

# **JSON API Guide**

## **Privileged User Manager**

**June 2013**



## Legal Notice

NetIQ Product Name is protected by United States Patent No(s): nnnnnnnn, nnnnnnnn, nnnnnnnn.

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, NETIQ CORPORATION PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU.

For purposes of clarity, any module, adapter or other similar material ("Module") is licensed under the terms and conditions of the End User License Agreement for the applicable version of the NetIQ product or software to which it relates or interoperates with, and by accessing, copying or using a Module you agree to be bound by such terms. If you do not agree to the terms of the End User License Agreement you are not authorized to use, access or copy a Module and you must destroy all copies of the Module and contact NetIQ for further instructions.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of NetIQ Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of NetIQ Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

**© 2013 NetIQ Corporation and its affiliates. All Rights Reserved.**

For information about NetIQ trademarks, see <https://www.netiq.com/company/legal/>.

---

# Contents

<b>About This Guide</b>	<b>5</b>
<b>1 JSON API</b>	<b>7</b>
1.1 SPF.Util Interface	7
1.1.1 callModuleEx	8
1.1.2 callMasterEx	8
1.1.3 callModule	8
1.1.4 callMaster	9
1.1.5 callModuleA	9
1.1.6 callMasterA	10
1.1.7 listConsoles	11
1.1.8 listAllConsoles	11
1.1.9 getVersion	11
1.1.10 lookupModule	12
1.1.11 lookupService	12
1.1.12 serviceInfo	12
1.2 Calling the Framework User Manager Module (Auth)	13
1.2.1 Logon	13
1.2.2 listUsers	15
1.2.3 listGroups	18
1.2.4 userInfo	19
1.2.5 groupInfo	20
1.2.6 addUser	20
1.2.7 addGroup	21
1.2.8 delUser	21
1.2.9 delGroup	21
1.2.10 modUser	22
1.2.11 modGroup	23
1.3 Calling the Audit Module	23
1.3.1 listLogs	23
1.3.2 getLogEntries	24
1.3.3 getSession	24
1.4 Calling the Command Control Module	25
1.4.1 getEntries	26
1.4.2 evalTemplate	27
1.4.3 Add User Group	27
1.4.4 Get User Group Entity	29
1.4.5 Modify Account Group	30
1.4.6 Add Account Group	31
1.4.7 Get Account Group Entity	31
1.4.8 Add Command	32
1.4.9 Get Command Entity	32
1.4.10 Add Host Group	34
1.4.11 Get Host Group Entity	34
1.4.12 Add Rule	35
1.4.13 Get Rule Entity	36
1.4.14 Modify Command	37
1.4.15 Modify Rule Condition	38
1.4.16 Modify Host Group	39
1.4.17 Modify Rule Condition	39
1.4.18 Modify User Group	41
1.4.19 Get Command Control Tree	42

1.5	Miscellaneous Functions .....	46
1.5.1	List Updates.....	47
1.5.2	Install Package .....	48
1.5.3	Promote Manager Module.....	48
1.5.4	Change Agent Address .....	49
1.5.5	Create Framework Host.....	49
1.5.6	List Framework Hosts .....	49
1.5.7	List Package Manager Packages .....	50

---

# About This Guide

This API provides an interface for AJAX and architectures that allow JSON HTTP requests to be made, such as Java.

## **Audience**

This guide is intended for users who manage the Privileged User Manager product.

## **Feedback**

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.



---

# 1 JSON API

JSON requests are performed by making an HTTP POST with a Content-Type of `application/javascript` to the `SPF.Util` module to the PUM Administration Manager via the `https://localhost/SPF.Util` URL.

This API provides an interface for AJAX and architectures such as Java that allow JSON HTTP requests to be made.

- ♦ [Section 1.1, “SPF.Util Interface,” on page 7](#)
- ♦ [Section 1.2, “Calling the Framework User Manager Module \(Auth\),” on page 13](#)
- ♦ [Section 1.3, “Calling the Audit Module,” on page 23](#)
- ♦ [Section 1.4, “Calling the Command Control Module,” on page 25](#)
- ♦ [Section 1.5, “Miscellaneous Functions,” on page 46](#)

## 1.1 SPF.Util Interface

You can make a number of different types of requests to `SPF.Util`. All calls to `SPF.Util` have the following form:

```
{
  method: [One of the SPF.Util methods as listed below],
  params: {
    [Method-specific arguments]
  }
}
```

- ♦ [Section 1.1.1, “callModuleEx,” on page 8](#)
- ♦ [Section 1.1.2, “callMasterEx,” on page 8](#)
- ♦ [Section 1.1.3, “callModule,” on page 8](#)
- ♦ [Section 1.1.4, “callMaster,” on page 9](#)
- ♦ [Section 1.1.5, “callModuleA,” on page 9](#)
- ♦ [Section 1.1.6, “callMasterA,” on page 10](#)
- ♦ [Section 1.1.7, “listConsoles,” on page 11](#)
- ♦ [Section 1.1.8, “listAllConsoles,” on page 11](#)
- ♦ [Section 1.1.9, “getVersion,” on page 11](#)
- ♦ [Section 1.1.10, “lookupModule,” on page 12](#)
- ♦ [Section 1.1.11, “lookupService,” on page 12](#)
- ♦ [Section 1.1.12, “serviceInfo,” on page 12](#)

## 1.1.1 callModuleEx

The `callModuleEx` method performs a synchronous request to an NPUM Agent for a module. Requests to `callModuleEx` have the following form:

```
{
  "method" : "callModuleEx",
  "params" : {
    "pkt" : {
      "module" : [Name of a module to call],
      "method" : [Method to call on above module],
      [Module/Method specific arguments]
    }
    "svc_name" : [Optional: Registered agent name of an agent to call],
    "svc_object" : [Optional] {
      "host" : [hostname],
      "port" : [port number]
    }
  }
}
```

A request to `callModuleEx` can have an optional `svc_name` or `svc_object` argument defined to force the call to be made to a specific NPUM Agent. When neither `svc_name` nor `svc_object` are defined, the call is made to the nearest agent as defined in the host hierarchy.

The results from the module call are returned to the caller in the following response:

```
{
  "svc":[name of agent that processed the request],
  "vrm":[version of the NPUM Framework],
  [Result from module]
}
```

## 1.1.2 callMasterEx

The `callMasterEx` method performs a synchronous request to the Primary NPUM agent for a module. Requests to `callMasterEx` have the following form:

```
{
  "method" : "callMasterEx",
  "params" : {
    "module" : [Name of a module to call],
    "method" : [Method to call on above module],
    [Module/Method-specific arguments]
  }
}
```

The results from the module call are returned to the caller in the following response:

```
{
  "svc":[name of agent that processed the request],
  "vrm":[version of the NPUM Framework],
  [Result from module]
}
```

## 1.1.3 callModule

Deprecated. See [callModuleEx](#).

The `callModule` method behaves like `callModuleEx`, except when a non-successful response is received from the module. Non-successful responses have the following form:



```

{
  "error" : {
    "message":[Error message returned from the module],
    "code":400
  }
  "spf": {
    "svc":[name of agent that processed the request],
    "vrm":[version of the NPUM Framework],
    [The actual result returned from the module]
  }
}

```

## 1.1.4 callMaster

Deprecated. See [callMasterEx](#).

The `callMaster` method behaves like `callMasterEx`, except when a non-successful response is received from the module. Non-successful responses have the following form:

```

{
  "error" : {
    "message":[Error message returned from the module],
    "code":400
  }
  "spf": {
    "svc":[name of agent that processed the request],
    "vrm":[version of the NPUM Framework],
    [The actual result returned from the module]
  }
}

```

## 1.1.5 callModuleA

The `callModuleA` method performs an asynchronous request to a module on an NPUM Agent. Requests to `callModuleA` have the following form:

```

{
  "method" : "callModuleA",
  "params" : {
    "pkt" : {
      "module" : [Name of a module to call],
      "method" : [Method to call on above module],
      [Module/Method specific arguments]
    }
    "svc_name" : [Optional: Registered agent name of an agent to call],
    "svc_object" : { [Optional]
      "host" : [hostname],
      "port" : [port number]
    }
  }
}
}

```

A request to `callModuleA` can have an optional `svc_name` or `svc_object` argument defined to force the call to be made to a specific NPUM Agent. When neither `svc_name` nor `svc_object` are defined, the call is made to the nearest agent as defined in the host hierarchy.

The result from `callModuleA` is a unique request ID that can be used to poll for the response:

```

{
  "svc":[name of agent that processed the request],
  "vrm":[version of the NPUM Framework],
  "AsyncReq": {
    "id":[Unique request id]
  }
}

```

To retrieve the subsequent response, poll the admin asyncResponse method, passing the request ID:

```
{
  "method": "callModuleEx",
  "svc_name": [svc that processed original request],
  "params": {
    "pkt": {
      "module": "admin",
      "method": "asyncResponse",
      "AsyncReq": {
        "id": [Unique request id]
      }
    }
  }
}
```

While the request is still pending, the response status from asyncResponse is either 100 or 202. When the request has finished processing, the asyncResponse method returns the actual response from the original request:

```
{
  "svc": [name of agent that processed the request],
  "vrm": [version of the NPUM Framework],
  [Result from module]
}
```

## 1.1.6 callMasterA

The callMasterA method performs an asynchronous request to the primary module on an NPUM Agent. Requests to callMasterA have the following form:

```
{
  "method" : "callMasterA",
  "params" : {
    "module" : [Name of a module to call],
    "method" : [Method to call on above module],
    [Module/Method specific arguments]
  }
}
```

The result from callMasterA is a unique request ID that can be used to poll for the response:

```
{
  "svc": [name of agent that processed the request],
  "vrm": [version of the NPUM Framework],
  "AsyncReq": {
    "id": [Unique request id]
  }
}
```

To retrieve the subsequent response, poll the admin asyncResponse method, passing the request ID:

```
{
  "method": "callModuleEx",
  "svc_name": [svc that processed original request],
  "params": {
    "pkt": {
      "module": "admin",
      "method": "asyncResponse",
      "AsyncReq": {
        "id": [Unique request id]
      }
    }
  }
}
```

While the request is still pending, the response status from `asyncResponse` is either 100 or 202. When the request has finished processing, the `asyncResponse` method returns the actual response from the original request:

```
{
  "svc": [name of agent that processed the request],
  "vrm": [version of the NPUM Framework],
  [Result from module]
}
```

### 1.1.7 listConsoles

Requests a list of consoles installed on the NPUM Administration manager that the current user is authorized to see.

```
{
  "method": "listConsoles"
  "params": {
    "identity": [authentication token]
  }
}
```

The response contains all consoles installed on the NPUM Administration manager.

```
{
  "Console": [
    [Array of Console definitions]
  ]
}
```

### 1.1.8 listAllConsoles

Requests a list of all consoles installed on the NPUM Administration manager.

```
{
  "method": "listAllConsoles"
}
```

The response contains all consoles installed on the NPUM Administration manager.

```
{
  "Console": [
    [Array of Console definitions]
  ]
}
```

### 1.1.9 getVersion

Called to retrieve the framework version:

```
{
  "method": "getVersion"
}
```

Returns the framework version:

```
"2.1.5"
```

## 1.1.10 lookupModule

Called to locate services that have a given module installed:

```
{
  "method": "lookupModule"
  "params": {
    "module": [name of module to look up]
  }
}
```

Returns a list of agents that have the module installed:

```
{
  "Service": [
    [Array of agent service definitions]
  ]
}
```

## 1.1.11 lookupService

Called to retrieve information about a specific NPUM Agent service:

```
{
  "method": "lookupService"
  "params": {
    "service": [agent name to look up]
  }
}
```

Returns an agent service definition:

```
{
  "Service": {
    "name": [agent name],
    "host": [host name],
    "port": [port number],
    "id": [unique agent id]
  }
}
```

## 1.1.12 serviceInfo

Called to retrieve service information for a specific NPUM Agent service:

```
{
  "method": "serviceInfo",
  "params": {
    "service": {
      "host": [host name],
      "port": [port number],
    }
  }
}
```

Returns information about the NPUM Agent service:

```

{
  "name":[agent name],
  "host":[host name],
  "port":[port number],
  "id":[service id],
  "ou":[location in domain tree],
  "status":[online status],
  "uptime":[time process has been running in seconds],
  "utc":[time on the host in UTC],
  "sessions":[number of active sessions],
  "tasks":[number of active tasks],
  "sysmachine":[hardware type],
  "sysname":[operating system name]
  "sysversion":[operating system version],
  "Root": {
    "free":[free bytes on filesystem],
    "path":[NPUM installation directory],
    "avail":[available bytes on filesystem],
    "size":[size of the filesystem]
  },
  "Memory": {
    "heap":[approximate heap usage of the process]
  },
  "Module": [
    [Array of installed modules]
  ]
  "CertInfo": [
    [Array of certificate information for certificate hierarchy]
  ]
}

```

## 1.2 Calling the Framework User Manager Module (Auth)

The Framework User Manager (auth) module provides the following interface to query and update the user configuration:

- ◆ [Section 1.2.1, “Logon,” on page 13](#)
- ◆ [Section 1.2.2, “listUsers,” on page 15](#)
- ◆ [Section 1.2.3, “listGroups,” on page 18](#)
- ◆ [Section 1.2.4, “userInfo,” on page 19](#)
- ◆ [Section 1.2.5, “groupInfo,” on page 20](#)
- ◆ [Section 1.2.6, “addUser,” on page 20](#)
- ◆ [Section 1.2.7, “addGroup,” on page 21](#)
- ◆ [Section 1.2.8, “delUser,” on page 21](#)
- ◆ [Section 1.2.9, “delGroup,” on page 21](#)
- ◆ [Section 1.2.10, “modUser,” on page 22](#)
- ◆ [Section 1.2.11, “modGroup,” on page 23](#)

### 1.2.1 Logon

Authenticates to the NPUM Framework:

```

{
  "module": "auth",
  "method": "login",
  "Credentials": {
    "name": [user name],
    "passwd": [user password],
  }
}

```

The response contains error or logon information, including the authentication token needed for future calls.

Error:

```

{
  "message": "Invalid user name or password",
  "status": 401,
  "svc": "pmmaster"
}

```

Successful logon:

```

{
  "message": "User user@pmmaster(192.168.1.1) successfully authenticated",
  "status": 0,
  "svc": "pmmaster",
  "User": {
    "name": "admin",
    "ACT_LAST_SUCC_LOGON": {
      "value": 1325842225
    },
    "ACT_LAST_UNSUCC_LOGON": {
      "value": 1325841725
    }
  },
  "Identity": {
    "content": "eJzj8ExJzSvJLk1kyOd0LC3JCMnPTs1jkGbJS8xNZU1Myc3My+TIzEtMLsksS92h4MlWUly
SmFvQ8mPWv5Ji7qLU3PyS1PiM\
OISPkNLiZlDMwsg1jM0sijmLEhNLQLLsCYVZSZnJ7IE5eekAg0uAlKMWgwofIhFDEAnBGem5yWWlBalMlQ
xF2emNzLo2hzVPHsmaUmRiP1tY92dftumb7WI360tzVIheFtYYaUe3\
Jagx2c69yt7Q4zGC6Zf+ru7Fs+dzft2b3tU1bbK44lOjwzNE\
/W7689Nw5N\
/esU2JyWc80St9\
lN+Q9LA\
5Xas5LppVy7cnHlmcuRHQ43\
9WK3qp+1B1fvvrTo4UaHgCWndXKpYFVNmikjkfr83RyZcWBOfYOHEHWKtdKjwga8DAwCyh3\
/1"
  },
  "AuthToken": {
    "name": "admin",
    "inactive": 7200,
    "tstamp": 1325842292,
    "remote_host": "192.168.1.1",
    "peer_host": "pmmaster",
    "Role": [
      { "role": "*" },
      { "role": "admin" }
    ]
  }
}

```

The Identity Content attribute is passed into any functions that require logon security rights. For example:

```

{
  "method" : "callModuleEx",
  "params" : {
    "pkt" : {
      "module": "distrib",
      "method": "listUpdates",
      "uid": <Identity Content>
    }
  }
}

```

## 1.2.2 listUsers

Provides a list of the users defined in the Framework User Manager database:

```

{
  "module": "auth",
  "method": "listUsers"
  "uid": <Identity Content>,
  "User": {
    "name": [Optional user name to search for (glob)],
    "group": [Optional Boolean flag to include group membership in
list],
    "role": [Optional Boolean flag to include roles in list],
    "brief": [Optional Boolean flag to request brief details in list]
  }
}

```

The optional arguments can be used to look up and control the level of detail returned in the list.

The framework user manager responds with a list of users.

```

{
  "User": [
    {
      "name": [user name],
      [user attributes]
    }, ...
  ],
  "vrm": [framework version]
}

```

### User Attributes

---

ACT_ACCESS_AUDIT	'Host Access Control Audit settings'
ACT_ACCESS_ALLOW	'Host Access Control Allow list'
ACT_ACCESS_DENY	'Host Access Control Deny list'
ACT_ACCESS_ORDER	'Host Access Control Order'
ACT_COMMENT	'Account description'
ACT_DESC	'Account description'
ACT_DISABLED	'Account disabled'
ACT_EMAIL	'Account email address'
ACT_FULL_NAME	'Account full name'
ACT_GROUPS	'Account group membership'
ACT_INACTIVITY	'Account inactivity timeout in seconds'

---

---

ACT_LAST_SUCC_LOGON	'Last successful logon seconds since epoch'
ACT_LAST_UNSUCC_LOGON	'Last unsuccessful logon seconds since epoch'
ACT_LOCKOUT_NUM	'Account lockout count'
ACT_LOGON_SCRIPT	'Perl script to execute after logon'
ACT_MAPS	'Account mappings for native and LDAP logon'
ACT_MOTD	'Message of the day'
ACT_NUM_BADLOGONS	'Number of bad logons'
ACT_PASSWD	'Password'
ACT_ROLES	'User permissions'
ACT_STAFF_ID	'Account identifier'
ACT_SUPER	'Super user'
ACT_TELEPHONE	'Account telephone number'
ACT_UNUSED_DELETE	'Account unused limit (seconds) deletes account '
ACT_UNUSED_LIMIT	'Account unused limit (seconds) locks account'
PWD_EXPIRED	'Password expired'
PWD_FMT_MIN_ALPHA	'Minimum alpha characters'
PWD_FMT_MIN_NUMERIC	'Minimum numeric characters'
PWD_HISTORY	'Password history'
PWD_HISTORY_NUM	'Number of passwords in history'
PWD_LAST_CHG	'Password last changed'
PWD_MAXAGE	'Password maximum age'
PWD_MINIMUM_LENGTH	'Password minimum length'

---

Example of output:

```
{
  "User": [
    {
      "name": "admin",
      "ACT_COMMENT": {
        "value": "Administration Account"
      },
      "PWD_MAXAGE": {
        "value": false
      },
      "ACT_UNUSED_LIMIT": {
```



```

    "value":false
  },
  "ACT_UNUSED_DELETE":{
    "value":false
  },
  "ACT_LOCKOUT_NUM":{
    "value":0
  },
  "ACT_CREATED":{
    "value":1318932953
  },
  "ACT_PASSWD":{
    "value":"$apr1$hu7JjT8E$6GMmofFJIjFkVQyovaksn."
  },
  "PWD_EXPIRED":{
    "value":false
  },
  "PWD_LAST_CHG":{
    "value":1318933625
  },
  "PWD_HISTORY":{
    "value":{
      "$apr1$Q.BUyimQ$6n2ayUx9tFqrEp.ixnW07." : {
        }
      }
    },
  "ACT_DESC":{
    "value":"Admin User"
  },
  "ACT_LAST_UNSUCC_LOGON":{
    "value":1319100610
  }

```

```

    },
    "ACT_LAST_SUCC_LOGON": {
        "value": 1319101930
    }
},
]
}

```

### 1.2.3 listGroups

Provides a list of the groups defined in the Framework User Manager database:

```

{
    "module": "auth",
    "method": "listGroups",
    "uid": <Identity Content>,
    "Group": {
        "name": [Optional group name to search for (glob)],
        "user": [Optional boolean flag to include group membership in list],
        "brief": [Optional boolean flag to request brief details in list]
    }
}

```

The optional arguments can be used to look up and control the level of detail returned in the list.

The framework user manager responds with a list of groups:

```

{
    "Group": [
        {
            "name": [group name],
            [group attributes]
        }, ...
    ],
    "vrm": [framework version]
}

```

#### Group Attributes

---

GRP_COMMENT	'Group Comment'Group
GRP_MEMBER	'Group members'
GRP_MGR	'Group manager's name'
GRP_MGR_EMAIL	'Group manager's email address'
GRP_MGR_TEL	'Group manager's telephone number'
GRP_ROLES	'Group permissions'
GRP_SUPER	'Super group flag'
SUB_GROUPS	'Sub groups'

---

Example output:

```

{
  "Group": [
    {
      "name": "admin",
      "GRP_COMMENT": {
        "value": "Global Administrators"
      }
    },
    {
      "name": "testgroup",
      "GRP_COMMENT": {
        "value": "comment"
      },
      "GRP_MGR": {
        "value": "manager"
      },
      "GRP_MGR_TEL": {
        "value": "111222333"
      },
      "GRP_MGR_EMAIL": {
        "value": "manager@foo.com"
      }
    }
  ]
}

```

## 1.2.4 userInfo

Provides a mechanism to query details for a user account:

```

{
  "module": "auth",
  "method": "userInfo",
  "uid": <Identity Content>,

  "User": {
    "name": [Required user name]
  },
  "Attrib": { [Optional empty Object to flag Attributes are required]
  }
}

```

The optional `Attrib` argument can be used to look up the user's attributes.

The framework user manager responds with the user details:

```
{
  "User": {
    "name": [user name],
    [user attributes]
  },
  "Attrib": {
    [list of available user attributes with default values where defined]
  }
  "vrm": [framework version]
}
```

## 1.2.5 groupInfo

Provides a mechanism to query details for a group account:

```
{
  "module": "auth",
  "method": "groupInfo",
  "uid": <Identity Content>,
  "Group": {
    "name": [Required group name]
  }
  "Attrib": { [Optional empty Object to flag Attributes are required]
  }
}
```

The optional Attrib argument can be used to look up the group's attributes.

The framework user manager responds with the group details:

```
{
  "Group": {
    "name": [user name],
    [group attributes]
  },
  "Attrib": {
    [list of available group attributes with default values where defined]
  }
  "vrm": [framework version]
}
```

## 1.2.6 addUser

Provides a mechanism to add a new user to the framework user manager database:

```
{
  "module": "auth",
  "method": "addUser",
  "uid": <Identity Content>,
  "User":
  {
    "name": [user name],
    "passwd": [user password],
  }
}
```

The Framework Access Manager responds with the status of the request. If the response status is present and non-zero, the message attribute contains the error message.

```
{
  "status": [Status (non zero is error)]
  "message": [error message]
  "vrm": [framework version]
}
```

## 1.2.7 addGroup

Provides a mechanism to add a new group to the framework user manager database:

```
{
  "module": "auth",
  "method": "addUser",
  "uid": <Identity Content>,
  "User": {
    "name": [user name],
    "passwd": [user password],
  }
}
```

The framework access manager responds with the status of the request. If the response status is present and non-zero, the message attribute contains the error message.

```
{
  "status": [Status (non zero is error)]
  "message": [error message]
  "vrm": [framework version]
}
```

## 1.2.8 delUser

Provides a mechanism to delete a user from the framework user manager database:

```
{
  "module": "auth",
  "method": "delUser",
  "uid": <Identity Content>,
  "User": {
    "name": [user name],
  }
}
```

The user can optionally be an array. If this is the case, all the listed users are deleted. The framework access manager responds with the status of the request. If the response status is present and non-zero, the message attribute contains the error message.

```
{
  "status": [Status (non zero is error)]
  "message": [error message]
  "vrm": [framework version]
}
```

## 1.2.9 delGroup

Provides a mechanism to delete a group from the framework user manager database:

```

{
  "module": "auth",
  "method": "delGroup",
  "uid": <Identity Content>,
  "Group": {
    "name": [group name],
  }
}

```

The group can optionally be an array. If this is the case, all the listed groups are deleted. The framework access manager responds with the status of the request. If the response status is present and non-zero, the message attribute contains the error message.

```

{
  "status": [Status (non zero is error)]
  "message": [error message]
  "vrm": [framework version]
}

```

## 1.2.10 modUser

Provides a mechanism to modify a user in the framework user manager database:

```

{
  "module": "auth",
  "method": "modUser",
  "uid": <Identity Content>,
  "User": {
    "name": [user name],
    [user attributes to update]
  }
}

```

The attributes to update are passed as JSON objects with the name of the Account attribute with an action attribute of "set".

For example, to disable an account, the request looks like the following:

```

{
  "module": "auth",
  "method": "modUser",
  "uid": <Identity Content>,
  "User": {
    "name": "foo",
    ACT_DISABLED: {
      "action": "set",
      "value": true
    },
    ACT_PASSWD: {
      "action": "set",
      "value": "foobar123"
    }
  }
}

```

The framework access manager responds with the status of the request. If the response status is present and non-zero, the message attribute contains the error message

```

{
  "status": [Status (non zero is error)]
  "message": [error message]
  "vrm": [framework version]
}

```

## 1.2.11 modGroup

Provides a mechanism to modify a group in the framework user manager database:

```
{
  "module": "auth",
  "method": "modGroup",
  "uid": <Identity Content>,
  "Group": {
    "name": [group name],
    [group attributes to update]
  }
}
```

The Attributes to update are passed as JSON objects with the name of the Account attribute with an action attribute of "set".

For example, to add user "foo" to group "bar," the request looks like the following:

```
{
  "module": "auth",
  "method": " modGroup ",
  "uid": <Identity Content>,
  "User": {
    "name": "bar",
    GRP_MEMBER: {
      "value": {
        "foo": {
          "action": "set"
        }
      }
    }
  }
}
```

The framework access manager responds with the status of the request. If the response status is present and non-zero, the message attribute contains the error message.

```
{
  "status": [Status (non zero is error)]
  "message": [error message]
  "vrm": [framework version]
}
```

## 1.3 Calling the Audit Module

The Audit Manager (audit) module provides the following interface to search the recorded audit information.

- ♦ [Section 1.3.1, "listLogs," on page 23](#)
- ♦ [Section 1.3.2, "getLogEntries," on page 24](#)
- ♦ [Section 1.3.3, "getSession," on page 24](#)

### 1.3.1 listLogs

Provides a list of the audit log files available on the audit manager:

```

{
  "module": "audit",
  "method": "listLogs",
  "uid": <Identity Content>,
  "DbList": {
    "dbgrp": [option database group to restrict the list]
  }
}

```

The optional `DbList.dbgrp` can be used to list only a certain type of audit logs, such as `"cmdctrl"`.

The Audit manager responds with a list of log files:

```

{
  "svc": [agent name that processed this request],
  "DbList": {
    "cmdctrl": [
      {
        "keyid": [encryption key identifier],
        "rolltime": [time log file was rolled over],
        "uuid": [unique log file identifier],
        "fname": [file name],
        "online": [online status]
      }
    ]
  },
  "vrm": [framework version]
}

```

### 1.3.2 getLogEntries

Provides a mechanism to query the contents of an audit log file:

```

{
  "module": "audit",
  "method": "getLogEntries",
  "uid": <Identity Content>,
  "Results": {
    "fname": [audit log file to query],
    "Result": {
      "content": [SQL query]
    }
  }
}

```

The Audit manager responds with the results of the query:

```

{
  "svc": [agent name that processed this request],
  "Results": {
    "Result": [
      {
        [results of the SQL query passed in]
      }
    ]
  },
  "vrm": [framework version]
}

```

### 1.3.3 getSession

Requests the keystroke records for a given command control session:



```

{
  "module": "audit",
  "method": "getSession",
    "uid": <Identity Content>,
  "Session": {
    "groupID": [ID identifying the keystroke session],
    "stdin": [pass stdin of 1 to retrieve stdin],
    "count": [defines how many records to return in each request],
    "start": [defines the key to continue searching from],
    "fname": [name of audit log to use],
    "fattach": [name of audit log to attach to continue searching],
    "to_codeset": [codeset to convert characters into],
    "from_codeset": [codeset to convert characters from],
    "term": [terminal type to use to parse control characters]
  }
}

```

The audit manager returns the processed list of keystroke events:

```

{
  "status": [status of request],
  "Session": {
    "svc": [name of NPUM Audit manager that processed this request],
    "groupID": [ID identifying the keystroke session],
    "count": [defines how many records to return in each request],
    "start": [defines the key to continue searching from],
    "fname": [name of audit log to use],
    "fattach": [name of audit log to attach to continue searching],
    "to_codeset": [codeset to convert characters into],
    "from_codeset": [codeset to convert characters from],
    "term": [terminal type to use to parse control characters]
    "Results": [
      "SessionData": {
        "fkey": [audit key],
        "delta": [time of keystroke event],
        "name": [type of keystroke e.g. stdout.output]
        "SessionDatum": [
          "value": [contents of the keystroke event],
          "type": [type of keystroke event e.g. text or code],
          "name": [ the name of the control code]
        ]
      }
    ]
  }
}
"vrm": [framework version]
}

```

## 1.4 Calling the Command Control Module

The Command Control Manager (cmdctrl) module provides the following interface to query the policy configuration. The hierarchical structure of the rules and categories is stored in the CCTree structure. Entity details are stored separately.

- ◆ [Section 1.4.1, “getEntries,” on page 26](#)
- ◆ [Section 1.4.2, “evalTemplate,” on page 27](#)
- ◆ [Section 1.4.3, “Add User Group,” on page 27](#)
- ◆ [Section 1.4.4, “Get User Group Entity,” on page 29](#)
- ◆ [Section 1.4.5, “Modify Account Group,” on page 30](#)
- ◆ [Section 1.4.6, “Add Account Group,” on page 31](#)
- ◆ [Section 1.4.7, “Get Account Group Entity,” on page 31](#)
- ◆ [Section 1.4.8, “Add Command,” on page 32](#)

- ◆ Section 1.4.9, “Get Command Entity,” on page 32
- ◆ Section 1.4.10, “Add Host Group,” on page 34
- ◆ Section 1.4.11, “Get Host Group Entity,” on page 34
- ◆ Section 1.4.12, “Add Rule,” on page 35
- ◆ Section 1.4.13, “Get Rule Entity,” on page 36
- ◆ Section 1.4.14, “Modify Command,” on page 37
- ◆ Section 1.4.15, “Modify Rule Condition,” on page 38
- ◆ Section 1.4.16, “Modify Host Group,” on page 39
- ◆ Section 1.4.17, “Modify Rule Condition,” on page 39
- ◆ Section 1.4.18, “Modify User Group,” on page 41
- ◆ Section 1.4.19, “Get Command Control Tree,” on page 42

## 1.4.1 getEntries

Retrieves configuration information from the command control manager:

```
{
  "module": "cmdctrl",
  "method": "getEntries",
  "uid": <Identity Content>,
  "[CCTree]": {
    "id": [id of cctree]
  }
  "[Entity]": {
    "key": [optional key of entity]
    "name": [wildcard entity name to search for]
  }
}
```

When requesting the CCTree structure, you can select the whole hierarchy or a specific subset of the hierarchy. The relevant IDs for sections of the CCTree structure are shown below:

CCTree ID	Description
0	Base of the CCTree hierarchy
1	Rule hierarchy
2	Account Group hierarchy
3	User Group hierarchy
4	Host Group hierarchy
5	Command hierarchy
6	Script hierarchy
7	Access Times hierarchy
9	Report hierarchy

For example, to retrieve the complete rule hierarchy, you use the following call:

```

{
  "module": "cmdctrl",
  "method": "getEntries",
  "uid": <Identity Content>,
  "Rule": {
    "id": 1
  }
}

```

To find an entity details by name, you can make a call like this:

```

{
  "module": "cmdctrl",
  "method": "getEntries",
  "uid": <Identity Content>,
  "Rule": {
    "name": "Test*"
  }
}

```

The command control manager responds with the data requested:

```

{
  [CCTree]: {
    [CCTree hierarchy]
  },
  [Entity]: {
    [Entity details]
  },
  "vrm": [framework version]
}

```

## 1.4.2 evalTemplate

Evaluates a PSP report template against the command control database:

```

{
  "module": "cmdctrl",
  "method": "evalTemplate",
  "uid": <Identity Content>,
  "Report": {
    "Template": {
      "value": [psp template]
    }
  }
}

```

The command control manager responds with the output from the report:

```

{
  "Report": {
    "content": [result of report],
  }
  "vrm": [framework version]
}

```

## 1.4.3 Add User Group

Adds a new user group entity. When the user group has been added, its contents can be modified.

```

{
  "method": "modEntries",
  "module": "cmdctrl",
  "remote_host": "127.0.0.1",

  "local_port": 443,
  "CCTree": {
    "id": 0,
    "UserGroup": {
      "id": 3,
      "UserGroup": {
        "action": "set",
        "name": "usrgrp1"
      }
    }
  }
}

```

The response contains the newly added ID. For example:

```

{
  "UserGroup": [
    {
      "name": "Everyone",
      "type": 0,
      "disabled": 0,
      "key": 1
    },
    {
      "name": "Submit User",
      "type": 0,
      "disabled": 0,
      "key": 2
    },
    {
      "name": "usergrp",
      "type": 0,
      "disabled": 0,
      "key": 3
    },
    {
      "name": "usrgrp1",
      "type": 0,
      "disabled": 0,
      "key": 4
    }
  ],
  "CCTree": {
    "id": 0,
    "UserGroup": {
      "id": 3,
      "UserGroup": [
        {
          "id": 102,
          "key": 1,
          "name": "Everyone",
          "type": 0,
          "disabled": 0
        },
        {
          "id": 103,
          "key": 2,
          "name": "Submit User",
          "type": 0,
          "disabled": 0
        },
        {
          "id": 204,

```

```

        "key":3,
        "name":"usergrp",
        "type":0,
        "disabled":0
      },
      {
        "id":211,
        "key":4,
        "name":"usrgrp1",
        "type":0,
        "disabled":0
      }
    ]
  }
}

```

## 1.4.4 Get User Group Entity

Gets a specified user group entity:

```

{
  "method":"getEntries",
  "module":"cmdctrl",
  "remote_host":"127.0.0.1",
  "local_port":443,
  "UserGroup":{
  },
  "CCTree":{
    "id":3
  }
}

```

Example output:

```

{
  "UserGroup":[
    {
      "name":"Everyone",
      "type":0,
      "disabled":0,
      "key":1
    },
    {
      "name":"Submit User",
      "type":0,
      "disabled":0,
      "key":2
    },
    {
      "name":"usergrp",
      "type":0,
      "disabled":0,
      "key":3
    },
    {
      "name":"usrgrp1",
      "type":0,
      "disabled":0,
      "key":4
    }
  ],
  "CCTree":{
    "id":0,
    "UserGroup":{
      "id":3,
      "UserGroup":[
        {

```



```

        "value":1
      }
    ],
    "HostGroup":[
      {
        "value":4
      }
    ],
    "UserGroup":[
      {
        "value":4
      }
    ]
  ]
}
}
}

```

## 1.4.6 Add Account Group

Adds a new account group. After the account group has been created, the contents can be modified.

```

{
  "method": "modEntries",
  "module": "cmdctrl",
  "remote_host": "127.0.0.1",
  "local_port": 443,
  "CCTree": {
    "id": 0,
    "AccountGroup": {
      "id": 2,
      "AccountGroup": {
        "action": "set",
        "name": "accgrptst"
      }
    }
  }
}
}
}

```

The response contains the newly added ID.

## 1.4.7 Get Account Group Entity

Gets the details of an account group:

```

{
  "method": "getEntries",
  "module": "cmdctrl",
  "remote_host": "127.0.0.1",
  "local_port": 443,
  "AccountGroup": {
  },
  "CCTree": {
    "id": 2
  }
}

```

Example Output:

```

{
  "AccountGroup":[
    {
      "name":"acgrp1",
      "type":0,
      "disabled":0,
      "key":1
    },
    {
      "name":"accgrptst",
      "type":0,
      "disabled":0,
      "key":2
    }
  ],
  "CCTree":{
    "id":0,
    "AccountGroup":{
      "id":2,
      "AccountGroup":[
        {
          "id":206,
          "key":1,
          "name":"acgrp1",
          "type":0,
          "disabled":0
        },
        {
          "id":208,
          "key":2,
          "name":"accgrptst",
          "type":0,
          "disabled":0
        }
      ]
    }
  }
}

```

## 1.4.8 Add Command

Adds a command entity. The response contains the newly added ID.

```

{
  "method":"modEntries",
  "module":"cmdctrl",
  "remote_host":"127.0.0.1",
  "local_port":443,
  "CCTree":{
    "id":0,
    "Command":{
      "id":5,
      "Command":{
        "action":"set",
        "name":"cmdtst1"
      }
    }
  }
}

```

## 1.4.9 Get Command Entity

Gets specified command information:



```

{
  "method": "getEntries",
  "module": "cmdctrl",
  "remote_host": "127.0.0.1",
  "local_port": 443,
  "Command": {
  },
  "CCTree": {
    "id": 5
  }
}

```

Example output:

```

{
  "Command": [
    {
      "name": "RDP Session",
      "type": 0,
      "disabled": 0,
      "key": 1
    },
    {
      "name": "SSH Session",
      "type": 0,
      "disabled": 0,
      "key": 2
    },
    {
      "name": "cmd",
      "type": 0,
      "disabled": 0,
      "key": 3
    },
    {
      "name": "cmdtst1",
      "type": 0,
      "disabled": 0,
      "key": 4
    }
  ],
  "CCTree": {
    "id": 0,
    "Command": {
      "id": 5,
      "Command": [
        {
          "id": 106,
          "key": 1,
          "name": "RDP Session",
          "type": 0,
          "disabled": 0
        },
        {
          "id": 107,
          "key": 2,
          "name": "SSH Session",
          "type": 0,
          "disabled": 0
        },
        {
          "id": 207,
          "key": 3,
          "name": "cmd",

```



```

{
  "HostGroup":[
    {
      "name":"All Hosts",
      "type":0,
      "disabled":0,
      "key":1
    },
    {
      "name":"Submit Host",
      "type":0,
      "disabled":0,
      "key":2
    },
    {
      "name":"hostgroup",
      "type":0,
      "disabled":0,
      "key":3
    },
    {
      "name":"hostgrp1",
      "type":0,
      "disabled":0,
      "key":4
    }
  ],
  "CCTree":{
    "id":0,
    "HostGroup":{
      "id":4,
      "HostGroup":[
        {
          "id":104,
          "key":1,
          "name":"All Hosts",
          "type":0,
          "disabled":0
        },
        {
          "id":105,
          "key":2,
          "name":"Submit Host",
          "type":0,
          "disabled":0
        },
        {
          "id":205,
          "key":3,
          "name":"hostgroup",
          "type":0,
          "disabled":0
        },
        {
          "id":210,
          "key":4,
          "name":"hostgrp1",
          "type":0,
          "disabled":0
        }
      ]
    }
  ]
}

```

## 1.4.12 Add Rule

Adds a new rule entity. The response contains the newly added ID.

```

{
  "method": "modEntries",
  "module": "cmdctrl",
  "remote_host": "127.0.0.1",
  "local_port": 443,
  "CCTree": {
    "id": 0,
    "Rule": {
      "id": 1,
      "Rule": {
        "action": "set",
        "name": "rulest1",
        "SubmitGroups": {
        }
      },
      "Commands": {
      }
    }
  }
}

```

### 1.4.13 Get Rule Entity

Retrieves specific rule information.

```

{
  "method": "getEntries",
  "module": "cmdctrl",
  "remote_host": "127.0.0.1",
  "local_port": 443,
  "Rule": {
  },
  "CCTree": {
    "id": 1
  }
}

```

Example output:

```

{
  "Rule": [
    {
      "name": "all",
      "type": 0,
      "disabled": 0,
      "key": 1
    },
    {
      "name": "Test rule 1",
      "type": 0,
      "disabled": 0,
      "key": 2
    },
    {
      "name": "Test rule 2",
      "type": 0,
      "disabled": 0,
      "key": 3
    }
  ],
  "CCTree": {
    "id": 0,
    "Rule": {
      "id": 1,
      "Rule": [
        {
          "id": 201,
          "key": 1,

```



## 1.4.15 Modify Rule Condition

Modifies the rule's condition logic:

```
{
  "method": "modEntries",
  "module": "cmdctrl",
  "remote_host": "127.0.0.1",
  "local_port": 443,
  "Rule": {
    "key": 4,
    "Match": {
      "action": "set",
      "Logic": [
        {
          "type": "UserGroup",
          "value": "AND",
          "key": 4,
          "UserGroup": {
            "user": "Passwd.username",
            "value": 4
          }
        },
        {
          "type": "HostGroup",
          "value": "AND",
          "key": 4,
          "HostGroup": {
            "host": "Host.name",
            "value": 4
          }
        },
        {
          "type": "Command",
          "value": "AND",
          "key": 4,
          "Command": {
            "cmd": "Command.cmd",
            "value": 4
          }
        }
      ]
    }
  }
}
```

Example output:

```
{
  "Rule": {
    "key": 4,
    "status": 0,
    "Match": {
      "action": "set",
      "Logic": [
        {
          "type": "UserGroup",
          "value": "AND",
          "key": 4,
          "UserGroup": {
            "user": "Passwd.username",
            "value": 4
          }
        },
        {
          "type": "HostGroup",
          "value": "AND",
          "key": 4,
```



```

{
  "method": "modEntries",
  "module": "cmdctrl",
  "remote_host": "127.0.0.1",
  "local_port": 443,
  "Rule": {
    "name": "rulest1",
    "disabled": 0,
    "key": 4,
    "type": 0,
    "action": "set",
    "Match": {
      "Logic": [
        {
          "type": "UserGroup",
          "value": "AND",
          "key": 4,
          "UserGroup": {
            "user": "Passwd.username",
            "value": 4
          }
        },
        {
          "type": "HostGroup",
          "value": "AND",
          "key": 4,
          "HostGroup": {
            "host": "Host.name",
            "value": 4
          }
        },
        {
          "type": "Command",
          "value": "AND",
          "key": 4,
          "Command": {
            "cmd": "Command.cmd",
            "value": 4
          }
        }
      ]
    },
    "Metadata": {
      "Exec": {
        "runAs": "root",
        "runHost": "host1"
      },
      "UserMessage": {
        "content": "This is the User Message"
      },
      "SessionCapture": {
        "value": "yes"
      },
      "Authorized": {
        "value": "yes"
      }
    },
    "Disabled": {
    },
  },
}

```



```

    "Description":{
      "value":"description"
    },
    "Stop":{
      "value":0
    },
    "Audit":{
      "group":"auditgrp"
    },
    "Risk":{
      "value":2
    }
  }
}

```

## 1.4.18 Modify User Group

Modifies a user group entity:

```

{
  "method":"modEntries",
  "module":"cmdctrl",
  "remote_host":"127.0.0.1",
  "local_port":443,
  "UserGroup":{
    "action":"set",
    "key":4,
    "name":"usrgrp1",
    "type":0,
    "Disabled":{
      "value":false
    },
    "RunUsers":{
      "value":true
    },
    "SubmitUsers":{
      "value":true
    },
    "Description":{
      "value":"description"
    },
    "MgrAccount":{
      "value":"admin"
    },
    "MgrName":{
      "value":"The Managers Name"
    },
    "MgrTel":{
      "value":"111222333"
    },
    "MgrEmail":{
      "value":"admin@foo.com"
    },
    "External":{
      "value":false
    },
    "UserList":{
      "User":[
        {

```

```

        "value": "user1"
      },
      {
        "value": "user2*"
      }
    ],
    "UserGroup": [
      {
        "value": 3
      }
    ]
  }
}
}
}

```

## 1.4.19 Get Command Control Tree

Retrieves the complete command control policy tree:

```

{
  "method": "getEntries",
  "module": "cmdctrl",
  "remote_host": null,
  "CCTree": {
    "id": 0
  },
  "Rule": {
  },
  "AccountGroup": {
  },
  "UserGroup": {
  },
  "HostGroup": {
  },
  "Command": {
  },
  "Script": {
  },
  "Tme": {
  }
}

```

Example output:

```

{
  "CCTree": {
    "id": 0,
    "Rule": {
      "id": 1,
      "Rule": [
        {
          "id": 201,
          "key": 1,
          "name": "all",
          "type": 0,
          "disabled": 0,
          "ref": 1
        },
        {
          "id": 202,
          "key": 2,
          "name": "Test rule 1",
          "type": 0,
          "disabled": 0,
          "ref": 1
        },
        {
          "id": 203,

```

```

        "key":3,
        "name":"Test rule 2",
        "type":0,
        "disabled":0,
        "ref":1
    },
    {
        "id":212,
        "key":4,
        "name":"rulest1",
        "type":0,
        "disabled":0,
        "ref":1
    }
]
},
"AccountGroup":{
    "id":2,
    "AccountGroup":[
        {
            "id":206,
            "key":1,
            "name":"acgrp1",
            "type":0,
            "disabled":0
        },
        {
            "id":208,
            "key":2,
            "name":"accgrptst",
            "type":0,
            "disabled":0
        }
    ]
},
"UserGroup":{
    "id":3,
    "UserGroup":[
        {
            "id":102,
            "key":1,
            "name":"Everyone",
            "type":0,
            "disabled":0
        },
        {
            "id":103,
            "key":2,
            "name":"Submit User",
            "type":0,
            "disabled":0
        },
        {
            "id":204,
            "key":3,
            "name":"usergrp",
            "type":0,
            "disabled":0
        },
        {
            "id":211,
            "key":4,
            "name":"usrgrp1",
            "type":0,
            "disabled":0
        }
    ]
},
"HostGroup":{
    "id":4,

```

```

"HostGroup": [
  {
    "id":104,
    "key":1,
    "name":"All Hosts",
    "type":0,
    "disabled":0
  },
  {
    "id":105,
    "key":2,
    "name":"Submit Host",
    "type":0,
    "disabled":0
  },
  {
    "id":205,
    "key":3,
    "name":"hostgroup",
    "type":0,
    "disabled":0
  },
  {
    "id":210,
    "key":4,
    "name":"hostgrp1",
    "type":0,
    "disabled":0
  }
]
},
"Command": {
  "id":5,
  "Command": [
    {
      "id":106,
      "key":1,
      "name":"RDP Session",
      "type":0,
      "disabled":0
    },
    {
      "id":107,
      "key":2,
      "name":"SSH Session",
      "type":0,
      "disabled":0
    },
    {
      "id":207,
      "key":3,
      "name":"cmd",
      "type":0,
      "disabled":0
    },
    {
      "id":209,
      "key":4,
      "name":"cmdtst1",
      "type":0,
      "disabled":0
    }
  ]
},
"Script": {
  "id":6
},
"Tme": {
  "id":7
},

```

```

    "RuleTemplate":{
      "id":8
    },
    "Report":{
      "id":9
    },
    "CCTree":{
    }
  },
  "Rule":[
    {
      "name":"all",
      "type":0,
      "disabled":0,
      "key":1
    },
    {
      "name":"Test Rule 1",
      "type":0,
      "disabled":0,
      "key":2
    },
    {
      "name":"Test Rule 2",
      "type":0,
      "disabled":0,
      "key":3
    },
    {
      "name":"rulest1",
      "type":0,
      "disabled":0,
      "key":4
    }
  ],
  "AccountGroup":[
    {
      "name":"acgrp1",
      "type":0,
      "disabled":0,
      "key":1
    },
    {
      "name":"accgrptst",
      "type":0,
      "disabled":0,
      "key":2
    }
  ],
  "UserGroup":[
    {
      "name":"Everyone",
      "type":0,
      "disabled":0,
      "key":1
    },
    {
      "name":"Submit User",
      "type":0,
      "disabled":0,
      "key":2
    },
    {
      "name":"usergrp",
      "type":0,
      "disabled":0,
      "key":3
    }
  ],

```

```

        "name": "usrgrpl",
        "type": 0,
        "disabled": 0,
        "key": 4
    },
    "HostGroup": [
        {
            "name": "All Hosts",
            "type": 0,
            "disabled": 0,
            "key": 1
        },
        {
            "name": "Submit Host",
            "type": 0,
            "disabled": 0,
            "key": 2
        },
        {
            "name": "hostgroup",
            "type": 0,
            "disabled": 0,
            "key": 3
        },
        {
            "name": "hostgrpl",
            "type": 0,
            "disabled": 0,
            "key": 4
        }
    ],
    "Command": [
        {
            "name": "RDP Session",
            "type": 0,
            "disabled": 0,
            "key": 1
        },
        {
            "name": "SSH Session",
            "type": 0,
            "disabled": 0,
            "key": 2
        },
        {
            "name": "cmd",
            "type": 0,
            "disabled": 0,
            "key": 3
        },
        {
            "name": "cmdtst1",
            "type": 0,
            "disabled": 0,
            "key": 4
        }
    ]
}

```

## 1.5 Miscellaneous Functions

- ♦ [Section 1.5.1, "List Updates," on page 47](#)
- ♦ [Section 1.5.2, "Install Package," on page 48](#)
- ♦ [Section 1.5.3, "Promote Manager Module," on page 48](#)

- ♦ [Section 1.5.4, “Change Agent Address,”](#) on page 49
- ♦ [Section 1.5.5, “Create Framework Host,”](#) on page 49
- ♦ [Section 1.5.6, “List Framework Hosts,”](#) on page 49
- ♦ [Section 1.5.7, “List Package Manager Packages,”](#) on page 50

## 1.5.1 List Updates

Lists the NPUM package updates that are waiting to be applied:

```
{ "method" : "callModuleEx",
  "params" : {
    "pkt" : {
      "module": "distrib",
      "method": "listUpdates",
      "uid": <Identity Content>
    }
  }
}
```

Example output:

```
{
  "svc": "pmmgr",
  "Module": [
    {
      "content": "\n ",
      "type": "manager",
      "name": "auth",
      "Title": {
        "content": "Access Manager"
      },
      "Description": {
        "content": "Provides Framework authentication"
      },
      "Package": {
        "rel": "beta",
        "rev": "23194",
        "build": "5086",
        "version": "2,3,0,3",
        "name": "auth-2-3-beta-linux-x86_64-2.6"
      },
      "System": {
        "version": "2.6",
        "hw": "x86_64",
        "os": "linux",
        "major": "2",
        "minor": "6"
      },
      "Depends": {
        "content": "\n ",
        "Patch": {
          "version": "2,1,6,0",
          "name": "spf"
        },
        "Module": {
          "version": "2,0,2,0",
          "name": "registry"
        }
      }
    },
    {
      "content": "\n ",
      "type": "manager",
      "name": "audit",
      "Title": {
        "content": "Audit Manager"
      }
    }
  ]
}
```

```

    },
    "Description":{
      "content":"Provides storage and reporting of system and application
audit events"
    },
    "Package":{
      "rel":"beta",
      "rev":"23193",
      "build":"5086",
      "version":"2,3,0,3",
      "name":"audit-2-3-beta-linux-x86_64-2.6"
    },
    "System":{
      "version":"2.6",
      "hw":"x86_64",
      "os":"linux",
      "major":"2",
      "minor":"6"
    },
    "Depends":{
      "content":"\n ",
      "Patch":{
        "rev":"20351",
        "version":"2,2,1,0",
        "name":"spf"
      }
    }
  },
}
]
}

```

## 1.5.2 Install Package

Installs an updated package.

```

{
  "method" : "callModuleEx",
  "params" : {
    "pkt" : {
      "module": "distrib",
      "method": "pullPackage",
      "Pull":{
        "<type>":{
          "name": "<name>",
          "backup":true,
          "Package":{
            "name": "<package>"
          }
        }
      }
    }
  },
  "uid":<Identity Content>
}
}
}

```

<type> is console, module, or patch.

<name> is the short name of the package, such as "auth."

<package> is the long name of the package, such as "auth-2-3-beta-linux-x86\_64-2.6."

## 1.5.3 Promote Manager Module

Promotes a specified module as master:



```

{
  "method" : "callModuleEx",
  "params" : {
    "pkt" : {
      "module": "registry",
      "method": "setMaster",
      "Module" : {
        "name" : "<short-module-name>"
      },
      "uid":<Identity-Content>
    },
  }
}

```

## 1.5.4 Change Agent Address

Changes the address (IP or DNS) that an agent exists on:

```

{
  "method" : "callModuleEx",
  "params" : {
    "pkt" : {
      "module": "registry",
      "method": "setSvcInfo",
      "Service": {
        "host": "<new-dns-or-ip-adress>",
        "id":<agent-svc-id>,
      },
      "uid":<Identity-Content>
    },
  }
}

```

## 1.5.5 Create Framework Host

Creates a framework host:

```

{
  "method": "svcCreate",
  "module": "registry",
  "remote_host": "192.168.1.2",
  "local_port": 443,
  "Service": {
    "name": "pumagnt",
    "orgid": 0
  }
}

```

## 1.5.6 List Framework Hosts

Lists framework hosts:

```

{
  "method": "orgList",
  "module": "registry",
  "remote_host": "127.0.0.1",
  "local_port": 443,
  "OrgUnit": {
    "services": 1,
    "recursive": 1
  }
}

```

Example output:

```

{
  "OrgUnit":{
    "id":0,
    "parent":-1,
    "ou":"0",
    "ou_idx":0,
    "name":"",
    "Service":[
      {
        "id":"7uHu67KGSOL/zGQOSplbcWEd014=",
        "name":"pmmgr",
        "host":"pmmgr",
        "port":29120,
        "orgid":0,
        "desc":"NPUM Manager"
      }
    ]
  }
}

```

## 1.5.7 List Package Manager Packages

Lists package manager packages:

```

{
  "module":"pkgman",
  "method":"listPackages",
  "Module":{
  },
  "Engine":{
  },
  "Console":{
  }
}

```

Example output:

```

{
  "Module":
  [
    {
      "content":"\n ",
      "type":"manager",
      "name":"auth",
      "Title":{
        "content":"Access Manager"
      },
      "Description":{
        "content":"Provides Framework authentication"
      },
      "Package":{
        "rel":"beta",
        "rev":"23194",
        "build":"4998",
        "version":"2,3,0,3",
        "name":"auth-2-3-beta-linux-x86_64-2.6"
      },
      "System":{
        "version":"2.6",
        "hw":"x86_64",
        "os":"linux"
      },
      "Depends":{
        "content":"\n ",
        "Patch":{
          "version":"2,1,6,0",
          "name":"spf"
        }
      },
    },
  ]
}

```

```

        "Module":{
            "version":"2,0,2,0",
            "name":"registry"
        }
    },
    {
        "content": "\n ",
        "type": "manager",
        "name": "auth",
        "Title":{
            "content": "Access Manager"
        },
        "Description":{
            "content": "Provides Framework authentication"
        },
        "Package":{
            "rel": "",
            "rev": "20429",
            "build": "4975",
            "version": "2,2,2,0",
            "name": "auth-2-2-linux-x86_64-2.6"
        },
        "System":{
            "version": "2.6",
            "hw": "x86_64",
            "os": "linux"
        },
        "Depends":{
            "content": "\n ",
            "Patch":{
                "version": "2,1,6,0",
                "name": "spf"
            },
            "Module":{
                "version": "2,0,2,0",
                "name": "registry"
            }
        }
    },
}
]
}

```

