

IBM Tivoli System Automation for Multiplatforms



Guide and Reference

Version 1.2

IBM Tivoli System Automation for Multiplatforms



Guide and Reference

Version 1.2

Note!

Before using this information and the product it supports, read the information in “Notices” on page 277.

Fifth Edition (April 2005)

This edition of the *IBM Tivoli System Automation for Multiplatforms® Guide and Reference* applies to IBM Tivoli System Automation for Multiplatforms Version 1.2, program numbers 5639–N53 and 5655–I53, and to all subsequent releases of this product until otherwise indicated in new editions.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Str. 220
D-71032 Boeblingen
Federal Republic of Germany

FAX (Germany): 07031+16-3456
FAX (Other Countries): (+49)+7031-16-3456

Internet e-mail: eservdoc@de.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 2003, 2004. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book	ix
Who should use this book	ix
How to use this book	ix
Where to find more information	ix
Summary of changes for Release 2	x
Conventions	x
ISO 9000	xi
Related information	xi
How to obtain publications	xi
How to reach us by e-mail	xi
Chapter 1. Introduction	1
Overview	1
High availability and resource monitoring	1
Policy based automation	1
Automatic recovery	1
Automatic movement of applications	1
Resource grouping	1
IBM Tivoli System Automation terms	2
Cluster / peer domain	2
Resource	2
Resource attributes	2
Resource class	2
Resource group	3
Managed resource	3
Nominal state	3
Equivalency	3
Relationships	3
Quorum	3
Tie breaker	4
Components of IBM Tivoli System Automation	5
Introducing resources managers provided by IBM Tivoli System Automation	5
Chapter 2. Installing IBM Tivoli System Automation	7
Planning for the installation	7
Coexistence with other products:	7
Contents of the CD:	7
Electronic distribution of IBM Tivoli System Automation	7
Supported platforms:	7
Supported network interfaces	8
Preparing for installation	9
Prerequisites	9
Installing and upgrading IBM Tivoli System Automation for Multiplatforms	11
Installing the product	11
Installing the product license	11
Languages supported by IBM Tivoli System Automation	11
Migrating the product	13
Uninstalling IBM Tivoli System Automation	16
Installing service	16
Chapter 3. Getting started	17
Step 1: Defining and administering a cluster	19
Creating a two nodes cluster	19

Adding a node to an existing cluster	21
Taking an entire cluster or individual nodes offline	22
Removing nodes from a cluster, or removing a complete cluster	23
Administering the recovery resource manager	24
Step 2: Defining RSCT resources	25
Creating application resource apache1	25
Creating IP address resource apache1IP	26
Creating an equivalency for the network adapters	27
Step 3: Defining the automation policy	28
Creating a resource group	28
Defining relationships	28
Bringing a resource group online	28
Chapter 4. Using resources	31
What is a resource?	31
What is a resource class?	31
What are resource attributes?	31
What is the difference between persistent attributes and dynamic attributes?	31
What is a managed resource?	32
Working with resources	32
Attributes used by resources	32
NodeNameList attribute	32
SelectFromPolicy attribute	33
ResourceType attribute	33
OpState attribute.	34
Nominal state of a resource.	34
Chapter 5. Using resource groups	35
What is a resource group?	35
Rules for using resource groups	36
Attributes used by resource groups	38
AllowedNode attribute	38
MemberLocation attribute	40
Name attribute	40
NominalState attribute.	40
Priority attribute	40
ExcludedList attribute	41
ActivePeerDomain	41
OpState attribute.	41
TopGroup attribute	42
AutomationDetails attribute	42
MoveStatus	43
ConfigValidity	43
Attributes used for resource group members	44
Mandatory attribute	44
MemberOf attribute	44
Defining and administering a resource group	45
Making (creating) a resource group	45
Adding a member resource to a resource group	45
Listing a resource group or its resource members	45
Starting and stopping a resource group	46
Changing attributes of a resource group	46
Changing the attributes of resource group members.	47
Removing a member resource from a resource group	47
Removing a resource group	47

Chapter 6. Using equivalencies	49
What is an equivalency?	49
Rules for using equivalencies	49
Attributes used by equivalencies	50
MemberClass attribute	50
Membership attribute	50
SelectString attribute	50
SelectFromPolicy attribute	50
Attributes used by members of equivalencies	50
Defining and administering equivalencies	51
Making (creating) an equivalency	51
List one or more equivalencies	51
Changing an equivalency	51
Removing an equivalency	51
Chapter 7. Using managed relationships	53
What is a managed relationship?	53
Attributes used by managed relationships	54
Name attribute	54
Source attribute	54
Target attribute	54
Relationship attribute	54
Condition attribute	55
Relationships for start / stop behavior	56
StartAfter relationship	56
StopAfter relationship	60
DependsOn relationship	62
DependsOnAny relationship	67
ForcedDownBy relationship	68
Location relationships	69
Conditions IfOnline, IfOffline, IfNotOnline, and IfNotOffline	69
Rules for using the Location relationships	70
Collocated relationship	71
AntiCollocated relationship	73
Affinity relationship	75
IsStartable relationship	77
Creating and administering relationships	79
Creating a relationship	79
Listing a relationship	79
Changing a relationship	80
Removing a relationship	80
Chapter 8. How IBM Tivoli System Automation processes the system information	81
Location relationship resolution: Binding algorithm	81
Events that might allow a resource group to become Online	83
Behavior Patterns of IBM Tivoli System Automation for Multiplatforms	84
General considerations	84
How IBM Tivoli System Automation reacts to the possible OpState changes of a resource that is online on a node	86
How System Automation composes the OpState of a resource group	88
How System Automation reacts to OpState changes of a resource that is started or stopped	89
How System Automation reacts if a resource is Online on a certain node and the MonitorCommand reports an OpState for the resource on another node at the same time	93
Chapter 9. Protecting your resources – quorum support	95
Overview	95

Configuration quorum	96
Operational quorum	96
Setting critical resources	97
Getting quorum information	98
Setting up and administering a tie breaker	99
Using a tie breaker	100
Overriding the operational quorum	104
Chapter 10. Setting up a high available network	105
Running a one or two node cluster: detecting network interface failures	105
Two nodes cluster, each node has one ethernet interface	105
Two nodes cluster, each node has two network interfaces	107
Two physically separated networks, move ServiceIP between nodes	107
Three logical networks in one physical network, move ServiceIP between network interfaces	108
Two physically separated networks, dynamic routing and VIPA	109
Chapter 11. Controlling and administering IBM Tivoli System Automation	111
Controlling IBM Tivoli System Automation	111
TimeOut and RetryCount	111
Stop operations.	112
Automation	112
ExcludedNodes	113
ResourceRestartTimeout	113
Examples	113
Automation policy management	114
Saving configuration elements	114
Restoring a saved configuration or replacing it with a new configuration	114
Moving resource groups with the rgreg command	116
Scope of a move	116
Processing of a move request	116
Move and relationships	116
Diagnosing IBM Tivoli System Automation Resources	117
Using the IBM Tivoli System Automation TEC event interface	119
What is the Tivoli Enterprise Console?	119
What are events?	119
Sending events to the TEC	119
Enabling the TEC publisher function	119
Publishing IBM Tivoli System Automation internal attributes into the RSCT infrastructure	122
Enabling IBM Tivoli System Automation for Coordinated Cross Platform Disaster Recovery for zSeries	122
Customizing the Tivoli Enterprise Integration Facility (EIF)	123
Installing Cpint	124
xDR heartbeating	124
DASD error reporting daemon	124
Stopping an xDR cluster / node for maintenance	124
Dynamically verifying resources.	126
IBM Tivoli System Automation Hints and Tips.	128
Rebooting a node	128
Stopping a node	128
Generating events in case of failures	128
Chapter 12. Resource managers provided by IBM Tivoli System Automation	129
Using the Global Resource Manager	129
What is the IBM.Application resource class?	129
What is the IBM.ServiceIP resource class?	137
Using the Test Resource Manager.	142
What is the IBM.Test resource class?	142

Attributes used by IBM.Test	142
Example: Create a test resource and manipulate its OpState	144
Chapter 13. IBM Tivoli System Automation commands	145
RSCT RMC commands and files	145
Setting the environment variables	145
addrgmbr	147
chequ	151
chrel	154
chrg	159
chrgmbr	162
lsequ	165
lsrel	168
lsrg	173
lsrgreq	178
lssamctrl	181
mkequ	183
mkrel	187
mkrq	191
rgmbrreq	194
rgreq	197
rmequ	200
rmrel	202
rmsg	206
rmsgmbr	208
samcfg	211
samctrl	213
samdiag	216
samlcm	219
Additional useful commands	220
pidmon	220
Using expressions.	222
SQL Restrictions	222
Supported Base Data Types	222
Structured Data Types	223
Data Types That Can Be Used for Literal Values	223
How Variable Names Are Handled	224
Operators That Can Be Used in Expressions	224
Default Precedence of Operators	226
Pattern Matching	228
Examples of Expressions	228
Appendix A. Messages used with IBM Tivoli System Automation	231
Appendix B. Troubleshooting	273
Resource Manager diagnostic files.	273
Recovering from RMC and Resource Manager problems	273
Using the ctsnap command	274
SRC-controlled commands	274
Notices	277
Trademarks	278
Index	279

About this book

This book provides the information needed to implement the IBM Tivoli System Automation for Multiplatforms policy based self-healing capability running on xSeries[®], zSeries[®], iSeries[®], pSeries[®] and AIX.

Who should use this book

This book is intended for system administrators who want to use the automation and failover capabilities of IBM Tivoli System Automation for Multiplatforms.

How to use this book

This book contains all of the information that you need to understand and use the IBM Tivoli System Automation for Multiplatforms (IBM Tivoli System Automation) product.

- Chapter 1 provides an introduction to IBM Tivoli System Automation. It gives an overview of IBM Tivoli System Automation, introduces the components, and explains the technical terms used in this manual.
- Chapter 2 provides information on how to install, upgrade, migrate, and uninstall IBM Tivoli System Automation.
- Chapter 3 describes how clusters and nodes are administered and how an automation policy is defined.
- Chapter 4 describes the common attributes of IBM Tivoli System Automation.
- Chapter 5 describes how resource groups are created and used.
- Chapter 6 describes how equivalencies are created and used.
- Chapter 7 describes how managed relationships are created and used.
- Chapter 8 describes how IBM Tivoli System Automation processes system information.
- Chapter 9 describes how IBM Tivoli System Automation protects your resources.
- Chapter 10 describes how a high available network is set up.
- Chapter 11 describes how IBM Tivoli System Automation is controlled and administered.
- Chapter 12 describes the resource managers provided by IBM Tivoli System Automation.
- Chapter 13 provides reference-type information describing the IBM Tivoli System Automation commands.
- Appendix A. lists the messages that IBM Tivoli System Automation generates.
- Appendix B provides useful information for troubleshooting.

Where to find more information

IBM Tivoli System Automation Home Page

IBM Tivoli System Automation has a home page on the World Wide Web, which offers up-to-date information and services, and other items of interest to IBM Tivoli System Automation users.

You can find the IBM Tivoli System Automation home page at:

<http://www-3.ibm.com/software/tivoli/products/sys-auto-linux/>

Summary of changes for Release 2

The following is a list of new functionality in Release 1.2 of IBM Tivoli System Automation:

- AIX: is supported as additional platform.
- National Language Support: the following languages are supported with Release 1.2:
 - French.
 - German.
 - Italian.
 - Portuguese (Brazilian).
 - Spanish.
 - Japanese.
 - Korean.
 - Chinese (Simplified).
 - Chinese (Traditional).
- Automation policy management: allows to save and restore the configuration.
- Usability enhancements:
 - Enhanced debugging functionality with the `samdiag` command: supports the user with detailed automation information about resources, group, equivalencies, and nodes.
 - Show top-level group: displays information about the top group if groups are nested.
 - Enhanced `-f` command option: enhances the usability of input files for the IBM Tivoli System Automation Command Line Interface (CLI).
 - Dynamic verification of resources: implements the verify methods when IBM Tivoli System Automation is notified, through RMC event notification, that a resource has changed. Formerly, resource verifications were mostly done at configuration time when a resource was defined to the daemon through the CLI. Now verification is performed synchronously.
 - ConfigValidity: After a policy was established, several things can happen to make the policy invalid. This attribute shows the reason for invalidity.
- Support of additional relationships: introduces two new relationships `StopAfter` and `ForcedDownBy`.
- Controlling resources and resource groups with requests:
 - Requests for starting and stopping resource groups and single resources.
 - Requests for moving resource groups to other nodes.
 - Multiple request sources and priorities.
- Generation TEC events: Whenever IBM Tivoli System Automation changes its configuration, resource status or encounters any problems, the Tivoli Enterprise Console (TEC) event interface can be used to notify the system administrator.

This new functionality is briefly described in this chapter.

In addition to the new functionality, IBM Tivoli System Automation supports additional hardware and software platforms now, including Linux on all eServers and AIX. For a detailed list of all supported platforms, see <http://www-3.ibm.com/software/tivoli/products/sys-auto-linux/requirements.html>.

Conventions

The following highlighting conventions are used in this book:

Bold	Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
<i>Italic</i>	Identifies parameters whose actual names or values are to be supplied by the user.

monospace	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.
-----------	---

This manual uses symbols to show resources, resource groups, equivalencies, and relationships. The symbols used are as follows:

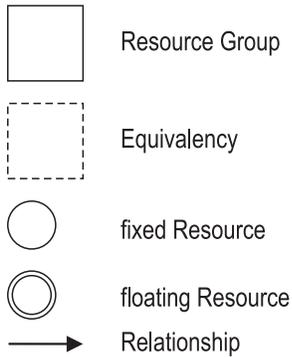


Figure 1.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Related information

The following RSCT documents are part of your IBM Tivoli System Automation CD:

- *IBM Reliable Scalable Cluster Technology for Linux, Administration Guide*, SA22-7892.
- *IBM Reliable Scalable Cluster Technology for Linux, Technical Reference*, SA22-7893.
- *IBM Reliable Scalable Cluster Technology for Linux, Messages*, GA22-7894.
- *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615-00

RSCT documents can also be found at the following Web site:

<http://www-1.ibm.com/servers/eserver/clusters/library/>

You might also need to refer to the following IBM Redpaper:

- *Linux on IBM zSeries and S/390®: High Availability for z/VM® and Linux*

It can be found at the following Web site:

<http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0220.html>

How to obtain publications

The IBM Tivoli System Automation publications are also available (valid at the time of release) at this Web sites:

<http://www-1.ibm.com/servers/eserver/clusters/library/>
<http://www-1.ibm.com/servers/eserver/zseries/software/sa/>
http://www.tivoli.com/support/public/Prodman/public_manuals/td/TD_PROD_LIST.html

How to reach us by e-mail

If you would like to contact us by e-mail, send your comments to eservdoc@de.ibm.com

Chapter 1. Introduction

Overview

IBM Tivoli System Automation manages the availability of applications running in Linux systems or clusters on xSeries, zSeries, iSeries, pSeries, and AIX systems or clusters. It consists of the following features:

High availability and resource monitoring

IBM Tivoli System Automation provides a high availability environment. High availability describes a system which is continuously available and which has a self-healing infrastructure to prevent downtime caused by system problems. A self-healing infrastructure detects improper operation of system, transactions and processes, and initiates corrective action without disrupting users. IBM Tivoli System Automation offers mainframe-like high availability by using fast detection of outages and sophisticated knowledge about application components and their relationships. It provides quick and consistent recovery of failed resources and whole applications either in place or on another system of a Linux cluster or AIX cluster without any operator intervention. Thus it relieves operators from manual monitoring, remembering application components and relationships, and therefore eliminates operator errors.

Policy based automation

IBM Tivoli System Automation allows to configure high availability systems through the use of policies that define the relationships among the various components. These policies can be applied to existing applications with minor modifications. Once the relationships are established, IBM Tivoli System Automation will assume responsibility for managing the applications on the specified nodes as configured. This reduces implementation time and the need for complex coding of applications. In addition, systems can be added without modifying scripts, and resources can be easily added, too. There are sample policies available for IBM Tivoli System Automation . You can download them from the following web page <ftp://ftp.software.ibm.com/software/tivoli/products/sys-auto-linux/>.

Automatic recovery

IBM Tivoli System Automation quickly and consistently performs an automatic restart of failed resources or whole applications either in place or on another system of a Linux or AIX cluster. This greatly reduces system outages.

Automatic movement of applications

IBM Tivoli System Automation manages the cluster-wide relationships among resources for which it is responsible. If applications need to be moved among nodes, the start and stop relationships, node requirements and any preliminary or follow-up actions are automatically handled by IBM Tivoli System Automation. This again relieves the operator from manual command entry, reducing operator errors.

Resource grouping

Resources can be grouped together in IBM Tivoli System Automation. Once grouped, all relationships among the members of the group can be established, such as location relationships, start and stop relationships, and so on. After all configuration is completed, operations can be performed against the entire group as a single entity. This once again eliminates the need for operators to remember the application components and relationships, reducing the possibility of errors.

IBM Tivoli System Automation terms

This section gives an overview of the terms this manual uses when describing IBM Tivoli System Automation.

Cluster / peer domain

The group of host systems upon which IBM Tivoli System Automation manages resources is known as a cluster. A cluster can consist of one or more systems or nodes. Throughout this manual 'peer domain' is also used when referring to a cluster. The two terms are interchangeable. IBM Tivoli System Automation supports up to 32 nodes within a cluster.

Resource

A resource is any piece of hardware or software that can be defined to IBM Tivoli System Automation. These resources can be either defined manually by the administrator using the `mkrsrc` (make resource) command or through the 'harvesting' functionality of the cluster infrastructure, whereby resources are automatically detected and prepared for use. All resources are controlled through the appropriate resource managers as described in "Introducing resources managers provided by IBM Tivoli System Automation" on page 5. Resources have characteristics, or attributes, which can be defined. For example, when considering an IP address as a resource, attributes would include the IP address itself and the net mask. There are two types of resources: fixed resources and floating resources.

Fixed resource

A fixed resource is a resource that has only a single instance within the cluster. It represents one entity that is defined for a single node, and this is the only node on which it runs.

Floating resource

A floating resource is a resource which can run on several nodes in the cluster. You can find a detailed definition of a floating resource under "ResourceType attribute" on page 33.

Resource attributes

A resource attribute describes some characteristics of a resource. There are two types of resource attributes: persistent attributes and dynamic attributes.

Persistent attributes

The attributes of the IP address just mentioned (the IP address itself and the net mask) are examples of persistent attributes – they describe enduring characteristics of a resource. While you could change the IP address and net mask, these characteristics are, in general, stable and unchanging.

Dynamic attributes

Dynamic attributes, on the other hand, represent changing characteristics of the resource. Dynamic attributes of an IP address, for example, would identify such things as its operational state.

Resource class

A resource class is a collection of resources of the same type. For example, if an application is a resource, then all applications defined in the cluster would comprise a resource class. Resource classes allow you to define the common characteristics among the resources in its class. In the case of applications, the resource class can define identifying characteristics, such as the name of the application, and varying characteristics, such as whether or not the application is running. So each resource in the class can then be noted by its characteristics at any given time. Resource classes are managed by various resource managers – see "Introducing resources managers provided by IBM Tivoli System Automation" on page 5.

Resource group

Resource groups are logical containers for a collection of resources. This container allows you to control multiple resources as a single logical entity. Resource groups are the primary mechanism for operations within IBM Tivoli System Automation. Resource groups can also be nested, meaning that applications can be split into several resource groups which themselves are part of another higher level resource group. Also resource groups can be defined in such a way that their members can be located on different systems in the cluster.

Managed resource

A managed resource is a resource that has been defined to IBM Tivoli System Automation. To accomplish this, the resource is added to a resource group, at which time it becomes manageable through IBM Tivoli System Automation.

Nominal state

The nominal state of a resource group indicates to IBM Tivoli System Automation whether the resources with the group should be Online or Offline at this point in time. So setting the nominal state to "Offline" indicates that you wish for IBM Tivoli System Automation to stop the resources in the group, and setting the nominal state to "Online" is an indication you wish to start the resources in the resource group. You can change the value of the NominalState resource group attribute, but you cannot set the nominal state of a resource directly. See "NominalState attribute" on page 40.

Equivalency

An equivalency is a collection of resources that provides the same functionality. For example, equivalencies are used for selecting network adapters that should host an IP address. If one network adapter goes offline, IBM Tivoli System Automation selects another network adapter to host the IP address.

Relationships

IBM Tivoli System Automation allows the definition of relationships between resources in a cluster. There are two different relationship types:

- **Start-/stop relationships**

Relationships are used to define start and stop dependencies between resources. You can use the StartAfter, StopAfter, DependsOn, DependsOnAny, and ForcedDownBy relationships to achieve this. For example, a resource must only be started after another resource was started. You can define this by using the policy element StartAfter relationship.

- **Location relationships**

Location relationships are applied when resources must, or should if possible, be started on the same or a different node in the cluster. IBM Tivoli System Automation provides the following location relationships: Collocation, AntiCollocation, Affinity, AntiAffinity, and IsStartable. A simple example is that a webserver and its corresponding service IP address, which could be started on any node in the cluster, should always be kept together. In the past this behavior had to be defined by writing complex scripts. Now IBM Tivoli System Automation allows the usage of a location relationship which simplifies the policy definition for the administrator.

Relationships provide the following additional features:

- The possibility to define relationships between resource groups, resources, and equivalencies.
- The possibility to define relationships between resources running on different systems in the cluster.

Quorum

The main goal of quorum operations is to keep data consistent and to protect critical resources. Quorum can be seen as the number of nodes in a cluster that are required to modify the cluster definition or perform certain cluster operations. There are two types of quorum:

Introduction

Configuration quorum

This quorum determines when configuration changes in the cluster will be accepted. Operations affecting the configuration of the cluster or resources are only allowed when the absolute majority of nodes is online. See “Configuration quorum” on page 96 for a detailed description.

Operational quorum

This quorum is used to decide whether resources can be safely activated without creating conflicts with other resources. In case of a cluster splitting resources can only be started in the subcluster which has a majority of nodes or obtained a tie breaker. See “Operational quorum” on page 96 for a detailed description.

Tie breaker

In case of a tie in which a cluster has been partitioned into subcluster with an equal number of nodes, the tie breaker is used to determine which subcluster will have an operational quorum.

Components of IBM Tivoli System Automation

Reliable Scalable Cluster Technology, or RSCT, is a product fully integrated into IBM Tivoli System Automation. RSCT is a set of software products that together provide a comprehensive clustering environment for AIX and Linux. RSCT is the infrastructure to provide clusters with improved system availability, scalability, and ease of use.

RSCT provides three basic components, or layers, of functionality:

- **RMC** (Resource Monitoring and Control), provides global access for configuring, monitoring, and controlling resources in a peer domain.
- **HAGS** (High Availability Group Services), is a distributed coordination, messaging, and synchronization service.
- **HATS** (High Availability Topology Services), provides a scalable heartbeat for adapter and node failure detection, and a reliable messaging service in a peer domain.

Introducing resources managers provided by IBM Tivoli System Automation

Resource classes are managed by the various resource managers (RM), depending on what type of resource is being managed. A resource manager is a software layer between a resource and RMC. The following resource managers are provided by IBM Tivoli System Automation:

Recovery RM (IBM.RecoveryRM)

This resource manager serves as the decision engine for IBM Tivoli System Automation. Once a policy for defining resource availabilities and relationships is defined, this information is supplied to the Recovery RM. This RM runs on every node in the cluster, with exactly one Recovery RM designated as the master. The master evaluates the monitoring information from the various resource managers. Once a situation develops that requires intervention, the Recovery RM drives the decisions that result in start or stop operations on the resources as needed.

Global Resource RM

The Global Resource RM (IBM.GblResRM) supports two resource classes:

IBM.Application

The IBM.Application resource class defines the behavior for general application resources. This class can be used to start, stop, and monitor processes. As a generic class, it is very flexible and can be used to monitor and control various kind of resources. Most of the applications that you will automate will be done using this class. For more information, refer to “Using the Global Resource Manager” on page 129.

IBM.ServiceIP

This application class defines the behavior of Internet Protocol (IP) address resources. It allows you to assign IP addresses to an adapter. In effect, it allows IP addresses to ‘float’ among nodes. For more information, refer to “What is the IBM.ServiceIP resource class?” on page 137.

Configuration RM

The Configuration RM (IBM.ConfigRM) is used in cluster definition. In addition, quorum support, which is a means of insuring data integrity when portions of a cluster lose communication, is provided.

Event response RM

The Event Response RM (IBM.ERRM) provides the ability to monitor conditions in the cluster in order for the RMC system to react in certain ways.

Test RM

The Test resource manager (IBM.TestRM) manages test resources and provides functions to manipulate the operational state of these resources. The resource manager is operational in a peer domain mode only

Introduction

and provides the resource class IBM.Test. The Test resource manager does not control real resources. A detailed description of the Test RM is given in Chapter 12, “Resource managers provided by IBM Tivoli System Automation,” on page 129.

Figure 2 shows a diagram of the previously described components.

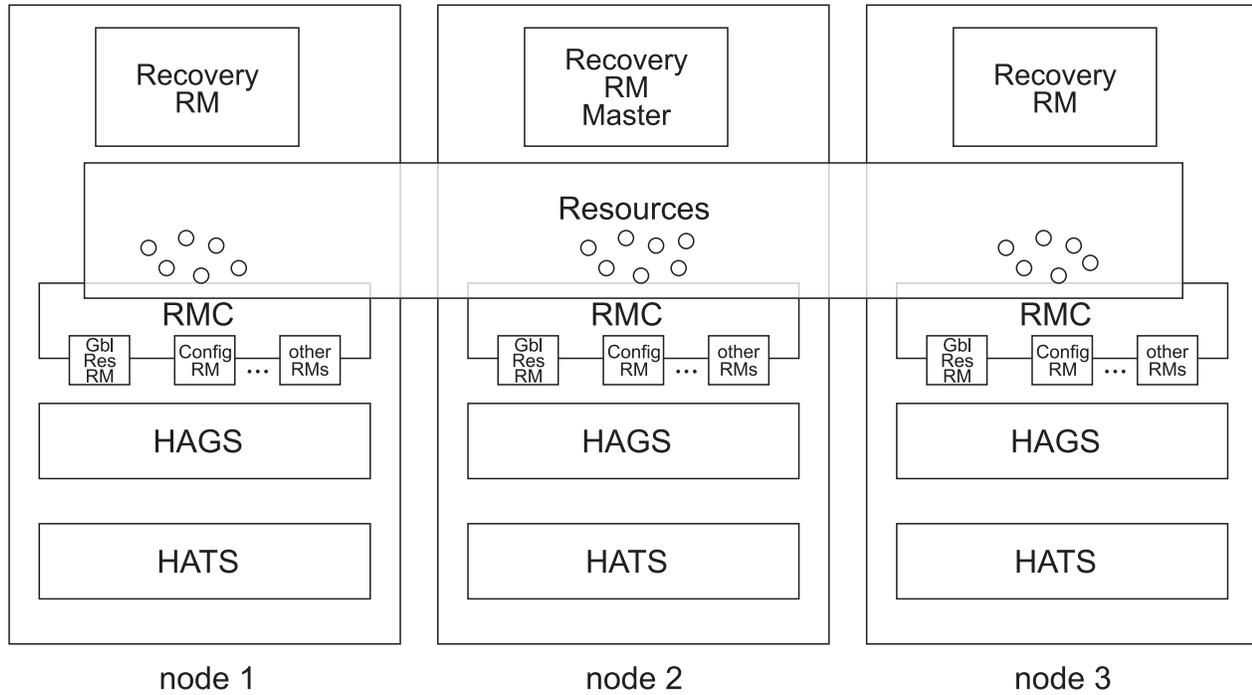


Figure 2. Overview of IBM Tivoli System Automation architecture

Chapter 2. Installing IBM Tivoli System Automation

This chapter describes how you install, configure, and migrate IBM Tivoli System Automation, in these main sections:

- “Planning for the installation.”
- “Preparing for installation” on page 9
- “Installing and upgrading IBM Tivoli System Automation for Multiplatforms” on page 11.
- “Uninstalling IBM Tivoli System Automation” on page 16
- Chapter 3, “Getting started,” on page 17

It also mentions how you can backup the current automation policy in section:

- “Installing service” on page 16

Planning for the installation

Coexistence with other products:

IBM Tivoli System Automation can coexist with General Parallel File System (GPFS) or Cluster Systems Management (CSM). If these products are installed, IBM Tivoli System Automation shares packages with those products. You can check if any of these packages is installed with the commands:

```
rpm -q gpfs
```

or

```
rpm -q csm
```

respectively.

If you find that GPFS prior to version 2.2 is installed, IBM Tivoli System Automation **cannot** be used with this version of GPFS at the same time.

Contents of the CD:

The CD labelled “IBM Tivoli System Automation for Multiplatforms” contains this manual, a README file, scripts, and software packages for each platform and the corresponding architecture.

Electronic distribution of IBM Tivoli System Automation

If you prefer electronic distribution to delivery on the CD, we offer you the possibility to download the product from the Web. After you have purchased IBM Tivoli System Automation you get an URL where you can download a tar file.

Supported platforms:

Version 1.2 of IBM Tivoli System Automation supports the Linux on the zSeries, xSeries, pSeries, iSeries, and AIX 5.2 platforms.

The following web site provides you with up-to-date information about supported platforms:

<http://www-3.ibm.com/software/tivoli/products/sys-auto-linux/platforms.html>.

Installation

Table 1. Supported platforms distributions

	xSeries (see Note)	zSeries	pSeries	iSeries
SUSE SLES 7 (32 Bit)		x		
SUSE SLSS/SLES 8 (32 Bit) United Unix 1.0	x	x		
SUSE SLES 7 (64 Bit)		x		
SUSE SLSS/SLES 8 (64 Bit) United Unix 1.0		x	x	x
SUSE SLES 9 (64 Bit)	x		x	
RedHat 7.2 (32 Bit)	x			
RedHat 7.3 (32 Bit)	x			
RedHat AS 2.1 (32 Bit)	x			
RedHat RHEL 3.0 (32 Bit)	x			
RedHat RHEL 3.0 (64 Bit)	x		x	x
AIX 5.2			x	

Note: xSeries and any other 32 Bit Intel based Servers or AMD Opteron (64Bit).

Supported network interfaces

All platforms support 10 Megabit Ethernet, Fast Ethernet, and Gigabit Ethernet. In addition, the zSeries platform also supports Hipersockets, CTC, and VM Guest LAN.

Preparing for installation

IBM Tivoli System Automation is contained in several packages which must be installed on every node in the cluster to be automated. The type of packages and content depends on the operation system:

Operating system	Type of package	Content
Linux	'rpm' stands for RedHat Packaging Manager. It manages installation and uninstallation of software packages in RPM format.	System Automation rpms and RSCT rpms. RSCT is the underlying infrastructure.
AIX	'installp' filesets	Only System Automation installp filesets. RSCT is part of AIX. However, a more recent level of RSCT may be required.

Note that the scripts *installSAM* and *uninstallSAM* are supplied to ensure that packages are installed or uninstalled in the correct order. The scripts and the RPM files must be made available on nodes where IBM Tivoli System Automation is to be installed. For example, you may use FTP to transfer the files from a PC (with the CDRom mounted) to the node. Also you may install the RPMs over a shared Network File System. You have to use the packages from the subdirectory according to the operating system and platform:

Table 2. Operating systems, platforms, and base directories

Operating system returned by command: 'uname -s'	Platform	Directory
Linux	iSeries	ppc (returned by command 'arch')
	pSeries	ppc (returned by command 'arch')
	xSeries	i386 (returned by command 'arch')
	zSeries	s390 (returned by command 'arch')
AIX	pSeries	AIX

Use command **uname -s** to determine the operating system.

In order to determine the platform for operating system Linux use command **arch** which returns the architecture equivalent to the directory.

Note: The directory <arch> containing the packages and scripts for the xSeries platform is named i386.

However, depending on the processor you are using, when issuing the arch command, you may get back values ranging from i386 to i686. On zSeries the directory is called s390. These packages are suitable for the s390 (zSeries 31 bit) and s390x (zSeries 64 bit) architecture.

Prerequisites

Before starting the installation you must fulfill these requirements:

- Install the Public Domain Korn Shell (pdksh) package (if not already done).
- Perl is required to use the command line interface of IBM Tivoli System Automation for Multiplatforms including native RSCT commands. It is per default installed on your Linux or AIX systems as part of the operating system, but if you are using IBM Tivoli System Automation in a language other than English, a special version of Perl may be required. Due to known problems with Perl 5.8.0 and how it handles UTF-8 encoded locales, some characters may not be properly displayed. This can occur on systems with Perl 5.8.0 installed, while using a UTF-8 encoded locale. When previous or subsequent versions of Perl are used, or non-UTF-8 encoded locales are used, this problem does not occur. AIX 5.2 uses Perl 5.8.0 and there is currently no opportunity to order a different version of Perl for that AIX release.

If you decide to upgrade your Perl version on a Linux distribution, perform the following steps:

Installation

1. Download Perl 5.8.1 source, referring <http://dev.perl.org/perl5/news/2003/perl-5.8.1.html> .
 2. Unzip and tar -xvf on any directory.
 3. Compile and install on the UTF-8 machine, referring the instruction provided with the downloaded files.
 4. Change the symbolic link pointing to the directory of the Perl version that is used by IBM Tivoli System Automation from: `/usr/sbin/rsct/perl5/bin/perl->/usr/bin/perl` to the directory where the new version of Perl is per default installed:
`/usr/sbin/rsct/perl5/bin/perl->/usr/local/bin/perl`.
- Set the following environment variable for all users of IBM Tivoli System Automation on all nodes: **CT_MANAGEMENT_SCOPE=2** (peer domain scope). You can set the variable permanently if you set it in the profile.
 - Also make sure that the directory **/usr/sbin** has at least 100 MB free space, and that the directory **/var** also provides at least 100 MB free space.
 - During installation of IBM Tivoli System Automation on AIX the correct level of RSCT will be checked and a higher level of RSCT may be required. If this is required for your systems, download and install the appropriate RSCT filesets from the AIX service center.
 - For other operating systems specific requirements, see the requirements web page at <http://www-306.ibm.com/software/tivoli/products/sys-auto-linux/requirements.html>.
 - For languages using the double-byte character set (DBCS), the Telnet dialog buffer must be large enough to ensure that long messages are properly displayed . If this is not the case, enlarge the Telnet dialog buffer.

Installing and upgrading IBM Tivoli System Automation for Multiplatforms

If this is a first time installation of the product, go to “Installing the product” below. If a previous version of IBM Tivoli System Automation is already installed, then there are some steps that have to be performed before the new version of IBM Tivoli System Automation can be installed. To perform a migration to a new version of the product, go to “Migrating the product” on page 13.

Installing the product

If you downloaded the tar file from the Internet, extract the file, using the following command:

```
tar -xvf <tar file>
```

If you got the product on a CD, mount the CD and change to the directory where the CD is mounted.

Now change to the appropriate directory for your platform according to Table 2 on page 9:

```
cd SAM12/<arch>
```

Install the product with the *installSAM* script:

```
./installSAM
```

Before installation starts, the License Agreement and the License Information is displayed. You can scroll forward line by line using the “Enter” key, and page by page using the “spacebar”, which is basically the “more” functionality in UNIX®. Once you have scrolled to the bottom of the License information file and you want to accept the License Information, type ‘y’. Any other input will cancel the installation.

For Linux, you may now issue the following command to see which packages have been installed:

```
rpm -qa | grep -E "^src|^rsct|^sam"
```

For AIX, you may now issue the following command to see which packages have been installed:

```
lslpp -l sam*
```

See the rpm man page for details about the rpm command.

Installing the product license

IBM Tivoli System Automation requires that a valid product license is installed on each system it is running on. The license is contained on the installation medium in the ‘license’ sub directory. The installation of the license is usually performed during the product installation process. In case this did not succeed, or you want to upgrade from a Try & Buy license to a full license of the product, issue the following command to install the license:

```
samlcm -i license_file
```

In order to display the license, issue:

```
samlcm -s
```

See “samlcm” on page 219 for a detailed description of the **samlcm** command.

Languages supported by IBM Tivoli System Automation

This section is only of interest for you if you want to use IBM Tivoli System Automation for Multiplatforms in a language other than English as shown in the following tables.

The following encoding is supported for the Linux distribution:

Installation

Language	UTF-8	ISO-8859-1	EUC/GBK	Euro	GB18030/BIG5
German	de_DE.UTF-8	de_DE, de_DE.ISO-8859-1		de_DE@euro	
Spanish	es_ES.UTF-8	es_ES, es_ES.ISO-8859-1		es_ES@euro	
French	fr_FR.UTF-8	fr_FR, fr_FR.ISO-8859-1		fr_FR@euro	
Italian	it_IT.UTF-8	it_IT, it_IT.ISO-8859-1		it_IT@euro	
Japanese	ja_JP.UTF-8		ja_JP.eucJP		
Korean	ko_KR.UTF-8		ko_KR.eucKR		
Portugese/ Brazilian	pt_BR.UTF-8	pt_BR			
Simplified Chinese	zh_CN.UTF-8		zh_CN.GBK, zh_CN.GB2312		zh_CN.GB18030
Traditional Chinese	zh_TW.UTF-8		zh_TW.eucTW (only supported under RedHat AS 2.1)		zh_TW.Big5, zh_TW

The following encoding is supported on the AIX distribution:

Language	UTF-8	ISO-8859-1	EUC/GBK	SJIS/GB18030/BIG5
German	DE_DE	de_DE		
Spanish	ES_ES	es_ES		
French	FR_FR	fr_FR		
Italian	IT_IT	it_IT		
Japanese	JA_JP		ja_JP	Ja_JP
Korean	KO_KR		ko_KR	
Portugese/Brazilian	PT_BR	pt_BR		
Simplified Chinese	ZH_CN		zh_CN	Zh_CN
Traditional Chinese	ZH_TW		zh_TW	Zh_TW

Be aware that you have to carry out the following steps if you are both using a SUSE LINUX distribution and a language other than English:

- Start YaST2.
- Select "System" icon from a list.
- Select "Editor for /etc/sysconfig" from a pane.
- Select "Base-Administration" from a list. Click the "+" icon.
- Select "Localization" from a list. Click the "+" icon.
- Select "rc_lang" from the list and set a correct locale from the locale table to RC_LANG parameter.
- Select "root_uses_lang" from a list. Set "yes" to ROOT_USES_LANG parameter.
- Press the "Save" button. When the "Save sysconfig variables" dialog box appears, press the "OK" button.
- Restart the system.

In order to verify that your system is set to the locales supported by this product (reference our locale support tables), perform the following steps:

1. Log in as root and issue the following command:

```
locale
```

Verify that the LANG value is listed on the language of your choice.

2. If the returned values are not set to a locale that is supported (reference our locale support tables) or set to POSIX, continue with the following steps:
3. Issue the following command:

```
export LANG=xx_XX
```

You have to choose a locale which can be displayed by your terminal.

4. In order to verify the terminal has been set to the locale you wanted, issue this command:

```
locale
```

and make sure LANG is set to **xx_XX**.

5. Proceed with regular product tasks.

You need to repeat step 3 to step 5 each time you start new terminal window in order to issue IBM Tivoli System Automation commands.

Migrating the product

If IBM Tivoli System Automation 1.1 is already installed, the product can be migrated to the new version IBM Tivoli System Automation 1.2.

Before migrating consider the following:

- The migration process starts when any node within the active cluster is upgraded to the higher version code.
- You can always upgrade from a lower code level to a higher code level, but a downward migration is not possible.
- The migration process is only complete when the active version number is equal to the highest installed code version number. Until then, different code levels can coexist. See “Verifying the active and installed version number” on page 14 and “Completing migration” on page 15 how to complete the migration process.

You can use one of the following ways to migrate IBM Tivoli System Automation, but we recommend to use the procedure as described under “Migrating an entire domain” on page 13:

Migrating an entire domain

Keep the following in mind when migrating an entire domain:

1. The domain will not be available for automation during the upgrade.
2. Stop all online resource groups by setting their NominalState to Offline:

```
chrg -o Offline <resource-group-name>
```

3. If the domain is online, stop the domain:

```
stoprpdomain <domain-name>
```

4. Run **installSAM** on all nodes.

5. Start the domain:

```
startrpdomain <domain-name>
```

6. Check the code levels with the **lssrc -ls IBM.RecoveryRM** command (see sample in “Verifying the active and installed version number” on page 14). All the nodes should have the newly installed code level, but the active code level should still be the previous one.

Installation

7. In order to activate the new version continue with “Completing migration” on page 15.

Migrating a node step by step

This has the advantage that IBM Tivoli System Automation is still available during migration. Keep the following in mind when migrating a node step by step:

1. Make sure that the node to be migrated is excluded from automation, so that resources of other nodes are activated.

```
samctrl -u a <node>
```

Note that if a resource group was running on the node to be excluded, automation will try to move it to another node. This may take a little while.

2. Stop the node, and verify that it is stopped:

```
stoprnode <node>
```

3. Run **installSAM** to upgrade the node.

4. Start the node:

```
startnode <node>
```

5. Take the newly upgraded node back to automation:

```
samctrl -u d <node>
```

6. The newly upgraded node can now join the existing cluster. Use the **lssrc -ls IBM.RecoveryRM** command (see sample in “Verifying the active and installed version number”) to display the installed version and the active version of the product. The new code features will not be activated until the active IBM Tivoli System Automation version number is equal to the highest IBM Tivoli System Automation version number installed within the cluster, and you cannot fully utilize these new code features until all the nodes are upgraded.

7. Repeat the steps 1-6 for other nodes within the cluster.

8. In order to activate the new version continue with “Completing migration” on page 15.

Verifying the active and installed version number

After the upgrade the new features of the new code are not yet activated. The previous and new code levels can coexist until the migration is completed. The **lssrc -ls IBM.RecoveryRM** command shows you the active version number **AVN**(1.2.0.0 in the sample below) and the installed version number **IVN** (1.2.0.0 in the sample below) of the product. When IVN and AVN are the same, migration is complete. The output looks like:

```
Subsystem      : IBM.RecoveryRM
PID            : 27973
Cluster Name   : ws
Node Number    : 1
Daemon start time : Wed Apr 21 08:09:10 2004
```

Daemon State:

```
My Node Name      : lnxcm3x
Master Node Name  : lnxcm3x (node number = 1)
Our IVN           : 1.2.0.0
Our AVN           : 1.2.0.0
Our CVN           : 11082527751 {0x140861007}
Total Node Count  : 1
Joined Member Count : 1
Config Quorum Count : 1
Startup Quorum Count : 1
Operational Quorum State: HAS_QUORUM
In Config Quorum   : TRUE
In Config State    : TRUE
Replace Config State : FALSE
```

In order to activate the new version continue with “Completing migration” on page 15.

Completing migration

In order to check and finish migration perform the following steps:

1. Issue the **lsrpdomain** command to see the current RSCT active version number and mixed version status:

```
Name           OpState RSCTActiveVersion  MixedVersions  TSPort  GSPort
SA_Domain      Online  2.3.2.1             Yes            12347  12348
```

2. Issue the **lsrpnod**e command to see the current RSCT install version number on all nodes. All nodes must be online:

```
Name  OpState RSCTVersion
node01 Online  2.3.3.0
node02 Online  2.3.3.0
node03 Online  2.3.2.1
```

3. If the RSCT Peer Domain is running under mixed version mode (MixedVersions = Yes), issue the following command on one of the nodes. Keep in mind that all nodes must be online.

```
runact -c IBM.PeerDomain CompleteMigration Options=0
```

This upgrades the RSCT Active Version after all the nodes have been upgraded to the new release of IBM Tivoli System Automation. See the additional RSCT migration preparation procedures described in Chapter 3 of the *IBM RSCT Administration Guide* before you start the RSCT CompleteMigration action.

4. Run the **samctrl -m** command (see “samctrl” on page 213) to activate the new features of the new code and finish migration.

The code version of the ActiveVersion and the InstalledVersion of IBM Tivoli System Automation should now be the same for all nodes. Until this is true, the new code features have not been activated and cannot be used.

Uninstalling IBM Tivoli System Automation

Use the *uninstallSAM* script that is provided for your operating system to uninstall IBM Tivoli System Automation. For example, run *uninstallSAM* from the AIX sub directory for AIX systems and from the s390 sub directory for your Linux on zSeries systems. This will ensure a proper deinstallation of the product. Before uninstalling you should save your configuration with the **samcfg -S** command. See “Automation policy management” on page 114 and “samcfg” on page 211 how to save IBM Tivoli System Automation.

Note: This will also remove all configuration information that you defined for the cluster. Never use *uninstallSAM* before you want to upgrade to a new version.

Check if a domain is still online by entering the command:

```
lsrpdomain
```

In order to stop a domain enter the command:

```
stoprpdomain <domain>
```

Uninstall the product with the *uninstallSAM* script:

```
./uninstallSAM
```

If CSM or GPFS (which also use RSCT and System Resource Controller (SRC) packages) is installed on a Linux system from which you want to uninstall IBM Tivoli System Automation, RPM will ensure that RSCT and SRC will not be uninstalled with IBM Tivoli System Automation. RPM messages will indicate this.

If you want to verify which packages were uninstalled for the Linux operating system, use the following command :

```
rpm -qa | grep -E "^src|^rsct|^sam"
```

If you want to verify which packages were uninstalled for the AIX operating system, use the following command :

```
lslpp -l sam*
```

Any packages left installed will be listed. If no packages required by other products are left installed, no packages will be listed.

Installing service

Installing service means upgrading IBM Tivoli System Automation from release 1.2.0.0. Therefore, release 1.2.0.0 must have been installed before any service can be applied. First backup your system configuration. See “Automation policy management” on page 114 how to do this. Then apply the following steps on each node in the peer domain:

1. Check if any resources are online on the node you want to service.
2. If the resources are online and must be kept available, exclude the node from automation using the command

```
samctrl -u a Node
```

The resources will then be restarted on other nodes in the peer domain.

3. If the resources need not be kept available during service, set the resource groups offline.
 4. Run the same steps as explained in “Installing the product” on page 11.
 5. If you had excluded the node in step 2, include the node into automation using the command
- ```
samctrl -u d Node
```
6. If you require the resource groups to be online set the resource groups online. Otherwise delay this step until after the last node in the peer domain is serviced.

---

## Chapter 3. Getting started

This chapter lists and describes the steps shown below which you have to perform to start IBM Tivoli System Automation:

### Step 1: Defining and administering a cluster

This step shows how you can create and remove a cluster, how you can add nodes to a cluster and remove nodes from a cluster, and how you can check the status of the IBM Tivoli System Automation daemon.

### Step 2: Defining RSCT resources

This step shows how you can create a resource like a web server and how you can create an equivalency relationship.

### Step 3: Defining the automation policy

This step shows you can define the relationships among the components created in Step 1 and Step 2. This is called defining the automation policy.

Before beginning to create a cluster you should ensure that your network setup is correct:

- IP, netmask and broadcast addresses must be consistent on each cluster node.
- Make sure that the name resolution is correct, DNS entries are consistent or entries in your local */etc/hosts* files on all nodes are identical.
- Do not define more than one network interface on a node to the same subnet.

See Chapter 10, “Setting up a high available network,” on page 105 for more details.

The following gives you an overview of the *Reliable Scalable Cluster Technology (RSCT) for Linux* commands you will use when working with cluster definitions. You will need some of these commands when you are going through Step 1 and Step 2.

|                                     |                                                                                                                                                                                                                                                                                      |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>preprnode</b>                    | This command prepares the security settings for the node to be included in a cluster. When issued, public keys are exchanged among the nodes, and the RMC access control list (ACL) is modified to enable access to cluster resources by all the nodes of the cluster.               |
| <b>mkrpdomain</b>                   | This command creates a new cluster definition. It is used to specify the name of the cluster, and the list of nodes to be added to the cluster.                                                                                                                                      |
| <b>lsrpdomain</b>                   | This command lists information about the cluster to which the node where the command runs belongs.                                                                                                                                                                                   |
| <b>startprdomain / stopprdomain</b> | These commands are used to bring the cluster online and offline, respectively.                                                                                                                                                                                                       |
| <b>addrpnode</b>                    | Once a cluster has been defined and is operational, this command is used to add new nodes to the cluster.                                                                                                                                                                            |
| <b>startprnode / stopprnode</b>     | These commands are used to bring individual nodes online and offline to the cluster. They often used when performing maintenance to a particular system. The node is stopped, repairs or maintenance is performed, then the node is restarted, at which time it rejoins the cluster. |
| <b>lsrpnode</b>                     | This command is used to view the list of nodes defined to a cluster, as well as the operating state (OpState) of each node. Note that this command is useful only on nodes that are Online in the cluster, otherwise it will not display the list of nodes.                          |
| <b>rmrpdomain</b>                   | This command removes a defined cluster.                                                                                                                                                                                                                                              |
| <b>rmrpnode</b>                     | This command removes one or more nodes from a cluster definition.                                                                                                                                                                                                                    |

## Getting started

For detailed descriptions of these commands, refer to the appropriate man pages or to these manuals, which you can find on the IBM Tivoli System Automation CD:

- *IBM Reliable Scalable Cluster Technology for Linux, Administration Guide*, SA22-7892.
- *IBM Reliable Scalable Cluster Technology for Linux, Technical Reference*, SA22-7893.
- *IBM Reliable Scalable Cluster Technology for AIX 5L™: Administration Guide.*, SA22-7889.
- *IBM Reliable Scalable Cluster Technology for AIX 5L: Technical Reference.*, SA22-7890.

You can find these documents at this IBM Web site:

<http://www-1.ibm.com/servers/eserver/clusters/library/>

Chapter 13, “IBM Tivoli System Automation commands,” on page 145 provides a complete list and description of the IBM Tivoli System Automation commands. You will use some of these commands in Step 3.

## Step 1: Defining and administering a cluster

The following scenarios show how you can create a cluster, add nodes to the cluster, and how you can check the status of the IBM Tivoli System Automation daemon (IBM.RecoveryRM).

### Creating a two nodes cluster

To create this cluster, you need to:

1. Access a console on each node in the cluster and log in as root.
2. Set the environment variable `CT_MANAGEMENT_SCOPE=2` on each node.
3. Issue the **preprnode** command on all nodes to allow communication between the cluster nodes.

```
preprnode node01 node02
```

4. You can now create a cluster with the name **SA\_Domain** running on node01 and node02. The following command can be issued from any node.

```
mkrpdomain SA_Domain node01 node02
```

Note that when creating RSCT peer domains (clusters) using **mkrpdomain**, the characters used for the peer domain name are limited to the following ASCII characters: A-Z, a-z, 0-9, . (period), and \_(underscore).

5. To look up the status of **SA\_Domain**, issue the **lsrpdomain** command:

```
lsrpdomain
```

Output:

| Name      | OpState | RSCTActiveVersion | MixedVersions | TSPort | GSPort |
|-----------|---------|-------------------|---------------|--------|--------|
| SA_Domain | Offline | 2.3.3.0           | No            | 12347  | 12348  |

The cluster is defined but offline.

6. Issue the **starttrpdomain** command to bring the cluster online.

```
starttrpdomain SA_Domain
```

When you run the **lsrpdomain** command again, you see that the cluster is still in the process of starting up, the OpState is Pending Online.

| Name      | OpState        | RSCTActiveVersion | MixedVersions | TSPort | GSPort |
|-----------|----------------|-------------------|---------------|--------|--------|
| SA_Domain | Pending online | 2.3.3.0           | No            | 12347  | 12348  |

After a short time the cluster will be started, so when issuing the **lsrpdomain** again, you see that the cluster is now online:

| Name      | OpState | RSCTActiveVersion | MixedVersions | TSPort | GSPort |
|-----------|---------|-------------------|---------------|--------|--------|
| SA_Domain | Online  | 2.3.3.0           | No            | 12347  | 12348  |

#### Notes:

1. You may get an error message like:

**2632-044** The domain cannot be created due to the following errors that were detected while harvesting information from the target nodes:

node1: **2632-068** This node has the same internal identifier as node2 and cannot be included in the domain definition.

This error most often occurs if you have cloned Linux images.

Something went wrong with the cluster and the entire configuration should be reset. Solve such problems by running the

```
/usr/sbin/rsct/install/bin/recfgct
```

command on the node which is named in the error message in order to reset the node id. Continue with the **preprnode** command.

## Getting started

2. You may also get an error message like:

```
2632-044 The domain cannot be created due to the following errors that were
detected while harvesting information from the target nodes:
node1: 2610-418 Permission is denied to access the resources or resource class specified
in this command.
```

Check your hostname resolution. Make sure that all entries for each node of the cluster in your local */etc/hosts* files on all nodes and the nameserver entries are identical.

## Adding a node to an existing cluster

After having created a two nodes cluster, you might want to add a third node to **SA\_Domain**. In order to do this, you need to:

1. Issue the **lsrpdomain** command to see if your cluster is online:

```
Name OpState RSCTActiveVersion MixedVersions TSPort GSPort
SA_Domain Online 2.3.3.0 No 12347 12348
```

Issue the **lsrpnode** command to see which nodes are online:

```
Name OpState RSCT Version
node02 Online 2.3.3.0
node01 Online 2.3.3.0
```

2. Issue the following **preprnode** commands to allow communication between the existing nodes and the new node.

Log on to node03 and enter:

```
preprnode node01 node02
```

Log on to node02 and enter:

```
preprnode node03
```

Log on to node01 and enter:

```
preprnode node03
```

You are strongly recommended to issue a **preprnode** command on each node for all nodes.

3. In order to add the node03 to the cluster definition, issue the **addrpnode** command on node01 or node02, which are already online on the cluster.

```
addrpnode node03
```

Again issue the **lsrpnode** command to see the status of all nodes:

```
Name OpState RSCT Version
node02 Online 2.3.3.0
node03 Offline 2.3.3.0
node01 Online 2.3.3.0
```

4. Start node03 from an online node:

```
starttrnode node03
```

After a short time node03 should be online, too.

## Getting started

### Taking an entire cluster or individual nodes offline

In order to perform node maintenance or make application upgrades, you might want to take an entire cluster or individual nodes of a cluster offline:

- In order to perform maintenance on cluster **SA-Domain**, you might wish to take it offline. Use the **stoprpdomain** command from any online node in the cluster to do this.

```
stoprpdomain SA_Domain
```

Issue the **lsrpdomain** command to check the status of cluster **SA-Domain**:

```
Name OpState RSCTActiveVersion MixedVersions TSPort GSPort
SA_Domain Offline 2.3.3.0 No 12347 12348
```

Stopping a cluster does not remove the cluster definition: the cluster can therefore be brought back online using the **startrpdomain** command.

- To take one or more cluster *nodes* offline, you use the **stoprpnode** command. You might need to do this to perform application upgrades, to perform maintenance on a node, or before removing the node from the cluster. Also, since a node may be defined in multiple clusters, but online in only one cluster at a time, you might need to take a node offline in one cluster so that you may bring it online in another cluster. To take a node offline, issue the **stoprpnode** command from any online node in the cluster, and pass to it the cluster node name of the node to take offline. For example, do this to stop node03:

```
stoprpnode node03
```

**Note:** Be careful when stopping multiple nodes of a cluster. You will lose quorum if less than half of the nodes are online. This may lead to outages if there are resources running on the online nodes of the cluster. See Chapter 9, “Protecting your resources – quorum support,” on page 95 for more information.

Issue the **lsrpnode** command to see if node03 has gone offline:

```
lsrpnode node03
```

```
Name OpState RSCT Version
node03 Offline 2.3.3.0
```

## Removing nodes from a cluster, or removing a complete cluster

When upgrading hardware or otherwise reorganizing your overall cluster configuration, you may need to remove individual nodes from a cluster, or remove an entire cluster definition.

- To remove a node from a cluster, you use the **rmrpnode** command. In order to remove a node from a cluster, the node must be offline. If the node you wish to remove is not currently offline, you must use the **stoprpnode** command to take it offline. You can also remove *multiple* nodes from the cluster, using the **rmrpnode** command. In order to see which node is offline, issue the **lsrpnode** command from any online node in the cluster.

```
lsrpnode
```

| Name   | OpState | RSCT Version |
|--------|---------|--------------|
| node02 | Online  | 2.3.3.0      |
| node03 | Offline | 2.3.3.0      |
| node01 | Online  | 2.3.3.0      |

Then issue the **rmrpnode** command from any online node in the cluster to remove node03.

```
rmrpnode node03
```

Issue the **lsrpnode** command again to see if node03 has been removed.

```
lsrpnode
```

| Name   | OpState | RSCT Version |
|--------|---------|--------------|
| node02 | Online  | 2.3.3.0      |
| node01 | Online  | 2.3.3.0      |

- To remove a complete cluster definition, you use the **rmrpdomain** command. Removing a cluster involves removing the cluster definition from each node on the cluster. To do this efficiently, all nodes in the cluster should be online. You can bring individual nodes online using the **startrpnode** command, or you can bring *all* offline nodes in the cluster online using the **startrpdomain** command. The **rmrpdomain** command removes the cluster definition on all of the nodes that are reachable from the node where the command was issued. If the command is issued from an online node in a cluster, and all the nodes are online, then the command will attempt to remove all of their cluster definition files. If a node is not reachable from the node where the **rmrpdomain** is run (for example, the node is offline or inoperative), the **rmrpdomain** command will not be able to remove the cluster definition on that node. In case the cluster cannot be brought online, you can use the force option **-f** to remove nodes or the cluster.

Issue the **startrpdomain** command to bring all nodes of cluster **SA\_Domain** online:

```
startrpdomain SA_Domain
```

Then issue the **rmrpdomain** command to remove cluster **SA\_Domain**:

```
rmrpdomain SA_Domain
```

## Getting started

### Administering the recovery resource manager

On each online node in the cluster an IBM Tivoli System Automation daemon (IBM.RecoveryRM) is running. You can check the status and the process id of the daemon with the command `lssrc`:

```
lssrc -s IBM.RecoveryRM
```

You get the following output:

| Subsystem      | Group   | PID   | Status |
|----------------|---------|-------|--------|
| IBM.RecoveryRM | rsct_rm | 18283 | active |

This daemon runs on each node which is online in the cluster. It is started automatically if the cluster node starts. If necessary, you can manually stop the daemon with the command:

```
stopsrc -s IBM.RecoveryRM
```

To start the daemon you use:

```
startsrc -s IBM.RecoveryRM
```

One of the daemons is the so called 'master daemon'. This daemon is responsible for driving all the necessary decisions. You can find out the node the master daemon is located on with the command:

```
lssrc -ls IBM.RecoveryRM | grep Master
```

and you get the following output:

```
Master Node Name : node03 (node number = 3)
```

In the example the master daemon runs on a node called node03.

The other daemons are called 'peer daemons'. These peer daemons are a hot standby if the master daemon or the node the master daemon is located on runs into problems. In this case, one of the peer daemons becomes the master. Of course, the takeover between the daemons is done without interruption of the automation functionality of IBM Tivoli System Automation.

## Step 2: Defining RSCT resources

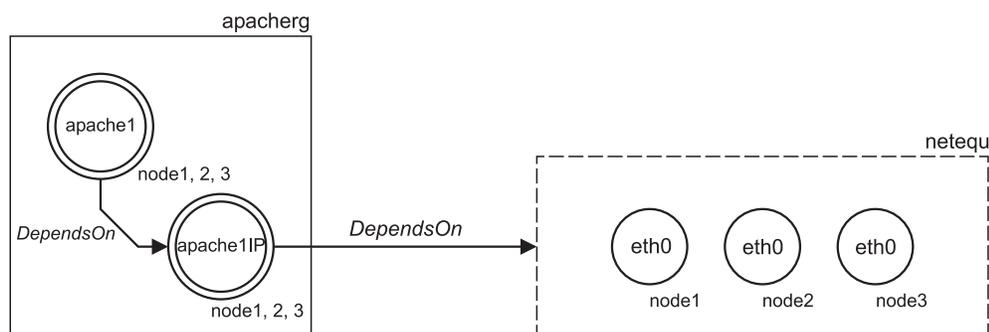
The following example shows how to define a high-available web server on the three nodes of the cluster SA\_Domain. See “Step 1: Defining and administering a cluster” on page 19 how this cluster and the nodes node01, node02, and node03 were defined.

The following requirements have to be met for the high-available web server::

- The web server should be startable on any node in the cluster, but will only run on one node at any point in time.
- The web server should be restarted automatically on the same or another node in the cluster in case of a failure. This mechanism also allows a planned outage of nodes for service and maintenance.
- The web server should be addressable with the same IP address regardless of the node it currently runs on. Thus the location of the web server is transparent outside the cluster where no adaption has to be performed, when the web server is moved from one node to another.

As the base for automation, the components involved must first be described in a set of RSCT defined resources. Due to often uncommon characteristics of resources, there are various RSCT resource classes to accommodate the differences. In this example we will need to define three RSCT resources from different classes:

1. An application resource named **apache1**, which represents the web server daemon. The resource is from a class called IBM.Application. **apache1** will be a floating resource, since the web server is not tied to a specific node in the cluster.
2. An IP address named **apache1IP**, which is used to represent the web server’s IP address. **apache1IP** is from a class called IBM.ServiceIP. **apache1IP** will be a floating resource, since it can move around in the cluster following the location of the web server.
3. An representation named **netequ** for the network interface cards that can be used for the **apache1IP** address. This is called an equivalency and belongs to a class IBM.Equivalency. The characteristic “floating” or “fixed” is not meaningful for this class.



### Creating application resource apache1

As part of the definition of application resource **apache1**, commands or scripts for starting, stopping and querying the web server have to be specified. These commands and/or scripts can be different ones, but it is often convenient to gather these functions in a single script, which has a command line parameter to select start/stop/status actions. These scripts will often be user-written. See Chapter 12, “Resource managers provided by IBM Tivoli System Automation,” on page 129 for details on the requirements of such scripts.

In this example, we use a script

```
/cluster/scripts/apache
```

which has the following content for a Linux system:

## Getting started

```
#!/bin/bash

OPSTATE_ONLINE=1
OPSTATE_OFFLINE=2

Action=${1}

case ${Action} in
 start)
 /usr/sbin/apachectl start >/dev/null 2>&1
 logger -i -t "SAM-apache" "Apache started"
 RC=0
 ;;
 stop)
 /usr/sbin/apachectl stop >/dev/null 2>&1
 logger -i -t "SAM-apache" "Apache stopped"
 RC=0
 ;;
 status)
 ps -ax |grep -v "grep"|grep "/usr/sbin/httpd">/dev/null
 if [$? == 0]
 then
 RC=${OPSTATE_ONLINE}
 else
 RC=${OPSTATE_OFFLINE}
 fi
 ;;
esac
exit $RC
```

Make sure to make the script accessible on all nodes with the same directory path.

RSCT resource definitions are created with the command **mkrsrc**. All resource characteristics can be provided in command line parameters, but the **mkrsrc** command also accepts a definition file in plain text. We will use the second approach with a definition file named `apache1.def`, which may look like the following:

```
PersistentResourceAttributes::
 Name="apache1"
 StartCommand="/cluster/scripts/apache start"
 StopCommand="/cluster/scripts/apache stop"
 MonitorCommand="/cluster/scripts/apache status"
 MonitorCommandPeriod=5
 MonitorCommandTimeout=5
 NodeNameList={"node01","node02","node03"}
 StartCommandTimeout=10
 StopCommandTimeout=10
 UserName="root"
 ResourceType=1
```

The resource definition can now be created with the **mkrsrc** command using the definition file.

```
mkrsrc -f apache.def IBM.Application
```

## Creating IP address resource apache1IP

The web server's IP address `apache1IP` is a separate IP address in the cluster and does not match any IP address assigned to the network adapters on each cluster node, that are made in system definitions outside of IBM Tivoli System Automation. The address for `apache1IP` is in contrast created by IBM Tivoli System Automation and is an additional alias address on an appropriate network adapter on the node where the web server resides. When the web server moves to a new location, the alias address is removed from the former node and recreated on the new node, where the web server is about to be restarted.

In this example `apache1IP` has the following attributes:

- IP 9.152.172.11
- Netmask 255.255.255.0
- The IP address may be created on any node in the cluster.

This time, we use command line parameters to the `mkrsrc` command to create the `apache1IP` resource:

```
mkrsrc IBM.ServiceIP \
 NodeNameList="{ 'node01', 'node02', 'node03' }" \
 Name="apache1IP" \
 NetMask=255.255.255.0 \
 IPAddress=9.152.172.11
```

Note that the command shown is split onto separate lines for readability only. In fact, the `apache1IP` has more attributes than the ones specified in the command shown. We leave the rest of them to their default values, such as the `ResourceType` attribute, which marks the resource as “floating” by default.

Also note that the managed resources are not started/stopped by a third party like, for example, the Linux run level or manually by the operator.

## Creating an equivalency for the network adapters

When a node in the cluster has multiple network attachments, not all of them might be equally suited to host the `apache1IP` address as an alias. An equivalency definition will specify the network adapters that can be used to carry the `apache1IP` address. Equivalency means that each of the adapters in the equivalency can provide the same required function regardless of its own unique characteristics. Since the web server should be startable on each node in the cluster, at least one of the adapters on each node has to appear in the equivalency.

An equivalency groups together a set of resources from another class. Network adapters belong to a class named `IBM.NetworkInterface`. There is no need to provide resource definitions for all the network adapters on the cluster nodes, since RSCD has a harvesting function which automatically creates appropriate resource definitions for many system defined resources.

The following command creates an equivalency named **netequ**, which contains a network adapter from each node of the cluster:

```
mkequ netequ IBM.NetworkInterface:eth0:node01,eth0:node02,eth0:node03
```

### Step 3: Defining the automation policy

The following examples show, how the resources **apache1** and **apache1IP** are turned into IBM Tivoli System Automation managed resources (see “What is a managed resource?” on page 32) providing high availability for a web server in a cluster environment. See “Step 2: Defining RSCT resources” on page 25 how the resources **apache1** and **apache1IP** were defined.

To turn resources **apache1** and **apache1IP** into managed resources, they have to be added to a resource group. When this has been done, IBM Tivoli System Automation starts controlling the resource group and its included resources.

In most cases, it is not enough to automate a managed resource on its own, because the resources are often related to each other. For instance, both resources **apache1** and **apache1IP** from our example must be made available on the same node. Such dependencies between managed resources must be described and defined with managed relationships (see “What is a managed relationship?” on page 53).

At last, an automation goal has to be provided, that is: should a managed resource be available/started in the cluster or should it be offline.

### Creating a resource group

A resource group is created with the **mkrp** command. The following command creates a resource group named **apacherg**:

```
mkrp apacherg
```

Both resources **apache1** and **apache1IP** will be added to the resource group **apacherg**. This is done with the **addrgrp** command. Adding the resources to the resource group turns them into managed resources:

```
addrgrp -g apacherg IBM.Application:apache1
addrgrp -g apacherg IBM.ServiceIP:apache1IP
```

### Defining relationships

There are two conditions that relate resources **apache1** and **apache1IP** to one another. First, both resources must be started/available on the same node in the cluster. This is called a collocated relationship (see “Collocated relationship” on page 71). Furthermore, it is of no use to start the web server **apache1** on a node, on which the IP address **apache1IP** has not been established yet. That is: **apache1IP** must be available before **apache1** can be started.

IBM Tivoli System Automation provides a relationship type called **DependsOn** (see “DependsOn relationship” on page 62) that gathers both required conditions. A managed relationship is defined with the **mkrel** command. The following command creates a managed relationship named **apache1\_dependson\_ip1** that establishes the dependency of resource **apache1** on the IP address **apache1IP**:

```
mkrel -p DependsOn -S IBM.Application:apache1 -G IBM.ServiceIP:apache1IP apache1_dependson_ip1
```

Our example needs a second relationship. In “Step 2: Defining RSCT resources” on page 25 we have created an equivalency of those network adapters, that can be used for aliasing the **apache1IP** address. This will be described in a second relationship called **apache1IP\_dependson\_netequ**, that ties **apache1IP** and the equivalency **netequ** together:

```
mkrel -p DependsOn -S IBM.ServiceIP:apache1IP -G IBM.Equivalency:netequ apache1IP_dependson_netequ
```

### Bringing a resource group online

When resources are added to resource groups, they become managed resources with a default automation goal of offline. This can be changed at the level of the resource group with the **chrg** command. To bring resource group **apacherg** online use the command:

```
chrg -o online apacherg
```

After your clusters and nodes have been created and configured, you can begin to use IBM Tivoli System Automation commands to:

- make, remove, change, and list *resource groups*. See Chapter 5, “Using resource groups,” on page 35 for details.
- make, remove, and change the resource group *member resources*. See Chapter 4, “Using resources,” on page 31 for details.
- make, remove, and change *equivalency resources*. See Chapter 6, “Using equivalencies,” on page 49 for details.
- make, remove, change, and list *managed relationship* resources. See Chapter 7, “Using managed relationships,” on page 53 for details.

In addition, a *complete list* of IBM Tivoli System Automation commands is provided in Chapter 13, “IBM Tivoli System Automation commands,” on page 145.



---

## Chapter 4. Using resources

This chapter describes how you use *resources*, in these main sections:

- “What is a resource?”
- “What is a managed resource?” on page 32
- “Attributes used by resources” on page 32

These are the IBM Tivoli System Automation commands that you use together with managed resources:

Table 3. IBM Tivoli System Automation commands used with managed resources

| Command  | Description                                                                 | For details, see page: |
|----------|-----------------------------------------------------------------------------|------------------------|
| addrgmbr | Add one or more resources to a resource group                               | 147                    |
| chrgmbr  | Change persistent attribute value of a managed resource in a resource group | 162                    |
| lsrg     | List an already-defined resource group or its member resources              | 173                    |
| rmmrgmbr | Remove one or more resources from the resource group                        | 208                    |

---

### What is a resource?

A *resource* is any piece of hardware or software that has been defined to IBM’s RMC (*Resource Monitoring and Control*) using either the:

- RMC **mkrsrc** (“Make Resource”) command.
- “harvesting” function of RMC, in which resources are *automatically* detected and prepared for use with IBM Tivoli System Automation.

As described in “Components of IBM Tivoli System Automation” on page 5, IBM Tivoli System Automation uses as its basis the functionality of RMC. A resource is therefore sometimes referred to as an *RMC resource*.

Resources (adapter, program, disk, and so on) are controlled by a Resource Manager (abbreviated to **RM**).

---

### What is a resource class?

A resource class is a set of resources of the same type. For example, while a resource might be a particular file system or particular host machine, a resource class would be the set of file systems, or the set of host machines. A resource class defines the common characteristics that instances of the resource class can have (for example, all file systems will have identifying characteristics (such as a name), as well as changing characteristics (such as whether or not it is mounted). Each individual resource instance of the resource class will then define what its particular characteristic values are (for example, this file system is named *"/var"*, and it is currently a mounted file system).

---

### What are resource attributes?

A resource attribute describes some characteristics of a resource. If the resource represents a host machine, its attributes would identify such information as the host name, size of its physical memory, machine type, and so on.

### What is the difference between persistent attributes and dynamic attributes?

There are two types of resource attributes – persistent attributes and dynamic attributes. The attributes of a host machine just mentioned (host name, size of physical memory, and machine type) are examples of

## Using Resources

persistent attributes – they describe enduring characteristics of a resource. While you could change the host name or increase the size of its physical memory, these characteristics are, in general, stable and unchanging.

Dynamic attributes, on the other hand, represent changing characteristics of the resource. Dynamic attributes of a host resource, for example, would identify such things as the average number of processes that are waiting in the run queue, processor idle time, the number of users currently logged on, and so on.

---

## What is a managed resource?

A resource becomes an *IBM Tivoli System Automation managed resource* (referred to simply as a *managed resource*) as soon as the resource has been inserted in an IBM Tivoli System Automation resource group. This is done using the IBM Tivoli System Automation **addrgmbr** command (described in “addrgmbr” on page 147). From this point onwards, the resource can be managed using the IBM Tivoli System Automation commands and programs.

Managed resources are supplied in IBM Tivoli System Automation resource class IBM.ManagedResource.

---

## Working with resources

Note that IBM Tivoli System Automation does not allow you to start or stop **resources** directly.

Resource starting and stopping is based on the setting of the NominalState attribute of a **resource group**. For example, setting the NominalState attribute of a resource group to “Online” indicates that you wish to start the resources in the resource group. Setting the NominalState attribute of a resource group to “Offline” indicates that you wish to stop the resources in the resource group. See the description in “Starting and stopping a resource group” on page 46 and “NominalState attribute” on page 40.

---

## Attributes used by resources

Resources can have the following attributes:

|                          |                                                                                                                                 |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>NodeNameList</b>      | indicates on which nodes the resource is allowed to run. This is an attribute of an RSCT resource.                              |
| <b>SelectFrom Policy</b> | defines the list of nodes the resource is available on. This is an attribute of a managed resource.                             |
| <b>ResourceType</b>      | indicates whether the resource is allowed to run on multiple nodes, or a single node. This is an attribute of an RSCT resource. |
| <b>OpState</b>           | specifies the operational state of a resource or a resource group. This is an attribute of an RSCT resource.                    |

## NodeNameList attribute

The NodeNameList persistent attribute represents the set of nodes on which the resource can run.

The nodes resources started on by IBM Tivoli System Automation are controlled by the order of the nodes in the NodeNameList of the resources. By default, each floating resource will be placed on the first node in its node list where it can be started with respect of all its relationships to other resources. This behavior can be modified by using the SelectFromPolicy attribute described below. Floating resources which should be collocated are placed before independent floating resources (these are resources which do not have relationships to other resources) and anticollocated floating resources.

If there are collocated resources with different node lists, the resources are placed on the node which is chosen by the majority of resources. In a tie situation, one node is chosen randomly.

## SelectFromPolicy attribute

As described above, the NodeNameList persistent attribute defines the list of nodes the resource is available on. This list is ordered, either in the sequence as specified by the user or in the sequence the resource manager owning the resource put them in. The SelectFromPolicy attribute gives the user more flexibility. It allows to tell IBM Tivoli System Automation which algorithm to use when selecting a node from the list. This can be either ordered, meaning that IBM Tivoli System Automation always starts from the beginning of the list when determining the next available node, or any, meaning that IBM Tivoli System Automation picks a node randomly.

## ResourceType attribute

The ResourceType attribute is either defined by the resource manager or during creation of the resource. The ResourceType persistent attribute specifies whether a resource is:

- Serial fixed (its NodeNameList attribute contains a *single* node entry).

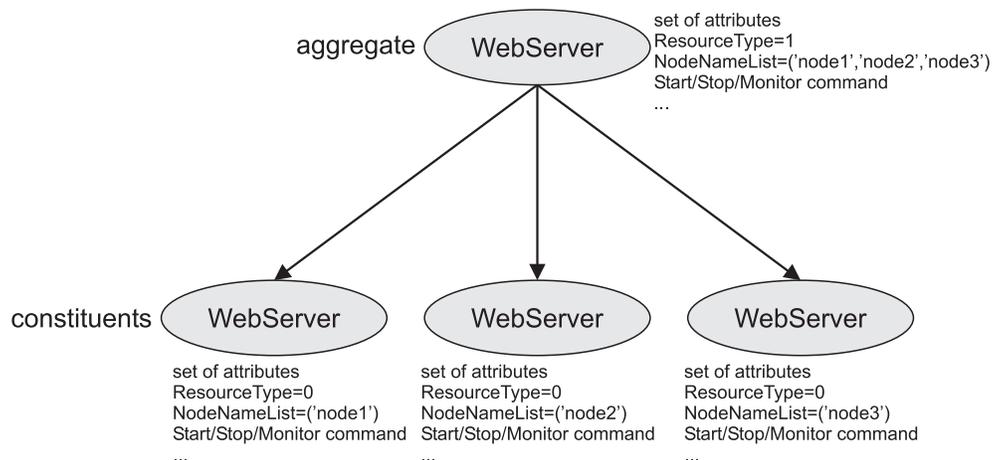
### Fixed resource

A fixed resource is a resource of which there is only a single instance within the cluster. It is defined upon a single node, and that is where it runs. It represents one entity such as a process, a mount point, or a network adapter.

- Serial floating (its NodeNameList attribute contains *one or multiple* entries). Although multiple nodes are defined for possible use, only one instance of the resource may be active at any time. For example, an IP address that can be moved from one machine to another is a floating resource; although multiple machines may have use of the IP address at some point, only one machine at a time will use it.

### Floating resource

A floating resource is a resource which can run on several nodes in the cluster. A floating resource is represented in RMC in the following way: You have one aggregate resource and one constituent resource on each node belonging to the aggregate resource.



The aggregate resource has a ResourceType attribute value of 1. The set of nodes where the resource should be able to run is defined in the NodeNameList attribute of the aggregate resource. The other attributes of this resource are defined by the resource manager and its class definition.

If you create a floating resource you create the aggregate resource. The resource manager responsible for this type of resource will create constituent resources on each node the resource is supposed to run on. The constituent resources have their own values of the attributes. The ResourceType of a constituent resource is 0 (a fixed resource), and the NodeNameList contains one node only. At time of creation the other attributes have identical values as the aggregate resource.

The following happens if you change attributes of a floating resource:

## Attributes used by resources

- A change of the NodeNameList of the aggregate resource causes deletion or creation of constituent resources.
- If you change an attribute of the aggregate resource this changes the according attributes of all constituent resources.
- If you change an attribute of a constituent resource this affects the constituent resource only and is not conveyed to other constituent resources or the aggregate resource.

A floating resource represents an automatable entity such as an application or a service ip address which can run on several nodes.

## OpState attribute

RMC uses the OpState dynamic attribute to specify the operational state of a resource. It is mandatory for resources added to a resource group.

These are the possible values that the OpState attribute can have:

|                        |                                                              |
|------------------------|--------------------------------------------------------------|
| <b>Offline</b>         | The resource is not started.                                 |
| <b>Pending Online</b>  | The resource has been started, but is not yet ready for work |
| <b>Online</b>          | The resource is ready for work.                              |
| <b>Pending Offline</b> | The resource is in the process of being stopped              |

Some of the operational states indicate problems:

|                       |                                                                                                                                                                                                                                            |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Failed Offline</b> | The resource is broken and cannot be used. You have to reset the resource when you have fixed it.                                                                                                                                          |
| <b>Stuck Online</b>   | The resource was being started, but did not become ready for work within the expected time interval and cannot be brought offline. Another possibility is that the resource was online, but an offline request could not bring it offline. |
| <b>Unknown</b>        | IBM Tivoli System Automation is unable to obtain reliable state information from the RMC managing the resource.                                                                                                                            |

**Note:** You might have to reset a resource that has the Failed Offline state. To do so, use the RMC command **resetrsrc**. For details, refer to the man page for this command.

When a node of a resource is Offline, the resource is considered to be Failed Offline, even though its operation state at that point is Unknown. IBM Tivoli System Automation can do this because it has separate state data for the resources node.

## Nominal state of a resource

Resources do not have nominal state information. You cannot set the nominal state of a resource directly. Resources must be defined within the resource groups. Each resource group has a nominal state. This is either Online or Offline and tells IBM Tivoli System Automation whether the resources within the resource group should be Online or Offline at this point in time. You can change the resource group nominal state value.

---

## Chapter 5. Using resource groups

This chapter describes how you use resource groups, in these main sections:

- “What is a resource group?”
- “Attributes used by resource groups” on page 38
- “Making (creating) a resource group” on page 45
- “Changing attributes of a resource group” on page 46
- “Removing a resource group” on page 47
- “Listing a resource group or its resource members” on page 45
- “Adding a member resource to a resource group” on page 45
- “Removing a member resource from a resource group” on page 47
- “Changing the attributes of resource group members” on page 47

### Related Section:

- “Events that might allow a resource group to become Online” on page 83

These are the IBM Tivoli System Automation commands that you use together with resource groups:

*Table 4. IBM Tivoli System Automation commands used with resource groups*

| Command  | Description                                                                                         | For details, see page: |
|----------|-----------------------------------------------------------------------------------------------------|------------------------|
| mkrq     | Make a resource group                                                                               | 191                    |
| rmrg     | Remove a resource group                                                                             | 206                    |
| chrg     | Change persistent attributes of a resource group (including starting and stopping a resource group) | 159                    |
| lsrg     | List one or more resource groups                                                                    | 173                    |
| addrgmbr | Add member resources to a resource group                                                            | 147                    |
| rmrgmbr  | Remove member resources from a resource group                                                       | 208                    |
| chrgmbr  | Change attributes of the member resources of the resource group                                     | 162                    |
| rgreq    | Start, stop, cancel, or move a resource group                                                       | 197                    |
| rmrgmbr  | Remove one or more resources from a resource group                                                  | 208                    |

---

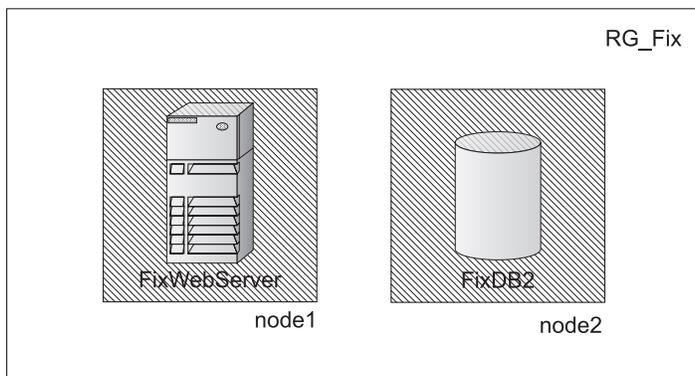
## What is a resource group?

The central unit in IBM Tivoli System Automation are the resource groups. They are a logical container for a collection of resources that can be treated as one logical instance. One aspect of resource groups is that you can use them to control all of their members collectively. For example, if you set a resource group's NominalState to Online all members are started and kept online. Vice versa, if you set the NominalState to Offline all members are stopped and kept offline. Another aspect of resource groups is that it is possible to monitor their OpState which provides a consolidation of the OpStates of the individual resource group members. Members of a resource group can be of type:

- Serial fixed.
- Serial floating.
- And even resource groups itself which means that nested groups can be defined.

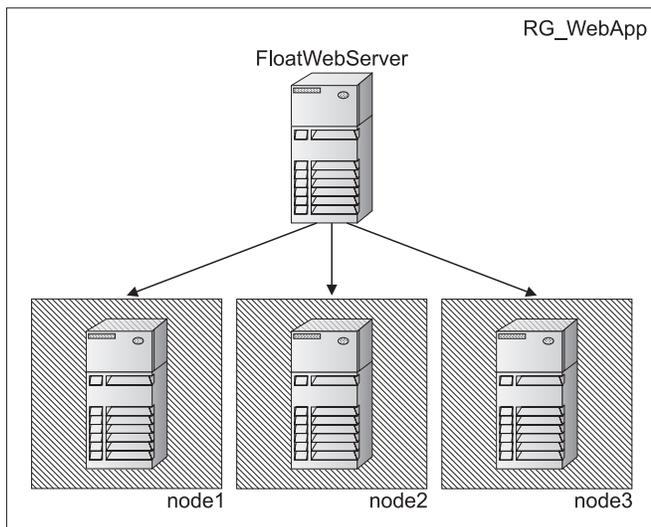
An example for a resource group containing fixed resources is a resource group RG\_Fix which contains serial fixed resources. These are a web server FixWebServer which can only run on node1, and a database resource FixDB2 located on node2.

## Resource Groups



In order to start both resources FixWebServer and FixDB2, set the NominalState of RG\_Fix to Online. This example also shows that IBM Tivoli System Automation can handle resource group members of a resource group which are distributed on different nodes in a cluster.

An example for floating resource group members is the following: A web server apache1 could run either on node1, node2, or node3. The resource group RG\_WebApp would look very similar except that the web server could be started on either of the three nodes



This example shows that resource groups can contain a mixture of members of different resource types.

The concept of resource groups is very powerful as it allows defining resource groups as members of other resource groups. An example is resource group RG\_A which has as members resource A, which is a fixed resource, and RG\_WebApp, the resource group from the previous example. Nested resource groups allow structuring complex environments in several layers. The nesting level is 50.

Another flexibility of the resource groups functionality is that all kinds of relationships like start /stop relationships and location constraint relationships can be defined with resource groups as source or target resource. Furthermore it is allowed that resource group members can be part of such relationships as source or target resource.

Resource groups are defined in IBM Tivoli System Automation resource class IBM.ResourceGroup.

## Rules for using resource groups

These are the rules for using resource groups:

1. A resource group *cannot* contain an equivalency or vice versa.

2. A resource can only be in one group.
3. A member cannot be in a group and in an equivalency.
4. The nesting level of a resource group is limited to 50.
5. The number of resources linked by groups or relationships is limited to 100.

### Attributes used by resource groups

A resource group provides the following persistent RSCT attributes which can be defined by the user:

|                         |                                                                                                                                                                                                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AllowedNode</b>      | limits the nodes on which the resource group members are allowed to be started.                                                                                                                                                                                        |
| <b>MemberLocation</b>   | defines if all members of a resource group have to be collocated or not. Collocated means that all members have to run on the same node. None means that the members are not dependent from each other and can arbitrarily run on the nodes on which they are defined. |
| <b>Name</b>             | defines a unique name for a resource group.                                                                                                                                                                                                                            |
| <b>NominalState</b>     | is the desired state of the resource group. IBM Tivoli System Automation tries to bring up and keep the resource group in this state.                                                                                                                                  |
| <b>Priority</b>         | defines the importance of a resource group in a conflict situation.                                                                                                                                                                                                    |
| <b>ExcludedList</b>     | defines a list of nodes which are temporarily excluded from the node list of the group.                                                                                                                                                                                |
| <b>ActivePeerDomain</b> | is the name of the peer domain the group is defined to.                                                                                                                                                                                                                |

**Note:** The persistent attributes described in this section can only be modified if the resource group containing them is **Offline**. An exception is the NominalState attribute which can be modified if the resource group is **Online**.

A resource group provides the following dynamic attributes:

|                          |                                                                                   |
|--------------------------|-----------------------------------------------------------------------------------|
| <b>OpState</b>           | Specifies the aggregate operational state of the collection of managed resources. |
| <b>TopGroup</b>          | Shows the name of the top level resource group of a resource group.               |
| <b>AutomationDetails</b> | Shows IBM Tivoli System Automation internal states of the resource group.         |
| <b>MoveStatus</b>        | Shows the progress of a move of a resource group initiated by a rgreq command.    |
| <b>ConfigValidity</b>    | Shows if a policy has become invalid.                                             |

### AllowedNode attribute

You use the AllowedNode parameter to define a set of nodes in a cluster on which the members of a resource group are limited to run.

You can choose between the following parameters:

|                 |                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>All</b>      | is the default value. It means that no limitation is made by the resource group. It can run on all nodes in the cluster.                                                             |
| <b>One node</b> | defines a specific node on which all resource group members have to run on. If the specified node is removed from the cluster at a later time, then AllowedNode will default to All. |

### Equivalency of nodes

contains a set of nodes where the resource group members are limited to run on. Only static equivalencies are allowed. See also Chapter 6, "Using equivalencies," on page 49.

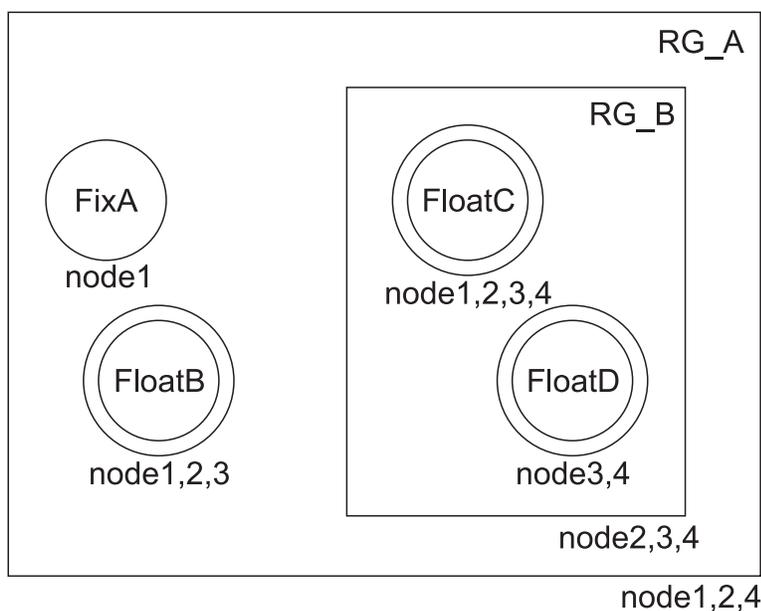
### Node limitation aspect of AllowedNode parameter

In specific cases it might be necessary to limit a member of a resource group to run on a set of nodes. For example, when a floating resource is defined a NodeNameList has to be specified which is in general

independent from the IBM Tivoli System Automation usage. The `NodeNameList` of the floating resource is used by the resource managers (for instance `GblResRM`) which own a floating resource. For them the `NodeNameList` defines on which nodes the constituents of a floating resource could potentially run. Whereas the `AllowedNode` attribute belongs to a resource group parameter that has different resource group members which are floating resources. Here the `AllowedNode` attribute allows to limit the nodes on which the resource group members are allowed to run. This means for resource group members of type:

- **Fixed resource**, that the `NodeNameList` which contains only one node on which the fixed resource is located has to be part of the `AllowedNode` parameter.
- **Floating resource**, that the intersection of the `NodeNameList` and the `AllowedNode` parameter defines the set of nodes on which the floating resource is started and controlled by IBM Tivoli System Automation.
- **Resource Group**, that an intersection of the `AllowedNode` parameter of the inner and the outer group derives a resulting list of allowed nodes for the inner group. The resulting list defines the nodes on which the members of the inner resource group are allowed to run on.

The following example explains the behavior of the `AllowedNode` parameter on the node limitation: Given an outer resource group `RG_A` with a member `FixA` with a `NodeNameList` = {node1}, a floating member `FloatB` with a `NodeNameList` = {node1, node2, node3} and a resource group `RG_B` with `AllowedNode` = {node2, node3, node4}. `RG_B` contains `FloatC` with a `NodeNameList` of {node1, node2, node3, node4} and `FloatD` with `NodeNameList` of {node3, node4}.



The result of this scenario is that:

- `FixA` can only be started by IBM Tivoli System Automation on node1.
- `FloatB` can only be started by IBM Tivoli System Automation on node1 and node2.
- `RG_B`'s members are limited to run on node2 and node4.
- `FloatC` can only be started by IBM Tivoli System Automation on node node2 and node4.
- `FloatD` can only be started by IBM Tivoli System Automation on node4.

The `AllowedNode` list of an inner group is not affected when its outer group has no node limitation due to its intersection of the own `AllowedNode` parameter and the intersection with the `AllowedNode` list of its outer group

## Resource Groups

### MemberLocation attribute

You use the MemberLocation persistent attribute to specify the default location between resources in the resource group. Valid values are either:

- |                                 |                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Collocated (the default)</b> | Collocated means that all members have to run on the same node. If you specify Collocated, you cannot apply the Affinity, AntiAffinity, and AntiCollocated managed relationships between the resources in the resource group to specify on which nodes the member resources should be located. See Chapter 7, “Using managed relationships,” on page 53 for a detailed explanation of relationships. |
| <b>None</b>                     | None means that the resource group implies no restriction regarding the location of its members.                                                                                                                                                                                                                                                                                                     |

The MemberLocation attribute is applied to *all* member resources of the resource group.

If resource groups are nested, the value of the MemberLocation attribute of the outer resource groups must allow the inner-group(s) specification. Therefore, if a resource group's MemberLocation attribute is collocated, only collocated resource groups are allowed as members.

### Name attribute

The Name of a resource group has to be unique in a cluster.

### NominalState attribute

You use the NominalState persistent attribute to control a resource group. By setting the NominalState of a resource group to Online all of its members are started and kept online.

The NominalState attribute can be either Online or Offline. A NominalState of Offline causes that all resource group members are stopped and kept offline. There are exceptions:

1. If resource groups are nested, a NominalState of online of an outer group overrules a NominalState of offline of an inner group and starts these inner groups. You cannot stop such an inner group separately.
2. If resource groups are nested, a NominalState of offline of an outer group will not overrule a NominalState of online of an inner group. A NominalState of online of an inner group will start this group and groups contained in this inner group. The outer groups' OpState will change, but the other group members remain untouched.
3. Cross group dependencies might force individual group members to be started. See “Relationships for start / stop behavior” on page 56.

The default value for the NominalState of a resource group is Offline.

This attribute can be modified at any time.

**Note:** You can modify this attribute with the **chrg -o** command (see “chrg” on page 159). The numeric value of this attribute is

- 1 Offline**
- 0 Online**

### Priority attribute

You use the Priority persistent attribute to specify the relative priority of this resource group in relationship to other resource groups.

The Priority attribute is used to resolve conflicts when resource groups are being started that have conflicting *managed relationships* (described in Chapter 7, “Using managed relationships,” on page 53) to

other started or online resources. These conflicts may be between the resource group or a managed resource contained in that group and any other started or running resource.

For instance, you have a cluster with only one node online, and two resource groups with an AntiCollocated relationship against each other. This means that the resource groups must never be started on the same node at a time. IBM Tivoli System Automation now uses the value of the Priority attribute of the resource groups to find out which of the resource groups should be online, and which cannot be online because of the conflicting AntiCollocated relationship.

If a lower priority active resource group prevents the activation of a higher priority resource group because of conflicting relationships, the lower priority resource group is stopped in order to allow the activation of the resource group with higher priority to proceed.

If resource groups are nested, the outer resource group must have a higher or equal priority than any of the inner resource groups.

The default value of the Priority attribute is zero, which is the lowest value. The maximum value is 200.

**Hint:** IBM Tivoli System Automation also uses the Mandatory attribute of the managed resources (described in “Attributes used for resource group members” on page 44) to determine which resources will be started in a conflict situation. In case of nested resource groups, the resource groups that are non-mandatory members should have a lower priority than mandatory members. Otherwise Mandatory members may be discarded.

## ExcludedList attribute

You use the ExcludedList attribute to temporarily exclude one or a list of nodes from the node list of the group. When excluding a node, the resources residing on the node being excluded are not automatically forced offline. The move must be triggered by means of the `rgreq` command.

This means that placing a node in the exclude list causes IBM Tivoli System Automation to not consider the node as a potential candidate for hosting the resource. It can be used to gradually and non-disruptively move resources away from a node in preparation for an EXCLUDE (via the `samctrl` command) at a later point in time.

The following rules apply:

1. Excluding a node means for a fixed resource group member that the resource cannot be started anymore.
2. Excluding a node means for a floating resource group member that its constituent on that node cannot be started any more.

## ActivePeerDomain

This attribute shows the name of the RSCT Peer Domain the group is defined in.

## OpState attribute

IBM Tivoli System Automation uses the OpState dynamic attribute to specify the aggregate operational state of the collection (of managed resources). It is determined from the individual operational states of the member resources of the resource group. These are the possible values that the OpState attribute can have:

| Status  | Value | Description                                                                                                      |
|---------|-------|------------------------------------------------------------------------------------------------------------------|
| Unknown | 0     |                                                                                                                  |
| Online  | 1     | Specifies that all of the Mandatory member resources are Online. The Non-Mandatory member resources are ignored. |

## Resource Groups

| Status          | Value | Description                                                                                                                                                                         |
|-----------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Offline         | 2     | Specifies that all of the member resources are Offline.                                                                                                                             |
| Failed Offline  | 3     | Specifies that one or more member resources contained in the resource group, are FailedOffline. In this case, all resources contained in the resource group will be set to Offline. |
| Stuck Online    | 4     | Specifies that a member resource is stuck online.                                                                                                                                   |
| Pending Online  | 5     | Specifies that a Start command is executed (the resource group's NominalState attribute is set to Online). The resource group must begin processing an Online action.               |
| Pending Offline | 6     | Specifies that an Offline action has been initiated.                                                                                                                                |

## TopGroup attribute

You use the TopGroup dynamic attribute to view the top level resource group of a resource group.

In IBM Tivoli System Automation resource groups can be members of another resource group, meaning that resource groups can be nested. The TopGroup attribute shows the name of the top level resource group of the current group. The attribute can be displayed using the **lsrg** command as described on page 173 and shown below.

**Note:** When using the `lsrg -g` command to query a resource group, the NominalState of the top group is shown in the output for user convenience together with the TopGroup attribute. The output looks similar to:

```
TopGroup = apacherg
TopGroupNominalState = Offline
```

## AutomationDetails attribute

This attribute shows System Automation internal states of the group. These states include:

|                        |                                                                                                                                                                                        |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CompoundState</b>   | Overall status of the resource group including group dependencies. An example is "Satisfactory" – resource/group has reached the requested user status.                                |
| <b>DesiredState</b>    | User requested status of the resource group. An example is "online" – user requested that the resource group should be online.                                                         |
| <b>ObservedState</b>   | Real status of the resource group from an automation point of view. An example is "online" – the resource group is currently online.                                                   |
| <b>BindingState</b>    | Status indicating if the resource group is bound to a specific system. An example is "bound" – the resource group is currently bound to a specific system.                             |
| <b>AutomationState</b> | Status indicating if the resource group is currently being automated. An example is "Idle" – IBM Tivoli System Automation is currently not trying to start or stop the resource group. |
| <b>StartableState</b>  | Status indicating if the resource group can be started. An example is "startable" – it is currently possible to start this resource group.                                             |
| <b>HealthState</b>     | currently not used, reserved for future releases.                                                                                                                                      |

To show the automation details as indicated above you use the `lsrg` command with the `-A d` and the `-V` option. For example, to show the automation details of a resource group named "apacherg", you use the command:

```
lsrg -A d -V -g apacherg
```

## MoveStatus

Shows the progress of a move of a resource group initiated by a `rgreq` command.

To show the move status you use the `lsrg` command with the `-A d` and the `-V` option. For example, to show the move status of a resource group named "apacherg", you use the command:

```
lsrg -A d -V -g apacherg
```

## ConfigValidity

After a policy was established, several things can happen to make the policy invalid. This attribute shows the reason for invalidity.

For example, if a node is removed from the peer domain which was the only common node for the members of a collocated resource group, the resources can no longer be started. This will be indicated by the `ConfigValidity` attribute.

### Attributes used for resource group members

In addition for each resource group member a user has to define persistent attribute:

- Mandatory** Defined for each resource group member and specifies whether it is mandatory for the group. Alternatively a member can also be non-mandatory.
- MemberOf** The name of the resource group of which the resources are members.
- SelectFromPolicy** Used to tell IBM Tivoli System Automation where floating resources should be preferably started.
- ConfigValidity** This attribute is reserved for future use.

### Mandatory attribute

You use the Mandatory persistent attribute to specify whether a managed resource is Mandatory or Non-Mandatory.

When a resource group is started, all managed resources within that group that are Mandatory must also be started. Managed resources that are Non-Mandatory (whose Mandatory attribute is set to False) might not be started when a conflict exists. If a managed resource that is Mandatory fails, the entire resource group is stopped and started on another node, but if a non-mandatory member of a resource group fails, the resource group stays Online on that node.

Resources that are members of a resource group are implicitly mandatory unless this attribute value is explicitly set to False.

Member resources whose Mandatory managed resource attribute is False may be sacrificed in order to activate the resource group.

### MemberOf attribute

Indicates that the resource is contained in a resource group resource. The MemberOf persistent attribute is generated implicitly when resources (including nested resource groups) are added to a resource group. The MemberOf attribute is used to determine the set of resources to be started and stopped when the resource group is activated or deactivated (either explicitly through a stop order, or implicitly through a non-recoverable member resource failure). A resource can be a member of one and only one resource group.

---

## Defining and administering a resource group

### Making (creating) a resource group

To make (create) a resource group you use the **mkr** command, in which you define to IBM Tivoli System Automation:

- Where the resource group is allowed to run.
- The relative importance of the resource group in relation to other resource groups (using the *Priority* attribute, as explained in “Priority attribute” on page 40).
- The *Location* relationship among the member resources of the resource group (explained in “Location relationships” on page 69).

Newly-created resource groups will default to an NominalState of Offline. This allows a user or administrator to fully configure the resource group and its resources.

For example, to define a new resource group called **apacherg2** with location relationship “None” and allowed node name “node03”, you would enter:

```
mkr -l None -n node03 apacherg2
```

For further details, see either the **mkr** man page, or “mkr” on page 191.

To establish the node list for a resource group, you can use either:

- **mkr** command, to create a new resource group.
- **chr** command, for an already-existing resource group.

To specify the node list to be used with the above two commands, you can either:

- Enter a node name when using the **mkr/chr** command.
- Establish as an *equivalency*, the set of nodes on which the resource group can be activated. This must be done before the node list is established. You then use the **mkr/chr** command, and the required equivalency will be attached to the resource group. (For details of equivalencies, see Chapter 6, “Using equivalencies,” on page 49).

### Adding a member resource to a resource group

To add one or more new member resources to a resource group, you use the **addr** command.

#### Notes:

1. A member resource cannot be included in more than one resource group at the same time.
2. A member resource cannot be in a resource group *and* in an equivalency at the same time.

For example, to add member resource apache1, belonging to resource class IBM.Application, to a resource group apacherg2, you would enter:

```
addr -g apacherg2 IBM.Application:apache1
```

For further details, see either the **addr** man page, or “addr” on page 147.

### Listing a resource group or its resource members

To list a resource group or its members, you use the **lsr** command. If the resource group name is omitted, all of the resource groups are listed. If the resource group name is specified with the **-m** option, the member resource names and resource classes are listed.

If the *Attr* parameter is specified, then the attributes specified for the resource group are listed.

Three resource groups are defined in the following examples:

## Resource Groups

**Example 1:** If command **lsrg** is entered, this type of information is displayed:

```
Resource Group Names:
apacherg2
apacherg3
apacherg4
```

**Example 2:** To list all members of all resource groups, enter:

```
lsrg -m
```

This information is then displayed:

```
Displaying Member Resource information:
Class:Resource:Node[ManagedResource] Mandatory MemberOf OpState
IBM.Application:apache1 True apacherg2 Offline
```

**Example 3:** apacherg2 contains one resource. To list the members of apacherg2, this command is entered:

```
lsrg -m -g apacherg2
```

This information is then displayed:

```
Member Resource 1:
Class:Resource:Node[ManagedResource] = IBM.Application:apache1
Mandatory = True
MemberOf = apacherg2
OpState = Offline
```

**Example 4:** To list the attributes of a resource group apacherg2, this command is entered:

```
lsrg -g apacherg2
```

This information is then displayed:

```
Resource Group 1:
Name = apacherg2
MemberLocation = None
Priority = 0
AllowedNode = node03
NominalState = Offline
OpState = Offline
```

For further details, see either the **lsrg** man page, or “lsrg” on page 173.

## Starting and stopping a resource group

To start or stop a resource group you set the **NominalState** attribute of the resource group to online or offline respectively. Use the **chrg** command to do this.

For example, to start a resource group called apacherg2, you would enter:

```
chrg -o online apacherg2
```

For further details, see either the **chrg** man page, or “chrg” on page 159.

## Changing attributes of a resource group

To change the persistent attribute values of one or more resource groups, you use the **chrg** command. The name of a resource group can also be changed with this command, using the **-c** option.

**Example 1:** to change the location relationship of a group apacherg2 to collocated, you would enter:

```
chrg -l collocated apacherg2
```

**Example 2:** to change the name of a group apacherg3 to apacherg4, you would enter:

```
chrg -c apacherg4 apacherg3
```

For further details, see either the **chrg** man page, or “chrg” on page 159.

## Changing the attributes of resource group members

To change the attributes of the specified member resources, you use the **chrgmbr** command.

This command also allows you to specify changes to the Mandatory attribute of a managed resource by using the **-m** option, and to change the resource group to which the resource belongs by using the **-c** option.

For example, to change the resource group to which member resource apache2 of resource class IBM.Application belongs, from the current resource group apacherg2 to resource group apacherg3, you would enter:

```
chrgmbr -c apacherg3 -g apacherg2 IBM.Application:apache2
```

For further details, see either the **chrgmbr** man page, or “chrgmbr” on page 162.

## Removing a member resource from a resource group

You use the **rmmrgmbr** command to remove:

- all the member resources of a specified resource group.
- only the specified member resources of the specified resource group.
- the member resources that match a selection string.

IBM Tivoli System Automation also ensures that any associated managed relationship or equivalency is also updated.

For example, to remove member resource apache2 that belongs to resource class IBM.Application, from resource group apacherg3, you would enter:

```
rmmrgmbr -g apacherg3 IBM.Application:apache2
```

For further details, see either the **rmmrgmbr** man page, or “rmmrgmbr” on page 208.

## Removing a resource group

To remove one or more resource groups, you use the **rmmrg** command. The resource groups are specified either using the *Resource\_group* parameter, or by matching a selection string. The member resources associated with the removed resource groups, are also removed by IBM Tivoli System Automation. If the resource group to be removed is still online, the resource group is not removed. Note that relationships where these deleted resource groups are the source, are also deleted.

For example, to remove resource groups called apacherg2 and apacherg3, you would enter:

```
rmmrg apacherg2 apacherg3
```

For further details, see either the **rmmrg** man page, or “rmmrg” on page 206.



---

## Chapter 6. Using equivalencies

This chapter describes equivalencies in these main sections:

- “What is an equivalency?”
- “Attributes used by equivalencies” on page 50
- “Rules for using equivalencies”
- “Making (creating) an equivalency” on page 51
- “Changing an equivalency” on page 51
- “Removing an equivalency” on page 51
- “List one or more equivalencies” on page 51

These are the IBM Tivoli System Automation commands that you use together with equivalencies:

*Table 5. IBM Tivoli System Automation commands used with equivalencies*

| Command | Description                    | For details, see page: |
|---------|--------------------------------|------------------------|
| mkequ   | Make an equivalency resource   | 183                    |
| rmequ   | Remove an equivalency resource | 200                    |
| chequ   | Change an equivalency resource | 151                    |
| lsequ   | List equivalency resources     | 165                    |

---

### What is an equivalency?

An equivalency is a collection of resources that provide the same functionality. An equivalency consists of a set of fixed resources from the same resource class.

For example, network adapters might be defined as equivalencies. If one network adapter goes offline, another network adapter can take over the processing from the offline adapter.

Equivalencies are also used for establishing a resource group's node list.

From this equivalency, one or more resources can be selected to satisfy a managed relationship. But only one member is started on a node to satisfy a managed relationship. For details of managed relationships, see Chapter 7, “Using managed relationships,” on page 53.

There are 2 types of equivalencies:

1. one with a static membership list. This type of equivalency contains a certain set of resources which a user explicitly added to the equivalency.
2. one with a SelectString list which dynamically determines at run-time which resources are contained within the equivalency. If RMC resources are created which match the dynamic select string, these are automatically contained within the equivalency. Specifying a policy is not reasonable for this type of equivalency as the resources are not ordered.

Equivalencies are supplied in IBM Tivoli System Automation resource class IBM.Equivalency.

### Rules for using equivalencies

These are the rules for using Equivalencies:

1. A resource can be a member of either an equivalency or a resource group, but not both.
2. A resource may be in more than one equivalency.
3. The specified resources must all be from the same resource class.
4. Equivalencies cannot be members of an equivalency.

## Equivalencies

5. Resource groups cannot be members of an equivalency.
6. Equivalencies cannot be members of a resource group.
7. An equivalency that satisfies a relationship for an active resource cannot be modified.
8. An equivalency can only be the *target* of a managed relationship (it cannot be the *source* of a managed relationship).
9. The members of an equivalency must be fixed resources. Floating resources are not allowed.

---

## Attributes used by equivalencies

### MemberClass attribute

IBM Tivoli System Automation uses the MemberClass persistent attribute to determine the resource class of all the member resources.

### Membership attribute

IBM Tivoli System Automation uses the Membership persistent attribute to determine the set of resources contained within the equivalency. If a Membership attribute is specified, no SelectString attribute is allowed.

### SelectString attribute

IBM Tivoli System Automation uses the SelectString persistent attribute to dynamically determine the resources contained within an equivalency. If resources matching the selection string are inserted in, or removed from, the system, then IBM Tivoli System Automation automatically modifies the equivalency. If a SelectString attribute is specified, no Membership attribute is allowed.

### SelectFromPolicy attribute

IBM Tivoli System Automation uses the SelectFromPolicy persistent attribute to determine the policy to be used in making a selection from the equivalency. This attribute can be modified when the resource groups referencing the equivalency are offline. This policy can be either:

|                          |                                                                                                                                                                               |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Ordered</b>           | In this case, if a resource contained within an equivalency fails, IBM Tivoli System Automation always starts from the beginning of the selection list.                       |
| <b>Any (the default)</b> | In this case, if a resource contained within an equivalency fails, IBM Tivoli System Automation chooses any resource without referring to a pre-specified order of selection. |

**Note:** An Ordered policy may not be used when a dynamic SelectString is used.

---

## Attributes used by members of equivalencies

- A resource that is to be added to an equivalency *must* have this attribute:
  - OpState
- A resource that is to be added to an equivalency *may* have these attributes:
  - NodeNameList
  - ResourceType

For details of these attributes, refer to “Attributes used by resources” on page 32.

---

## Defining and administering equivalencies

### Making (creating) an equivalency

To make an equivalency among resources, you use the **mkequ** command.

For further details, see either the **mkequ** man page, or “mkequ” on page 183.

For example, to create an static equivalency called NetworkInterfaces of two ethernet interfaces eth0 located on Linux systems node01 and node02 of the resource class IBM.NetworkInterface, you would enter:

```
mkequ NetworkInterfaces IBM.NetworkInterface:eth0:node01,eth0:node02
```

To create a dynamic equivalency called NetworkInterfacesDynamic containing all available ethernet interfaces in a cluster of Linux systems, you would enter:

```
mkequ -D "Name like 'eth%'" NetworkInterfacesDynamic IBM.NetworkInterface
```

In a cluster of AIX systems, you would enter:

```
mkequ -D "Name like 'en%'" NetworkInterfacesDynamic IBM.NetworkInterface
```

### List one or more equivalencies

To list one or more equivalencies, you use the **lsequ** command.

For further details, see either the **lsequ** man page, or “lsequ” on page 165.

If you omit an equivalency name, *all* of the defined equivalencies will be listed. If you specify an equivalency, the persistent attributes of the this equivalency will be listed. If you specify the attribute name as operand, the attributes specified for the equivalency will be listed.

For example, to list the persistent attributes of the equivalency NetworkInterfaces, you would enter:

```
lsequ -A p -e NetworkInterfaces
```

### Changing an equivalency

To add, remove, or totally replace the resources contained in an equivalency, you use the **chequ** command.

For further details, see either the **chequ** man page or “chequ” on page 151.

You can also use this command to change the name of the equivalency.

For example, to add the resource eth1 located on system node01 that belongs to the resource class IBM.NetworkInterface, to an equivalency called NetworkInterfaces, you would enter:

```
chequ -u a NetworkInterfaces IBM.NetworkInterface:eth1:node01
```

### Removing an equivalency

To remove one or more equivalencies, you use the **rmequ** command.

For further details, see either the **rmequ** man page, or “rmequ” on page 200.

You specify one or more equivalencies using either the Equivalency name as operand, or selection string.

For example, to remove an equivalency called NetworkInterfaces, you would enter:

```
rmequ NetworkInterfaces
```

## Equivalencies

---

## Chapter 7. Using managed relationships

This chapter describes how you use managed relationships in these main sections:

- “What is a managed relationship?”
- “Attributes used by managed relationships” on page 54
- “Relationships for start / stop behavior” on page 56
- “Location relationships” on page 69
- “Creating and administering relationships” on page 79

These are the IBM Tivoli System Automation commands that you use for managed relationships:

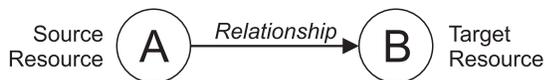
*Table 6. IBM Tivoli System Automation commands used with managed relationships*

| Command | Description                   | For details, see page: |
|---------|-------------------------------|------------------------|
| mkrel   | Make a managed relationship   | 187                    |
| lsrel   | List managed relationships    | 168                    |
| rmrel   | Remove a managed relationship | 202                    |
| chrel   | Change a managed relationship | 154                    |

---

### What is a managed relationship?

A managed relationship exists between a source resource, and one or more target resources. As a first example, during a resource group startup the source resource must be started after the target resource has become online: this example of a managed relationship uses a StartAfter value of the Relationship attribute.



As shown in the example relationships always indicate a direction. Relationships can cross node boundaries.

As a second example, a source resource should if possible be started on the same node as the target resource: this example of a managed relationship uses an Affinity value of the Relationship attribute (described in “Relationship attribute” on page 54).

Relationship attributes may be used with Condition attributes (described in “Condition attribute” on page 55). As a third example, a source resource should, if possible, be started on the same node as the target resource, but only if the target resource is online. This example of a managed relationship uses an Affinity value of the Relationship attribute, together with an IfOnline value of the Condition attribute.

The Relationship attribute is described in “Relationship attribute” on page 54, the Condition attribute is described in “Condition attribute” on page 55.

By using combinations of managed relationships, complex automation scenarios can be defined.

As mentioned above, a relationship is defined between a source and one or more target resources. If you remove the source resource of a relationship (can be a resource group member or the underlying RMC resource), the relationship is deleted. If you remove the last target resource from the list of target resources (can be a resource group member or the underlying RMC resource), the relationship is not deleted. You must remove this relationship with the **rmrel** command.

## Managed Relationships

The reason for this behavior is that no relationships should accidentally be deleted. There was a reason why, for example, a DependsOn was defined from a source resource to a target resource. The source resource cannot function properly without the target resource. So the relationship should not be deleted automatically unless you tell IBM Tivoli System Automation to do so.

Managed relationships are supplied in IBM Tivoli System Automation resource class IBM.ManagedRelationship.

---

## Attributes used by managed relationships

### Related Sections:

- “Attributes used by resources” on page 32
- “Attributes used by resource groups” on page 38
- “Attributes used by equivalencies” on page 50

The following picture shows another example of a managed relationship:



A managed relationship has the attributes described in the sections below.

### Name attribute

You use the Name persistent attribute to specify the name you wish to use for the managed relationship. This attribute is optional. It makes it easier to change or delete relationships.

### Source attribute

You use the Source persistent attribute to specify the source resource of the managed relationship.

### Target attribute

You use the Target persistent attribute to specify the list of target resources of the managed relationship.

### Relationship attribute

You use the Relationship persistent attribute to specify the relationship that is to be applied between source and target resources. There are two types of relationships, start / stop dependencies and location dependencies:

#### Start / Stop dependencies:

- StartAfter
- StopAfter
- DependsOn
- DependsOnAny
- ForcedDownBy

Start / Stop dependencies are used to define a start / stop behavior.

#### Location dependencies

- Collocated
- AntiCollocated
- Affinity
- AntiAffinity
- IsStartable

Location dependencies are used for locating resources on nodes.

### Condition attribute

The Condition persistent attribute specifies a condition to be used together with all Location relationships (described in “Location relationships” on page 69), *except for* the IsStartable managed relationship. The Condition persistent attribute defines when the relationship is considered applicable.

These are the conditions that can be applied:

- IfOnline
- IfNotOnline
- IfOffline
- IfNotOffline
- None

---

### Relationships for start / stop behavior

IBM Tivoli System Automation provides the following relationships which can be used to define a start/stop behavior:

- **StartAfter**
- **StopAfter**
- **DependsOn**
- **DependsOnAny**
- **ForcedDownBy**

The source of a start/stop relationship is either a member of a resource group or a resource group. See “What is a resource group?” on page 35 for more information about resource groups.

The target of a start/stop relationship is either

- a member of a resource group or a resource group.
- an equivalency.
- an RSCT resource (which is not a managed resource) which has to provide an OpState attribute.

Note that in case of a DependsOn relationship and source or target resources or both being groups, these groups must have a member location of collocated.

A start command cannot be issued against a resource directly. Therefore you start a resource by setting the nominal state of the resource group of which resource is a member of to online.

### StartAfter relationship

Use the StartAfter relationship to ensure that the source resource is only started when the target resource(s) are online.

The StartAfter relationship provides the following behavior scheme:



- With the start behavior StartAfter defines a start sequencing for resources A and B: When source resource A has to be started, then the target resource B is started first. After resource B has become online, resource A is started.  
Note that resource A and resource B can be started on different nodes.

The StartAfter relationship does not provide a force down behavior (see “DependsOn relationship” on page 62).

### Details on the start behavior of the StartAfter relationship

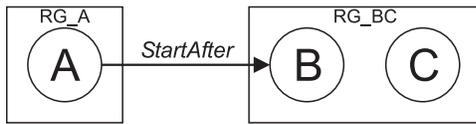
The start behavior is controlled via the operational state (OpState) of the target resource. At the time when the operational state of resource B has become online, resource A is started.

In many cases resource A and resource B are members of the same resource group.



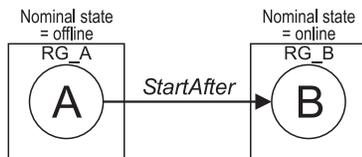
Setting their resource group’s nominal state to online causes that both members A and B are started. Due to the StartAfter relationship from A to B resource B is started first. When the operational state of resource B is online, resource A is started.

If resource A is a member of resource group RG\_A, and resource B is a member of resource group RG\_BC, and a StartAfter relationship is defined between A and B. Then the start behavior of the StartAfter relationship is triggered by setting the nominal state of RG\_A to online.



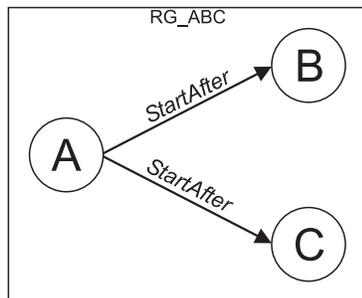
Due to the start sequence of the StartAfter relationship resource B has to be started first. In case RG\_BC's nominal state is set to offline, the following conflict exists: RG\_BC wants resource B to be offline whereas the StartAfter relationship forces B to be started. IBM Tivoli System Automation resolves this conflict in such a way that the online request is always more important than the offline request. Therefore resource B is started even though other possible group members of RG\_BC will not be started since the nominal state of their group is offline. After resource B is online, IBM Tivoli System Automation will start resource A. Resource C is not started.

The start order only acts in the forward direction of the relationship. In case resource A and resource B are part of different resource groups (A belongs to RG\_A and B belongs to RG\_B), then setting the nominal state of RG\_B to online does not cause any action on resource A since resource B has no forward relationship to resource A.



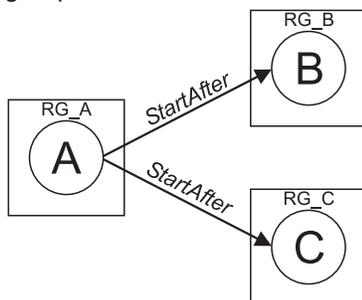
When RG\_A's nominal state is set to online, the resource A can be started right away since resource B is already online.

In another scenario it also might be the case that resource A has a StartAfter relationship to resource B and resource C.



In this case starting A requires that both resources B and C are online before IBM Tivoli System Automation can start resource A. For instance A, B, and C are members of the resource group RG\_ABC. Setting the nominal state of RG\_ABC to online causes that resources B and C are started in parallel first. When the operational state of both resources is online, then resource A is started.

It is also possible that resource A is member of resource group RG\_A, resource B is member of resource group RG\_B, and resource C is member of resource group RG\_C.



A has a StartAfter relationship to both B and C. Setting RG\_A's nominal state to online causes that due to the StartAfter relationship resource C and resource B are started. After both resources B and C are online, A is started.

## Managed Relationships

### Details on the stop behavior of the StartAfter relationship

Target resource B cannot be stopped while source resource A is online. If the NominalState attribute of source resource A is changed to Offline, target resource B automatically stops. Both resources can be simultaneously stopped.



In many cases source resource A and target resource B are members of the same resource group. So their NominalState values are identical.



Set the NominalState attribute of the RG\_AB to offline to stop both members A and B. Since the StartAfter relationship does not require a stop sequence, resources A and B can be simultaneously stopped.

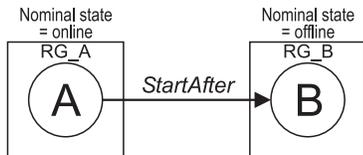


Provided that RG\_B has a NominalState of Online, you can start and stop RG\_A without affecting resource group RG\_B. It remains online.

If you set the NominalState of RG\_B to Offline and set the NominalState of RG\_A to Online, target resource B will start before source resource A.

If you set the NominalState of RG\_A to Offline, then resources A and B are simultaneously stopped.

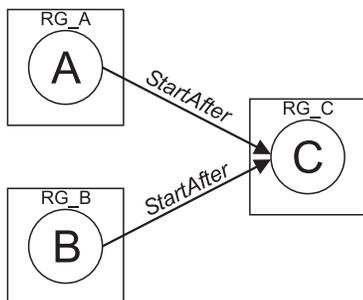
Consider the following stop behavior:



Provided the NominalState of RG\_A is Online and the NominalState of RG\_B is Offline, resource A and resource B are online. Now set the NominalState of RG\_A to Offline. Resource A and resource B simultaneously stop. The reason for this behavior is that resource B started due to the start request on resource group RG\_A which was passed on via the StartAfter relationship. Setting RG\_A to Offline removes the start request, and the NominalState of Offline from resource group RG\_B causes resource B to be stopped.

The StartAfter relationship causes the typical stop behavior: Resource A and B can be simultaneously stopped.

Resources A, B, and C are members of individual resource groups RG\_A, RG\_B, and RG\_C.



Resource C must be online to support both resources A and B. As long as the NominalState of either or both RG\_A and RG\_B is online, resource C must be kept Online, even if RG\_C's NominalState is Offline. Only when both RG\_A and RG\_B have a NominalState of Offline, resource C can be stopped. This will be the case if RG\_C's NominalState is Offline, too.

### Rules for using the StartAfter relationship

1. The StartAfter relationship must not conflict with an existing DependsOn relationship.
2. The StartAfter relationship does not assume that a Location relationship exists between managed resources. If you wish to define a Location relationship (see “Location relationships” on page 69), you must create an additional relationship for this purpose.
3. If IBM Tivoli System Automation is requested to start the source resource, it will however always attempt to first start the target resource.
4. If the target resource fails, this does not mean that the source resource will then be stopped.

## Managed Relationships

### StopAfter relationship

Use the StopAfter relationship to ensure that the source resource can only be stopped when the target resource has been already stopped.

The StopAfter relationship provides the following behavior scheme:



- Resource A will not be stopped unless the target resource has been brought Offline before (including Failed Offline).

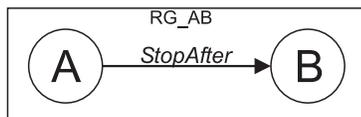
The StopAfter relationship does not provide a start and a force down behaviour (see “StartAfter relationship” on page 56 and “DependsOn relationship” on page 62).

### Details on the stop behavior of the StopAfter relationship

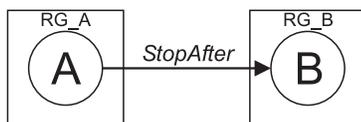
Source resource A cannot be stopped while target resource B is Online. If the OpState attribute of target resource B changes to Offline or Failed Offline, source resource A automatically stops.



In many cases source resource A and target resource B are members of the same resource group. Set the NominalState attribute of the RG\_AB to Online to start both members A and B. Since the StopAfter relationship does not require a start sequence, resources A and B can be simultaneously started. Setting their resource group’s NominalState attribute to Offline causes that members are stopped. Due to the relationship from A to B, resource B is stopped first. When the operational state of resource B is Offline, resource A is stopped.

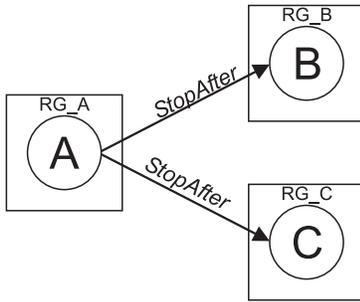


In case resource A and B are part of different resource groups (A belongs to RG\_A, and B belongs to RG\_B) and RG\_B has a NominalState of Offline, you can start and stop RG\_A without any dependency to resource group RG\_B. If you set the NominalState of RG\_B to Online and set the NominalState of RG\_A to Offline, source resource A cannot stop as long as target resource B is Online.



If the NominalState of RG\_A is Offline, you can start or stop RG\_B without any dependency to resource A.

It is also possible that resource A is a member of resource group RG\_A, resource B is a member of resource group RG\_B, and resource C is a member of resource group RG\_C. A has a StopAfter relationship to both B and C.



If the NominalState of RG\_A is Online and you want to stop it, RG\_A cannot be stopped as long as the NominalState of both RG\_B and RG\_C is Online. Only when both RG\_B and RG\_C have a NominalState of Offline or Failed Offline, resource A can be stopped.

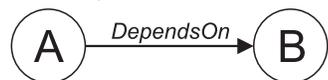
## Managed Relationships

### DependsOn relationship

IBM Tivoli System Automation uses the DependsOn relationship to ensure that the source resource can only be started when the target resource(s) is online. It is used in a similar way to the StartAfter relationship, except:

- A DependsOn relationship also includes an implicit collocation (explained in “Collocated relationship” on page 71) between the source and target resources.
- If a target resource fails, the source resource will also be stopped.

The DependsOn relationship provides the following three behavior schemes:



Resource A depends on the functionality of resource B, and this means that resource A cannot function without resource B. Therefore the force down behavior is introduced (see item 3 in the following list).

1. With the start behavior DependsOn defines a start sequencing for resources A and B with an implicit collocation:  
When resource A (source) has to be started then the target resource B is started first. After resource B has become online, resource A (source) is started on the same node.
2. With the stop behavior DependsOn defines a stop sequence for resource A and B:  
When resource B (target) has to be stopped, then source resource A is stopped first. After resource A has become offline, resource B (target) is stopped
3. Force down behavior in case the target resource fails: When target resource B has failed resource A is also stopped. Then a restart is triggered according to the start behavior described in 1.

### Details on the start behavior of the DependsOn relationship

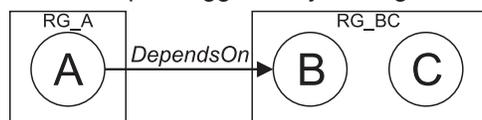
The start sequencing of the DependsOn relationship is controlled via the operational state (OpState) of the target resource. At the time when the operational state of resource B has become online, resource A is started. In addition to the start sequence, DependsOn provides a collocated constraint which causes that resource A has to be started on the same node where resource B was started. Therefore resource B is already started on a node where resource A can be started afterwards. The collocated constraint which is part of the DependsOn relationship corresponds to the behavior of the collocated relationship. For further details on this behavior see “Collocated relationship” on page 71.

In many cases resource A and resource B are members of the same resource group.



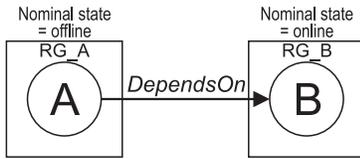
Setting their resource group’s nominal state to online causes that both members A and B are started. Due to the DependsOn relationship from A to B resource B is started first. When resource B’s operational state is online, resource A is started on the same node.

If resource A is a member of resource group RG\_A, and resource B is a member of resource group RG\_BC, and a DependsOn relationship is defined from A to B, then the start behavior of the DependsOn relationship is triggered by setting the nominal state of RG\_A to online.



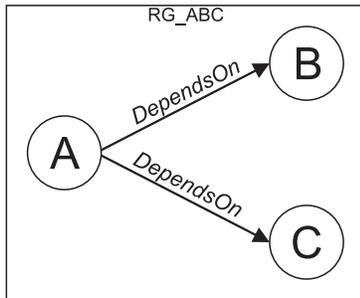
Due to the start sequence of the DependsOn relationship, resource B has to be started first. In case RG\_BC’s nominal state is set to offline there is the following conflict: RG\_BC wants resource B to be offline whereas the DependsOn relationship forces B to be started. IBM Tivoli System Automation resolves the conflict in such a way that the online request is always more important than the offline request. Therefore resource B is started even though other possible group members of RG\_BC will not be started as their group’s nominal state is offline. After resource B is online, IBM Tivoli System Automation will start resource A. Of course the resources A and B are started on the same node. Resource C is not started.

The start order only takes effect in the forward direction of the relationship. In case resource A and resource B are part of different resource groups (A belongs to RG\_A and B belongs to RG\_B),



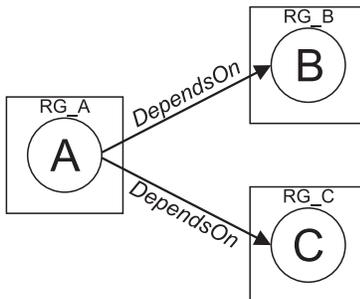
then setting the nominal state of RG\_B to online does not cause any action on resource A as resource B has no forward relationship to resource A. When RG\_A's nominal state is then also set to online, the resource A can be started right away on the same node as resource B is already online

In another scenario it also might be the case that resource A has a DependsOn relationship to resource B and resource C.



In this case starting A requires that both resources B and C are online before IBM Tivoli System Automation can start resource A. For instance A, B, and C are members of the resource group RG\_ABC. Setting the nominal state of RG\_ABC to online causes that resources B and C are started in parallel first. When both resources' operational state is online then resource A is started. All three resources are started on the same node as A has to be started on the same node where B and C are running .

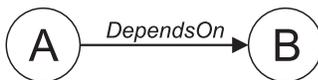
It is also possible that resource A is member of resource group RG\_A, resource B is member of resource group RG\_B and resource C is member of resource group RG\_C.



A has a DependsOn relationship to both B and C. Setting RG\_A's nominal state to online causes resource B and resource C to be started. After both resources B and C are online, A is started on the same node.

### Details on the stop behavior of the DependsOn relationship

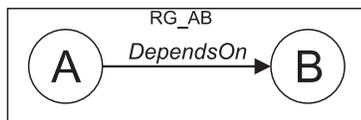
You can control the stop sequencing of the DependsOn relationship via the operational state (OpState) of the source resource.



When the resource A's OpState has become offline, then resource B can be stopped.

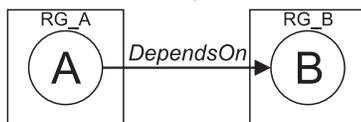
## Managed Relationships

In many cases resource A and resource B are members of the same resource group.



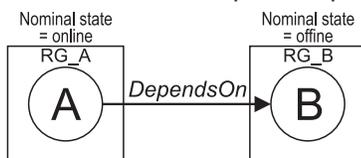
Set the nominal state attribute of the resource group to offline to stop both members A and B. Due to the DependsOn relationship resource A is stopped first. When resource A is offline resource B is stopped

Resource A is a member of resource group RG\_A, and resource B is a member of resource group RG\_B, and a DependsOn relationship is defined from A to B. You can trigger the stop behavior of the DependsOn relationship by setting the nominal state of RG\_B to offline (stopping resources directly is not possible in IBM Tivoli System Automation). Due to the DependsOn relationship resource A should stop first. There is a conflict if the nominal state of resource group A is set to online: RG\_A wants resource A to be online whereas the DependsOn relationship causes it to be stopped.



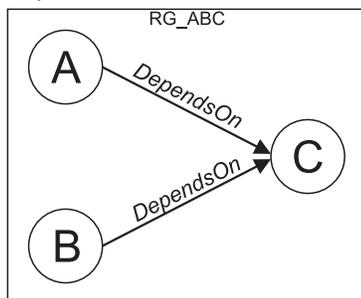
This conflict is resolved in such a way that in IBM Tivoli System Automation always the online request is more important than the offline request. Therefore resource A is kept online and resource B cannot be stopped. Only if the nominal state of RG\_A is set to offline, resource A can be stopped. When resource A is offline, resource B is stopped afterwards.

There is also an implicit stop behavior to consider:



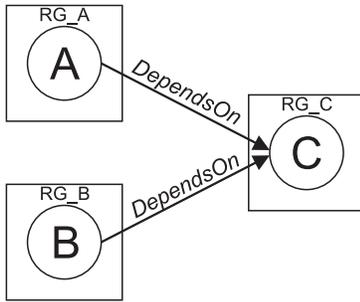
When the nominal state of RG\_A is online, and the nominal state of resource group RG\_B is set to offline, then as described above in the start scenario resource A and B are online. Now the nominal state of RG\_A is set to offline. This causes resource A to stop. In addition, resource B will be stopped. The reason for this is that it was started due to the start request on resource group RG\_A which was propagated via the DependsOn relationship to resource B. Since this resource group RG\_A is set to offline, the start request is removed and the nominal state of offline from resource group RG\_B causes B to be stopped. The DependsOn relationship causes the typical stop behavior: Resource B cannot be stopped before resource A is stopped. Therefore resource A is stopped first. When A is offline, then resource B is stopped.

Another scenario is that resource A and B have a DependsOn relationship to C. Stopping resource C requires that both resources A and B are brought offline first.



RG\_ABC.

Setting the nominal state of RG\_ABC to offline causes that resources A and B are stopped first. When the operational state of both resources is offline, then resource C is stopped. An alternative example is that resources A, B, and C are members of individual resource groups RG\_A, RG\_B, and RG\_C, respectively.

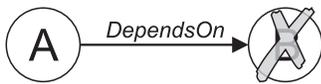


Setting the nominal state of RG\_C to offline triggers the stop behavior of the DependsOn relationship. Here the nominal state of the resource groups RG\_A and RG\_B might overrule the stop behavior. As long as the nominal state of RG\_A or RG\_B is online, resource C cannot be stopped. The reason for this is that in conflict situations an online request always overrules an offline request. Therefore the stop behavior of the DependsOn relationship is deferred until the nominal state of RG\_A and RG\_B is set to offline. When their members A and B are offline, then resource C is also stopped.

**Details on the force down behavior of the DependsOn relationship**

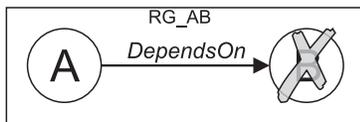
The basic principle of the DependsOn relationship is that source A depends on the functionality of the target resource B. When the target resource B fails then the source resource A cannot function anymore. Therefore, it is not sufficient to restart B. Due to a failure of B resource A will also be forced down. And then both resources will be restarted according to the start behavior: First B, then A.

As example one can define a resource A which has a DependsOn relationship to resource B.



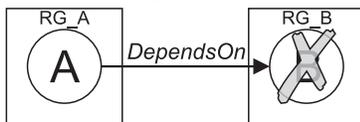
Both resources are online. In case resource B fails, resource A stops and then the normal start behavior occurs. Resource B will be restarted, and then resource A will be started.

Resource A and resource B are members of the same resource group RG\_AB.



In addition, the relationship resource A DependsOn resource B is defined. When RG\_AB is set to online, resource B is started first, then resource A is started. In case resource B fails or goes offline, resource A is also stopped. Afterwards a normal restart is performed with the start sequence of DependsOn: B is started before A is started.

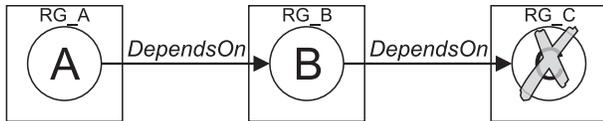
It could also be the case that resource A is member of resource group RG\_A, and resource B is member of resource group RG\_B, and A has a DependsOn relationship with resource B.



When resource group RG\_A is set to online and RG\_B's nominal state is offline, then resource B is started first and resource A is started afterwards. The force down behavior of the DependsOn relationship is triggered by a failure of resource B. It causes that resource A will also be stopped. This will occur even though RG\_A's nominal state is online. In IBM Tivoli System Automation such a conflict is always resolved in such a way that a force down behavior is always more important than the online request of a resource group.

The force down behavior is propagated through chains of DependsOn relationship. Given the following scenario: resource A is member of resource group RG\_A, Resource B is member of resource group RG\_B, and resource C is member of resource group RG\_C with the relationships A DependsOn B and B DependsOn C.

## Managed Relationships



Let's assume that resource group RG\_A is set to online which causes that the three resources C, B, and A were sequentially started and are in online state. Now resource C fails. This causes resources A and B to be forced down: First, resource A is stopped, then resource B is stopped. The reason is that the force down behavior has a higher importance than a normal online request.

### Rule for using the DependsOn relationship

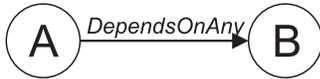
There is one rule for using the DependsOn relationship:

1. If the source or target resource is a group, all members of the group must be collocated.

## DependsOnAny relationship

The behavior of the DependsOnAny relationship is identical to the DependsOn relationship except that it does not provide the collocated constraint for the start sequence. Therefore source and target resources can be started either on the same node or on different nodes.

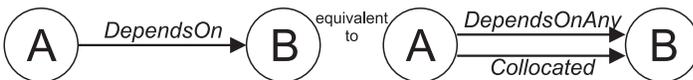
The DependsOnAny relationship provides the following three behavior schemes:



1. With the start behavior DependsOnAny defines a start sequencing for resource A and B **without** a location relationship:  
When resource A (source) has to be started, then the target resource B is started first. After resource B has become online, resource A (source) is started. Note that the only difference to the DependsOn relationship is that Resource A and resource B can be started on different nodes.
2. With the stop behavior DependsOnAny defines a stop sequence for resource A and B:  
When resource B (target) has to be stopped, then source resource A is stopped first. After resource A has become offline, resource B (target) is stopped.
3. Force down behavior in case the target resource fails: When target resource B has failed resource A is also stopped. Then a restart is triggered according to the start behavior described in 1.

See DependsOn relationship for further details on the DependsOnAny relationship.

**Note:** The scenario **A ---> DependsOn ---->B** corresponds to the scenario **A ---> DependsOnAny ----> B** and **A----> Collocated---->B**



## Managed Relationships

### ForcedDownBy relationship

Use the ForcedDownBy relationship to ensure that the source resource will be brought down if the target resource comes offline.

The ForcedDownBy relationship provides the following behavior scheme:



- Resource A must be forced Offline in the event that either the target resource goes Offline unexpectedly or the target resource itself is forced Offline. The stop of resources A and B can happen in parallel. The force down of resource A will be triggered when resource B enters any of the regular down states (Offline) after having previously been in an Online state or in any of the terminal down states (Failed Offline), regardless of its former state.

The ForcedDownBy relationship does not provide a start and a stop behaviour (see “StartAfter relationship” on page 56, “StopAfter relationship” on page 60, and “DependsOn relationship” on page 62).

### Details on the force down behavior of the ForcedDownBy relationship

The basic principle of the ForcedDownBy relationship is that source A must be forced Offline when target resource B goes Offline or fails.

As example one can define a resource A which has a ForcedDownBy relationship to resource B.

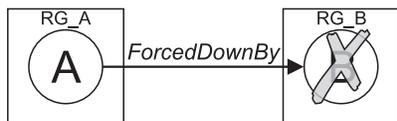


Both resources are Online. In case resource B is stopped or fails, resource A will be forced down.

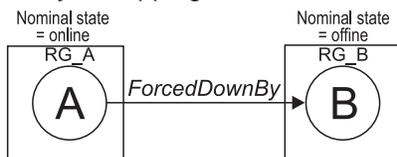
It could also be the case that resource A is member of resource group RG\_A, and resource B is member of resource group RG\_B, and A has a ForcedDownBy relationship with resource B.

When the NominalState attribute of the resource groups RG\_A and RG\_B is set to Online, then resource A and B are started without any dependencies on each other. The force down behavior of the ForcedDownBy relationship is triggered either:

- By a failure of resource B. It causes that resource A will be stopped, too. This will occur even though RG\_A's nominal state is Online. But since the nominal state of RG\_A is still Online in this case, resource A will be restarted by IBM Tivoli System Automation.



- Or by a stopping of resource B.



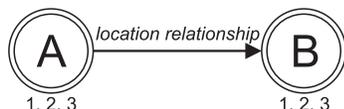
Setting the nominal state of RG\_B to Offline causes that resource A will be stopped, too. This will occur even though RG\_A's nominal state is Online. But since the nominal state of RG\_A is still Online in this case, resource A will be restarted by IBM Tivoli System Automation.

## Location relationships

IBM Tivoli System Automation provides the following relationships which can be used to define location relationships:

- **Collocated**
- **AntiCollocated**
- **Affinity**
- **AntiAffinity**
- **IsStartable**

For example, resources A and B are floating resources which can be started on node1, node2, and node3:



The idea behind these relationships is to define location constraints between resources. Resource types like floating resources, and groups provide a list of nodes on which these can be started. Resource A and resource B are floating resources which can be started on node1, node2, and node3.

A requirement could be that resource A must always be started on the node where resource B is already running or supposed to be running. This behavior can be specified by defining a Collocated relationship from A to B.

The opposite behavior which requires that resource A must not be started on the node where resource B is already running can be specified by defining the AntiCollocated relationship.

In case of the requirement that resource A should - if possible - be started on the node where resource B is running, otherwise it can be started elsewhere, the Affinity relationship is used. Compared to the Collocated relationship, the Affinity relationship has 'soft' location relationships.

The AntiAffinity relationship is used to define that resource A should not be started - if possible - where B is already running. Only if this requirement cannot be satisfied, process A can be started on the node where B is located. Like the Affinity relationship also the AntiAffinity relationship has 'soft' location constraints compared to the AntiCollocated relationship.

The IsStartable relationship defines that source resource A can only be placed on a node where target resource B is startable. This relationship is only considered if the source and target resources have nominal state online. When one of the resources (source or target) does not have a nominal state of online, the IsStartable relationship will be discarded together with the resources that have a nominal state of Offline.

## Conditions IfOnline, IfOffline, IfNotOnline, and IfNotOffline

You can specify the following conditions together with all location relationships except IsStartable. These conditions are:

- |                    |                                                                                                                                                                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>IfOnline</b>    | IfOnline defines that a location relationship is only evaluated when the target resource's OpState is Online. Otherwise the location is ignored. IfOnline does not include states such as Pending Online, and Pending Offline.                   |
| <b>IfOffline</b>   | IfOffline means that a location relationship is only evaluated when the target resource's OpState is either Offline or Failed Offline or Unknown. Otherwise the location relationship is ignored.                                                |
| <b>IfNotOnline</b> | IfNotOnline means that a location relationship is only evaluated when the target resource is <b>not</b> in an Online state. IfNotOnline includes states such as Pending Online and Stuck Online. Otherwise the location relationship is ignored. |

## Managed Relationships

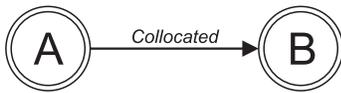
**IfNotOffline** IfNotOffline means that a location relationship is only evaluated when the target resource is **not** in an Offline or Failed Offline or Unknown state. Otherwise the location relationship is ignored.

### Rules for using the Location relationships

1. The source of a location relationship is either a member of a resource group or a resource group. See “What is a resource group?” on page 35 for more information about resource groups.
2. The target of a location relationship is either
  - a member of a resource group or a resource group.
  - an RMC resource (which is not a managed resource) which has to provide a start /stop method and an OpState attribute.
3. The target resource of a Location relationship *cannot* be an equivalency.
4. If the source or target resource is a group, all members of the group must be collocated.

## Collocated relationship

IBM Tivoli System Automation uses the Collocated relationship to ensure that the source resource and its target resource are located on the *same* node. The Collocated relationship provides the following behavior scheme:



- The Collocated relationship defines that on start of resource A it can only be started on the node where resource B is already running.

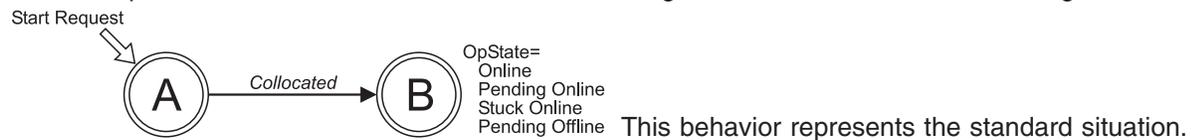
The Collocated relationship can be used together with a Condition attribute as described on page 72.

### Details on the principle behavior of the Collocated relationship

The following describes in detail four states the Collocated relationship can take:

#### Case I:

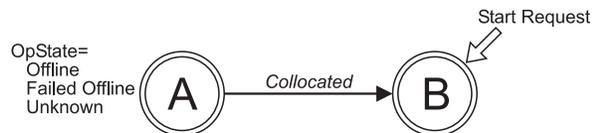
On start of resource A place it on the same node where resource B is already running. 'Running' means that the OpState of resource B is either Online, Pending Online, Stuck Online, or Pending Offline.



The collocated relationship tries to optimize the node selection based on predictions for future situations. Here the following cases are possible:

#### Case II:

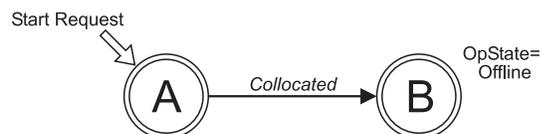
Resource B is started and resource A is in an Offline, Failed Offline or Unknown state.



Generally you would expect that the node selection for resource B is independent of resource A. But when IBM Tivoli System Automation selects a node for resource B, then a node is selected on which resource A could also be started in future. The reason for this prediction approach is that it later simplifies the start behavior for resource A: if no error situation occurs it is ensured that after resource B was started, resource A can be started on the same node where resource B runs.

#### Case III:

Resource A is started and resource B is in an offline state.



Theoretically resource A could now be placed on any of the nodes of its node list since A cannot be bound to a node on which B is running. Here again the prediction approach tries to find a node location for resource A where resource B could also be started in future. Therefore, IBM Tivoli System Automation determines the same node location for both resources A and B even though it will only start resource A. The internal IBM Tivoli System Automation behavior works as follows: When resource A has to be started IBM Tivoli System Automation determines a node location for both resources A and B, and then starts resource A. (Note: the start of resource B is not driven by the collocated relationship. This is done by another start/stop relationship or a group behavior).

## Managed Relationships

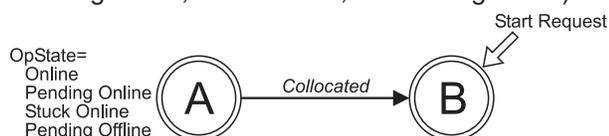
A summary of the prediction approach is: If either resource A or resource B is started, and the other resource is in an offline state, then IBM Tivoli System Automation determines a node location where both resources A and B are logically bound before one of them is started.

Note that the optimization for the node location is just a prediction based on the current circumstances. The prerequisites the decision of the node selection was based upon can change over time.

A scenario of an erroneous prediction of a node selection could be the following: Resources A and B are floating resources and can be located on node 1, 2, 3. The relationship A -- Collocated ---> B is defined. Now resource B has to be started. Due to the collocated relationship IBM Tivoli System Automation could select node1 for resources A and B. Then resource B is started. After a while an administrator usage error causes that resource A cannot be started on node 1 anymore. The OpState of the resource on node 1 is FailedOffline. Then a request causes that resource A has to be started. Since resource A cannot be started on node 1 anymore, a conflict situation occurs which has to be solved as described later.

### Case IV:

Another possible state could be that resource A is already in a running state (OpState is either Online, PendingOnline, StuckOnline, or PendingOffline) when resource B is started.



At the time when resource A was started resource B already got the same node selected. If no error occurred resource B can be started there. If there was a problem which prevented resource B from starting on the previous selected node, the resource gets unbound and at start time of resource B a new node location has to be found. This means that resource B can be started on another node.

The following relationships with conditions can be defined:

- **Collocated/IfOnline**

The relationship A ---> Collocated/IfOnline -----> B means that the location relationship is only considered when resource B is in a online state. Otherwise the location relationship is ignored. IfOnline does not include states such as Pending Online, and Pending Offline.

- **Collocated/IfOffline**

The relationship A ---> Collocated/IfOffline -----> B means that the location relationship is only valid when resource B is in an Offline, Failed Offline or Unknown state.

- **Collocated/IfNotOnline**

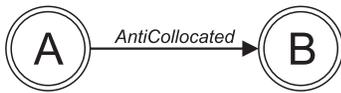
The relationship A ---> Collocated/IfNotOnline -----> B means that the location relationship is only valid, when resource B is not in an online state.

- **Collocated/IfNotOffline**

The relationship A ---> Collocated/IfNotOffline -----> B means that the location relationship is only valid, when resource B is not in an Offline, Failed Offline or Unknown state.

## AntiCollocated relationship

IBM Tivoli System Automation uses the AntiCollocated relationship to ensure that the source resource and its target resource are located on *different* nodes. The AntiCollocated relationship provides the following behavior scheme:



- The AntiCollocated relationship defines that on start of resource A it can only be started on a different node where resource B is already running.

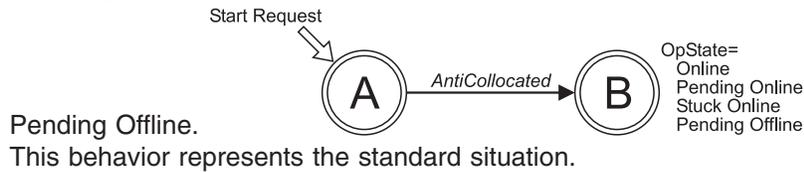
The AntiCollocated relationship can be used together with the Condition attribute as described on page 74.

### Details on the principle behavior of the AntiCollocated relationship

The following describes in detail four states the AntiCollocated relationship can take:

#### Case I:

On start of resource A place it on a different node than the one where resource B is currently running. 'Running' means that the OpState of resource B is either Online, Pending Online, Stuck Online, or



The AntiCollocated relationship tries to optimize the node selection based on predictions for future situations. Here the following cases are possible:

#### Case II:

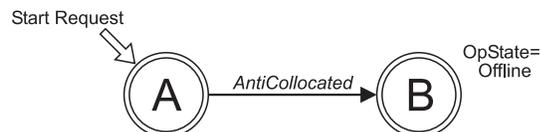
Resource B is started and resource A is in an Offline, Failed Offline or Unknown state.



Generally you would expect that the node selection for resource B is independent from resource A. But when IBM Tivoli System Automation selects a node for resource B, then a node is selected which allows that resource A can be started on another node in the future. The reason for this prediction approach is that it later simplifies the start behavior for resource A: if no error situation occurs it is ensured that after resource B was started, resource A can be started on another node where resource B is not running. This corresponds to the description of **Case I**.

#### Case III:

Resource A is started and resource B is in an offline state (Offline, Failed Offline).



Theoretically resource A could now be placed on any of the nodes of its node list. Here again the prediction approach tries to find a node location for resource A that allows resource B to be started on another node in the future. Therefore IBM Tivoli System Automation determines a node location for resource B even though it will only start resource A.

## Managed Relationships

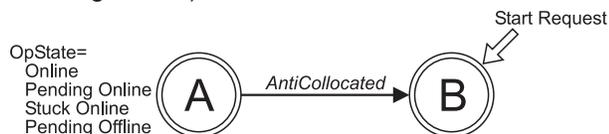
Summary of the prediction approach is:

If resource A is in an offline state and either resource A or resource B is started (see Case II and Case III), then IBM Tivoli System Automation determines a different node location for both resources A and B before one of them is started

As already mentioned in the description of the Collocated relationship it may happen that the prediction based on the current circumstances might be wrong over time. Nevertheless the prediction approach will simplify the automation behavior in most cases.

### Case IV:

Resource A is already in a running state (OpState is either Online, Pending Online, Stuck Online, or Pending Offline) when resource B is started.



At the time when resource A was started (see Case III) resource B already got another node selected. If no error occurred, resource B can be started there. If there was a problem which caused that resource B cannot be started anymore on its previous selected node, at start time of resource B a new node location is found. This means that resource B can be started anywhere, even where resource A is already running.

The following relationships with conditions can be defined:

- **AntiCollocated/IfOnline**

The relationship A ---> AntiCollocated/IfOnline -----> B means that the location relationship is only valid, when resource B is in a online state. Otherwise the location relationship is ignored. IfOnline does not include states such as Pending Online, and Pending Offline.

- **AntiCollocated/IfOffline**

The relationship A ---> AntiCollocated/IfOffline -----> B means that the location relationship is only valid when resource B is in an Offline, Failed Offline or Unknown state.

- **AntiCollocated/IfNotOnline**

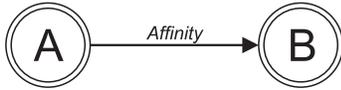
The relationship A --->AntiCollocated/IfNotOnline -----> B means that the location relationship is only valid, when resource B is not in an online state.

- **AntiCollocated/IfNotOffline**

The relationship A ---> AntiCollocated/IfNotOffline -----> B means that the location relationship is only valid, when resource B is not in an Offline, Failed Offline or Unknown state.

## Affinity relationship

The Affinity relationship provides the following behavior scheme:



- The Affinity relationship defines that on start of resource A the same node is chosen where resource B is already running, if possible. If other location relationships are inhibiting this, resource A can also run on another node.

The Affinity relationship is very similar to the Collocated relationship. Therefore the Affinity relationship defines a soft location relationship whereas the Collocated relationship is a hard location relationship.

The Affinity relationship can be used together with the Condition attribute (described in “Condition attribute” on page 55).

The following relationships with conditions can be defined:

- **Affinity/IfOnline**

The relationship A ---> Affinity/IfOnline -----> B means that the location relationship may only be considered when resource B is in a online state. Otherwise the location relationship is ignored. IfOnline does not include states such as Pending Online, and Pending Offline.

- **Affinity/IfOffline**

The relationship A --->Affinity/IfOffline -----> B means that the location relationship may be only valid when resource B is in an Offline, Failed Offline or Unknown state.

- **Affinity/IfNotOnline**

The relationship A ---> Affinity/IfNotOnline -----> B means that the location relationship may only be valid, when resource B is not in an online state.

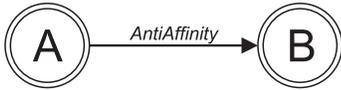
- **Affinity/IfNotOffline**

The relationship A ---> Affinity/IfNotOffline -----> B means that the location relationship may only be valid, when resource B is not in an Offline, Failed Offline or Unknown state.

## Managed Relationships

### AntiAffinity relationship

The AntiAffinity relationship provides the following behavior scheme:



- The AntiAffinity relationship defines that on start of resource A a different node than the one where resource B is already running is chosen, if possible. If other location relationships are inhibiting this, resource A can also run on the same node.

The AntiAffinity relationship is very similar to the AntiCollocated relationship. Therefore the AntiAffinity relationship defines a soft location relationship whereas the AntiCollocated relationship is a hard location relationship.

The AntiAffinity relationship can be used together with the Condition attribute (described in “Condition attribute” on page 55).

See also “Location relationships” on page 69.

The following relationships with conditions can be defined:

- **AntiAffinity/IfOnline**

The relationship A ---> AntiAffinity/IfOnline -----> B means that the location relationship may only be valid, when resource B is in a online state. Otherwise the location relationship is ignored. IfOnline does not include states such as Pending Online, and Pending Offline.

- **AntiAffinity/IfOffline**

The relationship A ---> AntiAffinity/IfOffline -----> B means that the location relationship may only be valid when resource B is in an Offline, Failed Offline or Unknown state.

- **AntiAffinity/IfNotOnline**

The relationship A ---> AntiAffinity/IfNotOnline -----> B means that the location relationship may only be valid, when resource B is not in an online state.

- **AntiAffinity/IfNotOffline**

The relationship A ---> AntiAffinity/IfNotOffline -----> B means that the location relationship may only be valid, when resource B is not in an Offline, Failed Offline or Unknown state.

## IsStartable relationship

The IsStartable relationship provides the following behavior scheme:



- The IsStartable relationship defines that resource A can only be placed on a node where resource B is startable when the resources A and B have a nominal state of online.

IsStartable does not imply that the target resource will actually be startable at a later time. This is because resource failures may prevent all of its relationships from being resolved at that later time.

See also “Location relationships” on page 69.

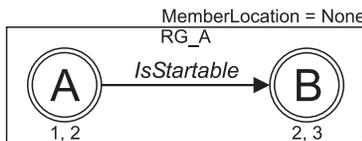
### Details on the principle behavior of the IsStartable relationship:

The IsStartable relationship causes the following behavior:

The IsStartable relationship defines that the source resource can only be placed on a node where the target resource is startable. This relationship is only considered if the source and target resources have nominal state online. When one of the resources (source or target) does not have a nominal state of online, the IsStartable relationship will be discarded together with the resources that have a nominal state of Offline.

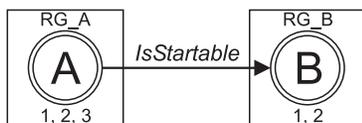
The following example explains the behavior of the IsStartable relationship:

Resource A and resource B are floating resources and members of the same resource group RG\_A. Resource A can run on node1 and node2, and resource B on node2 and node3. An IsStartable relationship is defined from resource A to resource B.



Both members are started when the nominal state of the resource group is set to online. Based on the IsStartable relationship resource A and resource B are started on node2, as this node is the intersecting node for both resources. When resource B is in failed offline state on node 2, starting of the resource group RG\_A does not start resource A, as no node exists where both resource A and resource B can be started.

The following example provides more information about the IsStartable relationship. In this scenario resource A can run on node1, node2, and node3, and is member of the resource group RG\_A. Resource B can run on node1 and node2, and is member of resource group RG\_B. An IsStartable relationship from resource A to resource B is defined.



The following describes the possible states in this example:

- RG\_A’s nominal state is set to online while RG\_B is offline. Since the IsStartable relationship is only taken into consideration if the source and target resources have a nominal state of online (here RG\_A and RG\_B), and RG\_B’s nominal state is offline in this case, the relationship will be ignored. Therefore resource A can either start on node1, or node2, or on node3.
- RG\_A’s nominal state is set to online while RG\_B is already online. In this case the IsStartable relationship is taken into account and IBM Tivoli System Automation starts resource A on a node where resource B can start (node1 or node2).

## Managed Relationships

- Due to a problem resource B cannot start on node1 and node2, and the nominal state of RG\_B is online. Starting resource group RG\_A causes that resource A cannot become online because resource B is not startable on the intersecting nodes node1 and node2.
- Due to a problem resource B cannot start on node1 and node2, and the nominal state of the resource group RG\_B is offline. When resource group RG\_A's nominal state is set to online, IBM Tivoli System Automation discards resource B, and the IsStartable relationship is ignored because of the desired state offline of resource group RG\_B.

## Creating and administering relationships

### Creating a relationship

To create a relationship between a source resource and one or more target resources, you use the **mkrel** command.

The source resource must be a member of a resource group. The target resource does not have to be in a resource group.

For example, to define an **AntiCollocated** relationship for a source resource **FloatWebServerA** of class **IBM.Application** to target resource **FloatWebServerB** of class **IBM.Application** with condition 'IfOnline' and name 'Rel1', you would enter:

```
mkrel -p anticollocated -o ifonline -S IBM.Application:FloatWebServerA -G IBM.Application:FloatWebServerB Rel1
```

For further details, see either the **mkrel** man page, or “mkrel” on page 187.

### Listing a relationship

To list a relationship, you use the **lsrel** command.

If you do not enter a relationship name, all relationships currently defined will be listed:

```
lsrel
```

Displaying Managed Relations :

```
Name Class:Resource:Node[Source] ResourceGroup[Source]
Rel1 IBM.Application:FloatWebServerA RG_WebApp
```

If you specify a relationship name with the **-M** option, the persistent attributes of the specified relationship will be listed. For example, to list the attributes of the relationship **Rel1** you would enter:

```
lsrel -M Rel1
```

Displaying Managed Relationship Information:  
for Managed Relationship "Rel1".

```
Managed Relationship 1:
Name = Rel1
Class:Resource:Node[Source] = IBM.Application:FloatWebServerA
Class:Resource:Node[Target] = {IBM.Application:FloatWebServerB}
Relationship = AntiCollocated
Conditional = IfOnline
ResourceGroup[Source] = RG_WebApp
```

You may get a similar output if you list all relationships where **IBM.Application:FloatWebServerA** is the source of (**-S** option):

```
lsrel -S IBM.Application:FloatWebServerA
```

Displaying Managed Relationship Information:

```
Managed Relationship 1:
Name = Rel1
Class:Resource:Node[Source] = IBM.Application:FloatWebServerA
Class:Resource:Node[Target] = {IBM.Application:FloatWebServerB}
Relationship = AntiCollocated
Conditional = IfOnline
ResourceGroup[Source] = RG_WebApp
```

For further details, see either the **lsrel** man page, or “lsrel” on page 168.

## Managed Relationships

### Changing a relationship

To change a relationship, you use the **chrel** command.

For example, to change a relationship named Rel1 (created above) to AntiAffinity, you would enter:

```
chrel -p antiaffinity Rel1
```

For further details, see either the **chrel** man page, or “chrel” on page 154.

### Removing a relationship

To remove a relationship between source and target resources, you use the **rmrel** command.

For example, to remove a relationship for a source resource **FloatWebServerA** of class IBM.Application, you would enter:

```
rmrel -S IBM.Application:FloatWebServerA
```

For further details, see either the **rmrel** man page, or “rmrel” on page 202.

---

## Chapter 8. How IBM Tivoli System Automation processes the system information

This chapter describes in its first part the binding algorithm, also called binder. This is a IBM Tivoli System Automation internal function responsible for the node placements of all resources. The second part of this chapter discusses events which allow a resource group to become online. The third part of this chapter is about the behavior patterns of IBM Tivoli System Automation .

---

### Location relationship resolution: Binding algorithm

The binder is invoked whenever a resource has to be started for which IBM Tivoli System Automation has not yet assigned a node placement. Resources which have a node location assigned are also called bound. An example is a floating resource A which could potentially run on several nodes. Here, the binding algorithm has to determine (bind) a node location for the floating resource considering all of its location relationships. Based on previous binding algorithm runs which already bound resources, a new solution has to be found. Binding solutions do not necessarily have to be non-ambiguous. Many constellations allow several alternative solutions where it is arbitrary which one is chosen by IBM Tivoli System Automation.

An example for an ambiguous scenario is a resource group with a collocated location relationship containing two floating resources A and B which can run on node1 and node2. When the group is started two alternative solutions are possible: either A and B are bound to node1 or both are bound to node2.



If the binding algorithm can find a solution for the node placement of all involved resources, the resource(s) are started. It is obvious that location relationships can lead to conflicting situations which have to be resolved.

For example, two floating resources A and B could be located on node1 and node2. Due to a performance constraint both resources must never run on the same node. You therefore must specify an AntiCollocated relationship from A to B and from B to A. It is assumed that resource A is already running on node 1. Then node 2 fails. If now a user started resource B, it would cause a location relationship conflict as resource A and B cannot be started on the same node. A perfect solution where both resources are running cannot be found in this situation. Therefore IBM Tivoli System Automation performs a specific conflict resolution called discarding step to resolve the situation.

It is possible that resources already online are part of the problem. These resources get an additional priority bonus of 10 to their priority set by the resource group.

The following section describes in detail IBM Tivoli System Automation's solution finding for location relationships and its conflict resolution handling. This whole process is called the binding algorithm

The binding algorithm consists of several steps:

1. **Discovery step: Determining configuration subsets for which the location relationships can be independently solved**

The discovery algorithm consists of several substeps:

- a. **Step 1a: Find all involved resources (configuration subset)**

Location relationships might separate a customer configuration into several configuration subsets which can be solved independently. The reason for this is that location relationships often affect only a subset of resources of the configuration. An example is the configuration with A -->

## IBM Tivoli System Automation logic

Collocated --> B -->, B--> Collocated --> C, and D --> Collocated --> E. Here the location relationships for A, B, and C can be independently solved from D and E. For those two subsets all following steps are separately made .

b. **Step 1b: Ignore all resources with OpState = Failed Offline**

It is obvious that all instances which have an OpState of Failed Offline cannot contribute to a binding solution. Those instances are removed from the configuration subset which is used to find a binding solution. An example for this is a Resource Group R1 containing two floating resources A and B which can run on node1 and node2. The Group has a collocated parameter set which means that resource A and B have to be started on the same node. Assume that node2 is broken down which causes that the constituents of the floating resources A and B on node2 are in a Failed Offline state. Therefore those are removed from the configuration subset since instances on node2 will not help to solve the binding problem

c. **Step 1c: Cleaning up resource groups which cannot be started.**

In case mandatory resource group members are in a Failed Offline state, the resource group cannot be started according to the resource group behavior. Therefore all other resource group members of such a resource group have to be stopped.

An example is resource group R1 with floating resource A and B as described above. If floating resource A cannot be started on either of the nodes due to an application error, and if it is a mandatory resource group member, the floating resource B is also stopped (see resource group members).

### 2. Perfect solution step: Try to find a 'perfect' solution

At first, the recovery resource manager tries to find a perfect solution of all involved location relationships for a configuration subset. In this step it tries to find bindings as described in "Location relationships" on page 69. Since in this first step the goal is to find a perfect solution, all Affinity and AntiAffinity relationships are treated as if they were pure Collocated and AntiCollocated relationships. In addition, even resources which are Offline and are not intended to start are also tried to be bound if necessary. If no location relationship conflict occurs, the necessary resources are bound and the binding algorithm is done. As a next step IBM Tivoli System Automation can start those resources which have to be started.

There are situations in which this binding step gets into a conflict situation with contradicting constraints that cannot be overcome. To resolve this IBM Tivoli System Automation provides a discarding step consisting of several substeps as described below.

### 3. Discarding step: Resolve situations with conflicting location relationships

The discarding step consist of a a number of substeps:

a. **Step 3a: Ignore all Affinity and AntiAffinity relationships**

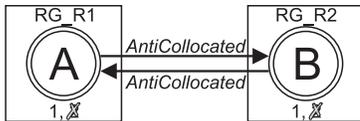
The first approach to overcome the conflicting situation is to ignore all Affinity and AntiAffinity relationships since those are 'soft' location relationships. Based on the previous bindings IBM Tivoli System Automation tries to find a solution for the resources which have to be bound. Since all Affinity and AntiAffinity relationships are ignored, the location relationships are simplified and the probability that a binding solution can be found is increased. In case a solution can be found, the sacrifice step is left. But there is still the chance that the conflicting situation cannot be overcome. Then the next level of the sacrifice step is reached.

b. **Step 3b: Ignore all resources with OpState = Offline and which do not have to be started**

If ignoring all Affinity and AntiAffinity relationships did not help to find a solution for the binding problem (see step 3a) then the next level is to ignore all resources from the binding evaluation which are Offline and which are currently not intended to be started. This increases the possibility that a binding solution can be found.

In case there is a binding solution available, then the sacrifice step is left. Otherwise the next step of the discarding process is reached.

An example is ResourceGroup R1 containing floating resource A, and resource group R2 containing floating resource B, and a relationship A AntiCollocated B. Floating resource A and B can run on node1 and node2, but node2 is broken down.



Now R1's nominal state is set to online which causes that resource A has to be bound before it can be started. At first IBM Tivoli System Automation tries to find a perfect solution. Therefore it tries to bind A and B. But here a solution cannot be found. Then IBM Tivoli System Automation ignores all Affinity and AntiAffinity relationships, which does not provide a solution either. Then it ignores all resources with an Offline state and which do not have to be started. This causes that resource B is ignored for the evaluation. Now it is possible to bind resource A to node1.

c. **Step 3c: Stopping least important resource group members**

The next level of the sacrifice step is to stop resource group members and ignore those members in the binding evaluation. Since each resource group has a priority value assigned, resource group members of the group(s) with the lowest priority are stopped first, and then a binding solution is tried to find without them. In case this does not satisfy the binding constraints, resource group members with the next group priority level are chosen. In addition to the priority schema, the stopping and removing of resource group members is performed in two substeps: First only all non-mandatory members of group priority level are stopped and ignored for the binding solution. Only if this does not help to solve the conflicting situation, then also the mandatory members of the same group priority level are stopped and removed from the binding evaluation. If the conflict still exists, the next lowest group priority is taken and all group members are stopped as just described. This is done iteratively until a binding solution can be found.

**Hints:**

- Outer groups must have the same or higher priority than the inner groups. Otherwise the outer groups would be discarded before the inner groups. But if the outer groups are discarded, the inner groups are stopped automatically.
- Non-mandatory members should have a lower priority than mandatory members. Otherwise mandatory members may be discarded.

---

## Events that might allow a resource group to become Online

All "root" resource groups whose NominalState attribute is Online will be *automated*: this means, an attempt will be made to start such "root" resource groups and the managed resources within these groups, *providing* the managed relationships of the managed resources within the resource groups can be satisfied.

If a resource group or a managed resource cannot be brought Online (when one or more of its member resources fail completely to reach the Online required state), the resource group is in an Offline state. The resource group remains Offline until an event occurs that informs IBM Tivoli System Automation that it should again attempt to start the resource group.

These are the possible events which might cause an Offline resource group to become Online:

- Changing the *AllowedNodes attribute* (explained in "AllowedNode attribute" on page 38) of the resource group, for example to include an additional node where the resource group can be started. For details, see "chrg" on page 159.
- Removing a managed resource from a resource group. As a result, the other member resources might then be startable because a resource which cannot be started is removed. For details, see "rmrgmbr" on page 208.
- Adding a resource to a resource group which is a target resource of a managed relationship, but which is currently not a member of a resource group. It will be automated by IBM Tivoli System Automation then. As a result, the other member resources might then be startable because a managed relationship can be satisfied. For details, see "addrgmbr" on page 147.
- Starting a resource which is not controllable by IBM Tivoli System Automation and which has a managed relationship to a resource group member. As a result, the resource goes Online and the

## IBM Tivoli System Automation logic

managed relationship is satisfied. This might allow the resource group to also be brought online. To start the resource, you might use the RMC **startsrc** command (for details, see the man page for this command).

- Adding a constituent to an aggregate that will make the aggregate resource available on more nodes, and may result in IBM Tivoli System Automation being able to satisfy all of the managed-relationship constraints. If the constituent is a piece of hardware, this would require that you install the hardware, or define it correctly. If the resource is a floating resource, you add a constituent by adding a node name to the `NodeNameList` attribute. For details, refer to the RMC documentation and man pages.
- A new resource is found by an equivalency that uses a dynamic select string. As a result, this resource is added to the equivalency, and may resolve a managed relationship to this equivalency.
- Making a managed resource `NotMandatory`, which allows this resource to be sacrificed. As a result, the other managed resources can be started. For details, see “`chrgmbr`” on page 162.
- Performing a Reset on an aggregate or one of its constituents after a failure has been fixed. As a result, the resource will be `Offline`, and can then be started by IBM Tivoli System Automation. For details, see the man page for the RMC **resetsrc** command.
- A node that was `Offline` becomes `Online`. As a result, IBM Tivoli System Automation may be able to bring the resource group online.
- Changing the priority attribute of a resource group (explained in “Priority attribute” on page 40). A resource group might not be startable because of a priority conflict with another resource group. In this case increase the priority of the group you want to start or decrease the priority of the other group.
- Stopping a higher-priority resource group, which prevents a lower-priority resource group from starting. As a result, a managed relationship conflict is avoided. For details, see “`chrg`” on page 159.

If a resource group is currently at its `NominalState` value, the following events may cause additional automation actions:

1. The `NominalState` attribute value changes from `Offline` to `Online`.
2. The `NominalState` attribute value changes from `Online` to `Offline`.

---

## Behavior Patterns of IBM Tivoli System Automation for Multiplatforms

This section describes how IBM Tivoli System Automation behaves and reacts in certain situations.

### General considerations

The following describes issues relating to the `StartCommand`, `MonitorCommand`, and `StopCommand`.

### StartCommand issues

IBM Tivoli System Automation for Multiplatforms uses the command specified in the `StartCommand` attribute of a resource to bring a resource `Online`. The `StartCommand` of a resource is executed in the following situations:

- Immediately, after the `NominalState` attribute of a resource group has been changed to `Online`, and all start dependencies of this resource are satisfied.
- Immediately, after the `OpState` of a resource has changed from `Online` to `Offline` caused by a failure of the resource (Note: This is not true, if the `NominalState` of the resource group has been changed to `Offline`, or if the resource has been stopped/forced down by IBM Tivoli System Automation to satisfy a dependency to another resource.)
- If the `StartCommand` had already been executed for a resource, but this resource is still `Offline` the time the `Online` time-out is reached **and** the `RetryCount` for the number of `StartCommand` executions has not been reached. The `Online` time-out for a resource is calculated by the following formula:

```
MAX(StartCommandTimeout, MonitorCommandPeriod) + 10
```

Note that this is not an absolute value, as IBM Tivoli System Automation does not use a real timer. The IBM Tivoli System Automation daemons are woken up frequently, and this may result in the `Online`

time-out being within a range of 10 - 13 seconds. Also note also that this Online time-out is only evaluated in case the resource did not change its OpState during the previous execution of the StartCommand (e.g. to Pending Online or Online). Then the Online time-out timer is canceled, and the StartCommand of the resource is executed immediately after the OpState of the resource has changed to Offline again.

The StartCommand is synchronously executed by IBM Tivoli System Automation, meaning that IBM Tivoli System Automation waits for the command to finish and obtains knowledge of any return code. Furthermore there is an attribute StartCommandTimeout for each resource which determines how long it takes at maximum to execute the StartCommand. If the StartCommand does not return within the StartCommandTimeout time period, the StartCommand is killed by IBM Tivoli System Automation using the SIGKILL command. If this happens, a message is logged into the system log of that node. However this may lead to problems, if an application process that is started within the StartCommand does not return control. In this case the application process is killed every time after the StartCommandTimeout is reached, resulting in IBM Tivoli System Automation cannot start this application as a resource. To get this working the application process has to be detached from the calling StartCommand by using one of the following methods:

- Redirect all file handles to a file and start the application process in the background, e. g.  

```
/usr/bin/application >/outputfile 2>&1 &
```
- Create a little wrapper application that uses the 'setuid()' C-function to get the application process detached from the calling StartCommand.
- If the above methods do not work or are not appropriate for a certain application, then the value of the attribute RunCommandsSync of the resource has to be set to 0. In this case IBM Tivoli System Automation does not honor the StartCommandTimeout attribute for this resource, and therefore the StartCommand and all its child processes may stay until forever on this node. But in this case IBM Tivoli System Automation does not wait for any return code of this StartCommand and therefore a resource is not failed over even if the StartCommand failed. Instead, the StartCommand is executed again if the resource does not come Online during the Online time-out period until the RetryCount is reached.

### MonitorCommand issues

The MonitorCommand of an IBM.Application resource is used by IBM Tivoli System Automation to determine the OpState of this resource on a node. IBM Tivoli System Automation starts monitoring a resource the time it is added to a resource group. The monitoring is performed on any node this resource is allowed to run on (NodeNameList). After the first execution, the MonitorCommand is executed in a frequency defined in the MonitorCommandPeriod attribute. This monitoring of the resource now goes on forever on every node the resource is defined on until the resource is removed from the resource group.

Starting with IBM Tivoli System Automation release 1.2 the MonitorCommand is also executed immediately after the StartCommand or StopCommand of a resource has finished execution (only for synchronous commands, if RunCommandsSync attribute for this resource is set to 1). This has been introduced to enhance the performance of the start/stop of an entire resource group, as now the OpState of the resource is immediately checked after the StartCommand or StopCommand has finished. After this execution of the MonitorCommand, the frequency of MonitorCommandPeriod seconds is honored again, meaning that the next MonitorCommand is executed after MonitorCommandPeriod seconds. There are two issues regarding the MonitorCommand that should be kept in mind to avoid trouble or strange behavior:

1. The MonitorCommand is executed on all nodes the resource is allowed to run on (that are defined in the resources NodeNameList attribute). If a resource should be down (NominalState of the resource group is Offline) and an operator starts this resource manually, IBM Tivoli System Automation will notice this with the MonitorCommand of that resource and finally execute the StopCommand for this resource to bring it back Offline again. This is how IBM Tivoli System Automation is designed: automate resources. If it is necessary to bring a single resource of a resource group Online (or Offline), for instance to perform a backup, then an IBM Tivoli System Automation request has to be used (rgreq command). This will overrule the NominalState of the resource group and will allow a resource of a resource group to be started, even if the NominalState of the resource is Offline.

## IBM Tivoli System Automation logic

2. There is an attribute MonitorCommandTimeout, which will result in a SIGKILL command against a running MonitorCommand if this has not finished before the time out has been reached. If the MonitorCommand has been killed, a message will be logged into the system log of that node, and the OpState of the resource will be set to Unknown. Since the OpState Unknown is serious, IBM Tivoli System Automation will not go on automating this or any dependent resource. If this message is seen frequently in the system log, the value of the MonitorCommandTimeout attribute should be checked and adjusted if necessary.

### StopCommand issues

The StopCommand is by default synchronously executed by IBM Tivoli System Automation, meaning that IBM Tivoli System Automation waits for the command to finish and obtain knowledge of any return code. Furthermore there is an attribute StopCommandTimeout for each resource which determines how long it takes at most to execute the StopCommand. If the StopCommand does not return within the StopCommandTimeout time period, the StopCommand is killed by IBM Tivoli System Automation using the SIGKILL command. If this happens a message is logged into the system log of that node. However, this may lead to problems, as the StopCommand is only called once, meaning the RetryCount attribute value is not honored. If resource could not be stopped because of the StopCommandTimeout or for another reason IBM Tivoli System Automation cannot go on automating this and any dependent resource. Therefore it is important to choose an appropriate value for the StopCommandTimeout attribute and make sure that the StopCommand will really stop the resource, if it is called by IBM Tivoli System Automation.

### How IBM Tivoli System Automation reacts to the possible OpState changes of a resource that is online on a node

The following sample configuration is used for the discussion in the next section:

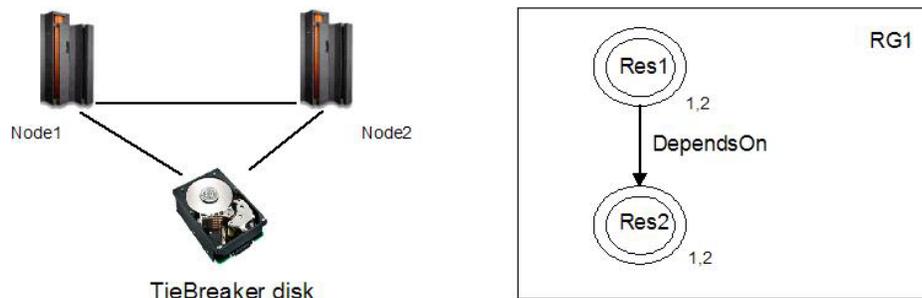


Figure 3. Sample configuration setup

The setup of this sample configuration is as follows:

- Cluster of 2 nodes.
- Disk TieBreaker

- Node1: production system.
- Node2: stand-by system.
- Resource group: **RG1**. with
  - Floating resource: Res1
  - Floating resource: Res2
  - Relationship: Res1 **DependsOn**. Res2
- Resources are **Online** on Node1.

The following diagram shows for reference the usual flow of the OpStates of the resources under control of IBM Tivoli System Automation:

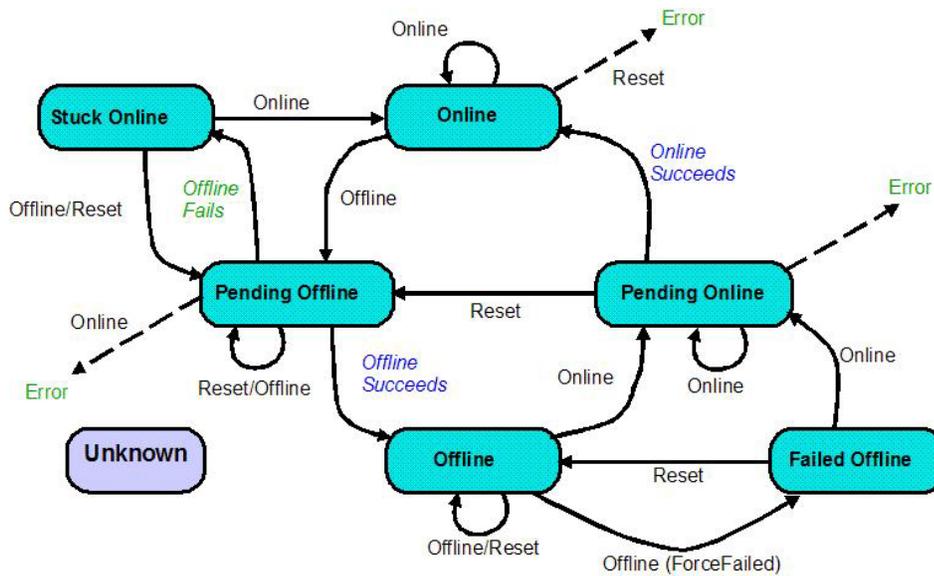


Figure 4. resource state diagram of IBM Tivoli System Automation

There are seven values the OpState of a resource can have. The OpState of a resource is determined by IBM Tivoli System Automation with the MonitorCommand, the actual OpState of a resource is provided to IBM Tivoli System Automation with the return code of the MonitorCommand. Note that it is sufficient for a MonitorCommand to return the OpState values Online and Offline to IBM Tivoli System Automation, the other OpState values a resource can have may be exploited optionally

Some OpState values like Unknown or Failed Offline may be also set by IBM Tivoli System Automation, for instance if the OpState Unknown is set for a resource and if the MonitorCommand for this resource timed out. IBM Tivoli System Automation therefore has no knowledge about the OpState of this resource anymore.

The following two tables will illustrate how IBM Tivoli System Automation reacts to an OpState change of the resources Res1 and Res2 from the example above. Note that the tables in this chapter contain all possible OpState values of a resource, even if a particular OpState does not make sense in this situation. The columns which contain these unlikely OpState values have been meshed, to make it clear, that an OpState is unlikely in a special situation.

## IBM Tivoli System Automation logic

### OpState change of resource Res1

The current status of Res1 and Res2 on node1 is Online. The following table shows the actions that IBM Tivoli System Automation performs depending on the return value of the MonitorCommand for Res1.

Table 7. System Automation actions regarding OpState changes of resource Res1

| MonitorCommand (OpState) |    | First action of System Automation                | Second action of System Automation                                    |
|--------------------------|----|--------------------------------------------------|-----------------------------------------------------------------------|
| RC=0 (Unknown)           | => | Nothing, wait for next MonitorCommand with RC<>0 | Nothing                                                               |
| RC=1 (Online)            | => | Nothing                                          | Nothing                                                               |
| RC=2 (Offline)           | => | Start Res1                                       | Nothing                                                               |
| RC=3 (Failed Offline)    | => | Stop Res2                                        | After Res2 is Offline, start both resources on node2 in correct order |
| RC=4 (Stuck Online)      | => | Nothing: wait for Operator action                | Nothing                                                               |
| RC=5 (Pending Online)    | => | Unlikely, wait for Online                        | Nothing                                                               |
| RC=6 (Pending Offline)   | => | Wait for Offline                                 | Nothing                                                               |

### OpState change of resource Res2

The current status of Res1 and Res2 on node1 is Online. The following table shows the actions that IBM Tivoli System Automation performs depending on the return value of the MonitorCommand for Res2.

Table 8. System Automation actions regarding OpState changes of resource Res2

| MonitorCommand (OpState) |    | First action of System Automation                | Second action of System Automation                                    |
|--------------------------|----|--------------------------------------------------|-----------------------------------------------------------------------|
| RC=0 (Unknown)           | => | Nothing, wait for next MonitorCommand with RC<>0 | Nothing                                                               |
| RC=1 (Online)            | => | Nothing                                          | Nothing                                                               |
| RC=2 (Offline)           | => | Force down Res1                                  | After Res1 is Offline, start Res2, after Online of Res2 -> start Res1 |
| RC=3 (Failed Offline)    | => | Force down Res1                                  | Start both resources on node2 in correct order                        |
| RC=4 (Stuck Online)      | => | Nothing: wait for Operator action                | Nothing                                                               |
| RC=5 (Pending Online)    | => | Unlikely, wait for Online                        | Nothing                                                               |
| RC=6 (Pending Offline)   | => | Wait for Offline                                 | Nothing                                                               |

## How System Automation composes the OpState of a resource group

IBM Tivoli System Automation for Multiplatforms is a policy-based automation product. The control point for the automation is the resource group level, meaning that an Operator usually starts or stops an entire resource group rather than starting or stopping single resources. This is done by changing the NominalState attribute of a resource group to Online or Offline. Immediately after changing this attribute IBM Tivoli System Automation will decide which resources needs to be started or stopped to meet the rules of the changed policy.

The OpState (Operational State) attribute of a resource group is an aggregation of the OpState attributes of all resources contained in that resource group **in relation to the NominalState value of the resource group**. So if the NominalState of a resource group has been changed to Online, the OpState of this resource group is showing Pending Online until all of the resources in that resource group are Online. Finally, if all resources of that resource group have reached the value of the NominalState attribute of the resource group, the OpState of the resource group changes to Online, and this value of the OpState attribute of a resource group can now be used to monitor the status of the resources in that group.

The following table shows how IBM Tivoli System Automation composes the value of the OpState attribute of a resource group based on the OpState of the two contained resources Res1 and Res2 from the above example. Note that this picture becomes more complex the more resources are contained in a single resource group.

Table 9. Resource group OpState determination

| OpState of Res2 | OpState of Res1 |    | OpState of Resource Group | Action of System Automation |
|-----------------|-----------------|----|---------------------------|-----------------------------|
| Unknown         | Unknown         | => | Unknown                   | Nothing                     |
| Offline         | Offline         | => | Offline                   | Nothing                     |
| Pending Online  | Offline         | => | Pending Online            | Wait until Res2 is Online   |
| Online          | Offline         | => | Pending Online            | Start Res1                  |
| Online          | Pending Online  | => | Pending Online            | Wait until Res1 is Online   |
| Online          | Online          | => | Online                    | Nothing                     |
| Online          | Pending Offline | => | Pending Offline           | Wait until Res1 is Offline  |
| Online          | Failed Offline  | => | Pending Offline           | Stop Res2                   |
| Pending Offline | Offline         | => | Pending Offline           | Wait until Res2 is Offline  |
| Failed Offline  | Offline         | => | Offline                   | Nothing                     |

## How System Automation reacts to OpState changes of a resource that is started or stopped

IBM Tivoli System Automation usually automates resources based on the NominalState of the resource group and the OpState values of the resources. The goal is to achieve and maintain a state where the OpState of the resource and the NominalState of the comprising resource group are the same. Furthermore, IBM Tivoli System Automation takes action, if the OpState of a resource changes, for example a resource that was running is now monitored Offline.

Another trigger for automation actions is the return code of the StartCommand. If the StartCommand returns an error (a non-zero return code) and the resource is not monitored Online, then IBM Tivoli System Automation also takes action and performs a fail over of the resource to another eligible node. The following sections describe the actions that are performed by IBM Tivoli System Automation if the OpState of a resource changes during or shortly after the execution of the StartCommand or StopCommand of that resource.

### StartCommand

The following tables illustrate how IBM Tivoli System Automation reacts to OpState changes during the execution of the StartCommand. There is one table for each of the three possible situations where the MonitorCommand can report an OpState change:

1. The StartCommand is still under execution (long running StartCommand).
2. The StartCommand has successfully finished (this is the normal situation).
3. The StartCommand has finished with an error or has timed out.

## IBM Tivoli System Automation logic

### **StartCommand is still under execution:**

Table 10. System Automation actions and StartCommand still under execution

| StartCommand                            | MonitorCommand         | Action of System Automation                                                                                         |
|-----------------------------------------|------------------------|---------------------------------------------------------------------------------------------------------------------|
| StartCommand started, but not finished. | RC=0 (Unknown)         | No action, wait for MonitorCommand returning RC<>0                                                                  |
|                                         | RC=1 (Online)          | Start other resources, if any                                                                                       |
|                                         | RC=2 (Offline)         | Wait for Online                                                                                                     |
|                                         | RC=3 (Failed Offline)  | Stop command against the resource and then fail over to another node, probably force other dependent resources down |
|                                         | RC=4 (Stuck Online)    | Unlikely, wait for Operator action                                                                                  |
|                                         | RC=5 (Pending Online)  | Wait for Online                                                                                                     |
|                                         | RC=6 (Pending Offline) | This is unlikely, wait for Online                                                                                   |

Note that once the MonitorCommand has reported the resource as Online, IBM Tivoli System Automation does not take care about the still running StartCommand anymore, as the goal to bring the resource Online is already achieved.

**StartCommand successfully finished:** This table describes the usual behavior of IBM Tivoli System Automation:

Table 11. System Automation actions after StartCommand successfully finished

| StartCommand                                                                             | MonitorCommand         | Action of System Automation                                                                                                                              |
|------------------------------------------------------------------------------------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| RC=0 (successful) and actual retry count < RetryCount (samctrl)                          | RC=0 (Unknown)         | No action, wait for MonitorCommand returning RC<>0                                                                                                       |
|                                                                                          | RC=1 (Online)          | Start other resources, if any                                                                                                                            |
|                                                                                          | RC=2 (Offline)         | After Online timeout: perform start retry, increase retry count                                                                                          |
|                                                                                          | RC=3 (Failed Offline)  | Stop command against the resource and then fail over to another node, probably force other dependent resources down                                      |
|                                                                                          | RC=4 (Stuck Online)    | Unlikely, wait for Operator action                                                                                                                       |
|                                                                                          | RC=5 (Pending Online)  | Wait for Online                                                                                                                                          |
|                                                                                          | RC=6 (Pending Offline) | This is unlikely, wait for Online                                                                                                                        |
| RC=0 (successful) and actual retry count = RetryCount (samctrl) and after Online timeout |                        | Set resource to Failed Offline, send stop command against the resource and then fail over to another node. probably force down other dependent resources |

**StartCommand finished with an error or timed out:** The following table describes the behavior of IBM Tivoli System Automation if the StartCommand for a resource returns with an error or times out, depending on the OpState of the resource:

Table 12. System Automation actions after StartCommand finished with an error or timed out

| MonitorCommand         | StartCommand                                 | Action of System Automation                                                                                                                                                                                                                                                                                       |
|------------------------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RC=0 (Unknown)         | RC=1 (not zero) not successful, or timed out | After MonitorCommand reported Unknown, IBM Tivoli System Automation waits for MonitorCommand returns RC<>0, especially the <b>StartCommand RC (or time out) is ignored</b> .<br><br>If next valid Monitor is Online, resource stays Online, if next valid Monitor is Offline, retry of StartCommand is performed. |
| RC=1 (Online)          | RC=1 (not zero) not successful, or timed out | After MonitorCommand had returned Online, <b>StartCommand RC (or time out) is ignored</b> . No further actions, resource stays Online.                                                                                                                                                                            |
| RC=2 (Offline)         | RC=1 (not zero) not successful, or timed out | Immediately after the StartCommand returns with RC=1, the resource is stopped by IBM Tivoli System Automation, and then a fail over takes place.                                                                                                                                                                  |
| RC=3 (Failed Offline)  | RC=1 (not zero) not successful, or timed out | Immediately after the StartCommand returns with RC=1, a fail over of the resource takes place (no StopCommand execution, as the resource is already failed).                                                                                                                                                      |
| RC=4 (Stuck Online)    | RC=1 (not zero) not successful, or timed out | Unlikely, immediately after the StartCommand returns with RC=1, the resource is stopped by IBM Tivoli System Automation, and then wait for Operator action.                                                                                                                                                       |
| RC=5 (Pending Online)  | RC=1 (not zero) not successful, or timed out | Immediately after the StartCommand returns with RC=1, the resource is stopped by IBM Tivoli System Automation, and then a fail over takes place after the resource is reported Offline.                                                                                                                           |
| RC=6 (Pending Offline) | RC=1 (not zero) not successful, or timed out | Unlikely, immediately after the StartCommand returns with RC=1, the resource is stopped by IBM Tivoli System Automation, and then a fail over takes place after the resource is reported Offline.                                                                                                                 |

Note that the return code of the StartCommand in the table above is ignored if the MonitorCommand already monitored this resource as Online. In this case the result of the two commands is inconsistent: the StartCommand tells IBM Tivoli System Automation, that the start of the resource has failed, but the MonitorCommand already has monitored the resource Online. This is considered as a script error - either within the StartCommand or within the MonitorCommand.

Note also that the return code of the StartCommand has no effect if the resource is monitored Unknown. In this case IBM Tivoli System Automation waits for a valid (non-Unknown) OpState of the resource and the automation will proceed after it receives the next valid (non-Unknown) return code of the MonitorCommand.

## IBM Tivoli System Automation logic

### StopCommand

The following tables illustrate how IBM Tivoli System Automation reacts to OpState changes during the execution of the StopCommand. There is one table for each of the three possible situations where the MonitorCommand can report an OpState change:

1. The StopCommand is still under execution (long running StopCommand).
2. The StopCommand has successfully finished (this is the normal situation).
3. The StopCommand has finished with an error or has timed out.

#### **StopCommand is still under execution:**

Table 13. System Automation actions and StopCommand still under execution

| StopCommand                            | MonitorCommand         | Action of System Automation                   |
|----------------------------------------|------------------------|-----------------------------------------------|
| StopCommand started, but not finished. | RC=0 (Unknown)         | No action, wait for MonitorCommand with RC<>0 |
|                                        | RC=1 (Online)          | Wait for Offline                              |
|                                        | RC=2 (Offline)         | Go on stopping other resources                |
|                                        | RC=3 (Failed Offline)  | Go on stopping other resources                |
|                                        | RC=4 (Stuck Online)    | Wait for Operator action                      |
|                                        | RC=5 (Pending Online)  | This is unlikely, wait for Offline            |
|                                        | RC=6 (Pending Offline) | Wait for Offline                              |

Note that once the MonitorCommand has reported the resource as Offline or Failed Offline, IBM Tivoli System Automation does not take care about the still running StopCommand anymore, as the goal to bring the resource Online is already achieved.

**StopCommand successfully finished:** This table describes the usual behavior of IBM Tivoli System Automation:

Table 14. System Automation actions after StopCommand successfully finished

| StopCommand       | MonitorCommand         | Action of System Automation                   |
|-------------------|------------------------|-----------------------------------------------|
| RC=0 (successful) | RC=0 (Unknown)         | No action, wait for MonitorCommand with RC<>0 |
|                   | RC=1 (Online)          | Wait for Offline                              |
|                   | RC=2 (Offline)         | Go on stopping other resources                |
|                   | RC=3 (Failed Offline)  | Go on stopping other resources                |
|                   | RC=4 (Stuck Online)    | Wait for Operator action                      |
|                   | RC=5 (Pending Online)  | This is unlikely, wait for Offline            |
|                   | RC=6 (Pending Offline) | Wait for Offline                              |

**StopCommand finished with an error or timed out:**

Table 15. System Automation actions after StopCommand finished with an error or timed out

| StopCommand                                   | MonitorCommand         | Action of System Automation                   |
|-----------------------------------------------|------------------------|-----------------------------------------------|
| RC=1 (not zero: not successful, or timed out) | RC=0 (Unknown)         | No action, wait for MonitorCommand with RC<>0 |
|                                               | RC=1 (Online)          | Wait for Offline                              |
|                                               | RC=2 (Offline)         | Go on stopping other resources                |
|                                               | RC=3 (Failed Offline)  | Go on stopping other resources                |
|                                               | RC=4 (Stuck Online)    | Wait for Operator action                      |
|                                               | RC=5 (Pending Online)  | This is unlikely, wait for Offline            |
|                                               | RC=6 (Pending Offline) | Wait for Offline                              |

IBM Tivoli System Automation does not honor the return code of the StopCommand. In any case, the StopCommand is only called once and IBM Tivoli System Automation expects the resource to become Offline. If this does not happen, no further automation action can be performed on this or any dependent resource. The RetryCount has no effect for the StopCommand execution.

**How System Automation reacts if a resource is Online on a certain node and the MonitorCommand reports an OpState for the resource on another node at the same time**

The following table shows the actions IBM Tivoli System Automation will perform, if a resource is Online on a node, and the MonitorCommand returns a certain OpState for this resource on another node.

Table 16. System Automation actions after the MonitorCommand reports an OpState change for the resource on another node

| MonitorCommand on stand-by Node | Action of System Automation                                                                                                                          |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| RC=0 (Unknown)                  | No automation action possible for this resource until MonitorCommand returns with RC<>0.                                                             |
| RC=1 (Online)                   | The resources on both nodes are stopped (and all resources, that depend on that resource). Then the resources are started on one of the nodes again. |
| RC=2 (Offline)                  | No action, this is usual OpState of the resource on a stand-by node.                                                                                 |
| RC=3 (Failed Offline)           | No action, but no fail over is possible to this node anymore.                                                                                        |
| RC=4 (Stuck Online)             | Unlikely, script error (requires Online OpState before ...)                                                                                          |
| RC=5 (Pending Online)           | Same as Online                                                                                                                                       |
| RC=6 (Pending Offline)          | Unlikely, script error (requires Online OpState before ...)                                                                                          |

Most important to notice is the fact, that IBM Tivoli System Automation will stop the resource on both nodes, if it is monitored Online on more than one node at a time, and not just the resource on the stand-by node. This also implies all resources that have a dependency to this resource may also be stopped. For instance all resources that have a DependsOn relationship to this resource will be stopped too. It is therefore recommended not to start and stop applications and resources manually that are under control of IBM Tivoli System Automation.



---

## Chapter 9. Protecting your resources – quorum support

This chapter describes how IBM Tivoli System Automation protects your resources by using configuration and operational quorum.

---

### Overview

A cluster (also known as peer domain) may split into two or more subclusters in case no more communication is possible between the elements in the cluster. Since each subcluster is not aware of one another, it may occur that IBM Tivoli System Automation starts a new instance of an application that is already running in one of the other subclusters. If the application requires access to a shared disk, for example to perform failure recovery, data corruption may occur due to simultaneous access to the disk.

Such resources are characterized as critical. A critical resource is a resource that may not be running on more than one node at any point in time. If such a resource is active on two or more separated nodes, then data integrity of the cluster is endangered. In order to protect such critical resources, IBM Tivoli System Automation ensures that only one of the subclusters survives while the others are dissolved. Thus IBM Tivoli System Automation prevents data corruption that is caused by system failures or network partitions.

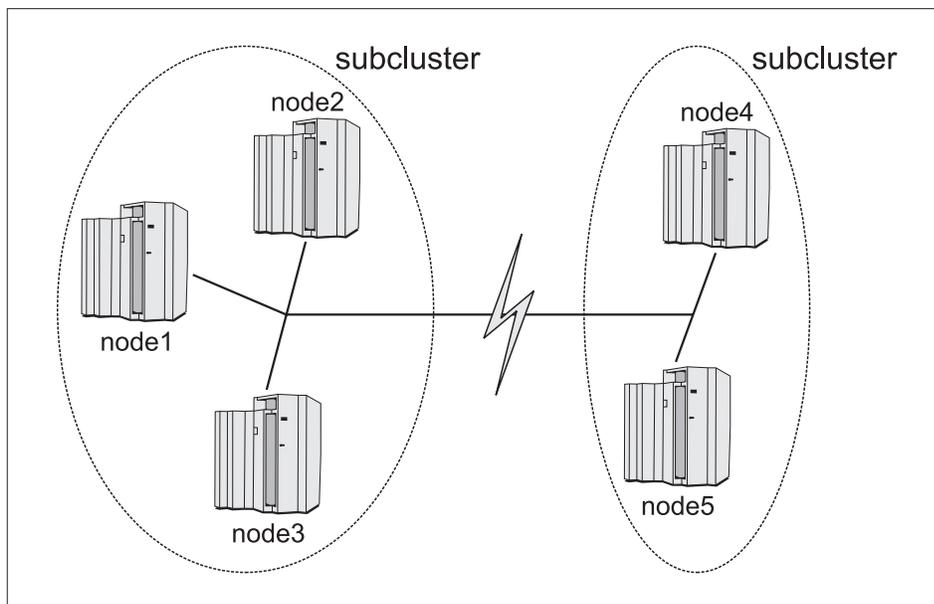


Figure 5. Quorum – majority of nodes

If a cluster falls apart into two or more subclusters, the configuration resource manager (ConfigRM) determines which of the subclusters has the majority of nodes. The majority is given when the subcluster has more than half of all defined nodes in the cluster. The subcluster with the majority of nodes will have an operational quorum. It will survive and become the active cluster, while the other subcluster(s) will be dissolved.

The protection of critical resources is achieved by

- Configuration quorum
- Operational quorum

---

### Configuration quorum

Configuration Quorum determines when configuration changes in the cluster will be accepted. The integrity of the cluster definition is ensured by following the majority rule. Operations affecting the configuration of the cluster are only allowed when  $n/2+1$  nodes are active, where  $n$  is the number of nodes defined in the cluster. However, for some operations, the majority rule can be overridden or different configuration quorum rules apply:

- You can remove nodes using the **rmrpnod** command if exactly half of the nodes is online and if the configuration can be successfully removed from at least one of the offline nodes. You can also use the **-f** option of this command to override the majority rule.
- The quorum rule for the **starttrpdomain** command is  $n/2$ , but you can override it with the all nodes (**-A**) option or the local node (**-L**) option.

**Note:** In a tie situation you can start and stop resource groups using the command **chrg -o online/offline group\_name**.

For more details refer to the *IBM Reliable Scalable Cluster Technology for Linux, Administration Guide, SA22-7892*, and *IBM Reliable Scalable Cluster Technology for Linux, Technical Reference, SA22-7893*.

---

### Operational quorum

The operational quorum is used to decide whether resources can be safely activated without creating conflicts with other resources. Operational quorum is determined based on the number of online nodes and a tie breaker to resolve certain tie situations. A subcluster has an operational quorum if it has more than half of the nodes active.

If an operational quorum exists, IBM Tivoli System Automation can manipulate resources or resource groups and bring them online. If no quorum exists, IBM Tivoli System Automation cannot take any action on a resource.

If critical resources are active on a subcluster that has lost quorum, the ConfigRM uses the "CritRsrcProtMethod" attribute of each node in the subcluster to determine on which way the system should be terminated. The protection methods are based on immediate system shutdown by means of kernel panic simulation. There are 5 protection methods:

| Meaning                                           | Value |
|---------------------------------------------------|-------|
| Hard reset and reboot operating system (default). | 1     |
| Halt operating system.                            | 2     |
| Hard reset and reboot operating system with Sync. | 3     |
| Halt with Sync.                                   | 4     |
| No protection. System continues operating.        | 5     |

When protection methods without sync are applied, the data consistency is ensured, but the file system can be corrupted. Therefore we recommend to use a journaling file system, which is a safe method. When protection methods with sync are applied, the in-memory data will be saved, but the data consistency is not guaranteed due to a possible file system flushing. This is an unsafe method. But sync, however, will enhance the probability of consistency data.

The protection methods on the nodes may be different. However, the normal case is that the same protection method is set for each node in the entire cluster.

In the case of a tie in which the cluster has been partitioned into subclusters with an equal number of nodes, the configuration resource manager uses a tie breaker to determine which subcluster has an operational quorum. A subcluster will have an operational quorum if it has the tie breaker reserved.

The following tie breaker types exist:

1. Operator - this tie breaker asks the operator or administrator for a decision. Until the administrator explicitly breaks the tie, neither subcluster will have an operational quorum. The operational quorum state is set to "PendingQuorum" and stays in this state until either the network is repaired, failing nodes are repaired and brought online, or the operator determines which subcluster is the winning one and which is the losing one. This is done by invoking the "ResolveOpQuorumTie" action on a node of each active subcluster.
2. Fail - this actually is a pseudo tie breaker which means that it does not resolve the tie. Neither subcluster will have operational quorum.
3. SCSI - this tie breaker is specific for Linux on iSeries, Linux on pSeries, and Linux on xSeries. It assumes that an SCSI disk is shared by all nodes of the cluster. Tie breaker reservation is done by the SCSI reserve or persistent reserve command.
4. ECKD - this tie breaker is specific for Linux on zSeries. It assumes that an ECKD disk is shared by all nodes of the cluster. Exclusive access to the disk is done by means of the ECKD reserve command.
5. DISK - This tie breaker type is specific for AIX. This tie breaker type enables you to specify an SCSI or SCSI-like physical disk using an AIX device name, and assumes that the SCSI disk is shared by one or more nodes of the cluster. Tie breaker reservation is done by the SCSI reserve or persistent reserve command. If creating a tie breaker of this type, you need to set the DeviceInfo persistent resource attribute to identify the physical disk. Only SCSI and SCSI-like physical disks are supported. Physical disks attached via Fiber Channel, iSCSI and Serial Storage Architecture Connections are suitable.

If you have an odd number of nodes in the cluster, the subcluster that has more than half of the nodes available has Quorum. For example, on a three node cluster, the subcluster with two nodes available has Quorum. The other subcluster having only one node available gets neither operational nor configuration quorum and therefore no resources will be started on that node. If a critical resource is already running on that node, the protection method defined in the "CritRsrcProtMethod" attribute will be applied to the node (see on page 96).

If you have an even number of nodes in the cluster and one of the subclusters consists of half of the nodes in case of a cluster split, then a tie breaker decides which subcluster is allowed to run critical resources. Nodes with critical resources that lose the competition for the tiebreaker are subject to resource protection, which means that they will be stopped or rebooted immediately. To perform an automatic tie breaking without operator action, you need a disk tie breaker (SCSI for Linux on xSeries, pSeries and iSeries, or ECKD for Linux on zSeries, or DISK for AIX). A disk tie breaker is a shared disk which is accessible from all cluster nodes.

## Setting critical resources

Use the ProtectionMode persistent attribute to specify whether the resource is critical. If it is critical, then the Configuration RM (IBM.ConfigRM) decides whether the resource can be started as requested. The attribute may have the integer values 0 (non-critical) or 1 (critical). Per default IBM.Application resources are non-critical, and IBM.ServiceIP resources are critical. If the resource is set to critical, monitoring will immediately start.

Issue the RSCT command lsrsrc to list the value of the ProtectionMode attribute:

```
lsrsrc IBM.Application Name NodeNameList ProtectionMode
```

Issue the RSCT command chrsrc to define the resource as critical by setting the ProtectionMode to 1:

```
chrsrc -s "Name='apache1'" IBM.Application ProtectionMode=1
```

To define a resource as non-critical set ProtectionMode to 0:

```
chrsrc -s "Name='apache1'" IBM.Application ProtectionMode=0
```

Issue the following to verify whether critical resources are currently active on a node for resource class IBM.Application:

## Quorum support

```
lsrsrc IBM.Application Name NodeNameList OpState ProtectionMode
```

This provides the following output:

```
resource 1:
 Name = "apache1"
 NodeNameList = {"node1","node2"}
 OpState = 1
 ProtectionMode = 1
resource 2:
 Name = "apache1"
 NodeNameList = {"node1"}
 OpState = 2
 ProtectionMode = 1
resource 3:
 Name = "apache1"
 NodeNameList = {"node2"}
 OpState = 1
 ProtectionMode = 1
```

Critical resource apache1 is active on the node2.

Issue the following to verify if critical resources are currently active on the nodes:

```
lsrsrc IBM.PeerNode Name CritRsrcActive
```

The output is as follows:

Resource Persistent and Dynamic Attributes for IBM.PeerNode

```
resource 1:
 Name = "node1"
 CritRsrcActive = 0
resource 2:
 Name = "node2"
 CritRsrcActive = 1
```

Critical resources are active on node2.

## Getting quorum information

Use the **lssrc** command for the IBM.RecoveryRM daemon to obtain the current quorum states.

```
node02:~/build # lssrc -ls IBM.RecoveryRM
```

You get the following output:

```
Daemon State:
My Node Name : node02
Master Node Name : node01 (node number = 1)
Our CVN : 61035379498
Total Node Count : 2
Joined Member Count : 2
Config Quorum Count : 2
Startup Quorum Count : 1
Operational Quorum State : HAS_QUORUM
In Config Quorum : TRUE
In Config State : TRUE
```

The meaning of the various attributes is as follows:

### Total Node count

Is the number of nodes defined in the cluster.

### Joined Member Count

is the number of IBM.RecoveryRM daemons running in the cluster. This is equivalent to the number of active nodes in the (sub)cluster.

### Config Quorum Count

is the number of IBM.RecoveryRM daemons that must be active in order to make a configuration change by means of the IBM Tivoli System Automation commands.

### Startup Quorum Count

Is the number of IBM.RecoveryRM daemons that must be active before the IBM Tivoli System Automation automation engine is activated.

### Operational Quorum State

Indicates (sub)cluster wide whether this subcluster can survive or must immediately dissolve in case critical resources are running on the node(s) in the subcluster. The operational quorum state is provided by the dynamic attribute OpQuorumState of PeerDomain class. OpQuorumState can have the following values:

- 0 – HAS\_QUORUM** IBM Tivoli System Automation may start resources
- 1 – PENDING\_QUORUM** Indicates that a tie situation occurred that is not yet resolved. IBM Tivoli System Automation does not start resources.
- 2 – NO\_QUORUM** IBM Tivoli System Automation is not allowed to start resources.

### In Config Quorum

Indicates whether enough nodes hosting IBM.RecoveryRM daemons are active to accept configuration changes by IBM Tivoli System Automation commands. Shows TRUE if the total number of "joined" IBM.RecoveryRM daemon group members within the cluster is equal or above the Config Quorum count.

### In Config State

Indicates whether the master IBM.RecoveryRM daemon has completed the verification of the system registry content at startup time. If the state equals to FALSE, any IBM Tivoli System Automation command will be rejected.

Enter the following to list OpQuorumState:

```
lsrsrc IBM.PeerDomain Name OpQuorumState
```

You get the following output:

```
Resource Persistent and Dynamic Attributes for:IBM.PeerDomain
resource 1:
 Name = "myCluster"
 OpQuorumState = 0
```

## Setting up and administering a tie breaker

The IBM.TieBreaker resource class allows you to configure a tie breaker such as ECKD or SCSI. Additionally two tie breakers are predefined, Operator and Fail. The operator tie breaker provides an undetermined result when a tie occurs and it is left to the administrator to resolve the tie through granting or denying the operational quorum. When a tie occurs and a tie breaker of type "Fail" is active, the attempt to reserve the tie breaker is always denied. Default tie breaker type is set to 'Operator'.

To list the available tie breaker type:

```
lsrsrc -c IBM.TieBreaker
```

You get the following output on a Linux system running on xSeries, pSeries or iSeries :

```
Resource Class Persistent Attributes for: IBM.TieBreaker
resource 1:
 AvailableTypes = [{"SCSI", ""}, {"Operator", ""}, {"Fail", ""}]
```

To list the tie breaker name:

## Quorum support

```
lsrsrc IBM.TieBreaker
```

You get the following output:

```
Resource Persistent Attributes for: IBM.TieBreaker
```

```
resource 1:
 Name = "FAIL"
 Type = "FAIL"
 DeviceInfo = ""
 ReprobeData = ""
 ReleaseRetryPeriod = 0
 HeartbeatPeriod = 0
 PreReserveWaitTime = 0
 PostReserveWaitTime = 0
 NodeInfo = {}
resource 2:
 Name = "Operator"
 Type = "Operator"
 DeviceInfo = ""
 ReprobeData = ""
 ReleaseRetryPeriod = 0
 HeartbeatPeriod = 0
 PreReserveWaitTime = 0
 PostReserveWaitTime = 0
 NodeInfo = {}
resource 3:
 Name = "myTieBreaker"
 Type = "SCSI"
 DeviceInfo = "ID=0 LUN=0 CHAN=0 HOST=2"
 ReprobeData = ""
 ReleaseRetryPeriod = 0
 HeartbeatPeriod = 5
 PreReserveWaitTime = 0
 PostReserveWaitTime = 0
 NodeInfo = {}
```

Although you can define several tie breaker resources in the resource class IBM.TieBreaker, only one of them can be active in the cluster at the same time. Issue the following command to list the tie breaker that is currently active in the cluster :

```
lsrsrc -c IBM.PeerNode OpQuorumTieBreaker
```

You get the following output:

```
Resource Class Persistent Attributes for: IBM.PeerNode
```

```
resource 1:
 OpQuorumTieBreaker = "Operator"
```

The active tie breaker is set with the following command:

```
chrsrc -c IBM.PeerNode OpQuorumTieBreaker="Operator"
```

To grant/deny the operational quorum when tie breaker is "Operator":

```
runact -c IBM.PeerDomain ResolveOpQuorumTie Ownership=1 (0 to deny)
```

**Note:** In order to avoid race conditions, the operator tie breaker must be denied first for the losing subcluster(s) before granting it to the subcluster which is supposed to continue.

## Using a tie breaker

To create a basic setup for the tie breaker you need a cluster of two (or other even number) of nodes. Also you need a disk that is shared between all nodes of the cluster. The tie breaker disk is shared between all cluster nodes.

**Attention:** When defining tie breaker resources, be aware that the disk on which IBM.TieBreaker resources are stored should not also be used to store file systems.

The following three examples show how to use a tie breaker with an ECKD, SCSI or DISK device. Note that the tie breaker needs not be formatted or partitioned. Then it will only be marked active without size information (in case of ECKD).

### Example 1: ECKD tie breaker setup for a two nodes cluster

Note the following when defining the tie breaker disk under VM:

- Full pack minidisk should be defined.
- If minidisk cache is used, its value should be set to off.
- ECKD disk is shared between both nodes.

The ECKD tie breaker type is specific for Linux on zSeries. If you want to create an ECKD tie breaker object, you need to set the DeviceInfo persistent resource attribute to indicate the ECKD device number. This type of tie breaker uses a reserve/release mechanism and needs to be re-reserved periodically to hold the reservation. For this reason, we strongly recommend that you also specify the HeartbeatPeriod persistent resource attribute when creating a tie breaker of this type. The HeartbeatPeriod persistent resource attribute defines the interval at which the reservation request is re-issued.

Attach the tie breaker to the system

```
node01:~ # cat /proc/subchannels
Device sch. Dev Type/Model CU in use PIM PAM POM CHPIDs

50DE 0A6F 3390/0A 3990/E9 F0 A0 FF 7475E6E7 FFFFFFFF
```

and add it to the /proc/dasd/devices list:

```
node01:~ # cat /proc/dasd/devices
50dc(ECKD) at (94: 0) is : active at blocksize: 4096, 601020 blocks, 2347 MB
50dd(ECKD) at (94: 4) is : active at blocksize: 4096, 601020 blocks, 2347 MB
50de(ECKD) at (94: 8) is : active at blocksize: 4096, 601020 blocks, 2347 MB
50df(ECKD) at (94: 12) is : active at blocksize: 4096, 601020 blocks, 2347 MB
```

Perform the following steps to use the tie breaker:

1. Create a tie breaker resource object in IBM.TieBreaker class. **DeviceInfo** shows the ECKD device number. It can be obtained from /proc/dasd/devices file.

```
node01:~ # mkrsrc IBM.TieBreaker Name=myTieBreaker Type=ECKD DeviceInfo="ID=50de" HeartbeatPeriod=5
node01:~ # lsrsrc IBM.TieBreaker
Resource Persistent Attributes for: IBM.TieBreaker
resource 1:
 Name = "Operator"
 Type = "Operator"
 DeviceInfo = ""
 ReprobeData = ""
 ReleaseRetryPeriod = 0
 HeartbeatPeriod = 0
 PreReserveWaitTime = 0
 PostReserveWaitTime = 0
 NodeInfo = {}
resource 2:
 Name = "Fail"
 Type = "Fail"
 DeviceInfo = ""
 ReprobeData = ""
 ReleaseRetryPeriod = 0
 HeartbeatPeriod = 0
 PreReserveWaitTime = 0
 PostReserveWaitTime = 0
 NodeInfo = {}
resource 3:
```

## Quorum support

```
Name = "myTieBreaker"
Type = "ECKD"
DeviceInfo = "ID=50de"
ReprobeData = ""
ReleaseRetryPeriod = 0
HeartbeatPeriod = 5
PreReserveWaitTime = 0
PostReserveWaitTime = 0
NodeInfo = {}
```

2. Change OpQuorumTieBreaker attribute in IBM.PeerNode class to one of the tie breaker resource objects.

```
node01:~ # chrsrc -c IBM.PeerNode OpQuorumTieBreaker="myTieBreaker"
```

```
node01:~ # lsrsrc -c IBM.PeerNode
```

```
Resource Class Persistent Attributes for: IBM.PeerNode
```

```
resource 1:
```

```
CommittedRSCTVersion = ""
ActiveVersionChanging = 0
OpQuorumOverride = 0
CritRsrcProtMethod = 1
OpQuorumTieBreaker = "myTieBreaker"
```

**Hint:** If the node reserving a tie breaker is down and cannot be rebooted, manual access on another node is needed to break the reservation and take it over on this other node. The tie breaker disk can

- be either still attached to the healthy node, provided this node has not been rebooted in the mean time:

```
node01:~ # cat /proc/subchannels
Device sch. Dev Type/Model CU in use PIM PAM POM CHPIDs

50DE 0A6F 3390/0A 3990/E9 F0 A0 FF 7475E6E7 FFFFFFFF
```

```
node01:~ # cat /proc/dasd/devices
50de(ECKD) at (94: 8) is dasdc : active at blocksize: 4096, 601020 blocks, 2347 MB
```

- be boxed, if this node has been rebooted and cannot recognize the tie breaker disk anymore:

```
node01:~ # cat /proc/subchannels
Device sch. Dev Type/Model CU in use PIM PAM POM CHPIDs

50DE 0A6F FFFF/00 F0 A0 FF 7475E6E7 FFFFFFFF
```

```
node01:~ # cat /proc/dasd/devices
50de(ECKD) at (94: 8) is dasdc : boxed
```

To break the tie breaker disk reservation enter the command `/usr/sbin/rsct/bin/tb_break`:

```
tb_break -t ECKD /dev/dasdc
```

The tie breaker disk should now be reserved by the healthy node.

**Note:** If the `tb_brk` command does not work the first time, issue it again.

### Example 2: SCSI tie breaker setup for a two nodes cluster

The SCSI tie breaker type is specific for Linux on xSeries, pSeries, and iSeries. If you want to create an SCSI tie breaker object, you need to specify the SCSI device using the DeviceInfo persistent resource attribute. If the SCSI configuration is different on different nodes in the cluster, you can also use the NodeInfo persistent resource attribute to reflect those differences. This type of tie breaker uses a reserve/release mechanism and needs to be re-reserved periodically to hold the reservation. For this reason, we strongly recommend that you also specify the HeartbeatPeriod persistent resource attribute when creating a tie breaker of this type. The HeartbeatPeriod persistent resource attribute defines the interval at which the reservation request is re-issued.

SCSI devices can be identified by four integer values for the attributes HOST, CHAN, ID, and LUN:

```
node1:~# dmesg | grep "Attached scsi disk"
```

Normally these parameters are identical on each cluster node.

For example, for node1 and node2 these are HOST=0 CHAN=0 ID=4 LUN=0.

You can then create the tie breaker object:

```
mkrsrc IBM.TieBreaker Name=myTieBreaker Type=SCSI DeviceInfo=" HOST=0 CHAN=0 ID=4 LUN=0"
```

The four values above may also be different for different nodes (even if the target device is same). In that case the **NodeInfo** field should be used.

Use the four integer values from the command output:

```
dmesg | grep "Attached scsi disk"
Attached scsi disk sdf at scsi2, channel 2, id 4, lun 0
```

For disk sdf is HOST=2, CHAN=2, ID=4, LUN=0.

For example, a SCSI device is connected to 2 nodes named node1 and node2 and has the following SCSI identifiers:

```
node1: HOST=0 CHAN=0 ID=4 LUN=0
node2: HOST=2 CHAN=2 ID=4 LUN=0
```

You can then create the tie breaker object as

```
mkrsrc IBM.TieBreaker Name=scsi Type=SCSI DeviceInfo="ID=4 LUN=0"
NodeInfo='["node1", "HOST=0 CHAN=0"], ["node2", "HOST=2 CHAN=2"]'
```

IBM Tivoli System Automation handles **DeviceInfo** and **NodeInfo** in such a way that it merges the two strings, **DeviceInfo** first and then **NodeInfo**. For example, for node1 the merged string is

```
"ID=4 LUN=0 HOST=0 CHAN=0"
```

which will then be parsed.

Also, any duplicated keywords will be allowed and the last one will be used. Therefore, the same command can be specified as

```
mkrsrc IBM.TieBreaker Name=myTieBreaker Type=SCSI DeviceInfo="ID=4 LUN=0
HOST=0,CHAN=0" NodeInfo='["node2", "HOST=2 CHAN=2"]'
```

This simplification may be useful as often and most likely the SCSI id is the same for many nodes.

**Hint:** If the node reserving a tie breaker is down and cannot be rebooted, manual access on another node is needed to release the SCSI tie breaker disk. To release a disk, run the command:

```
tb_break [-f] HOST CHAN ID LUN
```

for example,

```
/usr/sbin/rsct/bin/tb_break -f HOST=0 CHAN=0 ID=4 LUN=0
```

### Example 3: AIX DISK tie breaker setup for a two nodes cluster

The DISK tie breaker type is specific to AIX. If you want to create a DISK tie breaker object, you need to set the DeviceInfo persistent resource attribute to indicate the AIX device name. The AIX device name must specify a SCSI or SCSI-like physical disk that is shared by all nodes of the peer domain. Physical disks attached via Fiber Channel, iSCSI, and Serial Storage Architecture may serve as a DISK tie breaker. However, IDE hard disks do not support the SCSI protocol and cannot serve as a DISK tie breaker. Logical volumes also cannot serve as a DISK tie breaker. This type of tie breaker uses a reserve/release mechanism and needs to be re-reserved periodically to hold the reservation. For this reason, we strongly recommend that you also specify the HeartbeatPeriod persistent resource attribute when creating a tie breaker of this type. The HeartbeatPeriod persistent resource attribute defines the interval at which the reservation request is re-issued.

To print every known physical volume in the system along with its physical disk name, enter the **lspv** command:

## Quorum support

`lspv`

An output similar to the following one is displayed:

```
hdisk0 000000371e5766b8 rootvg active
hdisk1 000069683404ed54 None
```

In order to verify that a disk is a SCSI or SCSI-like disk and so a suitable candidate for a DISK tie breaker, use the **lsdev** command. For example:

```
lsdev -C -l hdisk1
```

An output similar to the following one is displayed:

```
hdisk1 Available 10-60-00-0,0 16 Bit SCSI Disk Drive
```

In order to serve as a tie breaker disk, the disk must be shared by all nodes of the peer domain. Check the physical volume ID returned by the `lspv` command to determine if the disk is shared between nodes (in the preceding output for the `lspv` command, the physical volume ID is listed in the second column; the volume ID for `hdisk1` is `000069683404ed54`.) Be aware, however, that AIX remembers all disks that have been attached to the system, and the disks listed by the `lspv` command may no longer be attached. If such a disk was moved to another machine, it might appear as if the disk is shared, when in fact it is no longer attached to the original machine.

The disk on which IBM.TieBreaker resources are stored should not also be used to store file systems. If the nodes of the cluster share more than one disk, it may be difficult to determine which one is the tie breaker disk, and which one is used for application data. The output from the `lsdev` command shows the SCSI address associated with the disk. (In the preceding output for the `lsdev` command, the SCSI address is listed in the third column; the SCSI address for `hdisk0` is `10-60-00-0,0`.) This information will help you to identify the correct disk if you are aware of the address of the disk prior to its installation.

Once you know the device name, you can issue the **mkrsrc** command:

```
mkrsrc IBM.TieBreaker Name=myTieBreaker Type=DISK DeviceInfo="DEVICE=/dev/hdisk1" HeartbeatPeriod=5
```

**Hint:** If the node reserving a tie breaker is down and cannot be rebooted, manual access on another node is needed to release the SCSI tie breaker disk. To release the disk, run the command:

```
/usr/sbin/rsct/bin/tb_break -f -t DISK "DEVICE=/dev/hdisk1"
```

## Overriding the operational quorum

In order to remove nodes from the cluster, at least one node of the cluster must be online to initiate the **rmrpnod** command. If there are not enough nodes to ever achieve an operational quorum, there is no chance to adjust the cluster size by administrative means so that the quorum can be reestablished.

If for any reasons the operational quorum function must be deactivated, the persistent attribute `OpQuorumOverride` must be set to 1:

```
chrsrc -c IBM.PeerNode OpQuorumOverride=1
```

In this case operational quorum State is always `HAS_QUORUM` and resource protection is not ensured anymore

---

## Chapter 10. Setting up a high available network

When talking about a setting up a high available network, we should distinguish between two situations:

- A more reliable cluster communication in which the cluster infrastructure (RSCT) takes care that all available communication paths are used to ensure cluster integrity and configuration data replication.
- A representation of a high available IT service in which automation takes care of an IP address (further called ServiceIP) which represents an IT service to clients outside the cluster.

The same communication interface is often used for both tasks, but this is not necessary and often not the best way to do. The following section describes this.

---

### Running a one or two node cluster: detecting network interface failures

In case you run a one or two node cluster you need some additional configuration to detect network interface failures. The cluster software periodically tries to reach each network interface of the cluster. If there is a two node cluster and one interface fails on one node, the other interface on the other node is not able to get response from the peer and will also be flagged offline.

To avoid this behavior the cluster software must be told to contact a network instance outside the cluster. Best practise is to use the default gateway of the subnet the interface is in.

On each node create following file:

```
/usr/sbin/cluster/netmon.cf
```

Each line of this file should contain the machine name or IP address of the external instance. An IP address should be specified in dotted decimal format.

This is an example of a `/usr/sbin/cluster/netmon.cf` file:

```
this is default gateway for all interfaces in 192.168.1.0 network
192.168.1.1

this is default gateway for all interfaces in 192.168.2.0 network
gw.de.ibm.com
```

---

### Two nodes cluster, each node has one ethernet interface

The following network setup is given:

|              | <b>Name:</b> | <b>Device:</b> | <b>IP:</b>       |
|--------------|--------------|----------------|------------------|
| Cluster node | Inxcm1       | eth0           | 9.152.172.1/24   |
| Cluster node | Inxcm2       | eth0           | 9.152.172.2/24   |
| Router       | gw           | eth0           | 9.152.172.254/24 |
| ServiceIP    | —            | —              | 9.152.172.3/24   |

## network setup

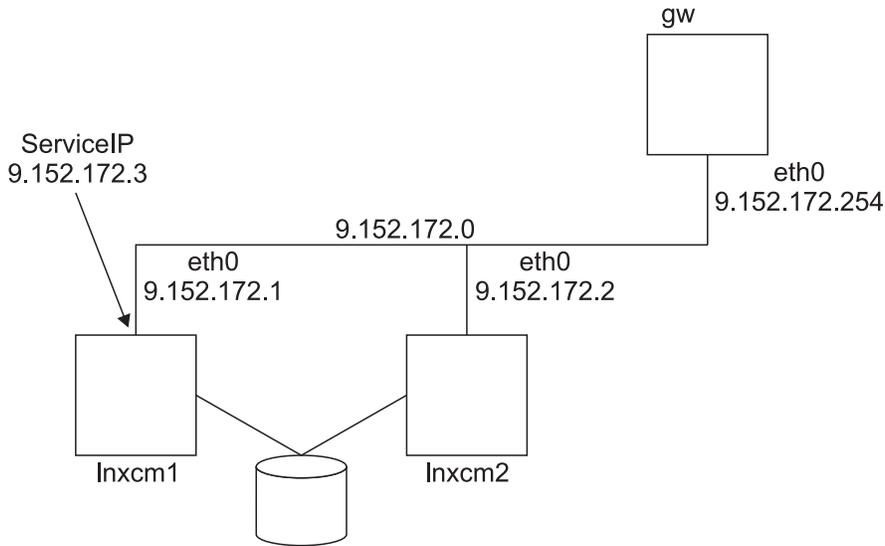


Figure 6. Two nodes, one interface

In this setup the cluster communication and the presentation of the high available IT service uses the same communication path, the 9.152.172.0 network.

Automation can assign the ServiceIP either on the Inxcm1 interface eth0 or on the Inxcm2 interface eth0. If one interface fails, automation moves the ServiceIP to the other node. Thus it satisfies the policy which requires assigning the ServiceIP on a running network interface.

In this setup the failure of one network interface will lead to a break in the cluster communication with all the problems as described in Chapter 9, “Protecting your resources – quorum support,” on page 95. If the communication breaks as shown in Figure 7, the tie breaker decides which node is able to go on with automation. If this is node Inxcm1, automation will find on Inxcm1 no online network interface to assign the ServiceIP on.

In this example the network 9.152.172.0 served two purposes:

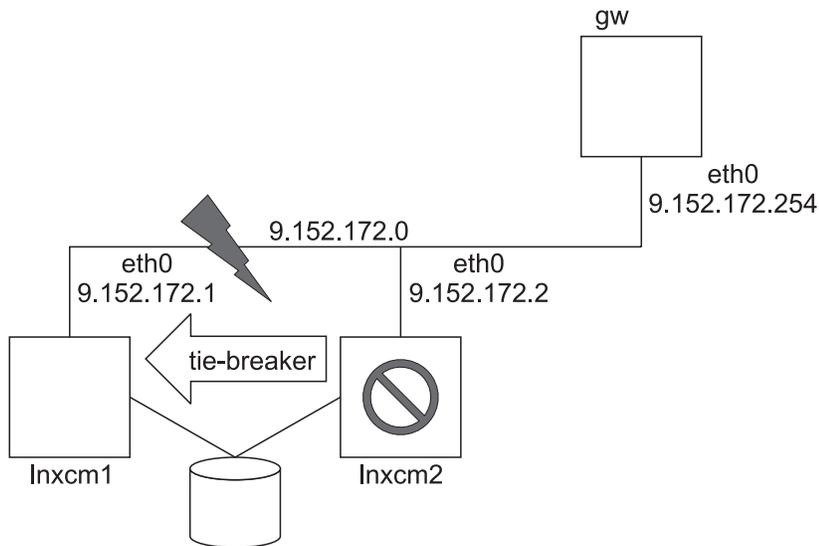


Figure 7. Two nodes, one interface – interface failure

1. Representing the network for the high available IT service.
2. Used for internal cluster communication.

Sample IBM Tivoli System Automation policy:

```
lnxcm1# mkequ NetInt IBM.NetworkInterface:eth0:lnxcm1,eth0:lnxcm2
lnxcm1# mkrsrc IBM.ServiceIP Name="SIP"
IPAddress="9.152.172.3"
NetMask="255.255.255.0"
NodeNameList="{ 'lnxcm1', 'lnxcm2' }"
lnxcm1# mkrgr rg
lnxcm1# addrgmbr -g rg IBM.ServiceIP:SIP
lnxcm1# mkrel -p dependson -S IBM.ServiceIP:SIP -G IBM.Equivalency:NetInt
```

| Advantage                       | Disadvantage                                       |
|---------------------------------|----------------------------------------------------|
| Very easy setup.                | Each communication problem leads to cluster split. |
| Less network hardware required. | ServiceIP moves only between nodes.                |

---

## Two nodes cluster, each node has two network interfaces

Before starting with this setup keep in mind that it is not possible to have more than one static configured network interface in the same IP subnet. Each IP address will cause an entry in the kernel routing table. In case of two interfaces in the same subnet there will be 2 routes for the same subnet. If the interface, which created the first entry, fails the communication for this subnet will break down even if there is another interface which still is able to communicate.

## Two physically separated networks, move ServiceIP between nodes

The following network setup applies:

|              | Name:  | Device:      | IP:                              |
|--------------|--------|--------------|----------------------------------|
| Cluster node | lnxcm1 | eth0<br>eth1 | 9.152.172.1/24<br>192.168.1.1/24 |
| Cluster node | lnxcm2 | eth0<br>eth1 | 9.152.172.2/24<br>192.168.1.2/24 |
| Router       | gw     | eth0         | 9.152.172.254/24                 |
| ServiceIP    | -      | -            | 9.152.172.3/24                   |

## network setup

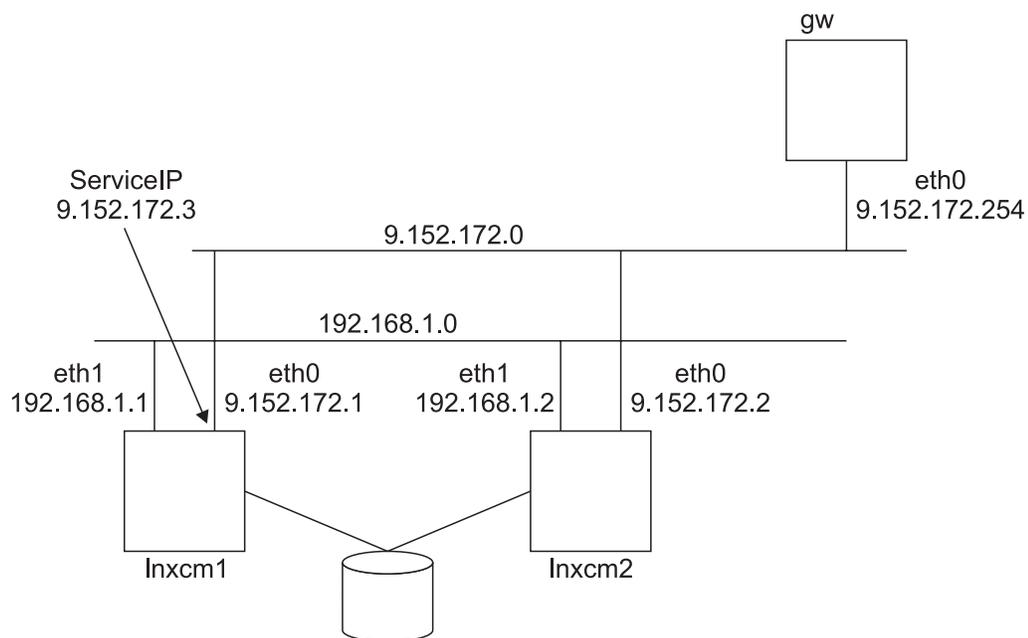


Figure 8. Two nodes, two interfaces, two physically separated networks

There are now two networks 192.168.1.0 and 9.152.172.0 for the cluster communication. If there is a failure in one network interface the cluster will not break.

- Network 9.152.172.0 represents the network for the high available IT service.
- Network 192.168.1.0 makes cluster internal communication more reliable.

Since only the network of the ServiceIP is connected to the gateway, a failure of interface eth0 on Inxcm1 will cause the automation to move the ServiceIP to the interface eth0 on the other node Inxcm2. Because of the physical separation of the two networks it is not possible to move the ServiceIP from eth0 to eth1 within the same node.

The sample IBM Tivoli System Automation policy is the same as shown on page on page 107.

| Advantage                            | Disadvantage                        |
|--------------------------------------|-------------------------------------|
| Easy setup.                          | ServiceIP moves only between nodes. |
| Redundancy in cluster communication. |                                     |

## Three logical networks in one physical network, move ServiceIP between network interfaces

Another network setup is required to not only move the ServiceIP between nodes in the cluster but also between interfaces within one node. Needed is a separate logical network for each interface of a node, and an additional network for the ServiceIP. Choosing an existing network (one of eth0 or eth1) would cause routing problems. Make sure to connect all interfaces to the same physical network. This allows each interface to hold addresses of all the logical networks.

The following network setup applies:

|              | Name:  | Device:      | IP:                              |
|--------------|--------|--------------|----------------------------------|
| Cluster node | Inxcm1 | eth0<br>eth1 | 192.168.1.1/24<br>192.168.2.1/24 |

|              | Name:  | Device:      | IP:                              |
|--------------|--------|--------------|----------------------------------|
| Cluster node | Inxcm2 | eth0<br>eth1 | 192.168.1.2/24<br>192.168.2.2/24 |
| Router       | gw     | eth0         | 9.152.172.254/24                 |
| ServiceIP    | -      | -            | 9.152.172.3/24                   |

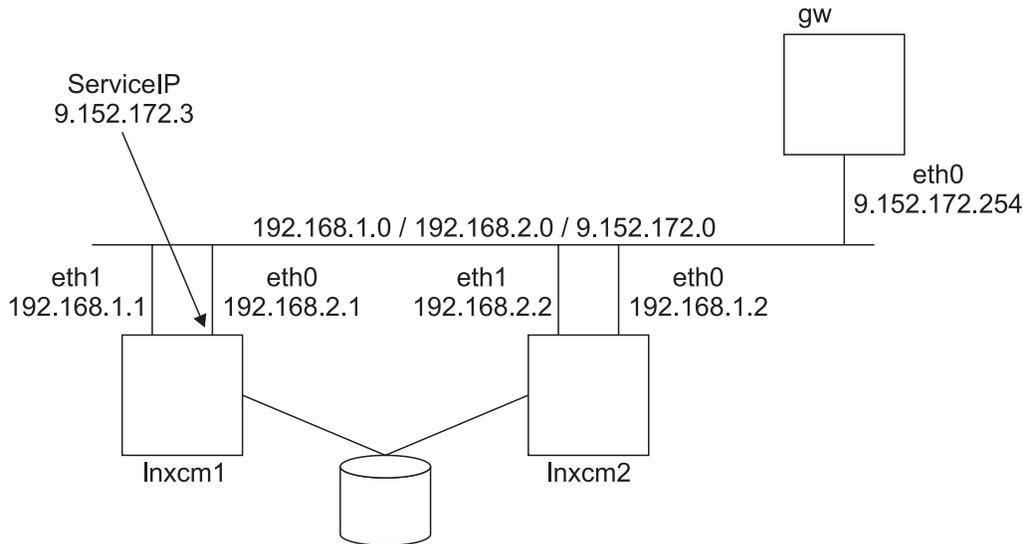


Figure 9. Two nodes, two interfaces, one physical network

- Network 9.152.172.0 represents the network for the high available IT service.
- Network 192.168.1.0 represents the first cluster internal communication network.
- Network 192.168.2.0 represents the second cluster internal communication network .

Sample IBM Tivoli System Automation policy:

```
Inxcm1# mkequ NetInt
IBM.NetworkInterface:eth0:Inxcm1,eth1:Inxcm1,eth0:Inxcm2,eth1:Inxcm2
Inxcm1# mkrsrc IBM.ServiceIP Name="SIP" IPAddress="9.152.172.3"
NetMask="255.255.255.0" NodeNameList="{ 'Inxcm1', 'Inxcm2' }"
Inxcm1# mkrng rg
Inxcm1# addrgmbr -g rg IBM.ServiceIP:SIP
Inxcm1# mkrel -p dependson -S IBM.ServiceIP:SIP -G IBM.Equivalency:NetInt
```

| Advantage                                       | Disadvantage                                |
|-------------------------------------------------|---------------------------------------------|
| Easy setup.                                     | 3 logical networks in 1 physical network.   |
| Redundancy in cluster communication.            | Traffic of 3 networks on 1 physical medium. |
| ServiceIP can move between interfaces and nodes |                                             |

## Two physically separated networks, dynamic routing and VIPA

The detailed description of this setup extends the scope of this manual. Basically the ServiceIP is assigned to a virtual network within the kernel of a cluster node. Dynamic routing on all cluster nodes and the gateway makes sure that a route to the ServiceIP is established.

## network setup

The following network setup applies:

|              | Name:  | Device:      | IP:                                  |
|--------------|--------|--------------|--------------------------------------|
| Cluster node | Inxcm1 | eth0<br>eth1 | 9.152.170.1/24<br>9.152.171.1/24     |
| Cluster node | Inxcm2 | eth0<br>eth1 | 9.152.170.2/24<br>9.152.171.2/24     |
| Router       | gw     | eth0<br>eth1 | 9.152.170.254/24<br>9.152.171.254/24 |
| ServiceIP    | -      | -            | 9.152.172.3/24                       |

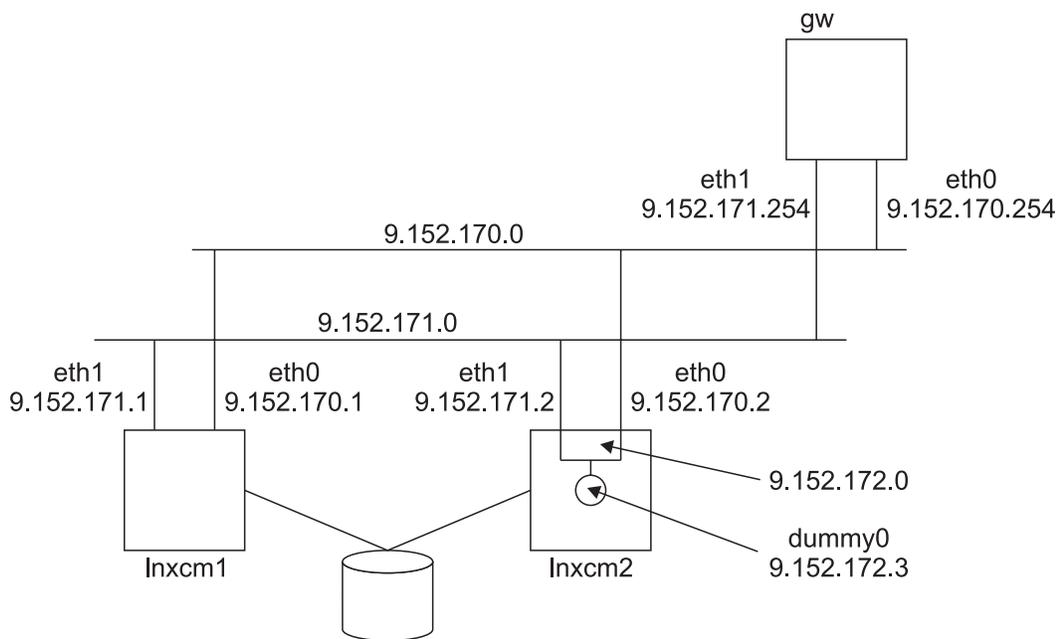


Figure 10. Two physically separated networks, dynamic routing and VIPA

| Advantage                                                           | Disadvantage                                                                               |
|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| There is no dependency to the physical network device.              | Complicate setup.                                                                          |
| Concept of finding dynamically the best way to a host (IP address). | Dynamic routing required.                                                                  |
| No need to move ServiceIP between interfaces                        | Setup is not restricted to the cluster nodes; gateway also has to support dynamic routing. |

---

## Chapter 11. Controlling and administering IBM Tivoli System Automation

This chapter describes various parameters which can be used to control and change the general behavior of IBM Tivoli System Automation. It also gives some insight into the infrastructure of IBM Tivoli System Automation and offers some helpful hints and tips.

---

### Controlling IBM Tivoli System Automation

There are several attributes you can use to change the general behavior of IBM Tivoli System Automation. You can start/stop the automation functionality, define some timeouts or exclude some nodes from automation, e.g. for maintenance reasons

These attributes are:

- **TimeOut**  
Specifies the timeout value in seconds for a start control operation executed by IBM Tivoli System Automation. After the timeout expires the operation is repeated if the `RetryCount` is not exceeded.
- **RetryCount**  
Number of allowed attempts if a control operation fails or times out.
- **Automation**  
Flag to enable or disable automation by IBM Tivoli System Automation.
- **ExcludedNodes**  
List of nodes on which IBM Tivoli System Automation actively pushes resources away or stops them. For example, this can be used for maintenance reasons.
- **ResourceRestartTimeOut**  
Amount of time in seconds IBM Tivoli System Automation waits to restart resources which were located on a failed node on another node.

The current values of the attributes described above can be listed with the command `Issamctrl`. The attributes are changed with the `samctrl` command. Refer to Chapter 13, "IBM Tivoli System Automation commands," on page 145 for a listing and description of these commands.

### TimeOut and RetryCount

The `TimeOut` and `RetryCount` parameter are always used together. They control how long IBM Tivoli System Automation will wait for an resource manager to do something and how many times it will tell the resource manager to try doing it again if it did not work the first time. In general, if it did not work the first time, the chances of it working on the second or subsequent attempts is fairly low.

### Start operations

The operation timer is started when IBM Tivoli System Automation first sends a resource start control operation to a resource. After that timer has started, there are 3 possibilities:

1. The resource changes to the desired state (online or offline) within the timeout period. With this the resource is in the state IBM Tivoli System Automation wants it to be, and no further actions are triggered .
2. The resource rejects the start control within the timeout period. What happens next depends on the reject code. If it indicates that the error is recoverable, IBM Tivoli System Automation will issue another start control operation against the resource. Every control operation try is counted and IBM Tivoli System Automation stops issuing control operations if the `RetryCount` is exceeded. If the error is not recoverable, the resource will go into a problem state. If it is a fixed resource nothing more will happen. If it is a floating resource, IBM Tivoli System Automation will try another instance, providing

## Controlling and administering IBM Tivoli System Automation

the instance that was tried to be started before is either Offline or Failed Offline. To get the resource out of the problem state, you should issue a reset operation against it.

3. The resource does not reach the desired state (online) within the timeout period. IBM Tivoli System Automation first issues a reset operation against the resource and waits until the reset operation has been accepted and the resource is offline. Then IBM Tivoli System Automation issues another start control operation against the resource. Every control operation try is counted and IBM Tivoli System Automation stops issuing control operations if the RetryCount is exceeded or the maximum timeout (which is  $\text{TimeOut} * \text{RetryCount}$ ) expires, whichever comes first.

If IBM Tivoli System Automation stops issuing control operations for either a fixed resource or a constituent of a floating resource, the OpState of this resource is set to failed offline. This indicates that this resource is no longer usable. Now manual intervention is required, the cause of the failure must be removed, and the resource must be reset with the RMC `resetsrc` command.

Note that no thresholding is implemented, the counter of the retries is reset if the resource reaches its desired state. For example, if the resource is started, is then online for a short period and stops after that, it is then restarted by IBM Tivoli System Automation in a loop.

Default values are:

- TimeOut = 60
- RetryCount = 3

You use the **samctrl -t** *Timeout* command to change the TimeOut value and **samctrl -r** *Retry\_count* to change the RetryCount value.

The IBM.Application class provides its own timeout value. If you add such a resource to a group, the general TimeOut is not used for this resource. As TimeOut for this group member the larger value of StartCommandTimeout or MonitorCommandPeriod attribute ( which are attributes of the IBM.Application resource) is used.

## Stop operations

If a resource should be stopped, no timeout is active. If the stop control operation is rejected and if the rejection code indicates recoverable error, it may be retried until the RetryCount parameter is exceeded . Once the stop operation has been accepted, or while acceptance/rejection is pending, IBM Tivoli System Automation will not resend the stop control operation.

After the stop control operation has been accepted, a time out and retry cycle does not make sense. If the order was accepted, then it has been issued and RMC is in the process of stopping the resource. There is no reason to believe that a subsequent stop order would do anything different (or could even be processed) or that a reset order could be handled by the resource at this point. If it has hung during a shutdown, then it is probably waiting on either a lock or user intervention (or the stop command was coded incorrectly). There is no use retrying the offline operation, a more forceful offline operation could result in data loss or corruption

## Automation

This flag indicates if IBM Tivoli System Automation automation functionality is enabled or not. If automation is disabled, IBM Tivoli System Automation will stop control operation. The state of resources will remain unchanged.

Default value is AUTO mode, which means that automation is turned on.

You use **samctrl -M F** to enable automation, **samctrl -M T** to disable automation.

## ExcludedNodes

This is a list of nodes where IBM Tivoli System Automation will stop all resources on and move them to another node if possible. For example, you have floating resource A which can run on four nodes node05, node06, node07 and node08. It is a member of resource group RG\_A. After you made the group online it is started on node05. If you add node05 to the list of excluded nodes, IBM Tivoli System Automation will stop the resource on node05 and restart it on one of the other nodes.

Caution: If you exclude a node and one or more mandatory members of a group cannot be restarted on another node, this may cause the whole group to stop.

Default value is an empty list, that means all nodes in the peer domain can be used.

You use **samctrl -u a** to add one or more nodes to the list of excluded nodes, **samctrl -u d** to delete nodes from that list and **samctrl -u r** to replace nodes in the list.

## ResourceRestartTimeout

The ResourceRestartTimeout is the amount of time IBM Tivoli System Automation waits to restart resources on an online node which were located on a failed node. The reason behind this is to give the resources or the failed nodes a chance to do some cleanup before the resources are moved to another system.

Default value is 5 seconds.

You use **samctrl -o** to specify the resource restart timeout.

## Examples

To list the current IBM Tivoli System Automation control parameters you use the **Issamctrl** command.

```
Issamctrl
```

Displaying IBM Tivoli System Automation Control Information:

```
Displaying IBM Tivoli System Automation Control Information:
```

```
SAMControl:
 TimeOut = 60
 RetryCount = 3
 Automation = Auto
 ExcludedNodes = {}
 ResourceRestartTimeout = 5
 ActiveVersion = [1.2.0.0,Tue 04 May 2004 12:30:48 PM EDT]
 Enable Publisher = Disabled
```

To add a node node05 to the list of excluded nodes, this command is entered:

```
samctrl -u a node05
```

To set the RetryCount parameter to 5, this command is entered:

```
samctrl -r 5
```

### Automation policy management

A core element of IBM Tivoli System Automation is the policy where the definitions are made how the system behaves. The policy consists of resource groups, relationships between resources, and/or groups and equivalencies. The main task of an administrator is to maintain this policy and to ensure that it is correct and recreatable.

To maintain one or more policies you can use the `samcfg` command. The command can be used in three different flavors:

1. To save a policy and restore it at a later point in time.
2. To modify a running policy.
3. To totally replace a policy.

### Saving configuration elements

The elements saved by `samcfg` are the elements of a policy which were created using the commands `mkrq`, `addrgmbr`, `mkequ`, and `mkrel`. The `samcfg` command cannot save elements provided by RSCT such as resources of the classes `IBM.Application`, `IBM.ServiceIP`, `IBM.NetworkInterface`, or `IBM.Test`.

The policy is saved in a file as a sequence of IBM Tivoli System Automation commands `mkrq`, `addrgmbr`, `mkequ`, and `mkrel` together with the corresponding options and parameters for each entity. You can take the default file name or define the name of the file yourself. This file is saved on the originator node where the `samcfg` command was invoked. An administrator may manually copy or backup this file at multiple nodes and locations.

Once the policy is written to a file, you can manually edit the file to change the policy.

For example, to save the policy for your cluster **SA\_Domain**, just enter:

```
samcfg -S
```

and the configuration is saved to a file with the default file name

```
/var/ct/SA_Domain/cfg/SA_Domain030904.135022
```

As stated above, resources which were defined using the RSCT command **mkrsrc** are **not** being saved by the **samcfg** command.

Nevertheless the RSCT resource definitions must be saved, too. There is no functionality in RSCT comparable to the `samcfg` command. The easiest way to save the RSCT definitions is to put the resource creation (`mkrsrc`) commands prior to definition in a script and run this script to create the resources.

### Restoring a saved configuration or replacing it with a new configuration

Use the `samcfg -R` command to restore a saved configuration.

Before you run the `samcfg` restore command, you must make sure that all needed RSCT resources exist, otherwise restoration will fail.

Before restoration, IBM Tivoli System Automation does a limited validation on the policies, and also stops all automation activities. This does not mean that the resources managed by IBM Tivoli System Automation are stopped. They remain untouched until the new configuration is committed.

This functionality can be used to modify an online configuration.

The file containing the saved configuration must be on the same node where `samcfg -R` will be invoked. For example, to restore the configuration for cluster **SA\_Domain**, enter:

```
samcfg -R SA_Domain030904.135022
```

You get the following information and question:

```
Reading configuration from file SA_Domain030904.135022:
```

```
.....
```

```
Verifying SAM configuration:
```

```
.....
```

```
Ready to commit new Configuration. Are you Sure? [Y/N]:
```

```
y
```

After you entered the "y" to commit the new configuration, the current master daemon terminates and a peer daemon takes over with the new policy and becomes master. The terminated daemon will automatically be restarted. Due to the termination of the daemon a message in the syslog is generated which looks similar to

```
Mar 10 15:51:42 lnxcml0x srcmstr: src_error=-9035, errno=0,
module='srchevn.c'@line:'248', 0513-035 The IBM.RecoveryRM Subsystem ended
abnormally. SRC will try and restart it.
```

This message can be ignored.

Using a text editor, you can modify an existing saved configuration as well as create a totally new one. Use the `samcfg -R` command as shown above to replace the current configuration with a changed or new one.

### Moving resource groups with the `rgreq` command

There are situations where the user would like the capability to move individual resource groups to another cluster node without affecting all resources currently running on the same node. For example, in a load-balancing situation, moving only one or several resource groups to another node may achieve the user's workload and performance objectives. Adjustments to the placement of resources can be done using the `rgreq -o move` command.

#### Scope of a move

The scope of a move is all the members of a top level resource group. Resources that are dependent upon one or more resources involved in the move might be affected, i.e. stopped and started. A move request cannot be issued against a single managed resource.

If the `MemberLocation` attribute of the top level resource group is set to `collocated`, no list of nodes has to be provided with the `rgreq` command. In this case, all resources are located on the same node and will be moved away from that node. If the resource group is not `collocated`, a list of one or more nodes has to be specified with the `-n` option of the `rgreq` command. All resources will be moved away from those nodes.

A move request issued against a resource group that contains only fixed resources will not be accepted. Also, a move request issued against an offline resource group will be refused. Once a move request is accepted, it cannot be cancelled by the issuer. While there is a move request already in progress, another move request on the same resource group will be rejected.

#### Processing of a move request

An offline phase is processed in the course of the move request, in which all the members of the moved resource group will be stopped first, even if they are running on nodes that are not contained in the list of nodes to move away from. This is done to avoid complications when placing the resources later on. The binder assigns a new node placement for all resource group members, and the resource group is restarted. This is the online phase of the move. Note that if it turns out to be impossible to restart the mandatory members of the top level resource group while honouring the list of nodes to move away from, the list will be ignored and resources may be restarted upon it. Likewise, if the only place to restart a resource is the original system it was running on, then it will be restarted there if it is a mandatory resource.

A move request is automatically removed when the move action is carried out. The `MoveStatus` dynamic attribute of the resource group being moved will show values indicating the progress of the move.

#### Move and relationships

In addition to performing a move request on the members of a top level resource group, there may be other resource groups and/or resource group members outside the moved top level group, which placement must be adjusted according to defined relationships constraints. This applies to the following relationships:

- `Collocation`
- `AntiCollocation`
- `DependsOn`
- `DependsOnAny`
- `StopAfter`
- `ForcedDownBy`

In addition, the `Affinity` and `AntiAffinity` relationships might not be fulfilled after the move.

## Diagnosing IBM Tivoli System Automation Resources

To get more information about resources managed by IBM Tivoli System Automation you can use the **samdiag** command, which is documented under “samdiag” on page 216. This command is mainly intended to be used in situations where it is not obvious to the user what is happening on the system and why.

**Note:** The samdiag command provided for Release 1 of IBM Tivoli System Automation could only be executed on the node where the master daemon was running on. So running samdiag on a Release 2 daemon will generate an error if Release 1 and Release 2 daemons coexist in the same cluster, and if the master daemon is on a Release 1 node.

To get information about a resource group called “apacherg”, you use the command:

```
samdiag -g apacherg
```

### Output:

```
Diagnosis::Resource: apacherg/ResGroup/IBM.ResourceGroup
type: CHARM Resource Group
Status -
 Observed: Offline - SoftDown
 Desired: Offline - Requested Offline
 (Nominal: Offline - Nominal State: Offline)
 Automation: Idle - CharmBase trigger linked
 Startable: Yes - Resource is startable
 Binding: Unbound - Binding status initialized
 Compound: Satisfactory - Satisfactory

Resource Based Quorum: None -
Members and Memberships:
+---bind/HasMember ---> RA/Float/IBM.Test
Group Constraint: None
Binding Constraints:
Flags:
 None
Orders:
 Outstanding Order: None - Resource is Unavailable
Dependencies:
 Start: Satisfied
 +---InCluster ---> Cluster
 Stop: Satisfied
Binding exceptions:
 There are unbound members.
Static Relationships:
 +---InCluster ---> Cluster
Dynamic Relationships:
 +---bind/HasMember ---> RA/Float/IBM.Test"
```

The following provides an interpretation of some of the information given in the example:

- The ObservedState should show the same value as OpState and NominalState. If this is not the case, contact the support center serving your location.
- Different values for DesiredState and NominalState indicate that a request has been issued against the resource group.
- The AutomationState can be 'busy' or 'idle'. 'Busy' means that the IBM Tivoli System Automation daemon (IBM.RecoveryRM) is waiting for another resource manager to start or stop a resource. After this has been completed the AutomationState changes to 'idle'. If this is not the case, contact the support center serving your location.
- A Startable state of 'No' indicates that some relationships, for example a DependsOn relationship, are not correctly set.

## Controlling and administering IBM Tivoli System Automation

- A BindingState of 'Unbound' comes along with a Observedstate 'Offline'. It indicates that the resource group is offline. Before it can be set to online a binding step must be performed which chooses the correct constituent. Then the BindingState is set to 'Bound'. A BindingState of 'Bound' for resource groups being offline is an error.
- A CompoundState of 'Satisfactory' indicates that ObservedState and DesiredState are the same. CompoundStates 'Inhibited', 'Denied', or 'Broken' indicate errors like relationships which have not been fulfilled or 'broken' resources.
- Resource Based Quorum equals 'None' means that all floating resources of the group do not support a Resource Based Quorum of their own and the Resource Based Quorum flag is not set.
- 'Bind/HasMember' describes the relationship between resource group apacherg and its floating member RA/Float/IBM.Test. When performing the binding step mentioned above a constituent is selected for each of the 'Bind/HasMember' relationships. This constituent is temporarily bound to the resource group before this resource group is set to online.
- Outstanding Orders refer to the AutomationState. If AutomationState is not 'idle', the pending command is shown here.
- Start/Stop Dependencies show when a policy prevents the start of a resource.
- Binding exceptions provide a closer explanation of BindingState.
- Static relationships mean that all constituent resources and resource groups are members of the cluster.
- Dynamic relationships are temporary relationships caused by the binding step.

## Using the IBM Tivoli System Automation TEC event interface

Whenever IBM Tivoli System Automation changes its configuration, resource status or encounters any problems, the Tivoli Enterprise Console (TEC) event interface can be used to notify the system administrator. There are two ways to notify the administrator:

1. Sending events to the IBM Tivoli Enterprise Console. As a prerequisite the TEC publisher function must be enabled (see “Enabling the TEC publisher function”).
2. Publishing IBM Tivoli System Automation internal attributes into the RSCT infrastructure. The administrator must subscribe to one of the Event Resource Manager (ERRM) scripts in order to get event information. Refer to the manual *IBM Reliable Scalable Cluster Technology for Linux, Technical Reference*, SA22–7983, how to do this; also see the description in the preceding section “Generating events in case of failures” on page 128.

## What is the Tivoli Enterprise Console?

The Tivoli Enterprise Console (TEC) is a rule-based event management application that uses a central server to process incoming events. TEC acts as a central collection point for alarms and events from a variety of sources, including those of Tivoli applications, Tivoli partner applications, customer applications, network management platforms, and relational database systems.

## What are events?

Events are units of information which can represent performance data or also can indicate problems, status or changes regarding resources. Usually IBM Tivoli System Automation sends events when the help of an administrator is required.

A language called Basic Recorder of Objects in C (BAROC) is used to define the structure of events and their properties. These definitions are stored in files with the extension **.baroc**.

## Sending events to the TEC

IBM Tivoli System Automation uses the TEC event interface ‘Tivoli Event Integration Facility (EIF)’ to send events to the TEC. The events will be sent to the TEC server non-TME event port.

The following reasons cause IBM Tivoli System Automation to send events to the TEC:

- The status of a resource or cluster changed.
- A resource was added or deleted.
- A relationship was added or deleted.

## Enabling the TEC publisher function

In order to receive and view the events coming from different sources like programs, systems or network devices you have to enable the TEC publisher by performing the following steps:

1. Import, compile, load, and activate the TEC baroc file in the TEC server (`/usr/sbin/rsct/samples/tec/SystemAutomation.baroc`). Refer to manual *IBM Tivoli Enterprise Console Rule Builder’s Guide*, GC32–0669, how to do this.
2. Copy files `/usr/sbin/rsct/samples/tec/samPublisher.conf` and `/usr/sbin/rsct/samples/tec/TECPublisher.conf` into `/etc/Tivoli/tec` on each cluster node.
3. Customize the publisher configuration file `/etc/Tivoli/tec/samPublisher.conf` and the TEC EIF file `/etc/Tivoli/tec/TECPublisher.conf` on each cluster node.
4. Enable the publisher with the command **samctrl -e P** on a IBM Tivoli System Automation master node. By default the publisher is disabled.
5. Restart all cluster nodes using the **stoprpnode/startrpnode** commands in order to activate the new configuration files.

## Controlling and administering IBM Tivoli System Automation

### Publisher configuration file `/etc/Tivoli/tec/samPublisher.conf`

The publisher configuration file specifies a list of all target consumers and their parameters. This is the syntax format of the publisher configuration file:

```
#
Publisher configuration file
file name: /etc/Tivoli/tec/samPublisher.conf
#
File format:
<keyword>=<value>
#
Publisher - unique name of the publisher
LibraryPath - name of the publisher library
ConfigPath - full path to the TEC EIF configuration file
#
Multiple entries of the Publisher, LibraryPath and ConfigPath can be specified.
One triplet for each publisher target consumer.

Publisher=TEC
LibraryPath=libTECPublisher.so
ConfigPath=/etc/Tivoli/tec/TECPublisher.conf

Publisher=TEC2
LibraryPath=libTECPublisher.so
ConfigPath=/etc/Tivoli/tec/TECPublisher2.conf
```

Figure 11. Syntax format and sample of the publisher configuration file

The following syntax rules apply:

- lines starting with # and blank lines will be ignored.
- parameter format: `<keyword>=<value>`.
- keyword "Publisher" starts a new triplet of "Publisher", "LibraryPath" and "ConfigPath" parameters.
- keyword "Publisher" specifies the unique name of the publisher.
- keyword "LibraryPath" specifies the full path to the publisher library.
- keyword "ConfigPath" specifies the full path to the TEC EIF configuration file

### TEC EIF configuration file `/etc/Tivoli/tec/TECPublisher.conf`

The TEC EIF configuration file specifies all parameters needed to connect to a specific TEC server. The file name must match the name specified as "ConfigPath" parameter in the publisher configuration file.

The syntax format of the TEC EIF file for the TEC publisher is the existing TEC EIF configuration file syntax.

```

TEC EIF configuration file
#
File format:
<keyword>=<value>
#
ServerLocation - name of the host where the TEC server is running
ServerPort - port number on which the TEC server is listening
to non-TME TEC events. TME TEC events are not supported
5529 - default non-TME port for TEC servers on Windows
0 - default non-TME port for TEC servers on AIX and Linux
ConnectionMode - distinguishes between connection_oriented OR connection_less
- (default is connection_oriented)
BufferEvents - specifies whether the event buffering cache file is enabled
(YES | MEMORY_ONLY | NO) (default is YES)
BufEvtPath - specifies the full path name of the cache file
(default: /etc/Tivoli/tec/cache)
FilterMode - specifies whether events that match a Filter are sent to
the event server (FilterMode=IN) or are discarded
(FilterMode=OUT)
Filter - Filter:Class=class_name;[attribute=value[;attribute=value]*]
#
For a description of all supported keywords and their values refer to manual:
"Tivoli Event Integration Facility User's Guide", GC32-0691,
Chapter: "Appendix B. Keywords for Configuration Files".
ServerLocation=tecserver.ibm.com
ServerPort=5529
ConnectionMode=connection_less
BufferEvents=YES
BufEvtPath=/etc/Tivoli/tec/TECPublisher.cache

Default Filters
Filter all relationship add / delete events
Filter:Class=SystemAutomation_Relationship_Configuration_Change
Filter all resource add / delete events
Filter:Class=SystemAutomation_Resource_Configuration_Change
Filter resource status events with severity HARMLESS
Filter:Class=SystemAutomation_Resource_Status_Change;severity=HARMLESS
Filter resource status events with severity WARNING
Filter:Class=SystemAutomation_Resource_Status_Change;severity=WARNING

#Put the server name or IP address of the server on which the TEC is running into
#the "ServerLocation" field.

```

Figure 12. Syntax format and sample of the TEC configuration file

To avoid that the TEC is flooded with a huge amount of messages, filters are provided in the # TEC EIF configuration file. Per default all filters are enabled, which results in only critical messages being sent to the TEC. If you want additional messages to be sent to the TEC, disable the corresponding filter by using the comment character #.

### Enabling the publisher

Per default the Publisher function is disabled. To query the status of the publisher issue the following command:

```
node1:/usr/sbin/rsct/samples/tec # lssamctrl
```

The following IBM Tivoli System Automation control information is displayed:

```
SAMControl:
 Timeout = 60
 RetryCount = 3
 Automation = Auto
```

## Controlling and administering IBM Tivoli System Automation

```
ExcludedNodes = {}
ResourceRestartTimeout = 5
ActiveVersion = [1.2.0.0,Fri Apr 16 16:05:50 2004]
EnablePublisher = Disabled
```

To enable the publisher issue this command on the master node:

```
node1:/usr/sbin/rsct/samples/tec # samctrl -e P
```

To disable the publisher issue this command on the master node:

```
node1:/usr/sbin/rsct/samples/tec # samctrl -d P
```

## Publishing IBM Tivoli System Automation internal attributes into the RSCT infrastructure

This function makes IBM Tivoli System Automation internal attributes known to the RSCT infrastructure. For this purpose, the resource classes IBM.ResourceGroup, IBM.Equivalency, and IBM.ManagedResource are extended with the dynamic data structure attribute AutomationDetails. The dynamic data structure AutomationDetails has the following attributes:

- **CompoundState** – overall status of the resource including group dependencies. Shows how far the resource has reached the **DesiredState**. For example, "Satisfactory" means that the resource or resource group has reached the requested user status.
- **DesiredState** – user requested status of the resource. For example, "online" means that the user requested that the resource should be online.
- **ObservedState** – actual status of the resource from an automation perspective. For example, "online" means that the resource is currently online.
- **BindingState** – status indicating if the resource is bound to a specific system. For example, "bound" means that the resource is currently bound to a specific system.
- **AutomationState** – status indicating if the resource is currently being automated. For example, "Idle" means that IBM Tivoli System Automation is currently not trying to start or stop the resource.
- **StartableState** – status indicating if the resource can be started. For example, "startable" means that it is currently possible to start this resource.
- **HealthState** – health status of the resource. This is reserved for future releases.

The **lsequ** and **lsrg** commands have been extended to show these attributes.

Each change of the value of one of these attributes indicates a status change of a resource and will be published to RSCT. If the TEC publisher is enabled (see "Enabling the TEC publisher function" on page 119), these status changes are also shown as TEC events.

---

## Enabling IBM Tivoli System Automation for Coordinated Cross Platform Disaster Recovery for zSeries

Today businesses and companies depend on disaster recovery solutions to recover critical data.

Therefore, IBM Tivoli System Automation supports Coordinated Cross Platform Disaster Recovery for zSeries (xDR). xDR for zSeries is based on functionality which is planned for Geographically Dispersed Parallel Sysplex (GDPS) 3.1.

xDR for zSeries provides a coordinated disaster recovery solution for systems running on zSeries, including z/OS and Linux for zSeries under z/VM.

Note that if you want to use the xDR functionality, certain versions of z/VM, Linux for zSeries, GDPS and IBM Tivoli System Automation must be installed. The functionality and these versions are explained and described in detail in the GDPS (Geographically Dispersed Parallel Sysplex) manuals.

Also note that the xDR naming convention requires that the names of cluster and nodes must not exceed a length of 32 characters. For xDR cluster names are not case sensitive.

To use xDR, IBM Tivoli System Automation has to be customized as described in the following.

### Customizing the Tivoli Enterprise Integration Facility (EIF)

xDR uses the event interface 'Tivoli Enterprise Integration Facility (EIF)' of the Tivoli Enterprise Console (TEC) to send events. In order to enable xDR to send events, add the following lines to the publisher configuration file `/etc/Tivoli/tec/samPublisher.conf` (see "Publisher configuration file `/etc/Tivoli/tec/samPublisher.conf`" on page 120 for a sample of this file).

---

```
Publisher=XDR_GDPS
LibraryPath=/usr/sbin/rsct/lib/libXDRPublisher.so
ConfigPath=/etc/Tivoli/tec/xdr.conf
```

---

If you want to send events to a second GDPS K system, add another entry to the publisher configuration file `/etc/Tivoli/tec/samPublisher.conf` like

---

```
Publisher=XDR_GDPS2
LibraryPath=/usr/sbin/rsct/lib/libXDRPublisher.so
ConfigPath=/etc/Tivoli/tec/xdr2.conf
```

---

Create the xDR configuration file `/etc/Tivoli/tec/xdr.conf` and `/etc/Tivoli/tec/xdr2.conf` respectively, similar to the TEC EIF configuration file (see "TEC EIF configuration file `/etc/Tivoli/tec/TECPublisher.conf`" on page 120), with the content and values for `ServerLocation` and `ServerPort` as shown here:

```
#
XDR EIF configuration file
#
File format:
<keyword>=<value>
#
ServerLocation - name of the host where the GDPS is running
ServerPort - port number of the NetView Event Receiver
BufferEvents - specifies if the event buffering cache file is enabled
(YES | MEMORY_ONLY | NO) (default is YES)
BufEvtPath - specifies the full path name of the cache file
(default: /etc/Tivoli/tec/cache)

ServerLocation=netview.ksys.ibm.com
ServerPort=5529
BufferEvents=NO
BufEvtPath=/etc/Tivoli/tec/xdrBuffer.cache
```

Figure 13. Syntax format and sample of the xDR EIF configuration file

This must be performed on every node in the cluster.

Enable the publisher with the command **samctrl -e P**. By default the publisher is disabled.

## Controlling and administering IBM Tivoli System Automation

### Installing Cpint

Cpint is an interface to call VM CP commands from Linux. It comes with the SUSE LINUX distribution. xDR requires Cpint to be installed and loaded on all nodes in the cluster. If you cannot find it in the distribution, you can download the latest code from <http://linuxvm.org/Patches/index.html>.

#### Example how to install the package cpint-1.1.5.tar.gz:

1. Unzip the file:

```
gunzip -d cpint-1.1.5.tar.gz
tar -xvf cpint-1.1.5.tar
```

2. build cpint:

```
cd cpint-1.1.5
make
```

3. install cpint:

```
make install
```

Issue the command **cpint\_load** to load the device and use cpint.

### xDR heartbeating

Make sure that EIF has been customized as described under “Customizing the Tivoli Enterprise Integration Facility (EIF)” on page 123 and that the cluster has been restarted each time the configuration files changed. Stop the cluster using the command **xdrStopCluster** and the command **xdrstoperpd** on each node. And restart the cluster with the command **startprdomain *clustername***. Enable the publisher with the command **samctrl -e P**. By default the publisher is disabled.

Add the following line to the file */etc/inittab*:

```
xdr:2345:respawn:/usr/bin/XDRHeartbeat
```

Then use the command **telinit q** in order to activate this new entry and to start heartbeating. You must do this for each node in the cluster.

### DASD error reporting daemon

The DASD error reporting daemon (erpd) must be made available using IBM Tivoli System Automation. Make sure that the publisher configuration file and the TEC EIF configuration file are configured as shown above. In order to run the enableErpd-script which defines a resource, defines a resource group, adds the resource to this group, and starts this group, type:

```
enableErpd
```

Issue this command on each node of the cluster. This ensures that the DASD error reporting daemon is running.

Use the **chrg** command to stop it and set the resource group offline:

```
chrg -o Offline <groupname>
```

where groupname is xdrerpd<node-name> group.

### Stopping an xDR cluster / node for maintenance

In order to stop xDR for a whole cluster, use the xdrStopCluster-script. It uses the **stopprdomain** command to bring down the cluster and sends an EIF event to GDPS, disabling xDR for this cluster. To run this script just type:

```
xdrStopCluster
```

Then stop the error reporting daemon by issueing the command **xdrstoperpd** on each node.

In order to stop xDR for one node CP only, use the **samctrl** command, for example:

```
samctrl -u a <nodename>
```

In order to restart xDR on this node, type:

```
samctrl -u d <nodename>
```

### Dynamically verifying resources

Usually resource verifications are performed at configuration time when a resource is defined to IBM Tivoli System Automation. Then the user immediately gets notified when a problem occurs, and the definition of a new resource fails.

This may, however, not be the case after a resource has been defined and then a configuration change occurs. IBM Tivoli System Automation gets notified after the configuration change has been made, and then it has to accept and react to these changes, which eventually may lead to one or more resources becoming invalid.

This may happen when you, for example, define a resource using the `mkrsrc` command, change the values of a resource using the `chrsrc` command, or remove a defined resource using the `rmrsrc` command.

In order to verify such configuration changes and to convey the validity of a resource to the user, the resource classes `IBM.ResourceGroup`, `IBM.ManagedResource`, `IBM.Equivalency`, and `IBM.ManagedRelationship` contain a dynamic attribute `ConfigValidity`. `ConfigValidity` contains a string which explains why the resource is invalid.

Use the `lsrsrc -Ad` command to display the value of `ConfigValidity` together with the values of the other dynamic attributes of a resource.

The following verifications are performed:

- **A resource group's AllowedNode attribute is empty**

When a node is removed, it may cause an equivalency to contain an empty member list. A resource group will become invalid if it uses this empty equivalency as its "AllowedNode" attribute. When this happens, the "ConfigValidity" dynamic attribute of the resource group will contain string "AllowedNode is empty".

- **Intersection of nested resource group's AllowedNode is empty**

In a collocated resource group, all nested inner resource groups and the containing resource group must have at least one node in common. If there is only one node in common, and that node is removed, all the resource groups become invalid. When this happens, the "ConfigValidity" dynamic attribute of the resource group will contain the string "No common node in collocated nested resource group".

- **No node to run a resource**

In a resource group, there may be the case when there is only one node in common between the resource group's AllowedNode and a member resource's NodeNameList. If this node is removed, the resource group becomes invalid. When this happens, the "ConfigValidity" dynamic attribute of the resource group will contain string "No common node to start a resource".

- **No node to satisfy a relationship**

In a DependsOn relationship, with implied collocation, the NodeNameList of the source and NodeNameList of the target resource must have at least one node in common. If this node is removed, the relationship becomes invalid. When this happens, the "ConfigValidity" dynamic attribute of the managed relationship will contain string "No common node between source and target".

- **Cannot satisfy an AntiCollocated relationship - 1**

If two mandatory floating resources in a resource group have an AntiCollocated relationship among each other, and removal of nodes leaves only one node in the resource group's AllowedNode, the resource group becomes invalid. When this happens, the "ConfigValidity" dynamic attributes of the resource group and the AntiCollocation managed relationship will contain string "An AntiCollocated relationship cannot be satisfied".

- **Cannot satisfy an AntiCollocated relationship - 2**

A removal of nodes causes two floating resources to have only one constituent left on the same node. But the two have an AntiCollocated relationship. When this happens, the "ConfigValidity" dynamic attributes of the AntiCollocated managed relationship will contain string "An AntiCollocated relationship cannot be satisfied".

- **Propagated invalidity**

Any invalid inner resource group will cause all enclosing resource groups to become invalid. When this happens, the "ConfigValidity" dynamic attribute of the affected resource group will contain string "An enclosed resource group is invalid."

### IBM Tivoli System Automation Hints and Tips

This section provides various hints and tips which are helpful when operating a cluster with System Automation for Linux.

#### Rebooting a node

Do not reboot a node when any resources are running on this node. First stop all running resources, using the following command:

```
samctrl -u a <node_name>
```

Now you can safely reboot the node,

This command also prevents resources from being started on this node. Reenable the node for running resources by entering:

```
samctrl -u d <node_name>
```

#### Stopping a node

Before you stop a node (e.g. with the RSCT command **stoprnode**), you must exclude this node from automation with the following command:

```
samctrl -u a <node_name>
```

The exclude must be done even if there are currently no resources online on the node.

After you have started the node again and it is online, you must reestablish automation on that node by entering:

```
samctrl -u d <node_name>
```

#### Generating events in case of failures

RSCT has the ability to generate events in case a dynamic attribute of a resource changes. This is done by the event response resource manager (ERRM). The event response resource manager provides a set of commands that enable you to monitor events of interest (called conditions) and have the RMC system react in particular ways (called responses) if the event occurs.

So, for example, you can subscribe for the OpState of a resource group and get an email if the status changes. You can also monitor different resources in your system which are critical. For further explanation how to generate such events, see the chapter on basic resource monitoring in the *IBM Reliable Scalable Cluster Technology for Linux, Technical Reference* manual, SA22-7893.

---

## Chapter 12. Resource managers provided by IBM Tivoli System Automation

This chapter shows the resource managers provided by IBM Tivoli System Automation.

---

### Using the Global Resource Manager

This section describes the characteristics of the Global Resource RM.

The Global Resource RM (IBM.GblResRM) provides the following two resource classes:

1. IBM.Application:

This class allows additional types of resources (e.g. business applications) to be monitored and controlled through the RMC subsystem. These resources can then be automated or recovered by management applications such as IBM Tivoli System Automation.

2. IBM.ServiceIP:

This class is used to manage IP addresses that can be started, stopped, and moved between network adapters and nodes within a peer domain under the control of the RMC subsystem. These IP addresses will typically be provided to clients that are connecting to some service that is running within the domain. IBM Tivoli System Automation can be used to keep the service and its associated IP address active, even through failures within the domain.

The resource manager (and access to its classes) is operable in peer domain mode only.

The following subsections describe the external characteristics of the resource classes that are supported by this resource manager. Each of the subsections will describe one resource class, including its persistent and dynamic attributes, actions, etc.

### What is the IBM.Application resource class?

The IBM.Application resource class allows new types of floating and fixed resources to be created, monitored and controlled through the RMC subsystem. These resources can then be automated or recovered by IBM Tivoli System Automation. In order to create a new resource, the following three scripts (resp. commands) must be provided:

1. A start script (or command) to bring the resource online.
2. A stop script (or command) to take the resource offline.
3. A script (or command) to monitor the resource through polling.

Besides these scripts, there are the following basic parameters to the IBM.Application resource class:

1. The name of the resource.
2. The nodes where the resource can run.
3. A user name used to start/stop/monitor the application.
4. Method to be used to synchronously or asynchronously start/stop the application.
5. Different timeouts.
6. Characterization of the resource as either critical or non-critical.
7. Determination of the monitoring frequency.
8. Identification of the resource as fixed or floating.

Each generic resource that is instrumented through the IBM.Application resource class is considered to be a global resource meaning that it is not tied to a single node. However, the resource may be defined to exist on only a subset of the nodes of the cluster. For each generic resource, one instance of the IBM.Application resource class must be created. This instance is called an aggregate resource since it

## Global resource RM

represents the floating resource that can move between nodes. In addition, there will be one instance of the IBM.Application resource class for each node where the generic resource exists. These are called constituent resources of the aggregate resource. Constituent resources are fixed resources in the sense that they exist on exactly one node of the cluster. Figure 14 illustrates the difference between aggregate and constituent resources.

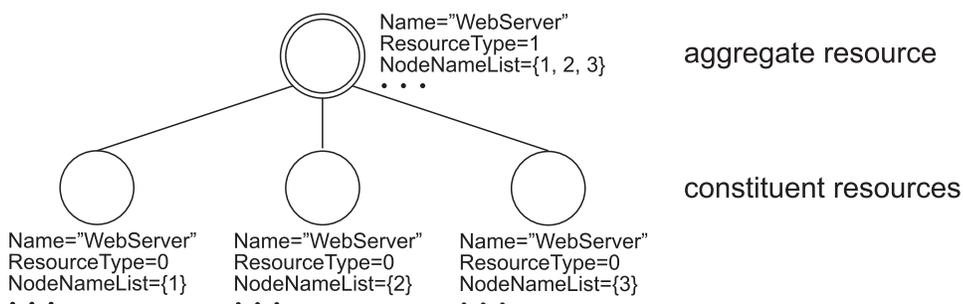


Figure 14. Aggregate resource and constituent resources

The constituent resources are automatically created or deleted as the definition of the aggregate resource is changed. Most management operations are done through the aggregate resource, but some applications may choose to monitor or operate on the constituents directly. Changes made to the aggregate resource are automatically applied to all constituents, whereas the change of an attribute of a constituent affects this constituent only and is not delivered to other resources (for example, the constituent on one node may have a different start command or monitoring interval).

When nodes are removed from the NodeNameList of the aggregate resource, the constituents are deleted automatically.

### Attributes used by IBM.Application

This section describes the attributes that are used by resources of the IBM.Application resource class.

When a resource of this class is created with the RMC command **mkrsrc**, it must have these attributes:

- Name
- StartCommand
- StopCommand
- MonitorCommand
- UserName

Resources of this class may have these attributes:

- NodeNameList
- ResourceType
- StartCommandTimeout
- StopCommandTimeout
- MonitorCommandTimeout
- MonitorCommandPeriod
- RunCommandsSync
- ProtectionMode

Resources of this class have the following dynamic attribute:

- OpState

**Name attribute**

The Name persistent attribute is a user– defined name for the generic application resource. Both the aggregate and constituent resource will have the same value for this attribute.

A value for this attribute must be specified when a new IBM.Application resource is created and it must be unique.

The attribute must be of type character string.

**NodeNameList attribute**

The NodeNameList persistent attribute is an array of strings that indicates which nodes the IBM.Application resource is available on.

If the resource is floating, the Global Resource RM will ensure that there is one constituent (i.e. fixed) resource for each node name in this list. Constituent resources are implicitly created or deleted as necessary to match the entries in this list. Constituent resources will only contain one entry in this array since they are fixed resources and thus only available on one node. This attribute for a floating resource is implicitly modified if a constituent resource is explicitly removed by the administrator so that the aggregate and constituent resource relationship is always consistent.

This list may be empty for a floating resource which means that it is not available anywhere and constituents may be added separately.

The list may contain at most one name if the resource is fixed (i.e. ResourceType=0). If no name is given for a fixed resource, then RMC will provide a default because the fixed resource is tied to a node and therefore cannot be created without a node name or node id.

The value of this attribute for aggregate resources may be changed with the chrsrc command. An attempt to modify this attribute for a constituent resource will generate an error.

**ResourceType attribute**

With the ResourceType persistent attribute you identify whether the resource is fixed or floating. An integer value of 0 indicates that the resource is fixed, a value of 1 indicates that it is a floating resource. This attribute defaults to floating if not specified when a new IBM.Application resource is created.

**StartCommand attribute**

The value of the StartCommand persistent attribute contains the exact command and arguments that will be executed when the resource manager receives a start request for the corresponding resource instance. The command is only executed by constituent resources even if the online request was issued to the aggregate resource. In this case, the resource manager will choose a constituent resource if not specified to execute the online request.

The command is executed under the user id specified with the UserName attribute.

The command is run with the authority and environment of the specified user.

Whether the resource manager waits for the command to complete is controlled by the RunCommandsSync attribute (see below for details). The command name must be a character string and it must be an absolute path (i.e. it must begin with a '/'). It must exist and be executable on each node where the resource is accessible (i.e. where there is a constituent).

This attribute must be specified when a new IBM.Application resource is defined.

The command may return the following values:

- 0** Command has run successfully.
- != 0** Error occurred during command processing.

See “How the resource manager handles return codes of the StartCommand, StopCommand, and MonitorCommand” on page 135.

## Global resource RM

### StopCommand attribute

The value of the StopCommand persistent attribute contains the exact command and arguments that will be executed when the resource manager receives a stop request for the resource instance. A stop request for the aggregate will be issued by all constituents. All other aspects related to running the command are the same as for StartCommand. This attribute must be specified when a new IBM.Application resource is defined. The command may return the following values:

- 0 Command has run successfully.
- != 0 Error occurred during command processing.

See “How the resource manager handles return codes of the StartCommand, StopCommand, and MonitorCommand” on page 135.

### MonitorCommand attribute

The value of the MonitorCommand persistent attribute contains the exact command and arguments that will be executed periodically to determine or update the operational state (OpState attribute) of the resource. The exit value from the command is used as the new OpState of the resource:

- Unknown=0
- Online=1
- Offline=2
- Failed Offline=3
- Stuck Online=4
- Pending Online=5
- Pending Offline=6

At least the Online and Offline status should be set by the MonitorCommand script. The IBM.GblResRM runs this command every MonitorCommandPeriod seconds when there are any subscribers to the OpState dynamic attribute. All other aspects of running the command are identical to those described under StartCommand. To avoid consuming system resources, this command should be as efficient as possible. The name of the MonitorCommand must be an absolute path (i.e. it must begin with a '/'). It must exist and be executable on each node where the resource is accessible (i.e. where there is a constituent).

This attribute must be specified when a new IBM.Application resource is defined. To learn more about the return value of the MonitorCommand attribute see “Important issues when defining IBM.Application resources” on page 134 and “How the resource manager handles return codes of the StartCommand, StopCommand, and MonitorCommand” on page 135.

### MonitorCommandPeriod attribute

The value of the MonitorCommandPeriod persistent attribute specifies the amount of time (number of seconds) to wait between invocations of the MonitorCommand. This period is started after the prior invocation completes. This attribute must be of type integer and it must be greater than 0. It defaults to 5 seconds.

### MonitorCommandTimeout attribute

With the MonitorCommandTimeout persistent attribute you specify the amount of time a monitor command is allowed to run before it is killed via killpg(). If the command times out, the operational state (OpState attribute) of the resource is set to Unknown=0. This attribute must be of type integer and it must be greater or equal to 0. The value of this attribute must be lower or equal to the MonitorCommandPeriod. A value of 0 means no timeout. The default value for this attribute is 5 seconds.

### StartCommandTimeout attribute

With the StartCommandTimeout persistent attribute you specify the amount of time a start command is allowed to run before it is killed via killpg(). Furthermore, this attribute also

specifies the amount of time after which IBM Tivoli System Automation expects the resource to be online, i.e. IBM Tivoli System Automation uses this value instead of the default timeout given with the control parameters.

This attribute must be of type integer and it must be greater or equal to 0. A value of 0 for this attribute means no timeout. The attribute is not used if the RunCommandSync attribute is set to 0. The default value for this attribute is 5 seconds.

#### **StopCommandTimeout attribute**

With the StopCommandTimeout persistent attribute you specify the amount of time a stop command is allowed to run before it is killed via killpg(). This attribute must be of type integer and it must be greater or equal to 0. A value of zero means no timeout. The default value for this attribute is 5 seconds.

#### **RunCommandsSync attribute**

You use the RunCommandsSync persistent attribute to control whether the start/stop commands are executed synchronously with the online()/offline() method. If the value of this attribute is set to the integer value 1 which is the default, then the response to the online()/offline() methods will not be completed until the command completes or times out. Any stderr/stdout outputs will be returned in the response for this case. If the value of this attribute is 0, then the IBM.GblResRM will "fire and forget" the start/stop commands. As soon as the fork/exec completes successfully, the resource manager forgets about them and they run completely unattached from the resource manager.

This attribute defaults to 1. Timeouts will not be applied to commands when this attribute is set to 0.

If your StartCommand is the application executable and the command does not return in a certain amount of time, but is running as long as your application is running, you cannot use the synchronous mode. In case of the synchronous mode the resource manager would kill the command after the StartCommand timeout has expired. Use RunCommandsSync=1 when you know how long the StartCommand normally takes to complete. Adapt the StartCommandTimeout attribute to this time. During heavy system load this time may be longer even in case of a normal start up without any error conditions. Use RunCommandsSync=0 when the command does not complete until your application is running (e.g. the executable of your application). If you want to run the application executable directly (asynchronous mode), but use the synchronous command mode and send the command to the shells background, redirect I/O file descriptors. If the resource manager is still connected to the process, I/O descriptors the StartCommand timeout will kill your process.

**Example:** "bin/myprocess" is an application executable not a start command. It will run as long as the application is running.

```
StartCommand="/bin/myprocess > /dev/null 2>&1 &" RunCommandsSync=1
```

#### **UserName attribute**

The UserName persistent attribute defines a user name under which the MonitorCommand, StartCommand and StopCommand are run. The commands are run with the authority and environment of the specified user. A check will be made on each node to ensure that the user name exists whenever the configuration of the resource is modified.

This attribute must be specified when a new IBM.Application resource is defined. The attribute must be of type character string.

#### **ProtectionMode attribute**

You use the ProtectionMode persistent attribute to specify whether the resource is critical. If it is critical, then the IBM.ConfigRM decides if the resource can be started as requested. (For further detail on this behavior, see Chapter 9, "Protecting your resources – quorum support," on page 95).

The attribute may have the integer values 0 (Non-Critical) or 1 (Critical), it defaults to Non-critical. If the resource is set to Critical, monitoring will immediately start, even if no subscriber to this resource exists.

## Global resource RM

### OpState attribute

The value of this dynamic state attribute contains the operational state of the resource as determined by the exit code from running the MonitorCommand periodically. The possible values for this attribute as defined by the state transition diagram in the RMC architecture are:

Unknown=0  
Online=1  
Offline=2  
Failed Offline=3  
Stuck Online=4  
Pending Online=5  
Pending Offline=6

This attribute is available from both the aggregate resource and all constituent resources. The value for an aggregate resource is a roll-up of the states from each of the constituents.

### HealthCommand

Reserved for future use.

### HealthCommandPeriod

Reserved for future use

### HealthCommandTimeout

Reserved for future use

### InstanceName

Reserved for future use

### InstanceLocation

Reserved for future use

## Actions used by IBM.Application

This section describes the actions that can be performed on resources of the IBM.Application resource class.

### The refreshOpState action

In normal operation the MonitorCommandPeriod attribute determines the interval, the OpState of an IBM.Application resource is evaluated by its monitoring script. In case a resource is able to detect a failure by itself, it is possible to trigger an immediate run of the monitoring script which is monitoring the resource. This will result in an immediate OpState refresh of the application resource.

For example, to refresh the OpState of the resource 'WebServer' running on node02, issue the following command on any node:

```
runact -s "Name='WebServer' && NodeNameList={'node02'}" IBM.Application refreshOpState
```

**SendEIFevent** Reserved for future use.

## Important issues when defining IBM.Application resources

Keep the following in mind when defining an IBM.Application resource:

1. In order to satisfy goal driven automation IBM Tivoli System Automation resources are always monitored. No matter if the resource is started or stopped, the monitor command has to determine the actual state of the resource. Don not put the monitor command on a filesystem which is not always present (e.g. a NFS mount which is part of a policy and is only mounted if the NFS resource is started) .If you see an IBM.Application OpState of "unknown", check the system log and take care the GblResRM resource manager can access the monitor command at any time.

2. The GblResRM resource manager will kill any command which is running longer than the timeout value allows. If you see IBM.Application resources reporting OpState “unknown” during heavy system load, the command probably takes longer to complete than the timeout value of this resource allows. If there is more CPU time, this will not lead to problems. The monitor command will be able to complete and report a valid OpState again. Check the system log to see if the command was killed cause of the timeout. If commands are killed during normal operation, check and adjust the timeout values in your resource definition.
3. The monitor must clearly identify the process or application it is responsible for. If the monitor, for instance looks into the process table for a particular process, this process must be uniquely identified. If the monitor monitors another process than automation actually controls, the behavior of the whole resource would be somehow unpredictable.
4. If you try to automate base system services like printer spooler or mail transport agent, take care this services are only started/stopped by the automation and not by the system runlevel or init process.
5. If you have a more sophisticated monitor which could not only determine if a process exists but also if the application is able to deliver service, there is no possibility to report this fact to the automation engine and trigger a stop or shutdown of this application. If the monitor finds out that the process or application is available but hang or stuck, the monitor itself should kill this process/application. Automation will recognize the application is not available anymore (next monitoring) and will try to restart the application in place or move it to another node.
6. If you try to remove an online resource of class IBM.Application (e.g. with rmrsrc), the command will be rejected. You can force removal by setting Force=1. For example, to remove a resource called WebServer you would enter:

```
rmrsrc -s "Name=='WebServer' && ResourceType==1" IBM.Application Force=1
```

### How the resource manager handles return codes of the StartCommand, StopCommand, and MonitorCommand

The resource manager handles the return codes for the StartCommand, StopCommand, and MonitorCommand as follows:

#### **StartCommand:**

1. If the StartCommand was able to start the resource it should return a value of 0 to indicate that the resource was properly started and should go online within the next few seconds.
2. If the StartCommand was not able to start the resource it should return a value other than 0. This signals the automation not to start the resource again and set the resource operational state to Failed Offline. This indicates that you have to manually intervene to fix the resource. When then the resource is able to start, reset the Failed Offline operational state with the **resetrsrc** command. Note that whenever a StartCommand failed, automation will issue a StopCommand to ensure application leftovers of the failing start are removed.
3. If the StartCommand completes successfully and returns a value of 0, but the MonitorCommand of the resource does not report an online state after a certain amount of time (depending on the settings in the automation control configuration), the automation will try to restart the resource. There will be a total of 3 attempts, and if the resource does not go online after the 3rd attempt a StopCommand is issued and the resource is set to Failed Offline.
4. If the StartCommand of a resource is not able to complete within the time specified in the StartCommandTimeout attribute, the resource manager will kill the StartCommand and treat the start like a failing start command as described under item 2.
5. In case the StartCommand was valid when the resource was defined, but is later removed or not present (e.g. missing NFS mount), the start procedure is treated like a failing StartCommand as described under item 2.

#### **StopCommand:**

1. If the StopCommand was able to stop the resource it should return a value of 0 to indicate that the resource was properly stopped and should go offline within the next few seconds.

## Global resource RM

2. There is no mechanism to handle a failed StopCommand in IBM Tivoli System Automation for Multiplatforms. The StopCommand may indicate a failing stop of the application by returning a value other than 0, but this will not result in an automation action. There will be no retry of the StopCommand.
3. If the StopCommand of a resource is not able to complete within the StopCommandTimeout, the resource manager will kill the command and treat the stop like a failing StopCommand as described under item 2.
4. In case the stop command was valid when the resource was defined, but is later removed or not present (e.g. missing NFS mount), the stop procedure is treated like a failing StopCommand as described under item 2.

### **MonitorCommand:**

1. If the MonitorCommand was able to determine the operational state of a resource it should return one of the valid RMC operational states (see page 134). Keep in mind that in this case 0 is not the return value for RMC operational state Online, but the return value for operational state Unknown, which is the most critical state for the automation. A resource with an unknown operational state will not be automated anymore, and this may also affect other resources which have dependencies to this resource.
2. If the monitor command of a resource is not able to complete within the MonitorCommandTimeout, the resource manager will kill the MonitorCommand and set the RMC operational state to Unknown, which indicates a major problem with the resource. There will be no automation with this resource, until the MonitorCommand returns an operational state other than Unknown.
3. In case the monitor command was valid when the resource was defined, but later is removed or not present (e.g. missing NFS mount), the operational state is set to Unknown, which indicates a major problem with the resource.
4. In both cases the MonitorCommand may continue to report valid RMC operational states after system load has decreased or NFS is present again, and now automation will continue automating the resource.

### **Example: Implement the lpd printer spooler as an IBM.Application resource**

The following example shows how to prepare the lpd printer spooler on a SUSE based Linux system to be managed by IBM Tivoli System Automation.

1. Remove the lpd from the default runlevel of the system. If you want to run this resource as a floating resource on more than one node, you have to check the runlevel on each node.
2. For the start and stop command of the IBM.Application use the default init scripts shipped with your lp daemon:

```
StartCommand: /etc/init.d/lp start
StopCommand: /etc/init.d/lp stop
```

3. For the monitor command we use a simple shell script which checks for the lpd process in the process table:

```
File: /root/lpmon
#!/bin/bash
```

```
OPSTATE_ONLINE=1
OPSTATE_OFFLINE=2
```

```
ps -ax | grep -v "grep" | grep "/usr/sbin/lpd" > /dev/null
if [$? == 0]
then
 exit $OPSTATE_ONLINE
else
 exit $OPSTATE_OFFLINE
fi
```

Alternatively you can use the **pidmon** command shipped with IBM Tivoli System Automation. It basically searches the process table for a given command string. If the command string was found, The RMC OpState is returned. See “pidmon” on page 220 for a detailed description of this command.

```
MonitorCommand: /root/lpmon
```

or

```
MonitorCommand: /usr/sbin/rsct/bin/pidmon '/usr/sbin/lpd'
```

4. In case of a floating resource make sure that all nodes can access the start/stop and monitor command under the same path. Since the lpd is a small and simple application, the default Start-/Stop- and MonitorCommandTimeout values (default is 5 seconds) can be used. . In order to start lpd via the init scripts provide root as the user name for the IBM.Application.

Now the IBM.Application resource can be defined using the **mkrsrc** command:

```
mkrsrc IBM.Application \
 Name = "line_printer_daemon" \
 ResourceType = 1 \
 StartCommand = "/etc/init.d/lpd start" \
 StopCommand = "/etc/init.d/lpd stop" \
 MonitorCommand = "/usr/sbin/rsct/bin/pidmon '/usr/sbin/lpd' " \
 MonitorCommandPeriod = 15 \
 MonitorCommandTimeout = 5 \
 StartCommandTimeout = 5 \
 StopCommandTimeout = 5 \
 UserName = "root" \
 RunCommandsSync = 1 \
 ProtectionMode = 0 \
 NodeNameList = "'node01','node02'"
```

This command results in three resources being created: An aggregate resource named "line\_printer\_daemon" which can potentially be brought online on nodes "node01" and "node02" and two constituent resources also named "line\_printer\_daemon", one on node "node01" and the other on node "node02". If a start request is issued against the aggregate resource, then the Global Resource RM chooses one of the constituents and starts it with the script (or command) specified with the StartCommand attribute.

## What is the IBM.ServiceIP resource class?

The IBM.ServiceIP resource class is used to manage IP addresses that can be started, stopped and moved between adapters and nodes within a peer domain. Each resource of this class identifies one IP address. These IP addresses will typically be provided to clients that are connecting to some service that is running within the domain. A recovery management application can be used to keep the service and its associated IP address active, even through failures within the domain.

A Service IP address is an aggregate resource with one constituent resource per node where the administrator wants to allow that resource to be potentially brought on line. It is a floating aggregate resource since it can only have one constituent active at a time.

The IBM.ServiceIP resource class uses the following basic parameters:

1. The name of the resource.
2. The nodes where the resource can run.
3. The IP address which can be moved.
4. The netmask of the IP address.

The broadcast address and flag for the network interface will be taken from the parent interface where the ServiceIP resource is aliased on during the start.

Keep in mind that IBM.ServiceIP will generate a static routing entry for the network the IBM.ServiceIP is in. Take care that this network/route will not destroy the network configuration of the device the ServiceIP is aliased on.

## Global resource RM

Also automation does not take care of dynamic routing. If you specify a ServiceIP which is not in the subnet of the parent network interface, this physical device will host two different networks. Be sure to properly set up routing outside the automation cluster in order to support this network.

### Characteristics of IBM.ServiceIP

The IBM.Service IP class has two different flavours:

#### 1. IBM.ServiceIP automatically chooses a suitable network interface

If the IBM.ServiceIP receives a start request, the resource manager tries to choose a suitable network interface. If it can find one, the service IP is aliased on the that interface. This interface is called the supporting resource or supporting network interface.

In order to determine a suitable network interface, the resource manager compares the IPAddress attribute of the IBM.ServiceIP resource to the IP address of all existing network interfaces. If there is a suitable network (subnet) match found, the resource manager assigns the alias address to that network interface. If no matching subnet is found, the online request fails.

In case of the automatic interface choosing algorithm automation has no chance to evaluate the actual state of the network interface. As long as the network interface is configured and running, the ServiceIP can be assigned to the network interface. Keep in mind that UNIX/Linux do not change the status of a network interface, for example if the network cable is unplugged. Even if the device driver is able to detect the missing cable, the interface will stay configured and running.

If you want to exploit RSCT heartbeat mechanism to detect a network interface failure like a missing cable, use the supporting resource setup of the IBM.ServiceIP described in the next section.

Example

An IBM.ServiceIP is defined as:

```
IPAddress=192.168.1.5
NetMask="255.255.255.0"
NodeNameList="{ 'node01', 'node02' }"
```

The following network interfaces are available in the cluster:

1. IPAddress=192.168.1.1  
Netmask=255.255.255.0  
NodeNameList="{ 'node01' }"
2. IPAddress=9.152.172.91  
Netmask=255.255.255.0  
NodeNameList="{ 'node02' }"
3. IPAddress=192.168.2.1  
Netmask=255.255.255.0  
NodeNameList="{ 'node03' }"

Only interface Number 1 is able to hold the service IP. All other interfaces do not match the address (subnet) of the service IP. Calls to start the IBM.ServiceIP on other nodes than node01 will fail, because there is no suitable supporting network interface.

#### 2. IBM.ServiceIP receives a supporting resource to alias the service IP address on

In this case there are no limitations for the IPAddress and the NetMask attribute of the IBM.ServiceIP resource. The service IP must have a DependsOn relationship (see “DependsOn relationship” on page 62) to an equivalency of network interfaces (see Chapter 6, “Using equivalencies,” on page 49 for supporting resources). If IBM Tivoli System Automation decides to bring a service IP resource on a particular node online, it picks a suitable network interface from the equivalency of network interfaces and provides it to the IBM.ServiceIP as supporting resource. This is the interface hosting the alias.

In this configuration automation is able to monitor the operational state of the network interface the service IP address is aliased on. In case of an interface failure detected by RSCT heartbeat, automation will force down the dependency chain and stop the ServiceIP. If there is another online network interface in the equivalency, automation will choose this device to assign the ServiceIP on.

Example:

An IBM.ServiceIP is defined as:

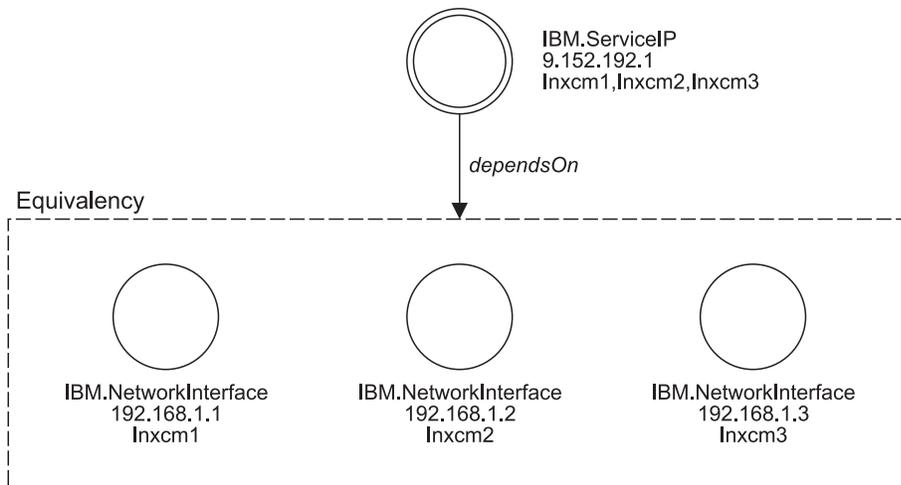
```
IPAddress=9.152.192.1
NetMask="255.255.255.0"
NodeNameList="{ 'node01', 'node02', 'node03' }"
```

The following network interfaces are available in the cluster:

1. IPAddress=192.168.1.1  
Netmask=255.255.255.0  
NodeNameList="{ 'node01' }"
2. IPAddress=192.168.1.2  
Netmask=255.255.255.0  
NodeNameList="{ 'node02' }"
3. IPAddress=192.168.1.3  
Netmask=255.255.255.0  
NodeNameList="{ 'node03' }"

All three network interfaces form an equivalency.

The following picture illustrates the setup:



If IBM Tivoli System Automation decides to start the service IP, it will pick a network interface from the equivalency. In this example the service IP can float between node01, node02 and node03, because all three nodes have a network interface in the equivalency which can act as supporting resource.

### Attributes used by IBM.ServiceIP

This section describes the attributes that are used by resources of the IBM.ServiceIP resource class.

When a resource of this class is created with the RMC command **mkrsrc**, it must have these attributes:

- Name
- IPAddress

Resources of this class may have these attributes:

- NodeNameList
- ResourceType
- NetMask
- ProtectionMode

Resources of this class have the following dynamic attribute:

- OpState

## Global resource RM

### Name attribute

The Name persistent attribute is a user defined name for this service IP address (e.g. mail-server-ip). Both the aggregate and constituent resource will have the same value for this attribute.

A value for this attribute must be specified when a new IBM.ServiceIP resource is created and it must be unique.

The attribute must be of type character string.

### NodeNameList attribute

The NodeNameList persistent attribute is an array of strings that indicates which nodes the IBM.ServiceIP resource is available on.

If the resource is floating, the Global Resource RM will ensure that there is one constituent (fixed) resource for each node name in this list. Constituent resources are implicitly created or deleted as necessary to match the entries in this list. Constituent resources will only contain one entry in this array since they are fixed resources and thus only available on one node. This attribute for a floating resource is implicitly modified if a constituent resource is explicitly removed by the administrator so that the aggregate and constituent resource relationship is always consistent.

This list may be empty for a floating resource which means that it is not available anywhere and constituents may be added separately.

The list may contain at most one name if the resource is fixed (i.e. ResourceType=0). If no name is given for a fixed resource, then RMC will provide a default because the fixed resource is tied to a node and therefore cannot be created without a node name or node id.

The value of this attribute for aggregate resources may be changed with the chsrc command. An attempt to modify this attribute for a constituent resource will generate an error.

### ResourceType attribute

You use the ResourceType persistent attribute to identify whether the resource is fixed or floating. An integer value of 0 indicates that the resource is fixed, a value of 1 indicates that it is a floating resource. This attribute defaults to floating if not specified when a new IBM.ServiceIP resource is created.

### IPAddress attribute

With the IPAddress persistent attribute you specify the IP address that will be aliased onto a network interface where the resource is brought online. This attribute is required when a new IBM.ServiceIP resource is created. The IP address must be given in 'dotted decimal' notation as a character string, e.g. 9.152.80.251.

### NetMask attribute

You use the NetMask persistent attribute to specify the netmask that will be assigned to the IP address defined in the IP address attribute. The attribute must be given as a character string, for example 255.255.255.0.

### ProtectionMode attribute

The ProtectionMode persistent attribute specifies whether the resource is critical. If it is critical, then the IBM.ConfigRM decides if the resource can be started as requested. (For further detail on this behavior, see Chapter 9, "Protecting your resources – quorum support," on page 95.

The attribute may have the integer values 0 (Non-Critical) or 1 (Critical), it defaults to Critical.

### OpState attribute

The value of this dynamic state attribute contains the operational state of the resource as determined by the resource manager. Typical values for this state are Online (value is 1) and Offline (value is 2) meaning that the IP address is either operational or not operational.

**Example 1: Define an IP address as an IBM.ServiceIP resource**

In order to define an IP address which has the address IP 9.152.172.11, netmask 255.255.255.0 and potentially runs on the nodes node05 and node06, issue the following RMC command:

```
mkrsrc IBM.ServiceIP
 Name="WebServerIP"
 NodeNameList="{ 'node05', 'node06' }"
 IPAddress=9.152.172.11
 NetMask=255.255.255.0
```

**Example 2: Define an IP address as an IBM.ServiceIP resource and use an IBM.Equivalency of network interfaces**

As shown in the preceding example, define an IP address which has the address IP 9.152.172.11, netmask 255.255.255.0 and potentially runs on the nodes node05 and node06:

```
mkrsrc IBM.ServiceIP
 Name="WebServerIP"
 NodeNameList="{ 'node05', 'node06' }"
 IPAddress=9.152.172.11
 NetMask=255.255.255.0
```

The nodes node05 and node06 have each more than one network interface. To form an equivalency containing eth1 device of node node05 and node node06, type:

```
mkequ MyInterfaces IBM.NetworkInterface:eth1:node05,:eth1:node06
```

Now you can connect the ServiceIP with the equivalency:

```
mkrel -p dependson -S IBM.ServiceIP:WebServerIP -G IBM.Equivalency:MyInterfaces WebIp_depon_MyInterfaces
```

### Using the Test Resource Manager

This section describes the characteristics of the Test resource manager.

The IBM Test resource manager (IBM.TestRM) manages test resources and provides functions to manipulate the operational state of these resources. The resource manager is operational in a peer domain mode only and provides the resource class IBM.Test. IBM.TestRM does not control real resources.

### What is the IBM.Test resource class?

The IBM.Test resource class allows new types of fixed and floating resources to be created, monitored and controlled through the RMC subsystem. These resources are no real resources but just the containers to define, monitor and control them. These resources can then be automated or recovered by IBM Tivoli System Automation. The purpose of the IBM.Test class is to provide a lightweight and easy to handle resource to simulate automation scenarios without the overhead of real resources. Each resource controlled by the IBM.Test class is considered to be a globalized resource which divides in one aggregate and one constituent on every node the resource is defined. See Figure 14 on page 130 for details.

The IBM.Test resource class provides a set of persistent resource attributes to simulate the behavior of real resources.

### Attributes used by IBM.Test

This section describes the persistent attributes that are used by the IBM.Test resource class.

When a resource of this class is created with the RMC command **mkrsrc**, it must have the following persistent attribute

- Name

Resources of this class may have these attributes:

- NodeNameList
- ResourceType
- ForceOpState
- TimeToStart
- TimeToStop
- WriteToSyslog
- MoveTime
- MoveFail

Resources of this class have the following dynamic attributes:

- OpState
- MoveState
- OpQuorumState

#### Name attribute

The Name persistent attribute is a user-defined name for the test resource. Both the aggregate and constituent resource will have the same value for this attribute. A value for this attribute must be specified when a new IBM.Test resource is created and it must be unique.

#### NodeNameList attribute

The NodeNameList persistent attribute is an array of strings that indicates which nodes the IBM.Test resource is available on.

If the resource is floating, the TestRM will ensure that there is one constituent (fixed) resource for each node name in this list. Constituent resources are implicitly created or

deleted as necessary to match the entries in this list. Constituent resources will only contain one entry in this array since they are fixed resources and thus only available on one node. This attribute for a floating resource is implicitly modified if a constituent resource is explicitly removed by the administrator so that the aggregate and constituent resource relationship is always consistent.

This list may be empty for a floating resource which means that it is not available anywhere and constituents may be added separately.

#### ResourceType attribute

You use the ResourceType persistent attribute to identify whether the resource is fixed or floating. An integer value of 0 indicates that the resource is fixed, a value of 1 indicates that it is a floating resource. This attribute defaults to fixed if not specified when a new IBM.Test resource is created.

#### ForceOpState attribute

You use this attribute to initiate an OpState change of the test resource via the RMC **chsrc** command. This could be used to simulate a failure in the resource. The last state change is saved in this persistent resource attribute. Specifying this attribute during the creation of the resource has no effect. Normally ForceOpState changes should be done on constituent resources, as the aggregate resource collects the OpState of the whole resource. Allowed values of this attribute are:

- Unknown=0
- Online=1
- Offline=2
- Failed Offline=3
- Stuck Online=4
- Pending Online=5
- Pending Offline=6

**TimeToStart** After a test resource receives the start command, the TimeToStart attribute specifies the amount of time (in seconds) it takes a resource to change its OpState from pending online to online. The default value is 0 seconds, then the resource immediately goes online.

**TimeToStop** After a test resource receives the stop command, the TimeToStop attribute specifies the amount of time (in seconds) it takes a resource to change its OpState from pending offline to offline. The default value is 0 seconds, then the resource immediately goes offline.

#### WriteToSyslog

A resource of the class IBM.Test is capable to log online, offline and ForceOpState events in the Linux syslog facility. You use the WriteToSyslog attribute to turn on/off the writing to the syslog daemon. Allowed values of this attribute are:

- 0** Do not write to syslog (this is default)
- 1** Write to syslog

#### OpState attribute

The value of this dynamic state attribute contains the operational state of the resource. IBM.Test resource OpState follows the RMC start/stop commands or the ForceOpState event from an operator or test script (automated testcase). The possible values for this attribute as defined by the state transition diagram in the RMC architecture are:

- Unknown=0
- Online=1
- Offline=2
- Failed Offline=3
- Stuck Online=4
- Pending Online=5

## Test resource manager

Pending Offline=6

**MoveTime** Reserved for internal use.

**MoveFail** Reserved for internal use.

**MoveState** Reserved for internal use.

**OpQuorumState**  
Reserved for internal use.

## Example: Create a test resource and manipulate its OpState

In order to create an IBM.Test resource on 2 nodes, issue the following RMC command:

```
mkrsrc IBM.Test \
 Name="mytest" \
 NodeNameList="{ 'node01', 'node02' }" \
 ResourceType=1 \
 TimeToStart=5 \
 TimeToStop=2 \
 WriteToSyslog=1
```

The following command causes a constituent on node02 to change its OpState to Failed Offline. If the resource is automated by IBM Tivoli System Automation, the automation manager starts the resource on another node.

```
chrsrc -s "Name='myTest' && NodeNameList='{ 'node02' }" IBM.Test ForceOpState=3
```

---

## Chapter 13. IBM Tivoli System Automation commands

These are the IBM Tivoli System Automation commands described here:

| Command   | Description                                                                                               | For details, see page: |
|-----------|-----------------------------------------------------------------------------------------------------------|------------------------|
| addrgmbr  | Add one or more resources to a resource group                                                             | 147                    |
| chequ     | Change a resource equivalency                                                                             | 151                    |
| chrel     | Change one or more managed relationships between resources                                                | 154                    |
| chrg      | Change persistent attribute values of a resource group (including starting and stopping a resource group) | 159                    |
| chrgmbr   | Change persistent attribute value of a managed resource in a resource group                               | 162                    |
| lsequ     | List equivalencies and their attributes.                                                                  | 165                    |
| lsrel     | List managed relationships                                                                                | 168                    |
| lsrg      | List a resource group or its resource member                                                              | 173                    |
| lsrgreq   | Lists outstanding requests applied against resource groups or managed resources                           | 178                    |
| lssamctrl | Lists IBM Tivoli System Automation controls                                                               | 181                    |
| mkequ     | Make an equivalency resource                                                                              | 183                    |
| mkrel     | Make a managed relationship between resources                                                             | 187                    |
| mkrq      | Make a resource group                                                                                     | 191                    |
| rgmbrreq  | Requests a managed resource to be started, stopped, or cancelled                                          | 194                    |
| rgreq     | Requests a resource group to be started, stopped, cancelled, or moved                                     | 197                    |
| rmequ     | Remove one or more resource equivalencies                                                                 | 200                    |
| rmrel     | Remove a managed relationship between resources                                                           | 202                    |
| rmrg      | Remove a resource group                                                                                   | 206                    |
| rmrgmbr   | Remove one or more resources from the resource group                                                      | 208                    |
| samcfg    | Save and restore the IBM Tivoli System Automation configuration.                                          | 211                    |
| samctrl   | Set the IBM Tivoli System Automation control parameters                                                   | 213                    |
| samdiag   | Gather snap-shot information on a resource                                                                | 216                    |
| samlicm   | Install, list and upgrade product license                                                                 | 219                    |

---

### RSCT RMC commands and files

You can find the RSCT RMC commands and files in the *IBM Reliable Scalable Cluster Technology for Linux, Technical Reference* manual, SA22-7893.

---

### Setting the environment variables

#### CT\_CONTACT

When the CT\_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

## CT\_LOCAL\_SCOPE

The CT\_LOCAL\_SCOPE environment variable sets the RMC subsystem scope to the stand-alone environment. By default, the RMC subsystem monitors and controls the resources and resource classes on all nodes in the cluster. A cluster can consist of one or more nodes. When a cluster consists of one node, it is said to operate in a stand-alone environment. The scope of a command in the stand-alone environment is the local node. The scope of a command in a cluster environment of more than one node is the entire cluster. For a cluster of more than one node, to return or change resources or resource classes only on the node where RMC is running, set the CT\_LOCAL\_SCOPE environment variable to 1, as follows:

```
export CT_LOCAL_SCOPE=1
```

For example, assume that a cluster consists of nodes A, B, and C and the following command is run:

```
lsrsrc IBM.FileSystem
```

Resources for the file systems associated with nodes A, B, and C are returned.

If CT\_LOCAL\_SCOPE is set to 1 and the **addrgmbr** command is run on Node A, then only the file system resources associated with node A are returned.

---

## addrgmbr

### Name

**addrgmbr** – Adds one or more resources to a resource group

### Synopsis

**addrgmbr** [-h] [ -f *data\_input\_file*] [-T] [-V]

**addrgmbr** [-h] -s [-m T | F] [-p A | O] [-T] [-V] -g *Resource\_group* *Resource\_class:"selection\_string"* [*Resource\_class:"selection\_string"* [...]]

**addrgmbr** [-h] [-m T | F] [-p A | O] [-T] [-V] -g *Resource\_group* *Resource\_class:Resource\_name[:Node][,Resource\_name[:Node][, ...]]* [*Resource\_class:Resource\_name[:Node][,Resource\_name[:Node][, ...]] [...]]*

### Description

The **addrgmbr** command adds one or more new resources to a resource group. If a *data\_input\_file* and the -f option is not specified, then either a selection string or a resource name must be specified when this command is used. A resource name or selection string must be specified with the resource class to which it belongs. A fixed resource must include the resource class and the node name at which it resides. A member resource cannot be included in more than one resource group at the same time. A member resource cannot be in a resource group and in an equivalency at the same time. Each member of the resource group must exist as an RMC resource before being added to a resource group. A resource group member can only be defined on an online node in a cluster.

You can only add multiple resources to a resource group if they are offline and if the nominal state of the resource group is offline. If the OpState of a resource to be added is online, the addition of that resource would cause an immediate stop of that resource. If the resource is in the process of stopping, then the addition of another resource would be rejected.

Member resources and their attribute flags can be specified on the command line as the syntax indicates or from a file when used with the -f option. The value must be the same data type that is defined for this resource attribute. Use the **lsrsrcdef** command to verify the data type and attribute field for each attribute.

### Options

-f *data\_input\_file*

Name of the file which contains member resource attribute information.

-g *Resource\_group*

The unique name of the resource group, to which the member resources are to be added. This represents the MemberOf attribute of the Managed Resource.

-h Help. Writes the command's usage statement to standard output.

-m T|F

Mandatory Attribute. Specifies whether this managed resource is required in the resource group.

T TRUE (the Default). These managed resources are required by the resource group.

F FALSE. These managed resources are not required by the resource group.

-p A|O

Specifies the method to be used when selecting a node for placing the resource on. The value can be:

A Any.

O Ordered. This is the default.

## addrgmbr

- s Specifies that a selection string will be used instead of the resource name.
- T Writes the command's trace messages to standard error. For your software-service organization's use only.
- V Writes the command's verbose messages to standard output.

## Parameters

*Resource\_class:"selection\_string"*

Specifies the selection string. The **-s** option determines this parameter. Each selection string must be preceded by a resource class. The specified selection string is applied to its corresponding *Resource\_class* attributes to determine which resources are to be made members of the *Resource\_group*. The *selection\_string* and its resource class must be separated by the colon or " : " delimiter.

The selection string must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s IBM.Application:'Name == "testing"'
-s IBM.Application:'Name ?= "test"'
-s IBM.Application:'Name like "%"' (For all resources)
```

For information on how to specify selection strings, see “Using expressions” on page 222.

*Resource\_class:Resource\_name[:Node]*

Member Resources. This specifies one or more member resources that are to be acted on. Resources belonging to different resource classes can also be specified using this syntax. However, member resources belonging to different classes must be separated by spaces. A member resource is identified by the resource's class ( *Resource\_class*), the resource's name ( *Resource\_name*), and optionally the node (*Node*) the resource is on. The *Resource\_class*, *Resource* and *Node* must be separated by a colon. *Resource\_class* is the name of the resource class the member resource belongs to. *Resource* is the name of the actual member resource in its class. The node may have to be included when trying to act on a fixed member resource. The node and its resource must be separated by the colon or ":" delimiter. Multiple resources belonging to the same resource class can also be specified and are separated by a comma: *Resource\_class:Resource[:Node][,Resource[:Node][,...]]*

The member resources must exist in the resource group for the requested action to take place.

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To create member resources using the data input file, do the following:
  - a. To generate a template to aid in the defining of these resources in a file, enter the command:
 

```
lsrsrdef -i IBM.ManagedResources >/tmp/MgdR.rdef
```
  - b. Edit the file */tmp/MgdR.rdef* with your preferred file editor: (Enter values for all of the attributes, substituting an appropriate value for the type, or leave it blank if you want the default value.)
  - c. Run the **addrgmbr** command with the file as follows:
 

```
addrgmbr -f /tmp/MgdR.rdef
```

### Sample Data Input file for addrgmbr:

You can enter values for all of the attributes and thus change the value shown below, or remove the line with the attribute if you want the default value.

#### PersistentResourceAttributes:

##### resource 1:

```
ManagedResource = "0x001b 0xffff 0x35c05b13 0x00000000 0x9a6ee5e0 0x0de8934c"
MemberOf = "charmC"
```

##### resource 2:

```
ManagedResource = IBM.Application:Res1:node1
MemberOf = "charmC"
Mandatory = false
```

2. To add a member resource **tester** belonging to resource class IBM.Application, to a resource group **foo**, enter:
 

```
addrgmbr -g foo IBM.Application:tester
```
3. To add member resources **tester**, **Jfoo**, and **Dfoo**, belonging to resource class IBM.Application, to a resource group **foo**, enter:
 

```
addrgmbr -g foo IBM.Application:tester,Jfoo,Dfoo
```
4. To add member resources **tester** and **Jfoo** belonging to resource class IBM.Application, and fixed resource **en0** of class IBM.ServiceIP, residing at node number 1, to a resource group **foo**, enter:
 

```
addrgmbr -g foo IBM.Application:tester,Jfoo IBM.ServiceIP:en0:1
```
5. To add selected member resources of the resource class IBM.Application, to a resource group **foo** as a mandatory resource, enter:
 

```
addrgmbr -m T -g foo IBM.Application:jfoo,foo
```

Sample data input file.

```
PersistentResourceAttributes::
resource 1:
ManagedResource = "0x001b 0xffff 0x35c05b13 0x00000000
0x9a6ee5e0 0x0de8934c"
MemberOf = "charmC"
resource 2:
ManagedResource = "0x001b 0xffff 0x35c05b13 0x00000000
0xb4d7e1b0 0x0def4ba2"
MemberOf = "charmC"
Mandatory = 0
NominalState = 0
```

## Files

**/usr/sbin/rsct/bin/addrgmbr** Location of the **addrgmbr** command.

*data\_input\_file*

**addrgmbr**

## **See Also**

The **samctrl**, **chrel**, **chrg**, **chrgmbr**, **lsrg**, **mkrg**, **mkrel**, **rmrel**, **rmrg**, **rmrgmbr** commands.

The **rmcli** General Information file.

The **Resource\_Data\_Input** file

---

## chequ

### Name

**chequ** – Changes an already-defined resource equivalency.

### Synopsis

**chequ** [-h] [-i] -u a l d l r [-p A l O ] [-m *Minimum\_Necessary*] [-T] [-V] -S “*Static\_select\_string*”  
*Equivalency Resource\_class*

**chequ** [-h] [-i] -u r [-p A l O ] [-m *Minimum\_Necessary*] [-T] [-V] -D “*Dynamic\_select\_string*”  
*Equivalency Resource\_class*

**chequ** [-h] [-i] [-u a l d l r ] [-p A l O ] [-m *Minimum\_Necessary*]  
[-c *New\_Equiv*] [-T] [-V] *Equivalency* [*Resource\_class:Resource\_name[:Node]*]  
[,*Resource\_name[:Node][,...]*]

### Description

The **chequ** command changes a resource equivalency. Resources can be added, removed, or totally replaced in an equivalency. All resources in an equivalency must be from the same resource class. Even the name of the equivalency can be changed with this command.

Equivalencies and their attribute flags can be specified on the command line as the syntax indicates. The value must be the same data type that is defined for the resource attributes. Use the **lsrsrdef** command to verify the data type and attribute field for each attribute.

### Options

#### -u a l d l r

Update Equivalency resources.

**a** Adds the specified resources to the specified equivalency.

**d** Deletes the specified resources from the specified equivalency.

**r** Replace/Overwrite Equivalency resources. Replaces the equivalency resources with the specified equivalency resources.

#### -c *New\_Equiv*

Specifies a new name that replaces the current *Equivalency* parameter.

#### -D “*Dynamic\_selection\_string*”

The specified selection string will be saved as the *SelectString* attribute for the concerned equivalency. This *dynamic\_selection\_string* is then applied to all resources of the specified resource class by the Recovery Manager to dynamically determine what members are to be included in the specified equivalency. This option must be used with the **r** option and can also not be combined with the *Resource\_name* [: *Node*] operand. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-D 'Name == "testing"'
```

```
-D 'Name ?= "test"'
```

-D cannot be combined with -S.

**-h** Help. Writes the command’s usage statement to standard output.

**-i** Interactive Prompt. Prompt before changing equivalency.

## chequ

### -p A | O

Specifies the equivalency select-from policy. The value can be:

- A** Any
- O** Ordered

The default value is **Any**.

### -m *Minimum\_Necessary*

Minimum Necessary Equivalency. Specifies the minimum necessary members to make an equivalency valid.

### -S "*Static\_selection\_string*"

The specified selection string will be applied to the specified resource class to determine the resources that make up the equivalency. These resources will then be saved as the MemberShip attribute for the concerned equivalency. This option cannot be combined with the *Resource\_name* [: *Node*] operand. Resources can be added, deleted and replaced when using this string. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-S 'Name == "testing"'
```

```
-S 'Name ?= "test"'
```

-S cannot be combined with -D.

For information on how to specify selection strings, see "Using expressions" on page 222.

- T** Writes the command's trace messages to standard error. For your software-service organization's use only.
- V** Writes the command's verbose messages to standard output.

## Parameters

### *Equivalency*

Specifies the unique name of the already defined equivalency to be changed.

### *Resource\_class:Resource\_name[:Node]*

Specifies one or more resources that will be added to or deleted from the specified equivalency. All of the resources must be from the same resource class. An equivalency resource is identified by the resource class (*Resource\_class*) and the resource name (*Resource\_name*), and, optionally, the node (*Node*) at which the resource resides. *Resource\_class*, *Resource\_Name*, and *Node* must be separated by a colon. *Resource\_class* is the name of the Resource class the equivalency resource belongs to. The first resource or *Resource\_name[:Node]* must be preceded by the *Resource\_class* it belongs to and it must be separated by the colon (:) delimiter. Multiple resources can be specified and are separated by a comma. Since all resources must be from the same class, the *Resource\_class* is to be specified only once as the syntax indicates.

## Exit Status

- 0** Command has run successfully.
- 1** Error occurred with RMC.
- 2** Error occurred with CLI script.
- 3** Incorrect flag on command line.
- 4** Incorrect parameter on command line.
- 5** Error occurred with RMC that was based on faulty command line input.

- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To add another resource **tester** that belongs to resource class IBM.Application to an equivalency called **foo**. type:  

```
chequ -u a foo IBM.Application:tester
```
2. To delete a resource **tester1** that belongs to resource class IBM.Application from an equivalency called **test**, enter:  

```
chequ -u d test IBM.Application:tester1
```
3. To add an equivalency called **Jfoo** with selected members of the resource class IBM.Application by using a select string, enter:  

```
chequ -u a -S "ResourceType==0" Jfoo IBM.Application
```
4. To overwrite an equivalency called **Jfoo** with selected members of the resource class IBM.Application by using a dynamic select string, enter:  

```
chequ -u r -D "ResourceType==0" Jfoo IBM.Application
```

## Files

**/usr/sbin/rsct/bin/chequ** Location of the **chequ** command.

## See Also

The **samctrl**, **lsequ**, **mkequ**, **rmequ** commands.

The **rmcli** General Information file

---

## chrel

### Name

**chrel** – Changes one or more managed relationships between resources.

### Synopsis

```
chrel [-h] [-i] [-u a l d | r] -s "Selection_string" [-c relation_name] [-p relationship] [-o condition] [-W
New_Target_Class[:Resource_name[:Node][,New_Target_Class[:Resource_name[:Node][,...]]]] [-w
"New_Target_string"] [-T] [-V]
```

```
chrel [-h] [-i] [-u a l d | r] [-c relation_name] [-p relationship] [-o condition] [-W
New_Target_Class[:Resource_name[:Node][,New_Target_Class[:Resource_name[:Node][,...]]]] [-w
"New_Target_string"] [-T] [-V] Managed_Relation
```

```
chrel [-h] [-i] [-u a l d | r] [-c relation_name] [-p relationship] [-o condition]
[-W New_Target_Class[:Resource_name[:Node][,New_Target_Class
[:Resource_name[:Node][,...]]]] [-w "New_Target_string"] -S Source_Class
[:Resource_name[:Node]] [-s "Source_string"] [-G Target_Class[:Resource_name[:Node]
,Target_Class[:Resource_name[:Node][,...]]]] [-g "Target_string"]
[-P Current_relationship] [-O Current_condition] [-T] [-V]
```

### Description

The **chrel** command modifies a managed relationship between resources. The managed relationship is between a source resource and one or more target resources. The source resource must be a member of a resource group. A target resource does not have to be in a resource group. Managed relationships and their attributes can be specified on the command line as the syntax indicates. The attribute value must be of the same data type that is defined for the resource attribute. Use the **lsrsrdef** command to verify the data type and attribute field for each attribute.

### Options

#### -u a l d | r

Update targets.

- a** Add the specified new target resources to the relationship defined for the source resources.
- d** Delete the specified new target resources from the relationship defined for the source resources.
- r** Replaces (overwrites) the relationship defined for the source resource with the specified new target resources.

#### -o condition

Specifies the new condition to be used when a Location relationship (Collocated, AntiCollocated, Affinity, or AntiAffinity) is defined. The value can be the numeric value or the word (not case sensitive):

##### **0x0000 NoCondition**

Specifies unconditional location relationship.

##### **0x0001 or IfOnline**

Specifies that the target resource is online.

##### **0x0002 or IfOffline**

Specifies that the target resource is offline, failed offline, or unknown.

**0x0003 or IfNotOnline**

Specifies that the target resource is not online.

**0x0004 or IfNotOffline**

Specifies that the target resource is neither offline nor failed offline.

**-O condition**

Specifies the condition value to query all the defined relationship to find a match with an existing condition. The value can be the numeric value or the word (not case sensitive):

**0x0000 NoCondition**

Specifies unconditional location relationship.

**0x0001 or IfOnline**

Specifies that the target resource is online.

**0x0002 or IfOffline**

Specifies that the target resource is offline, failed offline, or unknown.

**0x0003 or IfNotOnline**

Specifies that the target resource is not online.

**0x0004 or IfNotOffline**

Specifies that the target resource is neither offline nor failed offline.

**-g Target\_string**

The *Target\_string* is applied to all resources of the resource class specified by the **-G** flag, to determine the targets of the relationship.

If the **-g** option is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for the **-G** option .

**-G Target\_class:[Resource\_name[:Node]]**

Specifies the target resources to add, delete or replace with the relationship. The target resource or resources will be used with the source and any other specified query to find a match from all of the defined relationships. A target resource is identified by the resource's class (*Target\_class*), the resource name (*Resource\_name*), and, optionally, the node (*Node*) at which the resource resides. *Target\_class*, *Resource\_name*, and *Node* must be separated by colons. Multiple target resources can be specified and are separated by commas. The colons are required, but omitting *Target\_class* or *Resource\_name* causes the previously specified class or name to be used when multiple target resources are specified.

If the **-g** option is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for the **-G** option.

**-h** Writes the command's usage statement to standard output.

**-i** Interactive. Prompt before changing relationships.

**-c relation\_name**

Specifies the new name for the specified managed relationship. This change-of-name can only be applied to *one* relationship. This means that if a name change is required, Queries and Selection strings should match with only one relationship.

**-p relationship**

Specifies the new managed relationship to apply. The value can be the numeric value or the word (not case sensitive):

**0x0001 or Collocated**

Specifies that the source and the target resources are to be located on the same node.

**0x0002 or AntiCollocated**

Specifies that the source and the target resources are to be located on different nodes.

**0x0003 or Affinity**

Specifies that the source resource and its specified affinity resource are to be located on the same node, if possible.

**0x0004 or AntiAffinity**

Specifies that the source resource and its specified affinity resource are to be located on different nodes, if possible.

**0x0005 or IsStartable**

Specifies that the source and its target resources are startable.

**0x0006 or StartAfter**

Specifies that the source is to be started after the target resources.

**0x0007 or DependsOn**

Specifies that the target resources must be online before the source resource is started.

- A DependsOn relationship also includes an implicit collocation (explained in “Collocated relationship” on page 71) between the source and target resources.
- If a target resource fails, the source resource will also be stopped.

**0x0008 or DependsOnAny**

Specifies that the target must be online before the source resource is started. It is identical to the DependsOn relationship except that it does not provide the collocated constraint for the start sequence. Therefore the source and target resources may not be started on the same node.

**0x0009 or StopAfter**

Specifies that the source resource may not be stopped until after the target resource has been brought offline.

**0x000A or ForcedDownBy**

Specifies that the source resource must be forced offline in the event that either the target resource goes offline unexpectedly or the target resource itself is forced offline.

**-P relationship**

Specifies the managed relationship value used to query all the defined relationships to find a match. The value can be a numeric value or the word (not case sensitive):

**0x0001 or Collocated**

Specifies that the source and the target resources are to be located on the same node.

**0x0002 or AntiCollocated**

Specifies that the source and the target resources are to be located on different nodes.

**0x0003 or Affinity**

Specifies that the source resource and its specified affinity resource are to be located on the same node, if possible.

**0x0004 or AntiAffinity**

Specifies that the source resource and its specified affinity resource are to be located on different nodes, if possible.

**0x0005 or IsStartable**

Specifies that the source and its target resources are startable.

**0x0006 or StartAfter**

Specifies that the source is to be started after the target resources.

**0x0007 or DependsOn**

Specifies that the target resources must be online before the source resource is started.

- A DependsOn relationship also includes an implicit collocation (explained in “Collocated relationship” on page 71) between the source and target resources.
- If a target resource fails, the source resource will also be stopped.

**0x0008 or DependsOnAny**

Specifies that the target must be online before the source resource is started. It is identical to the DependsOn relationship except that it does not provide the collocated constraint for the start sequence. Therefore the source and target resources may not be started on the same node.

**0x0009 or StopAfter**

Specifies that the source resource may not be stopped until after the target resource has been brought offline.

**0x000A or ForcedDownBy**

Specifies that the source resource must be forced offline in the event that either the target resource goes offline unexpectedly or the target resource itself is forced offline.

**-w** *New\_Target\_string*

Specifies the target selection string for the resources to add, delete or replace with the relationship. *New\_Target\_string* is applied to all resources of the resource class specified by the **-G** option to determine the targets of the relationship.

If the **-g** option is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for the **-G** option .

**-W** *New\_Target\_Class: Resource\_name[: Node]*

New Target resources. These resources will be added, deleted or replaced as required to or from the current list of target resources for the matching relationships. A target resource is identified by the resource's class (*Resource\_class*), the resource's name (*Resource\_name*), and optionally the node (*Node*) the resource is on. The *Resource\_class*, *Resource\_name* and *Node* must be separated by a colon. Multiple target resources can be specified and are separated by a comma. The colons are required but omitting the *Resource\_class* or *Resource\_name* causes the previously specified class or name to be used when specifying multiple target resources.

If the **-g** flag is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for **-G**.

**-s** *Selection\_string*

Specifies the source selection string of the relationship to change. *Selection\_string* is applied to all resources of the resource class specified by the **-S** option to determine the source of the relationship. The result of the selection may contain one or more resources.

If the **-s** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-S** option .

**-S** *Source\_class:[Resource\_name[:Node]]*

Specifies the source resource of the relationship to change. A source resource is identified by the resource class (*Resource\_class*) of the resource, the resource name (*Resource\_name*), and, optionally, the node (*Node*) at which the resource resides. The result of the selection may contain one or more resources.

If the **-s** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-S** option.

**-T** Writes the command's trace messages to standard error. For your software-service organization's use only.

**-V** Writes the command's verbose messages to standard output.

## Parameters

*Managed\_Relation*

Name of the managed relationship to be changed. Using the **-c** flag, even the name of the relationship can be modified.

## chrel

### Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

### Security

This command requires root authority.

### Examples

1. To add a relationship for a resource **tester** that belongs to resource class IBM.Application with target resources from resource class IBM.Application, whose ResourceType is set to zero, enter:  

```
chrel -u a -S IBM.Application:tester -W IBM.Application -w "ResourceType==0" -G IBM.Application -g "ResourceType==1"
```
2. To modify a relationship for a source resource **narten** that belongs to resource class IBM.Application to add resources **tr0** and **en0** of resource class IBM.NetworkInterface, enter:  

```
chrel -u a -S IBM.Application:narten -W IBM.NetworkInterface:tr0,:en0
```

### Files

`/usr/sbin/rsct/bin/chrel` Location of the **chrel** command.

### See Also

The **addrgmbbr**, **chequ**, **chrg**, **chrgmbr**, **lsrg**, **mkequ**, **mkrel**, **mkrg**, **rmequ**, **rmrel**, **rmrg**, **rmrgmbr** commands.

The **rmccli** General Information file

---

## chrg

### Name

**chrg** – Changes persistent attribute values of one or more resource groups (including starting and stopping resource groups).

### Synopsis

**chrg** [-h] [-i] [-u a | d | r] -s “selection\_string” [-l relationship] [-n node\_name | -e equiv\_name] [-o nominal\_state] [-p priority] [-T] [-V] [-x node1, ...noden]

**chrg** [-h] [-i] [-u a | d | r] [-c new\_name] [-l relationship] [-n node\_name | -e equiv\_name] [-o nominal\_state] [-p priority] [-T] [-V] [-x node1, ...noden] Resource\_group [ ... ]

### Description

The **chrg** command changes the persistent attribute values of one or more resource groups. The name of the resource group is specified by *Resource\_group*. The resource groups to be changed can also be determined using the selection string with the **-s** option. The name of a resource group can also be changed with this command using the **-c** option. IBM Tivoli System Automation will then update all the associated member resources with this change. To allow IBM Tivoli System Automation to monitor and control these resource groups, the resource group must be in an Online Nominal State.

### Options

**-c** *new\_name*

Specifies the new *Resource\_group* name. Renames the *Resource\_group* name with *new\_name*.

**-e** *equiv\_name*

Specifies the equivalency name of the nodes on which the *Resource\_group* can run. This option cannot be combined with the **-n** option. To set the resource group with the equivalency of all nodes in the cluster, specify *equiv\_name* as the string ALL.

**-h** Writes the command's usage statement to standard output.

**-i** Interactive. Prompt before changing resource groups.

**-u a | d | r**

Specifies what should be done with the specified node list. Note that these options require the **-x** option.

**a** Adds the specified nodes to the list of excluded nodes.

**d** Deletes the specified nodes from the list.

**r** Replaces the exclude list with the specified nodes.

**-x** *node1, ...noden*

Specifies the excluded node list. Only to be used in conjunction with **-u a | d | r**.

**-l** *relationship*

Specifies the location relationship among the members of the resource group. The location relationship value can be entered as the numeric value or as the word (not case sensitive):

**0x0000 None**

None. Specifies that member resources of the resource group can be on any node

**0x0001 or Collocated**

Specifies that member resources of the resource group are to be located on the same node.

## chrg

### **-n** *node\_name*

Specifies the node on which the *Resource\_group* can run. This option cannot be combined with the **-e** option .

### **-o** *nominal\_state*

Specifies the Nominal state of the resource group, which can be online or offline. Nominal state values can be entered as the numeric value or as the word (not case sensitive):

#### **0x0000 or Online**

Online. Specifies that the desired state of *Resource\_group* is online.

#### **0x0001 or Offline**

Offline. Specifies that the desired state of *Resource\_group* is offline.

### **-p** *priority*

Specifies the relative importance of this resource group compared to other resource groups. Priority can be any integer from 0 to 200: the higher the integer, the higher the priority. The default priority value is 0.

### **-s** *Selection\_string*

Specifies the source selection string of the persistent attribute values to change. *Selection\_string* is applied to all existing resource groups.

The selection string must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
-s 'Name like "%"' (For all resources)
```

For information on how to specify selection strings, see “Using expressions” on page 222.

**-T** Writes the command’s trace messages to standard error. For your software-service organization’s use only.

**-V** Writes the command’s verbose messages to standard output.

## Parameters

### *Resource\_group*

The unique name of the resource group to be changed. This resource group must already exist for this operation to succeed. Multiple resource groups can be specified but must be separated by spaces.

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To change the persistent-attribute location relationship to Collocated, Nominal state to Online, and the priority value to 34 of a resource group called **foo** , enter:

```
chrg -l collocated -o 0 -p 34 foo
```

2. To change the name of resource group **foo** to **clusterfoo** with location relationship now changed to None, and allowed node name of cluster, enter:

```
chrg -c clusterfoo -l 0 -n cluster foo
```

3. To change the location relationship to **None**, and allowed node name **cluster** for all resource groups containing the name **foo**, enter:

```
chrg -l none -n cluster -s 'Name like "%foo%''
```

## Files

**/usr/sbin/rsct/bin/chrg**            Location of the **chrg** command.

## See Also

The **addrgmbr**, **samctrl**, **chrgmbr**, **lsrg**, **mkrsg**, **rmrg**, **rmrgmbr** commands.

The **Resource\_Data\_Input** file.

The **rmccli** General Information file.

---

## chrgmbr

### Name

**chrgmbr** – Changes the persistent attribute value(s) of a managed resource in a resource group

### Synopsis

```
chrgmbr [-h] [-i] [a | d | r] [-c New_group] [-m T | F] [-T] [-V]
-g Resource_group [-p A | O]
```

```
chrgmbr [-h] [-i] -s [a | d | r] -c New_group] [-m T | F] [-T] [-V]
-g Resource_group] [-p A | O] "selection_string"
```

```
chrgmbr [-h] [-i] [a | d | r] [-c New_group] [-m T | F] [-T] [-V]
-s [-g Resource_group] [-p A | O] [Resource_class: "selection_string" [Resource_class: "selection_string" [...]]
```

```
chrgmbr [-h] [-i] [a | d | r] [-c New_group] [-m T | F] [-T] [-V]
-g Resource_group] [-p A | O] Resource_class:Resource_name[:Node][, Resource_name[:Node][, ...]]
[Resource_class:Resource_name[:Node][, Resource_name[:Node][, ...]] [...]
```

### Description

The **chrgmbr** command changes the attributes of the specified member resources. When the selection string is the only parameter, then it is applied directly to IBM.ManagedResource class. Otherwise, a resource name or selection string must be specified with the class to which it belongs. This command allows the user to specify changes to the Mandatory attribute of a managed resource by using the **-m** option and also allows the user to change the resource group to which the resource belongs by using the **-c** option. A fixed resource must include the resource class and the node name at which it resides. The resource class and resource name, the resource name and the node must be separated by a colon.

### Options

**-c** *New\_group*

Specifies a new resource group name, which changes the MemberOf attribute of the managed resource. This option specifies the *Resource\_group*, or changes the *Resource\_group* of which this managed resource is a member, to the *New\_group*.

**-g** *Resource\_group*

Specifies the name of the resource group of which the resources are members. This represents the MemberOf attribute of the Managed Resource.

**-h** Writes the command's usage statement to standard output.

**-i** Interactive. Prompt before changing resource groups.

**a | d | r**

Specifies what should be done with the specified node list.

**a** Adds the specified nodes to the list of excluded nodes.

**d** Deletes the specified nodes from the list.

**r** Replaces the exclude list with the specified nodes.

**-m T | F**

Specifies whether this managed resource is required in the resource group.

**T** TRUE. These managed resources are required by the resource group.

**F** FALSE. These managed resources are not required by the resource group.

- s** Specifies that a selection string will be used.
- T** Writes the command's trace messages to standard error. For your software-service organization's use only.
- V** Writes the command's verbose messages to standard output.
- p A|O** Specifies the method to be used when selecting a node for placing the resource on. The value can be:
 

|          |                               |
|----------|-------------------------------|
| <b>A</b> | Any.                          |
| <b>O</b> | Ordered. This is the default. |

## Parameters

### *Resource\_class*

Specifies the name of the resource class to which the member resource belongs. The resource and the resource class must be separated by the colon (:) delimiter. Also the selection\_string and its resource\_class must be separated by the colon (:) delimiter.

### *Resource\_class:"selection\_string"*

The -s option flag determines this operand. Each selection string must be preceded by a resource class. Resource\_class indicates the name of the resource class the selection\_string will be applied to. The selection\_string and its Resource Class must be separated by the colon (:) delimiter. The specified selection string will be applied to its corresponding Resource\_class attributes to determine which member resources are to be changed. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
-s 'Name like "%"' (For all resources)
```

For information on how to specify selection strings, see “Using expressions” on page 222.

### *Resource\_class:Resource\_name[:Node]*

This specifies one or more member resources that are to be changed. Resources belonging to different resource classes can also be specified using this syntax. However, member resources belonging to different classes must be separated by spaces. A member resource is identified by the resource's class ( Resource\_class), the resource's name ( Resource\_name), and optionally the node ( Node) the resource is on. The Resource\_class, Resource\_Name and Node must be separated by a colon.

Resource\_class is the name of the resource class the member resource belongs to.

Resource\_name is the name of the actual member resource in its class. Node is the name of the node the resource is located on. The Node is required when attempting to change a fixed member resource. The Node and its resource must be separated by the colon delimiter (:). Multiple resources belonging to the same resource class can also be specified and are separated by a comma. In this case the first resource or Resource\_name[:Node] of a resource class must be preceded by the resource\_class it belongs to and it must be separated by the colon delimiter (:), while the remaining resources are separated by a comma.

*Resource\_class:Resource\_name[:Node][Resource\_name[:Node][,...]]*

The member resources must exist in the resource group for it to be changed.

### *Selection\_string*

Specifies the selection string. The -s option determines this parameter. When this is the only parameter, then the selection string is applied to the member resources. But when included with Resource\_class, each selection string in this case will be applied to its corresponding Resource\_class attributes to determine which resources are to be modified in the Resource\_group. The selection\_string and its resource\_class must be separated by the colon (:)

## chrgmbr

delimiter. The selection string must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
-s 'Name like "%"' (For all resources)
```

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To change the Mandatory attribute to TRUE of a member resource **tester** that belongs to resource class IBM.Application, enter:  

```
chrgmbr -m T IBM.Application:tester
```
2. To change the resource group to which member resources **tester**, **Jfoo**, and **Dfoo** of resource class IBM.Application belong from the current resource group **foo** to resource group **footest**, enter:  

```
chrgmbr -c footest -g foo IBM.Application:tester,Jfoo,Dfoo
```
3. To change the Mandatory attribute to FALSE of selected member resources of resource group **foo** with ResourceType attribute set to one, belonging to resource class IBM.Application, enter:  

```
chrgmbr -m F -s -g foo IBM.Application:"ResourceType==1"
```

## Files

**/usr/sbin/rsct/bin/chrgmbr** Location of the **chrgmbr** command.

## See Also

The **addrgmbr**, **samctrl**, **chrel**, **chrg**, **lsrg**, **mkrel**, **mkrq**, **mkrq**, **rmrel**, **rmrgmbr** commands.

The **rmcli** General Information file.

“RSCT RMC commands and files” on page 145 contains more information regarding RMC operations.

---

## Isequ

### Name

**Isequ** – Lists already-defined resource equivalencies and their attributes.

### Synopsis

```
Isequ [-h] [-s "selection_string"] [-e Equivalency] [-A p | d | b] [-l | -t | -d | -D Delimiter] [-T] [-V]
[Attr...]
```

### Description

The **Isequ** command lists the Equivalencies. If the equivalency name is omitted, *all* of the defined equivalencies will be listed. If an equivalency is specified, the persistent attributes of the specified equivalency will be listed. If the *Attr* operand is specified, the attributes specified for the equivalency will be listed.

If the attribute flag is specified, the persistent or/and dynamic attributes of the equivalencies will be listed. If specified, the *Attr* operand will override the **-A** flag. The *Attr* operand must be a valid equivalency attribute, when listing equivalency information. When the *AttrOperands* are specified, the exact value of these attributes will be listed. Otherwise, some of these attribute values are translated into English words.

### Options

#### **-A p | d | b**

Attribute type. By default, only persistent attributes are displayed. To view all the attributes of the member resources, this option must be used with **-A b** attribute type. The value can be:

- p** Display only persistent attributes. For best performance, specify this value.
- d** Display only dynamic attributes.
- b** Display both persistent and dynamic attributes.

**-d** Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the **-D** option if you wish to change the default delimiter.

#### **-D *Delimiter***

Specifies delimiter-formatted output that uses the specified delimiter. Use this option to specify something other than the default colon (:); for example, when the data to be displayed contains colons. Use this option to specify a delimiter of one or more characters.

#### **-e *Equivalency***

The name of the defined Equivalency to be listed.

**-h** Writes the command's usage statement to standard output.

**-l** Specifies long formatted output. Each attribute is displayed on a separate line. This is the default display format.

#### **-s "Selection\_string"**

Specifies a selection string. Used without the **-e** option, this option is applied to all the defined equivalencies. If the **-e** option is specified with a selection string, then this selection is applied only to the specified equivalency. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
```

Only persistent attributes may be included in a selection string.

## Isequ

For information on how to specify selection strings, see “Using expressions” on page 222.

- t** Specifies tabular formatted output. Each attribute is displayed in a separate column, one resource per line.
- T** Writes the command’s trace messages to standard error. For your software-service organization’s use only.
- V** Writes the command’s verbose messages to standard output.

## Parameters

*Attr* Specifies the names of one or more attributes whose values are to be displayed. These attributes must be separated by spaces.

## Exit Status

- 0** Command has run successfully.
- 1** Error occurred with RMC.
- 2** Error occurred with CLI script.
- 3** Incorrect option on command line.
- 4** Incorrect parameter on command line.
- 5** Error occurred with RMC that was based on faulty command line input.
- 6** Resource specified was not found.
- 7** Resource already exists.

## Security

This command requires root authority.

## Examples

1. To list all currently-defined equivalencies, enter:

```
lsequ
```

Output:

Displaying Equivalencies:

```
foo
foo1
equ1
equ2
```

2. To list the persistent attributes of the equivalency **foo**, enter:

```
lsequ -A p -e foo
```

Output:

Displaying Equivalency Information:

Persistent Attributes  
for Equivalency "foo".

Equivalency 1:

```
Name = foo
MemberClass = IBM.Application
Resource:Node[Membership] = {IBM.Application:Nate}
SelectString = ""
SelectFromPolicy = Any
MinimumNecessary = 1
```

3. To list all the attributes of the equivalency **equ1**, enter:

```
lsequ -A b -e equ1
```

**Output:**

```
Displaying Equivalency Information:
Equivalency 1:
Name = equ1
MemberClass = IBM.Application
Resource:Node [Membership] = {Nate:node1,Shoo:node2}
SelectString = ""
SelectFromPolicy = Ordered
MinimumNecessary = 1
Resource:Node [ValidSelectResources] = {Nate:node1,Shoo:node2}
Resource:Node [InvalidResources] = {}
AutomationDetails = [10,1,100,2,80,1,0]
```

4. To list all the attributes of the equivalency **equ1** in verbose mode, enter:

```
lsequ -V -A b -e equ1
```

**Output:**

```
Displaying Equivalency Information:
Equivalency 1:
Name = equ1
MemberClass = IBM.Application
Resource:Node [Membership] = {Nate:node1,Shoo:node2}
SelectString = ""
SelectFromPolicy = Ordered
MinimumNecessary = 1
Resource:Node [ValidSelectResources] = {Nate:node1,Shoo:node2}
Resource:Node [InvalidResources] = {}
AutomationDetails = [10,1,100,2,80,1,0]
 CompoundState = Automation
 DesiredState = Online
 ObservedState = Pending Online
 BindingState = Bound
 AutomationState = Internal
 StartableState = Yes
 HealthState = Not Applicable
```

5. To list the dynamic attributes of equivalency **foo**, enter:

```
lsequ -A d -e foo
```

## Files

**/usr/sbin/rsct/bin/lsequ** Location of the **lsequ** command.

## See Also

The **chequ**, **samctrl**, **mkequ**, **rmequ** commands.

The **rmccli** General Information file

---

## Isrel

### Name

**Isrel** – Lists an already-defined managed relationship and its attributes.

### Synopsis

```
Isrel [-h] [-s "selection_string"] [-A p | d | b] [-l | -t | -d | -D Delimiter] [-T] [-V] [Attr...]
```

```
Isrel [-h] [-M relation_name] [-A p | d | b] [-l | -t | -d | -D Delimiter] [-T] [-V] [Attr...]
```

```
Isrel [-h] [-S Source_Class[:Resource_name[:Node]]] [-s "Source_string"]
[-G Target_Class[:Resource_name[:Node]][,Target class[:Resource_name[:Node][...]]]]
[-g "Target_string"] [-P Current_relationship]
[-O Current_condition] [-A p | d | b] [-l | -t | -d | -D Delimiter]
[-T] [-V] [Attr...]
```

### Description

The **Isrel** command lists the managed relationships. If the relationship name is omitted, *all* managed relationships will be listed. If the resource relationship is specified, the persistent attributes of the specified relationship will be listed. If the *Attr* parameter is specified, the attributes specified for the relationship will be listed.

If the attribute option is specified, the persistent or dynamic attributes of the relationships will be listed. If specified, the *Attr* parameter will override the **-A** option. The *Attr* parameter must be a valid relationship attribute when listing relationship information. If the *Attr* parameters are specified, the exact value of these attributes will be listed. Otherwise, some of these attribute values are translated to English words.

### Options

#### **-A p | d | b**

Specifies an attribute type. By default, only persistent attributes are displayed. This option can be used only when no attribute names are specified on the command line. To view all the persistent attributes of the member resources, this flag must be used with **-p** attribute type.

- p** Displays only persistent attributes.
- d** Displays only dynamic attributes.
- b** Displays both persistent and dynamic attributes.

For best performance, specify the **-A p** option.

**-d** Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the **-D** option if you wish to change the default delimiter.

#### **-D Delimiter**

Specifies delimiter-formatted output that uses the specified delimiter. Use this option to specify something other than the default colon (:); for example, when the data to be displayed contains colons. Use this option to specify a delimiter of one or more characters.

#### **-g Target\_string**

The *Target\_string* is applied to all resources of the resource class specified by the **-G** flag, to determine the targets of the relationship.

If the **-g** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-G** option .

**-G Target\_class[:Resource\_name[:Node]]**

Specifies the target resources to display with the relationship. The target resource or resources will be used with the source and any other specified query to find a match from all of the defined relationships. A target resource is identified by the resource's class (*Target\_class*), the resource name (*Resource\_name*), and, optionally, the node (*Node*) at which the resource resides. *Target\_class*, *Resource\_name*, and *Node* must be separated by colons. Multiple target resources can be specified and are separated by commas. The colons are required, but omitting *Target\_class* causes the previously specified class to be used when multiple target resources are specified.

If the **-g** option is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for the **-G** option.

- h** Writes the command's usage statement to standard output.
- l** Specifies long formatted output. Each attribute is displayed on a separate line. This is the default display format.
- M** *relation\_name*  
Specifies the name of the managed relationship to be listed.
- P** *Current\_relationship*  
Specifies the managed relationship value used to query all the defined relationships to find a match. The value can be a numeric value or the word (not case sensitive):
  - 0x0001 or Collocated**  
Specifies that the source and the target resources are to be located on the same node.
  - 0x0002 or AntiCollocated**  
Specifies that the source and the target resources are to be located on different nodes.
  - 0x0003 or Affinity**  
Specifies that the source resource and its specified affinity resource are to be located on the same node, if possible.
  - 0x0004 or AntiAffinity**  
Specifies that the source resource and its specified affinity resource are to be located on different nodes, if possible.
  - 0x0005 or IsStartable**  
Specifies that the source and its target resources are startable.
  - 0x0006 or StartAfter**  
Specifies that the source is to be started after the target resources.
  - 0x0007 or DependsOn**  
Specifies that the target resources must be online before the source resource is started and includes an implicit collocation between the source and target resources.
  - 0x0008 or DependsOnAny**  
Specifies that the target must be online before the source resource is started. It is identical to the DependsOn relationship except that it does not provide the collocated constraint for the start sequence. Therefore the source and target resources may not be started on the same node.
  - 0x0009 or StopAfter**  
Specifies that the source resource may not be stopped until after the target resource has been brought offline.
  - 0x000A or ForcedDownBy**  
Specifies that the source resource must be forced offline in the event that either the target resource goes offline unexpectedly or the target resource itself is forced offline.

## Isrel

### **-O** *Current\_condition*

If condition query. Specifies the condition value to query all the defined relationships to find a match. The value can be the numeric value or the word (not case sensitive) as shown below:

#### **0x0000** or **NoCondition**

Specifies unconditional location relationship.

#### **0x0001** or **IfOnline**

Specifies that the target resource is online.

#### **0x0002** or **IfOffline**

Specifies that the target resource is offline, failed offline, or unknown.

#### **0x0003** or **IfNotOnline**

Specifies that the target resource is not online.

#### **0x0004** or **IfNotOffline**

Specifies that the target resource is neither offline nor failed offline

### **-s** "*selection\_string*"

Specifies a selection string. This option without the **-S** option will be applied to all the defined relationships and its attribute fields. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
```

```
-s 'Name ?= "test"'
```

Only persistent attributes may be listed in a selection string.

For information on how to specify selection strings, see "Using expressions" on page 222.

### **-s** *Source\_string*

Specifies the source selection string of the relationship to list. The *Source\_string* is applied to all resources of the resource class specified by the **-S** option to determine the source of the relationship. The result of the selection may contain one or more resources.

If the **-s** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-S** option .

### **-S** *Source\_class[:Resource\_name[:Node]]*

Specifies the source resource of the relationship to list. A source resource is identified by the resource class (*Resource\_class*) of the resource, the resource name (*Resource\_name*), and, optionally, the node (*Node*) at which the resource resides. The result of the selection may contain one or more resources.

If the **-s** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-S** option.

**-t** Specifies tabular formatted output. Each attribute is displayed in a separate column, one resource per line.

**-T** Writes the command's trace messages to standard error. For your software-service organization's use only.

**-V** Writes the command's verbose messages to standard output.

## Parameters

*Attr* Specifies the names of one or more attributes whose values are to be displayed. These attributes

must be separated by spaces. When the **-m** option is specified, these attributes must match the member resource attribute fields; when the **-m** option is not specified, the attributes must match the resource group attribute fields.

**Node** Specifies the node name. The node name may have to be included when an attempt has been made to list a fixed member resource. The node and its resource must be separated by the colon (:).

#### *Resource\_class*

Specifies the name of the resource class to which the member resource belongs. The resource and the resource class must be separated by spaces.

#### *Resource\_name*

Specifies the resource name. One or more resources may be specified along with the resource class to which it belongs. Each resource or set of resources must be preceded by its resource class. The resource and the resource class must be separated by spaces.

## Exit Status

- 0** Command has run successfully.
- 1** Error occurred with RMC.
- 2** Error occurred with CLI script.
- 3** Incorrect flag on command line.
- 4** Incorrect parameter on command line.
- 5** Error occurred with RMC that was based on faulty command line input.
- 6** Resource specified was not found.
- 7** Resource already exists.

## Security

This command requires root authority.

## Examples

1. To list all the currently-defined managed relationships enter:

```
lsrel
```

#### Output:

```
Displaying Managed Relations :
Name Resource:Node[Source] ResourceGroup[Source]
foo IBM.Application:ja Foo
jaffinity IBM.Application:ja Foo
 IBM.ResourceGroup:john Ja
```

2. To list the persistent attributes of the managed relationship **foo**, enter:

```
lsrel -A p -M foo
```

#### Output is as follows:

```
Displaying Managed Relationship Information:
Persistent Attributes
for Managed Relationship "foo".
```

```
Managed Relationship 1:
Class:Resource:Node[Source] = IBM.Application:ja:nodename
```

## Isrel

```
Class:Resource:Node[Target] = {IBM.Application:Nate:Nodename}
Relationship = Collocated
Conditional = NoCondition
Name = foo
```

3. To list the managed relationship that match the source **foo** from class IBM.Application, having a Collocated relationship value with target resource **narten** from class IBM.ResourceGroup, enter:

```
Isrel -S IBM.Application:Foo -P Collocated -G IBM.ResourceGroup:narten
```

Output is as follows:

```
Displaying Managed Relationship Information:
Managed Relationship 1:
Name = Foo
Class:Resource:Node[Source] = IBM.Application:Foo
Class:Resource:Node[Target] = {IBM.ResourceGroup:Narten ,IBM.Application:ja}
Relationship = Collocated
Conditional = NoCondition
ResourceGroup[Source] = charm
```

4. To list the dynamic attributes of the managed relationship name **foo**, enter:
5. To list the persistent attributes of the managed relationship name **foo** by using a selection string, enter:

```
Isrel -A d -M foo
```

```
Isrel -A p -s 'Name == "foo"'
```

## Files

**/usr/sbin/rsct/bin/Isrel** Location of the **Isrel** command.

## See Also

The **addrgmbr**, **samctrl**, **chrg**, **chrgmbr**, **mkrgr**, **rgrg**, **rgrgmbr** commands.

The **rmccli** General Information file.

---

## lsrg

### Name

**lsrg** – Lists already-defined resource groups and their members.

### Synopsis

```
lsrg [-h] [-m] [-s "Selection_string"] [-A p | d | b] [-l | -t | -d | -D Delimiter] [-g Resource_group] [-T] [-V] [[Attr...]
```

```
lsrg [-h] [-m] -c [-s "Selection_string"] [-A p | d | b] [-l | -t | -d | -D Delimiter] [-g Resource_group] [-T] [-V]
```

```
Resource_class:Resource[:Node][,Resource[:Node][,...]][Resource_class:Resource[:Node][,Resource[:Node][,...]] [...]
```

### Description

The **lsrg** command lists resource groups or the members of a resource group. If the resource group name is omitted, all of the resource groups are listed. If the resource group name is specified without the **-m** option, the persistent attributes of the specified group will be listed. If the *Attr* parameter is specified, then the attributes specified for the resource group are listed. If the *Attr* parameter is specified here, the attributes specified for the resource group will be listed.

If the resource group name is specified with the **-m** option, the members of that resource group will be listed. If the attribute option (**-A**) is specified, the persistent or dynamic attributes of the resource group members will be listed. Default for the **-m** option will list the members Resource class name, Resource name and the MemberOf attribute. If the *Attr* operand is specified, the attributes specified for the managed resources (member resources) will be listed.

The *Attr* parameter if specified will override the **-A** option. The *Attr* parameter must be a valid resource group attribute when listing resource group information, and likewise be a valid member resource attribute when listing a managed resource. If the *Attr* parameters are specified, then the exact value of these attributes will be listed. Otherwise, some of these attribute values are translated into English words.

### Options

#### **-A p | d | b**

Specifies an attribute type. By default, only persistent attributes are displayed. This option can be used only when no attribute names are specified on the command line. To view all the persistent attributes of the member resources, this flag must be used with **p** attribute type.

- p** Displays only persistent attributes.
- d** Displays only dynamic attributes.
- b** Displays both persistent and dynamic attributes.

For best performance, specify the **-A p** option.

- c** Specifies the resource and its class. This option indicates that parameters are resource classes, resource names, and, optionally, node names. The attributes of the member resources are listed or displayed. The attributes and selection strings when this option is used are applied to member resources and not the resource groups; therefore, these attributes must exist for the member resources.
- d** Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the **-D** option if you wish to change the default delimiter.

## lsrg

### **-D** *Delimiter*

Specifies delimiter-formatted output that uses the specified delimiter. Use this option to specify something other than the default colon (:); for example, when the data to be displayed contains colons. Use this option to specify a delimiter of one or more characters.

### **-g** *Resource\_group*

Displays resource group information such as the name and the persistent and dynamic attributes of the defined resource group. When the **-g** option is used with the **-m** option and the **-c** option, the member resources have to belong to the specified resource group.

**-h** Writes the command's usage statement to standard output.

**-l** Specifies long formatted output. Each attribute is displayed on a separate line. This is the default display format.

**-m** Specifies member resources and lists or displays the attributes of the member resources. The attributes and selection strings when this option is used are applied to member resources and not the resource groups; therefore, the attribute fields must exist for the member resources.

### **-s** *"Selection\_string"*

Specifies a selection string. This option without the **-m** option is applied to the resource-group attribute fields; with the **-m** option or **-c** option, the selection string is applied to the member-resource attribute fields. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name != "test"'
```

For information on how to specify selection strings, see "Using expressions" on page 222.

**-t** Specifies tabular formatted output. Each attribute is displayed in a separate column, one resource per line.

**-T** Writes the command's trace messages to standard error. For your software-service organization's use only.

**-V** Writes the command's verbose messages to standard output.

## Parameters

*Attr* Specifies the names of one or more attributes whose values are to be displayed. These attributes must be separated by spaces. When the **-m** option is specified, these attributes must match the member resource attribute fields; when the **-m** option is not specified, the attributes must match the resource group attribute fields.

*Node* Specifies the node name. The node name may have to be included when an attempt has been made to list a fixed member resource. The node and its resource must be separated by the colon (:) delimiter.

### *Resource\_class*

Specifies the name of the resource class to which the member resource belongs. The resource and the resource class must be separated by the colon (:) delimiter.

### *Resource\_name*

Specifies the resource name. One or more resources may be specified along with the resource class to which it belongs. Each resource or set of resources must be preceded by its resource class. The resource and the resource class must be separated by the colon (:) delimiter.

## Exit Status

**0** Command has run successfully.

- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To list all the currently-defined resource groups, enter:

```
lsrg
```

Output is similar to:

Resource Group Names:

```
foo
clusterfoo
```

2. To list all the attributes of the resource group **foo**, enter:

```
lsrg -A b -g foo
```

Output is as follows:

Displaying Resource Group Information:

Resource Group 1:

```
Name = "foo"
MemberLocation = Collocated
Priority = 22
AllowedNodes = "node1"
NominalState = Online
OpState = Offline
TopGroup = foo
AutomationDetails = [10,1,100,2,80,1,0]
```

3. To list all the attributes of the resource group **foo** in verbose mode, enter:

```
lsrg -A b -V -g foo
```

Output is as follows:

Displaying Resource Group Information:  
Starting to list resource group information.

Displaying Resource Group information:

All Attributes  
For Resource Group "foo".

Resource Group 1:

```
Name = "foo"
MemberLocation = Collocated
Priority = 22
AllowedNodes = "node1"
NominalState = Online
OpState = Offline
TopGroup = foo
AutomationDetails = [10,1,100,2,80,1,0]
CompoundState = Automation
```

## lsrg

```
DesiredState = Online
ObservedState = Pending Online
BindingState = Bound
AutomationState = Internal
StartableState = Yes
HealthState = Not Applicable
```

Completed listing resource group information.

4. To list the attributes of the members of resource group **foo**, enter:

```
lsrg -m -V -Ab -g foo
```

Output is as follows:

```
Displaying Member Resource information:
for Resource Group "foo".
Member Resource 1:
Resource:Node [ManagedResource] = IBM.Application:charm
Mandatory = False
MemberOf = foo
OpState = Offline
AutomationDetails = [10,1,100,2,80,1,0]
 CompoundState = Automation
 DesiredState = Online
 ObservedState = Pending Online
 BindingState = Bound
 AutomationState = Internal
 StartableState = Yes
 HealthState = Not Applicable
```

5. To list the persistent attributes of the member resources of resource group **foo**, enter:

```
lsrg -m -A p -g foo
```

Output is as follows:

```
Displaying Member Resource information:
for Resource Group "foo".
Member Resource 1:
 Resource:Node[ManagedResource] = IBM.Application:charm
 Mandatory = False
 MemberOf = foo
```

6. To list the persistent attributes of the member resources of a resource group **foo** using a selection string, enter:

```
lsrg -m -A p -s 'MemberOf == "foo"'
```

Output is as follows:

```
Displaying Member Resource information:
for Resource Group "foo".
Member Resource 1:
 Resource:Node[ManagedResource] = IBM.Application:charm
 Mandatory = False
 MemberOf = foo
```

7. To list the attributes of a managed resource, enter:

```
lsrg -m
```

Output is as follows:

```
Displaying Member Resource information:
Class:Resource:Node[ManagedResource]Mandatory MemberOf OpState
IBM.Application:WebServer True DemoRG Online
IBM.ServiceIP:WebIP True DemoRG Online
```

## Files

**/usr/sbin/rsct/bin/lsrg** Location of the **lsrg** command.

## See Also

The **addrgubr**, **samctrl**, **chrel**, **chrg**, **chrgubr**, **mkrel**, **mkrg**, **rmrel**, **rmrg**, **rmrgubr** commands.

The **rmcli** General Information file.

---

## lsrgreq

### Name

**lsrgreq** – lists the outstanding requests applied against the resource groups or managed resources.

**lsrgreq** belongs to a group of commands (rgreq, rgmbreq, and lsrgreq) which allow an operator to introduce persistent requests into IBM Tivoli System Automation. Using this set of commands the operator can start, stop, cancel, or move resource groups and managed resources.

### Synopsis

**lsrgreq** [-h] [-L ] [-m] [-I | -t | -d | -D *Delimiter*] [-g *Resource\_group*] [-T] [-V]

**lsrgreq** [-h] [-L ] -m [-I | -t | -d | -D *Delimiter*] [-T ] [-V] [-L]  
*Resource\_class:Resource[:Node][,Resource[:Node][,...]]*  
*[Resource\_class:Resource[:Node][,Resource[:Node][,...]]]*

**lsrgreq** [-h] [-L ] -m [-I | -t | -d | -D *Delimiter*] -s [-T] [-V] *Resource\_class:"Selection\_String"*  
*[Resource\_class:"Selection\_String" [...]]*

### Description

The **lsrgreq** command lists the outstanding requests applied against the resource groups or managed resources. By means of the **lsrgreq** command either the active or all the outstanding requests may be listed. The “- m” option will only list the action requests on the managed or member resources. Otherwise, the action requests on the resource groups will be listed.

- h** Help. Writes the command’s usage statement to standard output.
- g** *Resource\_group*  
The name of the resource group whose actions are to be listed.
- L** Lists all the action requests on the specified resource group or managed resources. If this option is not specified then only the active requests will be displayed or listed.
- m** Member Resources. Lists or displays the actions on the member resources.
- s** Specifies that a selection string will be used.
- I** Long formatted output. Each attribute will be displayed on a separate line. This is the default display format.
- t** Tabular formatted output. Each attribute will be displayed on a separate column, one resource per line.
- d** Default delimiter formatted output. Default delimiter is a colon. Use the -D option if you wish to change the delimiter.
- D** *Delimiter*  
Delimiter formatted output. Use this flag to specify something other than the default colon(:). An example is when the data to be displayed contains colons. One or more characters can be used as the Delimiter.
- T** Trace. Writes the command’s trace messages to standard error. For software-service organization’s use only.
- V** Verbose. Writes the command’s verbose messages to standard output.

## Parameters

*Resource\_class*:*selection\_string*

The **-s** option flag determines this operand. Each selection string must be preceded by a resource class. *Resource\_class* indicates the name of the Resource class the *selection\_string* will be applied to. The *selection\_string* and its *Resource\_class* must be separated by colon or ":" delimiter. The specified selection string will be applied to its corresponding *Resource\_class* attributes to determine which member resources are to be removed from the *Resource\_group*. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

- s IBM.Application:`Name=="testing"``
- s IBM.Application:`Name ?="test"``
- s IBM.Application:`Name like "%"`` (For all resources)

For information on how to specify selection strings, see "Using expressions" on page 222.

*Resource\_class*:*Resource*[:*Node*]

Member Resources. This specifies one or more member resources whose action requests are to be listed. Resources belonging to different resource classes can also be specified using this syntax. However, member resources belonging to different classes must be separated by spaces. A member resource is identified by the resource's class (*Resource\_class*), the resource's name (*Resource*), and optionally the node (*Node*) the resource is on. The *Resource\_class*, *Resource* and *Node* must be separated by a colon. *Resource\_class* is the name of the resource class the member resource belongs to. *Resource* is the name of the actual member resource in its class. The node may have to be included when trying to act on a fixed member resource. The node and its resource must be separated by the colon or ":" delimiter. Multiple resources belonging to the same resource class can also be specified and are separated by a comma:

*Resource\_class*:*Resource*[:*Node*][,*Resource*[:*Node*][,...]]

The member resources must exist in the resource group for the requested action to take place.

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To list the active action requests on the resource group RG1, enter:  
lsrgreq -g RG1
2. To list the requests on all the resources belonging to resource group RG1 in tabular format, enter:

## lsrgreq

```
lsrgreq -m -g RG1
```

Displaying Member Resource request information:

Active Requests

For Resource Group "RG1":

Member Resource 1:

Class:Resource:Node[ManagedResource] = IBM.Application:R1

Priority = Force

Action = start

Source = Operator

ActiveStatus = Active

Member Resource 2:

Class:Resource:Node[ManagedResource] = IBM.Application:R2

Priority = low

Action = start

Source = ExtSched

ActiveStatus = Active

3. To list all the action requests applied to resource group RG1 , enter:

```
lsrgreq -L -t -g RG1
```

Displaying Resource Group request information:

All request information

For Resource Group "RG1".

| ResourceGroup | Priority | Action | Source     | NodeList | ActiveStatus | MoveStatus |
|---------------|----------|--------|------------|----------|--------------|------------|
| RG1           | low      | stop   | ExtSched   | {}       | InActive     | None       |
| RG1           | High     | start  | Automation | {}       | Active       | None       |
| RG1           | Force    | stop   | Operator   | {}       | Active       | None       |

## Files

**/usr/sbin/rsct/bin/lsrgreq** Location of the **lsrgreq** command.

## See Also

The **addrgmb**, **chrg**, **chrgmbr**, **lsrg**, **mkr**, **rgreq**, **rmsg**, and **rmsgmbr** commands.

---

## Issamctrl

### Name

**Issamctrl** – lists already defined IBM Tivoli System Automation control parameters and its attributes.

### Synopsis

**Issamctrl** [-h] [-A p | d | b] [-l | -t | -d | -D *Delimiter*] [-T] [-V] [*Attr...*]

### Description

The **Issamctrl** command lists the control parameter values already set for IBM Tivoli System Automation. If the *Attr* operand is specified, then the value of the attributes specified will be listed. If the attribute flag -A is specified then the persistent or/and dynamic attributes of IBM Tivoli System Automation will be listed. The *Attr* operand if specified will override the -A flag. The *Attr* operand must be a valid attribute of IBM Tivoli System Automation Control Class. When the *AttrOperands* are specified then the exact value of these attributes will be listed. Otherwise, some of these attribute values are translated to English words.

### Options

- h Help. Writes the command's usage statement to standard output.
- A p | d | b Attribute type. By default only persistent attributes are displayed. To view all the persistent attributes of the member resources this flag must be used with *p* attribute type.
  - p - Displays only persistent attributes.
  - d - Displays only dynamic attributes.
  - b - Displays both persistent and dynamic attributes.
 For best performance specify the -A p flag.
- l Long formatted output. Each attribute will be displayed on a separate line. This is the default display format.
- t Tabular formatted output. Each attribute will be displayed on a separate column, one resource per line.
- d Default delimiter formatted output. Default delimiter is a colon. Use the -D flag if you wish to change the delimiter.
- D *Delimiter* Delimiter formatted output. Use this flag to specify something other than the default colon(:). An example is when the data to be displayed contains colons. One or more characters can be used as the Delimiter.
- T Trace. Writes the command's trace messages to standard error. For your software-service organization's use only.
- V Writes the command's verbose messages to standard output.

### Parameters

*Attr* The name of the attributes whose values are to be displayed. These attributes must be separated by spaces.

### Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.

## Issamctrl

- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

To list all the currently defined IBM Tivoli System Automation parameters, type the following command:

```
Issamctrl
```

Output:

Displaying IBM Tivoli System Automation Control Information:

```
SAMControl:
 Timeout = 60
 RetryCount = 3
 Automation = Auto
 ExcludedNodes = {}
 ResourceRestartTimeout = 5
 ActiveVersion = [1.2.0.0,Tue 04 May 2004 12:30:48 PM EDT]
 Enable Publisher = Disabled
```

## Files

`/usr/sbin/rsct/bin/Issamctrl` Location of the **Issamctrl** command.

## See Also

The **samctrl** command.

The **rmccli** General Information file

---

## mkequ

### Name

**mkequ** – Makes a resource equivalency.

### Synopsis

**mkequ** [-h] -f *data\_input\_file* [-T] [-V]

**mkequ** [-h] [-p **A** | **O**] [-m *Minimum\_Necessary*] [-T] [-V] -S "*Static\_select\_string*" *Equivalency Resource\_class*

**mkequ** [-h] [-p **A** | **O**] [-m *Minimum\_Necessary*] [-T] [-V] -D "*Dynamic\_select\_string*" *Equivalency Resource\_class*

**mkequ** [-h] [-p **A** | **O**] [-m *Minimum\_Necessary*] [-T] [-V] *Equivalency Resource\_class:Resource\_name[:Node][,Resource\_name[:Node][,...]]*

### Description

The **mkequ** command makes a resource equivalency among the specified resources. The resources must all be from the same resource class.

Equivalencies and their attribute flags can be specified on the command line as the syntax indicates or from a file when the -f option is specified. The value must be of the same data type that is defined for this resource attribute. Use the **lsrsrcdef** command to verify the data type and attribute field for each attribute.

### Options

-D "*dynamic\_select\_string*"

Dynamic Selection String. The specified selection string will be saved as the SelectString attribute for the concerned equivalency. This *selection\_string* is then applied to all resources of the specified resource class by the Recovery Manager, to dynamically determine what members are to be included in the specified equivalency. This option cannot be combined with the *Resource\_name[:Node]* parameter. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-D 'Name == "testing"'
```

```
-D 'Name ?= "test"'
```

-D cannot be combined with -S.

For information on how to specify selection strings, see "Using expressions" on page 222.

-f *data-input\_file*

Specifies the name of the file that contains equivalency attribute information.

-h Writes the command's usage statement to standard output.

-p **A** | **O**

Specifies the equivalency select-from policy. The value can be:

**A** Any

**O** Ordered

The default value is **Any**.

## mkequ

### -S "*static\_select\_string*"

Static Selection String. The specified selection string will be applied to the specified resource class, to determine the resources that make up the equivalency. These resources will then be saved as the MemberShip attribute for the equivalency. This option cannot be combined with the *Resource\_name* [:*Node*] parameter. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-S 'Name == "testing"'
```

```
-S 'Name ?= "test"'
```

-S cannot be combined with -D.

For information on how to specify selection strings, see "Using expressions" on page 222.

### -m *Minimum-Necessary*

Minimum Necessary Equivalency. Specifies the minimum necessary members to make an equivalency valid. The default value is 1.

-T Writes the trace messages of the command to standard error. For your software service organization's use only.

-V Writes the verbose messages of the command to standard error.

## Parameters

### *Equivalency*

Specifies the unique name of the new equivalency to be created.

### *Resource\_class:Resource\_name[:Node]*

Specifies one or more resources that will be members of the equivalency. All of the resources must be from the same resource class. A equivalency resource is identified by the resource name (*Resource\_name*), and, optionally, the node (*Node*) where the resource resides. *Resource\_name* and *Node* must be separated by colons. Multiple resources can be specified and are separated by a comma. *Resource\_class* is the name of the resource class the equivalency resource belongs to. The first resource or *Resource\_name[:Node]* must be preceded by the *Resource\_class* it belongs to and it must be separated by the colon (:) delimiter. Multiple resources can be specified and are separated by a comma. Since all resources must be from the same class the *Resource\_class* is to be specified only once as the syntax indicates.

## Exit Status

- |   |                                                                      |
|---|----------------------------------------------------------------------|
| 0 | Command has run successfully.                                        |
| 1 | Error occurred with RMC.                                             |
| 2 | Error occurred with CLI script.                                      |
| 3 | Incorrect flag on command line.                                      |
| 4 | Incorrect parameter on command line.                                 |
| 5 | Error occurred with RMC that was based on faulty command line input. |
| 6 | Resource specified was not found.                                    |
| 7 | Resource already exists.                                             |

## Security

This command requires root authority.

## Examples

1. To create an equivalency using the data input file, do the following:
  - a. To generate a template to aid in the defining of these resources in a file, enter:
 

```
lsrsrdef -i IBM.Equivalency >& /tmp/Equ.rdef
```
  - b. Edit the file `/tmp/Equ.rdef` with your preferred text editor:(Enter values for all of the attributes, substituting an appropriate value for the type, or leave it blank if you want the default value.)
  - c. Run the **mkequ** command with the file as follows:
 

```
mkequ -f /tmp/Equ.rdef
```

### Sample Data Input file for mkequ:

You can enter values for all of the attributes and thus change the value shown below, or remove the line with the attribute if you want the default value.

#### PersistentResourceAttributes:

##### resource 1:

```
Name = "Equiv"
MemberClass = "IBM.NetworkInterface"
Membership = "{0x000..."
```

##### resource 2:

```
Name = "Equiv2"
MemberClass = "IBM.Application"
Membership = "{App1,App2,App3}"
SelectFromPolicy = "Any"
MinimumNecessary = 2
```

##### resource 3:

```
Name = "Equiv3"
MemberClass = "IBM.Test"
SelectString = "Name ?="Test""
SelectFromPolicy = "uint32"
MinimumNecessary = "uint32"
```

2. To make an equivalency called **foo** with all the members of the resource class `IBM.Application`, enter:
 

```
mkequ -S 'Name like "%"' foo IBM.Application
```
3. To make an equivalency called **foo** with selected members **tester**, **Jfoo**, and **Dfoo** of the resource class `IBM.Application`, and set its policy to ordered, enter:
 

```
mkequ -p 0 foo IBM.Application:tester,Jfoo,Dfoo
```
4. To make an equivalency called **Jfoo** with selected members of the resource class `IBM.Application` by using a static select string, enter:
 

```
mkequ -S "ResourceType==0" Jfoo IBM.Application
```
5. To make an equivalency called **Jfoo** with selected members of the resource class `IBM.Application` using a dynamic select string, enter:
 

```
mkequ -D "ResourceType==0" Jfoo IBM.Application
```

## Files

- |                                       |                                                                             |
|---------------------------------------|-----------------------------------------------------------------------------|
| <code>/usr/sbin/rsct/bin/mkequ</code> | Location of the <b>mkequ</b> command.                                       |
| <code>data_input_file</code>          | See the examples, and see the man page for <b>Resource_Data_Input</b> file. |

**mkequ**

## **See Also**

The **chequ**, **chrgmbr**, **lsequ**, **mkrel**, **rmequ** commands.

The **rmcli** General Information file

The **Resource\_Data\_Input** file.

---

## mkrel

### Name

**mkrel** – Makes a managed relationship between resources.

### Synopsis

**mkrel** [-h] -f *data\_input\_file* [-T] [-V]

**mkrel** [-h] [-p *relationship*] [-o *condition*] -S *Source\_class* [:*Resource\_name*[:*Node*]] [-s "*Source\_selection\_string*"] -G *Target\_class*[:*Resource\_name*[:*Node*]][,*Target\_class*[:*Resource\_name*[:*Node*]][,...]] [ -g "*Target\_selection\_string*" ] [-T] [-V] [*Managed\_Relation*]

### Description

The **mkrel** command makes a relationship between resources. The relationship is between a source resource and one or more target resources. The source resource must be a member of a resource group. A target resource does not have to be in a resource group.

Managed relationships and their attribute flags can be specified on the command line as the syntax indicates or from a file when used with the -f option. The value must be the same data type that is defined for this resource attribute. Use the **lsrsrcdef** command to verify the data type and attribute field for each attribute.

Although optional, it is recommended to specify a name for the managed relationship.

### Options

-o *condition*

Specifies an if condition to be used when a LocateWith relationship (Collocated, AntiCollocated, Affinity, or AntiAffinity) is defined. The value can be a numeric value or a word (not case sensitive):

**0x0000 NoCondition**

Specifies unconditional location relationship.

**0x0001 or IfOnline**

Specifies that the target resource is online.

**0x0002 or IfOffline**

Specifies that the target resource is offline, failed offline, or unknown.

**0x0003 or IfNotOnline**

Specifies that the target resource is not online.

**0x0004 or IfNotOffline**

Specifies that the target resource is neither offline nor failed offline.

-f *data-input\_file*

Specifies the name of the file that contains managed-relationship attribute information.

-g *Target\_selection\_string*

Specifies the target selection string, which is applied to all resources of the resource class specified by the -G option to determine the targets of the relationship.

If the -g option is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for the -G option.

-G *Target\_class*[:*Resource\_name*[:*Node*]]

## mkrel

Specifies the target resources of the relationship. A target resource is identified by the resource class (*Target\_class*) of the resource, the resource name (*Resource\_name*), and, optionally, the node (*Node*) at which the resource resides. *Target\_class*, *Resource\_name*, and *Node* must be separated by colons. Multiple target resources can be specified and are separated by commas. The colons are required, but omitting *Target\_class* or *Resource\_name* causes the previously specified class or name to be used when multiple target resources are specified.

If the **-g** option is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for the **-G** option.

**-h** Writes the command's usage statement to standard output.

**-p** *relationship*

Specifies the managed relationship to apply. The value can be a numeric value or a word (not case sensitive):

**0x0001 or Collocated**

Specifies that the source and the target resources are to be located on the same node.

**0x0002 or AntiCollocated**

Specifies that the source and the target resources are to be located on different nodes.

**0x0003 or Affinity**

Specifies that the source resource and its specified affinity resource are to be located on the same node if possible.

**0x0004 or AntiAffinity**

Specifies that the source resource and its specified affinity resource are to be located on different nodes if possible.

**0x0005 or IsStartable**

Specifies that the source and its target resources are startable.

**0x0006 or StartAfter**

Specifies that the source is to be started after the target resources.

**0x0007 or DependsOn**

Specifies that the target resources must be online before the source resource is started and includes an implicit collocation between the source and target resources.

**0x0008 or DependsOnAny**

Specifies that the target must be online before the source resource is started. It is identical to the DependsOn relationship except that it does not provide the collocated constraint for the start sequence. Therefore the source and target resources may not be started on the same node.

**0x0009 or StopAfter**

Specifies that the source resource may not be stopped until after the target resource has been brought offline.

**0x000A or ForcedDownBy**

Specifies that the source resource must be forced offline in the event that either the target resource goes offline unexpectedly or the target resource itself is forced offline.

**-s** *Source\_selection\_string*

Specifies the source selection string, which is applied to all resources of the resource class specified by the **-S** option to determine the source of the relationship. The result of the selection must contain only one resource.

If the **-s** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-S** option.

**-S** *Source\_class[:Resource\_name[:Node]]*

Specifies the source resource of the relationship. A source resource is identified by the resource class (*Source\_class*) of the resource, the resource name (*Resource\_name*), and, optionally, the node (*Node*) at which the resource resides. *Source\_class*, *Resource\_name*, and *Node* must be separated by colons.

If the **-s** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-S** option.

- T** Writes the command's trace messages to standard error. For your software-service organization's use only.
- V** Writes the command's verbose messages to standard output.

## Parameters

### *Managed\_Relation*

Optional name of the managed relationship.

## Exit Status

- 0** Command has run successfully.
- 1** Error occurred with RMC.
- 2** Error occurred with CLI script.
- 3** Incorrect flag on command line.
- 4** Incorrect parameter on command line.
- 5** Error occurred with RMC that was based on faulty command line input.
- 6** Resource specified was not found.
- 7** Resource already exists.

## Security

This command requires root authority.

## Examples

1. To create a new relationship using the data input file, do the following:
  - a. To generate a template to aid in the defining of these relationships in a file, enter:
 

```
lsrsrcdef -i IBM.ManagedRelationship > /tmp/Rel.rdef
```
  - b. Edit the file */tmp/Rel.rdef* with your preferred text editor:(Enter values for all of the attributes, substituting an appropriate value for the type or leaving it blank if you want the default value.)
  - c. Run the **mkrel** command with the file as follows:

```
mkrel -f /tmp/Rel.rdef
```

### Sample Data Input file for mkrel:

You can enter values for all of the attributes and thus change the value shown below, or remove the line with the attribute if you want the default value.

#### PersistentResourceAttributes:

##### resource 1:

```
Name = "Rel1"
Source = "0x606b 0xffff 0x0fe8018b 0x3343381b 0x0e8341f8 0x7da8a450"
Target = {"0x606b 0xffff 0x0fe8018b 0x3343381b 0x0e8341fa 0x42a72de8", "0x...."}
Relationship = anticollated
Conditional =i offline
```

## mkrel

### resource 2:

Source = IBM.Application:App1  
Target = "{IBM.NetworkInterface:eth0:node2,IBM.Application:App0}"  
Relationship = collocated

2. To make a relationship named **myrel** for a resource **tester** that belongs to resource class IBM.Application with target resources from resource class IBM.Application, whose ResourceType is set to one, enter:

```
mkrel -p collocated -S IBM.Application:tester -G IBM.Application -g "ResourceType==1" myrel
```

3. To make an Affinity relationship called **test** for a resource **tester** belonging to resource class IBM.Application with target resource **tester1** belonging to resource class IBM.Application and target resources **Jfoo** belonging to resource class IBM.Application, enter:

```
mkrel -p affinity -S IBM.Application:tester -G IBM.Application:tester1,Jfoo test
```

4. To define a Collocated relationship named **myrel** for a source resource **narten** of class IBM.Application to target resources **tr0** and **en0** of class IBM.ServiceIP, enter:

```
mkrel -p 1 -S IBM.Application:narten -G IBM.ServiceIP:tr0,:en0 myrel
```

5. To define a Collocated relationship named **myrel** for a source resource from a selection string applied to the class IBM.Application with target resources **tr0** and **en0** of class IBM.ServiceIP, enter:

```
mkrel -p collocated -S IBM.Application -s 'Name=="narten"' -G IBM.ServiceIP:tr0,:en0 myrel
```

## Files

**/usr/sbin/rsct/bin/mkrel** Location of the **mkrel** command.

*data\_input\_file* See the examples, and see the man page for **Resource\_Data\_Input** file.

## See Also

The **addrgmbbr**, **chequ**, **chrel**, **chrg**, **chrgmbr**, **lsrg**, **mkequ**, **mkrq**, **rmequ**, **rmrel**, **rmrg**, **rmrgmbr** commands.

The **rmcli** General Information file

The **Resource\_Data\_Input** file.

---

## mkrp

### Name

**mkrp** – Makes one or more new resource groups.

### Synopsis

**mkrp** [-h] -f *data\_input\_file* [-T] [-V]

**mkrp** [-h] [-l *relationship*] [-n *node\_name* | -e *equiv\_name*] [-p *priority*] [-T] [-V] [-x *node1, ...noden*] *Resource\_group* [...]

### Description

The **mkrp** command defines (makes) one or more new resource groups that can be monitored and controlled by IBM Tivoli System Automation. A resource group can only be defined on an online node in a cluster. A resource group definition defines the following to IBM Tivoli System Automation:

- Where the resource group is allowed to run,
- Relative importance of resource group to other resource groups, and
- Location relationship among the member resources of the resource group.

The new resource groups will default to an Offline state. This is to allow a user or administrator to fully configure the resource group and its resources, before allowing IBM Tivoli System Automation to take control of the resource group.

To allow IBM Tivoli System Automation to monitor and control the resource groups, the resource group must be in Online State or condition. See the **chrg** command to set this or any other option on an existing resource group.

The resource group and its attribute options can be specified on the command line as the syntax indicates or from a file when the -f option is used. The value must be the same data type that is defined for this resource attribute. Use the **lsrsrcdef** RMC command to verify the data type and attribute field for each attribute.

### Options

-e *equiv\_name*

Specifies the equivalency name of the nodes on which the *Resource\_group* can run. This option cannot be combined with the -n option. The default value is ALL, which implies an equivalency of all nodes in the cluster.

-f *data\_input\_file*

Name of the file which contains resource group attribute information.

-h Writes the command's usage statement to standard output.

-l *relationship*

Specifies the location relationship among the members of the resource group. The location relationship value can be:

**0x0000 or None**

Specifies that member resources of the resource group can be on any node

**0x0001 or Collocated**

Specifies that member resources of the resource group are to be located on the same node.

The default value is **Collocated**.

## mkrg

### **-n** *node\_name*

Specifies the node on which the *Resource\_group* can run. This option cannot be combined with the **-e** option. The default value is ALL, which implies all nodes in the cluster.

### **-p** *priority*

Specifies the relative importance of this resource group compared to other resource groups. Priority can be any integer from 0 to 200: the higher the integer, the higher the priority. The default priority value is 0.

**-T** Writes the command's trace messages to standard error. For your software-service organization's use only.

**-V** Writes the command's verbose messages to standard output.

### **-x** *node1, ...noden*

Specifies the excluded node list.

## Parameters

### *Resource\_group*

The unique name of the new resource group(s) to be created.

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To create resource groups using the data input file, do the following:
  - a. To generate a template to aid in the defining of these resources in a file, enter:

```
lsrsrcdef -i IBM.ResourceGroup >& /tmp/RG.rdef
```
  - b. Edit the file */tmp/RG.rdef* with your preferred text editor: (Enter values for all of the attributes, substituting the appropriate value for the type, or remove the line with the attribute if you want the default value.)
  - c. Run the **mkrg** command with the file as follows:

```
mkrg -f /tmp/RG.rdef
```

### **Sample Data Input file for mkrg:**

You can enter values for all of the attributes and thus change the value shown below, or remove the line with the attribute if you want the default value.

#### **PersistentResourceAttributes:**

##### **resource 1:**

Name = InputFile1a

```
MemberLocation = "1"
Priority = "32"
NominalState = "1"
```

**resource 2:**

```
Name = InputFile1b
MemberLocation = collocated
NominalState = offline
AllowedNode = narten
```

**resource 3:**

```
Name = InputFile1c
MemberLocation = 1
Priority = 2
NominalState = offline
AllowedNode = "0x00010001 0x00000000 0x0069684c 0x0d065e26 0x30060ab9"
```

2. To define a new resource group called **foo** with all default values, enter:

```
mkrp foo
```

3. To define new resource groups called **foo** and **foo1** with location relationship "none", and allowed node name "cluster1", enter:

```
mkrp -l none -n cluster1 foo foo1
```

4. To define a new resource group called **foo** with location relationship "Collocated", , priority value of 22, and allowed equivalency name "cluster1", enter:

```
mkrp -l collocated -e cluster1 -p 22 foo
```

## Files

**/usr/sbin/rsct/bin/mkrp** Location of the **mkrp** command.

*data\_input\_file*

## See Also

The **addrgmbr**, **samctrl**, **chrg**, **chrgmbr**, **lsrg**, **rmrg**, **rmrgmbr** commands.

The **Resource\_Data\_Input** file.

The **rmcli** General Information file.

---

## rgmbrreq

### Name

**rgmbrreq** – Requests for a managed resource to be started, stopped, or cancelled.

**rgmbrreq** belongs to a group of commands (rgreq, rgmbrreq, and lsrgreq) which allow an operator to introduce persistent requests into IBM Tivoli System Automation. Using this set of commands the operator can start, stop, cancel, or move resource groups and managed resources.

### Synopsis

```
rgmbrreq [-h] [-p priority] -o action [-S source] [-T] [-V]
Resource_class:Resource[:Node][,Resource[:Node][,...]]
[Resource_class:Resource[:Node][,Resource[:Node][,...]][...]]
```

```
rgmbrreq [-h] [-p priority] -o action -s [-S source] [-T] [-V]
Resource_class:"Selection_String" [Resource_class:"Selection_String" [...]]
```

### Description

The **rgmbrreq** command allows requests into IBM Tivoli System Automation against managed resources. By means of the **rgmbrreq** command the operator or any originator (source) asks for the managed resource to be started, stopped, or cancelled. Start and Stop requests will last until explicitly removed or cancelled by the operator. A request coming from the same source replaces a former request from that source - that is, it cancels the previous request automatically. A move request is not allowed for a managed resource. A stop or a start request can be cancelled by the same source that enabled it.

### Options

**-h** Help. Writes the command's usage statement to standard output.

**-p** *priority*

Determines how important the request is with regard to other requests within the resource structure. The priority allows IBM Tivoli System Automation to solve conflicting requests for the same resource. Can be one of the following:

**low** low priority. This is the default.

**high** high priority.

**force** takes precedence of request posted with high priority.

**-o** *action*

is the actual action request. It can be one of the following:

**start** sets the nominal state of the resource to online.

**stop** sets the nominal state of the resource to offline.

**cancel**

cancels a previously entered request. The request is identified by the source of the request (-S parameter) and the resource name.

**-s** Specifies that a selection string will be used.

**-S** *source*

Identifies the originator of the request. This can be one of the following strings:

- Operator
- ExtSched
- Automation

Any others will be allowed, but will get the lowest priority. If source is not specified then, it will default to "Operator". When cancelling the request, the same source string must be specified. Note that each source can only have one active request of each type against each resource. If the source makes a second request directly against the resource, it will replace their first request. The source option allows the installation to identify the different organizational units and the roles they are playing. Another usage is to distinguish between who made or where the request came from, for example from the operator versus an automation shell script.

- T Trace. Writes the command's trace messages to standard error. For your software-service organization's use only.
- V Verbose. Writes the command's verbose messages to standard output.

## Parameters

*Resource\_class:"selection\_string"*

The **-s** option flag determines this operand. Each selection string must be preceded by a resource class. *Resource\_class* indicates the name of the Resource class the *selection\_string* will be applied to. The *selection\_string* and its *Resource\_class* must be separated by colon or ":" delimiter. The specified selection string will be applied to its corresponding *Resource\_class* attributes to determine which member resources are to be removed from the *Resource\_group*. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

- s IBM.Application:`Name=="testing"``
- s IBM.Application:`Name ?="test"``
- s IBM.Application:`Name like "%"`` (For all resources)

For information on how to specify selection strings, see "Using expressions" on page 222.

*Resource\_class:Resource[:Node]*

Member Resources. This specifies one or more member resources that are to be acted on. Resources belonging to different resource classes can also be specified using this syntax. However, member resources belonging to different classes must be separated by spaces. A member resource is identified by the resource's class ( *Resource\_class*), the resource's name ( *Resource\_name*), and optionally the node (*Node*) the resource is on. The *Resource\_class*, *Resource* and *Node* must be separated by a colon. *Resource\_class* is the name of the resource class the member resource belongs to. *Resource* is the name of the actual member resource in its class. The node may have to be included when trying to act on a fixed member resource. The node and its resource must be separated by the colon or ":" delimiter. Multiple resources belonging to the same resource class can also be specified and are separated by a comma:

*Resource\_class:Resource[:Node][,Resource[:Node][,...]]*

The member resources must exist in the resource group for the requested action to take place.

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.

## **rgmbrreq**

7 Resource already exists.

## **Security**

This command requires root authority.

## **Examples**

1. To request a start action on application App1 of IBM.Application on Node1, enter:

```
rgmbrreq -o start IBM.Application:App1:Node1
```

## **Files**

**/usr/sbin/rsct/bin/rgmbrreq** Location of the **rgmbrreq** command.

## **See Also**

The **addrgmbr**, **chrg**, **chrgmbr**, **lsrg**, **lsrgreq**, **mkrg**, **rgreq**, **rmrg**, and **rmrgmbr** commands.

---

## rgreq

### Name

**rgreq** – Requests for a resource group to be started, stopped, cancelled, or moved.

**rgreq** belongs to a group of commands (**rgreq**, **rgmbrreq**, and **lsrgreq**) which allow an operator to introduce persistent requests into IBM Tivoli System Automation. Using this set of commands the operator can start, stop, cancel, or move resource groups and managed resources.

### Synopsis

**rgreq** [-h] [-p *priority*] [-o *action*] [-S *source*] [-n *node1,....noden*] [-T] [-V] *Resource\_group*

### Description

The **rgreq** allows requests into IBM Tivoli System Automation against resource groups. The operator or any originator (also called source) asks for the resource group to be started, stopped, cancelled or moved. Start and Stop requests will last until explicitly removed or cancelled by the operator. A request coming from the same source replaces a former request from that source - it cancels the previous request automatically. Start and Stop requests cancel each other, but Move requests do not. Since Move and Start/Stop request are of different types, both a Move request and a Start or a Stop request from the same source against the same resource may be allowed. For example, request a resource group to start, then request it to move. When the resource group has been moved, the Start request should still be in place. A Move request is automatically removed when the move action is carried out or cancelled by the recovery resource manager (IBM.RecoveryRM). A Move request cannot be cancelled by any source. Stop and Start requests may be cancelled at any time. However, the cancel requests must come from the same source as the original Start and Stop requests.

### Options

**-h** Help. Writes the command's usage statement to standard output.

**-p** *priority*

Determines how important the request is with regard to other requests within the resource structure. The priority allows IBM Tivoli System Automation to solve conflicting requests for the same resource. Can be one of the following:

**low** low priority. This is the default.

**high** high priority.

**force** takes precedence of request posted with high priority.

**-o** *action*

is the actual action request. It can be one of the following:

**start** sets the nominal state of the resource to online.

**stop** sets the nominal state of the resource to offline.

**move** moves the resource group and its underlying resources to another node in the cluster. This is the process of stopping an active set of resources and starting them on a different node(s) in a coordinated fashion.

**cancel**

cancels a previously entered request. The request is identified by the source of the request (-S parameter) and the resource name. Unlike the start/stop the move action cannot be cancelled.

**-S** *source*

identifies the originator of the request. This can be one of the following strings:

## rgreq

- Operator
- ExtSched
- Automation

Other originators are allowed, but will get the lowest priority. If source is not specified then, it will default to "Operator". When cancelling the request, the same source string must be specified. There are two types of request. Start and Stop requests are of one type, and the Move request is of another type. Note that each source can only have one active request of each type against each resource. If the source makes a second request directly against the resource, it will replace their first request. The source option allows the installation to distinguish where the request came from, for example from the operator versus an automation shell script.

### **-n** *node1,..noden*

The originating nodes. This option can only be applied for a move action. It identifies the nodes to move the resource group or its underlying resources from. This option is required for an anticollocated resource group.

### **-T** Trace. Writes the command's trace messages to standard error. For your software-service organization's use only.

### **-V** Verbose. Writes the command's verbose messages to standard output.

## Parameters

### *Resource\_group*

the name of the resource group. This is the entry point at which the request is introduced into the resource structure and from which it propagates outwards.

## Exit Status

- |   |                                                                      |
|---|----------------------------------------------------------------------|
| 0 | Command has run successfully.                                        |
| 1 | Error occurred with RMC.                                             |
| 2 | Error occurred with CLI script.                                      |
| 3 | Incorrect flag on command line.                                      |
| 4 | Incorrect parameter on command line.                                 |
| 5 | Error occurred with RMC that was based on faulty command line input. |
| 6 | Resource specified was not found.                                    |
| 7 | Resource already exists.                                             |

## Security

This command requires root authority.

## Examples

1. To move all the resources from node1 in resource group RG1, enter:  

```
rgreq -n node1 -o move RG1
```
2. To stop all the resources in resource group RG2, enter:  

```
rgreq -o stop RG2
```

## Files

**/usr/sbin/rsct/bin/rgreq** Location of the **rgreq** command.

## See Also

The `lsrgreq`, `mkrq`, `chrg`, `rmrg`, `lsrg` commands.

## rmequ

### Name

**rmequ** – Removes one or more already-defined resource equivalencies.

### Synopsis

**rmequ** [-h] [-i] -s "*selection\_string*" [-T] [-V] [*Equivalency* [...]]

**rmequ** [-h] [-i] [-T] [-V] *Equivalency* [...]

### Description

The **rmequ** command removes one or more resource equivalencies specified by *Equivalency* parameters or equivalencies derived from selection strings.

### Options

- h Writes the command's usage statement to standard output.
- i Interactive. Prompt before removing equivalencies.
- s "*selection\_string*"  
Specifies the equivalency selection string. This *selection\_string* is applied to all the existing equivalencies when an equivalency parameter is not specified. When equivalency parameters are specified, then the specified selection string is only applied to the specified equivalency parameters. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:  

```
-s 'Name == "testing"'
```

```
-s 'Name ?= "test"'
```

  
For information on how to specify selection strings, see "Using expressions" on page 222.
- T Writes the command's trace messages to standard error. For your software-service organization's use only.
- V Writes the command's verbose messages to standard output.

### Parameters

#### *Equivalency*

Specifies the name of the defined resource equivalency to be removed. Multiple equivalencies may be specified, but must be separated by spaces.

### Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.

7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To remove an equivalency called **foo**, enter:  
rmequ foo
2. To remove equivalencies called **foo**, **foo1**, and **foo2**, enter:  
rmequ foo foo1 foo2

## Files

`/usr/sbin/rsct/bin/rmequ` Location of the **rmequ** command.

## See Also

The **chequ**, **samctrl**, **lsequ**, **mkequ** commands.

The **rmcli** General Information file

---

## rmrel

### Name

**rmrel** – Removes a managed relationship between resources,

### Synopsis

**rmrel** [-h] [-i] **-s** "selection\_string" [-T] [-V]

**rmrel** [-h] [-i] **-S** *Source\_Class[:Resource\_name[:Node]]* [**-s** "Source\_string"]  
[-G *Target\_Class[:Resource\_name[:Node]][,Target\_class[:Resource\_name[:Node][. ...]]]*]**-g** "Target\_string" [-P *Current\_relationship*]  
[-O *Current\_condition*] [-T] [-V]

**rmrel** [-h] [-i] [-T] [-V] *Managed\_Relation* [*Managed\_Relation ...*]

### Description

The **rmrel** command removes a relationship between resources. The relationship to be removed is determined by the source parameter (*Source\_class[:Resource\_name[:Node]]*) or by the selection string when it is applied to the relationship itself or the name(s) of the managed relationship(s).

### Options

#### **-O** *Current\_condition*

Specifies the condition value to query all defined relationships, in order to find a match. The value can be the numeric value or the word (not case sensitive):

##### **0x0000 NoCondition**

Specifies unconditional location relationship.

##### **0x0001 or IfOnline**

Specifies that the target resource is online.

##### **0x0002 or IfOffline**

Specifies that the target resource is offline, failed offline, or unknown.

##### **0x0003 or IfNotOnline**

Specifies that the target resource is not online.

##### **0x0004 or IfNotOffline**

Specifies that the target resource is neither offline nor failed offline.

#### **-g** *Target\_string*

The *Target\_string* is applied to all resources of the resource class specified by the **-G** flag, to determine the targets of the relationship.

If the **-g** option is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for the **-G** option .

#### **-G** *Target\_class[:Resource\_name[:Node]]*

Specifies the target resource to be used with the target query, and any other specified query, to find a match from all defined relationships. A target resource is identified by the class of the resource (*Target\_class*), the name of the resource (*Resource\_name*), and optionally the node (*Node*) on which the resource is located. The *Resource\_class*, *Resource\_name* and *Node* must be separated by a colon. Multiple target resources can be specified and are separated by a comma. The colons are required, but if the *Target\_class* is omitted, the previously-specified class will be used when specifying multiple target resources.

If the **-g** option is used to specify a target selection string, *Resource\_name* and *Node* must not be specified for the **-G** option.

**-h** Writes the command's usage statement to standard output.

**-i** Interactive. Prompt before removing relationships.

**-P** *Current\_relationship*

Specifies the managed relationship value used to query all the defined relationships to find a match. The value can be a numeric value or the word (not case sensitive):

**0x0001 or Collocated**

Specifies that the source and the target resources are to be located on the same node.

**0x0002 or AntiCollocated**

Specifies that the source and the target resources are to be located on different nodes.

**0x0003 or Affinity**

Specifies that the source resource and its specified affinity resource are to be located on the same node, if possible.

**0x0004 or AntiAffinity**

Specifies that the source resource and its specified affinity resource are to be located on different nodes, if possible.

**0x0005 or IsStartable**

Specifies that the source and its target resources are startable.

**0x0006 or StartAfter**

Specifies that the source is to be started after the target resources.

**0x0007 or DependsOn**

Specifies that the target resources must be online before the source resource is started and includes an implicit collocation between the source and target resources.

**0x0008 or DependsOnAny**

Specifies that the target must be online before the source resource is started. It is identical to the DependsOn relationship except that it does not provide the collocated constraint for the start sequence. Therefore the source and target resources may not be started on the same node.

**0x0009 or StopAfter**

Specifies that the source resource may not be stopped until after the target resource has been brought offline.

**0x000A or ForcedDownBy**

Specifies that the source resource must be forced offline in the event that either the target resource goes offline unexpectedly or the target resource itself is forced offline.

**-s** *"selection\_string"*

Specifies the selection string. If the **-s** option is specified without the **-S** option, the selection string associated with it will be applied to all managed relationships of resources. The selection string must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
```

For information on how to specify selection strings, see "Using expressions" on page 222.

**-s** *Source\_string*

Specifies the source selection string. The *Source\_string* is applied to all resources of the resource class specified by the **-S** option to determine the source of the relationship. The result of the selection may contain one or more resources.

## rmrel

If the **-s** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-S** option .

**-S** *Source\_class[:Resource\_name[:Node]]*

Specifies the source resource of the query. This source resource, with any other query that may be specified, will be queried against all defined relationship to find a match. A source resource is identified by the class of the resource ( *Source\_class*), the name of the resource (*Resource\_name*), and optionally the node ( *Node*) on which the resource is located. The *Resource\_class*, *Resource\_Name* and *Node* must be separated by a colon.

If the **-s** option is used to specify a source selection string, *Resource\_name* and *Node* must not be specified for the **-S** option.

**-T** Writes the command's trace messages to standard error. For your software-service organization's use only.

**-V** Writes the command's verbose messages to standard output.

## Parameters

### *Managed\_Relation*

Name of the managed relationship to be removed. Multiple relationships can be specified on the command line, but have to be separated by spaces.

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To remove a relationship for a resource **tester** that belongs to resource class IBM.Application, enter:  

```
rmrel -S IBM.Application:tester
```
2. To remove the relationship of a resource by using a selection string, enter:  

```
rmrel -s 'Name == "tester"'
```
3. To remove a relationship for a source resource **foo** from a selection string applied to the class IBM.Application, enter:  

```
rmrel -s 'Name=="foo"' -S IBM.Application
```
4. To remove all relationship for source resources from class IBM.Application:  

```
rmrel -s 'Name like "%"' -S IBM.Application
```
5. To remove relationship for resources with targets from class IBM.Application with a collocated relationship and a specific condition, enter:  

```
rmrel -S IBM.Application:narten -G IBM.Application:ha,:test -P collocated -O 1
```

6. To remove relationships named **narten** and **foo**, enter:

```
rmrel narten foo
```

## Files

**/usr/sbin/rsct/bin/rmrel** Location of the **rmrel** command.

## See Also

The **addrgmbr**, **chequ**, **chrel**, **chrg**, **chrgmbr**, **lsrg**, **mkequ**, **mkrel**, **mkrg**, **rmequ**, **rmrg**, **rmrgmbr** commands.

The **rmcli** General Information file.

---

**rmrg****Name**

**rmrg** – Removes one or more already-defined resource groups.

**Synopsis**

**rmrg** [-h] [-i] -s "*selection\_string*" [-T] [-V] [*Resource\_group* ...]

**rmrg** [-h] [-i] [-T] [-V] *Resource\_group* [...]

**Description**

The **rmrg** command removes one or more resource groups specified by the *Resource\_group* parameter, or that match the specified selection string. The member resources associated with the removed resource groups, are also removed by IBM Tivoli System Automation. If any member of the resource groups to be deleted are part of a managed relationship between resources, the managed relationships are also removed by IBM Tivoli System Automation. The resource group must already exist for it to be removed. If the resource group to be removed is still online, the resource group is not removed.

When the -h option is specified, this command's usage statement is written to standard output. All verbose messages are also written to standard output.

All trace messages are written to standard error.

**Parameters***Resource\_group*

The name of the defined resource group to be removed. Multiple resource groups can be specified but must be separated by spaces.

**Options**

- h Writes the command's usage statement to standard output.
- i Interactive. Prompt before removing relationships.
- s "*selection\_string*"  
Specifies the selection string. If the *Resource\_group* parameters are not specified, the selection string will be applied to all existing resource groups. The selection string must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:  

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
-s 'Name like "%"' (For all resources)
```
- T Writes the command's trace messages to standard error. For your software-service organization's use only.
- V Writes the command's verbose messages to standard output.

**Exit Status**

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.

- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. To remove a resource group called **foo**, enter:  

```
rmrg foo
```
2. To remove resource groups called **foo**, **foo1**, and **foo2**, enter:  

```
rmrg foo foo1 foo2
```
3. To prompt before removing resource groups called **foo** and **foo2**, enter:  

```
rmrg -i foo foo2
```
4. To remove a resource group called **foo** using a select string, enter:  

```
rmrg -s 'Name=="foo"'
```
5. To remove all resource groups containing the word **foo** as part of its name, enter:  

```
rmrg -s 'Name ?="foo"'
```

## Files

**/usr/sbin/rsct/bin/rmrg** Location of the **rmrg** command.

## See Also

The **addrgubr**, **samctrl**, **chrg**, **chrgubr**, **lsrg**, **mkrg**, **rmrgubr** commands.

The **rmccli** General Information file

---

## **rmrgmbr**

### **Name**

**rmrgmbr** – Removes one or more resources from their resource group.

### **Synopsis**

**rmrgmbr** [-h] [-i] [-T] [-V] -g *Resource\_group*

**rmrgmbr** [-h] [-i] -s [-T] [-V] [-g *Resource\_group*] "*selection string*"

**rmrgmbr** [-h] [-i] -s [-T] [-V] [-g *Resource\_group*]  
*Resource\_class:"selection\_string"* [*Resource\_class:"selection\_string"* [...]]

**rmrgmbr** [-h] [-i] [-T] [-V] [-g *Resource\_group*]  
*Resource\_class:Resource\_name* [:Node][,*Resource\_name*[:Node][,...]]  
[*Resource\_class:Resource\_name* [:Node][,*Resource\_name*[:Node][,...]] [...]]

### **Description**

The **rmrgmbr** command removes all the member resources of the specified resource group, only the specified member resources of the specified resource group, or the member resources that match the selection string. IBM Tivoli System Automation ensures that the associated relationship and equivalency are also updated, if possible. When the selection string is the only parameter, then it is applied directly to the IBM.ManagedResource class.

A resource name or selection string must be specified with the resource class to which it belongs. The user must input the resource class, resource name, and the node number or name if it is a fixed resource. The resource class and the resource name, the resource name and the node must be separated by a colon.

### **Options**

- h Writes the command's usage statement to standard output.
- i Interactive. Prompt before removing relationships.
- g *Resource\_group*  
Specifies the unique name of the resource group from which the member resources are to be removed.
- s Specifies that a selection string will be used instead of the resource name.
- T Writes the command's trace messages to standard error. For your software-service organization's use only.
- V Writes the command's verbose messages to standard output.

### **Parameters**

#### *Resource\_class*

Specifies the name of the resource class to which the member resource belongs. The resource and the resource class must be separated by the colon (:) delimiter.

#### *Resource\_class:Resource\_name[:Node]*

This specifies one or more member resources that are to be removed from the specified resource group. Resources belonging to different resource classes can also be specified using this syntax. However, member resources belonging to different classes must be separated by spaces. A member resource is identified by the resource's class (*Resource\_class*), the resource's name (*Resource\_name*), and optionally the node (*Node*) the resource is on. The *Resource\_class*,

Resource\_Name and Node must be separated by a colon. Resource\_class is the name of the resource class the member resource belongs to. Resource\_name is the name of the actual member resource in its class. The Node may have to be included when trying to remove a fixed resource from the resource group. The Node and its resource must be separated by the colon (:) delimiter.

Multiple resources belonging to the same resource class can also be specified and are separated by a comma. In this case the first resource or Resource\_name[:Node] of a resource class must be preceded by the Resource\_class it belongs to and it must be separated by the colon (:) delimiter while the remaining resources are separated by a comma.

*Resource\_class:Resource\_name[:Node]*

The member resources must exist in the resource group for it to be removed.

#### *Resource\_class:"selection\_string"*

Selection string. The -s option flag determines this operand. Each selection string must be preceded by a resource class. *Resource\_class* indicates the name of the resource class the selection\_string will be applied to. The selection\_string and its Resource Class must be separated by the colon (:) delimiter. The specified selection string will be applied to its corresponding resource\_class attributes to determine which member resources are to be removed from the resource\_group. The selection string must be enclosed within double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
-s 'Name like "%"' (For all resources)
```

For information on how to specify selection strings, see “Using expressions” on page 222.

#### *Selection\_string*

Specifies the selection string. The -s option determines this parameter. When this is the only parameter, then the selection string is applied to the member resources. But, when the selection string is included with *Resource\_group*, the selection string will be applied to the member resources that belong to the specified resource group. When each of the selection strings are preceded by a resource class, the specified selection string in this case will be applied to its corresponding *Resource\_class* attributes, to determine which resources are to be removed from the Resource\_group. The selection string must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
-s 'Name like "%"' (For all resources)
```

For information on how to specify selection strings, see “Using expressions” on page 222.

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## rmrgmbr

### Security

This command requires root authority.

### Examples

1. To remove a member resource **tester** that belongs to resource class IBM.Application from a resource group **foo**, enter:  

```
rmrgmbr -g foo IBM.Application:tester
```
2. To remove member resources **tester**, **Jfoo**, and **Dfoo** that belong to resource class IBM.Application, from resource group **foo**, enter:  

```
rmrgmbr -g foo IBM.Application:tester,Jfoo,Dfoo
```
3. To remove member resources **tester** and **Jfoo** that belong to resource class IBM.Application and fixed resource **en0** that belongs to class IBM.ServiceIP, enter:  

```
rmrgmbr IBM.Application:tester,Jfoo IBM.ServiceIP:en0
```
4. To remove selected member resources of the resource class IBM.Application from a resource group **foo** whose ResourceType is set, enter:  

```
rmrgmbr -s -g foo IBM.Application:"ResourceType==1"
```
5. To remove all member resources of resource group **foo** by using a selection string, enter:  

```
rmrgmbr -s 'MemberOf == "foo"'
```

### Files

**/usr/sbin/rsct/bin/rmrgmbr** Location of the **rmrgmbr** command.

### See Also

The **addrgmbr**, **chrel**, **chrg**, **chrgmbr**, **lsrel**, **lsrg**, **mkrel**, **mkrg**, **rmrel**, **rmrg**, commands.

The **rmccli** General Information file.

---

## samcfg

### Name

**samcfg** – Saves and restores the IBM Tivoli System Automation configuration.

### Synopsis

**samcfg** [-h] [-T] [-V] -S [*Filename* ]

**samcfg** [-h] [-T] [-V] -R [*Filename* ]

### Description

The **samcfg** command saves and restores the configuration of IBM Tivoli System Automation.

### Options

- h Help. Writes the command's usage statement to standard output.
- S Save Action. Saves the current IBM Tivoli System Automation configuration to a file.
- R Restore. Restores the configuration from a file.
- T Trace. Writes the command's trace messages to standard error. For your software-service organization's use only.
- V Writes the command's verbose messages to standard output.

### Parameters

#### *Filename*

Name of the file or absolute path with file name where the configuration will be saved to or restored from. The default path is **/var/ct/{cluster}/cfg/IBM.RecoveryRM**. This file will contain the configuration data. *Filename* is optional for the save operation. The default file name for save will follow the following format:

**{domain}\_samcfg<month><day><yr>.{index}**

where

**domain** domain (cluster) name

**index** file version, current index +1, default is 0

Example: CHARM\_samcfg022703.2, bignet\_samcfg022303.0

The user must provide the file name for the restore operation. If an absolute path is not specified, the file will be expected to be found at the default path specified above.

### Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.

## **samcfg**

6 Resource specified was not found.

7 Resource already exists.

## **Security**

This command requires root authority.

## **Examples**

### **Files**

**/usr/sbin/rsct/bin/samctrl** Location of the **samctrl** command.

### **See Also**

The **Issamctrl** command.

The **rmcli** General Information file

---

## samctrl

### Name

**samctrl** – Sets the IBM Tivoli System Automation control parameters.

### Synopsis

**samctrl** [-h] -m [-f] [-T] [-V] [*new\_activeversion*]

**samctrl**[-h] [-e P|-d P] [-M T|F] [-u a | d | r] [-t *Timeout*] [-r *Retry\_count*]  
[-o *ResourceRestartTimeout*] [-T] [-V] [*Node* [*Node* [...]]]

### Description

The **samctrl** command sets the default control parameter values for IBM Tivoli System Automation. This command, when used, must change at least one control parameter for IBM Tivoli System Automation.

### Options

- e P** Enables the TEC (Tivoli Enterprise Console) publisher. Publisher is enabled.
- d P** Publisher is disabled. This is the default.
- f** Force Migration. It requires the **-m** option. A force option (**-mf**) can be specified if user insists to upgrade the code version when the joined IBM Tivoli System Automation daemon count is less than the total defined node count and the installed version number information of the down daemon is lower than the rest of the up daemons.
- h** Writes the command's usage statement to standard output.
- m** Migration. This is used to trigger a IBM Tivoli System Automation code version migration complete action.
- M T|F**
  - T (True)** Manual. Specifies that all control operations will not be controlled and monitored by System Automation Manager.
  - F (False)** Automated. Specifies that all control operations will be controlled and monitored by System Automation Manager.
- u a | d | r**
  - Specifies that one or more nodes are to be added, removed, or replaced with the excluded list of nodes.
  - a** Adds one or more specified nodes to the excluded list of nodes for control operations.
  - d** Deletes one or more specified nodes from the excluded list of nodes for control operations.
  - r** Replaces one or more specified nodes with the excluded list of nodes for control operations.
- r *Retry\_count***
  - Specifies the default number of retries a control operation can perform before the operation is declared as failed.
- o *ResourceRestartTimeout***
  - Resource Restart Timeout value. Specifies the default timeout value (in seconds) before a failed resource is restarted. IBM Tivoli System Automation waits for the specified period of time before it restarts the resource on another node.

## samctrl

### -t *Timeout*

Specifies the default timeout value (in seconds) of a control operation. A control operation is declared stalled or failed if the operation is not successfully completed within the specified time interval.

**-T** Writes the command's trace messages to standard error. For your software-service organization's use only.

**-V** Writes the command's verbose messages to standard output.

## Parameters

*Node* Specifies the name of one or more nodes that will be added, removed or replaced with the excluded node list for control operations. The **-u** option and its parameter determines the action required to update the excluded node list.

### *new\_activeversion*

This specifies the desired new active version value for IBM Tivoli System Automation. It requires the **-m** option

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred with CLI script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred with RMC that was based on faulty command line input.
- 6 Resource specified was not found.
- 7 Resource already exists.

## Security

This command requires root authority.

## Examples

1. Set IBM Tivoli System Automation to be in automated mode, with retries at 4 , resource restart timeout as 5 seconds.

```
samctrl -M F -r 4 -o 5
```

2. Exclude some nodes from IBM Tivoli System Automation Control:

```
samctrl -u a narten jarden varten
```

3. 3. Migrate to the latest version possible, enter the following:

```
samctrl -m
```

4. 4. To force a migration to the latest version possible, enter the following:

```
samctrl -m -f
```

## Files

**/usr/sbin/rsct/bin/samctrl** Location of the **samctrl** command.

## See Also

The **Issamctrl** command.

The **rmcli** General Information file

---

## samdiag

### Name

**samdiag** – allows to request detailed information on a resource.

### Synopsis

**samdiag** [-h] [-T] [-V] [-x *extra\_info*] -r *Resource\_handle*

**samdiag** [-h] [-T] [-V] [-x *extra\_info*] -g *Resource\_group*

**samdiag** [-h][-T] [-V] [-x *extra\_info*] **Resource\_class:Resource:[Node]**

**samdiag** [-h] [-T] [-V] [-x *extra\_info*] -e *Engine\_name*

### Description

You can request detailed resource information using this command. You can invoke it on any node where a IBM Tivoli System Automation daemon is active.

**Note:** The samdiag command provided for Release 1 of IBM Tivoli System Automation could only be executed on the node where the master daemon was running on. So running samdiag on a Release 2 daemon will generate an error if Release 1 and Release 2 daemons coexist in the same cluster, and if the master daemon is on a Release 1 node.

**samdiag** provides the following information about a specified resource:

- Status
- Dependencies
- Binding constraints
- Outstanding actions
- Flags and exceptions
- Requests and votes
- History
- Variables

### Options

**-h** Writes the command's usage statement to standard output.

**-T** Writes the command's trace messages to standard error. For your software-service organization's use only.

**-V** Writes the command's verbose messages to standard output.

**-x *extra\_info***

Extra Information. This option requests for extra information on the specified resource. The operand "extra\_info" can have the following valid strings:

reqs requests information.

votes votes information.

hist history information.

vars variable information.

all all information.

**-g** *Resource\_group*

The name of the resource group. This implies that a resource group is specified.

**-r** *Resource\_handle*

Resource handle. This option implies that a resource handle of a resource is specified.

**-e** *Engine\_name*

Engine resource name. This option implies that an engine name of a resource is specified. These are the engine name formats currently in use:

- Cluster
- Resource\_name/<type>/Resource\_class[/Node]

**Cluster** This implies that diagnostic information is requested on the cluster or shared domains known to IBM Tivoli System Automation.

**Resource\_name/<type>/Resource\_class[/Node]**

This is the most commonly used engine name format. Here the engine resource is identified by the resource's name ( Resource\_name), the <type> of resource, the resource's class( Resource\_class), and optionally the node ( Node) the resource is on. The Resource\_class, Resource\_Name and Node must be separated by a forward slash ("/"). Resource\_class is the name of the Resource class the resource belongs to. Resource\_name is the name of the actual resource in its class. The Node may have to be included when trying to act on a fixed resource. The Node and its resource must be separated by the forward slash or "/" delimiter. The <type> of the resource can be any of the following:

ResGroup

For resource groups.

Equivalency

For equivalencies.

Fixed

For fixed resources.

Float

For floating resources.

## Parameters

*Resource\_class: Resource\_name[:Node]*

Member Resources. This specifies one resource that is to be acted on. A resource is identified by the resource's class (Resource\_class), the resource's name (Resource\_name), and optionally the node (Node) the resource is on. *Resource\_class*, *Resource\_Name*, and *Node* must be separated by a colon. *Resource\_class* is the name of the Resource class the resource belongs to.

*Resource\_name* is the name of the actual resource in its class. The Node may have to be included when trying to act on a fixed resource. The Node and its resource must be separated by the colon or ":" delimiter.

## Exit Status

- 0 Command has run successfully.
- 1 Error occurred with RMC.
- 2 Error occurred due to an underlying error in the command script.
- 3 Incorrect flag on command line.
- 4 Incorrect parameter on command line.
- 5 Error occurred due to a user error.
- 6 Resource specified was not found.

## **samdiag**

### **Security**

This command requires root authority.

### **Files**

**/usr/sbin/rsct/bin/samdiag** Location of the **samdiag** command.

### **See Also**

The **lssamctrl** command.

The **rmccli** General Information file

---

## samlicm

### Name

**samlicm** – allows to install, list, and upgrade the product license.

### Synopsis

**samlicm** [-h] [-s] [-t] [-i *license\_file*]

### Description

The **samlicm** command installs, lists, and upgrades the product license for IBM Tivoli System Automation. Since every node in the cluster checks for a license, the license has to be installed on every single node in the cluster. Depending on the installation medium a try and buy or full license has already been installed during product installation. Use the **samlicm** command to check your license status or upgrade a try and buy license to a full license.

### Options

- h Writes the command's usage statement to standard output.
- s Prints the license status to standard output. Use this option to see when your license expires.
- t Tests the license. Use this option to check the installed license
- i *license\_file*  
Installs or upgrades the product license with the specified license file. Use the -i option for both, the initial license installation and license upgrade.

### Exit Status

- 0 Command has run successfully.
- 1 Error occurred during command processing.

### Security

This command requires root authority.

### Examples

1. Install a license or upgrade a license:  

```
samlicm -i try_buy.lic
```

If everything works fine you will not get any output from the command. In case of an error the error message will be written to standard error, and the corresponding exit status is returned.

### Files

**/usr/sbin/rsct/bin/samlicm** Location of the **samlicm** command.

### See Also

The IBM License Use Management (LUM) documentation.

## Additional useful commands

### pidmon

#### Name

**pidmon** – Searches process list for command string or takes OpState from file.

#### Synopsis

**pidmon** [-h] [-d *debugmode*] [-u *uid*] [-c] *command\_string*

**pidmon** [-h] [-d *debugmode*] -f *opstate\_file*

#### Description

The **pidmon** command searches the process list for a given command string. If the command string was found, the RMC OpState Online is returned. The **pidmon** command can be used as a generic monitor for the IBM.Application resource class. Optional the RMC OpState can be taken from a specified file.

#### Options

- h Writes the command's usage statement to standard output.
- d *debugmode* Specifies the debug mode.
- u *uid* Specifies a user id for the command string search.
- c Command string search (this is default).
- f OpState is taken from file.

#### Parameters

**debugmode** can be one of the following options:

- 1 Write debug messages to syslog.
- 2 Write debug messages to standard output/error.
- 3 Write detailed debug messages to standard output/error and syslog.

**uid** Searches process list for command string which belongs to the specified user id. RMC\_OPSTATE\_ONLINE is only reported if there is a process with the specified command string and owner of this process is uid.

**opstate\_file** This is a plain text file containing just one of the RMC OpStates shown below. The **pidmon** command reads this text file and returns OpState from the file.

#### Exit Status

Against usual command convention **pidmon** does not return 0 on success or another value on error. It returns the RMC OpState. An exit status of 0 means RMC OpState unknown which indicates an error in the **pidmon** command. Valid RMC OpStates are:

- 0 RMC\_OPSTATE\_UNKNOWN.
- 1 RMC\_OPSTATE\_ONLINE.
- 2 RMC\_OPSTATE\_OFFLINE
- 3 RMC\_OPSTATE\_FAILED\_OFFLINE
- 4 RMC\_OPSTATE\_STUCK\_ONLINE
- 5 RMC\_OPSTATE\_PENDING\_ONLINE
- 6 RMC\_OPSTATE\_PENDING\_OFFLINE

## Security

This command does not require special permissions.

## Examples

1. Searches for the process with the command string ' /bin /bash ' and returns OpState online (exit status 1).

```
pidmon ' /bin/bash/ '
echo $?
1
```

2. Searches for the process with the command string ' /bin /bash ' , but it does not belong to the user id foo. **pidmon** returns OpState offline (exit status 2).

```
pidmon -u foo ' /bin/bash/ '
echo $?
2
```

3. **pidmon** reads myopstate file, which only contains the number '3', and returns OpState failed offline (exit status 3).

```
pidmon -f myopstate
echo $?
3
```

4. Use the **pidmon** command as a generic monitor command for an IBM.Application resource definition:

```
lsrsrc -s "Name='syslog'" IBM.Application
resource 1:
 Name = "syslogd"
 ResourceType = 0
 AggregateResource = "0x3fff 0xffff 0x00000000 0x00000000 0x00000000 0x00000000"
 StartCommand = "/etc/init.d/syslog start"
 StopCommand = "/etc/init.d/syslog stop"
 MonitorCommand = "/usr/sbin/rsct/bin/pidmon '/sbin/syslogd'"

```

## Files

**/usr/sbin/rsct/bin/pidmon**      Location of the **pidmon** command.

## See Also

The **MonitorCommand** attribute of IBM.Application on page 132.

---

### Using expressions

The information in this section is for advanced users who want to:

- Modify predefined expressions.
- Select resources.
- Filter audit log records by compiling and running a complex mathematical expression against a set of values.

Permissible data types, operators, and operator order of precedence are described below. RMC uses these functions to match a selection string against the persistent attributes of a resource and to implement the evaluation of an event expression or a rearm expression.

An expression is similar to a C language statement or the WHERE clause of an SQL query. It is composed of variables, operators, and constants. The C and SQL syntax styles may be intermixed within a single expression. The following table relates the RMC terminology to SQL terminology:

| RMC            | SQL                             |
|----------------|---------------------------------|
| attribute name | column name                     |
| select string  | WHERE clause                    |
| operators      | predicates, logical connectives |
| resource class | table                           |

### SQL Restrictions

SQL syntax is supported for selection strings, with the following restrictions:

- Only a single table may be referenced in an expression.
- Queries may not be nested.
- The IS NULL predicate is not supported because there is no concept of a NULL value.
- The period (.) operator is not a table separator (for example, table.column). Rather, in this context, the period (.) operator is used to separate a field name from its containing structure name.
- The pound sign (#) is hard-coded as the escape character within SQL pattern strings.
- All column names are case sensitive.
- All literal strings must be enclosed in either single or double quotation marks. Bare literal strings are not supported because they cannot be distinguished from column and attribute names.

### Supported Base Data Types

The term *variable* is used in this context to mean the column name or attribute name in an expression. Variables and constants in an expression may be one of the following data types that are supported by the RMC subsystem:

| Symbolic Name | Description                                  |
|---------------|----------------------------------------------|
| CT_INT32      | Signed 32-bit integer                        |
| CT_UINT32     | Unsigned 32-bit integer                      |
| CT_INT64      | Signed 64-bit integer                        |
| CT_UINT64     | Unsigned 64-bit integer                      |
| CT_FLOAT32    | 32-bit floating point                        |
| CT_FLOAT64    | 64-bit floating point                        |
| CT_CHAR_PTR   | Null-terminated string                       |
| CT_BINARY_PTR | Binary data – arbitrary-length block of data |

|                    |                                                                                              |
|--------------------|----------------------------------------------------------------------------------------------|
| CT_RSRC_HANDLE_PTR | Resource handle – an identifier for a resource that is unique over space and time (20 bytes) |
|--------------------|----------------------------------------------------------------------------------------------|

## Structured Data Types

In addition to the base data types, aggregates of the base data types may be used as well. The first aggregate data type is similar to a structure in C in that it can contain multiple fields of different data types. This aggregate data type is referred to as *structured data* (SD). The individual fields in the structured data are referred to as *structured data elements*, or simply *elements*. Each element of a structured data type may have a different data type which can be one of the base types in the preceding table or any of the array types discussed in the next section, except for the structured data array.

The second aggregate data type is an array. An array contains zero or more values of the same data type, such as an array of CT\_INT32 values. Each of the array types has an associated enumeration value (CT\_INT32\_ARRAY, CT\_UINT32\_ARRAY). Structured data may also be defined as an array but is restricted to have the same elements in every entry of the array.

## Data Types That Can Be Used for Literal Values

Literal values can be specified for each of the base data types as follows:

**Array** An array or list of values may be specified by enclosing variables or literal values, or both, within braces {} or parentheses () and separating each element of the list with a comma. For example: { 1, 2, 3, 4, 5 } or ( "abc", "def", "ghi" ).

Entries of an array can be accessed by specifying a subscript as in the C programming language. The index corresponding to the first element of the array is always zero; for example, List [2] references the third element of the array named List. Only one subscript is allowed. It may be a variable, a constant, or an expression that produces an integer result. For example, if List is an integer array, then List[2]+4 produces the sum of 4 and the current value of the third entry of the array.

### Binary Data

A binary constant is defined by a sequence of hexadecimal values, separated by white space. All hexadecimal values comprising the binary data constant are enclosed in double quotation marks. Each hexadecimal value includes an even number of hexadecimal digits, and each pair of hexadecimal digits represents a byte within the binary value. For example:

```
"0xabcd 0x01020304050607090a0b0c0d0e0f1011121314"
```

### Character Strings

A string is specified by a sequence of characters surrounded by single or double quotation marks (you can have any number of characters, including none). Any character may be used within the string except the null '\0' character. Double quotation marks and backslashes may be included in strings by preceding them with the backslash character.

### Floating Types

These types can be specified by the following syntax:

- A leading plus (+) or minus (-) sign
- One or more decimal digits
- A radix character, which at this time is the period (.) character
- An optional exponent specified by the following:
  - A plus (+) or minus (-) sign
  - The letter 'E' or 'e'
  - A sequence of decimal digits (0–9)

### Integer Types

These types can be specified in decimal, octal, or hexadecimal format. Any value that begins with

## Using Expressions

the digits 1-9 and is followed by zero or more decimal digits (0-9) is interpreted as a decimal value. A decimal value is negated by preceding it with the character '-'. Octal constants are specified by the digit 0 followed by 1 or more digits in the range 0-7. Hexadecimal constants are specified by a leading 0 followed by the letter x (uppercase or lowercase) and then followed by a sequence of one or more digits in the range 0–9 or characters in the range a–f (uppercase or lowercase).

### Resource Handle

A fixed-size entity that consists of two 16-bit and four 32-bit words of data. A literal resource handle is specified by a group of six hexadecimal integers. The first two values represent 16-bit integers and the remaining four each represent a 32-bit word. Each of the six integers is separated by white space. The group is surrounded by double quotation marks. The following is an example of a resource handle:

```
"0x4018 0x0001 0x00000000 0x0069684c 0x00519686 0xaf7060fc"
```

### Structured Data

Structured data values can be referenced only through variables. Nevertheless, the RMC command line interface displays structured data (SD) values and accepts them as input when a resource is defined or changed. A literal SD is a sequence of literal values, as defined in “Data Types That Can Be Used for Literal Values” on page 223, that are separated by commas and enclosed in square brackets. For example, [‘abc’,1,{3,4,5}] specifies an SD that consists of three elements: (a) the string ‘abc’, (b) the integer value 1, and (c) the three-element array {3,4,5}.

Variable names refer to values that are not part of the expression but are accessed while running the expression. For example, when RMC processes an expression, the variable names are replaced by the corresponding persistent or dynamic attributes of each resource.

The elements of a structured data value can be accessed by using the following syntax:

```
<variable name>.<element name>
```

For example, a.b

The variable name is the name of the table column or resource attribute, and the element name is the name of the element within the structured data value. Either or both names may be followed by a subscript if the name is an array. For example, a[10].b refers to the element named b of the 11th entry of the structured data array called a. Similarly, a[10].b[3] refers to the fourth element of the array that is an element called b within the same structured data array entry a[10].

## How Variable Names Are Handled

Variable names refer to values that are not part of an expression but are accessed while running the expression. When used to select a resource, the variable name is a persistent attribute. When used to generate an event, the variable name is a dynamic attribute. When used to select audit records, the variable name is the name of a field within the audit record.

A variable name is restricted to include only 7-bit ASCII characters that are alphanumeric (a-z, A-Z, 0-9) or the underscore character (\_). The name must begin with an alphabetic character. When the expression is used by the RMC subsystem for an event or a rearm event, the name can have a suffix that is the '@' character followed by 'P', which refers to the previous observation.

## Operators That Can Be Used in Expressions

Constants and variables may be combined by an operator to produce a result that in turn may be used with another operator. The resulting data type or the expression must be a scalar integer or floating-point value. If the result is zero, the expression is considered to be FALSE; otherwise, it is TRUE.

**Note:** Blanks are optional around operators and operands unless their omission causes an ambiguity. An ambiguity typically occurs only with the word form of operator (that is, AND, OR, IN, LIKE, etc.). With these operators, a blank or separator, such as a parenthesis or bracket, is required to

distinguish the word operator from an operand. For example, aANDb is ambiguous. It is unclear if this is intended to be the variable name aANDb or the variable names a, b combined with the operator AND. It is actually interpreted by the application as a single variable name aANDb. With non-word operators (for example, +, -, =, &&, etc.) this ambiguity does not exist, and therefore blanks are optional.

The set of operators that can be used in strings is summarized in the following table:

| Operator | Description           | Left Data Types | Right Data Types | Example                                     | Notes                                                           |
|----------|-----------------------|-----------------|------------------|---------------------------------------------|-----------------------------------------------------------------|
| +        | Addition              | Integer,float   | Integer,float    | "1+2" results in 3                          | None                                                            |
| -        | Subtraction           | Integer,float   | Integer,float    | "1.0-2.0" results in -1.0                   | None                                                            |
| *        | Multiplication        | Integer,float   | Integer,float    | "2*3" results in 6                          | None                                                            |
| /        | Division              | Integer,float   | Integer,float    | "2/3" results in 1                          | None                                                            |
| -        | Unary minus           | None            | Integer,float    | "-abc"                                      | None                                                            |
| +        | Unary plus            | None            | Integer,float    | "+abc"                                      | None                                                            |
| ..       | Range                 | Integers        | Integers         | "1..3" results in 1,2,3                     | Shorthand for all integers between and including the two values |
| %        | Modulo                | Integers        | Integers         | "10%2" results in 0                         | None                                                            |
|          | Bitwise OR            | Integers        | Integers         | "2 4" results in 6                          | None                                                            |
| &        | Bitwise AND           | Integers        | Integers         | "3&2" results in 2                          | None                                                            |
| ~        | Bitwise complement    | None            | Integers         | ~0x0000ffff results in 0xffff0000           | None                                                            |
| ^        | Exclusive OR          | Integers        | Integers         | 0x0000aaaa^0x0000ffff results in 0x00005555 | None                                                            |
| >>       | Right shift           | Integers        | Integers         | 0x0fff>>4 results in 0x00ff                 | None                                                            |
| <<       | Left shift            | Integers        | Integers         | "0x0fff<<4" results in 0xffff0              | None                                                            |
| ==       | Equality              | All but SDs     | All but SDs      | "2==2" results in 1<br>"2=2" results in 1   | Result is true (1) or false (0)                                 |
| !=       | Inequality            | All but SDs     | All but SDs      | "2!=2" results in 0<br>"2<>2" results in 0  | Result is true (1) or false (0)                                 |
| >        | Greater than          | Integer,float   | Integer,float    | "2>3" results in 0                          | Result is true (1) or false (0)                                 |
| >=       | Greater than or equal | Integer,float   | Integer,float    | "4>=3" results in 1                         | Result is true (1) or false (0)                                 |
| <        | Less than             | Integer,float   | Integer,float    | "4<3" results in 0                          | Result is true (1) or false (0)                                 |
| <=       | Less than or equal    | Integer,float   | Integer,float    | "2<=3" results in 1                         | Result is true (1) or false (0)                                 |
| =_       | Pattern match         | Strings         | Strings          | "abc"=_ "a.*" results in 1                  | Right operand is interpreted as an extended regular expression  |

## Using Expressions

| Operator                   | Description           | Left Data Types | Right Data Types | Example                       | Notes                                                                     |
|----------------------------|-----------------------|-----------------|------------------|-------------------------------|---------------------------------------------------------------------------|
| !_                         | Not pattern match     | Strings         | Strings          | "abc"!_"a.*" results in 0     | Right operand is interpreted as an extended regular expression            |
| =?<br>LIKE<br>like         | SQL pattern match     | Strings         | Strings          | "abc"=? "a%" results in 1     | Right operand is interpreted as a SQL pattern                             |
| !?<br>NOT LIKE<br>not like | Not SQL pattern match | Strings         | Strings          | "abc"!?"a%" results in 0      | Right operand is interpreted as a SQL pattern                             |
| <<br>IN<br>in              | Contains any          | All but SDs     | All but SDs      | "{1..5} <{2,10}" results in 1 | Result is true (1) if left operand contains any value from right operand  |
| ><<br>NOT IN<br>not in     | Contains none         | All but SDs     | All but SDs      | "{1..5}><{2,10}" results in 1 | Result is true (1) if left operand contains no value from right operand   |
| &<                         | Contains all          | All but SDs     | All but SDs      | "{1..5}&<{2,10}" results in 0 | Result is true (1) if left operand contains all values from right operand |
| <br>OR<br>or               | Logical OR            | Integers        | Integers         | "(1<2)   (2>4)" results in 1  | Result is true (1) or false (0)                                           |
| &&<br>AND<br>and           | Logical AND           | Integers        | Integers         | "(1<2)&& (2>4)" results in 0  | Result is true (1) or false (0)                                           |
| !<br>NOT<br>not            | Logical NOT           | None            | Integers         | "!(2==4)" results in 1        | Result is true (1) or false (0)                                           |

## Default Precedence of Operators

When integers of different signs or size are operands of an operator, standard C style casting is implicitly performed. When an expression with multiple operators is evaluated, the operations are performed in the order defined by the precedence of the operator. The default precedence can be overridden by enclosing the portion or portions of the expression to be evaluated first in parentheses ().

For example, in the expression "1+2\*3", multiplication is normally performed before addition to produce a result of 7. To evaluate the addition operator first, use parentheses as follows: "(1+2)\*3". This produces a result of 9. The default precedence rules are shown in the following table. All operators in the same table cell have the same or equal precedence.

| Operators | Description                       |
|-----------|-----------------------------------|
| .         | Structured data element separator |
| ~         | Bitwise complement                |
| !         | Logical not                       |
| NOT       |                                   |
| not       |                                   |
| -         | Unary minus                       |
| +         | Unary plus                        |
| *         | Multiplication                    |
| /         | Division                          |
| %         | Modulo                            |
| +         | Addition                          |
| -         | Subtraction                       |
| <<        | Left shift                        |
| >>        | Right shift                       |
| <         | Less than                         |
| <=        | Less than or equal                |
| >         | Greater than                      |
| >=        | Greater than or equal             |

## Using Expressions

| Operators | Description             |
|-----------|-------------------------|
| ==        | Equality                |
| !=        | Inequality              |
| =?        | SQL match               |
| LIKE      |                         |
| like      |                         |
| !?        | SQL not match           |
| =_        | Reg expr match          |
| !_        | Reg expr not match      |
| ?=        | Reg expr match (compat) |
| <         | Contains any            |
| IN        |                         |
| in        |                         |
| ><        | Contains none           |
| NOT IN    |                         |
| not in    |                         |
| &<        | Contains all            |
| &         | Bitwise AND             |
| ^         | Bitwise exclusive OR    |
|           | Bitwise inclusive OR    |
| &&        | Logical AND             |
|           | Logical OR              |
| ,         | List separator          |

## Pattern Matching

Two types of pattern matching are supported; extended regular expressions and that which is compatible with the standard SQL LIKE predicate. This type of pattern may include the following special characters:

- The percentage sign (%) matches zero or more characters.
- The underscore (\_) matches exactly one character.
- All other characters are directly matched.
- The special meaning for the percentage sign and the underscore character in the pattern may be overridden by preceding these characters with an escape character, which is the pound sign (#) in this implementation.

## Examples of Expressions

Some examples of the types of expressions that can be constructed follow:

1. The following expressions match all rows or resources that have a name which begins with 'tr' and ends with '0', where 'Name' indicates the column or attribute that is to be used in the evaluation:

```
Name =~'tr.*0'
Name LIKE 'tr%0'
```

2. The following expressions evaluate to TRUE for all rows or resources that contain 1, 3, 5, 6, or 7 in the column or attribute that is called IntList, which is an array:

```
IntList|<{1,3,5..7}
IntList in (1,3,5..7)
```

3. The following expression combines the previous two so that all rows and resources that have a name beginning with 'tr' and ending with '0' and have 1, 3, 5, 6, or 7 in the IntList column or attribute will match:

```
(Name LIKE "tr%0")&&(IntList|<{1,3,5..7})
(Name=~'tr.*0') AND (IntList IN {1,3,5..7})
```

## Using Expressions

---

## Appendix A. Messages used with IBM Tivoli System Automation

This chapter is for any user of IBM Tivoli System Automation who needs to know what a message means and what should be done in response to that message. This chapter lists all of the messages that are generated by IBM Tivoli System Automation and describes solutions for each of these messages.

---

**2621-001**    **Attribute "*attribute\_name*" cannot be specified when defining a new resource.**

**Explanation:** An error occurred when an attribute that can not be used in defining a resource has been specified.

**User Response:** Specify only the attributes which can be specified in defining the resource.

---

**2621-002**    **Attribute "*attribute\_name*" appears in request more than once.**

**Explanation:** An error occurred as the same attribute is used multiple times.

**User Response:** Use the attribute only once.

---

**2621-003**    **Class name "*class\_name*" is not recognized by this resource manager.**

**Explanation:** An internal error occurred when attempting to create an RCCP for a class which does not exist.

**User Response:** Contact your software service organization.

---

**2621-004**    **Could not initialize control point for class "*class\_name*".**

**Explanation:** An internal error occurred when attempting to create an RCCP for the class.

**User Response:** Contact your software service organization.

---

**2621-005**    **Attribute "*attribute\_name*" must be specified when defining a new resource.**

**Explanation:** This error was detected when a mandatory attribute is excluded in defining a resource.

**User Response:** Use "`!lsrsrcdef classname`" to check the attribute properties, include all mandatory (ReqdForDefine) attributes.

---

**2621-006**    **Unrecognized resource class id: *resource\_class\_id*.**

**Explanation:** An error occurred while extracting the class info for this class id.

**User Response:** If a resource handle is used, check if it is correct. Otherwise, contact your software service organization.

---

**2621-007**    **Time out waiting for resource enumeration responses for the IBM.PeerNode class.**

**Explanation:** A time out error occurred while waiting for peer nodes enumeration registration response.

**User Response:** Check if the RMC and ConfigRM are running (use "`!ssrc -a`" command). If either of the subsystem is inoperative, consult the RMC/ConfigRM user guides to check how to restart the non-operational subsystem.

---

**2621-008**    **Failed to update resource because of configuration data replication errors.**

**Explanation:** An error occurred while replicating data.

**User Response:** Retry the operation. If the problem continues, report to your software service organization.

---

## Messages

---

**2621-009 Command not allowed - IBM.RecoveryRM not initialized.**

**Explanation:** The resource manager is not yet initialized to perform client's request.

**User Response:** Wait for initialization and retry the command.

---

**2621-010 Command not allowed - IBM.RecoveryRM not in configuration quorum.**

**Explanation:** The cluster is probably running with several IBM.RecoveryRM daemons failure. The operation needs the quorum of IBM.RecoveryRM daemons online.

**User Response:** Try to restart the failed IBM.RecoveryRM daemons to bring back the configuration quorum.

---

**2621-011 Command not allowed - node failed configuration data replication.**

**Explanation:** The node had failed updating the configuration data - can not perform the client's request.

**User Response:** Retry the operation. If the problem continues, report to your software service organization.

---

**2621-012 Command not allowed - configuration data replacement in process.**

**Explanation:** The system is in transition to a new configuration data - can not perform the client's request.

**User Response:** Retry the operation. If the problem continues, report to your software service organization.

---

**2621-013 Command not allowed - replication can not be started.**

**Explanation:** An error occurred in initializing the replication process.

**User Response:** Retry the operation. If the problem continues, report to your software service organization.

---

**2621-014 Command not allowed - one or more related resource groups are online.**

**Explanation:** An error was generated while changing a managed resource, resource group, or relationship which involves one or more online resource groups.

**User Response:** Before changing the resource, the resource groups have to be offline.

---

**2621-015 Failed to get resource class persistent attributes for class "*class\_name*".**

**Explanation:** An error was returned while querying the class to get its persistent attributes.

**User Response:** Check if the resource manager to which the class belongs to is operational. If the resource manager is inoperative, consult its user guide to check how to restart.

---

**2621-016 Failed to get definition of resource persistent attributes for class "*class\_name*".**

**Explanation:** An error was returned while querying a resource to get its persistent attributes.

**User Response:** Check if the resource manager to which the class belongs to is operational. If the resource manager is inoperative, consult its user guide to check how to restart.

---

**2621-017 Time out waiting for attribute query responses for the "*class\_name*" class.**

**Explanation:** A time out error occurred while waiting for attribute query response from RMC.

**User Response:** Check if the resource manager to which the class belongs to is operational. If the resource manager is inoperative, consult its user guide to check how to restart.

---

---

**2621-018 Error encountered when calling function "*function\_name*". return code = *return\_code***

**Explanation:** An error condition was returned when this function was invoked.

**User Response:** This is an internal error, report it to your software service organization.

---

**2621-019 Time out waiting for response of event registration with resource handle.**

**Explanation:** A time out error occurred while waiting for a response on an event registration with resource handle.

**User Response:** Check if the RMC and the resource manager for the resource are running (use "`lssrc -a`" command). If either of the subsystem is inoperative, consult the corresponding user guide to check how to restart the non-operational subsystem.

---

**2621-020 Resource has an invalid resource class name "*resource\_class\_name*".**

**Explanation:** The referenced resource class name is not found.

**User Response:** Use "`lssrc`" command to display the supported resource classes. Correct the class name and retry the operation.

---

**2621-021 Resource does not exist.**

**Explanation:** The referenced resource is not found.

**User Response:** Use "`lssrc class_name`" to display all resources with in a class.

---

**2621-022 Resource encounters event registration error. The error message is: [*error\_message*]**

**Explanation:** An error message was returned by the RMC in response to an event registration for a resource.

**User Response:** Check the error message and take the appropriate measure. Restart the RMC and the resource manager if they are not running.

---

**2621-023 Resource RCP not found**

**Explanation:** An internal error was detected while a control point for a resource was not found.

**User Response:** Contact your software service organization.

---

**2621-024 Timed out waiting for resource enumeration responses for the IBM.PeerDomain class.**

**Explanation:** A time out error occurred while waiting for peer domains enumeration registration

**User Response:** Check if the RMC and ConfigRM are running (use "`lssrc -a`" command). If either of the subsystem is inoperative, consult the RMC/ConfigRM user guides to check how to restart the non-operational subsystem.

---

**2621-025 Value "*input\_parameter\_value*" provided for input parameter "*input\_parameter\_name*" in SubmitRequest action is not valid.**

**Explanation:** Value provided for an input parameter in SubmitRequest action is not valid

**User Response:** Check value provided for input parameter of SubmitRequest action. Correct the value and retry the action

---

**2621-026 This functionality is not supported by current active version of the cluster.**

**Explanation:** The cluster is in co-existence or migration mode. The current active version does not support the functionality.

**User Response:** Retry after migration is completed.

---

## Messages

---

**2621-027** There is no previous request from source [*source\_name*]. Cancellation failed.

**Explanation:** Cancellation of request is received but there is no previous request from the specified source.

**User Response:** Check value provided for source of SubmitRequest action. Correct the value and retry the action

---

**2621-028** Command not allowed - complete migration action in process.

**Explanation:** The system is in transition to a new active code version level - can not perform the client's request.

**User Response:** Retry the operation. If the problem continues, report to your software service organization.

---

**2621-029** Request to move can not be granted as there is already a move request on the resource.

**Explanation:** Request to move can not be granted as there is already a move request on the same resource group.

**User Response:** Wait for five minutes and Retry the operation.

---

**2621-030** Timed out waiting for query persistent class attribute request responses for the IBM.PeerNode class.

**Explanation:** A time out error occurred while waiting for peer node QuorumType registration

**User Response:** Check if the RMC and ConfigRM are running (use "Issrc -a" command). If either of the subsystem is inoperative, consult the RMC/ConfigRM user guides to check how to restart the non-operational subsystem.

---

**2621-050** Resource group name "*resource\_group\_name*" already defined.

**Explanation:** An error was detected while an existing resource group name is used to define a new resource group.

**User Response:** Use an unique name and retry the command.

---

**2621-051** Resource group has an invalid allowed node as the equivalency members do not belong to IBM.PeerNode.

**Explanation:** The AllowedNode of the defined resource group comes from an equivalency whose MemberClass attribute is not IBM.PeerNode.

**User Response:** Correct the problem and retry the operation.

---

**2621-052** Resource group member location is incompatible with its outer resource group.

**Explanation:** The MemberLocation attribute of the inner resource group is incompatible with its outer resource group.

**User Response:** Correct the problem and retry the operation.

---

**2621-053** Resource group member location is incompatible with its members relationships.

**Explanation:** The MemberLocation attribute of the resource group is incompatible with a defined location sensitive (Collocated, DependsOn, etc ) managed relationship. The resource group may have one or more members participating in the conflicted relationship.

**User Response:** Fix the location compatibility problem either by changing the attribute or the relationship or both.

---

**2621-054** Resource group priority must be between 0 and 200.

**Explanation:** The priority value specified by the user is not valid.

**User Response:** Specify a priority with in the valid range 0 and 200.

---

---

**2621-055** Resource group priority *resource\_group\_priority* exceeds the priorities of outer resource groups.

**Explanation:** An error was detected in a nested resource group where an inner resource group has a higher priority than an outer resource group.

**User Response:** Reset the priority so that outer resource group has the same or higher priority than an inner resource group, and retry the command.

---

**2621-056** Resource group has an invalid value for attribute MemberLocation: *invalid\_value*.

**Explanation:** The value entered for attribute MemberLocation is invalid.

**User Response:** Define resource group with a valid value for attribute MemberLocation.

---

**2621-057** Resource group has an invalid value for attribute NominalState: *invalid\_value*.

**Explanation:** The value entered for attribute NominalState is invalid.

**User Response:** Enter a valid value for attribute NominalState.

---

**2621-058** Resource group nesting level exceeds the maximum of 50 levels.

**Explanation:** The resource groups nesting level can not exceed maximum of 50 levels.

**User Response:** Reduce the resource groups nesting level.

---

**2621-059** Node ID can not be found.

**Explanation:** A node id in SubmitRequest resource action does not correspond to a node in cluster

**User Response:** Check value of node id. Correct the value and retry the action

---

**2621-060** ExcludedList is not supported for Resource Group in the current active version of the cluster.

**Explanation:** The cluster is in co-existence or migration mode. The current active version does not support the ExcludedList attribute for Resource Groups.

**User Response:** Retry after migration is completed.

---

**2621-061** Resource group has an invalid ExcludedList as all or some of its members do not belong to the current cluster.

**Explanation:** The ExcludedList of the defined resource has at least one of its members or nodes that do not belong in the current cluster.

**User Response:** Correct the problem and retry the operation.

---

**2621-062** A node specified in move action is not in the allowed node list of resource group.

**Explanation:** Resource group is being requested to move away from a node it is not allowed.

**User Response:** Check value provided for input parameter of SubmitRequest action. Correct the value and retry the action

---

**2621-063** The resource group is in offline state and can not be moved.

**Explanation:** Resource group being requested to move is in offline state.

**User Response:** Check target of move request. Correct the target and retry the action.

---

## Messages

---

**2621-064 Node name list not specified when moving a non-collocated resource group.**

**Explanation:** A list of node names to move resources away from must be specified when moving a non-collocated resource group.

**User Response:** Retry the command with node name list specified.

---

**2621-065 Resource group "*resource\_group*" cannot be brought online - Empty Tree.**

**Explanation:** The specified resource group cannot be brought online because it did not have any resources that could be brought online.

**User Response:** Add a real resource to the group and then bring it online.

---

**2621-066 Resource group can not be moved as it is not a top-level resource group.**

**Explanation:** The specified resource group cannot be moved because it is not a top-level resource group.

**User Response:** Issue the move request against a top-level resource group.

---

**2621-067 Resource group can not be moved as it contains only fixed resources.**

**Explanation:** The specified resource group cannot be moved because it contains only fixed resources.

**User Response:** Rethink the needs to move this resource group.

---

**2621-101 Managed resource cannot be in more than one resource group, already in "*resource\_group\_name*".**

**Explanation:** An error was detected when a resource that already exists as a managed resource was attempted to be included in the same or another resource group.

**User Response:** Retry the operation with correct parameters.

---

**2621-102 Managed resource has an invalid resource handle.**

**Explanation:** A resource manager has responded back with an invalid resource handle.

**User Response:** If this internal problem continues, contact your software service organization

---

**2621-103 Managed resource "*resource\_name*" does not have control interface.**

**Explanation:** An error was generated while it was detected that resource class does not support the control interface.

**User Response:** Check the attributes of the resource class. Add only the resources whose OpState can be changed.

---

**2621-104 Managed resource's resource group "*resource\_group\_name*" does not exist.**

**Explanation:** A dangling managed resource is detected whose resource group does not exist any more.

**User Response:** This may be an internal error. Contact your software service organization.

---

**2621-105 Managed resource has an invalid resource class id *class\_id*.**

**Explanation:** The referenced managed resource class has an invalid class id.

**User Response:** Use "`lsrsrcdef -c class-name`" to display the correct class id. Use the correct class name and retry the operation.

---

---

**2621-106** Managed resource has an invalid resource type *resource\_type*.

**Explanation:** A managed resource can either be fixed or floating type. An error occurred as an invalid typed resource was used.

**User Response:** Correct the problem and retry the operation.

---

**2621-107** Managed resource is a fixed resource, but is not defined on any node.

**Explanation:** An error was generated as a fixed managed resource was not found on any node.

**User Response:** Either remove the resource from the resource group or modify type as floating.

---

**2621-108** Resource group's member location is incompatible with its outer resource group.

**Explanation:** The MemberLocation attribute of the inner resource group is incompatible with its outer resource group.

**User Response:** Correct the problem and retry the operation.

---

**2621-109** Managed resource is a fixed resource, but is defined on *number\_of\_nodes* nodes.

**Explanation:** An error was generated as a fixed resource was defined on multiple nodes.

**User Response:** Either remove the resource from the resource group or modify the node list or change type to floating.

---

**2621-110** Can not find resource class information for managed resource.

**Explanation:** An error was generated when enumerating the class information for a resource.

**User Response:** Verify that the correct class name was used. If error continues, report to your software service organization.

---

**2621-111** Resource group can not be a member of itself.

**Explanation:** A managed resource (resource group) can not be nested within itself.

**User Response:** Retry the operation with correct parameters.

---

**2621-112** Value of attribute Mandatory: *invalid\_value* for managed resource is not valid.

**Explanation:** The value entered for attribute Mandatory is invalid.

**User Response:** Define managed resource with a valid value for attribute Mandatory.

---

**2621-113** Managed resource "*resource\_name*" is the target of a resource in another tree.

**Explanation:** A managed resource can not be the target of two independent resource groups. The error was generated due to the violation of this constraint.

**User Response:** Remove the conflict and retry the operation.

---

**2621-114** Managed resource "*resource\_name*" can not be defined in an equivalency.

**Explanation:** An error was generated when a managed resource was being included as a member of an equivalency.

**User Response:** Define the equivalency only with resources that are not defined as managed resources.

---

## Messages

---

### 2621-115 **Managed resource is from an invalid class.**

**Explanation:** A resource designated as managed has a resource handle that does not belong to the referenced class.

**User Response:** Correct the class name and retry the operation.

---

### 2621-116 **Resource does not have dynamic attribute - OpState.**

**Explanation:** An error was generated while it was detected that resource class does not have the dynamic attribute - OpState.

**User Response:** Check dynamic attributes of the resource class. Add only the resources which has OpState.

---

### 2621-117 **Can not change source managed resource while there are targets.**

**Explanation:** An error was generated while trying to change the source of a relationship when the relationship has one or more targets.

**User Response:** Remove the relationship and recreate a new one with appropriate source.

---

### 2621-118 **Can not change the resource handle of a managed resource.**

**Explanation:** A managed resource always represents the resource for which it was created.

**User Response:** Remove the member representing the managed resource from the resource group and add a new member that points to the correct resource.

---

### 2621-119 **Resource group does not exist.**

**Explanation:** An error was generated when a non-existent resource group was referenced.

**User Response:** Check the valid resource groups using "lsrg" command. Retry the operation using correct resource group.

---

### 2621-120 **Operation temporarily disallowed, as resource registration is in progress.**

**Explanation:** Previous attempt to register for resource events failed. While daemon is retrying the registration, some functionalities is not available.

**User Response:** Correct the registration problem, then retry the operation.

---

### 2621-121 **Value of attribute SelectFromPolicy: *invalid\_value* for managed resource is not valid.**

**Explanation:** The value entered for attribute SelectFromPolicy is invalid.

**User Response:** Define managed resource with a valid value for attribute SelectFromPolicy.

---

### 2621-122 **SelectFromPolicy is not supported for Managed resources in the current active version of the cluster.**

**Explanation:** The cluster is in co-existence or migration mode. The current active version does not support the SelectFromPolicy feature for managed resources.

**User Response:** Retry after migration is completed.

---

### 2621-150 **Managed relationship has a source resource that is not a managed resource.**

**Explanation:** An error was detected when a non-managed resource was used as the source of a managed relationship. The source must be a managed resource.

**User Response:** Correct the problem, and retry the operation.

---

---

**2621-151 Managed relationship has a target resource that is not valid.**

**Explanation:** An error was generated as an invalid or non-existent resource was used as the target of a managed relationship.

**User Response:** Correct the problem and retry the operation.

---

**2621-152 Managed relationship name "*relationship\_name*" already defined.**

**Explanation:** A managed relationship with this name was already defined.

**User Response:** Use an unique name and retry the operation.

---

**2621-153 Managed relationship target specified more than once.**

**Explanation:** An error was generated as a resource was used more than once as the target of a managed relationship.

**User Response:** Correct the problem and retry the operation.

---

**2621-154 Managed relationship has an invalid value 0 for attribute - Relationship.**

**Explanation:** A new managed relationship was being defined with no (NONE) relationship value.

**User Response:** Define managed relationship with a valid value for relationship.

---

**2621-155 Managed relationship source defined as target resource.**

**Explanation:** An error condition was detected. A managed relationship's source and target can not be the same.

**User Response:** Remove source resource from the target, and retry the operation.

---

**2621-156 Managed relationship conflicts with a previously defined relationship.**

**Explanation:** A new relationship was defined which conflicts with a previously defined location sensitive or order sensitive relation.

**User Response:** Remove the conflict and retry the operation.

---

**2621-157 Managed relationship duplicates a previously defined relationship.**

**Explanation:** An error was generated due to re-definition of a relationship.

**User Response:** Define only one instance of a relationship.

---

**2621-158 Managed relationship has a source or target that is not of an allowed class.**

**Explanation:** The resource from this class can't be the source or target of the defined relationship.

**User Response:** Correct the source or target resource and retry the operation.

---

**2621-159 Managed relationship is not consistent with source's resource group member location.**

**Explanation:** A conflict was detected in location association of the relationship.

**User Response:** Check the source's MemberLocation attribute and correct the problem.

---

**2621-160 Managed relationship introduces circular relationship from a resource back to itself.**

**Explanation:** For StartAfter or DependsOn, or DependsOnAny relationship, a circular dependency was detected which may create a deadlock.

**User Response:** Remove the circular dependency and retry the operation.

---

## Messages

---

**2621-161 Specified Relationship attribute for Managed relationship is not supported by current active version of the cluster.**

**Explanation:** The cluster is in co-existence or migration mode. The current active version does not support the specified relationship attribute.

**User Response:** Retry after migration is completed.

---

**2621-200 Equivalency name "*equivalency\_name*" already defined.**

**Explanation:** An equivalency with this name was already defined.

**User Response:** Use an unique name and retry the operation.

---

**2621-201 Equivalency membership must either be explicitly enumerated or specified with a select string.**

**Explanation:** An error was generated while both a select string and an enumerated set of resource handles are used to define equivalency membership.

**User Response:** Use either enumeration or select string (not both) and retry the operation.

---

**2621-202 Equivalency has an invalid select string.**

**Explanation:** User entered an invalid select string.

**User Response:** Correct the problem, especially check the attribute name and value. Retry the operation.

---

**2621-203 Equivalency has an invalid member.**

**Explanation:** An error condition was detected as an equivalency member was found which was either a managed resource or did not belong to the same class as other members.

**User Response:** Correct the problem and retry the operation.

---

**2621-204 Equivalency of nodes has invalid member.**

**Explanation:** A node in an equivalency do not belong to IBM.PeerNode class.

**User Response:** Correct the problem and retry the operation.

---

**2621-205 Equivalency has members belonging to different resource classes.**

**Explanation:** An error was detected in an equivalency members. Resources within the equivalency are from different classes. All resources within the an equivalency must be from the same class.

**User Response:** Correct the problem and retry the operation.

---

**2621-206 Equivalency membership has a duplicated resource handle `\resource_handle\`.**

**Explanation:** Two members of an equivalency has the same resource handle.

**User Response:** Remove one and then retry the operation.

---

**2621-207 Equivalency membership has resource handles containing multiple class ids (*class\_id1* and *class\_id2*).**

**Explanation:** An error was generated while defining an equivalency. Two resource members within the equivalency are from different classes. All resources within the an equivalency must be from the same class.

**User Response:** Correct the problem and retry the operation.

---

---

**2621-208 Equivalency need to specify both select string and a valid resource class name.**

**Explanation:** An error was generated while an invalid resource class name was used with the dynamic select string.

**User Response:** Correct the class name and retry the operation.

---

**2621-209 Can not find resource class information for equivalency "*equivalency\_name*".**

**Explanation:** An error was generated while enumerating the class information for the member class.

**User Response:** This may be an internal error, report to your software service organization.

---

**2621-210 Resource class "*resource\_class\_name*" is invalid for equivalency.**

**Explanation:** A resource from this class is not valid for this equivalency. Class id may be different from other members.

**User Response:** Use the resources from the same class to correct the problem.

---

**2621-211 Resource class may not be changed in an equivalency without replacing the members.**

**Explanation:** An error was generated while user was trying to change the MemberClass attribute value of an equivalency which has members (may be from a different class).

**User Response:** Remove the members before changing the MemberClass.

---

**2621-212 ManagedResource "*equivalency\_name*" can not be a member of an Equivalency.**

**Explanation:** An error was generated as a managed resource was placed as a member of this equivalency.

**User Response:** Add only the resources that are not member of any resource group.

---

**2621-213 A ManagedResource referenced as an AllowedNode may not change its class.**

**Explanation:** Error was detected as class of a managed resource referenced as an AllowedNode was changed.

**User Response:** Remove the error condition and retry.

---

**2621-214 Resource class may not be changed for an equivalency if there are members.**

**Explanation:** An error was generated while user was trying to change the MemberClass attribute value of an equivalency which has members (may be from a different class).

**User Response:** Remove the members before changing the MemberClass.

---

**2621-215 A selection Policy other than ANY is invalid with a selection string.**

**Explanation:** An error was generated while parsing an equivalency definition with selection string where the selection policy was not ANY.

**User Response:** Change the selection policy to ANY and retry the operation.

---

**2621-216 Minimum value must between 1 and 100.**

**Explanation:** An error was generated while an invalid MinimumNecessary attribute value was entered by the user.

**User Response:** Use a valid value from 1 - 100 and retry the command.

---

**2621-217 Equivalency contains no member resource**

**Explanation:** This error message indicates that an IBM.Equivalency resource is defined with a NULL select string, and an empty membership list.

**User Response:** Define an equivalency with either a select string or a non-empty membership list

---

## Messages

---

**2621-218 Floating resource with resource handle `\\"resource_handle\\"` can not be a member of Equivalency `\\"Equivalency_name\\"`.**

**Explanation:** Floating resources are not allowed to be a member of an equivalency.

**User Response:** Add only fixed or constituents resources as members of equivalencies.

---

**2621-300 Node specified by resource handles for excluded nodes could not be found.**

**Explanation:** The excluded nodes contains one or more nodes that are not defined in IBM.PeerNode resource class.

**User Response:** Correct the problem and retry the operation.

---

**2621-301 Input to the `"class_action_name"` action is not valid.**

**Explanation:** A class action activation request is received which contains structured data with invalid elements.

**User Response:** Correct the problem and retry the operation.

---

**2621-302 attribute `"invalid_value"` has an invalid or out of range value.**

**Explanation:** During the process of attempting to add or change attribute values, it was discovered that the value is either invalid or out of range.

**User Response:** Correct the problem and retry the operation.

---

**2621-303 The automation engine is not initialized.**

**Explanation:** The decision engine is not initialized.

**User Response:** Wait for five minutes and retry the operation. If problem continues, report to IBM service organization.

---

**2621-304 Invalid action, RecoveryRM is not in ReplaceConfig mode.**

**Explanation:** An invalid restore configuration option entered.

**User Response:** Correct the problem and retry the operation.

---

**2621-305 Feature not enabled, RecoveryRM is either running in down graded or migration mode.**

**Explanation:** A feature option that can not be activated is entered.

**User Response:** Complete migration to an appropriate version level.

---

**2621-306 Input to the `"CompleteMigration"` action is not valid. Our IVN is incompatible to this `NewActiveVersion: "active_version"`**

**Explanation:** The NewActiveVersion release name is incompatible to the current installed RecoveryRM daemon version.

**User Response:** Correct the problem and retry the operation.

---

**2621-307 Input to the `"CompleteMigration"` action is not valid. The range of `NewActiveVersion` is between version `"1.1.1.0"` and version `"256.256.256.256"` - `"version"`**

**Explanation:** The value of the NewActiveVersion release name is invalid. It needs to be within the range of `"1.1.1.0"` and `"256.256.256.256"`.

**User Response:** Correct the problem and retry the operation.

---

---

**2621-308** Input to the "CompleteMigration" action is not valid. NewActiveVersion ("*new\_active\_version*") cannot higher than IVN ("*installed\_version*") or lower than current AVN ("*%3\$s*  
*current\_active\_version*").

**Explanation:** The value of the NewActiveVersion release name is invalid. It cannot lower than the current active version number or higher than installed version number.

**User Response:** Correct the problem and retry the operation.

---

**2621-309** Command not allowed as daemon does not have a valid license.

**Explanation:** The command being issued is not allowed to run on this daemon as a valid license is not installed.

**User Response:** Install a valid license and retry the operation.

---

**2621-310** Cannot perform the "CompleteMigration" action. Current joined RecoveryRM daemon member count (*daemon\_member\_count*) is different than the total node count (*total\_node\_count*). Use force option ("-f") to override.

**Explanation:** Since not all the RecoveryRM daemons are up and running, we cannot start the migration complete action.

**User Response:** Make sure that all the RecoveryRM daemons are up and running or use the force option ("-f") to start the migration complete action.

---

**2621-311** Publisher is not supported in the current active version of the cluster.

**Explanation:** The cluster is in co-existence or migration mode. The current active version does not support the Publisher feature.

**User Response:** Retry after migration is completed.

---

**2621-312** License file "*license\_file\_name*" does not exist.

**Explanation:** The license file to be installed does not exist.

**User Response:** Find the correct location of the license file to be installed, and retry the operation.

---

**2621-313** Configuration command not allowed - daemon is in replace configuration mode.

**Explanation:** The command being issued is not allowed in replace configuration mode.

**User Response:** Wait for the completion of replace configuration action.

---

**2621-314** Input to the "CompleteMigration" action is not valid. Current RSCT active version number is incompatible to this NewActiveVersion: "*version*"

**Explanation:** The current RSCT active version number is incompatible to the selected new RecoveryRM daemon active version number.

**User Response:** Correct the problem and retry the operation.

---

**2621-750** SA event with unknown reason in domain "*domain\_name*".

**Explanation:** Reason: SA\_UNKNOWN (reason unknown)

**User Response:**

---

**2621-751** SA detected status change for domain "*domain\_name*".

**Explanation:** Reason: (any domain state change)

**User Response:**

## Messages

---

**2621-753 SA Automation Manager started successfully in domain "domain\_name".**

**Explanation:** Reason: SA\_STARTED\_AUTOMATION\_MANAGER (sam master engine started)

**User Response:**

---

**2621-754 SA Automation Manager stopped successfully in domain "domain\_name".**

**Explanation:** Reason: SA\_STOPPED\_AUTOMATION\_MANAGER (sam master engine stopped)

**User Response:**

---

**2621-755 SA detected a configuration policy refresh in domain "domain\_name".**

**Explanation:** Reason: SA\_POLICY\_REFRESHED (configuration policy refreshed)

**User Response:**

---

**2621-756 SA detected a new configuration in domain "domain\_name".**

**Explanation:** Reason: SA\_CONFIGURATION\_CREATED (new resource or relationship)

**User Response:**

---

**2621-757 SA detected a configuration change in domain "domain\_name".**

**Explanation:** Reason: SA\_CONFIGURATION\_MODIFIED (any persistent attribute changed)

**User Response:**

---

**2621-758 SA detected a deleted configuration in domain "domain\_name".**

**Explanation:** Reason: SA\_CONFIGURATION\_DELETED (deleted resource or relationship)

**User Response:**

---

**2621-759 SA detected a status change of resource "resource\_name" in domain "domain\_name".**

**Explanation:** Reason: any SA\_STATUS\_xxx\_CHANGED (any state change)

**User Response:**

---

**2621-770 SA started automation operation for resource group "resource\_name" in domain "domain\_name".**

**Explanation:** Reason: SA\_STARTED\_AUTOMATION (CS: "Satisfactory or awaiting automation" -> "In Automation")

**User Response:**

---

**2621-771 SA completed automation operation for resource group "resource\_name" in domain "domain\_name".**

**Explanation:** Reason: SA\_FINISHED\_AUTOMATION (CS: "In Automation" -> "Satisfactory")

**User Response:**

---

**2621-772 SA detected that the system "system\_name" switched into maintenance mode by an operator command in domain "domain\_name".**

**Explanation:** Reason: SA\_STARTED\_MAINTENANCE (excluded flag "not excluded" -> "excluded")

**User Response:**

---

---

**2621-773** SA detected that the system "*system\_name*" switched back to automation mode by an operator command in domain "*domain\_name*".

**Explanation:** Reason: SA\_FINISHED\_MAINTENANCE (excluded flag "excluded" -> "not excluded")

**User Response:**

---

**2621-777** SA detected that the resource "*resource\_name*" has the requested state in domain "*domain\_name*".

**Explanation:** Reason: SA\_OK (no state in "problem", "failed", "sacrificed", "degraded", etc.)

**User Response:**

---

**2621-778** SA turned the resource "*resource\_name*" online as requested in domain "*domain\_name*".

**Explanation:** Reason: SA\_RESOURCE\_ONLINE

**User Response:**

---

**2621-779** SA turned the resource "*resource\_name*" offline as requested in domain "*domain\_name*".

**Explanation:** Reason: SA\_RESOURCE\_OFFLINE

**User Response:**

---

**2621-780** SA detected that the system "*system\_name*" joined the domain "*domain\_name*".

**Explanation:** Reason: SA\_SYSTEM\_JOINED\_CLUSTER (resource node OS "offline" -> "online")

**User Response:**

---

**2621-781** SA detected that the system "*system\_name*" left the domain "*domain\_name*".

**Explanation:** Reason: SA\_SYSTEM\_LEFT\_CLUSTER (resource node OS "online" -> "offline")

**User Response:**

---

**2621-782** SA detected a warning in domain "*domain\_name*" for resource "*resource\_name*".

**Explanation:** Reason: SA\_WARNING (not all states in SA\_OK and no state in SA\_PROBLEM)

**User Response:**

---

**2621-784** SA detected a warning in domain "*domain\_name*". The resource "*resource\_name*" has been degraded.

**Explanation:** Reason: SA\_RESOURCE\_DEGRADED (CS: Degraded, OS: Online, Degraded)

**User Response:**

---

**2621-785** SA detected a warning in domain "*domain\_name*". The resource "*resource\_name*" has a performance problem.

**Explanation:** Reason: SA\_RESOURCE\_ILL (CS: Degraded, OS: Online, Degraded, HS: Unsatisfied)

**User Response:**

---

**2621-786** SA detected a warning in domain "*domain\_name*". The resource "*resource\_name*" has problems caused by a dependency to another resource.

**Explanation:** Reason: SA\_RESOURCE\_INHIBITED (SS: Inhibited)

**User Response:**

---

## Messages

---

**2621-787** SA detected a problem in domain "*domain\_name*" for resource "*resource\_name*".

**Explanation:** Reason: SA\_PROBLEM (any state in "problem", "failed" or "sacrificed")

**User Response:**

---

**2621-788** SA detected a problem in domain "*domain\_name*". No online system found to start resource "*resource\_name*" based on the configuration.

**Explanation:** Reason: SA\_RESOURCE\_SACRIFICED (BS: "any state" -> "sacrificed")

**User Response:**

---

**2621-789** SA detected a problem in domain "*domain\_name*". The resource "*resource\_name*" can't be turned into the requested state.

**Explanation:** Reason: SA\_RESOURCE\_HUNG (CS: Problem, OS: Starting, Online, Degraded, Stopping, Problem)

**User Response:**

---

**2621-790** SA detected a problem in domain "*domain\_name*". The resource "*resource\_name*" needs a manual reset by the operator.

**Explanation:** Reason: SA\_RESOURCE\_BROKEN (CS: Problem, OS: Offline, Failed\_Offline)

**User Response:**

---

**2621-791** SA detected a problem in domain "*domain\_name*". The resource "*resource\_name*" has problems caused by a offline system.

**Explanation:** Reason: SA\_SYSTEM\_GONE (OS: "system gone" or AS: "system gone" + excluded: "not excluded")

**User Response:**

---

**2621-792** SA detected a problem in domain "*domain\_name*". The operational quorum is lost.

**Explanation:** Reason: SA\_NO\_QUORUM (cluster quorum "any state" -> "no quorum")

**User Response:**

---

**2622-001** *program\_name*: **2622-001 Not a recognized flag: *flag*.**

**Explanation:** The flag or combination of flags passed to *program\_name* was either an unsupported flag or an invalid combination of flags.

**User Response:** Refer to the Usage statement which lists the valid flag. Correct the command and re-enter.

---

**2622-002** *program\_name*: **2622-002 Flag "*flag*" is not allowed with the unsupported operand "*operand*".**

**Explanation:** The operand passed with the flag was an unsupported or invalid operand or combination of operands.

**User Response:** Refer to the Usage statement which lists the valid operand. Correct the command and re-enter.

---

**2622-003** *program\_name*: **2622-003 The combination of flags "*flag*" and "*flag or operand*" is not allowed.**

**Explanation:** The combination of the specified flags or flag and operand is not allowed.

**User Response:** Either choose a different flag or do not enter the operand. Refer to the Usage statement which lists the valid flag, operands, and flag operand combinations.

---

---

**2622-004** *program\_name:* **2622-004 The "flag" flag is required.**

**Explanation:** The indicated flag must be specified when using this command.

**User Response:** Check the command syntax and issue the command again.

---

**2622-005** *program\_name:* **2622-005 The "flag" flag requires an operand.**

**Explanation:** The indicated flag must be specified with an operand when using this command.

**User Response:** Check the command syntax and issue the command again.

---

**2622-006** *program\_name:* **2622-006 There are no resource groups defined.**

**Explanation:** There were no Resources Groups.

**User Response:** Add Resources Groups.

---

**2622-007** *program\_name:* **2622-007 There are no member resources defined.\n**

**Explanation:** There were no member Resources.

**User Response:** Add member resources to groups.

---

**2622-008** *program\_name:* **2622-008 Not a valid number of operands.**

**Explanation:** *program\_name* was called with an invalid number of operands.

**User Response:** Refer to the Usage statement which lists the valid operand. Correct the command and re-enter.

---

**2622-009** *program\_name:* **2622-009 An unexpected RMC error occurred. The RMC return code was *rmc\_return\_code*.**

**Explanation:** A call to an RMC CLI script returned an unexpected error.

**User Response:** Check to see if RMC is operational and retry the command.

---

**2622-010** *program\_name:* **2622-010 error\_exit received an unexpected return code *return\_code*.**

**Explanation:** All IBM Tivoli System Automation for Multiplatforms return codes should be converted to an appropriate CLI (command line interface) return code. The actual API return code or other internal subroutine return code is printed. It is treated as a severe error and processing of the command stops.

**User Response:** This is an internal script error and should not occur. Report the command name, and this message to the IBM Support Center at your convenience.

---

**2622-011** *program\_name:* **2622-011 The specified selection string "*selection\_string*" did not match any resource groups.**

**Explanation:** The specified selection string did not match any resource groups.

**User Response:** Check the selection string and issue the command again.

---

**2622-012** *program\_name:* **2622-012 The resource group "*resource\_group*" already exists.**

**Explanation:** The resource group specified already exists.

**User Response:** Check the resource group name and issue the command again.

---

## Messages

---

**2622-013**    *program\_name:* **2622-013 The resource group "*resource\_group*" does not exist.**

**Explanation:** The specified resource group name does not exist.

**User Response:** Check the resource group name and issue the command again.

---

**2622-014**    *program\_name:* **2622-014 The resource "*resource\_name*" not found in the class "*resource\_class*".**

**Explanation:** The resource specified does not belong in the specified class.

**User Response:** Check the resource and class name combination and issue the command again.

---

**2622-015**    *program\_name:* **2622-015 None of the resources with the specified types exists in their respective classes.**

**Explanation:** All the resources specified were not found in their respective classes

**User Response:** Make sure the resources with the specified types are part of their classes and then re-enter command.

---

**2622-016**    *program\_name:* **2622-016 No resources were found with the selection string "*selection\_string*" from the class "*resource\_class*".**

**Explanation:** No resources were found using the specified selection string.

**User Response:** Check the class name and selection string combination and then issue the command again.

---

**2622-017**    *program\_name:* **2622-017 Resource "*resource*" already exists as a Member of Group "*resource\_group*".**

**Explanation:** Resource already exist as a member resource.

**User Response:** No action required.

---

**2622-018**    *program\_name:* **2622-018 Resource "*resource*" is not a member resource of Resource Group "*resource\_group*".**

**Explanation:** Resource is not a member resource of the specified resource group.

**User Response:** May be required to add this resource to the specified resource group.

---

**2622-019**    *program\_name:* **2622-019 All of the specified resources are already managed resources.**

**Explanation:** All member resources already exist as a managed Resource.

**User Response:** No action required.

---

**2622-020**    *program\_name:* **2622-020 None of the resources specified are member resources of resource group "*resource\_group*".**

**Explanation:** All member resources don't exist for the specified resource group.

**User Response:** Add these resources as member resources.

---

**2622-021**    *program\_name:* **2622-021 Multiple entries of the flag *flag* not allowed.**

**Explanation:** User had entered the same flag with operands more than once.

**User Response:** Re-enter the command with only one occurrence of the concerned flag.

---

---

**2622-022** *program\_name:* **2622-022 None of the resources specified are managed resources.**

**Explanation:** None of the resources specified exist as managed resources.

**User Response:** Check resources and re-enter with valid resources.

---

**2622-023** *program\_name:* **2622-023 Resource "resource" is not a managed resource.**

**Explanation:** Specified resource is not a managed resource.

**User Response:** Check Resources and re-enter with valid resources.

---

**2622-024** *program\_name:* **2622-024 Either flag "flag1" or "flag2" or "flag3" is required with the -N flag.**

**Explanation:** One the three flags are required when using current flag.

**User Response:** Add the required flags to the command and try again.

---

**2622-025** *program\_name:* **2622-025 The specified source resource "source\_resource" does not exist.**

**Explanation:** The specified source resource name does not exist in the specified class.

**User Response:** Check the resource name and class name combination and then issue the command again.

---

**2622-026** *program\_name:* **2622-026 The specified source selection string "source\_string" did not match any resource.**

**Explanation:** The specified source selection string did not match any resource in the specified class.

**User Response:** Check the selection string or class name combination and then issue the command again.

---

**2622-027** *program\_name:* **2622-027 More than one relations were found with the same name "source\_resource".**

**Explanation:** More than one relations was returned for the source when the operation only requires one relationship.

**User Response:** Make the query more specific and issue the command again.

---

**2622-028** *program\_name:* **2622-028 Name change not allowed for more than one relations that matched from "query".**

**Explanation:** The selection string or query returned more than one source resource. Relation name has to be unique.

**User Response:** Check the selection string or query and issue the command again.

---

**2622-029** *program\_name:* **2622-029 The specified equivalency "equivalency" already exists.**

**Explanation:** Specified equivalency already exists.

**User Response:** Re-enter the command with an equivalency name that does not already exist.

---

**2622-030** *program\_name:* **2622-030 The specified equivalency "equivalency" does not exist.**

**Explanation:** Specified equivalency does not exist.

**User Response:** Make sure the equivalency name already exist, re-issue the command.

---

**2622-031** *program\_name:* **2622-031 The node "node\_name" does not exist in the cluster.**

**Explanation:** The node specified does not belong in the cluster.

**User Response:** Check the node name and issue the command again.

---

## Messages

---

**2622-032** *program\_name:* **2622-032 Resource "resource\_name" of resource class "class\_name" is not from the allowed nodes for resource group "Resource\_group\_name".**

**Explanation:** Member resource is not from the allowed nodes for the new resource group.

**User Response:** Member resources must be part of the allowed node list of resource group.

---

**2622-033** *program\_name:* **2622-033 Resource Group "resource\_group" has no member resources.**

**Explanation:** The specified resource group name does not contain any member resources.

**User Response:** Check the resource group name and issue the command again.

---

**2622-034** *program\_name:* **2622-034 No resource groups were found that matched the specified query.**

**Explanation:** No existing resource groups matched the specified Query.

**User Response:** Check the resource group name and issue the command again.

---

**2622-035** *program\_name:* **2622-035 No member resources were found that matched the specified query.**

**Explanation:** No existing member resources matched the specified query.

**User Response:** Check the selection string or resource group name and issue the command again.

---

**2622-036** *program\_name:* **2622-036 actual\_error\_message**

**Explanation:** Display the error message from lsrsrc-api.

**User Response:** Check the error Message for appropriate action.

---

**2622-037** *program\_name:* **2622-037 Invalid user error. The return code was return Code**

**Explanation:** Command line interface user Error.

**User Response:** Check the error message for appropriate action.

---

**2622-038** *program\_name:* **2622-038 No resources were found. The return code was return Code**

**Explanation:** No resources were found.

**User Response:** Check the error message for appropriate action.

---

**2622-039** *program\_name:* **2622-039 No resources were specified for class class\_name**

**Explanation:** No resources were specified for the class in the command line.

**User Response:** Repeat the command with resources associated with the specified class.

---

**2622-040** *program\_name:* **2622-040 Multiple resources of the same type were matched for resource "resource\_name" from class "resource\_class**

**Explanation:** Multiple resources of the same resource type were found for the specified resource.

**User Response:** Make sure only one resource is matched with same Type and then re-enter the command.

---

**2622-901** *program\_name:* **2622-901 Attributes and Arguments are mismatched.**

**Explanation:** Arguments are missing for some of the attributes in the input file.

**User Response:** Repeat the command by updating the input file.

---

---

**2622-902** *program\_name:* **2622-902 Do not specify attribute *attribute* when defining resource *resource\_class***  
*row:* *row*.

**Explanation:** Arguments are missing for some of the attributes in the input file.

**User Response:** Repeat the command by updating the input file.

---

**2622-903** *program\_name:* **2622-903 Error processing attribute *attribute\_name* value "*attribute\_value*".**

**Explanation:** The value for this particular attribute is not valid. See the preceding error message for additional details.

**User Response:** Verify the data type for this particular attribute using the `lsrsrcdef` command. Verify and correctly specify the value so that it is appropriate for this data type. For example, if the data type is an array, the values must be enclosed within curly braces { }. See the appropriate command man page and the man page for the Resource Data Input File for valid syntax values.

---

**2622-904** *program\_name:* **2622-904 Required argument *argument\_name* *argument name* (SD element name) for resource *resource\_class* name of the resource class must be specified.**

**Explanation:** The specified command arguments for this resource class and this command must be specified.

**User Response:** Run `lscmdargdef` command to see the list of argument names and data types expected as input for this command.

---

**2622-905** *program\_name:* **2622-905 No resources were entered for resource class "*resource\_class*".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-906** *program\_name:* **2622-906 Failed to create some or all resources from input file "*input\_file\_name*".**

**Explanation:** Failed to create some or all resources from input file for specified class.

**User Response:** Check the file data, correct if possible and re-run command.

---

**2622-907** *program\_name:* **2622-907 Fixed resource *resource* on node *node\_name* is not a managed resource.**

**Explanation:** Specified fixed resource is not a managed resource.

**User Response:** Check Resources and re-enter with valid resources.

---

**2622-908** *program\_name:* **2622-908 Resource "*resource\_name*" of resource class "*class\_name*" is not from the allowed nodes for the Top resource group "*Top\_Resource\_group\_name*".**

**Explanation:** Member resource is not from the allowed nodes for the new resource group's top most group.

**User Response:** Member resources must be part of the allowed node list of resource group and its top most group.

---

**2622-041** *program\_name:* **2622-041 No resources were entered for resource class "*resource\_class*".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-042** *program\_name:* **2622-042 No selection string was entered for resource class "*resource\_class*".**

**Explanation:** Selection string was not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

## Messages

---

**2622-061** *program\_name:* **2622-061 The specified resource group input file "input\_file\_name" does not exist.**

**Explanation:** Specified resource file name was not found.

**User Response:** Make sure the resource group file name exist, re-enter the command.

---

**2622-062** *program\_name:* **2622-062 Failed to create resource groups from the specified resource group input file "input\_file\_name".**

**Explanation:** Failed to create resource groups from the specified resource file.

**User Response:** Correct the associated error with this and, re-enter the command.

---

**2622-063** *program\_name:* **2622-063 None of the nodes specified exists in the cluster.**

**Explanation:** The specified node names do not exist in the cluster.

**User Response:** Check the node names and issue the command again.

---

**2622-081** *program\_name:* **2622-081 None of the specified resource groups were found or could not be removed.**

**Explanation:** None of the specified resource group or selection string matched any existing resource groups or some of them just couldn't be removed.

**User Response:** Check the message that preceded this message for the actual cause, if resource groups already exists.

---

**2622-082** *program\_name:* **2622-082 Resource group "resource\_group" does not exist or could not be removed.**

**Explanation:** The specified resource group was not found or even if it was found it just couldn't be removed.

**User Response:** Check the message that preceded this message for the actual cause, if resource group already exists.

---

**2622-083** *program\_name:* **2622-083 Resource Groups matched by the specified selection string "selection\_string" could not be removed or did not exist.**

**Explanation:** The specified select string did not match any existing group or even if it was found it just couldn't be removed.

**User Response:** Check the message that preceded this message for the actual cause, if resource group already exists.

---

**2622-101** *program\_name:* **2622-101 Must change attributes of resource group "resource\_group".**

**Explanation:** At least one of the resource group attributes must be changed when using chrg command.

**User Response:** Enable the appropriate options to change required attributes of resource group.

---

**2622-102** *program\_name:* **2622-102 Cannot change name of more than one resource group.**

**Explanation:** Can only change name of one resource group at a time.

**User Response:** Repeat the command with only one resource group.

---

**2622-103** *program\_name:* **2622-103 None of the specified resource groups were found or could not be changed.**

**Explanation:** None of the specified resource group or selection string matched any existing resource groups or some of them just couldn't be changed.

**User Response:** Check the message that preceded this message for the actual cause, if resource groups already exists.

---

**2622-104** *program\_name:* **2622-104 Resource Group "resource\_group" does not exist or could not be changed.**

**Explanation:** The specified resource group was not found or even if it was found it just couldn't be changed.

**User Response:** Check the message that preceded this message for the actual cause, if resource group already exists.

---

**2622-105** *program\_name:* **2622-105 Allowed node of Resource Group "resource\_group" could not be changed to node "node\_name".**

**Explanation:** The specified resource group's allowed node could not be changed to the specified node because its members are not from this node.

**User Response:** Specify a node or equivalency that contains all the nodes of the resource groups members.

---

**2622-106** *program\_name:* **2622-106 Allowed node of Resource Group "resource\_group" could not be changed to the equivalency "equiv\_name".**

**Explanation:** The specified resource group's allowed node could not be changed to the specified equivalency because its members are not from the nodes in the equivalency.

**User Response:** Specify a node or equivalency that contains all the nodes of the resource groups members.

---

**2622-107** *program\_name:* **2622-107 Allowed node of all the specified resource groups could not be changed.**

**Explanation:** The specified resource group's allowed node could not be changed to the specified equivalency or node because its members are not from the node or nodes in the equivalency.

**User Response:** Specify a node or equivalency that contains all the nodes of the resource groups members.

---

**2622-108** *program\_name:* **2622-108 The specified node "excluded\_node" does not exist in the list.**

**Explanation:** The specified node name does not exist in the excluded node list.

**User Response:** Check the node name and issue the command again.

---

**2622-109** *program\_name:* **2622-109 The specified node "excluded\_node" already exists in the list.**

**Explanation:** The specified node name already exists in the excluded node list.

**User Response:** Check the node name and issue the command again.

---

**2622-110** *program\_name:* **2622-110 None of the nodes specified exists in the list.**

**Explanation:** The specified node names do not exist in the excluded node list.

**User Response:** Check the node names and issue the command again.

---

**2622-111** *program\_name:* **2622-111 None of the nodes specified exists in the cluster.**

**Explanation:** The specified node names do not exist in the cluster.

**User Response:** Check the node names and issue the command again.

---

**2622-112** *program\_name:* **2622-112 All the nodes specified already exists in the list.**

**Explanation:** All the nodes specified already exist in the excluded node list.

**User Response:** Check the node names and issue the command again.

## Messages

---

**2622-113** *program\_name:* **2622-113 Too many groups were specified to change to the new ExcludedList.**

**Explanation:** Only one group name can be specified to change the excluded node list attribute.

**User Response:** Check the group name and issue the command again.

---

**2622-121** *program\_name:* **2622-121 No resources were entered for resource class "resource\_class".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-122** *program\_name:* **2622-122 No Selection string entered for Resource class "resource\_class".**

**Explanation:** Selection string was not entered with the command for the specified class.

**User Response:** Enter the selection string and retry the command.

---

**2622-123** *program\_name:* **2622-123 None of the specified member resources are from the allowed nodes for resource group "resource\_group".**

**Explanation:** All of the specified resources are not in the allowed node list of the new resource group.

**User Response:** Member resources must be part of the allowed node list of resource group.

---

**2622-124** *program\_name:* **2622-124 The specified managed resource input file "input\_file\_name" does not exist.**

**Explanation:** Specified resource file name was not found.

**User Response:** Make sure the managed resource file name exist, re-enter the command.

---

**2622-125** *program\_name:* **2622-125 Failed to create resources from the specified managed resource input file "input\_file\_name".**

**Explanation:** Failed to create resources from the specified resource file.

**User Response:** Correct the associated error with this and, re-enter the command.

---

**2622-126** *program\_name:* **2622-126 None of the nodes specified exists in the cluster.**

**Explanation:** The specified node names do not exist in the cluster.

**User Response:** Check the node names and issue the command again.

---

**2622-127** *program\_name:* **2622-127 Fixed resource "resource\_name" is not allowed to have an ExcludedList.**

**Explanation:** The specified fixed resource can not be moved, therefore point less to carry an ExcludedList.

**User Response:** Add all the fixed resources without an ExcludedList.

---

**2622-128** *program\_name:* **2622-128 None of the specified managed resources were created.**

**Explanation:** None of the specified managed resources were created because they were fixed resources.

**User Response:** Add all the fixed resources without an ExcludedList or Policy.

---

**2622-141** *program\_name:* **2622-141 No resources were entered for resource class "resource\_class".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

---

**2622-142** *program\_name:* **2622-142 No selection string was entered for resource class "resource\_class".**

**Explanation:** Selection string was not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-161** *program\_name:* **2622-161 Must change attributes of member resources.**

**Explanation:** At least one of the member resource attributes must be changed when using chrgmbr command.

**User Response:** Enable the appropriate options to change required attributes of resource group.

---

**2622-162** *program\_name:* **2622-162 No resources were entered for resource class "resource\_class".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-163** *program\_name:* **2622-163 No selection string was entered for Resource class "resource\_class".**

**Explanation:** Selection string was not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-164** *program\_name:* **2622-164 None of the specified member resources are from the allowed nodes for resource group "resource\_group".**

**Explanation:** All of the specified resources are not in the allowed node list of the new resource group.

**User Response:** Member resources must be part of the allowed node list of resource group.

---

**2622-165** *program\_name:* **2622-165 The specified node "excluded\_node" does not exist in the list.**

**Explanation:** The specified node name does not exist in the excluded node list.

**User Response:** Check the node name and issue the command again.

---

**2622-166** *program\_name:* **2622-166 The specified node "excluded\_node" already exists in the list.**

**Explanation:** The specified node name already exists in the excluded node list.

**User Response:** Check the node name and issue the command again.

---

**2622-167** *program\_name:* **2622-167 None of the nodes specified exists in the list.**

**Explanation:** The specified node names do not exist in the excluded node list.

**User Response:** Check the node names and issue the command again.

---

**2622-168** *program\_name:* **2622-168 None of the nodes specified exists in the cluster.**

**Explanation:** The specified node names do not exist in the cluster.

**User Response:** Check the node names and issue the command again.

---

**2622-169** *program\_name:* **2622-169 All the nodes specified already exists in the list.**

**Explanation:** All the nodes specified already exist in the excluded node list.

**User Response:** Check the node names and issue the command again.

---

## Messages

---

**2622-170** *program\_name:* **2622-170 Too many resources were specified to change to the new ExcludedList.**

**Explanation:** Only one resource can be specified to change the excluded node list attribute.

**User Response:** Check the resource count and issue the command again.

---

**2622-171** *program\_name:* **2622-171 Fixed resource "resource\_name" is not allowed to have an ExcludedList or Policy.**

**Explanation:** The specified fixed resource can not be moved, therefore point less to carry an ExcludedList or SelectFromPolicy.

**User Response:** Change all the fixed resources without an ExcludedList or Policy.

---

**2622-172** *program\_name:* **2622-172 None of the specified managed resources were created.**

**Explanation:** None of the specified managed resources were created because they were fixed resources.

**User Response:** Change all the fixed resources without an ExcludedList or Policy.

---

**2622-181** *program\_name:* **2622-181 The specified Source resource source\_resource is not a member resource.**

**Explanation:** specified Source Resource is not a member resource.

**User Response:** Make sure the source is a member resource, then re-issue the command.

---

**2622-182** *program\_name:* **2622-182 The specified source resource matched from selection string "source\_string" is not a member resource.**

**Explanation:** specified source resource matched from selection string is not a member resource.

**User Response:** Make sure the source is a member resource, then re-issue the command.

---

**2622-183** *program\_name:* **2622-183 Source resource "source\_resource" already exists.**

**Explanation:** Source resource cannot already exist when creating it.

**User Response:** use chrel to make changes to the relationships related to this source resource.

---

**2622-184** *program\_name:* **2622-184 Source resource matched from selection string "source\_resource" already exists.**

**Explanation:** Source resource cannot already exist when creating it.

**User Response:** use chrel to make changes to the relations related to this source resource.

---

**2622-185** *program\_name:* **2622-185 None of the specified source resources Exists.**

**Explanation:** There were no source resources.

**User Response:** None.

---

**2622-186** *program\_name:* **2622-186 None of the specified target resources Exists.**

**Explanation:** There were no target resources.

**User Response:** No action is required.

---

---

**2622-187** *program\_name:* **2622-187 More than one source resources were matched for "source".\n**

**Explanation:** More than one source was matched for specified source name.

**User Response:** No action is required.

---

**2622-188** *program\_name:* **2622-188 More than one source resource were matched using selection string "selection\_string" on class\_name class .\n**

**Explanation:** More than one source was matched with source selection string.

**User Response:** No action is required.

---

**2622-189** *program\_name:* **2622-189 Specified condition is not allowed with the relationship "relationship".\n**

**Explanation:** Only locate-with relationships are allowed to have conditions.

**User Response:** No action is required.

---

**2622-190** *program\_name:* **2622-190 The specified managed relationship input file "input\_file\_name" does not exist.**

**Explanation:** Specified managed resource file name was not found.

**User Response:** Make sure the managed relationship file name exist, re-enter the command.

---

**2622-191** *program\_name:* **2622-191 Failed to create resources from the specified relationship input file "input\_file\_name".**

**Explanation:** Failed to create resources from the specified relationship input file.

**User Response:** Correct the associated error with this and, re-enter the command.

---

**2622-201** *program\_name:* **2622-201 No relationships matched the query "source\_query".**

**Explanation:** Source query did not match any existing relationships.

**User Response:** Check the query and re-run the command.

---

**2622-202** *program\_name:* **2622-202 There are no relationships defined for source "source" matched from selection string "selection\_string".**

**Explanation:** Source resource must already have a relationship defined.

**User Response:** use mkrel to make relationships related to this source resource.

---

**2622-203** *program\_name:* **2622-203 None of the specified target resources exists.**

**Explanation:** There were no target resources.

**User Response:** No action is required.

---

**2622-204** *program\_name:* **2622-204 None of the specified target resources have relationships with the source.**

**Explanation:** The specified target resources were not related to the source.

**User Response:** Re-enter the command with existing target resources for the source.

---

## Messages

---

**2622-205** *program\_name: 2622-205* **All of the specified target resources already have relationships with the source.**

**Explanation:** All specified target resources already have relationship with the source.

**User Response:** Re-enter the command with target resources that don't already exist for the source.

---

**2622-206** *program\_name: 2622-206* **Target resource *target\_resource* already has relationship with the source.**

**Explanation:** Target resource already maintains relationship with the Source.

**User Response:** Use the delete target resource option to remove targets or do nothing.

---

**2622-207** *program\_name: 2622-207* **Target resource "*target\_resource*" does not have any relationship with the source.**

**Explanation:** Target resource does not maintain any relationship with the Source.

**User Response:** Use mkrel to make relationships related to this source resource.

---

**2622-208** *program\_name: 2622-208* **At least one change must be made to the relationship.**

**Explanation:** No changes were requested by the command.

**User Response:** Use chrel to change one or more attributes of the relations related to this source resource.

---

**2622-209** *program\_name: 2622-209* **There are no relations defined for sources matched from selection string "*selection\_string*".**

**Explanation:** Selection string did not match a relationship.

**User Response:** Use mkrel to make relationships related to this source resource.

---

**2622-210** *program\_name: 2622-210* **The selection string "*selection\_string*" did not match any relationships.**

**Explanation:** Selection string did not match a relationship.

**User Response:** Re-enter command with string that will match at least a relationship.

---

**2622-211** *program\_name: 2622-211* **The managed relation "*managed\_relation*" does not exist as a relationship.**

**Explanation:** The managed relation specified does not exist.

**User Response:** Re-enter command with an existing managed relationship.

---

**2622-212** *program\_name: 2622-212* **None of the queried target resources Exists.**

**Explanation:** The target resources used to query the relationships do not exist.

**User Response:** No action required.

---

**2622-213** *program\_name: 2622-213* **None of the specified source resources exists.**

**Explanation:** There were no source resources.

**User Response:** No action is required.

---

**2622-214** *program\_name: 2622-214* **Specified condition is not allowed with the relationship "*relationship*".\n**

**Explanation:** Only locate-with relationships are allowed to have conditions.

**User Response:** No action is required.

---

---

**2622-221** *program\_name:* **2622-221 No relationships matched the query "source\_query."**

**Explanation:** Source query did not match any existing relationships.

**User Response:** Check the query and re-run the command.

---

**2622-222** *program\_name:* **2622-222 There are no relationships defined for source resource "source" matched from selection string "selection\_string".**

**Explanation:** Source resource must already have a relationship defined.

**User Response:** Use mkrel to make relationships related to this source resource.

---

**2622-223** *program\_name:* **2622-223 There are no relationships defined for any of the specified source resources.**

**Explanation:** Source resource must already have a relationship defined.

**User Response:** Use mkrel to make relationships related to this source resource.

---

**2622-224** *program\_name:* **2622-224 The selection string "selection\_string" did not match any managed relations.**

**Explanation:** Selection string did not match any managed relations.

**User Response:** Re-enter the command with string that will match at least one managed relations.

---

**2622-225** *program\_name:* **2622-225 None of the specified relation "relation\_names" exists.**

**Explanation:** All the managed relations specified does not exist.

**User Response:** Re-enter the command with at least one existing managed relations.

---

**2622-226** *program\_name:* **2622-226 None of the specified queried target resources exists.**

**Explanation:** The target resources used to query the relationships do not exist.

**User Response:** No action is required.

---

**2622-227** *program\_name:* **2622-227 None of the specified source resources exists.**

**Explanation:** There were no source resources.

**User Response:** No action is required.

---

**2622-228** *program\_name:* **2622-228 Relationship "relation\_name" does not exist or could not be removed.**

**Explanation:** Specified relationship was not found or even if it was found it just couldn't be removed.

**User Response:** Check the message that preceded this message for the actual cause if already exists.

---

**2622-229** *program\_name:* **2622-229 None of the specified Relationships were found or could not be removed.**

**Explanation:** None of the specified relationships or selection string matched any existing relationships or some of them just couldn't be removed.

**User Response:** Check the message that preceded this message for the actual cause.

---

## Messages

---

**2622-261** *program\_name:* **2622-261 The specified logicdeck file "*logicdeck\_filename*" with the "*option\_flag*" flag does not exist.**

**Explanation:** The specified logic deck file name does not exist in the specified location.

**User Response:** Check the logic deck file name and issue the command again.

---

**2622-262** *program\_name:* **2622-262 At least one change must be made to the IBM Tivoli System Automation for Multiplatforms Controls.**

**Explanation:** No changes were requested by the command.

**User Response:** Use appropriate samctrl flags to change one or more sam controls attributes.

---

**2622-263** *program\_name:* **2622-263 The specified node "*excluded\_node*" does not exist in the list.**

**Explanation:** The specified node name does not exist in the excluded node list.

**User Response:** Check the node name and issue the command again.

---

**2622-264** *program\_name:* **2622-264 The specified node "*excluded\_node*" already exists in the list.**

**Explanation:** The specified node name already exists in the excluded node list.

**User Response:** Check the node name and issue the command again.

---

**2622-265** *program\_name:* **2622-265 None of the nodes specified exists in the list.**

**Explanation:** The specified node names do not exist in the excluded node list.

**User Response:** Check the node names and issue the command again.

---

**2622-266** *program\_name:* **2622-266 None of the nodes specified exists in the cluster.**

**Explanation:** The specified node names do not exist in the cluster.

**User Response:** Check the node names and issue the command again.

---

**2622-267** *program\_name:* **2622-267 All the nodes specified already exists in the list.**

**Explanation:** All the nodes specified already exist in the excluded node list.

**User Response:** Check the node names and issue the command again.

---

**2622-268** *program\_name:* **2622-268 Specified engine action failed.**

**Explanation:** Specified engine action failed to process successfully.

**User Response:** No action required.

---

**2622-269** *Program\_name:* **2622-269 Migration action failed with return code "*return\_code*".**

**Explanation:** Specified Migration action failed to process successfully.

**User Response:** No action required.

---

**2622-270** *Program\_name:* **2622-270 Install License Action failed with return code "*return\_code*".**

**Explanation:** Install License action failed to process successfully.

**User Response:** No action required.

---

---

**2622-271** *program\_name: 2622-271 Specified license file "FileName" does not exist.*

**Explanation:** License file could not be found at specified location.

**User Response:** Re-enter the command with a valid file name.

---

**2622-281** *program\_name: 2622-281 The specified equivalency input file "input\_file\_name" does not exist.*

**Explanation:** Specified equivalency file name was not found.

**User Response:** Make sure the equivalency file name exist, re-enter the command.

---

**2622-282** *program\_name: 2622-282 No resources were entered for resource class "resource\_class".*

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-283** *program\_name: 2622-283 No Selection string was entered for Resource class "resource\_class".*

**Explanation:** Selection string was not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-284** *program\_name: 2622-284 Policy other than Any is not allowed with option "option\_flag".*

**Explanation:** Selection string was not entered with SelectFromPolicy Any.

**User Response:** Retry making equivalency with Policy Any.

---

**2622-285** *program\_name: 2622-285 Failed to create resources from the specified equivalency input file "input\_file\_name".*

**Explanation:** Failed to create resources from the specified equivalency input file.

**User Response:** Correct the associated error with this and, re-enter the command.

---

**2622-286** *program\_name: 2622-286 All of the specified members for equivalency "equivalency\_name" are floating resources.*

**Explanation:** All of the specified members are floating resources, but as a rule for equivalency they can not be included as a members of an equivalency

**User Response:** Add at least a fixed resources as a member of an equivalency.

---

**2622-301** *program\_name: 2622-301 The equivalency "equivalency" already has a SelectString defined.*

**Explanation:** There is already a SelectString defined therefore won't allow an add or delete function.

**User Response:** Try using the overwrite function for command.

---

**2622-302** *program\_name: 2622-302 No changes were made to the equivalency "equivalency".*

**Explanation:** Specified equivalency did not change.

**User Response:** Re-enter the command based on the information that preceded this error.

---

**2622-303** *program\_name: 2622-303 At least one change must be made to the equivalency "equivalency".*

**Explanation:** Must make a change when using this command.

**User Response:** Specify a change to the equivalency when issuing the command.

---

## Messages

---

**2622-304** *program\_name:* **2622-304 No resources were entered for resource class "resource\_class".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-305** *program\_name:* **2622-305 No selection string was entered for resource class "resource\_class".**

**Explanation:** Selection string was not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-306** *program\_name:* **2622-306 Policy other than Any is not allowed with option "option\_flag".**

**Explanation:** Selection string was not entered with SelectFromPolicy Any.

**User Response:** Retry changing equivalency with Policy Any.

---

**2622-307** *program\_name:* **2622-307 Equivalency "equivalency" does not exist or could not be changed.**

**Explanation:** Specified equivalency was not found or even if it was found it just could not be changed.

**User Response:** Check the message that preceded this message for the actual cause if it already exists.

---

**2622-308** *program\_name:* **2622-308 All of the specified members for equivalency "equivalency\_name" are floating resources.**

**Explanation:** All of the specified members are floating resources, but as a rule for equivalency they can not be included as a members of an equivalency. The members will not be updated and will be left alone.

**User Response:** Add at least a fixed resources as a member of an equivalency.

---

**2622-321** *program\_name:* **2622-321 None of the specified equivalencies were found or could not be removed.**

**Explanation:** None of the specified equivalencies were found or just could not be removed.

**User Response:** Check the message that preceded this message for the actual cause.

---

**2622-322** *program\_name:* **2622-322 Equivalency "equivalency" does not exist or could not be removed.**

**Explanation:** Specified equivalency was not found or even if it was found it just could not be removed.

**User Response:** Check the message that preceded this message for the actual cause if it already exists.

---

**2622-323** *program\_name:* **2622-323 The selection string "selection\_string" did not match any equivalencies.**

**Explanation:** The specified selection string did not match any equivalencies.

**User Response:** Check the equivalency select string and issue the command again.

---

**2622-341** *program\_name:* **2622-341 No relationships matched the query "source\_Query".**

**Explanation:** Source query did not match any existing relationships.

**User Response:** Check the query and re-run the command.

---

**2622-342** *program\_name:* **2622-342 There are no relationships defined that matched from selection string "selection\_string".**

**Explanation:** Selection string did not match to an existing relationship.

**User Response:** use mkrel to make this relationships.

---

---

**2622-343**    *program\_name:* **2622-343 The managed relation "selection\_string" does not exist.**

**Explanation:** The managed relation specified does not exist.

**User Response:** Re-enter command with an existing managed relationship.

---

**2622-344**    *program\_name:* **2622-344 There are no relationships defined.**

**Explanation:** There are no managed relations currently defined.

**User Response:** Re-enter command with an existing managed relationship.

---

**2622-345**    *program\_name:* **2622-345 The specified target selection string "target\_string" did not match any resource.**

**Explanation:** The specified target selection string did not match any resource in the specified class.

**User Response:** Check the selection string or class name combination and issue the command.

---

**2622-346**    *program\_name:* **2622-346 The specified target resources "target\_resource" does not exist.**

**Explanation:** The specified resource was not found in the specified class.

**User Response:** Check the resource and issue the command again.

---

**2622-347**    *program\_name:* **2622-347 None of the specified queried target resources exists.**

**Explanation:** The Target resources used to query the relationships does not exist in the relationship.

**User Response:** No action is required.

---

**2622-348**    *program\_name:* **2622-348 None of the specified relationships exists.**

**Explanation:** None of the relationships specified exists.

**User Response:** No action is required.

---

**2622-349**    *program\_name:* **2622-349 None of the specified source resources Exists.**

**Explanation:** There were no source resources.

**User Response:** No action is required.

---

**2622-361**    *program\_name:* **2622-361 There are no equivalencies defined that matched from selection string "selection\_string".**

**Explanation:** No equivalencies matched the select string.

**User Response:** No action is required.

---

**2622-362**    *program\_name:* **2622-362 The equivalency "equivalency" does not exist.**

**Explanation:** The equivalency specified does not exist.

**User Response:** Re-enter command with an existing equivalency.

---

**2622-363**    *program\_name:* **2622-363 There are No Equivalencies defined.**

**Explanation:** There are no equivalencies currently defined.

**User Response:** Re-enter command after creating an existing equivalency.

---

## Messages

---

**2622-381** *program\_name:* **2622-381 List License Action failed with return code "return\_code".**

**Explanation:** List License action failed to process successfully.

**User Response:** No action required.

---

**2622-401** *program\_name:* **2622-401 No resources were entered for resource class "resource\_class".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-402** *program\_name:* **2622-402 Samdiag action failed with return code "error\_code".**

**Explanation:** Samdiag action failed to acquire information on specified resource.

**User Response:** Refer to related error messages or return code and then rerun command \n based on those messages.

---

**2622-403** *program\_name:* **2622-403 Resource handle "resource\_handle" could not be resolved.**

**Explanation:** Resource handle used to acquire information could not be resolved to an actual resource.

**User Response:** Enter resource handles that exist.

---

**2622-421** *program\_name:* **2622-421 Samcfg action failed with return code "error\_code".**

**Explanation:** Samcfg action failed for the specified reason.

**User Response:** Refer to related error messages or return code and then rerun command \n based on those messages.

---

**2622-422** *program\_name:* **2622-422 Specified file "resource\_handle" is not in the IBM Tivoli System Automation for Multiplatforms Configuration file format.**

**Explanation:** File associated with the command does not seem to be the format designed for IBM Tivoli System Automation for Multiplatforms Configuration.

**User Response:** Correct the format of file if possible and re-run command with file again.

---

**2622-423** *program\_name:* **2622-423 Invalid Command "Command" in file "FileName".**

**Explanation:** Command found in configuration file is not valid.

**User Response:** Correct the entry in file and then re-run the command again.

---

**2622-424** *program\_name:* **2622-424 Specified file "FileName" not found.**

**Explanation:** Configuration file could not be found at the specified location.

**User Response:** Enter the correct path for the file and re-run the command.

---

**2622-425** *program\_name:* **2622-425 None of the clusters are online on this node.**

**Explanation:** None of the clusters defined on this node are online.

**User Response:** Bring this node online in a cluster, and re-run the command.

---

**2622-426** *program\_name:* **2622-426 Specified file "FileName" does not exist.**

**Explanation:** Restore file could not be found at specified location.

**User Response:** Re-enter the command with a valid file name.

---

---

**2622-427** *program\_name: 2622-427 Specified file "FileName" did not contain any IBM Tivoli System Automation for Multiplatforms commands.*

**Explanation:** Restore file were missing commands.

**User Response:** Re-enter the command with a valid file name containing commands.

---

**2622-428** *program\_name: 2622-428 Timed out waiting for user prompt while restoring from file "FileName".*

**Explanation:** User did not respond in 90 secs when prompted to continue restoration.

**User Response:** Re-enter the command but respond as when prompted promptly.

---

**2622-429** *program\_name: 2622-429 Error opening specified file "FileName": Error.*

**Explanation:** Specified file could not be opened.

**User Response:** Re-enter the command after the problem indicated by the error is resolved.

---

**2622-441** *program\_name: 2622-441 Samdvs action failed with return code "error\_code".*

**Explanation:** Samdvs action failed to acquire information on specified resource.

**User Response:** Refer to related error messages or return code and then rerun command \n based on those messages.

---

**2622-442** *program\_name: 2622-442 The specified input file "input\_filename" does not exist.*

**Explanation:** Specified DVS input file was not found at the specified location.

**User Response:** Make sure the DVS file exists, then re-enter the command.

---

**2622-443** *program\_name: 2622-443 The specified output file "DVS\_Outputfile" already exist and was not overwritten.*

**Explanation:** Specified DVS Output file already exist at the specified location and was not overwritten.

**User Response:** Make sure the DVS file don't exist or is overwritten when prompted, then re-enter the command.

---

**2622-444** *"program\_name: 2622-444 The specified file "filename" failed to open with return code "return\_code".*

**Explanation:** Specified DVS file failed to open.

**User Response:** Retry the command with the trace option to find the error messages indicating the reason for this failure. Fix it if possible then re -enter the command.

---

**2622-445** *"program\_name: 2622-445 The specified file "filename" failed to close with return code "return\_code".*

**Explanation:** Specified DVS file failed to close.

**User Response:** Retry the command with the trace option to find the error messages indicating the reason for this failure. Fix it if possible then re-enter the command.

---

**2622-446** *"program\_name: 2622-446 Writing to the specified file "filename" failed with return code "return\_code".*

**Explanation:** Specified DVS file failed when attempted to be written .

**User Response:** Retry the command with the trace option to find the error messages indicating the reason for this failure. Fix it if possible then re-enter the command.

---

## Messages

---

**2622-461** *program\_name:* **2622-461 Must be a move action.**

**Explanation:** This must a Move action.

**User Response:** Re-enter the request as a move action.

---

**2622-462** *program\_name:* **2622-462 "node\_name" name is not a valid node name.**

**Explanation:** The specified node name is not valid node.

**User Response:** Repeat the command with a valid node name.

---

**2622-463** *program\_name:* **2622-463 request action on resource group failed with return code "error\_code".**

**Explanation:** Request action on the specified resource group failed.

**User Response:** Refer to related error messages or return code and then rerun command \n based on those messages.

---

**2622-481** *program\_name:* **2622-481 Must be a move action.**

**Explanation:** This must a Move action.

**User Response:** Re-enter the request as a move action.

---

**2622-482** *program\_name:* **2622-482 "node\_name" name is not a valid node name.**

**Explanation:** The specified node name is not valid node.

**User Response:** Repeat the command with a valid node name.

---

**2622-483** *program\_name:* **2622-483 request action on managed resources failed with return code "error\_code".**

**Explanation:** Request action on the specified managed resources failed.

**User Response:** Refer to related error messages or return code and then rerun command based on those messages.

---

**2622-484** *program\_name:* **2622-484 No resources were entered for resource class "resource\_class".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-485** *program\_name:* **2622-485 No Selection string entered for Resource class "resource\_class".**

**Explanation:** Selection string was not entered with the command for the specified class.

**User Response:** Enter the selection string and retry the command.

---

**2622-486** *program\_name:* **2622-486 Move action is not allowed for managed resources.**

**Explanation:** Move Action is not allowed when requested on a managed resource.

**User Response:** Move can only be applied to a resource group.

---

**2622-501** *program\_name:* **2622-501 No resources were entered for resource class "resource\_class".**

**Explanation:** Resources were not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

---

**2622-502** *program\_name: 2622-502* **No selection string was entered for resource class "resource\_class".**

**Explanation:** Selection string was not entered with the command for the specified class.

**User Response:** Enter the resources and retry the command.

---

**2622-503** *program\_name: 2622-503* **List request action failed with return code "error\_code".**

**Explanation:** List request action on the specified managed resources failed.

**User Response:** Refer to related error messages or return code and then rerun command based on those messages.

---

**2661-001** **Attribute "attribute\_name" cannot be specified when defining a new resource.**

**Explanation:** While attempting to create an Application resource, an attribute was encountered that is not allowed to be defined.

**User Response:** Do not specify this attribute when attempting to create an Application resource.

---

**2661-002** **The value of the MonitorCommandPeriod attribute must be greater or equal to the value of the MonitorCommandTimeout attribute.**

**Explanation:** While attempting to create or change an Application resource, a combination of MonitorCommandPeriod and MonitorCommandTimeout attributes was encountered that is not valid.

**User Response:** Correct the value for either the MonitorCommandPeriod or the MonitorCommandTimeout attribute, such that the MonitorCommandPeriod is greater or equal to the MonitorCommandTimeout.

---

**2661-003** **Class name "class\_name" is not recognized by this resource manager.**

**Explanation:** The Resource Manager does not recognize the named resource class as belonging to it. This is either an internal error or indicates a corrupted RMC configuration.

**User Response:** Record the above information and contact your software service organization.

---

**2661-004** **Could not initialize control point for class "class\_name".**

**Explanation:** The Resource Manager was unable to create and initialize RCCP for the named resource class.

**User Response:** Make sure the system has plenty of resources (paging space, available /var file system space, and so on). If this does not resolve the problem, record the above information and contact your software service organization.

---

**2661-005** **The value of the MonitorCommandPeriod attribute must be greater than 0.**

**Explanation:** While attempting to create or change an Application resource, a MonitorCommandPeriod attribute was encountered that is lower than zero.

**User Response:** Correct the value for the MonitorCommandPeriod attribute to be greater than 0.

---

**2661-006** **The start command did not complete successfully, exit code is *the exit value from the start command*.  
*Stdout = stdout generated by the start command* *Stderr = stderr generated by the start command***

**Explanation:** In response to a request to bring a resource online, the associated start command was executed but it did not complete successfully. The exit code, stderr and stdout are listed in the error message.

**User Response:** Attempt to correct the problems identified by the output from the command and retry the operation.

---

## Messages

---

**2661-007**    **The stop command did not complete successfully, exit code is** *the exit value from the stop command*.  
**Stdout =** *stdout generated by the stop command* **Stderr =** *stderr generated by the stop command*

**Explanation:** In response to a request to take a resource offline, the associated stop command was executed but it did not complete successfully. The exit code, stderr and stdout are listed in the error message.

**User Response:** Attempt to correct the problems identified by the output from the command and retry the operation.

---

**2661-008**    **The value of the ResourceType attribute must be 0 or 1.**

**Explanation:** While attempting to create or change an Application resource, a ResourceType attribute was encountered that is not valid.

**User Response:** Correct the value for the ResourceType attribute to be either 0 (for fixed resources) or 1 (for floating resources).

---

**2661-009**    **The value of the RunCommandsSync attribute must be 0 or 1.**

**Explanation:** While attempting to create or change an Application resource, a RunCommandsSync attribute was encountered that is not valid.

**User Response:** Correct the value for the RunCommandsSync attribute to be either 0 (for "fire and forget" of start/stop commands) or 1 (starting/stopping resources will wait for the completion of start/stop commands).

---

**2661-010**    **The user name specified is not valid or does not exist.**

**Explanation:** While attempting to create or change an Application resource, a user name was encountered that is not valid or does not exist.

**User Response:** Correct the user name or ensure that the user name is defined on each target node where the resource is to exist.

---

**2661-011**    **The command specified for attribute** *the attribute name that the command in error is specified for* **is NULL, not a absolute path, does not exist or has insufficient permissions to be run.**

**Explanation:** While attempting to create or change an Application resource, a command string for the specified attribute was encountered that is not valid. The command is either a NULL string, is not an absolute path, does not exist or has insufficient permissions to be run.

**User Response:** Correct the value to be one a valid command on all the target nodes or ensure that the specified command exists on all target node and has the proper permissions.

---

**2661-012**    **The specified netmask** *"the specified netmask in error"* **is not valid.**

**Explanation:** While attempting to create or change a ServiceIP resource, an invalid value for the NetMask attribute was encountered.

**User Response:** Correct the value to be one of the valid representations for specifying a netmask such as xxx.xxx.xxx.xxx or an empty string in which case the netmask will be inherited from the network interface that it is activated on.

---

**2661-013**    **The specified IP address** *"the specified IP address in error"* **is not valid.**

**Explanation:** While attempting to create or change a ServiceIP resource, an invalid value for the IPAddress attribute was encountered.

**User Response:** Correct the value to be one of the valid representations for specifying an IP address such as xxx.xxx.xxx.xxx.

---

---

**2661-014** The network interface specified for use in bringing an IP address online does not exist or is not valid.

**Explanation:** The resource manager was requested to bring an IP address online and the specified supporting resource (a network interface) does not exist or is not usable.

**User Response:** Chose a different supporting resource to use in the online request.

---

**2661-015** The maximum number of aliases are already active on network interface *interface\_name*.

**Explanation:** The resource manager was requested to bring an IP address online but the specified interface to be used already has the maximum number of aliases active.

**User Response:** Either deactivate some aliases or chose another interface to be used for this service IP address.

---

**2661-016** The reset command did not complete successfully, exit code is *the exit value from the reset command*. **Stdout** = *stdout generated by the reset command* **Stderr** = *stderr generated by the reset command*

**Explanation:** In response to a request to reset a resource, the associated reset command was executed but it did not complete successfully. The exit code, stderr and stdout are listed in the error message.

**User Response:** Attempt to correct the problems identified by the output from the command and retry the operation.

---

**2661-017** The start command timed out, exit code is *the exit value from the start command*. **Stdout** = *stdout generated by the start command* **Stderr** = *stderr generated by the start command*

**Explanation:** In response to a request to bring a resource online, the associated start command was run but it exceeded the time limit and was ended. The exit code, stderr and stdout are listed in the error message.

**User Response:** Attempt to correct the problems identified by the output from the command and retry the operation.

---

**2661-018** The stop command timed out, exit code is *the exit value from the stop command*. **Stdout** = *stdout generated by the stop command* **Stderr** = *stderr generated by the stop command*

**Explanation:** In response to a request to take a resource offline, the associated stop command was run but it exceeded the time limit and was ended. The exit code, stderr and stdout are listed in the error message.

**User Response:** Attempt to correct the problems identified by the output from the command and retry the operation.

---

**2661-019** The reset command timed out, exit code is *the exit value from the reset command*. **Stdout** = *stdout generated by the reset command* **Stderr** = *stderr generated by the reset command*

**Explanation:** In response to a request to reset a resource, the associated reset command was run but it exceeded the time limit and was ended. The exit code, stderr and stdout are listed in the error message.

**User Response:** Attempt to correct the problems identified by the output from the command and retry the operation.

---

**2661-020** The value specified for the attribute named ProtectionMode must be either 0 or 1.

**Explanation:** An invalid value was specified for the attribute named ProtectionMode.

**User Response:** Correct the value specified and retry the operation.

---

**2661-021** The command used to monitor the resource did not complete successfully, exit code is *the exit value from the monitor command*. **Stdout** = *stdout generated by the monitor command* **Stderr** = *stderr generated by the monitor command*

**Explanation:** While monitoring the operational state of a resource, the associated monitor command was executed but it did not complete successfully. The exit code, stderr and stdout are listed in the error message.

**User Response:** Attempt to correct the problems identified by the output from the command and retry the operation.

---

## Messages

---

**2661-022**    **The command used to monitor the resource timed out, exit code is** *the exit value from the monitor command*. \n Stdout = *stdout generated by the monitor command* \n Stderr = *stderr generated by the monitor command*

**Explanation:** While monitoring the operational state of a resource, the associated monitor command was run but it exceeded the time limit and was stopped. The exit code, stderr and stdout are listed in the error message.

**User Response:** Attempt to correct the problems identified by the output from the command and retry the operation.

---

**2661-023**    **The IP address specified is the base address for a network interface and therefore cannot be used.**

**Explanation:** The address specified is already the base IP address for a network interface which makes it invalid for use as a service IP address.

**User Response:** Choose a different IP address and retry the operation.

---

**2661-024**    **The resource "*resource\_name*" cannot be deleted because it is online.**

**Explanation:** The resource cannot be deleted because it is online.

**User Response:** Make sure the resource is offline and retry the operation.

---

**2661-025**    **The operation cannot be completed because one or more resources from the *resource class name* class is online.**

**Explanation:** The operation (stop node, stop domain or remove domain) cannot be completed because one or more resources are online on the target node(s).

**User Response:** Make sure the resource is offline and retry the operation.

---

**2661-026**    **Invalid parameter format passed to the configCoordinationAll action for resource class *resource class name*.**

**Explanation:** The configCoordinationAll action was received by the RM for the indicated class but the input format is not valid or not supported.

**User Response:** This is an internal error so contact your software service organization.

---

**2661-027**    **An online or offline operation is not valid when the resource is in the failed offline state.**

**Explanation:** An online or offline operation was targeted to a resource that is in the failed offline state. This is not a valid request. The only control operation that is permitted in the failed offline state is reset.

**User Response:** Perform the reset operation against the resource and then retry the online or offline operation. The command 'resetsrc' can be used to reset a resource.

---

**2661-028**    **Receiver of the SendEIFevent is not online, or not a constituent resource.**

**Explanation:** An internal error of the GblResRM.

**User Response:** Record the above information and contact your software service organization.

---

**2662-001**    **Attribute "*attribute\_name*" cannot be specified when defining a new resource.**

**Explanation:** While attempting to create an Application resource, an attribute was encountered that is not allowed to be defined.

**User Response:** Do not specify this attribute when attempting to create an Application resource.

---

---

**2662-002** Attribute "*attribute\_name*" appears in request more than once.

**Explanation:** An attribute appears more than once in a request.

**User Response:** Specify an attribute only once in a request.

---

**2662-003** Class name "*class\_name*" is not recognized by this resource manager.

**Explanation:** The Resource Manager does not recognize the named resource class as belonging to it. This is either an internal error or indicates a corrupted RMC configuration.

**User Response:** Record the above information and contact your software service organization.

---

**2662-004** Could not initialize control point for class "*class\_name*".

**Explanation:** The Resource Manager was unable to create and initialize RCCP for the named resource class.

**User Response:** Make sure the system has plenty of resources (paging space, available /var file system space, and so on). If this does not resolve the problem, record the above information and contact your software service organization.

---

**2662-005** Attribute "*attribute\_name*" must be specified when defining a new resource.

**Explanation:** The ResourceManager was unable to create the resource because a mandatory attribute was not defined.

**User Response:** Make sure to specify this attribute during creation of the resource.

---

**2662-006** The value of the attribute must be 0 or 1.

**Explanation:** The value of this attribute must be 0 or 1.

**User Response:** Change the attribute to 0 or 1. M: The value of this attribute must be 0 or 1. Change the attribute to 0 or 1.

---

**2662-007** The value of the ResourceType attribute must be 0 or 1.

**Explanation:** While attempting to create or change an Application resource, a ResourceType attribute was encountered that is not valid.

**User Response:** Correct the value for the ResourceType attribute to be either 0 (for fixed resources) or 1 (for floating resources).

---

**2662-008** The time in the Start/Stop/Move attribute must be  $\geq 0$  and  $< 600$  (10 min)

**Explanation:** The specified value for this attribute is not within the allowed range.

**User Response:** Specify a value between 0 and 600.

---

**2662-009** The value of ForceOpState must be a valid rmc opstate (0,1,2,3,4,5,6,8)

**Explanation:** The specified value does not match a valid rmc opstate.

**User Response:** Specify one of the following opstates 0,1,2,3,4,5,6,8.

---

**2662-010** Resource does not support move protocol

**Explanation:** The resource received a move action but does not support the move protocol.

**User Response:** Record the above information and contact your software service organization.

---

## Messages

---

**2662-011 Resource is not an aggregate resource**

**Explanation:** The resource received a move action but is not an aggregate resource.

**User Response:** Record the above information and contact your software service organization.

---

**2662-012 Resource not in move state NONE when prepare action received**

**Explanation:** The resource received a move prepare action but is not in move state none.

**User Response:** Record the above information and contact your software service organization.

---

**2662-013 Resource not in move state READY when complete action received**

**Explanation:** The resource received a move complete action but is not in move state ready.

**User Response:** Record the above information and contact your software service organization.

---

**2662-014 Complete/Cancel action or cleanup already in progress**

**Explanation:** The complete/cancel process is already in progress.

**User Response:** Record the above information and contact your software service organization.

---

**2662-015 Prepare action already in progress**

**Explanation:** The prepare process is already in progress.

**User Response:** Record the above information and contact your software service organization.

---

**2662-016 Resource not in READY or FAILED state when cancel action received**

**Explanation:** The resource received a move cancel action but is not in move state ready or failed.

**User Response:** Record the above information and contact your software service organization.

---

**2662-017 Attribute "*attribute\_name*" cannot be specified when defining a new resource.**

**Explanation:** While attempting to create a Test resource, an attribute was encountered that is not allowed to be defined.

**User Response:** Do not specify this attribute when attempting to create a Test resource.

---

**2662-018 Wrong input specified for the action.**

**Explanation:** An internal error of the TestRM.

**User Response:** Record the above information and contact your software service organization.

---

---

## Appendix B. Troubleshooting

This discusses diagnostics and, problem determination and refers to the Messages appendix.

Files are created in the `"/var/ct/<clustername>/log/mc/IBM.RecoveryRM"` directory to contain internal trace output that is useful to a software service organization for resolving problems. An internal trace utility tracks the activity of the resource manager daemon. Multiple levels of detail may be available for diagnosing problems. Some minimal level of tracing is on at all times. Full tracing can be activated with the command:

```
traceson -s IBM.RecoveryRMd
```

Minimal tracing can be activated with the command:

```
tracesoff -s IBM.RecoveryRMd
```

---

### Resource Manager diagnostic files

All trace files are written by the trace utility to the `"/var/ct/<clustername>/log/mc/IBM.RecoveryRM"` directory. Each file in this directory that is named **trace<.n>** corresponds to a separate run of the resource manager. The latest file that corresponds to the current run of the resource manager is called **trace**. Trace files from earlier runs have a suffix of `.n`, where `n` starts at 0 and increases for older runs.

Use the **rpitr** command to view these files. Records can be viewed as they are added for an active process by adding the **-f** option to the **rpitr** command.

Any core files that result from a program error are written by the trace utility to the `"/var/ct/<clustername>/run/mc/IBM.RecoveryRM"` directory. Like the trace files, older core files have a `.n` suffix that increases with age. Core files and trace files with the same suffix correspond to the same run instance.

The log and run directories have a default limit of 10MB. The resource managers ensure that the total amount of disk space used is less than this limit. Trace files without corresponding core files are removed first when the resource manager is over the limit. Then pairs of core and trace files are removed, starting with the oldest. At least one pair of core and trace files is always retained.

---

### Recovering from RMC and Resource Manager problems

This section describes the tools that you can use to recover from infrastructure problems. It tells you how to determine if the components of the monitoring system are running and what to do if the RMC subsystem or one of the resource managers should abnormally stop. Common troubleshooting problems and solutions are also described.

The Audit Log, Event Response, File System, and Host resource managers recover from most errors because they have few dependencies. In some cases, the recovery consists of terminating and restarting the appropriate daemon. These resource managers can recover from at least the following errors:

1. Losing connection to the RMC daemon, probably caused by the terminating of the RMC daemon or another system problem.
2. Programming errors that cause the process to abnormally terminate. In this case, the SRC subsystem restarts the daemon. This includes errors such as incorrect memory references and memory leaks.
3. The `/var` or `/tmp` directories filling up. When this happens, core and trace files cannot be captured.

In addition, all parameters received from the RMC subsystem are verified to avoid impacting other clients that may be using the same resource manager.

## Troubleshooting

The following tools are described:

1. **ctsnap** command
2. SRC-controlled commands
3. **rmctrl** command for the RMC subsystem
4. Audit log

## Using the ctsnap command

For debugging purposes, the **ctsnap** command can be used to **tar** the RSCT and resource-manager programs and send them to the software service organization. The **ctsnap** command gathers system configuration information and compresses the information into a **tar** file, which can then be downloaded to disk or tape and transmitted to a remote system. The information gathered with the **ctsnap** command may be required to identify and resolve system problems. See the man page for the **ctsnap** command for more information.

## SRC-controlled commands

The RMC subsystem and the resource managers are controlled by the System Resource Controller (SRC). They can be viewed and manipulated by SRC commands. For example:

To see the status of all resource managers, type:

```
lssrc -g rsct_rm
```

To see the status of an individual resource manager, type:

```
lssrc -s rmname
```

where *rmname* can be:

- IBM.AuditRM
- IBM.DMSRM
- IBM.ERRM
- IBM.FSRM
- IBM.HostRM
- IBM.Sensor

To see the status of all SRC-controlled subsystems on the local machine, type:

```
lssrc -a
```

The output of `lssrc -a` may look as follows. It is not a problem when some of the subsystems are not active.

| Subsystem      | Group   | PID | Status      |
|----------------|---------|-----|-------------|
| IBM.ConfigRM   | rsct_rm | 387 | active      |
| ctcas          | rsct    | 419 | active      |
| cthats         | cthats  | 523 | active      |
| cthags         | cthags  | 524 | active      |
| ctrmc          | rsct    | 600 | active      |
| IBM.ERRM       | rsct_rm | 615 | active      |
| IBM.RecoveryRM | rsct_rm | 637 | active      |
| IBM.AuditRM    | rsct_rm | 662 | active      |
| IBM.GblResRM   | rsct_rm | 695 | active      |
| IBM.TestRM     | rsct_rm | 696 | active      |
| IBM.SensorRM   | rsct_rm |     | inoperative |
| IBM.HostRM     | rsct_rm |     | inoperative |

Not all subsystems are active as default. Some of the subsystems start automatically, and some of them when they are first referenced.

To see the status of a particular subsystem, for example, the RMC subsystem, which is known to SRC as **ctrmc**, type:

```
lssrc -s ctrmc
```

The SRC has these commands:

- **lssrc**
- **startsrc**
- **stopsrc**
- **traceson**
- **tracesoff**

For more information, see the command man pages.

For more information about SRC, see *System Management Concepts: Operating System and Devices* or *IBM RSCT Administration Guide*.



---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Department LJEB/P905  
2455 South Road Road  
Poughkeepsie, New York 12601-5400  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- IBM, AIX, AIX 5L, Netfinity, RS/6000, SP, and the e (logo) are trademarks or registered trademarks of International Business Machines Corporation.
- Equinox is a trademark of Equinox Systems, Inc.
- Linux is a registered trademark of Linus Torvalds.
- Myrinet is a trademark of Myricom, Inc.
- Red Hat and RPM are trademarks of Red Hat, Incorporated.
- Java and all Java-based trademarks and logos are registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Other company, product, or service names may be the trademarks or service marks of others.

---

# Index

## Special characters

(EIF) Tivoli Event Integration Facility 119

### A

about this book ix  
adding a member resource to  
    resource group 45  
addrgubr command 45, 147  
addrpnode command 21  
affinity relationship 75  
aggregate resource 130  
AllowedNode resource group attribute 38  
AntiAffinity relationship 76  
AntiCollocated relationship 73  
attributes  
    dynamic 2, 31  
    NodeNameList 32  
    persistent 2, 31  
    resource attributes, description of 2, 31  
    SelectFromPolicy 33  
    used by equivalencies 50  
    used by IBM.Application 130  
    used by IBM.ServiceIP 139  
    used by IBM.Test 142  
    used by managed relationships 54  
    used by resource groups 38  
    used by resources 32  
audience of this book ix  
automation  
    events affecting automation 83  
    parameters 112  
automation policy management 114  
AutomationDetails attribute 42

### B

baroc file 119  
base data types, supported 222  
binder  
    definition 81  
blanks, use of in expressions 224

### C

changing  
    a resource group 46  
    an equivalency 51  
    attributes of resource group members 47  
chequ command 51, 151  
chrel command 154  
chrg command 46, 159  
chrgubr command 47, 162  
Cluster Systems Management (CSM) 7  
clusters  
    adding nodes to 21  
    defining and administering 19

clusters (*continued*)  
    removing 23  
    restoring a configuration 115  
    taking offline 22  
collocated relationship 71  
commands  
    addrgubr 45, 147  
    addrpnode 21  
    chequ 51, 151  
    chrel 154  
    chrg 46, 159  
    chrgubr 47, 162  
    ctsnap 274  
    IBM Tivoli System Automation 146  
    lsequ 51, 165  
    lsrel 79, 168  
    lsrg 45, 173  
    lsrgreq 178  
    lsrpnod 23  
    lssamctrl 181  
    mkequ 51, 183  
    mkrel 79, 187  
    mkrgr 45, 191  
    mkrpdomain 19  
    pidmon 220  
    preprnode 19  
    rgubrreq 194  
    rgreq 197  
    rmequ 51, 200  
    rmrel 80, 202  
    rmrg 47, 206  
    rmrgubr 47, 208  
    rmrpdomain 23  
    rmrpnod 23  
    RSCT 146  
    samcfg 211  
    samctrl 128, 213  
    samdiag 117, 216  
    samlicm 219  
    startpdomain 19, 23  
    startpnode 21  
    stoppdomain 22  
    stoppnode 22, 23  
Condition attribute 55  
conditions for location relationships 69  
configuration quorum 4, 96  
ConfigValidity, dynamic attribute 126  
constituent resources 130  
core files 273  
creating  
    an equivalency 51  
    relationships 79  
    resource groups 45  
critical resource 95  
ctsnap command 274

## D

- data types
  - base 222
  - structured 223
  - used for literal values 223
- DependsOn
  - behavior schemes 62
  - relationship attribute 62
  - start behavior 62
  - stop behavior 63
- DependsOnAny relationship attribute 67
- diagnosing resources 117
- dynamic attribute 31
  - AutomationDetails 38
  - ConfigValidity 126
  - definition 2
  - MoveStatus 43, 116
  - OpQuorumState 99
  - OpState 34, 38, 130, 139
  - TopGroup 38

## E

- e-mail address xi
- equivalencies
  - attributes used by 50
  - changing 51
  - description of 3, 49
  - listing 51
  - making (creating) 51
  - removing 51
  - rules for using 49
- establishing an equivalency 51
- establishing the node list for
  - resource group members 45
- event response resource manager (ERRRM) 128
- events affecting automation 83
- ExcludedList attribute 41
- ExcludedNodes parameter 113
- expressions
  - operators for 224
  - pattern matching supported in 228
  - using 222

## F

- fail tiebreaker 99
- fixed resources
  - definition 33
- floating resources
  - definition 33
- force down behavior of DependsOn 62
- ForcedDownBy relationship 68

## G

- General Parallel File System (GPFS) 7
- getting quorum information 98

## H

- highlighting x
- how to use this book ix

## I

- IBM Tivoli System Automation
  - commands 146
  - getting started 17
  - installing 7, 11
  - migrating 13
  - setting up a high available network 105
  - uninstalling 16
- IBM.Application 130
- IBM.RecoveryRM 24
- IBM.ServiceIP
  - attributes 139
  - definition 137
  - detecting network interface failures 105
- IBM.TestRM 142
- IFNotOffline condition 70
- IFNotOnline condition 69
- IOffline condition 69
- IOnline condition 69
- installing IBM Tivoli System Automation 7, 11, 17
- installing service 16
- ISO 9000 xi
- IsStartable relationship 77

## L

- listing
  - an equivalency 51
  - relationships 79
  - resource groups 45
- location relationships 69
- lsequ command 51, 165
- lsrel command 79, 168
- lsrg command 45, 173
- lsrgreq command 178
- lsrnode command 23
- lssamctrl command 181

## M

- making a resource group 45
- managed relationships
  - attributes used by 54
  - description of 53
- managed resource
  - description of 3, 32
- Mandatory attribute of ManagedResource class 44
- MemberClass equivalency attribute 50
- MemberLocation resource group attribute 40
- MemberOf attribute 44
- Membership equivalency attribute 50
- migrating IBM Tivoli System Automation 13
  - completing migrating 15
  - considerations 13
  - migrating a node step by step 14

migrating IBM Tivoli System Automation *(continued)*  
  migrating an entire domain 13  
mkequ command 51, 183  
mkrel command 79, 187  
mkrq command 45, 191  
mkrpdomain command 19  
modifying a policy 115  
modifying predefined expressions 222  
MoveStatus, dynamic attribute 43, 116

## N

Name attribute 54  
Name resource group attribute 40  
node list for resource 32  
node list, establishing 45  
NodeNameList attribute for resource 32  
nodes  
  rebooting a node 128  
  removing from a cluster 23  
  stopping a node 128  
  taking offline 22  
NominalState resource group attribute 40

## O

operational quorum 4, 96  
operator precedence 226  
operator tie breaker 99  
operators available for use in expressions 224  
OpState attribute for resource 34  
OpState resource group attribute 41

## P

parameters  
  automation 112  
  ExcludedNodes 113  
  ResourceRestartTimeout 113  
  TimeOut and RetryCount 111  
pattern matching supported in expressions 228  
pdksh package 9  
persistent attribute 31  
  AllowedNode 38  
  changing values 46  
  Condition 55  
  definition 2  
  ExcludedList 41  
  ForceOpState 143  
  IPAddress 140  
  Mandatory 44  
  MemberClass 50  
  MemberLocation 40  
  MemberOf 44  
  Membership 50  
  MonitorCommand 132  
  MonitorCommandPeriod 132  
  MonitorCommandTimeout 132  
  Name 40, 54, 131, 140, 142  
  NetMask 140  
  NodeNameList 32, 131, 140, 142

persistent attribute *(continued)*  
  NominalState 40  
  OpQuorumOverride 104  
  Priority 40  
  ProtectionMode 97, 133, 140  
  Relationship 54  
  ResourceType 33, 131, 140, 143  
  RunCommandsSync 133  
  SelectFromPolicy 50  
  SelectString 50  
  Source 54  
  StartCommand 131  
  StartCommandTimeout 132  
  StopCommand 132  
  StopCommandTimeout 133  
  Target 54  
  TimeToStart 143  
  TimeToStop 143  
  UserName 133  
  WriteToSyslog 143  
pidmon command 220  
policy  
  modifying a policy 114  
  replacing a policy 114  
  saving and restoring a policy 114  
precedence of operators 226  
predefined expressions  
  modifying 222  
preprnode command 19  
prerequisite knowledge for this book ix  
Priority resource group attribute 40  
ProtectionMode attribute 97  
Public Domain Korn Shell (pdksh) package 9  
publisher configuration file 120  
publishing IBM Tivoli System Automation internal  
  attributes 122

## Q

quorum  
  configuration quorum 4, 96  
  getting information 98  
  majority of nodes 95  
  operational quorum 4, 96  
  tie breaker 97

## R

recovery resource manager (IBM.RecoveryRM) 24  
recovery support 273  
recovery support for resource managers 273  
relationship  
  DependsOn 62  
  DependsOnAny 67  
Relationship attribute 54  
relationships  
  Affinity 75  
  AntiAffinity 76  
  AntiCollocated 73  
  Collocated 71  
  creating 79

- relationships (*continued*)
  - ForcedDownBy 68
  - IsStartable 77
  - listing 79
  - location relationship 69
  - removing 80
  - StartAfter 56
  - StopAfter 60
- removing
  - a member resource from a resource group 47
  - an equivalency 51
  - relationships 80
  - resource groups 47
- removing a member resource
  - resource group 47
- resource
  - attributes used by 32
  - critical 95, 97
  - description of 2, 31
  - description of resource attributes 2, 31
  - description of resource class 2, 31
  - nominal state 3, 34
  - verifying 126
- resource class
  - definition 2
- resource classes
  - definition 31
  - IBM.Application 129
  - IBM.ServiceIP 137
  - IBM.Test 142
  - IBM.TieBreaker 99
- resource group
  - adding a member resource to 45
  - allowing it to become Online 83
  - attributes used by 38
  - changing attributes of 46
  - description of 35
  - establishing the node list for 45
  - listing (incl. its resource members) 45
  - making (creating) 45
  - moving 116
  - removing 47
  - starting 46
  - states 40
  - stopping 46
- resource group members
  - changing the attributes of 47
- resource manager
  - Global Resource RM 129
  - resource class IBM.Application 129
  - resource class IBM.ServiceIP 137
  - resource class IBM.Test 142
  - test resource manager 142
- resource manager diagnostic files 273
- resource type 33
- ResourceRestartTimeout parameter 113
- ResourceType attribute for resource 33
- restoring a cluster configuration 115
- restoring a policy 115
- rgmbrreq command 194
- rgreq command 197

- rmequ command 51, 200
- rmrel command 80, 202
- rmrq command 47, 206
- rrmgbr command 47
- rrmgbrg command 208
- rmrpdomain command 23
- rmrpnode command 23
- RSCT command 146

## S

- samcfg command 211
- samctrl command 128, 213
- samdiag command 117, 216
- samlcm command 219
- saving a policy 114
- select string 222
- SelectFromPolicy attribute 33
- SelectFromPolicy equivalency attribute 50
- SelectString equivalency attribute 50
- setting up a high available network 105
- Source attribute 54
- SQL syntax 222
- start behavior of DependsOn 62
- StartAfter relationship 56
- starting a resource group 46
- startprdomain command 19, 23
- startprnode command 21
- state transitions for a resource group 41
- stop behavior of DependsOn 63
- StopAfter relationship 60
- stopping a resource group 46
- stopprdomain command 22
- stopprnode command 22, 23
- structured data types 223

## T

- Target attribute 54
- TEC (Tivoli Enterprise Console) 119
- TEC (Tivoli Enterprise Console) event interface 119
- TEC EIF configuration file 120
- TEC publisher function 119
- Telnet dialog buffer 10
- test resource manager 142
- tie breaker 4, 99
- Timeout and RetryCount parameter 111
- Tivoli Enterprise Console (TEC) 119
- Tivoli Enterprise Console (TEC) event interface 119
- Tivoli Event Integration Facility (EIF) 119
- TopGroup resource group attribute 42
- trace files 273
- trademarks 278

## U

- uninstalling IBM Tivoli System Automation 16
- upgrading IBM Tivoli System Automation 11
- using select strings in expressions 222

## V

- variable names 224
- verifying resources 126
- verifying the version number 14
- version number of IBM Tivoli System Automation 14



---

# Readers' Comments — We'd Like to Hear from You

IBM Tivoli System Automation for Multiplatforms  
Guide and Reference  
Version 1.2

Publication No. SC33-8210-03

Overall, how satisfied are you with the information in this book?

|                      | Very Satisfied           | Satisfied                | Neutral                  | Dissatisfied             | Very Dissatisfied        |
|----------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Overall satisfaction | <input type="checkbox"/> |

How satisfied are you that the information in this book is:

|                          | Very Satisfied           | Satisfied                | Neutral                  | Dissatisfied             | Very Dissatisfied        |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Accurate                 | <input type="checkbox"/> |
| Complete                 | <input type="checkbox"/> |
| Easy to find             | <input type="checkbox"/> |
| Easy to understand       | <input type="checkbox"/> |
| Well organized           | <input type="checkbox"/> |
| Applicable to your tasks | <input type="checkbox"/> |

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH  
Department 3248  
Schoenaicher Strasse 220  
D-71032 Boeblingen  
Federal Republic of Germany



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5639-N53  
5655-I53

Printed in USA

SC33-8210-03



Spine information:



IBM Tivoli System Automation  
for Multiplatforms

IBM Tivoli System Automation Guide and  
Reference

Version 1.2

SC33-8210-03