Autodesk®
SystemCentral 2010
Extension 1

User Guide

Autodesk®

Autodesk® Visual Effects and Finishing 2010 Extension 1

Trademarks

Disclaimer

Published by: Autodesk, Inc.
111 McInnis Parkway
San Rafael, CA 94903, USA

Title: Autodesk SystemCentral 2010 Extension 1 User Guide
Document Version: 2
Date: November 12, 2009

Contents

Introduction 1

Topics in this chapter:

About SystemCentral

SystemCentral is a tool that combines the usefulness and power of command-line utilities with the ease of use of a Web-based interface. It enables you to access system information and run tests remotely on any Autodesk® Visual Effects, Finishing, and Colour Grading (Autodesk® Inferno®, Autodesk® Flame®, Autodesk® Flint®, Autodesk® Flare™, Autodesk® Smoke®, Autodesk® Backdraft® Conform, and Autodesk® Lustre®) Linux® workstation in your network.

Its modular, plug-in based architecture allows you to expand its capabilities by creating new plug-ins.

New in SystemCentral 2010 Extension 1

This section lists the new features and plug-ins introduced in the current version of SystemCentral. Follow the links for more information about the features.

- You can now save plug-in graphs as PNG images. See Graph View on page 18.
- The new *Platform Information* plug-in in the *Support* category displays information on the workstation hardware and Operating System.

Architectural Overview

If this is the first time you are using SystemCentral, read this section to familiarize yourself with the key components of the application.

SystemCentral features a Web-based architecture that allows you to access its graphical user interface from any computer in your network. This architecture eliminates the need to be at a Visual Effects and Finishing workstation to get hardware and system information, or to run various system tests.

SystemCentral comprises four main components.

SystemCentral Framework The main component of the system. It installs on the workstation and enables users to run one or more utilities on the workstation, either from a local terminal or from any compatible Web browser on the network.

SystemCentral Plug-ins Represent the actual utilities that are loaded into the framework and perform the tests, or retrieve the system information requested by the user.

SystemCentral comes with several predefined plug-ins, which provide system hardware and software information, and allow you to run various performance tests. You can create additional plug-ins using the application's plug-in editor.

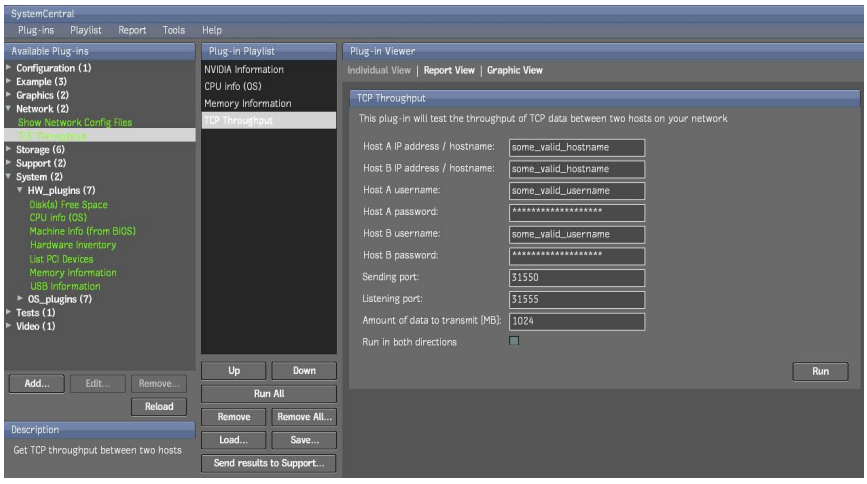
SystemCentral Database A database that runs on the workstation and stores the results of executed plug-ins that support this feature. Storing plug-in results to the database allows you to generate graphs that compare results from multiple executions of the same plug-in. The SystemCentral user interface provides an easy way of performing database cleanup from time to time. See SystemCentral Database Clean-up on page 33.

SystemCentral Graphical User Interface An Adobe® Flash® Player compatible interface that can be accessed through any compatible Web browser, on any computer in your network. It communicates with the SystemCentral Framework via the HTTP protocol. The SystemCentral graphical user interface allows you to create or edit plug-ins, run them, and view their results.

SystemCentral User Interface

If this is the first time you are using SystemCentral, read this section to familiarize yourself with the SystemCentral graphical user interface.

The SystemCentral graphical user interface enables you to browse the list of available plug-ins, select the ones you want to run, set their parameters if needed, and view their results.



Available Plug-ins

The Available Plug-ins panel displays the plug-ins that are currently installed in SystemCentral.

When the application starts, the available plug-ins are automatically loaded and organized into several categories, based on the directory structure of the plug-in files on disk.

Each branch of the plug-in hierarchy displays the number of plug-ins available in that category.

Plug-in Playlist

The playlist contains the plug-ins you intend to run. The playlist is manually populated by adding plug-ins from the Available Plug-ins list. The playlist enables you to run selected plug-ins or all the plug-ins you added to it.

You can save the current playlist and load previously-saved playlists.

Plug-in Viewer

The Plug-in Viewer panel has multiple uses:

- Individual View enables you to view the description of the selected plug-in in the playlist, set its parameters if necessary, run it, and view its results.
- Report View displays an aggregated report of the results of all the plug-ins that were executed.
- Graph View displays graphs that allow you to compare results saved in the SystemCentral database from multiple executions of a plug-in.

Menu Bar

The SystemCentral menu bar enables you to set application preferences, view HTML reports of the executed plug-ins, view the application log, perform database clean-up, and consult the SystemCentral online help.

About this Guide

This guide provides information on installing, configuring and using SystemCentral, as well as instructions on creating custom plug-ins.

Notation Conventions

A number of style conventions are used throughout your documentation. These conventions and examples of their use are shown as follows.

Convention	Example
Text that you enter in a command line or shell appears in Courier bold. Press the Enter key after each command.	**`install rpm -qa`**
Variable names appear in Courier, enclosed in angle brackets.	`<filename>`
Feedback from the command line or shell appears in Courier.	`limit coredumpsize`
Directory names, filenames, URLs, and command line utilities appear in italics.	*`/usr/discreet`*

Related Documentation

Documentation for this release is installed with the product as PDF files and as an HTML help system, and is also available on the Autodesk web site at *http://www.autodesk.com/me-documentation*. From this page you can access the complete documentation library.

You should also refer to the product release notes for all late-breaking release information.

Contacting Customer Support

For Autodesk Media and Entertainment Customer Support, visit *http://www.autodesk.com/support*.

Customer support is also available through your Autodesk reseller. To find a reseller near you, consult the reseller look-up database at *http://www.autodesk.com/resellers*.

Getting Started

2

Topics in this chapter:

Installing SystemCentral

The current version of SystemCentral is available as a download from Autodesk. Install SystemCentral on the Visual Effects, Finishing and Colour Grading workstation you want to diagnose.

SystemCentral requires the Apache Web Server version 2.0.52 or greater, and PostgreSQL 7.4.8 or greater. Apache and PostgreSQL should already be installed and configured by default with Red Hat® Enterprise Linux on your workstation. For details, consult your Red Hat Enterprise Linux documentation.

To install SystemCentral:

1. Open a terminal and log in as root.
2. Download the SystemCentral installation package. The download link is provided in the Release Announcement you received from Autodesk.
3. Unpack the downloaded *tar* file, and navigate to the newly created directory. For example:

   ```
   tar -zxvf SystemCentral_2010.1_LINUX64_RHEL.tar.gz
   cd SystemCentral_2010.1_LINUX64_RHEL
   ```
4. Run the installation script:

   ```
   ./INSTALL
   ```

   The installation starts. A window appears asking you to create a SystemCentral administrator password.
5. Type an administrator password for SystemCentral.

This password will be used to authenticate users when creating plug-ins that require root permissions to run.

TIP To avoid potential security risks, do not use the Linux root password as the SystemCentral administrator password.

SystemCentral is installed and automatically configured for the Apache Web server. The SystemCentral database is automatically installed and configured. No additional configuration is necessary.

Browser Configuration

The SystemCentral graphical user interface runs in any Web browser that supports the Adobe® Flash Player plug-in, version 9 or later. This includes Mozilla® Firefox® 1.x or later, Apple® Safari™ 1.x or later, and Microsoft® Internet Explorer 6 or later.

If you already have the Adobe Flash Player plug-in installed for your browser, you don't need to perform any additional configuration to use SystemCentral.

If your browser does not have the Adobe Flash Player plug-in, you can download it for free from the Adobe Web site.

NOTE Currently, there is no Adobe Flash Player plug-in for 64-bit Linux browsers. To be able to access SystemCentral from the browser of your Visual Effects and Finishing workstation, perform the following procedure.

To be able to use the SystemCentral client on a Visual Effects and Finishing workstation:

1. Install the latest version of the *Discreet Kernel Utility* (DKU). The DKU automatically installs the 32-bit version of Mozilla Firefox 2 on your workstation. For instructions about installing the DKU, see the latest *Autodesk Visual Effects and Finishing Installation and Configuration Guide*.
2. Using Firefox 2, browse to the Adobe Web site to download the latest version of the Adobe Flash Player plug-in.
3. Follow the instructions on the Adobe Web site to install the plug-in for your browser.

   Once the plug-in is installed, you are able to access SystemCentral from your Visual Effects and Finishing workstation.

Accessing SystemCentral

There are two ways to access SystemCentral. You can access it locally from a Linux terminal, or you can access the SystemCentral graphical user interface (GUI) from a compatible Web browser on any computer in your network.

Accessing SystemCentral from a Terminal

The SystemCentral command line interface provides a quick and easy way to run plug-ins from a terminal on the workstation where SystemCentral is installed.

The command line interface is especially useful to run plug-ins as part of batch shell scripts or scheduled tasks.

For information on running SystemCentral plug-ins or playlists from a terminal, see Running Plug-ins from a Terminal on page 13.

Accessing the SystemCentral Graphical User Interface

To access the SystemCentral User Interface:

1. Open the browser and point it to the workstation where SystemCentral is installed:

   `https://<workstation_host_name>`

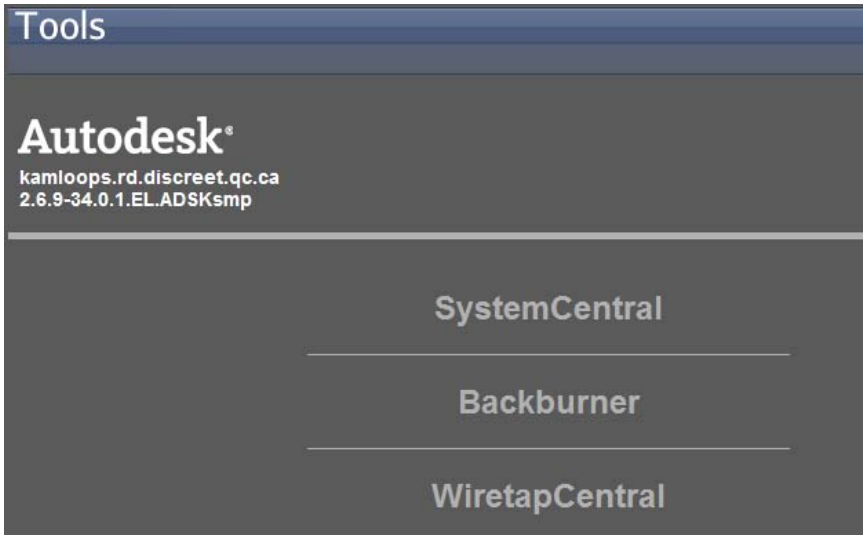
   where `<workstation_host_name>` represents the host name or IP address of the workstation. For example:

   `https://myworkstation.autodesk.com`

2. Optional: Accept the Web site security certificate, if necessary.

   NOTE If you are accessing SystemCentral from Microsoft Internet Explorer, and receive a message that there is a problem with the site's security certificate, click the "Continue to this website" link.

   The Autodesk Tools page opens and displays all the Autodesk Web applications available on the workstation.



   NOTE The page also displays the workstation host name, and the version of the Linux kernel, for example *2.6.9-34.0.1.EL.ADSKsmp*.

3. Click the SystemCentral link.

   The SystemCentral user interface appears.

Running Plug-ins

3

Topics in this chapter:

Workflow for Running Plug-ins

The following steps are the recommended workflow when using SystemCentral plug-ins.

To run SystemCentral plug-ins:

1. Select the plug-ins you want to run and add them to the current playlist. See Adding Plug-ins to the Playlist on page 10.
2. Optional: Set the parameters of the plug-ins you want to run. See Setting Plug-in Parameters on page 11.
3. Optional: Save the playlist if you want to reuse it later, or if you want to be able to run it from a terminal. See Saving and Loading Playlists on page 11.
4. Run the selected plug-ins or the entire playlist, either from the graphical user interface, or from a terminal. See Running Plug-ins from the Graphical User Interface on page 12, and Running Plug-ins from a Terminal on page 13.

Adding Plug-ins to the Playlist

The Available Plug-ins list contains all the plug-ins available to SystemCentral.

To view detailed information about a plug-in, set its parameters, and run it, you must add it to the Plug-in Playlist.

To add a plug-in to the playlist:

1. In the Available Plug-ins list, browse to the plug-in you intend to run. The list organizes plug-ins by category.
2. Use the arrow icons to expand or collapse the branches of the plug-in hierarchy. Some categories have several branches that represent various presets of a plug-in. Select the preset that best suits your needs.

   TIP You can also double-click a category to expand or collapse it.

3. Optional: To see a short description of a plug-in before adding it to the playlist, select it in the Available Plug-ins list. A description of the plug-in is displayed in the Description panel below the list.

   NOTE Plug-ins that might have a negative impact on your system are displayed in orange. You receive a warning when attempting to run such a plug-in. Read the warning and assess the potential risks before proceeding.

4. To add a plug-in to the playlist, double-click it in the Available Plug-ins list, or drag and drop it into the Plug-in Playlist. You may select multiple plug-ins by holding **Shift** or **Ctrl** while clicking the plug-ins. You can also drag entire categories of plug-ins into the playlist.

After you have added the plug-in to the playlist, the Plug-in Viewer panel displays a more detailed description of the plug-in allowing you to configure and run it. See Running Plug-ins on page 9.

NOTE You can change the order of plug-ins in the playlist by either dragging them to a new position, or by selecting one or more plug-ins and using the Up and Down buttons at the bottom of the playlist to change their position.

Removing Plug-ins from the Playlist

The playlist buttons enable you to remove only the selected plug-ins, or all plug-ins from the playlist.

If you remove a plug-in that has already been executed, its settings and results are also removed from the Plug-in Viewer. Save or print the plug-in results before you remove a plug-in from the playlist.

To remove selected plug-ins from the playlist:

1. Select the plug-in you want to remove from the playlist. Hold **Ctrl** or **Shift** to select multiple entries or ranges.
2. Click Remove at the bottom of the playlist, or click Playlist, then click Remove in the SystemCentral main menu.

   The selected plug-ins are removed from the playlist, and their results are removed from the Plug-in Viewer panel.

To remove all plug-ins from the playlist:

➤ Click Remove All at the bottom of the playlist, or select Playlist | Remove All from the application menu.

All the plug-ins and their results are removed from the playlist and from the Plug-in Viewer.

Setting Plug-in Parameters

Once you have added one or more plug-ins to the playlist, you can view their description and set their parameters, if necessary.

To view the details of a plug-in and set its parameters:

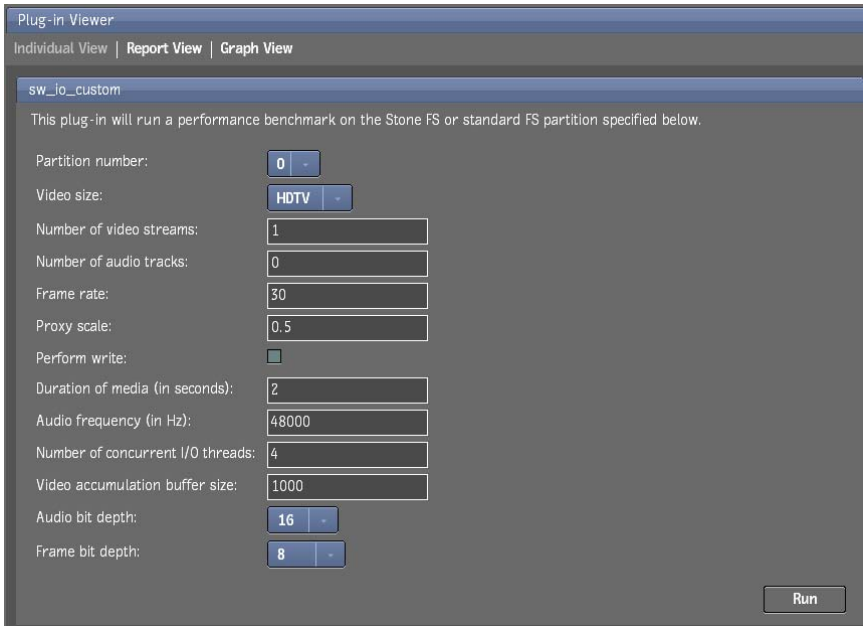
1 Select the plug-in in the playlist.

The Individual View tab of the Plug-in Viewer displays a description of the selected plug-in.

NOTE If you select multiple plug-ins in the playlist, only the description of the last plug-in in the selection is displayed in the Plug-in Viewer.

2 If the plug-in can be configured, use the fields in the Individual View to modify the parameters, or use the default values.

NOTE The new parameters of the plug-in are saved with the playlist. Removing the plug-in from the playlist discards all custom parameters.



Saving and Loading Playlists

SystemCentral allows you to save the current playlist for later use.

NOTE Playlists are saved on the SystemCentral workstation, not on the system you are accessing the SystemCentral from.

To save the current playlist:

1 Click Save at the bottom of the playlist, or click Playlist, and then Save in the SystemCentral main menu.

The Save Playlist dialog box appears.

2 Enter a name for the playlist, and the path to the directory where to save the playlist.

If you do not enter a path, the playlist is saved in */usr/discreet/SystemCentral/playlists/*.

NOTE Make sure the directory you are trying to save to has read, write, and execute permissions for all users (mode 777).

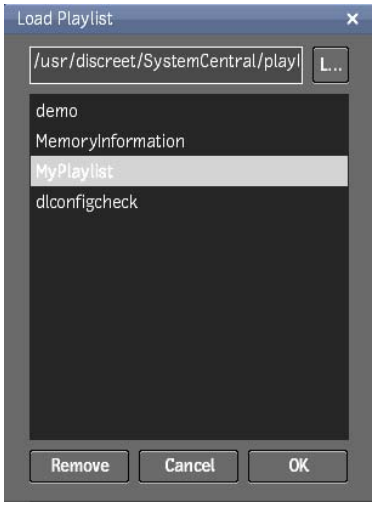
3 Click OK to save the playlist.

The playlist is saved, along with any custom parameters you set for the plug-ins in the playlist. If some of the plug-ins support the graph view feature, their graph settings are also saved with the playlist.

To load a previously-saved playlist:

1 Click Load at the bottom of the playlist, or click Playlist, and then Load in the SystemCentral main menu.

The Load Playlist window appears, and lists the playlists available in the default playlist directory */usr/discreet/SystemCentral/playlists/*.



2 Optional: To load a playlist from another directory, enter the path in the directory text box, and click the button to the right of the text box to refresh the list of playlists.

3 Optional: To delete a playlist, select it and click Remove. A confirmation dialog box appears before the file is deleted.

WARNING A playlist cannot be recovered once you delete it.

4 To load a playlist, select it and click OK.

The selected playlist is loaded and replaces the current playlist.

NOTE The newly loaded playlist does not display plug-ins that are no longer available (for example, if some plug-ins were deleted from SystemCentral or if the playlist was copied over from another workstation that had a different set of plug-ins).

Running Plug-ins from the Graphical User Interface

Use the following procedures to run a single plug-in or the entire current playlist from the SystemCentral graphical user interface.

To run a single plug-in:

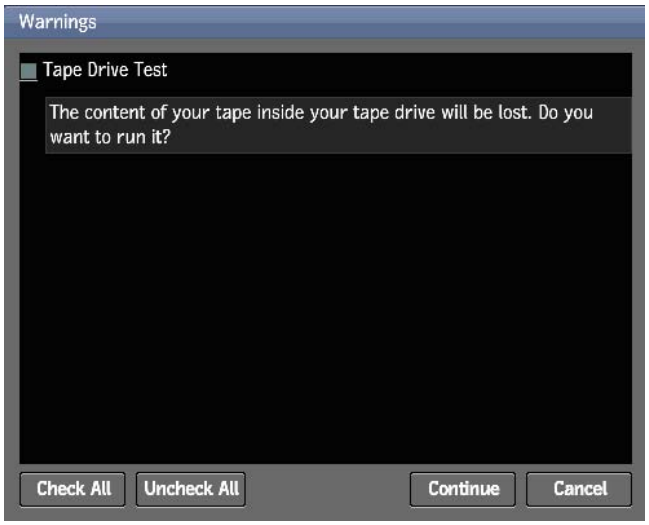
➤ Select the plug-in in the playlist, set its parameters if necessary, and click Run in the Individual View tab of the Plug-in Viewer.

NOTE If the plug-in can have a negative impact on your system, a warning is displayed prior to running the plug-in. Read the risks and confirm or cancel the plug-in execution accordingly.

The plug-in executes, and its results are displayed in the Individual View tab.

To run all plug-ins in the playlist:

1. Click Run All at the bottom of the playlist, or click Playlist, and then Run All in the SystemCentral menu.
2. Optional: If some plug-ins in the playlist can have a negative impact, a list of warnings is displayed, explaining the risk of each plug-in. Check the boxes next to the plug-ins you decide to run, and click Continue to execute the playlist.



While plug-ins are running, a progress dialog box is displayed. Depending on the number and nature of the plug-ins to be executed, this process could take several seconds.

3. Optional: To stop the execution of the plug-ins, click Cancel in the progress dialog box.

Running Plug-ins from a Terminal

Use the SystemCentral command line interface to run individual plug-ins or entire playlists from a local terminal on the SystemCentral workstation.

You can set the plug-ins to display their results in the terminal or to save them as text files. Result files are located in the */usr/discreet/SystemCentral/results/* directory and have the execution date and time as the filename.

The SystemCentral command line interface is especially useful for running SystemCentral plug-ins as part of batch scripts or scheduled tasks. For instructions on scheduling tasks in Linux, open a terminal and type **`man crontab`**.

NOTE For the list of options available in the command line version of SystemCentral, open a terminal and type **`SystemCentral --help`**.

Running Individual Plug-ins

Use the following procedures to list all the plug-ins available in SystemCentral, and to run one or more plug-ins from the terminal.

NOTE Executing individual plug-ins from the terminal is recommended only for plug-ins that do not require any parameters to be set. If you need to set parameters for some of the plug-ins, or if you need to execute more than a few plug-ins at a time, add them to a playlist and run the playlist. See Running Playlists on page 14.

To list available plug-ins:

➤ Open a terminal on the SystemCentral workstation and type:

`SystemCentral -l`

A list of all the categories (topics) and plug-ins available in SystemCentral is displayed. Each plug-in is listed with its path and name. For example:

```
[Topic] Video
            [Plugin] Video/videoIOSync (Video I/O Sync)
```

Write down the path (*Video/videoIOSync* in the previous example) of the plug-ins you plan to execute. You need to know the path in order to execute a plug-in from the terminal.

To run individual plug-ins:

➤ Open a terminal on the SystemCentral workstation, and type:

`SystemCentral -x` `<plug-in_path>` **`-o`** `<output_type>` **`-f`**

- `<plug-in_path>` is the path to the plug-in (for example *Video/videoIOSync*).
- `<output_type>` is *shell* or *file*.

  NOTE If `<output_type>` is not specified, it defaults to *shell*, and plug-in results are displayed in the terminal. You can use both *shell* and *file*, separated by a comma.

- Use the **`-f`** option at the end of the command line to ignore any warnings from plug-ins, and force their execution. If you do not use the **`-f`** option, a warning is displayed for each plug-in that could have a negative impact on your system, and you are asked to confirm their execution.

You can specify multiple plug-ins, separated by commas. For example:

`SystemCentral -x Graphics/getNvidiaInfo,System/HW_plugins/getCpuInfoFromOS`

The plug-ins are executed and their results are displayed in the terminal or saved to a text file, depending on the output type specified. Results are also saved in the SystemCentral database for plug-ins that support this feature.

Running Playlists

Running plug-ins as part of a playlist is more practical than running individual plug-ins, as it allows you to set plug-in parameters before execution.

After saving one or more playlists in the SystemCentral graphical user interface, use the following procedures to list all available playlists, and to run a playlist from the terminal. For information on creating playlists, see Adding Plug-ins to the Playlist on page 10.

To list all playlists:

➤ Open a terminal on the SystemCentral workstation, and type:

`SystemCentral -p`

A list of all the playlists available in SystemCentral is displayed. For example:

```
[1] MyPlaylist
[2] WeeklyTests
```

To run a playlist:

➤ Open a terminal, and type:

`SystemCentral -r` `<playlist_name>` **`-o`** `<output_type>`

- `<playlist_name>` is the name of an existing playlist.
- `<output_type>` is *shell* or *file*.

  NOTE If `<output_type>` is not specified, it defaults to *shell*, and plug-in results are displayed in the terminal. You can use both *shell* and *file*, separated by a comma.

- Use the **`-f`** option at the end of the command line to ignore any warnings from plug-ins, and force their execution. If you do not use the **`-f`** option, a warning is displayed for each plug-in that could have a negative impact on your system, and you are asked to confirm their execution.

Each plug-in in the playlist runs, and displays its results in the terminal, or saves them to a text file, depending on the output type specified. Results are also saved in the SystemCentral database for plug-ins that support this feature.

NOTE You can only run one playlist at a time.

Viewing Plug-in Results

4

Topics in this chapter:

Individual View

After each plug-in is executed, its results are displayed in the Individual View tab of the Plug-in Viewer.

Depending on the type of plug-in, the results can be displayed in a table or in a terminal-style panel. The text displayed in the terminal-style panel can be selected and copied to the clipboard.

To sort the rows in the result table:

1. To sort the results by name, click the heading of the Label column.

   An icon is displayed on the right-hand side of the column to indicate whether the sorting is ascending or descending.

2. To sort the results by value, click the heading of the Value column.

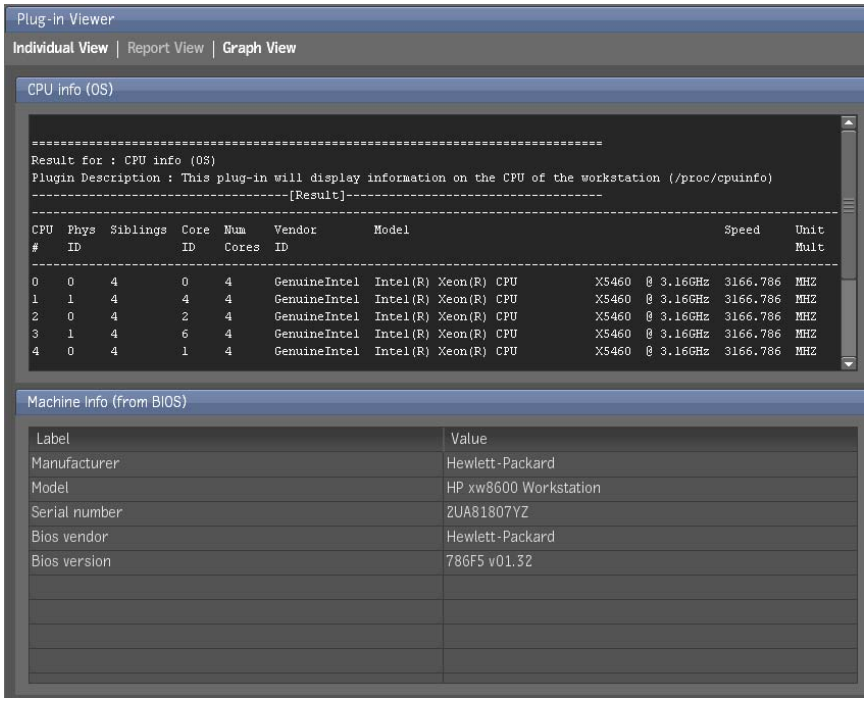
3. To toggle between ascending and descending sort order, click the column headings again.

NOTE You can change the order of the columns in the table by dragging a column heading on top of the other.

Report View

The Report View tab displays an aggregated report of all the plug-ins in the playlist that were executed.

The results of each plug-in in the Report View tab are displayed in the same format as in the Individual View.



NOTE If you remove a plug-in from the playlist after having executed it, its results are removed from Report View.

Graph View

Graph View allows you to generate graphs that compare results from several executions of a plug-in.

The data for the graph is retrieved from the SystemCentral database. The database stores the results of all plug-ins that support Graph View.

To support Graph View, a plug-in must meet the following two requirements:

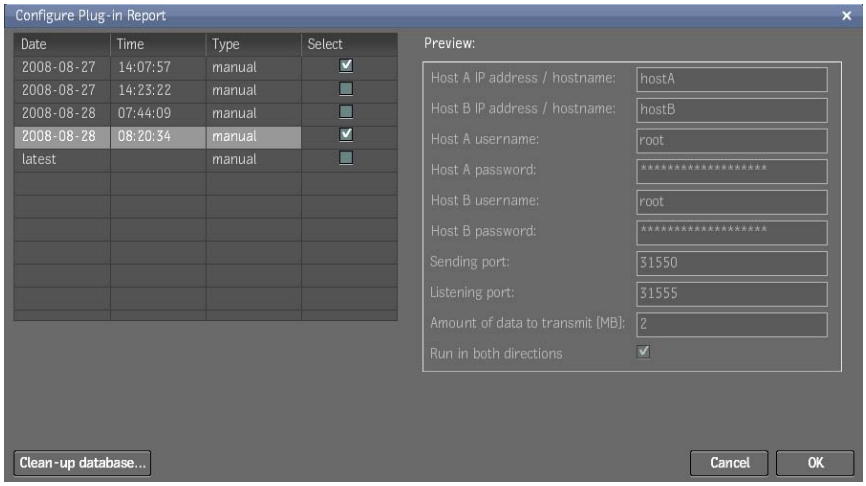
- The plug-in must use the SystemCentral API to return results. All plug-ins that display results in a table are API-compliant.
- All results of the plug-in must be numerical values.

To generate a graph for a plug-in:

1. Add the plug-in to the playlist. See Adding Plug-ins to the Playlist on page 10.
2. Optional: Run the plug-in to obtain a new set of results for the graph.
3. Click the Graph View tab of the Plug-in Viewer.

   Graph View displays a graph box for each supported plug-in in the playlist. By default, the graph boxes are empty. You need to set up the graph for a plug-in before it can be displayed.
4. Click Settings to customize and display the graph for the plug-in.

The Configure Plug-in Report window appears.



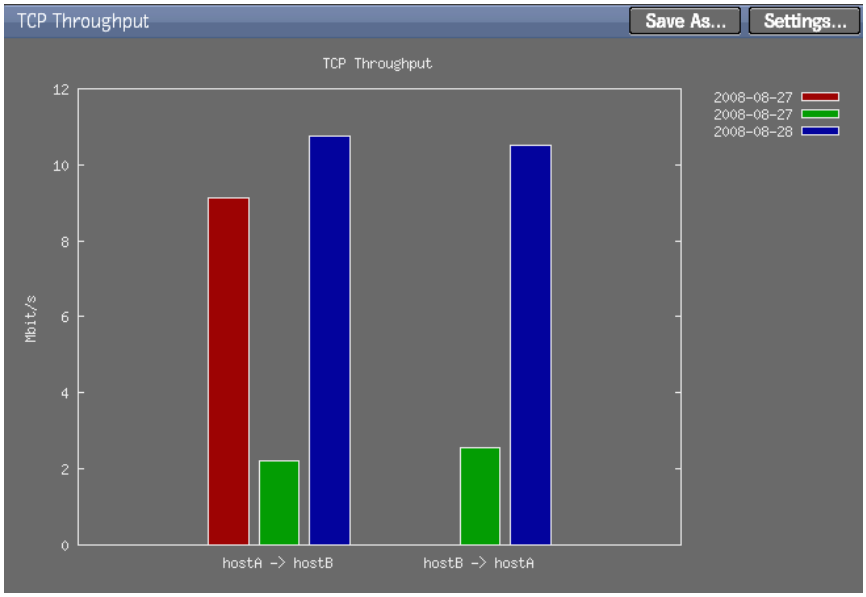
5 In the table on the left, select the plug-in executions that you want to include in the graph.

NOTE Entries in the table are ordered chronologically.

When you click an entry in the table, an overview of the parameters that were set for that plug-in execution is displayed.

6 Click OK to generate the graph.

The graph for the selected results is displayed.



Each group of graph bars represents a value in the plug-in results, and each colour in a group represents a plug-in execution.

For example, the illustration above depicts a graph for three executions of the TCP Throughput plug-in. The two groups of graph bars represent the two values in the plug-in results (transmission rate from hostA to hostB, and transmission rate from hostB to hostA, in Mbit/s). Note how the group of bars for the “hostB to hostA” value only illustrates two executions of the plug-in, as that value was not part of the results in the third execution.

7 Optional: To save the graph as an image file, click “Save as...”.

The graph is saved as a PNG image on the computer you are accessing SystemCentral from.

HTML Report

You can view the results of all executed plug-ins from the current playlist as a HTML report. The HTML report is easy to save, print, or send by e-mail.

To view the HTML report:

1 Select Report | View HTML Report from the application menu.

A new browser window or tab displaying the report appears.

NOTE If no new browser window or tab appears, your browser’s pop-up blocker might prevent it from opening. Set your browser to always allow pop-ups from the workstation running SystemCentral.

2 Use your browser menu to save, print, or e-mail the report.

NOTE The report page is generated and hosted on the Visual Effects, Finishing, and Colour Grading workstation, not on the client system. If you do not save the report from your browser, it will be overwritten when the current playlist is discarded or replaced.

Sending Plug-in Results to Autodesk Customer Support

When reporting a problem to Autodesk Customer Support, you might be asked to run certain SystemCentral plug-ins and send in their results.

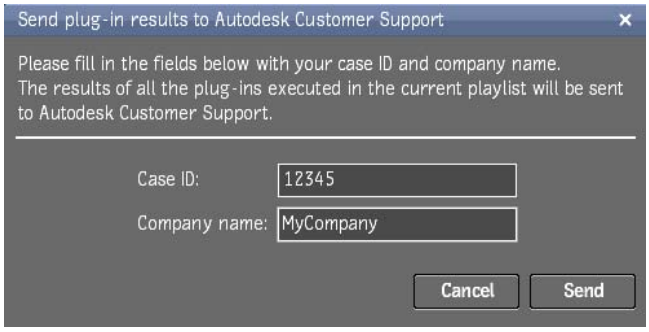
SystemCentral enables you to easily send the results of all executed plug-ins from the current playlist to Customer Support.

To send plug-in results to Customer Support:

1 Create a playlist of the required plug-ins, and execute it. See Adding Plug-ins to the Playlist on page 10, and Running Plug-ins from the Graphical User Interface on page 12.

2 Click Send results to Support at the bottom of the playlist.

The Send plug-in results window appears.



3 Fill in the case ID assigned to you by Customer Support, as well as your contact information, and then click Send.

The results of all executed plug-ins from the current playlist are sent to Autodesk Customer Support.

Managing Plug-ins

5

Topics in this chapter:

Overview

SystemCentral allows you to easily extend its functionality by creating new plug-ins.

You can set your plug-ins to run existing Linux commands or executables, as well as custom scripts you have created. Scripts written in Python can use the SystemCentral API to customize the way results are displayed in the Plug-in Viewer, and to add entries to the SystemCentral log. See SystemCentral API Reference on page 26.

Each SystemCentral plug-in relies on an XML configuration file located in a subdirectory of the */usr/discreet/SystemCentral/plugins/* directory on the workstation.

The category under which a plug-in is displayed in the Available Plug-ins list is the name of the subdirectory where the plug-in XML file is located on disk. For example, the plug-in */usr/discreet/SystemCentral/plugins/Graphics/getNvidiaInfo.xml* appears in the UI under the Graphics category.

The XML configuration file defines several important settings for a plug-in, such as the command or script to be executed, any parameters that can be set by the user before running the plug-in, the way results are displayed and saved to the SystemCentral database, and whether the plug-in is API-compliant.

The following example illustrates a typical plug-in XML file.

```
<?xml version="1.0" ?>
 <item asRoot="False" description="This plug-in will display information on the installed
NVIDIA card and driver" exec="getNvidiaInfo" help="" id="plugins/Configuration/getNvidiaInfo"
 inputType="None" label="NVIDIA Information" needRoot="False" outputType="apicompliant"
paramSwf="None" provider="adsk" resultSwf="None" resultformatgui="raw" saveResults="False"
tooltip="Get information on the NVIDIAcard and driver" uuid="None" version="-1" warning="">
  <params/>
</item>
```

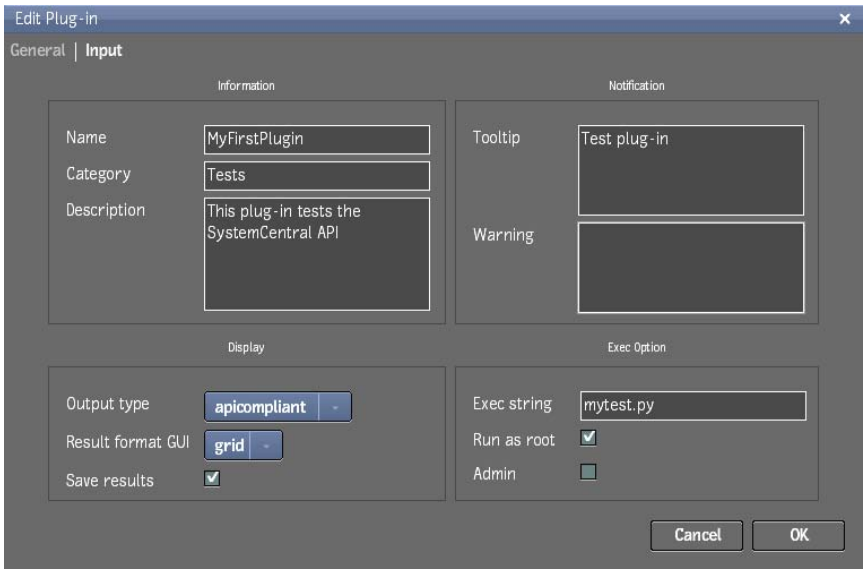
If you configured the plug-in to execute an API-compliant Python script, the script and all other associated files are stored in the same directory as the plug-in XML configuration file.

Creating Plug-ins

The easiest way to create a new plug-in is to use the SystemCentral plug-in editor. The plug-in editor allows you to easily define plug-in settings, and to create input fields for parameters the user can customize before running the plug-in. Based on your settings, the plug-in editor automatically generates the plug-in XML configuration file.

To add a plug-in to SystemCentral:

1. In the Available Plug-ins list, select the category under which you want the plug-in to be saved.
2. Click Add at the bottom of the list.
3. The Edit Plug-in window appears.



4. Type a name for the plug-in.

   NOTE The plug-in name can contain only alphanumeric characters.

   This is the name displayed in the Available Plug-ins list. It also determines the name of the plug-in configuration file. For example, if you set the name to *MyFirstPlugin*, the *MyFirstPlugin.xml* configuration file is created when saving the plug-in.

5. Type the name of the category under which you want the new plug-in to appear in the Available Plug-ins list.

By default, this field contains the name of the category that was selected when clicking Add. You can type the name of another category, or a new name, if you want a new category to be created.

NOTE The category name can contain only alphanumeric characters.

The category name determines the name of the directory under which the plug-in configuration file is saved to disk. For example, if you set the category to *MyCategory*, the plug-in file is saved in the */usr/discreet/SystemCentral/plugins/MyCategory/* directory.

6 Optional: Type a description for the plug-in.

This description is displayed in the Individual View section of the Plug-in Viewer when the plug-in is added to the playlist.

7 Optional: Type a short description for the plug-in in the Tooltip field.

This description is displayed in the Description panel under the Available Plug-ins list when you select a plug-in in the list.

8 Optional: If the plug-in may have a negative impact on the system, enter a warning in the Warning field.

Plug-ins with a warning appear in orange in the Available Plug-ins list. The warning is displayed when attempting to run the plug-in.

9 Type the command or the script to be executed by the plug-in in the Exec string field.

The plug-in executable can be any existing Linux command (for example *df*), or a custom script created by you.

NOTE For information on creating API-compliant Python scripts for your plug-ins, see SystemCentral API Reference on page 26.

If the command requires command-line parameters that are identical at each execution of the plug-in, type them with the command in the Exec string field. For example, *df -P*. If plug-in parameters must change before each execution of the plug-in, define input fields for them in the Input tab. See the next procedure.

10 Optional: If the plug-in executable requires root privileges, enable the Run as root check box.

NOTE If the Run as root check box is enabled, you are asked to provide the SystemCentral administrator password when attempting to save the plug-in.

11 Optional: If you want to prevent other users from editing or deleting the plug-in, enable the Only editable by the Administrator checkbox.

If this option is enabled, users are asked for the SystemCentral administrator password when attempting to Edit or Remove the plug-in.

12 Select one of the following options from the Output type list:

- *apicompliant*, if the plug-in is based on a Python script that uses the SystemCentral API to send the results back to the SystemCentral engine. See SystemCentral API Reference on page 26.
- *noncompliant*, if the plug-in is based on a Linux command, or on a script that does not use the SystemCentral API. Plug-in results will be displayed in a terminal-style panel after the plug-in is executed.

13 Select one of the following options from the Result format GUI list:

- *raw*, to display the plug-in results in a terminal-style panel after the plug-in is executed. This option is mandatory if the plug-in does not use the SystemCentral API.

- *grid*, to display the plug-in results in a table. Use this option only if the plug-in uses the SystemCentral API.

WARNING Do not enable the Save results check box. Saving plug-in results to the SystemCentral database is not supported in this release, except for certain plug-ins created by Autodesk.

14 If you want to define input fields for plug-in parameters, click the Input tab, and follow the next procedure. Otherwise, click OK to save the plug-in.

To define input fields for a plug-in:

1 Click the Input tab of the Edit plug-in window.

2 Click Add to create a new field.

A new entry is created in the fields list to the left, and the new field is displayed in the Preview panel.

3 Type a name for the new field in the fields list.

This name is only used by the SystemCentral engine, and does not appear in the user interface when setting plug-in parameters.

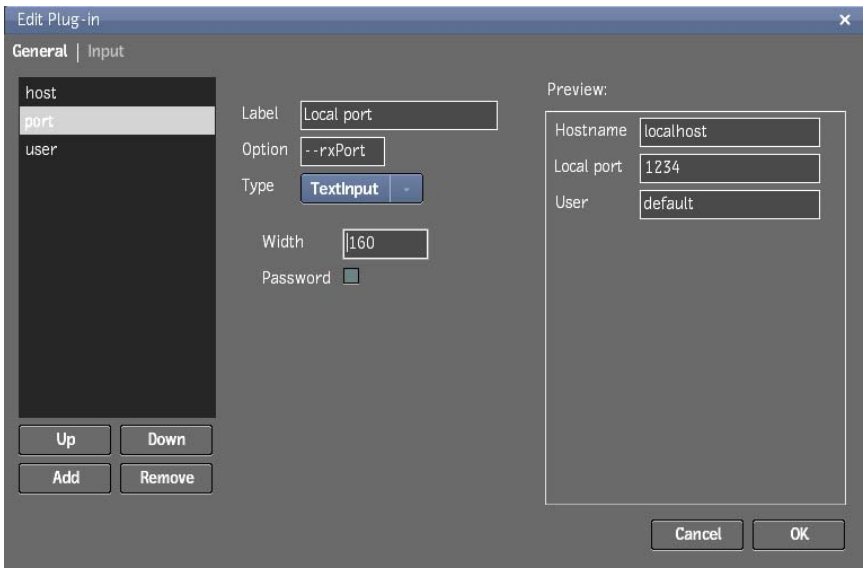
4 Type a label for the field.

The field label appears in the user interface when setting plug-in parameters.

5 In the Option field, type the name of the command-line argument that passes the value of this field to the command or script executed by the plug-in, if necessary.

For example, in the illustration below, the value of the Local port field is passed to the plug-in script through the `--rxPort` argument. If the user enters a value of `1234` in this field before running the plug-in, the plug-in script will be executed with `--rxPort 1234` as an argument.

NOTE The entry in the Option field can also be used in API-compliant plug-in scripts to retrieve the value of a plug-in parameter.



6 Select the type of field from the Type list.

Select	To
TextInput	Create a simple text field.

Select	To
	NOTE You can set the width of the field (in pixels), and whether the field is a password field. Characters in password fields are masked by asterisks.
ComboBox	Create a dropdown list of predefined values. A Values panel appears, allowing you to add predefined values to the dropdown list. For the list to be usable, you must add at least one value.
CheckBox	Create a check box. **NOTE** This type of field does not need any additional configuration.

7 Optional: To set a default value for a field, type the value in the Preview panel entry for the field.

NOTE You can also set the default state of check boxes and the default selected value in dropdown lists.

8 Repeat steps two to seven as many times as necessary to add new fields.

9 Optional: To change the position of a field, use the Up and Down buttons at the bottom of the fields list.

10 Click OK to save the plug-in.

NOTE If the Run as root check box is enabled in the General tab, you are asked to provide the SystemCentral administrator password when attempting to save the plug-in.

Editing and Deleting Plug-ins

To edit a plug-in:

1 Select the plug-in in the Available Plug-ins list and click Edit at the bottom of the list.

NOTE The Edit button is disabled for plug-ins created by Autodesk.

2 Enter the SystemCentral administrator password if asked to.

3 When the Edit Plug-in window appears, follow the instructions in Creating Plug-ins on page 22 to edit the plug-in settings and fields.

To delete a plug-in:

1 Select the plug-in in the Available Plug-ins list and click Remove at the bottom of the list.

NOTE The Remove button is disabled for plug-ins created by Autodesk.

If the plug-in was set to be editable only by the SystemCentral administrator, you are asked to enter the administrator password.

2 Enter the SystemCentral administrator password if necessary.

A delete confirmation dialog box appears.

3 Click Yes to confirm the deletion.

The plug-in is removed from the Available Plug-ins list, and its associated XML configuration file is deleted from disk.

NOTE Only the XML configuration file is deleted from disk when removing a plug-in. Other associated files, such as scripts, are not deleted. This protects your script files from an accidental removal of the plug-in.

SystemCentral API Reference

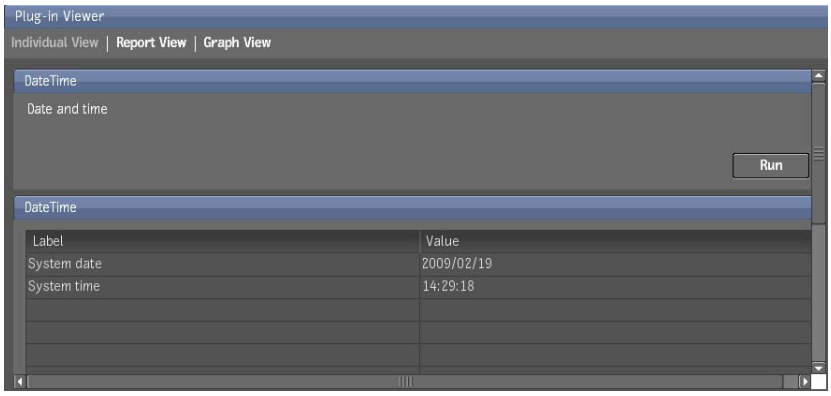
Plug-in scripts written in Python can use the SystemCentral public API to retrieve plug-in configuration information, send plug-in results back to the SystemCentral engine, and add records to the SystemCentral log.

NOTE Saving results to the SystemCentral database is not supported in this release, except for certain plug-ins created by Autodesk.

The following example illustrates the basic structure of an API-compliant plug-in script. It retrieves the system date and time, sends the information back to the SystemCentral engine to be displayed in the Plug-in Viewer, and adds an entry to the SystemCentral log.

```
from framework.api.APIplugin import CAPIplugin
import time
def run(api, testPattern):
     pluginName = testPattern.getName()
     myDate = time.strftime("%Y/%m/%d")
     myTime = time.strftime("%H:%M:%S")
     api.SendResult(pluginName, myDate, "System date")
     api.SendResult(pluginName, myTime, "System time")
     api.SendLogMsg("Plug-in executed successfully.", "INFO")
```

The results of a plug-in based on the previous example script would be similar to the following.



Mandatory Statements

The following statements are mandatory in an API-compliant plug-in script.

from framework.api.APIplugin import CAPIplugin

Each API-compliant plug-in script must import the *CAPIplugin* class. The class exposes all the functions available in the SystemCentral public API.

def run(api, testPattern)

It is mandatory to define a *run* function in the plug-in script.

The *run* function is the entry point to every API-compliant plug-in. When the plug-in is executed, the SystemCentral engine calls this function with the *api* and *testPattern* objects as arguments.

- The *api* object exposes methods for sending plug-in results back to the SystemCentral engine and for adding entries to the SystemCentral log. See api Object Methods on page 28.
- The *testPattern* object exposes methods for retrieving configuration information about the plug-in. This includes the plug-in name, plug-in description, and any plug-in parameters set in the SystemCentral user interface. See testPattern Object Methods on page 27.

NOTE You can write the entire code of your plug-in script inside the *run* function. However, it is recommended to write the plug-in specific code inside a separate function and call that function from within the *run* function. This way, the *run* function remains generic enough that you can easily copy it into other plug-in script files.

testPattern Object Methods

Use the methods of the *testPattern* object to retrieve configuration information about the plug-in.

testPattern.getName() is mandatory. All other *testPattern* methods are optional.

string testPattern.getName()

This mandatory method returns a string containing the name of the current plug-in. The plug-in name is necessary when using the *api.SendResult()* method to send plug-in results back to the SystemCentral engine.

NOTE This method takes no arguments.

Usage example:

```
name = testPattern.getName()
api.SendLogMsg("Plug-in "+name+" ran successfully.","INFO")
```

string testPattern.getArguments()

This optional method returns a string containing the plug-in parameters that were set in the User Interface before running the plug-in. The parameters are separated by space. For each parameter, the string contains the associated argument (as defined in the Option field of the plug-in editor), and the value.

For example, if the plug-in has two fields with arguments "-- firstArg" and "-- secondArg", and the user enters a value of "1" in the first field and "2" in the second field, the *testPattern.getArguments()* method returns the string "-- firstArg 1 -- secondArg 2".

The actual names of the fields are not returned by this method.

NOTE This method takes no arguments.

Usage example:

```
pluginArgs = testPattern.getArguments()
api.SendLogMsg("Plug-in parameters are: "+pluginArgs,"INFO")
```

dict testPattern.getArgumentsDict()

This optional method returns a Python dictionary containing the names and values of the plug-in parameters that were set in the user interface before running the plug-in. Each parameter is represented by a key-value pair in the array.

NOTE This method takes no arguments.

Usage example:

```
pluginArgs = testPattern.getArgumentsDict()
if "field1":
     api.SendLogMsg("The value of field1 is "+pluginArgs["field1"], "INFO")
```

string testPattern.getDescription()

This optional method returns a string containing the description of the plug-in.

NOTE This method takes no arguments.

Usage example:

```
pluginDescr = testPattern.getDescription()
api.SendLogMsg(pluginDescr, "INFO")
```

string testPattern.getCfgFileName()

This optional method returns a string containing the file path of the plug-in XML configuration file, for example */usr/discreet/SystemCentral/plugins/test/myPlugin.xml*.

NOTE This method takes no arguments.

Usage example:

```
pluginPath = testPattern.getCfgFileName()
api.SendLogMsg("The plug-in XML file is: "+pluginPath, "INFO")
```

api Object Methods

Use the methods exposed by the *api* object to send plug-in results back to the SystemCentral engine, and to add entries to the SystemCentral log.

void api.SendResult(string pluginName, string result, string label, colour=(string foreground, string background))

This method sends a plug-in result to the SystemCentral engine, as a name-value pair.

The engine displays the result in the Plug-in Viewer if the plug-in is executed from the SystemCentral user interface. If the plug-in is executed from a terminal, the results are displayed in the terminal, or saved to a file, depending on user preferences.

This method takes the following arguments:

- **pluginName** Mandatory string that contains the plug-in name, as returned by the *testPattern.getName()* method.
  The plug-in name is used by the SystemCentral engine to associate the result name-value pair to the plug-in.
- **result** Mandatory string that contains the value of the plug-in result.
  This argument is displayed in the "value" column of the results table in the Plug-in Viewer.
- **label** Mandatory string that contains a descriptor for the result, for example "System time:" if the result is the current system time.
  This argument is displayed in the "label" column of the results table in the Plug-in Viewer.
- **colour** Optional parameter that contains an array of two strings defining the foreground and background colour for the result text. Each colour is defined by a RGB hex triplet, for example *0xFF0000* for red.

If your plug-in returns multiple results, call the *api.SendResult()* method for each individual result you want to display in the plug-in output.

NOTE This method does not return any value.

Usage example:

```
pluginName = testPattern.getName()
firstResult = 1
secondResult = 2
api.SendResult(pluginName, firstResult, "First result:", colour =("0xFF0000","0xFFFFFF"))
api.SendResult(pluginName, secondResult, "Second result:", colour =("0x00FF00",""))
```

The first result in this example is displayed in red text on a white background, and the second result is displayed in green text on the default background.

void api.SendLogMsg(string message, string level)

This optional method allows you to add entries to the SystemCentral log.

This method takes the following arguments:

- **message** Mandatory string that contains the message to be displayed in the SystemCentral log.
- **level** Mandatory string that sets the log level at which the message is displayed.
  Depending on the importance of the message, set one of the following log levels: "CRITICAL", "ERROR", "WARNING", "INFO", "DEBUG". Each level in the list encompasses the previous one, "CRITICAL" being the most restricted, and "DEBUG" the most verbose.

NOTE This method does not return any value.

Usage example:

```
pluginName = testPattern.getName()
api.SendLogMsg("Fatal error in "+pluginName, "CRITICAL")
```

Appendix: Log and Database Tools

6

Topics in this chapter:

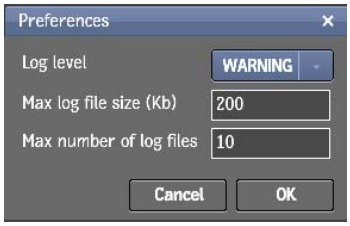
Configuring the SystemCentral Log

The SystemCentral log provides important information for troubleshooting problems you might have with the application or with a plug-in.

To set the log parameters:

1 Select Tools | Preferences from the application menu.

The Preferences dialog box appears.



2 Select one of the five log levels:

- **DEBUG** The most verbose log level. All errors, warnings, and information messages are logged, as well as detailed debugging information.

- **INFO** This level enables the logging of all errors, warnings, and information messages.
- **WARNING** All errors and warnings are logged.
- **ERROR** This level enables the logging of all errors.
- **CRITICAL** When this log level is used, only critical errors are logged.

NOTE It is recommended to use the WARNING log level on a daily basis, and switch to other levels only when required.

3 Set the maximum size of log files, and the number of files to keep.

NOTE For best performance, it is recommended to keep the maximum file size below 1024 KB, and the maximum number of log files below 20.

4 Click OK to save the new log settings.

Viewing the SystemCentral Log

Consult the SystemCentral log to view application activity, or trace back a problem with the application, the playlists, or the plug-ins.

To view the SystemCentral log:

1 Select Report | SystemCentral Log from the application menu.
The SystemCentral Log window appears.

2 Optional: Select and copy the text in this window to the clipboard if necessary.

NOTE The log window displays information only from the most recent log file (*/var/log/autodesk/systemcentral/SystemCentral.log*). To view the contents of older log files, go to */usr/discreet/SystemCentral/log/* and open the log files in a text editor.

SystemCentral Database Clean-up

The SystemCentral database stores results from all executions of the plug-ins that support saving results.

Delete old results from the database from time to time, when you no longer need that data.

The following procedures provide instructions on deleting database entries for a single plug-in or for all the plug-ins.

To delete the results for a plug-in:

1. Add the plug-in to the playlist. See Adding Plug-ins to the Playlist on page 10.
2. Click the Graphic View tab of the Plug-in Viewer.
   Graphic View displays a graph box for each supported plug-in in the playlist.
3. Click Settings for the plug-in you want to clean up.
   The Configure Plug-in Report window appears.
4. Click Clean-up database.
   The Database clean-up dialog box appears.
5. Click the calendar icon to pick the date up to which to purge records, and click OK.
   All the results for the selected plug-in, up to the specified date, are deleted from the database.

To purge the entire results database:

1. Click Tools, and then Database clean-up in the SystemCentral main menu.
   The Database clean-up dialog box appears.
2. Click the calendar icon to pick the date up to which to purge records, and click OK.
   All the results up to the specified date for all plug-ins are deleted from the database.

Index

U

V

W