Reporting 7.5

CCPulse+

Administrator's Guide

About Genesys

Genesys is the world's leading provider of customer service and contact center software - with more than 4,000 customers in 80 countries. Drawing on its more than 20 years of customer service innovation and experience, Genesys is uniquely positioned to help companies bring their people, insights and customer channels together to effectively drive today's customer conversation. Genesys software directs more than 100 million interactions every day, maximizing the value of customer engagement and differentiating the experience by driving personalization and multi-channel customer service - and extending customer service across the enterprise to optimize processes and the performance of customer-facing employees. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys and the Genesys logo are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders. © 2012 Genesys Telecommunications Laboratories, Inc. All rights reserved.

The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 9. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 75rt_ad_ccpulse+_08-2012_v7.5.102.00

Table of Contents

Preface

Welcome to the *Reporting 7.5 CCPulse+ Administrator's Guide*. This document presents information that CCPulse+ administrators need in order to fine-tune and troubleshoot CCPulse+ installation. This guide is valid only for the 7.5.*x* release of this product.

Note: For releases of this guide created for other releases of this product, please visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

This preface has these sections:

In brief, you will find the following information in this guide:

- Special considerations concerning CCPulse+ historical reporting
- Instructions for customizing the user interface
- Instructions for creating and formatting custom statistics
- Explanations of the Thresholds and Actions scripts
- A list of the historical statistics associated with the CCPulse+ real-time statistics
- Troubleshooting suggestions

Intended Audience

This guide is primarily intended for CCPulse+ administrators. It assumes that you have a basic understanding of:

- Computer-telephony integration concepts, processes, terminology, and applications.

- Network operation.
- Your own network configurations.

You should also be familiar with Genesys Framework architecture and functions, particularly Stat Server statistics and configuration.

CCPulse+ administrators have access to CCPulse+ functions not available to other users. CCPulse+ administrators can:

- Operate the Threshold and Action wizards.
- Operate the Template Wizard to create, modify, or delete CCPulse+ templates.
- Use the Import/Export Utility to copy CCPulse+ templates, thresholds, and actions from one storage to another.

A strong knowledge of VBScript may be required for some functionality, such as creating custom formulas.

Chapter Summaries

In addition to this opening chapter, this guide contains these chapters:

- Chapter 1, “Installing CCPulse+ Silently,” on page 11, describes how to use InstallShield Silent for installing CCPulse+ *silently;* that is, for performing installations that do not require the installer to enter any settings.
- Chapter 2, “Fine-Tuning CCPulse+ Configuration,” on page 17, describes how to customize storage file locations and user interface settings.
- Chapter 3, “Historical Reporting Considerations for CCPulse+,” on page 37, introduces the functionality for composite metrics, and presents some issues that you should know about before using historical reporting functionality in CCPulse+.
- Chapter 4, “Server Connections to CCPulse+,” on page 41, describes how CCPulse+ behaves when its server connections fail and how and when you can resume normal operation.
- Chapter 5, “Using Custom Statistics,” on page 47, describes the syntax of the formula-driven statistics provided with your solution and how to affect the display of results that CCPulse+ returns.
- Chapter 6, “Thresholds and Actions,” on page 59, describes the Genesys-provided action and threshold scripts and the objects available for use in these scripts.
- Chapter 7, “Measuring CCPulse+/Stat Server Traffic,” on page 77, discusses some of the elements contributing to CCPulse+/Stat Server traffic and how you can measure them.

- Chapter 8, “Troubleshooting,” on page 79, lists some problems you may encounter when using CCPulse+ and offers suggestions for correcting them.

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 85.

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to Techpubs.webadmin@genesyslab.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Contacting Genesys Technical Support

If you have purchased support directly from Genesys, contact Genesys Technical Support at the regional numbers that follow:

Note: The following contact information was correct at time of publication. For the most up-to-date contact information, see the Contact Information on the Tech Support website. Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

Genesys Technical Support Contact Information

Region	Telephone	E-mail
North America	+888-369-5555 (toll-free)	support@genesyslab.com
Latin America	+55 11 2948 5776	la-support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-127 645-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au

Genesys Technical Support Contact Information (Continued)

Region	Telephone	E-mail
Japan	+81-3-6361-8950	support@genesyslab.co.jp
India	000-800-100-7136 (toll-free) +61-7-3368-6868	support@genesyslab.com.au

Document Change History

This section lists topics that are new in the current release of this document, or that have changed significantly from the preceding release.

New in Release 7.5.1

This release of CCPulse+ introduces the following new features:

- Support for a new `Capacities` property inside the `state` object. See "Using Predefined Objects in Formulas" on page 49.
- New formatting options that allow CCPulse+ users to choose how to display capacity information in the CCPulse+ workspace. See "Formatting the Results" on page 55.
- Three new CCPulse+ configuration options related to enabling and displaying agent and agent group capacity:
  - `EnableAgentCapacity`, described on page 25
  - `EnableAgentCapacityStatus`, described on page 25
  - `EnableAgentgroupCapacity`, described on page 25
- A new configuration option that allows users to specify the available media types when formatting capacity information using the CCPulse+ template wizard:
  - `MediaTypes`, described on page 28
- Availability of capacity information in threshold and formula scripts, described on page 49.
- Simulation within CCPulse+ of capacity information for agent groups, described on page 25.

Chapter

1 Installing CCPulse+ Silently

This chapter describes how to use InstallShield Silent for installing CCPulse+ *silently;* that is, for performing installations that do not require the installer to enter any settings. Refer to the *Reporting Deployment Guide* for instructions using wizards to configure and install CCPulse+ using wizards.

This chapter includes the following sections:

Overview

InstallShield Silent is a third-party installation program that Genesys uses to facilitate electronic software distribution for both server and GUI applications on Windows platforms. “Silent” installations eliminate the need for interactive dialog during the installation process. Instead, you create a single response file filled with the necessary parameters that InstallShield Silent references during subsequent silent installations.

After creating your response file and performing a silent installation, review the log file for a successful result code or any errors encountered.

You must take some preliminary actions before installing CCPulse+. Refer to the “Predeployment Measures” chapter in the *Reporting Deployment Guide* for more information.

Creating the Response File

To select setup options and automatically record the InstallShield Silent response file, perform Steps 1–3 below.

Note: Use this procedure instead of double-clicking `setup.exe` from Windows Explorer.

1. Open a console window.
2. Enter the path to the directory in which you deployed the CCPulse+ installation files.
3. Issue the following command using the `-r` command-line parameter:

   ```
   setup.exe -r
   ```

   Answer the prompts as appropriate. InstallShield records all your setup choices in a response file named `setup.iss` and places the file in the Windows directory. For example, if you are using Windows XP, this is the `C:\WINDOWS` directory.

Use the configured `setup.iss` file any time you must install an application with the specified parameters.

The script in Figure 1 shows a sample `setup.iss` file for CCPulse+. Blank lines have been inserted between sections to facilitate reading.

Playing Back the Response File

To install CCPulse+ silently on a local workstation:

1. Open a console window.
2. Enter the path to the directory in which you deployed the CCPulse+ installation files.
3. Launch InstallShield Silent using the following command:

   ```
   setup.exe -s /f1"ResponsePath" /f2"LogPath"
   ```

   where:

   - `-s` is the silent flag.
   - `/f1"ResponsePath"` is an optional command-line parameter that specifies the full path to the response file. If you do not specify this parameter, InstallShield assumes that the name of the response file is `setup.iss`, and that it is located in the directory containing `setup.exe`.
   - `/f2"LogPath"` is an optional command-line parameter that specifies the full path to the log file. If you do not specify this parameter, InstallShield assigns the default file name of `setup.log` and stores the file in the directory containing `Setup.exe`.

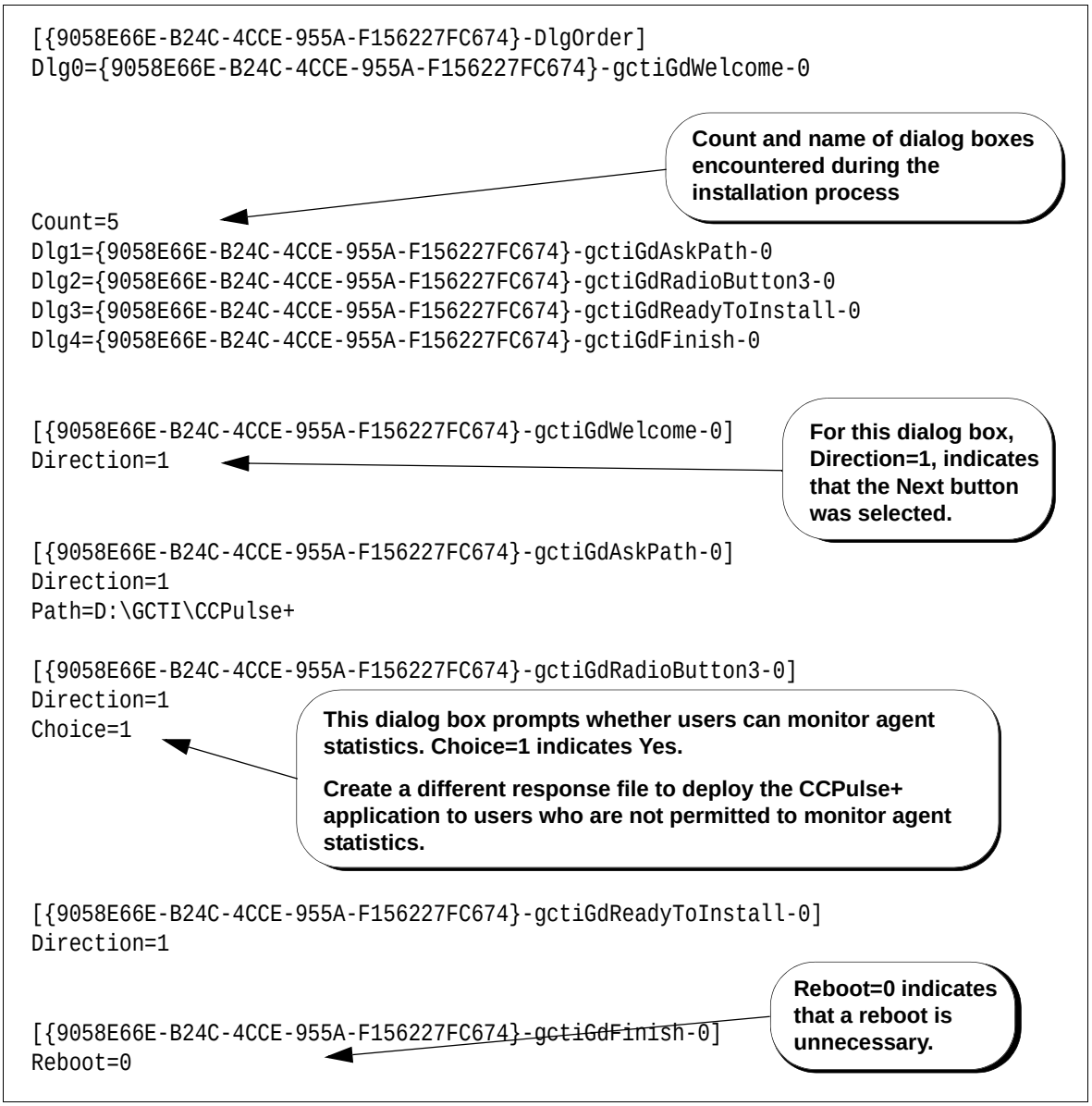

Figure 1: Sample CCPulse+ `setup.iss` File

> Note: Do not use spaces between the `/f1` or `/f2` parameter and its value in double quotation marks.

The following illustrates some commands you could issue to play back the response file:

- `setup.exe -s`

  Both `/f1"*\[`*`Setup.exe directory`*`]\setup.iss"` and `/f2"*\[`*`setup.exe directory`*`]\setup.log"` are implied on a Windows XP system.

- `setup.exe -s /f1"c:\WINDOWS\MonitorAgents.iss" /f2"*\[`*`Setup.exe directory`*`]\setup.log"` is implied on a Windows XP system.
- `setup.exe -s /f2"c:\WINDOWS\SuzysLog.log" /f1"*\[`*`Setup.exe directory`*`]\setup.iss"` is implied on a Windows XP system.

Remote Silent Installation

Using standard operating system tools, your network administrator can design the scripts necessary to deploy the response file to any number of machines where CCPulse+ can be deployed remotely.

Analyzing the Log File

When the response file has completed its playback, InstallShield Silent prints installation results to the file specified by `/f2`, as shown in the previous section. This log file contains one section:

- The `[ResponseResult]` section contains the result code indicating whether the silent install succeeded. See Table 1 for the meanings of the `ResultCode` key given in this section.

Table 1: Log File ResultCode Key

Value	Meaning
0	Success
-1	General error
-2	Invalid mode
-3	Required data not found in the `setup.iss` file
-4	Not enough memory
-5	File does not exist
-6	Cannot write to the response file
-7	Unable to write to the uninstallation log file
-8	Invalid path to the InstallShield Silent response file
-9	Not a valid list type (string or number)
-10	Data type is invalid

Table 1: Log File ResultCode Key (Continued)

Value	Meaning
-11	Unknown error during setup
-12	Dialog boxes are out of order
-51	Cannot create the specified folder
-52	Cannot access the specified file or folder
-53	Invalid option selected

The text in Figure 2 shows the contents of a log file for a CCPulse+ application that was successfully installed using InstallShield Silent.

```
[ResponseResult]
ResultCode=0
```

Figure 2: Log File Showing Successful Deployment

Chapter

2 Fine-Tuning CCPulse+ Configuration

After the initial configuration of the CCPulse+ `Application` object and installation, you can use Configuration Manager to perform additional customization in the various sections of the application. This chapter describes how to fine-tune the CCPulse+ configuration in the following sections:

The CCPulse+ application itself does not consider the case of configuration information it reads from Configuration Server. However, the RDBMS of your Configuration Server database may be case sensitive. As you fine-tune your CCPulse+ `Application` object, keep this in mind. Also, for those configuration options for which you specify `true/false` values, any of the following additional values are also valid:

- `yes/no`
- `y/n`
- `1/0`
- `on/off`

Storage Section

A *storage* is a file in which specific information types reside. Storage locations are configured on the `Storage` section of your CCPulse+ application `Options` tab using Configuration Manager, as shown in Figure 3.

To enter a storage location:

1. Double-click the desired item in the `Name` column to open the `Edit Option` window.
2. In the `Value` text box, enter the full path to the location where the storage file should reside.
3. Click `OK`. The `Storage` list now displays the updated storage location (see Figure 3).

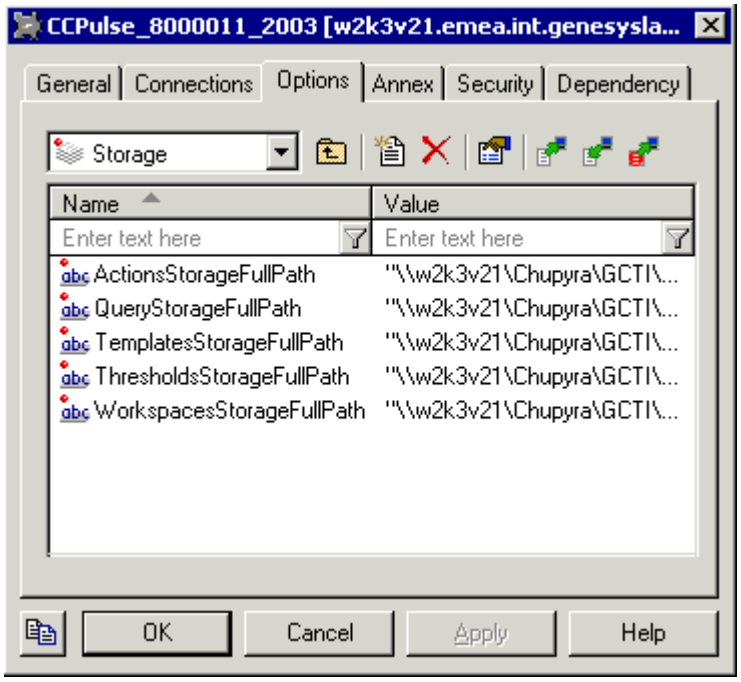

Figure 3: Storage Location Configuration

Table 2 describes the six configuration options you can define in the `Storage` CCPulse+ section.

Table 2: Configuration Options for the Storage Section

Option	Description
ActionsStorageFullPath	Defines where CCPulse+ actions are stored when an administrator clicks `Finish` in the Actions Wizard.

Table 2: Configuration Options for the Storage Section (Continued)

Option	Description
QueryStorageFullPath	Defines the location of the XML file where CCPulse+ queries are stored.
StatProfileStorageFullPath	Defines the location of `StatProfile.cfg`, which is set up during CCPulse+ application configuration when you use the Real-Time Reporting Wizard. You can also manually add the location to CCPulse+ application properties in Configuration Manager.
TemplatesStorageFullPath	Defines where CCPulse+ templates are stored when an administrator clicks `Finish` in the Template Wizard.
ThresholdsStorageFullPath	Defines where CCPulse+ thresholds are stored when an administrator clicks `Finish` in the Thresholds Wizard.
WorkspacesStorageFullPath	Defines where CCPulse+ workspaces are stored when users save workspaces. If a workspace is used by multiple users, make sure all users can access the location of the `Workspaces` storage file.

UserInterface Section

The `UserInterface` section holds configuration options with which you can specify how CCPulse+ is to display data in its interface. (See Table 3.)

Table 3: Configuration Options for the UserInterface Section

Option	Description
DisableThresholdsDlg	Use this option to enable contact center operators configured as CCPulse+ Users, to set thresholds, or to prevent them from doing this. This option controls visibility of the Set Threshold menu item on the shortcut menu that is invoked when the user right-clicks a statistic or statistics group. Setting this configuration option has no meaning for users configured as CCPulse+ Administrators. Full threshold and action functionality is always available to CCPulse+ Administrators. Default Value: `true` Valid Values: `true, false` Change Takes Effect: Upon restart

Table 3: Configuration Options for the UserInterface Section (Continued)

Option	Description
EnableDNAliases	Use this option to configure CCPulse+ to display either DN numbers or DN aliases in the CCPulse+ workspace. To display DN aliases, specify `true` for this option's value. To display DN numbers instead, specify `false`. CCPulse+ synchronizes DN aliases with any changes you might make to a DN object's properties. Default Value: `true` Valid Values: `true, false` Change Takes Effect: Upon restart
ShowAgentLoginID	Use this option to configure CCPulse+ to display the agent's login ID in the `Extended Current Status` window. Specify `true` to display login IDs, or false to hide them. This window might display `n/a` (for not applicable) in place of a login ID when the ID is unavailable, or when the agent is logged out. This option does not apply to Agent Group objects or other object types. Default Value: `true` Valid Values: `true, false` Change Takes Effect: Upon restart
ShowAgentSkills	Use this option to configure CCPulse+ to display the skills associated with selected agents in the `Extended Current Status` window. Specify `true` to display agent skills or `false` to hide them. If no skills are defined in Configuration Manager or if skills have not been assigned to the selected agent(s) and `ShowAgentSkills` is set to `true`, CCPulse+ displays `n/a` (for not applicable) in the `Extended Current Status` window. The 7.0.1 release of CCPulse+ introduced a `Skill` icon (), which precedes the listing of skills in the `Extended Current Status` window. Default Value: `false` Valid Values: `true, false` Change Takes Effect: Upon restart
ShowStatusReasons	Use this option to configure CCPulse+ to display data that is attached to a TEvent in the form of reason codes. This data provides a reason for an object's current state and is associated with the agents you select in the `Extended Current Status` window. CCPulse+ obtains this information from Stat Server, which gets this information from one of two attributes supplied by each T-Server `TEvent`: • The `Extensions` attribute–its `ReasonCode` key (for hardware-related reasons) • The `Reasons` attribute (for data supplied by software such as the Genesys SoftPhone or ActiveX)

Table 3: Configuration Options for the UserInterface Section (Continued)

Option	Description
ShowStatusReasons (continued)	Specify `true` to display the value of this reason code or `false` to hide it. Default Value: `false` Valid Values: `true, false` Change Takes Effect: Upon restart Reason codes apply only to the following noncall-related statuses: • `LoggedIn` • `LoggedOut` • `AfterCallWork` • `Ready` • `NotReady` As reason codes are associated with a particular DN, CCPulse+ cannot display reason codes if the agent is not logged in to a DN. In addition, CCPulse+ cannot display reason codes if they are attached to a `TEvent` in other than the `Reasons` or `Extension` attributes. If T-Server attaches no data to the TEvent, CCPulse+ displays `n/a` (for not applicable) in the `Extended Current Status` window. The 7.0.1 release of CCPulse+ introduced a `Reasons` icon (), which precedes the listing of reasons in the `Extended Current Status` window. Refer to the *Genesys 7 Events and Models Reference Manual* for information about `TEvent` structure. **Note:** In some cases, CCPulse+ might show user data and business attributes; (for example, `MediaType`), in addition to actual reasons. Use a string for the reasons values; (for example, `Lunch` or `Meeting`), to separate actual reasons from possible non- reason data. `Extended Current State` window and `Current state` in the View can show reasons if enabled by the appropriate option.

Table 3: Configuration Options for the UserInterface Section (Continued)

Option	Description
ShowDurationBefore StatusName	Use this option to effect the format by which CCPulse+ displays information about current status in the `Views` pane. If `true`, CCPulse+ displays the duration of a particular state first followed by the current state using the format: `(hh:mm:ss) currstatus`. If `false`, CCPulse+ displays the name of the current status first and then duration: `currstatus (hh:mm:ss)`. By default, CCPulse+ displays status duration after the current status, even if the option is absent from the configuration of your CCPulse+ application object. Such placement facilitates sorting by current status should you desire to use this functionality. (Incidentally, CCPulse+ does not allow you to sort by duration.) You may, however, prefer duration precede to current status. Figure 4 provides an illustration of this preference. Default Value: `false` Valid Values: `true, false` Change Takes Effect: Upon restart

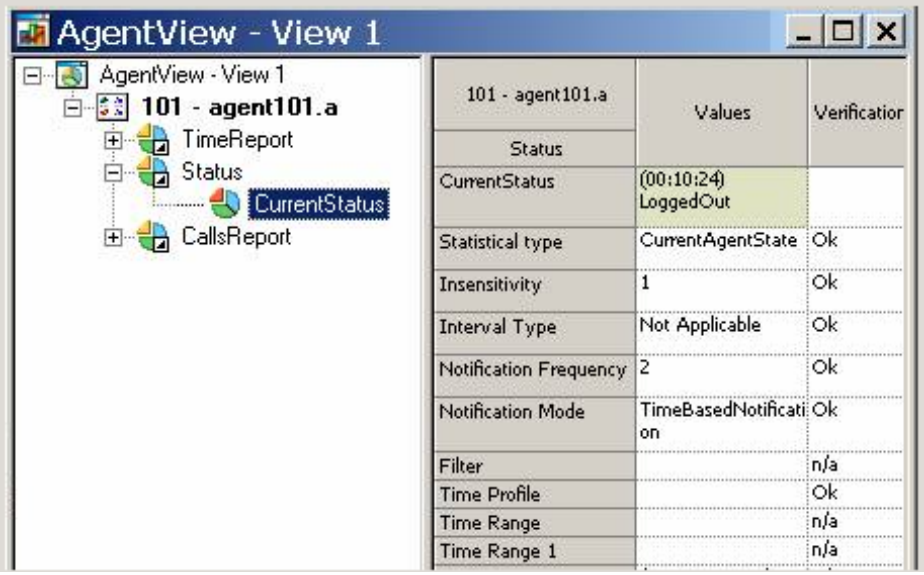

Figure 4: CCPulse+ Displays Duration First When Option Is Set to True

IconStyle Section

The `IconStyle` section holds one configuration option which controls the appearance of the logged-out icon in the CCPulse+ interface. (See Table 4.)

Table 4: Configuration Options for the IconStyle Section

Option	Description
DNLoggedOut	Use this option to configure CCPulse+ to display one of three styles of logged out icons by using the `DNLoggedOut` option. The value you specify for this option relates to the CCPulse+ release in which these styles appeared. The three styles of the `DNLoggedOut` icon and the corresponding CCPulse+ release are shown below: •  , `DNLoggedOut` icon for CCPulse+ release prior to `6.1.3`. •  `DNLoggedOut` icon for CCPulse+ release `6.1.3` to `6.1.301.04`. The CCPulse+ template provided with the 7.5 installation package supplies a value of `6.1.301.04` for this option. •  `DNLoggedOut` icon for CCPulse+ release `6.1.301.05` to `7.5`. This icon style is also used when the value of the `DNLoggedOut` configuration option is empty or invalid, or when the `IconStyle` section is absent. In the CCPulse+ `Options` tab (shown below in Figure 5), a value of `6.1.301.04` has been entered for the `DNLoggedOut` option. Default Value: `6.1.301.04` Valid Values: *a CCPulse+ release number* Change Takes Effect: Upon restart

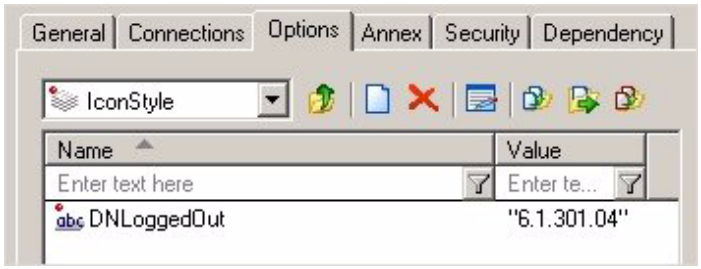

Figure 5: Setting the DNLoggedOut Configuration Option

Workspace Section

The `Workspace` section holds configuration options that affect your CCPulse+ workspace. (See Table 5.) Note that because the options defined in this section are defined within the CCPulse+ `Application` object in Configuration Server, their settings affect all CCPulse+ instances that are configured to use the object. You cannot change these settings for every instance of CCPulse+ individually. Instead, you have to define a new `Application` object for each combination of configuration settings that you wish to set and then configure each instance to use the appropriate `Application` object.

Table 5: Configuration Options for the Workspace Section

Option	Description
CurrentStateEnable	Use this option to enable or disable the current state for all configuration objects selected for monitoring on the Call Center Objects pane. Consider using this option if you wish to increase runtime performance of CCPulse+ and decrease startup time—especially if you regularly use workspaces with a large numbers of objects. Default Value: `true` Valid Values: `true, false` Change Takes Effect: Upon restart
DisableAgentCurrentState	Use this option to enable or disable the visibility of current-state information on individual agents in all CCPulse+ panes for contact center operators configured as CCPulse+ Users. The visibility of current-state information for other object types remains unaffected by the setting of this option; instead it is controlled by the `CurrentStateEnable` configuration option. The setting of this configuration option has no meaning for users configured as CCPulse+ Administrators. Administrators can always view current-state information on individual agents, regardless of this option's setting. In situations where your configuration includes one-person agent groups, consider adding a pseudo `Person` object to these groups, to preserve an environment where operators should not be able to obtain current-state information on individual agents. Default Value: `false` Valid Values: `true, false` Change Takes Effect: Upon restart

Table 5: Configuration Options for the Workspace Section (Continued)

Option	Description
EnableAgentCapacity	This option enables or disables the receipt of capacity per media information from Stat Server for individual agents in the workspace. Default Value: `false` Valid Values: `true`, `false` Change Takes Effect: Upon restart
EnableAgentCapacityStatus	This option enables or disables the display of agent or agent group capacity information in the object tree. Default Value: `false` Valid Values: `true`, `false` Change Takes Effect: Upon restart
EnableAgentGroupCapacity	This option enables or disables the calculation of capacity per media information by CCPulse+ for agent groups in the workspace. Agent group capacity is defined as the sum of the corresponding media capacity values for the agents in the group: `Group max capacity/current interactions/routable interactions` = `Sum (Agent max capacity/current interactions/routable interactions)` **Note:** Displaying agent group capacity will have a linear impact on CCPulse+ performance (depending on the number of calls that the group members receive). Default Value: `false` Valid Values: `true`, `false` Change Takes Effect: Upon restart
IgnoreNewObjectNotifications	Use this option to specify whether CCPulse+ should accept or ignore notifications from Configuration Server about newly created reportable objects. If set to `true`, CCPulse+ ignores notifications of new objects—any new reportable objects will not appear in the workspace. If set to `false`, new objects will appear instantly in the `Call Center Objects` pane. When the workspace is saved, these objects become a permanent part of the workspace, like the other configuration objects that were explicitly selected for monitoring when the workspace was initially created. Default Value: `false` Valid Values: `true`, `false` Change Takes Effect: Upon restart

Table 5: Configuration Options for the Workspace Section (Continued)

Option	Description
SafeScriptMode	Use this option to specify whether CCPulse+ will spawn a separate thread to execute VBScripts. If set to `true`, threshold and action VBScripts will be executed using a separate thread. This is a safe but slower method of operation. If set to `false`, all scripts will be executed using the main process, and the `Timeout` option, shown on the `Threshold Creation` and `Action Creation` pages of the Threshold and Action wizards (respectively), will become disabled. Figure 6 shows the bottom half of the `Threshold Creation` page with the timeout area disabled. Default Value: `true` Valid Values: `true, false` Change Takes Effect: Upon restart
UseStatServerVirtual Groups	Use the `UseStatServerVirtualGroups` configuration option to control whether CCPulse+ obtains information about virtual agent group membership from Stat Server or Configuration Server. Stat Server enables you to specify a wider range of criteria in order to qualify virtual agent group membership. These criteria include a consideration of the ACD queue or switch into which an agent is logged in, skill expressions configured for the agent, and real-time data such as status. Configuration Server uses skill expressions alone to determine group composition. Refer to "Virtual Agent Groups" in the *Framework Stat Server User's Guide* for additional information. If set to `true`, CCPulse+ obtains information about group membership from Stat Server. If set to `false`, CCPulse+ obtains information from Configuration Server. Please note that regardless of the setting of this option, the new feature—creating dynamic agent- and virtual agent-group views—uses Stat Server as the source for obtaining group membership. Refer to the "Creating Dynamic Virtual/Agent Group Views" topic in *Reporting 7.5 CCPulse+ Help* for more information about dynamic views. **Note:** This feature is available only in the CCPulse+ real-time environment. CCPulse+ historical reporting does not capture dynamic changes to the content of virtual agents groups. Default Value: `false` Valid Values: `true, false` Change Takes Effect: Upon restart

Table 5: Configuration Options for the Workspace Section (Continued)

Option	Description
WorkspaceAutoLoad	This option determines whether CCPulse+ loads the last workspace used when CCPulse+ starts. If `true`, CCPulse+ loads the workspace you were using the last time you closed CCPulse+, enabling you to bypass the explicit selection steps otherwise required. If `false`, CCPulse+ prompts you to select one of the available workspaces in your environment. CCPulse+ stores the name of the most recently used workspace in the Microsoft Windows registry. Default Value: `true` Valid Values: `true, false` Change Takes Effect: Upon restart

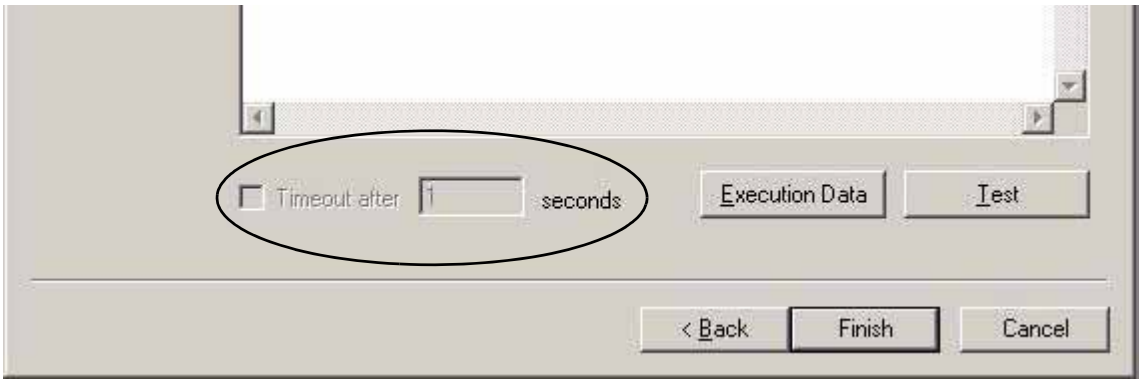

Figure 6: No Timeout Specification with SafeScriptMode=False

CustomStatistic Section

The `CustomStatistic` section holds one configuration option to affect the display of custom statistics within the CCPulse+ interface. (See Table 6.)

Table 6: Configuration Option for the CustomStatistic Section

Option	Description

ExtendedCurrentStatus	Use this option to configure CCPulse+ to display reasons in the current state statistics provided through agent views. If set to `true`, CCPulse+ displays the reasons for the current state statistic as one or more strings separated by semi-colons. If set to `false`, the reasons for the current state statistics are not displayed.
	To fully implement this feature, you must also define the underlying stat type for agent current state statistics as follows:
	`Category=CurrentState`
	`MainMask=*`
	`Objects=Agent`
	`Subject=DNAction`
	This `DNAction`-based stat type enables Stat Server to provide data along with duration and actual status data.
	In Figure 7, the `AgentStateReason` view displays a reason of `1` for the agent `Employee_ID_601` who is in the state `NotReadyForNextCall`. In this example, a threshold and action have also been applied to this statistic. For information on setting thresholds and actions, refer to the discussion starting on page 59. Table 16 on page 60 shows how to use reasons in thresholds.
	Default Value: `false`
	Valid Values: `true, false`
	Change Takes Effect: Upon restart

MediaTypes	This option populates the drop-down list of media types available in the `Media Filter` section of the `Properties` dialog box for the `CurrentAgentState` statistic category. Default Value: `"voice"` Valid Values: <comma separated list of media> (Example: `"voice, email, voip"`) Change Takes Effect: Upon restart

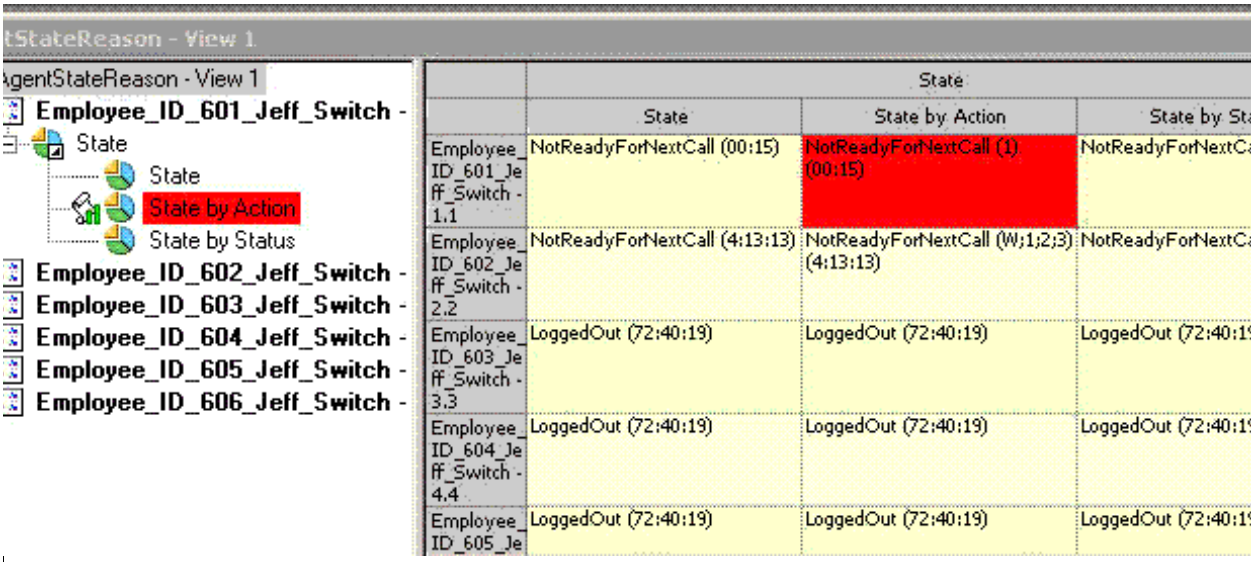

Figure 7: CCPulse+ Displays Reason When Option is Set to True

View Section

The `View` section holds configuration options that affect the presentation of data in CCPulse+ views. (See Table 7.)

Table 7: Configuration Options for the View Section

Option	Description
CurrentStateRefresh Interval	This option specifies the time interval at which `CurrentState` statistics will be refreshed in the view. If the view does not contain any `CurrentState` statistics, CCPulse+ ignores this option. CCPulse+ adjusts any value specified over the maximum value to the maximum value. **Note:** If you enter a `0` (zero), a negative number, or any alphanumeric string, you will disable this setting. The current state duration will be not be refreshed until it receives an update from the Stat Server. Default Value: `5` (seconds) Valid Values: `0` to `86399` Change Takes Effect: Upon restart

Table 7: Configuration Options for the View Section (Continued)

Option	Description
DelayBeforeRemoval	This option specifies the number of seconds that logged out agents remain visible, but in a dimmed (greyed out) state, in a real-time V/AG dynamic membership view, before CCPulse+ removes them from the view entirely. A value of `0` instructs CCPulse+ to immediately remove agents from the view; however, in reality, some small amount time could pass before the view is updated. CCPulse+ resets any value specified outside the range of valid values to `0`. Default Value: `15` (seconds) Valid Values: `0` to `3600` Change Takes Effect: Upon restart
DisableCreateRemove Views	Use this option to enable CCPulse+ Administrators to restrict the creation and removal of views by contact center operators, or to prevent them from doing this. The setting of this configuration option has no meaning for users configured as CCPulse+ Administrators. Administrators can always create and/or review views, regardless of this option's setting. However, setting this value to `true`, prevents agents configured as CCPulse+ Users from performing these actions. Default Value: `false` Valid Values: `true, false` Change Takes Effect: Upon restart
DisableHideShowActions OnGraph	Use this option to enable or disable hide/show actions on the `Graph` view. Setting this configuration option will greatly reduce the time it takes for CCPulse+ to open a workspace with hide/show actions. Default Value: `false` Valid Values: `true, false` Change Takes Effect: Upon restart

Table 7: Configuration Options for the View Section (Continued)

Option	Description
DisableRowColVisibility Change	Use this option to enable contact center operators configured as CCPulse+ Users, to alter the visibility of rows and/or columns in the `Table` view, or to prevent them from doing this. In addition, if you set this configuration option to `true`, the shortcut menu items to hide and show rows and columns are disabled for CCPulse+ Users. The setting of this configuration option has no meaning for users configured as CCPulse+ Administrators. Administrators can always hide and show rows and columns in the `Table` view and the corresponding shortcut menu items remain enabled, regardless of this option's setting. Default Value: `false` Valid Values: `true, false` Change Takes Effect: Upon restart
MaxNumberOfAgents	This option specifies the maximum number of agents that CCPulse+ displays in a real-time V/AG dynamic membership view regardless of the number of available agents or their current status. The default value provided in the CCPulse+ 7.5 application template allows up to 75 agents in the display; this is also the value that CCPulse+ uses if this option is not set. CCPulse+ sets any value outside the range of valid values to `1`. Default Value: `75` (agents) Valid Values: `1` to `450` Change Takes Effect: Upon restart

Changing the Display of Dates and Time

CCPulse+ release 7.0.1 introduced functionality that allows you to define how time and dates appear in hourly and daily historical reports. Specify the desired format on the `Date` and `Time` tabs of your Control Panel's `Regional Settings`, and CCPulse+ automatically changes its column titles display of your hourly and daily reports to the format specified. CCPulse+ also uses these regional settings for date and time format on the Daily and Hourly Historical report titles and in query-based reports.

Figure 8 shows instances of two different hour-format settings. The first image uses a regional hour format of M/d/yy representing July 27, 2001 as `7/27/01`. The second image uses the `dd/MM/yyyy` format and displays the same date as `27/07/2001`.

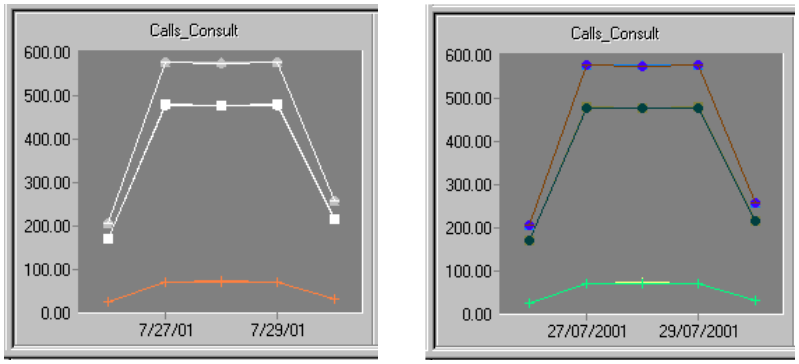

Figure 8: Different Hour-Format System Settings Reflected in Historical Daily Views

Figure 9 shows instances of two different time-format settings. The first image uses a regional time format of `h:mm:ss tt`, representing 4 o'clock in the afternoon as `4:00:00 PM`. The second image uses the `HH:mm:ss` format and displays the same time as `16:00:00`.

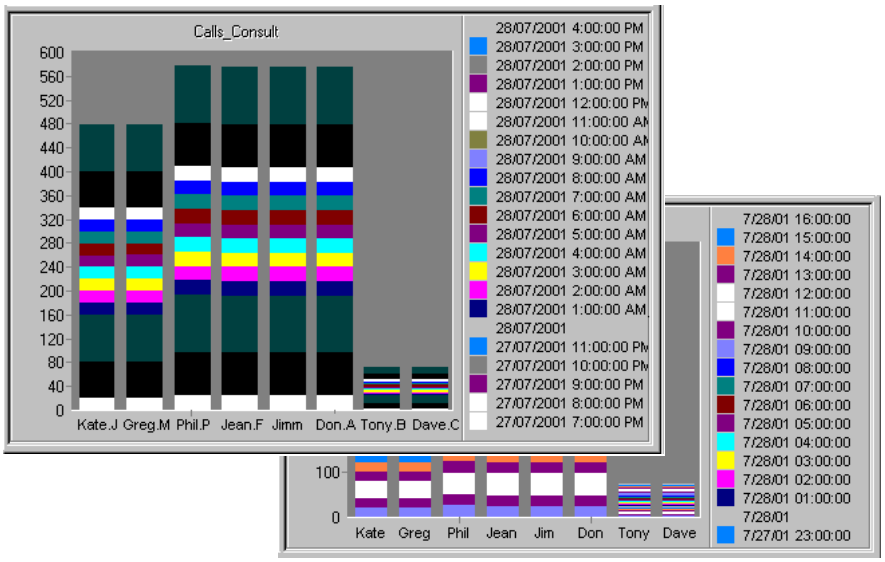

Figure 9: Different Time-Format System Settings Reflected in Historical Hourly Views

Although CCPulse+ accepts dynamic system changes of regional time and dates, if CCPulse+ is currently displaying a view when the change is made, you must deactivate and reactivate the view in order to observe its effect.

CCPulse+ Registry Settings

You can edit these five CCPulse+ configuration settings only by using the Microsoft Windows Registry Editor:

- GridFontSize
- GraphTileFontSize
- GraphBkColor
- GraphAxisFontSize
- FormatAgentInfo

In addition, you can configure the following settings in the registry or in CCPulse+:

- DeltaAvgTime
- DeltaTotalTime
- DeltaCntCalls
- IntervalForCurrentUpdate
- HistoricalUpdateFrequency
- NotifyEvery

Where applicable, Genesys recommends that you use the Settings menu option in CCPulse+ to set these values. However, if you prefer to configure these settings directly using the Registry, refer to the path information provided for each setting below.

Note: Unlike settings configured in the CCPulse+ Application object, which apply to all CCPulse+ instances that use that application, registry key changes affect only the local workstation.

Access the root for all these settings through the following path in the Registry:

HKEY_CURRENT_USER\Software\GCTI\CallCenter\

Configure your user interface by entering the desired value next to the appropriate key. Table 8 describes these settings.

Table 8: CCPulse+ Registry Settings

Registry Setting	Description
GridFontSize	Specifies the font size of the characters in the Table view of the Views pane. Default Value: 7 Valid Values: Positive integers Path: HKEY_CURRENT_USER\Software\GCTI\CallCenter\Attributes

Table 8: CCPulse+ Registry Settings (Continued)

Registry Setting	Description
GraphTileFontSize	Specifies the font size of the title caption above graphs in the `Graphs` view of the `Views` pane. Default Value: `7` Valid Values: positive integers Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\Attributes`
GraphBkColor	Specifies the background color of all graphs in the `Graphs` view of the `Views` pane. Default Value: `0x7f7f7f` (gray) Valid Values: Six-digit hexadecimal numbers (interpreted as Blue-Green-Red color encoding). Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\Attributes`
GraphAxisFontSize	Specifies the font size of the captions under graphs in the `Graphs` view of the `Views` pane. Default Value: `7` Valid Values: Positive integers Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\Attributes`

FormatAgentInfo	Specifies how CCPulse+ displays agent information on the `Call Center Objects` pane of the CCPulse+ main window.
	Default Value: Varies depending on the language detected by your operating system
	Valid Values: Strings containing any combination of the following parameters:
	`%n`=Employee ID
	`%F`=First name
	`%f`=Abbreviated first name
	`%L`=Last name
	`%l`=Abbreviated last name
	For example, if you have an agent named Vince Duncan, whose employee number is `9001`, the string:
	`FormatAgentInfo = %n-%f.%L`
	yields the following display:
	`9001 - V. Duncan`
	Note: Separate the parameters using punctuation with which you want the elements to display.
	Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\CUSTOM DEFINITION`

Table 8: CCPulse+ Registry Settings (Continued)

Registry Setting	Description
DeltaAvgTime	Specifies the default value, in seconds, of the `Insensitivity` parameter for statistics in the `Average Time` category. Default Value: `2` Valid Values: `2–600` Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\CUSTOM DEFINITION`
DeltaTotalTime	Specifies the default value, in seconds, of the `Insensitivity` parameter for statistics in the `Total Time` category. Default Value: `10` Valid Values: `10–600` Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\CUSTOM DEFINITION`
DeltaCntCalls	Specifies the default value, in seconds, of the `Insensitivity` parameter for statistics in the `Current Calls` category. Default Value: `1` Appropriate Values: `1–100` Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\CUSTOM DEFINITION`
IntervalForCurrent Update	Specifies the default value, in seconds, of the `Notification Frequency` property for `Current State` and `Current Number` statistics. Default Value: `2` Valid Values: `2–600` Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\CUSTOM DEFINITION`
HistoricalUpdate Frequency	Specifies the default value, in seconds, of the `Notification Frequency` property for statistics in the `Historical Update` category. Default Value: `60` Valid Values: `60–600` Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\CUSTOM DEFINITION`
NotifyEvery	Specifies the default value, in seconds, for the `Notification Frequency` property for statistics in the `Notify Every` category. Default Value: `60` Valid Values: `60–1800` Path: `HKEY_CURRENT_USER\Software\GCTI\CallCenter\CUSTOM DEFINITION`

Chapter

3 Historical Reporting Considerations for CCPulse+

This chapter discusses CCPulse+ historical reporting. It covers certain points to consider before using CCPulse+ historical reporting functionality. Using this functionality is completely optional. If you intend to only use CCPulse+'s real-time functionality, you do not have to connect CCPulse+ to a Data Mart and you can skip this chapter. If you are interested in employing CCPulse+'s historical reporting functionality, read the following sections:

Data Mart Connections

You must configure a Data Mart connection on the `Connections` tab of your CCPulse+ `Application` object. If CCPulse+ cannot connect to the Data Mart, it cannot perform any historical reporting. If you use the CCPulse+ Wizard to configure CCPulse+, you can specify a Data Mart application on the ETL Runtime window. Otherwise, you must update your CCPulse+ `Application` object manually.

CCPulse+ does not maintain a constant connection to Data Mart. Instead, it connects for each request made and then disconnects until the next request. This functionality saves Data Mart resources.

Note: For CCPulse+, you can use only a Data Mart that has only one active Configuration Server defined to it. (*Active* indicates that Configuration Server has an ODS [Operational Data Storage] assigned to it within ETL Assistant.) If your Data Mart collects information from multiple Configuration Servers, you cannot use it with CCPulse+.

Composite Metrics

Prior to release 7.0, you could not use CCPulse+ to supply composite metrics in reports—except for those metrics already provided by the ready-to-use reports. To correct this shortcoming, the 7.0 release introduced a custom-formula capability, allowing you to define averages, percentages, and more. Refer to the "Template Wizard" topic in *Reporting 7.5 CCPulse+ Help* for information on how to use this feature.

Column Names

Data Mart stores information in folder templates that include only a single object (Agent, Place, Queue, and so on). However, in CCPulse+, you can create custom templates that include multiple compatible objects. To link object data stored in Data Mart, CCPulse+ identifies metrics by the Data Mart column name.

For this to function properly, the Data Mart folder template for each object to be incorporated into the CCPulse+ template must include the *identical* column name used in the CCPulse+ template. Otherwise, data from the Data Mart template is omitted from the CCPulse+ template.

Warning! If more than one custom template uses the same column name to specify different metrics, CCPulse+ may incorrectly combine these metrics.

For more information about the Data Mart folder templates and column names provided in reports from Genesys solutions, refer to the "Data Mart Folder Templates" and "Data Mart Composite Metrics" chapters in *Solution Reporting Templates* book of the *Reporting Technical Reference 7.6* series.

The Last N Intervals Parameter

CCPulse+ lets you select `Last N Intervals` as a time parameter for historical reporting. However, if no data exists in double the requested interval, CCPulse+ returns no data.

For example, if you request data from the last three weeks but data collection ceased several months ago for that object or metric, CCPulse+ searches for data in the previous six-week interval (double three weeks), and then returns a null response. This optimization methodology prevents resource-intensive searches for data existing at some unspecified point in the past. Refer to the "CCPulse+ Performance" section of the *Reporting Reference Manual* to learn about other parameters that affect both real-time and historical performance.

To retrieve such data, make a request for data from a specified set of dates.

Historical Associations for Statistics

You cannot set historical associations for custom report layouts until you complete the following:

1. Create a layout template using Data Modeling Assistant (DMA)
2. Assign a specific column name to it.
3. Create and activate a report layout based on the new layout template.

After the statistic has been propagated to Data Mart—which usually takes a few hours—you can make the historical association in CCPulse+.

> `Note:` CCPulse+ reads all available Data Mart metrics at start-up only. In order for CCPulse+ to accept newly created or changed metrics, you must restart it.

Propagation of Template Changes

Earlier releases of CCPulse+ did not propagate to existing historical views any changes that you made to the underlying CCPulse+ historical template. Starting with the 7.0.2 release, however, CCPulse+ does propagate such changes upon reload of your workspace. This behavior now parallels how CCPulse+ propagates changes made to real-time templates.

Deactivated Report Layouts

If you deactivate a report layout in DMA, CCPulse+ views based on that report layout cannot display any related data that may exist in the Data Mart.

Info Mart Connections for Query-Based Functionality

To use the new query functionality, add a connection to a Genesys Info Mart application on the `Connections` tab of your CCPulse+ `Application` object. If you do not add a connection to your properly configured Genesys Info Mart application, CCPulse+ will display all available Genesys Info Mart applications, and prompt you to choose one for queries. You cannot use the CCPulse+ 7.5 Wizard to configure a Genesys Info Mart application. Refer to the *Reporting Deployment Guide* for instructions on configuring and installing a Genesys Info Mart application.

Chapter

4 Server Connections to CCPulse+

The CCPulse+ `Application` object defines connections to Stat Server for monitoring real-time data and optionally to DB Server (through a connection to the specified Data Mart and its database access point) for monitoring historical data. In addition, the CCPulse+ `Application` object, which is defined in a particular Configuration Server, connects to this server (although this connection is not visibly defined within the application's properties).

This chapter describes CCPulse+ behavior when any of these server connections, or connections to the designated backup servers, fail. This chapter also describes what actions you should take to regain CCPulse+ monitoring under these circumstances.

The scenarios in this chapter begin with a CCPulse+ session for which all connections are established. For any number of reasons, any of the following connection failures, in any order, could occur to affect normal CCPulse+ operation:

Note: Except where specifically noted, this chapter describes the behavior of CCPulse+ release 7.0.2 and later, when disconnected from Configuration Server 7.0.2 (and later), Stat Server 7.0.3 (and later), and DB Server 7.*x*. CCPulse+'s reconnection procedures differ slightly for previous releases of these servers.

Configuration Server Connection Failure

CCPulse+ maintains two persistent connections to Configuration Server (or Configuration Server Proxy, if so configured). One connection is opened during login. If this connection is dropped, the CCPulse+ status bar instantly changes from `CfgServer ONLine` to `CfgServer OFFLine`. A broken Configuration Server connection means that CCPulse+ cannot receive updates about changes to configuration such as:

- Group composition changes (for example, changes to agent groups and place groups).
- Object additions or deletions (for example, the addition/deletion of DNs, agents, and queues).

When the first change occurs to an object in Configuration Server, CCPulse+ opens a second persistent connection to Configuration Server. This second connection runs in the background.

With regard to changes you make to the CCPulse+ `Application` object itself, CCPulse+ only reads this information upon startup; therefore, a Configuration Server disconnection after an initial successful connection affects CCPulse+'s ability to read its own configuration only upon next startup.

CCPulse+ next executes its reconnection procedure:

1. CCPulse+ displays the message box in Figure 10, which prompts you to select one of the three options:
   - `Connect to Backup Server(s)`
   - `Retry`
   - `Cancel`

Figure 10: Reconnection to Configuration Server Message

Selecting `Cancel` terminates the reconnection procedure. CCPulse+ makes no further attempts to reconnect to any Configuration Server.

> `Note:` CCPulse+ disables the first option if no backup Configuration Server has been designated.

2. If you select `Retry`, CCPulse+ attempts to reconnect to Configuration Server.

An unsuccessful attempt returns CCPulse+ to Step 1. A successful attempt terminates the reconnection procedure. CCPulse+ resumes receipt of Configuration Server updates.

3. If at Step 1, you select `Connect to Backup`, CCPulse+ attempts to reconnect to the Configuration Server designated as backup.

   An unsuccessful attempt returns CCPulse+ to Step 1. A successful attempt terminates the reconnection procedure. CCPulse+ now receives Configuration Server updates from the backup Configuration Server.

Upon a successful connection to either the primary or backup Configuration Server, the CCPulse+ status bar reads:

`CfgServer ONLine`

If, however, you opt to leave the reconnection loop without connecting to any Configuration Server, when Configuration Server (or its backup, if specified) does get back online, you must close and then reopen your workspace in order for CCPulse+ to acknowledge the connection and receive Configuration Server updates. CCPulse+ checks the status of its connection to Configuration Server every time a workspace is opened.

If Configuration Server fails when running in warm standby mode, the Solution Control Server (SCS) attempts to start the backup Configuration Server, if specified, and then designates this backup as primary. In hot standby mode, SCS immediately designates the backup as primary upon notification of failure.

CCPulse+'s connection to Configuration Server can fail for a number of reasons including:

- Someone closes Configuration Server.
- Someone kills the Configuration Server process.
- Network problems break CCPulse+'s connection to Configuration Server.

You can continue to use a previously opened and activated CCPulse+ workspace without a connection to Configuration Server; however, you cannot open or create a new workspace without this connection.

Stat Server Connection Failure

CCPulse+ maintains a continuous connection to Stat Server when there is an active workspace open. When this connection fails, the CCPulse+ status bar instantly changes from `StatServer ONLine` to `StatServer OFFLine`. A broken Stat Server connection means that:

- CCPulse+ cannot receive real-time statistical updates.
- You cannot apply thresholds or actions to real-time statistics (but you can apply these to historical statistics).
- You cannot open the `Extended Current Status` window for any object.

- You cannot open any real-time views.

CCPulse+ next executes its reconnection procedure which is identical to that for reconnection to Configuration Server. Refer to Steps 1 through 3 on page 42. If Stat Server restarts within the CCPulse+ reconnection loop, CCPulse+ resumes sending statistical requests to and receiving statistical updates from Stat Server, and the CCPulse+ status bar reads:

```
StatServer ONLine
```

If, however, you opt to leave the reconnection loop without connecting to any Stat Server, when Stat Server (or its backup, if specified) does get back online, you must close and then reopen your workspace in order for CCPulse+ to acknowledge the connection and resume normal operation. CCPulse+ checks the status of its connection to Stat Server only when you open a workspace or upon executing its reconnection procedure.

If Stat Server fails when running in warm standby mode, SCS attempts to start the backup Stat Server, if specified, and then designates this backup as primary. You cannot specify a hot-standby mode for a backup Stat Server.

CCPulse+'s connection to Stat Server can fail for a number of reasons including:

- Stat Server becomes overloaded.
- Someone closes Stat Server.
- Someone kills the Stat Server process.
- Network problems break CCPulse+'s connection to Stat Server.

You can continue to use a previously opened and activated CCPulse+ workspace without a connection to its Stat Server; although this mode of operation is not particularly useful. However, you cannot open or create a new workspace without this connection.

Data Mart Connection Failure

CCPulse+ does not maintain a continuous connection to the DB Server to Data Mart. Instead, CCPulse+ connects to this server on a request-by-request basis, such as when you refresh or create an historical view.

As a result, CCPulse+ has no reconnection procedure for DB Server. CCPulse+ either succeeds in connecting to DB Server for each request, or it does not. If CCPulse+ cannot connect to DB Server, CCPulse+ silently attempts to connect to its designated backup in order to access the Data Mart DBMS. This means you receive no notification of connection failure. If this backup is not running, CCPulse+ does not receive the historical information requested, and your historical view, for example, displays no data. Under these circumstances, you can continue to use CCPulse+ for reporting real-time data gathered directly from Stat Server, provided that this server is running.

Once DB Server is back online, you do not need to close and then reopen your workspace to regain historical reporting functionality. Merely re-request the historical information you seek by re-opening the related historical view.

CCPulse+'s automatic connection to a designated backup DB Server was a feature introduced in the 7.0.2 release.

Chapter

5 Using Custom Statistics

When creating or modifying templates, you can build custom statistics from the template's basic statistics by constructing your own formulas using JScript. A basic statistic has assigned to it a Stat Server stat type definition. This chapter includes the following topics:

Identifying Custom Statistics

In the CCPulse+ interface, you can distinguish basic statistics, which are predefined in CCPulse+ templates, from custom statistics by the appearance of the icon preceding the statistic name in the `Predefined Statistics` window of the Templates Wizard:

 denotes a basic statistic.

 denotes a custom statistic.

Figure 11, for example, shows four basic statistics and seven custom statistics from the `Callback Queue` template in a cutaway of the `Predefined Statistics` window.

Using JScript Language

CCPulse+ supports the JScript language for the custom statistics you define. As a result, you can use the arithmetic operators and delimiters available in this language, including: `+ / * – . ? : == ( ) ; != +=`

Refer to your JScript documentation for a description of these operators and syntax guidelines.

Referencing Basic Statistics

CCPulse+ uses the following syntax for referencing basic statistics in formulas:

```
ccpulse.group("StatisticGroupName").statistic("StatisticName");
```

The trailing semicolon is optional. CCPulse+ includes the name of the statistic group to distinguish the statistic, in case another statistic of the same name exists in another statistical group. If either `StatisticGroupName` or `StatisticName` contains no spaces, CCPulse+ drops the corresponding `group` and `statistic` delimiters, the parentheses, and the double quotes:

```
ccpulse.StatisticGroupName.StatisticName
```

Example 1 demonstrates the definition of the `All Entered` custom statistic, which is provided in the `Callback Queue` template.

Example 1

The `Callback Queue` template (provided with the `Voice Callback` [`VCB`] option) makes use of several custom statistics. Figure 11 shows a cutaway of CCPulse+'s `Predefined Statistics` window and many of this template's statistics. The `Requested Statistics` list in this window contains the template's categorization of statistic groups and statistics, and it highlights one custom statistic in particular, `All Entered`, which belongs to the `Total Entered` statistic group.

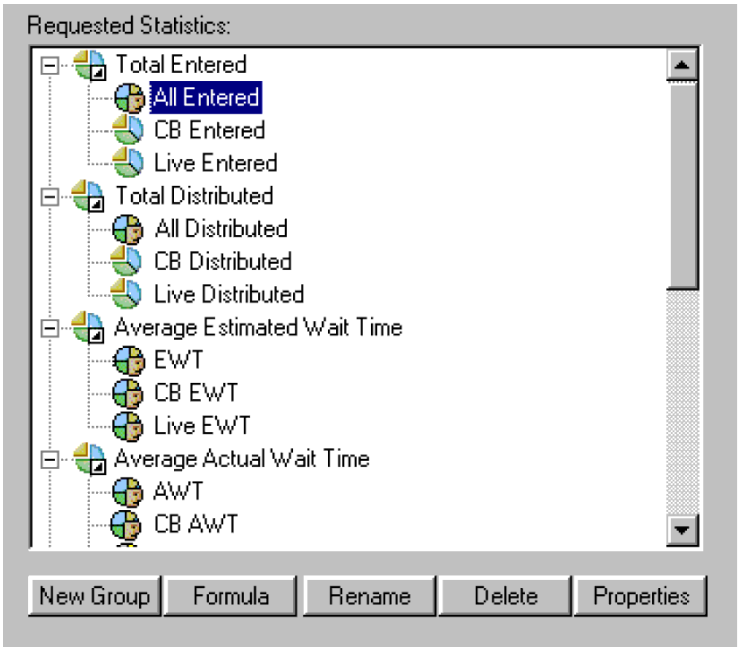

Figure 11: The Predefined Statistics Window

When you open the properties of this statistic, CCPulse+ opens the `ActiveX Scripting Expression` dialog box and displays the statistic's formula. Figure 12 provides a cutaway of the `ActiveX Scripting Expression` dialog box, showing that the definition of `All Entered` is based on two basic statistics, `CB Entered` and `Live Entered`, both of which belong to the `Total Entered` group.

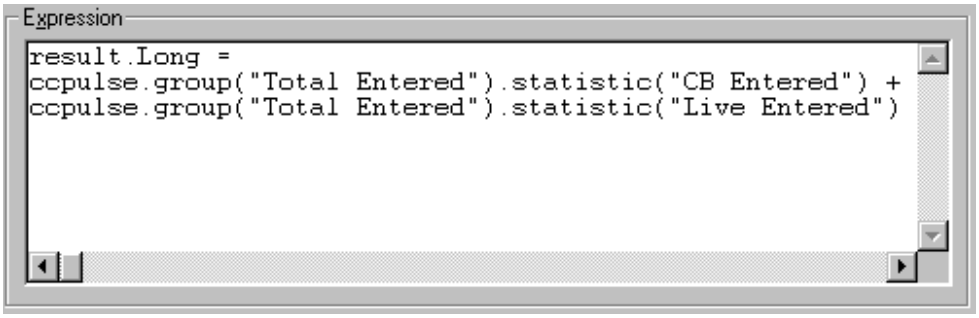

Figure 12: Definition of the All Entered Custom Statistic

Using Predefined Objects in Formulas

In the calculation of custom statistics, you can use qualitative information about a statistic other than its value by applying the `state`, `dn`, and/or `Action` predefined objects within the formula. Example 4 on page 54 provides an expansive example that uses the `state` predefined object in a formula.

Status-Related Information

To obtain information about the current status of a statistic in a formula, use the `state` predefined object and its properties (see Tables 9 and 10). To

activate this functionality, be sure to set the `ExtendedCurrentStatus` configuration option to `True` and to define the statistic's underlying stat type appropriately. The description of the option on page 28 provides these details.

Table 9: Properties of the state Object for AgentState

Property	Type	Description
type	String	Type of `state` object. In the 7.5.*x* release, extended current status information is provided only for `Agent` and `Agent Group` objects. Genesys recommends that you validate an object's `type` property as `AgentState` before using other properties of this object. If the `state` object is not available, the value of the `type` property is `undefined`. In this case, all other properties of the `state` object will be unavailable, which may result in errors.
StartTime	OLE Date	Coordinated Universal Time (UTC) date and time when the current status was established. Suitable for use in the `Date` object that is available in the JScript.
Status	String	Current status name, as provided by Stat Server.
AgentID	String	`EmployeeID`, usually set within Configuration Server.
PlaceID	String	ID of the place where the agent is logged in.
LoginID	String	Login ID used by the agent.
DNs	Collection	Contains information about the DNs used by or available to the agent. The list can be iterated by using `Enumerator` object (available in Jscript) to obtain specific DN data. Refer to the `GetDNState` function in Example 2 for an example on how to use this property. The list can also be accessed using the `Count` property and the `Value` function.
Capacities	Collection	Contains the collection of capacity items for an agent. Each capacity item contains the following properties: `MediaType`, `Current Interactions`, `RoutableInteractions`, and `MaxInteractions` (see Table 11). To return this collection of capacity items, set the `EnableAgentCapacity [Workspace]` configuration option to `true`.

Table 10: Properties of the state Object for AgentGroupState

Property	Type	Description
type	String	Type of `state` object. In the 7.5.*x* release, extended current status information is provided only for `Agent` and `Agent Group` objects. Genesys recommends that you validate an object's `type` property as `AgentGroupState` before using other properties of this object. If the `state` object is not available, the value of the `type` property is `undefined`. In this case, all other properties of the `state` object will be unavailable, which may result in errors.
StartTime	OLE Date	UTC date and time when the current status was established. Suitable for use in the `Date` object that is available in the JScript.
AgentGroupID	String	Name of the agent group.
Capacities	Collection	Contains the collection of capacity items for an agent group. Each capacity item contains the following properties: `MediaType`, `CurrentInteractions`, `RoutableInteractions`, and `MaxInteractions` (see Table 11). To return this collection of capacity items, set the `EnableAgentGroupCapacity [Workspace]` configuration option to `true`.

Table 11 provides a description of the properties of the Capacities items.

Table 11: Properties of the Capacity Items

Property	Type	Description
MediaType	String	The name of the media.
CurrentInteractions	Integer	The number of current interactions.
RoutableInteractions	Integer	The number of available routable interactions.
MaxInteractions	Integer	The maximum number of interactions.

The formula script can count the Capacities collection by using either the enumerator or the index (for example, by using `.value(index)` method). The number of capacity items are available through the `.Count` property. Certain capacity items for specific media are also available by using

`.value("media_name");` for example, `var voice_capacity = state.Capacities.Values("voice")`. Refer to Example 2 and Example 3.

Example 2

The following is an example of the formula script for `Agents`:

```
if(state.type == "AgentState")
{
      var i, item, s = "";
      for(i = 0; i < state.Capacities.Count; i++)
      {
          item = state.Capacities.Value(i);
          s += item.MediaType;
          s += "(" + item.CurrentInteractions + "," +
item.MaxInteractions + "," + item.RoutableInteractions + "),";
      }
      result.Text = s;
}
else
      result.Text = "n\\a";
```

Example 3

The following is an example of the formula script for `Agent Groups`:

```
if(state.type == "AgentGroupState")
{
      var i, item, s = "";
      for(i = 0; i < state.Capacities.Count; i++)
      {
          item = state.Capacities.Value(i);
          s += item.MediaType;
          s += "(" + item.CurrentInteractions + "," +
item.MaxInteractions + "," + item.RoutableInteractions + "), ";
      }
      result.Text = s;
}
else
      result.Text = "n\\a";
```

Note: In the CCPulse+ workspace, the `Graph` view displays only one number—current capacity per media.

DN-Related Information

To obtain DN-related information in a formula, use the `DN` predefined object with its properties listed in Table 12. This object can only be used in conjunction with the `state` predefined object.

Table 12: Properties of the DN Object

Property	Type	Description
type	String	This string contains the value `DNState`.
DNType	String	The directory number's type, such as `Position`, `Extension`, or `Chat`.
DN	String	The directory number.
StartTime	Long	Greenwich Mean Time (GMT) date and time when current status has been established. Suitable for use in the `Date` object available in the JScript.
Status	String	Current status name of the directory number as provided by Stat Server.
SwitchID	String	The switch's identification number, as registered in Configuration Server, to which the DN belongs.
Actions	Collection	Contains information about the list of actions currently occurring at a DN. DNs can hold several simultaneous actions. The list can be iterated by using `Enumerator` object (available in Jscript) to obtain specific agent data. Refer to the `GetAction` function in Example 4 for an example showing how to use this property.

Action-Related Information

To obtain action-related information use the `Action` predefined object and its properties described in the Table 13. This object can only be used in conjunction with DN predefined object. Please note that the names of masks can coincide with the names of statuses.

Table 13: Properties of the Action Object

Property	Type	Description
type	String	This string contains the value `Action`.
Action	String	Name of the action, such as those listed on page 64 without the `SDNA`, `SRPA`, or `SCMPA` prefixes.
StartTime	Long	GMT date and time when this action has been established. Suitable for use in the `Date` object available in the JScript.

Example 4

The following example illustrates how you can use the `state` predefined object within a customized statistic's formula to determine an agent's current state:

```
result.Text = GetAgentState();
function GetAgentState()
{
     if(state.type != "AgentState")
                 return "n/a";
     var r = "(" + FormatDate(state.StartTime) + ") ";
     r += state.AgentID;
     r += " [Place: " + state.PlaceID;
     r += " Login: " + state.LoginID + "]";
     r += ", Status: " + state.Status;
     r += "; DNs: ";

     for(var e = new Enumerator(state.DNs); !e.atEnd();
e.moveNext())
     {   r += GetDNState(e.item());   }
     return r;
}

function FormatDate(dateVal)
{
     var dateObj = new Date(dateVal);
     return dateObj.getHours() + ":" + dateObj.getMinutes() + ":"
             + dateObj.getSeconds();
}

function GetDNState(dn)
{
     var r = "(" + FormatDate(dn.StartTime) + ") ";
     r += dn.DN;
     r += " [Switch: " + dn.SwitchID;
     r += ", Type: " + dn.DNType + "], Status: " + dn.Status;
    r += "; Actions: "

     for(var e = new Enumerator(dn.Actions); !e.atEnd();
e.moveNext())
     {   r += GetAction(e.item()) + " ";   }
     return r;
}

function GetAction(a)
{
     var r = "(" + FormatDate(a.StartTime) + ") ";
     r += a.Action;
     return r;
}
```

The information that CCPulse+ returns is the same information that appears in the `Extended Current Status` window when requested for `Agent` objects. Figure 13 illustrates sample data returned in the Example 4 script. Note that the time information returned from running the script reflects startup time (the time that the current status occurred)—not duration as shown in the `Extended Current Status` window.

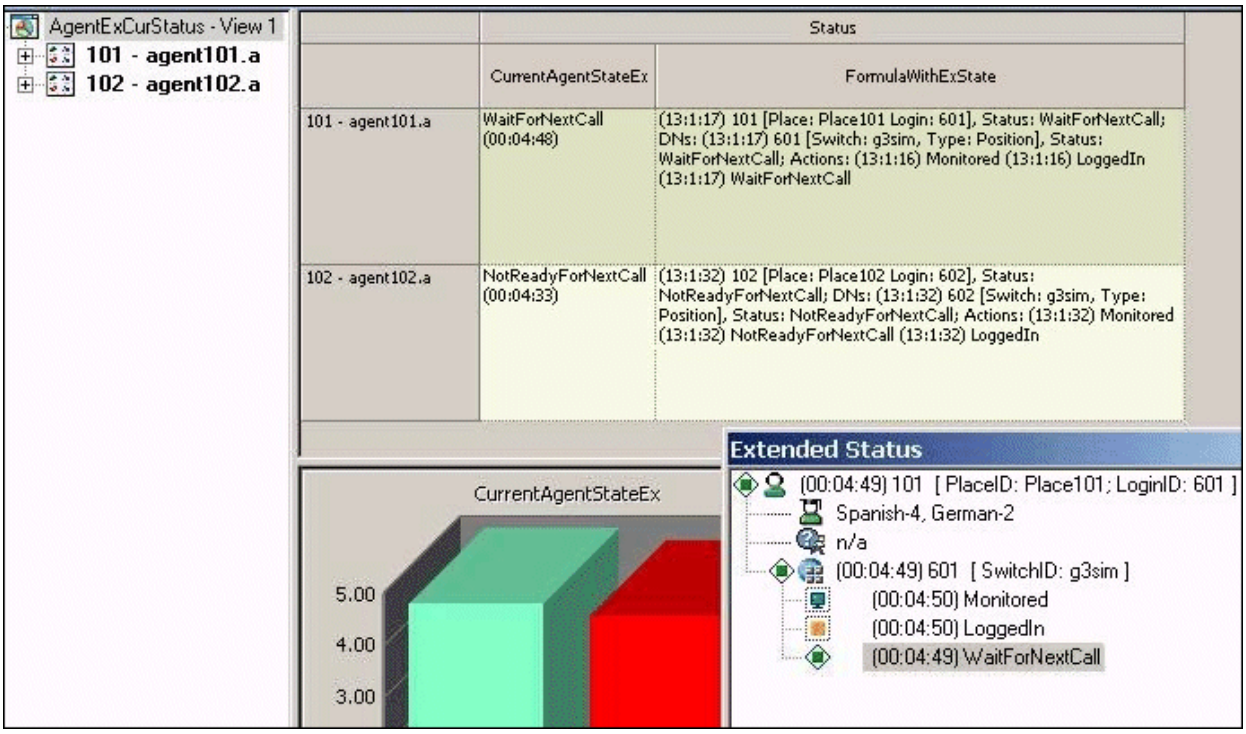

Figure 13: Example 4 Results

Formatting the Results

CCPulse+ enables you to specify the meaning of the result CCPulse+ returns for custom statistics:

- `result.Duration = formula`
  Displays the returned result in `hh:mm:ss` format.
- `result.Long = formula`
  Displays the returned result in integer format.
- `result.Float = formula`
  Displays the returned result in floating-point format.
- `result.Text = formula`
  Displays the returned result in string format.

If you omit the `result` object from the formula, CCPulse+ might not return the result as you expect it. For example, if you try to calculate duration based on a set of time statistics that you create, and if you do not use `result.duration` in

your formula, CCPulse+ might return a `Long` value and not apply the `hh:ss:mm` or other time format.

Figure 12 on page 49 illustrates the use of `result.Long` in the definition of the `All Entered` statistic.

Displaying Capacity Information

CCPulse+ provides two options for formatting the Capacities collection of statistics in the table and graph views.You can:

- Enter a user-defined format string (see Table 14).
- Select a predefined format (see Table 15).

To chose either option, from the `Statistic` dialog window in the CCPulse+ template wizard:

- Select the media for which CCPulse+ will display the capacity information.

> Note: The list of available media is populated by the values set for the `MediaTypes` configuration option. See Chapter 2 for more information on this option.

- Select the media format from the drop-down menu, or enter a user-defined format.

Figure 14 shows the predefined formats in a cutaway of the `Statistic` dialog box.

> Note: Capacity information is available in the `CurrentAgentTargetState` or `CurrentAgentGroupTargetState` statistic (`CurrentTargetState` statistical category).

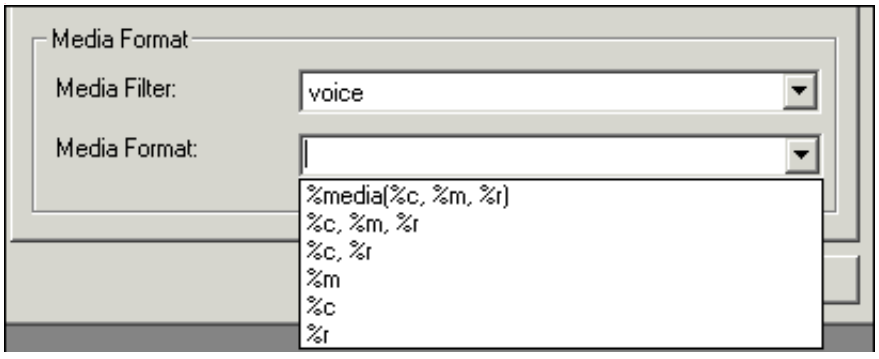

Figure 14: Cutaway of the Statistic Dialog Box

Table 14 lists the placeholder values for which you can substitute a user-defined sting. Note that you must preface the strings entered with a % sign.

Table 14: User-Defined Format Strings for Capacity Information

Placeholder	Type	Description
%media	String	Displays the name of the media in a single cell.
%m	Single integer	Displays the maximum interactions in a single cell.
%c	Single integer	Displays the current interactions in a single cell.
%r	Single integer	Displays routable interactions in a single cell.

Table 15 lists the predefined formats you can select from a drop-down menu of format choices.

Table 15: Predefined Formats for Capacity Information

Format	Example	Description
%media(%c, %m, %r)	voice(0, 3, 3)	Displays media name, current interactions, maximum interactions, and routable interactions in a single cell. This is the default format.
%c, %m, %r	0, 3, 3	Displays current interactions, maximum interactions, and routable interactions in a single cell delimited by commas.
%c, %r	0, 3	Displays current and routable interactions in a single cell delimited by commas.
%c	0	Displays the current interactions.
%r	3	Displays the routable interactions.
%m	3	Displays the maximum interactions.

Chapter

6 Thresholds and Actions

By linking thresholds and actions, you can configure CCPulse+ so that it performs a specific action whenever the criteria you set in a threshold definition are met. This chapter contains the following sections:

Overview

This chapter presents:

- The predefined objects that you can use in the threshold and action VBScripts to create and edit thresholds and actions.
- The predefined thresholds installed with CCPulse+.
- The predefined actions installed with CCPulse+.

All existing thresholds and actions that are applied to the root level of a view are propagated automatically to any new objects that are added to the view. You do *not* have to manually apply thresholds to statistics of newly added objects.

You can add a customized threshold to a statistic of a specific contact center object, even if there is a pre-existing threshold applied at the root level. In this case, the customized threshold will take precedence. Any changes to the root-level threshold will not affect a statistic with a customized threshold. A new icon will designate the associated customized threshold.

For instructions on using the Threshold Wizard and Action Wizard to create and edit thresholds and actions, and for information on associating thresholds and actions for specific contact center objects, see *Reporting 7.5 CCPulse+ Help*.

Predefined Objects

Use the predefined objects listed and described in Table 16 to modify the predefined CCPulse+ thresholds and actions or to create new ones using the CCPulse+ Threshold and/or Action wizards.

Table 16: Predefined Objects for Use in Threshold and Action Scripts

Object	Type	Description
Threshold Use this object to set and retrieve the data used in threshold calculation, which is available in both threshold and action scripts.		
Result	Boolean	Contains the result of the execution of a threshold script. If the result is not set in the script, it is assumed to be `False`. Example: `Threshold.Result = True`
StatValue	Variant	Contains the current value of the statistic to which the threshold script is applied. Example: `If (Threshold.StatValue = x) then ...`
StatAlias	String	Alias of the statistic, as defined in the template. **Note:** The alias is an empty string if a threshold is applied to a configuration object in the main tree.
ActionDelay	Integer	Used to delay action execution by *n* seconds. If the threshold script that set this value is executed again before its actions are completed, and the result is `False`, action execution is canceled. This allows for so-called duration thresholds, the actions of which are executed if a certain condition is `True` for the amount of time specified. Example scenario: The value of a statistic changes. CCPulse+ executes the `Threshold1` script, which contains an action delay of 10 seconds. The script returns a `True` value. You can expect CCPulse+ to execute the associated actions 10 seconds later. • If the value of the same statistic changes 5 seconds later, CCPulse+ re-executes the `Threshold1` script. If the script returns `False`, CCPulse+ cancels execution of the delayed actions. • If there is no statistic changes before the 10 seconds elapse, CCPulse+ executes the actions, thereby showing that a particular condition was `True` for the duration of 10 seconds. Example: `Threshold.ActionDelay = 10`

Table 16: Predefined Objects for Use in Threshold and Action Scripts (Continued)

Object	Type	Description
ActionDelay (continued)		**Note:** ActionDelay functionality is based on the duration of Current Status only—not Current Status and Reason together, unless status and reason are coupled together at the moment of an actual change in Current Status. To obtain the correct time duration when Current Status and Reason are coupled together, you can configure your statistic to use the `CurrentTime` statistical category with a specific status mask that is filtered by a specific reason.
StatReasons	String	Use this read-only property to evaluate reasons in threshold and action scripts. The following threshold script illustrates how you can use this property when it is applied to an agent `CurrentState` statistic: `If (Threshold.StatValue = StatAction.SDNANotReadyForNnextCall and Threshold.StatReasons = "Lunch") then Threshold.Result = True end if` Note that reasons and action codes can appear only for following agent states: • `LoggedIn` • `NotReadyForNextCall` • `AfterCallWork` • `WaitForNextCall` Reasons can originate from more than one source (T-Server, hardware, or other), and they can be provided to CCPulse+, via Stat Server, in string or integer format. CCPulse+ converts all reasons to string format, separating each with a semicolon (;). To effectively evaluate the concatenated string, the CCPulse+ Administrator might have to parse it using VBScript.
Configuration Object Properties These represent the primary properties of any configuration object (also available in action scripts).		
CFGTenantDbid	Long	tenant dbid Example: `Threshold.CFGTenantDbid`
CFGDbid	Long	object dbid Example: `Threshold.CFGDbid`
CFGObjectID	String	object ID Example: `Threshold.CFGObjectID`

Table 16: Predefined Objects for Use in Threshold and Action Scripts (Continued)

Object	Type	Description
CFG*Type*	Integer	Type of object (`short CfgObjectType` enumeration value, for instance, value `4` means `CFGPlace`). The enumeration values for the available object types are as follows: • `Switch–1` • `DN–2` • `Person–3` • `Place–4` • `AgentGroup–5` • `PlaceGroup–6` • `Tenant–7` • `Script–12` • `DNGroup–17` • `CallingList–26` • `Campaign–27` Example: `Threshold.CFGDNGroup` CCPulse+ support for the `CFGScript` predefined object was introduced in the 7.0.1 release.
CCPulseNotifier Use this object to send notifications to CCPulse+ (also available in action scripts).		
SetColor	Color: Integer	Notifies CCPulse+ of a color change. A member of the Color object can be passed as a parameter. Example: `CCPulseNotifier.SetColor(Color.Red)`. Or a user-defined RGB value can be passed instead. Example: `CCPulseNotifier.SetColor(RGB(50,50,50))`. For further information about the RGB function, consult your Visual Basic Script (VBScript) documentation. **Note:** You must set color commands for color names using the English language.
ResetColor	N/A	Notifies CCPulse+ that the color of the object (set by `SetColor`) should be reset to its original color. Example: `CCPulseNotifier.ResetColor`
Hide	N/A	Use this operation to hide a configuration object in the `Call Center Objects` pane or `Tree` view, depending on the threshold application.
Show	N/A	Use this operation to show a configuration object on the `Call Center Objects` pane, the `Tree` view, and the `Graph` view. Used to display an object that was previously hidden using the `Hide` action associated with a different threshold.

Table 16: Predefined Objects for Use in Threshold and Action Scripts (Continued)

Object	Type	Description
Color (Enumeration Object) The properties of this object are constant RGB values that you can use with the `SetColor` and `ResetColor` methods (also available in action scripts). Some commonly used color codes are listed below. Examples: `CCPulseNotifier.SetColor(Color.Red)` or `CCPulseNotifier.SetColor(RGB(255,0,0))` The following commands (and their corresponding RGB values) are available with the `Color` object: Red RGB (255,0,0) Blue RGB (0,0,255) Green RGB (0,255,0) DarkRed RGB (128,0,0) DarkGreen RGB (0,128,0) Cyan RGB (0,255,255) DarkCyan RGB (0,128,128) Magenta RGB (255,0,255) DarkMagenta RGB (128,0,128) Yellow RGB (255,255,0) White RGB (255,255,255) Black RGB (0,0,0) Gray RGB (192,192,192) DarkGray RGB (128,128,128) **Note:** You must set color commands for color names using the English language.		

Table 16: Predefined Objects for Use in Threshold and Action Scripts (Continued)

Object	Type	Description
StatAction (Enumeration Object) The properties of this object are the names of the DNs, route points, and campaign actions as described in the statistics library (available only in threshold scripts). Example: `if ( Threshold.StatValue = StatAction.SDNALoggedIn ) then ...`		

Current Status Name	Current State Value
SDNAWaitForNextCall	4
SDNACallDialing	6
SDNACallRinging	7
SDNANotReadyForNextCall	8
SDNACallUnknown	18
SDNACallConsult	19
SDNACallInternal	20
SDNACallOutbound	21
SDNACallInbound	22
SDNALoggedOut	23
SDNACallOnHold	13
SDNANotMonitored	0
SDNAMonitored	1
SDNALoggedIn	2
SDNAOnHook	3
SDNAOffHook	5
SDNAAfterCallWork	9
SRPANotMonitored	0
SRPAMonitored	1
SRPANotReadyForNextCall	2
SDNAASM_Engaged	16
SDNAASM_Outbound	17
SRPAWaitForNextCall	13
SRPACallWait	9
SCMPAStatusDeactivated	0
SCMPAStatusActivated	1
SCMPAStatusRunning	2

Each Current Status has an associated Current State value which is constant.

The prefixes in the listing to the left mean the following:

DNA—Directory Number Action

RPA—Routing Point Action

CMPA—Campaign Action

The *S* prefix on each indicates that the item is a Stat Server entity (rather than an entity of Configuration Server or any other server). Elsewhere within the CCPulse+ interface, these actions appear without these prefixes.

Example:

```
if (Threshold.StatValue=StatAction.SDNANotMonitored) then ...
   Threshold.Result=True
end if
```

CCPulse+ executes the corresponding action script when the selected DN statistic achieves `NotMonitored` status.

Note: *ASM* is a reference to the Active Switching Matrix call model used in conjunction with predictive dialing in the Outbound Contact solution. Refer to Genesys Outbound Contact documentation for further details.

Predefined Thresholds

This section describes the threshold scripts provided with each Genesys solution. The purpose of providing these scripts is to demonstrate how you might apply thresholds within your own CCPulse+ environment.

The scripts and threshold definitions are identical for each Genesys solution and are stored in a file named `thresholds.stg`.

Note: Because these are sample scripts, it is likely that their absolute values have little significance within your environment. You are encouraged to modify them to more meaningful values.

How to Use Thresholds

The following discussion assumes you are familiar with the VBScript programming language and with the CCPulse+ predefined objects described in “Predefined Objects” on page 60.

To be effective, you must use thresholds in conjunction with actions. When using thresholds and actions, keep in mind these points:

- CCPulse+ executes associated actions only if the threshold returns a `True` result.

  This means that you cannot create a threshold to perform one action if its result is `True` and another action if its result is `False`.
- With CCPulse+, you can apply more than one threshold to a statistic.

  You can have one threshold call the first action, the second threshold call the second action, and so on.
- You can apply one or more actions to a particular threshold.
- You can apply a chain of thresholds to a particular statistic.

  CCPulse+ executes the thresholds in the order you set in the `Threshold/Action Association` dialog box. The first threshold returning a `True` result stops the launch of all remaining thresholds in the chain. Any attached action(s) will be run in the order set. The next statistic’s value repeats the process.
- You can also use object linking and embedding (OLE) automation objects within threshold scripts, because VBScript allows this.

  Note: It is *not* recommended to use OLE objects with either a GUI-like appearance or with dialog boxes that require explicit user confirmation.

  And, starting with release 7.2, you can configure the `SafeScriptMode` option to specify whether CCPulse+ spawns a separate thread to execute

VBScripts. Refer to “SafeScriptMode” on page 26 for more information about this option.

Real-World Example

Consider the following business scenario, which uses thresholds and actions to aid a CCPulse+ operator in managing a queue named `ClientQueue`.

Management has decided that 5 calls or fewer waiting in `ClientQueue` is the desired situation; 10 or more calls waiting represents an alarming condition; and anywhere in between is acceptable.

Three custom thresholds, `UnderCapacity`, `AtCapacity`, and `OverCapacity`, have been applied to the `CallsWaiting` statistic, which is a statistic that is monitored for `ClientQueue`. The custom thresholds are defined as follows:

UnderCapacity

```
if Threshold.StatValue < 5 then
   Threshold.Result = true
end if
```

AtCapacity

```
if Threshold.StatValue >= 5 and Threshold.StatValue < 10 then
   Threshold.Result = true
end if
```

OverCapacity

```
if Threshold.StatValue >= 10 then
   Threshold.Result = true
end if
```

A color action is assigned to each threshold to change the background color of a table cell holding the `CallsWaiting` statistic when CCPulse+ evaluates the threshold to be `True`. The `UnderCapacity` threshold triggers the `SetGreen` action. `AtCapacity` triggers the `SetYellow` action, and `OverCapacity` triggers the `SetRed` action.

Now CCPulse+ users have a visual cue to quickly assess `ClientQueue` performance.

Explanations of the Predefined Thresholds

For each predefined threshold, this section gives the VBScript and the object types to which you can apply it.

Note: Some of these scripts include numbers by default. These numbers are intended simply to indicate where to insert a value in the script. You must replace these numbers with ones suited to your environment.

AgentLoggedIn

This threshold script returns a `True` value when the agent is logged in—that is, when CCPulse+ receives the `Logged In` status from Stat Server.

```
if Threshold.StatValue = StatAction.SDNALoggedIn then
   Threshold.Result = true
end if
```

Used with the `Agent` object.

Note: Stat Server reports to its clients the highest ranking status of an object. After an agent logs in, his/her status almost immediately changes from `Logged In` to `Ready` or `Not Ready`, which holds a higher ranking. Hence, the threshold script above is rarely tripped even though an agent may be logged in for quite some time. For this reason, Genesys recommends that you revise your script to reflect the condition when agents have not logged out:

```
if Threshold.StatValue <> StatAction.SDNALoggedOut then
   Threshold.Result = true
end if
```

AgentLoggedOut

The threshold returns a `True` value when the agent is logged out.

```
if Threshold.StatValue = StatAction.SDNALoggedOut then
   Threshold.Result = true
end if
```

Used with the `Agent` object.

Using the `Hide` action with this threshold removes the logged out agent from the `Table` view as well as the `Tree` view.

Note: This functionality does not apply to the `Graph` view.

CampaignActivated

This threshold script returns a `True` value when the `Campaign` object has been activated.

```
if Threshold.StatValue = StatAction.SCMPAStatusActivated then
   Threshold.Result = true
end if
```

Used with the `Campaign` object.

CampaignDeactivated

This threshold script returns a `True` value when the `Campaign` object has been deactivated.

```
if Threshold.StatValue = StatAction.SCMPAStatusDeactivated then
   Threshold.Result = true
end if
```

Used with the `Campaign` object.

ExactValue

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied equals the value you set. By default, the value is set to `3`.

```
if Threshold.StatValue = 3 then
   Threshold.Result = true
else
   Threshold.Result = false
end if
```

ExactValueWDuration

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied equals the value you set. By default, this value is set to `3`. If `True`, CCPulse+ executes the associated action a specified number of seconds later. By default, `ActionDelay` is set to `10`.

```
if Threshold.StatValue = 3 then
   Threshold.ActionDelay = 10
   Threshold.Result = true
else
   Threshold.Result = false
end if
```

GreaterThanLimit

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied is greater than the value you set. By default, this value is set to `10`.

```
if ( Threshold.StatValue > 10) then
    Threshold.Result = True
else
    Threshold.Result = False
end if
```

GreaterThanLimitWDuration

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied is greater than the value you set. By default, this value is set to `10`. If `True`, CCPulse+ executes the associated action a specified number of seconds later. By default, `ActionDelay` is set to `10`.

```
if Threshold.StatValue > 10 then
   Threshold.ActionDelay = 10
   Threshold.Result = True
else
   Threshold.Result = False
end if
```

LessThanSpecifiedLimit

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied is less than the value you set. By default, this value is set to `777`.

```
if (Threshold.StatValue < 777) then
    Threshold.Result = True
else
    Threshold.Result = False
end if
```

Used with the `Calling List` object.

LessThanLimit

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied is less than the value you set. By default, this value is set to `10`.

```
if ( Threshold.StatValue < 10 ) then
    Threshold.Result = True
else
    Threshold.Result = False
end if
```

LessThanLimitWDuration

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied is less than the value you set. By default, this value is set to `10`. If `True`, CCPulse+ executes the associated action a specified number of seconds later. By default, `ActionDelay` is set to `10`.

```
if Threshold.StatValue < 10 then
   Threshold.ActionDelay = 10
   Threshold.Result = True
else
   Threshold.Result = False
end if
```

NotEqual

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied equals any nonnull value other than the value you set. By default, this value is set to `3`.

```
if Threshold.StatValue <> 3 then
   Threshold.Result = true
else
   Threshold.Result = false
end if
```

NotEqualWDuration

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied equals any nonnull value other than the value you set. By default, this value is set to `3`. If `True`, CCPulse+ executes the associated action a specified number of seconds later. By default, `ActionDelay` is set to `10`.

```
if Threshold.StatValue <> 3 then
   Threshold.ActionDelay = 10
   Threshold.Result = true
else
   Threshold.Result = false
end if
```

OutOfRange

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied lies outside the range of values between the values you set. By default, these values are set to `5` and `10`.

```
if Threshold.StatValue < 5 Or Threshold.StatValue > 10 then
   Threshold.Result = true
else
```

```
   Threshold.Result = false
end if
```

OutOfRangeWDuration

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied lies outside the range of values between the values you set. By default, these values are set to `5` and `10`. If `True`, CCPulse+ executes the associated action a specified number of seconds later. By default, `ActionDelay` is set to `10`.

```
if Threshold.StatValue < 5 Or Threshold.StatValue > 10 then
   Threshold.ActionDelay = 10
   Threshold.Result = true
else
   Threshold.Result = false
end if
```

WithinRange

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied lies exclusively between the values you set. By default, these values are set to `5` and `10`.

```
if Threshold.StatValue  > 5 and Threshold.StatValue < 10 then
   Threshold.Result = true
else
   Threshold.Result = false
end if
```

WithinRangeOfValues

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied lies exclusively between the values you set. By default, these values are set to `50` and `1000`.

```
if Threshold.StatValue  > 50 and Threshold.StatValue < 1000 then
   Threshold.Result = true
else
   Threshold.Result = false
end if
```

WithinRangeWDuration

This threshold script returns a `True` value if the value of the statistic to which the threshold is applied lies exclusively between the values you set. By default, these values are set to `5` and `10`. If `True`, CCPulse+ executes the associated action a specified number of seconds later. By default, `ActionDelay` is set to `10`.

```
if Threshold.StatValue  > 5 and Threshold.StatValue < 10 then
   Threshold.ActionDelay = 10
   Threshold.Result = true
else
   Threshold.Result = false
end if
```

Predefined Actions

This section describes the action scripts provided with each Genesys solution. Actions enable CCPulse+ to alert you when a particular condition has been met with respect to a statistic's value.

This section provides the information you need to customize the action scripts to meet your contact center requirements. The scripts are identical for each Genesys solution and are stored in a file named `actions.stg`.

Actions have little meaning outside the context of their associated thresholds. Therefore, review "Predefined Thresholds" on page 65 and see the "Threshold/Action Association" topic in *Reporting 7.5 CCPulse+ Help*.

How Actions Function

The results of actions persist until CCPulse+ evaluates the result of the associated threshold to be `False`. For example, if you have used the `SetColor` action to apply a `Green` color when the value of `TotalCalls` is greater then `100`, the `Green` color persists until the value of `TotalCalls` falls below `100`.

In this example:

- `TotalCalls` is the targeted statistic.
- `GreaterThan100` is the applied threshold.
- `SetColorGreen` is the threshold's associated action.

The following discussion assumes you are familiar with the VBScript programming language and with the CCPulse+ predefined objects described in "Predefined Objects" on page 60.

Explanations of the Predefined Actions

In the descriptions below, the term *targeted statistic* refers to the statistic to which CCPulse+ applies the action's associated threshold.

Hide

This action hides the object to which the action's associated threshold has been assigned on the `Call Center Objects` pane, in the `Tree` view, and in the `Grid` view (the row containing the specified statistic). CCPulse+ hides the object

when the conditions set for the associated threshold are `True`. The object remains hidden unless you configure a second threshold/action combination for the targeted statistic using an alternative threshold condition and the `Show` action.

```
CCPulseNotifier.Hide
```

Show

This action displays or makes reappear the targeted object in the `Call Center Objects` pane, in the `Tree` view, and in the `Graph` view (the row containing the specified statistic). The object remains visible unless you configure a second threshold/action combination for the targeted statistic using an alternative threshold condition and the `Hide` action.

```
CCPulseNotifier.Show
```

ResetColor

This action sets the background color of the table cell containing the targeted statistic to its default color. The default color is either the color set in the Registry settings (see “GraphBkColor” on page 34) or the color specified in a previous `SetColor` action, if any.

```
CCPulseNotifier.ResetColor
```

SetColor

This action sets the background color of the table cell containing the targeted statistic to whatever color you choose, for example, dark red.

```
CCPulseNotifier.SetColor( Color.DarkRed )
```

You can modify this action script to use any of the CCPulse+ predefined colors or you can specify a custom color in RGB format; for example:

```
CCPulseNotifier.SetColor( RGB( 7, 77, 177 ) )
```

The available predefined colors are:

- `Blue`
- `Green`
- `Red`
- `White`
- `Black`
- `Gray`
- `DarkGray`
- `DarkRed,`
- `Yellow`
- `DarkYellow`
- `DarkGreen`
- `Cyan`
- `DarkCyan`
- `DarkBlue`
- `Magenta`
- `DarkMagenta`

WAVAudioAlert

You can implement COM objects using Visual Basic or C++.

The sample provided uses a COM component defined in the `GsSampleTE Objects.DLL` file. The source code for this file is included in the CCPulse+ installation package. For this script to work, you must compile it using Visual Basic.

Note: This sample is provided “as is” for your reference. Although every effort has been taken to ensure that this sample functions correctly, Genesys does not offer product support for it.

To create the Genesys Sound object:

```
Set SoundObject =
CreateObject("GsSampleTEObjects.GsActionSoundObject")
CreateObject("GsSampleTEObjects.GsActionFSObject")
```

Specify the full path to a .WAV file in order to play it—for example:

```
SoundObject.PlaySoundFile (“c:\winnt\media\tada.wav”)
```

PCSpeakerAudioAlert

This action uses a COM component defined in the GsSampleTEObjects.DLL file. The source code for this file is included in the CCPulse+ installation package. For this script to work, you must compile it using Visual Basic.

To create the Genesys Sound object:

```
Set SoundObject =
CreateObject("GsSampleTEObjects.GsActionSoundObject")
SoundObject.Beep
```

ExecuteProcess

The sample provided uses a COM component defined in the GsSampleTE Objects.DLL file. The source code for this file is included in the CCPulse+ installation package. For this script to work, you must compile it using Visual Basic.

Note: This sample is provided “as is” for your reference. Although every effort has been taken to ensure that this sample functions correctly, Genesys does not offer product support for it.

To create the Genesys File System object (FSObject), use this command:

```
Set FSObject =
CreateObject("GsSampleTEObjects.GsActionFSObject")
```

Specify the full path to an executable file in order to run it. For example, to open Internet Explorer using Windows NT, enter:

```
FSObject.Execute("C:\Program Files\Plus!\Microsoft
Internet\iexplore.exe")
```

SendE-MailWithCDO

Preinstall Microsoft Outlook 98 or Microsoft Exchange Server version 5.5 to execute this script.

Body of the e-mail message:

```
Dim msgBody
Dim thresholdInfo
msgBody="A mail from the CCPulse+ threshold engine!"+vbNewLine
 thresholdInfo="StatValue:"+CStr(Threshold.StatValue)+vbNewLine+
 "StatAlias:"+CStr(Threshold.StatAlias)+vbNewLine+
 "CFGTenantDbid:"+CStr(Threshold.CFGTenantDbid)+vbNewLine+
 "CFGDbid:"+CStr(Threshold.CFGDbid) + vbNewLine +
 "CFGObjectID:"+CStr(Threshold.CFGObjectID)+vbNewLine+
 "CFGType:"+CStr(Threshold.CFGType)+vbNewLine
```

Call this function using `profile`, `recipient`, `message`, and `subject`, as shown below:

```
SendMail "","",msgBody+thresholdInfo,"Alarm notification from
CCPulse+"


Sub SendMail(profile,recipient,message,subject)
    Dim objSession, objAddrEntry, oInbox, colMessages, oMessage,
colRecipients, oRecipient

    Set objSession = CreateObject("MAPI.Session")
    objSession.Logon profile,"",True,False,0, False

    Set objAddrEntry = objSession.CurrentUser
    Set oInbox = objSession.Inbox
    Set colMessages = oInbox.Messages
    Set oMessage = colMessages.Add()
    Set colRecipients = oMessage.Recipients

    If recipient = "" Then
        colRecipients.Add  objAddrEntry
    Else
        colRecipients.Add  recipient
    End If
    colRecipients.Resolve

    oMessage.Subject = subject
    oMessage.Text = msg
    oMessage.Send
```

```
    objSession.Logoff
    Set objSession = nothing

End Sub
```

ActivateWebBrowser

This action opens Microsoft Internet Explorer and navigates to the specified URL. Use the following script format:

```
Dim Browser
```

Create the Explorer object:

```
Set Browser =
CreateObject("InternetExplorer.Application.1")
```

Then open the URL:

```
Browser.Navigate("www.genesyslab.com")
```

To show the browser window:

```
Browser.Visible = True
```

Chapter

7 Measuring CCPulse+/Stat Server Traffic

If your environment places CCPulse+ and Stat Server on different local area networks (LANs), you might be interested in determining the impact of CCPulse+/Stat Server traffic on your wide area network (WAN). This chapter shows how you can approximate CCPulse+/Stat Server traffic flow and includes the following sections:

Determining Factors

Traffic between CCPulse+ and Stat Server is dependent on many factors, including:

- The speed of both computers.
- The number of requests and responses in the data stream.
- Network speed—for example, a network operating at 100 Mbits/second might yield poor performance.

The data stream is dependent on how fast your network operates. Obviously, a fast network prevents many traffic problems.

The Number of Requests

The traffic specifically from CCPulse+ to Stat Server depends on:

- The number of statistics requested by CCPulse+ views.

- The number of monitored agents, agent groups, places, place groups, queues, groups of queues, routing points, and groups of routing points in the CCPulse+ workspace. CCPulse+ issues one current state request for each.

Moreover, if you request the extended current status for particular object, CCPulse+ requests current state information for the entire status tree of that object which includes subordinate objects. For the extended current status of an agent group, for example, CCPulse+ also receives the current state for all agents, places, DNs, and DN actions within that group. Thus, any formula measuring the number of requests is dependent on the content of a particular workspace and the objects for which you requested the extended current status.

The Number of Responses

The traffic specifically from Stat Server to CCPulse+ is dependent on the notification mode specified for each statistic along with each request. There are four types of notification modes:

- Change-based—Stat Server sends the current value whenever the value of the statistic significantly changes. A statistic's `Insensitivity` setting determines what a significant change in value is. Higher `Insensitivity` settings typically reduce network traffic, but they also reduce reporting accuracy because values are not updated as frequently.
- Time based—Stat Server sends the current value every *x* seconds, in accordance with the specified notification frequency. Lower notification frequencies reduce network traffic, but they also affect the availability of data in real time. For instance, instead of having updates every 2 seconds, a lower frequency specification may provide updates every 60 seconds.
- Reset based—Stat Server sends the current value right before setting it to `0`.
- No notification—Stat Server sends a response only when you issue a `Peek Statistic` request.

Depending on which notification mode you define for each statistic, Stat Server might send no response, one response, or constant responses. In addition to these four notification modes, the user can request updates at any time by using the Peek Statistics feature. Refer to the *Framework Stat Server User's Guide* for further information about statistics.

The Size of One Request/Response

Except for extended current status requests and responses, the size of each statistical request and response is fixed and significantly less than 256 bytes. The response size is variable for extended current status requests and depends on the object type and subject parameter of the request.

Chapter

8 Troubleshooting

This chapter presents some scenarios that you might have to troubleshoot while using CCPulse+ and suggests possible solutions for each.

If you are experiencing difficulties with your CCPulse+ application, browse the following list for the resolutions to some commonly encountered issues.

If you continue to have problems after reviewing this chapter and verifying that your hardware is functioning properly and that your CCPulse+ configuration settings are correct, contact Genesys Technical Support.

These issues are covered in this chapter:

Agents Having Trouble Logging In

Confirm that the user name and password are correct, keeping in mind that both are case-sensitive. Confirm that the `Settings` fields on the CCPulse+ `Configuration Server` tab are entered correctly.

Wizards Are Disabled

In Configuration Manager, check that you have been assigned the rank of `Administrator`. CCPulse+ disables the Template, Threshold, and Action wizards for persons with no rank, or with the rank of `User`. Figure 15, for example, shows that `Administrator` ranking is being added to the properties of `Agent 7005`, who previously had no assigned ranking.

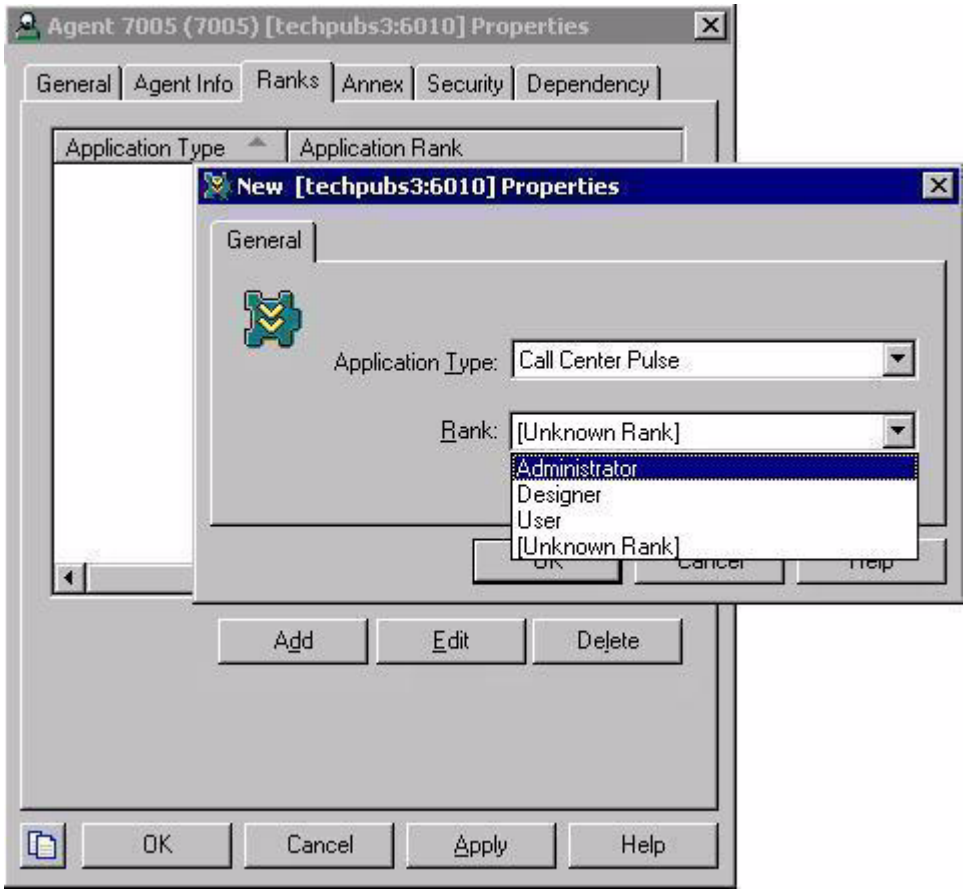

Figure 15: Adding Administrator Ranking to Agent Properties

Import/Export Utility Is Disabled

In Configuration Manager, check that you have been assigned the rank of `Administrator` (see Figure 15). CCPulse+ disables the Import/Export Utility for persons with no rank, or with the rank of `User`.

No DB Server Connection Error Message

If you receive the error message shown in Figure 16, verify in Configuration Manager that you have correctly configured the database access point to the Data Mart in the CCPulse+ `Application` object.

Figure 16: No DB Server Connection Error Message

Agents Missing from Call Center Objects Pane

When selecting object types in the Configuration Wizard, select the `IncludeAll (Type:Agent)` check box. See Figure 17.

Be sure that agents have been set up properly in Configuration Manager. See the “Configuration Wizard” topic in *Reporting 7.5 CCPulse+ Help* for more information.

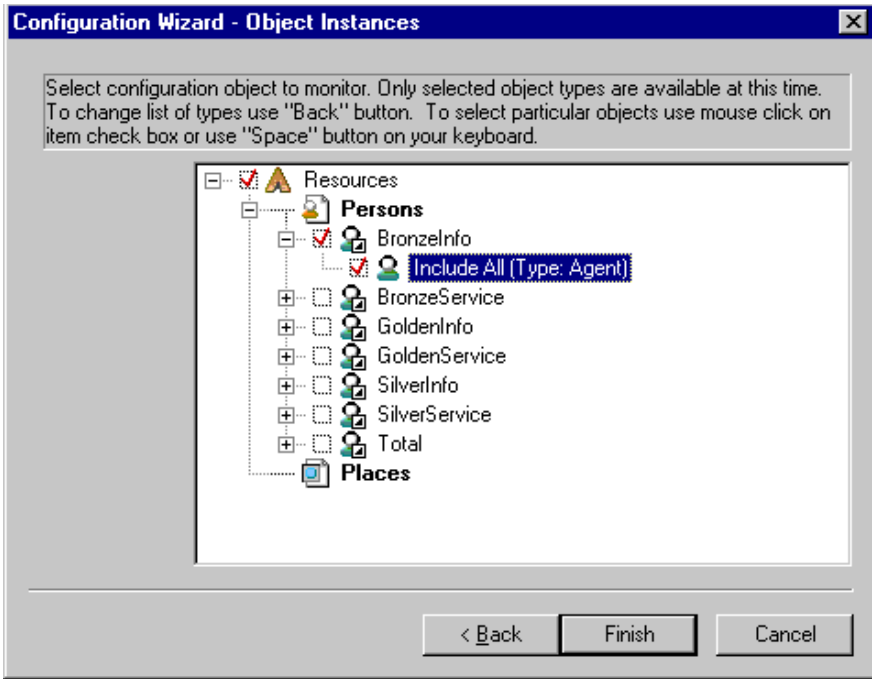

Figure 17: Object Instances

General SQL Server Error Message

"General SQL Server error: Check messages from the SQL Server. Invalid column name ' ... '."

The most probable cause of this error message is that your Data Mart has not been upgraded to release 6.5 or later. To use 6.5 (or later) CCPulse+ templates, you must upgrade all Reporting components to release 6.5 or later.

For more information, refer to the Reporting chapters of the *Genesys Migration Guide* or the "Configuring Genesys Solution Reporting" chapter of the *Reporting 7.5 Deployment Guide*.

Agents Cannot Monitor Agent Activity

Reinstall CCPulse+. At the Question dialog box, shown in Figure 18, click Yes to enable operators to monitor agent statistics.

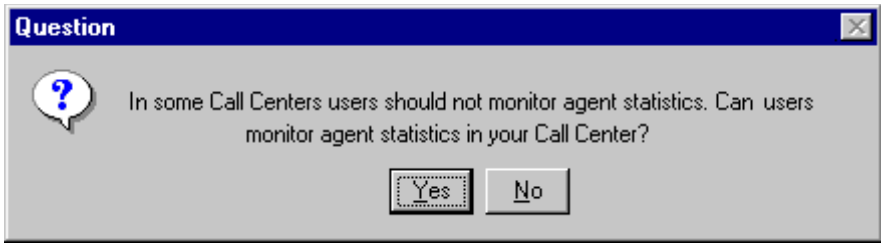

Figure 18: Allowing Users to Monitor Agent Statistics

Refer to "Installing Real-Time Reporting" in the *Reporting 7.5 Deployment Guide* for more information.

Sudden Stat Server Disconnection

Aberrant behavior may result when these events in CCPulse+ occur simultaneously:

- Stat Server is suddenly disconnected.
- Connection to the backup Stat Server cannot be established.
- The current workspace has been changed.

If the StatServer disconnected error message appears, click OK. When the Would you like to save your workspace? message appears, click one of the following:

- Yes, to save your current workspace but not your data.
- No, to clear both workspace and data.
- Cancel, to save the data but not the workspace.

Stat Server Missing from New Workspace

You must specify a Stat Server to which CCPulse+ is to connect. In Configuration Manager, check the `Connections` tab on the `Properties` dialog box of your CCPulse+ `Application` object to ensure the correct Stat Server is listed.

No Historical Associations Error Messages

If you are creating a historical view, and you receive the error message shown in Figure 19, verify that you have entered an associated historical statistic for each applicable real-time statistic you are using.

Figure 19: No Data Available Error Message

If you are trying to create a historical view, and you receive the error message in Figure 20, confirm that you have correctly configured your Data Mart database access point in the CCPulse+ `Application` object.

Figure 20: No Historical Associations Error Message

Cannot Recover Workspace

Under the very rare circumstance when your workspace becomes corrupt, CCPulse+ will not enable you to recover it. Instead, if you try to open the workspace, CCPulse+ displays the following error message:

```
Workspace [name] corrupted! It cannot be recovered. Please delete.
```

A workspace can become corrupt, for instance, if it was previously only partially saved as is the case when the related CCPulse+ process is killed

during the save operation or given file system problems. For this reason, Genesys recommends that you periodically back up your storage (*.stg) files.

Can’t View Multi-Tenant Data

The tenant to which a user belongs and the whether the user can view environment-level data is determined by how the user was originally defined to Configuration Server. CCPulse+ will display data for only those tenants that are monitored by the specific Stat Server application to which CCPulse+ is connected.

Supplements

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

Reporting

- *Reporting 7.5 CCPulse+ Help*, which describes how to navigate the CCPulse+ interface, operate the wizards, create templates and views, and associate actions to specific thresholds that you might set.
- The *Reporting 7.6 Deployment Guide*, which will help you configure, install, start, stop, and uninstall your Reporting environment. This guide also provides instructions for setting up CCPulse+ administrators and users.
- The *Framework Stat Server User's Guide*, which will help you understand statistics configuration.
- The *Reporting 7.6 Reference Manual*, which provides general information about performance measurements, how Reporting behaves during time shifts, and how to set up custom reports for skills-based and partial-period reporting.
- The *Reporting 7.6 Master Index*, which will help you find where other related topics are documented.

Genesys

- *Genesys Technical Publications Glossary*, which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.

- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

Information about supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys Supported Operating Environment Reference Manual*
- *Genesys Supported Media Interfaces Reference Manual*

Consult these additional resources as necessary:

- *Genesys Interoperability Guide*, which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and G*plus* Adapters Interoperability.
- *Genesys Database Sizing Estimator Worksheets*, which provides a range of expected database sizes for various Genesys products.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the system level documents by release tab in the Knowledge Base Browse Documents Section.

Genesys product documentation is available on the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

75rt_ad_ccpulse+_08-2012_v7.5.102.00

You will need this number when you are talking with Genesys Technical Support about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 17 describes and illustrates the type conventions that are used in this document.

Table 17: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 88).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .

Table 17: Type Styles (Continued)

Type Style	Used For	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W