

Infrastructure management using the HP SIM command line interface



Introduction	2
Inventory management	2
Configuring HP SIM	2
Running discovery	4
Running reports	6
Non-HP device management	8
Managing collections	9
Event Management	14
Configuring HP SIM	14
WBEM	14
SNMP	15
Managing HP SIM tasks	16
Managed server setup	16
Managing tasks	16
Managing task execution	19
Managing users and authorizations	20
Managing users	21
Managing system groups	22
Managing toolboxes	23
Adding tools to toolboxes	24
Managing authorizations	25
HP SIM configuration	25
HP SIM initialization	26
HP SIM services	26
HP SIM log	26
HP SIM information	27
HP SIM settings	28
HP SIM Server certificate	28
HP SIM passwords	29
Appendix	30

Introduction

HP Systems Insight Manager (HP SIM) is a robust, full-featured infrastructure management tool. Users have the option of accessing HP SIM functionality through the web-based GUI, or through the set of command line interfaces (CLI) provided. This paper discusses how to use the HP SIM CLI for managing your infrastructure.

CLI use is described in terms of the CLI(s) required to perform specific tasks, like inventory management, event management, etc. A list of all CLIs is provided in the Appendix.

Detailed man pages are available from the command line.

HP-UX

- CLI command line usage man pages are specified as section 1M
- CLI Extensible Markup Language (XML) usage man pages are specified as section 4

For example:

```
% man mxtask           ; can also be man 1m mxtask
% man 4 mxtask
```

Linux

- CLI command line usage man pages are specified as section 8
- CLI XML usage man pages are specified as section 4

For example:

```
# man mxtask           ; can also be man 8 mxtask
# man 4 mxtask
```

Windows

The man pages are located in the following directory:

```
Program Files\HP\System Insight
Manager\hpwebadmin\webapps\mxhelp\mxportal\en\man
```

Double-click a man page file to view its contents in a web browser.

Inventory management

Configuring HP SIM

Before adding systems to HP SIM, protocols used to talk to the managed systems must be setup and configured. The command **mxglobalprotocolsettings** enables you to set most of the protocol settings available in HP SIM. The command **mxnodesecurity** enables you to set WBEM and SNMP credentials on a per-system basis. Both commands accept XML formatted data, either from a file or from the command line, and sets the global protocol settings accordingly. These commands must be run before running discovery in order for protocol authentication on the managed systems to be accepted.

See the man page **mxglobalprotocolsettings** and **mxnodesecurity** for details on using the CLI. See the man page **mxglobalprotocolsettings(4)** and **mxnodesecurity(4)** for XML formatting details.

To change a specific global protocol setting, use the following command:

```
% mxglobalprotocolsettings -s enable-snmp=true
```

The **enable-snmp** setting is one of many available. The available settings are documented in the **mxglobalprotocolsettings(4)** man page.

A convenient way to change multiple protocol settings is to use an XML file. For example:

```
% mxglobalprotocolsettings -s -f globalprotocol.xml
```

The globalprotocol.xml file might look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<global-protocol-settings xmlns="http://www.hp.com/hpsim5.0.0.0">
  <use-icmp-for-ping value="true" />
  <default-ping-timeout-seconds value="5" />
  <default-ping-retries value="2" />
  <enable-wbem value="true" />
  <enable-http-and-https value="true" />
  <enable-snmp value="true" />
  <default-snmp-timeout-seconds value="5" />
  <default-snmp-retries value="1" />
  <enable-dmi value="false" />
</global-protocol-settings>
```

Run the following command to view the current global protocol settings in XML format:

```
% mxglobalprotocolsettings -lf
```

Alternatively, use the **-ld** option to have the settings displayed in a text format:

```
% mxglobalprotocolsettings -ld
```

The command **mxnodesecurity** enables you to set and view WBEM and SNMP credentials on per-system basis. Both options accept XML formatted data, either from a file or from the command line, and sets the protocol settings accordingly.

Run the following commands to add a credential for WBEM and SNMP for the system **nodeName1**:

```
% mxnodesecurity -a -p wbem -c root:password -n nodeName1 ; WBEM
% mxnodesecurity -a -p snmp -c public:private -n nodeName1 ; SNMP
```

The **-p** and **-c** parameters must be provided. The **-n** parameter is optional. If provided, the credential is for the selected system only. If not provided, the credential is the default credentials used for WBEM in the global protocol settings.

The **mxnodesecurity** command can also use an XML file to add the credentials for WBEM and SNMP for a list of systems. For example,

```
% mxnodesecurity -a -f credential.xml
```

The `credential.xml` file might contain the following:

```
<?xml version="1.0" encoding="UTF-8" ?>
<nodelist>
  <node name="nodeName1">
    <credential protocol="wbem" username="root"
      password="password" />
    <credential protocol="snmp" username="public"
      password="private" />
  </node>
  <node name="nodeName2">
    <credential protocol="wbem" username="root"
      password="password" />
    <credential protocol="snmp" username="public"
      password="private" />
  </node>
</nodelist>
```

mxnodesecurity is very useful when there are more credentials in use than **mxglobalprotocolsettings** allows, and in environments where each system has unique credentials.

Run the following command to remove the WBEM or SNMP credentials from a system:

```
% mxnodesecurity -r -p wbem -n nodeName1
% mxnodesecurity -r -p snmp -n nodeName1
```

Run the following command to remove credentials from a list of system:

```
% mxnodesecurity -r -f credential.xml
```

Running discovery

The command **mxnode** enables you to add, delete, modify, identify, or list systems in HP SIM. Before using **mxnode**, global protocol settings must be configured, and access credentials set, as described above. After systems are added, identification and data collection run automatically.

Systems can be added using its host name or its IP address, and can be provided on the command line or in an XML file. See the man page **mxnode** for command usage details, and the man page **mxnode(4)** for XML file formatting details.

The **mxnode** command provides two methods to add a system:

```
% mxnode -a nodeName1 ipAddr1
% mxnode -a -f nodes.xml
```

The first example shows adding two systems, one by its system name, and one by its IP address. The second example uses an XML file, `nodes.xml`, to add all of the systems in the file. After the systems are added, identification is run on them. The following is an example of what `nodes.xml` might contain:

```
<?xml version="1.0" encoding="UTF-8"?>
<node-list>
  <node name="nodeName1"/>
  <node name="nodeName2"/>
</node>
```

```
<sw-attribute name="IPAddress">192.1.1.1</sw-attribute>
</node>
</node-list>
```

The syntax to modify and re-identify systems is:

```
% mxnode -m nodeName1 nodeName2
% mxnode -m -f nodes.xml
```

The **-m** option only accepts system names, unlike the **-a** option that also accepts IP addresses.

To remove systems, use the **-r** option. The **-r** option accepts system names, host names, or IP addresses, or in conjunction with the **-f** option, the name of an XML file containing the names or IP addresses of the nodes to remove. For example:

```
% mxnode -r nodeName1 ipAddress1 nodeName2
% mxnode -r -f nodes.xml
```

There are several formats available for displaying the existing set of systems in the HP SIM database.

```
% mxnode -l n [systemnames]
% mxnode -l d [systemnames]
% mxnode -l t [systemnames]
% mxnode -l f [systemnames]
% mxnode -l mpo
```

The **systemnames** parameter is an optional list of one or more system names. If provided, the command displays the requested data just for the systems listed. Otherwise, the command displays the requested data for all systems known to HP SIM.

The **-ln** option displays the system names. The **-ld** option displays detailed information for each system in the database. The **-lt** option displays the system name, the host name, the OS name, the date the system was added to the database, and the IP address for the systems in a tabular format. The **-lf** option displays detailed information for the systems in XML format. Finally, the **-lmpo** option displays systems from the database that have any management path objects (MPO). The man page for **mxnode** contains a description of MPOs.

The **mxnode -lf** command can be used to create an XML file that allows easier management of your system list. Redirecting the output to a file creates a file in the proper XML format that is easy to modify when adding or removing systems from HP SIM.

To use the CLI to create an automatic discovery task, the simplest way is to copy the default **System Automatic Discovery** task, modify the task name, task schedule, and IP address range fields, and then create a new automatic discovery task. The command **mxtask** is used to accomplish these steps. Refer to the **mxtask** man page for a complete description of this command, and to the **mxtask(4)** man page for a full description of the XML syntax for creating tasks.

To create a copy of the **System Automatic Discovery** task, enter the following command:

```
% mxtask -lf "System Automatic Discovery" > autodisco.xml
```

This creates an XML description of the task in the file `autodisco.xml`. Open the file in a text editor, and modify the following fields:

<task name> - Change the name from **System Automatic Discovery** to a name you want.

<scheduleinfo> - Set the interval and time fields, defining when you want the task to run.

ipInclusionRange – Enter the IP address ranges you want discovered.

After making the changes, save the file. Now, create the new discovery task by entering the following command:

```
% mxtask -cf autodisco.xml
```

If you enabled periodic execution in the <scheduleinfo> field, the task automatically runs at the time and frequency provided. If you did not enable periodic execution, you must manually execute the task by entering the following command:

```
% mxtask -e TaskName
```

TaskName is the name of the task entered in the <task name> field of the XML file.

mxtask also has a build-in help facility that describes the command, parameters, and sample uses. Enter, **mxtask -h**, to see the available information.

Running reports

The **mxreport** command enables an HP SIM user with sufficient privilege to run reports, and to add, delete, and list reports and report categories in HP SIM. A complete description of the **mxreport** command is found in the **mxreport** man page. A complete description of the XML format used by **mxreport** is found in the **mxreport(4)** man page.

To create a report in HP SIM, first create an XML file describing the report – report name, collection to run the report against, and reporting fields. You can define multiple reports in the XML file. Here is a simple example of the XML file to create a report:

```
<?xml version="1.0" encoding="UTF-8"?>
<reportList>
  <report reportName="Inventory 7 - Servers" displayFlag="0"
    creatorId="1" editFlag="1">
    <group groupLocale="en_US" parent="System reports" child=""/>
    <displayName displayLocale="en_US">Inventory 7 -
      Servers</displayName>
    <query queryType="0" queryLocale="en_US">All Servers</query>
    <item viewColName="DeviceName" viewName="R_Inventory"/>
    <item viewColName="ProductType" viewName="R_Inventory"/>
    <item viewColName="ProductName" viewName="R_Inventory"/>
    <item viewColName="SerialNumber" viewName="R_Inventory"/>
    <item viewColName="AssetTag" viewName="R_Inventory"/>
    <item viewColName="MemorySize" viewName="R_Inventory"/>
    <item viewColName="OSName" viewName="R_Inventory"/>
    <item viewColName="OSVendor" viewName="R_Inventory"/>
    <item viewColName="Location" viewName="R_Inventory"/>
    <item viewColName="DeviceOwner" viewName="R_Inventory"/>
    <viewOption>0</viewOption>
    <privilege>0</privilege>
  </report>
</reportList>
```

This report will be called *Inventory 7 - Servers*, and will be run against the **All Servers** collection. If this file is called *inv7server.xml*, the command to add this report to HP SIM is,

```
% mxreport -a -f <path>inv7server.xml
```

where *<path>* is the fully qualified path to the file `inv7server.xml`.

If a report is defined in the XML file that has the name of an existing report in HP SIM, the duplicate report is not processed. **mxreport** does not have a modify option. Therefore, to modify a report, the existing report must first be deleted, and the modified report added. To delete an existing report:

```
% mxreport -d report-name
```

You can only delete a report that you created, or that you have permission to delete. You cannot delete system-defined reports and report categories. There are many predefined report categories in HP SIM. A report category contains a logical grouping of reportable items. The **mxreport** command enables you to define your own report categories to be used to report against. Run the following command to define your own report categories:

```
% mxreport -c -f categories.xml
```

where *categories.xml* contains the XML definitions of the categories. The man page **mxcategoryitem(4)** provides the details for creating the XML definitions for creating report categories.

To remove a report category, use the **-r** option:

```
% mxreport -r report-category-name
```

The following command is used to execute reports from the command line:

```
% mxreport [-x format] [-o report.out] -e report-name
```

The optional **-x** option is used to specify the output format, either **XML** (the default), **HTML**, or **CSV**. The optional **-o** option is used to specify a file, **report.out** in the example, where the report output will be saved. If **-o** is not provided, the report output will be displayed on the console.

The **mxreport** command also provides options to view report or report category information. The data is displayed on the console in XML format, so output redirection is required to save the data in a file.

The general format for viewing report or report category data is,

```
% mxreport -l -x [report | category] [-n r-or-c-name]
```

To see a list of all reports in HP SIM, enter either of these commands,

```
% mxreport -l -x  
% mxreport -l -x report
```

To see a list of all report categories in HP SIM,

```
% mxreport -l -x category
```

The **-n** option enables you to select a specific report, or specific report category, for display. The output is XML formatted data that includes details of the report or report category. To see details of a report, enter,

```
% mxreport -l -x report -n "report name"
```

Similarly, to see the details of a report category,

```
% mxreport -l -x category -n "report category name"
```

Non-HP device management

HP SIM can manage non-HP equipment if the equipment management agents and fault management mechanisms adhere to industry standards like SNMP and WBEM. HP SIM provides system definitions for dozens of non-HP systems. The System Type Manager enables you to modify existing definitions, add new definitions for equipment not known to HP SIM, and remove definitions.

The CLI to perform this task is **mxstm**. Through the command line, **mxstm** enables you to add, remove, and list system type definitions. Further details for this command are available in the **mxstm** man page.

To add a new system type definition, use one of the following commands:

```
% mxstm -a -n product-name -p protocol (-x attribute=value)+  
% mxstm -a -f stmfile.xml
```

The **-n product-name** option is used to specify the product name of the product being added. For example, HP rx1600 is a product name. The **-p protocol** selects the protocol to be used for data collection, and should be set to **SNMP**. The **-x** option provides the definitions for one or more *attributes* for the system. Examples of SNMP attributes are the system OID, product type, product subtype, OS name, and so on.

The second form of the add command enables you to provide the system type definitions in XML format. In the example above, the **-f stmfile.xml** command tells HP SIM to get the system type definitions from the file `stmfile.xml`. The file `/opt/mx/dtd/stmrules.dtd` provides a complete description of the XML format required.

To remove system type definitions, use one of the following commands:

```
% mxstm -r -n product-name  
% mxstm -r -f stmfile.xml
```

The first command removes the system type with the product name *product-name*. The second command removes all of the system types defined in the XML file `stmfile.xml`.

To see a listing of existing system type definitions, use one of the following commands:

```
% mxstm -l [-n product-name]  
% mxstm -l -f stmfile.xml
```

The first command lists all of the product names for all of the system type definitions in HP SIM. By using the **-n product-name** option the full definition of the product *product-name* is displayed. The product name, protocol, OID data, product type, vendor, and so on, are all displayed. The second command produces the full definition for every system type in HP SIM in XML format, saving the data in the file `stmfile.xml`.

Managing collections

HP SIM enables you to create groupings of systems or events called collections. Each collection is a query into the HP SIM database. HP SIM users can create collections that all HP SIM users can see, called shared collections, or collections that only that user can see, called private collections. Collections enable you to define a hierarchical organization of collections. For example, the following hierarchy exists in the predefined collections of HP SIM:

```
Systems
  Private
  Shared
    Systems by Operating System
      HP-UX
      Microsoft Windows Server 2003
```

The items **Systems**, **Shared**, and **Systems by Operating System** are collections that contain other collections as their members. The items **HP-UX** and **Microsoft Windows Server 2003** are also collections, but they also define a query that is performed on the HP SIM database. In this section, the term query collection will be used to refer to collections that define a query on the HP SIM database.

The command **mxcollection** enables you to add, modify, remove, or list collections, or members within a collection in HP SIM. A full description for **mxcollection** exists in the **mxcollection** man page, and a full description of the XML syntax required to manage users through an XML file exists in the **mxcollection(4)** man page. The command **mxquery** is used to add, modify, list, remove, and execute query collections. Full descriptions of the command and the XML format for query collections is available in the **mxquery** and **mxquery(4)** man pages, respectively.

Note: In earlier releases of HP SIM, categories could be created and manipulated with the **mxquery** command. Beginning with HP SIM 5.0, collections are an extension of categories, and are managed with the **mxcollection** command. **mxcollection** enables you to choose members individually by name. You must continue to use **mxquery** to the search attributes when creating and managing queries.

Query collections are powerful, but they can be complex. The **mxquery** command provides many options to help create and manage query collections. The following XML is an example of a query collection, called Test Query.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<query-list>
  <query id="64" name="Test Query" type="DeviceViews"
    category-name="SysSharedLists" owner="root">
    <criteria name="CriteriaByOSNameComparison"
      sense="INCLUDE">
      <value operator="contains">HP</value>
    </criteria>
    <criteria name="ProductByName" sense="INCLUDE">
      <value>ia64</value>
    </criteria>
    <criteria name="CriteriaByNameComparison" sense="INCLUDE">
      <value operator="contains">ABC</value>
    </criteria>
  </query>
</query-list>
```

This query finds all systems that have **HP** in the Operating System name and has **ia64** in the Product name, and has **ABC** in the System name. To get this output, use the following command,

```
% mxquery -lf "Test Query"
```

The **-lf** option displays the output in XML format for the specified query collection. If no query collection name is specified, XML for all collections and query collections is displayed. If a collection name is specified, an error is displayed saying no such query exists.

mxquery has the following options for displaying query information:

```
-l cat          list all collections where you can add a query collection
-l crit        list the criteria that can be used to construct a query collection
-l mode "crit" list the operators that can be used with criteria "crit"
-l val "crit"  list the defined values associated with criteria "crit"
```

Note: If the criteria type is Text, there will not be any associated value for the criteria.

From our example above, the output for the command **mxquery -l mode CriteriaByNameComparison** is:

```
does not start with
does not contain
contains
ends with
does not end with
is equal to
is not equal to
starts with
```

This is because **CriteriaByNameComparison** is a text criteria, and those operators apply to text fields in the database. Other criteria will be selection criteria, and their operators will be **is** and **is not**, since selection criteria will be specific values to be compared, not free-form text.

CriteriaByNameComparison is a text criteria and does not have any values. However, the criteria **ProductByName** is a selection criteria. To find the values this criteria can have, enter,

```
% mxquery -l val ProductByName
```

There are two additional options to the **mxquery -l** command. The **n** option displays a list of all the query collections in HP SIM. The **t** option displays all the query collections in a tabular format, showing the query ID, query name, query type, and query owner.

To create a new query collection in HP SIM, first create a file that contains the XML definition for the query collection. Using the example above, the XML would be put in a file that we will call `testquery.xml`. The following command adds that query collection to HP SIM,

```
% mxquery -a -f testquery.xml
```

You can have more than one query collection definition in the XML file. **mxquery** processes all of the queries that are defined. If a query definition has a problem, that definition is skipped, but that does not impact processing of the other definitions.

The XML field **category-name** defines where in the collection hierarchy the query will reside. For our example query, it will reside in the collection **SysSharedLists**, which is the collection Shared in the HP SIM collection hierarchy.

The XML file is also used to modify one or more query collections once they have been added to HP SIM, by using the command,

```
% mxquery -m -f testquery.xml
```

Only query collection definitions that have been modified in `testquery.xml` will be processed.

There are two ways to remove query collections from HP SIM, illustrated below.

```
% mxquery -r "Test Query"  
% mxquery -r -f testquery.xml
```

The first example removes the query collection **Test Query**. The second example removes all of the query collections defined in the file `testquery.xml`.

Once you have your query collections added to HP SIM, you can execute them from the command line by performing the following command.

```
% mxquery -e "Test Query"
```

The output is a list of information about each system that matches the query criteria. Since it is possible for many systems to match the query criteria, HP recommends that you redirect the output of the **-e** option to a file for further processing. The following example shows the output for one system from the example query, **Test Query**.

```
NoticeID: 602  
HWStatus: 1  
MPStatus: 0  
SWStatus: 12  
DeviceName: tincup  
DeviceType: 1  
DeviceAddress: 15.1.53.133  
ProductName: ia64  
OSName: HP-UX  
DeviceAddress:  
deviceKey: 591  
fullDNSName: tincup.fc.hp.com  
associatedDeviceKey:  
AssociatedType: 0  
AssociatedName:  
associatedDeviceKey: 0  
timeStamp: 1144308295806
```

The command **mxcollection** enables you to manage the HP SIM collections that help organize query collections. The following command displays all of the collections in a hierarchical tree,

```
% mxcollection -ln
```

The hierarchy shown at the beginning of this section is taken from the output of the above command. Here is the full hierarchy:

```
Systems  
  Private
```

- Shared
 - Systems by Type
 - All VSE Resources
 - HP BladeSystem
 - Spare Systems
 - Systems Needing Maintenance
 - Storage Systems
 - Systems by Status
 - Systems by Operating System
 - Clusters by Type
 - Clusters by Status
 - System Functions
- Events
 - Private
 - Shared
 - Events by Severity
 - Login Events
 - Service Events

To see the members of each collection, the **-lf collname** option is used instead of the **-ln** option, where **collname** is the name of the collection. For example, the command:

```
% mxcollection -lf Shared
```

displays the following output:

```
<?xml version="1.0" encoding="UTF-8"?>
<collection-list xmlns="http://www.hp.com/hpsim5.0.0.0">
  <collection name="Shared" type="system">
    <member name="Systems by Type" display-status="-1"
type="collection" default-view="treeview" hidden="false" />
    <member name="Systems by Status" display-status="-1"
type="collection" default-view="treeview" hidden="false" />
    <member name="Systems by Operating System" display-status="-1"
type="collection" default-view="treeview" hidden="false" />
    <member name="Clusters by Type" display-status="-1"
type="collection" default-view="treeview" hidden="false" />
    <member name="Clusters by Status" display-status="-1"
type="collection" default-view="treeview" hidden="false" />
    <member name="System Functions" display-status="-1"
type="collection" default-view="treeview" hidden="false" />
    <member name="Test Query" display-status="0" type="query" default-
view="tableview" hidden="false" />
  </collection>
  <collection name="Shared" type="event">
    <member name="Events by Severity" display-status="-1"
type="collection" default-view="none" hidden="false" />
    <member name="Login Events" display-status="-1" type="collection"
default-view="none" hidden="false" />
    <member name="Service Events" display-status="-1" type="collection"
default-view="none" hidden="false" />
  </collection>
</collection-list>
```

As shown in the hierarchy above, both the Systems and Events collections have a sub-collection called Shared. This is highlighted in the XML output. The items shown for the Shared collection are the same as what is shown above in the hierarchy list, with one notable exception that is highlighted. Our test query collection is shown as part of the **-lf** output for a collection, but not in the **-ln** output.

This is because **-ln** only shows the names of collections, and a query is not a collection. **-lf** shows all of the members of a collection, which can include both collections and query collections.

The **mxcollection(4)** man page details the XML schema for creating collections. When creating collections, you define the name of the collection, whether it is a system or event collection, where the collection goes in the hierarchy, and all of the members. The collection members can be collections, systems or events (depending on the collection type), or nodes in the system. Here is an example of a system collection that has another collection and one node as its members:

```
<?xml version="1.0" encoding="UTF-8" ?>
<collection-list xmlns="http://www.hp.com/hpsim5.0.0.0">
  <collection name="Example" type="system" parent="Systems by Type">
    <member name="All Servers" type="query" display-status="0"
default-view="tableview" hidden="false" />
    <member name="server001" type="node" display-status="0" default-
view="tableview" hidden="false" />
  </collection>
</collection-list>
```

This system collection would be placed under Systems by Type in the hierarchy, and contains two members, the collection All Servers, and the node server001. The type field is very important because it is possible to have collections with the same name reside under **Systems** or **Events**.

To add collections and members of collections to HP SIM, the **-a -f collection.xml** option is used, where `collection.xml` contains the XML description, as shown above. For example,

```
% mxcollection -a -f collection.xml
```

The following shows the XML required to add the query collection Test Query to the Shared collection under Systems, as shown above.

```
<?xml version="1.0" encoding="UTF-8" ?>
<collection-list xmlns="http://www.hp.com/hpsim5.0.0.0">
  <collection name="Shared" type="system">
    <member name="Test Query" display-status="0" type="query" default-
view="tableview" hidden="false" />
  </collection>
</collection-list>
```

Since the collection Shared already exists, you only need to add the members to the collection. The command to add new members to collections is:

```
% mxcollection -a mem -f testcoll.xml
```

where `testcoll.xml` contains the XML shown above.

To remove members of collections, simply replace the **-a** option with **-r**, and all of the members defined in the XML file will be removed from the collection specified in the file `testcoll.xml`. To remove the collection itself, use the **-d CollName** option, where *CollName* is the name of the collection to be removed.

The next example creates a new collection called Test Collection that has two members, the Test Query query collection, and one server, and will be placed under Systems by Type in the hierarchy. The following is the XML for Test Collection.

```
<?xml version="1.0" encoding="UTF-8" ?>
<collection-list xmlns="http://www.hp.com/hpsim5.0.0.0">
```

```

    <collection name="Test Collection" type="system" parent="Systems by
Type">
    <member name="Test Query" type="query" display-status="0" default-
view="tableview" hidden="false" />
    <member name="cupux09" type="node" display-status="0" default-
view="tableview" hidden="false" />
    </collection>
</collection-list>

```

The command to add this collection is, **mxcollection -a -f testcoll.xml**, where `testcoll.xml` contains the XML above.

Event Management

HP SIM can be used to manage alerts from the systems you are managing. HP SIM accepts SNMP traps and WBEM indications. This section discusses how to use the CLI to manage traps and indications. For this discussion, the term event refers to either an SNMP trap or a WBEM indication.

Configuring HP SIM

In the section Inventory management, the section on Configuring HP SIM discusses how to use **mxglobalprotocolsettings** and **mxnodesecurity** to configure SNMP and WBEM authorization credentials. This is particularly important for WBEM because root authorization is required to subscribe to WBEM indications.

WBEM

The command **mxwbemsub** enables you to manage WBEM indication subscriptions for managed systems that are able to send WBEM indications. Each managed server must have WBEM Services (B8465BA) and the System Fault Management (SysFaultMgmt) products installed to enable WBEM indications. Storage devices that support SMI-S are able to send WBEM indications as well.

A WBEM subscription tells the managed system where to send indications using the WBEM protocol. To create a subscription, use one of the following commands:

```

% mxwbemsub -a [destCMS] -n nodenames
% mxwbemsub -a [destCMS] -f nodefile.txt

```

-n is used when specifying the system names on the command line. **-f** is used when specifying a file that contains the system names. When using a file, each system name must be on a separate line. The optional parameter **destCMS** tells the systems being subscribed where to send the indications. If **destCMS** is not provided, then indications are sent to the system where **mxwbemsub** is being executed.

To remove a subscription, replace the **-a** option with the **-r** option. Removing a subscription means the systems on the command line or in the file will no longer send WBEM indications to the specified CMS.

To list the subscriptions for systems, replace the **-a** option with the **-l** option. If the optional parameter **destCMS** is provided, then subscriptions for CMS **destCMS** are displayed.

mxwbemsub also enables you to move WBEM subscriptions from one CMS to another. The command syntax is,

```

% mxwbemsub -m [currentCMS] newCMS -n nodenames

```

```
% mxwbemsub -m [currentCMS] newCMS -f nodefile.txt
```

This command moves the WBEM subscription from *currentCMS* to *newCMS* for the systems on the command line (using **-n**), or for the systems in the text file (using **-f**). The parameter *currentCMS* is optional, and if omitted the WBEM subscription is moved from the current system to *newCMS*.

SNMP

Each managed system must have its SNMP trap destination configured to point to the CMS. Once configured, HP SIM can receive SNMP traps.

There are two commands for managing SNMP MIB information. The command **mcompile** compiles SNMP MIB files into an intermediate format, called a *cfg* file. For example,

```
% mcompile [-d mibdir] mibfile.mib
```

compiles *mibfile.mib* into the intermediate file *mibfile.cfg*. If the **-d** option is provided, *mibfile.mib* must reside in the directory *mibdir*. If the MIB file contains IMPORTS from other MIB files, those MIB files must reside in the same directory as *mibfile.mib*. The *cfg* file will be placed in the same directory where *mibfile.mib* is located.

After compiling the MIBs, the command **mxmib** enables you to process the *cfg* file. You can register, un-register, list registered MIBs, and list the traps for a specific registered MIB.

To register a newly-compiled MIB file, use the **-a** option:

```
% mxmib -a mibfile.cfg
```

This command replaces the data on an already-registered MIB if the MIB has been updated.

To un-register a mib, use the **-d** option,

```
% mxmib -d mibfile.cfg
```

If you have several MIBs you would like to register, you can create a text file and list the *.cfg* files for the MIBs you would like to have registered. Instead of multiple **mxmib -a** commands, one simple command will register all the MIBs. If the text file is called *cfglist.txt*, the command to register multiple MIBs using a text file would be:

```
% mxmib -f cfglist.txt
```

The following command lists all of the MIBs registered in HP SIM:

```
% mxmib -l
```

The output is a list of all the *.mib* files registered in HP SIM. To see a list of the traps associated with a MIB, use the **-t** option:

```
% mxmib -t mibfile.mib
```

Managing HP SIM tasks

A very powerful feature of HP SIM is the ability to create, manage, and schedule tasks. Within HP SIM, all tools can be executed immediately, or scheduled to be run at a future time. This section describes how to manage tasks within HP SIM.

Note: The command **mxtool** is used to add tools to HP SIM. The whitepaper, “*Creating custom tool definition files for HP Systems Insight Manager*”, documents how to create tool definition files, and how to add tools to HP SIM.

Managed server setup

Tool execution on managed servers is controlled by the Distributed Task Facility (DTF) of HP SIM. The DTF communicates with managed servers through SSH. The command **mxagentconfig** pushes the public key for the CMS to a users SSH key directory on the managed servers, and then appends the key to the authenticated keys file. The user must be a valid user on the managed servers. This authentication must also be performed on the CMS so that the DTF can execute tools on the CMS.

The following command pushes the SSH public key to the identified user on one or more managed servers:

```
% mxagentconfig -a [-n host] [-u login] [-o host | user] [-p password]
% mxagentconfig -a [-n host] [-u login] [-o host | user] [-p password] [-f file.txt]
```

The **-n** option identifies the systems to configure. If the **-n** option is not specified, the command will apply to the CMS. The **-u** and **-p** options specify the account on the systems identified with the **-n** option. If the **-u** and **-p** options are not specified, the account information for the account running the command will be used. The **-o** option enables you to choose host-based authentication, **-o host**, or user key-based authentication, **-o user**. If the **-o** option is not specified, host-based authentication is used.

The command,

```
% mxagentconfig [-n host] [-u username]
```

is used to check whether the hosts specified have been authenticated for CMS access by the user specified in the **-u** option. If the host is not specified, CMS authentication is checked. If the user name is not specified, the user name for the user executing the command is used for the check.

To remove authentication, use the **-r** option:

```
% mxagentconfig -r [-n host]
```

This removes the host key for system **host** from the CMS' known hosts file. If no host is provided, the CMS removes authentication to itself.

Managing tasks

The command **mxtask** enables you to create, remove, execute, list, and change ownership of HP SIM tasks, either from the command line or through XML files. A full description for **mxtask** exists in the **mxtask** man page, and a full description of the XML syntax required to manage users with an XML file exists in the **mxtask(4)** man page.

To create a new task in HP SIM, the task must have a name, the name of a collection that will be used to create the list of systems that will execute the selected tool, any parameters required by the tool, scheduling information if the task is not to be executed immediately, the name of a time filter, if required, specifying timeframes for tool execution, and optionally the name of the owner of the task.

The syntax for creating a task is:

```
% mxtask -c tname -q qname -t toolname -w sched [-A tparams | -f tpf.xml] \
[-o towner] [-i tfilter]
```

The **-c tname** option is used to create a task with name *tname*. **-q qname** identifies the collection to use for the task, and **-t toolname** identifies the name of the tool to be executed. Both *qname* and *toolname* must exist in HP SIM. If the tool *toolname* requires parameters, then one of the **-A** or **-f** options must be used. **-A** is used to list the task parameters, *tparams*, on the command line. **-f tpf.xml** is used if the parameters defined in the file *tpf.xml*. As the file name implies, the parameters are defined in XML format as defined in the man page **mxtask(4)**. If the **-o towner** parameter is provided, the new task will be owned by *towner*. The default owner is the user that is running the **mxtask** command. The **-i tfilter** parameter is used to have the task use time filter *tfilter* to define when the task will execute.

The **-w sched** parameter defines the execution schedule for the task. The general format for the **-w sched** parameter is:

```
-w T<tasktype>.I<intervaltype>.P<periodic>.D<day|date>.M<time>
```

<tasktype> defines the type of task, and can be one of the following values:

- add – task runs when the query criteria are met
- remove – task runs when query criteria are no longer met
- manual – task runs only when manually invoked (see next section)
- schedule – task runs based on the P and D time definitions

The rest of the *sched* parameter is required only when the *tasktype* is **schedule**.

I<intervaltype>: Values can be **minutes** (runs every X minutes), **hours** (runs every X hours), **days** (runs every X days at the specified time), **weeks** (runs every X weeks at the specified day and time), **months** (runs every X months at the specified day and time), **daily** (at the specified time), or **runonce** (at the specified date and time). Here, X is defined by the **P** option below.

P<periodic>: Defines the periodic interval for the task, and can be any number greater than zero. For example, P3 means every third time.

D<day|date>: Defines the day or time for task execution, and is only required if the **I** parameter is **weeks**, **months**, or **runonce**. For **weeks**, valid values are the days of the week (**sunday**, **monday**, and so on). For **months**, valid values range from 1-31, representing a day of the month. For **runonce**, the valid date value has the form mm/dd/yy.

M<time>: Defines the time for task execution, and is required for all **I** parameter values except **minutes** and **hours**. The format is hh:mm, using the 24 hour format. For example, 1pm is represented as 13:00.

The **-c** option for creating a task can be complicated if all of the options are needed to properly define the task. You can create a task by defining the task in XML format, and have **mxtask** use the XML definition. The command is:

```
% mxtask -cf taskfile.xml
```

The XML format for the task is defined in the **mxtask(4)** man page. The general format of the XML syntax for the task is:

```
<?xml version="1.0" encoding="UTF-8"?>
  <task-list>
    <task name="task name" type="add|remove|manual|schedule"
owner="taskowner" runmode="runnow|runatstartup">
      <toolname>tool name</toolname>
      <queryname>query name</queryname>
      <scheduleinfo interval="x" periodic="x" time="x" day="x"
date="x"/>
      <timefilter />
      <toolparams />
    </task>
  </task-list>
```

To remove an existing task, use:

```
% mxtask -r taskname
```

taskname is the name of the task to be removed.

To run an existing task immediately, run:

```
% mxtask -e taskname
```

To change the owner of a task, use the **-o** option:

```
% mxtask -o taskname newowner
```

Here, *newowner*, an existing user in HP SIM, becomes the owner of task *taskname*. The user executing this command to change ownership must have authorization to change task ownership (see the section on Managing users and authorizations).

There are different options available for displaying task information. To see a tabular list of all existing tasks in HP SIM, use one of the following commands:

```
% mxtask
% mxtask -lt
```

The output is a table that displays the task name, the name of the tool associated with the task, the time the task was last run, and the schedule for the task. To see details about a specific task, use,

```
% mxtask -ld taskname
```

This command displays the name of the task, tool, and collection associated with the task, task owner, task schedule information, the times of the last run and next run, the last time the task was modified, and XML format of any parameters for the tool associated with this task.

The **-lf** option displays the task definition in XML format. By using file redirection, you can save the output for use in creating additional tasks. For example,

```
% mxtask -lf "Daily Device Identification"
```

displays on the terminal the XML for the "Daily Device Identification" task. However,

```
% mxtask -lf "Daily Device Identification" > deviceid.xml
```

will save the XML for the "Daily Device Identification" task in the file `deviceid.xml`.

There is additional information beyond the data provided in the man pages for **mxtask**. This information is displayed on the console terminal, and is available by using the following command:

```
% mxtask -h
```

HP recommends that the output be piped to **more**, or redirected to a file for easier viewing.

Managing task execution

The command **mxexec** enables you to execute tools that are defined in HP SIM, whether those tools are HP SIM tools, or tools that you have created. As discussed in the previous section, **mxtask** is used to create tasks to enable scheduled execution of tools, and **mxexec** is used for immediate execution of a tool, as well as to manage other jobs running within HP SIM. A full description for **mxexec** exists in the **mxexec** man page.

The command for executing a tool is:

```
% mxexec -t tool [-A arg ...] [-h | -O file | -o dir] [-n target ... | -q query]
```

The only required argument is **-t tool**, where *tool* is the name of the tool to execute. The **-A** option is used to pass required arguments to the tool. The arguments are passed to the tool in the order they appear in the command, so the first argument in the command is the first tool argument, and so on. Any argument not being provided is identified by "".

The **-h**, **-O**, and **-o** options control the output of the running job, and are mutually exclusive. **-h** suppresses the job information headers and sends job output directly to **stdout** and **stderr** as appropriate. **-O** directs all **stdout** and **stderr** output to the file, *file*. If *file* cannot be created in the given path location, HP SIM attempts to open *file* in `/var/tmp`. If that fails, all tool output is sent to **stdout**. **-o** creates one result file per target system in the directory *dir*. The result file contains both **stdout** and **stderr** data, and has a file name that looks like `nodename.job_ID`.

The **-n** and **-q** options control the target systems and system groups that the tool will execute on, and are mutually exclusive. The **-n** option is to be used when providing the target system names and system group names on the command line. To specify a system group, preface the system group name with the prefix, **g:**. The **-q** option is used to specify the name of a collection that will be used to generate the target system list.

mxexec verifies that the user executing the command is authorized to run the selected tool on the target systems. If not, a message is logged in the HP SIM audit log (`/var/opt/mx/logs/mx.log`), sent to **stderr**, and the job is aborted.

The following command cancels a specific job:

```
% mxexec -c [-k] -j job_ID
```

This command cancels the job with ID *job_ID*. The effects of job cancellation depend on the state of the job. A job can be in one of four states:

- Pending – nothing has started
- Copying files – entered if there are files to copy as part of the tool execution
- Running tool – entered if there is a command line to execute as part of the tool execution
- Complete – the job has completed and results are made available by the DTF

A job can only be cancelled if the job is in either Pending or Copying files states. Nothing happens if the job is in Running tool or Complete. Any files copied in the Copying files state are left on the target systems. If a file copy was in progress, that file is restored to its original contents before the copy began.

The **-k** option is used to kill the job. In addition to cancellation steps, **-k** sends the kill signal to the shell process spawned to run the command line associated with the tool.

Note: Killing a job can be dangerous. While unlikely, it is possible to leave a target system in an inconsistent state when killing jobs.

mxexec allows many different ways to display job information. The following commands are used to list job information:

```
% mxexec
% mxexec [-l n | t] [-i taskname ... | -j job_ID ...] [-d date]
% mxexec -l d [-h] -j job_ID ...
```

If no options are used as in the first example above, all jobs run by the user that have not completed are displayed. The **-l n** option specifies to list job IDs only. The **-l t** option lists job information in a tabular format, providing task name, job ID, HP SIM user, HP SIM tool name, and the job state. The **-i** and **-j** options are mutually exclusive. **-i** is used to specify one or more task names for which data for any job associated with the task names will be displayed. **-j** is used to specify one or more job IDs to display information for. Note that since **-l n** specifies to only show job IDs, **-j** is not needed in conjunction with **-l n**. The **-d date** option is used to limit displaying information for those jobs that completed after the *date* parameter. **date** must have the following format:

```
"Month/day/year hour:minute AM | PM"
```

The quotes are required so the string is interpreted as one parameter, including the embedded spaces.

In the third example above, the **-l d** option is used to display detailed information about the job ID listed with the **-j** option. Only one job ID is allowed. **stdout**, **stderr**, and completion status per target system is displayed.

Managing users and authorizations

In HP SIM, an authorization is comprised of three components: a user, the set of systems the user is managing, and the set of tools the user can use on the set of systems. This section discusses using the CLI for managing users, managing toolboxes, and managing authorizations.

Managing users

The command **mxuser** enables you to add, modify, remove, or list users in HP SIM. A full description for **mxuser** exists in the **mxuser** man page, and a full description of the XML syntax required to manage users with an XML file exists in the **mxuser(4)** man page.

The command **mxuser -a user** adds the user with the login name *user* to HP SIM. The user must exist as a user on the operating system in order to sign in to HP SIM, but does not need to exist to be added to HP SIM. This user will have limited access rights and no authorizations.

The **-d description** option enables you to provide a description for the user. A common use for the description is to provide the role of the user, like Database Administrator, or HPUX Server Manager. For example,

```
% mxuser -a Jane -d "HPUX Administrator"
```

The **-p full | limited | none** option enables you to specify the rights of the user. A **full** rights user has unlimited access HP SIM capabilities. A **limited** rights user can create tools in HP SIM, and like the **none** rights user, has access to the listing options of HP SIM commands.

The **-C source** option is useful when creating several users with the same authorizations. See Managing authorizations below for details on authorizations.

The **-g userGroup** option can replace *user* in the **-a** option to create a user group. The **-d** and **-p** options can be used to provide a description and access rights for the group.

The **-m user** option, used in conjunction with **-d** and **-p**, enables you to modify the description and access rights of *user*. In the example above, to modify Jane's description, the command will look like,

```
% mxuser -m Jane -d "Database Administrator"
```

Using **-m -g userGroup**, in conjunction with **-d** and **-p**, modifies the characteristics of group *userGroup*.

The commands **mxuser -r user** and **mxuser -r -g userGroup** can be used to remove a user and a group, respectively. All authorizations for the user and group are also removed.

The **-f file** option enables you to add, modify, or remove multiple users and user groups. **File** contains XML syntax, allowing multiple entries to be processed. Use this option with **-a**, **-m**, and **-r**, replacing the user name or user group name with **-f file**. Another benefit to using a file is that you can define more characteristics for the users and groups. For example, you can specify e-mail and phone contact information for each user. The **mxuser(4)** man page provides details of the XML file syntax.

There are several ways to display user information on the screen. You can also save the user information in an XML-formatted file.

```
% mxuser -l t ; displays the user name, rights, and description in tabular format
% mxuser -l d ; displays all user information in a screen-viewable format
% mxuser -l n ; displays only the user names
% mxuser -l f ; displays all user information in XML format
```

For each of the **-l** options, you can specify one or more user names and one or more user group names to get the information for only those users and user groups. For example,

```
% mxuser -l d Jane Bob
```

This displays the details for users Jane and Bob.

Managing system groups

If a user or set of users is managing several systems, it is convenient to put them into a system group. This enables any authorization created for the users to be created with one command as described in the Managing authorizations section below. If a user is only managing a few systems, it is just as easy to create the authorizations on a system-by-system basis.

The command **mxngroup** enables you to create, remove, modify, and list system groups in HP SIM. A full description for **mxngroup** exists in the **mxngroup** man page, and a full description of the XML syntax for managing system groups from an XML file exists in the **mxngroup(4)** man page.

The command to create a system group is:

```
% mxngroup -a -g grpname [-d description] [-n [member1 member2 ...]]
```

where *grpname* is the name of the system group. The optional parameter **-d description** enables you to provide a description of the group, like PA Systems, or All Windows Servers. The optional parameter **-n** enables you to provide a list of systems that will be members of the new group. If the **-n** parameter is left off, or if there are no members provided for the **-n** parameter, the newly-created system group will be empty.

When creating a system group that contains many systems, it is easier to create an XML file that contains the list of systems to be added to the group. The syntax to create a system group from an XML file is,

```
% mxngroup -a -f grpfile.xml
```

Here is a simple example of the syntax required to create a system group, "group1", providing a description, and adding two systems to the group:

```
<?xml version="1.0" encoding="UTF-8"?>
<node-group-list>
  <node-group name="group1">
    <description>This is group1.</description>
    <member-node value="system1.corp.com" />
    <member-node value="system2.corp.com" />
  </node-group>
</node-group-list>
```

To modify existing system groups, the syntax is identical to that for adding system groups, simply replace the **-a** with **-m**:

```
% mxngroup -m -g grpname [-d description] [-n [member1 member2 ...]]
```

If both optional parameters **-d** and **-n** are not provided, then the group is not modified. Providing the **-d** option replaces the existing group description with the one provided. Using the **-n** option is only recommended for groups that have very few entries because the systems must be listed in the same order as they were added, with new system names in the place of the systems they are replacing. New systems being added must be provided at the end of the list.

The following are examples of modifying a group. The group we are modifying is *grp*, and it has three members, **n1**, **n2**, and **n3**, listed in that order when the group was created.

```
% mxngroup -m -g grp -n n4 n2 n3 ; replace n1 with n4.
% mxngroup -m -g grp -n n5 ; n5 is the only member of the group
% mxngroup -m -g grp -n n5 n4 n3 n2 ; add n4, n3, n2 to the group
```

Care must be taken when modifying groups from the command line, as the second example illustrates, because the **-m** option is actually *replacing* systems in the group.

Using XML syntax to modify a group is more reliable, especially when the group has many members. To modify a group that is described in the file `grp.xml`,

```
% mxngroup -m -f grp.xml
```

The same rules apply when replacing systems using XML syntax.

To remove a system group, or several system groups, use one of the following commands:

```
% mxngroup -r -g grp ; remove group "grp"
% mxngroup -r -f grprm.xml ; remove groups listed in XML file "grprm.xml"
```

When modifying or removing system groups, HP SIM modifies or removes, respectively, any authorizations associated with the groups.

The **mxngroup** command provides two basic ways to display group information:

```
% mxngroup -l [d | f | m | n] -g grpname
% mxngroup -l [d | f | m | n] -n member
```

The first command displays information about group *grpname*. The second command displays information for all the groups that contain system *member*. If either the **-g grpname** or **-n member** options are left off, the information for all system groups in HP SIM are displayed.

The optional parameter defines the formatting style and amount of information provided. It must be one of the following four values:

```
d – displays detailed, screen-viewable listing of group information
f – displays group information in XML format
m – displays the members of the group
n – displays the names of the system groups
```

Managing toolboxes

The command **mxtoolbox** enables you to add, modify, remove, or list toolboxes in HP SIM. A full description for **mxtoolbox** exists in the **mxtoolbox** man page, and a full description of the XML syntax required to manage toolboxes via a XML file exists in the **mxtoolbox(4)** man page.

Toolboxes are an integral part of creating authorizations in HP SIM. A toolbox contains a set of HP SIM tools that one or more users can run on the systems they are authorized to manage. A simple example of using toolboxes is to create a toolbox containing tools that apply only to HP-UX systems, and a toolbox containing tools that apply only to Windows systems. This allows HP-UX administrators using HP SIM to only see the tools they need, and the same for Windows administrators.

HP SIM has two pre-defined toolboxes, called All Tools and Monitor Tools. All Tools contains everything, and cannot be modified in any way. Monitor Tools contains tools that monitor, display, view, and report information, and can be modified if so desired.

The following commands add, modify, and remove toolboxes, respectively:

```
% mxtoolbox -a tbname [-d description] ; add toolbox "tbname"  
% mxtoolbox -m tbname [-d description] [-N newtbname] ; modify toolbox "tbname"  
% mxtoolbox -r tbname ; remove toolbox "tbname"
```

The **-d** *description* is used to provide a description of the toolbox. In the modify toolbox example, **-N** *newtbname* is used to rename the toolbox from *tbname* to *newtbname*. Both **-d** and **-N** are optional parameters.

Toolbox names are limited to a length of 16 characters, and must start with an alphanumeric character. The rest of the name can contain alphanumeric characters, embedded spaces, underscores (_), or hyphens (-).

Instead of providing toolbox information on the command line, an XML file can be created as defined in the **mxtoolbox(4)** man page. The **-f** *tbfile* option is used to specify the XML file to be used to add, modify, or remove toolboxes. This is useful when managing multiple toolboxes. For example, to add toolboxes defined in the file *tbfile.xml*,

```
% mxtoolbox -a -f tbfile.xml
```

The command **mxtoolbox -l** produces a columnar list of all the toolboxes in HP SIM. The following options produce different presentations:

```
-n ; columnar list of all toolbox names  
-t ; tabular display of all toolbox attributes  
-f ; XML format of all toolbox attributes
```

Adding tools to toolboxes

The command **mxtool** enables you to add, modify, remove, or list tools in HP SIM. A full description for **mxtool** exists in the **mxtool** man page, and a full description of the XML syntax required to manage toolboxes from an XML file exists in the **mxtool(4)** man page.

This paper does not discuss the details of creating and managing tools. Please refer to the white paper, *“Creating custom tool definition files for HP Systems Insight Manager”* for a detailed discussion of that topic. However, important items in that paper that involve toolboxes are discussed here.

The previous section discussed how to create toolboxes to be used in creating authorizations. When creating tools, the following XML entry is used to put the tool in a specific toolbox:

```
<toolbox toolbox-name="My toolbox" />
```

You can have more than one `<toolbox.../>` entry to have the tool included in multiple toolboxes.

Important: Do not add tools to the Monitor Tools toolbox that provide root or administrator level functionality, since this toolbox is assigned to the operator and user templates, and the tools can run with root or administrator access on the CMS.

Managing authorizations

The command **mxauth** enables you to add, remove, or list authorizations in HP SIM. A full description for **mxauth** exists in the **mxauth** man page, and a full description of the XML syntax required to manage authorizations from an XML file exists in the **mxauth(4)** man page.

Before creating authorizations, the users, system groups (optional), and toolboxes must be in place as describe in the sections above. Once the users, tools, and toolboxes are created, authorizations can be created.

The commands to create a node authorization are:

```
% mxauth -a -u username -R toolboxname -n nodename
% mxauth -a -u username -R toolboxname -g groupname
% mxauth -a -f authfile.xml
```

The first form creates a node authorization that enables *username* to use the tools in *toolboxname* to manage system *nodename*. The second form creates a group authorization that enables *username* to use the tools in *toolboxname* to manage all the systems in system group *groupname*. The third form creates authorizations as defined in the XML file *authfile.xml*. Here is a simple example of what *authfile.xml* would look like to create three system authorizations:

```
<?xml version="1.0" encoding="UTF-8"?>
<auth-list>
  <node-authorization user-name="user1"
    toolbox-name="webadmin"
    node-name="node1.corp.com" />
  <node-authorization user-name="user2"
    toolbox-name="operator"
    node-name="node1.corp.com" />
  <node-authorization user-name="user3"
    toolbox-name="dbadmin"
    node-name="node2.corp.com" />
</auth-list>
```

In the three commands above, replace the **-a** with **-r**, and the three command remove the system authorization, remove the group authorization, and remove authorizations as specified in the XML file, respectively.

To view authorizations, the command, **mxauth -lt** lists all of the authorizations in the format,

```
user:toolboxname:n:nodename
```

for system authorizations, and in the format,

```
user:toolboxname:g:groupname
```

for group authorizations. The command **mxauth -lf** displays the authorizations in XML format.

HP SIM configuration

There are many commands that are used to configure the HP SIM software after it is installed. Some of the functionality available is service start and stop, connecting to an Oracle database, initializing HP SIM, and managing trusted certificates.

HP SIM initialization

After installing the HP SIM software, HP SIM must be initialized. The command **mxinitconfig** performs initialization of HP SIM.

```
% mxinitconfig -l           ; show configuration status
% mxinitconfig -a [-F]     ; configure HP SIM
% mxinitconfig -r         ; unconfigure HP SIM
```

mxinitconfig, or **mxinitconfig -l**, shows the configuration status, including kernel parameter checks, database setup, web server setup, and thirteen other items. The **-a** parameter configures HP SIM – web services are started, HP SIM services are started, the database is initialized, and so. The **-r** parameter unconfigures HP SIM by stopping all services, removing the database, and so on. The configure option **-F** specifies force mode, meaning do not stop configuration when an error is detected.

If an Oracle database is being used, the command **mxoracleconfig** must be run *before* running **mxinitconfig**. This is because **mxinitconfig** initializes the database, so it must be in place before initialization. The syntax for **mxoracleconfig** is:

```
% mxoracleconfig -h host [-n port] -d db -u u -p pw [-i jar] [-f ]
```

-h *host* specifies the host name where the Oracle database is installed. **-d** *db* specifies the instance name of the database HP SIM will use. This should be a database created for the exclusive use of HP SIM, created to use the Unicode character set, AL32UTF8. **-u** *user* specifies the user authorized to log in to the Oracle database instance. **-p** *pw* specifies the password for *user*. The optional parameter **-n** *port* can be used to specify the Oracle TCP/IP listener service port. If not specified, the default port 1521 is used. Optional parameter **-i** *jar* specifies the location of the thin client jar file containing the jdbc driver. This file is typically named `ojdbc14.jar`. The optional parameter **-f** is used to rerun this command. Typically, **mxoracleconfig** only needs to be run once, but if it needs to be run again, the **-f** option allows a rerun.

HP SIM services

There are two commands that enable you to stop and start the HP SIM services. The command **mxstop** stops all of the HP SIM services and the Tomcat web server. The command **mxstart** starts all of the HP SIM services and the Tomcat web server. Neither command has any options. The HP SIM services can get in a bad state if the services are restarted too soon after stopping. After running **mxstop**, HP recommends that **mxstart** not be run until all of the services and related processes are stopped. This can take several minutes. On HP-UX and Linux operating systems, run the command **ps -ef | grep mx** until there are no more processes that start with **mx** running. If there are still **mx** processes running after several minutes, manually kill them by running **kill PID**, where *PID* is the process ID displayed from the **ps -ef | grep mx** command.

HP SIM log

The command **mxlog** enables you to create an entry in a log file, or to print a log entry to **stdout**. The basic forms of the command are:

```
% mxlog -l
% mxlog -p
```

The **-l** option logs the entry to the log file. The **-p** option prints a preview of the log entry to **stdout**. These two options are mutually exclusive.

The following options can be used with either the **-l** or **-p** options. Some of the options listed have a default value provided if the option is not specified on in the command.

-a action: The *action* that occurred. If this option is not used, the action STAR" will be used by default. Valid values include ACTION_NONE, ADD, MODIFY, DELETE, RUN, LIST, ENABLE, DISABLE, RESTORE, SAVE, SETUP, UNSETUP, START, and DONE.

-m message: The specific *message* to be logged. This message will appear on the line after the log entry in the log file.

-n appl: The name of the object or application that is making the log entry. If this option is not used, the object NONE will be used by default.

-o object: The type of object or application that is making the log entry. If this option is not used, the default value of TYPE_NONE will be used. Valid values include APPLICATION, GUI, and TYPE_NONE.

-r result: The result of the event that occurred. If this option is not used, the default value of SUCCESS will be used. Value values include IN_PROGRESS, SUCCESS, FAILURE, SOME_FALURES, CANCELED, KILLED, and RESULT_NONE.

-s sessionID: The session ID in string form.

-t transactionID: The transaction ID in string form.

-u user: The user that submitted this log entry. If this option is not used, the default value is the user name of the user running the command. This is typically used to specify the name of the application logging the message rather than the run-as user of the application.

-v verbosity: The level of verbosity of this log entry. If this option is not used, the default value of SUMMARY is used. Valid values include SUMMARY, DETAIL, VERBOSE, ERROR, WARNING, and NOTE.

In addition to the above options, certain columns in the log entry are filled-in to default values at all times. This includes a timestamp of when the entry was created, and the Category column, which is always set to APPLICATION.

HP SIM information

The command **mxgethostname** displays the host name and DNS name of the local server. The **-i** option adds the IP address of the local server to the display. The **-n host** option displays the host name and DNS name of system *host*.

The command **mxversion** displays the full version string for HP SIM. The full version string includes the version name, the version number, and the date and time the version was built. The version string will look like:

Systems Insight Manager 5.0 with Update 1 – HP-UX C.05.00.01.00.06 (2006-01-05 17:13)

The command **mxgetdbinfo** displays information about the HP SIM database. Without any parameters, the database port, database name, user ID, database type, path to executables, and path to data are all displayed. The following options are supported:

- a**: same output as if no option was provided.
- h**: returns the host name of the server where the database is installed.
- p**: returns the TCP/IP port number used to connect to the database.
- d**: returns the name of the database used by HP SIM.
- u**: returns the username/userID used by HP SIM to log into the database.
- t**: returns the database type – SQLServer, PostgreSQL, or Oracle.
- r**: (HP-UX only) returns the path to the postgresSQL data directory.
- b**: (HP-UX only) returns the path to the postgresSQL program directory.

HP SIM settings

The command **mxglobalsettings** is used to set and view global settings that are stored in the file `/etc/opt/mx/config/globalsettings.props` on HP-UX and Linux, or in the HP SIM installation directory under `config\globalsettings.props` on Windows. This file is a text file that contains various configuration items and their default values. The command syntax is:

```
% mxglobalsettings -s [-f] setting=value ...      ; set value(s)
% mxglobalsettings -ld [setting ...]             ; display value(s)
```

The **-s** option is used to set and/or create entries in `globalsettings.props`. If **setting** exists in the file, then it is set to the new value, *value*. If **setting** is not in the file, then **mxglobalsettings** prompts for confirmation to create the new entry. If confirmed, the entry is created, and its initial value will be *value*. The **-f** option forces creation of the new entry, bypassing the confirmation.

The **-ld** option is used to display settings and their current value in **name = value** format. If one or more setting names are provided, then only those settings will be listed. Otherwise, all settings will be shown.

HP SIM Server certificate

The command **mxcert** is used to manage the server and trusted certificates in the HP SIM key store. The following command can be used to create a new server certificate:

```
% mxcert -n CN=common_name OU=organization_unit O=organization_name \
          L=locality_name
```

To import a signed server certificate, enter:

```
% mxcert -i -f filespec
```

filespec is the full path to the signed certificate file. To import a trusted certificate, enter:

```
% mxcert -t -f filespec
```

To tell HP SIM to require, or not require, trusted certificates, use the **-b** option:

```
% mxcert -b [1 | 0]
```

The value 1 is used to require trusted certificates; the value 0 is used to not require trusted certificates.

To list the certificates in the key store, use either **-ln** or **-ld**:

```
% mxcert -ln | -ld
```

-ln lists basic information about the certificate, and **-ld** lists detailed information about the certificates, including certificate fingerprint data.

The **-s** option synchronizes the server certificate with the server certificate in the SSL shared directory. The **-c** option creates a Certificate Signing Request.

HP SIM passwords

The command **mxpassword** is used to add, remove, modify, and list SSH key passwords stored by HP SIM. **Note:** The passwords are displayed in clear text. Take great care when using this command to insure that passwords are not seen by unintended eyes.

The command to add a new password is:

```
% mxpassword -a -x key1=value1 [-x key2=value ...]
% mxpassword -a -f file
```

The **-x** option assigns password *value* to password key name *key*. There can be multiple key and value pairs provided to the add command. The **-f** option adds the passwords defined in the file, *file*. This is a more secure option because the passwords are not available for viewing on the terminal screen, or in the command line history.

The command to modify an existing password is:

```
% mxpassword -m -x key1=newvalue [-x key2=newvalue ...]
```

To remove existing passwords, use:

```
% mxpassword -r -x key1 [-x key2 ...]
```

To list existing passwords, use:

```
% mxpassword -l [-x key ...]
```

If specific keys are provided using the **-x** option, the passwords for those keys are displayed. Otherwise, all known passwords are displayed.

mxpassword has three other options available. **-g** launches a GUI, **-c** checks for the existence of a key store, and **-n** creates a new server certificate.

Appendix

mcompile – Compiles SNMP MIB files for registration.
mxagentconfig – Configures SSH on a managed system.
mxauth – Manages HP SIM authorizations.
mxcert – Manages server and trusted certificates.
mxcollection – Manages collections.
mxdtf – Starts the mxdtf daemon.
mxexec – Executes an HP SIM tool.
mxgetdbinfo – Displays information about the HP SIM database.
mxgethostname – Displays the name of the local host.
mxglobalprotocolsettings – Configures global protocol settings.
mxglobalsettings – Manages HP SIM settings.
mxinitconfig – Performs initial configuration of HP SIM.
mxlog – Adds or displays log entries.
mxmib – Manages MIB registration.
mxngroup – Manages system groups.
mxnode – System management.
mxnodesecurity – Manages system credentials.
mxoracleconfig – Configures HP SIM to use an Oracle database.
mxpassword – Manages passwords.
mxquery – Manages collections.
mxreport – Manages reports.
mxstart – Starts HP SIM and the Tomcat web server.
mxstm – Manages system type manager rules.
mxstop – Stops HP SIM and the Tomcat web server.
mxtask – Manages HP SIM tasks.
mxtool – Manages HP SIM tools.
mxtoolbox – Manages HP SIM toolboxes.
mxuser – Manages HP SIM users.
mxversion – Displays the version of HP SIM installed on the server.
mxwbemsub – Manages WBEM event subscriptions.

© Copyright 2004-2007 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation
436331-001 01/2007