

# Neuron Management Server

USER'S GUIDE

Product Version: 1.16.0

Document Revision: 1.0.0



## Copyright

Copyright © 1995-2011 Halcyon Monitoring Solutions, Inc.  
All rights reserved. This product and related documentation is protected by copyright and distributed under licenses restricting its use, copying, distribution and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Halcyon Monitoring Solutions, Inc. and its licensors.

Corporate Headquarters  
Halcyon Monitoring Solutions  
800 Bellevue Way NE,  
Suite 400  
Bellevue, WA 98004, USA

Tel: 416-932-4600  
Fax: 416-932-4711  
Email: [info@HalcyonInc.com](mailto:info@HalcyonInc.com)  
URL: [www.HalcyonInc.com](http://www.HalcyonInc.com)

## License Agreement

Downloading Halcyon software constitutes acceptance of the End User Binary Code License agreement, which can be found here: <http://www.halcyoninc.com/products/license.php>

Without purchasing a license, the Halcyon Software will only operate for a trial period of 14 days from installation. If you wish to purchase a license to use the Halcyon Software, please contact us at:

Web: <http://www.HalcyonInc.com>  
Email: [info@HalcyonInc.com](mailto:info@HalcyonInc.com)  
Tel: 416-932-4647  
Fax: 416-932-4711

## Technical Support

For assistance with any Halcyon products, contact Technical Support:

Tel: 1-877-932-4666 (Toll Free in North America)  
Tel: 1-416-932-4666 (International)  
Email: [support@HalcyonInc.com](mailto:support@HalcyonInc.com)

### Halcyon Forums:

Halcyon experts actively participate in the online Halcyon Forums. The experts are constantly monitoring the forums, answering questions and posting useful tips, tricks, and general knowledge base information. Whether you have a technical question or just wish to expand our knowledge base, this is the place for you. <http://forums.HalcyonInc.com>

## About Halcyon

Halcyon delivers Infrastructure Management solutions that provide operational visibility, availability, and reliability for business critical services and their underlying infrastructure. Since 1994, numerous Fortune 100 and SMEs, spanning every major geography and sector, have adopted Halcyon solutions.

At Halcyon, we believe the health of the IT infrastructure is integral to the success of a business. Our clients rely on us for complete end-to-end monitoring solutions that are straightforward, easy to deploy and use, and cost-effective, coupled with a history of client service excellence.

### **Your Infrastructure is Our Business**

# Contents

---

<b>1</b>	<b><u>PREFACE</u></b>	<b>4</b>
1.1	PURPOSE OF THE DOCUMENT	4
1.2	INTENDED AUDIENCE	4
1.3	RELATED DOCUMENTS	4
<b>2</b>	<b><u>GENERAL OVERVIEW</u></b>	<b>6</b>
<b>3</b>	<b><u>NEURON MANAGEMENT PORTAL</u></b>	<b>7</b>
3.1	REQUIREMENTS	8
3.2	TROUBLESHOOTING	8
<b>4</b>	<b><u>USER MANAGEMENT</u></b>	<b>9</b>
4.1	USERS	9
4.2	GROUPS	9
4.3	DOMAINS	9
4.4	PASSWORDS	10
4.5	USER MANAGER UTILITY	10
4.5.1	LISTING USERS	10
4.5.2	ADDING USERS	10
4.5.3	MODIFYING USERS	11
4.5.4	DELETING USERS	11
4.5.5	CHANGING PASSWORDS	11
4.6	ROLES	12
<b>5</b>	<b><u>LICENSE MANAGEMENT</u></b>	<b>19</b>
5.1	LISTING USERS	19
5.2	ADDING LICENSES	20
5.3	REMOVING LICENSES	20
5.4	LICENSE FILE	20

## Tables

---

TABLE 1.2-1: INTENDED AUDIENCE	4
TABLE 1.3-1: RELATED DOCUMENTS	4
TABLE 4.6-1: USER ROLES	12
TABLE 4.6-2: MANAGER ROLES	15
TABLE 4.6-3: ADMINISTRATOR ROLES	17

# 1 Preface

## 1.1 Purpose of the Document

The purpose of this document is to describe the Neuron Management Server and how the end-users, managers, and administrators will use it.

## 1.2 Intended Audience

This guide is written for the following type of audience:

**Table 1.2-1: Intended Audience**

Role	Usage
End User	The User's Guide is intended for end users who use the product on a daily basis. This guide provides information on how to use the product for tasks such as generating and viewing reports.
Manager	The User's Guide provides information for managers who are responsible for preparing the product for use by end users.
Administrator	The User's Guide contains information for administrators in order to configure the product.

## 1.3 Related Documents

This solution is composed of a series of underlying products. For further information regarding the configuration, the usage and administration of the products, please refer to the following documents.

These documents may be located in the doc folder of the solution distribution or on the website ([www.halcyoninc.com/docs](http://www.halcyoninc.com/docs)).

**Table 1.3-1: Related Documents**

Component Name	Related Documents
Neuron Management Suite	<ul style="list-style-type: none"> <li>▪ Neuron Management Suite Installation Guide</li> </ul>
Neuron Management Server	<ul style="list-style-type: none"> <li>▪ Neuron Management Server Release Notes</li> <li>▪ README.config</li> </ul>
Neuron Event Manager	<ul style="list-style-type: none"> <li>▪ Neuron Event Manager Release Notes</li> <li>▪ Neuron Event Manager User's Guide</li> </ul>

---

Neuron Inventory Manager	<ul style="list-style-type: none"><li>▪ Neuron Inventory Manager Release Notes</li><li>▪ Neuron Inventory Manager User's Guide</li></ul>
--------------------------	--

---

Neuron Configuration Manager	<ul style="list-style-type: none"><li>▪ Neuron Configuration Manager Release Notes</li><li>▪ Neuron Configuration Manager User's Guide</li></ul>
------------------------------	--

---

Neuron Health Manager	<ul style="list-style-type: none"><li>▪ Neuron Health Manager Release Notes</li><li>▪ Neuron Health Manager User's Guide</li></ul>
-----------------------	--

---

Neuron Performance Manager	<ul style="list-style-type: none"><li>▪ Neuron Performance Manager Release Notes</li><li>▪ Neuron Performance Manager User's Guide</li></ul>
----------------------------	--

---

## 2 General Overview

---

*Neuron Management Server* is the core of the Neuron Management Suite. It is a lightweight centralized event management framework which enables organizations to perform monitoring of their hardware and software. It can also be used to leverage their existing Enterprise Management Framework (EMF) to gain visibility into the health, performance, and status of critical business services in addition to the hardware and software components that make up the services.

The *Neuron Management Server* provides the main functionality for all other *Neuron Management Suite* products. The server also comes with the *Neuron Management Portal* which is a web-based user interface for managing the modules, adapters, and services that are part of the *Neuron Management Suite*. Please refer to section 3 for more details.

## 3 Neuron Management Portal

The *Neuron Management Portal* is a web-based user interface used to access much of the *Neuron Management Suite* functionality.

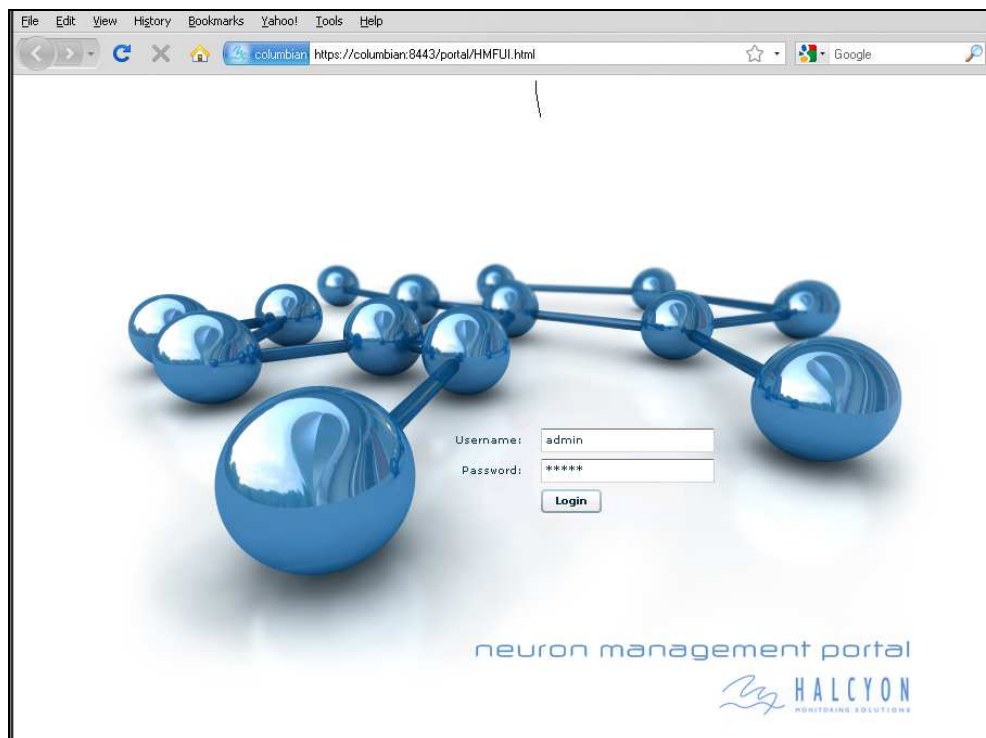
The first step in using the Portal is to login. To do this, you use the URL provided at the end of the installation:

```
https://<hostname>:<port>/portal
```

The hostname is the name of the host where the *Neuron Management Suite* was installed.

The port number can be found in the managementPortal.config file located in the [LOCALDIR]/conf directory.

By default LOCALDIR is /var/opt/HMF, however it can be specified during installation; please refer to the Neuron Management Suite Installation Guide for details on determining what LOCALDIR is.



**Figure 3-1: Neuron Management Portal Login Screen**

Please note that during installation the default user *admin* is set up with a secure password. In order to change the default password of this user or to create additional users use the *User Manager* utility. Please see section 4 below for more information.

Once logged in, you will have access to all the Neuron Management Suite functionality described through the various user guides:

- Neuron Event Manager User's Guide
- README.config
- Neuron Inventory Manager User's Guide
- Neuron Configuration Manager User's Guide
- Neuron Health Manager User's Guide

## 3.1 Requirements

Adobe Flash must be installed on the system you are using to access the Neuron Management Portal. Halcyon recommends Adobe Flash version 10 or later.

**NOTE:** If you do not have Adobe Flash installed, or you have an unsupported version of Adobe Flash installed, a message will appear asking you to Get Flash. Halcyon officially supports the Flash version mentioned above as the product was tested against it. It is possible the product may function correctly with other versions.

The Neuron Management Portal is officially supported on the following browsers:

- Mozilla Firefox 3.0-5.0
- Microsoft Internet Explorer 7-8

**NOTE:** Halcyon officially supports the browsers and versions listed above as the product was tested against these releases. It is possible the product may function correctly with other browsers and versions.

## 3.2 Troubleshooting

If you see an RSL error when browsing to the link noted above, please clear your browser cache and authenticated sessions and then refresh your browser window.

## 4 User Management

---

*Neuron Management Server* provides extensive support for use by multiple users. Each user must be part of a group and a domain. All user configuration is done in the following files which can be found in [LOCALDIR]/conf (see *Neuron Management Suite Installation Guide* for more information):

- users.xml
- groups.xml
- domains.xml

The following sections describe these files and how to modify them using the *User Manager* utility.

### 4.1 Users

The *users.xml* file defines all users that may use the *Neuron Management Server*. There are several preconfigured users already defined. Make sure not to remove them or the product will not work properly.

For each user that may use the product one entry is required.

### 4.2 Groups

The *groups.xml* file defines all groups that are supported by the *Neuron Management Server*. Each of the preconfigured users is already in this file. Any new users must be added to the appropriate group. The following groups are available:

users	Users have read-only access to most features in the <i>Neuron Management Suite</i> . For example, they can view events in the <i>Event Viewer</i> or rules in the <i>Event Manager</i> . They can also access configuration and licensing information. However, they do not have authority to change this information.
managers	Managers have read and write access to most features in the <i>Neuron Management Suite</i> . For example, they can modify rules in the <i>Event Manager</i> or update templates and schedules in the <i>Performance Manager</i> .
administrators	Administrators have full access to all features in the <i>Neuron Management Suite</i> . They can configure all aspects of the product.

### 4.3 Domains

The *domains.xml* file defines all Neuron domains that are supported by the *Neuron Management Server*. Each of the preconfigured users is already in this file. Any new users must be added to the appropriate domain.

Note: Currently, only the *default* domain is supported.

## 4.4 Passwords

The User Manager utility checks whether provided passwords meet the following requirements:

- must be at least 8 characters
- must not contain the following characters: ' (single quote), " (double quote)

## 4.5 User Manager Utility

The User Manager utility provides an easy way to list, add, modify, and delete users. It is also used to change the password of users. This utility must be executed by the root user.

### 4.5.1 Listing Users

In order to list all currently configured users invoke the utility as follows:

```
# [BASEDIR]/bin/UserManager -list
```

Example output:

```
#/opt/HMF/bin/UserManager -list
User      Group Domain
-----
admin     admins default
broker    admins default
gateway   users  default
guest     users  default
router    users  default
```

### 4.5.2 Adding Users

In order to add a user invoke the utility as follows:

```
# [BASEDIR]/bin/UserManager -add <user> <group>
```

where user is the name of the user to add and group is one of the supported groups (see section 4.2). The utility will ask for the password of the new user.

Example output:

```
#/opt/HMF/bin/UserManager -add testUser admins
Please enter the password for testUser:
Please verify password:
User testUser has been added.
```

### 4.5.3 Modifying Users

In order to modify a user invoke the utility as follows:

```
# [BASEDIR]/bin/UserManager -mod <user> <group>
```

where user is the name of the user to add and group is one of the supported groups (see section 4.2). This will update the group of the provided user.

Example output:

```
#/opt/HMF/bin/UserManager -mod testUser users  
User testUser has been modified.
```

### 4.5.4 Deleting Users

In order to delete an existing user invoke the utility as follows:

```
# [BASEDIR]/bin/UserManager -del <user>
```

where user is the name of the user to delete.

Example output:

```
#/opt/HMF/bin/UserManager -del testUser  
User testUser has been deleted.
```

### 4.5.5 Changing Passwords

In order to change the password of an existing user invoke the utility as follows:

```
# [BASEDIR]/bin/UserManager -passwd <user>
```

where user is the name of the user whose password is to be changed. The utility will ask for the new password of the user.

Example output:

```
#/opt/HMF/bin/UserManager -passwd testUser  
Please enter the password for testUser:  
Please verify password:  
Password of user testUser has been changed.
```

## 4.6 Roles

As outlined section 4.2 the *Neuron Management Server* supports multiple groups. The following table lists all roles that are associated with these groups.

**Table 4.6-1: User Roles**

Role Name	Role Description
au.pending.read	Read pending actions.
au.schedule.read	Read automation schedule.
au.schedules.read	Read automation schedules.
cf.configurations.read	Read configurations.
cf.configuration.properties.read	Read configuration properties
cl.tab.configuration	Read Configuration Tab data.
cl.tab.inventory	Read Inventory Tab data.
cl.tab.summary	Read Summary Tab data.
cm.asset.read	Read asset from CMDB.
cm.asset.aliases.read	Read asset aliases from CMDB.
cm.asset.operations.read	Read asset operations from CMDB.
cm.asset.types.read	Read asset types from CMDB.
cm.assets.read	Read assets from CMDB.
cm.configuration.read	Read configuration asset from CMDB.
cm.configurations.read	Read configuration assets from CMDB.
cm.credential.read	Read credential from CMDB.
cm.credential.types.read	Read credential types from CMDB.
cm.event.read	Read event details from CMDB.
cm.license.read	Read license details from CMDB.
cm.mapping.read	Read mapping from CMDB.
cm.metric.read	Read metric from CMDB.
cm.metric.type.read	Read metric type from CMDB.
cm.metrics.read	Read metrics from CMDB
cm.operation.read	Read operation from CMDB.
cm.operations.read	Read operations from CMDB.
cm.query	Query information from CMDB.
cm.rule.read	Read rule from CMDB.
cm.rulebase.read	Read rule base from CMDB.

cm.snapshot.compare	Compare snapshots.
cm.snapshot.read	Read snapshots.
cm.tag.read	Read tag from CMDB.
cm.tag.type.read	Read tag type from CMDB.
cm.tags.read	Read tags from CMDB.
co.task.execute	Execute a task.
em.message.send	Send email message.
em.message.store	Store email message.
ew.execute	Execute EnergyWise request.
ew.ping	Execute EnergyWise ping.
ht.execute	Execute HTTP request.
in.configuration.read	Read configuration information.
ip.execute	Execute IPMI request.
ip.sel.read	Read IPMI SEL information.
ip.server.ping	Execute IPMI server ping.
jx.attribute.read	Read JMX attribute information.
jx.server.ping	Execute JMX server ping.
jx.notification.subscribe	Subscribe to JMX notifications.
li.license.read	Read license information.
li.licenses.read	Read license information.
lo.log.read	Read log information.
lo.log.write	Write log information.
lo.logs.read	Read log information.
mb.enterpriseid.read	Read MIB enterprise ID information.
mb.enterpriseids.read	Read MIB enterprise ID information.
mb.oid.read	Read MIB OID information.
mb.trap.read	Read MIB trap information.
ms.query	Query information from message database (MSGDB).
ms.request.create	Create request in message database (MSGDB).
ms.request.delete	Delete request from message database (MSGDB).
ms.request.read	Read request from message database (MSGDB).
ms.request.update	Update request in message database (MSGDB).
ms.requests.read	Read requests from message database (MSGDB).
nf.receive	Receive notification.
nw.ping.check	Execute ping request.

nw.tcp.check	Execute TCP request.
nw.udp.check	Execute UDP request
op.discovery.execute	Execute discovery request.
op.discovery.finish	Execute end of discovery request.
op.discovery.start	Execute start of discovery request.
op.discovery.stop	Execute stop of discovery request.
pd.data.collect	Collect data from agents.
pd.data.export	Export information from performance database (PERFDB).
pd.message	Process message containing performance data (PERFDB).
pd.report.create	Create report from information in performance database (PERFDB).
pd.report.convert	Convert data to report in performance database (PERFDB).
pd.report.export	Export report from performance database (PERFDB).
pd.statistics.export	Export statistics information from performance database (PERFDB).
pd.statistics.read	Read statistics information from performance database (PERFDB).
pd.status.collect.read	Read data collection status information.
ru.actions.read	Read rule actions.
ru.assert	Assert rule facts.
ru.rule.read	Read rule information.
ru.rulefile.read	Read rule file information.
ru.rules.read	Read rule information.
sn.return.address.read	Read SNMP return address.
sn.snmp.get	Execute SNMP GET request.
sn.snmp.getnext	Execute SNMP GETNEXT request.
sn.trap.log	Execute SNMP trap log request.
sn.trap.read	Execute SNMP trap read request.
sn.trap.mappings.clear	Clear SNMP trap mappings.
sn.trap.write	Execute SNMP trap write request.
sn.trap.walk	Execute SNMP walk request.
so.execute	Execute SOAP request.
sr.run.	Execute script.
ss.execute	Execute SSH request.
st.execute	Execute Service Tag request.
sv.availability	Execute availability request.
sv.configuration	Execute configuration data request.

sv.fault	Execute fault monitoring request.
sv.health	Execute health monitoring request.
sv.performance	Execute performance monitoring request.
sv.statistics	Execute statistics monitoring request.
sv.vacuum	Execute database vacuum request.
wm.execute	Execute WMI request.
wm.get	Execute WMI GET request.
wm.ping	Execute WMI PING request.
wm.query	Execute WMI QUERY request.

**Table 4.6-2: Manager Roles**

Role Name	Role Description
Users	Include all user roles.
au.pending.cancel	Cancel pending actions.
au.pending.run	Run pending actions.
au.schedule.cancel	Cancel automation schedule.
au.schedule.create	Create automation schedule.
au.schedule.delete	Delete automation schedule.
au.schedule.execute	Execute automation schedule.
au.schedule.update	Update automation schedule.
cm.arguments.update	Update operation arguments in CMDB.
cm.asset.assoc.types.update	Update asset association types in CMDB.
cm.asset.assocs.update	Update asset associations in CMDB.
cm.asset.attributes.update	Update asset attributes in CMDB.
cm.asset.delete	Delete asset in CMDB.
cm.asset.operation.update	Update asset operation in CMDB.
cm.asset.operations.update	Update asset operations in CMDB.
cm.asset.types.update	Update asset types in CMDB.
cm.asset.rename	Rename asset in CMDB.
cm.asset.update	Update asset in CMDB.
cm.assets.create	Create assets in CMDB.
cm.assets.import	Import assets into CMDB.
cm.assets.merge	Merge assets in CMDB.
cm.assets.update	Update assets in CMDB.

cm.commands.update	Update operation commands in CMDB.
cm.credential.copy	Copy credential in CMDB.
cm.credential.create	Create credential in CMDB.
cm.credential.delete	Delete credential in CMDB.
cm.credential.update	Update credential in CMDB.
cm.jobs.update	Update operation jobs in CMDB.
cm.license.update	Update license in CMDB.
cm.metric.update	Update metric in CMDB.
cm.operation.types.update	Update operation types in CMDB.
cm.operations.update	Update operations in CMDB.
cm.rule.create	Create rule in CMDB.
cm.rule.delete	Delete rule in CMDB.
cm.rule.update	Update rule in CMDB.
cm.rules.copy	Copy rules in CMDB.
cm.rules.delete	Delete rules in CMDB.
cm.rules.purge	Purge rules in CMDB.
cm.rules.undelete	Undelete rules from CMDB.
cm.snapshot.create	Create snapshot.
cm.tag.create	Create tag in CMDB.
cm.tag.delete	Delete tag in CMDB.
cm.tag.update	Update tag in CMDB.
cm.tag.type.create	Create tag type in CMDB.
cm.tag.type.delete	Delete tag type in CMDB.
cm.tag.type.update	Update tag type in CMDB.
cm.variables.update	Update operation variables in CMDB.
jx.attribute.write	Write JMX attribute information.
jx.operation.execute	Execute JMX operation.
li.license.update	Update license information.
li.licenses.update	Update license information.
mb.enterpriseid.create	Create MIB enterprise ID information.
mb.oid.create	Create MIB OID information.
mb.trap.copy	Copy MIB trap information.
mb.trap.create	Create MIB trap information.
mb.trap.delete	Delete MIB trap information.
mb.trap.update	Update MIB trap information.

ms.event.update	Update event in message database.
op.operation.execute	Execute operation request.
op.operations.execute	Execute operations request.
pd.agents.update	Update agent in performance database (PERFDB).
pd.asset.rename	Rename asset in performance database (PERDB).
pd.assets.merge	Merge assets in performance database (PERFDB).
pd.statistics.collect	Collect statistics information in performance database (PERFB).
ru.actions.test	Test rule actions.
ru.filter.delete	Delete rule filter.
ru.pending.clear	Clear pending rule actions.
ru.pending.run	Run pending rule actions.
ru.rulebase.export	Export rule base information.
ru.rulebase.import	Import rule base information.
ru.rule.create	Create rule information.
ru.rule.delete	Delete rule information.
ru.rule.disable	Disable rule.
ru.rule.enable	Enable rule.
ru.rule.update	Update rule information.
ru.rules.copy	Copy rule information.
ru.rules.delete	Delete rule information.
ru.rules.disable	Disable rules.
ru.rules.enable	Enable rules.
ru.rules.purge	Purge rules.
ru.rules.undelete	Undelete rules.
sn.snmp.set	Execute SNMP SET request.
sv.database.reorg	Execute database reorganization request.

**Table 4.6-3: Administrator Roles**

<b>Role Name</b>	<b>Role Description</b>
Managers	Include all manager roles.
cf.configuration.properties.update	Update configuration properties.
cf.configurations.update	Update configurations.
cf.module.disable	Disable module.

cf.module.enable	Enable module.
cf.pending.requests.create	Create pending requests.
cf.pending.requests.read	Read pending requests.
cm.configuration.create	Create configuration asset in CMDB.
cm.configuration.update	Update configuration asset in CMDB.
cm.namespaces.update	Update namespaces in CMDB.
cm.revision.history.delete	Delete revision history in CMDB.
cm.revision.history.write	Write revision history of CMDB.
cm.rulebase.create	Create rule base in CMDB.
cm.rulebase.delete	Delete rule base in CMDB.
cm.rulebase.update	Update rule base in CMDB.
de.module.update	Deploy update module.
in.container.register	Register container.
lo.log.add	Add log file to monitor.
lo.log.remove	Remove log file to monitor.
lo.revision.history.write	Write revision history to log file.
mb.revision.history.write	Write revision history of MIBDB.
ru.rulebase.create	Create rule base information.
ru.rulebase.delete	Delete rule base information.
ru.rulebase.update	Update rule base information.

## 5 License Management

After installation *Neuron Management Suite* components are licensed for a trial period of 14 days. In order to continue using the product a license is required. The following sections describe how to list, add, and remove licenses using the *License Manager* utility. This utility must be executed by the root user.

### 5.1 Listing Users

In order to list the licenses of all components invoke the utility as follows:

```
# [BASEDIR]/bin/LicenseManager -user <user> -list
```

where user is a Neuron Management Server user.

Example output:

```
#!/opt/HMF/bin/LicenseManager -user admin -list
Password for user admin:
Component                Version  Description                ... Expiration
-----
BEMAdapter               1.1.9   Adapter for forwardi ... permanent
CiscoBaseModule          1.0.2   Module for basic man ... permanent
ConfigurationManager     1.15.0  Manage and view conf ... permanent
EmailAdapter             1.15.0  Adapter for forwardi ... permanent
EventManager             1.15.0  Manage and view even ... permanent
GenericAdapter           1.15.0  Adapter for forwardi ... permanent
GridEngineModule         1.0.2   Module for monitorin ... permanent
HardwareModule           1.0.9   Module for basic man ... permanent
HealthManager           1.15.0  Manage and view heal ... permanent
InventoryManager         1.15.0  Manage and view inve ... permanent
LinuxBaseModule          1.0.4   Module for basic man ... permanent
MessageLogger            1.15.0  Module for logging m ... permanent
NetcoolAdapter           1.1.8   Adapter for forwardi ... permanent
OpenViewAdapter          1.1.8   Adapter for forwardi ... permanent
OpsCenterAdapter         1.15.0  Adapter for integrat ... permanent
PerformanceManager       1.6.0   Manage and view perf ... permanent
SolarisBaseModule        1.0.5   SolarisBaseModule     ... permanent
SolarisVirtualizationModule 1.0.3   Module for virtualiz ... permanent
SunMCAdapter             1.1.11  Adapter for processi ... permanent
SunMCServerAdapter       1.1.11  The Sun MC Server Ad ... permanent
TivoliAdapter            1.1.8   Adapter for forwardi ... permanent
WindowsBaseModule        1.0.2   Module for basic dis ... permanent
```

## 5.2 Adding Licenses

In order to add licenses from a file provided by Halcyon invoke the utility as follows:

```
# [BASEDIR]/bin/LicenseManager -user <user> -add <file>
```

where user is the name of a Neuron Management Server manager or administrator and file is the complete path to a license file.

Example output:

```
#/opt/HMF/bin/LicenseManager -user admin -add /tmp/halcyon-licenses.dat  
Password for user admin:  
License file /tmp/halcyon-licenses.dat was successfully added.
```

## 5.3 Removing Licenses

In order to remove the license for a specific component invoke the utility as follows:

```
# [BASEDIR]/bin/LicenseManager -user <user> -remove <component>
```

where user is the name of a Neuron Management Server manager or administrator and component is a licensed component (see section 5.1).

Example output:

```
#/opt/HMF/bin/LicenseManager -user admin -remove BEMAdapter  
Password for user admin:  
License for BEMAdapter was successfully removed.
```

## 5.4 License File

The *Neuron Management Server* exports current license information after adding or removing licenses to the file [LOCALDIR]/conf/halcyon-licenses.dat.