Roles and Users Management

This section includes the following topics:

Roles management

Manage roles command

Managing roles using CLI is done using the following command:

Windows
```
infra\bin\psin_cli.bat
-i3-user <user_name>
{-i3-encrypted-password <encrypted_password> | -i3-clear-password <clear_password>}
-action manage-roles
-roles-parametersfile <roles_parameters.xml>
```

UNIX
```
./infra/bin/psin_cli.sh
-i3-user <user_name>
{-i3-encrypted-password <encrypted_password> | -i3-clear-password <clear_password>}
-action manage-roles
-roles-parametersfile <roles_parameters.xml>
```

Table 1 Elements of the Manage roles command

Element	Description
action	**Values**: manage-roles **Mandatory**: yes
i3-user	See Authenticate to CLI Utility on page 8.
is-encrypted-password	See Authenticate to CLI Utility on page 8.
roles-parametersfile	**Values**: the parameters file that holds the roles definitions. **Mandatory**: Yes

The parameters file contains the definitions for one or more roles. The file structure is as follows

```
<root>
   <role>
      Role definition
   </role>
   …additional roles definitions
</root>
```

> When managing only one role the <role> tag is not required.

```
<root>
   Role definition
</root>
```

Required permissions

To activate the roles management command the user must have ADMINISTRATE.EXECUTE permissions for Precise technology.

Roles definitions limitations

A user activating this command can add/edit/delete roles only if the permissions the role contains are in the scope of the user's permissions.

For example, if a user has ADMINISTRATE.VIEW permission on an Oracle instance he can create a new role with ADMINISTRATE.VIEW permission on the Oracle instance he has the same permission on, however, he will not be able to assign ADMINISTRATE.VIEW on another instance or define ADMINISTRATE.EXECUTE on the same instance.

Handling errors

The CLI mechanism always skips to the next role and does not halt the whole operation in case of error. In case the CLI fails for a specific role a message is issued to the screen and the problem is logged in the CLI log file.

Adding a new role

The definition for adding a new role is as follows:

```
<root>
    <parameter name="action" value="add"/>
    <parameter name="role-name" value="role-name"/>
    <parameter name="role-scope" value="role-scope"/>
    <complex name="permissions">
        <parameter permission-type="permission-type" permission-operation="permission-operation"/>
        …additional permissions definitions
    </complex>
    <complex name="resources" [value="*"]> [<parameter [resource-information]/>
        …additional resources definitions ]
    </complex>
    [<complex name="nodes" [value="*"]>
        <parameter node-name="node-name"/>
        …additional nodes definitions
    </complex>]
</root>
```

Table 2 Parameter values for Adding a new role

Parameter	Description
action	The action we wish to perform on the defined role. **Value**: add.
role-name	Value: The name of the role to be added. **Mandatory**: Yes
role-scope	The scope of the role the user wants to define the permissions on. **Value**: technology, application, Tier, or instance. See Table 9-3 on page 111. **Mandatory**: Yes
permissions	The permissions we wish to assign to this role. When assigning permission to a role all the dependencies of this permission are automatically assigned to the role as well (i. e. assigning MONITOR.WHAT_IF will automatically assign MONITOR.EXPLAIN as well). **Mandatory**: Yes
permission-type	The permission type name we wish to assign to this role, for example: monitor. **Mandatory**: Yes
permission-operation	The permission operation name we wish to assign to this role, for example: view. **Mandatory**: Yes
resources	The resources the role permissions apply to. **Mandatory**: Yes
resource-information	Resource information holds the information of the resource the permission is granted on. This information is derived from the role scope parameter, as shown in Table 9-3 on page 111. **Mandatory**: Yes

nodes	This parameter is relevant only to 'technology' role scope. Use this parameter to define technologies permissions on specific nodes. If this parameter is not defined the technology permissions will apply on all nodes. **Value**: Name of the node To define permission on all the resources of a specific type and all future resources as well (i.e. all instances), do not specify any resource information. Instead, define the resources value attribute as "*". To define permission on all the nodes do not specify any node name. Instead, define the nodes value attribute as "*". **Mandatory**: No

Table 3 Role scope parameters

Role scope parameter	Resource information	Example
technology	technology-code: the technology code the permission should be granted on	<parameter technology-code=" OR"/>
environment	environment-name: the name of the application the permission should be granted on.	<parameter environment-name=" Default"/> View permission can only be set on the application resource.
apptier	apptier-name: the name of the Tier the permission should be granted on environment-name: the name of the application the Tier belongs to.	<parameter apptier-name=" Oracle" environment-name=" Default"/>
instance	**instance-name**: the name of the instance the permission should be granted on. **technology-code**: this parameter is optional, specifies the instance technology code. This parameter should be used if the instance name is not unique. **server-name**: this parameter is optional, specifies the server the instance is installed on. If this parameter is specified the technology code parameter must be specified as well. This parameter should be used if the instance name and technology are not unique.	<parameter instance-name=" ORCL" technology-code="OR" server-name="orcl-server"/>

Example

In this example we will be adding two roles as follows:

- First role definition:
  - role name: test-role1
  - role scope: technology
  - role permissions:
    - MONITOR.FULL_CONTROL
    - ADMINISTRATE.EXECUTE
  - role resources
    - Oracle technology
    - Oracle Applications technology
  - role nodes
    - node1
    - node2
- Second role definition
  - role name: test-role2
  - role scope: application
  - role permissions:
    - MONITOR.VIEW
  - role resources:
    - All applications

The roles parameters file will look as follows:

```
<root>
    <role>
        <parameter name="action" value="add"/>
        <parameter name="role-name" value=" test-role1"/>
        <parameter name="role-scope" value="technology"/>
        <complex name="permissions">
            <parameter permission-type="monitor" permission-operation="full_control"/>
            <parameter permission-type="administrate" permission-operation="execute"/>
        </complex>
        <complex name="resources">
            <parameter technology-code="OR"/>
            <parameter technology-code="OA"/>
        </complex>
        <complex name="nodes">
            <parameter node-name="node1"/>
            <parameter node-name="node2"/>
        </complex>
    </role>
    <role>
        <parameter name="action" value="add"/>
        <parameter name="role-name" value=" test-role2"/>
        <parameter name="role-scope" value="application"/>
        <complex name="permissions">
            <parameter permission-type="monitor" permission-operation="view"/>
        <complex name="resources" value="all=true"/>
    </role>
</root>
```

Deleting a role

The definition for deleting a role is as follows:

```
<root>
    <parameter name="action" value="delete"/>
    <parameter name="role-name" value="role-name"/>
</root>
```

The role ‘I3 Manager’ cannot be deleted.

Verifying user’s roles assignment before deletion

Deleting a role can cause a situation where one or more users will be left with no roles assigned to them (i.e. this role is the only role assigned to one of the users).

It is possible to define that if a role deletion creates this kind of situation, an error will be issued. This definition can be done by setting the following registry parameter to ‘false’:

products\i3fp\registry\products\infrastructure\roles\settings\ignore-last-role-on-delete

This parameter default value is ‘true’.

Parameters specification

Table 4 Parameter values for Deleting a role

Parameter	Description
action	The action we wish to perform on the defined role. **Value**: delete. **Mandatory**: Yes
role-name	The name of the role we wish to delete. **Mandatory**: Yes

Example

In this example we will be deleting one role ‘test-role1’:

```
<root>
    <parameter name="action" value="delete"/>
    <parameter name="role-name" value=" test-role1"/>
</root>
```

Editing a role

The definition for editing a new role is as follows:

```
<root>
    <parameter name="action" value="edit"/>
    <parameter name="role-name" value="role-name"/>
    [<parameter name="role-scope" value="role-scope"/>]
    [<parameter name="role-new-name" value="role-new-name"/>]
    <complex name="permissions">
        <parameter permission-type="permission-type" permission-operation="permission-operation"/>
        …additional permissions definitions
    </complex>
    <complex name="resources" [value="*"]>
        [<parameter [resource-information]/>
        …additional resources definitions ]
    </complex>]
    [<complex name="nodes" [value="*"]>
        [<parameter node-name="node-name"/>
        …additional nodes definitions ]
    </complex>]
</root>
```

> The role ‘I3 Manager’ cannot be edited.

Parameters specification

Table 5 Parameter values for Editing a role

Parameter	Description
Action	The action we wish to perform on the defined role. **Value**: Edit **Mandatory**: Yes
Role-name	The name of the role we wish to edit. **Mandatory**: Yes
Role-scope	The scope of the rode the user wants to define the permissions on. **Values**: technology, application, Tier, or instance. **Mandatory**: Yes
Role-new-name	The new role name. **Mandatory**: No
Permissions	The permissions we wish to assign to this role. **Mandatory**: No When assigning permission to a role, all the dependencies of this permission are automatically assigned to the role as well (i. e. assigning MONITOR.WHAT_IF will automatically assign MONITOR.EXPLAIN as well). • **Permission-type**. The permission type name we wish to assign to this role. • **Permission-operation**. The permission operation name we wish to assign to this role. Permissions that are already assigned to the role and are not specified in edit will be removed from the role definition.

Resources	The resources the role permissions apply on. • **Resource-information**. See Table 9-3 on page 111. Resources that are already assigned to the role and are not specified in edit will be removed from the role definition. **Mandatory**: Yes, if permissions are changed.
Nodes	This parameter is relevant only to 'technology' role scope. Use this parameter to define technologies permissions on specific nodes. If this parameter is not defined the technology permissions will apply on all nodes. **Node-name**: the name of the node Nodes that are already assigned to the role and are not specified in edit will be removed from the role definition. If no proxies are specified in edit mode then the role's proxies will remain unchanged. **Mandatory**: No

Example

In this example we will be editing the following role:

- Role name: test-role1
- Role scope: technology
- Role permissions
  - MONITOR.FULL_CONTROL
  - TUNE.EXECUTE
  - ADMINISTRATE.EXECUTE
- Role resources:
  - Oracle technology
  - Oracle Applications technology
- Role nodes
  - node1
  - node2

We will remove the first two permissions and add a new permission – MONITOR.EXECUTE. We will set the role to apply on all nodes and also change the role name to 'test-role1-updated'. After the edit operation is completed, the role definition will be:

- Role name: test-role1-updated (the new name)
- Role scope: technology (wasn't affected by the update)
- Role permissions
  - MONITOR.EXECUTE (new permission)
  - ADMINISTRATE.EXECUTE (redefined by the update)
- Role resources
  - Oracle technology (redefined by the update)
  - Oracle Applications technology (redefined by the update)
- Role nodes: all nodes (updated definition) The roles parameters file will look as follows:

```
<root>
   <parameter name="action" value="edit"/>
   <parameter name="role-name" value=" test-role1"/>
   <parameter name="role-new-name" value=" test-role1-updated"/>
   <complex name="permissions">
      <parameter permission-type="monitor" permission-operation="execute"/>
      <parameter permission-type="administrate" permission-operation="execute"/>
   </complex>
   <complex name="resources">
      <parameter instance-name="ORCL" technology-code="OR" server-name="srv1"/>
      <parameter instance-name="OA1" technology-code="OA" server-name="srv2"/>
   </complex>
   <parameter name="nodes" value="*"/>
   <parameter name="role-scope" value="INSTANCE"/>
</root>
```

Users management

Manage users command

Managing users using CLI is done using the following command:

Windows infra\bin\psin_cli.bat
-i3-user <*user_name*>
{-i3-encrypted-password <*encrypted_password*> | -i3-clear-password <*clear_password*>}
-action manage-users
-roles-parametersfile <users_parameters.xml>

UNIX ./infra/bin/psin_cli.sh
-i3-user <*user_name*>
{-i3-encrypted-password <*encrypted_password*> | -i3-clear-password <*clear_password*>}
-action manage-users
-roles-parametersfile <users_parameters.xml>

Table 6 Elements for the Manage users command

Parameter	Description
i3-user	See Authenticate to CLI Utility on page 8.
is-encrypted-password	See Authenticate to CLI Utility on page 8.
role-parametersfile	**Values**: the parameters file that holds the users definitions **Mandatory**: Yes
action	**Values**: manage-users **Mandatory**: Yes

The parameters file contains the definitions for one or more users. The file structure is as follows

```
<root>
   <user>
      User definition
   </user>
   …additional users definitions
</root>
```

When managing only one user the <user> tag is not required.

```
<root>
   User definition
</root>
```

Required permissions

Activating the user's management command requires ADMINISTRATE.EXECUTE permissions on Precise technology.

User roles definitions limitations

A user activating this command can add/remove roles to the managed user, only if the permissions of the roles granted/removed from the managed user, are in the scope of the managing user's roles permissions.

For example, if a user has ADMINISTRATE.VIEW permission on an Oracle instance he can create a new user and assign a role with ADMINISTRATE. VIEW permission on the Oracle instance he has the same permission on, however, he will not be able to assign a role with ADMINISTRATE.VIEW on another instance or ADMINISTRATE.EXECUTE on the same instance.

Handling errors

CLI mechanism always skips to the next user and does not halt the whole operation in case of error. In case the CLI fails for a specific user, a message is issued to the screen and the problem is logged in the CLI log file.

Adding a new user

The definition for adding a new user is as follows:

```
<root>
    <parameter name="action" value="add"/>
    <parameter name="user-name" value="user-name"/>
    {<parameter name="user-clear-password" value="clear-password"/> |
    <parameter name="user-encrypted-password" value="encrypted-password"/>}
    [<complex name="user-roles">
        <parameter role-name="role-name"/>
        …additional roles definitions
    </complex>]
</root>
```

Table 7 Parameters for Adding a new user

Parameter	Description
action	The action we wish to perform on the defined user. **Values**: Add **Mandatory**: Yes
User-name	The name of the user we wish to add. **Mandatory**: Yes
user-clear-password or user-encrypted-password	The user's password as clear or encrypted text. **Mandatory**: Yes
User-roles	The roles we wish to assign to this user • **Role-name**: The name of the role we wish to assign to this user. ⓘ The Role name is case sensitive. **Mandatory**: Yes

Example

In this example we will be adding two users.

```
<root>
    <user>
        <parameter name="action" value="add"/>
        <parameter name="user-name" value="koby"/>
        <parameter name="user-clear-password" value="1234"/>
        <complex name="user-roles">
            <parameter role-name="Tuxedo Manager"/>
            <parameter role-name="SQL Server Administrator"/>
        </complex>
    </user>
    <user>
        <parameter name="action" value="add"/>
        <parameter name="user-name" value="yossi"/>
        <parameter name="user-clear-password" value="5678"/>
        <complex name="user-roles">
            <parameter role-name="Web Manager"/>
        </complex>
    </user>
</root>
```

Deleting a user

The definition for deleting a user is as follows:

```
<root>
    <parameter name="action" value="delete"/>
    <parameter name="user-name" value="user-name"/>
</root>
```

ⓘ The 'admin' user cannot be deleted. The user activating this command cannot delete himself.

Table 8 Parameters for Deleting a user

Parameter	Description
Action	The action we wish to perform on the defined user. **Values**: Delete **Mandatory**: Yes
User-name	The name of the user we wish to delete. **Mandatory**: Yes

Example

In this example we will be deleting one user ‘koby’:

```
<root>
    <parameter name="action" value="delete"/>
    <parameter name="user-name" value="koby"/>
</root>
```

Editing a user

The definition for editing or adding a user is as follows:

```
<root>
    <parameter name="action" value="edit"/>
    <parameter name="user-name" value="user-name"/> [<parameter name="user-new-name" value="user-new-name"/>] [{<parameter name="user-clear-password" value="clear-password"/> |
    <parameter name="user-encrypted-password" value="encrypted-password"/>}]
    [<complex name="user-roles">
        <parameter role-name="role-name"/>
        …additional roles definitions
    </complex>]
</root>
```

Table 9 Parameters for Editing a user

Parameter	Description
-action	The action we wish to perform on the defined user. **Values**: Edit **Mandatory**: Yes
User-name	The name of the user we wish to edit. **Mandatory**: Yes
user-clear-password or user-encrypted-password	The user’s password as clear or encrypted text. **Mandatory**: Yes
User-roles	The roles we wish to assign to this user.
Role-name	The name of the role we wish to assign to this user. ⓘ The Role name is case sensitive. ⓘ Roles that are already assigned to the user and are not specified in edit will be removed from the user assigned roles. **Mandatory**: Yes

Example

In this example we will be editing the user ‘user1’. This user has the following roles assigned to him:

- Tuxedo Manager
- SQL Server Administrator

We will remove the role ‘SQL Server Administrator’ from his assigned roles and add the following role – ‘Oracle Administrator’. After the edit operation the user will have the following roles assigned to him:

- Tuxedo Manager
- Oracle Administrator

We don’t want to change the user’s password or name.

```
<root>
    <user>
        <parameter name="action" value="edit"/>
        <parameter name="user-name" value="user1"/>
        <complex name="user-roles">
            <parameter role-name="Oracle Administrator"/>
            <parameter role-name="Tuxedo Manager"/>
        </complex>
    </user>
</root>
```

Exporting users/roles

The roles-export command prints the current users/roles defined in Precise. The export result can later be used to define/update roles/users.

Windows infra\bin\psin_cli.bat
-i3-user <*user_name*>
{-i3-encrypted-password <*encrypted_password*> | -i3-clear-password <*clear_password*>}
-action roles-export
-mode {users|roles|all}
[-output-file <*file_name*>]

UNIX ./infra/bin/psin_cli.sh
-i3-user <*user_name*>
{-i3-encrypted-password <*encrypted_password*> | -i3-clear-password <*clear_password*>}
-action roles-export
-mode {users|roles|all}
[-output-file <*file_name*>]

Table 10 Elements for Exporting users/roles

Elements	Description
Mode	The required export mode. **Values**: export users, export roles, or export users and roles. **Mandatory**: Yes
output-file	The file path to which the export will be written. **Value**: If not specified: <precise_root>\infra\cli2\output\cli_expo rt_<mode>.xml. **Mandatory**: No

Command output

The roles export output is written to an output file as described in the previous table.

Output format

The output format can be users, roles or all.

Export users output

```
<users>
    <user>
        <parameter name="action" value="add" />
        <parameter name="user-encrypted-password" value="IAJDFKJBI@" />
        <parameter name="user-name" value="usr1" />
        <complex name="user-roles">
            <parameter role-name="Oracle view only" />
        </complex>
    </user>
    <user>
        <parameter name="action" value="add" />
        <parameter name="user-encrypted-password" value="ICDDFK@FFA" />
        <parameter name="user-name" value="usr2" />
        <complex name="user-roles">
            <parameter role-name="Precise Manager" />
        </complex>
    </user>
    ...additional users
</users>
```

Export roles output

```
<roles>
    <role>
        <parameter name="action" value="add" />
        <parameter name="nodes" value="*" />
        <complex name="permissions">
            <parameter permission-operation="FULL_CONTROL" permission-type="ADMINISTRATE" />
            <parameter permission-operation="FULL_CONTROL" permission-type="TUNE" />
            <parameter permission-operation="FULL_CONTROL" permission-type="MONITOR" />
        </complex>
        <parameter name="resources" value="*" />
        <parameter name="role-name" value="Precise Manager" />
        <parameter name="role-scope" value="TECHNOLOGY" />
    </role>
    <role>
        <parameter name="action" value="add" />
        <parameter name="nodes" value="*" />
        <complex name="permissions">
            <parameter permission-operation="VIEW" permission-type="MONITOR" />
            <parameter permission-operation="VIEW" permission-type="ADMINISTRATE" />
            <parameter permission-operation="VIEW" permission-type="TUNE" />
        </complex>
        <complex name="resources">
            <parameter technology-code="OR" />
        </complex>
        <parameter name="role-name" value="Oracle view only" />
        <parameter name="role-scope" value="TECHNOLOGY" />
    </role>
    ...additional roles
</roles>
```

Export all output

```
<all>
    <user>
        <parameter name="action" value="add" />
        <parameter name="user-encrypted-password" value="IAJDFKJBI@" />
        <parameter name="user-name" value="usr1" />
        <complex name="user-roles">
            <parameter role-name="Oracle view only" />
        </complex>
    </user>
    <user>
        <parameter name="action" value="add" />
        <parameter name="user-encrypted-password" value="ICDDFK@FFA" />
        <parameter name="user-name" value="usr2" />
        <complex name="user-roles">
            <parameter role-name="Precise Manager" />
        </complex>
    </user>
    ...additional users
    <role>
        <parameter name="action" value="add" />
        <parameter name="nodes" value="*" />
        <complex name="permissions">
            <parameter permission-operation="FULL_CONTROL" permission-type="ADMINISTRATE" />
            <parameter permission-operation="FULL_CONTROL" permission-type="TUNE" />
            <parameter permission-operation="FULL_CONTROL" permission-type="MONITOR" />
        </complex>
        <parameter name="resources" value="*" />
        <parameter name="role-name" value="Precise Manager" />
        <parameter name="role-scope" value="TECHNOLOGY" />
    </role>
    <role>
        <parameter name="action" value="add" />
        <parameter name="nodes" value="*" />
        <complex name="permissions">
            <parameter permission-operation="VIEW" permission-type="MONITOR" />
            <parameter permission-operation="VIEW" permission-type="ADMINISTRATE" />
            <parameter permission-operation="VIEW" permission-type="TUNE" />
        </complex>
        <complex name="resources">
            <parameter technology-code="OR" />
        </complex>
        <parameter name="role-name" value="Oracle view only" />
        <parameter name="role-scope" value="TECHNOLOGY" />
    </role>
    ...additional roles
</all>
```

User permissions summary

User permissions summary prints a summary of the permissions a user has. Managing users using CLI is done using the following command:

Windows infra\bin\psin_cli.bat
-i3-user <*user_name*>
{-i3-encrypted-password <*encrypted_password*> | -i3-clear-password <*clear_password*>}
-action permissions-summary
[-user-name <*user_name*>] [-output-file <*file_name*>]

UNIX ./infra/bin/psin_cli.sh
-i3-user <*user_name*>
{-i3-encrypted-password <*encrypted_password*> | -i3-clear-password <*clear_password*>}
-action permissions-summary
[-user-name <*user_name*>] [-output-file <*file_name*>]

Table 11 Elements for the User permissions summary

Element	Description
user-name	The user we wish to generate the permissions summary for. If this parameter is not specified, the permissions summary will be generated for the user activating this command according to the i3-user parameter. **Mandatory**: Yes
output-file	The file the command output will be written to. If this parameter is not specified the output will be written as follows: • An xml file will be generated under the following folder: infra\cli2\permissions. • The file name will be of the following format: permissions_summary_YYYY.MM.DD_HH_MM_S S.xml. **Mandatory**: No

Command output

The user permissions summary is printed to an output file as described above in the Parameters specification section.

Output format

The output xml structure will be as follows:

```
<user-permissions-summary user-name="user-name">
    <roles-permissions-summary>
        <role role-name="role-name">
            <permission>
            <description>permission description</description>
            <permission-type>permission type</permission-type>
            <permission-operation>permission operation</permission-operation>
            <resource resource-type="resource type" [resource-information] />
            <affected-instances>
            <instance>
            <instance-name>instance name</instance-name>
            <server-name>instance server name</server-name>
            <technology-code>instance technology</technology-code>
            </instance>
            …additional instances
        </affected-instances>
        </permission>
        …additional permissions
    </role>
    …additional roles
    </roles-permissions-summary>
</user-permissions-summary>
```

Output specification

- **User-name**. The name of the user this summary was generated for
- **Role-name**. The name of a role assigned to this user
- **Permission**. Role's permission specification
- **Description**. The description of this permission
- **Permission-type**. The permission type
- **Permission-operation**. The permission operation
- **Resource**. The resource this permission was granted on
- **Resource-type**. The type of the Resource-type can have one of the following values: technology, application, Tier or instance
- **Resource-information**. See resource information definitions in Table 9-3 on page 111.
- **Affected-instances**. The list of instances derived from the resource the permission was granted on (i.e. if the resource is the 'Default' application all the instances connected to the 'Default' application will be listed here)
- **Instance-name**. The name of the instance
- **Server-name**. The name of the server the instance is installed on
- **Technology-code**. The technology code of the instance

Example

This is an example of a permissions summary file for user 'usr1' with roles that contains the following permissions:

- MONITOR.VIEW on the 'Default' application
- ADMINISTRATE.FULL_CONTROL on an SQL Server instance
- MONITOR.EXECUTE on 'SQL Server' Tier in the 'Default' application

```
<user-permissions-summary user-name="usr1">
    <roles-permissions-summary>
        <role role-name="monitor default environment" role-scope="ENVIRONMENT"
            <permission>
                <description>'Monitor.View' permission on the selected applications</description>
                <permission-type>MONITOR</permission-type>
                <permission-operation>VIEW</permission-operation>
                <resource resource-type="ENVIRONMENT" environment-name="Default" />
                <affected-instances>
                    <instance>
                        <instance-name>PIFA1000</instance-name>
                        <server-name>pifa1000</server-name>
                        <technology-code>SQ</technology-code>
                    </instance>
                    <instance>
                        <instance-name>H47_TEST</instance-name>
                        <server-name>poolhp3</server-name>
                        <technology-code>SP</technology-code>
                    </instance>
                    <instance>
```

```
                <instance-name>H47_TEST2</instance-name>
                <server-name>poolhp3</server-name>
                <technology-code>SP</technology-code>
            </instance>
        </affected-instances>
    </permission>
</role>
<role role-name="monitor sql apptier" role-scope="APPTIER">
    <permission>
        <description>'Monitor.View' permission on the selected Tiers</description>
        <permission-type>MONITOR</permission-type>
        <permission-operation>VIEW</permission-operation>
        <resource resource-type="APPTIER" environment-name="Default" apptier-name="SQL Server" />
        <affected-instances>
            <instance>
                <instance-name>PIFA1000</instance-name>
                <server-name>pifa1000</server-name>
                <technology-code>SQ</technology-code>
            </instance>
        </affected-instances>
    </permission>
    <permission>
        <description>'Monitor.Execute' permission on the selected Tiers</description>
        <permission-type>MONITOR</permission-type>
        <permission-operation>EXECUTE</permission-operation>
        <resource resource-type="APPTIER" environment-name="Default" apptier-name="SQL Server" />
        <affected-instances>
            <instance>
                <instance-name>PIFA1000</instance-name>
                <server-name>pifa1000</server-name>
                <technology-code>SQ</technology-code>
            </instance>
        </affected-instances>
    </permission>
</role>
<role role-name="administrate sql instance" role-scope="INSTANCE">
    <permission>
        <description>'Administrate.Execute' permission on the selected instances</description>
        <permission-type>ADMINISTRATE</permission-type>
        <permission-operation>EXECUTE</permission-operation>
        <resource resource-type="INSTANCE" instance-name="PIFA1000" server-name="pifa1000" technology-code="SQ" />
        <affected-instances>
            <instance>
                <instance-name>PIFA1000</instance-name>
                <server-name>pifa1000</server-name>
                <technology-code>SQ</technology-code>
            </instance>
        </affected-instances>
    </permission>
    <permission>
        <description>'Administrate.Full Control' permission on the selected instances</description>
        <permission-type>ADMINISTRATE</permission-type>
        <permission-operation>FULL_CONTROL</permission-operation>
        <resource resource-type="INSTANCE" instance-name="PIFA1000" server-name="pifa1000" technology-code="SQ" />
        <affected-instances>
            <instance>
                <instance-name>PIFA1000</instance-name>
                <server-name>pifa1000</server-name>
                <technology-code>SQ</technology-code>
            </instance>
        </affected-instances>
    </permission>
    <permission>
        <description>'Administrate.View' permission on the selected instances</description>
        <permission-type>ADMINISTRATE</permission-type>
        <permission-operation>VIEW</permission-operation>
        <resource resource-type="INSTANCE" instance-name="PIFA1000" server-name="pifa1000" technology-code="SQ" />
        <affected-instances>
            <instance>
                <instance-name>PIFA1000</instance-name>
                <server-name>pifa1000</server-name>
                <technology-code>SQ</technology-code>
            </instance>
        </affected-
```