CHAPTER 19

Cisco BAC Support Tools and Advanced Concepts

This chapter contains information on, and explains the use of, tools that help you maintain Cisco Broadband Access Center (BAC) as well as speed and improve the installation, deployment, and use of this product.

This chapter discusses these topics:

- Using the deviceExport.sh Tool, page 19-1
- Using the disk_monitor.sh Tool, page 19-4
- Using the resetAdminPassword.sh Tool, page 19-5
- Using the runEventMonitor.sh Tool, page 19-5
- Using the changeARProperties.sh Tool, page 19-8
- Using the changeNRProperties.sh Tool, page 19-9

For a list of other tools that are supported in this release, see Cisco BAC Tools, page 9-5.

Note This section contains examples of tool use. In many cases, the tool filenames include a path specified as *BPR_HOME*. This indicates the default installation directory location.

Using the deviceExport.sh Tool

You can obtain information about devices by using the device export tool, which retrieves device information from the Cisco BAC system and exports it to a flat file. This file can, in turn, be used to import data into an external application.

The **deviceExport.sh** tool, located at the BPR_HOME/rdu/bin directory, exports device information from the backup snapshot of the RDU database to a Comma Separated Value (CSV) format file.

Note You can use the device export tool only on the backup database; the tool does not export device information from the live RDU database.

You must provide a list of device properties that are to be exported in the control file. The control file is an XML file which defines the fields required for export. The tool provides an option to generate a sample control file, which you can edit to configure which properties to export.

You can generate the list of properties predefined in Cisco BAC and available for export by running the **deviceExport.sh -samplectrl** command. (For sample control output, see Example 19-2.)

The CSV format is used widely to exchange data between applications. Keep the following rules in mind about a CSV format file:

- Each device outputs to one line.
- Each line terminates with the UNIX format line separator (\n).
- Each field is separated by a comma (,).
- If a field contains white space, a comma, or a line separator, it is enclosed by double quotes ("). If a field contains double-quotes, repeat the character twice to escape it; for instance, "file name" becomes ""file name"".
- A boolean field outputs as *true* or *false*.
- A byte array outputs to a string with UTF-8 encoding.
- If a field is a list, it converts into a formatted string with each item separated by comma; for instance, a node list outputs as "node1, node2, node3".
- If a field is a map, it is converted into a long string. The key and data is separated by a comma; for instance, a map output looks like: "(key1, data1)(key2, data2)(key3, data3)".
- If the field value is null or does not exist, the output is an empty string followed by a comma.
- The first line is the field name separated by a comma.
- There is no comma at the end of each record.

Example 19-1 Sample CSV Format

```
74:7b:7b:f0:e7:80,admin,true,2,"node1,node2,node3","(prop1,value1)(prop2,value2)",,,
```

Syntax Description

To use the **deviceExport.sh** command, use this syntax:

```
# ./deviceExport.sh [-help] [-samplectrl] controlfile backupdir outputdir
```

- *controlfile*—Identifies the path to the control file, which defines the fields required for export.
- *backupdir*—Identifies the path to the directory, which contains backed-up database files that are to be used as data source. (To back up your database, use the **backupDb.sh** tool; see Backup and Recovery, page 10-4.)
- *outputdir*—Identifies the target location for the output files. If the directory does not exist, a new directory is created.
- **help**—Generates tool usage information.
- **samplectrl**—Generates the sample control file, which contains the supported properties and device types, in the current directory. The control file is an XML file that contains the supported properties and device types. You can remove unwanted properties or choose to export only certain types of devices by editing the XML file. See Example 19-2 for output of a sample control file.

Example 19-2 Sample Control File

```
# ./deviceExport.sh -samplectrl
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE CONTROLFILE SYSTEM "device-export-control.dtd">

<!--SAMPLE CONTROL FILE-->
```

```
<CONTROLFILE>

    <!--Start of field list(REQUIRED)
        The field list specifies the device properties that will be exported.
        All supported standard fields are listed below. Remove unwanted
        fields by deleting the line that contains the field name. Customer
        defined properties are not listed but can be added to the list.
    -->
    <FIELDLIST>
        <FIELD>GenericObjectKeys.OID_REVISION_NUMBER</FIELD>
        <FIELD>DeviceDetailsKeys.DEVICE_TYPE</FIELD>
        <FIELD>DeviceDetailsKeys.OWNER_ID</FIELD>
        <FIELD>DeviceDetailsKeys.NODE_DETAILS</FIELD>
        <FIELD>DeviceDetailsKeys.DEVICE_ID</FIELD>
        <FIELD>DeviceDetailsKeys.FQDN</FIELD>
        <FIELD>DeviceDetailsKeys.HOST</FIELD>
        <FIELD>DeviceDetailsKeys.DOMAIN</FIELD>
        <FIELD>DeviceDetailsKeys.IS_IN_REQUIRED_PROV_GROUP</FIELD>
        <FIELD>DeviceDetailsKeys.IS_REGISTERED</FIELD>
        <FIELD>DeviceDetailsKeys.IS_PROVISIONED</FIELD>
        <FIELD>DeviceDetailsKeys.PROV_GROUP</FIELD>
        <FIELD>DeviceDetailsKeys.CLASS_OF_SERVICE</FIELD>
        <FIELD>DeviceDetailsKeys.CLASS_OF_SERVICE_SELECTED</FIELD>
        <FIELD>DeviceDetailsKeys.PROPERTIES</FIELD>
        <FIELD>DeviceDetailsKeys.PROPERTIES_DETECTED</FIELD>
        <FIELD>DeviceDetailsKeys.PROPERTIES_SELECTED</FIELD>
        <FIELD>DeviceDetailsKeys.REASON</FIELD>
        <FIELD>DeviceDetailsKeys.EXPLANATION</FIELD>
        <FIELD>DeviceDetailsKeys.CONFIGURATION_REVISION</FIELD>
        <FIELD>DeviceDetailsKeys.FIRMWARE_CONFIGURATION_REVISION</FIELD>
        <FIELD>DeviceDetailsKeys.REPORTED_IP_ADDRESS</FIELD>
        <FIELD>DeviceDetailsKeys.SOURCE_IP_ADDRESS</FIELD>
        <FIELD>DeviceDetailsKeys.ROUTABLE_IP_ADDRESS</FIELD>
        <FIELD>DeviceDetailsKeys.DEVICE_FAULTS</FIELD>
        <FIELD>DeviceDetailsKeys.PENDING_ON_CONNECT_OPERATION_IDS</FIELD>
        <FIELD>DeviceDetailsKeys.PASSWORD_IS_PROTECTED</FIELD>
        <FIELD>IPDeviceKeys.HOME_PROV_GROUP</FIELD>
        <FIELD>IPDeviceKeys.CPE_PASSWORD</FIELD>
        <FIELD>IPDeviceKeys.CONNECTION_REQUEST_USERNAME</FIELD>
        <FIELD>IPDeviceKeys.CONNECTION_REQUEST_PASSWORD</FIELD>
    </FIELDLIST>
    <!--End of field list-->

</CONTROLFILE>
```

Note The `DOCTYPE CONTROLFILE SYSTEM` references a .dtd file, device-export-control.dtd, which is used for XML validation. The file is installed in the BPR_HOME/rdu/bin directory.

Example 19-3 Exporting Data from Backup Snapshot

This is an example of exporting data from a backup snapshot:

```
# ./deviceExport.sh control.xml rdu-backup-20061227-145538 /data/rduexport
Starting exporting devices...

Using backup database in /tmp/rdu-backup-20061227-145538
Device export finished in 28m11s.
```

Note The exported file is generated in the specified directory; in the above example, in the /data/rduexport directory. You do not need to specify the full path to the directory.

Following a successful export from the Cisco BAC backup database, the Device Export tool creates a device file, which contains the list of device records that are successfully exported from the Cisco BAC backup database. The filename is bac-device-details-*yyyyMMdd-HHmmss*.csv:

Where *yyyyMMdd-HHmmss* identifies the time the file was generated.

Using the disk_monitor.sh Tool

Monitoring available disk space is an important system administration task. You can use a number of custom written scripts or commercially available tools to do so. The disk_monitor.sh tool is a sample tool to accomplish this.

The **disk_monitor.sh** tool, located in the BPR_HOME/rdu/sample/tools directory, sets threshold values for one or more file systems. When these thresholds are surpassed, an alert is generated through the syslog facility, at 60-second intervals, until additional disk space is available.

Note Cisco recommends that, at a minimum, you use the disk_monitor.sh script to monitor the BPR_DATA and BPR_DBLOG directories.

Syntax Description

```
# ./disk_monitor.sh file system-directory x
```

- *file system-directory*—Identifies any directory in a file system to monitor.
- *x*—Identifies the percentage threshold applied to the specified file system.

Example 19-4 Monitoring Disk Space

Assume that you want to be notified when a file system (*/var/CSCObac,* for example) with database logs reaches 80% of its capacity. Enter the command:

```
# ./disk_monitor.sh /var/CSCObac 80&
```

When the database logs disk space reaches 80% capacity, an alert is sent to the syslog file:

```
Dec 7 8:16:03 perf-u80-1 BPR: [ID 702911 local6.warning] File system /var/bpr usage is 81%
(threshold is 80%)
```

Note Make sure to configure Solaris or Linux to this on start-up, so that it is started after system reboots automatically.

Using the resetAdminPassword.sh Tool

Cisco BAC supports both local authentication and TACACS+ authentication and you can use the **resetAdminPassword.sh** tool to reset the password for both local and TACACS+ authentication. Run the **resetAdminPassword.sh** tool from the BPR_HOME/rdu/internal/db/bin directory.

When the authentication is local if you use the **resetAdminPassword.sh** tool, it allows you to login with the default password **changeme**. After logging in, it will prompt you to change the password and you can change the password as you want.

When TACACS+ authentication is enabled, you can use the **resetAdminPassword.sh** tool to temporarily reset bacadmin authentication to local RDU. This tool allows you to login once with the password **changeme** to the RDU using the local reset admin password. After logging in, it will prompt you to change the password where you have to use the **changeme** password again. Then in the TACACS+ Defaults page you can change the password, as you want.

To enable local authentication, you have to manually change the authentication mode setting, after you login to the RDU. Otherwise, it will automatically fall back to TACACS+ authentication mode from next login. To enable local authentication:

Step 1 Choose **Configuration** on either the Primary Navigation bar or Main Menu page.

Step 2 Choose **Defaults** from the Secondary Navigation bar.

The Configure Defaults page appears.

Step 3 Click TACACS+ Defaults link on the left pane.

The TACACS+ Defaults page appears

Step 4 Check the TACACS+ Authentication Disabled check box.

Step 5 Click **Submit**.

Note The authentication setting for other Cisco BAC users will not be affected by this tool. You must use the TACACS+ server administrative procedure to change the passwords.

Using the runEventMonitor.sh Tool

You can run the **runEventMonitor.sh** tool to view the events that are being fired in Cisco BAC. You can run this tool from the *BPR_HOME/rdu/internal/bin* directory.

Table 19-1 describes the types of events that you can view from the event monitor:

Table 19-1 Events that can be viewed from the Event Monitor

Event	Sub-Event	Description
Batch	Completion	Displays when a batch submitted by a client application ends. Contains the batch status.
Class of service	New	Indicates when a class of service is added to the system.

Table 19-1 Events that can be viewed from the Event Monitor (continued)

Event	Sub-Event	Description
Class of service	Deleted	Indicates when a class of service is deleted from the system.
Configuration	Generated	Indicates when a configuration is generated.
Configuration	Uncommitted Generated	Indicates when a configuration that is temporarily stored at the DPE is generated.
Configuration	Rollback Uncommitted	Indicates that the uncommitted configuration should be discarded from the DPE.
Device	Changed Class Of Service	Indicates when a device changes its Class of Service.
Device	Changed IP Address	Indicates when a device's IP address changes.
Device	Deleted	Indicates when a device is deleted.
Device	Deleted Voice Service	Indicates when a voice service is deleted from a device.
Device	New Provisioned Device	Indicates when a device is added through the provisioning API.
Device	New Unprovisioned Device	Indicates when a device is added when booting on the network.
Device	New Voice Service	Indicates when a voice service is added to a device
Device	Roaming	Indicates when a device roams provisioning groups.
DHCP Criteria	New	Indicates when a DHCP criteria is added to the system.
DHCP Criteria	Deleted	Indicates when a DHCP criteria is deleted from the system.
External File	Added	Indicates when a file is added to the system.
External File	Deleted	Indicates when a file is deleted from the system.
External File	Replaced	Indicates when a file is replaced in the system.
Messaging	Connection Up	Indicates when a connection on the local instance of the messaging system starts.
Messaging	Connection Down	Indicates when a connection on the local instance of the messaging system stops.
Messaging	Queue Full	Indicates when the queue on the local instance of the messaging system is full and starts dropping messages.

Table 19-1 Events that can be viewed from the Event Monitor (continued)

Event	Sub-Event	Description
Provisioning Group	Changed	Indicates when the provisioning group is changed.
Server Properties	Common Properties	Indicates when common properties that effect the RDU or DPE change.
System Configuration	Server Defaults Changed	Indicates when properties are changed on an user, RDU, or DPE.
System Configuration	System Configuration Changed	Indicates when the system configuration is changed.
System Configuration	System Defaults Changed	Indicates when defaults are changed.

Syntax Description

To run the event monitor, enter:

```
# /opt/CSCObac/rdu/internal/bin/runEventMonitor.sh [options]
```

Options are used to specify the RDU connection parameters and amount of output. You have the following options:

- **-noverbose**—Forces the event monitor to display only the types of events being fired, not their contents.
- **-host** *host*—Specifies the host where the RDU is located. Default is the localhost.
- **-port** *port*—Specifies the port on which the RDU is listening. Default is 49187.

Example 19-5 Sample Event Monitor Output.

```
If need help, please restart command with '?' parameter.

Verbose mode: true
RDU host: localhost
RDU port: 49187

Connecting to RDU...ok

Listening for events...

ExternalFileEvent added filename=gold.cm
    rev=1014671115124(Mon Feb 25 16:05:15 EST 2002)
    source=BPR Provisioning API:BPR Regional Distribution Unit:AddExternalFile command

DeviceEvent newProvDevice ID=1,6,01:02:03:04:05:06
    rev=1014671179380(Mon Feb 25 16:06:19 EST 2002)
    source=BPR Provisioning API:BPR Regional Distribution Unit:AddIPDevice command IP=null
    FQDN=null group=null
```

Using the changeARProperties.sh Tool

The Cisco BAC installation program establishes values for configuration properties used by Cisco BAC extensions that are incorporated into the Access Registrar RADIUS server. You use the **changeARProperties.sh** command, which is found in the BPR_HOME/car_ep/bin directory, to change key configuration properties.

Invoking the script without any parameters displays a help message listing the properties that can be set.

To run this command:

Step 1 Change directory to BPR_HOME/car_ep/bin.

Step 2 Run the **changeARProperties.sh** command using this syntax:

changeNRProperties.sh options

Where options are:

– -**help**—Displays this help message. The **-help** option must be used exclusively. Do not use this with any other option.

– -**d**—Displays the current properties. The **-d** option must be used exclusively. Do not use this with any other option.

– -**s** *secret*—Identifies the Cisco BAC shared secret. For example, if the shared secret is the word secret, enter **-s secret**.

– -**q** *queue_size* —Identifies the maximum number of the Access-Requests to be queued. For example, if you want to set the queue size to 1000, enter **-q 1000.**

– -**t** *thread_count*—Identifies the maximum number of threads to be used to process the requests. For example, if you want to set the thread count to 10, enter -**t 10.**

– -**host** *addr*—Identifies the DPE Auth HTTP interface you want to use. For example, if you want to use the interface localhost, enter **-host localhost.**

– -**port** *port*—Identifies the DPE Auth HTTP interface port you want to use. For example, if you want to use the port 7551, enter **-port 7551.**

– -**url** *url*—Identifies the DPE Auth HTTP service url you want to use. For example, if you want to use the url /auth, enter **-url /auth.**

– -**ssl** *enabled|disabled*—Enables or disables the SSL/TLS connection between CAR and DPE. Enter **-ssl enabled** to enable it and **-ssl disabled** to disable it.

– **-x** *[true/false]*—Enables or disables XML Schema validation e.g. **-x true**

Using the changeNRProperties.sh Tool

The Cisco BAC installation program establishes values for configuration properties used by Cisco BAC extensions that are incorporated into the Network Registrar DHCP server. You use the **changeNRProperties.sh** command, which is found in the BPR_HOME/cnr_ep/bin directory, to change key configuration properties.

Invoking the script without any parameters displays a help message listing the properties that can be set.

To run this command:

Step 1 Change directory to BPR_HOME/cnr_ep/bin.

Step 2 Run the **changeNRProperties.sh** command using this syntax:

```
changeNRProperties.sh options
```

Where options are:

- **-help**—Displays this help message. The **-help** option must be used exclusively. Do not use this with any other option.
- **-d**—Displays the current properties. The**-d** option must be used exclusively. Do not use this with any other option.
- **-s** *secret*—Identifies the Cisco BAC shared secret. For example, if the shared secret is the word secret, enter **-s secret.**
- **-f** *fqdn*—Identifies the RDU FQDN. For example, if you use rdu.example.com as the fully qualified domain name, enter **-f rdu.example.com**.
- **-p** *port*—Identifies the RDU port you want to use. For example, if you want to use port number 49187, enter **-p 49187**.
- **-g** *prov_group*—Identifies the provisioning group. For example, if you are using provisioning group called group1, enter **-g group1.**

Examples

This is an example of changing the Network Registrar extensions by using the NR Extensions Properties tool:

```
#sh changeNRProperties.sh -g primary1
Current NR Properties:
RDU Port: 49187
RDU FQDN: rdu.example.com
Provisioning Group: primary1
Shared Secret: fggTaLg0XwKRs
```

Note You must restart your NR DHCP server for the changes to take effect.

This is an example of viewing the current properties:

```
#sh changeNRProperties.sh -d
Current NR Properties:
RDU Port: 49187
RDU FQDN: rdu.example.com
Provisioning Group: primary1
Shared Secret: fggTaLg0XwKRs
```