Oracle® Solaris Cluster Data Services
Reference Manual
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 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 END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 Corporation 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 États-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l’utilise pour le compte du Gouvernement des États-Unis, la notice suivante s’applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 pas conçu ni n’est destiné à être utilisé 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, dites ou 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.

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 d’Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l’accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l’accès à des contenus, produits ou services tiers, ou à leur utilisation.
# Contents

**Preface** ..................................................................................................................................................... 5

**OSC33DS 4** .................................................................................................................................................. 9
custom_action_file(4) .................................................................................................................................. 10

**OSC33DS 5** .................................................................................................................................................. 17
ORCL.obiee_cluster_controller(5) ........................................................................................................... 18
ORCL.obiee_presentation_service(5) ......................................................................................................... 20
ORCL.obiee_scheduler(5) ......................................................................................................................... 22
ORCL.obiee_server(5) ................................................................................................................................. 24
ORCL.ohs(5) .............................................................................................................................................. 26
ORCL.opmn(5) ............................................................................................................................................ 28
ORCL.oracle_external_proxy(5) .................................................................................................................. 30
ORCL.sapcentr(5) ...................................................................................................................................... 34
ORCL.sapdia(5) ......................................................................................................................................... 38
ORCL.saprepenq(5) ................................................................................................................................... 43
ORCL.saprepenq_preempt(5) ..................................................................................................................... 47
ORCL.sapstartsrv(5) .................................................................................................................................. 51
SUNW.apache(5) ...................................................................................................................................... 55
SUNW.asm_diskgroup(5) ........................................................................................................................... 61
SUNW.dns(5) ............................................................................................................................................. 69
SUNW.hadb(5) .......................................................................................................................................... 73
SUNW.hadb_ma(5) .................................................................................................................................... 76
SUNW.iws(5) ............................................................................................................................................. 80
SUNW.jsas(5) ............................................................................................................................................. 86
SUNW.jsas-na(5) ....................................................................................................................................... 90
SUNW.krb5(5) .......................................................................................................................................... 94
| SUNW.ldom(5) | 97 |
| SUNW.nfs(5) | 101 |
| SUNW.oracle_listener(5) | 105 |
| SUNW.oracle_rac_server(5) | 107 |
| SUNW.oracle_server(5) | 112 |
| SUNW.slas(5) | 117 |
| SUNW.s1mq(5) | 121 |
| SUNW.sap_as(5) | 124 |
| SUNW.sap_ci(5) | 127 |
| SUNW.sapdb(5) | 130 |
| SUNW.sapenq(5) | 135 |
| SUNW.sap_livecache(5) | 141 |
| SUNW.saprepl(5) | 144 |
| SUNW.sapscs(5) | 150 |
| SUNW.sapwebas(5) | 156 |
| SUNW.sap_xserver(5) | 162 |
| SUNW.sblgtwy(5) | 165 |
| SUNW.sblsrvr(5) | 167 |
| SUNW.scalable_acfs_proxy(5) | 169 |
| SUNW.scalable_asm_diskgroup_proxy(5) | 173 |
| SUNW.scalable_asm_instance(5) | 177 |
| SUNW.scalable_asm_instance_proxy(5) | 184 |
| SUNW.scalable_rac_listener(5) | 189 |
| SUNW.scalable_rac_server(5) | 194 |
| SUNW.scalable_rac_server_proxy(5) | 202 |
| SUNW.sybase(5) | 212 |
| SUNW.wls(5) | 216 |

**Index** ........................................................................................................................................ 221
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 intro 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 | 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. The following special characters are used in this section: |
| [ ] | 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 ioctl(2) system call is called ioctl and generates its own heading. ioctl calls for a specific device are listed alphabetically (on the man page for that specific device). |
ioctl calls are used for a particular class of devices. All these calls have an io ending, such as `mtio(7I)`.

**OPTIONS**
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.

**OPERANDS**
This section lists the command operands and describes how they affect the actions of the command.

**OUTPUT**
This section describes the output – standard output, standard error, or output files – generated by the command.

**RETURN VALUES**
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.

**ERRORS**
On failure, most functions place an error code in the global variable `errno` 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.

**USAGE**
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:

- Commands
- Modifiers
- Variables
- Expressions
- Input Grammar

**EXAMPLES**
This section provides examples of usage or of how to use a command or function. Wherever possible, a complete
example, which includes command-line entry and machine response, is shown. Whenever an example is given, the prompt is shown as example%, or if the user must be superuser, example#. Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS, and USAGE sections.

<table>
<thead>
<tr>
<th>ENVIRONMENT VARIABLES</th>
<th>This section lists any environment variables that the command or function affects, followed by a brief description of the effect.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXIT STATUS</td>
<td>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.</td>
</tr>
<tr>
<td>FILES</td>
<td>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.</td>
</tr>
<tr>
<td>ATTRIBUTES</td>
<td>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.</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>This section lists references to other man pages, in-house documentation, and outside publications.</td>
</tr>
<tr>
<td>DIAGNOSTICS</td>
<td>This section lists diagnostic messages with a brief explanation of the condition that caused the error.</td>
</tr>
<tr>
<td>WARNINGS</td>
<td>This section lists warnings about special conditions that could seriously affect your working conditions. WARNINGS is not a list of diagnostics.</td>
</tr>
<tr>
<td>NOTES</td>
<td>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.</td>
</tr>
<tr>
<td>BUGS</td>
<td>This section describes known bugs and, wherever possible, suggests workarounds.</td>
</tr>
</tbody>
</table>
REFERENCE

OSC33DS 4
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 resource type.
- **Oracle Real Application Clusters (RAC) server resources.** These resources are instances of the SUNW.scalable_rac_server 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:

```plaintext
{
[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.

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

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:

**NONE**

Specifies that the server fault monitor ignores the error.

**STOP**

Specifies that the server fault monitor is stopped.

**RESTART**

Specifies an action that depends on the type of resource for which the fault monitor that is being customized:

- **HA Oracle server resource**. Specifies that the server fault monitor stops and restarts the entity that is specified by the value of the `Restart_type` extension property of the `SUNW.oracle_server` resource.

- **Oracle RAC server resource**. Specifies that the server fault monitor stops and restarts the Oracle RAC server resource.

**SWITCH**

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 RAC server fault monitor. For the Oracle 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:

* **co** - Specifies that the entry applies only if the server fault monitor is attempting to connect to 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:

* **co** - Specifies that the server fault monitor must disconnect from the database and reconnect immediately to 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

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)

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;  
ACTION=NONE;  
CONNECTION_STATE=*;  
NEW_STATE=*;  
MESSAGE="Timeout #5 has occurred.";
}
**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)

* Oracle Solaris Cluster Data Service for Oracle Guide
REFERENCE

OSC33DS 5
ORCL.obiee_cluster_controller

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

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

The HA-Oracle BIEE Cluster Controller resource is configured in a failover resource group when creating an Oracle BIEE installation that consists of more than one BIEE 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 BIEE Cluster Controller resource by using the crresource(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**: Anytime

Retry_count

- **Default**: 2
- **Tunable**: Anytime

Retry_interval

- **Default**: 1330
- **Tunable**: Anytime

Thorough_probe_interval

- **Default**: 30
- **Tunable**: Anytime

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*
ORCL.obiee_presentation_service

<table>
<thead>
<tr>
<th>Name</th>
<th>ORCL.obiee_presentation_service, obiee_presentation_service – resource type implementation for HA-Oracle Business Intelligence Enterprise Edition (Oracle BI EE) Presentation Service</th>
</tr>
</thead>
</table>
| 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), clresource_type(1CL), clresource(1CL)

Oracle Solaris Cluster Data Services Planning and Administration Guide
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
**ORCL.obiee_server(5)**

<table>
<thead>
<tr>
<th>Name</th>
<th>ORCL.obiee_server, obiee_server – resource type implementation for HA Oracle Business Intelligence Enterprise Edition (Oracle BIEE) Server</th>
</tr>
</thead>
</table>
| 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 BIEE Cluster Controller is not used.  
|               | ■ In a multi-master resource group when the BIEE 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.  

---

Name ORCL.obiee_server, obiee_server – resource type implementation for HA Oracle Business Intelligence Enterprise Edition (Oracle BIEE) 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 BIEE Cluster Controller is not used.
- In a multi-master resource group when the BIEE 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
ORCL.ohs, ohs – resource type implementation for failover Oracle HTTP Server (OHS)

**Description**
The ORCL.ohs resource type represents one of the HA for Oracle Web Tier components available to an Oracle Solaris Cluster configuration. The HA for Oracle HTTP Server resource can be configured only as a failover service.

The ORCL.ohs resource type is derived from, and extends, the SUNW.gds resource type. As a consequence the Probe\_command, Start\_command, Stop\_command, and Validate\_command properties are not tunable. However, all the SUNW.gds standard and extension resource type properties can be tuned. For a list of the SUNW.gds standard and extension properties, see the SUNW.gds(5) man page.

The ORCL.ohs resource type defines the following extension properties, in addition to the SUNW.gds extension properties that the resource type uses.

**Extension Properties**

<table>
<thead>
<tr>
<th>Component_instance</th>
<th>The name of the Oracle HTTP Server instance that is listed in the opmnctl output.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
<td>Required (if not specified, the default value is used)</td>
</tr>
<tr>
<td><strong>Per Node</strong></td>
<td>False</td>
</tr>
<tr>
<td><strong>Data Type</strong></td>
<td>String</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>ohs1</td>
</tr>
<tr>
<td><strong>Tunable</strong></td>
<td>When disabled</td>
</tr>
</tbody>
</table>

**Debug\_level**
The Debug\_level property determines how much debugging information is produced during resource creation, update and during probe cycles.

| **Category**     | Optional                                                                   |
| **Per Node**     | True                                                                      |
| **Data Type**    | Integer                                                                   |
| **Minimum**      | 0                                                                         |
| **Maximum**      | 2                                                                         |
| **Default**      | 0                                                                         |
| **Tunable**      | Any time                                                                  |

**Attributes**
See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>ha-cluster/data-service/ohs</td>
</tr>
</tbody>
</table>
See Also `pmfadm(1M) scha_resource_get(3HA), clresource_type(1CL), attributes(5), ORCL.opmn(5), r_properties(5), SUNW.gds(5)`

*Oracle Solaris Cluster Data Services Planning and Administration Guide, Oracle Solaris Cluster Data Service for Oracle Web Tier Guide*
ORCL.opmn, opmn – resource type implementation for failover Oracle Process Management and Notification Server (OPMN)

The ORCL.opmn resource type represents one of the HA for Oracle Web Tier components available to an Oracle Solaris Cluster configuration. The HA for Oracle HTTP Server resource can be configured only as a failover service.

The ORCL.opmn resource type is derived from, and extends, the SUNW.gds resource type. As a consequence the Probe_command, Start_command, Stop_command, and Validate_command properties are not tunable. However, all the SUNW.gds standard and extension resource type properties can be tuned. For a list of the SUNW.gds standard and extension properties, see the SUNW.gds(5) man page.

The ORCL.opmn resource type defines the following extension properties, in addition to the SUNW.gds extension properties that the resource type uses.

**Extension Properties**

**Debug_level**

The Debug_level property determines how much debugging information is produced during resource creation, updates, and probe cycles.

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per Node</td>
<td>True</td>
</tr>
<tr>
<td>Data Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Minimum</td>
<td>0</td>
</tr>
<tr>
<td>Maximum</td>
<td>2</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
</tbody>
</table>

**Instance_name**

The instance name. A directory of this name must exist within the ORACLE_HOME/instances subdirectory.

<table>
<thead>
<tr>
<th>Category</th>
<th>Required (if not specified, the default value is used)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per Node</td>
<td>False</td>
</tr>
<tr>
<td>Data Type</td>
<td>String</td>
</tr>
<tr>
<td>Default</td>
<td>instance1</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

**Oracle_home**

The absolute path of the ORACLE_HOME of the web tier component of Oracle Fusion Middleware.

<table>
<thead>
<tr>
<th>Category</th>
<th>Required</th>
</tr>
</thead>
</table>

Per Node: False
Data Type: String
Default: Null
Tunable: When disabled

Attributes: See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>ha-cluster/data-service/opmn</td>
</tr>
</tbody>
</table>

See Also pmfadm(1M), scha_resource_get(3HA), cl_resource(1CL), cl_resourcetype(1CL), attributes(5), ORCL.ohs(5), r_properties(5), SUNW.gds(5)

Oracle Solaris Cluster Data Services Planning and Administration Guide, Oracle Solaris Cluster Data Service for Oracle Web Tier Guide
Name: ORCL.oracle_external_proxy, oracle_external_proxy – resource type implementation for the Oracle External Proxy managed by Oracle Solaris Cluster

Description: The ORCL.oracle_external_proxy resource type interrogates the Oracle Database and the Oracle Real Application Clusters (RAC) services and interprets the availability of those services as an Oracle Solaris Cluster resource state or status in an Oracle Solaris Cluster configuration.

Note: The ORCL.oracle_external_proxy resource type can be used within a multiple master resource group. A single resource of this type can run concurrently on multiple nodes, but does not use network load balancing.

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

- To register this resource type, use the clresource command.
- To create instances of this resource type, use the clresource 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:

<table>
<thead>
<tr>
<th>Property</th>
<th>Minimum</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init_timeout</td>
<td>60</td>
<td>300</td>
</tr>
<tr>
<td>Fini_timeout</td>
<td>60</td>
<td>300</td>
</tr>
<tr>
<td>Prenet_Start_timeout</td>
<td>60</td>
<td>300</td>
</tr>
<tr>
<td>Postnet_Stop_timeout</td>
<td>60</td>
<td>300</td>
</tr>
<tr>
<td>Validate_timeout</td>
<td>60</td>
<td>300</td>
</tr>
</tbody>
</table>

Name: ORCL.oracle_external_proxy(5)
The extension properties of the ORCL.oracle_external_proxy resource type are as follows.

**Debug level**
This property indicates the level to which debug messages for the ORCL.oracle_external_proxy resources are logged. When the debug level is increased, more debug messages are written to the terminal, the console, and 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 and End messages

For messages to appear in the system log:
1. Edit the /etc/syslog.conf file and make sure that the debug is set.
   ```
   *.err;kern.debug;daemon.debug;mail.crit /var/adm/messages
   ```
2. Disable the system log.
   ```
   # svcadm disable system-log
   ```
3. Enable the system log.
   ```
   # svcadm enable system-log
   ```

**Dbuser**
This property specifies the server-side Oracle Database user that the proxy monitor uses to connect to the database.

- **Data Type**: String
- **Default**: hauser
- **Tunable**: When disabled

**Ons_nodes**
This property specifies the Remote Oracle Notification Server (ONS) nodes such as node:port, node:port that the proxy monitor uses to connect to the database.

- **Data Type**: String
- **Default**: None
- **Tunable**: When disabled
Plugin_name
This property specifies the plugin module name that the proxy monitor uses to connect to the database.

Data Type String
Default OracleExternalProxy
Tunable When disabled

Service_name
This property specifies the Oracle Database and Oracle RAC service name that the proxy monitor uses to connect to the database.

Data Type String
Default None
Tunable When disabled

Tns_admin
This property specifies the client-side location for the Oracle Tns_admin path that the proxy monitor uses to connect to the database.

Data Type String
Default /var/opt/oracle
Tunable When disabled

Examples

**EXAMPLE 1** Creating an ORCL.oracle_external_proxy Resource

This example shows the commands to create a scalable multi-master ORCL.oracle_external_proxy resource on a two-node cluster. These commands perform the following operations:

- Creating the oep-rs resource group
- Registering the ORCL.oracle_external_proxy resource type
- Adding the oep-rs resource to the oep-rs resource group

This example assumes that default values are being used for the dbuser, the tns_admin, and the plugin_name extension properties. It also assumes that the dbuser password has been encrypted on each cluster node.

```
phys-schost-1# clresourcetype register ORCL.oracle_external_proxy
phys-schost-1# clresourcegroup create -S oep-rs
phys-schost-1# clresource create -g oep-rs \
-t ORCL.oracle_external_proxy \
-p service_name=orcl \
-p ons_nodes=binks-scan-lh:6200 \
-d oep-rs
phys-schost-1# clresourcegroup online -M oep-rs
```
EXAMPLE 1 Creating an ORCL.oracle_external_proxy Resource (Continued)

phys-schost-1# clresource enable oep-rs

Attributes See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>ha-cluster/data-service/oracle-external-proxy</td>
</tr>
</tbody>
</table>

See Also clresource(1CL), clresource_type(1CL), clsetup(1CL), attributes(5), r_properties(5)
ORCL.sapcentr resource type implementation for processing central services of Oracle Solaris Cluster HA for SAP NetWeaver

**Description**
The ORCL.sapcentr resource type represents the SAP central services component in an Oracle Solaris Cluster configuration. This resource type is part of a set of resource types for the SAP NetWeaver platform. The other resource types are:

- The SAP replicated enqueue server, represented by the ORCL.saprepenq resource type.
- The SAP sapstartsrv component, represented by the ORCL.sapstartsrv resource type.
- The SAP dialogue instance, represented by the ORCL.sapdia resource type.
- The SAP replicated enqueue preempter, represented by the ORCL.saprepenq_preempt resource type.

Create the following dependencies when you configure the Oracle Solaris Cluster HA for SAP NetWeaver data service.

- The ORCL.sapcentr resource type must be dependent on the ORCL.sapstartsrv resource in the same resource group. If an ORCL.saprepenq_preempt resource is configured, it must be dependent on the ORCL.sapcentr resource.
- The ORCL.saprepenq resources and the ORCL.sapcentr resources must not be placed in the same resource group as they do not fail over together.
- The resource group weak positive affinities must ensure that the SAP central service resource group fails over to the node where the SAP replicated enqueue resource group is online. If an ORCL.saprepenq_preempt resource is not configured, it must be implemented by strong negative affinities such that the replicated enqueue server resource group is off-loaded from the failover target node before the SAP central service resource group is started.

For more information, see Oracle Solaris Cluster Data Service for SAP NetWeaver Guide.

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

**Standard Properties**
None of the standard resources properties have been overridden for this resource type.

For a description of all standard resource properties, see the r_properties(5) man page.

**Extension Properties**
The extension properties of the ORCL.sapcentr resource type are as follows:

**Debug_level**
This property indicates the level to which debug messages for the SAP NetWeaver 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 and 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  
**Default**  0  
**Range**  0–2  
**Tunable**  Any time

**Sap_user**  
This property indicates the administrative user for an SAP NetWeaver installation.

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

**Sid**  
This property indicates the SAP NetWeaver System Identifier (SID). This is SAPSYSTEMNAME in the SAP profile.

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

**Instance_name**  
This property indicates the name of the SAP central service component instance. This is INSTANCE_NAME in the SAP profile.

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

**Instance_number**  
This property indicates the two-digit SAP system number for the SAP central service component instance. This is SAPSYSTEM in the SAP profile.

**Data type**  Number  
**Default**  None defined
Range Not applicable
Tunable When disabled

Host
This property indicates the host alias on which the central services are configured.

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

Start_script
This property indicates the full path to the start script for the instance.

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

Stop_script
This property indicates the full path to the stop script for the instance.

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

Yellow
This property indicates the SAP NetWeaver probe return value for the central services YELLOW status.

Data type Number
Default 10
Range 1–50
Tunable Any time

Examples

**EXAMPLE 1** Creating an ORCL.sapcentr Resource

This example shows the commands to create an ORCL.sapcentr resource. These commands perform the following operations:

- Registering ORCL.sapcentr
EXAMPLE 1 Creating an ORCL.sapcentr Resource (Continued)

- Adding the scs-rs resource to the scs-rg resource group

This example makes the following assumptions:
- The bash shell is used.
- The failover resource group scs-rg is already created.
- The logical host resource is already created.
- The database resource db-rs, the sapstartsrv resource scs-start-rs, and the storage resource hsp-rs are already created.

```bash
# clrt register ORCL.sapcentr
# clrs create -d -g scs-rg -t sapcentr \ 
  -p sid=QE3 \ 
  -p sap_user=qe3adm \ 
  -p instance_number=00 \ 
  -p instance_name=ASCS00 \ 
  -p HOST=bono-1 \ 
  -p yellow=20 \ 
  -p debug_level=0 \ 
  -p resource_dependencies_offline_restart=hsp-rs \ 
  -p resource_dependencies=db-rs,scs-start-rs \ 
  scs-rs
```

Attributes See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>ha-cluster/data-service/sapnetweaver</td>
</tr>
</tbody>
</table>

See Also clresource(1CL), clresourcegroup(1CL), clresource(1CL), attributes(5), r_properties(5), ORCL.sapdia(5), ORCL.saprenq(5), ORCL.saprenq_preempt(5), ORCL.sapstartsrv(5)
The ORCL.sapdia resource type represents the SAP dialogue component in an Oracle Solaris Cluster configuration. This resource type is part of a set of resource types for the SAP NetWeaver platform. The other resource types are:

- The SAP replicated enqueue server, represented by the ORCL.saprepenq resource type.
- The SAP sapstartsrv component, represented by the ORCL.sapstartsrv resource type.
- The SAP central services, represented by the ORCL.sapcentr resource type.
- The SAP replicated enqueue preempter, represented by the ORCL.saprepenq_preempt resource type.

The ORCL.sapdia resource type is used for the following SAP components:

- Primary application instances
- Additional dialogue instances
- Single instances that combine the primary application instance and the central services in a single deployment

Create the following dependencies when you configure the Oracle Solaris Cluster HA for SAP NetWeaver data service.

- The ORCL.sapdia resource type must be dependent on the ORCL.sapstartsrv resource in the same resource group.
- The ORCL.sapdia resource type must depend on the database resource.
- The ORCL.sapdia resource type supports failover and multiple master deployments.

For more information, see Oracle Solaris Cluster Data Service for SAP NetWeaver Guide.

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

None of the standard resources properties have been overridden for this resource type.

For a description of all standard resource properties, see the r_properties man page.

The extension properties of the ORCL.sapdia resource type are as follows:

Debug_level

This property indicates the level to which debug messages for the SAP NetWeaver 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
All debug messages and function Begin and End messages

Data type: Integer
Default: 0
Range: 0–2
Tunable: Any time

Sap_user
This property indicates the administrative user for an SAP NetWeaver installation.

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

Sid
This property indicates the SAP NetWeaver System Identifier (SID). This is SAPSYSTEMNAME in the SAP profile.

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

Instance_name
This property indicates the name of the SAP central service component instance. This is INSTANCE_NAME in the SAP profile.

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

Instance_number
This property indicates the two-digit SAP system number for the SAP central service component instance. This is SAPSYSTEM in the SAP profile.

Data type: Number
Default: None defined
Range: Not applicable
Tunable: When disabled
Host
This property indicates the host alias on which the central services are configured.

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

Start_script
This property indicates the full path to the start script for the instance.

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

Stop_script
This property indicates the full path to the stop script for the instance.

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

Yellow
This property indicates the SAP NetWeaver probe return value for the central services YELLOW status.

- **Data type**: Integer
- **Default**: 10
- **Range**: 1–50
- **Tunable**: Any time

Architecture
A flag to indicate if a dialogue instance is a combined instance or a normal dialogue instance. The valid values are space or comb.

- **Data type**: String
- **Default**: None defined
- **Range**: Not applicable
- **Tunable**: When disabled
**Examples**

**EXAMPLE 1**  Creating an ORCL.sapdia Resource for Failover Deployment

This example shows the commands to create an ORCL.sapdia resource for failover deployment. These commands perform the following operations:

- Registering ORCL.sapdia
- Adding the pas-rs resource to the pas-rg resource group

This example makes the following assumptions:

- The bash shell is used.
- The failover resource group pas-rg is already created.
- The logical host resource is already created.
- The database resource db-rs, the central service resource scs-rs, the sapstartsrv resource pas-start-rs, and the storage resource hsp-rs are already created.

```bash
# clrt register ORCL.sapdia
# /usr/cluster/bin/clrs create -d -g pas-rg -t sapdia \
  -p sid=QE3 \ 
  -p sap_user=qe3adm \ 
  -p instance_number=02 \ 
  -p instance_name=DVEBMGS02 \ 
  -p HOST=bono-4 \ 
  -p debug_level=0 \ 
  -p resource_dependencies_offlineRestart=hsp-rs \ 
  -p resource_dependencies=db-rs,scs-rs,pas-start-rs \ 
  pas-rs
```

**EXAMPLE 2**  Creating an ORCL.sapdia Resource for Multiple Master Deployment

This example shows the commands to create an ORCL.sapdia resource for multiple master deployment. These commands perform the following operations:

- Registering ORCL.sapdia
- Adding the dia-rs resource to the dia-rg resource group

This example makes the following assumptions:

- The bash shell is used.
- The multiple master resource group dia-rg is already created.
- The database resource db-rs, the central service resource scs-rs, the sapstartsrv resource dia-start-rs, and the storage resource hsp-rs are already created.

```bash
# clrt register ORCL.sapdia
# /usr/cluster/bin/clrs create -d -g dia-rg -t sapdia \
  -p sid=QE3 \ 
  -p sap_user=qe3adm \ 
  -p instance_number=03 \ 
```
EXAMPLE 2  Creating an ORCL.sapdia Resource for Multiple Master Deployment  (Continued)

-p instance_number(pbno2)=04 \n-p instance_name(pbno1)=D03 \n-p instance_name(pbno2)=D04 \n-p HOST(pbno1)=pbno1 \n-p HOST(pbno2)=pbno2 \n-p debug_level=0 \n-p resource_dependencies_offline_restart=hsp-rs \n-p resource_dependencies=db-rs,scs-rs,dia-strt-rs \n-dia-rs

Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>ha-cluster/data-service/sapnetweaver</td>
</tr>
</tbody>
</table>

See Also  clresource(1CL), clresourcegroup(1CL), clresourcetype(1CL), attributes(5), r_properties(5), ORCL.sapcentr(5), ORCL.saprepenq(5), ORCL.saprepenq_preempt(5), ORCL.sapstartsrv(5)
ORCL.saprepenq resource type represents the SAP replicated enqueue server component in an Oracle Solaris Cluster configuration. This resource type is part of a set of resource types for the SAP NetWeaver platform. The other resource types are:

- The SAP dialogue instance, represented by the ORCL.sapdia resource type.
- The SAP sapstartsrv component, represented by the ORCL.sapstartsrv resource type.
- The SAP central services, represented by the ORCL.sapcentr resource type.
- The SAP replicated enqueue preemt, represented by the ORCL.saprepenq_preemt resource type.

Create the following dependencies when you configure the Oracle Solaris Cluster HA for SAP NetWeaver data service.

- The ORCL.saprepenq resource type must be dependent on the ORCL.sapstartsrv resource in the same resource group.
- The ORCL.saprepenq resources and the ORCL.sapcentr resources must not be placed in the same resource group as they do not fail over together.
- The resource group weak positive affinities must ensure that the SAP central service resource group fails over to the node where the SAP replicated enqueue resource group is online. If an ORCL.saprepenq_preemt resource is not configured, it must be implemented strong negative by affinities such that the replicated enqueue server resource group is off-loaded from the failover target node before the SAP central service resource group is started.
- If the replicate enqueue server is running on a different instance number than the central services (which is the standard SAP installation), you must also configure a replicated enqueue preemt resource in the central service resource group.
- Resource dependencies must ensure that the replicated enqueue server resource depends on the central service resource.

For more information, see Oracle Solaris Cluster Data Service for SAP NetWeaver Guide.

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

None of the standard resources properties have been overridden for this resource type.

For a description of all standard resource properties, see the r_properties(5) man page.
The extension properties of this resource type are as follows:

**Debug_level**
This property indicates the level to which debug messages for the SAP NetWeaver 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 and End messages

**Data type**  Number
**Default**  0
**Range**  0–2
**Tunable**  Any time

**Sap_user**
This property indicates the administrative user for an SAP NetWeaver installation.

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

**Sid**
This property indicates the SAP NetWeaver System Identifier (SID). This is `SAPSYSTEMNAME` in the SAP profile.

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

**Instance_name**
This property indicates the name of the SAP central service component instance. This is `INSTANCE_NAME` in the SAP profile.

**Data type**  String
**Default**  None defined
**Range**  Not applicable
**Tunable**  When disabled
Instance_number
  This property indicates the two-digit SAP system number for the SAP central service component instance. This is SAPSYSTEM in the SAP profile.

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

Host
  This property indicates the host alias on which the central services are configured.

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

Start_script
  This property indicates the full path to the start script for the instance.

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

Stop_script
  This property indicates the full path to the stop script for the instance.

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

Yellow
  This property indicates the SAP NetWeaver probe return value for the central services YELLOW status.

  Data type  Number
  Default  10
  Range  1–50
  Tunable  Any time
EXAMPLE 1  Creating an ORCL.saprepenq Resource

This example shows the commands to create an ORCL.saprepenq resource. These commands perform the following operations:

- Registering ORCL.saprepenq
- Adding the rep-rs resource to the rep-rg resource group

This example makes the following assumptions:

- The bash shell is used.
- The failover resource group rep-rg is already created.
- The logical host resource is already created.
- The database resource db-rs, the central service resource scs-rs, the sapstartsrv resource rep-start-rs, and the storage resource hsp-rs are already created.

```
# clrt register ORCL.saprepenq
#/usr/cluster/bin/clrs create -d -g rep-rg -t saprepenq \
  -p sid=QE3 \n  -p sap_user=qe3adm \n  -p instance_number=10 \n  -p instance_name=ERS10 \n  -p HOST=bono-3 \n  -p debug_level=0 \n  -p resource_dependencies_offline_restart=hsp-rs \n  -p resource_dependencies=scs-rs,rep-start-rs \n  rep-rs
```

Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>ha-cluster/data-service/sapnetweaver</td>
</tr>
</tbody>
</table>

See Also  clresource(1CL), clresourcegroup(1CL), clresourcetype(1CL), attributes(5), r_properties(5), ORCL.sapcentr(5), ORCL.sapdia(5), ORCL.saprepenq_preempt(5), ORCL.sapstartsrv(5)
ORCL.saprepenq_preempt, saprepenq_preempt – resource type implementation for processing the replicated enqueue server preemt component of Oracle Solaris Cluster HA for SAP NetWeaver

**Description**
The ORCL.saprepenq_preempt resource type represents the SAP replicated enqueue server preemt component in an Oracle Solaris Cluster configuration. This resource type is part of a set of resource types for the SAP Netweaver platform. The other resource types are:

- The SAP replicated enqueue server, represented by the ORCL.saprepenq resource type.
- The SAP sapstartsrv component, represented by the ORCL.sapstartsrv resource type.
- The SAP dialogue instance, represented by the ORCL.sapdia resource type.
- The SAP central services, represented by the ORCL.sapcent resource type.

Create the following dependencies when you configure the Oracle Solaris Cluster HA for SAP NetWeaver data service.

- The ORCL.saprepenq_preempt resource type must be offline restart dependent on one or more ORCL.sapcent resources in the same resource group.
- The ORCL.saprepenq_preempt resource must be configured if the central services and the replicated enqueue server are running on different instance numbers. In all other cases, it must not be configured.
- The ORCL.saprepenq_preempt resources and the ORCL.sapcent resources must be placed in the same resource group as they fail over together.
- The resource group weak positive affinities must ensure that the SAP central service resource group fails over to the node where the SAP replicated enqueue resource group is online. If an ORCL.saprepenq_preempt resource is not configured, it must be implemented by strong negative affinities such that the replicated enqueue server resource group is off-loaded from the failover target node before the SAP central service resource group is started. If the replicated enqueue preemt resource is configured, it is the task of this resource to off-load the replicated enqueue server resource group to a spare node after the enqueue tables are copied.

- If the replicated enqueue server is running on a different instance number than the central services (which is the standard SAP installation), you must also configure a replicated enqueue preemt resource in the central service resource group.

For more information, see *Oracle Solaris Cluster Data Service for SAP NetWeaver Guide*.

Standard properties and extension properties that are defined for the ORCL.saprepenq_preempt resource type are described in the subsections that follow. To set these properties for an instance of the ORCL.saprepenq_preempt resource type, use the clresourceype(1CL) command.
None of the standard resources properties have been overridden for this resource type. For a description of all standard resource properties, see the `r_properties`(5) man page.

The extension properties of this resource type are as follows:

**Debug_level**
This property indicates the level to which debug messages for the SAP NetWeaver 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 and End messages

**Data type** Number
**Default** 0
**Range** 0–2
**Tunable** Any time

**Sap_user**
This property indicates the administrative user for an SAP NetWeaver installation.

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

**Sid**
This property indicates the SAP NetWeaver System Identifier (SID). This is `SAPSYSTEMNAME` in the SAP profile.

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

**Repqres**
This property indicates the name of the SAP replicated enqueue server resource name.

**Data type** String
**Default** None defined
**Range** Not applicable
**Enq_instnr**
This property indicates a list of two-digit SAP system numbers for the SAP central service components in this resource group. This instance number is SAPSYSTEM in the SAP profile.

- **Data type**: String array
- **Default**: None defined
- **Range**: Not applicable

**Timeout_return**
This property indicated the return code when the probe method reaches 80 percent of the probe timeout.

- **Data type**: Number
- **Default**: 10
- **Range**: Not applicable

**Examples**

**EXAMPLE 1** Creating an ORCL.saprepenq_preempt Resource

This example shows the commands to create an ORCL.saprepenq_preempt resource. These commands perform the following operations:

- Registering ORCL.saprepenq_preempt
- Adding the preempt-rs resource to the scs-rg resource group

This example makes the following assumptions:

- The bash shell is used.
- The failover resource groups scs-rg and rep-rg are already created.
- The logical host resource is already created.
- The database resource db-rs, the central service resource scs-rs, the replicated enqueue resource rep-rs, and the storage resource hsp-rs are already created.

```bash
# clrt register ORCL.saprepenq_preempt
# /usr/cluster/bin/clrs create -d -g scs-rg -t saprepenq_preempt\ 
  -p sid=QE3 \ 
  -p sap_user=qe3adm \ 
  -p reponqres=rep-rs \ 
  -p enq_instnr=00 \ 
  -p debug_level=0 \ 
  -p resource_dependencies_offline_restart=scs-rs,hsp-rs \ 
  -p resource_dependencies=db-rs \ 
```
EXAMPLE 1  Creating an ORCL.saprepenq_preempt Resource  (Continued)

preempt-rs

Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>ha-cluster/data-service/sapnetweaver</td>
</tr>
</tbody>
</table>

See Also  clresource(1CL), clresourcegroup(1CL), clresourcetype(1CL), attributes(5), r_properties(5), ORCL.sapcentr(5), ORCL.sapdia(5), ORCL.saprepenq(5), ORCL.sapstartsrv(5)
**Name**  
ORCL.sapstartsrv, sapstartsrv – resource type implementation for processing sapstartsrv of Oracle Solaris Cluster HA for SAP NetWeaver

**Description**  
The ORCL.sapstartsrv resource represents the SAP component for the sapstartsrv daemon in an Oracle Solaris Cluster configuration. This resource type is part of a set of resource types for the SAP NetWeaver platform. The other resource types are:
- The SAP replicated enqueue server, represented by the ORCL.saprepenq resource type.
- The SAP dialogue instance, represented by the ORCL.sapdia resource type.
- The SAP central services, represented by the ORCL.sapcentr resource type.
- The SAP replicated enqueue pre empter, represented by the ORCL.saprepenq_preempt resource type.

The ORCL.sapstartsrv resource starts, stops, and monitors the sapstartsrv process of an instance. Since the sapstartsrv process controls the instance, it is extremely important to make it highly available. All instance resources must have a strong dependency on the sapstartsrv resource. The sapstartsrv resource supports failover as well as multiple master deployments.

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

None of the standard resources properties have been overridden for this resource type.

For a description of all standard resource properties, see the r_properties(5) man page.

The extension properties of this resource type are as follows:

- **Debug_level**
  This property indicates the level to which debug messages for the SAP NetWeaver 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 and End messages
  **Data type** Number
  **Default** 0
  **Range** 0–2
  **Tunable** Any time

- **Sap_user**
  This property indicates the administrative user for an SAP NetWeaver installation.
Data type  String
Default    None defined
Range      Not applicable
Tunable    When disabled

Sid
This property indicates the SAP NetWeaver System Identifier (SID). This is SAPSYSTEMNAME in the SAP profile.

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

Instance_name
This property indicates the name of the SAP central service component instance. This is INSTANCE_NAME in the SAP profile.

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

Instance_number
This property indicates the two-digit SAP system number for the SAP central service component instance. This is SAPSYSTEM in the SAP profile.

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

Timeout_return
This property indicates the return code when the probe method of the sapstartsrv process reaches 80 percent of the probe timeout.

Data type  Number
Default    10
Range      Not applicable
Tunable    Anytime
Host
This property indicates the host alias on which the central services are configured.

**Data type**  String

**Default**  None

**Range**  1–50

**Tunable**  When disabled

**Examples**

**EXAMPLE 1  Creating an ORCL.sapstartsrv Resource for Failover Deployment**

This example shows the commands to create an `ORCL.sapstartsrv` resource for failover deployment. These commands perform the following operations:

- Registering `ORCL.sapstartsrv`
- Adding the `scs-strt-rs` resource to the `scs-rg` resource group

This example makes the following assumptions:
- The bash shell is used.
- The logical host resource is already created.
- The failover resource group `scs-rg` and the storage resource `hsp-rs` are already created.

```
# clrt register ORCL.sapstartsrv
#/usr/cluster/bin/clrs create -d -g scs-rg -t sapstartsrv -p sid=QE3 -p sap_user=qe3adm -p instance_number=00 -p instance_name=ASCS00 -p HOST=bono-1 -p timeout_return=20 -p debug_level=0 -p resource_dependencies_offline_restart=hsp-rs -p resource_dependencies_online_start= -p resource_dependencies_online_stop=scs-strt-rs
```

**EXAMPLE 2  Creating an ORCL.sapstartsrv Resource for Multiple Master Deployment**

This example shows the commands to create an `ORCL.sapstartsrv` resource for multiple master deployment. These commands perform the following operations:

- Registering `ORCL.sapstartsrv`
- Adding the `dia strt rs` resource to the `dia rg` resource group

This example makes the following assumptions:
- The bash shell is used.
- The multiple master resource group `dia rg` is already created.
- The storage resource `hsp-rs` is already created.
EXAMPLE 2  Creating an ORCL.sapstartsrv Resource for Multiple Master Deployment  (Continued)

```bash
# clrt register ORCL.sapstartsrv
#/usr/cluster/bin/clrs create -d -g dia-rg -t sapstartsrv \
  -p sid=QE3 \
  -p sap_user=qe3adm \
  -p instance_number{pbono1}=03 \
  -p instance_number{pbono2}=04 \
  -p instance_name{pbono1}=D03 \
  -p instance_name{pbono2}=D04 \
  -p timeout_return=20 \
  -p debug_level=0 \
  -p resource_dependencies_offline_restart=hsp-rs \
  dia-strt-rs
```

Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>ha-cluster/data-service/sapnetweaver</td>
</tr>
</tbody>
</table>

See Also  clresource(1CL), cresourcegroup(1CL), cresource(1CL), attributes(5), r_properties(5), ORCL.sapcentr(5), ORCL.sapdia(5), ORCL.saprepenq(5), ORCL.saprepenq_preempt(5)
SUNW.apache, apache – resource type implementation for failover and scalable Apache Web Server

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.

If you set the Scalable resource property to TRUE, you can use other Standard properties, such as Scalable, Network_resources_used, Resource_dependencies, Port_list, Conn_threshold, Round_robin, Load_balancing_policy, and Load_balancing_weights for 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 r_properties for a complete description of the following resource properties.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNW.apache</td>
<td>resource type implementation for failover and scalable Apache Web Server</td>
</tr>
<tr>
<td>Description</td>
<td>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).</td>
</tr>
</tbody>
</table>

You must set the following properties on an Apache resource by using clresource:

If you set the Scalable resource property to TRUE, you can use other Standard properties, such as Scalable, Network_resources_used, Resource_dependencies, Port_list, Conn_threshold, Round_robin, Load_balancing_policy, and Load_balancing_weights for 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 r_properties for a complete description of the following resource properties.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conn_threshold (integer)</td>
<td>Maximum number of active connections or clients supported when Round_robin load distribution is enabled. TCP connections are considered active if the connection endpoint remains alive on the server node. UDP sessions are considered active if there is traffic flow within the UDP session active timeout window setting (see the udp_session_timeout cluster property).</td>
</tr>
<tr>
<td>Category</td>
<td>Optional</td>
</tr>
<tr>
<td>Default</td>
<td>10000</td>
</tr>
<tr>
<td>Tunable</td>
<td>WHEN_DISABLED</td>
</tr>
<tr>
<td>Load_balancing_policy</td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td>LB_WEIGHTED</td>
</tr>
<tr>
<td>Tunable</td>
<td>At creation</td>
</tr>
<tr>
<td>Load_balancing_weights</td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td>NULL</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
</tbody>
</table>

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** Anytime

**Port_list**

**Default** 80/tcp

**Tunable** At creation

**Retry_count**

**Minimum** 0

**Maximum** 10

**Default** 2

**Tunable** Anytime

**Retry_interval**

**Minimum** 0

**Maximum** 3600

**Default** 620
**Tunable** Any time

**Round_robin (boolean)**
Assigns incoming requests to specific server nodes in a round-robin fashion taking into account the relative load_balancing_weights value assigned to each node. Requests are assigned on a connection basis for resources with a non-sticky load_balancing_policy setting; otherwise, requests are assigned on a per-client IP address basis.

Round_Robin should be enabled for resources that require deterministic load distribution of incoming requests where a small number of connections or clients are expected.

A resource property, Conn_threshold, and a cluster property, udp_session_timeout, support the Round Robin scheme, and may optionally be configured by the user if the Round_robin resource property is set for a service.

No existing resource type registration (RTR) files need to be upgraded to use the Round_robin property.

**Category** Optional

**Default** FALSE

**Tunable** WHEN_DISABLED

**Thorough_probe_interval**

**Minimum** 0

**Maximum** 3600

**Default** 60

**Tunable** Any time

**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 `cresource(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` 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 "InternalServerError" 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 2  Creating a Scalable Apache Resource  (Continued)

eexample# clresourcetype 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.)

eexample# 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 [ "$host" = "button-1.example.com" \
    -a "$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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscapc</td>
</tr>
</tbody>
</table>

See Also

pmadm(1M), scha_resource_get(3HA), clresource_type(1CL), clresource_group(1CL), attributes(5), r_properties(5), scalable_service(5)

*Oracle Solaris Cluster Data Service for Apache Guide, Oracle Solaris Cluster Data Services Planning and Administration Guide*
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 an 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. For more information on configuring resources for Oracle database instances, see *Oracle Solaris Cluster Data Service for Oracle Guide* or *Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide*.

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

- Oracle Solaris Cluster Manager
- 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.

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**: ""
- **Tunable**: Any time

### Port_list
- **Default**: ""
- **Tunable**: Any time

### Retry_count
- **Maximum**: 10
Tunable Anytime

Retry_INTERVAL

Maximum 3600
Default 370
Tunable Anytime

Scalable

Default FALSE
Tunable At creation

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
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Default</td>
<td>Tunable When disabled</td>
</tr>
<tr>
<td>Child_mon_level (integer)</td>
<td>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 -C option for <code>pmfadm(1M)</code> All child processes and their descendents are monitored.</td>
</tr>
<tr>
<td>Category</td>
<td>Optional</td>
</tr>
<tr>
<td>Default</td>
<td>-1</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
<tr>
<td>debug_level</td>
<td>Note – All SQL*Plus and srvmgr messages that the ASM disk group resource issues are written to the log file <code>/var/opt/SUNWscor/oracle_asm/message_log.${RESOURCE}</code>. 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 <code>/var/adm/messages</code> as follows:</td>
</tr>
<tr>
<td>Data Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Range</td>
<td>0–2</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
<tr>
<td>Failover_Enabled (boolean)</td>
<td>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.</td>
</tr>
</tbody>
</table>
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.

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 | "/opt/SUNWscor/oracle_asm/bin/asm_control probe -R %RS_NAME -G %RG_NAME -T %RT_NAME"
**Tunable**

**Probe_timeout(integer)**
This property specifies the timeout value, in seconds, for the probe command.

- **Category**: Optional
- **Default**: 30 seconds
- **Tunable**: Anytime

**Start_command(string)**
Specifies the command that mounts the ASM diskgroup.

- **Category**: Required
- **Default**: 
  
  
  ```
  /opt/SUNWscor/oracle_asm/bin/asm_control start
  -R %RS_NAME -G %RG_NAME -T %RT_NAME
  ```

- **Tunable**: None

**Stop_command(string)**
Specifies the command that dismounts the ASM diskgroup.

- **Category**: Required
- **Default**: 
  
  ```
  /opt/SUNWscor/oracle_asm/bin/asm_control stop
  -R %RS_NAME -G %RG_NAME -T %RT_NAME
  ```

- **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# cresource type register SUNW.asm_diskgroup
phys-schost-1# cresourcegroup create asm-dg-rg
phys-schost-1# cresourcegroup set -p rg_affinities=++asm-inst-rg asm-dg-rg
phys-schost-1# cresource 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# cresourcegroup online -M asm-dg-rg
phys-schost-1# cresource 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-rs exists and contains a resource of type SUNW.ScalMountPoint that is named qfs-mp-rs.

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

OSC33DS5 67
EXAMPLE 2  Creating a Scalable asm_diskgroup Resource  (Continued)

phys-schost-1# cresource 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# cresource enable asm-dg-rs

See Also  **Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide**, **Oracle Solaris Cluster Data Service for Oracle Guide**, **Oracle Solaris Cluster Data Services Planning and Administration Guide**
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.

**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**  Anytime

**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 | SUNWscdns

**See Also**
- in.named(1M), pmfadm(1M), scha_resource_get(3HA), clresourcectype(1CL), clresourcegroup(1CL), named.conf(4), attributes(5), r_properties(5)

*Oracle Solaris Cluster Data Service for Domain Name Service (DNS) Guide*
SUNW.hadb-hadb – resource type implementation for Sun Java System Application Server EE (HADB) that is mastered on multiple nodes simultaneously

**Description**
The SUNW.hadb resource type represents the Sun Java System Application Server EE (HADB) component in an Oracle Solaris Cluster configuration.

You must set the following properties on a SUNW.hadb resource by using `clresource(1CL)`.

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

- **Thorough_probe_interval**
  - **Default**: 180
  - **Maximum**: 3600
  - **Tunable**: Any time

- **Stop_timeout**
  - **Default**: 180
  - **Minimum**: 120
  - **Tunable**: Any time

- **Start_timeout**
  - **Default**: 600
  - **Minimum**: 60
  - **Tunable**: Any time

**Extension Properties**
- **Confdir_list**
  - Type string array. This property is a path name set to the Sun Java System Application Server EE (HADB) configuration directory for the HADB database. For example, `/etc/opt/SUNWhadb/dbdef/hadb`. This property is tunable when the resource is disabled.

- **DB_Name**
  - Type string. No default. This property contains the name of the HADB database. This property is tunable when the resource is disabled.

- **HADB_ROOT**
  - Type string. No default. This property is a path name set to the Sun Java System Application Server EE (HADB) application binary location. For example, `/opt/SUNWappserver7/SUNWhadb/4`. This property is tunable when the resource is disabled.

- **Auto_recovery**
  - Type boolean. Default is FALSE. If the Auto_recovery extension property is set to TRUE and the agent is unable to start the database, the HADB data service attempts to recover the database by reinitializing it. The data service reinitializes the database by running the `hadbm`
clear -- fast command, and then running the command specified by the
 Auto_recovery_command extension property. This property is tunable at any time.

Auto_recovery_command
 Type string. Default is ". This property specifies the command to be run when the HADB
data service recovers the database by reinitializing the database. The HADB data service
runs the command after clearing the database. This property is tunable at any time.

DB_password_file
 Type string. Default is ". This property specifies the file that contains the password to be
used for the system user of the database. The hadbm commands that require a password will
have the DB_password_file extension property passed as the value of the hadbm
--dbpasswordfile command. This argument is needed if the Auto_recovery extension
property is set to TRUE because the hadbm --clear command requires a password. This
property is tunable at any time.

Examples
 The following examples show the creation of a resource group for a Sun Java System
Application Server EE (HADB) resource and the creation of a resource in this resource group.
The examples that show the creation of a Sun Java System Application Server EE (HADB)
resource illustrate alternative configurations of the Sun Java System Application Server EE
(HADB) resource.

The examples assume that the SUNWschadb package is already installed.

EXAMPLE 1 Creating a Resource Group for a Sun Java System Application Server EE (HADB) Resource
This example shows the creation of a resource group for a Sun Java System Application Server
EE (HADB) resource on a six-node cluster. Sun Java System Application Server EE (HADB) is
mastered on all nodes in the cluster. The resource group is named hadb-rg.

# clresourcegroup create \\n-p maximum_primaries=6 -p desired_primaries=6 hadb-rg

EXAMPLE 2 Creating a Sun Java System Application Server EE (HADB) Resource Without Auto_recovery
This example shows the creation of a Sun Java System Application Server EE (HADB)
resource without Auto_recovery that is named hadb-rs. This resource is created in an
existing resource group that is named hadb-rg. The creation of the hadb-rg resource group is
shown in the example for the creation of a resource group for a Sun Java System Application
Server EE (HADB) resource.

# clresource create -g hadb-rg -t SUNW.hadb \\
-p confdir_list=/etc/opt/SUNWhadb/dbdef/hadb \\
-p hadb_root=/opt/SUNWappserver7/SUNWhadb/4 \\
-p db_name=hadb hadb-rs

EXAMPLE 3 Creating a Sun Java System Application Server EE (HADB) Resource With Auto_recovery
This example shows the creation of a Sun Java System Application Server EE (HADB)
resource with Auto_recovery that is named hadb-rs. This resource is created in an existing
resource group that is named hadb-rs. The creation of the hadb-rs resource group is shown in the example for the creation of a resource group for a Sun Java System Application Server EE (HADB) resource.

```
# clresource create -g hadb-rg -t SUNW.hadb \
-p confdir_list=/etc/opt/SUNWhadb/dbdef/hadb \
-p hadb_root=/opt/SUNWappserver7/SUNWhadb/4 -p db_name=hadb \
-p auto_recovery=true \
-p auto_recovery_command=/usr/local/etc/create-session-store \
-p db_password_file=/usr/local/etc/hadb-password-file hadb-rs
```

**Attributes**  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWschadb</td>
</tr>
</tbody>
</table>

**See Also**  clresourcetype(1CL), clresourcegroup(1CL), scha_resource_get(3HA), pmfadm(1M), attributes(5), r_properties(5), scalable_service(5)

*Oracle Solaris Cluster Data Service for Sun Java System Application Server EE (HADB), Oracle Solaris Cluster Data Services Planning and Administration Guide*
**SUNW.hadb_ma**

**Name**  
SUNW.hadb_ma, hadb_ma – resource type implementation for Sun Java System Application Server EE (HADB) (hadb_ma)

**Description**  
The SUNW.hadb_ma resource type represents the Sun Java System Application Server EE (HADB version 4.4) application in an Oracle Solaris Cluster configuration.

Standard properties and extension properties that are defined for the SUNW.hadb_ma resource type are described in the subsections that follow. To set these properties for an instance of the SUNW.hadb_ma 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
Retry_count

  Maximum 10
  Default 2
  Tunable Any time

Retry_interval

  Maximum 3600
  Default 970
  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:

HADB_MA_CFG
  The full path to the configuration file that is used to start the HADB Management Agent Server.

  Data type  String
  Default   /etc/opt/SUNWhadb/mgt.cfg
  Range     Not applicable
  Tunable   When disabled

HADB_MA_START
  The full path to the script that is used to start and stop the HADB Management Agent Server. This script must be able to start and stop the MA Server without any input from the user. Any configuration parameters must be specified in the file that is indicated by the extension property HADB_MA_CFG.

  Data type  String
  Default   /etc/init.d/ma-initd
  Range     Not applicable
  Tunable   When disabled

HADB_MA_USER
  The user name of the user who starts the HADB Management Agent Server.

  Data type  String
DATA

HADB_ROOT
The complete path to the HADB installation directory. This directory contains the
directory structure bin/, which contains the files ma and hadbm.

Data type String
Default /opt/SUNWhadb/4
Range Not applicable
Tunable When disabled

HADBM_PASSWORDFILE
The complete path to the file that contains the HADBM administrative password. This
property must be set if the HADB management domain is created with an administrative
password.

Data type String
Default Null
Range Not applicable
Tunable Anytime

Examples
EXAMPLE 1 Creating a Resource for SUNW.hadb_ma
For this example to work, the Oracle Solaris Cluster HA for Sun Java System Application
Server EE (HADB) data service must first be installed. This data service includes all the
packages to make Sun Java System Application Server EE (HADB) highly available.

This example creates an HADB MA resource mastered on multiple nodes. The resource is
created in an existing resource group, which is assumed to be mastered on multiple nodes. To
create this resource the following commands are run:

# clresourcetype register SUNW.hadb_ma
# clresource create -g hadb-ma-rg -t SUNW.hadb_ma hadb-ma-rs

In this example, the resource group is named hadb-ma-rg and the resource is named
hadb-ma-rs. The default values are used for the extension properties.

Attributes
See attributes(5) for descriptions of the following attributes:
See Also  clresource(1CL), clresourcegroup(1CL), scha_resource_get(3HA), pmfadm(1M),
attributes(5), r_properties(5), scalable_service(5)

Oracle Solaris Cluster Data Service for Sun Java System Application Server EE (HADB) Guide
and Oracle Solaris Cluster Data Services Planning and Administration Guide
SUNW.iws(5)

**Name**  SUNW.iws, iws – resource type implementation for failover and scalable Oracle iPlanet Web Server

**Description**  The SUNW.iws resource type represents the Oracle iPlanet Web Server application in an Oracle Solaris Cluster configuration.

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

If you set the Scalable resource property to TRUE, you can use other Standard properties, such as Scalable, Network_resources_used, Resource_dependencies, Port_list, Conn_threshold, Round_robin, Load_balancing_policy, and Load_balancing_weights for scalable resource types.

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 Scalable property is set at resource creation time to indicate the mode in which the service operates. The default is FALSE (failover mode).

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

**Conn_threshold (integer)**

Maximum number of active connections or clients supported when Round_robin load distribution is enabled. TCP connections are considered active if the connection endpoint remains alive on the server node. UDP sessions are considered active if there is traffic flow within the UDP session active timeout window setting (see the `udp_session_timeout` cluster property).

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>10000</td>
</tr>
<tr>
<td>Tunable</td>
<td>WHEN_DISABLED</td>
</tr>
</tbody>
</table>

**Load_balancing_policy**

<table>
<thead>
<tr>
<th>Default</th>
<th>LB_WEIGHTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunable</td>
<td>At creation</td>
</tr>
</tbody>
</table>

**Load_balancing_weights**

<table>
<thead>
<tr>
<th>Default</th>
<th>NULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunable</td>
<td>Anytime</td>
</tr>
</tbody>
</table>

**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

**Port_list**

**Default**  80/tcp  
**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**  Anytime

**Retry_count**

**Minimum**  0  
**Maximum**  10  
**Default**  2  
**Tunable**  Any time

**Retry_interval**

**Minimum**  0  
**Maximum**  3600  
**Default**  620
Tunable Any time

**Round_robin (boolean)**
Assigns incoming requests to specific server nodes in a round-robin fashion taking into account the relative load_balancing_weights value assigned to each node. Requests are assigned on a connection basis for resources with a non-sticky load_balancing_policy setting; otherwise, requests are assigned on a per-client IP address basis.

Round_Robin should be enabled for resources that require deterministic load distribution of incoming requests where a small number of connections or clients are expected.

A resource property, Conn_threshold, and a cluster property, udp_session_timeout, support the Round Robin scheme, and may optionally be configured by the user if the Round_robin resource property is set for a service.

No existing resource type registration (RTR) files need to be upgraded to use the Round_robin property.

Category Optional
Default FALSE
Tunable WHEN_DISABLED

**Thorough_probe_interval**

Minimum 0
Maximum 3600
Default 60
Tunable Any time

**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 SSL-based 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 c1resource(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 "". 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 Oracle Solaris Cluster HA for Oracle iPlanet 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 Oracle iPlanet Web Server 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.

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# clresourcetype register SUNW.iws
example# clresource create -g web-rg -t SUNW.iws
     -p Confdir_list=/global/iws/https-web webserver-failover
```

**EXAMPLE 2** Creating a Scalable Oracle iPlanet Web Server 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# clresourcetype register SUNW.iws
example# clresourcegroup create \
     -p Maximum_primaries=4 -p Desired_primaries=4 \
```
EXAMPLE 2    Creating a Scalable Oracle iPlanet Web Server Resource  (Continued)

-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 Oracle iPlanet Web Server 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.

eexample# clresourcetype register SUNW.iws
eexample# 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 Oracle iPlanet Web Server Resource That Contains Two Web Server Instances
For this example to work, the data service must first be installed. This example creates a scalable Oracle iPlanet Web Server resource named webserver-paired, which contains two iWS instances, one SSL-based and the other non-SSL based. 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.

eexample# clresourcetype register SUNW.iws
eexample# 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 Scalable Oracle iPlanet Web Server Instance
The following example shows how to set the Monitor_uri_list extension property when a scalable Oracle iPlanet Web Server instance is added to a configuration. The Monitor_uri_list extension property is not supported with an SSL-based Oracle iPlanet Web Server instance.

(Add a non-SSL Oracle iPlanet instance with default load balancing.)
EXAMPLE 5  Setting Monitor_uri_list for Scalable Oracle iPlanet Web Server Instance
(Continued)

eexample# clresource create -g resource-group-1 -t SUNW.iws \
-p Confdir_list=/opt/oracle/webserver \
-p Monitor_Uri_list=http://schost-1:8000/servlet/monitor \
-p Scalable=True -y Network_resources_used=schost-1 \
-p Port_list=8000/tcp insecure-1

EXAMPLE 6  Setting Monitor_uri_list for Failover Oracle iPlanet Web Server Instance

The following example shows how to set the Monitor_uri_list extension property when a
failover Oracle iPlanet Web Server instance is added to a configuration. The
Monitor_uri_list extension property is not supported with an SSL-based Oracle iPlanet
Web Server instance.

(Add a non-SSL Oracle iPlanet application resource instance.)

eexample# clresource create -g resource-group-1 -t SUNW.iws \
-p Confdir_list=/opt/oracle/webserver \
-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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWschtt</td>
</tr>
</tbody>
</table>

See Also  clresourcetype(1CL), clresourcegroup(1CL), sha_resource_get(1HA), pmfadm(1M),
aattributes(5), r_properties(5), scalable_service(5)

Oracle Solaris Cluster Data Services Planning and Administration Guide, Oracle Solaris Cluster
Data Service for Oracle iPlanet Web Server Guide
### SUNW.jsas(5)

**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.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network_resources_used</strong></td>
<td>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>. This property is updated automatically by the RGM, based on the setting of the <code>resource-dependencies</code> properties. You do not set this property directly. Instead, use the <code>Resource_dependencies</code> property.</td>
</tr>
<tr>
<td><strong>Resource_dependencies</strong></td>
<td>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 <code>clreslogicalhostname(1CL)</code> and <code>clressharedaddress(1CL)</code> man pages for more information. You can specify an alternate kind of dependency by using the <code>Resource_dependencies_weak</code>, <code>Resource_dependencies_restart</code>, or <code>Resource_dependencies_offline_restart</code> property instead of the <code>Resource_dependencies</code> property. For more information, see the <code>r_properties(5)</code> man page.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Conditional/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default</strong></td>
<td>The empty list</td>
</tr>
<tr>
<td><strong>Tunable</strong></td>
<td>When disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default</strong></td>
<td>The empty list</td>
</tr>
<tr>
<td><strong>Tunable</strong></td>
<td>Any time</td>
</tr>
</tbody>
</table>
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 \n   -p Domain_name=new-domain \n   -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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscjsas</td>
</tr>
</tbody>
</table>

See Also  clresourcetype(1CL), clresourcegroup(1CL), scha_resource_get(3HA), pnmadm(1M), attributes(5), r_properties(5), scalable_service(5)

*Oracle Solaris Cluster Data Service for Sun Java System Application Server Guide* and *Oracle Solaris Cluster Data Services Planning and Administration Guide*
**SUNW.jsas-na**

**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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Network_resources_used</code></td>
<td>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>. 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.</td>
</tr>
<tr>
<td>Category</td>
<td>Conditional/Optional</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>The empty list</td>
</tr>
<tr>
<td><strong>Tunable</strong></td>
<td>When disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Resource_dependencies</code></td>
<td>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 <code>clreslogicalhostname(1CL)</code> and <code>clressharedaddress(1CL)</code> man pages for more information. You can specify an alternate kind of dependency by using the <code>Resource_dependencies_weak</code>, <code>Resource_dependencies_restart</code>, or <code>Resource_dependencies_offline_restart</code> property instead of the <code>Resource_dependencies</code> property. For more information, see the <code>r_properties(5)</code> man page.</td>
</tr>
<tr>
<td>Category</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>The empty list</td>
</tr>
<tr>
<td><strong>Tunable</strong></td>
<td>Anytime</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Port_list</code></td>
<td>Null string</td>
</tr>
<tr>
<td>Category</td>
<td></td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td></td>
</tr>
</tbody>
</table>
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  

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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscjsas</td>
</tr>
</tbody>
</table>

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

Oracle Solaris Cluster Data Service for Sun Java System Application Server Guide and 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 SUNWsckrb5/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:

<table>
<thead>
<tr>
<th>Property</th>
<th>Override</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheap_probe_interval</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>3600 seconds</td>
</tr>
<tr>
<td>Minimum</td>
<td>10 seconds</td>
</tr>
<tr>
<td>Default</td>
<td>30 seconds</td>
</tr>
<tr>
<td>Tunable</td>
<td>Anytime</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>The empty list</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

**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 `cleslogicalhostname(1CL)` and `clessharedaddress(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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>The empty list</td>
</tr>
</tbody>
</table>
**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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNW.krb5</td>
</tr>
</tbody>
</table>

**See Also**  kinit(1), kadmin(1M), kadmind(1M), krb5kdc(1M), scrgadm(1M), svcadm(1M), r_properties(5)

_Oracle Solaris Cluster Data Service for Kerberos Guide_
SUNW.ldom, ldom – resource type implementation for failover guest domains.

Description
The HA for Oracle VM Server for SPARC 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 logical domain resource by using clresource(1CL) man page.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource_dependencies</td>
<td>Specifies a list of resources upon which a resource depends. This list includes any HAStoragePlus 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.</td>
</tr>
<tr>
<td>Category</td>
<td>Optional</td>
</tr>
<tr>
<td>Default</td>
<td>The empty list</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
</tbody>
</table>

Failover_mode (enum)

<table>
<thead>
<tr>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOFT</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

Retry_count (integer)

<table>
<thead>
<tr>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

Retry_interval (integer)

<table>
<thead>
<tr>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td>370</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

Thorough_probe_interval (integer)

<table>
<thead>
<tr>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>Any time</td>
</tr>
</tbody>
</table>
Domain name (string)
Indicates the name of the 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 for Oracle VM Server 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 HA for Oracle VM Server data service. You can specify None, Info, or Err for this property. When you specify None, diagnostic messages are not logged by the 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 HA for Oracle VM
Server 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 a logical
domain instance. You can modify the value for this property at any time.

Default     30
Tunable      Any time

Examples
EXAMPLE 1 Creating a Failover Oracle VM Server for SPARC Guest-Domain Resource
The following example creates a failover 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 guest-domain resource uses the plugin_probe script to
check the status of the LDomS guest domain.
EXAMPLE 1 Creating a Failover Oracle VM Server for SPARC Guest-Domain Resource

(Continued)

eexample# clresourcetype register SUNW.ldom
eexample# 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 HA for Oracle VM Server guest domain data service before performing the steps in the example.

Attributes

See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscdns</td>
</tr>
</tbody>
</table>

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
SUNW.nfs(5)

**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 exactly the same as that described in the dfstab(4) man page. 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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cheap Probe Interval</strong></td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>10</td>
</tr>
<tr>
<td>Maximum</td>
<td>3600</td>
</tr>
<tr>
<td>Default</td>
<td>20</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Conditional/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default</strong></td>
<td>The empty list</td>
</tr>
<tr>
<td><strong>Tunable</strong></td>
<td>When disabled</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
</table>
The empty list

Any time

Minimum 60

Maximum 3600

Default 120

Any time

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

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.

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.

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

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

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

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

Type Boolean. Default is FALSE. Indicates if the system is to be rebooted when a null rpc call on rpcbind fails.
**Rpcbind_nullrpc_timeout**
Type integer. Default is 120. Minimum value is 60. This property indicates the timeout value (in seconds) to use when probing rpcbind.

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

**Files**

<table>
<thead>
<tr>
<th>resource-name</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>/tmp/.hanfs/*</td>
<td>Critical state files used by the implementation.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNW nfs</td>
</tr>
</tbody>
</table>
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.
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. For more information, see Oracle Solaris Cluster Data Service for Oracle Guide.

You must set the following properties for an Oracle listener resource by using cresource(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.

```bash
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(IM), scha_resource_get(1HA), clresourcetype(1CL), clresource(1CL), r_properties(5), SUNW.oracle_server(5)

*Oracle Solaris Cluster Data Service for Oracle Guide, Oracle Solaris Cluster Data Services Planning and Administration Guide*
Name: SUNW.oracle_rac_server, oracle_rac_server – legacy resource type implementation for the Oracle Real Application Clusters (RAC) server managed by Oracle Solaris Cluster

Description: Note – This resource type is not required for OSC 3.3 configurations. This resource type is provided only to ensure that earlier configurations of Oracle Solaris Cluster Support for Oracle Real Application Clusters continue to function after an upgrade to OSC 3.3. In OSC 3.3 configurations, use the SUNW.oracle_rac_server resource type.

The SUNW.oracle_rac_server resource type represents the Oracle RAC server in a Sun Cluster 3.2 configuration. Each instance of the Oracle RAC server is represented by a single SUNW.oracle_rac_server resource, and is uniquely identified by its Oracle_Sid property.

Each resource belongs to a failover resource group. The resource group is restricted to run on only one node. Therefore, the Oracle RAC server resource in the resource group is also restricted to run on only one node. This restriction is enforced by specifying a single node in the node list when the resource group is created.

The resource group manager (RGM) performs only automated startup and shutdown of Oracle RAC server instances. The RGM does not restart or fail over Oracle RAC server instances. The RGM monitors Oracle RAC server resources only to enable the status of these resources to be obtained by using Oracle Solaris Cluster utilities.

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 Oracle RAC server resources and the RAC framework as follows:

- Create a strong positive affinity between the Oracle RAC server resource groups and the RAC framework resource group.
- Create a dependency between the Oracle RAC server resource and the RAC framework resource.

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

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

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

- **Failover_mode**
  - Default: NONE
  - Tunable: Any time.

- **Thorough_probe_interval**
  - Minimum: 1
  - Maximum: 2592000
  - Default: 30
Tunable: Any time.

For a description of standard resource properties, see the `r_properties(5)` man page.

**Auto_End_Bkp**

Type Boolean; defaults to `False`. This property specifies whether the Oracle RAC server resource automatically recovers the database if an Oracle relational database management system (RDBMS) hot backup is interrupted. When a hot backup is interrupted, the database fails to open because of files that remain in hot backup mode. During the startup of the Oracle RAC server resource, the resource tests for the interruption of a hot backup by testing for an occurrence the following RDBMS error:

`ORA-01113 file file needs media recovery`

To recover the database automatically, the Oracle RAC server resource performs the following actions:

- Releasing all files that remain in hot backup mode. The `sys.v$backup` view indicates which files remain in hot backup mode.
- Opening the database for use.

The permitted values for this property are as follows:

- `False` Specifies that the Oracle RAC server resource does not automatically recover the database. If a hot backup is interrupted, you must recover the database manually. In this situation, the status of the Oracle RAC server resource is set to `FAULTED`. The default value of this property is `False`.
- `True` Specifies that the Oracle RAC server resource automatically recovers the database.

You can modify this property at any time.

**Debug_level**

Type integer; defaults to 1, which logs `syslog` messages. Maximum value is 100. 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. You can modify this property at any time.

**Oracle_Home**

Type string. This property is set to the path to the Oracle home directory. The Oracle home directory contains the binary files, log files, and parameter files for the Oracle software. You can modify this property only when the resource is disabled.

**Oracle_Sid**

Type string. This property is set to the Oracle system identifier. This identifier is the name of the Oracle database instance. 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 NULL. When this property is NULL, the default Oracle mechanism is used to locate the parameter file. You can modify this property at any time.

User_env
Type string; defaults to NULL. This property is set to the name of the file that contains the environment variables to be set before database startup or shutdown. All environment variables that have values that differ from Oracle defaults must be defined in this file.

For example, a user's listener.ora file might not reside under the /var/opt/oracle directory or the $ORACLE_HOME/network/admin directory. In this situation, the TNS_ADMIN environment variable should 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 modify this property at any time.

Wait_for_online
Type Boolean; defaults to True. 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. The default value of this property is True.

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.

You can modify this property at any time.

Examples
EXAMPLE 1 Creating oracle_rac_server Resources and Resource Groups

The following example creates two oracle_rac_server resources and two oracle_rac_server resource groups on a two-node cluster. One oracle_rac_server resource group is created for each oracle_rac_server resource. Each resource and its corresponding resource group are named as shown in the following table.

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource Group Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAC1-rs</td>
<td>RAC1-rs</td>
</tr>
<tr>
<td>RAC2-rs</td>
<td>RAC2-rs</td>
</tr>
</tbody>
</table>
The example assumes that a SUNW.rac_framework resource group named rac-framework-rg has been created. The example also assumes that the constituent resources of this resource group have been created. These resources are instances of resource types as shown in the following table.

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNW.rac_framework</td>
<td>rac_framework</td>
</tr>
<tr>
<td>SUNW.rac_udlm</td>
<td>rac_udlm</td>
</tr>
<tr>
<td>SUNW.rac_svm</td>
<td>rac_svm</td>
</tr>
</tbody>
</table>

```
example# scrgadm -a -t SUNW.oracle_rac_server
example# scrgadm -a -g RAC1-rg -h node1 \       
   -y RG_AFFINITIES=+rac-framework-rg
example# scrgadm -a -g RAC2-rg -h node2 \       
   -y RG_AFFINITIES=+rac-framework-rg
example# scrgadm -a -j RAC1-rs -g RAC1-rg \     
   -t SUNW.oracle_rac_server \                  
   -y RESOURCE_DEPENDENCIES=rac_framework \    
   -x ORACLE_SID=RAC1 \                         
   -x ORACLE_HOME=/oracle
example# scrgadm -a -j RAC2-rs -g RAC2-rg \     
   -t SUNW.oracle_rac_server \                  
   -y RESOURCE_DEPENDENCIES=rac_framework \    
   -x ORACLE_SID=RAC2 \                         
   -x ORACLE_HOME=/oracle
```

The preceding commands create two oracle_rac_server resources and two oracle_rac_server resource groups on a two-node cluster by performing the following operations:

1. Registering the SUNW.oracle_rac_server resource type
2. Creating the RAC1-rg resource group for node node1
3. Creating the RAC2-rg resource group for node node2
4. Creating the RAC1-rs resource in the RAC1-rg resource group for node node1
5. Creating the RAC2-rs resource in the RAC2-rg resource group for node node2

**Attributes** See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscor</td>
</tr>
</tbody>
</table>
See Also  scrgadm(1M), scswitch(1M), attributes(5), SUNW.oracle_listener(5), SUNW.rac_framework(5), SUNW.rac_svm(5), SUNW.rac_udlm(5), SUNW.scalable_rac_server(5)

Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide, Oracle Solaris Cluster Data Services Planning and Administration Guide
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. For more information, see *Oracle Solaris Cluster Reference Manual*.

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

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

The extension property `Alert_log_file` is a 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.

Preset actions for logged alerts are listed in Appendix B, “Preset Actions for DBMS Errors and Logged Alerts,” in Oracle Solaris Cluster Data Service for Oracle Guide. To change the action that the server fault monitor performs, customize the server fault monitor as explained in “Customizing the HA for Oracle Server Fault Monitor” in Oracle Solaris Cluster Data Service for Oracle Guide.

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

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

STANDBY
Specifications 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
Specifications 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:

- **RESOURCE_RESTART**: Specifies that only this resource is restarted.
- **RESOURCE_GROUP_RESTART**: 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:

- **LOGICAL**: Specifies a logical standby database.
PHYSICAL

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, Oracle Solaris Cluster Data Service for Oracle Guide*
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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network_resources_used (string array)</td>
<td>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.</td>
</tr>
<tr>
<td>Category</td>
<td>Conditional/Optional</td>
</tr>
<tr>
<td>Default</td>
<td>The empty list</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
<tr>
<td>Resource_dependencies (string array)</td>
<td>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</td>
</tr>
</tbody>
</table>
property instead of the Resource_dependencies property. For more information, see the $\text{r\_properties}(5)$ man page.

**Category** Optional

**Default** The empty list

**Tunable** Anytime

**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

**Confdir_list**

Type string array. This property is a path name set to $\text{install\_dir/domains/domain/server}$ 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 URLs 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.)

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

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

```bash
# 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.

```bash
eample# scrgadm -a -t SUNW.s1as
eample# 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 s1as in a Scalable Resource Group  (Continued)

of /global/s1as/domains/domain1/server1. The URI
http://localhost:8000/servlets/testservlet is specified to the extension property
Monitor_uri_list.

example# scrgadm -a -g s1as-rg
-y Maximum_primaries=3 -y Desired_primaries=3
example# scrgadm -a -g s1as-rg -j scalable-app-server1 -t SUNW.s1as
-x Confdir_list=/global/s1as/domains/domain1/server1
-x Monitor_uri_list=http://localhost:8000/servlets/testservlet

Attributes
See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscs1as</td>
<td></td>
</tr>
</tbody>
</table>

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

Oracle Solaris Cluster Data Service for Sun Java System Application Server Guide, and Oracle Solaris Cluster Data Services Planning and Administration Guide
The SUNW.s1mq resource type represents the Sun Glassfish Message Queue application in an Oracle Solaris Cluster configuration. This application was formerly known as Sun Java System Message Queue.

You must set the following properties on an SUNW.s1mq resource by using `clresource(1CL)`. See `r_properties(5)` for a complete description of the following resource properties.

### Standard Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Category</th>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network_resources_used</td>
<td>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.</td>
<td>Conditional/Optional</td>
<td>The empty list</td>
<td>When disabled</td>
</tr>
<tr>
<td>Resource_dependencies</td>
<td>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.</td>
<td>Optional</td>
<td>The empty list</td>
<td>Anytime</td>
</tr>
<tr>
<td>Port_list</td>
<td></td>
<td>No default</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tunable At creation

Retry_count

Maximum 10
Default 2
Tunable Any time

Retry_interval

Maximum 3600
Default 740
Default Any time

Thorough_probe_interval

Maximum 3600
Default 60
Tunable Any time

Extension Properties

Bin_dir
Type string. This property indicates the location of Message Queue server binaries for Sun Glassfish Message Queue versions beginning with 4.4. You must specify this property at resource creation time.

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 s1mq instance directory. You must specify this property at resource creation time.

Broker_Name
Type string. No default. This property contains the name of the broker to start and monitor. The imqcmd command needs this name to stop the broker if Smooth_Shutdown is set to TRUE.

Broker_User
Type string. Default is "". This property contains the Message Queue user name of the managed broker. This user name is used to shut down the broker if Smooth_Shutdown is set to TRUE. Smooth_Shutdown defaults to FALSE. If Smooth_Shutdown=FALSE, the broker is sent SIGTERM to shut it down. If Smooth_Shutdown is set to TRUE the broker will be shut down by using imqcmd. Using imqcmd exposes the broker user password on the imqcmd command line.

Probe_timeout
Type string. Default is 120 seconds. Minimum value is 2. This property is tunable at anytime and sets the timeout value for the probe.
Examples

**EXAMPLE 1**  
Creating a Failover s1mq Resource in an Existing Group

For this example to work, the SUNWscs1mq data service package must first be installed. This example creates a failover s1mq resource named message-queue-failover in an existing resource group named mq-rg. The mq-rg resource group contains a LogicalHostname resource, which identifies the logical hostname associated with the resource group. This example includes the Bin_dir extension property used by Message Queue beginning in version 4.4.

```
example# clresourcetype register SUNW.s1mq
example# clresource create -g mq-rg -t SUNW.s1mq
    -p Confdir_list=/global/s1mq/instances/hamq1
    \-p Bin_dir=/opt/mq/bin \n    -p Network_Resources_used=logical host \n    -p Port_List=7676\tcp \n    -p Broker_Name=hamq1 message-queue-failover
```

In the preceding example, the s1mq resource created is named message-queue-failover. The s1mq resource listens on port 7676, with a corresponding s1mq instance in the directory /global/s1mq/instances/hamq1.

**EXAMPLE 2**  
Creating a Failover s1mq Resource with Smooth_Shutdown=TRUE

For this example to work, the SUNWscs1mq data service package must be first installed. This example creates a failover s1mq resource named message-queue-failover, which listens on port 7676 in an existing resource group named mq-rg.

```
example# clresourcetype register SUNW.s1mq
example# clresource create -g mq-rg -t SUNW.s1mq
    -p Confdir_list=/global/s1mq/instances/hamq1
    -p Network_Resources_used=logical host \n    -p Port_List=7676 -p Broker_Name=hamq1 -p Broker_User=admin \n    -p Smooth_Shutdown=TRUE message-queue-failover
```

Attributes

See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscs1mq</td>
</tr>
</tbody>
</table>

See Also

clresourcetype(1CL), clresourcegroup(1CL), scha_resource_get(3HA), pmfadm(1M), attributes(5), r_properties(5), scalable_service(5)

Oracle Solaris Cluster Data Service for Sun Java System Message Queue Guide, and Oracle Solaris Cluster Data Services Planning and Administration Guide
SUNW.sap_as(5)

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 SAPIID. 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:

<table>
<thead>
<tr>
<th>ATtributETYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsap</td>
</tr>
</tbody>
</table>

See Also  pmfadm(1M), scha_resource_get(1HA), clresource(1CL), clresource(1CL), attributes(5), SUNW.sap_ci(5)

Oracle Solaris Cluster Data Services Planning and Administration Guide
### SUNW.sap_ci(5)

<table>
<thead>
<tr>
<th>Name</th>
<th>SUNW.sap_ci, sap_ci, SUNW.sap_ci_v2, sap_ci_v2 – resource type implementations for Oracle Solaris Cluster HA for SAP central instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>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 <code>clresource_type (1CL)</code> command or a resource configuration GUI to set the following properties on an SAP central-instance resource.</td>
</tr>
<tr>
<td>Standard Properties</td>
<td>See <code>r_properties(5)</code> for a complete description of the following resource properties.</td>
</tr>
<tr>
<td>Failover_mode</td>
<td>Default: SOFT Tunable: Any time</td>
</tr>
<tr>
<td>Retry_count</td>
<td>Default: 3 Tunable: Any time</td>
</tr>
<tr>
<td>Retry_interval</td>
<td>Default: 3600 Tunable: Any time</td>
</tr>
<tr>
<td>Thorough_probe_interval</td>
<td>Default: 60 Tunable: Any time</td>
</tr>
<tr>
<td>Extension Properties</td>
<td>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.</td>
</tr>
<tr>
<td>Ci_instance_id</td>
<td>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.</td>
</tr>
<tr>
<td>Ci_services_string</td>
<td>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.</td>
</tr>
<tr>
<td>Message_server_name</td>
<td>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.</td>
</tr>
<tr>
<td>Monitor_retry_count</td>
<td>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 <code>-n</code> option passed to the <code>pmfadm(1M)</code> command. The RGM</td>
</tr>
</tbody>
</table>
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 SAP `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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsap</td>
</tr>
</tbody>
</table>

**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 xservers system. Therefore, you must set a dependency between the MaxDB resource group and the SAP xservers resource group. Create this dependency when you register and configure the Oracle Solaris Cluster HA for MaxDB data service. For more information, see “Registering and Configuring HA for MaxDB” in *Oracle Solaris Cluster Data Service for MaxDB Guide*.

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 `clresourceType(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
**DB_Name**

The name of the MaxDB database instance in uppercase. This name is created when MaxDB is installed and configured as explained in “Installing and Configuring MaxDB” in Oracle Solaris Cluster Data Service for MaxDB Guide.

**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. For more information, see “Installing and Configuring MaxDB” in Oracle Solaris Cluster Data Service for MaxDB Guide.

**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 as explained in “Installing and Configuring MaxDB” in Oracle Solaris Cluster Data Service for MaxDB Guide.
  
  **Data type**  String  
  **Default**  No default defined  
  **Range**  Not applicable  
  **Tunable**  When disabled
Examples  

**EXAMPLE 1**  Creating a SUNW.sapdb Resource

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

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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsapdb</td>
</tr>
</tbody>
</table>

**See Also**  
clresourcegroup(1CL), clresourcetype(1CL), clresource(1CL), attributes(5), r_properties(5), SUNW.sap_xserver(5)

*Oracle Solaris Cluster Data Service for MaxDB Guide*
SUNW.sapenq(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. saprep1 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. For more information, see Oracle Solaris Cluster Data Service for SAP Web Application Server Guide.

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 clresourceType(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
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.
**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**: 2
- **Range**: No range 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 \\
```
EXAMPLE 1  Creating Resources for SUNW.sapenq, SUNW.sapscs, and SUNW.saprepl (Continued)

-N sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3,sc_ipmp0@4 central-lh \  
central-lh-rs

To bring online in a managed state the SAP central services resource group, the following command is run:

# clresourcegroup online -M 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 in a managed state the SAP replica server resource group, the following command is run:

# clresourcegroup online -M 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.sapscs
# clresourcetype register SUNW.saprepl
EXAMPLE 1  Creating Resources for SUNW.sapenq, SUNW.sapscs, and SUNW.saprepl (Continued)

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 \n-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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsapenq</td>
</tr>
</tbody>
</table>

See Also  clresource(1CL), clresourcegroup(1CL), clresourcetype(1CL), attributes(5), r_properties(5), SUNW.saprepl(5), SUNW.sapscs(5), SUNW.sapwebas(5)
The SAP liveCache data service is managed by the Oracle Solaris Cluster Resource Group Manager (RGM) and is configured as a LogicalHostname resource, an 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 an SAP liveCache database resource using the clresource(1CL) command.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNW.sap_livecache</td>
<td>SUNW.sap_livecache, sap_livecache – resource type implementation for failover SAP liveCache database</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Standard Properties</th>
<th>Maximum: 10</th>
<th>Default: 2</th>
<th>Tunable: Any time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retry_count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retry_interval</td>
<td>Maximum: 3600</td>
<td>Default: 620</td>
<td>Tunable: Any time</td>
</tr>
<tr>
<td>Thorough_probe_interval</td>
<td>Maximum: 3600</td>
<td>Default: 60</td>
<td>Tunable: Any time</td>
</tr>
</tbody>
</table>

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

See r_properties(5) for a description of the following resource properties.
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 an 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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNW:slc</td>
</tr>
</tbody>
</table>

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

Oracle Solaris Cluster Data Services Developer’s Guide
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.sapmsg 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. For more information, see Oracle Solaris Cluster Data Service for SAP Web Application Server Guide.

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 clresourcetype(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
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**  Anytime

---

**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 in a managed state the SAP central services resource group, the following command is run:

```
# clresourcegroup online -M 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 in a managed state the SAP replica server resource group, the following command is run:

```
# clresourcegroup online -M 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:
EXAMPLE 1  Creating Resources for SUNW.sapenq, SUNW.sapscs, and SUNW.saprepl  

(Continued)

# 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.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 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.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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNW.scsaprepl</td>
</tr>
</tbody>
</table>

See Also  clresource(1CL), clresourcegroup(1CL), clresourcetype(1CL), attributes(5), r_properties(5), SUNW.sapenq(5), SUNW.sapscs(5), SUNW.sapwebas(5)
SUNW.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.saprep1` 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. For more information, see Oracle Solaris Cluster Data Service for SAP Web Application Server Guide.

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 + SAP_Instance_Number. If the listen port of the SAP message server to be probed is not equivalent to 3600 + 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 start-up script for the instance.

- **Data type**: String
- **Default**: /usr/sap/<SAP_SID>/SYS/exe/run/startsap
- **Range**: Not applicable
- **Tunable**: When disabled
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 in a managed state the SAP central services resource group, the following command is run:

```
# clresourcegroup online -M 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 in a managed state the SAP replica server resource group, the following command is run:

```
# clresourcegroup online -M 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
Creating Resources for SUNW.sapenq, SUNW.sapscs, and SUNW.saprepl

EXAMPLE 1

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

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.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/enrepsserver \
-p SAP_User=sc3adm \
-p Resource_Dependencies=enq-rs repl-rs
```

**Attributes**

See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsapscs</td>
</tr>
</tbody>
</table>

**See Also**

clresource(1CL), clresourcegroup(1CL), clresourcetype(1CL), attributes(5), r_properties(5), SUNW.sapenq(5), SUNW.saprepl(5), SUNW.sapwebas(5)
SUNW.sapwebas, sapwebas – resource type implementation for the SAP web application server component of Oracle Solaris Cluster HA for SAP Web Application Server

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.saprep1 resource type), and the SAP message server (represented by the SUNW.sapmsgcs 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. For more information, see Oracle Solaris Cluster Data Service for SAP Web Application Server Guide.

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 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  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).
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
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**
Determine if the start-up script process tree is run under Process Monitor Facility (PMF).
The possible values of this extension property are as follows:

- **True** - Specifies that the start-up script process tree is run under PMF.
- **False** - Specifies that the start-up 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 in a managed state the failover resource group for the SAP web application server component, the following command is run:

```
# clresourcegroup online -M 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_primates=4 
  -p Desired_primitives=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 in a managed state the scalable resource group for the SAP web application server component, the following command is run:

```
# clresourcegroup online -M 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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsapwebas</td>
</tr>
</tbody>
</table>

See Also  clresource(1CL), clresourcegroup(1CL), clresourcetype(1CL), attributes(5), r_properties(5), SUNW.sapenq(5), SUNW.sapscs(5), SUNW.saprepl(5)
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 PFM 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 username. 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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscle</td>
</tr>
</tbody>
</table>

**See Also**  pmfadm(1M), scha_resource_get(1HA), clresourcetype(1CL), clresource(1CL), clresourcegroup(1CL), attributes(5), r_properties(5)

*Oracle Solaris Cluster Data Service for MaxDB Guide,*
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.

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

**Standard 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 4, the fault monitor is restarted.
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.

Siebel_version
Type string. Default is 8.2 and this property is tunable when disabled. This property is set to the Siebel server version.

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 \ 
-p siebel_version=8.2 sblgtwy-rs
```

In this example, /global/siebel/gtwy is the Siebel gateway root directory.

Attributes
See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsbl</td>
</tr>
</tbody>
</table>

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
### 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.
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.

Siebel_version
Type string. Default is 8.2 and this property is tunable when disabled. This property is set
to the Siebel server version.

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 \
-p siebel_version=8.2 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:

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsbl</td>
</tr>
</tbody>
</table>

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_acfs_proxy, scalable_acfs_proxy – resource type implementation for an Oracle Automatic Storage Management Cluster File System (Oracle ACFS) that is managed by Oracle Solaris Cluster

**Description**  
The SUNW.scalable_acfs_proxy resource type represents the Oracle ACFS file system in an Oracle Solaris Cluster configuration. This resource type is introduced for use starting with Oracle 11g release 2 configurations.

**Note**  
Use the SUNW.scalable_acfs_proxy resource type only if you are using Oracle Grid Infrastructure for a cluster.

The SUNW.scalable_acfs_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_acfs_proxy resource represents an Oracle ACFS file system. Each Oracle ACFS file system is uniquely identified by the value of the acfs_mountpoint extension property on the node where the instance is running. Only mount the Oracle ACFS file system if the Oracle ASM disk group is mounted on the same cluster node as the file system. To ensure that these requirements are met, configure the Oracle ACFS file system proxy resource as follows:

- Create a strong positive affinity between the Oracle ACFS proxy resource group and the Oracle ASM disk-group proxy resource group.
- Create an offline-restart dependency between the Oracle ACFS file-system proxy resource and the Oracle ASM disk-group proxy resource.

You can create an Oracle ACFS file system for use as a general purpose file system or as an Oracle database home file system. Create these dependencies and affinities when you configure Oracle ACFS proxy resources for any applications that are managed by Oracle Solaris Cluster data services.

Configure Oracle Solaris Cluster resources for applications that use an Oracle ACFS file system with the following relationships:

- An offline-restart dependency on the corresponding Oracle ACFS file system proxy resource
- A strong positive affinity between the containing resource group and the Oracle ACFS proxy resource group

Create an Oracle Clusterware stop-trigger resource for every Oracle ACFS file system that is used by applications that are managed by Oracle Solaris Cluster. You must create this resource with hard—start and pull-up start dependencies and with a hard-stop dependency on the Oracle Clusterware ACFS resource.

To register and create instances of this resource type, use the following sequence of Oracle Solaris Cluster maintenance commands:
Register this resource type with the `cl_resource_type` command.
Create instances of this resource type with the `cl_resource` command.

To make an Oracle ACFS file system available in a zone cluster, configure its mount path under the zone-cluster root path. Use the `cl_zone_cluster` command to add this file system to the zone cluster.

To enable applications that are managed by Oracle Solaris Cluster to use an Oracle ACFS file system from a zone cluster, perform the following tasks:

- Create a `SUNW.wait_zc_boot` resource in the global zone and set the `ZCName` property to the name of the zone cluster.
- Create a `SUNW.scalable_acfs_proxy` resource in the zone cluster, with a strong positive affinity on the `SUNW.wait_zc_boot` resource group.
- Create an Oracle Clusterware proxy resource for the `SUNW.wait_zc_boot` resource.
- Add a hard-start dependency from the Oracle ACFS resource on the Oracle Clusterware proxy resource for the `SUNW.wait_zc_boot` resource.

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:

**Init_timeout**
- **Minimum**: 60
- **Default**: 300

**Fini_timeout**
- **Minimum**: 60
- **Default**: 300

**Prenet_start_timeout**
- **Minimum**: 60
- **Default**: 300

**Post_stop_timeout**
- **Minimum**: 60
- **Default**: 300

Extension Properties The extension properties of the `SUNW.scalable_acfs_proxy` resource type are as follows:

**acfs_mountpoint**
This property specifies the mount point of an Oracle ACFS file system.

**Data Type**: String
**Defaults**

- **Default**: No default defined
- **Minimum length**: 1
- **Tunable**: When disabled

**debug_level**

*Note –* All SQL*Plus messages that the Oracle ACFS 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 ACFS 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

**proxy_probe_interval**

This property specifies the interval, in seconds, between probes of the Oracle ACFS resource for which this resource is acting as a proxy.

- **Data Type**: Integer
- **Range**: 5–300
- **Default**: 30
- **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 Oracle ACFS resource for which this resource is acting as a proxy.

- **Data Type**: Integer
- **Range**: 5–120
- **Default**: 60
- **Tunable**: Any time
EXAMPLE 1    Creating a scalable_acfs_proxy Resource

This example shows the commands to perform the following operations, which create a scalable_acfs_proxy resource on a two-node cluster:

1. Registering the SUNW.scalable_acfs_proxy resource type
2. Creating the acfs-rg resource group and setting the resource-group affinity
3. Adding the acfs-rs resource to the acfs-rg resource group with offline-restart resource dependencies and setting the acfs_mountpoint extension property for one Oracle ACFS file system

The example makes the following assumptions:

- The bash shell is used.
- A resource group that is named asm-dg-rg exists and contains a resource of type SUNW.scalable_asm_diskgroup_proxy that is named asm-dg-rs.
- The Oracle ACFS file-system mount point is /acfs_mount.

```
phys-schost-1# clresourcetype register SUNW.scalable_acfs_proxy
phys-schost-1# clresourcegroup create -S -p rg_affinities=++asm-dg-rg acfs-rg
phys-schost-1# clresource create -g acfs-rg -t SUNW.scalable_acfs_proxy -p acfs_mountpoint=/acfs_mount -p resource_dependencies_offline_restart=asm-dg-rs -d acfs-rs
phys-schost-1# clresourcegroup online -M acfs-rg
phys-schost-1# clresource enable acfs-rs
```

Attributes    See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscor</td>
</tr>
</tbody>
</table>

See Also    clresource(1CL), clresourcegroup(1CL), clresourcetype(1CL)
The SUNW.scalable_asm_diskgroup_proxy resourcetype represents the clustered Oracle ASM disk group in an Oracle Solaris Cluster configuration. This resourcetype is introduced for use starting with Oracle 11g release 2 configurations.

Note – The SUNW.scalable_asm_diskgroup_proxy resourcetype can only be used when using Oracle Grid Infrastructure for Clusters.

The SUNW.scalable_asm_diskgroup_proxy resourcetype is a multiple-master resourcetype. 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 resourcetype 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:

- Oracle Solaris Cluster Manager
- The clsetup 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 command.
  - To create instances of this resource type, use the clresource 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:

<table>
<thead>
<tr>
<th>Property</th>
<th>Minimum</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Prenet_start_timeout</code></td>
<td>60</td>
<td>300</td>
</tr>
<tr>
<td><code>Prenet_stop_timeout</code></td>
<td>60</td>
<td>300</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>String array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Default</td>
<td>No default defined</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No debug messages</td>
</tr>
<tr>
<td>1</td>
<td>Function Begin and End messages</td>
</tr>
<tr>
<td>2</td>
<td>All debug messages and function Begin/End messages</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0–2</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
</tbody>
</table>
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-rs resource group
2. Registering the SUNW.scalable_asm_diskgroup_proxy resource type
3. Setting the resource group affinity
4. Adding the asm-dg-rs resource to the asm-dg-rs 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-rs 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-rs exists and contains a resource of type SUNW.ScalMountPoint that is named scal-mp-rs, for Oracle files.

```
phys-schost-1# cresource type register SUNW.scalable_asm_diskgroup_proxy
phys-schost-1# cresource group create -S asm-dg-rg
phys-schost-1# cresource group set -p rg_affinities=++asm-inst-rs asm-dg-rg
phys-schost-1# cresource create -g asm-dg-rs \
- t SUNW.scalable_asm_diskgroup_proxy \
```
EXAMPLE 1 Creating a scalable_asm_diskgroup_proxy Resource (Continued)

-p asm_diskgroups=data1 \n-p resource_dependencies_offline_restart=casm-inst-rs,qfs-mp-rs \n-d asm-dg-rs

phys-schost-1# clresourcegroup online -M asm-dg-rg

phys-schost-1# clresource enable asm-dg-rg

Attributes See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscor</td>
</tr>
</tbody>
</table>

See Also Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide, Oracle Solaris Cluster Data Service for Oracle Guide, and Oracle Solaris Cluster Data Services Planning and Administration Guide
SUNW.scalable_asm_instance, scalable_asm_instance – resource type implementation for the Oracle single instance Automated Storage Management (ASM) managed by Oracle Solaris Cluster

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:

- Oracle Solaris Cluster Manager
- The csetup(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:

<table>
<thead>
<tr>
<th>Failover_mode</th>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SOFT</td>
<td>Any time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load_balancing_policy</th>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LB_WEIGHTED</td>
<td>At creation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load_balancing_weights</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Configuration</td>
<td>Tunable</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Monitor_check_timeout</td>
<td>Minimum</td>
</tr>
<tr>
<td></td>
<td>Default</td>
</tr>
<tr>
<td>Monitor_start_timeout</td>
<td>Minimum</td>
</tr>
<tr>
<td></td>
<td>Default</td>
</tr>
<tr>
<td>Monitor_stop_timeout</td>
<td>Minimum</td>
</tr>
<tr>
<td></td>
<td>Default</td>
</tr>
<tr>
<td>Network_resources_used</td>
<td>Default</td>
</tr>
<tr>
<td>Port_list</td>
<td>Minimum</td>
</tr>
<tr>
<td></td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Tunable</td>
</tr>
<tr>
<td>Retry_count</td>
<td>Maximum</td>
</tr>
<tr>
<td></td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Tunable</td>
</tr>
<tr>
<td>Retry_Interval</td>
<td>Maximum</td>
</tr>
<tr>
<td></td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Tunable</td>
</tr>
<tr>
<td>Scalable</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Tunable</td>
</tr>
<tr>
<td>Start_timeout</td>
<td>Minimum</td>
</tr>
</tbody>
</table>
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 -C option for `pmfadm(1M)`. All child processes and their descendents are monitored.

**Category** Optional
**Default** -1
**Tunable** When disabled

default_level

**Note** – All SQL*Plus and svrmgr 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
Function Begin and End messages

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

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0–2</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>True</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Info</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
</tbody>
</table>

Network_aware (boolean)
This property specifies whether an application uses the network.

<table>
<thead>
<tr>
<th>Category</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>False</td>
</tr>
</tbody>
</table>
Tunable

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: "/opt/SUNWscor/oracle_asm/bin/asm_control probe -R %RS_NAME -G %RG_NAME -T %RT_NAME"
  - 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: "/opt/SUNWscor/oracle_asm/bin/asm_control start -R %RS_NAME -G %RG_NAME -T %RT_NAME"
  - Tunable: None

Stop_command(string)
- Specifies the command that stops the single instance ASM.
  - Category: Required
  - Default: "/opt/SUNWscor/oracle_asm/bin/asm_control stop -R %RS_NAME -G %RG_NAME -T %RT_NAME"
  - 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# cresource-type register SUNW.scalable_asm_instance
phys-schost-1# cresourcegroup create -S asm-inst-rg
phys-schost-1# cresource create -g asm-inst-rg \
-p Oracle_home=/export/home/oraasm/oracle/product/10.2.0/db_1 \
-p asm-inst-rs
phys-schost-1# cresourcegroup online -M asm-inst-rg
phys-schost-1# cresource 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# cresource create -g asm-inst-rg \
-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 Service for Oracle Real Application Clusters Guide, Oracle Solaris Cluster Data Service for Oracle Guide, and Oracle Solaris Cluster Data Services Planning and Administration Guide
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

The **SUNW.scalable_asm_instance_proxy** resourcetype 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. For more information on configuring resources for Oracle database instances, see *Oracle Solaris Cluster Data Service for Oracle Guide* or *Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide*.

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

- Oracle Solaris Cluster Manager
- 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**

<table>
<thead>
<tr>
<th>Minimum</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>300</td>
</tr>
</tbody>
</table>

**Prenet_stop_timeout**

<table>
<thead>
<tr>
<th>Minimum</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>300</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Range</th>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Not applicable</td>
<td>No default defined</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

**debug_level**

Note – All SQL*Plus and srvmgr 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:

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No debug messages</td>
</tr>
<tr>
<td>1</td>
<td>Function Begin and End messages</td>
</tr>
<tr>
<td>2</td>
<td>All debug messages and function Begin/End messages</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Data Type</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Default</td>
<td>No default defined</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Default</td>
<td>NULL</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>5–120</td>
</tr>
<tr>
<td>Default</td>
<td>60</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer</th>
</tr>
</thead>
</table>
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-fmwk-rg exists and contains resources that are shown in the following table:

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNW.crs_framework</td>
<td>crs_framework-rs</td>
</tr>
<tr>
<td>SUNW.rac_framework</td>
<td>rac_framework-rs</td>
</tr>
<tr>
<td>SUNW.rac_udlm</td>
<td>rac_udlm-rs</td>
</tr>
</tbody>
</table>

```
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-fmwk-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-fmwk-rs qfs-mp-rs -d asm-inst-rs
```

**See Also**

*Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide*, *Oracle Solaris Cluster Data Service for Oracle Guide*, and *Oracle Solaris Cluster Data Services Planning and Administration Guide*
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** 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.

For information about how to configure RAC listener resources, see "Configuring Resources for Oracle RAC Database Instances" in *Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide*.

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

- Oracle Solaris Cluster Manager
- 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.

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
<table>
<thead>
<tr>
<th>Setting</th>
<th>Minimum</th>
<th>Default</th>
<th>Tunable</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fini_timeout</td>
<td>5</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Init_timeout</td>
<td>5</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor_start_timeout</td>
<td>5</td>
<td>180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor_stop_timeout</td>
<td>5</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retry_count</td>
<td>-1</td>
<td>-1</td>
<td>Any time</td>
<td></td>
</tr>
<tr>
<td>Retry_interval</td>
<td>-1</td>
<td>2592000</td>
<td>600</td>
<td>Any time</td>
</tr>
<tr>
<td>Start_timeout</td>
<td>5</td>
<td>300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop_timeout</td>
<td>5</td>
<td>300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thorough_probe_interval</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0–100</td>
</tr>
<tr>
<td>Default</td>
<td>1, which logs syslog messages</td>
</tr>
<tr>
<td>Tunable</td>
<td>Any time</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Default</td>
<td>LISTENER</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>
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
EXmple 1 Creating a scalable_rac_listener Resource (Continued)

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.

The example makes the following assumptions:

- The C shell is used.
- A RAC framework resource group that is named rac-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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscor</td>
</tr>
</tbody>
</table>

See Also clresource(1CL), clresourcetype(1CL), clsetup(1CL), attributes(5), r_properties(5), SUNW.rac_framework(5), SUNW.rac_svm(5), SUNW.rac_udlm(5), SUNW.scalable_rac_server(5), SUNW.scalable_rac_server_proxy(5)

Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide, Oracle Solaris Cluster Data Services Planning and Administration Guide
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. For more information, see “Configuring Resources for Oracle RAC Database Instances” in Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide.

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

- Oracle Solaris Cluster Manager
- 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

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

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

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

Preset actions for logged alerts are listed in Appendix B, “Preset Actions for DBMS Errors and Logged Alerts,” in Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide. To change the action that the RAC server fault monitor performs, customize the server fault monitor as explained in “Customizing the Oracle 9i RAC Server Fault Monitor” in Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide.
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 syslog 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.
**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/init<oracle-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.

**probe_timeout**

This property specifies the timeout value, in seconds, that the fault monitor uses when checking the status of an Oracle RAC server.

**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
\textit{variable-name}=\textit{value}. Each definition must start on a new line in the environment file.

You can specify a different value of the \textit{user\_env} extension property for each node that can
master the resource.

\begin{tabular}{|l|l|}
\hline
\textbf{Data Type} & String \\
\hline
\textbf{Range} & Not applicable \\
\hline
\textbf{Default} & No default defined \\
\hline
\textbf{Tunable} & Any time \\
\hline
\end{tabular}

\textit{wait\_for\_online}

This property specifies whether the \textit{START} method of the Oracle RAC server resource waits
for the database to be online before the \textit{START} method exits. The permitted values for this
property are as follows:

\begin{itemize}
  \item \textbf{True} \hspace{1cm} Specifies that the \textit{START} method of the Oracle RAC server resource waits for the
database to be online before the \textit{START} method exits.
  \item \textbf{False} \hspace{1cm} Specifies that the \textit{START} method runs the commands to start the database but
does not wait for the database to come online before the \textit{START} method exits.
\end{itemize}

\begin{tabular}{|l|l|}
\hline
\textbf{Data Type} & Boolean \\
\hline
\textbf{Range} & Not applicable \\
\hline
\textbf{Default} & True \\
\hline
\textbf{Tunable} & Any time \\
\hline
\end{tabular}

\textbf{Examples} \hspace{1cm} \textbf{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 \textit{rac-db-rg} resource group
2. Registering the \textit{SUNW.scalable\_rac\_server} resource type
3. Adding the scalable\_rac\_server-rs resource to the \textit{rac-db-rg} resource group

A different value of the following extension properties is set for each node that can master the
resource:

\begin{itemize}
  \item alert\_log\_file
  \item oracle\_sid
\end{itemize}

The example makes the following assumptions:

\begin{itemize}
  \item The C shell is used.
\end{itemize}
EXAMPLE 1 Creating a scalable_rac_server Resource  

AA RAC framework resource group that is named rac-framework-rg exists and contains resources of types as shown in the following table:

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNW.rac_framework</td>
<td>rac_framework-rs</td>
</tr>
<tr>
<td>SUNW.rac_udlm</td>
<td>rac_udlm-rs</td>
</tr>
<tr>
<td>SUNW.rac_svm</td>
<td>rac_svm-rs</td>
</tr>
</tbody>
</table>

■ 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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscor</td>
</tr>
</tbody>
</table>

See Also  clresource(1CL), clresourcetype(1CL), clsetup(1CL), custom_action_file(4), attributes(5), r_properties(5), SUNW.rac_framework(5), SUNW.rac_svm(5), SUNW.rac_udlm(5), SUNW.scalable_rac_listener(5), SUNW.scalable_rac_server_proxy(5)

*Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide, Oracle Solaris Cluster Data Services Planning and Administration Guide*
SUNW.scalable_rac_server_proxy, scalable_rac_server_proxy – resourcetype
implementation for the Oracle Real Application Clusters (Oracle RAC) server proxy managed by Oracle Solaris Cluster

The SUNW.scalable_rac_server_proxy resourcetype 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 resourcetype 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 interoperate.

The SUNW.scalable_rac_server_proxy resourcetype 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. For more information, see “Configuring Resources for Oracle RAC Database Instances” in Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide.

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

- Oracle Solaris Cluster Manager
- 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.
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    Anytime

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Minimum</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>5</td>
<td>600</td>
</tr>
<tr>
<td>Default</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thorough_probe_interval

<table>
<thead>
<tr>
<th>Property</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Default</th>
<th>Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td>2592000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td></td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tunable</td>
<td></td>
<td>Any time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Update_timeout

<table>
<thead>
<tr>
<th>Property</th>
<th>Minimum</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td></td>
<td>240</td>
</tr>
</tbody>
</table>

Validate_timeout

<table>
<thead>
<tr>
<th>Property</th>
<th>Minimum</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td></td>
<td>120</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Range</td>
<td>1–25</td>
</tr>
<tr>
<td>Default</td>
<td>3</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Range</td>
<td>1–3600</td>
</tr>
<tr>
<td>Default</td>
<td>5</td>
</tr>
<tr>
<td>Tunable</td>
<td>When disabled</td>
</tr>
</tbody>
</table>
**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 syslog 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:

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNW.crs_framework</td>
<td>crs_framework-rs</td>
</tr>
<tr>
<td>SUNW.rac_framework</td>
<td>rac_framework-rs</td>
</tr>
<tr>
<td>SUNW.rac_udlm</td>
<td>rac_udlm-rs</td>
</tr>
</tbody>
</table>

```
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:

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscor</td>
</tr>
</tbody>
</table>

**See Also**  clresource(1CL), clresourcetype(1CL), clsetup(1CL), attributes(5), r_properties(5), SUNW.rac_udlm(5), SUNW.scalable_rac_server(5)
Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide, Oracle Solaris Cluster Data Services Planning and Administration Guide
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 \texttt{RUN} server script. This script is located in the \$SYBASE/ASE\textunderscore major\textunderscore version/install directory, where \texttt{major\textunderscore version} is the major version of Sybase ASE that you are using. For example, if you are using Sybase ASE version 12.5.1, \texttt{major\textunderscore 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 \texttt{RUN} server script. This script is located in the \$SYBASE/ASE\textunderscore major\textunderscore version/install directory, where \texttt{major\textunderscore version} is the major version of Sybase ASE that you are using. For example, if you are using Sybase ASE version 12.5.1, \texttt{major\textunderscore 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 \texttt{Connect\textunderscore 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:

\texttt{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 \textit{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 \texttt{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 \texttt{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 \texttt{RUN} server script. This script is located in the \$SYBASE/ASE\_\textit{major-version}/install directory, where \textit{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, \textit{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 \texttt{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 \texttt{RUN} server script. This script is located in the \$SYBASE/ASE\_\textit{major-version}/install directory, where \textit{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, \textit{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:
<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscsyb</td>
</tr>
</tbody>
</table>

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, Oracle Solaris Cluster Data Service for Sybase ASE Guide*
**Description**  The Resource Group Manager (RGM) manages Oracle Solaris Cluster HA for WebLogic Server for Oracle Solaris Cluster. Use the `clresource(1C)` 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 at any time.

**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_Uri 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_Uri**

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.

c resource create -g bea-rg -t SUNW.wls \ 
-p Configdir_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)

cresource 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.

cresource 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.

cresource create -g bea-rg -t SUNW.wls \
-p Confdir_list=/global/bea/beahome/weblogic700 \
-p Server_url=http://logical-host-1:710 \n-p Start_script=/global/bea/beahome/user_projects/ha-wls/startManagedWebLogic.sh \n-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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWWls</td>
</tr>
</tbody>
</table>

See Also  attributes(5), clresource(1CL), r_properties(5), scha_resource_get(3HA)

*Oracle Solaris Cluster Data Service for Oracle WebLogic Server Guide*
Index

A
acfs_mountpoint extension property, 170
alert_log_file extension property
   scalable_rac_server resource type, 196
   SUNW.scalable_rac_server resource type, 196
apache, 55
Architecture extension property, ORCL.sapdia
   resource type, 40
asm_diskgroups extension property, 174
Auto_End_Bkp extension property, 108

C
Child_mon_level extension property
   SUNW.sapenq resource type, 136
   SUNW.saprepl resource type, 145
client_retries extension property, 206
client_retry_interval extension property, 206
connect_cycle extension property
   scalable_rac_server resource type, 197
   SUNW.scalable_rac_server resource type, 197
connect_string extension property
   scalable_rac_server resource type, 197
   SUNW.scalable_rac_server resource type, 197
crs_home extension property, 186, 207
custom_action_file extension property
   scalable_rac_server resource type, 198
   SUNW.scalable_rac_server resource type, 198
custom_action_file file, 10

D
database users, 133
databases
   instance name, 108, 187, 198, 208
DB_Name extension property, 131
db_name extension property, 207
DB_User extension property, 131
dbmcli command
   path to, 132
   start option, 130
dbmcli_Start_Option extension property, 130
Debug_level extension property, 108
   ORCL.sapcentr resource type, 34
   ORCL.sapdia resource type, 38
   ORCL.saprepenq_preempt resource type, 48
   ORCL.saprepenq resource type, 44
   ORCL.sapstartsrv resource type, 51
debug_level extension property
   scalable_asm_diskgroup_proxy resource
type, 174
   scalable_asm_instance_proxy resource type, 171, 186
   scalable_rac_listener resource type, 191
   scalable_rac_server_proxy resource type, 207
   scalable_rac_server resource type, 198
   SUNW.scalable_asm_diskgroup_proxy resource
type, 174
   SUNW.scalable_asm_instance_proxy resource
type, 171, 186
   SUNW.scalable_rac_listener resource type, 191
   SUNW.scalable_rac_server_proxy resource
type, 207
debug_level extension property *(Continued)*
SUNW.scalable_rac_server resource type, 198
directories
Oracle home, 108, 187, 192, 198, 208
dns, 69

hot backup mode, 108

I
identifiers
system, 108, 187, 198, 208
Independent_Program_Path extension property,
SUNW.sapdb resource type, 132
Instance_name extension property
ORCL.sapcentr resource type, 35
ORCL.sapdia resource type, 39
ORCL.saprepenq resource type, 44
ORCL.sapstartsrv resource type, 52
Instance_number extension property
ORCL.sapcentr resource type, 35
ORCL.saprepenq resource type, 45
ORCL.sapstartsrv resource type, 52
iws, 80

J
jsas-na Sun Java System Application Server, resource
type implementation for failover Sun Java System
Application Server, 90
jsas Sun Java System Application Server, 86

K
kernel processes
identities
paths to files containing, 132
termination of parent, 133
krb5 resource type, 94

L
ldom, 97
libraries
MaxDB
SUNW.sapdb resource type, 132
listener_name extension property
  scalable_rac_listener resource type, 191
  SUNW.scalable_rac_listener resource type, 191
Log_Directory extension property
  SUNW.sapenq resource type, 137, 158
  SUNW.saprepl resource type, 145
log files
  Oracle RAC server proxy, 207
  RAC listener, 191
  RAC server, 198

M
MaxDB application
  paths to programs and libraries
    SUNW.sapdb resource type, 132
  process identities
    paths to files containing, 132
MaxDB database instance
  database user, 133
  name, 131
  operating system user, 131
  starting, 130
maximum values
  restarts
    SUNW.krb5 resource type, 95
    SUNW.sapdb resource type, 132
messages
monitor_probe_interval extension property, 207
Monitor_retry_count extension property
  SUNW.krb5 resource type, 95
  SUNW.sapdb resource type, 132
  SUNW.sapenq resource type, 137
  SUNW.saprepl resource type, 145
Monitor_retry_Count extension property
  SUNW.sapscs resource type, 151
  SUNW.sapwebas resource type, 157
Monitor_retry_interval extension property
  SUNW.krb5 resource type, 95
  SUNW.sapdb resource type, 132
  SUNW.sapenq resource type, 137
  SUNW.saprepl resource type, 146
Monitor_Retry_Interval extension property
  SUNW.sapscs resource type, 151
  SUNW.sapwebas resource type, 157
Msg_Server_Monitor extension property, SUNW.sapscs
  resource type, 152
Msg_Server_Port extension property, SUNW.sapscs
  resource type, 152

N
names
  MaxDB database instance, 131
  Oracle database instance, 108, 187, 198, 208
  nfs, 101

O
obiee_cluster_controller, 18
obiee_presentation_service, 20
obiee_scheduler, 22
obiee_server, 24
ohs, 26
operating system users, of MaxDB database
  instance, 131
opmn, 28
oracle_external_proxy, 30
Oracle_Home extension property, 108
oracle_home extension property
  scalable_asm_instance_proxy resource type, 187
  scalable_rac_listener resource type, 192
  scalable_rac_server_proxy resource type, 208
  scalable_rac_server resource type, 198
  SUNW.scalable_asm_instance_proxy resource type, 187
  SUNW.scalable_rac_listener resource type, 192
  SUNW.scalable_rac_server_proxy resource type, 208
  SUNW.scalable_rac_server resource type, 198
  oracle_listener, 105
  oracle_server, 112
Oracle_Sid extension property, 108
oracle_sid extension property
  scalable_asm_instance_proxy resource type, 187
oracle_sid extension property (Continued)
   scalable_rac_server_proxy resource type, 208
   scalable_rac_server resource type, 198
   SUNW.scalable_asm_instance_proxy resource type, 187
   SUNW.scalable_rac_server_proxy resource type, 208
   SUNW.scalable_rac_server resource type, 198
   ORCL.obiee_cluster_controller, 18
   ORCL.obiee_presentation_service, 20
   ORCL.obiee_scheduler, 22
   ORCL.obiee_server, 24
   ORCL.ohs, 26
   ORCL.opmn, 28
   ORCL.oracle_external_proxy, 30
   ORCL.sapcentr resource type, 34
   ORCL.sapdia resource type, 38
   ORCL.saprepenq_preempt resource type, 47
   ORCL.saprepenq resource type, 43
   ORCL.sapstartsrv resource type, 51

P
Parameter_file extension property, 109
   parameter_file extension property
      scalable_rac_server resource type, 199
      SUNW.scalable_rac_server resource type, 199
   paths
      dbmcli command, 132
      MaxDB programs and libraries
         SUNW.sapdb resource type, 132
         process identity files, 132
   Pid_Dir_Path extension property, 132
   probe_timeout extension property
      scalable_rac_listener resource type, 192
      scalable_rac_server resource type, 199
   Probe_Timeout extension property
      SUNW.krb5 resource type, 96
      SUNW.sapdb resource type, 133
      SUNW.sapenq resource type, 138
      SUNW.saprepl resource type, 146
   probe_timeout extension property
      SUNW.scalable_rac_listener resource type, 192
      SUNW.scalable_rac_server resource type, 199

Probe_Timeout extension property
   SUNW.sapscs resource type, 152
   SUNW.sapwebas resource type, 157
   processes
      identities
         paths to files containing, 132
         termination of parent, 133
   programs
      MaxDB
         SUNW.sapdb resource type, 132
   proxy_probe_interval extension property, 171
   proxy_probe_timeout extension property, 171, 175, 187, 208

R
Repenqres extension property,
   ORCL.saprepenq_preempt resource type, 48
   Replica_Profile extension property, SUNW.saprepl resource type, 146
   Replica_Server extension property, SUNW.saprepl resource type, 146
   resource type, ORCL.sapstartsrv, 51
   resource type implementation for failover and scalable
      Apache Web Server, 55
      resource type implementation for failover and scalable
      Oracle iPlanet Web Server, 80
      resource type implementation for failover Domain Name Service (DNS), 69
      resource type implementation for failover guest domains, 97
      resource type implementation for failover Oracle HTTP Server (OHS), 26
      resource type implementation for failover Oracle Process Management and Notification Server (OPMN), 28
      resource type implementation for failover SAP liveCache database, 141
      resource type implementation for failover Siebel gateway, 165
     resource type implementation for failover Siebel server, 167
resource type implementation for HA-Oracle Business Intelligence Enterprise Edition (Oracle BIEE) Cluster Controller, 18
resource type implementation for HA-Oracle Business Intelligence Enterprise Edition (Oracle BIEE) Presentation Service, 20
resource type implementation for HA-Oracle Business Intelligence Enterprise Edition (Oracle BIEE) Scheduler, 22
resource type implementation for HA-Oracle Business Intelligence Enterprise Edition (Oracle BIEE) server, 24
resource type implementation for HA Oracle server, 112
resource type implementation for Oracle Solaris Cluster HA for NFS, 101
resource type implementation for Oracle Solaris Cluster HA for SAP application server, 124
resource type implementation for Oracle Solaris Cluster HA for SAP central instance., 127
resource type implementation for Oracle Solaris Cluster HA for Sybase Adaptive Server Enterprise (ASE), 212
resource type implementation for scalable SAP server, 162
resource type implementation for the Oracle External Proxy managed by Oracle Solaris Cluster, 30
resource type implementation for the Oracle listener, 105
resource types (Continued)
sapwebas, 156
SUNW.krb5, 94
SUNW.sapdb, 130
SUNW.sapenq, 135
SUNW.saprepl, 144
SUNW.sapscs, 150
SUNW.sapwebas, 156
Restart_if_Parent_Terminated extension property, 133
reverts
interval between
SUNW.krb5 resource type, 95
SUNW.sapdb resource type, 132
maximum allowed
SUNW.krb5 resource type, 95
SUNW.sapdb resource type, 132
S
s1as Sun Java System Application Server, 117
s1mq Sun Glassfish Message Queue, 121
sap_as, 124
sap_as_v2, 124
SAP central server
resource type, 34, 38, 43, 47, 51, 150, 156
sap_ci, 127
sap_ci_v2, 127
SAP_Instance_Name extension property
SUNW.sapscs resource type, 152
SUNW.sapwebas resource type, 157
SAP_Instance_Number extension property
ORCL.sapdia resource type, 39
SUNW.sapscs resource type, 153
SUNW.sapwebas resource type, 157
sap_livecache, 141
SAP replicated enqueue replica server, resource type, 144
SAP replicated enqueue server, resource type, 135
SAP_SID extension property
SUNW.sapscs resource type, 153
SUNW.sapwebas resource type, 158
Sap_user extension property
ORCL.sapcentr resource type, 35
Sap_user extension property (Continued)
  ORCL.sapdia resource type, 39
  ORCL.saprepenq_preempt resource type, 48
  ORCL.saprepenq resource type, 44
  ORCL.sapstartsrv resource type, 51

SAP_User extension property
  SUNW.sapenq resource type, 138
  SUNW.saprepl resource type, 146
  SUNW.sapscs resource type, 153
  SUNW.sapwebas resource type, 158

sap_xserver, 162
sapcentr resource type, 34
sapdb resource type, 130
sapdia resource type, 38
sapenq resource type, 135
saprepenq_preempt resource type, 47
saprepenq resource type, 43
saprepl resource type, 144
sapscs resource type, 150
sapstartsrv resource type, 51
sapwebas resource type, 156
sblgtwy, 165
sblsrvr, 167

Scs_Shutdown_Script extension property,
  SUNW.sapscs resource type, 153

Scs_Startup_Script extension property, SUNW.sapscs
  resource type, 153

Sid extension property
  ORCL.sapcentr resource type, 35
  ORCL.sapdia resource type, 39
  ORCL.saprepenq_preempt resource type, 48
  ORCL.saprepenq resource type, 44
  ORCL.sapstartsrv resource type, 52

Start_script extension property
  ORCL.sapcentr resource type, 36
  ORCL.sapdia resource type, 40
  ORCL.saprepenq resource type, 45

starting, MaxDB database instance, 130

startup_wait_count extension property, 208

Stop_script extension property
  ORCL.sapcentr resource type, 36
  ORCL.sapdia resource type, 40
  ORCL.saprepenq resource type, 45

Stop_signal extension property
  SUNW.sapenq resource type, 138
  SUNW.saprepl resource type, 147

Sun Java System Application Server, 90
  resource type implementation for failover and
  scalable Sun Glassfish Message Queue, 121
  resource type implementation for failover and
  scalable Sun Java System Application Server, 90
  resource type implementation for failover and
  scalable SunJava System Application Server, 86
  resource type implementation for failover Sun
  Glassfish Message Queue, 121
  resource type implementation for failover Sun Java
  System Application Server, 86

Sun Java System Application Server”>resource type
  implementation for Sun Java System Application
  Server EE (HADB), 76

Sun Open Net Environment (Sun ONE) Application
  Server (iWS)
  resource type implementation for failover and
  scalable SunJava System Application Server, 117
  resource type implementation for failover Sun Java
  System Application Server, 117

Sun Open Net Environment (Sun ONE) HADB
  resource type implementation for Sun Java System
  Application Server EE (HADB), 73

SUNW.apache, 55
SUNW.dns, 69
SUNW.hadb maSunJava System Application
  Server, 76
SUNW.hadb Sun Java System Application Server EE
  (HADB), 73
SUNW.iws, 80
SUNW.jsas-na Sun Java System Application Server, 90
SUNW.jsas Sun Java System Application Server, 86
SUNW.krb5
  resource type for collecting data on system resource
  usage, 94
  resource type implementation of Kerberos KDC
  server, 94
SUNW.krb5 resource type, 94
SUNW.ldom, 97
SUNW.nfs, 101
SUNW.oracle_listener, 105
SUNW.oracle_server, 112
SUNW.s1as Sun Java System Application Server, 117
SUNW.s1mq Sun Glassfish Message Queue, 121
SUNW.sap_as, 124
SUNW.sap_as_v2, 124
SUNW.sap_ci, 127
SUNW.sap_ci_v2, 127
SUNW.sap_livecache, 141
SUNW.sap_xserver, 162
SUNW.sapdb resourcetype, 130
SUNW.sapenq resourcetype, 133
SUNW.saprepl resource type, 144
SUNW.sapscs resource type, 150
SUNW.sapwebas resource type, 156
SUNW.sblgtwy, 165
SUNW.sblsrvr, 167
SUNW.sybase, 212
system identifiers
Oracle, 108, 187, 198, 208

User_Key extension property, 133
users
of MaxDB database instance
database, 133
operating system, 131

Wait_for_online extension property, 109
wait_for_online extension property
scalable_rac_server resource type, 200
SUNW.scalable_rac_server resource type, 200
Webas_Probe_J2ee extension property, SUNW.sapwebas resource type, 158
Webas_Shutdown_Script extension property,
SUNW.sapwebas resource type, 159
Webas_Startup_Script extension property,
SUNW.sapwebas resource type, 159
Webas_Use_Pmf extension property, SUNW.sapwebas resource type, 159

.XUSER.62 file, and DB_User extension property, 131

Yellow extension property
SUNW.sapenq resource type, 36, 40, 45