Automatically configuring the target environment using the eG CLI

Restricted Rights Legend

Trademarks

Copyright

Table of Contents

Chapter 1

Introduction

eG Enterprise is a 100%, web-based management console that allows users to view performance metrics collected from a target infrastructure. Users with administrative rights can configure the infrastructure that needs to be monitored. Configuration typically involves a sequence of tasks that prepares the environment for monitoring - this includes identifying and adding the components to be monitored, configuring the tests pertaining to these components, setting thresholds, configuring additional external and remote agents for the environment, etc. Typically, a user must login to the web-based eG administrative interface as an admin user in order to perform the above-mentioned tasks.

In many large data centers, an orchestaration solution such as HP Operations Orchestration (HPOO) software, is used to automate routine IT tasks. In such environments, the eG manager must integrate seamlessly with the automation tool, so that the eG Enterprise system can be automatically administered via the tool.

To enable this integration, the eG Enterprise suite provides a command-line interface (CLI) which allows the automation tool/script to communicate with the eG manager and execute simple commands on the manager to perform critical configuration tasks. This integration minimizes user intervention in the configuration of the monitoring system. In addition, since commands can also be executed in bulk using the CLI, it significantly reduces the time required to perform simultaneous tasks. Administrators can also construct scripts (using any common scripting language) that use the CLI to automate routine administration tasks.

Note:

Bulk updates using the CLI are provided here to solicit customer inputs. Based on feedback, we expect this interface to change in future versions of the CLI.

Currently, the command-line interface provided by eG Enterprise - known as the **eG CLI -** can be used for performing the following administrative tasks only:

- Adding/modifying/removing components for monitoring
- Managing/Unmanaging components
- Adding/removing external agents
- Adding/removing remote agents

- Assigning/Disassociating agents from a manager in a redundant manager setup
- Adding/Modifying/Deleting Quick Maintenance Policies
- Assigning Quick Maintenance Policies to Components/Hosts/Tests
- Adding/Modifying/Deleting Independent Policies
- Adding/Modifying/Deleting Maintenance Policy Groups
- Assigning Independent Policy to Maintenance Policy Groups
- Associating Maintenance Policy Groups with Infrastructure Elements
- Enabling/Disabling Tests
- Displaying the Remote agents
- Displaying the External agents
- Displaying the Components
- Displaying the test names for a Component Type
- Include/Exclude Components for Test
- Include/Exclude Tests for Component
- Displaying the Maintenance Policies
- Displaying the details of the Maintenance Policies
- Displaying the details of the Tests
- Adding/Modifying/Deleting a User
- Associating Components to a User
- Adding/Modifying/Renaming/Deleting a Zone
- Adding/Modifying/Renaming/Deleting a Group

Note:

Other administrative tasks will be supported in future versions of the CLI.

1.1 How does the eG CLI Work?

The eG CLI is currently supported on Windows, Linux, and Solaris environments.

By default, the eG CLI is bundled with the eG manager. Accordingly, once the eG manager is installed, the CLI-related files automatically get installed into the **<EG_INSTALL_DIR>\egcli** directory (on Windows; on Unix, this will be the **/opt/egurkha/egcli** directory). You can then proceed to issue the CLI commands from the Windows command prompt / Unix shell prompt on the eG manager host, and easily administer the eG Enterprise system.

On the other hand, to install eG CLI on a remote Windows/Linux/Solaris host, you will have to run a special set-up program that is provided. This is called **eGCLI.exe** on Windows environments, **iCLI_linux** on Linux and **iCLI_solaris** on Solaris. If an automation tool is in use in your environment, the setup program has to be executed on the host on which the automation tool (say, HPOO) operates. On the

other hand, if no orchestration engine is in use in your environment, then, you can run the setup program on any remote host in your environment, so that eG CLI gets installed on that host.

Once installed, you need to configure the CLI to access the eG manager in the target environment, so that the CLI can execute commands/scripts on it to setup the environment for monitoring.

Where an automation tool pre-exists, soon after CLI installation, you will have to explicitly configure the tool with the commands supported by the CLI. Chapter 3 provides a detailed list of commands and their syntax. If a user then logs into the tool and invokes any of the configured commands, the CLI establishes an HTTP/HTTPS connection with the eG manager indicated by the command, and automatically executes the issued command on that manager.

If the eG CLI has been installed on a remote client instead, you can either directly issue the eG CLI commands from the command prompt of the client or can bundle the commands into a script that can be executed from the prompt. The command/script will then establish an HTTP/HTTPS connection with the indicated eG manager and perform the configuration / database-related tasks on that manager.

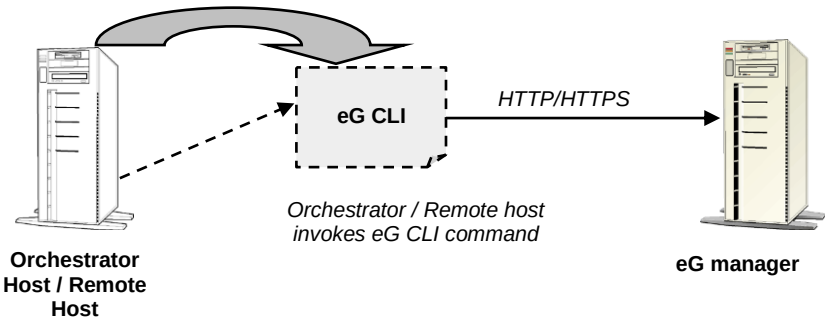

Figure 1.1: How the eG CLI works

In a redundant eG manager setup on the other hand, the eG CLI should be configured to communicate with the primary manager alone, as administrative operations can be performed only on this manager. Once this is done, the redundant manager architecture will ensure that the configuration changes effected on the primary manager are automatically replicated to the secondary manager(s).

The next chapter will discuss how to install the eG CLI and how to configure it to communicate with individual managers

Chapter 2

The eG CLI Commands for Automating Target Environment Configuration

The eG CLI supports commands for performing the following administrative activities on the eG manager:

- Adding/Modifying/Removing a Component
- Managing/Unmanaging Components
- Adding/Removing External Agents
- Adding/Removing Remote Agents
- Assigning/Disassociating agents from managers in a redundant setup
- Adding/Modifying/Deleting Quick Maintenance Policies
- Assigning Quick Maintenance Policies to Components/Hosts/Tests
- Adding/Modifying/Deleting Independent Policies
- Adding/Modifying/Deleting Maintenance Policy Groups
- Assigning Independent Policy to Maintenance Policy Groups
- Associating Maintenance Policy Groups with Infrastructure Elements
- Enabling/Disabling Tests
- Displaying the Remote agents
- Displaying the External agents
- Displaying the Components
- Displaying the test names for a Component Type
- Include/Exclude Components for Test
- Include/Exclude Tests for Component
- Displaying the Maintenance Policies
- Displaying the details of the Maintenance Policies
- Displaying the details of the Tests
- Adding/Modifying/Deleting a User

- Associating Components to a User
- Adding/Modifying/Renaming/Deleting a Zone
- Adding/Modifying/Renaming/Deleting a Group

Note:

Other administrative tasks will be supported in future versions of the CLI.

Note:

- Once you install the Admin CLI, you will need to reboot the system on which the Admin CLI is installed. This is required for the installation directory of the Admin CLI to be in the system's path. Otherwise, you will need to prepend **<EGADMINCLI_DIR>\bin** to the commands you want to execute.
- Similarly, if you have built scripts into which the eG CLI commands have been bundled, then make sure that each command invocation in the script is pre-fixed by the full path to the command (i.e., **<EGADMINCLI_DIR>\bin**). To avoid this, you will have to reboot the Admin CLI host after installation, and then proceed to write the scripts.
- The eG CLI commands can support multi-byte characters - i.e., the component names and agent nick names used in the command line can include multi-byte characters. However, to enable the eG CLI to support multi-byte component names/nick names, you need to install and configure the eG CLI on the corresponding multi-byte operating system.

2.1 Adding Components

Task	Command	Example
Adding an agent-based and port-based component	**eGCLI AddComponent -componenttype <ComponentType> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <nick name of the component> -port <port at which the component listens> -externalagents <Comma-separated list of external agents assigned to the server> -managerid <ManagerID> [-groupname <name of the group to which the component is to be added>] [-zonename <name of the zone to which the component is to be added]**	eGCLI AddComponent -componenttype Active Directory -**hostip** 192.168.10.228 -componentname ad228 -port 389 -externalagents 192.168.10.74,extagnt175 -managerid mgr157
		eGCLI AddComponent -componenttype Active Directory -**hostip** 192.168.10.228 -componentname ad228 -port 389 -externalagents 192.168.10.74,extagnt175 -managerid mgr157 -groupname GRP1
		eGCLI AddComponent -componenttype Active Directory -**hostip** 192.168.10.228 -componentname ad228 -port 389 -externalagents 192.168.10.74,extagnt175 -managerid mgr157 -zonename zone1
		eGCLI AddComponent -componenttype Active Directory -**hostname** egurkha25 -componentname ad228 -port 389 -externalagents 192.168.10.74,extagnt175 -managerid mgr157
Adding an agent-based and non-port-based component	**eGCLI AddComponent -componenttype <ComponentType> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <nick name of the component> -externalagents <Comma-separated list of external agents assigned to the server> -managerid <ManagerID> [-groupname <name of the group to which the component is to be added>] [-zonename <name of the zone to which the component is to be added]**	eGCLI AddComponent -componenttype Generic -**hostip** 192.168.10.228 -componentname gen228 -externalagents 192.168.10.74,extagnt175 -managerid 100manager
		eGCLI AddComponent -componenttype Generic -**hostname** egurkha25 -componentname gen228 -externalagents 192.168.10.74,extagnt175 -managerid 100manager

Adding an Oracle database server in an agent-based manner	**eGCLI AddComponent -componenttype <ComponentType> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <nick name of the component> -port <port at which the component listens> -sid <Comma-separated list of SIDs> -ispassive <yes/no> -externalagents <Comma-separated list of external agents assigned to the server> -managerid <ManagerID> [-groupname <name of the group to which the component is to be added>] [-zonename <name of the zone to which the component is to be added]**	eGCLI AddComponent -componenttype Oracle Database -**hostip** 192.168.10.8 -componentname ora8 -port 1521 -sid egora,bussdb -ispassive no -externalagents agt25,agt27 -managerid 100manager
		eGCLI AddComponent -componenttype Oracle Database -**hostname** egurkha25 -componentname ora8 -port 1521 -sid egora,bussdb -ispassive no -externalagents agt25,agt27 -managerid 100manager
Adding an IIS web server in an agent-based manner	**eGCLI AddComponent -componenttype <ComponentType> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <nick name of the component> -port <port at which the component listens> -mtsenabled <yes/no> -externalagents <Comma-separated list of external agents assigned to the server> -managerid <ManagerID> [-groupname <name of the group to which the component is to be added>] [-zonename <name of the zone to which the component is to be added]**	eGCLI AddComponent -componenttype IIS Web -**hostip** 192.168.10.72 -componentname iis72 -port 80 -mtsenabled no -externalagents agt25,agt27 -managerid 100mgr
		eGCLI AddComponent -componenttype IIS Web -**hostname** egurkha25 -componentname iis72 -port 80 -mtsenabled no -externalagents agt25,agt27 -managerid 100mgr
Adding an application executing on a VM	**eGCLI AddComponent -componenttype <ComponentType> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <nick name of the component> -port <port at which the component listens> -virtualenv <yes/no> -virtualserver <componentType:componentName:port> -externalagents <Comma-separated list of external agents assigned to the server> -managerid <ManagerID> [-groupname <name of the group to which the component is to be added>] [-zonename <name of the zone to which the component is to be added]**	eGCLI AddComponent -componenttype Microsoft SQL -**hostip** 192.168.10.163 -componentname sql163 -port 1433 -virtualenv yes -virtualserver VMware ESX:esx179:NULL -externalagents agt25,agt27 -managerid 201mgr
		eGCLI AddComponent -componenttype Microsoft SQL -**hostname** egurkha25 -componentname sql163 -port 1433 -virtualenv yes -virtualserver VMware ESX:esx179:NULL -externalagents agt25,agt27 -managerid 201mgr

Task	Command	Example
Adding a Windows application in an agentless manner	**eGCLI AddComponent -componenttype <ComponentType> -hostname <the name of the host on which the target component is executing> (or) -hostip<IP address of the component> -componentname <nick name of the component> -port <port at which the component listens> -agentless <yes/no> -os <Windows 2000/Windows 2003/Windows 2008/Windows NT/Windows XP> -mode Perfmon -remoteagent <The remote agent that monitors the target> -ispassive <yes/no> -externalagents <Comma-separated list of external agents assigned to the server> -managerid <ManagerID> [-groupname <name of the group to which the component is to be added>] [-zonename <name of the zone to which the component is to be added]**	eGCLI AddComponent -componenttype Oracle Database -**hostname** egurkha25 -componentname ora8 -port 1521 -sid egora -agentless yes -os Windows 2003 -mode Perfmon -remoteagent 192.168.10.7 -externalagents agt25,agt27 -ispassive no -managerid 102mgr
		eGCLI AddComponent -componenttype Oracle Database -**hostip** 192.168.10.8 -componentname ora8 -port 1521 -sid egora -agentless yes -os Windows 2003 -mode Perfmon -remoteagent 192.168.10.7 -externalagents agt25,agt27 -ispassive no -managerid 102mgr
Adding a Unix application in an agentless manner	**eGCLI AddComponent -componenttype <ComponentType> -hostname <the name of the host on which the target component is executing> (or) -hostip<IP address of the component> -componentname <nick name of the component> -port <port at which the component listens> -agentless <yes/no> -os <Solaris/Linux/AIX/HP-UX//Tru64/Other> -mode <Rexec/SSH> -remoteport <The port at which Rexec/SSH listens> -remoteuser <Valid user name on the target> -remotepwd <A valid password> -remoteagent <The remote agent that monitors the target> -ispassive <yes/no> -externalagents <Comma-separated list of external agents assigned to the server> -managerid <ManagerID> [-groupname <name of the group to which the component is to be added>] [-zonename <name of the zone to which the component is to be added]**	eGCLI AddComponent -componenttype Oracle Database -**hostname** egurkha25 -componentname ora8 -port 1521 -sid egsql -agentless yes -os Linux -mode SSH -remoteport 22 -remoteuser egtest -remotepwd egurkha2008 -remoteagent 192.168.10.7 -ispassive no -externalagents agt25,agt27 -managerid 100mgr
		eGCLI AddComponent -componenttype Oracle Database -**hostip** 192.168.10.8 -componentname ora8 -port 1521 -sid egsql -agentless yes -os Linux -mode SSH -remoteport 22 -remoteuser egtest -remotepwd egurkha2008 -remoteagent 192.168.10.7 -ispassive no -externalagents agt25,agt27 -managerid 100mgr
Assigning an existing internal agent to a new component	**eGCLI AddComponent -componenttype <ComponentType> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <nick name of the component> -port <port at which the component listens>**	eGCLI AddComponent -componenttype Oracle Database -**hostname** egurkha25 -componentname ora8 -port 1521 -sid egsql -internalagentassignment yes -internalagent web08 -ispassive no -externalagents agt25,agt2 -managerid 100mgr

	-internalagentassignment <yes/no> -internalagent <IP address/nick name of the internal agent> -ispassive <yes/no> -externalagents <Comma-separated list of external agents assigned to the server> -managerid <ManagerID> [-groupname <name of the group to which the component is to be added>] [-zonename <name of the zone to which the component is to be added]	eGCLI AddComponent -componenttype Oracle Database -**hostip** 192.168.10.8 -componentname ora8 -port 1521 -sid egsql -internalagentassignment yes -internalagent web08 -ispassive no -externalagents agt25,agt2 -managerid 100mgr

Note:

By default, a component added using the eG CLI can either be associated with a group or zone alone. You will not be permitted to associate a component added using the eG CLI to a group and zone simultaneously.

2.2 Managing Components

Task	Command	Example
Managing a non-port-based component	**eGCLI ManageComponent -componenttype <Component Type> -componentname <Nick name of the component> -managerid <ManagerID>**	eGCLI ManageComponent -componenttype Generic -componentname gen20 -managerid 100mgr
Managing a port-based component	**eGCLI ManageComponent -componenttype <Component Type> -componentname <Nick name of the component> -port <port number> -managerid <ManagerID>**	eGCLI ManageComponent -componenttype Microsoft SQL -componentname sql173 -port 1433 -managerid 100mgr
Managing an Oracle database server	**eGCLI ManageComponent -componenttype <Component Type> -componentname <Nick name of the component> -port <port number> -sid <SID of the Oracle server> -managerid <ManagerID>**	eGCLI ManageComponent -componenttype Oracle Database -componentname ora8 -port 1521 -sid egora -managerid 100mgr **Note: An Oracle database server that supports multiple SIDs cannot be managed using a single command; each SID should be passed to a separate 'ManageComponent' command.**

2.3 Modifying Components

Task	Command	Example
Changing the nick name	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -oldcomponentname <The current nick name of the component> -newcomponentname <The new nick name of the component> -port <Port> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Microsoft SQL -**hostip** 192.168.10.68 -oldcomponentname mssql68 -newcomponentname sql68 -port 1433 -ispassive no -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Microsoft SQL -**hostname** egurkha25 -oldcomponentname mssql68 -newcomponentname sql68 -port 1433 -ispassive no -externalagents agt25 -managerid 100mgr
Changing the IP address	**eGCLI ModifyComponent -componenttype <Component Type> -componentname <Nick name of the component> -oldhostip <The current IP address of the component> -newhostip <The new IP**	eGCLI ModifyComponent -componenttype Active Directory -componentname ad -oldhostip 192.168.10.199 -newhostip 192.168.10.191-port 389 -externalagents agt25

	address of the component> -port <Port> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>	-managerid 100mgr
Changing the Hostname	**eGCLI ModifyComponent -componenttype <Component Type> -componentname <Nick name of the component> -oldhostname <The current host name of the component> -newhostname <The new host name of the component> -port <Port> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Active Directory -componentname ad -oldhostname egurkha25 -newhostname eg25 -port 389 -externalagents agt25 -managerid 100mgr
Changing the port number	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -oldport <The old port number> -newport <The new port number> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -oldport 80 -newport 7077 -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -oldport 80 -newport 7077 -externalagents agt25 -managerid 100mgr
Changing monitoring mode of a Windows application from agent-based to agentless	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -agentless yes -os <Windows 2000/Windows 2003/Windows NT/Windows 2008/Windows XP> -mode Perfmon -remoteagent <The remote agent for monitoring this target> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -agentless yes -os Windows 2003 -mode Perfmon -remoteagent 192.168.10.29 -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -agentless yes -os Windows 2003 -mode Perfmon -remoteagent 192.168.10.29 -externalagents agt25 -managerid 100mgr

Task	Command	Example
Changing monitoring mode of a Unix application from agent-based to agentless	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -agentless yes -os <Solaris/Linux/AIX/HP-UX/FreeBSD/Tru64> -mode <SSH/Rexec> -remoteport <The remote port used by SSH/Rexec> -remoteuser <A valid user name on the target> -remotepwd <The password that corresponds to the user name> -remoteagent <The remote agent for monitoring this target> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -agentless yes -os Linux -mode Rexec -remoteport 512 -remoteuser egtest -remotepwd egurkha2008 -remoteagent 192.168.10.29 -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -agentless yes -os Linux -mode Rexec -remoteport 512 -remoteuser egtest -remotepwd egurkha2008 -remoteagent 192.168.10.29 -externalagents agt25 -managerid 100mgr
Changing monitoring mode agentless to agent-based	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -agentless no -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -agentless no -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -agentless no -externalagents agt25 -managerid 100mgr
Changing OS of Agentless system from Windows to Unix	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -agentless yes -os <Solaris/Linux/AIX/HP-UX/FreeBSD/Tru64> -mode <SSH/Rexec> -remoteport <The remote port used by SSH/Rexec> -remoteuser <A valid user name on the target> -remotepwd <The password that corresponds to the user name> -remoteagent <The remote agent for monitoring this target> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -agentless yes -os Linux -mode Rexec -remoteport 512 -remoteuser egtest -remotepwd egurkha2008 -remoteagent 192.168.10.29 -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -agentless yes -os Linux -mode Rexec -remoteport 512 -remoteuser egtest -remotepwd egurkha2008 -remoteagent 192.168.10.29 -externalagents agt25 -managerid 100mgr

Changing OS of Agentless system from Unix to Windows	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -agentless yes -os <Windows 2000/Windows 2003/Windows NT/Windows 2008/Windows XP> -mode Perfmon -remoteagent <The remote agent for monitoring this target> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -agentless yes -os Windows 2003 -mode Perfmon -remoteagent 192.168.10.29 -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -agentless yes -os Windows 2003 -mode Perfmon -remoteagent 192.168.10.29 -externalagents agt25 -managerid 100mgr

Task	Command	Example
Switching off the 'Internal Agent Assignment' flag	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -internalagentassignment no -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -internalagentassignment no -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -internalagentassignment no -externalagents agt25 -managerid 100mgr
Switching on the 'Internal Agent Assignment' flag	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -internalagentassignment yes -internalagent <The internal agent IP/nick name> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -internalagentassignment yes -internalagent 192.168.10.54 -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -internalagentassignment yes -internalagent 192.168.10.54 -externalagents agt25 -managerid 100mgr
Converting a Virtual Application into a Physical Application	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -virtualEnv no -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -virtualEnv no -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -virtualEnv no -externalagents agt25 -managerid 100mgr
Converting a Physical Application into a Virtual Application	**eGCLI ModifyComponent -componenttype <Component Type> -hostname <the name of the host on which the target component is executing> (or) -hostip <IP address of the component> -componentname <The nick name of the component> -port <Port number> -virtualEnv yes -virtualserver <ComponentType:ComponentName:port> -externalagents <Comma-separated list of external agents> -managerid <ManagerID>**	eGCLI ModifyComponent -componenttype Web -**hostip** 192.168.10.54 -componentname web54 -port 80 -virtualEnv yes -virtualserver vSphere/ESX(i):esx136:NULL -externalagents agt25 -managerid 100mgr
		eGCLI ModifyComponent -componenttype Web -**hostname** egurkha25 -componentname web54 -port 80 -virtualEnv yes -virtualserver vSphere/ESX(i):esx136:NULL -externalagents agt25 -managerid 100mgr

2.4 Unmanaging Components

Task	Command	Example
Unmanaging a non-port-based component	**eGCLI UnmanageComponent -componenttype <Component Type> -componentname <Nick name of the component> -managerid <ManagerID>**	eGCLI UnmanageComponent -componenttype Generic -componentname gen20 -managerid 100mgr
Unmanaging a port-based component	**eGCLI UnmanageComponent -componenttype <Component Type> -componentname <Nick name of the component> -port <port number> -managerid <ManagerID>**	eGCLI UnmanageComponent -componenttype Microsoft SQL -componentname sql173 -port 1433 -managerid 100mgr
Unmanaging an Oracle database server	**eGCLI UnmanageComponent -componenttype <Component Type> -componentname <Nick name of the component> -port <port number> -sid <SID of the Oracle server> -managerid <ManagerID>**	eGCLI UnmanageComponent -componenttype Oracle Database -componentname ora8 -port 1521 -sid egora -managerid 100mgr **Note: An Oracle database server that supports multiple SIDs cannot be unmanaged using a single command; each SID should be passed to a separate 'UnmanageComponent' command.**

2.5 Removing Components

Task	Command	Example
Deleting a Component	**eGCLI DelComponent -componenttype <Component Type> -componentname <The nick name of the component> -port <Port> -managerid <ManagerID>**	eGCLI DelComponent -componenttype Web -componentname web54 -port 80 -managerid 100mgr
Deleting an Oracle database server with multiple instances	As already stated, if an Oracle database server is added with multiple SIDs, then the eG Enterprise system will monitor each SID as a different Oracle server. Therefore, while removing an Oracle database server that supports multiple SIDs, you cannot issue a single command to remove all the SIDs at one shot. Instead, the **DelComponent** command should be invoked separately for each SID.	eGCLI DelComponent -componenttype Oracle Database -componentname ora8 -port 1521 -sid egdemo -managerid 100mgr eGCLI DelComponent -componenttype Oracle Database -componentname ora8 -port 1521 -sid egora -managerid 100mgr

2.6 Displaying Components

Task	Command	Example
Displaying Components	**eGCLI ShowComponents -managerid <ManagerID> -componenttype <Component type> [-csvformat <yes/no>]**	eGCLI ShowComponents -managerid m51 -componenttype Oracle Database
		eGCLI ShowComponents -managerid m51 -componenttype Oracle Database -csvformat yes

2.7 Displaying Test names for a Component Type

Task	Command	Example
Displaying Tests for a Component Type	**eGCLI ShowTests -managerid <ManagerID> -componenttype <Component type> [-category <Enabled / Disabled / All>] [-testtype <Performance / Configuration>] [-csvformat <yes/no>]**	eGCLI ShowTests -managerid m51 -componenttype Oracle Database
		eGCLI ShowTests -managerid m51 -componenttype Oracle Database -category Enabled -csvformat yes
		eGCLI ShowTests -managerid m51 -componenttype Oracle Database -category All -testtype Configuration -csvformat yes
		eGCLI ShowTests -managerid m51 -componenttype Oracle Database -category Disabled -testtype Performance

2.8 Include/Exclude Components for Test

Task	Command	Example
Include Components for a test	**eGCLI IncludeComponentsForTest -managerid <Manager ID> -componenttype <Component type> [-testtype <Performance / Configuration>] -testname <Test name> -componentname <Comma-separated list of Component names:Port number>**	eGCLI IncludeComponentsForTest -managerid mgr1 -componenttype 2X Client Gateway -testtype performance -testname 2X Gateway Status -componentname client_1:80,client_2:80

		eGCLI IncludeComponentsForTest -managerid mgr1 -componenttype 2X Client Gateway -testtype configuration -testname Drives -componentname client_1:80,client_2:80
Exclude Components for a test	**eGCLI ExcludeComponentsForTest -managerid <Manager ID> -componenttype <Component type> [-testtype <Performance / Configuration>] -testname <Test name> -componentname <Comma-separated list of Component names:Port number>**	eGCLI ExcludeComponentsForTest -managerid mgr1 -componenttype 2X Client Gateway -testtype performance -testname 2X Gateway Status -componentname client_1:80,client_2:80
		eGCLI ExcludeComponentsForTest -managerid mgr1 -componenttype 2X Client Gateway -testtype configuration -testname Drives -componentname client_1:80,client_3:80

2.9 Include/Exclude Tests for Component

Task	Command	Example
Include Tests for a Component	**eGCLI IncludeTestsForComponent -managerid <Manager ID> -componenttype <Component type> [-testtype <Performance / Configuration>] -componentname <Component name:Port number> -testname <Comma-separated list of tests>**	eGCLI IncludeTestsForComponent -managerid mgr1 -componenttype Tomcat -testtype performance -componentname TomComp:8080 -testname Network Traffic,TCP
		eGCLI IncludeTestsForComponent -managerid mgr1 -componenttype Tomcat -testtype configuration -componentname TomComp:8080 -testname Environment Entries,Drives Capacity
Exclude Tests for a Component	**eGCLI ExcludeTestsForComponent -managerid <Manager ID> -componenttype <Component type> [-testtype <Performance / Configuration>] -componentname <Component name:Port number> -testname <Comma-separated list of tests>**	eGCLI ExcludeTestsForComponent -managerid mgr1 -componenttype Active Directory -testtype performance -componentname actDir:389 -testname AD Replications,Application Events
		eGCLI ExcludeTestsForComponent -managerid mgr1 -componenttype Active Directory -testtype configuration -componentname activeDir:389 -testname Operating System,Drives

Note:

1. Specify the port number against the component name in the command: **-componentname <Component name:Port number>** only if it is a port-based component.
2. The default values for the following parameters are:
   - Category – Enabled
   - Testtype – Performance
   - Csvformat - no

2.10 Adding External Agents

This section discusses the commands used by the eG CLI to add an external agent for monitoring.

Task	Command	Example
Adding an External Agent (without client emulation)	**eGCLI AddExternalAgent -hostname <the name of the external agent host> (or) -hostip <HostIP> -agentname <Nickname> -managerid <ManagerID>**	eGCLI AddExternalAgent -**hostip** 192.168.10.55 -agentname ext55 -managerid 100mgr
		eGCLI AddExternalAgent -**hostname** egurkha25 -agentname ext55 -managerid 100mgr
Adding an External Agent (with client emulation)	**eGCLI AddExternalAgent -hostname <the name of the host on which the target component is executing> (or) -hostip <HostIP> -agentname <Nickname> -clientemulation <yes/no> -managerid <ManagerID>**	eGCLI AddExternalAgent -**hostip** 192.168.10.55 -agentname ext55 -clientemulation yes -managerid 100mgr
		eGCLI AddExternalAgent -**hostname** egurkha25 -agentname ext55 -clientemulation yes -managerid 100mgr

2.11 Deleting External Agents

Command	Example
eGCLI DelExternalAgent -agentname <Nickname> -managerid <ManagerID>	eGCLI DelExternalAgent -agentname ext55 -managerid 100mgr

2.12 Displaying External Agents

Task	Command	Example
Displaying External Agents	**eGCLI ShowExternalAgents -managerid <ManagerID> [-csvformat <yes/no>]**	eGCLI ShowExternalAgents -managerid m51
		eGCLI ShowExternalAgents -managerid m51 -csvformat yes

2.13 Adding Remote Agents

Command	Example
eGCLI AddRemoteAgent -hostname <the name of the remote agent host> (or) -hostip <Host IP> -agentname <Nick name> -managerid <ManagerID>	eGCLI AddRemoteAgent -**hostip** 192.168.10.77 -agentname rem77 -managerid 100mgr
	eGCLI AddRemoteAgent -**hostname** egurkha25 -agentname rem77 -managerid 100mgr

2.14 Deleting Remote Agents

Command	Example
eGCLI DelRemoteAgent -agentname <Nick name> -managerid <ManagerID>	eGCLI DelRemoteAgent -agentname rem77 -managerid 100mgr

2.15 Displaying Remote Agents

Task	Command	Example
Displaying Remote Agents	**eGCLI ShowRemoteAgents -managerid <ManagerID> [-csvformat <yes/no>]**	eGCLI ShowRemoteAgents -managerid m51
		eGCLI ShowRemoteAgents -managerid m51 -csvformat yes

2.16 Associating/Disassociating Agents from Managers in a Redundant Setup

Task	Command	Example
Associating agents with a manager	**eGCLI AssignAgent -managerip <IP of the eG manager to which agents are to be assigned> -agents <Comma-separated list of agents> -managerID <ID of the primary manager>**	eGCLI AssignAgent -managerip 192.168.10.173 -agents ora8,gen12 -managerID mgr153
Disassociating agents from a manager	**eGCLI UnassignAgent -managerip <IP of the eG manager from which agents are to be delinked> -agents <Comma-separated list of agents> -managerID <ID of the primary manager>**	eGCLI UnassignAgent -managerip 192.168.10.173 -agents ora8,gen12 -managerID mgr153

Note:

While issuing a command, if the values you provide for command parameters include any of the special characters, namely - **<** (lesser than), **>** (greater than), **-** (hyphen), &(ampersand) - then such values should be enclosed within double quotes. For instance, while adding a component of type *Sybase (> 12.5)*, your command specification should be:

eGCLI AddComponent -componenttype "Sybase (> 12.5)" -hostip 192.168.10.228 -componentname syb228 -port 5000 -externalagents 192.168.10.74,extagnt175 -managerid mgr157

2.17 Adding/Modifying/Deleting Quick Maintenance Policies

Task	Command	Example
Adding Quick Maintenance Policy	**eGCLI AddQuickMaintenancePolicy -managerid <Manager ID> -policyname <Policy name> -timefrequency <Daily / First day of month / Last day of month / Sunday / Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / <Start Date- End Date>=<Start Time-End Time>**	eGCLI AddQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -timefrequency Daily=1:00-2:00,First day of month=1:00-2:00, Last day of month=1:00-2:00, Sunday=1:00-2:00,08/30/2010-09/30/2010=1:00-2:00
Modifying Quick Maintenance Policy for adding timeperiod	**eGCLI ModifyQuickMaintenancePolicy -managerid <Manager ID> -policyname <Policy name> -addtimefrequency <Daily / First day of month / Last day of month / Sunday / Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / <Start Date-End Date>=<Start Time - End Time>**	eGCLI ModifyQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -addtimefrequency Daily=5:00-10:00,First day of month=5:00-10:00
Modifying Quick Maintenance Policy for removing timeperiod	**eGCLI ModifyQuickMaintenancePolicy -managerid <Manager ID> -policyname <Policy name> -rmtimefrequency <Daily / First day of month / Last day of month / Sunday / Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / <Start Date-End Date>=<Start Time - End Time>**	eGCLI ModifyQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -rmtimefrequency Daily=1:00-2:00,First day of month=1:00-2:00
Deleting Quick Maintenance Policy	**eGCLI DeleteQuickMaintenancePolicy -managerid <Manager ID> -policyname <Policy name> [-allpolicies <yes>]**	eGCLI DeleteQuickMaintenancePolicy -managerid mgr51 -policyname QMP1
Deleting Quick Maintenance Policy	**eGCLI DeleteQuickMaintenancePolicy -managerid <Manager ID> -allpolicies <yes>**	eGCLI DeleteQuickMaintenancePolicy -managerid mgr51 -allpolicies yes

Note:

- The Start Date / End Date must be in **MM/DD/YYYY** format.
- The Start Time / End Time must be in **HH:MM** format.

2.18 Assigning Quick Maintenance Policies to Hosts/Tests/Components

Task	Command	Example
Assigning Quick Maintenance Policy to Host	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor Host -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Host -disassociateelements agateport,Gen1,ora10 -associateelements ora2,win2
Assigning Quick Maintenance Policy to Zone Component	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor Component -componentsby Zone -zone <Zone name> -associateelements < Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Component -componentsby Zone -zone Zone1 -associateelements agateport:3900,ActDir1111:7077 -disassociateelements Gen1,Oracle221:15:oracleSID1112
Assigning Quick Maintenance Policy to Segment Component	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor <Component> -componentsby Segment -segment <Segment name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Component -componentsby Segment -segment Segm1 -associateelements win1,agate1:3900 -disassociateelements agate2:3900,tom1:8080
Assigning Quick Maintenance Policy to Service Component	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor Component -componentsby Service -service <Service name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Component -componentsby Service -service ServiceiiSWeb -associateelements win3,iisweb1:8080 -disassociateelements apacheweb1:8080
Assigning Quick Maintenance Policy to Component of a Particular Type	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor Component -componentsby Component Type -componenttype <Component type> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Component -componentsby Component Type -componenttype AGate -associateelements AgateTest2:3900,Gen1:3901 -disassociateelements agateport:3900
Assigning Quick Maintenance Policy to Test	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor Test -associateelements**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Test -associateelements

	<Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>	AGateAccess,AGateServer -disassociateelements AGateStatus,AGateTrans
Assigning Quick Maintenance Policy to Test For Host	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor Test For Host -test <Test name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Test For Host -test DiskActivity -associateelements ora2,ora3 -disassociateelements ftp4,ftp5
Assigning Quick Maintenance Policy to Test For Component	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor Test For component -test <Test name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Test For Component -test AGateAccess -associateelements agateport:3900,Gen1:3901 -disassociateelements AgateTest2:3900
Assigning Quick Maintenance Policy to Test For Component Type	**eGCLI AssignQuickMaintenancePolicy -managerid <manager Id> -policyname <Policy name> -associatefor Test For component type -componenttype <Component Type> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignQuickMaintenancePolicy -managerid mgr51 -policyname QMP1 -associatefor Test For Component Type -componenttype Oracle Database -associateelements IOWaits -disassociateelements MemoryDetails,Memory Usage

2.19 Adding/Modifying/Deleting Independent Policies

Task	Command	Example
Adding Independent Policy	**eGCLI AddMaintenancePolicy -managerid <Manager ID> -policyname <Policy name> -timefrequency <Daily/First day of month/Last day of month /Sunday/Monday/Tuesday/Wednesday /Thursday/Friday/Saturday / <Start Date - End Date> =<Start Time - End Time>**	eGCLI AddMaintenancePolicy -managerid mgr51 -policyname IP1 -timefrequency Daily=1:00-2:00,First day of month=1:00-2:00, Last day of month=1:00-2:00,Sunday=1:00-2:00,08/30/2010-09/30/2010=1:00-2:00
Modifying Independent Policy for adding Timeperiod	**eGCLI ModifyMaintenancePolicy -managerid <Manager ID> -policyname <Policy name> -addtimefrequency <Daily / First day of month / Last day of month / Sunday / Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / <Start Date - End Date>=<Start Time - End time>**	eGCLI ModifyMaintenancePolicy -managerid mgr51 -policyname IP1 -addtimefrequency Daily=5:00-10:00,First day of month=5:00-10:00
Modifying Independent Policy for adding Timeperiod	**eGCLI ModifyMaintenancePolicy -managerid <Manager ID> -policyname <Policy name> -rmtimefrequency <Daily / First day of month / Last day of month / Sunday / Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / <Start Date - End Date>=<Start Time - End time>**	eGCLI ModifyMaintenancePolicy -managerid mgr51 -policyname IP1 -rmtimefrequency Daily=1:00-2:00,First day of month=1:00-2:00

Deleting Independent Policy	**eGCLI DeleteMaintenancePolicy -managerid <Manager ID> -policyname <Policy name>**	eGCLI DeleteMaintenancePolicy -managerid mgr51 -policyname IP1,IP3
Deleting Independent Policy	**eGCLI DeleteMaintenancePolicy -managerid <Manager ID> -allpolicies <yes>**	eGCLI DeleteMaintenancePolicy -managerid mgr51 -allpolicies yes

2.20 Displaying the Maintenance Policies

Task	Command	Example
Displaying the Maintenance Policies	**eGCLI ShowMaintenancePolicyNames -managerid <Manager ID> [-csvformat <yes/no>]**	eGCLI ShowMaintenancePolicyNames -managerid mgr51
		eGCLI ShowMaintenancePolicyNames -managerid mgr51 –csvformat yes

2.21 Displaying the details of the Maintenance Policies

Task	Command	Example
Displaying the details of the Maintenance Policies	**eGCLI ShowMaintenancePolicyDetails -managerid <Manager ID> [-policyname <Comma-separated list of Policy names>] [-csvformat <yes/no>]**	eGCLI ShowMaintenancePolicyDetails -managerid mgr51
		eGCLI ShowMaintenancePolicyDetails -managerid mgr51 -policyname IP1,IP2
		eGCLI ShowMaintenancePolicyDetails -managerid mgr51 -policyname IP1,IP2 –csvformat yes

2.22 Adding/Modifying/Deleting Maintenance Policy Groups

Task	Command	Example
Adding Maintenance Policy Group	**eGCLI AddMaintenancePolicyGroup -managerid <Manager ID> -groupname <Group name> -policyname <Comma-Separated List of Policy names>**	eGCLI AddMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -policyname IP1,IP2
Modifying Maintenance Policy Group to Associate Policies	**eGCLI ModifyMaintenancePolicyGroup -managerid <Manager ID> -groupname <Group name> -associatepolicy <Comma-Separated List of Policy names>**	eGCLI ModifyMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatepolicy IP3
Modifying Maintenance Policy Group to Disassociate Policies	**eGCLI ModifyMaintenancePolicyGroup -managerid <Manager ID> -groupname <Group name> -disassociatepolicy <Comma-Separated List of Policy names >**	eGCLI ModifyMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -disassociatepolicy IP2
Deleting Maintenance Policy Group	**eGCLI DeleteMaintenancePolicyGroup -managerid <Manager ID> -groupname <Group name>**	eGCLI DeleteMaintenancePolicyGroup -managerid mgr51 -groupname PG1
Deleting Maintenance Policy Group	**eGCLI DeleteMaintenancePolicyGroup -managerid <Manager ID> -allgroups <yes>**	eGCLI DeleteMaintenancePolicyGroup -managerid mgr51 -allgroups yes

2.23 Assigning Independent Policy to Maintenance Policy Group

Task	Command	Example
Adding Independent Policy to Maintenance Policy Group	**eGCLI AddMaintenancePolicyToPolicyGroups -managerid <Manager ID> -policyname <Policy name> -groupnames <Comma-Separated List of Group names>**	eGCLI AddMaintenancePolicyToPolicyGroups -managerid mgr51 -policyname IP4 -groupnames PG1,PG2

Modifying Independent Policy that is added to Maintenance Policy Group	**eGCLI ModifyMaintenancePolicyGroup -managerid <Manager ID> -groupname <Group name> -associatepolicy <Comma-Separated List of Policy names>**	eGCLI ModifyMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatepolicy IP3
Modifying Independent Policy that is added to Maintenance Policy Group	**eGCLI ModifyMaintenancePolicyGroup -managerid <Manager ID> -groupname <Group name> -disassociatepolicy <Comma-Separated List of Policy names>**	eGCLI ModifyMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -disassociatepolicy IP2
Deleting Independent Policy from a Maintenance Policy Group	**eGCLI DeleteMaintenancePolicyGroup -managerid <Manager ID> -groupname <Group name>**	eGCLI DeleteMaintenancePolicyGroup -managerid mgr51 -groupname PG1
Deleting Independent Policy from a Maintenance Policy Group	**eGCLI DeleteMaintenancePolicyGroup -managerid <Manager ID> -allgroups <yes>**	eGCLI DeleteMaintenancePolicyGroup -managerid mgr51 -allgroups yes

2.24 Assigning Policy Groups to Infrastructure Elements

Task	Command	Example
Assigning Maintenance Policy Group to Host	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Host -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Host -disassociateelements agateport,Gen1,ora10 -associateelements ora2,win2
Assigning Maintenance Policy Group to Zone Component	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Component -componentsby Zone -zone <Zone name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Component -componentsby Zone -zone Zone1 -associateelements agateport:3900,ActDir1111:7077 -disassociateelements Gen1,Oracle221:15:oracleSID1112
Assigning Maintenance Policy Group to Segment Component	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Component -componentsby Segment -segment <Segment name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Component -componentsby Segment -segment Segm1 -associateelements agate1t:3900,agate2:3900

	List of Elements>	-disassociateelements win2,agate3:3900
Assigning Maintenance Policy Group to Service Component	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Component -componentsby Service -service <Service name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Component -componentsby Service -service ServiceiiSWeb -associateelements win3 -disassociateelements iisweb1:80
Assigning Maintenance Policy Group to Component of a Particular Type	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Component -componentsby ComponentType -componenttype <Component Type> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Component -componentsby ComponentType -componenttype AGate -associateelements AgateTest2:3900,Gen1:3901 -disassociateelements agateport:3900
Assigning Maintenance Policy Group to Test	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Test -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Test -associateelements AGateAccess,AGateServer -disassociateelements AGateStatus,AGateTrans
Assigning Maintenance Policy Group to Test For Host	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Test For Host -test <Test name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Test For Host -test DiskActivity -associateelements ora2,ora3 -disassociateelements ftp4,ftp5
Assigning Maintenance Policy Group to Test For Component	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Test For Component -test <Test name> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Test For Component -test AGateAccess -associateelements agateport:3900,Gen1:3901 -disassociateelements AgateTest2:3900
Assigning Maintenance Policy Group to Test For Component Type	**eGCLI AssignMaintenancePolicyGroup -managerid <manager Id> -groupname <Group name> -associatefor Test For Component Type -componenttype <Component Type> -associateelements <Comma-Separated List of Elements> -disassociateelements <Comma-Separated List of Elements>**	eGCLI AssignMaintenancePolicyGroup -managerid mgr51 -groupname PG1 -associatefor Test For Component Type -componenttype Oracle Database -associateelements IOWaits -disassociateelements MemoryDetails,Memory Usage

2.25 Enabling / Disabling Tests

Task	Command	Example
Enabling Tests	**eGCLI EnableTests -managerid <Manager ID> -componenttype <Component Type> [-testtype <Performance / Configuration>] -tests <Comma-Separated List of Test names>**	eGCLI EnableTests -managerid mgr51 -componenttype AGate –testtype Performance –tests HandlesUsage,MemoryDetails
		eGCLI EnableTests -managerid mgr51 -componenttype Oracle Database -testtype Configuration -tests Drives,Environment Entries
Disabling Tests	**eGCLI DisableTests -managerid <Manager ID> -componenttype <Component Type> [-testtype <Performance / Configuration>] -tests < Comma-Separated List of Test names>**	eGCLI DisableTests -managerid mgr51 -componenttype Oracle Database -testtype Performance -tests Memory Details,Oracle User Connections
		eGCLI DisableTests -managerid m51 -componenttype AGate -testtype ConfigurationTest -tests Drives,Software

Note:

- The test type syntax is case-insensitive and therefore the syntax can be represented as either **Performance / PerformanceTest / performance / performancetest or Configuration / ConfigurationTest / configuration / configurationtest.**
- The default test type is **Performance**.

2.26 Displaying the details of the Tests

Task	Command	Example
Displaying the details of the Tests	**eGCLI ShowTestsDetails -managerid <Manager ID> [-componenttype <Component type>] [-componentname <Component name>] [-testtype <Performance/Configuration>] [-testname <Test name>] [-csvformat <yes/no>]**	eGCLI ShowTestsDetails -managerid mgr1
		eGCLI ShowTestsDetails -managerid mgr1 -componenttype 2X Client Gateway -componentname client_1:80
		eGCLI ShowTestsDetails -managerid mgr1 -componenttype 2X Client Gateway -componentname client_1:80 -testtype performance -testname Handles Usage
		eGCLI ShowTestsDetails -managerid mgr1 -componenttype 2X Client Gateway -componentname client_1:80 -testtype configuration -testname FTP Server Details –csvformat yes

Note:

- In this case, if the test type is not specified, both the **Performance** and **Configuration** tests will be displayed as against the default test type of **Performance** for all other CLI commands.
- The default value for the **csvformat** parameter is **no**.

Note:

The Maintenance Policy Configuration and Test Configruation tasks cannot be performed in bulk i.e. through xml and csv files.

2.27 Displaying the Hosts managed in the target environment

Task	Command	Example
Displaying the hosts managed in the target environment	**eGCLI ShowManagedHosts -managerid <Manager ID> -agentname<Agent name> [-agenttype <externalagent/remoteagent>] [-csvformat <yes/no>]**	eGCLI ShowManagedHosts -managerid mgr1-agentname 192.168.8.202
		eGCLI ShowManagedHosts -managerid mgr1 -agentname 192.168.8.202 -agenttype externalagent

Note:

The **agenttype** is case insensitive

2.28 Adding a User

Task	Command	Example
Adding a user	**eGCLI AddUser -managerid <Manager ID> -userrole <User role> -userid <User ID> -password <Password> -expirydate < MM/dd/yyyy> [-alarmsbymail <Critical/Major/Minor/All> -to <comma-separated list of Mail IDs/Mobile numbers> -cc <comma-separated list of Mail IDs/Mobile**	eGCLI AddUser -managerid mgr1 -userrole admin -userid saran -password saran -expirydate 12/12/2014 -alarmsbymail critical -to saranya@eginnovations.com -cc saran1@eginnovations.com -bcc example@eginnovations.com

	numbers> -bcc <comma-separated list of Mail IDs/Mobile numbers>]	eGCLI AddUser -managerid mgr1 -userrole admin -userid saran -password saran -expirydate 12/12/2014 -alarmsbymail critical -to saranya@eginnovations.com,9840391695 -cc saran1@eginnovations.com -bcc example@eginnovations.com
		eGCLI AddUser -managerid mgr1 -userrole monitor -userid saran -password saran -expirydate 12/12/2014 -alarmsbymail critical -to saranya@eginnovations.com,9840391695 -cc saran1@eginnovations.com -bcc example@eginnovations.com

2.29 Modifying a User

Task	Command	Example
Modifying a user	**eGCLI ModifyUser -managerid <Manager ID -userid <User ID> [-userrole <User role> -password <Password> -expirydate < MM/dd/yyyy> -alarmsbymail <Critical/Major/Minor/All> -to <comma-separated list of Mail IDs/Mobile numbers> -cc <comma-separated list of Mail IDs/Mobile numbers> -bcc <comma-separated list of Mail IDs/Mobile numbers>]**	eGCLI ModifyUser -managerid mgr1 -userid saran -userrole admin -password saran -expirydate 12/13/2014 -alarmsbymail major -to saran1@eginnovations.com -cc saranya@eginnovations.com -bcc sample@eginnovations.com
		eGCLI ModifyUser -managerid mgr1 -userid saran -userrole admin -password saran -expirydate 12/13/2014 -alarmsbymail major -to saran1@eginnovations.com,9884011111 -cc saranya@eginnovations.com,9840391695 -bcc sample@eginnovations.com

2.30 Deleting a User

Task	Command	Example
Deleting a user	**eGCLI DeleteUser -managerid <Manager ID> -userid <comma-separated list of User IDs>**	eGCLI DeleteUser -managerid mgr1 -userid saran, john

2.31 Associating Components to a User

Task	Command	Example
Associating components to a user	**eGCLI AssociateComponentsToUser -managerid <Manager ID> -userid <User ID> -componenttype <Component Type> <-components <comma-separated list of Nick names/all> [-autoassociatetype <yes/no>]**	eGCLI AssociateComponentsToUser -managerid mgr1 -userid mon -componenttype 2x client gateway -components client_1:80,client_2:80
		eGCLI AssociateComponentsToUser -managerid mgr1 -userid mon -componenttype 2x client gateway -components all –autoassociatetype yes

Note:

It is mandatory to provide either the **components** or the **autoassociatetype** while associating the components to a user. If you wish to provide only the **autoassociatetype** option, then set this to **yes**. By default, the **autoassociatetype** option is set to **no**.

2.32 Adding a Zone

Task	Command	Example
Adding a zone	**eGCLI AddZone -managerid <manager Id> -zonename <Zone name> -elements <comma-separated list of elements> [-displayimage <Display image>] [-autoassociate <yes/no>]**	eGCLI AddZone -managerid mgr101 -zonename eastzone -elements IIS Web:web_2:80,Group:dbgroup,Segment:seg1,Service:myservce -displayimage Web
		eGCLI AddZone -managerid mgr101 -zonename eastzone -elements IIS Web:web_2:80,Group:dbgroup,Segment:seg1,Service:myservce -displayimage Web –autoassociate yes

2.33 Modifying a Zone

Task	Command	Example
Modifying a zone	**eGCLI ModifyZone -managerid <manager Id> -zonename <Zone name> [-associateelements <Elements>] [-disassociateelements <Elements>] [-displayimage <Display image>] [-autoassociate <yes/no>]**	eGCLI ModifyZone -managerid mgr101 -zonename eastzone -associateelements Group:citrixgroups –disassociateelements IIS Web:web_2:80 -displayimage Others
		eGCLI ModifyZone -managerid mgr101 -zonename eastzone -associateelements Group:citrixgroups –disassociateelements IIS Web:web_2:80 -displayimage Others Web –autoassociate yes

2.34 Renaming a Zone

Task	Command	Example
Renaming a zone	**eGCLI RenameZone -managerid <manager Id> -zonename <Zone name> -newzonename <New zone name>**	eGCLI RenameZone -managerid mgr101 -zonename eastzone -newzonename northeastzone

2.35 Deleting a Zone

Task	Command	Example
Deleting a zone	**eGCLI DeleteZone -managerid <manager Id> -zonename <comma-separate list of Zones>**	eGCLI DeleteZone -managerid mgr101 -zonename northeastzone,northwestzone

2.36 Adding a Group

Task	Command	Example
Adding a Group	**eGCLI AddGroup -managerid <manager Id> -groupname <Group name> -elements <Comma-separated list of elements>**	eGCLI AddGroup -managerid mgr101 -groupname mygroup –elements Generic:gen1,IIS Web:iis1:80

2.37 Modifying a Group

Task	Command	Example
Modifying a Group	**eGCLI ModifyGroup -managerid <manager Id> -groupname <Group name> [-associateelements <Elements> / -disassociateelements <Elements>]**	eGCLI ModifyGroup -managerid mgr101 -groupname mygroup -associateelements Active Directory:ad:1234 -disassociateelements IIS Web:iis1:80

2.38 Renaming a Group

Task	Command	Example
Renaming a Group	**eGCLI RenameGroup -managerid <manager Id> -groupname <Group name> -newgroupname <New group name>**	eGCLI RenameGroup -managerid mgr101 -groupname mygroup -newgroupname mynewgroup

2.39 Deleting a Group

Task	Command	Example
Deleting a Group	**eGCLI DeleteGroup -managerid <manager Id> -groupname <Comma-separated list of groups>**	eGCLI DeleteGroup -managerid mgr101 -groupname mynewgroup,dbgroup

2.40 Determining the Success/Failure of a Command

The eG CLI indicates the success/failure of a command by returning a message to that effect. While the successful execution of a command is indicated by a message that begins with the string "**Result :**", when a command fails, an error message is returned, which typically begins with the string "**Error :**". This **Error** message also briefly describes the reason for the failure of the command.

For instance, if a component addition completes successfully, the eG CLI will return the following message:

Result : Component added successfully.

Similarly, if a component could not be added owing to some reason, the CLI will return a message of this type:

Error : The host/nick name you are trying to add already exists.Please use another host/nick name.

Note:

Typically, the eG CLI displays the success/failure messages in the default language of the eG manager with which it connects. However, since the CLI logs into the eG manager as an **Admin** user, the language preference set by this user in his/her profile (in the eG web interface) will automatically override the default language setting of the manager; this implies that all messages returned by the CLI will be in the preferred language of the **Admin** user.

2.41 Command Logging

To enable you to track the status of the eG CLI commands and to troubleshoot issues with the same, the eG manager automatically logs the command line activities in the **CLILog0.log** file created in the **<EGADMINCLI_INSTALL_DIR>\logs** directory. By default, the **CLILog0.log** file will be allowed to grow upto a maximum size of 2MB, beyond which the contents of the **CLILog0.log** file will be copied to a **CLILog1.log** file, and the newer information will be logged in the **CLILog0.log** file. This way, a maximum of 5 log files will be created, with the last log file being **CLILog4.log** file. This log rotation mechanism helps ensure that the log file does not grow beyond control.

Chapter 3

Performing Operations in Bulk using the eG CLI

One of the key benefits of the eG CLI is that, unlike the eG web-based interface, where only one configuration operation can be performed at a given point in time, the eG CLI allows you to process commands in bulk, and thus allows you to perform multiple tasks simultaneously - for instance, you can add multiple components at one shot using the CLI. This saves administrators the time and trouble involved in performing redundant tasks.

To execute commands in bulk, the CLI requires an XML or a CSV file that contains the details of the operations to be performed. This file (XML or CSV) should be created on the system hosting the orchestration engine.

Once the file is created, invoke the relevant CLI command from the automation tool or the command prompt by providing the manager ID and the full path to the XML or CSV file. The command will then read the instructions defined in the XML or CSV file and then execute them on the specified manager to perform the operation.

Note:

To ensure that the XML and CSV files support multi-byte component names and agent nick names, do the following:

- Make sure that the eG CLI is installed on a multi-byte operating system;
- Make sure that you save the XML and CSV files in the **UTF-8** mode

The sections below discuss how the XML and CSV files can be used for performing multiple administrative operations on an eG manager.

3.1 Adding Components in Bulk

3.1.1 Component Addition Using an XML File

Typically, a dedicated XML file is required for each operation that needs to be performed by eG CLI. For instance, to add components in bulk, you will require a single XML that has to be configured with the details of only those components that need to be added to the eG Enterprise system. The XML created for the sole purpose of adding components should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="add">
   <Agent>
          <Details of component1>
   </Agent>
   <Agent>
          <Details of component2>
   </Agent>
                .
                .
                .
                .
</Component>
```

Within each *<Agent> </Agent>* segment in the XML block above, you need to provide the details of a component to be added. For instance, to add 2 Oracle database server components at the same time using the host IP of the component, your XML specification would be:

```
<Component action="add">
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.8</hostip>
          <componentname>ora8</componentname>
          <port>1521</port>
          <sid>egora</sid>
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152</externalagents>
   </Agent>
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.9</hostip>
          <componentname>ora9</componentname>
          <port>1521</port>
          <sid>egdemo</sid>
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152</externalagents>
   </Agent>
</Component>
```

If you would like to add 2 Oracle databse server compoennts at the same time using the host name of the component, your XML specification would be:

```
<Component action="add">
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostname>egurkha25</hostip>
          <componentname>ora8</componentname>
          <port>1521</port>
          <sid>egora</sid>
```

```
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152</externalagents>
   </Agent>
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostname>egurkha26</hostname>
          <componentname>ora9</componentname>
          <port>1521</port>
          <sid>egdemo</sid>
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152</externalagents>
   </Agent>
</Component>
```

While the XML tags such as *<componenttype>*, *<hostip>*, etc. cannot be altered, the data contained within each tag can change depending upon the component being added. **Note that the XML tags are the same as the input parameters of the 'AddComponent' command supported by eG CLI** (see Section 2.1 of this document).

You can remove tags that you deem are unnecessary for a component specification - for eg., while adding a non-port-based component, you may want to omit the *<port></port>* tag. Likewise, you can even add more tags depending upon the nature of components to be added.

If say, a component being added is associated with multiple external agents, then a comma-separated list of external agents will have to be provided within the *<externalagents></externalagents>* tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="add">
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.8</hostip>
          <componentname>ora8</componentname>
          <port>1521</port>
          <sid>egora</sid>
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152,ext72</externalagents>
   </Agent>
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.9</hostip>
          <componentname>ora9</componentname>
          <port>1521</port>
          <sid>egdemo</sid>
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152,ext75</externalagents>
   </Agent>
</Component>
```

You can either add the components using the host IP or the host name of the component.

Similarly, to add an Oracle database server with multiple SIDs, all the SIDs should be provided as a comma-separated list within the *<sid></sid>* tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="add">
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.8</hostip>
          <componentname>ora8</componentname>
          <port>1521</port>
          <sid>egora,egdemo</sid>
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152,ext72</externalagents>
```

```
    </Agent>
</Component>
```

Note:

If an Oracle database server with multiple SIDs is added to the eG Enterprise system via an XML file, then each SID will be registered as a separate Oracle database server in the eG Enterprise system.

Note:

- Each component specification should be included within the main *<Component></Component>* block only.
- A single XML file should not contain more than one *<Component></Component>* block.
- An XML file that is created for adding components should not include sections pertaining to any other operation such as component modification/deletion, addition of external agents, etc.
- A single XML file can be used for adding components of different types.

Once all the required entries are defined in the XML file, execute the following command to extract the component information from the file, connect to the required eG manager, and add the specified components to the eG Enterprise system:

eGCLI AddComponent -managerID <ManagerID> -file <Full_path_to_the_XML_file>

For example:

eGCLI AddComponent -managerID mgr153 -file c:\CLI.xml

3.1.2 Component Addition Using a CSV File

Like XML files, separate CSV files should be created for every administrative operation to be performed by eG CLI. This implies that for adding new components to the eG Enterpise system, a dedicated CSV file is required.

To configure a CSV file with the details of the components to be added, entries of the following format should be included in that file:

```
Element,action
Component,add
componenttype,hostip/hostname,componentname,port,externalagents
<Details of component1>
<Details of component2>
.
.
.
```

For example, if you want to add 3 IIS web servers to the eG Enterprise system using the host IP, do the following:

```
Element,action
Component,add
componenttype,hostip,componentname,port,mtsenabled,externalagents
IIS web,192.168.10.96,iis96,80,no,ext43
IIS web,192.168.10.173,iis173,7077,no,ext173
IIS web,192.168.10.90,web90,80,no,ext85
```

If you want to add 3 IIS web servers to the eG Enterprise system using the host name of the component, do the following:

```
Element,action
Component,add
componenttype,hostname,componentname,port,mtsenabled,externalagents
IIS web,egurkha25,iis96,80,no,ext43
IIS web,egurkha26,iis173,7077,no,ext173
IIS web,egurkha27,web90,80,no,ext85
```

Note that the column names (componenttype, hostip, etc.) used here are the same as the input parameters of the 'AddComponent' command supported by eG CLI (see Section 2.1 of this document). These column names cannot be changed. Also, while providing the details of the components to be added, ensure that you follow the same order of the column names.

While adding components of different types or those which support different parameter sets, make sure that you leave the columns not applicable for a component specification, blank. At the same time, ensure that the column names you specify in the CSV file are a super-set of the parameters supported by all the components that are being added. In other words, the column names provided in the CSV file should correspond to the following:

- the parameters that are common across all the components to be added, and;
- the parameters that are distinct/unique for each of the components being added;

For instance, you can add an IIS web server and a Generic server component using the same CSV file, with the help of the following specification:

```
Element,action
Component,add
componenttype,hostip,componentname,port,mtsenabled,externalagents
IIS web,192.168.10.96,iis96,80,no,ext43
Generic,192.168.10.173,gen173,,,ext180
```

In this case, note that the columns *componenttype*, *hostip*, *componentname*, and *externalagents* are common for both the IIS web server and the Generic server, but the *port* and *mtsenabled* columns are supported only by the IIS web server. Moreover, since the Generic server is a non-port-based component and does not support the *mtsenabled* parameter, the columns *port* and *mtsenabled* have been left blank in the specification that corresponds to the Generic server.

If you want to say, associate multiple external agents with a component, then your specification should include a comma-separated list of external agents provided within double-quotes:

```
Element,action
Component,add
componenttype,hostip,componentname,port,mtsenabled,externalagents
IIS web,192.168.10.96,iis96,80,no,"ext43,ext60"
IIS web,192.168.10.173,iis173,7077,no,ext173
```

Similarly, you can add an Oracle database sever with multiple SIDs.

Note:

If an Oracle database server with multiple SIDs is added to the eG Enterprise system, then each SID will be registered as a separate Oracle database server in the eG Enterprise system.

Once all the required entries are defined in the CSV file, execute the following command to extract the component information from the file, connect to the required eG manager, and add the specified components to the eG Enterprise system:

eGCLI AddComponent -managerID <ManagerID> -file <Full_path_to_the_CSV_file>

For example:

eGCLI AddComponent -managerID mgr153 -file c:\addiis.csv

3.2 Managing Components in Bulk

3.2.1 Managing Components Using an XML File

The first step towards this to create an XML file for the sole purpose of managing multiple components of different component-types at one shot. Given below is the format of the entries in such a file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="manage">
          <Agent>
                <Component1 to be managed>
          </Agent>
          <Agent>
                <Component2 to be managed>
          </Agent>
                    .
                    .
                    .
</Component>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="manage">
          <Agent>
                <componenttype>Generic</componenttype>
                <componentname>gen1</componentname>
          </Agent>
          <Agent>
                <componenttype>AGate</componenttype>
                <componentname>agte10</componentname>
                <port>3900</port>
          </Agent>
          <Agent>
                <componenttype>Oracle Database</componenttype>
                <componentname>ora8</componentname>
                <port>1521</port>
                <sid>egora</sid>
```

```
        </Agent>
    </Component>
```

Note that the XML tags are the same as the input parameters of the 'ManageComponent' command supported by eG CLI (see Section 2.2 of this document).

Note:

If an Oracle database server with multiple SIDs is be managed, then your *<sid></sid>* tag cannot include a comma-separated list of SIDs; instead, your XML file should include a separate *<Agent></Agent>* block for each SID to be managed.

Note:

- Each component specification should be included within the main *<Component></Component>* block only.
- A single XML file should not contain more than one *<Component></Component>* block.
- An XML file that is created for managing components should not include sections pertaining to any other operation.
- A single XML file can be used for managing components of different types.

Once the XML file is ready, execute the following command to extract the component information from the file, connect to the required eG manager, and manage the specified components:

eGCLI ManageComponent -managerID <ManagerID> -file <Full_path_to_the_XML_file>

For example:

eGCLI ManageComponent -managerID mgr153 -file c:\mancomp.xml

3.2.2 Managing Components using a CSV File

To manage components in bulk, you can create a dedicated CSV file for this purpose and configure it with entries related to each component to be managed. Given below is the format of the entries in such a file:

```
Element,action
Component,manage
componenttype,componentname,port,sid
<Component1 to be managed>
<Component2 to be managed>
.
.
.
```

For instance:

```
Element,action
Component,manage
componenttype,componentname,port,sid
```

```
Generic,gen1,,
AGate,agte10,3900,
Oracle Database,ora55,1521,multi
```

Note that the column names used in the CSV file are the same as the input parameters of the 'ManageComponent' command supported by the CLI (see Section 2.2 of this document).

Note:

- If an Oracle database server with multiple SIDs is be managed, then the entry for the Oracle server in your CSV file should not include a comma-separated list of SIDs; instead, you should provide a separate entry for each SID to be managed.
- If one/more column names in your CSV file are not applicable to a component specification, then make sure that such columns are left empty in the specification.

Once the CSV file is ready, execute the following command to extract the component information from the file, connect to the required eG manager, and manage the specified components:

eGCLI ManageComponent -managerID <ManagerID> -file <Full_path_to_the_CSV_file>

For example:

eGCLI ManageComponent -managerID mgr153 -file c:\mancomp.csv

3.3 Modifying Components in Bulk

3.3.1 Component Modification Using an XML File

The XML file that is created exclusively for modifying multiple component specifications, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="modify">
   <Agent>
         <Modification to component1>
   </Agent>
   <Agent>
         <Modification to component2>
   </Agent>
                .
                .
                .
                .
</Component>
```

For example, to change the nick name of an Oracle database server and the port number of an MS SQL server that pre-exist, your specification would be:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="modify">
```

```
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.8</hostip>
          <oldcomponentname>oracle8</oldcomponentname>
          <newcomponentname>oracle80</newcomponentname>
          <port>1521</port>
          <sid>egora</sid>
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152</externalagents>
   </Agent>
   <Agent>
          <componenttype>Microsoft SQL</componenttype>
          <hostip>192.168.10.9</hostip>
          <componentname>mssql</componentname>
          <oldport>1433</oldport>
          <newport>1434</newport>
          <ispassive>no</ispassive>
          <externalagents>192.168.10.152</externalagents>
   </Agent>
</Component>
```

You can either specify the hostip or the hostname while you try to modify the components.

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the XML file:

eGCLI ModifyComponent -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

eGCLI ModifyComponent -managerID mgr153 -file c:\modcomp.xml

Note:

- Each modification should be included within the main *<Component></Component>* block only.
- A single XML file should not contain more than one *<Component></Component>* block.
- An XML file that is created for modifying component details should not include sections pertaining to any other operation.
- A single XML file can be used for modifying the details of components of different types.

3.3.2 Component Modification Using a CSV File

As already mentioned, an exclusive CSV file will have to be created to handle bulk modifications to component details.

Entries in this file should be of the following format:

```
Element,action
Component,modify
componenttype,hostip/hostname,componentname,port,externalagents
<Modification to component1>
<Modification to component2>
.
```

```
.
.
```

For example, if you want to modify the port numbers of 2 IIS web servers, do the following:

```
Element,action
Component,modify
componenttype,hostip,componentname,oldport,Newport,mtsenabled,externalagents
IIS web,192.168.10.96,iis96,7077,8088,no,ext43
IIS web,192.168.10.173,iis173,7077,7078,no,ext173
```

The CSV file can also be used to modify the details of components of multiple types at one shot. While doing so, make sure that you leave the columns not applicable for a component specification, blank. At the same time, ensure that the column names you specify in the CSV file are a super-set of the parameters supported by all the components that are being modified. In other words, the column names provided in the CSV file should correspond to the following:

- the parameters that are common across all the components to be modified, and;
- the parameters that are distinct/unique for each of the components being modified;

For instance, say you want to modify the nick name of an Oracle database server, and want to change the monitoring mode of an MS SQL server from agent-based to agentless. The specification in this case will be, as follows:

```
Element,action
Component,modify
componenttype,hostip,componentname,oldcomponentname,
newcomponentname,port,sid,agentless,mode,os,externalagents,remoteagent
Oracle database,192.168.10.8,,ora8,ora08,1521,egora,,,,ext125,
Microsoft SQL,192.168.10.63,
sql63,,,1433,,yes,perfmon,Windows2003,ext173,rem12
```

In the above specification, you can find that the column list includes the the following:

- parameters such as *componenttype, hostip/hostname, port,* and *externalagents* that are common to both the Oracle and MS SQL servers
- the *oldcomponentname, newcomponentname*, and *sid* parameter that are available only for the Oracle component
- the *componentname, agentless*, *mode*, *os*, and *remoteagent* parameters that are relevant for only the MS SQL server being modified

From the above specification, it is also evident that columns not applicable to a component specification have been left blank.

If say, you want to add multiple SIDs to an Oracle database server, your specification should be as follows:

```
Element,action
Component,modify
componenttype,hostip,componentname,port,sid,externalagents
Oracle database,192.168.10.8,ora08,1521,"egora,egoracle",ext125
```

Note:

If an Oracle database server with multiple SIDs is added to the eG Enterprise system, then each SID will be registered as a separate Oracle database server in the eG Enterprise system.

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the CSV file:

eGCLI ModifyComponent -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

eGCLI ModifyComponent -managerID mgr153 -file c:\modcomp.csv

3.4 Unmanaging Components in Bulk

3.4.1 Unmanaging Components Using an XML File

Entries of the following format should be included in the XML file that is created exclusively for unmanaging multiple components of different types from the eG Enterprise system:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="unmanage">
   <Agent>
         <Component1 to be unmanaged>
   </Agent>
   <Agent>
         <Component2 to be unmanaged>
   </Agent>
                .
                .
                .
                .
</Component>
```

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="unmanage">
          <Agent>
                <componenttype>Generic</componenttype>
                <componentname>gen1</componentname>
          </Agent>
          <Agent>
                <componenttype>AGate</componenttype>
                <componentname>agte10</componentname>
                <port>3900</port>
          </Agent>
          <Agent>
                <componenttype>Oracle Database</componenttype>
                <componentname>ora8</componentname>
                <port>1521</port>
                <sid>egora</sid>
          </Agent>
   </Component>
```

Note that the XML tags are the same as the input parameters of the 'UnmanageComponent' command supported by eG CLI (see Section 2.4 of this document).

Note:

If an Oracle database server with multiple SIDs is be unmanaged, then do not provide a comma-separated list of SIDs within your *<sid></sid>* tag; instead, make sure that your XML file includes a separate *<Agent></Agent>* block for each SID to be unmanaged.

Note:

- The details of each unmanaged component should be included within the main *<Component></Component>* block only.
- A single XML file should not contain more than one *<Component></Component>* block.
- An XML file that is created for unmanaging components should not include sections pertaining to any other operation.
- A single XML file can be used for unmanaging components of different types.

Once the XML file is ready, execute the following command to extract the component information from the file, connect to the required eG manager, and unmanage the specified components:

eGCLI UnmanageComponent -managerID <ManagerID> -file <Full_path_to_the_XML_file>

For example:

eGCLI UnmanageComponent -managerID mgr153 -file c:\unmancomp.xml

3.4.2 Unmanaging Components Using a CSV File

The components to be unmanaged should be included in a CSV file that is created exclusively for this purpose.

Such a CSV file should contain entries of the following format:

```
Element,action
Component,unmanage
componenttype,componentname,port,sid
<Component1 to be unmanaged>
<Component2 to be unmanaged>
.
.
.
```

For example:

```
Element,action
Component,unmanage
componenttype,componentname,port,sid
Generic,gen1,,
AGate,agte10,3900,
Oracle Database,ora55,1521,multi
```

Note:

Components included in a zone, segment, or service cannot be unmanaged.

Note:

- If an Oracle database server with multiple SIDs is be managed, then the entry for the Oracle server in your CSV file should not include a comma-separated list of SIDs; instead, you should provide a separate entry for each SID to be managed.
- If one/more column names in your CSV file are not applicable to a component specification, then make sure that such columns are left empty in the specification.

Once the CSV file is ready, execute the following command to extract the component information from the file, connect to the required eG manager, and unmanage the specified components:

eGCLI UnmanageComponent -managerID <ManagerID> -file <Full_path_to_the_CSV_file>

For example:

eGCLI UnmanageComponent -managerID mgr153 -file c:\unmancomp.csv

3.5 Deleting Components in Bulk

3.5.1 Component Deletion Using an XML File

Entries of the following format should be included in the XML file that is created exclusively for deleting multiple components of different types from the eG Enterprise system:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="deletion">
   <Agent>
          <Component1 to be deleted>
   </Agent>
   <Agent>
          <Component2 to be deleted>
   </Agent>
                .
                .
                .
                .
</Component>
```

For example, to remove an Oracle and an Microsoft SQL database server from the eG Enterprise system, use the following specification:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="delete">
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.8</hostip>
          <componentname>oracle8</componentname>
          <port>1521</port>
          <sid>egora</sid>
   </Agent>
   <Agent>
          <componenttype>Microsoft SQL</componenttype>
          <hostip>192.168.10.9</hostip>
          <componentname>mssql</componentname>
          <port>1433</port>
   </Agent>
</Component>
```

If you wish to use the hostname of the Orace and Microsoft SQL database server from the eG Enterprise system instead of the IP address of the host, then use the following specification:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="delete">
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostname>egurkha25</hostname>
          <componentname>oracle8</componentname>
          <port>1521</port>
          <sid>egora</sid>
   </Agent>
   <Agent>
          <componenttype>Microsoft SQL</componenttype>
          <hostname>egurkha26</hostname>
          <componentname>mssql</componentname>
          <port>1433</port>
   </Agent>
</Component>
```

As already stated, if an Oracle database server is added with multiple SIDs, then the eG Enterprise system will monitor each SID as a different Oracle server. Therefore, while removing an Oracle database server that supports multiple SIDs, each SID should be treated as a different Oracle server, and a separate specification for each SID should be included in the XML file. For example, say, an Oracle database server has been added with the following SIDs: *egdemo,egora*. To remove this Oracle server completely, your XML file should contain the following entries:

```
<?xml version="1.0" encoding="UTF-8"?>
<Component action="delete">
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.8</hostip>
          <componentname>oracle8</componentname>
          <port>1521</port>
          <sid>egdemo</sid>
   </Agent>
   <Agent>
          <componenttype>Oracle database</componenttype>
          <hostip>192.168.10.8</hostip>
          <componentname>oracle8</componentname>
          <port>1521</port>
          <sid>egora</sid>
   </Agent>
</Component>
```

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to delete all the components specified in the XML file:

eGCLI DelComponent -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

eGCLI DelComponent -managerID mgr153 -file c:\delcomp.xml

Note:

- Each deletion entry should be included within the main *<Component></Component>* block only.
- A single XML file should not contain more than one *<Component></Component>* block.
- An XML file that is created for deleting component details should not include sections pertaining to any other operation.
- A single XML file can be used for deleting components of different types.
- Components included in a zone, segment, or service cannot be deleted.

3.5.2 Component Deletion Using a CSV File

The components to be deleted simultaneously from the eG Enterprise system should be included in a CSV file that is created exclusively for this purpose.

Such a CSV file should contain entries of the following format:

```
Element,action
Component,delete
componenttype,hostip/hostname,componentname,port
<Component1 to be deleted>
<Component2 to be deleted>
.
.
.
```

For example, if you want to delete an Oracle and an MS SQL server together using their respective host IPs, then your CSV file should include the following entries:

```
Element,action
Component,delete
componenttype,hostip,componentname,port,sid
Oracle database,192.168.10.96,ora96,1521,egora
Microsoft SQL,192.168.10.173,sql173,1433,
```

If you want to delete an Oracle and an MS SQL server together using their respective host names, then your CSV file should include the following entries:

```
Element,action
Component,delete
componenttype,hostname,componentname,port,sid
Oracle database,egurkha25,ora96,1521,egora
Microsoft SQL,egurkha26,sql173,1433,
```

As already stated, if an Oracle database server is added with multiple SIDs, then the eG Enterprise system will monitor each SID as a different Oracle server. Therefore, while removing an Oracle database server that supports multiple SIDs, each SID should be treated as a different Oracle server, and a separate specification for each SID should be included in the CSV file. For example, say, an Oracle database server has been added with the following SIDs: *egdemo,egora*. To remove this Oracle server completely, your CSV file should contain the following entries:

```
Element,action
Component,delete
componenttype,hostip,componentname,port,sid
Oracle database,192.168.10.96,ora96,1521,egora
Oracle database,192.168.10.96,ora96,1521,egdemo
```

Note:

Components included in a zone, segment, or service cannot be deleted.

3.6 Adding External Agents in Bulk

3.6.1 External Agent Addition Using an XML File

The XML file created for the sole purpose of adding multiple external agents to the eG Enterprise system at one go, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <ExternalAgent action="add">
          <Agent>
          <Details of External Agent1>
          </Agent>
          <Agent>
```

```
          <Details of External Agent2>
          </Agent>
                .
                .
   </ExternalAgent>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
   <ExternalAgent action="add">
          <Agent>
          <hostip>192.168.10.8</hostip>
          <agentname>ext8</agentname>
          </Agent>
          <Agent>
          <hostip>192.168.10.9</hostip>
          <agentname>ext9</agentname>
          </Agent>
   </ExternalAgent>
```

In the above example, the external agents are added using the IP address of the host. If you wish to use the host name instead of the *hostip* parameter, then your specification should be:

```
<?xml version="1.0" encoding="UTF-8"?>
   <ExternalAgent action="add">
          <Agent>
          <hostame>egurkha25</hostname>
          <agentname>ext8</agentname>
          </Agent>
          <Agent>
          <hostname>egurkha26</hostname>
          <agentname>ext9</agentname>
          </Agent>
   </ExternalAgent>
```

Note that the XML tags are the same as the input parameters of the 'AddExtenalAgent' command supported by eG CLI (see Section 2.6 of this document)

If the eG license enables the client emulation capability, the external agent specifications will include an additional *<clientemulation>* tag, as shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
   <ExternalAgent action="add">
          <Agent>
          <hostip>192.168.10.8</hostip>
          <agentname>ext8</agentname>
          <clientemulation>yes</clientemulation>
          </Agent>
          <Agent>
          <hostip>192.168.10.9</hostip>
          <agentname>ext9</agentname>
          </Agent>
   </ExternalAgent>
```

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to add multiple external agents to the eG Enterprise system:

eGCLI AddExternalAgent -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

eGCLI AddExternalAgent -managerID mgr153 -file c:\addextagent.xml

Note:

- Every external agent specification should be included within the main *<ExternalAgent></ExternalAgent>* block only.
- A single XML file should not contain more than one *<ExternalAgent></ExternalAgent>* block.
- An XML file that is created for adding external agents should not include sections pertaining to any other operation.

3.6.2 External Agent Addition Using a CSV File

The CSV file created specifically for adding multiple external agents to the eG Enterprise system, should contain the following entries:

```
Element,action
ExternalAgent,add
Hostip/hostname,agentname
<Details of extenalagent1>
<Details of extenalagent2>
<Details of extenalagent3>
.
.
.
.
```

For instance:

```
Element,action
ExternalAgent,add
hostip,agentname
192.168.10.8,ext8
192.168.10.10,ext10
192.168.10.12,lin12
```

If you use the host name instead of hostip, then your specification should be:

```
Element,action
ExternalAgent,add
hostname,agentname
egurkha25,ext8
egurkha26,ext10
egurkha27,lin12
```

Note that the column names (hostip,hostname,etc.) used here are the same as the input parameters of the 'AddExternalAgent' command supported by eG CLI (see Section 2.6 of this document).

If the eG license enables the client emulation capability, then the CSV file should include an additional *clientemulation* column. Therefore, if you want to add two external agents - one to be used for client emulation and another that is not used for client emulation - then, your CSV specification should be as follows:

```
Element,action
ExternalAgent,add
```

```
hostip,agentname,clientemulation
192.168.10.8,ext8,yes
192.168.10.10,ext10,no
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to add multiple external agents to the eG Enterprise system:

eGCLI AddExternalAgent -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

eGCLI AddExternalAgent -managerID mgr153 -file c:\addextagent.csv

3.7 Deleting External Agents in Bulk

3.7.1 External Agent Deletion Using an XML File

The XML file that is specifically created for deleting a number of external agents in bulk, should contain the following entries:

```
<?xml version="1.0" encoding="UTF-8"?>
   <ExternalAgent action="delete">
          <Agent>
          <Nick name of External Agent1>
          </Agent>
          <Agent>
          <Nick name of External Agent2>
          </Agent>
                .
                .
                .
   </ExternalAgent>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
   <ExternalAgent action="delete">
          <Agent>
          <agentname>ext8</agentname>
          </Agent>
          <Agent>
          <agentname>ext10</agentname>
          </Agent>
   </ExternalAgent>
```

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to delete multiple external agents from the eG Enterprise system:

eGCLI DelExternalAgent -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

eGCLI DelExternalAgent -managerID mgr153 -file c:\delextagent.xml

3.7.2 External Agent Deletion Using a CSV File

The CSV file that is specifically created for deleting a number of external agents in bulk, should contain the following entries:

```
Element,action
ExternalAgent,delete
agentname
<Nickname of external agent1>
<Nickname of external agent2>
<Nickname of external agent3>
```

For instance:

```
Element,action
ExternalAgent,delete
agentname
ext8
ext10
ext12
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to remove multiple external agents from the eG Enterprise system:

eGCLI DelExternalAgent -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

eGCLI DelExternalAgent -managerID mgr153 -file c:\remextagent.csv

3.8 Adding Remote Agents in Bulk

3.8.1 Remote Agent Addition Using an XML File

The XML file created for the sole purpose of adding multiple remote agents to the eG Enterprise system at one go, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <RemoteAgent action="add">
          <Agent>
          <Details of Remote Agent1>
          </Agent>
          <Agent>
          <Details of Remote Agent2>
          </Agent>
                .
                .
                .
   </RemoteAgent>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
   <RemoteAgent action="add">
          <Agent>
          <hostip>192.168.10.8</hostip>
          <agentname>ext8</agentname>
```

```
          </Agent>
          <Agent>
          <hostip>192.168.10.9</hostip>
          <agentname>ext9</agentname>
          </Agent>
   </RemoteAgent>
```

If you wish to add a remote agent using the host name instead of the host IP, then your specification should be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
   <RemoteAgent action="add">
          <Agent>
          <hostname>egurkha25</hostname>
          <agentname>ext8</agentname>
          </Agent>
          <Agent>
          <hostname>egurkha26</hostname>
          <agentname>ext9</agentname>
          </Agent>
   </RemoteAgent>
```

Note that the XML tags are the same as the input parameters of the 'AddRemoteAgent' command supported by eG CLI (see Section 2.12 of this document).

Note:

- Every Group specification should be included within the main *<RemoteAgent></RemoteAgent>* block only.
- A single XML file should not contain more than one *<RemoteAgent></RemoteAgent>* block.
- An XML file that is created for adding groups should not include sections pertaining to any other operation.

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to add multiple remote agents to the eG Enterprise system:

eGCLI AddRemoteAgent -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

eGCLI AddRemoteAgent -managerID mgr153 -file c:\addremagent.xml

3.8.2 Remote Agent Addition Using a CSV File

The CSV file created specifically for adding multiple remote agents to the eG Enterprise system, should contain the following entries:

```
Element,action
RemoteAgent,add
hostip,agentname
<Details of remoteagent1>
<Details of remoteagent2>
<Details of remoteagent3>
.
.
```

```
.
.
```

For instance:

```
Element,action
RemoteAgent,add
hostip,agentname
192.168.10.8,rem8
192.168.10.10,rem10
192.168.10.12,lin12
```

If you use the host name instead of hostip, then your specification should be:

```
Element,action
RemoteAgent,add
hostname,agentname
egurkha25,rem8
egurkha26,rem10
egurkha27,lin12
```

Note that the column names (hostip,hostname) used here are the same as the input parameters of the 'AddemoteAgent' command supported by eG CLI (see Section 2.12 of this document).

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to add multiple remote agents to the eG Enterprise system:

eGCLI AddRemoteAgent -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

eGCLI AddRemoteAgent -managerID mgr153 -file c:\addremagent.csv

3.9 Deleting Remote Agents in Bulk

3.9.1 Remote Agent Deletion Using an XML File

The XML file that is specifically created for deleting a number of remote agents in bulk, should contain the following entries:

```
<?xml version="1.0" encoding="UTF-8"?>
   <RemoteAgent action="delete">
          <Agent>
          <Nick name of Remote Agent1>
          </Agent>
          <Agent>
          <Nick name of Remote Agent2>
          </Agent>
                .
                .
                .
   </RemoteAgent>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
   <RemoteAgent action="delete">
          <Agent>
          <agentname>rem8</agentname>
```

```
        </Agent>
        <Agent>
        <agentname>rem10</agentname>
        </Agent>
   </RemoteAgent>
```

After creating the XML file on the orchestrator host, invoke the following command from the automation tool to extract the remote agent details from the XML file and add these remote agents to the eG Enterprise system:

eGCLI DelRemoteAgent -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

eGCLI DelRemoteAgent -managerID mgr153 -file c:\delremagent.xml

3.9.2 Remote Agent Deletion Using a CSV File

The CSV file that is specifically created for deleting a number of remote agents in bulk, should contain the following entries:

```
Element,action
RemoteAgent,delete
agentname
<Nickname of remote agent1>
<Nickname of remote agent2>
<Nickname of remote agent3>
```

For instance:

```
Element,action
RemoteAgent,delete
agentname
rem8
rem10
ext12
```

After creating the CSV file on the orchestrator host, invoke the following command from the automation tool to extract the remote agent details from the CSV file and add these remote agents to the eG Enterprise system:

eGCLI DelRemoteAgent -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

eGCLI DelRemoteAgent -managerID mgr153 -file c:\delremagent.csv

3.10 Adding Groups in Bulk

3.10.1 Adding Groups Using an XML File

The XML file created specifically for adding multiple groups to the eG Enterprise system, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Group action="add">
```

```
          <Element>
          <elements><comma-separated list of elements></elements>
          <groupname><Name of the group></groupname>
          </Element>
          <Element>
          <elements><comma-separated list of elements></elements>
          <groupname><Name of the group></groupname>
          </Element>
                .
                .
                .
   </Group>
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Group action="add">
          <Element>
          <elements>2x client gateway:client1:80,2x client
gateway:client2:80,2x Publishing Agent:agent1:20002</elements>
          <groupname>Group1</groupname>
          </Element>
          <Element>
          <elements>Microsoft Windows:wincli1,Microsoft Windows:wincli2
Microsoft Windows:wincli3</elements>
          <groupname>Group2</groupname>
          </Element>
          <Element>
          <elements>2x Publishing Agent:agent1:20002,2x
Agent:agent2:20002,2x client gateway:client3:80</elements>
          <groupname>Group3</groupname>
          </Element>
     </Group>
```

Note that the XML tags are the same as the input parameters of the 'AddGroup' command supported by eG CLI (see Section 2.36 of this document)

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to add multiple groups to the eG Enterprise system:

egcli addgroup -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example

egcli addgroup -managerid mgr153 -file E:\bulkgroupadd.xml

Note:

- Every Group specification should be included within the main *<Group></Group>* block only.
- A single XML file should not contain more than one *<Group></Group>* block.
- An XML file that is created for adding groups should not include sections pertaining to any other operation.

3.10.2 Adding Groups Using a CSV File

The CSV file created specifically for adding multiple groups to the eG Enterprise system, should contain the following entries:

```
Element,action
Group,add
groupname,elements
group1,"<comma-separated list of elements>"
group2,"<comma-separated list of elements>"
group3,"<comma-separated list of elements>"
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
Element,action
Group,add
groupname,elements
group1,"2X Client Gateway:client1:80,2X Client Gateway:client2:80,2X Client
Gateway:client3:80"
group2,"2X Publishing Agent:agent1:20002,2X Publishing Agent:agent2:20002,2X
Publishing Agent:agent3:20002"
group3,"Microsoft Windows:wincli1,Microsoft Windows:wincli2,Microsoft
Windows:wincli3"
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to add multiple groups to the eG Enterprise system:

egcli addgroup -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

egcli addgroup -managerid mgr153 -file E:\bulkgroupadd.csv

3.11 Modifying Groups in Bulk

3.11.1 Modifying Groups Using an XML File

An XML file is created exclusively for modifying multiple groups. The main purpose of modifying the groups is to associate and disassociate elements available in a group. In order to associate the elements in the Group, entries in this file should be of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Group action="modify">
          <Element>
          <associateelements><comma-separated list of
elements></associateelements>
          <groupname><Name of the group></groupname>
          </Element>
                .
                .
                .
   </Group>
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
<Group action="modify">
   <Element>
    <associateelements>2x client gateway:client4:80,2x client
gateway:client5:80</associateelements>
   <groupname>Group1</groupname>
   </Element>
</Group>
```

If you wish to disassociate the elements in the Group, entries in this file should be of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Group action="modify">
          <Element>
          <disassociateelements><comma-separated list of
elements></disassociateelements>
          <groupname><Name of the group></groupname>
          </Element>
                .
                .
                .
   </Group>
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
<Group action="modify">
   <Element>
    <disassociateelements>2x client gateway:client3:80,2x Publishing
Agent:agent1:20002</disassociateelements>
          <groupname>group2</groupname>
   </Element>
</Group>
```

Note that the XML tags are the same as the input parameters of the 'ModifyGroup' command supported by eG CLI (see Section 2.37 of this document)

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the XML file:

egcli modifyGroup -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

egcli modifyGroup -managerid mgr153 -file E:\bulkmodifygroup.xml

Note:

- Each modification should be included within the main *<Group></Group>* block only.
- A single XML file should not contain more than one *<Group></Group>* block.
- An XML file that is created for modifying group details should not include sections pertaining to any other operation.
- A single XML file can be used for modifying the details of groups of different types.

3.11.2 Modifying Groups Using a CSV File

As already mentioned, an exclusive CSV file will have to be created to handle bulk modifications to group details.

In order to associate the elements in the Group, entries in this file should be of the following format:

```
Element,action
Group,modify
groupname,associateelements
groupname,"<comma-separated list of elements>"
.
.
.
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
Element,action
Group,modify
groupname,associateelements
group1,"2x client gateway:client1:80,2x client gateway:client2:80"
group2,2x Publishing Agent:agent1:20002
group3,"Microsoft Windows:wincli1,Microsoft Windows:wincli2"
```

If you wish to disassociate the elements in the Group, entries in this file should be of the following format:

```
Element,action
Group,modify
groupname,disassociateelements
groupname,"<comma-separated list of elements>"
.
.
.
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
Element,action
Group,modify
groupname,disassociateelements
```

```
group1,"2x client gateway:client1:80,2x client gateway:client2:80"
group2,2x Publishing Agent:agent1:20002
group3,"Microsoft Windows:wincli1,Microsoft Windows:wincli2"
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the CSV file:

egcli modifyGroup -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

egcli modifyGroup -managerid mgr153 -file E:\bulkgroupmodify.csv

3.12 Renaming Groups in Bulk

3.12.1 Renaming Groups Using an XML File

The XML file created specifically for renaming multiple groups to the eG Enterprise system, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Group action="rename">
          <Element>
          <groupname><Name of the oldgroup></groupname>
          <newgroupname><Name of the newgroup></newgroupname>
          </Element>
                .
                .
                .
   </Group>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
<Group action="rename">
   <Element>
          <groupname>group3new</groupname>
          <newgroupname>group1</newgroupname>
   </Element>
   <Element>
          <groupname>group2new</groupname>
          <newgroupname>group2</newgroupname>
   </Element>
</Group>
```

Note that the XML tags are the same as the input parameters of the 'RenameGroup' command supported by eG CLI (see Section 2.38 of this document)

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the XML file:

egcli renamegroup -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

egcli renamegroup -managerid mgr153 -file E:\bulkrenamegroup.xml

.

Note:

- Every Group specification should be included within the main *<Group></Group>* block only.
- A single XML file should not contain more than one *<Group></Group>* block.
- An XML file that is created for renaming groups should not include sections pertaining to any other operation.

3.12.2 Renaming Groups Using a CSV File

The CSV file created specifically for renaming multiple groups to the eG Enterprise system, should contain the following entries:

```
Element,action
Group,rename
groupname,newgroupname
<Names of the old and new Groups>
.
.
.
```

For instance:

```
Element,action
Group,rename
groupname,newgroupname
group1,group1new
group2,group2new
group3,group3new
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the CSV file:

egcli renamegroup -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

egcli renamegroup -managerid mgr153 -file E:\bulkgrouprename.csv

3.13 Deleting Groups in Bulk

3.13.1 Deleting Groups Using an XML File

The XML file created specifically for deleting multiple groups to the eG Enterprise system, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Group action="delete">
         <Element>
         <groupname><comma-separated list of groups></groupname>
         </Element>
   </Group>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
<Group action="delete">
   <Element>
          <groupname>group1,group2,group3</groupname>
   </Element>
</Group>
```

Note that the XML tags are the same as the input parameters of the 'DeleteGroup' command supported by eG CLI (see Section 2.39 of this document)

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the XML file:

egcli deletegroup -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

egcli deletegroup -managerid mgr153 -file E:\bulkdeletegroup.xml

Note:

- Every Group specification should be included within the main *<Group></Group>* block only.
- A single XML file should not contain more than one *<Group></Group>* block.
- An XML file that is created for deleting groups should not include sections pertaining to any other operation.

3.13.2 Deleting Groups Using a CSV File

The CSV file created specifically for deleting multiple groups to the eG Enterprise system, should contain the following entries:

```
Element,action
Group,delete
groupname
<Names of the Groups>
.
.
.
```

For instance:

```
Element,action
Group,delete
groupname
group1
group2
group3
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the CSV file:

egcli deletegroup -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

egcli deletegroup -managerid mgr153 -file E:\bulkdeletegroup.csv

3.14 Adding Zones in Bulk

3.14.1 Adding Zones Using an XML File

The XML file created specifically for adding multiple Zones to the eG Enterprise system, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Zone action="add">
          <Element>
          <elements><comma-separated list of elements></elements>
          <zonename><Name of the zone></zonename>
          </Element>
          <Element>
          <elements><comma-separated list of elements></elements>
          <zonename><Name of the zone></zonename>
          </Element>
                .
                .
                .
   </Zone>
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zone action="add">
   <Element>
          <elements>2x client gateway:client1:80,2x client
gateway:client2:80,2x Publishing Agent:agent1:20002</elements>
          <zonename>zone1</zonename>
   </Element>
   <Element>
          <elements>Group:group1</elements>
          <zonename>zone2</zonename>
   </Element>
   <Element>
          <elements>Service:service1</elements>
          <zonename>zone3</zonename>
   </Element>
</Zone>
```

Note that the XML tags are the same as the input parameters of the 'AddZone' command supported by eG CLI (see Section 2.32 of this document)

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to add multiple zones to the eG Enterprise system:

egcli addZone -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example

egcli addzone -managerid mgr153 -file E:\bulkgroupadd.xml

Note:

- Every Zone specification should be included within the main *<Zone></Zone>* block only.
- A single XML file should not contain more than one *<Zone></Zone>* block.
- An XML file that is created for adding Zones should not include sections pertaining to any other operation.

3.14.2 Adding Zones Using a CSV File

The CSV file created specifically for adding multiple Zones to the eG Enterprise system, should contain the following entries:

```
Element,action
Zone,add
zonename,elements
zone1,"<comma-separated list of elements>"
zone2,"<comma-separated list of elements>"
zone3,"<comma-separated list of elements>".
.
.
.
.
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
Element,action
Zone,add
zonename,elements
zone1,"2x client gateway:client1:80,2x client gateway:client2:80,2x client
gateway:client3:80"
zone2,Group:group1
zone3,Segment:segment1
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to add multiple zones to the eG Enterprise system:

egcli addzone -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

egcli addzone -managerid mgr153 -file E:\bulkzoneadd.csv

3.15 Modifying Zones in Bulk

3.15.1 Modifying Zones Using an XML File

An XML file is created exclusively for modifying multiple Zone specifications. In order to associate the elements in the Zone, entries in this file should be of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Zone action="modify">
          <Element>
          <associateelements><comma-separated list of
elements></associateelements>
          <zonename><Name of the zone></zonename>
          </Element>
                .
                .
                .
   </Zone>
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zone action="modify">
   <Element>
          <associateelements>2x client gateway:client3:80,2x client
gateway:client4:80</associateelements>
          <zonename>zone1</zonename>
   </Element>
   <Element>
          <associateelements>Segment:segment1</associateelements>
          <zonename>zone2</zonename>
   </Element>
</Zone>
```

If you wish to disassociate the elements in the Zone, entries in this file should be of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Zone action="modify">
          <Element>
          <disassociateelements><comma-separated list of
elements></disassociateelements>
          <zonename><Name of the zone></zonename>
          </Element>

                .
                .
                .
   </Zone>
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For instance:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zone action="modify">
   <Element>
          <disassociateelements>Service:service1</disassociateelements>
          <zonename>zone3</zonename>
   </Element>
</Zone>
```

Note that the XML tags are the same as the input parameters of the 'ModifyZone' command supported by eG CLI (see Section 2.33 of this document)

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the XML file:

egcli modifyzone -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

egcli modifyzone -managerid mgr153 -file E:\bulkmodifyZone.xml

Note:

- Every Zone specification should be included within the main *<Zone></Zone>* block only.
- A single XML file should not contain more than one *<Zone></Zone>*block.
- An XML file that is created for modifying Zone details should not include sections pertaining to any other operation.
- A single XML file can be used for modifying the details of Zones of different types.

3.15.2 Modifying Zones Using a CSV File

As already mentioned, an exclusive CSV file will have to be created to handle bulk modifications to Zone details.

In order to associate the elements in the Zone, entries in this file should be of the following format:

```
Element,action
Zone,modify
zonename,associateelements
zonename,"<comma-separated list of elements>"
.
.
.
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For example:

```
Element,action
Zone,modify
zonename,associateelements
zonedemo,"2x client gateway:client3:80,2x Publishing
Agent:agent1:20002,Microsoft Windows:wincli1"
zonedemo1,"Group:group1,Group:group1new"
```

If you wish to disassociate the elements in the Zone, entries in this file should be of the following format:

```
Element,action
Zone,modify
zonename,disassociateelements
zonename,"<comma-separated list of elements>"
.
.
```

```
.
```

Note that the elements should be specified in the format:<Component Type>:<component name>:<Port>

For example:

```
Element,action
Zone,modify
zonename,disassociateelements
zonedemo,"2x client gateway:client3:80,2x Publishing
Agent:agent1:20002,Microsoft Windows:wincli1"
zonedemo1,"Group:group1,Group:group1new"
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the CSV file:

egcli modifyzone -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

egcli modifyzone -managerid mgr153 -file E:\bulkmodifyzone.csv

3.16 Renaming Zones in Bulk

3.16.1 Renaming Zones Using an XML File

The XML file created specifically for renaming multiple Zones to the eG Enterprise system, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Zone action="rename">
  <Element>
         <zonename><Name of the zone></zonename>
         <newzonename><Name of the newzone></newzonename>
  </Element>
               .
               .
               .
   </Zone>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zone action="rename">
   <Element>
         <zonename>zone3new</zonename>
         <newzonename>zone3</newzonename>
   </Element>
   <Element>
         <zonename>zone2new</zonename>
         <newzonename>zone2</newzonename>
   </Element>
</Zone>
```

Note that the XML tags are the same as the input parameters of the 'RenameZone' command supported by eG CLI (see Section 2.34 of this document)

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the XML file:

egcli renamezone -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

egcli renamezone -managerid mgr153 -file E:\bulkrenamezone.xml

Note:

- Every Zone specification should be included within the main *<Zone></Zone>* block only.
- A single XML file should not contain more than one *<Zone></Zone>* block.
- An XML file that is created for renaming Zones should not include sections pertaining to any other operation.

3.16.2 Renaming Zones Using a CSV File

The CSV file created specifically for renaming multiple Zones to the eG Enterprise system, should contain the following entries:

```
Element,action
Zone,rename
zonename,newzonename
Names of the old and new zones>
.
.
.
```

For instance:

```
Element,action
Zone,rename
zonename,newzonename
zone1,zone1new
zone2,zone2new
zone3,zone3new
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the CSV file:

egcli renamezone -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

egcli renamegroup -managerid mgr153 -file E:\bulkrenamezone.csv

3.17 Deleting Zones in Bulk

3.17.1 Deleting Zones Using an XML File

The XML file created specifically for deleting multiple Zones to the eG Enterprise system, should contain entries of the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Zone action="delete">
          <Element>
          <zonename><comma-separated list of zones></zonename>
          </Element>
   </Zone>
```

For instance:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zone action="delete">
   <Element>
          <zonename>zone1,zone2</zonename>
   </Element>
</Zone>
```

Note that the XML tags are the same as the input parameters of the 'DeleteZone' command supported by eG CLI (see Section 2.35 of this document)

Once the XML file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the XML file:

egcli deletezone -managerid <ManagerID> -file <Fullpath_to_the_XML_file>

For example:

egcli deletezone -managerid mgr153 -file E:\ bulkdeletezone.xml

Note:

- Every Zone specification should be included within the main *<Zone></Zone>* block only.
- A single XML file should not contain more than one *<Zone></Zone>* block.
- An XML file that is created for deleting Zones should not include sections pertaining to any other operation.

3.17.2 Deleting Zones Using a CSV File

The CSV file created specifically for deleting multiple Zones to the eG Enterprise system, should contain the following entries:

```
Element,action
Zone,delete
zonename
<Names of the zones>
.
.
```

```
.
```

For instance:

```
Element,action
Zone,delete
zonename
zone1
zone2
zone3
```

Once the CSV file is created on the orchestrator's host, invoke the following command from the orchestrator to update the eG manager with all the modifications contained in the CSV file:

egcli deletezone -managerid <ManagerID> -file <Fullpath_to_the_CSV_file>

For example:

egcli deletezone -managerid mgr153 -file E:\ bulkzonedelete.csv

3.18 Logging of Bulk Commands

When operations that are performed in bulk via the command line fail, an error message indicating which specific operations failed will appear at the command prompt. For instance, while adding ten components using an XML file, if the addition of five components fail, then the resulting error message will clearly indicate which component of which type could not be added. However, in order to troubleshoot a failure, it is not enough if you just know which operations failed; you would also require the knowledge of why they failed. To provide this information, the eG CLI automatically logs bulk operation failures to the **<EGADMINCLI_INSTALL_DIR>\logs\<MANAGERID>Log0.log** file. For instance, if one/more bulk operations performed on the eG manager **mgr147** fail, then the eG CLI will log the details of these failures to the **<EGADMINCLI_INSTALL_DIR>\logs\mgr147Log0.log** file. Administrators can use this log file to understand the reason for the failure of bulk operations, and then accordingly take the appropriate corrective steps.

By default, the **<MANAGERID>Log0.log** file will be allowed to grow upto a maximum size of 2MB. Beyond this limit, the contents of this log file will be copied to a **<MANAGERID>Log1.log** file, and the newer information will be logged in the original **<MANAGERID>Log0.log** file. This way, a maximum of 2 log files will be created, with the last log file being **<MANAGERID>Log1.log** file. This log rotation mechanism helps ensure that the log file does not grow beyond control.

Chapter 5

Conclusions

This document clearly explained the current capabilities of the eG CLI. Please note that these capabilities are not sacrosanct and are liable to change based on customer feedback.