instead of the `Failover_enabled` extension property because `Failover_mode` better controls failover behavior. For more information, see the descriptions of the `LOG_ONLY` and `RESTART_ONLY` values for `Failover_mode` in [r_properties\(5\)](#).

You can specify a different value of the `debug_level` extension property for each node that can master the resource.

Category	Optional
Default	True
Tunable	When disabled

`Log_level`

Specifies the level, or type, of diagnostic messages that are logged by GDS. You can specify `None`, `Info`, or `Err` for this property. When you specify `None`, diagnostic messages are not logged by GDS. When you specify `Info`, both information and error messages are logged. When you specify `Err`, only error messages are logged.

Category	Optional
Default	Info
Tunable	Any time

`Network_aware`(boolean)

This property specifies whether an application uses the network.

Category	Optional
Default	False

Tunable	At creation
Monitor_retry_count	This property specifies the number of PMF restart allowed for the fault monitor.
Default	4
Tunable	Any time
Monitor_retry_interval	This property specifies the number of PMF restart allowed for the fault monitor.
Default	2
Tunable	Any time
probe_command(string)	Specifies the command that periodically checks the health of the single instance ASM.
Category	Required
Default	<code>"/opt/SUNWscor/oracle_asm/bin/asm_control probe -R %RS_NAME -G %RG_NAME -T %RT_NAME"</code>
Tunable	None
Probe_timeout(integer)	This property specifies the timeout value, in seconds, for the probe command.
Category	Optional
Default	30 seconds
Tunable	Any time
Start_command(string)	Specifies the command that starts the single instance ASM.
Category	Required
Default	<code>"/opt/SUNWscor/oracle_asm/bin/asm_control start -R %RS_NAME -G %RG_NAME -T %RT_NAME"</code>
Tunable	None
Stop_command(string)	Specifies the command that stops the single instance ASM.
Category	Required
Default	<code>"/opt/SUNWscor/oracle_asm/bin/asm_control stop -R %RS_NAME -G %RG_NAME -T %RT_NAME"</code>
Tunable	None

Stop_signal(integer)

Specifies the command that stops the single instance ASM.

Category	Optional
Default	15
Tunable	When disabled

Validate_command(string)

Specifies the absolute path to the command that validates the application, although currently not used.

Category	Optional
Default	Null
Tunable	When disabled

oracle_home

This property specifies the full path to the Oracle home directory. The Oracle home directory contains the binary files, log files, and parameter files for the Oracle software.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	When disabled

oracle_sid

This property specifies the Oracle System Identifier (SID). The Oracle SID uniquely identifies the Oracle single instance ASM on the node where the instance is running.

Data Type	String
Range	Not applicable
Default	+ASM
Tunable	When disabled

Examples EXAMPLE 1 Creating a scalable_asm_instance Resource

This example shows the commands for performing the following operations to create a scalable scalable_asm_instance resource on a two-node cluster:

1. Creating the asm_inst_rg resource group
2. Registering the SUNW.scalable_asm_instance resource type
3. Adding the asm_inst_rs resource to the asm_inst_rg resource group

A different value for the oracle_sid extension properties is set for each node that can master the resource.

EXAMPLE 1 Creating a `scalable_asm_instance` Resource *(Continued)*

The example makes the assumption that the bash shell is used.

```
phys-schost-1# clresourcetype register SUNW.scalable_asm_instance
phys-schost-1# clresourcegroup create -S asm-inst-rg
phys-schost-1# clresource create -g asm-inst-rg \
-t SUNW.scalable_asm_instance \
-p Oracle_home=/export/home/oraasm/oracle/product/10.2.0/db_1 \
-d asm-inst-rs
phys-schost-1# clresourcegroup online -M asm-inst-rg
phys-schost-1# clresource enable asm-inst-rs
```

In the above example, the default value of `+ASM` is used for `oracle_sid`. If different values are required for `oracle_sid`, you should use the `per_node` property attribute as follows:

```
phys-schost-1# clresource create -g asm-inst-rg \
-t SUNW.scalable_asm_instance \
-p Oracle_home=/export/home/oraasm/oracle/product/10.2.0/db_1 \
-p "Oracle_sid{pmutt1}"=+ASM_node1 \
-p "Oracle_sid{pmutt2}"=+ASM_node2 \
-d asm-inst-rs
```

See Also [Oracle Solaris Cluster Data Services Planning and Administration Guide](#)

Name SUNW.scalable_asm_instance_proxy, scalable_asm_instance_proxy – resource type implementation for the Oracle Clustered Automated Storage Management (ASM) instance proxy managed by Oracle Solaris Cluster

Description The SUNW.scalable_asm_instance_proxy resource type represents a proxy for the Oracle clustered ASM instance in an Oracle Solaris Cluster configuration.

Note – Use the SUNW.scalable_asm_instance_proxy resource type *only* if you are using Oracle Cluster Ready Services (CRS). If you are *not* using CRS, use the [SUNW.scalable_asm_instance\(5\)](#) resource type.

From Oracle 10g, Oracle Cluster Ready Services (CRS) manage the startup and shutdown of clustered ASM instances. To be managed by the CRS, these instances must be registered with the CRS. The Oracle CRS software also provides automatic fault monitoring and failure recovery for clustered ASM instances. These instances are represented as resources to Oracle CRS.

A resource of type SUNW.scalable_asm_instance_proxy is a *proxy* resource. The proxy resource acts as a substitute for a resource that is managed by Oracle CRS. The proxy resource enables Oracle Solaris Cluster utilities to manage Oracle RAC server instances *through* Oracle CRS. In this way, the SUNW.scalable_asm_instance_proxy resource type enables the clustering frameworks that are provided by Oracle Solaris Cluster and Oracle Cluster Ready Services (CRS) to interoperate.

The SUNW.scalable_asm_instance_proxy resource type enables you to use Oracle Solaris Cluster utilities as an alternative to Oracle utilities to start and stop Oracle RAC database instances.

Each resource of type SUNW.scalable_asm_instance_proxy has a monitor that obtains the following status information from the CRS resource for which the SUNW.scalable_asm_instance_proxy resource is acting as a proxy.

- The online or offline status of Oracle CRS
- The status of an Oracle clustered ASM instance:
 - Online and enabled
 - Online but disabled
 - Offline and enabled
 - Offline and disabled
- The success or failure of an attempt to start or stop a clustered ASM instance
- The ability of Oracle CRS to manage the Oracle clustered ASM instance

The timeout period that the monitor uses for obtaining status information is determined by the proxy_probe_timeout extension property. If the timeout period is too short, timeouts

might cause the monitor to report the status of a valid RAC database instance as invalid. In this situation, consider increasing the value of the `proxy_probe_timeout` extension property.

This monitor only enables the status of Oracle clustered ASM instances to be monitored by Oracle Solaris Cluster utilities. This monitor does *not* provide fault monitoring and automatic fault recovery for Oracle clustered ASM instances. The Oracle CRS software provides this functionality.

Oracle clustered ASM server instances should be started only after the CRS framework and any storage resources are enabled on the cluster node. To ensure that this requirement is met, configure the Oracle clustered ASM instance proxy resource as follows:

- Create a strong positive affinity between the Oracle clustered ASM instance proxy resource group and the following resource groups:
 - The RAC framework resource group
 - Any resource group that contains storage resources for Oracle files
- Create a strong dependency between the Oracle clustered ASM instance proxy resource and the RAC framework resource.
- Create an offline-restart dependency between the Oracle clustered ASM instance proxy resource and the following resources:
 - The CRS framework resource
 - Any storage resources for Oracle files that you are using

Create these dependencies and affinities when you configure clustered ASM instance resources for the Oracle Solaris Cluster Support for Oracle data service or Oracle Solaris Cluster Support for Oracle RAC data service. .

To register this resource type and create instances of this resource type, use one of the following options:

- The `clsetup(1CL)` utility, specifying the option for configuring Oracle Solaris Cluster Support for Oracle Real Application Clusters
- The following sequence of Oracle Solaris Cluster maintenance commands:
 1. To register this resource type, use the `clresourcetype(1CL)` command.
 2. To create instances of this resource type, use the `clresource(1CL)` command.

Note – A Solaris project might be specified for a `SUNW.scalable_asm_instance_proxy` resource or the resource group that contains a `SUNW.scalable_asm_instance_proxy` resource. In this situation, the project affects *only* the processes for the `SUNW.scalable_asm_instance_proxy` resource. The project does *not* affect the processes for any resources that Oracle CRS control, including processes for clustered ASM instances.

Standard Properties For a description of all standard resource properties, see the [r_properties\(5\)](#) man page.

Standard resource properties are overridden for this resource type as follows:

Prenet_start_timeout

Minimum	60
Default	300

Prenet_stop_timeout

Minimum	60
Default	300

Extension Properties The extension properties of the SUNW.scalable_asm_instance_proxy resource type are as follows.

crs_home

This property specifies the full path to the Oracle CRS home directory. The Oracle CRS home directory contains the binary files, log files and parameter files for the Oracle CRS software.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	When disabled

debug_level

Note – All SQL*Plus and srmgr messages that the clustered ASM instance proxy resource issues are written to the log file `/var/opt/SUNWscor/oracle_asm/message_log. ${RESOURCE}`.

This property indicates the level to which debug messages from the monitor for the Oracle clustered ASM instance proxy are logged. When the debug level is increased, more debug messages are written to the system log `/var/adm/messages` as follows:

- 0 No debug messages
- 1 Function Begin and End messages
- 2 All debug messages and function Begin/End messages

You can specify a different value of the `debug_level` extension property for each node that can master the resource.

Data Type	Integer
Range	0–2

Default	0
Tunable	Any time

oracle_home

This property specifies the full path to the Oracle home directory. The Oracle home directory contains the binary files, log files, and parameter files for the Oracle software.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	When disabled

oracle_sid

This property specifies the Oracle System Identifier (SID). The Oracle SID uniquely identifies the Oracle clustered ASM database instance on the node where the instance is running.

You *must* specify a different value of the `oracle_sid` extension property for each node that can master the resource. The value for each node must correctly identify the instance that is running on the node.

Data Type	String
Range	Not applicable
Default	NULL
Tunable	When disabled

proxy_probe_timeout

This property specifies the timeout value, in seconds, that the proxy monitor uses when checking the status of the CRS resource for which this resource is acting as a proxy.

Data Type	Integer
Range	5–120
Default	60
Tunable	Any time

proxy_probe_interval

This property specifies the interval, in seconds, between probes of the CRS resource for which this resource is acting as a proxy.

Data Type	Integer
Range	5–120
Default	60

Tunable

Any time

Examples EXAMPLE 1 Creating a scalable_asm_instance_proxy Resource

This example shows the commands for performing the following operations to create a scalable_asm_instance_proxy resource on a two-node cluster:

1. Creating the asm-inst-rg resource group
2. Registering the SUNW.scalable_asm_instance_proxy resource type
3. Adding the asm-inst-rs resource to the asm-inst-rg resource group

A different value of the oracle_sid extension property is set for each node that can master the resource.

The example makes the following assumptions:

- The bash shell is used.
- A resource group that is named scal-mp-rg exists and contains a resource of type SUNW.ScalMountPoint that is named scal-mp-rs.
- A RAC framework resource group that is named rac-fwk-rg exists and contains resources that are shown in the following table:

Resource Type	Resource Name
SUNW.crs_framework	crs_framework-rs
SUNW.rac_framework	rac_framework-rs

```
phys-schost-1# clresourcetype register SUNW.scalable_asm_instance_proxy
phys-schost-1# clresourcegroup create -g asm-inst-rg
phys-schost-1# clresourcegroup set -p rg_affinities=++rac-fwk-rg asm-inst-rg
phys-schost-1# clresource create -g asm-inst-rg \
-t SUNW.scalable_asm_instance_proxy \
-p oracle_home=/global/11g/asm/product/11.1.0/db_1 \
-p crs_home=/global/11g/crs \
-p "oracle_sid {pmutt1}"=+ASM1 \
-p "oracle_sid {pmutt2}"=+ASM2 \
-p resource_dependencies_offline_restart=crs-fwk-rs qfs-mp-rs -d asm-inst-rs
phys-schost-1# clresourcegroup online -eM asm-inst-rg
```

See Also [Oracle Solaris Cluster Data Services Planning and Administration Guide](#)

Name SUNW.scalable_rac_listener, scalable_rac_listener – resource type implementation for the Oracle 9i Real Application Clusters (RAC) listener managed by Oracle Solaris Cluster

Description The SUNW.scalable_rac_listener resource type represents the Oracle 9i RAC listener in an Oracle Solaris Cluster configuration.

Note – Use the SUNW.scalable_rac_listener resource type *only* if you are using Oracle 9i RAC. If you are using Oracle 10g R2, use the [SUNW.scalable_rac_server_proxy\(5\)](#) resource type.

The SUNW.scalable_rac_listener resource type is a multiple-master resource type. A single resource of this type can run on multiple nodes concurrently, but does not use network load balancing.

Each SUNW.scalable_rac_listener resource represents all Oracle RAC listener instances that serve a database. Each instance of the RAC listener is uniquely identified by the value of the listener_name extension property on the node where the instance is running. The listener_name extension property is a per-node property. A single resource of this type can take a different value of this property for each node.

To register this resource type and create instances of this resource type, use one of the following means:

- The [clsetup\(1CL\)](#) utility, specifying the option for configuring Oracle Solaris Cluster Support for Oracle Real Application Clusters
- The following sequence of Oracle Solaris Cluster maintenance commands:
 1. To register this resource type, use the [clresourcetype\(1CL\)](#) command.
 2. To create instances of this resource type, use the [clresource\(1CL\)](#) command.

Standard Properties For a description of all standard resource properties, see the [r_properties\(5\)](#) man page.

Standard resource properties are overridden for this resource type as follows:

Boot_timeout

Minimum	5
Default	30

Failover_mode

Default	None
Tunable	Any time

Fini_timeout

Minimum	5
----------------	---

Default	30
Init_timeout	
Minimum	5
Default	30
Monitor_start_timeout	
Minimum	5
Default	180
Monitor_stop_timeout	
Minimum	5
Default	120
Retry_count	
Minimum	-1
Default	-1
Tunable	Any time
Retry_interval	
Minimum	-1
Maximum	2592000
Default	600
Tunable	Any time
Start_timeout	
Minimum	5
Default	300
Stop_timeout	
Minimum	5
Default	300
Thorough_probe_interval	
Minimum	1
Maximum	2592000
Default	30

Tunable	Any time
Update_timeout	
Minimum	5
Default	300
Validate_timeout	
Minimum	5
Default	120

Extension Properties The extension properties of the SUNW.scalable_rac_listener resource type are as follows.

debug_level

This property indicates the level to which debug messages from the Oracle RAC listener component are logged. When the debug level is increased, more debug messages are written to the log files. These messages are logged to the file `/var/opt/SUNWscor/scalable_rac_listener/message_log.rs`, where *rs* is the name of the resource that represents the Oracle RAC listener component.

You can specify a different value of the `debug_level` extension property for each node that can master the resource.

Data Type	Integer
Range	0–100
Default	1, which logs syslog messages
Tunable	Any time

listener_name

This property specifies the name of the Oracle listener instance that is to be started on the node where the instance is running. This name must match the corresponding entry in the `listener.ora` configuration file.

You can specify a different value of the `listener_name` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	LISTENER
Tunable	When disabled

oracle_home

This property specifies the full path to the Oracle home directory. The Oracle home directory contains the binary files, log files, and parameter files for the Oracle software.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	When disabled

probe_timeout

This property specifies the timeout value, in seconds, that the fault monitor uses when checking the status of an Oracle RAC listener.

Data Type	Integer
Range	1–99999
Default	180
Tunable	Any time

user_env

This property specifies the name of the file that contains the environment variables that are to be set before the listener starts up or shuts down. You must define all environment variables whose values differ from Oracle defaults in this file.

For example, a user's `listener.ora` file might not be located under the `/var/opt/oracle` directory or the `oracle-home/network/admin` directory. In this situation, the `TNS_ADMIN` environment variable must be defined.

The definition of each environment variable that is defined must follow the format *variable-name=value*. Each definition must start on a new line in the environment file.

You can specify a different value of the `user_env` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	Any time

Examples EXAMPLE 1 Creating a `scalable_rac_listener` Resource

This example shows the commands for performing the following operations to create a `scalable_rac_listener` resource on a two-node cluster:

1. Registering the `SUNW.scalable_rac_server` resource type
2. Adding the `scalable_rac_server-rs` resource to the `rac-db-rg` resource group

A different value of the `listener_name` extension property is set for each node that can master the resource.

EXAMPLE 1 Creating a scalable_rac_listener Resource *(Continued)*

The example makes the following assumptions:

- The C shell is used.
- A RAC framework resource group that is named rac-f-framework-rg exists.
- Logical hostname resources that are named lh1-rs and lh2-rs exist.

```
phys-schost-1# clresourcetype register \
SUNW.scalable_rac_listener
phys-schost-1# clresource create -g rac-db-rg \
-t SUNW.scalable_rac_listener \
-p resource_dependencies_weak=lh1-rs,lh2-rs \
-p oracle_home=/home/oracle/product/9.2.0 \
-p listener_name\{1\}=ORALISTNR1 \
-p listener_name\{2\}=ORALISTNR2 \
scalable_rac_listener-rs
...      Creation of RAC server resource
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/oracle-database

See Also [clresource\(1CL\)](#), [clresourcetype\(1CL\)](#), [clsetup\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.rac_framework\(5\)](#), [SUNW.scalable_rac_server\(5\)](#), [SUNW.scalable_rac_server_proxy\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.scalable_rac_server, scalable_rac_server – resource type implementation for the Oracle 9i Real Application Clusters (RAC) server managed by Oracle Solaris Cluster

Description The SUNW.scalable_rac_server resource type represents the Oracle 9i RAC server in an Oracle Solaris Cluster configuration.

Note – Use the SUNW.scalable_rac_server resource type *only* if you are using Oracle 9i RAC. If you are using Oracle 10g R2, use the [SUNW.scalable_rac_server_proxy\(5\)](#) resource type.

The SUNW.scalable_rac_server resource type is a multiple-master resource type. A single resource of this type can run on multiple nodes concurrently, but does not use network load balancing.

Each SUNW.scalable_rac_server resource represents all Oracle RAC server instances for a database. Each instance of the RAC server is uniquely identified by the value of the `oracle_sid` extension property on the node where the instance is running. The `oracle_sid` extension property is a per-node property. A single resource of this type can take a different value of this property for each node.

Oracle RAC server instances should be started only after the RAC framework is enabled on the cluster node. To ensure that this requirement is met, configure the Oracle RAC server resource and the RAC framework as follows:

- Create a strong positive affinity between the Oracle RAC server resource group and the RAC framework resource group.
- Create a strong dependency between the Oracle RAC server resource and the RAC framework resource.

Create these dependencies and affinities when you configure database resources for the Oracle Solaris Cluster Support for Oracle RAC data service.

To register this resource type and create instances of this resource type, use one of the following means:

- The `clsetup(1CL)` utility, specifying the option for configuring Oracle Solaris Cluster Support for Oracle Real Application Clusters
- The following sequence of Oracle Solaris Cluster maintenance commands:
 1. To register this resource type, use the `clresourcetype(1CL)` command.
 2. To create instances of this resource type, use the `clresource(1CL)` command.

Standard Properties For a description of all standard resource properties, see the [r_properties\(5\)](#) man page.

Standard resource properties are overridden for this resource type as follows:

`Boot_timeout`

Minimum	5
Default	30

Failover_mode	
Default	SOFT
Tunable	Any time
Fini_timeout	
Minimum	5
Default	30
Init_timeout	
Minimum	5
Default	30
Monitor_start_timeout	
Minimum	5
Default	120
Monitor_stop_timeout	
Minimum	5
Default	120
Retry_count	
Minimum	-1
Maximum	9999
Default	2
Tunable	Any time
Retry_interval	
Minimum	0
Maximum	2592000
Default	1330
Tunable	Any time
Start_timeout	
Minimum	5
Default	600

Stop_timeout	
Minimum	5
Default	600
Thorough_probe_interval	
Minimum	1
Maximum	2592000
Default	30
Tunable	Any time
Update_timeout	
Minimum	5
Default	240
Validate_timeout	
Minimum	5
Default	120

Extension Properties The extension properties of the SUNW.scalable_rac_server resource type are as follows.

alert_log_file

This property is set to the absolute path of the Oracle alert log file. The Oracle software logs alerts in this file. The Oracle RAC server fault monitor scans the alert log file for new alerts at the following times:

- When the RAC server fault monitor is started
- Each time that the RAC server fault monitor queries the health of the server

You can specify a different value of the `alert_log_file` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	NULL
Tunable	Any time

connect_cycle

This property specifies the number of fault monitor probe cycles that are performed before the fault monitor disconnects from the database.

You can specify a different value of the `connect_cycle` extension property for each node that can master the resource.

Data Type	Integer
Range	0–99999
Default	5
Tunable	Any time

connect_string

This property specifies the Oracle database user ID and password that the fault monitor uses to connect to the Oracle database. This property is specified as follows:

userid/password

userid

Specifies the Oracle database user ID that the fault monitor uses to connect to the Oracle database.

password

Specifies the password that is set for the Oracle database user *userid*.

The system administrator must define the database user ID and password for the fault monitor during the setup of Oracle RAC. To use Solaris authentication, type a slash (/) instead of a user ID and password.

You can specify a different value of the `connect_string` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	NULL
Tunable	Any time

custom_action_file

This property specifies the absolute path of the file that defines the custom behavior of the Oracle RAC server fault monitor. The format of this file is defined in the [custom_action_file\(4\)](#) man page.

You can specify a different value of the `custom_action_file` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	Empty string
Tunable	Any time

debug_level

This property indicates the level to which debug messages from the Oracle RAC server component are logged. When the debug level is increased, more debug messages are written to the log files. These messages are logged to the file `/var/opt/SUNWscor/scalable_rac_server/message_log.rs`, where *rs* is the name of the resource that represents the Oracle RAC server component.

You can specify a different value of the `debug_level` extension property for each node that can master the resource.

Data Type	Integer
Range	0–100
Default	1, which logs <code>syslog</code> messages
Tunable	Any time

oracle_home

This property specifies the full path to the Oracle home directory. The Oracle home directory contains the binary files, log files, and parameter files for the Oracle software.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	When disabled

oracle_sid

This property specifies the Oracle System Identifier (SID). The Oracle SID uniquely identifies the Oracle RAC database instance on the node where the instance is running.

You *must* specify a different value of the `oracle_sid` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	NULL
Tunable	When disabled

parameter_file

This property specifies the full path to the Oracle parameter file. This file contains parameters that are to be set when the Oracle database is started. This property is optional. If you do not set this property, the default parameter file that is specified by Oracle is used, namely: `oracle-home/dbs/initoracle-sid.ora`.

oracle-home

Specifies the Oracle home directory

oracle-sid

Specifies the Oracle system identifier of the database instance for which the file is to be used.

You can specify a different value of the `parameter_file` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	Any time

probe_timeout

This property specifies the timeout value, in seconds, that the fault monitor uses when checking the status of an Oracle RAC server.

Data Type	Integer
Range	1–99999
Default	300
Tunable	Any time

user_env

This property specifies the name of the file that contains the environment variables that are to be set before the database starts up or shuts down. You must define all environment variables whose values differ from Oracle defaults in this file.

For example, a user's `listener.ora` file might not be located under the `/var/opt/oracle` directory or the `oracle-home/network/admin` directory. In this situation, the `TNS_ADMIN` environment variable must be defined.

The definition of each environment variable that is defined must follow the format `variable-name=value`. Each definition must start on a new line in the environment file.

You can specify a different value of the `user_env` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	Any time

wait_for_online

This property specifies whether the START method of the Oracle RAC server resource waits for the database to be online before the START method exits. The permitted values for this property are as follows:

- | | |
|-------|--|
| True | Specifies that the START method of the Oracle RAC server resource waits for the database to be online before the START method exits. |
| False | Specifies that the START method runs the commands to start the database but does not wait for the database to come online before the START method exits. |

Data Type	Boolean
Range	Not applicable
Default	True
Tunable	Any time

Examples EXAMPLE 1 Creating a scalable_rac_server Resource

This example shows the commands for performing the following operations to create a scalable_rac_server resource on a two-node cluster:

1. Creating the rac-db-rg resource group
2. Registering the SUNW.scalable_rac_server resource type
3. Adding the scalable_rac_server-rs resource to the rac-db-rg resource group

A different value of the following extension properties is set for each node that can master the resource:

- alert_log_file
- oracle_sid

The example makes the following assumptions:

- The C shell is used.
- A RAC framework resource group that is named rac-framework-rg exists and contains resources of types as shown in the following table:

Resource Type	Resource Name
SUNW.rac_framework	rac_framework-rs

- Creation of a resource of type `SUNW.scalable_rac_listener` that is named `scalable_rac_listener-rs` is outside the scope of this example.

```
phys-schost-1# clresourcegroup create \
-p rg_affinities=++rac-framework-rg \
-p desired_primaries=2 \
-p maximum_primaries=2 \
rac-db-rg
...      Creation of RAC listener resource
phys-schost-1# clresourcetype register SUNW.scalable_rac_server
phys-schost-1# clresource create -g rac-db-rg -t SUNW.scalable_rac_server \
-p resource_dependencies=rac_framework-rs \
-p resource_dependencies_weak=scalable_rac_listener-rs \
-p oracle_home=/home/oracle/product/9.2.0 \
-p connect_string=scooter/t!g3r \
-p oracle_sid\{1\}=V920RAC1 \
-p oracle_sid\{2\}=V920RAC2 \
-p alert_log_file\{1\}=/home/oracle/9.2.0/rdbms/log/alert_V920RAC1.log \
-p alert_log_file\{2\}=/home/oracle/9.2.0/rdbms/log/alert_V920RAC2.log \
scalable_rac_server-rs
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/oracle-database

See Also [clresource\(1CL\)](#), [clresourcetype\(1CL\)](#), [clsetup\(1CL\)](#), [custom_action_file\(4\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.rac_framework\(5\)](#), [SUNW.scalable_rac_listener\(5\)](#), [SUNW.scalable_rac_server_proxy\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.scalable_rac_server_proxy, scalable_rac_server_proxy – resource type implementation for the Oracle Real Application Clusters (Oracle RAC) server proxy managed by Oracle Solaris Cluster

Description The SUNW.scalable_rac_server_proxy resource type represents a proxy for the Oracle RAC 10g release 2 or 11g server in an Oracle Solaris Cluster configuration.

Note – Use the SUNW.scalable_rac_server_proxy resource type *only* if you are using Oracle RAC 10g release 2 or 11g. If you are using Oracle 9i, use the SUNW.scalable_rac_server resource type.

Starting in Oracle 10g, Oracle Clusterware manages the startup and shutdown of Oracle RAC server instances. To be managed by the Oracle Clusterware, these instances must be registered with the Oracle Clusterware. The Oracle Clusterware software also provides automatic fault monitoring and failure recovery for Oracle RAC server instances. These instances are represented as resources to Oracle Clusterware.

A resource of type SUNW.scalable_rac_server_proxy is a *proxy* resource: The proxy resource acts as a substitute for a resource that is managed by Oracle Clusterware. The proxy resource enables Oracle Solaris Cluster utilities to manage Oracle RAC server instances *through* Oracle Clusterware. In this way, the SUNW.scalable_rac_server_proxy resource type enables the clustering frameworks that are provided by Oracle Solaris Cluster and Oracle Clusterware to inter-operate.

The SUNW.scalable_rac_server_proxy resource type enables you to use Oracle Solaris Cluster utilities as an alternative to Oracle utilities to start and stop Oracle RAC database instances.

Each resource of type SUNW.scalable_rac_server_proxy has a monitor that obtains the following status information from the Oracle Clusterware resource for which the SUNW.scalable_rac_server_proxy resource is acting as a proxy.

- The online or offline status of Oracle Clusterware
- The status of an Oracle RAC database instance:
 - Online and enabled
 - Online but disabled
 - Offline and disabled
- The success or failure of an attempt to start or stop a database instance
- The ability of Oracle Clusterware to manage the Oracle RAC database instance

The monitor probes the Oracle Clusterware determine if the Oracle Clusterware are managing the Oracle RAC database instance. If the Oracle Clusterware does not indicate that the Oracle Clusterware are managing the Oracle RAC database instance, the monitor warns that the instance is invalid. However, the Oracle Clusterware might not be

managing the Oracle RAC database instance because the instance is unregistered with the Oracle Clusterware. In this situation, the Oracle RAC database instance might be valid, despite the warning.

The timeout period that the monitor uses for obtaining status information is determined by the `proxy_probe_timeout` extension property. If the timeout period is too short, timeouts might cause the monitor to report the status of a valid Oracle RAC database instance as invalid. In this situation, consider increasing the value of the `proxy_probe_timeout` extension property.

This monitor only enables the status of Oracle RAC database instances to be monitored by Oracle Solaris Cluster utilities. This monitor does *not* provide fault monitoring and automatic fault recovery for Oracle RAC database instances. The Oracle Clusterware software provides this functionality.

Oracle RAC server instances should be started only after the Oracle RAC framework and any storage resources are enabled on the cluster node. To ensure that this requirement is met, configure the Oracle RAC server proxy resource as follows:

- Create a strong positive affinity between the Oracle RAC server proxy resource group and the following resource groups:
 - The Oracle RAC framework resource group
 - Any resource group that contains storage resources for Oracle files
- Create a strong dependency between the Oracle RAC server proxy resource and the RAC framework resource.
- Create an offline-restart dependency between the Oracle RAC server proxy resource and the following resources:
 - The Oracle Clusterware framework resource
 - Any storage resources for Oracle files that you are using

Create these dependencies and affinities when you configure database resources for the Oracle Solaris Cluster Support for Oracle RAC data service..

To register this resource type and create instances of this resource type, use one of the following means:

- The `clsetup(1CL)` utility, specifying the option for configuring Oracle Solaris Cluster Support for Oracle Real Application Clusters
- The following sequence of Oracle Solaris Cluster maintenance commands:
 1. To register this resource type, use the `clresourcetype` command.
 2. To create instances of this resource type, use the `clresource` command.

Note – A Solaris project might be specified for a `SUNW.scalable_rac_server_proxy` resource or the resource group that contains a `SUNW.scalable_rac_server_proxy` resource. In this situation, the project affects *only* the processes for the `SUNW.scalable_rac_server_proxy`

resource. The project does *not* affect the processes for any resources that Oracle Clusterware control, including processes for Oracle RAC database instances.

Standard Properties For a description of all standard resource properties, see the [r_properties\(5\)](#) man page.

Standard resource properties are overridden for this resource type as follows:

Boot_timeout

Minimum	5
Default	30

Failover_mode

Default	None
Tunable	Any time

Fini_timeout

Minimum	5
Default	30

Init_timeout

Minimum	5
Default	30

Load_balancing_policy

Default	LB_weighted
Tunable	At creation

Load_balancing_weights

Default	Empty string
Tunable	Any time

Network_resources_used (string array)

A list of logical-hostname or shared-address network resources upon which this resource has a dependency. This list contains all network-address resources that appear in the properties `Resource_dependencies`, `Resource_dependencies_weak`, `Resource_dependencies_restart`, or `Resource_dependencies_offline_restart`.

This property is updated automatically by the RGM, based on the setting of the resource-dependencies properties. You do not set this property directly. Instead, use the `Resource_dependencies` property.

Category Conditional/Optional

Default The empty list

Tunable When disabled

Resource_dependencies (string array)

Specifies a list of resources upon which a resource depends. This list includes any logical-hostname or shared-address network resources that are used by a resource. The default value for this property is null.

You can specify one or more resource names. Each network resource can contain one or more logical host names. See the [clreslogicalhostname\(1CL\)](#) and [clressharedaddress\(1CL\)](#) man pages for more information.

You can specify an alternate kind of dependency by using the Resource_dependencies_weak, Resource_dependencies_restart, or Resource_dependencies_offline_restart property instead of the Resource_dependencies property. For more information, see the [r_properties\(5\)](#) man page.

Category Optional

Default The empty list

Tunable Any time

Port_list

Default None

Tunable At creation

Retry_Count

Maximum 10

Default 2

Tunable Any time

Retry_Interval

Maximum 3600

Default 300

Tunable Any time

Start_timeout

Minimum 5

Default 600

Stop_timeout

Minimum	5
Default	600

Thorough_probe_interval

Minimum	1
Maximum	2592000
Default	20
Tunable	Any time

Update_timeout

Minimum	5
Default	240

Validate_timeout

Minimum	5
Default	120

Extension Properties The extension properties of the SUNW.scalable_rac_server_proxy resource type are as follows.

client_retries

This property specifies the maximum number of attempts by the resource's remote procedure call (RPC) client to connect to the proxy daemon.

Data Type	Integer
Range	1–25
Default	3
Tunable	When disabled

client_retry_interval

This property specifies the interval, in seconds, between attempts by the resource's remote procedure call (RPC) client to connect to the proxy daemon.

Data Type	Integer
Range	1–3600
Default	5
Tunable	When disabled

crs_home

This property specifies the directory in which the Oracle Clusterware software is located.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	When disabled

db_name

This property specifies the name that uniquely identifies the specific Oracle RAC database that is associated with this resource. This identifier distinguishes the database from other databases that might run simultaneously on your system. The name of the Oracle RAC database is specified during the installation of Oracle RAC.

Data Type	String
Range	Not applicable
Tunable	When disabled

debug_level

This property indicates the level to which debug messages from the monitor for the Oracle RAC proxy server are logged. When the debug level is increased, more debug messages are written to the log files.

The messages are logged to files in the directory `/var/opt/SUNWscor/oracle_server/proxyrs`. Messages for server-side components and client-side components of the proxy server resource are written to separate files:

- Messages for server-side components are written to the file `message_log.rs`.
- Messages for client-side components are written to the file `message_log.client.rs`.

In these file names and directory names, *rs* is the name of the resource that represents the Oracle RAC server component.

You can specify a different value of the `debug_level` extension property for each node that can master the resource.

Data Type	Integer
Range	0–100
Default	1, which logs <code>syslog</code> messages
Tunable	Any time

monitor_probe_interval

This property specifies the interval, in seconds, between probes of the Oracle Clusterware resource for which this resource is acting as a proxy.

Data Type	Integer
Range	10–3600
Default	300
Tunable	Any time

oracle_home

This property specifies the full path to the Oracle home directory. The Oracle home directory contains the binary files, log files, and parameter files for the Oracle software.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	When disabled

oracle_sid

This property specifies the Oracle System Identifier (SID). The Oracle SID uniquely identifies the Oracle RAC database instance on the node where the instance is running.

You *must* specify a different value of the `oracle_sid` extension property for each node that can master the resource. The value for each node must correctly identify the instance that is running on the node.

Data Type	String
Range	Not applicable
Default	NULL
Tunable	When disabled

proxy_probe_timeout

This property specifies the timeout value, in seconds, that the proxy monitor uses when checking the status of the Oracle Clusterware resource for which this resource is acting as a proxy.

Data Type	Integer
Range	5–3600
Default	120
Tunable	Any time

startup_wait_count

This property specifies the maximum number of attempts by this resource to confirm that the Oracle Clusterware software is started completely. The interval between attempts is twice the value of the `proxy_probe_timeout` extension property.

The resource requires confirmation that Oracle Clusterware software is started before attempting to start an Oracle RAC database instance. If the maximum number of attempts is exceeded, the resource does not attempt to start the database instance.

Data Type	Integer
Range	10–600
Default	20
Tunable	When disabled

user_env

This property specifies the name of the file that contains the environment variables that are to be set before the database starts up or shuts down. You must define all environment variables whose values differ from Oracle defaults in this file.

For example, a user's `listener.ora` file might not be located under the `/var/opt/oracle` directory or the `oracle-home/network/admin` directory. In this situation, the `TNS_ADMIN` environment variable must be defined.

The definition of each environment variable that is defined must follow the format *variable-name=value*. Each definition must start on a new line in the environment file.

You can specify a different value of the `user_env` extension property for each node that can master the resource.

Data Type	String
Range	Not applicable
Default	No default defined
Tunable	Any time

Examples EXAMPLE 1 Creating a scalable_rac_server_proxy Resource

This example shows the commands for performing the following operations to create a `scalable_rac_server_proxy` resource on a two-node cluster:

1. Creating the `rac-proxy-db-rg` resource group
2. Registering the `SUNW.scalable_rac_server_proxy` resource type
3. Adding the `scalable_rac_server_proxy-rs` resource to the `rac-proxy-db-rg` resource group

A different value of the `oracle_sid` extension property is set for each node that can master the resource.

The example makes the following assumptions:

- The C shell is used.

EXAMPLE 1 Creating a scalable_rac_server_proxy Resource (Continued)

- An Oracle RAC database that is named V1020RAC is registered with the Oracle Clusterware.
- A resource group that is named scal-dg-rg exists and contains a resource of type SUNW.ScalDeviceGroup that is named scal-dg-rs.
- An Oracle RAC framework resource group that is named rac-framework-rg exists and contains resources that are shown in the following table:

Resource Type	Resource Name
SUNW.crs_framework	crs_framework-rs
SUNW.rac_framework	rac_framework-rs

```

phys-schost-1# clresourcegroup create \
-p rg_affinities=++rac-framework-rg,++scal-dg-rg \
-p desired_primaries=2 \
-p maximum_primaries=2 \
rac-proxy-db-rg
phys-schost-1# clresourcetype register \
SUNW.scalable_rac_server_proxy
phys-schost-1# clresource create -g rac-proxy-db-rg \
-t SUNW.scalable_rac_server_proxy \
-p resource_dependencies=rac_framework-rs \
-p resource_dependencies_offline_restart=crs-framework-rs,scal-dg-rs \
-p oracle_home=/home/oracle/product/10.2.0/oracle_install \
-p crs_home=/home/oracle/product/10.2.0/crs_install \
-p db_name=V1020RAC \
-p oracle_sid\{1\}=V1020RAC1 \
-p oracle_sid\{2\}=V1020RAC2 \
scalable_rac_server_proxy-rs

```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/oracle-database

See Also [clresource\(1CL\)](#), [clresourcetype\(1CL\)](#), [clsetup\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#), [SUNW.scalable_rac_server\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.sybase, sybase – resource type implementation for Oracle Solaris Cluster HA for Sybase Adaptive Server Enterprise (ASE)

Description The SUNW. sybase resource type represents the Sybase ASE application in an Oracle Solaris Cluster configuration.

Standard properties and extension properties that are defined for the SUNW. sybase resource type are described in the subsections that follow. To set these properties for an instance of the SUNW. sybase resource type, use the `clresource(1CL)` command or a resource-configuration GUI.

Standard Properties Set the standard resource property `Failover` for all failover resource types.

Standard resource properties are overridden for this resource type as follows:

`Failover_mode`
Default: SOFT

Tunable: Any time

`Retry_count`
Minimum: 0

Default: 2

Tunable: Any time

`Retry_interval`
Minimum: 0

Default: 600

Tunable: Any time

`Thorough_probe_interval`
Minimum: 1

Default: 30

Tunable: Any time

For a description of these standard resource properties, see [r_properties\(5\)](#).

Extension Properties `Adaptive_Server_Log_File`

Type string. Set this property as the absolute path of the Sybase ASE data-server log to which Sybase ASE logs errors. As part of the probe execution, the Sybase ASE data-server fault monitor scans this log file for errors. The fault monitor matches the error numbers for possible actions with patterns that the Oracle Solaris Cluster HA for Sybase ASE action file, `/opt/SUNWsyb/etc/sybase_actions`, specifies. You can modify this property at any time. No default value exists for this field. Minimum value is 1. You must set this property.

Adaptive_Server_Name

Type string. This property specifies the adaptive-server name, which enables the Oracle Solaris Cluster HA for Sybase ASE data service to locate and execute the RUN server script. This script is located in the `$SYBASE/ASE_<major-version>/install` directory, where *major-version* is the major version of Sybase ASE that you are using. For example, if you are using Sybase ASE version 12.5.1, *major-version* is 12-5. You can modify this property only when you have disabled the resource. No default value exists for this field. Minimum value is 1. You must set this property.

Backup_Server_Name

Type string. This property specifies the backup-server name, which enables the Oracle Solaris Cluster HA for Sybase ASE data service to locate and execute the RUN server script. This script is located in the `$SYBASE/ASE_<major-version>/install` directory, where *major-version* is the major version of Sybase ASE that you are using. For example, if you are using Sybase ASE version 12.5.1, *major-version* is 12-5. You can modify this property only when you have disabled the resource. Setting this property is optional, but if you do not set the property, the Oracle Solaris Cluster HA for Sybase ASE data service will not manage the server.

Connect_cycle

Type integer. Default is 5. Minimum value is 1. The Sybase ASE data-server fault monitor uses the user ID and password that the `Connect_string` property specifies to periodically connect to the database. After executing the number of probes that this property specifies, the monitor disconnects and then reconnects. You can modify the value for this property at any time.

Connect_string

Type string. Set this property to the database user's user ID and password in fault-monitor transactions. Specify this property as follows:

userid/password

When you set up the Oracle Solaris Cluster HA for Sybase ASE data service, define the database user ID and password before you enable the server resource and the server resource's fault monitor. Do *not* use the `sa` account for the database user. You can modify this property at any time. No default value exists for this field. Minimum value is 1. You must set this property, even if you do not set the `Monitor_Server_Name` property.

Debug_level

Type integer. Default is 1. Minimum value is 0. Maximum value is 100. This property indicates the debug level for writing to the Oracle Solaris Cluster HA for Sybase ASE log. You can modify the value for this property at any time.

Environment_File

Type string. This property specifies the absolute file path of the environment file (typically `SYBASE.sh`) that is provided with the Sybase ASE distribution. Before executing any method or program, the Oracle Solaris Cluster HA for Sybase ASE data service reads this

file and sets the environment accordingly. You can modify this property only when you have disabled the resource. No default value exists for this field. Minimum value is 1. You must set this property.

Monitor_Server_Name

Type string. This property specifies the monitor-server name, which enables the Oracle Solaris Cluster HA for Sybase ASE data service to locate and execute the RUN server script. This script is located in the `$SYBASE/ASE_<major-version>/install` directory, where *major-version* is the major version of Sybase ASE that you are using. For example, if you are using Sybase ASE version 12.5.1, *major-version* is 12-5. You can modify this property only when you have disabled the resource. Setting this property is optional, but if you do not set the property, the Oracle Solaris Cluster HA for Sybase ASE data service will not manage the server.

Probe_timeout

Type integer. Default is 60 seconds. Minimum value is 1. This property is the timeout value that the fault monitor uses to probe a Sybase ASE server instance. You can modify the value for this property at any time.

Stop_File

Type string. This property indicates the absolute path to the script that the STOP method executes to stop the Sybase ASE servers. This file stores the password of the Sybase ASE system administrator (sa). Protect the path so that only the user and group that are associated with the Sybase ASE installation can access the file. The Oracle Solaris Cluster HA for Sybase ASE package includes the `sybase_stop_servers` template. You must replace the existing password. You can modify this property at any time. No default value exists for this field. Minimum value is 1. You must set this property.

Text_Server_Name

Type string. This property specifies the text-server name, which enables the Oracle Solaris Cluster HA for Sybase ASE data service to locate and execute the RUN server script. This script is located in the `$SYBASE/ASE_<major-version>/install` directory, where *major-version* is the major version of Sybase ASE that you are using. For example, if you are using Sybase ASE version 12.5.1, *major-version* is 12-5. You can modify this property only when you have disabled the resource. Setting this property is optional, but if you do not set the property, the Oracle Solaris Cluster HA for Sybase ASE data service will not manage the server.

Wait_for_online

Type Boolean. Default is TRUE. This property specifies whether the START method waits for the database to become active before exiting. If you set this property to TRUE, the START method starts the database and waits for the database to become active before exiting. You can modify the value for this property at any time.

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	ha-cluster/data-service/sybase

See Also [pmfadm\(1M\)](#), [scha_resource_get\(1HA\)](#), [clresource\(1CL\)](#), [clresourcegroup\(1CL\)](#), [attributes\(5\)](#), [r_properties\(5\)](#)

Oracle Solaris Cluster Data Services Planning and Administration Guide

Name SUNW.wls, wls – resource type implementation for failover WebLogic Server

Description The Resource Group Manager (RGM) manages Oracle Solaris Cluster HA for WebLogic Server for Oracle Solaris Cluster. Use the `clresource(1CL)` command or a resource configuration GUI to set the following properties on WebLogic Server resources.

Standard Properties See [r_properties\(5\)](#) for a complete description of the following resource properties.

Failover_mode

Default SOFT

Tunable Any time

Probe_timeout

Minimum 60

Default 180

Tunable Any time

Retry_count

Maximum 10

Default 2

Tunable Any time

Retry_interval

Maximum 3600

Default 1220

Tunable Any time

Thorough_probe_interval

Maximum 3600

Default 120

Tunable Any time

Extension Properties `Confdir_list`

Type string array. No default value exists. Use this property to set the complete path to the WebLogic Server home directory, `$WL_HOME`.

You can modify this property only when you create the resource.

`DB_Probe_Script`

Type string. Default is null. This extension property can be used to set the complete path to a database probe script. The HA-WLS probe method probes only the WLS instances. The

database probe script can be provided by using this extension property if the administrators want the HA_WLS to probe the database also. This probe script must return 0 for success. The WebLogic Servers are started only if this database probe returns success. If an HA-WLS probe detects a failure in a WebLogic Server instance and if this extension property is set, the HA-WLS probe takes action only if the database probe succeeds.

You can modify this property at any time.

Environment_File

Type string. Default is null. Use this property to set the absolute path to the WebLogic Server environment file. This file is typically installed in the \$WEBLOGIC_HOME/server/bin directory when WebLogic Server is configured by using the installation wizard.

You can modify this property when the resource is disabled.

Monitor Uri List

Type string. Default is null. This property indicates the URI or list of URIs, separated by a comma, that can be used by the fault monitor to test the functionality of the WebLogic Server by running an HTTP GET command on the URI. The `Server_Url` extension property is for simple probes on the hostname and port. This extension property can be used to probe the WebLogic Server functionality by retrieving a Java servlet or making the WebLogic Server run an application and return an HTTP server code. If the HTTP server return code is 500 (Internal Server Error) or if the connection fails, the probe takes action. See the probe method for more details.

Make sure that the hostnames used in the `server_url` and `monitor_uri_list` are resolvable. If you use Fully Qualified Domain Names (FQDNs), then DNS must be enabled and `/etc/nsswitch.conf` must have the correct entries to resolve the hostnames by using DNS.

You can modify this property at any time.

Server_name

Type string. Default is null. A single start script can be used to start all the Managed Servers by passing the Managed Server name as an argument to the START script. If Agent START methods have to pass the server name as an argument to the START script, configure the Managed Server name in this extension property.

You can modify this property at any time.

Server_Url

Type string. No default value exists. This property indicates the URL of the WebLogic Server. The URL includes the protocol that is used to connect to the server, that this resource is configured to start. The probe method uses this URL to check the health of the server by running an HTTP GET command on the URL. The protocol in the URL must be set to `http`. The complete URL should be in the following form.

```
http://host:port
```

Make sure that the hostnames used in the `server_url` and `monitor_uri_list` are resolvable. If you use Fully Qualified Domain Names (FQDNs), then DNS must be enabled and `/etc/nsswitch.conf` must have the correct entries to resolve the hostnames by using DNS.

You can modify this property only when you create the resource.

Smooth_shutdown

Type Boolean. Default is `False`. This extension property can be used to enable smooth shutdown by using the `WebLogic.Admin` class. This extension property must be set to `TRUE` if a smooth shutdown is desired before trying to kill the WebLogic Server process. If this extension property is `TRUE`, the `WLS_USER` and `WLS_PW` must be set in the `start_script` and not in `boot.properties`.

Set this extension property to `TRUE` if all of the following apply.

- Setting the username and password in the `start_script` is not a concern.
- A smooth shutdown is desirable instead of the default, killing the process.
- You are not concerned if the agent STOP method reads the user name and password from the START script and passes it to the `java weblogic.Admin` command.

You can modify this property at any time.

Start_Script

Type string. No default value exists. Use this property to set the complete path to the script that should be used to start the WebLogic Server instance (either Administration or Managed). This script is typically present under the domain name directory along with the `config.xml` file. A separate script or a single script can be configured for starting each WebLogic Server.

You can modify this property only when you create the resource.

Examples

EXAMPLE 1 Creating a Simple WebLogic Server Resource

This example assumes that the START script, `startWebLogic.sh`, can start the WebLogic Server without any arguments to the script. The username and password needed to start the WebLogic Server can be configured within this START script or in the `boot.properties` file.

```
clresource create -g bea-rg -t SUNW.wls \
-p Confdir_list=/global/bea/beahome/weblogic700 \
-p Server_url=http://logical-host-1:7001 \
-p Start_script=/global/bea/beahome/user_projects/ha-wls/startWebLogic.sh bea-rs
```

EXAMPLE 2 Creating a Managed Server Resource Whose Start Script Takes a Managed Server Name as an Input

For this example to work, the `Admin_URL` must be set within the `Start_script` `startManagedWebLogic.sh`.

EXAMPLE 2 Creating a Managed Server Resource Whose Start Script Takes a Managed Server Name as an Input *(Continued)*

```
clresource create -g bea-rg -t SUNW.wls \
-p Confdir_list=/global/bea/beahome/weblogic700 \
-p Server_url=http://logical-host-1:7004 \
-p Start_script=/global/bea/beahome/user_projects/ha-wls/startManagedWebLogic.sh \
-p Server_name=test1 bea-rs1
```

EXAMPLE 3 Creating a WebLogic Server Managed Server Resource Which Should Be Shut Down Smoothly

This example creates a resource that has the extension property `Smooth_shutdown` set to `TRUE`. Setting this extension property to `TRUE` specifies that WebLogic Server will shut down the resource smoothly. If this extension property is not set to `TRUE`, the `STOP` method sends `sigkill` to the WebLogic Server. For the resource created in this example, the WebLogic Server first tries a smooth shutdown by using the `WebLogic.Admin` class. If this attempt is not successful, WebLogic Server uses `sigkill`. The `WLS_PW` and `WLS_USER` must be set in the `START` script `startManagedWebLogic.sh`. If these two parameters are not set in the `START` script, the resource creation fails.

```
clresource create -g bea-rg -t SUNW.wls \
-p Confdir_list=/global/bea/beahome/weblogic700 \
-p Server_url=http://logical-host-1:7008 \
-p Start_script=/global/bea/beahome/user_projects/ha-wls/startManagedWebLogic.sh \
-p Server_name=test2 -p smooth_shutdown=true bea-rs2
```

EXAMPLE 4 Creating a WebLogic Server Resource that Should Probe the Database Before Taking Any Action on the WebLogic Server

This example creates a WebLogic Server resource that should probe the database before taking any action on the WebLogic Server. If the `db_probe` script is set, the WebLogic Server will not be started if the script returns a failure. When the WebLogic Server probe fails, action is taken only if the `db_probe_script` returns 0. This user-supplied database probe script must return 0 for success and non zero for failure.

```
clresource create -g bea-rg -t SUNW.wls \
-p Confdir_list=/global/bea/beahome/weblogic700 \
-p Server_url=http://logical-host-1:710 \
-p Start_script=/global/bea/beahome/user_projects/ha-wls/startManagedWebLogic.sh \
-p Server_name=test3 \
-p db_probe_script=/global/phys-pale-1/bea-db_probe_script bea-rs3
```

EXAMPLE 5 Creating a WebLogic Server Resource that Should Also Monitor Some URIs Along With the `Server_url` Monitoring

This example creates a WebLogic Server resource that monitors URIs by using the `Monitor_uri_list` extension property. Action is taken only if the URI returns an `http` error 500 or if the connection to the WebLogic Server fails.

EXAMPLE 5 Creating a WebLogic Server Resource that Should Also Monitor Some URIs Along With the Server_url Monitoring (Continued)

```
clresource create -g bea-rg -t SUNW.wls \
-p Confdir_list=/global/bea/beahome/weblogic700 \
-p Server_url=http://logical-host-1:7012 \
-p Start_script=/global/bea/beahome/user_projects/ha-wls/startManagedWebLogic.sh \
-p Server_name=test5 \
-p db_probe_script=/global/bea/db_probe_script \
-p monitor_uri_list=http://logical-host-1:7001/sctest bea-rs5
```

Attributes See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/data-service/weblogic

See Also [attributes\(5\)](#), [clresource\(1CL\)](#), [r_properties\(5\)](#), [scha_resource_get\(3HA\)](#)

Index

A

alert_log_file extension property
 scalable_rac_server resource type, 146
 SUNW.scalable_rac_server resource type, 146
apache, 26
asm_diskgroups extension property, 124

C

Child_mon_level extension property
 SUNW.sapenq resource type, 90
 SUNW.saprepl resource type, 99
client_retries extension property, 156
client_retry_interval extension property, 156
connect_cycle extension property
 scalable_rac_server resource type, 146
 SUNW.scalable_rac_server resource type, 146
connect_string extension property
 scalable_rac_server resource type, 147
 SUNW.scalable_rac_server resource type, 147
crs_home extension property, 136, 157
custom_action_file extension property
 scalable_rac_server resource type, 147
 SUNW.scalable_rac_server resource type, 147
custom_action_file file, 10

D

database users, 87

databases

 instance name, 137, 148, 158
db_name extension property, 157
DB_Name extension property, 85
DB_User extension property, 85
dbmcli command
 path to, 86
 start option, 84
dbmcli_Start_Option extension property, 84
debug_level extension property
 scalable_asm_diskgroup_proxy resource
 type, 124
 scalable_asm_instance_proxy resource type, 136
 scalable_rac_listener resource type, 141
 scalable_rac_server_proxy resource type, 157
 scalable_rac_server resource type, 148
 SUNW.scalable_asm_diskgroup_proxy resource
 type, 124
 SUNW.scalable_asm_instance_proxy resource
 type, 136
 SUNW.scalable_rac_listener resource type, 141
 SUNW.scalable_rac_server_proxy resource
 type, 157
 SUNW.scalable_rac_server resource type, 148
directories
 Oracle home, 137, 141, 148, 158
dns, 39

E

Enqueue_Instance_Number extension property,
SUNW.sapenq resource type, 90

Enqueue_Profile extension property, SUNW.sapenq
resource type, 90

Enqueue_Server extension property, SUNW.sapenq
resource type, 91

Enqueue_Server_Monitor extension property,
SUNW.sapenq resource type, 91

environment variables, 142, 149, 159

F

Failover_enabled extension property, 85

Failover_Enabled extension property, SUNW.sapsccs
resource type, 105

files

- custom_action_file, 10
- process identity, 86

H

home directory

- Oracle, 137, 141, 148, 158

I

identifiers

- system, 137, 148, 158

Independent_Program_Path extension property,
SUNW.sapdb resource type, 85

iws, 43

J

jsas-na Sun Java System Application Server, resource
type implementation for failover Sun Java System
Application Server, 52

jsas Sun Java System Application Server, 48

K

kernel processes

- identities

 - paths to files containing, 86
 - termination of parent, 87

krb5 resource type, 56

L

ldom, 59

libraries

- MaxDB

 - SUNW.sapdb resource type, 85

listener_name extension property

- scalable_rac_listener resource type, 141
- SUNW.scalable_rac_listener resource type, 141

Log_Directory extension property

- SUNW.sapenq resource type, 91, 112
- SUNW.saprepl resource type, 99

log files

- Oracle RAC server proxy, 157
- RAC listener, 141
- RAC server, 148

M

MaxDB application

- paths to programs and libraries

 - SUNW.sapdb resource type, 85

process identities

- paths to files containing, 86

MaxDB database instance

- database user, 87
- name, 85
- operating system user, 85
- starting, 84

maximum values

- restarts

 - SUNW.krb5 resource type, 57
 - SUNW.sapdb resource type, 86

messages

- debug, 124, 136, 141, 148, 157

monitor_probe_interval extension property, 157

Monitor_retry_count extension property
 SUNW.krb5 resource type, 57
 SUNW.sapdb resource type, 86
 SUNW.sapenq resource type, 91
 SUNW.saprepl resource type, 99
 Monitor_Retry_Count extension property
 SUNW.sapscs resource type, 105
 SUNW.sapwebas resource type, 111
 Monitor_retry_interval extension property
 SUNW.krb5 resource type, 57
 SUNW.sapdb resource type, 86
 SUNW.sapenq resource type, 91
 SUNW.saprepl resource type, 100
 Monitor_Retry_Interval extension property
 SUNW.sapscs resource type, 105
 SUNW.sapwebas resource type, 111
 Msg_Server_Monitor extension property, SUNW.sapscs
 resource type, 106
 Msg_Server_Port extension property, SUNW.sapscs
 resource type, 106

N

names
 MaxDB database instance, 85
 Oracle database instance, 137, 148, 158
 nfs, 63

O

obiee_cluster_controller, 18
 obiee_presentation_service, 20
 obiee_scheduler, 22
 obiee_server, 24
 operating system users, of MaxDB database
 instance, 85
 oracle_home extension property
 scalable_asm_instance_proxy resource type, 137
 scalable_rac_listener resource type, 141
 scalable_rac_server_proxy resource type, 158
 scalable_rac_server resource type, 148
 SUNW.scalable_asm_instance_proxy resource
 type, 137

oracle_home extension property (*Continued*)
 SUNW.scalable_rac_listener resource type, 141
 SUNW.scalable_rac_server_proxy resource
 type, 158
 SUNW.scalable_rac_server resource type, 148
 oracle_listener, 67
 oracle_server, 69
 oracle_sid extension property
 scalable_asm_instance_proxy resource type, 137
 scalable_rac_server_proxy resource type, 158
 scalable_rac_server resource type, 148
 SUNW.scalable_asm_instance_proxy resource
 type, 137
 SUNW.scalable_rac_server_proxy resource
 type, 158
 SUNW.scalable_rac_server resource type, 148
 ORCL.obiee_cluster_controller, 18
 ORCL.obiee_presentation_service, 20
 ORCL.obiee_scheduler, 22
 ORCL.obiee_server, 24

P

parameter_file extension property
 scalable_rac_server resource type, 148
 SUNW.scalable_rac_server resource type, 148
 paths
 dbmcli command, 86
 MaxDB programs and libraries
 SUNW.sapdb resource type, 85
 process identity files, 86
 Pid_Dir_Path extension property, 86
 probe_timeout extension property
 scalable_rac_listener resource type, 142
 scalable_rac_server resource type, 149
 Probe_timeout extension property
 SUNW.krb5 resource type, 58
 SUNW.sapdb resource type, 87
 SUNW.sapenq resource type, 92
 SUNW.saprepl resource type, 100
 probe_timeout extension property
 SUNW.scalable_rac_listener resource type, 142
 SUNW.scalable_rac_server resource type, 149

Probe_Timeout extension property
 SUNW.sapscs resource type, 106
 SUNW.sapwebas resource type, 111

processes

identities
 paths to files containing, 86
 termination of parent, 87

programs

MaxDB
 SUNW.sapdb resource type, 85

proxy_probe_timeout extension property, 125, 137, 158

R

Replica_Profile extension property, SUNW.saprepl
 resource type, 100

Replica_Server extension property, SUNW.saprepl
 resource type, 100

resource type implementation for failover and scalable
 Apache Web Server, 26

resource type implementation for failover and scalable
 Sun Java System Web Server, 43

resource type implementation for failover Domain
 Name Service (DNS), 39

resource type implementation for failover guest
 domains, 59

resource type implementation for failover SAP
 liveCache database, 95

resource type implementation for failover Siebel
 gateway, 119

resource type implementation for failover Siebel
 server, 121

resource type implementation for HA-Oracle Business
 Intelligence Enterprise Edition (Oracle BI EE)
 Cluster Controller, 18

resource type implementation for HA-Oracle Business
 Intelligence Enterprise Edition (Oracle BI EE)
 Presentation Service, 20

resource type implementation for HA-Oracle Business
 Intelligence Enterprise Edition (Oracle BI EE)
 Scheduler, 22

resource type implementation for HA-Oracle Business
 Intelligence Enterprise Edition (Oracle BI EE)
 server, 24

resource type implementation for HA Oracle server, 69
 resource type implementation for Oracle Solaris Cluster
 HA for NFS, 63

resource type implementation for Oracle Solaris Cluster
 HA for SAP application server, 78

resource type implementation for Oracle Solaris Cluster
 HA for SAP central instance., 81

resource type implementation for Oracle Solaris Cluster
 HA for Sybase Adaptive Server Enterprise
 (ASE), 161

resource type implementation for scalable SAP
 server, 116

resource type implementation for the Oracle
 listener, 67

resource types

krb5, 56
 sapdb, 84
 sapenq, 89
 saprepl, 98
 sapscs, 104
 sapwebas, 110
 SUNW.krb5, 56
 SUNW.sapdb, 84
 SUNW.sapenq, 89
 SUNW.saprepl, 98
 SUNW.sapscs, 104
 SUNW.sapwebas, 110

Restart_if_Parent_Terminated extension
 property, 87

restarts

interval between
 SUNW.krb5 resource type, 57
 SUNW.sapdb resource type, 86
 maximum allowed
 SUNW.krb5 resource type, 57
 SUNW.sapdb resource type, 86

S

s1as Sun Java System Application Server, 74

sap_as, 78

- sap_as_v2, 78
- SAP central server
 - resource type, 104, 110
- sap_ci, 81
- sap_ci_v2, 81
- SAP_Instance_Name extension property
 - SUNW.sapscs resource type, 106
 - SUNW.sapwebas resource type, 111
- SAP_Instance_Number extension property
 - SUNW.sapscs resource type, 106
 - SUNW.sapwebas resource type, 111
- sap_livecache, 95
- SAP replicated enqueue replica server, resource type, 98
- SAP replicated enqueue server, resource type, 89
- SAP_SID extension property
 - SUNW.sapscs resource type, 107
 - SUNW.sapwebas resource type, 112
- SAP_User extension property
 - SUNW.sapenq resource type, 92
 - SUNW.saprepl resource type, 100
 - SUNW.sapscs resource type, 107
 - SUNW.sapwebas resource type, 112
- sap_xserver, 116
- sapdb resource type, 84
- sapenq resource type, 89
- saprepl resource type, 98
- sapscs resource type, 104
- sapwebas resource type, 110
- sblgtwy, 119
- sblsrvr, 121
- Scs_Shutdown_Script extension property,
 - SUNW.sapscs resource type, 107
- Scs_Startup_Script extension property, SUNW.sapscs resource type, 107
- starting, MaxDB database instance, 84
- startup_wait_count extension property, 158
- Stop_signal extension property
 - SUNW.sapenq resource type, 92
 - SUNW.saprepl resource type, 101
- Sun Java System Application Server, 52
 - resource type implementation for failover and scalable Sun Java System Application Server, 52
- Sun Java System Application Server (*Continued*)
 - resource type implementation for failover and scalable SunJava System Application Server, 48
 - resource type implementation for failover Sun Java System Application Server, 48
- Sun Open Net Environment (Sun ONE) Application Server (iWS)
 - resource type implementation for failover and scalable SunJava System Application Server, 74
 - resource type implementation for failover Sun Java System Application Server, 74
- SUNW.apache, 26
- SUNW.dns, 39
- SUNW.iws, 43
- SUNW.jsas-na Sun Java System Application Server, 52
- SUNW.jsas Sun Java System Application Server, 48
- SUNW.krb5
 - resource type for collecting data on system resource usage, 56
 - resource type implementation of Kerberos KDC server, 56
- SUNW.krb5 resource type, 56
- SUNW.ldom, 59
- SUNW.nfs, 63
- SUNW.oracle_listener, 67
- SUNW.oracle_server, 69
- SUNW.s1as Sun Java System Application Server, 74
- SUNW.sap_as, 78
- SUNW.sap_as_v2, 78
- SUNW.sap_ci, 81
- SUNW.sap_ci_v2, 81
- SUNW.sap_livecache, 95
- SUNW.sap_xserver, 116
- SUNW.sapdb resource type, 84
- SUNW.sapenq resource type, 89
- SUNW.saprepl resource type, 98
- SUNW.sapscs resource type, 104
- SUNW.sapwebas resource type, 110
- SUNW.sblgtwy, 119
- SUNW.sblsrvr, 121
- SUNW.sybase, 161
- sybase, 161
- syslog messages, 124, 136, 141, 148, 157

system identifiers
Oracle, 137, 148, 158

X
.XUSER.62 file, and DB_User extension property, 85

T

timeouts
fault monitor
SUNW.krb5 resource type, 58
SUNW.sapdb resource type, 87

U

user_env extension property
scalable_rac_listener resource type, 142
scalable_rac_server_proxy resource type, 159
scalable_rac_server resource type, 149
SUNW.scalable_rac_listener resource type, 142
SUNW.scalable_rac_server_proxy resource
type, 159
SUNW.scalable_rac_server resource type, 149
User_Key extension property, 87
users
of MaxDB database instance
database, 87
operating system, 85

W

wait_for_online extension property
scalable_rac_server resource type, 150
SUNW.scalable_rac_server resource type, 150
Webas_Probe_J2ee extension property, SUNW.sapwebas
resource type, 112
Webas_Shutdown_Script extension property,
SUNW.sapwebas resource type, 113
Webas_Startup_Script extension property,
SUNW.sapwebas resource type, 113
Webas_Use_Pmf extension property, SUNW.sapwebas
resource type, 113